

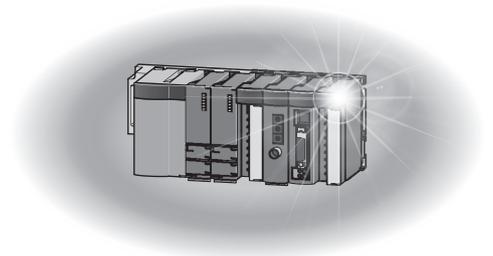


三菱電機 **汎用** シーケンサ

MELSEC **Q** series

Qn(H)/QnPH/QnPRHCPU ユーザズマニユアル (機能解説・プログラム基礎編)

- Q00JCPU
- Q00CPU
- Q01CPU
- Q02CPU
- Q02HCPU
- Q06HCPU
- Q12HCPU
- Q25HCPU
- Q02PHCPU
- Q06PHCPU
- Q12PHCPU
- Q25PHCPU
- Q12PRHCPU
- Q25PRHCPU



●安全上のご注意●

(ご使用前に必ずお読みください)

本製品のご使用に際しては、本マニュアルおよび本マニュアルで紹介している関連マニュアルをよくお読みいただくと共に、安全に対して十分に注意を払って、正しい取扱いをしていただくようお願いいたします。

この「安全上のご注意」では、安全注意事項のランクを「警告」, 「注意」として区分してあります。



取扱いを誤った場合に、危険な状況が起こりえて、死亡または重傷を受ける可能性が想定される場合。



取扱いを誤った場合に、危険な状況が起こりえて、中程度の傷害や軽傷を受ける可能性が想定される場合および物的損害だけの発生が想定される場合。

なお、注意に記載した事項でも、状況によっては重大な結果に結びつく可能性があります。

いずれも重要な内容を記載していますので必ず守ってください。

本マニュアルは必要なときに読めるよう大切に保管すると共に、必ず最終ユーザまでお届けいただくようお願いいたします。

【設計上の注意事項】

警告

- 外部電源の異常やシーケンサ本体の故障時でも、システム全体が安全側に働くようにシーケンサの外部で安全回路を設けてください。誤出力、誤動作により、事故の恐れがあります。
 - (1) 非常停止回路、保護回路、正転／逆転などの相反する動作のインタロック回路、位置決めの上限／下限など機械の破損防止のインタロック回路などは、シーケンサの外部で回路構成してください。
 - (2) シーケンサは次の異常状態を検出すると演算を停止し、出力は下記の状態になります。

異常状態	Qシリーズのユニット	AnS/Aシリーズのユニット
電源ユニットの過電流保護装置または過電圧保護装置が働いたとき	出力OFF	出力OFF
シーケンサCPUでウォッチドッグタイマエラーなど自己診断機能で異常を検出したとき	パラメータ設定により全出力を保持、またはOFF	出力OFF

また、CPU ユニットで検出できない入出力制御部分などの異常時は、全出力が ON することがあります。このとき、機械の動作が安全側に働くよう、シーケンサの外部でフェールセーフ回路を構成したり、安全機構を設けたりしてください。フェールセーフ回路例については、QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）の“フェールセーフ回路の考え方”を参照してください。

- (3) 出力ユニットのリレーやトランジスタなどの故障によっては、出力が ON の状態を保持したり、OFF の状態を保持することがあります。重大な事故につながるような出力信号については、外部で監視する回路を設けてください。

【設計上の注意事項】

警告

- 出力ユニットにおいて、定格以上の負荷電流または負荷短絡などによる過電流が長時間継続して流れた場合、発煙・発火の恐れがありますので、外部にヒューズなどの安全回路を設けてください。
- シーケンサ本体の電源立上げ後に、外部供給電源を投入するように回路を構成してください。外部供給電源を先に立ち上げると、誤出力、誤動作により事故の恐れがあります。
- ネットワークが交信異常になったときの各局の動作状態については、そのネットワークに関連するマニュアルを参照してください。誤出力、誤動作により事故の恐れがあります。
- CPU ユニットに周辺機器を接続、またはインテリジェント機能ユニットにパソコンなどを接続して、運転中のシーケンサのデータを変更するときは、常時システム全体が安全側に働くように、シーケンスプログラム上でインタロック回路を構成してください。また、プログラム変更、運転状態変更を行うときは、関連するマニュアルを熟読し、十分に安全を確認してから行ってください。特に外部機器から遠隔地のシーケンサに対する上記制御では、データ交信異常によりシーケンサ側のトラブルに即対応できない場合もあります。シーケンスプログラム上でインタロック回路を構成すると共に、データ交信異常が発生時のシステムとしての処置方法などを外部機器と CPU ユニット間で取り決めてください。

【設計上の注意事項】

注意

- 制御線や通信ケーブルは、主回路や動力線などと束線したり、近接したりしないでください。100mm 以上を目安として離してください。ノイズにより、誤動作の原因になります。
- 出力ユニットでランプ負荷、ヒータ、ソレノイドバルブなどを制御するとき、出力の OFF → ON 時に大きな電流（通常の 10 倍程度）が流れる場合がありますので、定格電流に余裕のあるユニットへの変更などの対策を行ってください。
- CPU ユニットの電源 OFF → ON またはリセット時、CPU ユニットが RUN 状態になるまでの時間が、システム構成、パラメータ設定、プログラム容量などにより変動します。RUN 状態になるまでの時間が変動しても、システム全体が安全側に働くように設計してください。

【取付け上の注意事項】

⚠ 注意

- シーケンサは、QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）に記載の一般仕様の環境で使用してください。
一般仕様の範囲以外の環境で使用すると、感電、火災、誤動作、製品の損傷あるいは劣化の原因になります。
- ユニット下部のユニット装着用レバーを押さえながら、ユニット固定用突起をベースユニットの固定穴に確実に挿入し、ユニット固定穴を支点として装着してください。
ユニットが正しく装着されていないと、誤動作、故障、落下の原因になります。
振動の多い環境で使用する場合は、ユニットをネジで締め付けください。
ネジの締め付けは、規定トルク範囲で行ってください。
ネジの締め付けがゆるいと、落下、短絡、誤動作の原因になります。
ネジを締め過ぎると、ネジやユニットの破損による落下、短絡、誤動作の原因になります。
- 増設ケーブルは、ベースユニットの増設ケーブル用コネクタに確実に装着してください。
装着後に、浮上りがないかチェックしてください。
接触不良により、誤入力、誤出力の原因になります。
- メモリカードは、メモリカード装着スロットに押し付けて確実に装着してください。
装着後に、浮上りがないかチェックしてください。
接触不良により、誤動作の原因になります。
- ユニットの着脱は、必ずシステムで使用している外部供給電源を全相遮断してから行ってください。
全相遮断しないと製品の損傷の恐れがあります。
オンラインユニット交換に対応した CPU ユニットを使用したシステムおよび MELSECNET/H リモート I/O 局では、オンライン中（通電中）でのユニット交換が可能です。
ただし、オンライン中（通電中）でのユニット交換が可能なユニットには制限があり、ユニットごとに交換手順が決められています。
詳細については、QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）、および該当ユニットのマニュアルに記載されているオンラインユニット交換の項を参照してください。
- ユニットの導電部分や電子部品には直接触らないでください。
ユニットの誤動作、故障の原因になります。
- モーション CPU ユニット、モーションユニットを使用するときは、電源を投入する前にユニットの組合せが正しいか必ず確認してください。
誤った組合せで使用した場合、製品が損傷する恐れがあります。
詳細については、モーション CPU ユニットのユーザーズマニュアルを参照してください。

【配線上の注意事項】



- 取付け，配線作業などは，必ずシステムで使用している外部供給電源を全相遮断してから行ってください。
全相遮断しないと，感電あるいは製品の損傷の恐れがあります。
- 配線作業後，通電，運転を行う場合は，必ず製品に付属の端子カバーを取り付けてください。
端子カバーを取り付けないと，感電の恐れがあります。

【配線上の注意事項】

注意

- FG 端子および LG 端子は、シーケンサ専用の D 種接地（第三種接地）以上で必ず接地を行ってください。
感電、誤動作の恐れがあります。
- 圧着端子は適合圧着端子を使用し、規定のトルクで締め付けてください。
先開形圧着端子を使用すると、端子ネジがゆるんだ場合に脱落し、故障の原因になります。
- ユニットへの配線は、製品の定格電圧および端子配列を確認した上で正しく行ってください。
定格と異なった電圧の入力や、電源を接続、誤配線をすると、火災、故障の原因になります。
- 外部接続用コネクタは、メーカー指定の工具で圧着、圧接または正しくハンダ付けしてください。
接続が不完全になっていると、短絡、火災、誤動作の原因になります。
- 制御線や通信ケーブルは、主回路や動力線と束線したり、近接したりしないでください。ノイズにより、誤動作の原因になります。
- 端子ネジの締め付けは、規定トルク範囲で行ってください。
端子ネジの締め付けがゆるいと、短絡、火災、誤動作の原因になります。
端子ネジを締め過ぎると、ネジやユニットの破損による落下、短絡、誤動作の原因になります。
- ユニット内に、切粉や配線クズなどの異物が入らないように注意してください。
火災、故障、誤動作の原因になります。
- ユニットは、配線時にユニット内へ配線クズなどの異物が混入するのを防止するため、ユニット上部に混入防止ラベルを貼り付けています。
配線作業中は、本ラベルをはがさないでください。
システム運転時は、放熱のために本ラベルを必ずはがしてください。
- 当社のシーケンサは、制御盤内に設置して使用してください。
制御盤内に設置されたシーケンサ電源ユニットへの主電源配線に関しては、中継端子台を介して行ってください。
また、電源ユニットの交換と配線作業は、感電保護に対して、十分に教育を受けたメンテナンス作業が行ってください。
配線方法は、QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）を参照してください。

【立上げ・保守時の注意事項】

警告

- 通電中に端子に触れないでください。
感電または誤動作の原因になります。
- バッテリコネクタは正しく接続してください。
バッテリーに充電，分解，加熱，火中投入，ショート，ハンダ付けなどを行わないでください。
バッテリーの取扱いを誤ると，発熱，破裂，発火などにより，ケガ，火災の恐れがあります。
- 清掃，端子ネジ，コネクタ取付けネジ，ユニット固定ネジの増し締めは，必ずシステムで使用している外部供給電源を全相遮断してから行ってください。全相遮断しないと，感電，ユニットの故障や誤動作の恐れがあります。ネジの締付けがゆるいと，落下，短絡，誤動作の原因になります。
ネジを締め過ぎると，ネジやユニットの破損による落下，短絡，誤動作の原因になります。

【立上げ・保守時の注意事項】

⚠ 注意

- 運転中の CPU ユニットに周辺機器を接続して行うオンライン操作（特にプログラム変更、強制出力、運転状態の変更）は、マニュアルを熟読し、十分に安全を確認してから行ってください。操作ミスにより機械の破損や事故の原因になります。
- ユニットの分解、改造はしないでください。故障、誤動作、ケガ、火災の原因になります。
- 携帯電話や PHS などの無線通信機器は、シーケンサの全方向から 25cm 以上離して使用するようになしてください。誤動作の原因になります。
- ユニットの着脱は、必ずシステムで使用している外部供給電源を全相遮断してから行ってください。全相遮断しないと、ユニットの故障や誤動作の原因になります。オンラインユニット交換に対応した CPU ユニットを使用したシステムおよび MELSECNET/H リモート I/O 局は、オンライン中（通電中）でのユニット交換が可能です。ただし、オンライン中（通電中）でのユニット交換が可能なユニットには制限があり、ユニットごとに交換手順が決められています。詳細については、QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）、およびオンラインユニット交換に対応したユニットのマニュアルのオンラインユニット交換の項を参照してください。
- 下記の着脱は、製品使用後、50 回以内（JIS B 3502、IEC 61131-2 に準拠）としてください。
 - ・ユニットとベースユニット
 - ・ユニットと端子台なお、50 回を超えた場合は、誤動作の原因となる恐れがあります。
- ユニットに装着するバッテリーには、落下・衝撃を加えないでください。落下・衝撃によりバッテリーが破損し、バッテリー液の液漏れをバッテリー内部で発生している恐れがあります。落下・衝撃を加えたバッテリーは使用せずに廃棄してください。
- ユニットに触れる前には、必ず接地された金属などに触れて、人体などに帯電している静電気を放電してください。静電気を放電しないと、ユニットの故障や誤動作の原因になります。

【廃棄時の注意事項】

注意

- 製品を廃棄するときは、産業廃棄物として扱ってください。
バッテリーを廃棄する際には各地域にて定められている法令に従い分別を行ってください。
(EU 加盟国内でのバッテリー規制の詳細については QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編) を参照してください。)

【輸送時の注意事項】

注意

- リチウムを含有しているバッテリーの輸送時には、輸送規制に従った取扱いが必要となります。
(規制対象機種の詳細については QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編) を参照してください。)

●製品の適用について●

- (1) 当社シーケンサをご使用いただくにあたりましては、万一シーケンサに故障・不具合などが発生した場合でも重大な事故にいたらない用途であること、および故障・不具合発生時にはバックアップやフェールセーフ機能が機器外部で系統的に実施されていることをご使用の条件とさせていただきます。
- (2) 当社シーケンサは、一般工業などへの用途を対象とした汎用品として設計・製作されています。したがって、以下のような機器・システムなどの特殊用途へのご使用については、当社シーケンサの適用を除外させていただきます。万一使用された場合は当社として当社シーケンサの品質、性能、安全に関する一切の責任（債務不履行責任、瑕疵担保責任、品質保証責任、不法行為責任、製造物責任を含むがそれらに限定されない）を負わないものとさせていただきます。
- ・各電力会社殿の原子力発電所およびその他発電所向けなどの公共への影響が大きい用途
 - ・鉄道各社殿および官公庁殿など、特別な品質保証体制の構築を当社にご要求になる用途
 - ・航空宇宙、医療、鉄道、燃焼・燃料装置、乗用移動体、有人搬送装置、娯楽機械、安全機械など生命、身体、財産に大きな影響が予測される用途
- ただし、上記の用途であっても、具体的に用途を限定すること、特別な品質（一般仕様を超えた品質等）をご要求されないこと等を条件に、当社の判断にて当社シーケンサの適用可とする場合もございますので、詳細につきましては当社窓口へご相談ください。

改訂履歴

※ 取扱説明書番号は、本説明書の裏表紙の左下に記載してあります。

印刷日付	※ 取扱説明書番号	改訂内容
2008年10月	SH(名)-080803-A	初版印刷
2009年2月	SH(名)-080803-B	<p>一部修正</p> <p>安全上のご注意, 本マニュアルの見方, 本マニュアルで使用する総称/略称, 1.2節, 1.3節, 1.4節, 2.2.2項, 2.2.3項, 2.3節, 2.3.3項, 2.3.4項, 2.3.5項, 2.3.6項, 2.4.1項, 2.4.4項, 3.3節, 3.5節, 3.6節, 3.7節, 3.8節, 3.8.1項, 3.8.2項, 4.1.1項, 4.1.2項, 4.2節, 4.2.2項, 4.2.3項, 5.1.1項, 5.1.2項, 5.2.1項, 5.2.2項, 5.2.4項, 5.2.5項, 5.2.6項, 5.2.8項, 5.2.10項, 5.3節, 5.4.2項, 5.4.3項, 5.4.4項, 6.1節, 6.2節, 6.3節, 6.4節, 6.5節, 6.6節, 6.6.1項, 6.6.2項, 6.6.3項, 6.6.4項, 6.6.5項, 6.7節, 6.8節, 6.9節, 6.10節, 6.11.3項, 6.12.1項, 6.12.2項, 6.12.3項, 6.13.3項, 6.14節, 6.15.2項, 6.17節, 6.19節, 6.20節, 6.21.1項, 6.21.2項, 6.22.2項, 6.22.4項, 6.27節, 7.1.4項, 7.1.5項, 7.2節, 第8章, 8.1節, 8.1.1項, 8.1.2項, 8.2節, 8.3節, 8.4節, 9.1節, 9.2節, 9.2.5項, 9.2.10項, 9.2.11項, 9.2.12項, 9.2.13項, 9.3.3項, 9.4節, 9.5.2項, 9.6.2項, 9.7.4項, 9.9節, 9.10節, 9.13.2項, 10.1.1項, 10.1.2項, 10.1.3項, 11.1.1項, 11.2.1項, 11.2.5項, 付1, 付2.1, 付3</p>
2010年2月	SH(名)-080803-C	<p>一部修正</p> <p>安全上のご注意, 2.2.3項, 4.2.2項, 9.2.10項, 9.6.1項, 9.7.2項, 9.7.4項, 12.1節, 12.2節</p> <p>追加</p> <p>製品の適用について</p>
2010年5月	SH(名)-080803-D	<p>一部修正</p> <p>安全上のご注意, 4.2.2項, 5.1.1項, 5.2.1項, 6.5節, 6.11.2項</p>
2011年5月	SH(名)-080803-E	<p>一部修正</p> <p>3.8.2項, 5.1.1項, 5.1.2項, 5.2.1項, 5.2.8項, 9.2節, 9.2.4項, 10.1.3項</p> <p>削除</p> <p>第12章</p>
2011年9月	SH(名)-080803-F	<p>一部修正</p> <p>安全上のご注意, はじめに, 本マニュアルで使用する総称/略称, 2.4.5項, 5.2.6項, 6.24節, 9.2節, 9.2.10項, 9.12.4項</p>
2011年10月	SH(名)-080803-G	<p>一部修正</p> <p>本マニュアルで使用する総称/略称, 1.5.2項, 3.8節, 4.2.2項, 6.10節, 6.22.2項, 6.22.4項, 10.1.2項</p>
2012年5月	SH(名)-080803-H	<p>一部修正</p> <p>5.2.8項, 6.2節, 6.12.3項, 8.1.1項, 8.1.2項, 9.1節, 9.2.13項, 9.3.2項, 9.3.3項, 9.7節, 9.7.4項, 11.1.2項, 11.2.2項, 付1</p>
2013年7月	SH(名)-080803-I	<p>一部修正</p> <p>4.2.1項, 5.2.1項, 5.2.10項, 6.12.3項, 9.1節, 10.1.3項</p>
2014年1月	SH(名)-080803-J	<p>一部修正</p> <p>本マニュアルで使用する総称/略称, 5.2.1項, 5.2.5項, 9.7.2項, 9.7.6項, 付2.2, 付2.3</p>
2014年4月	SH(名)-080803-K	<p>一部修正</p> <p>本マニュアルで使用する総称/略称, 1.5.2項, 3.8節, 4.2.2項, 6.22.2項, 6.22.4項, 10.1.2項</p>

印刷日付	※ 取扱説明書番号	改訂内容
2015年10月	SH(名)-080803-L	一部修正 1.2節, 5.2.1項, 5.2.7項, 5.2.8項
2017年6月	SH(名)-080803-M	一部修正 安全上のご注意, 本マニュアルで使用する総称/略称, 6.2節, 6.13.1項, 6.13.3項, 6.17節, 6.24節, 9.13.2項
2018年10月	SH(名)-080803-N	一部修正 マニュアル体系, 本マニュアルで使用する総称/略称, 5.2.1項, 5.4.4項
2019年4月	SH(名)-080803-O	一部修正 本マニュアルで使用する総称/略称

本書によって、工業所有権その他の権利の実施に対する保証、または実施権を許諾するものではありません。また本書の掲載内容の使用により起因する工業所有権上の諸問題については、当社は一切その責任を負うことができません。

© 2008 MITSUBISHI ELECTRIC CORPORATION

はじめに

本マニュアル「Qn(H)/QnPH/QnPRHCPU ユーザーズマニュアル（機能解説・プログラム基礎編）」は、Q シリーズ CPU ユニットをご使用いただくときに必要となる CPU ユニットのメモリマップ、機能、プログラム、入出力番号の割付方法、デバイスなどについてご理解いただくためのマニュアルです。

ご使用前に本マニュアルおよび関連マニュアルをよくお読みいただき、Q シリーズシーケンサの機能・性能を十分ご理解のうえ、正しくご使用くださるようお願い致します。

本マニュアルで紹介するプログラム例を実際のシステムへ流用する場合は、対象システムにおける制御に問題がないことを十分検証ください。

■対象 CPU ユニット

CPU ユニット	形名
ベーシックモデル QCPU	Q00JCPU, Q00CPU, Q01CPU
ハイパフォーマンスモデル QCPU	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
プロセス CPU	Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU
二重化 CPU	Q12PRHCPU, Q25PRHCPU

備考

本マニュアルでは、電源ユニット、ベースユニット、増設ケーブル、メモリカード、バッテリーの仕様などについては説明していません。

下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）

マルチ CPU システムについては、下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル（マルチ CPU システム編）

二重化システムについては、下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル（二重化システム編）

目次

安全上のご注意	A - 1
製品の適用について	A - 10
改訂履歴	A - 11
はじめに	A - 13
マニュアル体系	A - 20
本マニュアルの見方	A - 22
本マニュアルで使用する総称/略称	A - 24

第1章 概要

1-1 ~ 1-20

1.1 CPU ユニットの処理順序	1 - 1
1.2 プログラムの格納と実行	1 - 2
1.3 プログラムの構造化	1 - 3
1.4 プログラミングに便利なデバイス、命令	1 - 6
1.5 特長	1 - 11
1.5.1 ベーシックモデル QCPU の特長	1 - 11
1.5.2 ハイパフォーマンスモデル QCPU の特長	1 - 13
1.5.3 プロセス CPU の特長	1 - 15
1.5.4 二重化 CPU の特長	1 - 17
1.6 シリアル No. と機能バージョンの確認方法	1 - 19

第2章 シーケンスプログラム

2-1 ~ 2-51

2.1 シーケンスプログラムの概要	2 - 1
2.2 シーケンスプログラムの構成	2 - 3
2.2.1 メインルーチンプログラム	2 - 4
2.2.2 サブルーチンプログラム	2 - 5
2.2.3 割込みプログラム	2 - 6
2.3 複数のプログラムに分割した場合の設定	2 - 14
2.3.1 初期実行タイププログラム	2 - 19
2.3.2 スキャン実行タイププログラム	2 - 22
2.3.3 低速実行タイププログラム	2 - 23
2.3.4 待機タイププログラム	2 - 31
2.3.5 定周期実行タイププログラム	2 - 36
2.3.6 実行タイプの切換え例	2 - 40
2.4 シーケンスプログラムで使用できるデータ	2 - 42
2.4.1 BIN (2進数: Binary Code)	2 - 45
2.4.2 HEX (16進数: Hexadecimal)	2 - 46
2.4.3 BCD (2進化10進数: Binary Coded Decimal)	2 - 47
2.4.4 実数 (浮動小数点データ)	2 - 48
2.4.5 文字列データ	2 - 51

第3章 CPU ユニットの動作

3-1 ~ 3-15

3.1 イニシャル処理	3 - 1
3.2 I/O リフレッシュ (入出力ユニットのリフレッシュ処理)	3 - 2

3.3	プログラムの演算	3 - 2
3.4	END 処理	3 - 3
3.5	RUN 状態, STOP 状態, PAUSE 状態の演算処理	3 - 4
3.6	瞬停時の演算処理	3 - 6
3.7	データのクリア処理	3 - 7
3.8	入出力処理と応答遅れ	3 - 9
3.8.1	リフレッシュ方式	3 - 10
3.8.2	ダイレクト方式	3 - 13

第 4 章	ベースユニット／入出力番号の割付け	4-1 ~ 4-17
-------	-------------------	------------

4.1	ベースユニットの割付け	4 - 1
4.1.1	ベースモードとは	4 - 1
4.1.2	ベースユニットの割付け設定	4 - 2
4.2	入出力番号の割付け	4 - 4
4.2.1	入出力番号の割付けの考え方	4 - 5
4.2.2	入出力番号の設定	4 - 8
4.2.3	入出力番号の割付け例	4 - 14
4.2.4	入出力番号の確認	4 - 17

第 5 章	CPU ユニットで取り扱うメモリとファイルについて	5-1 ~ 5-51
-------	---------------------------	------------

5.1	ベーシックモデル QCPU で取り扱うメモリ	5 - 1
5.1.1	メモリ構成と格納できるデータ	5 - 1
5.1.2	プログラムメモリについて	5 - 3
5.1.3	標準 ROM について	5 - 6
5.1.4	標準 RAM について	5 - 7
5.1.5	標準 ROM のプログラムの実行 (ブート運転) と書込み	5 - 9
5.2	ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU で取り扱うメモリ	5 - 13
5.2.1	メモリ構成と格納できるデータ	5 - 13
5.2.2	プログラムメモリについて	5 - 16
5.2.3	標準 ROM について	5 - 19
5.2.4	標準 RAM について	5 - 20
5.2.5	メモリカードについて	5 - 22
5.2.6	GX Developer による標準 ROM / Flash カードへの書込み	5 - 26
5.2.7	メモリカード→標準 ROM 全データ自動書込み	5 - 31
5.2.8	標準 ROM / メモリカードのプログラムの実行 (ブート運転)	5 - 34
5.2.9	書き込んだファイルの詳細	5 - 39
5.2.10	有効にするパラメータの指定 (パラメータ有効ドライブ設定)	5 - 40
5.3	プログラムファイルの構成	5 - 41
5.4	GX Developer によるファイル操作および取扱い時の注意事項	5 - 43
5.4.1	ファイルの操作	5 - 43
5.4.2	ファイルの取扱い時の注意事項	5 - 44
5.4.3	ファイルのメモリ容量	5 - 45
5.4.4	ファイルのサイズ単位	5 - 48

6.1	機能一覧	6 - 1
6.2	コンスタントスキャン	6 - 4
6.3	ラッチ機能	6 - 8
6.4	STOP 状態→RUN 状態にしたときの出力 (Y) 状態の設定	6 - 11
6.5	時計機能	6 - 14
6.6	リモート操作	6 - 19
6.6.1	リモート RUN/STOP	6 - 19
6.6.2	リモート PAUSE	6 - 22
6.6.3	リモート RESET (リモートリセット)	6 - 25
6.6.4	リモートラッチクリア	6 - 28
6.6.5	リモート操作と CPU ユニットの RUN/STOP 状態との関係	6 - 30
6.7	Q シリーズ対応ユニットの入力応答時間選択 (I/O 応答時間)	6 - 31
6.8	エラー時出力モード設定	6 - 33
6.9	ハードウェアエラー時の CPU 動作モード設定	6 - 34
6.10	インテリジェント機能ユニットのスイッチ設定	6 - 35
6.11	モニタ機能	6 - 37
6.11.1	モニタ条件の設定	6 - 38
6.11.2	ローカルデバイスのモニタ・テスト	6 - 44
6.11.3	外部入出力の強制 ON/OFF	6 - 47
6.12	CPU ユニットが RUN 中のプログラム書込み	6 - 53
6.12.1	回路モードでの RUN 中書込み	6 - 53
6.12.2	ファイルの RUN 中書込み	6 - 57
6.12.3	RUN 中書込み時の注意事項	6 - 59
6.13	実行時間計測	6 - 64
6.13.1	プログラム一覧モニタ	6 - 64
6.13.2	割り込みプログラム一覧モニタ	6 - 68
6.13.3	スキャンタイム測定	6 - 69
6.14	サンプリングトレース機能	6 - 73
6.15	複数の GX Developer からのデバッグ機能	6 - 86
6.15.1	複数の GX Developer から同時にモニタする機能	6 - 86
6.15.2	複数の GX Developer から RUN 中書込みする機能	6 - 88
6.16	ウォッチドッグタイマ (WDT)	6 - 90
6.17	自己診断機能	6 - 92
6.17.1	エラー発生による割り込み	6 - 99
6.17.2	エラー発生による LED 表示	6 - 100
6.17.3	エラーの解除	6 - 101
6.18	エラー履歴	6 - 102
6.18.1	ベーシックモデル QCPU	6 - 102
6.18.2	ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU	6 - 103
6.19	システムプロテクト	6 - 104
6.19.1	パスワード登録	6 - 105
6.19.2	リモートパスワード	6 - 107
6.20	GX Developer による CPU ユニットのシステム表示	6 - 112

6.21 LED の表示	6 - 117
6.21.1 LED の消灯方法	6 - 118
6.21.2 LED 表示の優先順位	6 - 119
6.22 高速割込み機能	6 - 122
6.22.1 高速割込みプログラム実行機能	6 - 123
6.22.2 高速 I/O リフレッシュ/高速バッファ転送機能	6 - 124
6.22.3 処理時間	6 - 126
6.22.4 制約事項	6 - 127
6.23 インテリジェント機能ユニットからの割込み	6 - 130
6.24 シリアルコミュニケーション機能	6 - 131
6.25 サービス処理	6 - 140
6.25.1 ユニットサービス間隔読出し	6 - 140
6.26 デバイス初期値	6 - 141
6.27 メモリチェック機能	6 - 146

第 7 章 インテリジェント機能ユニットとの交信 7-1 ~ 7-10

7.1 CPU ユニットからインテリジェント機能ユニットとの交信	7 - 1
7.1.1 GX Configurator による初期設定, 自動リフレッシュ設定	7 - 2
7.1.2 デバイス初期値による初期設定	7 - 5
7.1.3 FROM/TO 命令による交信	7 - 5
7.1.4 インテリジェント機能ユニットデバイスによる交信	7 - 6
7.1.5 インテリジェント機能ユニット専用命令による交信	7 - 8
7.2 AnS/A シリーズ対応の特殊機能ユニットにアクセスする場合	7 - 10

第 8 章 パラメータ 8-1 ~ 8-34

8.1 PC パラメータ	8 - 2
8.1.1 ベーシックモデル QCPU	8 - 2
8.1.2 ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU	8 - 14
8.2 二重化パラメータ	8 - 27
8.3 ネットワークパラメータ	8 - 29
8.4 リモートパスワード	8 - 34

第 9 章 デバイスの説明 9-1 ~ 9-108

9.1 デバイス一覧	9 - 1
9.2 内部ユーザデバイス	9 - 5
9.2.1 入力 (X)	9 - 8
9.2.2 出力 (Y)	9 - 10
9.2.3 内部リレー (M)	9 - 11
9.2.4 ラッチリレー (L)	9 - 12
9.2.5 アナンシェータ (F)	9 - 14
9.2.6 エッジリレー (V)	9 - 19
9.2.7 リンクリレー (B)	9 - 20

9.2.8	リンク特殊リレー (SB)	9 - 22
9.2.9	ステップリレー (S)	9 - 23
9.2.10	タイマ (T)	9 - 24
9.2.11	カウンタ (C)	9 - 33
9.2.12	データレジスタ (D)	9 - 39
9.2.13	リンクレジスタ (W)	9 - 40
9.2.14	リンク特殊レジスタ (SW)	9 - 42
9.3	内部システムデバイス	9 - 43
9.3.1	ファンクションデバイス (FX, FY, FD)	9 - 43
9.3.2	特殊リレー (SM)	9 - 46
9.3.3	特殊レジスタ (SD)	9 - 47
9.4	リンクダイレクトデバイス	9 - 48
9.5	ユニットアクセスデバイス	9 - 52
9.5.1	インテリジェント機能ユニットデバイス	9 - 52
9.5.2	マルチ CPU 間共有デバイス	9 - 55
9.6	インデックスレジスタ (Z)	9 - 56
9.6.1	インデックスレジスタ (Z)	9 - 56
9.6.2	スキャン実行タイププログラムと低速実行タイププログラムの切換え時の処理	9 - 58
9.6.3	スキャン実行タイプ/低速実行タイププログラム→割込み/定周期実行タイプ プログラムへの切換え時の処理	9 - 59
9.7	ファイルレジスタ (R)	9 - 63
9.7.1	ファイルレジスタの格納場所	9 - 64
9.7.2	ファイルレジスタの容量	9 - 64
9.7.3	格納先メモリによるアクセス方法の違い	9 - 65
9.7.4	ファイルレジスタの登録手順	9 - 66
9.7.5	ファイルレジスタの指定方法	9 - 72
9.7.6	ファイルレジスタ使用時の注意事項	9 - 73
9.8	ネスティング (N)	9 - 76
9.9	ポインタ (P)	9 - 77
9.9.1	ローカルポインタ	9 - 78
9.9.2	共通ポインタ	9 - 80
9.10	割込みポインタ (I)	9 - 83
9.10.1	割込みポインタ番号と割込み要因の一覧	9 - 85
9.11	その他のデバイス	9 - 92
9.11.1	SFC ブロックデバイス (BL)	9 - 92
9.11.2	SFC 移行デバイス (TR)	9 - 92
9.11.3	ネットワーク No. 指定デバイス (J)	9 - 92
9.11.4	I/O No. 指定デバイス (U)	9 - 93
9.11.5	マクロ命令引数デバイス (VD)	9 - 94
9.12	定数	9 - 95
9.12.1	10 進定数 (K)	9 - 95
9.12.2	16 進定数 (H)	9 - 95
9.12.3	実数 (E)	9 - 96
9.12.4	文字列 (" ")	9 - 97
9.13	デバイスの便利な使い方	9 - 98
9.13.1	グローバルデバイス	9 - 98
9.13.2	ローカルデバイス	9 - 99

第 10 章 CPU ユニットの処理時間 10-1 ~ 10-22

10.1 スキャンタイム	10 - 1
10.1.1 スキャンタイムの構成	10 - 1
10.1.2 スキャンタイムに関連する要素の処理時間	10 - 6
10.1.3 スキャンタイムを延ばす要因	10 - 15
10.1.4 設定変更でスキャンタイムを短くできる要因	10 - 20

第 11 章 プログラムを CPU ユニットに書き込むまでの手順 11-1 ~ 11-15

11.1 ベーシックモデル QCPU	11 - 1
11.1.1 プログラムを作成する場合の検討事項	11 - 1
11.1.2 ハードウェアチェック	11 - 2
11.1.3 プログラムを書き込むまでの手順	11 - 4
11.1.4 ブート運転の手順	11 - 6
11.2 ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU	11 - 7
11.2.1 プログラムを作成する場合の検討事項	11 - 7
11.2.2 ハードウェアチェック	11 - 8
11.2.3 1本のプログラムを書き込むまでの手順	11 - 10
11.2.4 複数のプログラムを書き込むまでの手順	11 - 12
11.2.5 ブート運転の手順	11 - 15

付 録 付 -1 ~ 付 -13

付 1 パラメータ No. 一覧	付 - 1
付 2 バージョンアップによる機能の追加・変更	付 - 5
付 2.1 ベーシックモデル QCPU の機能アップ	付 - 5
付 2.2 ハイパフォーマンスモデル QCPU の機能アップ	付 - 8
付 2.3 プロセス CPU の機能アップ	付 - 10
付 2.4 二重化 CPU の機能アップ	付 - 11
付 3 デバイス点数割付けシート	付 - 12

索 引 索引 -1 ~ 索引 -5

マニュアル体系

基本的な仕様、機能、使用方法については基本マニュアルでご理解いただけます。
 他のマニュアルは、該当の CPU ユニットや機能を利用する場合にご利用ください。
 各マニュアルは、必要に応じて本表を参考にしてください。

番号	CPU ユニット
1)	ベーシックモデル QCPU
2)	ハイパフォーマンスモデル QCPU
3)	プロセス CPU
4)	二重化 CPU

●：基本マニュアル，○：該当 CPU ユニット／機能を利用する場合にご利用ください。

マニュアル名称 <マニュアル番号(形名コード) 標準価格>	記載内容	CPU ユニット			
		1)	2)	3)	4)
■ユーザーズマニュアル					
QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編) < SH-080472(13JP56) ¥4,000 >	CPU ユニット、電源ユニット、ベースユニット、増設ケーブル、メモリカードなどのハードウェア仕様と、システムの保守・点検、トラブルシューティング、エラーコードなど	●	●	●	●
Qn(H)/QnPH/QnPRHCPU ユーザーズマニュアル (機能解説・プログラム基礎編) < SH-080803(13JY95) ¥4,000 >	プログラム作成に必要な機能、プログラミング方法およびデバイスなど	●	●	●	●
QCPU ユーザーズマニュアル (マルチ CPU システム編) < SH-080475(13JP59) ¥3,000 >	マルチ CPU システムの構築に関する内容(システム構成、入出力番号、CPU ユニット間の通信、入出力ユニット/インテリジェント機能ユニットとの通信など)	○	○	○	
QnPRHCPU ユーザーズマニュアル (二重化システム編) < SH-080474(13JP58) ¥3,000 >	二重化システムの構築に関する内容(システム構成、機能、外部機器との通信、トラブルシューティングなど)				●
■プログラミングマニュアル					
MELSEC-Q/L プログラミングマニュアル(共通命令編) < SH-080804(13JC22) ¥4,000 >	シーケンス命令、基本命令および応用命令などの使用方法	●	●	●	●
MELSEC-Q/L/QnA プログラミングマニュアル(SFC 編) < SH-080023(13JC02) ¥3,000 >	MELSAP3 のシステム構成、性能仕様、機能、プログラミング、デバッグ、およびエラーコードなど	○	○	○	○
MELSEC-Q/L プログラミングマニュアル(MELSAP-L 編) < SH-080072(13JC03) ¥3,000 >	MELSAP-L 形式の SFC プログラムの作成に必要なプログラミング方法、仕様、機能など	○	○	○	○
MELSEC-Q/L プログラミングマニュアル(ストラクチャードテキスト編) < SH-080363(13JC11) ¥3,000 >	ストラクチャードテキスト言語のプログラミング方法	○	○	○	○
MELSEC-Q/L/QnA プログラミングマニュアル(PID 制御命令編) < SH-080022(13JC01) ¥1,000 >	PID 制御を行うための専用命令	○	○		○
MELSEC-Q プログラミング/構造化プログラミングマニュアル(プロセス制御命令編) < SH-080265(13JC09) ¥3,000 >	プロセス制御を行うための専用命令			○	○

その他の関連マニュアル

マニュアル名称	記載内容
MELSEC-Q CC-Link IE コントローラネットワークリファレンスマニュアル < SH-080649(13JD22) ¥4,000 >	CC-Link IE コントローラネットワークのコントローラネットワークの仕様、運転までの設定と手順、パラメータ設定、プログラミングおよびトラブルシューティング
Q 対応 MELSECNET/H ネットワークシステムリファレンスマニュアル (PC 間ネット編) < SH-080026(13JD04) ¥3,000 >	MELSECNET/H ネットワークシステムの PC 間ネットの仕様、運転までの設定と手順、パラメータ設定、プログラミング、トラブルシューティング
Q 対応 MELSECNET/H ネットワークシステムリファレンスマニュアル (リモート I/O ネット編) < SH-080123(13JD06) ¥3,000 >	MELSECNET/H ネットワークシステムのリモート I/O ネットの仕様、運転までの設定と手順、パラメータ設定、プログラミング、トラブルシューティング
Q 対応 Ethernet インタフェースユニット ユーザーズマニュアル (基本編) < SH-080004(13JQ36) ¥3,000 >	Ethernet ユニットの仕様、相手機器とのデータ交信手順、回線接続 (オープン/クローズ)、固定バッファ交信、ランダムアクセス用バッファ交信、トラブルシューティング
MELSEC-Q/L Ethernet インタフェースユニット ユーザーズマニュアル (応用編) < SH-080005(13JQ37) ¥3,000 >	Ethernet ユニットの電子メール機能、シーケンサ CPU の状態監視、MELSECNET/H、MELSECNET/10 を中継して交信する機能、データリンク用命令で交信する機能、ファイル転送 (FTP サーバ) を使用する場合など
MELSEC-Q CC-Link システムマスタ・ローカルユニットユーザーズマニュアル < SH-080395(13JP15) ¥4,000 >	QJ61BT11N のシステム構成、性能仕様、機能、取扱い、配線、およびトラブルシューティング
Q 対応シリアルコミュニケーションユニット ユーザーズマニュアル (基本編) < SH-080001(13JQ32) ¥3,000 >	シリアルコミュニケーションユニットを使用するための概要、適用システム構成、仕様、運転までの手順、相手機器との基本的なデータ交信方法、保守、点検、トラブルシューティング
MELSEC-Q/L シリアルコミュニケーションユニットユーザーズマニュアル (応用編) < SH-080002(13JQ33) ¥3,000 >	シリアルコミュニケーションユニットの特殊機能の仕様と使い方、特殊機能を使用するための設定、相手機器とのデータ交信方法
MELSEC コミュニケーションプロトコルリファレンスマニュアル < SH-080003(13JQ34) ¥4,000 >	シリアルコミュニケーションユニット / Ethernet ユニットを使用して CPU ユニットのデータ読出し、書込みなどを行う MC プロトコルの説明
GX Developer Version8 オペレーティングマニュアル < SH-080356(13JV69) ¥4,000 >	GX Developer でのプログラムの作成方法、プリントアウト方法など

本マニュアルの見方

注意事項の表示

本文中の「注●▲」は、ページ下の「注●▲」に対応しています。

参照先の表示

参照先や参照マニュアルは、のマークで記載しています。

章見出しの表示

ページ右側のインデックスで開いているページの章が一目で分かります。

第6章 機能

6.24 シリアルコミュニケーション機能 注 6.30

(1) シリアルコミュニケーション機能とは

CPUユニットのRS-232インタフェースとパソコン、表示器などをRS-232ケーブルで接続し、MCプロトコル*1により交信を行う機能です。シリアルコミュニケーション機能の仕様、機能、各種設定について説明します。

*1: MCプロトコルとは、MELSECコミュニケーションプロトコルの総称です。MELSECコミュニケーションプロトコルは、Qシリーズスタンダード（シリアルコミュニケーションユニット、Ethernetユニットなど）の交信手順に従い、相手機器からCPUユニットへアクセスするための通信方式の名前です。MCプロトコルについては、下記マニュアルを参照してください。

Q対応MELSECコミュニケーションプロトコルリファレンスマニュアル

図 6.91 パソコン、表示器との交信

Point

- パソコン、表示器などからシリアルコミュニケーション機能で交信できるのは、パソコン、表示器などと接続しているCPUユニットのみです。CC-Link IEコントローラネットワーク、MELSECNET/H、Ethernet、CC-Linkの他局との交信はできません。
- GX Developer、GX ConfiguratorとCPUユニットの接続には、シリアルコミュニケーション機能を使用しません。

注 6.30

Q00JCPU、ハイパフォーマンスモデルQCPU、プロセスCPU、二重化CPUでは、シリアルコミュニケーション機能は使用できません。

6 - 128

注意事項の説明

アイコンに対応した注意事項が記載されています。

節・項タイトルの表示

開いているページの節・項が一目で分かります。

※ このページは説明のために作成したもので、実際のページとは異なります。

アイコン				内 容
ベーシックモデル QCPU	ハイパフォーマンス モデル QCPU	プロセス CPU	二重化 CPU	
				アイコンは、記載している仕様の一部に注意事項があることを示します。

ほかに次の種類の説明があります。

Point 

そのページで説明した内容で、特に注意する事項や知っておきたい機能などを説明します。

備考

そのページで説明した内容に関連する参照先や、知っておくと便利な内容を説明します。

本マニュアルで使用する総称／略称

本マニュアルでは、特に明記する場合を除き、下記に示す総称および略称を使って説明します。

※ □は、複数の形名やバージョンなどを総称するときの変数部分を示しています。

(例) : Q33B, Q35B, Q38B, Q312B ⇒ Q3 □ B

総称／略称	総称／略称の内容
■ CPU ユニットのモデル名	
CPU ユニット	ベーシックモデル QCPU, ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU の総称
ベーシックモデル QCPU	Q00JCPU, Q00CPU, Q01CPU の総称
ハイパフォーマンスモデル QCPU	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU の総称
プロセス CPU	Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU の総称
二重化 CPU	Q12PRHCPU, Q25PRHCPU の総称
モーション CPU	Q172CPUN, Q173CPUN, Q172HCPU, Q173HCPU, Q172CPUN-T, Q173CPUN-T, Q172HCPU-T, Q173HCPU-T, Q172DCPU, Q173DCPU, Q172DCPU-S1, Q173DCPU-S1, Q172DSCPU, Q173DSCPU 形三菱電機モーショントローラの総称
パソコン CPU ユニット	PPC-CPU852(MS)-512 形株式会社コンテック製 MELSEC-Q シリーズ対応パソコン CPU ユニットの総称
C 言語コントローラユニット	Q06CCPU-V-H01, Q06CCPU-V, Q06CCPU-V-B, Q12DCCPU-V, Q24DHCCPUV 形 C 言語コントローラユニットの総称
■ CPU ユニットの形名	
QnHCPU	Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU の総称
Qn(H)CPU	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU の総称
QnPHCPU	Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU の総称
QnPRHCPU	Q12PRHCPU, Q25PRHCPU の総称
■ 二重化 CPU 関連	
制御系	二重化システムで制御, ネットワークの通信を行っているシステム
待機系	二重化システムでバックアップ用のシステム
A 系	トラッキングケーブルの A 系コネクタを接続したシステム
B 系	トラッキングケーブルの B 系コネクタを接続したシステム
■ ベースユニットのモデル名	
ベースユニット	基本ベースユニット, 増設ベースユニット, スリムタイプ基本ベースユニット, 電源二重化基本ベースユニット, 電源二重化増設ベースユニット, マルチ CPU 間高速基本ベースユニットの総称
基本ベースユニット	Q3 □ B, Q3 □ SB, Q3 □ RB, Q3 □ DB の総称
増設ベースユニット	Q5 □ B, Q6 □ B, Q6 □ RB, Q6 □ WRB, QA1S5 □ B, QA1S6 □ B, QA1S6ADP+A1S5 □ B/A1S6 □ B, QA6 □ B, QA6ADP+A5 □ B/A6 □ B の総称
スリムタイプ基本ベースユニット	Q3 □ SB の別称
電源二重化基本ベースユニット	Q3 □ RB の別称
電源二重化増設ベースユニット	Q6 □ RB の別称
二重化増設ベースユニット	Q6 □ WRB の別称
マルチ CPU 間高速基本ベースユニット	Q3 □ DB の別称
二重化ベースユニット	Q3 □ RB, Q6 □ RB, Q6 □ WRB の総称
電源二重化ベースユニット	Q3 □ RB, Q6 □ RB の総称
■ ベースユニットの形名	
Q3 □ B	Q33B, Q35B, Q38B, Q312B 形基本ベースユニットの総称
Q3 □ SB	Q32SB, Q33SB, Q35SB 形スリムタイプ基本ベースユニットの総称

総称／略称	総称／略称の内容
Q3□RB	Q38RB 形電源二重化システム用基本ベースユニットの別称
Q3□DB	Q35DB, Q38DB, Q312DB 形マルチ CPU 間高速基本ベースユニットの総称
Q5□B	Q52B, Q55B 形増設ベースユニットの総称
Q6□B	Q63B, Q65B, Q68B, Q612B 形増設ベースユニットの総称
Q6□RB	Q68RB 形電源二重化システム用増設ベースユニットの別称
Q6□WRB	Q65WRB 形 CPU・電源二重化システム用増設ベースユニットの別称
QA1S5□B	QA1S51B 形増設ベースユニットの別称
QA1S6□B	QA1S65B, QA1S68B 形増設ベースユニットの総称
QA6□B	QA65B,QA68B 形増設ベースユニットの総称
A5□B	A52B, A55B, A58B 形増設ベースユニットの総称
A6□B	A62B, A65B, A68B 形増設ベースユニットの総称
QA6ADP+A5□B/A6□B	QA6ADP を装着した A 大形増設ベースユニットの略称
QA1S6ADP+A1S5□B/A1S6□B	QA1S6ADP を装着した A 小形増設ベースユニットの略称
■ 電源ユニット	
電源ユニット	Q シリーズ電源ユニット, スリムタイプ電源ユニット, 二重化電源ユニットの総称
Q シリーズ電源ユニット	Q61P-A1, Q61P-A2, Q61P, Q61P-D, Q62P, Q63P, Q64P, Q64PN 形電源ユニットの総称
AnS シリーズ電源ユニット	A1S61PN, A1S62PN, A1S63P 形電源ユニットの総称
A シリーズ電源ユニット	A61P, A61PN, A62P, A63P, A68P, A61PEU, A62PEU 形電源ユニットの総称
スリムタイプ電源ユニット	Q61SP 形スリムタイプ電源ユニットの略称
二重化電源ユニット	Q63RP, Q64RP 形電源二重化システム用電源ユニットの総称
■ ネットワーク	
MELSECNET/H	MELSECNET/H ネットワークシステムの略称
Ethernet	Ethernet ネットワークシステムの略称
CC-Link	Control & Communication Link の略称
■ メモリカード	
メモリカード	SRAM カード, Flash カード, ATA カードの総称
SRAM カード	Q2MEM-1MBSN, Q2MEM-1MBS, Q2MEM-2MBSN, Q2MEM-2MBS, Q3MEM-4MBS 形 SRAM カードの総称
Flash カード	Q2MEM-2MBF, Q2MEM-4MBF 形 Flash カードの総称
ATA カード	Q2MEM-8MBA, Q2MEM-16MBA, Q2MEM-32MBA 形 ATA カードの総称
■ その他	
GX Developer	Q シリーズ対応の SW □ D5C-GPPW 形 GPP 機能ソフトウェアパッケージの製品名
QA6ADP	QA6ADP 形 QA 変換アダプタユニットの略称
QA1S6ADP	QA1S6ADP 形 Q-AnS ベースユニット変換アダプタの略称
増設ケーブル	QC05B, QC06B, QC12B, QC30B, QC50B, QC100B 形増設ケーブルの総称
トラッキングケーブル	QC10TR, QC30TR 形二重化 CPU 用トラッキングケーブルの総称
バッテリー	Q6BAT, Q7BATN, Q7BAT, Q8BAT 形 CPU ユニット用バッテリー, Q2MEM-BAT 形 SRAM カード用バッテリー, Q3MEM-BAT 形 SRAM カード用バッテリーの総称
GOT	三菱電機グラフィックオペレーションターミナル GOT-A ***シリーズ, GOT-F ***シリーズ, GOT1000 シリーズの総称

第1章 概要

CPU ユニットは、プログラムを実行することにより、シーケンス制御を行います。
本章では、CPU ユニットの処理順序、CPU ユニットへのプログラムの格納、プログラミングに便利なデバイスや命令などについて説明します。

1.1 CPU ユニットの処理順序

CPU ユニットは下記の順序で処理を行います。

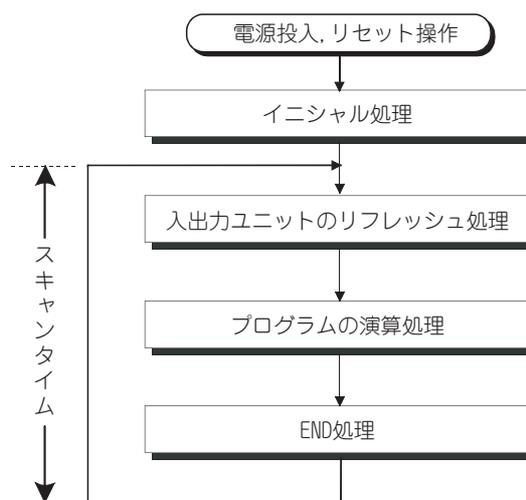


図 1.1 CPU ユニットの処理順序

(1) イニシャル処理 (☞ 3.1 節)

プログラムの演算処理を実行するための前処理です。
電源投入リセット操作時などに 1 回だけ実行されます。

(2) 入出力ユニットのリフレッシュ処理 (☞ 3.2 節)

入出力ユニット/インテリジェント機能ユニットから CPU ユニットへの ON/OFF データの入力と、CPU ユニットから入出力ユニット/インテリジェント機能ユニットへの ON/OFF データの出力を行います。

(3) プログラムの演算処理 (☞ 3.3 節)

CPU ユニットに格納されているプログラムのステップ 0 から END/FEND 命令までを順次実行します。

(4) END 処理 (☞ 3.4 節)

ネットワークユニットなどとのリフレッシュ処理や、外部機器との通信などを行います。

1.2 プログラムの格納と実行

CPU ユニットへのプログラムの格納と実行について示します。

(1) プログラムの作成

GX Developer で作成します。

プログラムの概要、構成、実行条件などについては第2章を参照してください。

(2) プログラムの格納

作成したプログラムやパラメータは、CPU ユニットの下記のメモリに格納します。(☞ 5.1 節, 5.2 節)

- プログラムメモリ
- 標準 ROM
- メモリカード

(3) プログラムの実行

CPU ユニットは、プログラムメモリに格納されているプログラムの演算を行います。

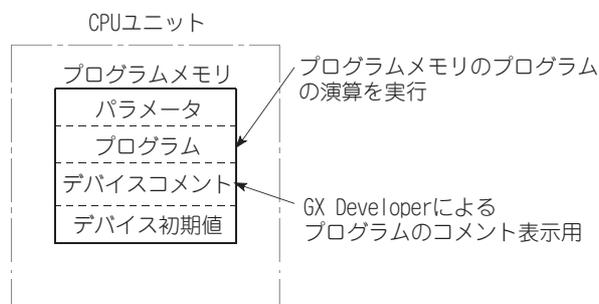


図 1.2 プログラムの実行

プログラムを標準 ROM / メモリカードに格納した場合は、CPU ユニットの電源 OFF → ON またはリセット解除時にプログラムメモリにブート (☞ 5.1.5 項) して実行できます。

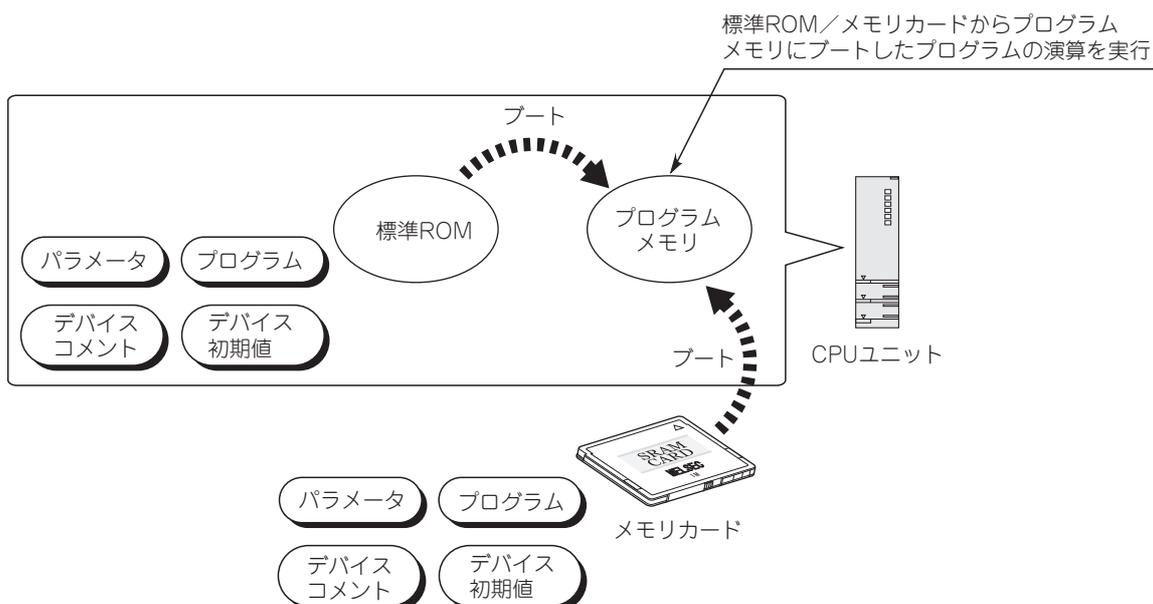


図 1.3 ブート運転によるプログラムの実行

1.3 プログラムの構造化

CPU ユニットで実行するプログラムを構造化できます。
プログラムの構造化には、下記の2種類があります。

- 同一プログラム内での構造化
- ファイルを分割した構造化 **注 1.1**

(1) 同一プログラム内での構造化

プログラムをメインルーチンプログラム (☞ 2.2.1 項)、サブルーチンプログラム (☞ 2.2.2 項)、割込みプログラム (☞ 2.2.3 項) を作成することにより構造化できます。

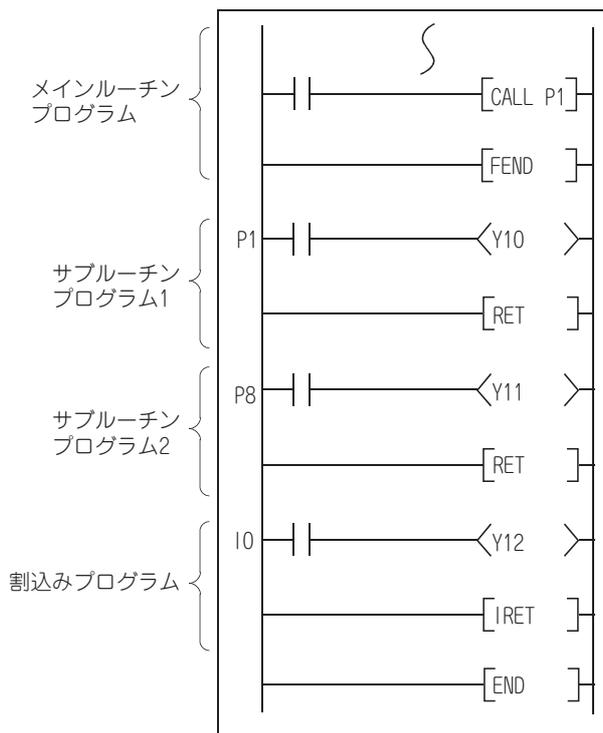


図 1.4 同一プログラム内での構造化



注 1.1

Basic

ベーシックモデル QCPU では、複数のプログラムが格納できないため、ファイルを分割した構造化はできません。

(2) ファイルを分割した構造化

プログラムは、ファイルとして格納します。

ファイル名を変えることにより、複数のプログラムをCPUユニットに格納できます。

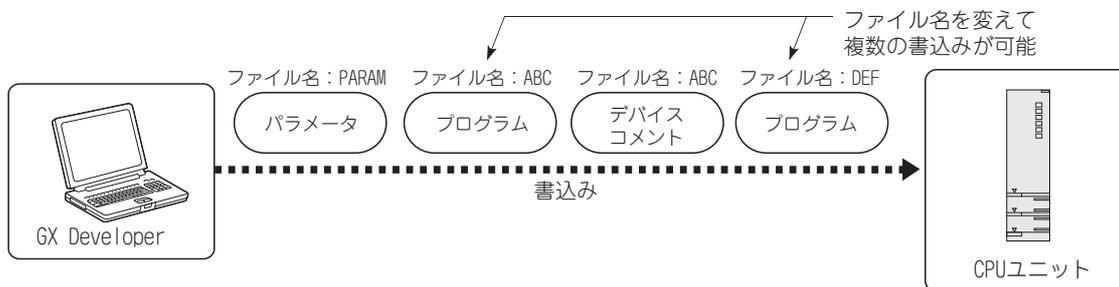


図 1.5 ファイルを分割した構造化

工程別や機能別にプログラムを分割すると、複数の設計者が並行してプログラミングできます。

また、分割して管理すると、他のプログラムへの流用や再利用が簡単にできます。

仕様変更が発生した場合は、該当するファイルのみを修正・デバッグできるので、効率的に対応できます。

(a) 工程別に分割した場合 * 1

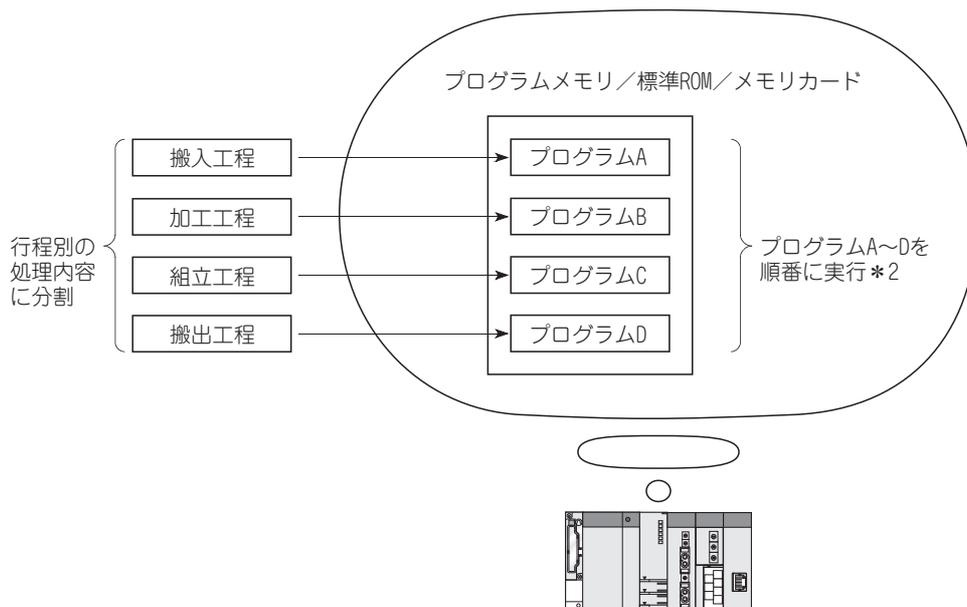


図 1.6 工程別に分割した場合

* 1 : 工程別の処理内容をさらに機能別に分割して管理できます。

* 2 : PC パラメータのプログラム設定 (2.3 節 (2)) で設定できます。

(b) 機能別に分割した場合

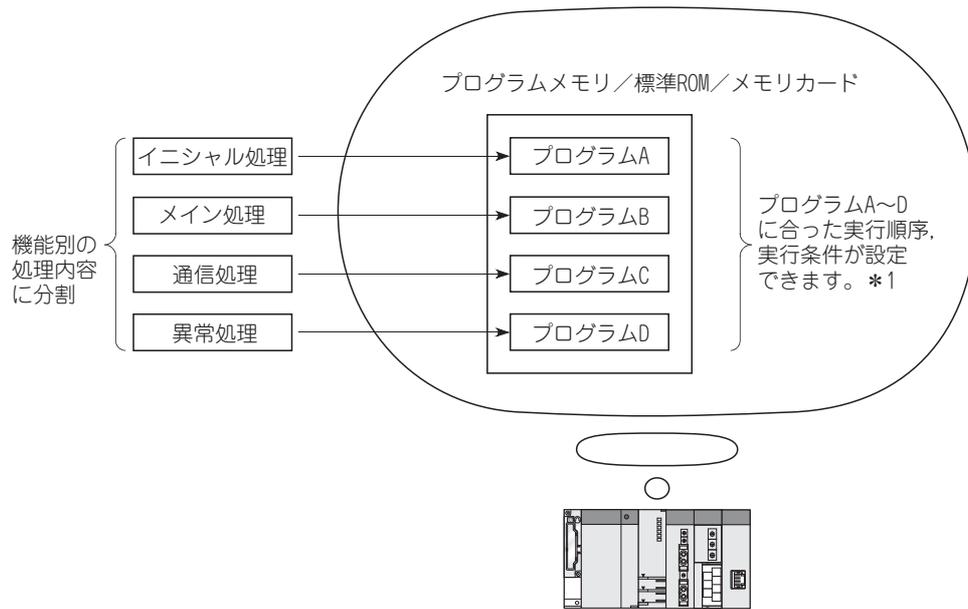


図 1.7 機能別に分割した場合

* 1 : PC パラメータのプログラム設定 (☞ 2.3 節 (2)) で設定できます。

1.4 プログラミングに便利なデバイス、命令

CPU ユニットには、プログラム作成に便利なデバイス、命令が用意されています。その主なものについての概要を説明します。

(1) 柔軟なデバイス指定が可能

(a) ワードデバイスの各ビットを接点、コイル扱いで使用

ワードデバイスのビット指定により、ワードデバイスの各ビットを接点、コイル扱いで使用できます。

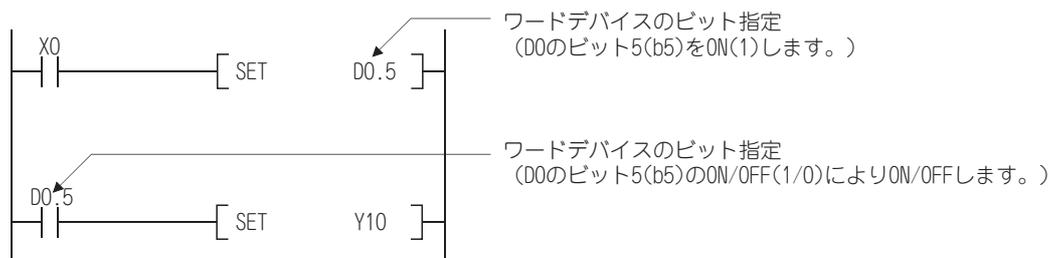


図 1.8 ワードデバイスのビット指定

(b) 1点単位のダイレクト処理が容易に可能

ダイレクトアクセス入力 (DX □), ダイレクトアクセス出力 (DY □) により、プログラム中で1点単位のダイレクト処理が容易に行えます。(☞ 3.8.2 項)

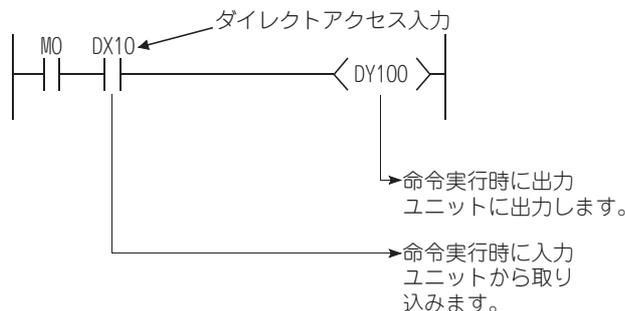


図 1.9 1点単位のダイレクト処理

(c) 微分接点の使用により入力のパルス化処理が不要

微分接点 (⌣ / ⌣) の使用により入力のパルス化処理が不要になります。

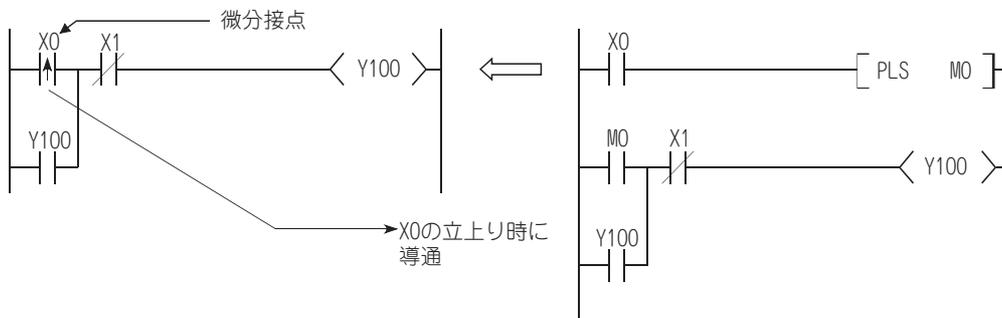


図 1.10 微分接点の使用

(d) インテリジェント機能ユニットのバッファメモリに直接アクセス

インテリジェント機能ユニットのバッファメモリを、デバイス扱いでプログラミングできます。
 (☞ 9.5.1 項)

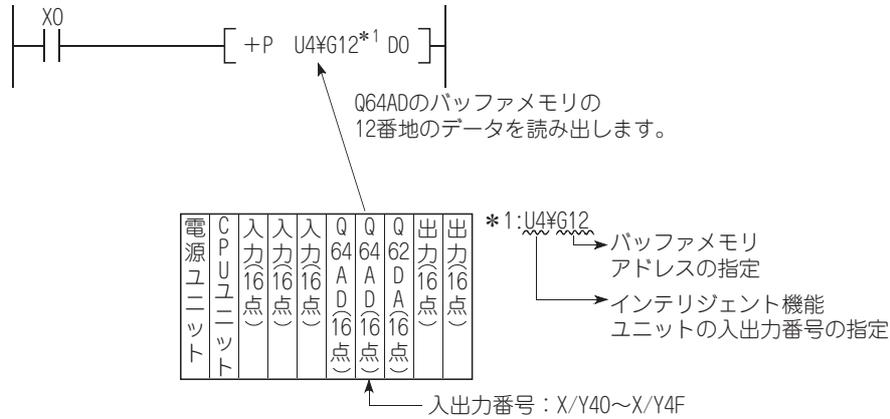


図 1.11 インテリジェント機能ユニットのバッファメモリへのアクセス

(e) リンクデバイスに直接アクセス

ネットワークユニットのリンクデバイス (LX, LY, LB, LW, SB, SW) は、リフレッシュ設定をしなくても
 直接アクセスできます。(☞ 9.4 節)

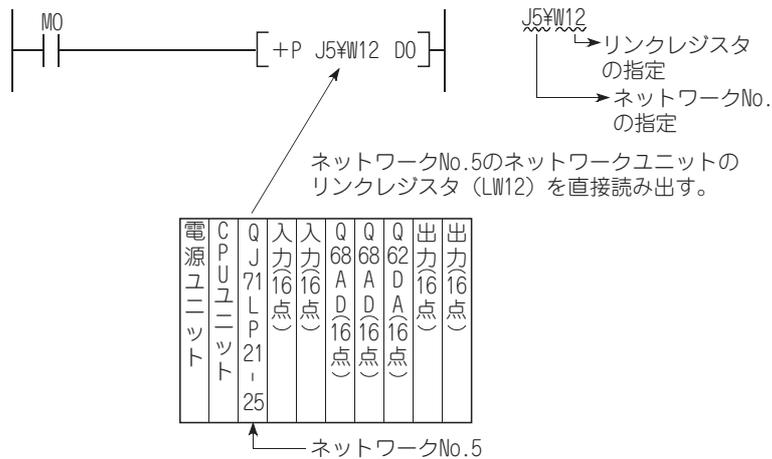


図 1.12 リンクデバイスへの直接アクセス

(2) プログラムを構造的に記述

インデックスレジスタおよびエッジリレーにより、パルス処理を含めたプログラムを簡単に構造化できます。
 (☞ 9.2.6 項)

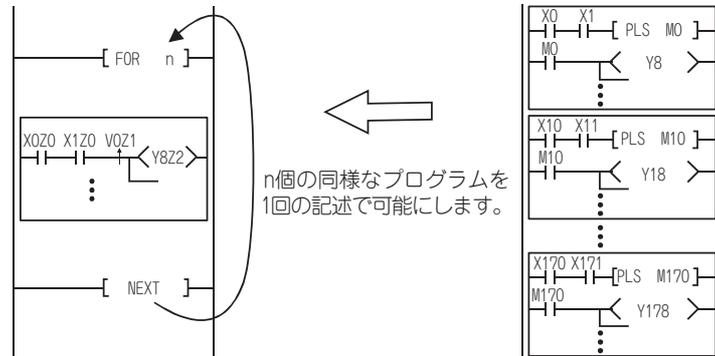


図 1.13 パルス処理を含めたプログラムの構造化

(3) データ処理が容易

(a) 実数および文字列定数をそのまま使用

実数（浮動小数点データ）、文字列定数をそのままプログラミングで使用できます。

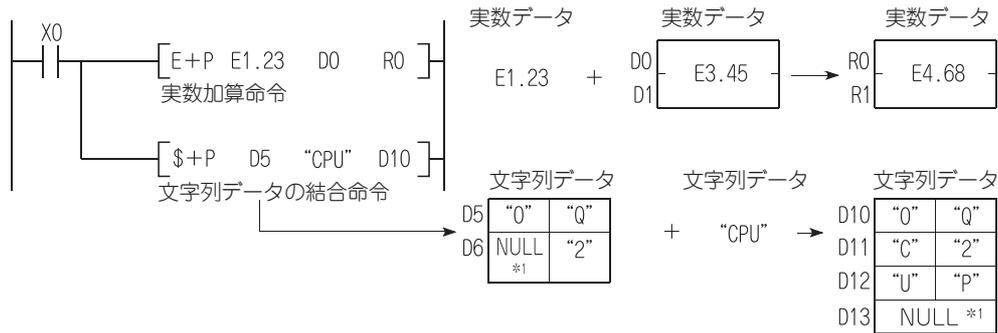


図 1.14 実数および文字列定数の使用

* 1 : NULL は “00H (文字列の最終)” を示しています。

(b) 大容量データの高速処理が可能

データテーブル処理などのデータテーブル操作命令の充実により大容量データの高速処理が可能です。

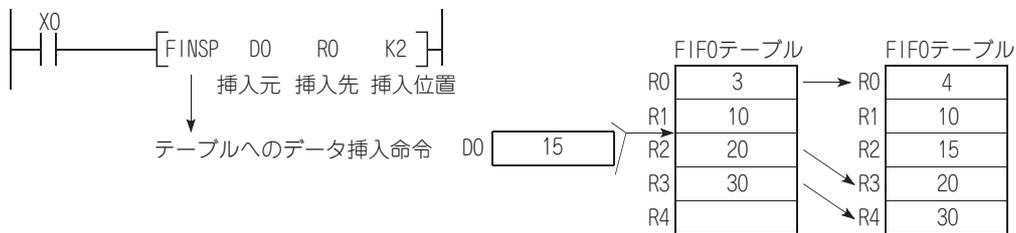


図 1.15 テーブル処理命令のデータ処理

1.4 プログラミングに便利なデバイス、命令

(4) 柔軟なサブルーチンプログラムの管理が可能

(a) サブルーチンプログラムの共用化

サブルーチンプログラムの共用化により、プログラムのステップ数を少なくできます。
また、プログラムの作成および管理も容易になります。

サブルーチンプログラムは、同一プログラム内に作成して呼び出し可能ですが、共通ポインタを使用することにより、他のプログラム内のサブルーチンプログラムも呼び出し可能です。

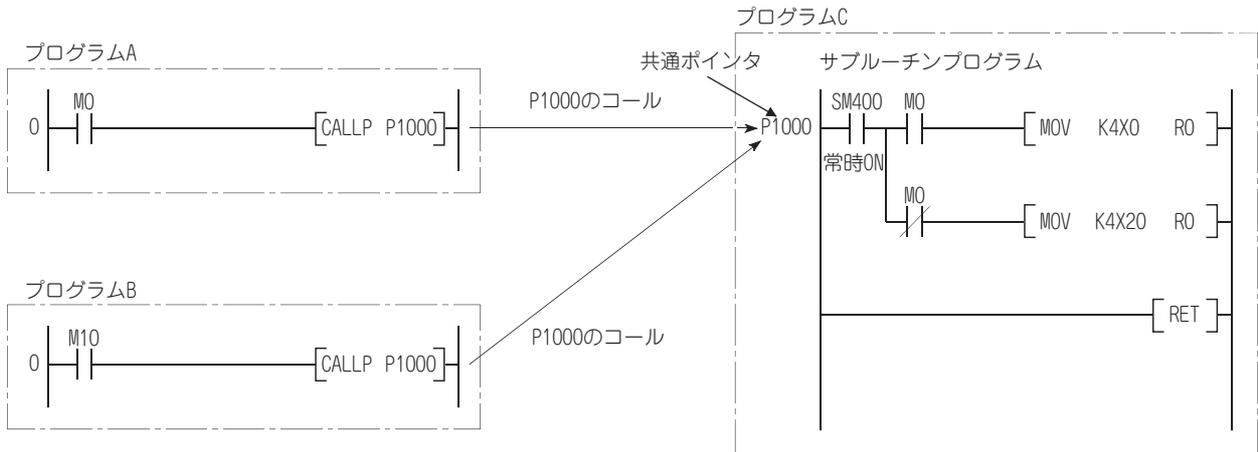


図 1.16 サブルーチンプログラムの共用化

(b) 引数付きサブルーチンコール命令

複数回呼び出しするサブルーチンプログラムの作成が容易になります。

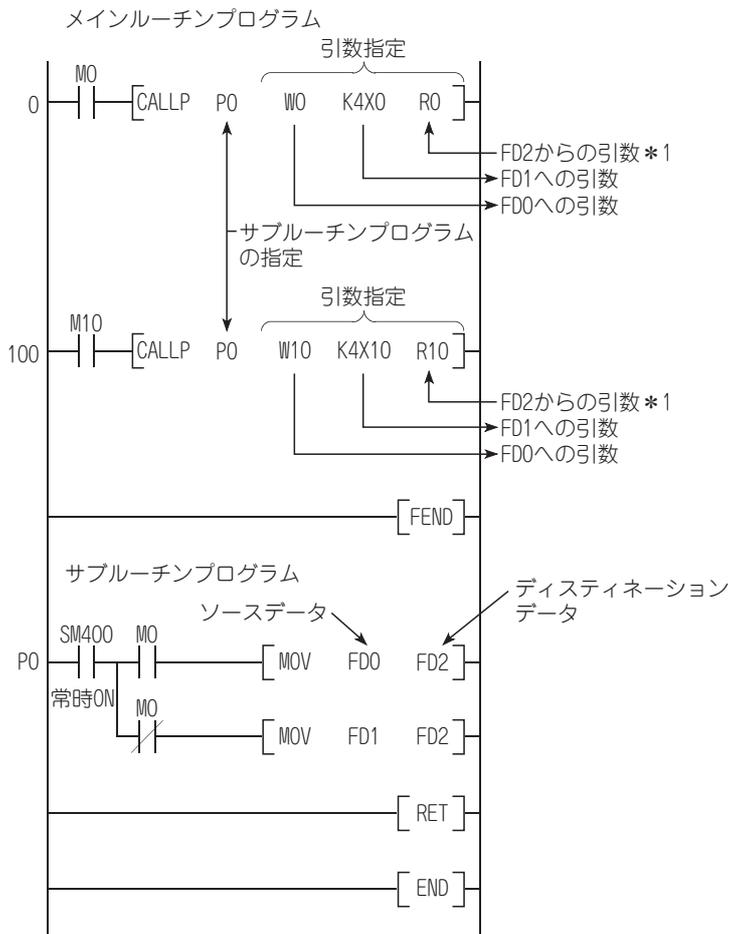


図 1.17 引数付きサブルーチンプログラムの呼び出し

* 1 : 引数のインプット／アウトプット条件については、9.3.1 項を参照してください。

1.5 特 長

各 CPU ユニットの特長について説明します。

1.5.1 ベーシックモデル QCPU の特長

ベーシックモデル QCPU 特有の特長を次に示します。

(1) 小規模システムに最適なコストパフォーマンス

ベーシックモデル QCPU は、小規模システムを対象とした簡単でコンパクトなシステムの制御に最適なユニットです。

これにより、小規模システムに最適なコストパフォーマンスを実現します。

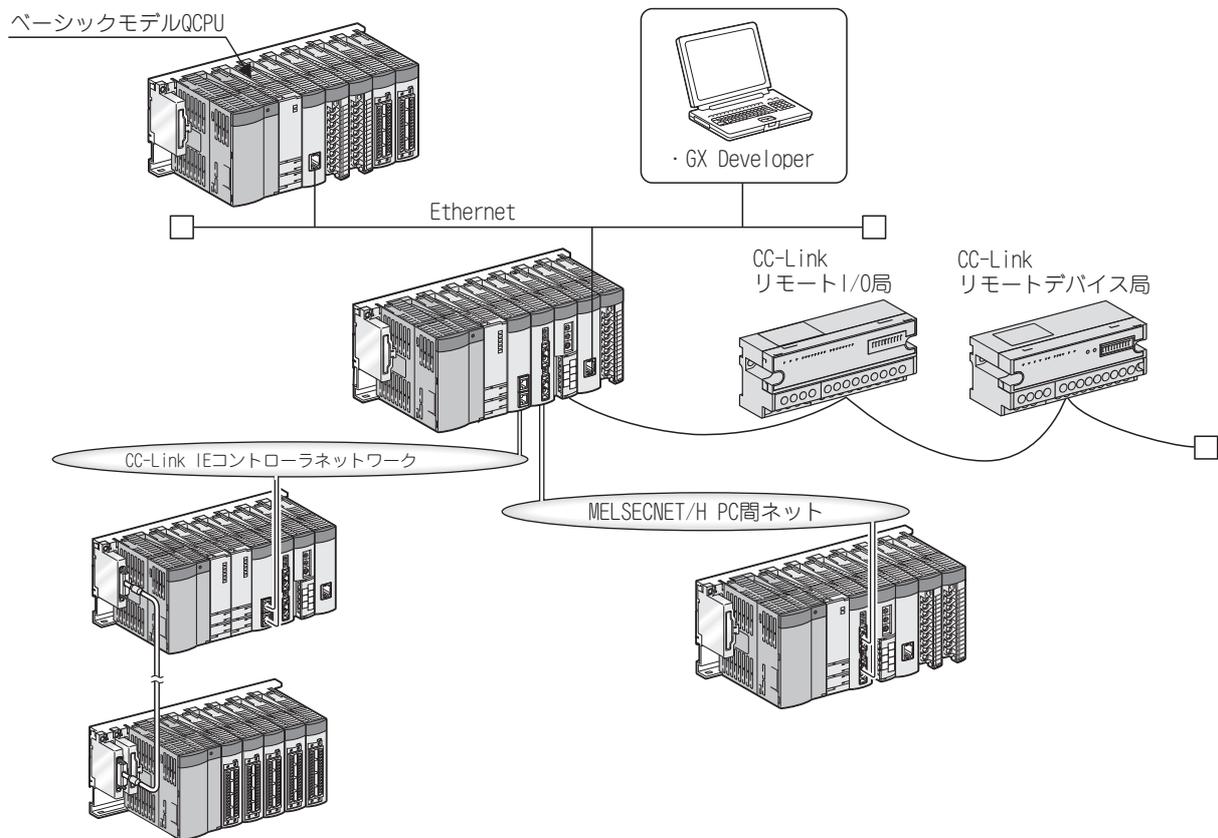


図 1.18 ベーシックモデル QCPU を使用したシステム例

(2) シリアルコミュニケーション機能によるパソコン、表示器との交信 (☞ 6.24 節)

Q00CPU, Q01CPU では、RS-232 インタフェースとパソコン/表示器などを接続することにより、MELSEC コミュニケーションプロトコル (以下、MC プロトコルと略します) での交信が可能です

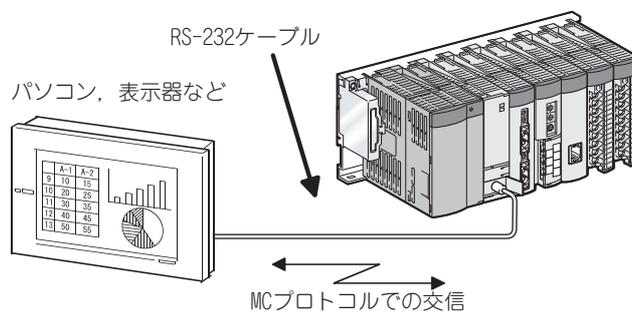


図 1.19 パソコン、表示器との交信

Point

CPU ユニットのシリアル No. や GX Developer のバージョンアップにより、CPU ユニットの機能が追加されています。ベーシックモデル QCPU での機能アップで追加された機能については、付 2.1 を参照してください。

1.5.2 ハイパフォーマンスモデル QCPU の特長

ハイパフォーマンスモデル QCPU 特有の特長を次に示します。

(1) 高性能で大容量

ハイパフォーマンスモデル QCPU は、小規模から大規模システムまでを対象とする、大容量で高速処理が可能なユニットです。

これにより、最適で高性能な設備の構築を実現します。

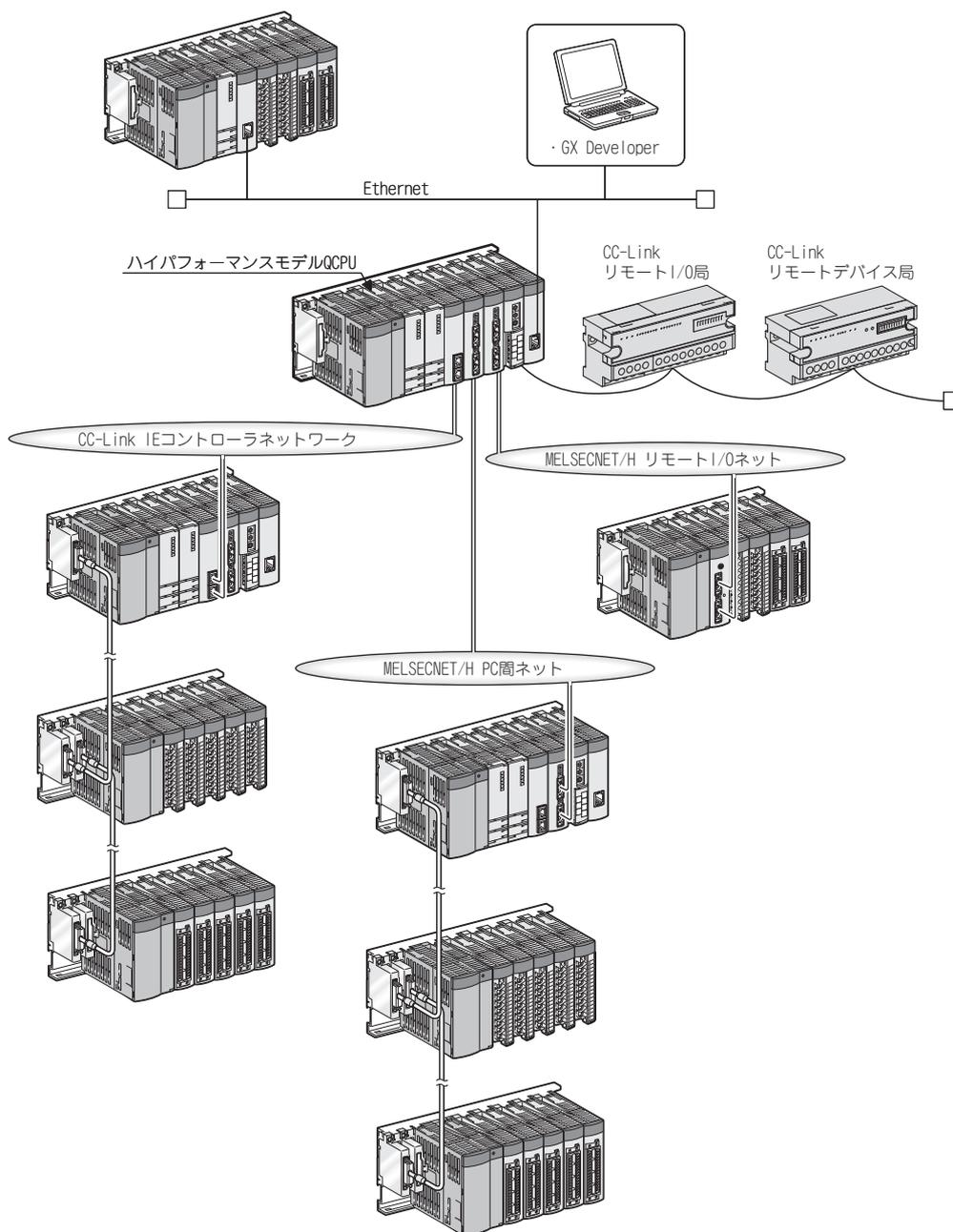


図 1.20 ハイパフォーマンスモデル QCPU を使用したシステム例

(2) AnS/A シリーズの入出力ユニット，特殊機能ユニットが使用可能

AnS/A シリーズ対応の増設ベースユニット（QA1S5 □ B，QA1S6 □ B，QA1S6ADP+A1S5 □ B/A1S6 □ B，QA6 □ B，QA6ADP+A5 □ B/A6 □ B）により，AnS/A シリーズの入出力ユニットや特殊機能ユニットを使用できます。

Point

CPU ユニットのシリアル No. や GX Developer のバージョンアップにより，CPU ユニットの機能が追加されています。ハイパフォーマンスモデル QCPU での機能アップで追加された機能については，付 2.2 を参照してください。

1.5.3 プロセス CPU の特長

プロセス CPU 特有の特長を次に示します。

(1) プロセス制御命令として 52 命令を追加

ハイパフォーマンスモデル QCPU をベースに、高度なプロセス制御に対応した 52 命令を追加しました。追加された命令の詳細については、下記マニュアルを参照してください。

☞ QnPHCPU/QnPRHCPU プログラミングマニュアル（プロセス制御命令編）

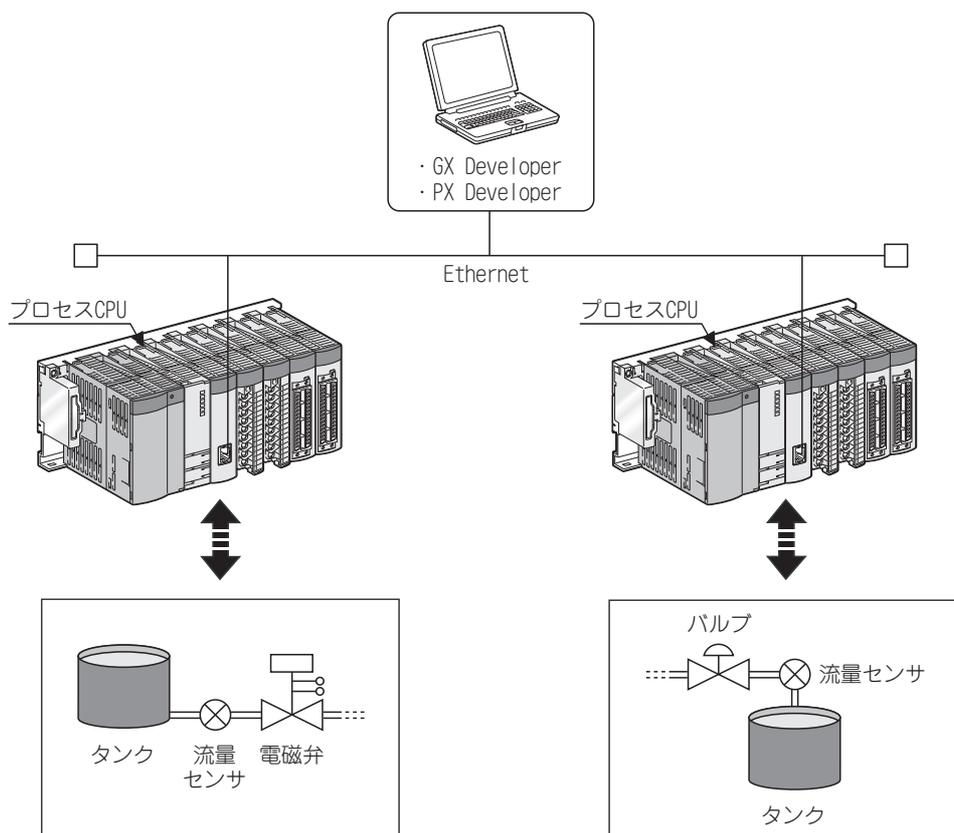


図 1.21 プロセス CPU の動作

(2) 2 自由度 PID 制御方式を採用

目標値変動，外乱変動の両方に最適応答が可能になります。

(3) オートチューニング機能（PID 定数初期値設定）

制御パラメータの調整作業の自動化，調整作業の短縮化，オペレータや制御エンジニアの省力化，調整結果の個人差解消を実現します。

オートチューニング機能の詳細については、下記マニュアルを参照してください。

☞ QnPHCPU/QnPRHCPU プログラミングマニュアル（プロセス制御命令編）

(4) オンライン中のユニット交換（オンラインユニット交換）が可能

ユニットの故障時に、システムを停止することなくユニットを交換できます。
オンラインユニット交換の詳細については、下記マニュアルを参照してください。

- ☞ QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）
- ☞ オンラインユニット交換に対応したユニットのマニュアル

(5) MELSECNET/H 多重リモート I/O システムの構築が可能

MELSECNET/H リモートマスタ局を装着すると、MELSECNET/H 多重リモート I/O システムが構築できます。

(6) 計装専用ソフトウェアパッケージ (PX Developer) に対応

プロセス制御用ソフトウェアパッケージ (PX Developer) を使用すると、PID 制御のプログラムをファンクションブロックにより簡単に作成できます。

プロセス制御用ソフトウェアパッケージ (PX Developer) との組合せにより、優れたエンジニアリング環境を提供します。

Point

- Q12PHCPU, Q25PHCPU を使用する場合には、Version 7.10L 以降の GX Developer を使用してください。
Q02PHCPU, Q06PHCPU を使用する場合には、Version 8.68W 以降の GX Developer を使用してください。
- PX Developer は、Version 7.12N 以降の GX Developer と組み合わせて使用します。
Q12PHCPU, Q25PHCPU を使用する場合には、Version 1.00A 以降の PX Developer を使用してください。
Q02PHCPU, Q06PHCPU を使用する場合には、Version 1.18U 以降の PX Developer を使用してください。
PX Developer の詳細については、PX Developer のマニュアルを参照してください。
- CPU ユニットのシリアルNo.やGX Developerのバージョンアップにより、CPUユニットの機能が追加されています。
プロセス CPU での機能アップで追加された機能については、付 2.3 を参照してください。

1.5.4 二重化 CPU の特長

二重化 CPU 特有の特長を次に示します。

(1) プロセス CPU の機能に加えて二重化システムに対応

(a) 二重化 CPU を使用した二重化システム

二重化 CPU を使用することにより、ベースユニット、電源ユニット、CPU ユニット（二重化 CPU）を含めたシステム全体を二重化できます。

制御系で故障が発生した場合でも待機系に制御が切り替わることにより、システムの高信頼化を実現できます。

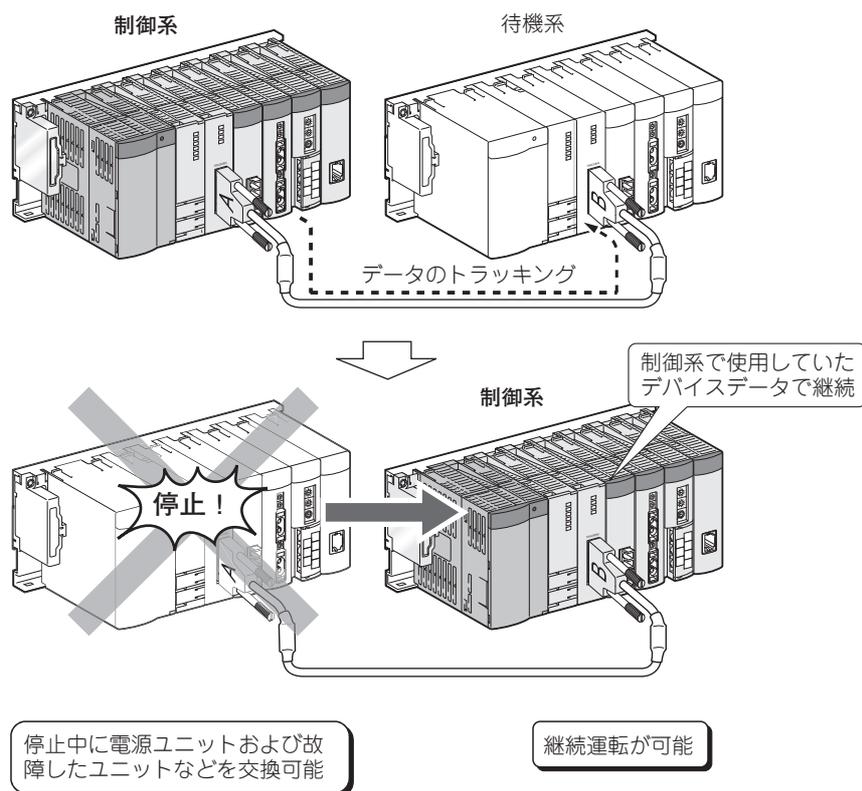


図 1.22 二重化 CPU の動作

(b) 電源二重化システム

電源二重化基本ベースユニット (Q3 □ RB), および二重化電源ユニット (Q63RP, Q64RP) をリモート I/O 局側に使用することにより, リモート I/O 局側の電源を二重化できます。

これにより, リモート I/O 局側の電源ユニットが故障した場合でも, システムを停止することなく電源ユニットの交換が可能です。

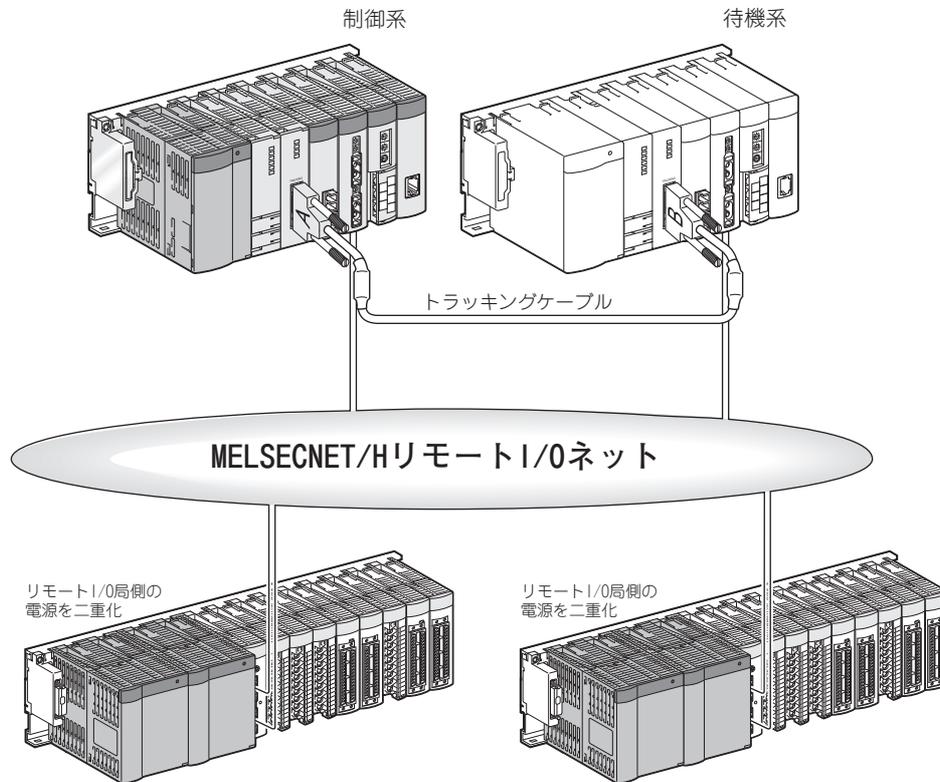


図 1.23 電源二重化システム

Point

- GX Developer と PX Developer を組み合わせて使用します。
 - Version 8.17T 以降の GX Developer
 - Version 1.05F 以降の PX Developer
- CPU ユニットのシリアル No. や GX Developer のバージョンアップにより, CPU ユニットの機能が追加されています。二重化 CPU での機能アップで追加された機能については, 付 2.4 を参照してください。

備考

二重化 CPU の特長, 機能などの詳細については, 下記マニュアルを参照してください。

☞ QnPRHCPU ユーザーズマニュアル (二重化システム編)

1.6 シリアル No. と機能バージョンの確認方法

CPU ユニットのシリアル No. と機能バージョンは、定格銘板やユニット前面、および GX Developer のシステムモニタで確認できます。

(1) 定格銘板での確認

定格銘板は、CPU ユニットの側面にあります。

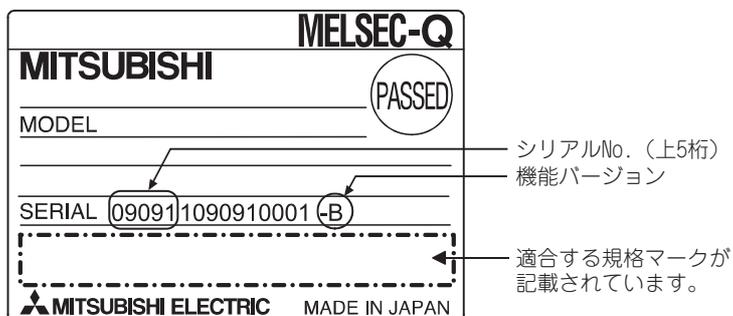


図 1.24 定格銘板

(2) ユニット前面での確認

ユニット前面（下部）に定格銘板に記載されているシリアル No. を表示しています。2007 年 9 月中旬以前に生産したユニットは対象外です。

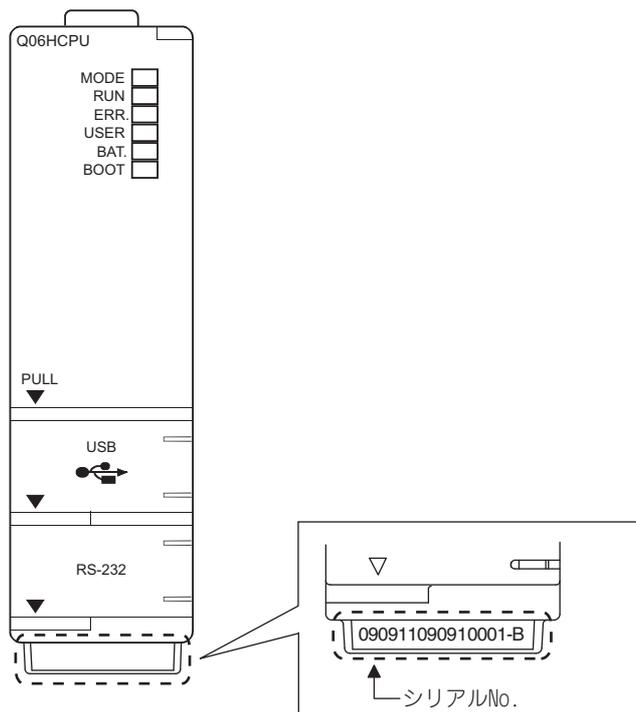


図 1.25 ユニット前面表示

(3) システムモニタ（製品情報一覧）での確認

システムモニタの表示は、GX Developer の [診断] → [システムモニタ] の **製品情報一覧** ボタンをクリックします。

また、インテリジェント機能ユニットのシリアル No. と機能バージョンの確認もできます。



図 1.26 システムモニタ（製品情報一覧）

【シリアル No., Ver.】

- シリアル No. 欄に、該当ユニットのシリアル No. が表示されます。
- Ver. 欄に、該当ユニットの機能バージョンが表示されます。

Point

定格銘板、ユニット前面に記載されているシリアル No. と、GX Developer の製品情報一覧に表示されるシリアル No. は異なることがあります。

- 定格銘板、ユニット前面のシリアル No. は、製品の管理情報を示しています。
- GX Developer の製品情報一覧に表示されるシリアル No. は、製品の機能情報を示しています。製品の機能情報は、機能追加時に更新されます。

第2章 シーケンスプログラム

2.1 シーケンスプログラムの概要

(1) シーケンスプログラムとは

CPU ユニットで実行できるプログラムです。

シーケンス命令、基本命令、応用命令などを使用して作成します。

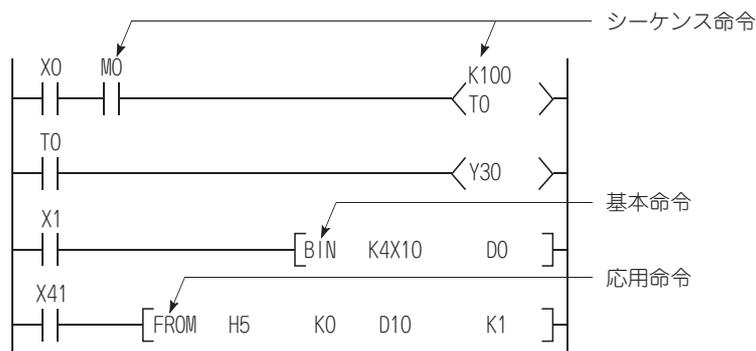


図 2.1 シーケンスプログラム

Point

シーケンスプログラムの各ステップの演算の実行/非実行結果を“シグナルフロー”といいます。

備考

シーケンスプログラムで使用する命令については、下記マニュアルを参照してください。

☞ MELSEC-Q/L プログラミングマニュアル（共通命令編）

(2) プログラミング方法

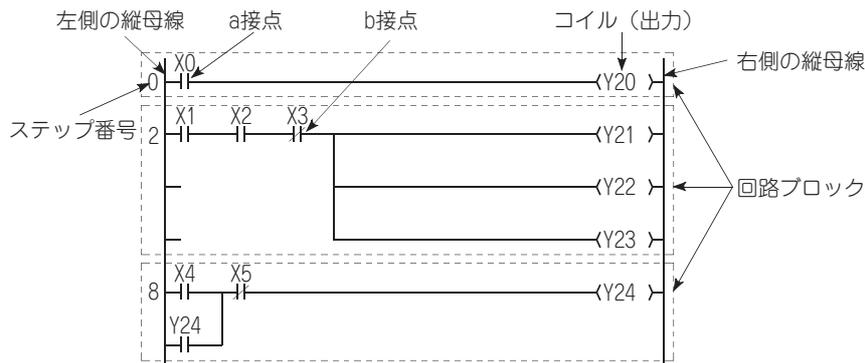
シーケンスプログラムのプログラミング方法には、主に下記の2種類があります。

- 回路モードによる方法
- リストモードによる方法

(a) 回路モードによる方法

回路モードは、リレー制御のシーケンス回路の考え方を基本にしたモードです。シーケンス図に近い表現で回路ブロック単位ごとにプログラミングができます。

回路ブロックは、左側の縦母線で始まり、右側の縦母線で終わる回路で、シーケンスプログラムの演算を行う最小単位です。



* X0~5は入力, Y20~24は出力を示しています。

図 2.2 回路モード

(b) リストモードによる方法

リストモードは、回路モードで使用していた接点、コイルなどの記号を、専用の命令に置き換えてプログラミングします。

a 接点, b 接点, コイルは次のような命令になります。

- a 接点 … LD, AND, OR
- b 接点 … LDI, ANI, ORI
- コイル … OUT

備考

シーケンスプログラムには、他にも SFC プログラムや ST プログラムがあります。詳細については、下記マニュアルを参照してください。

- 📖 MELSEC-Q/L/QnA プログラミングマニュアル (SFC 編)
- 📖 MELSEC-Q/L プログラミングマニュアル (MELSAP-L 編)
- 📖 MELSEC-Q/L プログラミングマニュアル (ストラクチャードテキスト編)

(3) シーケンスプログラムの演算

シーケンスプログラムの演算は、ステップ0からEND/FEND命令までを順次実行します。
回路モードの回路ブロックでは、左側の縦母線から右へ、上から下への順で演算を行います。

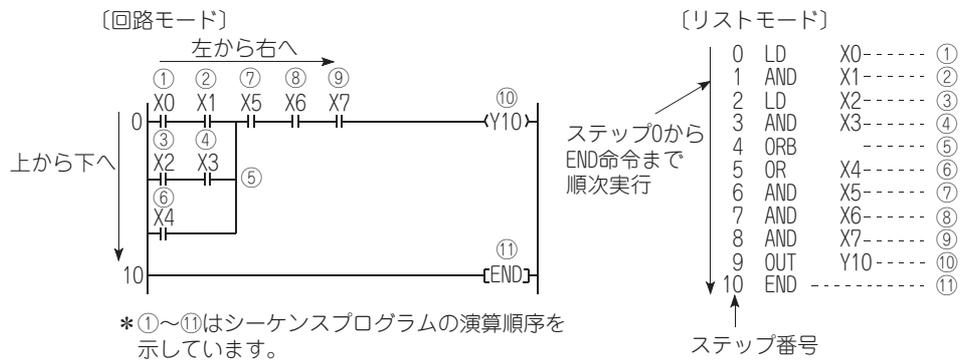


図 2.3 回路モードとリストモードの比較

2.2 シーケンスプログラムの構成

シーケンスプログラムは、下記の3種類に分類されます。

- メインルーチンプログラム
- サブルーチンプログラム
- 割込みプログラム

ファイルA *1

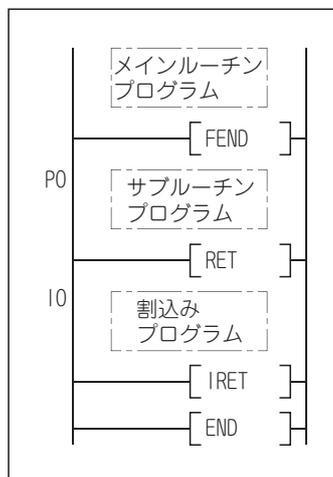


図 2.4 シーケンスプログラムの分類

* 1: ベーシックモデル QCPU では、複数のプログラムを実行できないため、ファイル名は“MAIN”固定となります。

2.2.1 メインルーチンプログラム

(1) メインルーチンプログラムとは

ステップ0 から END/FEND 命令までのプログラムです。

(2) メインルーチンプログラムの実行動作

メインルーチンプログラムは、ステップ0 から END/FEND 命令までのプログラムを実行し、END 処理を行います。

END 処理後に再度ステップ0 から演算を行います。

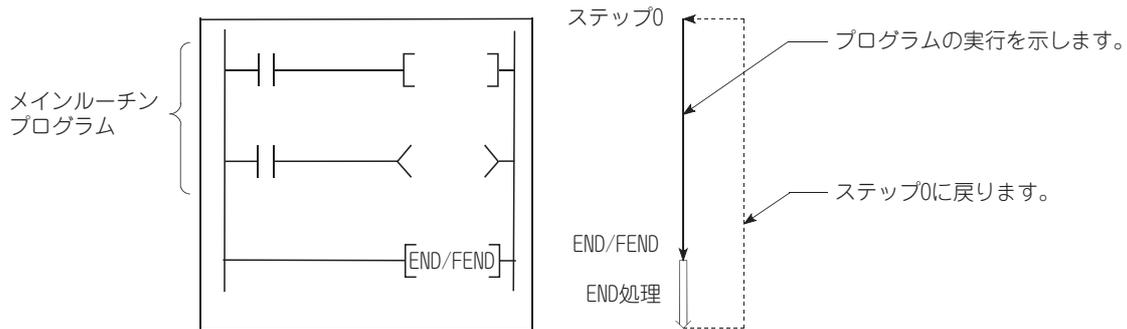


図 2.5 メインルーチンプログラム

Point

複数のプログラムを実行する場合、メインルーチンプログラムの END/FEND 命令実行後の動作は、設定されている実行条件により異なります。(☞ 2.3 節)

備考

END/FEND 命令の詳細については、下記マニュアルを参照してください。

☞ MELSEC-Q/L プログラミングマニュアル（共通命令編）

2.2.2 サブルーチンプログラム

(1) サブルーチンプログラムとは

ポインタ (P□) から RET 命令までのプログラムです。

メインルーチンプログラム内からサブルーチンプログラムのコール命令 (CALL(P), FCALL(P) など) でコールされた場合のみ実行されます。

(2) サブルーチンプログラムの用途

- 1 スキャン中に複数回実行されるプログラムをサブルーチンプログラムにすると、全体のステップ数が少なくなります。
- ある条件でしか実行しないプログラムをサブルーチンプログラムにすると、その分スキャンタイムが短くなります。

(3) サブルーチンプログラムの作成

メインルーチンプログラムの FEND 命令～ END 命令間に作成します。

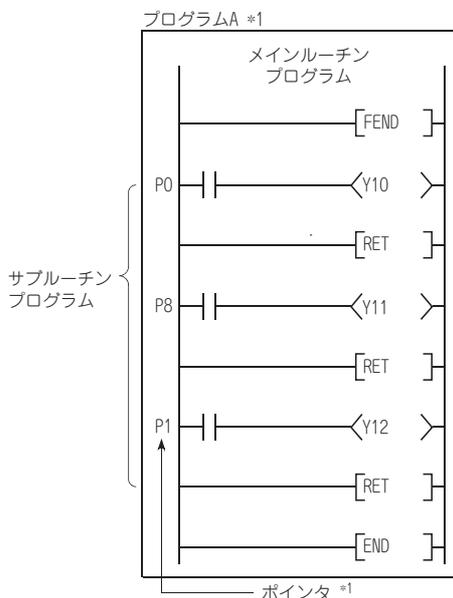


図 2.6 サブルーチンプログラムの作成箇所

- * 1：ベーシックモデル QCPU では、複数のプログラムを実行できないため、ファイル名は“MAIN”固定となります。
- * 2：ポインタを若番順にする必要はありません。

Point

サブルーチンプログラムは、別のプログラム (待機タイププログラム) として管理できます。(☞ 2.3.4 項)

備考

ネスタリングを使用してサブルーチンプログラムを入れ子構造にできます。(☞ 9.8 節)

2.2.3 割り込みプログラム

(1) 割り込みプログラムとは

割り込みポインタ (I□) から IRET 命令までのプログラムです。

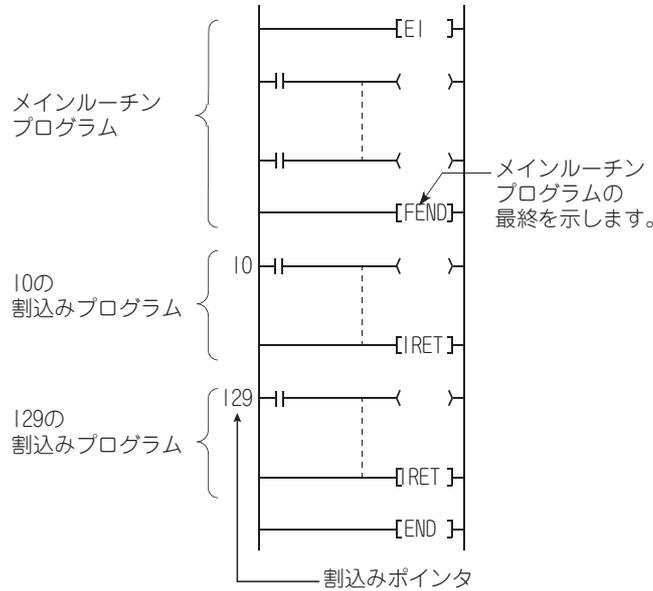


図 2.7 割り込みプログラム

割り込み要因によって、割り込みポインタ (I□) の番号が異なります。(☞ 9.10 節)

割り込み要因が発生すると、その要因に対応する割り込みポインタ番号の割り込みプログラムが実行されます。(割り込みプログラムは、割り込み要因が発生した場合のみ実行されます。)

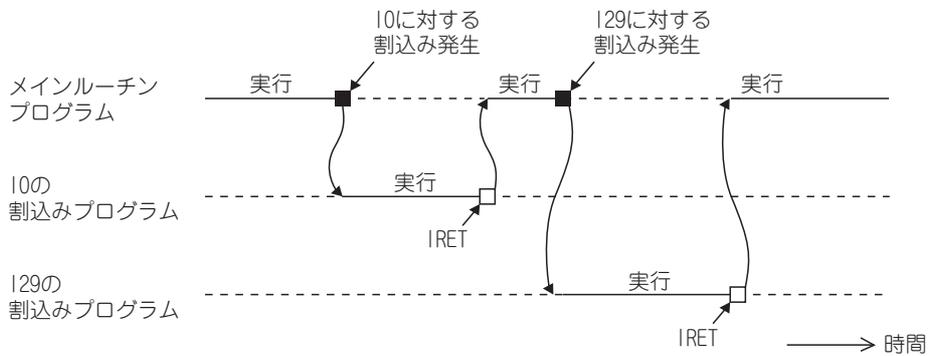


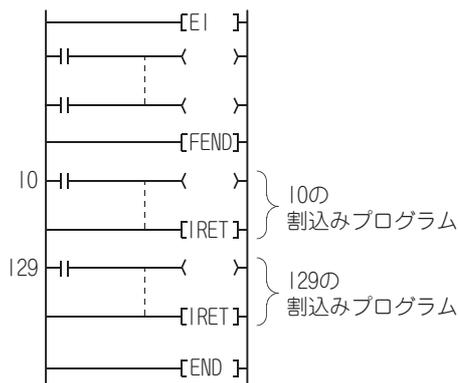
図 2.8 割り込みプログラムの実行タイミング

Point

- 割り込みポイントには、高速割り込み機能専用のポイント（149）**注 2.1**があります。149を使用した場合は、下記に示すポイントおよびプログラムは実行しないでください。
 - ・その他の割り込みポイント（149以外の割り込みポイント）
 - ・割り込みプログラム
 - ・定周期実行タイププログラム

上記に示すプログラムなどを実行すると、149の割り込みプログラムは設定した割り込み周期間隔で実行できません。

- 1つの割り込みポイント番号で作成可能な割り込みプログラムは1つです。



備考

割り込み要因、割り込みポイントの詳細については、9.10節を参照してください。



注 2.1

Basic

High performance

Process

Redundant

ベーシックモデル QCPU、Q02CPU、プロセス CPU、二重化 CPU では、高速割り込み機能専用ポイント（149）は使用できません。

(2) 割込みプログラムの作成

メインルーチンプログラムの FEND 命令～ END 命令間に作成します。

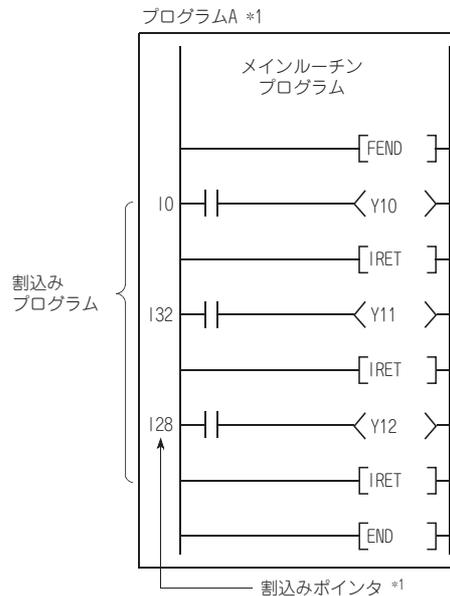


図 2.9 割込みプログラムの作成箇所

- * 1：ベーシックモデル QCPU では、複数のプログラムを実行できないため、ファイル名は“MAIN”固定となります。
- * 2：ポインタを若番順にする必要はありません。

Point

割込みプログラムは、別のプログラム（待機タイププログラム）として管理できます。（☞ 2.3.4 項）

(a) 割込みプログラムを実行する前に

1) ベーシックモデル QCPU の場合

EI 命令を実行して割込み許可状態にします。

2) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU の場合

割込みポインタ I32 ~ 47 の割込みプログラムは、IMASK 命令および EI 命令で割込み許可状態にして実行します。

割込みポインタ I0 ~ 31, I48 ~ 255 の割込みプログラムは、EI 命令で割込み許可状態にして実行します。

備考

IMASK 命令および EI 命令の詳細については、下記マニュアルを参照してください。

☞ MELSEC-Q/L プログラミングマニュアル（共通命令編）

(b) プログラム作成上の制約

1) PLS, PLF 命令について

PLS, PLF 命令は、命令実行後の次のスキャンで OFF 処理を行います。
ON したデバイスは、再度命令が実行されるまで ON のままになります。

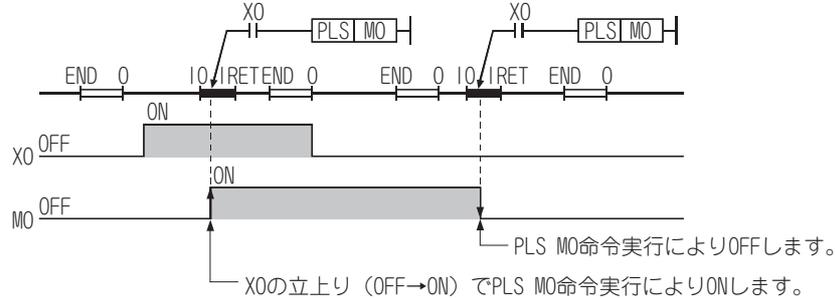


図 2.10 割込みプログラム内の PLS で ON したデバイス

2) EI/DI 命令について

割込みプログラムを実行中は、他の割込み処理を実行しないように割込み禁止 (DI) になっています。
割込みプログラム中では、EI/DI 命令を実行しないでください。

3) タイマ (T), カウンタ (C) について

割込みプログラムでは、タイマ (T) およびカウンタ (C) を使用しないでください。

割込みプログラム内のタイマ (T) およびカウンタ (C) は、1 スキャン中に割込みが複数回発生すると正常に計測できません。

また、カウンタ (C) は、割込みプログラム実行時の OUT C □命令の状態によっても正常に計測することができません。

4) 割込みプログラムで使用できない命令について

下記マニュアルの各命令部分を参照してください。

📖 MELSEC-Q/L プログラミングマニュアル (共通命令編)

(3) 割り込み要因発生時の動作

割り込みプログラムは、割り込み要因が発生したタイミングによって制約があります。

(a) 割り込み許可状態になる前に割り込み要因が発生した場合

CPU ユニットは、発生した割り込み要因を記憶します。

割り込み許可状態になった時点で、記憶した割り込み要因に対応する割り込みプログラムが実行されます。

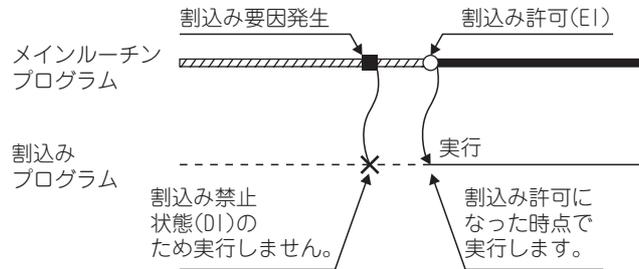


図 2.11 割り込み許可状態になる前に割り込み要因が発生した場合

割り込み許可状態になる前に同一割り込み要因が複数回発生した場合は、下記ようになります。

1) ベーシックモデル QCPU の場合

I0 ~ 15, I28 ~ 31, I50 ~ 127 の割り込み要因は 1 回のみ記憶されます。

2) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU の場合

I0 ~ 27, I28 ~ 31, I50 ~ 255, および定周期実行タイププログラムの割り込み要因は 1 回のみ記憶されます。

I32 ~ 41, I49 の割り込み要因では、割り込み禁止の間に発生した要因は破棄されます。

(b) STOP/PAUSE 状態で割り込み要因が発生した場合

CPU ユニットが RUN 状態になったあとに、割り込み許可になった時点で割り込みプログラムを実行します。

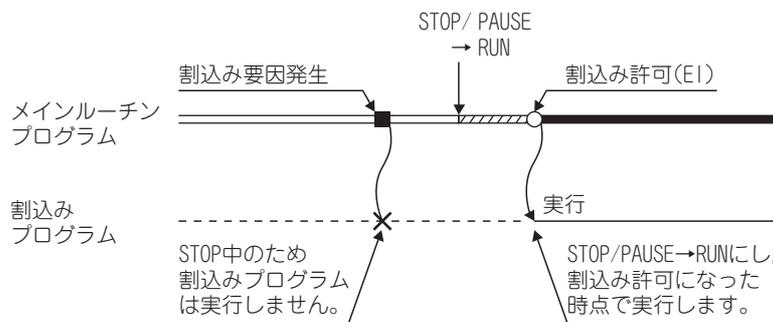


図 2.12 STOP/PAUSE 状態で割り込み要因が発生した場合

(c) 割込み許可状態中に複数の割込み要因が同時に発生した場合

優先順位の高い割込みポイント番号 (I□□) に対応した割込みプログラムから実行されます。(☞ 9.10.1 項)
他の割込みプログラムは、実行中の割込みプログラムの処理が完了するまで待ち状態になります。

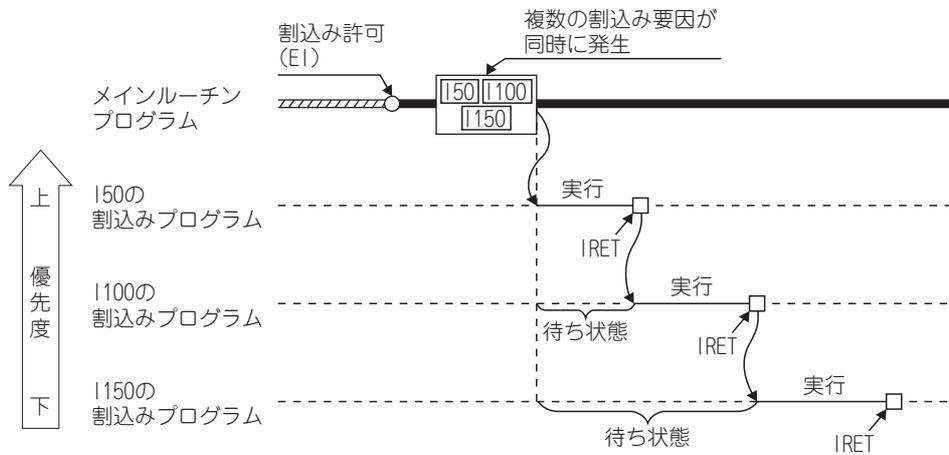


図 2.13 複数の割込み要因が同時に発生した場合

(d) 実行中の割込みプログラムと同一割込み要因が発生した場合

1) ベーシックモデル QCPU の場合

10 ~ 15, 128 ~ 31, 150 ~ 127 の割込み要因は 1 回のみ記憶され、割込みプログラムの実行終了後に、記憶されていた割込み要因の割込みプログラムを実行します。

複数回同一の割込み要因が発生しても記憶できるのは最初の 1 回のみです。

2 回目以降の割込み要因は無視されます。

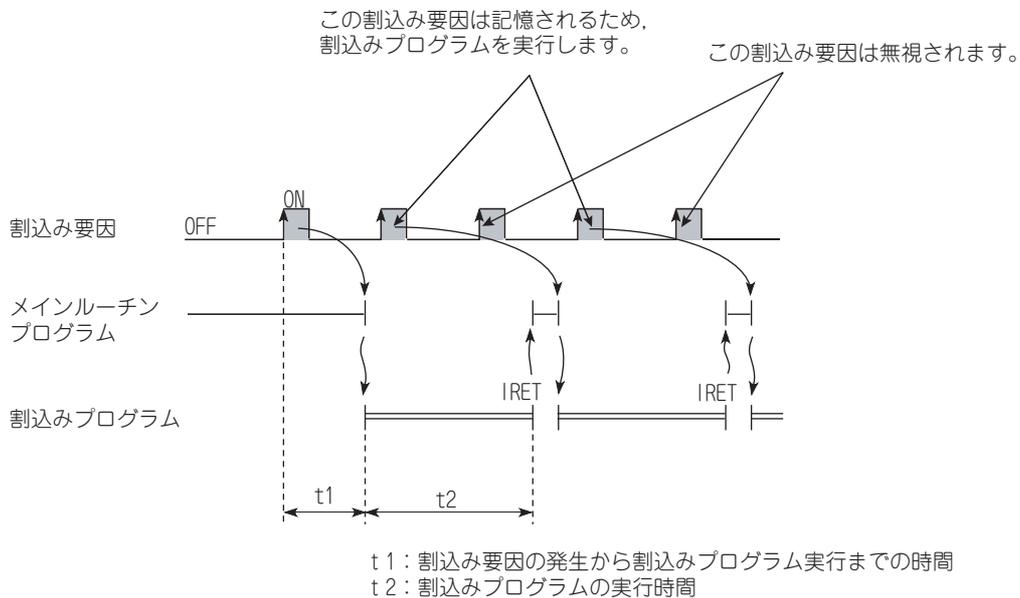


図 2.14 実行中の割込みプログラムと同一割込み要因が発生した場合の動作

2) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU の場合

- 10 ~ 27, 150 ~ 255 の割込み要因は 1 回のみ記憶され、割込みプログラムの実行終了後に、記憶されていた割込み要因の割込みプログラムを実行します。
- 128 ~ 31, および定周期実行タイププログラムは、発生した回数分の割込み要因はすべて記憶し、割込みプログラムの実行終了後に、記憶されていた割込み要因の割込みプログラムを実行します。
- 132 ~ 41, 149 の割込み要因は破棄されます。

(e) リンクリフレッシュ中に割込み要因が発生した場合

リンクリフレッシュを中断し、割込みプログラムを実行します。

CC-Link IE コントローラネットワークまたは、MELSECNET/H でサイクリックデータの局単位ブロック保証を行っても、割込みプログラムでリフレッシュ先に設定しているデバイスを使用すると、サイクリックデータの局単位ブロック保証ができません。

割込みプログラムでは、リフレッシュ先のデバイスを使用しないでください。

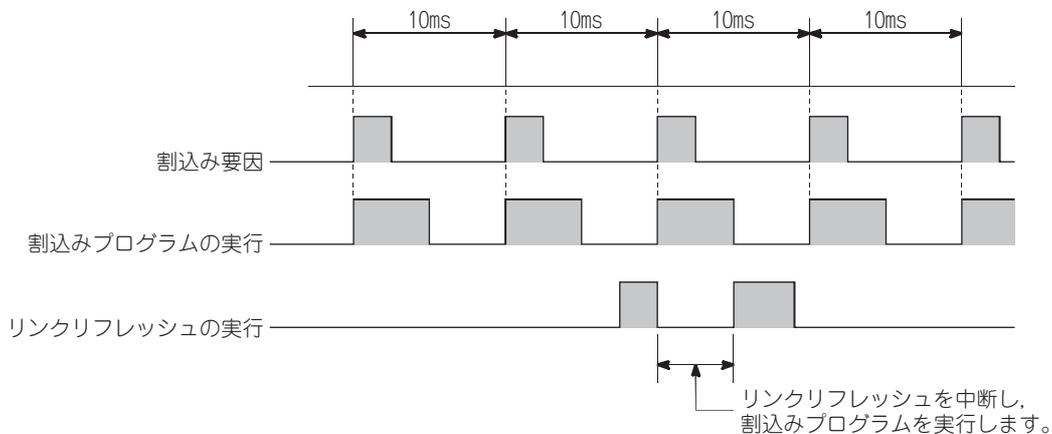


図 2.15 リンクリフレッシュ中に割込みが発生した場合

備考

サイクリックデータの局単位ブロック保証については、下記マニュアルを参照してください。

☞ 各ネットワークユニットのマニュアル

(f) END 処理中の割込み

コンスタントスキャン実行時、END 処理の待ち時間中に割込み要因が発生した場合は、割込み要因に対応する割込みプログラムを実行します。

(g) 他のユニットへのアクセス中に割込み要因が発生した場合

他のユニットへアクセスしているとき（サービス処理中、または命令処理中）に割込み要因が発生した場合、割込みプログラムは、実行中のサービス処理、または命令処理が完了するまで待ち状態になります。

割込みの待ち状態を短くするには、他のユニットへアクセスするデータ量を減らしてください。

(4) プログラム切換え時の処理

CPU ユニットは、スキャン実行タイプ/低速実行タイププログラムから割込みプログラムへの切換え時、下記の内容を退避および復帰します。(☞ 9.6.3 項)

- インデックスレジスタの内容
- ファイルレジスタのブロック No.

パラメータ設定で、上記の内容を退避および復帰するかどうかの設定ができます。

退避および復帰しない場合は、割込みプログラムのオーバヘッド時間を短くできます。(☞ 10.1.2 項)

(5) 注意事項

(a) デバイスを重複して使用する場合

メインルーチンプログラムの命令実行中に、命令の実行処理を中断して割込みプログラムが実行されることがあります。

メインルーチンプログラムと割込みプログラムでデバイスを重複して使用する場合は、デバイスデータに泣き別れが生じることがあるため、下記に示す対策を行ってください。

1) デバイスデータを別のデバイスに移す

割込みプログラムで書き込むデバイスをメインルーチンプログラムで直接指定せずに、転送命令などで別のデバイスに移して使用してください。

2) DI 命令で割込み禁止にする

メインルーチンプログラムの命令実行中に割り込まれると不都合が生じる命令は、DI 命令で割込み禁止にして実行してください。

ただし、命令の各引数のデバイスにアクセスしている最中に割込みプログラムが入り込むことはないため、各引数の単位でデータの泣き別れは発生しません。  **注 2.2**



注 2.2

Basic

ベーシックモデル QCPU では、各引数の単位でデータの泣き別れが発生する可能性があります。

2.3 複数のプログラムに分割した場合の設定 注 2.3

シーケンスプログラムを分割して複数作成した場合は、起動時に1回だけ起動するプログラム、一定間隔で実行するプログラムなどのように、プログラムごとに実行条件を設定できます。

(1) 複数のプログラムに分割した制御

プログラムは、制御単位ごとに複数のプログラムに分割し、CPUユニットに格納できます。これにより、1本のプログラムを処理単位ごとに分割して、複数人でプログラミングできます。

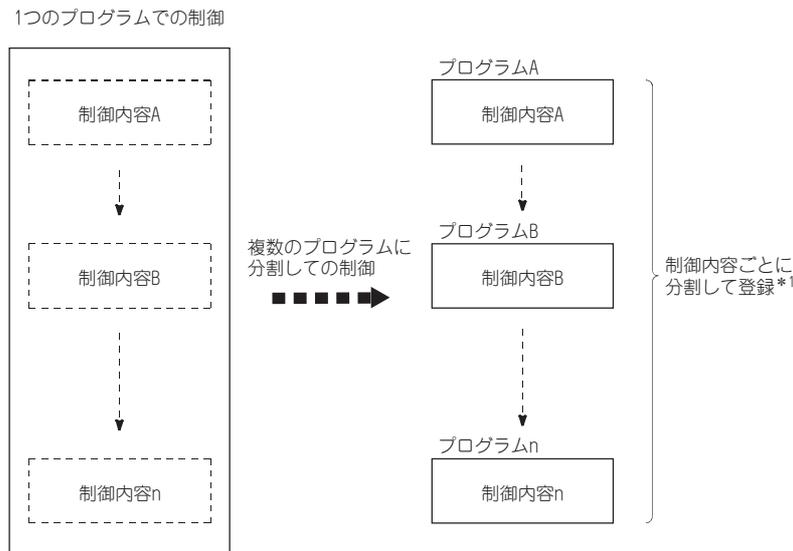


図 2.16 複数プログラムに分割した制御



注 2.3

Basic

ベーシックモデル QCPU では、複数のプログラムを実行できません。

(2) 複数のプログラムを実行する場合の設定

実行するプログラムのプログラム名（ファイル名）と実行条件の設定が必要です。
各設定は、PC パラメータのプログラム設定で行います。

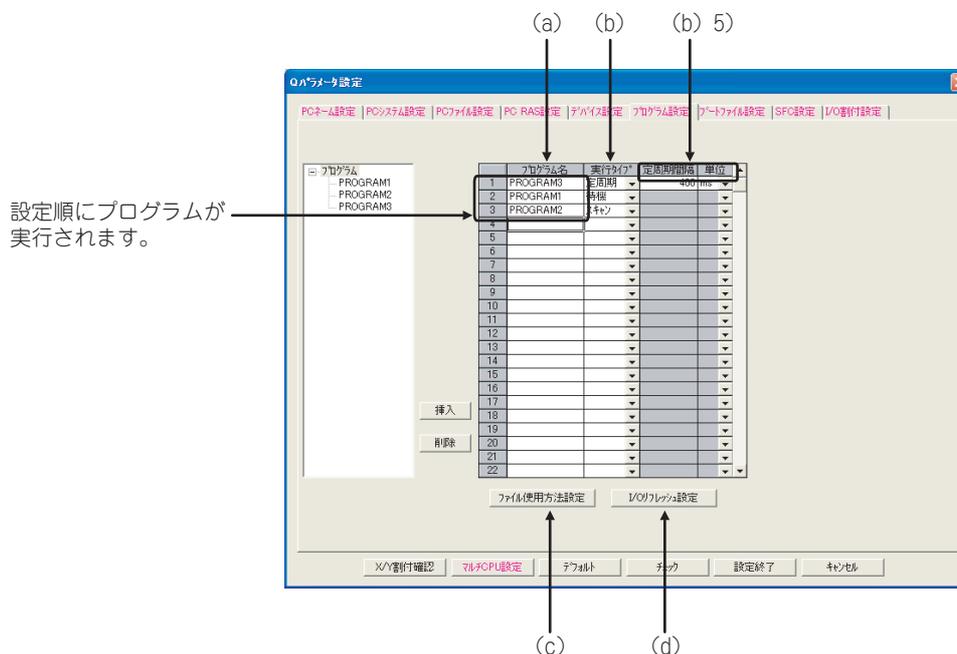


図 2.17 プログラム設定

(a) プログラム名

CPU ユニットで実行するプログラムのプログラム名（ファイル名）を設定します。

(b) 実行タイプ

プログラム名で設定したファイルの実行タイプを選択します。

CPU ユニットは、設定した実行タイプのプログラムを設定順に実行します。

1) 初期実行タイプ（“初期”）

電源 OFF → ON または STOP 状態から RUN 状態に切換え時に 1 回のみ実行されるプログラムです。

(☞ 2.3.1 項)

2) スキャン実行タイプ（“スキャン”）

初期実行タイププログラムを実行した次のスキャンから、1 スキャンに 1 回実行されるプログラムです。

(☞ 2.3.2 項)

3) 低速実行タイプ（“低速”） 注 2.4

コンスタントスキャン設定時または低速プログラム実行時間設定時のみ実行されるプログラムです。

(☞ 2.3.3 項)

4) 待機タイプ（“待機”）

実行要求があった場合のみ実行されるプログラムです。(☞ 2.3.4 項)



注 2.4

Redundant

二重化 CPU では、低速実行タイププログラムが使用できないため、実行タイプに“低速”を設定できません。

5) 定周期実行タイプ（“定周期”）

“定周期間隔”と“単位”で設定した時間ごとに実行されるプログラムです。（[図 2.3.5 項](#)）

- 定周期間隔
定周期実行タイププログラムの実行間隔を設定します。
定周期間隔で設定した単位により、設定範囲が異なります。
 - 単位が“ms”の場合：0.5～999.5ms（0.5ms 単位）
 - 単位が“s”の場合：1～60s（1s 単位）
- 単位
定周期間隔の単位（ms または s）を選択します。

(c) ファイル使用方法設定

PC パラメータの PC ファイル設定で設定したファイルレジスタ、デバイス初期値、デバイスコメント、ローカルデバイスのファイルを使用するかどうかをプログラムごとに設定します。



図 2.18 ファイル使用方法設定

デフォルトは，“PC ファイル設定に従う”（PC ファイル設定の設定をそのまま使用する）に設定されています。

ファイル使用方法設定を“使用しない”に設定すると、表 2.1 に示す処理になります。

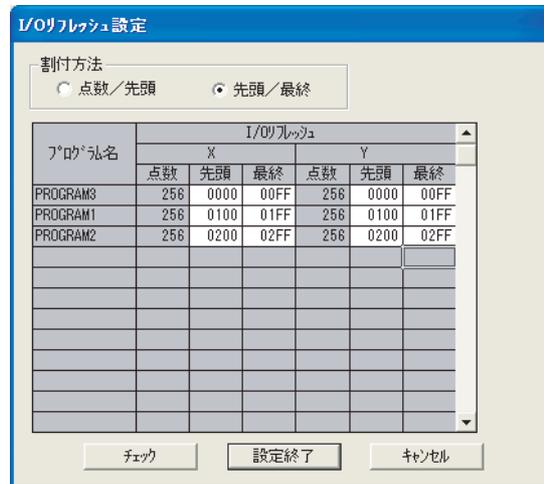
表 2.1 ファイル使用方法設定の項目と処理

設定項目	“使用しない”を選択時の処理
ファイルレジスタ	プログラムでのファイルレジスタが使用できません。
デバイス初期値	プログラムと同一ファイルのデバイス初期値の場合は、デバイス初期値をセットしません。
コメント	プログラムでのデバイスコメントが使用できません。
ローカルデバイス	プログラムの切換え時にローカルデバイスの退避と復帰を行いません。

(d) I/O リフレッシュ設定

CPU ユニットは、一括の I/O リフレッシュで入出力ユニット、インテリジェント機能ユニットの入出力の更新を行います。(☞ 3.8.1 項)

I/O リフレッシュ設定を行うと、設定したプログラムごとに指定範囲の I/O リフレッシュができます。



プログラム名	I/Oリフレッシュ					
	X			Y		
	点数	先頭	最終	点数	先頭	最終
PROGRAM3	256	0000	00FF	256	0000	00FF
PROGRAM1	256	0100	01FF	256	0100	01FF
PROGRAM2	256	0200	02FF	256	0200	02FF

図 2.19 I/O リフレッシュ設定

1) I/O リフレッシュ設定の用途

定周期実行タイププログラム実行前に使用する入力 (X) を取込み、定周期実行タイププログラムで ON/OFF した出力 (Y) の外部出力ができます。

Point

実行しているプログラム (定周期実行タイププログラムを除く) のスキャンタイムは、プログラム一覧モニタで確認できます。(☞ 6.13.1 項)

(3) CPU ユニットの各プログラムの流れ

CPU ユニットの電源 OFF → ON, または CPU ユニットの STOP 状態から RUN 状態にした場合の各プログラムの流れを, 図 2.20 に示します。

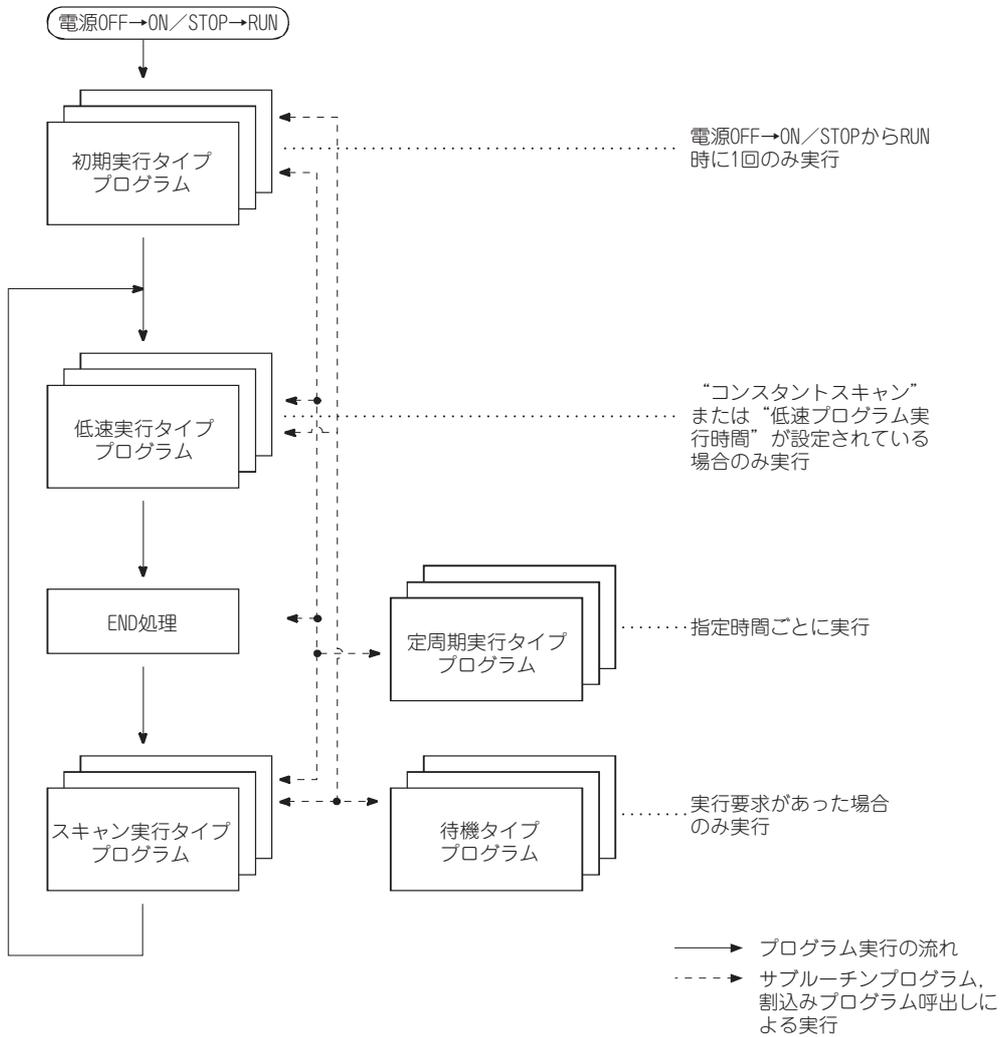


図 2.20 各プログラムの流れ

Point

初期実行タイププログラム, 待機タイププログラム, 定周期実行タイププログラムは必要に応じて使用してください。

2.3.1 初期実行タイププログラム

(1) 初期実行タイププログラムとは

CPU ユニットの電源 OFF → ON または STOP 状態から RUN 状態に切り換えたときに 1 回のみ実行されるプログラムです。

初期実行タイププログラムは、インテリジェント機能ユニットへのイニシャル処理のように、一度実行すれば次のスキャンから実行する必要のないプログラムに使用できます。

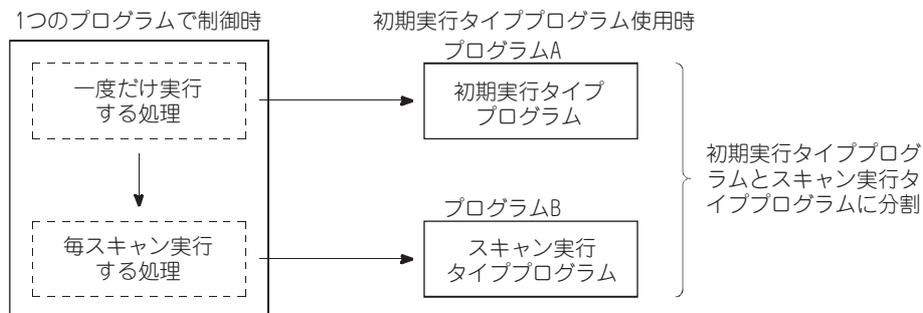


図 2.21 一度だけ実行する処理を初期実行タイププログラムとして分割した場合

(2) 初期実行タイププログラムの処理

(a) 実行順序

初期実行タイププログラムの実行がすべて完了すると END 処理を行い、次のスキャンからスキャン実行タイププログラムを実行します。

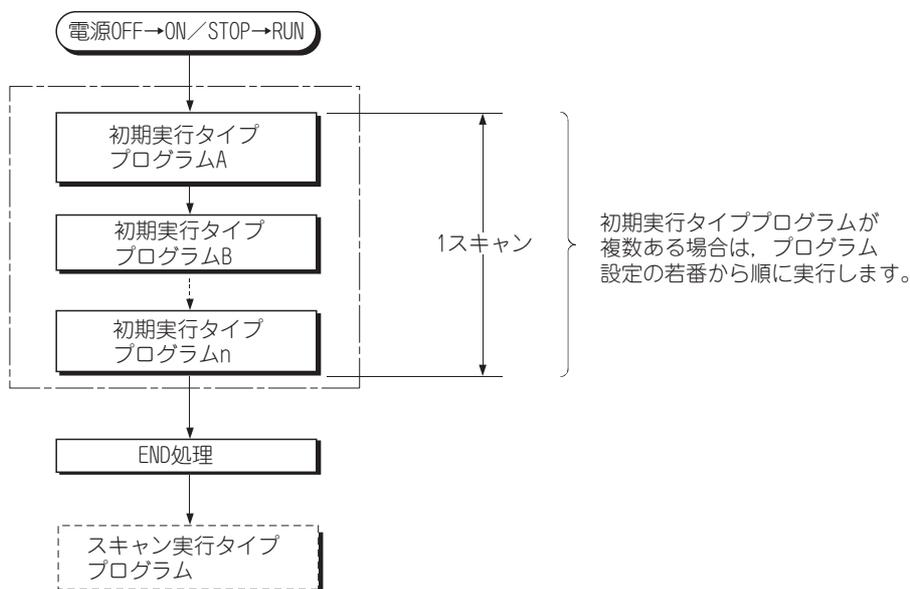


図 2.22 初期実行タイププログラムの実行順序

(b) 初期スキャンタイム

初期スキャンタイムは、初期実行タイププログラムの実行時間です。

複数の初期実行タイププログラムを実行した場合は、すべての初期実行タイププログラムの実行が完了するまでの時間です。

1) 初期スキャンタイムの格納場所

CPU ユニットは初期スキャンタイムを計測し、特殊レジスタ (SD522, SD523) に格納しています。SD522, SD523 をモニタすることにより、初期スキャンタイムの確認ができます。

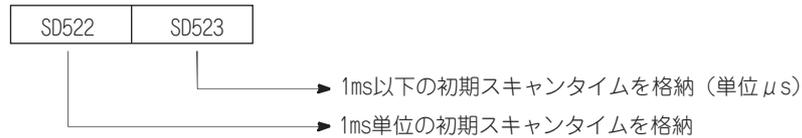


図 2.23 初期スキャンタイムの格納場所

例 SD522 が 3, SD523 が 400 の場合、初期スキャンタイムは 3.4ms となります。

2) 初期スキャンタイムの精度と計測

特殊レジスタに格納される初期スキャンタイムの精度は、 $\pm 0.1\text{ms}$ です。

また、初期スキャンタイムの計測は、シーケンスプログラムでウォッチドッグタイマリセット命令 (WDT) を実行しても継続されます。

3) 割り込み／定周期実行タイププログラムを実行した場合

初期実行タイププログラムの実行が完了する前に、割り込み／定周期実行タイププログラムを実行した場合は、初期実行タイププログラムの実行時間に割り込み／定周期実行タイププログラムの実行時間が加算されます。

(3) 初期実行タイププログラム作成上の注意事項

初期実行タイププログラムには、実行完了までに数スキャン必要な命令 (完了デバイスが存在する命令) を使用できません。

例 SEND, RECV 命令など

(4) 初期実行監視時間の設定

初期実行監視時間は、初期スキャンタイムを監視するタイマです。
PC パラメータの PC RAS 設定で設定します。
設定範囲は 10 ~ 2000ms です。(設定単位：10ms)
デフォルト値は設定されていません。

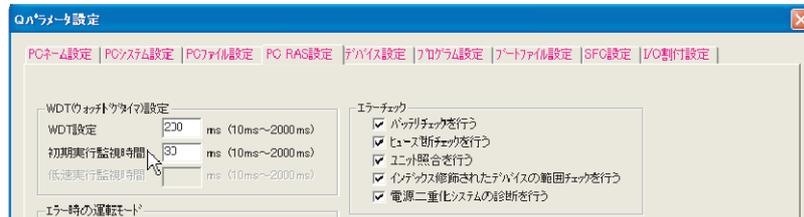


図 2.24 PC RAS 設定 (初期実行監視時間の設定)

(a) 初期実行監視時間を越えた場合

初期スキャンタイムが設定した初期実行監視時間を越えた場合は、“WDT ERROR” となり、CPU ユニッ
トは演算を停止します。

初期実行監視時間は、初期スキャンタイムの実行時間よりも長い時間を設定してください。

Point

- 初期実行タイププログラムと低速実行タイププログラムを実行する場合、初期実行タイププログラムの完了後に低速
実行タイププログラム (2.3.3 項) が実行されます。
初期実行監視時間は、初期スキャンタイムと低速実行タイププログラムの実行時間の合計値より長い時間を設定して
ください。
- 初期実行監視時間を設定した場合、計測値の誤差は 10ms です。
初期実行監視時間 (t) を 10ms に設定すると、初期スキャンタイムが $10\text{ms} < t < 20\text{ms}$ の範囲で “WDT
ERROR” になります。

2.3.2 スキャン実行タイププログラム

(1) スキャン実行タイププログラムとは

初期実行タイププログラムを実行した次のスキャンから、1スキャンに1回実行されるプログラムです。

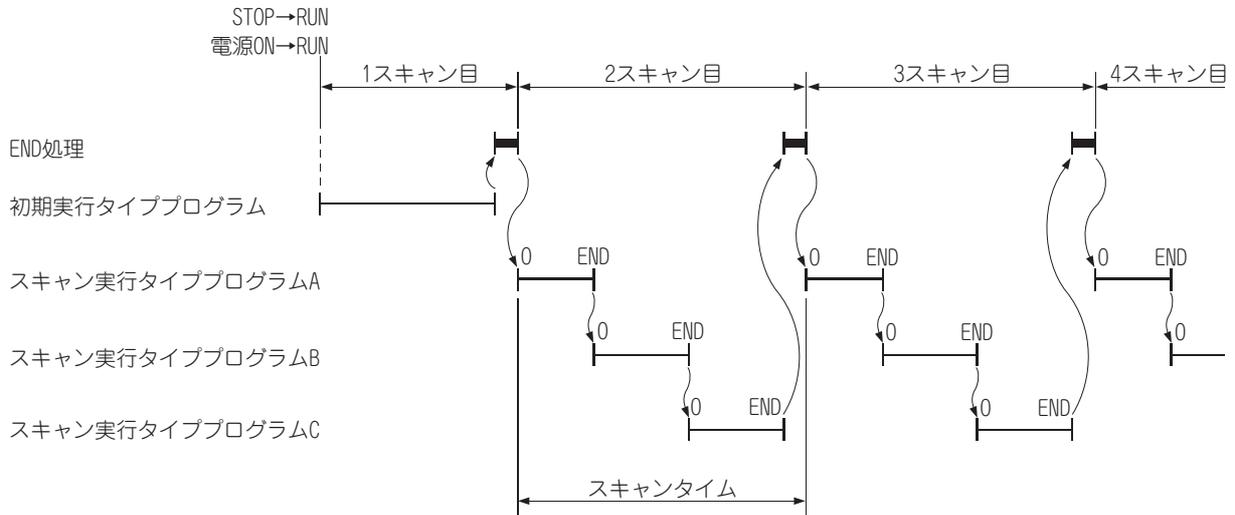


図 2.25 実行タイププログラムの実行順序

(2) スキャン実行タイププログラムの処理

複数のスキャン実行タイププログラムを実行した場合は、すべてのスキャン実行タイププログラムの実行が完了するまでの時間が、スキャン実行タイププログラムの実行時間になります。

割り込み/定周期実行タイププログラムを実行した場合は、各実行時間が加算された値がスキャンタイムになります。

2.3.3 低速実行タイププログラム 注 2.5

(1) 低速実行タイププログラムとは

コンスタントスキャン設定時または低速プログラム実行時間設定時のみ実行されるプログラムです。
プリンタへの出力など毎スキャン実行する必要のないプログラムに使用します。

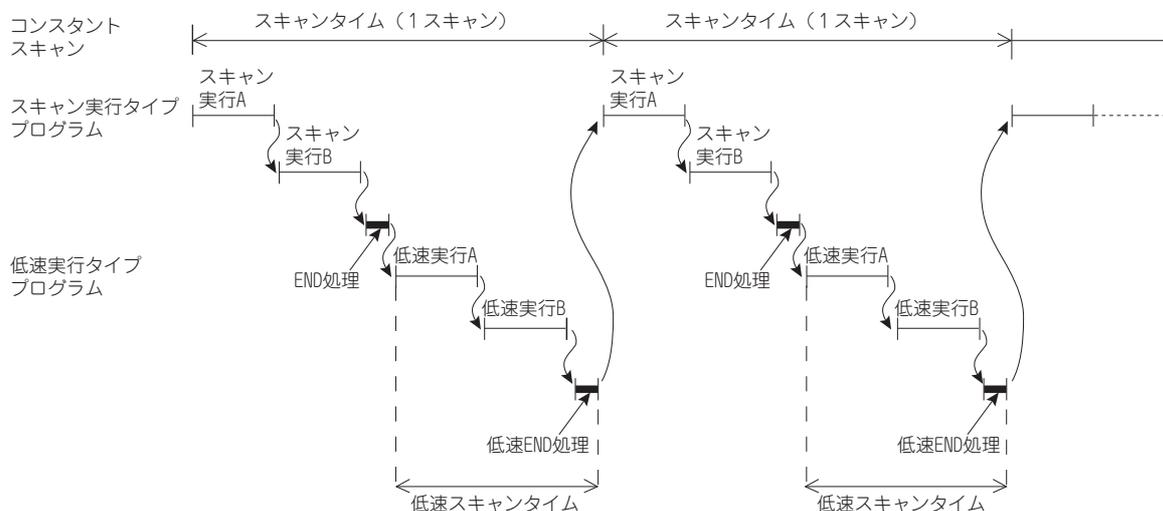


図 2.26 低速実行タイププログラムの実行 (コンスタントスキャンの余剰時間に実行した場合)

(2) 低速実行タイププログラムの処理

(a) 実行動作

PC パラメータの PC RAS 設定の設定 ( 本項(4)) により、実行動作が異なります。
必要に応じて使い分けてください。

1) スキャンタイムを一定にし、制御の精度を優先する場合

コンスタントスキャンを設定します。

2) 実行時間を確保し、低速実行タイププログラムを確実に実行させる場合

低速プログラム実行時間を設定します。



注 2.5

Redundant

二重化 CPU では、低速実行タイププログラムは使用できません。

- (b) 1 スキャン内ですべての低速実行タイププログラムの演算が完了して余剰時間がある場合
 低速実行タイププログラムの演算完了後の処理は、特殊リレー SM330 の ON/OFF 状態および低速実行タイププログラムの実行条件により異なります。

1) 非同期方式 (SM330 = OFF)

余剰時間内で低速実行タイププログラムの演算を続けて行う方式です。

2) 同期方式 (SM330 = ON)

余剰時間があっても低速実行タイププログラムの演算を続けて行わずに、次のスキャンから演算を行う方式です。

表 2.2 動作方式と設定の組合せ

低速実行タイププログラムの動作方式	SM330 の設定状態	低速実行タイププログラムの実行条件	
		コンスタントスキャン設定時	低速プログラム実行時間設定時
非同期方式	OFF	低速実行タイププログラムを再実行* ¹	低速実行タイププログラムを再実行* ²
同期方式	ON	コンスタントスキャンの待ち時間が発生* ³	スキャン実行タイププログラムの演算を開始* ⁴

- * 1: 低速実行タイププログラムは、コンスタントスキャンの余剰時間分繰返し実行されます。
 このため、スキャンごとに低速実行タイププログラムの実行時間が異なります。(図 2.27)
- * 2: 低速実行タイププログラムは、設定されている低速プログラム実行時間分繰返し実行されます。
 このため、スキャンタイムはスキャンごとに異なります。(図 2.29)
- * 3: 低速 END 処理完了後の余剰時間が待ち時間となります。
 設定されているコンスタントスキャンになると、スキャン実行タイププログラムが実行されます。
 このため、スキャンごとのスキャンタイムは一定です。(図 2.28)
- * 4: 低速 END 処理完了後の余剰時間を無視して、スキャン実行タイプのプログラムの演算を開始します。
 このためスキャンタイムはスキャンごとに異なります。(図 2.30)

《コンスタントスキャン設定時》

下記条件で低速実行タイププログラムを実行した場合の動作を示します。

- コンスタントスキャン：8ms
- スキャン実行タイププログラムの合計：4ms～5ms
- 低速実行タイププログラム A の実行時間：1ms
- 低速実行タイププログラム B の実行時間：3ms
- END 処理／低速 END 処理：0ms（説明を簡単にするため 0ms と仮定する）

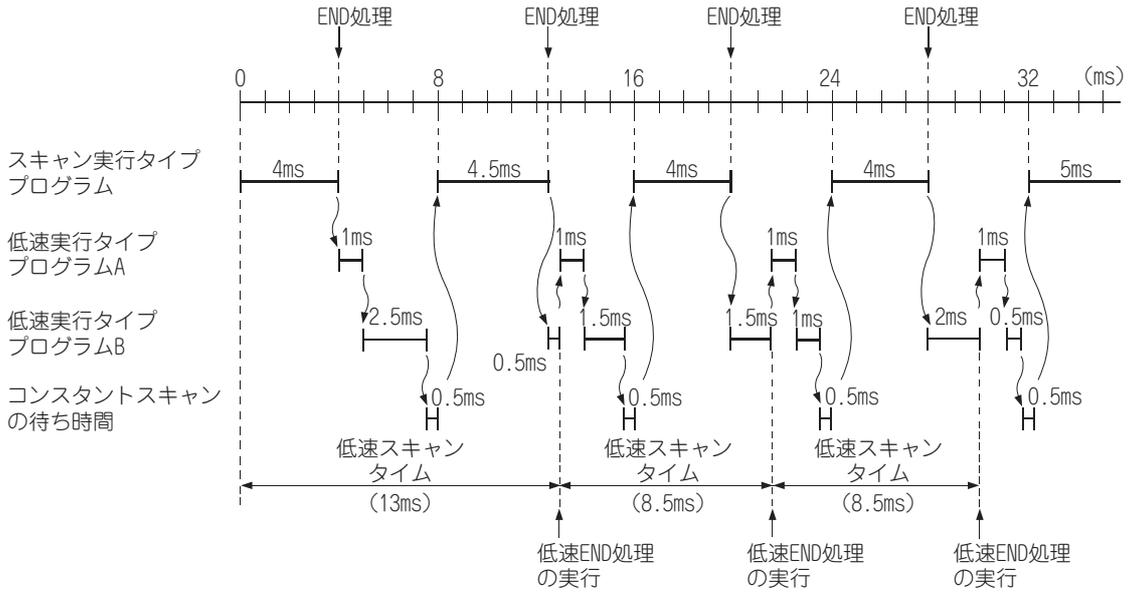


図 2.27 非同期方式 (SM330 : OFF)

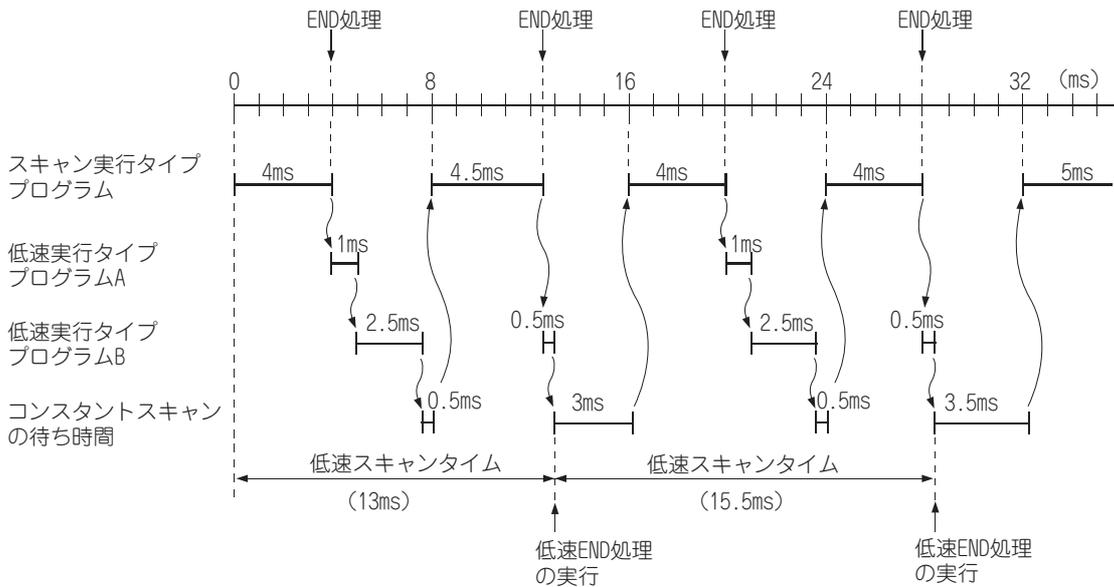


図 2.28 同期方式 (SM330 : ON)

《低速プログラム実行時間設定時》

下記条件で低速実行タイププログラムを実行した場合の動作を示します。

- 低速プログラム実行時間：3ms
- スキャン実行タイププログラムの合計：4ms～5ms
- 低速実行タイププログラム A の実行時間：1ms
- 低速実行タイププログラム B の実行時間：3ms
- END 処理／低速 END 処理：0ms（説明を簡単にするため 0ms と仮定する）

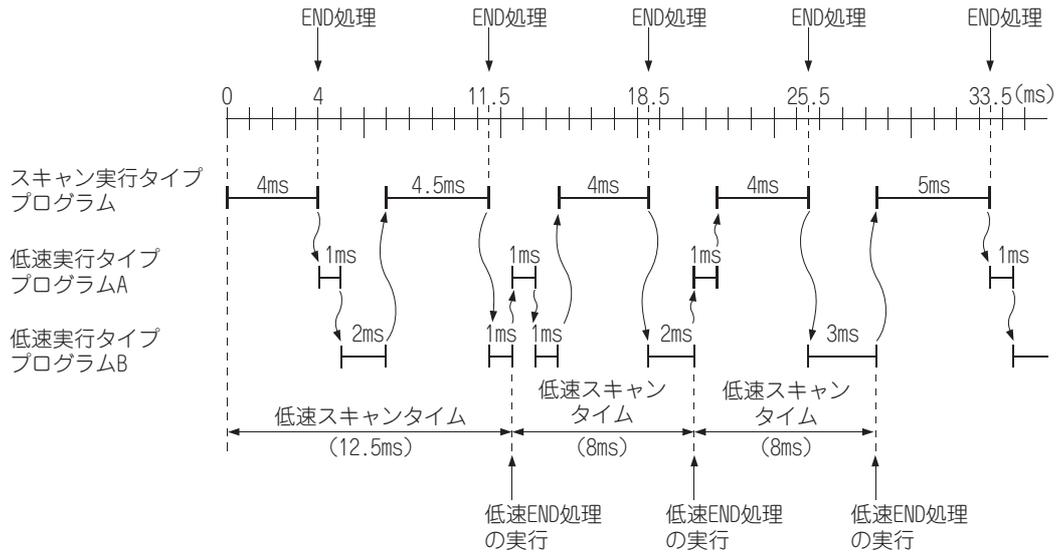


図 2.29 非同期方式 (SM330 : OFF)

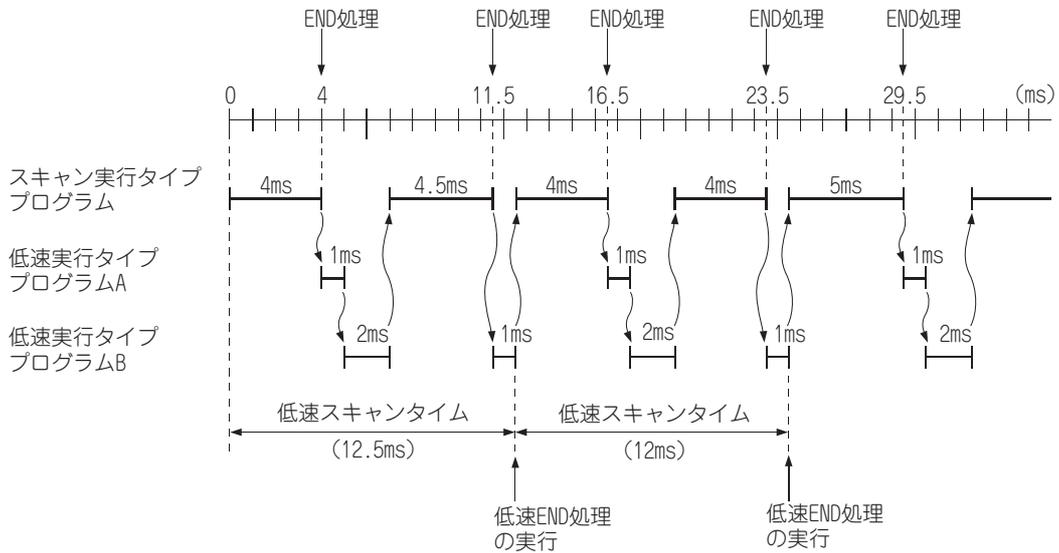


図 2.30 同期方式 (SM330 : ON)

2.3 複数のプログラムに分割した場合の設定
2.3.3 低速実行タイププログラム

(c) コンスタントスキヤンの余剰時間または低速プログラム実行時間内に低速実行タイププログラムが処理できなかった場合

プログラムの実行は一旦中断し、次のスキヤンに残りのプログラムが実行されます。

(d) 低速 END 処理

低速実行タイププログラムをすべて実行すると、低速 END 処理を行います。

低速 END 処理では、下記処理を行います。

- ・ 低速実行タイププログラム用特殊リレー／特殊レジスタのセット* 1
- ・ 低速実行タイププログラムの RUN 中書込み
- ・ 低速スキヤンタイムの計測
- ・ 低速実行タイププログラムのウォッチドッグタイマのリセット

低速 END 処理が終了すると、特殊リレー (SM330) が OFF でコンスタントスキヤンの余剰時間がある場合、再度先頭の低速実行タイププログラムを実行します。

* 1：低速実行タイププログラム用の特殊リレー／特殊レジスタは下記のとおりです。

- ・ SM330, SM404, SM405, SM510
- ・ SD430, SD510, SD528 ~ 535, SD544 ~ 547

Point

低速実行タイププログラム実行時は、実行している命令の最大処理時間+低速 END 処理時間分、コンスタントスキヤンがずれることがあります。

備考

低速 END 処理と END 処理の違いについては、図 2.26 を参照してください。

(e) 低速スキャンタイム

すべての低速実行タイププログラムの実行が完了するまでの時間と低速 END 処理時間の合計時間です。低速スキャンタイムとスキャンタイムの違いについては、図 2.31 を参照してください。

1) 低速スキャンタイムの格納場所

CPU ユニットは、低速スキャンタイムを計測し、特殊レジスタ (SD528 ~ 535) に格納しています。SD528 ~ 535 をモニタすることにより、低速スキャンタイムの確認ができます。

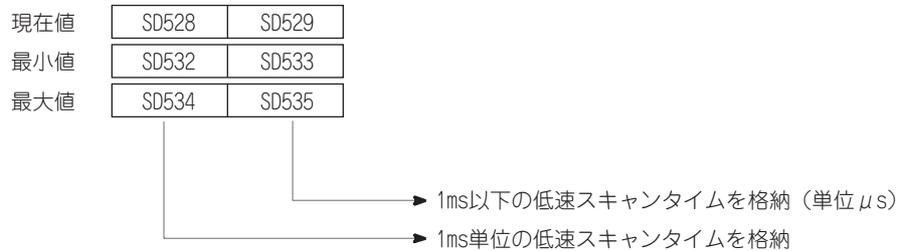


図 2.31 低速スキャンタイムの格納場所

例 SD528 が 50, SD529 が 400 の場合、低速スキャンタイムは 50.4ms になります。

備考

低速スキャンタイム、プログラムの実行時間などはプログラム一覧モニタで確認できます。(☞ 6.13.1 項)

2) 低速スキャンタイムの精度と計測

特殊レジスタに格納される各低速スキャンタイムの精度は、 $\pm 0.1\text{ms}$ です。

また、シーケンスプログラムでウォッチドッグタイマリセット命令 (WDT) を実行しても、各低速スキャンタイムの計測は継続されます。

3) 割り込み／定周期実行タイププログラムを実行した場合

低速スキャンタイムは、割り込み／定周期実行タイププログラムの実行時間が加算された値になります。

(3) 低速実行タイププログラム作成上の注意事項

(a) 低速プログラム実行時間設定について

低速プログラム実行時間を設定すると、設定分だけ低速実行タイププログラムの実行時間が確保されるため、スキャンタイムが延びます。

スキャンタイムが、WDT より小さくなるように低速プログラム実行時間を設定、または WDT の設定値を大きくしてください。

(b) 使用できない命令について

低速実行タイププログラムでは、COM 命令が使用できません。

(c) 実行タイミングについて

低速実行タイププログラムは、初期実行タイププログラムを実行したスキャンでも実行します。

初期実行タイププログラムを実行後に、低速実行タイププログラムを実行させない場合は、特殊リレー (SM402, SM403) でインタロックをとってください。

(d) 設定範囲について

コンスタントスキャンおよび低速プログラム実行時間は、いずれか一方のみ設定してください。

コンスタントスキャンと低速プログラム実行時間が設定されている場合、(コンスタントスキャンの余剰時間) < (低速プログラム実行時間) になると、“PRG. TIME OVER” (エラーコード : 5010) になります。

(e) プログラムの切換え時の処理

プログラムを切り換えたときのインデックスレジスタの退避と復帰については、9.6.3 項を参照してください。

(4) 低速実行タイププログラム実行時の設定

PC パラメータの PC RAS 設定で下記を設定します。

- コンスタントスキャン…設定範囲：0.5ms ～ 2000ms（設定単位：0.5ms）
- 低速プログラム実行時間…設定範囲：1ms ～ 2000ms（設定単位：1ms）



図 2.32 PC RAS 設定（コンスタントスキャン，低速プログラム実行時間，低速実行監視時間）

Point

低速実行タイププログラムを実行する場合は、コンスタントスキャンまたは低速プログラム実行時間のいずれか一方を必ず設定してください。

また、低速実行タイププログラムの実行時間を監視する場合は、低速実行監視時間を設定します。設定範囲は 10 ～ 2000ms です。（設定単位：10ms）
デフォルト値は設定されていません。（デフォルト値はありません。）

(a) 低速実行監視時間を超えた場合

低速スキャンタイムが設定した低速実行監視時間を超えた場合は、“PRG TIME OVER”（エラーコード：5010）になります。

Point

低速実行タイププログラムと初期実行タイププログラムを実行する場合、初期実行タイププログラム（[図 2.3.1](#) 項）の完了後に低速実行タイププログラムが実行されます。
初期実行監視時間は、初期スキャンタイムと低速実行タイププログラムの実行時間の合計値よりも長い時間を設定してください。

2.3.4 待機タイププログラム

(1) 待機タイププログラムとは

実行要求のあった場合のみ実行されるプログラムです。

また、シーケンスプログラムの命令により待機タイプを他の実行タイプに切り換えて使用できます。

(2) 待機タイププログラムの用途

(a) プログラムのライブラリ化

サブルーチンプログラム、割込みプログラムを待機タイププログラムにして、メインルーチンプログラムと別管理する場合に使用します。

1つの待機タイププログラムには、サブルーチンプログラム、割込みプログラムを複数作成できます。

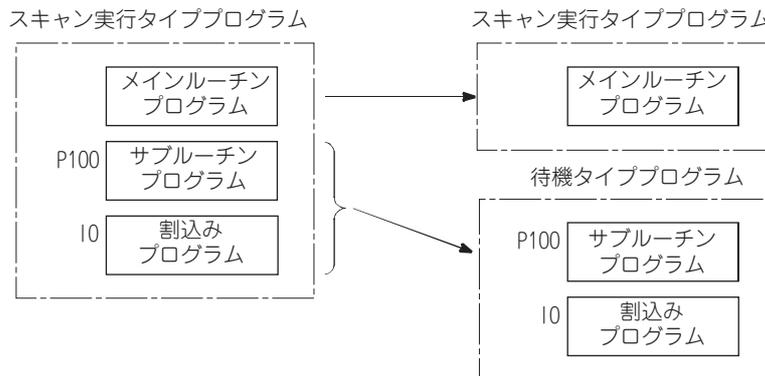


図 2.33 待機タイププログラムを使用したプログラムのライブラリ化

(b) プログラムの段取り替え

すべてのシステムに対応するプログラムをあらかじめ作成しておき、必要なプログラムのみ実行させる場合に使用します。

例えば、あらかじめ PC パラメータで待機タイプに設定したプログラムは、シーケンスプログラムでスキャン実行タイププログラムに変更し、実行できます。

(3) 待機タイププログラムの実行方法

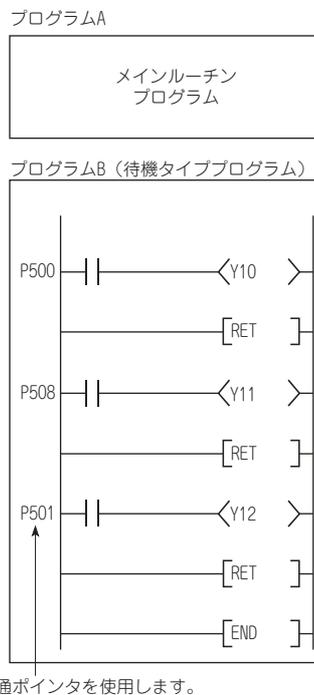
待機タイププログラムは、下記に示す方法で実行できます。

- 待機タイププログラム内にサブルーチンプログラム、割込みプログラムを作成し、割込み発生時またはポインタなどで呼び出して実行する方法
- プログラムの実行タイプを切り換える命令を使用し、待機タイププログラムを他の実行タイプに切り換えて実行する方法

(a) 待機タイププログラム内に、サブルーチンプログラムまたは割込みプログラムを1つにまとめる場合

サブルーチンプログラムまたは割込みプログラムを1つの待機タイププログラムに作成する場合は、ステップ0から作成します。

メインルーチンプログラムのあとには、サブルーチンプログラム、割込みプログラムを作成する際に使用する FEND 命令は必要ありません。



共通ポインタを使用します。

図 2.34 待機タイププログラムにサブルーチンプログラムをまとめる場合

1) 待機タイププログラム内のサブルーチンプログラム、割り込みプログラムを実行した場合の動作

待機タイププログラムの実行を終了すると、待機タイププログラム内のプログラムをコールしたプログラムを再開します。

待機タイププログラム内のサブルーチンプログラム、割り込みプログラムを実行した場合の動作を図 2.35 に示します。

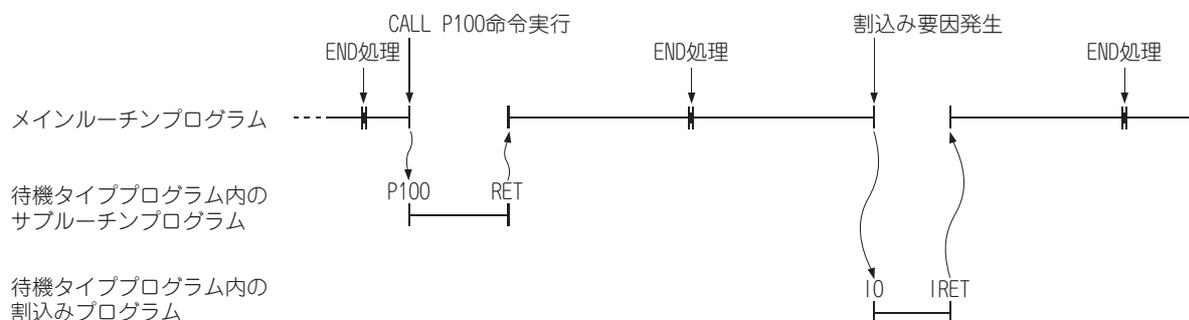


図 2.35 待機タイププログラム内のサブルーチンプログラム、割り込みプログラムを実行した場合の動作

Point

- サブルーチンプログラム、割り込みプログラム作成時の制約については、下記を参照してください。
 - ・ サブルーチンプログラム : [2.2.2 項](#)
 - ・ 割り込みプログラム : [2.2.3 項](#)
- ポインタは、共通ポインタを使用してください。([9.9.2 項](#))
ローカルポインタを使用した場合、他のプログラムから待機タイププログラム内のサブルーチンプログラムは実行できません。

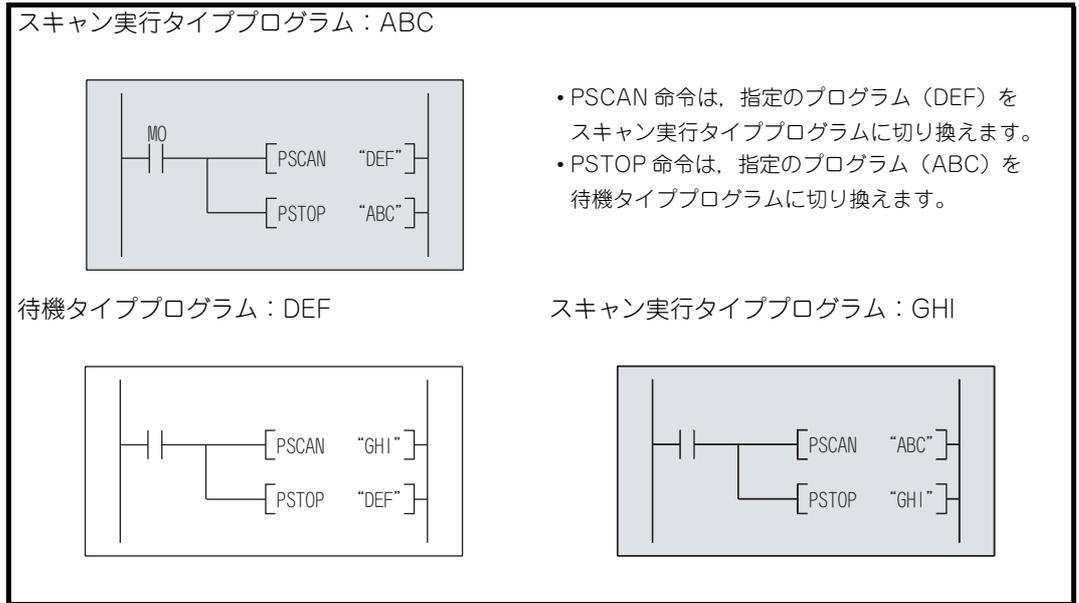
(b) 命令で実行タイプを切り換える場合

実行タイプの切り換えは、PSCAN 命令、PSTOP 命令、POFF 命令で行います。

1) スキャン実行タイププログラムにおいて実行タイプを切り換える例

- プログラム “ABC” と “GHI” は、スキャン実行タイププログラムに設定し、プログラム “DEF” は、待機タイププログラムに設定する場合
- 条件が成立 (図 2.36 にある内部リレー (M0) が ON) したときに、プログラム “DEF” をスキャン実行タイププログラムに切り換え、プログラム “ABC” を待機タイププログラムに切り換える場合

【PSCAN,PSTOP 命令実行前】



↓ MOをONした場合

【PSCAN,PSTOP 命令実行後】

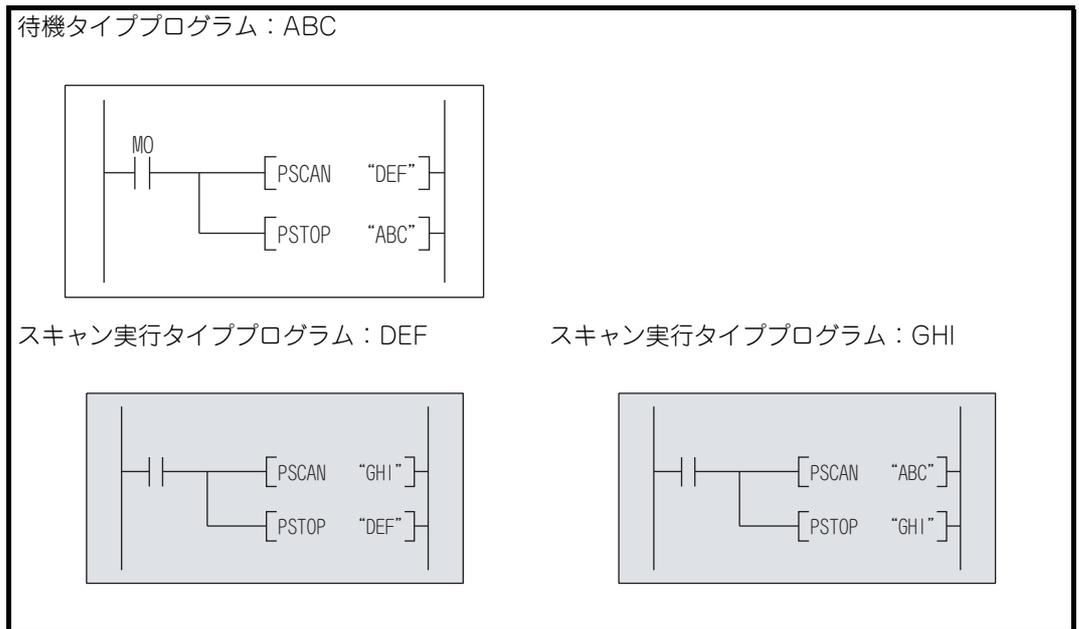


図 2.36 スキャン実行タイププログラムにおいて実行タイプを切り換える例

2.3 複数のプログラムに分割した場合の設定
2.3.4 待機タイププログラム

2) 実行タイプの切換えタイミング

プログラムの実行タイプの切換えは、END 処理で行います。

このためプログラムの実行途中に、プログラムの実行タイプは切り換わりません。

また、同一スキャンで同一プログラムに異なるタイプを設定した場合は、あとで実行した実行タイプの切換え命令の実行タイプになります。

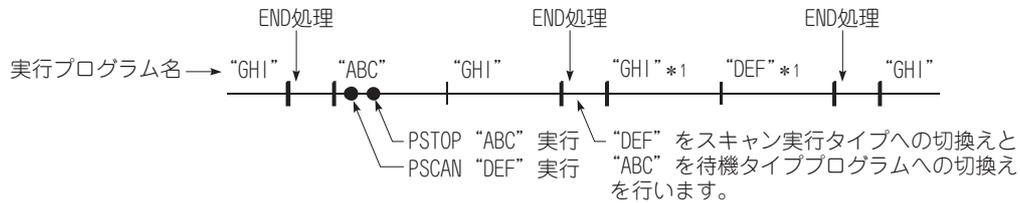


図 2.37 実行タイプの切換えタイミング

* 1: "GHI" と "DEF" のプログラムの実行順序は、PC パラメータのプログラム設定で設定した順になります

(4) 待機タイププログラム作成上の注意事項

(a) 使用できないデバイスについて

使用できないデバイスは、プログラムの種類（サブルーチンプログラム、割込みプログラム）、または命令により切り換える実行タイプに依存します。

(b) ローカルデバイスの使用について

ローカルデバイスを使用したサブルーチンプログラムの実行については、9.13.2 項を参照してください。

2.3.5 定周期実行タイププログラム

(1) 定周期実行タイププログラムとは

指定時間ごとに実行される割り込みプログラムです。

ただし、割り込みプログラムとは違い、割り込みポインタ、IRET 命令を記述せずにファイル単位での割り込みが可能です。

プログラム作成上の制約については、2.2.3 項 (2) (b) を参照してください。

(参照先の“割り込みプログラム”を“定周期実行タイププログラム”に読み替えてください。)

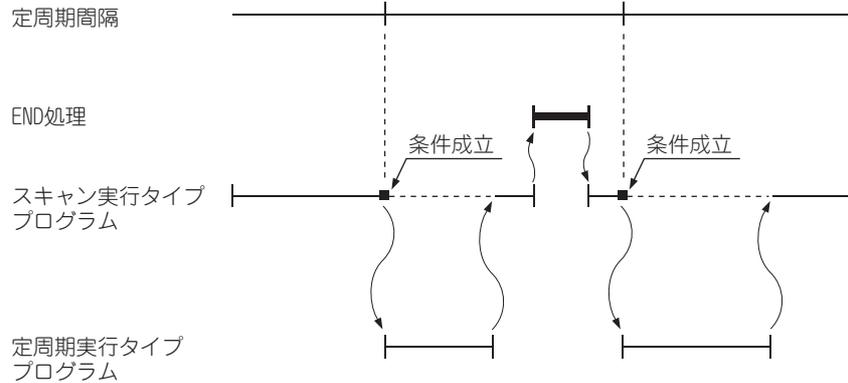


図 2.38 定周期実行タイププログラムの実行

Point

定周期実行タイププログラムを実行する場合は、初期実行タイプ/スキャン実行タイププログラムにおいて、EI 命令により割り込み許可状態にしてください。

(2) 定周期実行タイププログラムの処理

(a) 複数の定周期実行タイププログラムがある場合

定周期実行タイププログラムは、指定時間ごとに実行されます。

複数の定周期実行タイププログラムが、同一タイミングで指定時間に達した場合は、PC パラメータのプログラム設定の若番から順に実行されます。

(b) 定周期実行タイププログラムと割り込みプログラムがある場合

定周期実行タイププログラムと割り込みプログラム（I28～31）が、同一タイミングで指定時間に達した場合は、割り込みプログラムを優先して実行します。

(c) リンクリフレッシュ中に実行条件が成立した場合

リンクリフレッシュを中断し、定周期実行タイププログラムを実行します。

CC-Link IE コントローラネットワークまたは、MELSECNET/H でサイクリックデータの局単位ブロック保証を行っても、定周期実行タイププログラムでリフレッシュ先に設定しているデバイスを使用すると、サイクリックデータの局単位ブロック保証ができません。

定周期実行タイププログラムでは、リフレッシュ先のデバイスを使用しないでください。

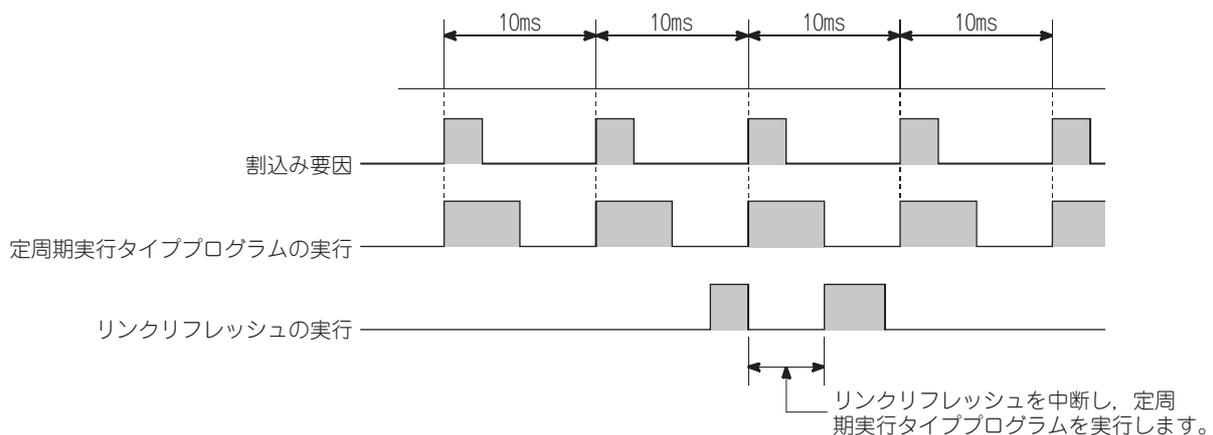


図 2.39 リンクリフレッシュ中に定周期実行タイププログラムが実行された場合

備考

サイクリックデータの局単位ブロック保証については、下記マニュアルを参照してください。

各ネットワークユニットのマニュアル

(d) END 処理中に実行条件が成立した場合

コンスタントスキャン実行時、および END 命令の待ち時間中に定周期実行タイププログラムの実行条件が成立した場合は、定周期実行タイププログラムを実行します。

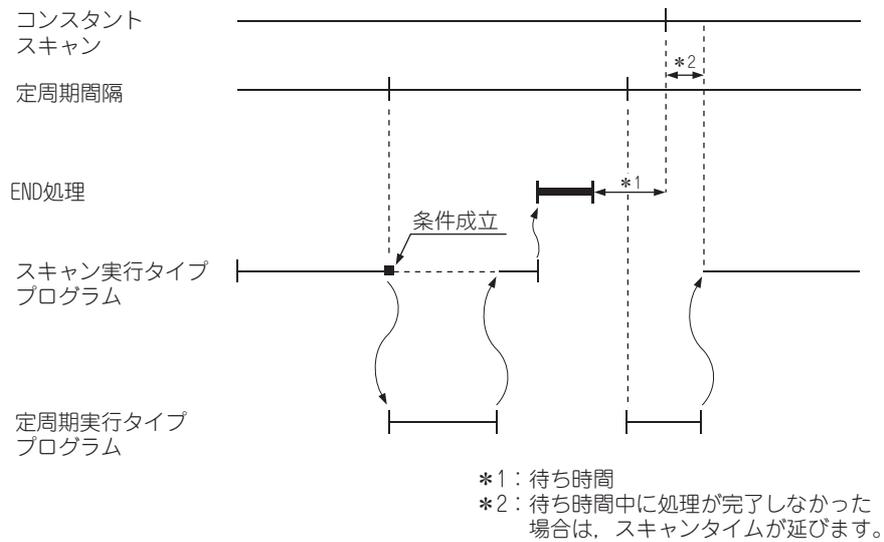


図 2.40 待ち時間中の定周期実行タイププログラムの実行

(3) プログラム切換え時の処理

プログラムを切り換えたときのインデックスレジスタの退避と復帰については、2.2.3 項 (4) を参照してください。(参照先の“割り込みプログラム”を“定周期実行タイププログラム”に読み替えてください。)

(4) 注意事項

(a) 定周期実行プログラムの実行間隔について

DI 命令による割込み禁止中の時間（割込み禁止時間）によっては，定周期実行プログラムの実行間隔が設定時間よりも遅くなる場合がありますので注意してください。

DI 命令による割込み禁止時間が長くなる場合には，定周期実行タイプを使用するのではなく，定周期割込み (I28 ~ I31) による割込みプログラムを使用してください。

$$\text{定周期実行間隔の最大公約数} * 1 < \text{割込み禁止時間} \cdots \text{式①}$$

* 1：定周期実行間隔の最大公約数とは，複数の定周期実行プログラムに設定されている実行間隔設定値の最大公約数のことです。

式①が成立していると，定周期実行間隔に設定した間隔より実際の定周期実行タイププログラムの実行間隔は，下式に示す時間分延びる場合があります。

$$\frac{\text{割込み禁止時間}}{\text{定周期実行間隔の最大公約数}} \times \text{該当プログラムの定周期実行間隔設定値}$$

定周期実行タイププログラムの実行時間の延び時間を，下記例に示します。

- 例**
- 定周期実行間隔 … 10ms, 5ms, 1ms, 0.5ms
 - 定周期実行間隔の最大公約数 … 0.5ms
 - 割込み禁止時間 (DI) … 5ms (割込み許可時間 (EI) … 0.5ms 未満)

上記設定の場合，式①は $0.5\text{ms} < 5\text{ms}$ となります。

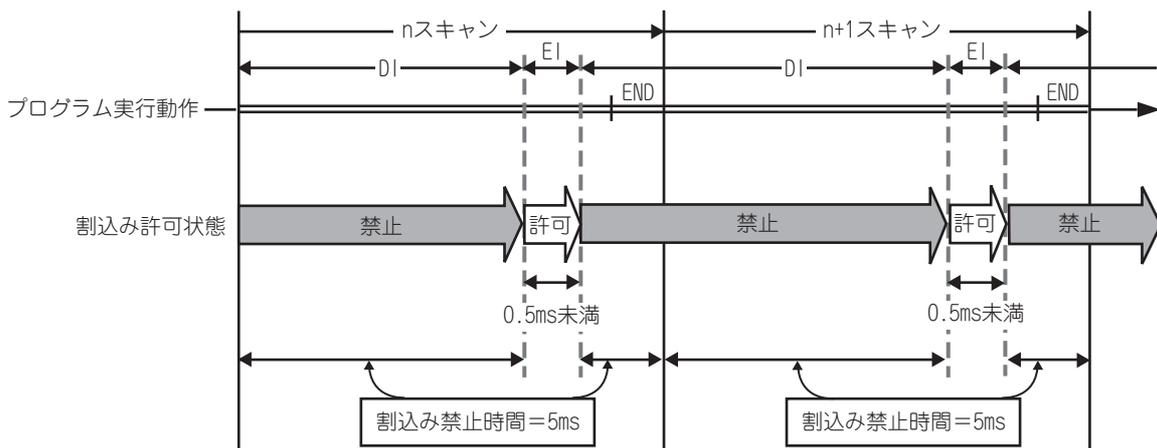


図 2.41 プログラムの実行動作と割込み許可状態

よって，10ms 設定の定周期実行タイププログラムの実行時間は， $5 \div 0.5 \times 10 = 100$ となるため，最大 100ms 延びます。

2.3.6 実行タイプの切換え例

(1) 命令による実行タイプの切換え例

(a) 実行タイプ切換えの命令

実行タイプは、命令を使用することによりシーケンスプログラム実行中でも変更できます。

実行タイプの変更は、PSCAN 命令、PLOW 命令 **注 2.6**、PSTOP 命令、POFF 命令で行います。

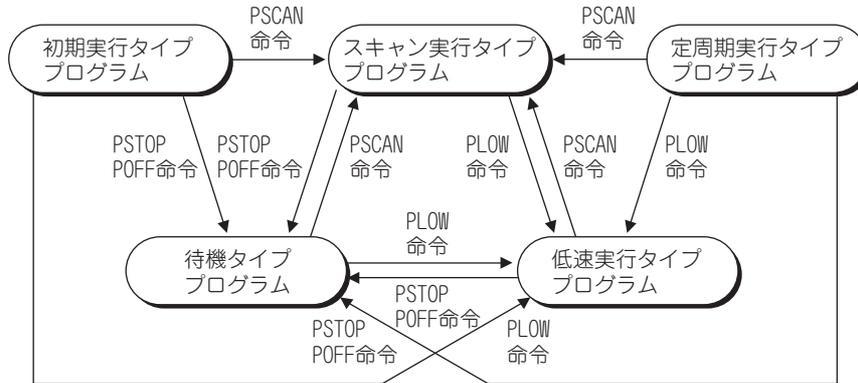


図 2.42 命令による実行タイプ切換えのパターン

表 2.3 命令による実行タイプ切換えのタイミング

変更前の実行タイプ	実行命令			
	PSCAN	PSTOP	POFF	PLOW
スキャン実行タイプ	スキャン実行タイプのまま変化しない	待機タイプになる	次のスキャンで出力を OFF する	低速実行タイプになる
初期実行タイプ	スキャン実行タイプになる	待機タイプになる	その次のスキャン以降待機タイプになる	
待機タイプ		待機タイプのまま変化しない	無処理	
低速実行タイプ	低速実行タイプの実行を中断し、次のスキャンからスキャン実行タイプになる(ステップ0から実行)	低速実行タイプの実行を中断し、次のスキャン以降待機タイプになる	低速実行タイプの実行を中断し、次のスキャンで出力を OFF する その次のスキャン以降待機タイプになる	低速実行タイプのまま変化しない
定周期実行タイプ	スキャン実行タイプになる	待機タイプになる	次のスキャンで出力を OFF する その次のスキャン以降待機タイプになる	低速実行タイプになる

Point

定周期実行タイププログラムを他の実行タイプに変更すると、定周期実行タイプに戻すことはできません。

2.3 複数のプログラムに分割した場合の設定
2.3.6 実行タイプの切換え例

注 2.6 **Redundant**

二重化 CPU では、低速実行タイププログラムは使用できません。

(b) 実行タイプの切換え例

管理プログラムにおいて、設定した条件に合った待機タイププログラムを、スキャン実行タイププログラムに変更して実行します。

使用しないスキャン実行タイププログラムは、待機タイププログラムに変更することも可能です。

管理プログラムで“ABC”、“DEF”、“GHI”、“JKL”の待機タイププログラムの実行タイプを切り換える場合の例を図 2.43 に示します。

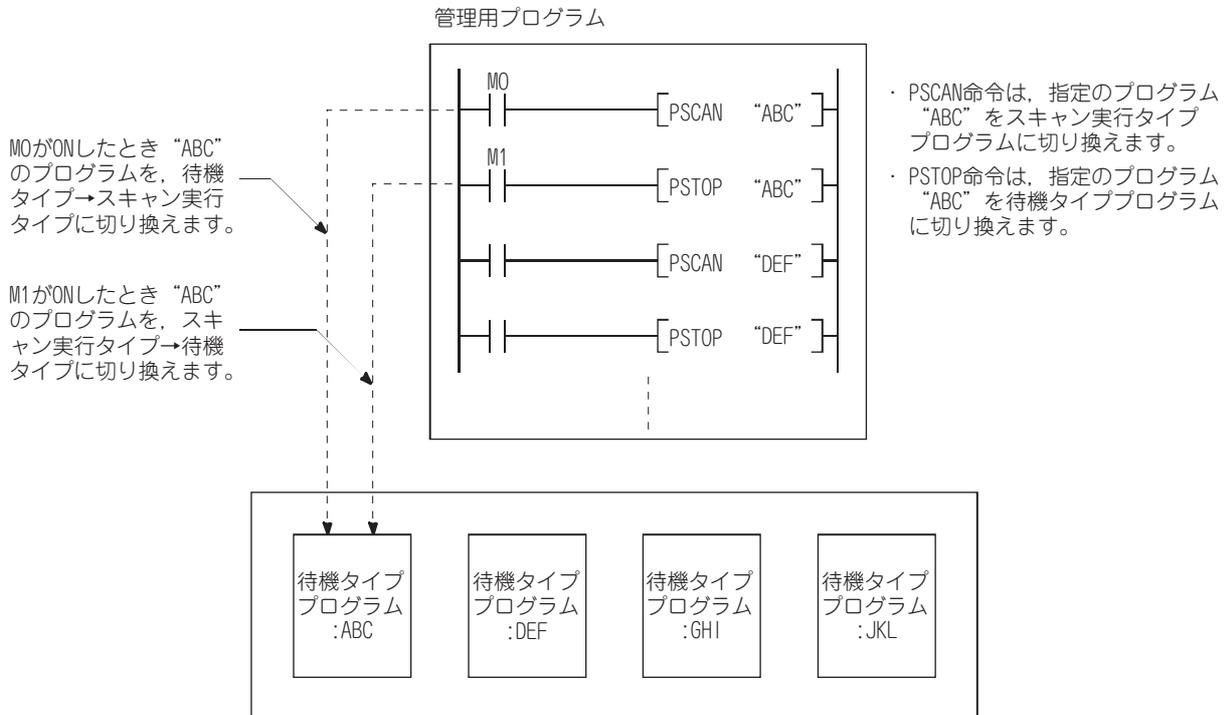


図 2.43 命令による実行タイプの切換え例

(2) プログラム一覧モニタによる実行タイプの切換え

GX Developer の [オンライン] → [モニタ] → [プログラム一覧モニタ] から切り換えることができます。

(☞ 6.13.1 項)

2.4 シーケンスプログラムで使用できるデータ

CPU ユニットでは、数値・アルファベットなどのデータを、0 (OFF) と 1 (ON) の2つの状態で表現しています。

この0と1で表現したデータをBIN (2進数) といいます。

また、BIN データを4ビットごとにまとめて表現するHEX (16進数)、BCD (2進化10進数) や実数を使用することもできます。

BIN, HEX, BCD, DEC (10進数) の数値表現を表 2.4 に示します。

表 2.4 BIN, HEX, BCD, DEC の数値表現

DEC (10進数)	HEX (16進数)	BIN (2進数)				BCD (2進化10進数)			
0	0				0				0
1	1				1				1
2	2				10				10
3	3				11				11
·	·				·				·
·	·				·				·
·	·				·				·
9	9				1001				1001
10	A				1010			1	0000
11	B				1011			1	0001
12	C				1100			1	0010
13	D				1101			1	0011
14	E				1110			1	0100
15	F				1111			1	0101
16	10			1	0000			1	0110
17	11			1	0001			1	0111
·	·				·				·
·	·				·				·
·	·				·				·
47	2F			10	1111			100	0111
·	·								
·	·								
·	·								
32766	7FFE	0111	1111	1111	1110	—			
32767	7FFF	0111	1111	1111	1111	—			
-32768	8000	1000	0000	0000	0000	1000	0000	0000	0000
-32767	8001	1000	0000	0000	0001	1000	0000	0000	0001
·	·								
·	·								
·	·								
-2	FFFE	1111	1111	1111	1110	—			
-1	FFFF	1111	1111	1111	1111	—			

(1) 外部から CPU ユニットへの数値の入力

外部からデジタルスイッチなどで CPU ユニットに数値を設定する場合は、下記 (b) に示す方法により、BCD (2 進化 10 進数) で DEC (10 進数) と同一の設定が可能です。

(a) CPU ユニット内部で扱われる数値

CPU ユニットは、BIN (2 進数) で演算を行っています。

BCD で設定した値をそのまま使用すると、CPU ユニットは設定した値を BIN として演算を行います。

このため、設定した値とは異なる値で演算が行われます。(☞ 本項(1)(b))

(b) BIN を意識することなく数値を入力する方法

BCD で設定したデータを CPU ユニットで使用する BIN に変換するためには、BIN 命令を使用します。

BIN 命令を使用すると、外部からの数値データ設定が BIN を意識することなく行えます。

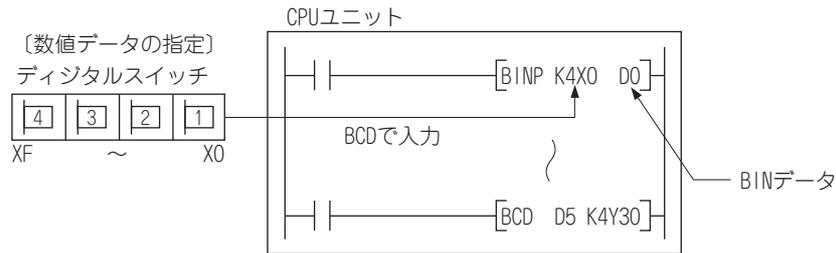


図 2.44 デジタルスイッチから CPU ユニットへのデータ取込み

備考

BIN 命令の詳細については、下記マニュアルを参照してください。

☞ MELSEC-Q/L プログラミングマニュアル (共通命令編)

(2) CPU ユニットから外部への数値の出力

CPU ユニットで演算した数値を外部で表示する場合には、デジタル表示器などが使用可能です。

(a) 数値を出力する方法

CPU ユニットは、BIN で演算を行っています。

CPU ユニットが使用している BIN をそのままデジタル表示器に出力しても、正常な表示が行えません。BIN で演算されたデータを、外部表示器などで使用する BCD に変換するためには、BCD 命令を使用します。

BCD 命令を使用すると、外部表示器などに DEC（10 進数）と同一の表示を行うことが可能になります。

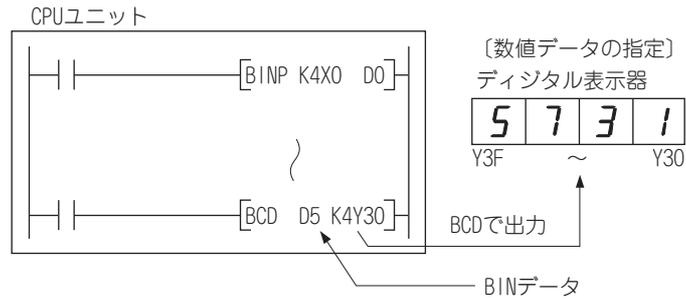


図 2.45 デジタル表示器による CPU ユニットの演算データの表示

備考

BCD 命令の詳細については、下記マニュアルを参照してください。

☞ MELSEC-Q/L プログラミングマニュアル（共通命令編）

2.4.1 BIN (2進数 : Binary Code)

(1) 2進数

BIN (2進数) は、0(OFF)と1(ON)で表現した数値です。

DEC (10進数) では0から数が増えて9までいくと、次に桁上りして10になります。

BINでは0,1の次に桁上りして、10 (10進数の2) になります。

BINとDECの数値表現を表2.5に示します。

表 2.5 BIN, DEC の数値表現

DEC (10進数)	BIN (2進数)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011

(2) BIN の数値表現

(a) CPU ユニットで使用している BIN のビット構成

CPU ユニットの各レジスタ (データレジスタ, リンクレジスタなど) は、16ビットで構成されています。

(b) CPU ユニットで使用できる数値データ

CPU ユニットの各レジスタには、-32768 ~ 32767 の数値を格納できます。

CPU ユニットの各レジスタの数値表現を図2.46に示します。



図 2.46 CPU ユニットの各レジスタの数値表現

Point

各々のレジスタの各ビットには、 2^n の数値が割り付けられています。

ただし、最上位のビットは、正負の判別用として使用しますので、符号無し (無符号) の BIN (0 ~ 65535) は使用できません。

- 最上位ビットが0…………… 正
- 最上位ビットが1…………… 負

2.4.2 HEX (16 進数 : Hexadecimal)

(1) HEX

HEX (16 進数) は, BIN データの 4 ビットを 1 桁として表現したものです。

BIN (2 進数) で 4 ビットを使用すると, 0 ~ 15 の 16 通りを表現できます。

HEX では, 0 ~ 15 を 1 桁で表現するために 9 の次の 10 を AH, 11 を BH とアルファベットで表し, FH(15) の次で桁上りします。

BIN, HEX, DEC (10 進数) の数値表現を表 2.6 に示します。

表 2.6 BIN, HEX, DEC の数値表現

DEC (10 進数)	HEX (16 進数)	BIN (2 進数)	
0	0		0
1	1		1
2	2		10
3	3		11
⋮	⋮		⋮
9	9		1001
10	A		1010
11	B		1011
12	C		1100
13	D		1101
14	E		1110
15	F		1111
16	10	1	0000
17	11	1	0001
⋮	⋮		⋮
47	2F	10	1111

← 桁上り

(2) HEX の数値表現

CPU ユニットの各レジスタ (データレジスタ, リンクレジスタなど) は, 16 ビットで構成されています。

16 ビットの場合, 16 進数では 0 ~ FFFFH が指定可能です。

2.4.3 BCD (2 進化 10 進数 : Binary Coded Decimal)

(1) BCD

BCD (2 進化 10 進数) は, DEC (10 進数) の 1 桁を BIN (2 進数) で表現したものです。HEX (16 進数) と同様に 4 ビット表現しますが, HEX の A ~ F を使用しません。BIN, BCD, DEC の数値表現を表 2.7 に示します。

表 2.7 BIN, BCD, DEC の数値表現

(10 進数)	BIN (2 進数)	BCD (2 進化 10 進数)	
0	0000		0
1	0001		1
2	0010		10
3	0011		11
4	0100		100
5	0101		101
6	0110		110
7	0111		111
8	1000		1000
9	1001		1001
10	1010	1	0000
11	1011	1	0001
12	1100	1	0010

← 桁上り

(2) BCD の数値表現

CPU ユニットの各レジスタ (データレジスタ, リンクレジスタなど) は, 16 ビットで構成されています。したがって, 各レジスタに格納できる数値を BCD で表すと 0 ~ 9999 の範囲となります。

2.4.4 実数（浮動小数点データ）

実数データには、単精度浮動小数点データと倍精度浮動小数点データがあります。

(1) 単精度浮動小数点データ

(a) 実数データの内部表現

CPU ユニットで取り扱う実数データの内部表現について示します。
実数データは、ワードデバイス2つを使って次のように表します。

[符号] 1. [仮数部] $\times 2$ [指数部]

実数データを内部表現するときのビット構成とその意味は次のとおりです。



図 2.47 実数データのビット構成

1) 符号

b31 は符号を表します。

- 0：正
- 1：負

2) 指数部

b23 ~ b30 にて 2^n の n を表現します。

b23 ~ b30 の BIN 値により n は次のようになります。

b23~b30	FFH	FEH	FDH		81H	80H	7FH	7EH		02H	01H	00H
n	未使用	127	126		2	1	0	-1		-125	-126	未使用

図 2.48 指数部に格納する値と指数の関係

3) 仮数部

b0 ~ b22 の 23 ビットにて、2 進数で 1.XXXXXX... としたときの XXXXXX... の値を示します。

(b) 計算例

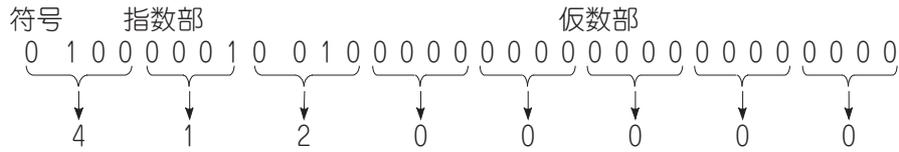
計算例を示します。(nnnnnn)x は X 進数で表したデータであることを示します。

1) 10 を格納したとき

$$(10)_{10} \rightarrow (1010)_2 \rightarrow (1.010000\dots \times 2^3)_2$$

符号 正 → 0
 指数部 3 → 82H → (10000010)₂
 仮数部 (010 00000 00000 00000 00000)₂

よってデータは



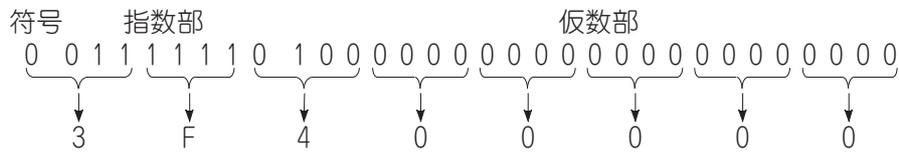
41200000H となります。

2) 0.75 を格納したとき

$$(0.75)_{10} \rightarrow (0.11)_2 \rightarrow (1.100\dots \times 2^{-1})_2$$

符号 正 → 0
 指数部 -1 → 7EH → (01111110)₂
 仮数部 (100 00000 00000 00000 00000)₂

よってデータは

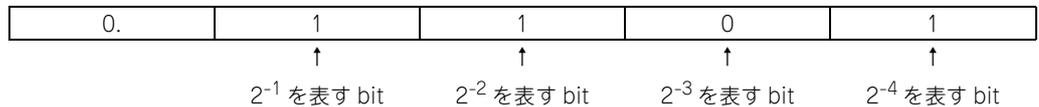


3F400000H となります。

Point

2 進数で小数点以下の値は次のように計算します。

例 (0.1101)₂ の場合



$$(0.1101)_2 = 2^{-1} + 2^{-2} + 2^{-4} = 0.5 + 0.25 + 0.0625 = (0.8125)_{10}$$

(c) 内部演算処理を倍精度で行う方法  注 2.7

PC パラメータの PC システム設定で “浮動小数点演算処理” にチェックを入れます。(デフォルトではチェックしてあります。)



図 2.49 PC システム設定画面

演算結果は、浮動小数点演算処理の設定に関係なく単精度です。

倍精度の内部演算処理は、内部演算のみ倍精度（64 ビット）で行います。

内部演算処理を倍精度で行うかどうかは、下記に応じて使い分けてください。

1) 内部演算処理を倍精度で行う場合

従来機種との互換性をもたせるために精度を要求する場合に使用します。

また SIN 命令、COS 命令のように内部演算で実数演算を多用する命令を使用する場合、内部演算処理を倍精度で行うと精度が高くなります。

2) 内部演算処理を倍精度で行わない場合

実数演算の高速化が必要な場合に使用します。

ただし、内部演算を単精度（32 ビット）で行うため実数演算は高速になりますが、一部精度が落ちることがあります。

Point 

- GX Developer のモニタ機能では、CPU ユニットの実数データをモニタできます。ただし “FFFFH” のようにモニタしようとしたデータが実数で表現できない場合は、“-” が表示されます。
- 0 を表すときには b0 ~ b31 をすべて 0 とします。



注 2.7

Basic

Process

Redundant

ベーシックモデル QCPU、プロセス CPU、二重化 CPU では、“内部演算処理を倍精度で行う” の設定はできません。

2.4.5 文字列データ

(1) 文字列データ

CPU ユニットで取り扱えるのは、シフト JIS コードの文字列です。

第3章 CPUユニットの動作

本章では、CPUユニットの動作について説明します。

3.1 イニシャル処理

イニシャル処理は、シーケンスプログラムの演算を実行するための前処理です。

表 3.1 に示す CPU ユニット状態の場合に 1 回のみ実行します。

イニシャル処理が完了すると、CPU ユニットは RUN/STOP スイッチ（ベーシックモデル QCPU では RUN/STOP/RESET スイッチ）で設定されている動作状態になります。（[図 3.5 節](#)）

表 3.1 イニシャル処理一覧

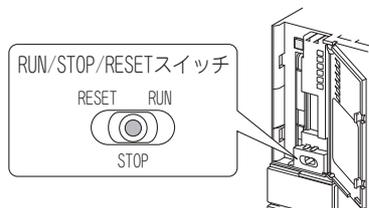
イニシャル処理項目	CPU ユニットの状態		
	電源投入時	リセット操作時	STOP 状態→RUN 状態時* 1
入出力ユニットの初期化	○	○	×
標準 ROM / メモリカードからのブート	○	○	×
PC パラメータのチェック	○	○	○
マルチ CPU システムパラメータの同一性チェック	○	○	○
ラッチ範囲外のデバイスの初期化 * 2 (ビットデバイス：OFF, ワードデバイス：0)	○	○	×
装着ユニットの入出力番号の自動割付け	○	○	○
CC-Link IE コントローラネットワーク、MELSECNET/H 情報のセット	○	○	×
インテリジェント機能ユニットのスイッチ設定	○	○	×
CC-Link 情報のセット	○	○	×
Ethernet 情報のセット	○	○	×
デバイス初期値のセット	○	○	○
シリアルコミュニケーション機能の設定	○	○	×

○：実行する，×：実行しない

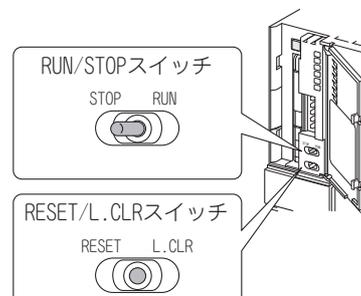
- * 1：STOP 状態にしてパラメータまたはプログラムを変更したあと、リセットを行わずに RUN 状態にした場合を示しています。
RUN/STOP スイッチ（ベーシックモデル QCPU では RUN/STOP/RESET スイッチ）を STOP → RUN → (RUN LED が点滅) → STOP → RUN と操作します。
上記操作では、パルス化命令 (PLS, □P) はプログラムの変更内容 (STOP 中の RUN 中書込みまたは PC 書込み) によっては前回情報が継続されないため、正常に動作しないことがありますので十分注意してください。
- * 2：二重化 CPU で、スタートモードがホットスタートモードの場合は、ラッチ範囲外のデバイスは初期化されません。
(ステップリレーおよびインデックスレジスタなどの一部デバイスを除く。)

- CPUユニットのSTOP, RUN, RESETスイッチは, CPUユニットごとに異なります。

ベーシックモデル QCPU の場合



ハイパフォーマンスモデル QCPU, プロセス CPU,
二重化 CPU の場合



- STOP 状態でパラメータまたはプログラムを変更した場合は, スイッチ操作により, リセットを行ってください。

3.2 I/O リフレッシュ (入出力ユニットのリフレッシュ処理)

シーケンスプログラムの演算開始前に下記を実行します。

- 入力ユニット/インテリジェント機能ユニットから CPU ユニットへの ON/OFF データ入力
- CPU ユニットから出力ユニット/インテリジェント機能ユニットへの ON/OFF データの出力

コンスタントスキャン実行時は, コンスタントスキャンの待ち時間が終了してから行います。(I/O リフレッシュは, コンスタントスキャン時間ごとに実行されます。)

3.3 プログラムの演算

CPU ユニットに格納されているプログラムのステップ 0 から END/FEND 命令までを順次実行します。

(☞ 第 2 章)

3.4 END 処理

ネットワークユニットなどのリフレッシュ処理や、外部機器との通信などを行います。

END 処理には、主に下記の処理があります。

表 3.2 END 処理一覧

項目	END 処理を行う CPU ユニット				参照
	ベーシック モデル QCPU	ハイパフォーマンス モデル QCPU	プロセス CPU	二重化 CPU	
ネットワークユニットのリフレッシュ	○	○	○	○	第 10 章
インテリジェント機能ユニットの自動 リフレッシュ	○	○	○	○	7.1.1 項
インテリジェント機能ユニット専用命令の 処理	○	○	○	○	第 10 章
サービス処理	○	○	○	○	6.25.1 項
ウォッチドッグタイマのリセット	○	○	○	○	6.16 節
マルチ CPU 間自動リフレッシュ	○	○	○	×	QCPU ユーザーズ マニュアル（マル チ CPU システム 編）
サンプリングトレース機能によるデバイス 内容の収集（トレースポイントがスキャン ごと（END 命令実行後）に設定されている 場合）	×	○	○	○	6.14 節
自己診断処理	○	○	○	○	6.17 節
特殊リレー／特殊レジスタへの値のセット （セットのタイミングが END 処理時のもの）	○	○	○	○	QCPU ユーザーズ マニュアル（ハー ドウェア設計・保 守点検編）

○：実行する ×：実行しない

Point

- コンスタントスキャン機能（ 6.2 節）を設定している場合、END 処理後または次のスキャンが始まるまでの間、END 処理時間の結果を保持しています。
- 低速実行タイププログラムを実行している場合、すべての低速実行タイププログラムの実行完了で低速 END 処理を行います。（ 2.3.3 項）

3.5 RUN 状態, STOP 状態, PAUSE 状態の演算処理

CPU ユニットの動作状態には, 次の 3 種類があります。

- RUN 状態
- STOP 状態
- PAUSE 状態

各動作状態における, CPU ユニットの演算処理について説明します。

(1) RUN 状態の演算処理

RUN 状態とは, シーケンスプログラムの演算をステップ 0 → END(FEND) 命令 → ステップ 0 の順で, 繰り返し演算を実行している状態です。

(a) RUN 状態に入るときの出力

RUN 状態に入るときは, パラメータの STOP → RUN 時の出力モード設定により, 下記のいずれかを出力します。(☞ 6.4 節)

- STOP 状態のときに退避した出力 (Y) の状態
- 1 スキャン後の演算結果

(b) 演算開始までの処理時間

STOP → RUN に切り換えてからシーケンスプログラム演算開始までの処理時間は, システム構成やパラメータ設定により変動します。(通常は 1 ~ 3 秒です。)

(2) STOP 状態の演算処理

STOP 状態とは, RUN/STOP/RESET スイッチ, またはリモート STOP (☞ 6.6.1 項) により, シーケンスプログラムの演算を中止している状態です。

また, 停止エラーが発生した場合も STOP 状態になります。

(a) STOP 状態に入るときの出力

STOP 状態に入るとき, 出力 (Y) の状態を退避して, 出力を全点 OFF します。
出力 (Y) 以外のデバイスメモリは, 保持されています。

(3) PAUSE 状態の演算処理

PAUSE 状態とは, リモート PAUSE (☞ 6.6.2 項) により, 1 スキャン実行後に出力およびデバイスメモリの状態を保持したまま, シーケンスプログラムの演算を中止している状態です。

(4) スイッチ操作時のCPUユニットの演算処理

表 3.3 スイッチ操作時の演算処理

RUN/STOP 状態	CPU ユニットの演算処理			
	シーケンスプログラムの演算処理	外部出力	デバイスメモリ	
			M,L,S,T,C,D	Y
RUN → STOP	END 命令まで実行し停止する。	STOP 状態になる直前の出力 (Y) の状態を退避して、全点 OFF する。	STOP 状態になる直前のデバイスメモリの状態を保持する。	STOP 状態になる直前の出力 (Y) の状態を退避して、全点 OFF する。
STOP → RUN	ステップ 0 から開始する。	PC パラメータの“STOP → RUN 時の出力モード”により決まる。 (☞ 6.4 節)	STOP 状態になる直前のデバイスメモリの状態を保持する。 ただしデバイス初期値が設定されている場合は、デバイス初期値の値をセットする。 またローカルデバイスはクリアする。	PC パラメータの“STOP → RUN 時の出力モード”により決まる。 (☞ 6.4 節)

Point

CPU ユニットの RUN 状態、STOP 状態、PAUSE 状態にかかわらず、下記の処理を行っています

- 入出力ユニットとのリフレッシュ処理
- ネットワークユニットのリフレッシュ処理
- インテリジェント機能ユニットの自動リフレッシュ処理
- 自己診断処理
- サービス処理
- インテリジェント機能ユニット専用命令の処理 (完了処理のみ)

このため STOP 状態、PAUSE 状態でも、下記の動作を実行できます。

- GX Developer による入出力のモニタやテスト操作
- MC プロトコルを使用した外部機器からの書込み/読出し
- CC-Link IE コントローラネットワーク、MELSECNET/H による他局との交信
- CC-Link のリモート局との交信

3.6 瞬停時の演算処理

CPU ユニットは、電源ユニットに供給される入力電源電圧が規定範囲より低くなったときに瞬停を検出し、下記の演算処理を行います。

(1) 許容瞬停時間以下の瞬停が発生したとき

瞬停が発生すると、エラー履歴の登録を行ってから、演算処理を中断します。
ただし、タイマデバイスの計測は継続されています。
また、出力状態は保持されます。

(a) SFC プログラムの続行スタート指定がある場合

システムの退避処理を行います。

(b) 瞬停が解除された場合

瞬停が解除されると、演算処理を続行します。

(c) 瞬停発生時のウォッチドッグタイマ (WDT) の測定

瞬停が発生して演算を中断していても、ウォッチドッグタイマ (WDT) の計測は継続されています。
例えば、PC パラメータの WDT 設定が 200ms で、スキャンタイムが 190ms のとき、15ms の瞬停が発生すると “WDT ERROR” になります。

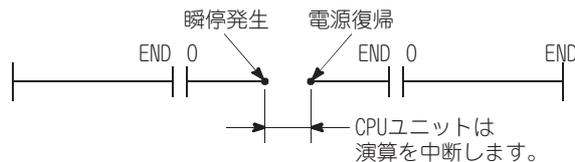


図 3.1 瞬停発生時の演算処理

(2) 許容瞬停時間を超える停電が発生したとき

CPU ユニットは、イニシャルスタートになります。
下記操作をしたときと同じ演算処理になります。

- ・シーケンサの電源投入
- ・RESET/L.CLR スイッチ(ベーシックモデル QCPU は RUN/STOP/RESET スイッチ)によるリセット操作
- ・GX Developer によるリモートリセット操作

Point

- 電源ユニットを二重化している場合、片方の電源ユニットで瞬停が発生しても演算は中断しませんが、片方の電源ユニットにのみ電源が供給されている状態で瞬停が発生したときは、演算を中断します。

電源二重化システムにおける瞬停の情報は、SM1782～1783 および SD1782～1783 に格納されます。
電源を二重化していない場合の瞬停の情報は、SM53 および SD53 に格納されます。
SM53 および SD53 の詳細については、下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編)

3.7 データのクリア処理

CPUユニットのデータのクリア方法、ラッチデータクリアに関する設定について説明します。

(1) データのクリア方法

CPUユニットは、リセット操作（スイッチ操作によるリセット、電源のOFF→ONによるリセットなど）を行うと、データがクリアされます。

ただし、下記(a)のデータは、上記に示した操作ではクリアできません。

(a) リセット操作でクリアされないデータ

- プログラムメモリのデータ
- 標準ROMのデータ
- メモリカード内のデータ
- ラッチ指定しているデバイスのデータ（ 本節(2)）
- ファイルレジスタのデータ

(b) リセット操作でクリアされないデータのクリア方法

1) プログラムメモリのデータ

下記のいずれかの方法でクリアします。

- PCパラメータのブートファイル設定で、“プログラムメモリをクリアする”を設定します。

注 3.1

- GX Developerの[オンライン]→[PCデータ削除]で行います。

2) 標準ROMのデータ

標準ROMへデータを書き込むときに、自動的にクリアされます。

3) メモリカード内のデータ

GX Developerの[オンライン]→[PCデータ削除]で行います。

4) ラッチ指定しているデバイスのデータ

本節(2)を参照してください。

5) ファイルレジスタのデータ

下記のいずれかの方法でクリアします。

- RST命令でリセットする
- MOV/FMOV命令でK0を転送する
 MELSEC-Q/Lプログラミングマニュアル（共通命令編）
- GX Developerの[オンライン]→[PCメモリクリア]の“ファイルレジスタオールクリア”を実行する



注 3.1

Basic

ベーシックモデル QCPU では、“プログラムメモリをクリアする”の設定はできません。

(2) デバイスのラッチ指定

デバイスのラッチ指定（ラッチ範囲設定）は、PC パラメータのデバイス設定で行います。（ 6.3 節 (5)）

(a) ラッチ範囲の設定

GX Developer で行うラッチ範囲の設定には、下記の 2 種類があります。

1) ラッチクリア操作有効（ラッチ (1) 先頭／最終）

RESET/L.CLR スイッチ  **注 3.2** によるラッチクリアまたはリモートラッチクリアでクリアできるラッチ範囲を設定します。

2) ラッチクリア操作無効（ラッチ (2) 先頭／最終）

RESET/L.CLR スイッチ  **注 3.2** によるラッチクリアまたはリモートラッチクリアでクリアできないラッチ範囲を設定します。

(b) ラッチクリア操作有効に設定したデバイスデータのクリア

RESET/L.CLR スイッチ  **注 3.2** によるラッチクリアまたはリモートラッチクリア（ 6.6.4 項）でクリアします。

(c) ラッチクリア操作無効に設定したデバイスのデータクリア

下記のいずれかの方法でクリアします。

- RST 命令でリセットする
- MOV/FMOV 命令で K0 を転送する
 MELSEC-Q/L プログラミングマニュアル（共通命令編）
- GX Developer の [オンライン] → [PC メモリクリア] の “ デバイスメモリオールクリア(ラッチを含む) ” を実行する

Point

二重化 CPU でスタートモードがホットスタートモードの場合は、ラッチ範囲に設定していないデータも保持されます。（インデックスレジスタおよびステップリレーなどの一部デバイスは除きます。）

備考

GX Developer の操作方法については、下記マニュアルを参照してください。

 GX Developer Version 8 オペレーティングマニュアル



注 3.2

Basic

ベーシックモデル QCPU では、スイッチ操作によるラッチクリアはできません。

3.8 入出力処理と応答遅れ

CPUユニットの入出力処理は、リフレッシュ方式です。

ただし、シーケンスプログラムでダイレクトアクセス入出力を使用することにより、各命令の実行時にダイレクト方式の入出力処理が可能です。

CPUユニットの入出力処理の方式と応答遅れについて説明します。

(a) リフレッシュ方式 (☞ 3.8.1 項)

リフレッシュ方式は、入出力ユニットとのアクセスをシーケンスプログラムの演算開始前に一括で行う方式です。

(b) ダイレクト方式 (☞ 3.8.2 項)

ダイレクト方式は、入出力ユニットとのアクセスをシーケンスプログラムの各命令実行時に行う方式です。ダイレクト方式で入出力ユニットとのアクセスを行う場合は、シーケンスプログラムでダイレクトアクセス入力、またはダイレクトアクセス出力を使用します。

(1) リフレッシュ方式とダイレクト方式の相違点

ダイレクト方式は、命令実行時に直接入出力ユニットとアクセスを行うため、リフレッシュ方式に比べ入力の取り込みは早くなります。

ただし、リフレッシュ方式に比べ命令の処理時間は長くなります。

各入出力でのリフレッシュ方式とダイレクト方式の使用可否を表 3.4 に示します。

表 3.4 各入出力でのリフレッシュ方式とダイレクト方式の使用可否

項目	リフレッシュ方式	ダイレクト方式
入出力ユニット	使用可	使用可
インテリジェント機能ユニットの入出力		
MELSEC-I/OLINK リモート I/O システムマスタユニット (AJ51T64/A1SJ51T64) の入出力 * 1	使用可	使用不可
CC-Link IE コントローラネットワーク, MELSECNET/H, CC-Link のリモート入出力	使用可	使用不可

* 1 : AnS/A シリーズ対応増設ベースユニット (QA1S5 □ B, QA1S6 □ B, QA1S6ADP+A1S5 □ B/A1S6 □ B, QA6 □ B, QA6ADP+A5 □ B/A6 □ B) に装着して使用します。(ハイパフォーマンスモデル QCPU のみ可能です。)

3.8.1 リフレッシュ方式

(1) リフレッシュ方式とは

入出力ユニットとのアクセスをシーケンスプログラムの演算開始前に一括で行う方式です。

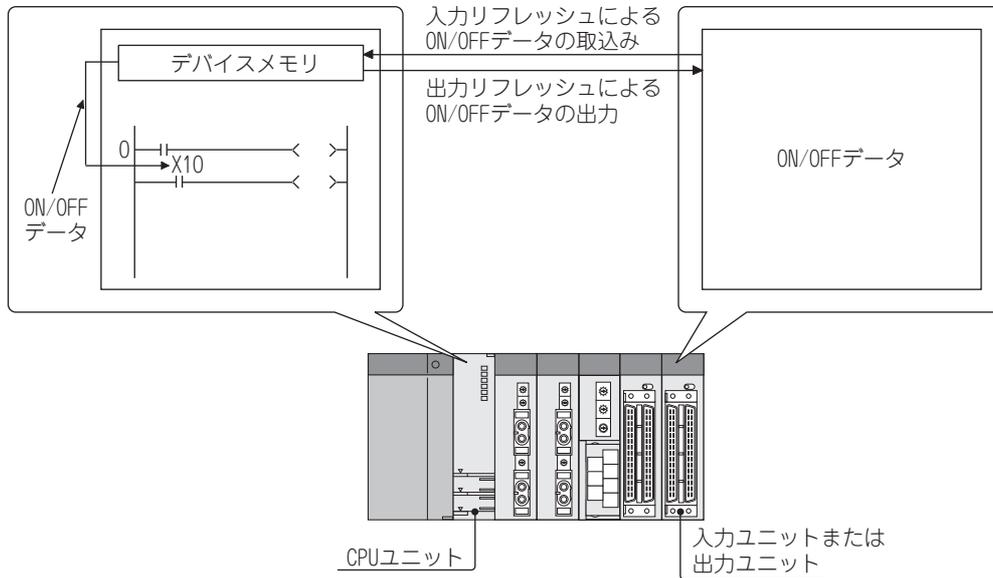


図 3.2 リフレッシュ方式

(2) 入力について

入力ユニットの ON/OFF データは、シーケンスプログラムの演算開始前に一括して CPU ユニット内部の入力ユニットとの交信用エリアに取り込みます。

シーケンスプログラム実行時は、入力 (X) 用デバイスメモリの ON/OFF データを使用して演算を行います。

(3) 出力について

出力 (Y) のシーケンスプログラムでの演算結果は、その都度 CPU ユニット内部の出力 (Y) 用デバイスメモリに出力し、シーケンスプログラムの演算開始前に出力 (Y) 用デバイスメモリの ON/OFF データを、一括して出力ユニットに出力します。

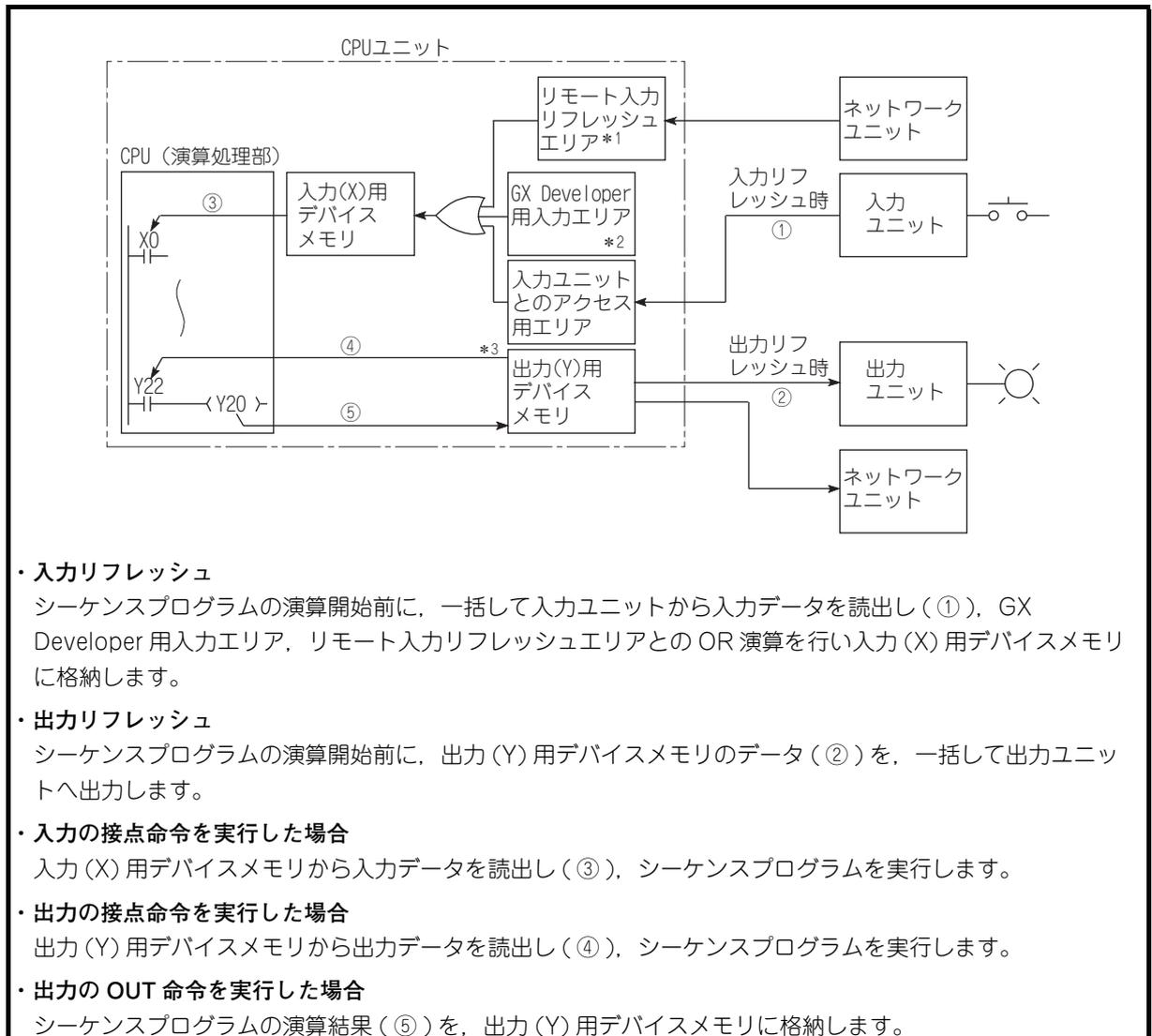
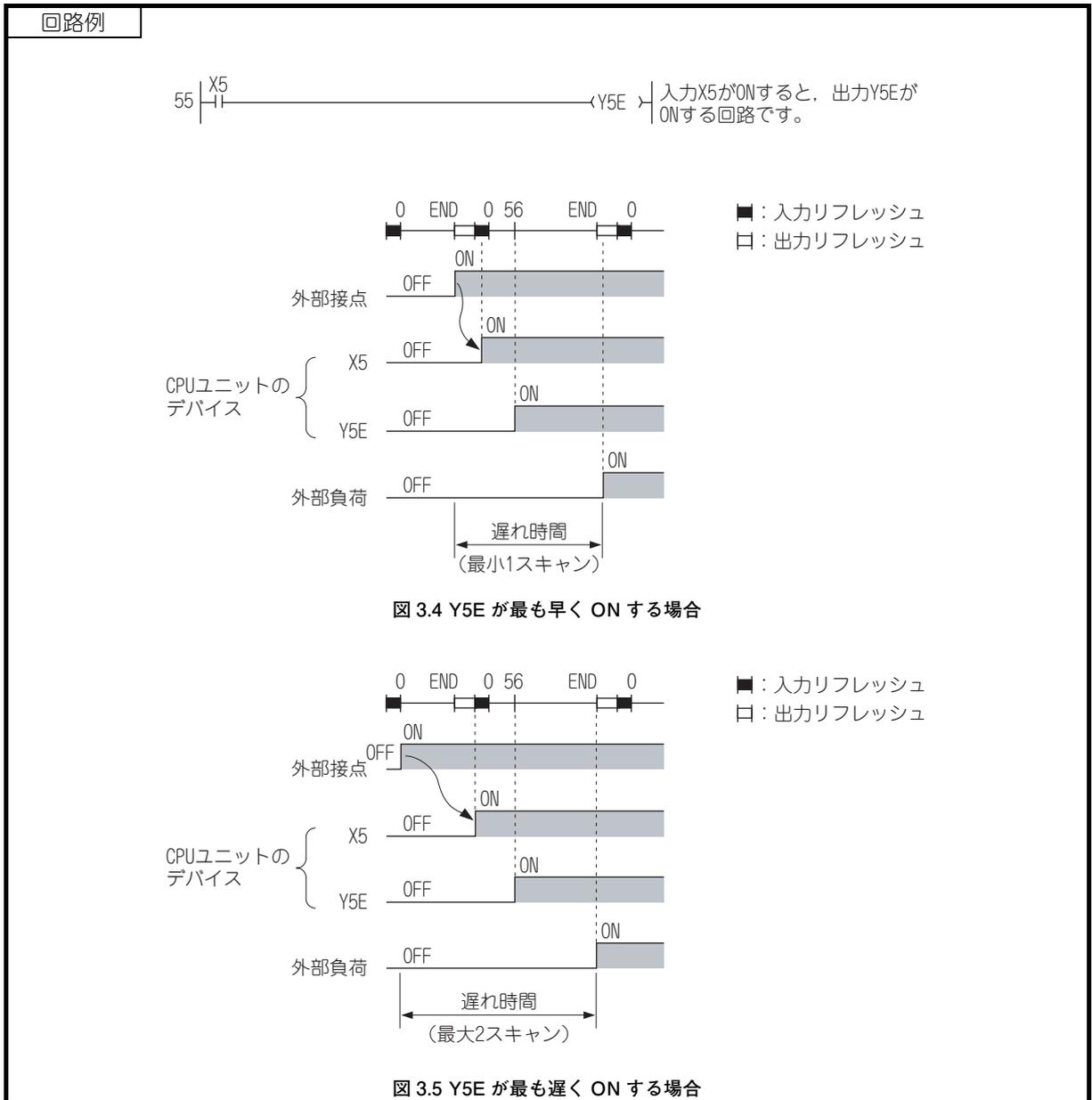


図 3.3 リフレッシュ方式の入力/出力情報の流れ

- * 1: リモート入力リフレッシュエリアは、CC-Link IE コントローラネットワーク、MELSECNET/H、CC-Link で入力 (X) に自動リフレッシュ設定した場合のエリアを示しています。
リモート入力リフレッシュエリアの自動リフレッシュは、END 処理時に行います。
- * 2: GX Developer 用入力エリアを ON/OFF できるものには、下記があります。
 - ・ GX Developer によるテスト操作
 - ・ ネットワークユニットからの書込み
 - ・ MC プロトコルを使用した外部機器からの書込み
- * 3: デバイスメモリ出力 (Y) 用を ON/OFF できるものには、下記があります。
 - ・ GX Developer によるテスト操作
 - ・ MC プロトコルを使用した外部機器からの書込み
 - ・ ネットワークユニットからの書込み

(4) 応答遅れ

入力ユニットの変化に対する出力の変化は、外部接点が ON するタイミングによって、最大 2 スキャンの遅れになります。



3.8.2 ダイレクト方式

(1) ダイレクト方式とは

入出力ユニットとのアクセスをシーケンスプログラムの各命令実行時に行う方式です。

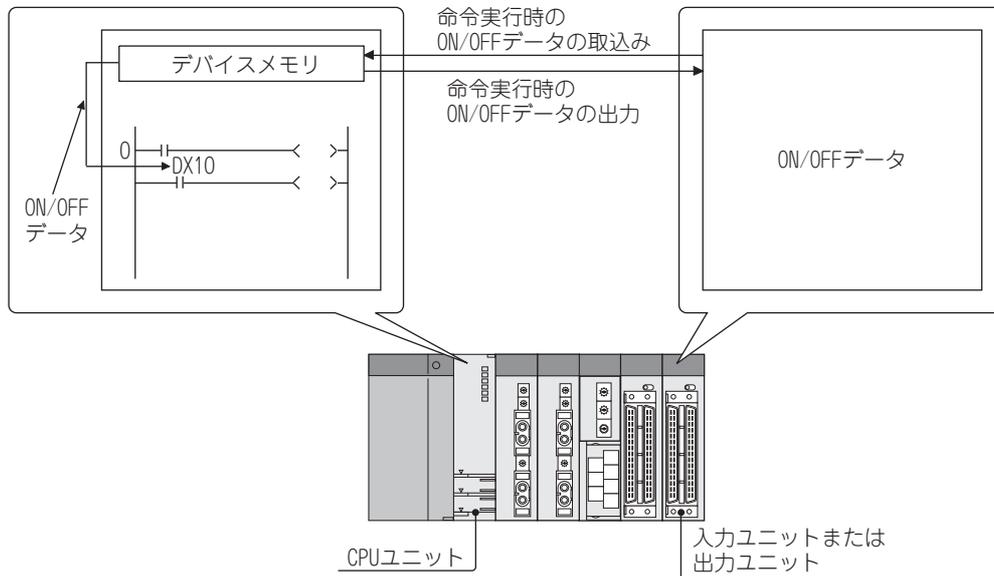
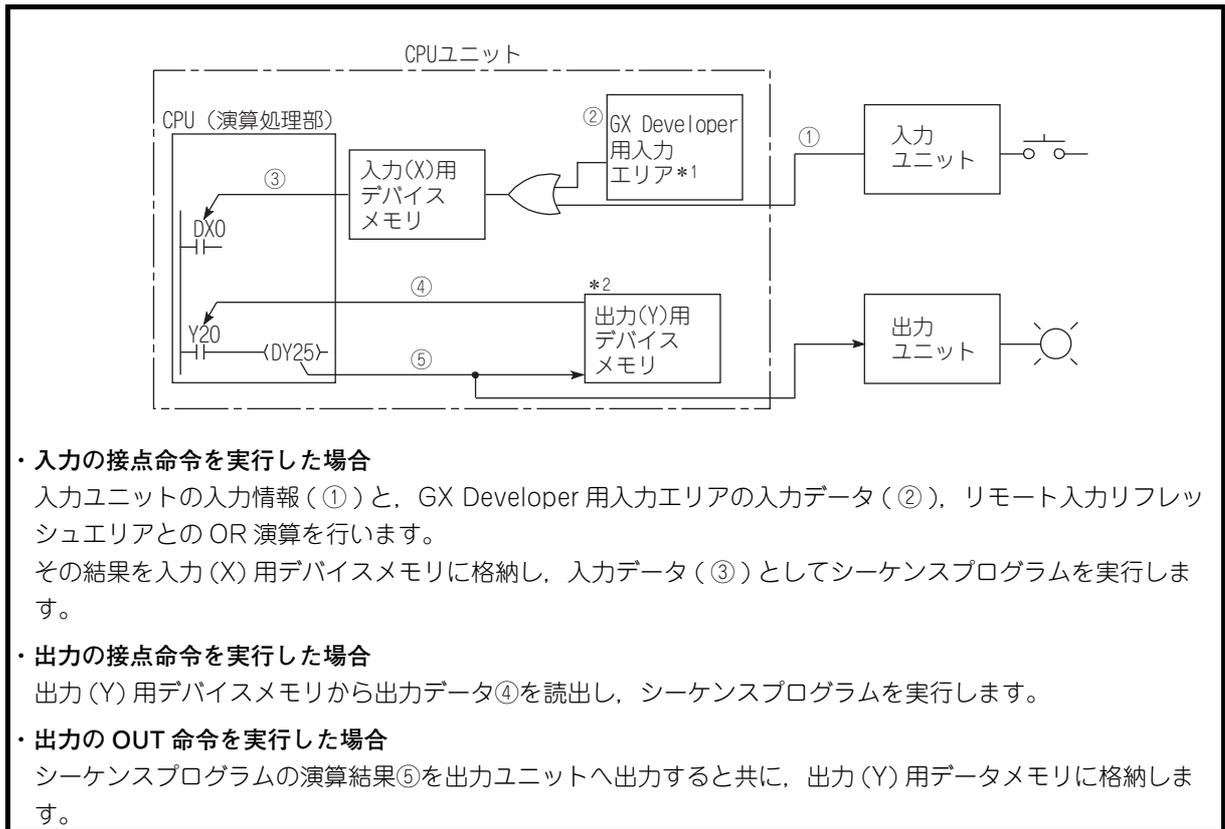


図 3.6 ダイレクト方式

CPU ユニットでは、ダイレクトアクセス入力 (DX)、ダイレクトアクセス出力 (DY) を使用して、ダイレクト方式の入出力処理を行います。



・ **入力の接点命令を実行した場合**

入力ユニットの入力情報(①)と、GX Developer 用入力エリアの入力データ(②)、リモート入力リフレッシュエリアとの OR 演算を行います。

その結果を入力(X)用デバイスメモリに格納し、入力データ(③)としてシーケンスプログラムを実行します。

・ **出力の接点命令を実行した場合**

出力(Y)用デバイスメモリから出力データ④を読み出し、シーケンスプログラムを実行します。

・ **出力の OUT 命令を実行した場合**

シーケンスプログラムの演算結果⑤を出力ユニットへ出力すると共に、出力(Y)用データメモリに格納します。

図 3.7 ダイレクト方式の入力／出力情報の流れ

* 1 : GX Developer 用入力エリアを ON/OFF できるものには、下記があります。

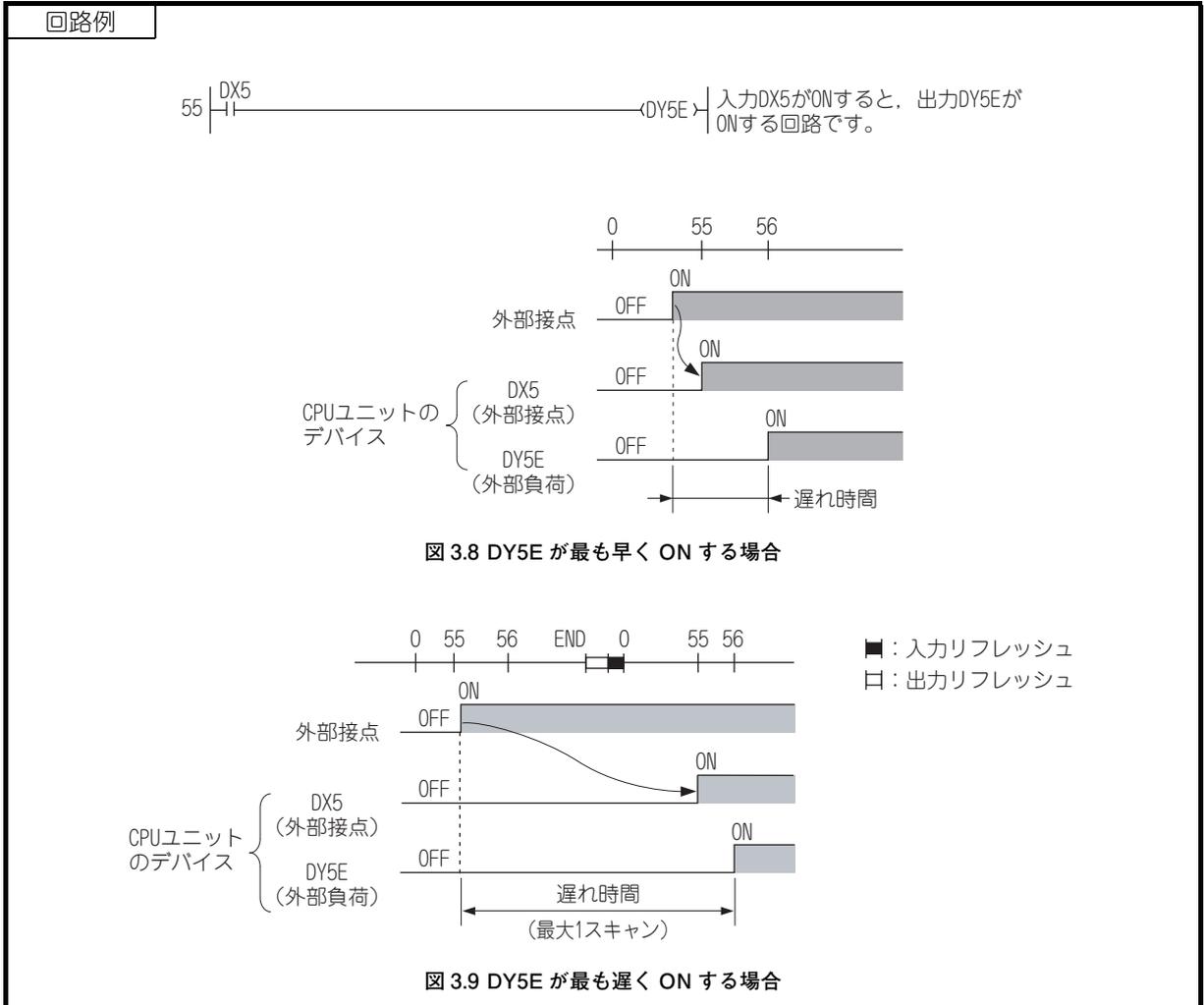
- ・ GX Developer によるテスト操作
- ・ ネットワークユニットからの書込み
- ・ MC プロトコルを使用した外部機器からの書込み

* 2 : デバイスメモリ出力(Y)用を ON/OFF できるものには、下記があります。

- ・ GX Developer によるテスト操作
- ・ MC プロトコルを使用した外部機器からの書込み
- ・ ネットワークユニットからの書込み

(2) 応答遅れ

入力ユニットの変化に対する出力の変化は、外部接点が ON するタイミングによって、最大 1 スキャンの遅れになります。



3.8 入力処理と応答遅れ
3.8.2 タイムアウト方式

第4章 ベースユニット／入出力番号の割付け

本章では、CPUユニットが入出力ユニット、インテリジェント機能ユニットとのデータの授受を行うために必要なベースユニット／入出力番号の割付けについて説明します。

4.1 ベースユニットの割付け

4.1.1 ベースモードとは

基本ベースユニット、増設ベースユニットに装着できるスロット数を割り付けるモードです。ベースモードには、下記の2種類があります。

- 自動モード
- 詳細モード

(1) 自動モードとは

使用するベースユニットの実際のスロット数でベースユニットの割付けを行うモードです。

(2) 詳細モードとは

ベースユニットごとに設定するモードです。

スロット数は、使用するベースユニットの実際のスロット数に関係なく設定できます。

(a) 実際のスロット数よりも多いスロット数を設定する場合

設定したスロット数分を占有します。

使用しているベースユニットのスロット数よりあとは、空きスロットになります。

例えば、5スロットのベースユニットを使用し、8スロットに設定すると3スロットが空きスロットになります。

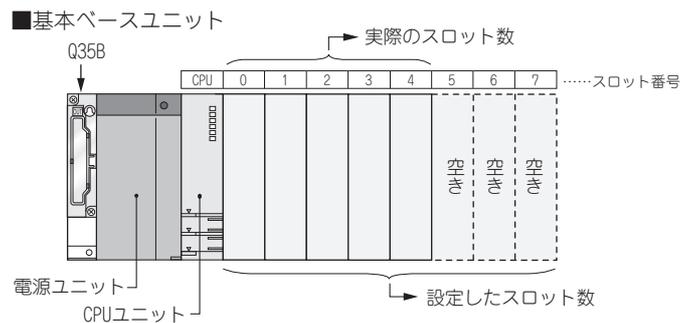


図 4.1 実際のスロット数よりも多いスロット数を設定する場合

空きスロットの点数は、PCパラメータのPCシステム設定で設定した空きスロット点数、またはPCパラメータのI/O割付で設定した点数になります。(デフォルトは16点です。)

(b) 実際のスロット数よりも少なくスロット数を設定する場合

ユニットを装着しないスロット以降を認識しないようにする場合に使用します。
 例えば、12 スロットのベースユニットを使用し、8 スロットに設定するとベースユニットの右から 4 スロットが使用禁止になります。
 (使用禁止のスロットにユニットを装着すると “SP.UNIT LAY ERR.” になります。)

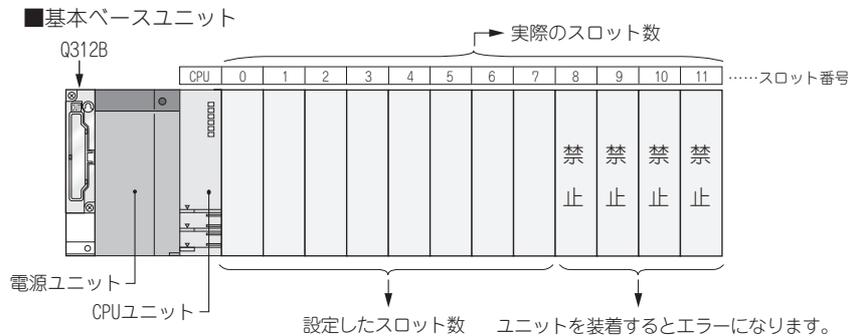


図 4.2 実際のスロット数よりも少なくスロット数を設定する場合

4.1.2 ベースユニットの割付け設定

PC パラメータの I/O 割付設定で行います。

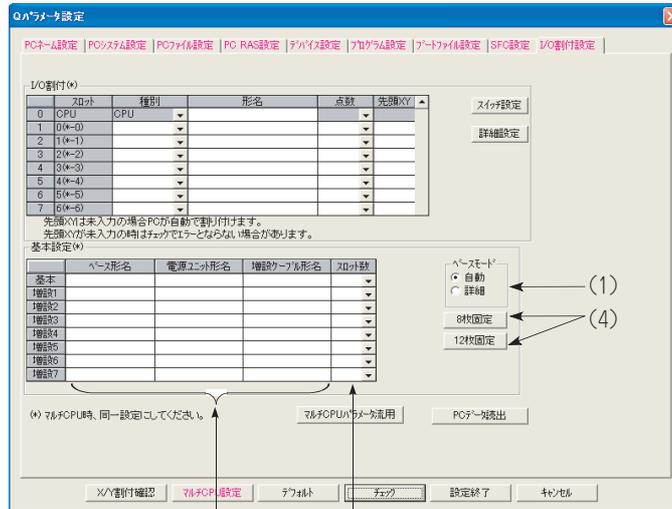


図 4.3 I/O 割付設定

(1) 自動／詳細

ベースユニットの割付けを自動モードで行うか、詳細モードで行うかを選択します。

(2) ベース形名、電源ユニット形名、増設ケーブル形名

メモまたはパラメータのプリントアウト用として、使用するベースユニットの形名、電源ユニットの形名、増設ケーブルの形名をそれぞれ半角 16 文字以下で入力します。
 CPU ユニットは入力内容を使用しません。

(3) スロット数

詳細モードの場合、使用するベースユニットのスロット数を下記から選択します。

2 (2 スロット), 3 (3 スロット), 5 (5 スロット), 8 (8 スロット), 10 (10 スロット),
12 (12 スロット)

(4) 8 枚固定 / 12 枚固定

詳細モードの場合、ベースユニットを一括で指定スロット数に設定するとき選択します。

Point

- 自動モードの場合は、ベースユニットの段数設定コネクタで増設段数を飛ばして設定しても空きの増設段数は確保できません。
将来拡張用に空きの増設段数を確保するには、詳細モードで設定してください。
 - 詳細モードの場合、使用するすべてのベースユニットに対してスロット数の設定を行ってください。
設定されていない場合は、I/O 割付けが正常に動作しないことがあります。
-

4.2 入出力番号の割付け

入出力番号は、下記の場合にシーケンスプログラムで使用するアドレスを示すためのものです。

- CPU ユニットへの ON/OFF データの取込みを行う
- CPU ユニットから外部への ON/OFF データの出力を行う

(1) ON/OFF データの取込みと出力

CPU ユニットへの ON/OFF データの取込みは入力 (X)、CPU ユニットからの ON/OFF データの出力は出力 (Y) で行います。

(2) 入出力番号の表現

入出力番号は 16 進数で表現します。

16 点の入出力ユニットを使用する場合、入出力番号は図 4.4 のように 1 スロットが□□0～□□F の 16 点で連番になります。

入力ユニットの場合は、入出力番号の先頭に“X”，出力ユニットの場合は入出力番号の先頭に“Y”を付けます。

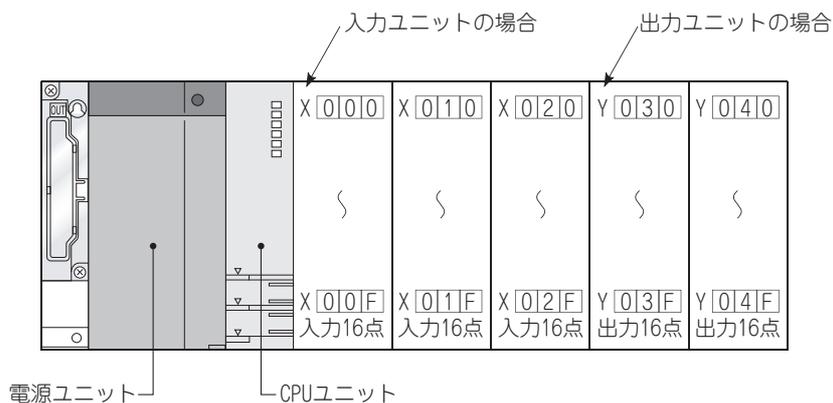


図 4.4 入力番号と出力番号

4.2.1 入出力番号の割付けの考え方

CPU ユニットは電源投入またはリセット解除時に、I/O 割付け設定に従って入出力番号の割付けを行います。

(1) 入出力番号の割付け

基本ベースユニットに CPU ユニットが装着されているシステムでのベースユニットの入出力番号の割付け例を図 4.5 に示します。

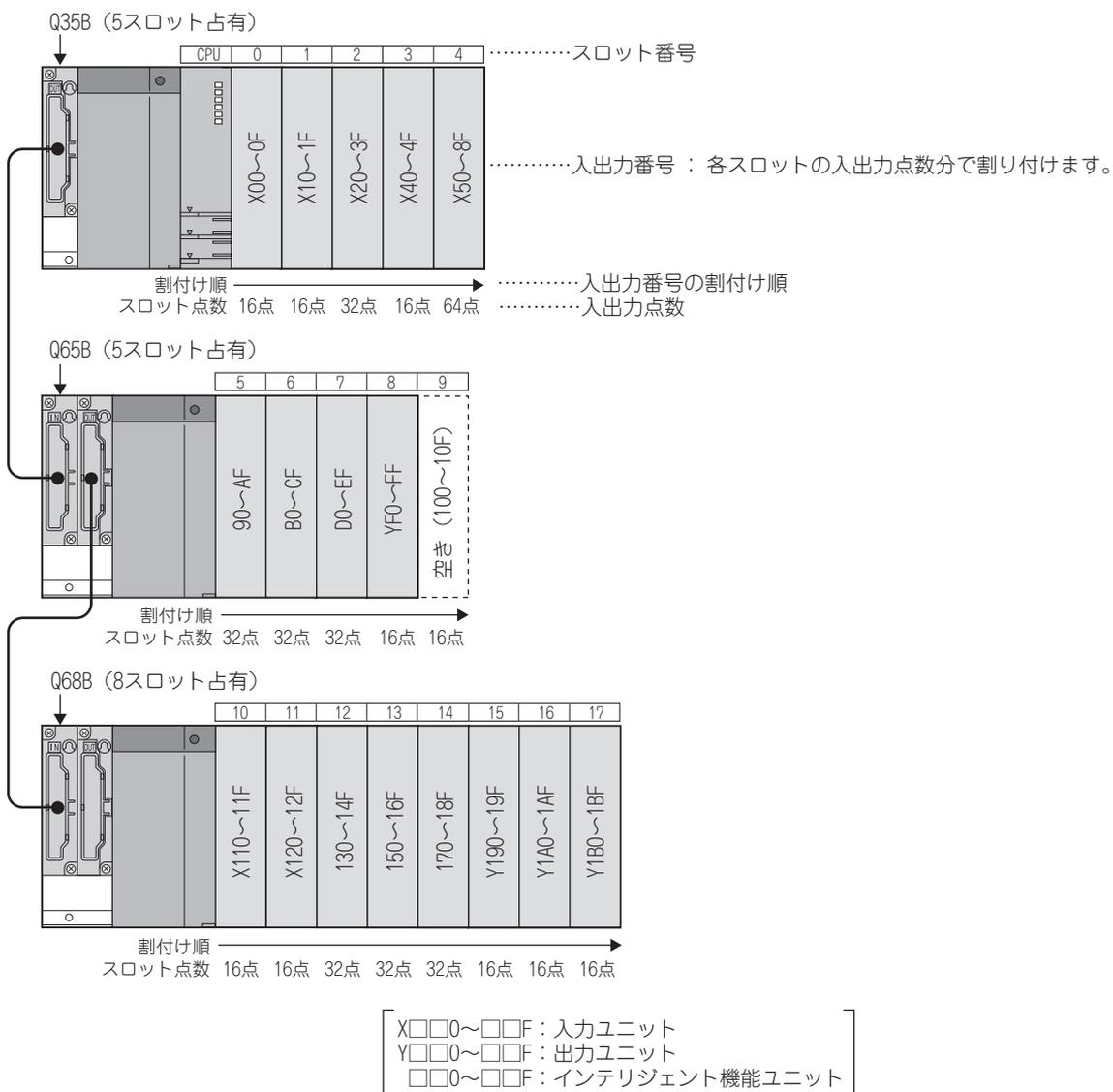


図 4.5 入出力番号の割付け例

(a) 入出力番号の割付け順

基本ベースユニットでは、CPU ユニットの右隣を 0H として、順に連番で割り付けます。
 増設ベースユニットでは、基本ベースユニットの入出力番号の次の番号から割り付けます。

(b) 各スロットの入出力番号

ベースユニットの各スロットは、装着されたユニットの点数分の入出力番号を占有します。

(2) リモート局での入出力番号の割付け  注 4.1

MELSECNET/H リモート I/O ネット、CC-Link などのリモート I/O システムでは、リモート局の入出力ユニット／インテリジェント機能ユニットに、CPU ユニットのデバイスの入力 (X)、出力 (Y) を割り付けて制御できます。

また、MELSECNET/H ユニットのリンク入出力 (LX, LY) のリフレッシュ先 (CPU ユニット側デバイス) に、入力 (X)、出力 (Y) が使用できます。

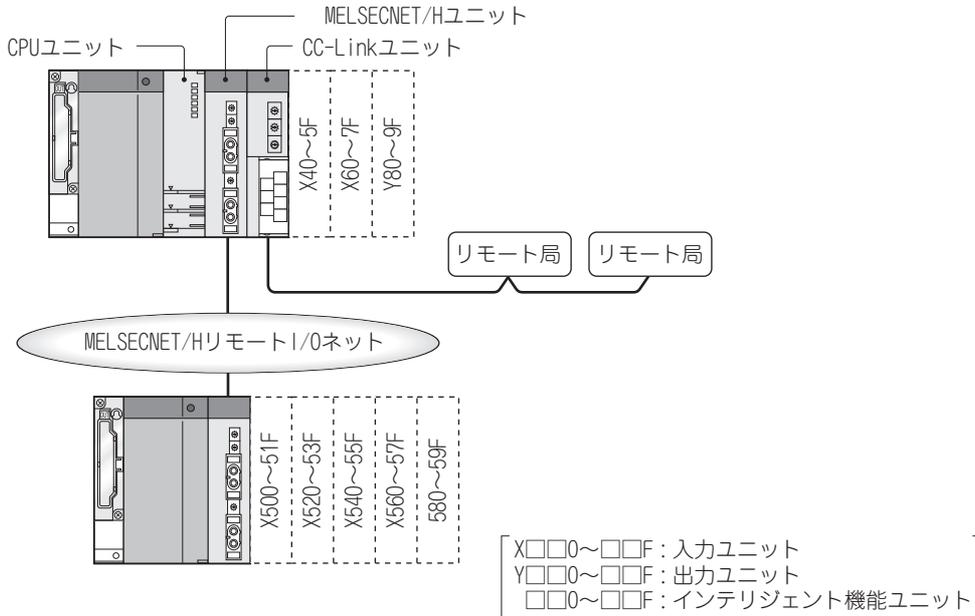


図 4.6 リモート局での入出力番号の割付け

(a) リモート局で使用可能な CPU ユニットの入出力番号

リモート局の入出力番号に、CPU ユニットの入力 (X)、出力 (Y) を使用する場合は、CPU ユニット側の入出力ユニットおよびインテリジェント機能ユニットで使用している入出力番号以降を割り付けてください。

例 CPU ユニット側の入出力ユニット／インテリジェント機能ユニットで X/Y0 ~ X/Y3FF (1024 点) を使用している場合、リモート局では X/Y400 以降が使用可能です。



注 4.1

Basic

High performance

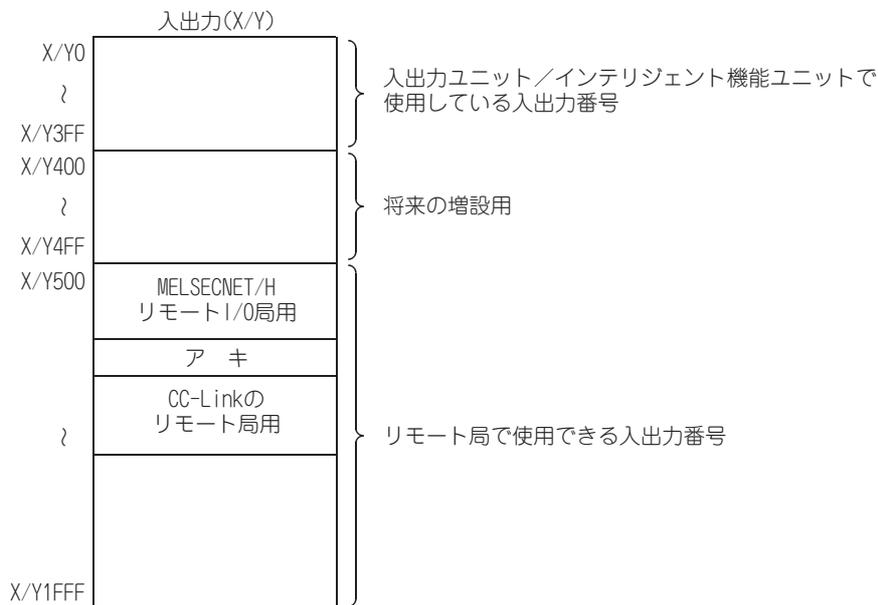
ベーシックモデル QCPU は、MELSECNET/H リモート I/O ネットに対応していません。

ハイパフォーマンスモデル QCPU で MELSECNET/H リモート I/O ネットを使用する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。( 付 2.2)

(b) リモート局の入出力番号使用時の注意事項

1) 将来の増設を考慮した設定

リモート局の入出力番号に、CPUユニットの入力(X)、出力(Y)を使用する場合は、CPUユニット側の入出力ユニット/インテリジェント機能ユニットの増設を考慮して設定してください。



入出力ユニット/インテリジェント機能ユニットで X/Y0~3FF(1024点)を使用し、将来の増設用に X/Y400~4FF(256点)を確保する場合

図 4.7 リモート局の入出力番号割付け

2) MELSECNET/H および CC-Link 使用時

MELSECNET/Hのリフレッシュ先(CPUユニット側デバイス)の入出力番号、およびCC-LinkリモートI/Oシステムの入出力番号は重複しないようにしてください。

Point

- CC-Linkでネットワークパラメータの設定を行っていない場合は、若番のCC-Linkのマスタ・ローカルユニットにX/Y1000~17FF(2048点)の点数が割り付けられます。
- MELSECNET/HリモートI/Oネット、CC-Linkなどの入出力番号の割付け順に制約はありません。
- MELSECNET/HリモートI/O局用とCC-Linkのリモート局用の間に空きを設けることもできます。

4.2.2 入出力番号の設定

入出力番号の設定は、I/O 割付設定で行います。

(1) 入出力番号の設定の目的

(a) 16 点ユニット以外のユニットに変更時の予約

現状使用しているユニットを、将来違うユニットに変更する場合に、入出力番号を変更しなくても済むように余裕を持たせて設定できます。

例えば、16 点入力ユニットが装着されているスロットを、将来使用する予定の 32 点入力ユニットとして割り付けることが可能です。

(b) ユニット取換え時の入出力番号の変化の防止

16 点以外の入出力ユニット、インテリジェント機能ユニットの故障でユニットを取りはずした場合に、入出力番号が変化するのを防止できます。

(c) プログラムで使用している入出力番号への変更

設計したプログラムで使用している入出力番号と実際のシステムの入出力番号が異なる場合、ベースユニットの各ユニットの入出力番号をプログラムで使用している入出力番号に変更できます。

Point

- I/O 割付設定を行っていない状態で 16 点以外の入出力ユニットが故障した場合、そのユニット以降の入出力番号が変化して誤動作につながる恐れがあるため、I/O 割付設定を行うことを推奨します。
- I/O 割付設定では、下記の設定もできます。
 - ・ 入力応答時間の設定 (I/O 応答時間) (☞ 6.7 節)
 - ・ エラー時出力モード設定 (☞ 6.8 節)
 - ・ インテリジェント機能ユニットのハードウェアエラー時の CPU ユニットの動作設定 (☞ 6.9 節)
 - ・ インテリジェント機能ユニット、割込みユニットのスイッチの設定 (☞ 6.10 節)

入力応答時間の設定、スイッチ設定を行う場合は、I/O 割付設定が必要です。

(2) I/O 割付設定

PC パラメータの I/O 割付設定で行います。

“I/O 割付” では、ベースユニットの-slot ごとに下記項目を個別に設定できます。

- “種別” (ユニット種別)
- “点数” (入出力点数)
- “先頭 XY” (先頭入出力番号)

例えば、指定 slot の入出力点数を変更する場合は、点数のみ設定できます。
設定しない項目は、ベースユニットの装着状態になります。



図 4.8 I/O 割付設定

(a) スロット

スロット No. とベースユニットの何段目の何スロット目であることを表示します。

ベースユニットを自動モードで設定した場合、ベースユニットの何段目かは “*” になり、何スロット目かは基本ベースユニットの 0 スロット目からのスロット数になります。

(b) 種別

装着しているユニットの種別を下記項目から選択します。

- 空き（空きスロット）
- 入力（入力ユニット）
- 高速入力（高速入力ユニット）
- 出力（出力ユニット）
- 入出力混合（入出力混合ユニット）
- インテリ（インテリジェント機能ユニット）
- 割込み（割込みユニット）

種別を設定しないスロットは、実装されているユニットの種別になります。

(c) 形名

装着しているユニットの形名を半角 16 文字以下で設定します。

CPU ユニットでは設定されている形名を使用しません。（ユーザでのメモになります。）

(d) 点数

各スロットの入出力点数を変更する場合は、下記点数から選択します。

0 点, 16 点, 32 点, 48 点, 64 点, 128 点, 256 点, 512 点, 1024 点

点数を設定しないスロットは、実装されているユニットの点数になります。

ただし、空きスロットには、PC パラメータの PC システム設定で設定した点数が割り付けられます。
（デフォルト：16 点）

(e) 先頭 XY

各スロットの入出力番号を変更する場合は、変更後の先頭入出力番号を設定します。

先頭 XY を設定しないスロットは、設定されているスロットから連番で入出力番号が割付けられます。

(3) 注意事項

(a) 種別の設定

I/O 割付設定の種別は、実装されているユニットと同一にしてください。

種別が異なる場合は、正常に動作しません。

また、インテリジェント機能ユニットは、入出力点数も同一にしてください。

実装されているユニットと I/O 割付設定の種別が異なる場合の動作を表 4.1 に示します。

表 4.1 実装ユニットと I/O 割付けが異なる場合の動作一覧

実装されているユニット	I/O 割付設定の種別	結果
入力ユニット, 出力ユニット, 入出力混合ユニット	・インテリ ・割込	“SP.UNIT.LAY.ERR.” になります。
インテリジェント機能ユニット	・入力 ・高速入力 ・出力 ・入出力混合	“SP.UNIT.LAY.ERR.” になります。
空きスロット	・入力 ・高速入力 ・出力 ・入出力混合 ・インテリ ・割込	空きスロット扱いになります。
全てのユニット	・空き	空きスロット扱いになります。
その他の組合せ	—	エラーにならないが、正常に動作しません。

(b) スロットの入出力点数

ユニットの実装よりも I/O 割付設定が優先されます。

1) 実装されている入出力ユニット点数より少ない点数を設定した場合

実装されている入出力ユニットの使用できる点数が減ります。

例えば、32 点の入力ユニットが装着されているスロットを I/O 割付設定で 16 点の入力ユニットに設定すると、32 点の入力ユニットの後半 16 点は使用できません。

2) 実装されている入出力ユニット点数より多い点数を設定した場合

実装の点数をオーバした分の点数は、入出力ユニットで使用しません。

3) 最後の入出力番号

I/O 割付設定を行った場合、最終の入出力番号が入出力点数の最大値の範囲内になるように設定してください。

最終の入出力番号が入出力点数の最大値を超える設定を行った場合は、“SP.UNIT LAY ERR.” になります。(GX Developer のシステムモニタでは、I/O アドレスに***が表示されます。)

4) 空きスロットに 0 点を設定した場合

種別を“空き”に設定して、点数を“0 点”に設定しても、1 スロットを占有します。

あるスロット以降を占有しないようにするには、詳細モードでスロット数を設定してください。

( 4.1.1 項)

(c) 先頭 XY の設定

先頭 XY が未入力の場合は、CPU ユニットが自動で先頭 XY を割り付けます。

このため、下記の 1) または 2) の場合、各スロットの先頭 XY の設定と、CPU ユニットが割り付ける先頭 XY の設定が重なることがあります。

- 1) 先頭 XY 設定で前後の入出力番号を入れ換えて設定している
- 2) 先頭 XY を設定するスロットと設定しないスロット（自動割付のスロット）が混在している

先頭 XY が重なる場合の例を図 4.9 に示します。



図 4.9 先頭 XY が重なる I/O 割付設定

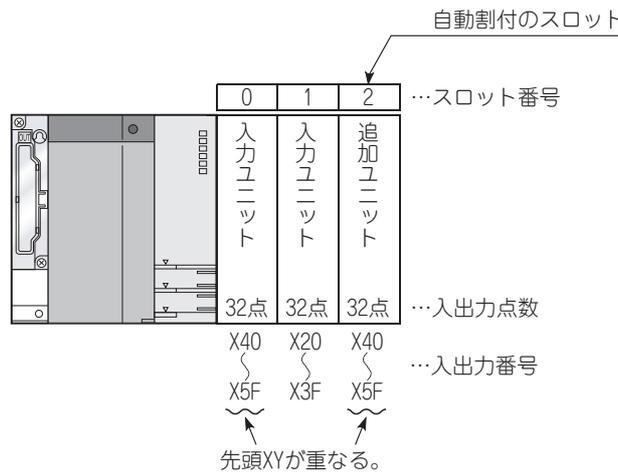


図 4.10 図 4.9 のように I/O 割付設定を行った場合の先頭 XY

各スロットの先頭 XY が重ならないように十分に注意してください。

上記例で先頭 XY が重ならないようにするには、追加ユニットの先頭 XY を設定してください。

(例：スロット 2 の“先頭 XY”に“0060”を入力する。)

ハイパフォーマンスモデル QCPU、プロセス CPU では、先頭 XY が重なっているスロットにユニットを装着すると、“SP.UNIT LAY ERR.”になります。

ベーシックモデル QCPU、二重化 CPU では、先頭 XY が重なっていると“SP.UNIT LAY ERR.”になります。(ユニットを装着していなくてもエラーになります。)

(d) AnS/A シリーズ対応の増設ベースユニットを使用する場合  注 4.2

AnS/A シリーズ対応の増設ベースユニット QA1S5 □ B, QA1S6 □ B, QA6 □ B を混在して使用する場合は、以下の注意事項に従って使用してください。

- 基本ベースユニットに近い方から Q5 □ B / Q6 □ B → QA1S5 □ B / QA1S6 □ B → QA6 □ B の順に接続してください。
- QA1S51B には、増設ケーブルコネクタ (OUT) がないため、QA6 □ B と混在させることはできません。
- ベースユニットに装着するユニットの入出力番号は、“Q シリーズ → A シリーズ” または “A シリーズ → Q シリーズ” の順にシリーズごとにまとめて割り付けてください。

ユニットの入出力番号を上記で割り付けない場合は、“SP.UNITLAY ERR.” になります。

QA6ADP+A1S5 □ B / A1S6 □ B を使用する場合は、下記マニュアルを参照してください。

 QA6ADP 形 QA 変換アダプタユニットユーザーズマニュアル

QA1S6ADP+A1S5 □ B / A1S6 □ B を使用する場合は、下記マニュアルを参照してください。

 QA1S6ADP 形 Q-AnS ベースユニット変換アダプタユーザーズマニュアル

 注 4.2

Basic

Process

Redundant

ベーシックモデル QCPU, プロセス CPU, 二重化 CPU では、AnS/A シリーズ対応の増設ベースユニットは使用できません。

4.2.3 入出力番号の割付け例

入出力番号の割付け例を下記に示します。

(1) 空きスロットの点数を 16 点から 32 点に変更する場合

現在空きスロットの位置（スロット No.3）に、将来 32 点の入力ユニットを装着したとき、スロット No.4 以降の入出力番号が変化しないように 32 点分の予約を行います。

(a) システム構成と設定前の入出力番号の割付け

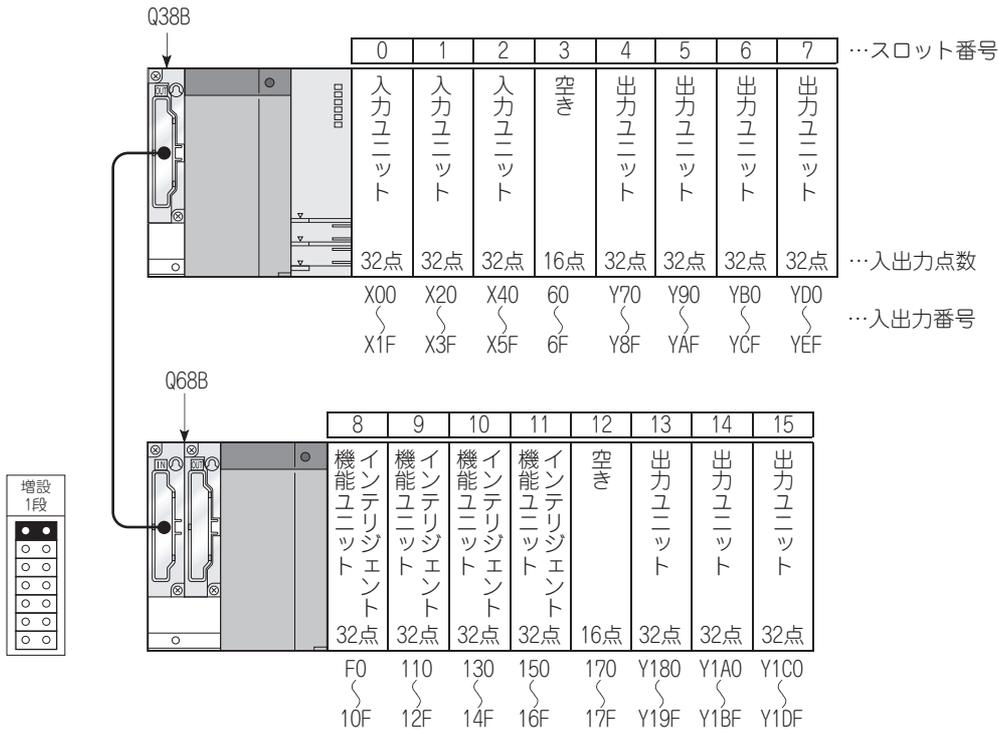


図 4.11 設定前の入出力番号の割付け（空きスロットの点数変更前）

(b) I/O 割付設定

PC パラメータの I/O 割付設定で、スロット No.3 の点数を “32 点” に設定します。

32点を選択します。
(種別を選択しない場合は
装着されているユニットの
種別になります。)



図 4.12 I/O 割付設定 (スロット 3 の空き点数を変更する場合)

(c) 設定後の入出力番号の割付け

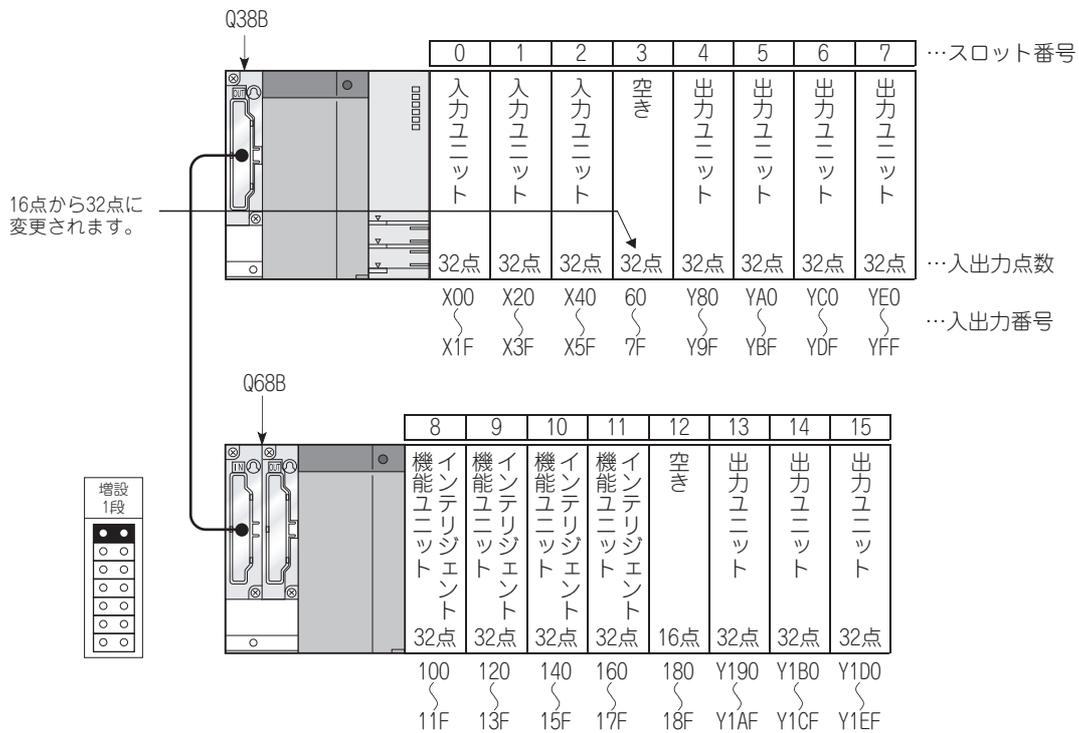


図 4.13 設定後の入出力番号 (空きスロットの空き点数変更後)

(2) スロットの入出力番号を変更

現在空きスロットの位置（スロット No.3）に 32 点の入力ユニットを装着することにより、スロット No.4 以降の入出力番号が変化しないように、スロット No.3 の入出力番号を X200 ~ 21F に変更します。

(a) システム構成と設定前の入出力番号の割付け

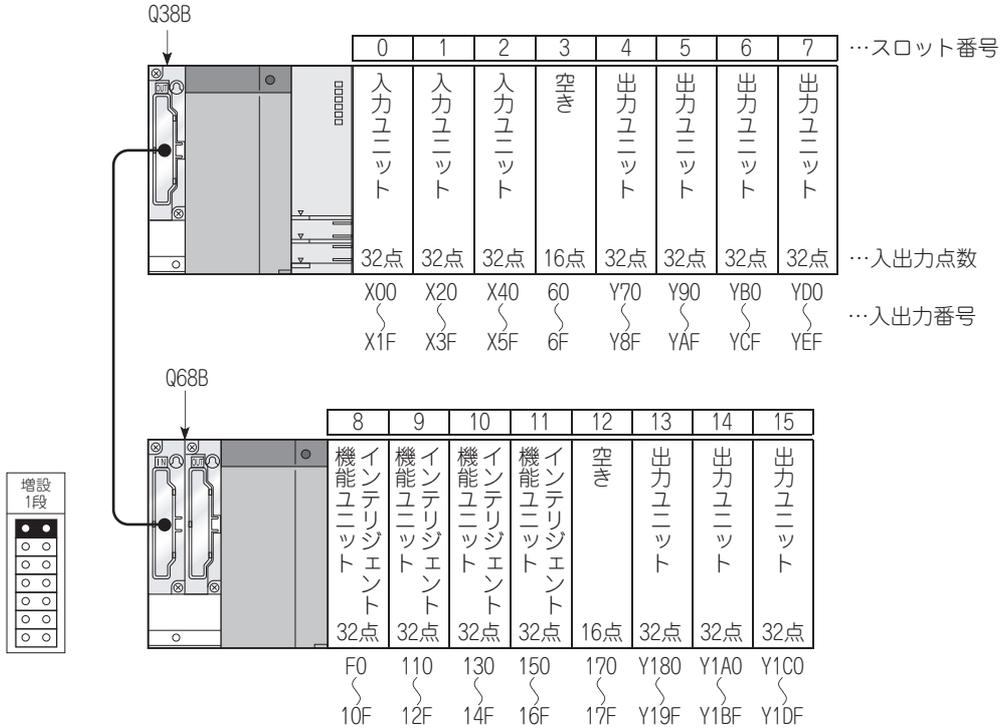


図 4.14 設定前の入出力番号の割付け（スロットの入出力番号変更前）

(b) I/O 割付設定

PC パラメータの I/O 割付設定で、スロット No.3 の先頭 XY を “200” に、スロット No.4 の先頭 XY を “70” に設定します。

先頭入出力番号を “200” に設定します。

先頭入出力番号を “70” に設定します。

（先頭入出力番号を設定しないと、3スロット目の次の入出力番号が割り付けられます。）



図 4.15 I/O 割付設定（スロット 3 の入出力番号を変更する場合）

4.2 入出力番号の割付け
4.2.3 入出力番号の割付け例

(c) 設定後の入出力番号の割付け

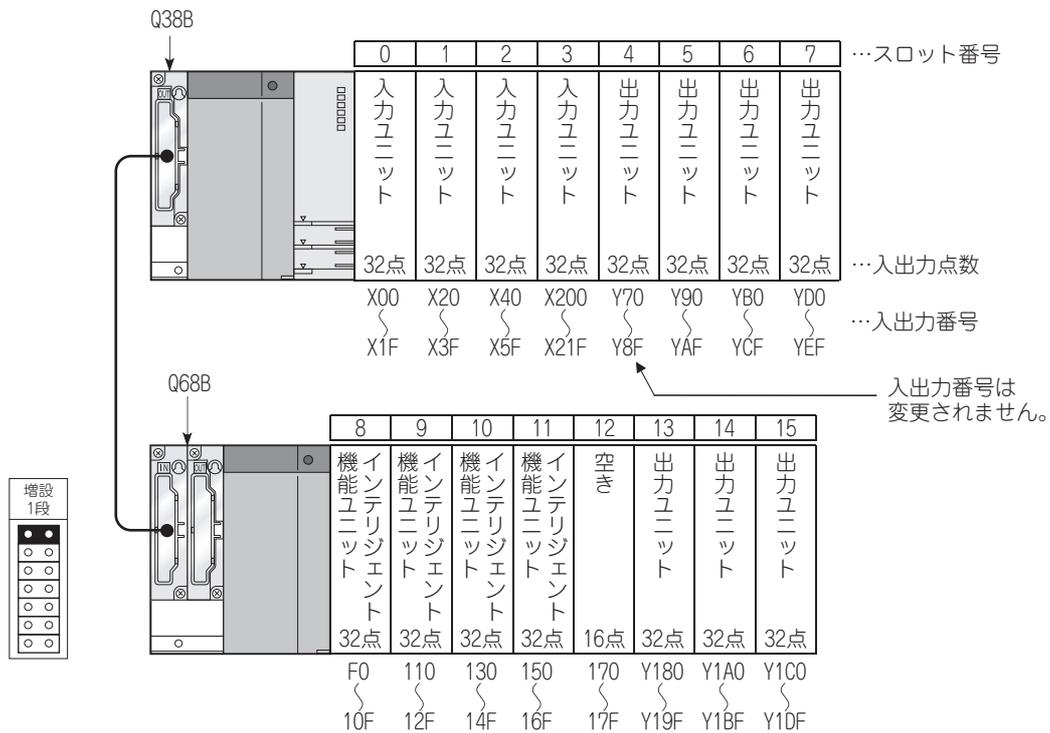


図 4.16 設定後の入出力番号の割付け (スロットの入出力番号変更後)

4.2.4 入出力番号の確認

GX Developer のシステムモニターで、実装されているユニットと入出力番号の確認ができます。

(☞ 6.20 節)

第5章 CPUユニットで取り扱うメモリとファイルについて

5.1 ベーシックモデル QCPU で取り扱うメモリ

5.1.1 メモリ構成と格納できるデータ

ベーシックモデル QCPU で取り扱うメモリと、格納できるデータについて説明します。

(1) メモリ構成

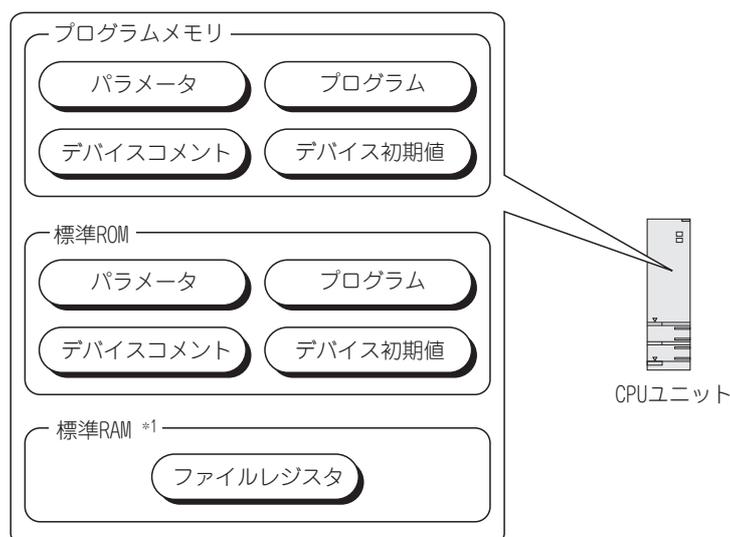


図 5.1 ベーシックモデル QCPU のメモリ構成

* 1 : Q00JCPU には、標準 RAM はありません。

(a) プログラムメモリ (☞ 5.1.2 項)

CPU ユニットが演算するためのプログラム、パラメータを格納するメモリです。

(b) 標準 ROM (☞ 5.1.3 項)

プログラム、パラメータなどのデータを格納するメモリです。
バッテリーバックアップなしでプログラム、パラメータを保存できます。

(c) 標準 RAM (☞ 5.1.4 項)

ファイルレジスタを使用するためのメモリです。

(2) 各メモリに格納できるデータ

各メモリに格納できるデータの一覧を表 5.1 に示します。

表 5.1 各メモリに格納できるデータ

項目	CPU ユニット内蔵メモリ			ファイル名 と拡張子
	プログラムメモリ	標準 RAM	標準 ROM	
	ドライブ 0 * 1	ドライブ 3 * 1	ドライブ 4 * 1	
パラメータ	◎	×	○	PARAM.QPA
インテリジェント機能ユニットパラメータ	○	×	○	IPARAM.QPA
シーケンスプログラム	◎ * 2	×	○ * 3	MAIN.QPG
SFC プログラム	◎ * 2	×	○ * 3	MAIN-SFC.QPG
ファイルレジスタ	×	○ * 4	×	MAIN.QDR
デバイスコメント	○ * 5	×	○ * 5	MAIN.QCD
デバイス初期値	○	×	○	MAIN.QDI
ラベルプログラム * 7	○	×	○	PROJINFO.CAB
ユーザ設定のシステムエリア * 6	○	×	×	—

◎：必須，○：格納可能，×：格納不可

* 1：ドライブ No. は、シーケンスプログラムや MC プロトコルを使用した外部機器などから書込み／読出しする対象メモリの指定に使用します。

GX Developer では、対象メモリ名で指定するため、ドライブ No. を意識する必要はありません。

* 2：シーケンスプログラム、ST プログラム、SFC プログラムのいずれかのデータが必要です。

* 3：標準 ROM に格納しているプログラムを実行する場合は、PC パラメータでプログラムメモリへのブート指定が必要です。

* 4：標準 RAM には、ファイルレジスタを 1 ファイルのみ格納できます。(☞ 9.7 節)

* 5：シーケンスプログラムの命令でデバイスコメントの読出しはできません。

* 6：システムで使用するエリアを設定します。(☞ 5.1.2 項 (2) (b))

* 7：ラベルプログラムの構成情報を格納したデータです。

ラベルプログラムについては、下記マニュアルを参照してください。

☞ GX Developer Version8 オペレーティングマニュアル

(3) メモリ容量とフォーマットの要否

各メモリの容量とフォーマットの要否を表 5.2 に示します。

フォーマットが必要なメモリは、使用前に必ず GX Developer でフォーマットしてください。

表 5.2 フォーマットの要否

	Q00JCPU	Q00CPU	Q01CPU	フォーマットの要否
プログラムメモリ	58k バイト	94k バイト	94k バイト	必要
標準 ROM	58k バイト	94k バイト	94k バイト	不要
標準 RAM	無	128k バイト * 1		必要 * 2

* 1：ベーシックモデル QCPU では、CPU ユニットのバージョンにより標準 RAM の容量が異なります。(☞ 付 2.1)

* 2：機能バージョン B 以降では、使用前にフォーマットする必要はありません。

(GX Developer から機能バージョン B 以降の CPU ユニットに標準 RAM のフォーマットを行うと、GX Developer にエラーコード：4150_H が表示されます。)

Point

データを各メモリに書き込む場合、書き込む CPU ユニットとメモリアreaにより、格納する容量の単位が異なります。

(☞ 5.4.4 項)

5.1.2 プログラムメモリについて

(1) プログラムメモリとは

CPUユニットが演算するためのプログラム、パラメータを格納するメモリです。

Point

格納するデータの総容量が、プログラムメモリの容量を超える場合、以下を行ってください。

- ユーザ設定のシステムエリアを減らす
- プログラム以外のデータを、標準ROMに移すことを検討する

(2) プログラムメモリの使用前に

必ずGX Developerでプログラムメモリのフォーマットを行ってください。

(a) フォーマットの実行

GX Developerの[オンライン] → [PCメモリフォーマット]で行います。

対象メモリは“プログラムメモリ/デバイスメモリ”を選択します。

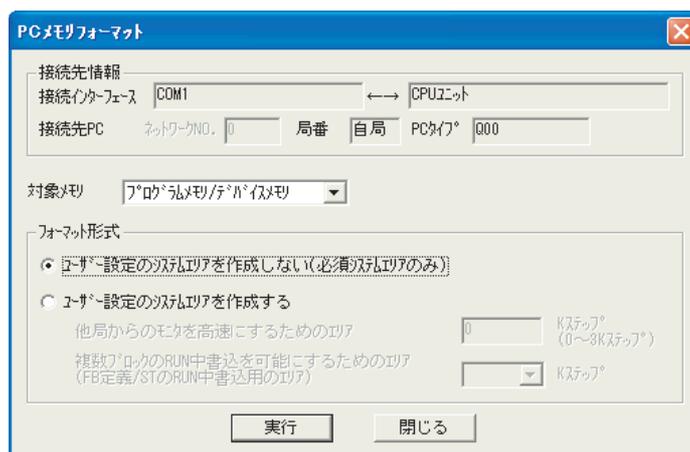


図 5.2 プログラムメモリのフォーマット

(b) ユーザ設定のシステムエリアの作成

プログラムメモリをフォーマットする場合、ユーザ設定のシステムエリアの容量を設定します。

1) ユーザ設定のシステムエリアを作成しない（必須システムエリアのみ）

ユーザ設定のシステムエリアを作成せずにフォーマットを行います。

2) ユーザ設定のシステムエリアを作成する

フォーマットのときにユーザ設定のシステムエリアを作成します。

ユーザ設定のシステムエリアには、表 5.3 のものがあります。

表 5.3 ユーザ設定のシステムエリアの形式

システムエリア形式	内容
他局からのモニタを高速にするためのエリア	本エリアを設定すると、シリアルコミュニケーションユニットなどに接続されている GX Developer からのモニタが早くなります。 本エリアは、複数箇所から同時にモニタを行う場合、シリアルコミュニケーションユニットなどに接続されている GX Developer からのモニタデータ登録に使用します。
複数ブロックの RUN 中書込を可能にするためのエリア（FB 定義 / ST の RUN 中書込用のエリア）	本エリアを設定すると、複数ブロックの RUN 中書込が可能になります。 RUN 中書込み可能なブロック数については、下記マニュアルを参照してください。 ☞ GX Developer Version 8 オペレーティングマニュアル

Point

ユーザ設定のシステムエリアを作成すると、エリアを作成したステップ数分だけ使用可能なエリアが減少します。

(c) フォーマット後のメモリ容量の確認

GX Developer の [オンライン] → [PC 読出] で確認できます。

1) PC 読出画面の対象メモリで、“プログラムメモリ / デバイスマemory” を選択します。

2) **空き容量** ボタンをクリックします。

3) 全空容量欄にメモリ容量が表示されます。

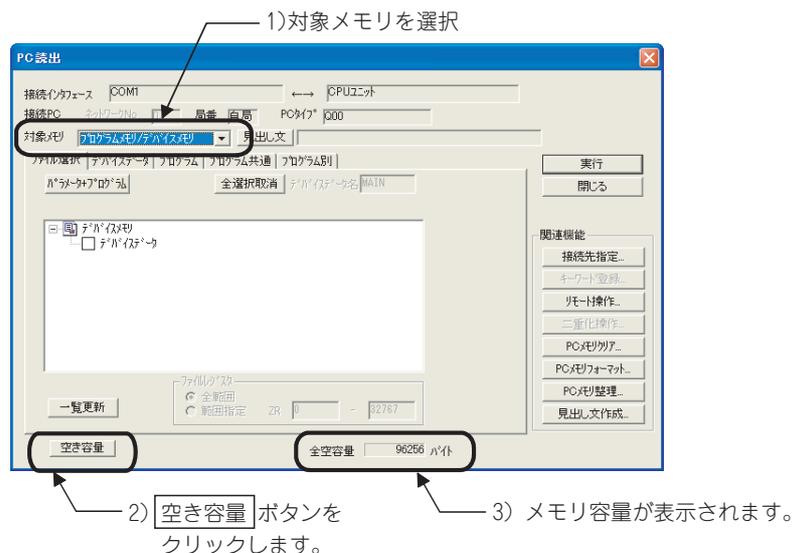


図 5.3 メモリ容量の確認手順

(3) プログラムメモリへの書込み

GX Developer の [オンライン] → [PC 書込] で行います。

PC 書込画面の対象メモリで、“プログラムメモリ/デバイスメモリ” を選択します。

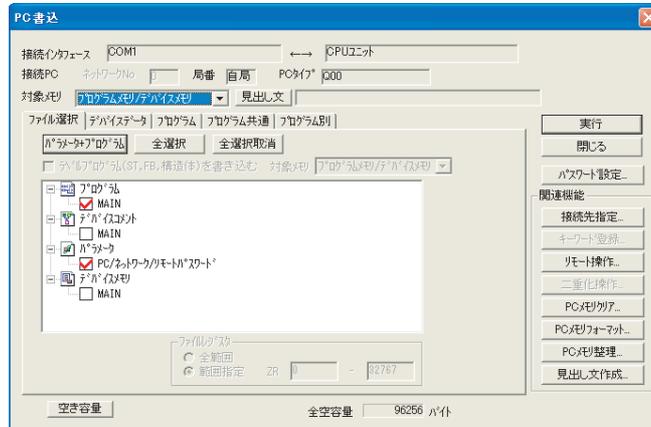


図 5.4 PC 書込画面

Point

ファイルサイズには、最小単位があります。(☞ 5.4.4 項)
占有するメモリ容量は、実際のファイルサイズよりも大きくなる場合があります。

5.1.3 標準 ROM について

(1) 標準 ROM とは

パラメータ、プログラムなどのデータを格納するメモリです。
バッテリーバックアップなしでプログラム、パラメータを保存できます。

(2) メモリ容量の確認

GX Developer の [オンライン] → [PC 読出] で確認できます。

- 1) PC 読出画面の対象メモリで、“標準 ROM” を選択します。
- 2) **空き容量** ボタンをクリックします。
- 3) 全空容量欄にメモリ容量が表示されます。

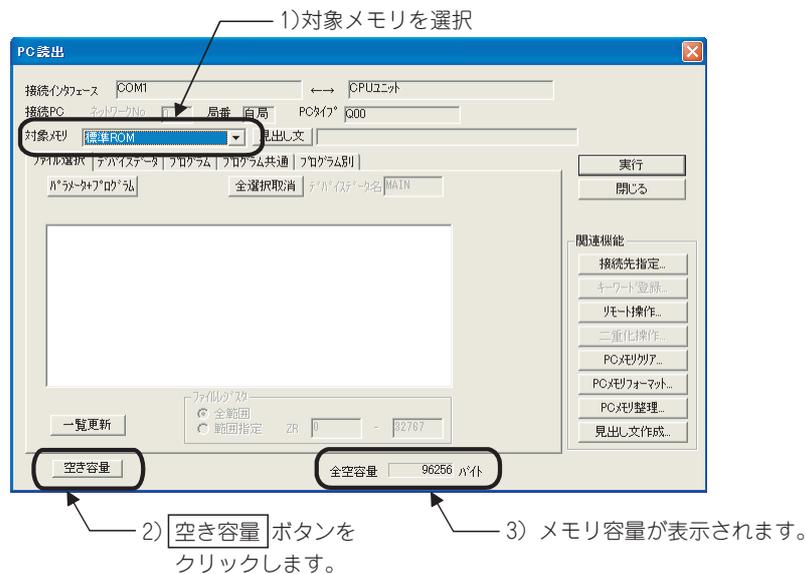


図 5.5 メモリ容量の確認手順

(3) 標準 ROM への書込み

GX Developer の [オンライン] → [PC 書込 (フラッシュ ROM)] → [プログラムメモリの ROM 化] で行います。(☞ 5.1.5 項)

Point

ファイルサイズには最小単位があります。(☞ 5.4.4 項)
占有するメモリ容量は、実際のファイルサイズよりも大きくなる場合があります。

5.1.4 標準 RAM について 注 5.1

(1) 標準 RAM とは

ファイルレジスタ用のメモリです。

(2) 標準 RAM の使用前に

機能バージョン A のベーシックモデル QCPU では、必ず GX Developer で標準 RAM のフォーマットを行ってください。

機能バージョン B 以降のベーシックモデル QCPU では、使用前の標準 RAM のフォーマットは不要ですが、標準 RAM のクリア操作が必要です。標準 RAM (ファイルレジスタ) のクリア操作については、9.7 節 (3) を参照してください。

(a) フォーマットの実行

GX Developer の [オンライン] → [PC メモリフォーマット] で行います。

対象メモリは“標準 RAM”を選択します。

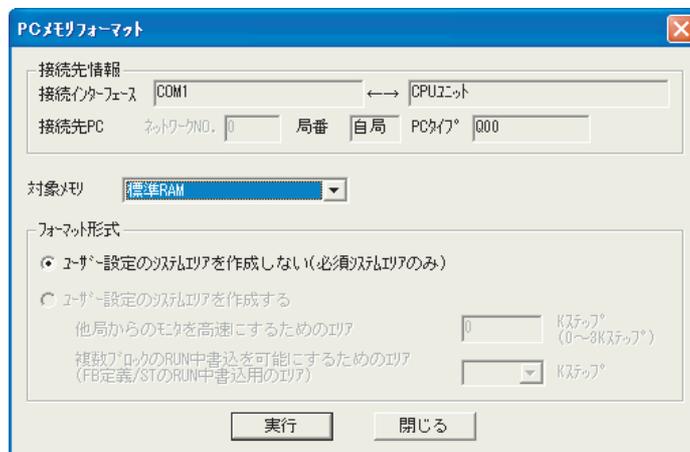


図 5.6 標準 RAM のフォーマット



注 5.1

Basic

Q00JCPU には、標準 RAM はありません。

(b) フォーマット後のメモリ容量の確認

GX Developer の [オンライン] → [PC 読出] で確認できます。

- 1) PC 読出画面の対象メモリで、“標準 RAM” を選択します。
- 2) **空き容量** ボタンをクリックします。
- 3) 全空容量欄にメモリ容量が表示されます。

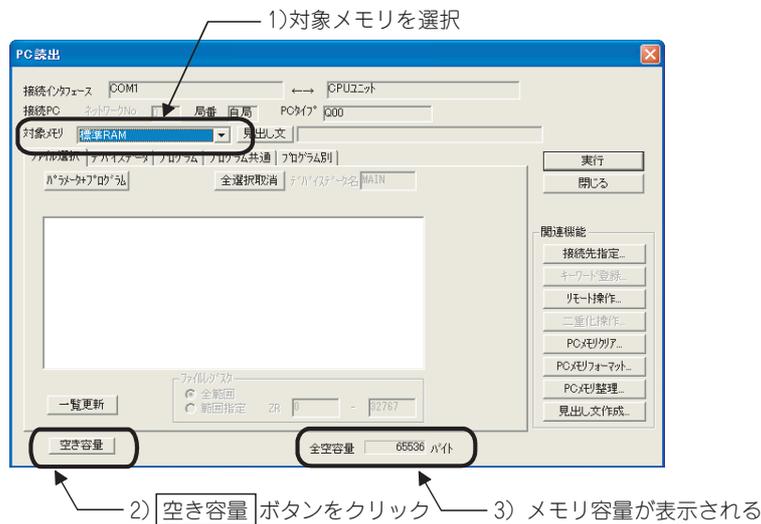


図 5.7 メモリ容量の確認手順

(3) 標準 RAM への書き込み

GX Developer の [オンライン] → [PC 書込] で行います。

PC 書込画面の対象メモリで、“標準 RAM” を選択します。

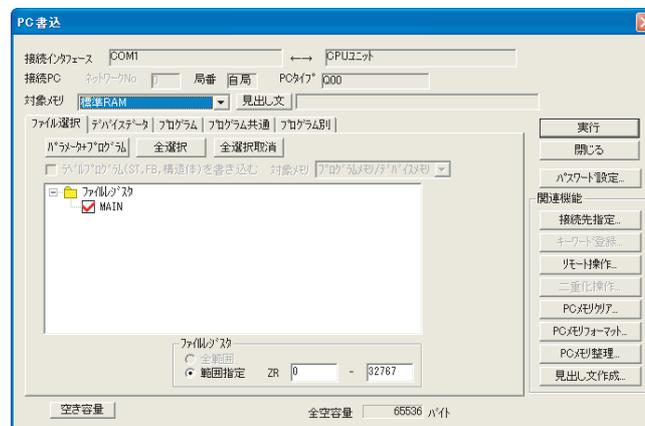


図 5.8 PC 書込画面

Point

ファイルサイズには最小単位があります。(☞ 5.4.4 項)
占有するメモリ容量は、実際のファイルサイズよりも大きくなる場合があります。

5.1.5 標準 ROM のプログラムの実行（ブート運転）と書き込み

標準 ROM に格納しているプログラムは演算を実行できないため、プログラムメモリにブートして使用します。

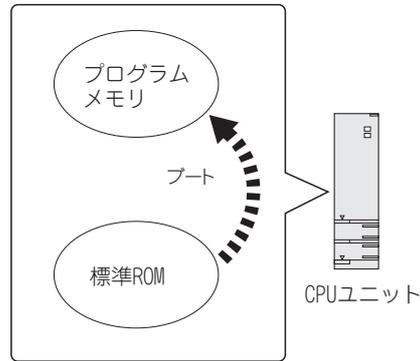


図 5.9 ブート運転

(1) プログラムの実行（ブート運転）

下記の手順で行います。

(a) GX Developer によるプログラムの作成

ブート運転を行うプログラムを作成します。

(b) ブート運転を行うための設定

PC パラメータのブートファイル設定により，“標準 ROM からのブートを行う” にチェックを入れます。

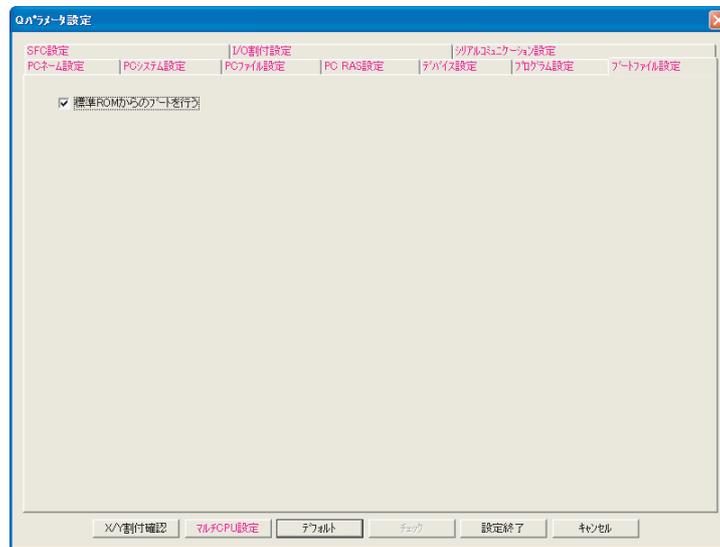


図 5.10 ブートファイル設定

(c) 標準 ROM への書込み

書込み前の操作と書込み方法について説明します。

1) 書込み前に

標準 ROM にファイルを書き込むと、標準 ROM に格納されている全ファイルは自動的に削除されるため、書込み前には、あらかじめ格納されているファイルを“PC 読出”で保存するなどしてください。

(☞ 5.1.3 項)

2) 書込み手順

GX Developer の [オンライン] → [PC 書込] でプログラムメモリにファイルを書き込みます。次にプログラムメモリに書き込んだファイルを、[オンライン] → [PC 書込 (フラッシュ ROM)] → [プログラムメモリの ROM 化] により、標準 ROM に書き込みます。

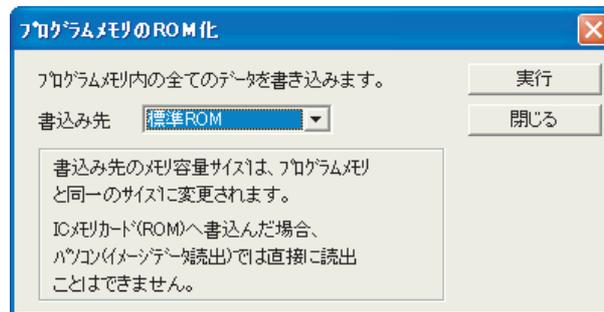


図 5.11 プログラムメモリの ROM 化

(d) プログラムの実行

RUN/STOP/RESET スイッチでリセットすると、標準 ROM からのブートを行います。

ブートが正常に完了しているかは、特殊リレー (SM660) により確認できます。

特殊リレー (SM660) の詳細については、下記マニュアルを参照してください。

(☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編))

(2) 標準 ROM のファイルに対する追加・変更

追加・変更は、下記の方法で行ってください。(格納されているファイルに直接追加・変更はできません。)

- 1) GX Developer の [オンライン] → [PC 読出] で、標準 ROM の全ファイルを読み出します。
- 2) 読み出したファイルに対して、追加・変更を行います。
- 3) 追加・変更を行ったファイルを、プログラムメモリに書き込みます。
- 4) [オンライン] → [PC 書込 (フラッシュ ROM)] → [プログラムメモリの ROM 化] で、標準 ROM にファイルをコピーします。

(3) ブート運転を中止するための操作

ブート運転を中止して、プログラムメモリに書き込まれたプログラム、パラメータを実行するには、下記の操作を行ってください。

- 1) プログラムメモリのフォーマットを行います。
- 2) GX Developer の [オンライン] → [PC 書込 (フラッシュ ROM)] → [プログラムメモリの ROM 化] を行います。(標準 ROM のプログラム、パラメータが消去されます。)
- 3) プログラムメモリにプログラム、パラメータの書込みを行います。

(4) 注意事項

(a) 標準 ROM に格納するファイルについて

ブート運転を行う場合、下記のファイルは標準 ROM に格納してください。

- パラメータ
- インテリジェント機能ユニットパラメータ
- プログラム * 1
- デバイスコメント
- デバイス初期値

* 1: シーケンスプログラム、ST プログラム、SFC プログラムのいずれかが必要です。

(b) ブート運転時の RUN 中書込みについて

標準 ROM からのブート運転時にプログラムメモリのプログラムを RUN 中書込みしても、ブート元の標準 ROM のプログラムには変更内容は反映されません。
STOP 状態にしてから標準 ROM への書込みを行ってください。

(c) 電源 OFF → ON またはリセット解除時にプログラムメモリの内容が変わる場合

プログラムメモリにプログラムを書き込んで CPU ユニットの電源 OFF → ON またはリセットを行ったとき、プログラムメモリの内容が書き変わる場合はブート運転になっている可能性があります。
本項 (3) を参照し、ブート運転を中止してください。

(d) GX Developer の通信タイムチェック時間の設定について

標準 ROM にファイルを書き込む場合、GX Developer の通信タイムチェック時間が 180 秒以下のときは、180 秒でチェックを行います。
エラーが発生する場合は、接続先指定の設定で通信タイムチェック時間を延ばしてください。

(e) CC-Link 経由の他局の GX Developer から書き込みを行う場合

標準 ROM へのファイルの書き込みは、処理に時間がかかるため、CC-Link の CPU 監視時間設定 (SW0A) を 180 秒以上に設定してください。

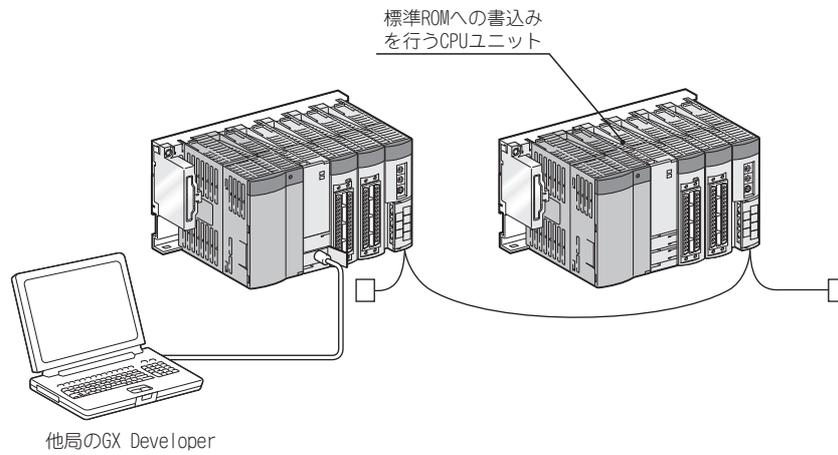


図 5.12 他局の GX Developer からの書き込み (CC-Link 経由)

備考

通信タイムチェック時間の設定および CPU 監視時間の設定については、下記マニュアルを参照してください。

☞ GX Developer Version8 オペレーティングマニュアル

5.2 ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU で取り扱うメモリ

5.2.1 メモリ構成と格納できるデータ

ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU で取り扱うメモリと、格納できるデータについて説明します。

(1) メモリ構成

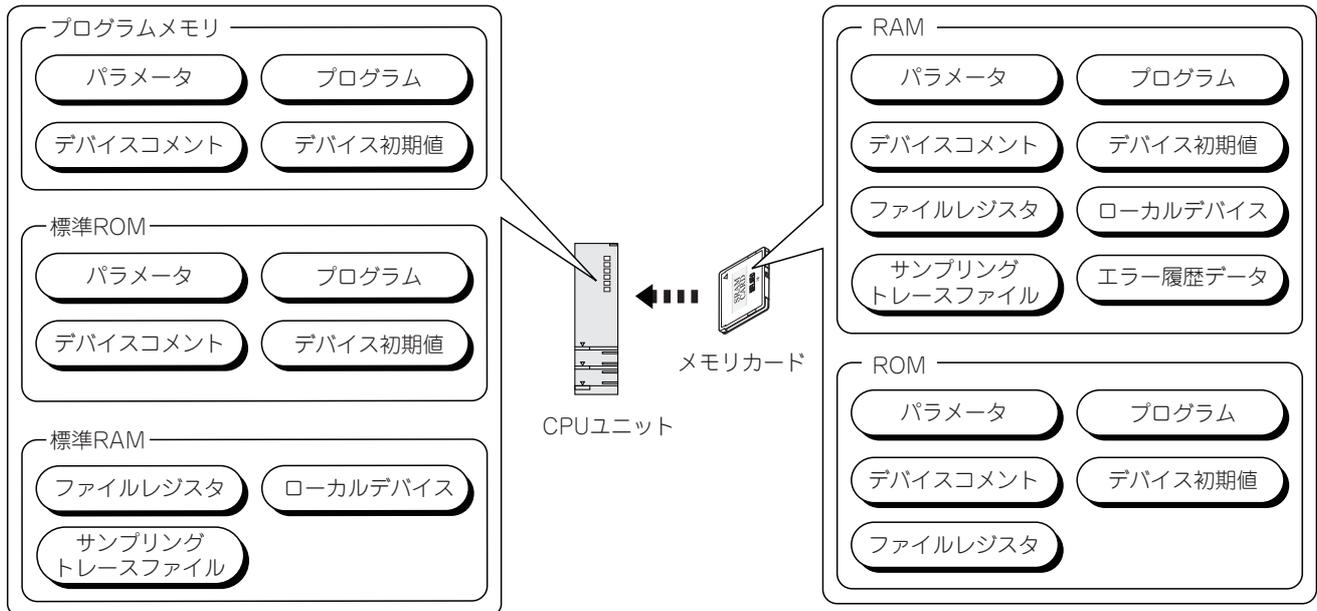


図 5.13 ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU のメモリ構成

(a) プログラムメモリ (☞ 5.2.2 項)

CPU ユニットが演算するためのプログラム、パラメータを格納するメモリです。

(b) 標準 ROM (☞ 5.2.3 項)

プログラム、パラメータなどのデータを格納するメモリです。
バッテリーバックアップなしでプログラム、パラメータを保存できます。

(c) 標準 RAM (☞ 5.2.4 項)

メモリカードを装着せずにファイルレジスタ、ローカルデバイス、サンプリングトレースファイルを使用するためのメモリです。

(d) メモリカード (☞ 5.2.5 項)

CPU ユニットの内蔵メモリの拡張で使用します。
使用できるメモリカードには、SRAM カード、Flash カード、ATA カードがあります。

(2) 各メモリに格納できるデータ

各メモリに格納できるデータの一覧を表 5.4 に示します。

表 5.4 各メモリに格納できるデータ

項目	CPU ユニットの内蔵メモリ			メモリカード (RAM)	メモリカード (ROM)		ファイル名と拡張子	備考
	プログラムメモリ	標準 RAM	標準 ROM	SRAM カード	Flash カード	ATA カード		
	ドライブ 0 * 1	ドライブ 3 * 1	ドライブ 4 * 1	ドライブ 1 * 1	ドライブ 2 * 1			
パラメータ	○	×	○	○	○	○	PARAM.QPA	1 データ / ドライブ
インテリジェント機能ユニットパラメータ * 2	○	×	○	○	○	○	IPARAM.QPA	1 データ / ドライブ
プログラム	◎	×	○* 3	○* 3	○* 3	○* 3	***.QPG	—
デバイスコメント	○* 4	×	○* 5	○* 5	○* 5	○* 5	***.QCD	—
デバイス初期値	○	×	○	○	○	○	***.QDI	—
デバイスデータ	×	×	×	×	×	×	***.QST	—
ファイルレジスタ	×	○* 6 * 7	×	○	○* 8	×	***.QDR	—
ローカルデバイス	×	○* 6	×	○	×	×	***.QDL	1 データ / CPU ユニット
サンプリングトレースファイル	×	○* 6	×	○	×	×	***.QTD	—
ブート設定ファイル	○	×	○	○	○	○	AUTOEXEC.QBT	—
リモートパスワード	○	×	○	○	○	○	00000000.QTM	—
エラー履歴データ	×	×	×	○	×	×	***.QFD	—
PC ユーザデータ	×	×	×	×	×	○* 9	***.CSV/BIN	—
ラベルプログラム * 11	○	×	○	×	×	○	PROJINFO.CAB	—
ユーザ設定のシステムエリア * 10	○	×	×	×	×	×	—	—

◎：必須，○：格納可能，×：格納不可

- * 1：ドライブ No. は、シーケンスプログラムや MC プロトコルを使用した外部機器などから書込み / 読出しする対象メモリの指定に使用します。
GX Developer では、対象メモリ名で指定するため、ドライブ No. を意識する必要はありません。
- * 2：インテリジェント機能ユニットパラメータはパラメータと同一ドライブに格納してください。
異なるドライブに格納した場合、インテリジェント機能ユニットパラメータは有効となりません。
- * 3：標準 ROM / メモリカードに格納しているプログラムを実行する場合は、PC パラメータでプログラムメモリへのブート指定が必要です。
- * 4：シーケンスプログラムの命令でデバイスコメントの読出しはできません。
- * 5：シーケンスプログラムからの読出しには数スキャンが必要です。
- * 6：標準 RAM には、ファイルレジスタ、ローカルデバイスおよびサンプリングトレースファイルを各 1 ファイル格納できます。
- * 7：格納可能なファイルレジスタの点数については、9.7 節を参照してください。
- * 8：シーケンスプログラムでは読出しのみ可能です。シーケンスプログラムからの書込みはできません。
- * 9：下記命令によりデータの読み書きができます。
 - ・ SP.FREAD (メモリカードの指定ファイルからの一括読出し。)
 - ・ SP.FWRITE (メモリカードの指定ファイルへの一括書込み。)
- * 10：システムで使用するエリアを設定します。(☞ 5.2.2 項 (2) (b))
- * 11：ラベルプログラムの構成情報を格納したデータです。
ラベルプログラムについては、下記マニュアルを参照してください。
☞ GX Developer Version8 オペレーティングマニュアル

(3) メモリ容量とフォーマットの要否

各メモリの容量とフォーマットの要否を表 5.5 に示します。

フォーマットが必要なメモリは、使用前に必ず GX Developer でフォーマットしてください。

表 5.5 メモリ容量とフォーマットの要否

	Q02CPU * 2	Q02HCPU * 2	Q06HCPU * 2	Q12HCPU * 2	Q25HCPU * 2	フォーマットの要否
プログラムメモリ	112k バイト (28k ステップ)	112k バイト (28k ステップ)	240k バイト (60k ステップ)	496k バイト (124k ステップ)	1008k バイト (252k ステップ)	必要* 1
標準 ROM	112k バイト	112k バイト	240k バイト	496k バイト	1008k バイト	不要
標準 RAM	64k バイト	128k バイト		256k バイト		必要* 1
メモリカード	SRAM カード	Q2MEM-1MBSN, Q2MEM-1MBS : 1M バイト Q2MEM-2MBSN, Q2MEM-2MBS : 2M バイト Q3MEM-4MBS : 4M バイト				必要 (GX Developer で行う。)
	Flash カード	Q2MEM-2MBF : 2M バイト Q2MEM-4MBF : 4M バイト				不要
	ATA カード	Q2MEM-8MBA : 8M バイト Q2MEM-16MBA : 16M バイト Q2MEM-32MBA : 32M バイト				必要 (GX Developer で行う。)

	Q02PHCPU	Q06PHCPU	Q12PHCPU	Q25PHCPU	Q12PRHCPU	Q25PRHCPU	フォーマットの要否
プログラムメモリ	112k バイト (28k ステップ)	240k バイト (60k ステップ)	496k バイト (124k ステップ)	1008k バイト (252k ステップ)	496k バイト (124k ステップ)	1008k バイト (252k ステップ)	必要* 1
標準 ROM	112k バイト	240k バイト	496k バイト	1008k バイト	496k バイト	1008k バイト	不要
標準 RAM	128k バイト		256k バイト				必要* 1
メモリカード	SRAM カード	Q2MEM-1MBSN, Q2MEM-1MBS : 1M バイト Q2MEM-2MBSN, Q2MEM-2MBS : 2M バイト Q3MEM-4MBS : 4M バイト					必要 (GX Developer で行う。)
	Flash カード	Q2MEM-2MBF : 2M バイト Q2MEM-4MBF : 4M バイト					不要
	ATA カード	Q2MEM-8MBA : 8M バイト Q2MEM-16MBA : 16M バイト Q2MEM-32MBA : 32M バイト					必要 (GX Developer で行う。)

* 1 : 初期状態またはバッテリー切れでメモリの内容が不定になると、CPU ユニットの電源 OFF → ON またはリセット時に自動でフォーマットされます。

* 2 : ハイパフォーマンスモデル QCPU では、CPU ユニットのバージョンにより標準 RAM の容量が異なります。(付 2.2)

Point

- ファイルを各メモリに書き込む場合、書き込む CPU ユニットとメモリアreaにより、格納する容量の単位が異なります。(付 5.4.4 項)
- メモリ容量の計算で 1 ステップは 4 バイトです。

5.2.2 プログラムメモリについて

(1) プログラムメモリとは

CPU ユニットが演算するためのプログラム、パラメータを格納するメモリです。

Point

格納するデータの総容量が、プログラムメモリの容量を超える場合、以下を行ってください。

- ユーザ設定のシステムエリアを減らす
- プログラム以外のデータを、標準 ROM またはメモリカードに移すことを検討する

(2) プログラムメモリの使用前に

必ず GX Developer でプログラムメモリのフォーマットを行ってください。

(a) フォーマットの実行

GX Developer の [オンライン] → [PC メモリフォーマット] で行います。

対象メモリは “プログラムメモリ/デバイスメモリ” を選択します。

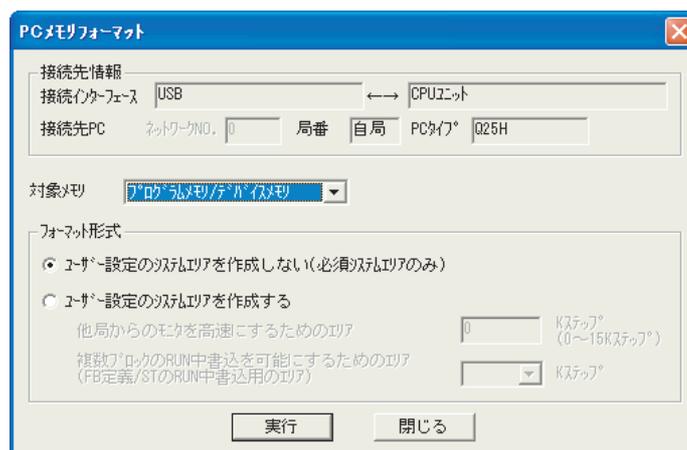


図 5.14 プログラムメモリのフォーマット

(b) ユーザ設定のシステムエリアの作成

プログラムメモリをフォーマットする場合、ユーザ設定のシステムエリアの容量を設定します。

1) ユーザ設定のシステムエリアを作成しない（必須システムエリアのみ）

ユーザ設定のシステムエリアを作成せずにフォーマットを行います。

2) ユーザ設定のシステムエリアを作成する

フォーマットのときにユーザ設定のシステムエリアを作成します。

ユーザ設定のシステムエリアには、表 5.6 のものがあります。

表 5.6 ユーザ設定のシステムエリアの形式

システムエリア形式	内容
他局からのモニタを高速にするためのエリア	本エリアを設定すると、シリアルコミュニケーションユニットなどに接続されている GX Developer からのモニタが早くなります。 本エリアは、GX Developer で RS-232 と USB を同時に使用する場合、シリアルコミュニケーションユニットなどに接続されている GX Developer からのモニタデータ登録に使用します。
複数ブロックの RUN 中書込を可能にするためのエリア（FB 定義 / ST の RUN 中書込用のエリア）	本エリアを設定すると、複数ブロックの RUN 中書込が可能になります。 RUN 中書込み可能なブロック数については、下記マニュアルを参照してください。 ☞ GX Developer Version 8 オペレーティングマニュアル

Point

ユーザ設定のシステムエリアを作成すると、エリアを作成したステップ数分だけ使用可能なエリアが減少します。

(c) フォーマット後のメモリ容量の確認

GX Developer の [オンライン] → [PC 読出] で確認できます。

- 1) PC 読出画面の対象メモリで、“プログラムメモリ / デバイスメモリ” を選択します。
- 2) **空き容量** ボタンをクリックします。
- 3) 全空容量欄にメモリ容量が表示されます。

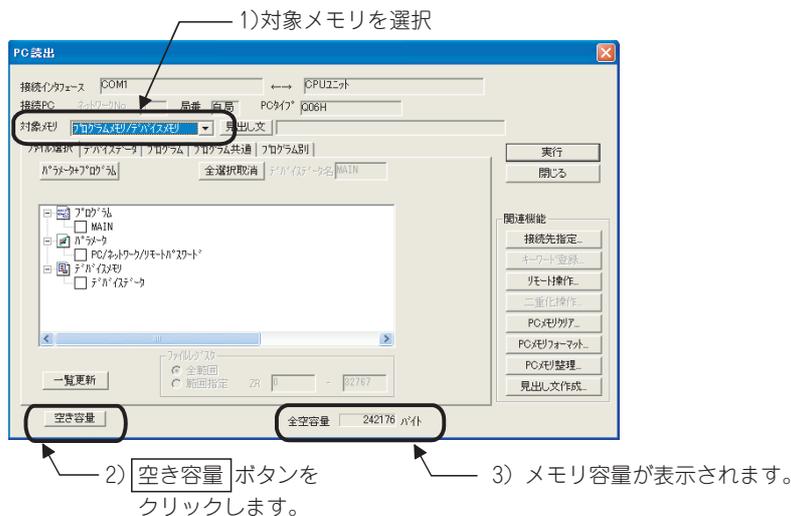


図 5.15 メモリ容量の確認手順

(3) プログラムメモリへの書込み

GX Developer の [オンライン] → [PC 書込] で行います。

PC 書込画面の対象メモリで、“プログラムメモリ/デバイスメモリ” を選択します。

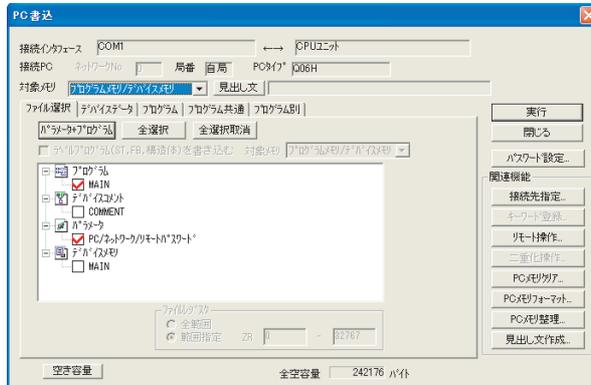


図 5.16 PC 書込画面

Point

ファイルサイズには、最小単位があります。(参照 5.4.4 項)

占有するメモリ容量は、実際のファイルサイズよりも大きくなる場合があります。

ファイル数が多くなると、占有するメモリ容量と実際のファイルサイズとの差は大きくなりますので注意してください。

5.2.3 標準 ROM について

(1) 標準 ROM とは

パラメータ、プログラムなどのデータを格納するメモリです。
バッテリーバックアップなしでプログラム、パラメータを保存できます。

(2) メモリ容量の確認

GX Developer の [オンライン] → [PC 読出] で確認できます。

- 1) PC 読出画面の対象メモリで、“標準 ROM” を選択します。
- 2) **空き容量** ボタンをクリックします。
- 3) 全空容量欄にメモリ容量が表示されます。

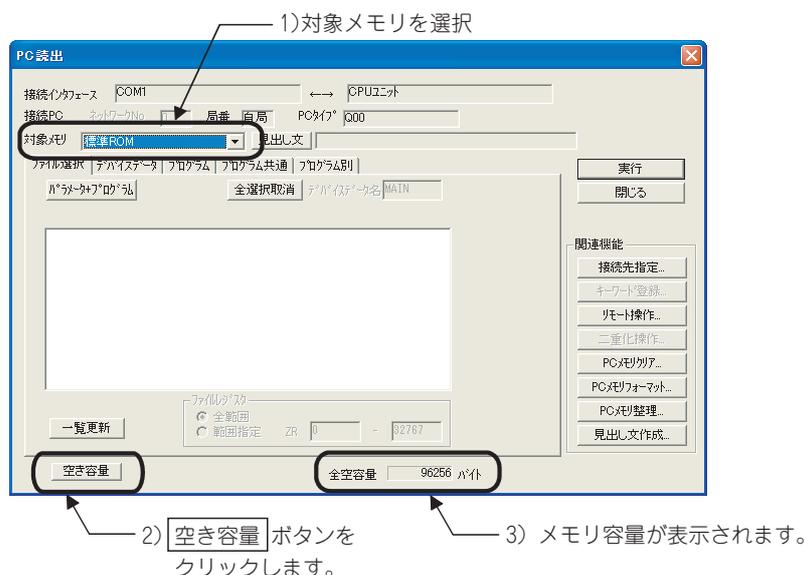


図 5.17 メモリ容量の確認手順

(3) 標準 ROM への書込み

下記に示す 3 種類があります。

- GX Developer の [オンライン] → [PC 書込 (フラッシュ ROM)] → [プログラムメモリの ROM 化] で、プログラムメモリのデータを一括して標準 ROM にコピーする方法 (☞ 5.2.6 項)
- GX Developer の [オンライン] → [PC 書込 (フラッシュ ROM)] → [PC 書込 (フラッシュ ROM)] で書き込む方法 (☞ 5.2.6 項)
- メモリカード → 標準 ROM 全データ自動書込み機能を使用して書き込む方法 (☞ 5.2.7 項)

Point

ファイルサイズには最小単位があります。(☞ 5.4.4 項)
占有するメモリ容量は、実際のファイルサイズよりも大きくなる場合があります。
ファイル数が多くなると、占有するメモリ容量と実際のファイルサイズとの差は大きくなりますので注意してください。

5.2.4 標準 RAM について

(1) 標準 RAM とは

メモ리카ードを装着せずにファイルレジスタ、ローカルデバイス、サンプリングトレースファイルを使用するためのメモリです。

標準 RAM をファイルレジスタとして使用した場合は、データレジスタと同様に高速なアクセスができます。

Point

- 標準 RAM に格納するファイルの容量が、標準 RAM の容量を超える場合、以下を行ってください。
 - ・メモ리카ードにファイルを格納する
 - ・ファイルレジスタ、ローカルデバイス、サンプリングトレースの点数を減らすことを検討する

ただし、メモ리카ードにファイルレジスタを格納した場合は、標準 RAM に比べアクセス速度が遅くなりますので注意してください。

- サンプリングトレースファイルを標準 RAM に格納する場合は、CPU ユニットと GX Developer のバージョンを確認してください。(付 2)

(2) 標準 RAM の使用前に

必ず GX Developer で標準 RAM のフォーマットを行ってください。

(a) フォーマットの実行

GX Developer の [オンライン] → [PC メモリフォーマット] で行います。

対象メモリは“標準 RAM”を選択します。

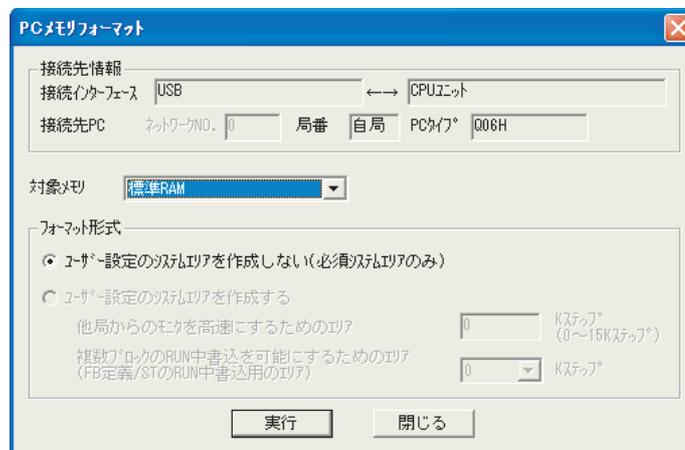


図 5.18 標準 RAM のフォーマット

(b) フォーマット後のメモリ容量の確認

GX Developer の [オンライン] → [PC 読出] で確認できます。

- 1) PC 読出画面の対象メモリで、“標準 RAM” を選択します。
- 2) **空き容量** ボタンをクリックします。
- 3) 全空容量欄にメモリ容量が表示されます。

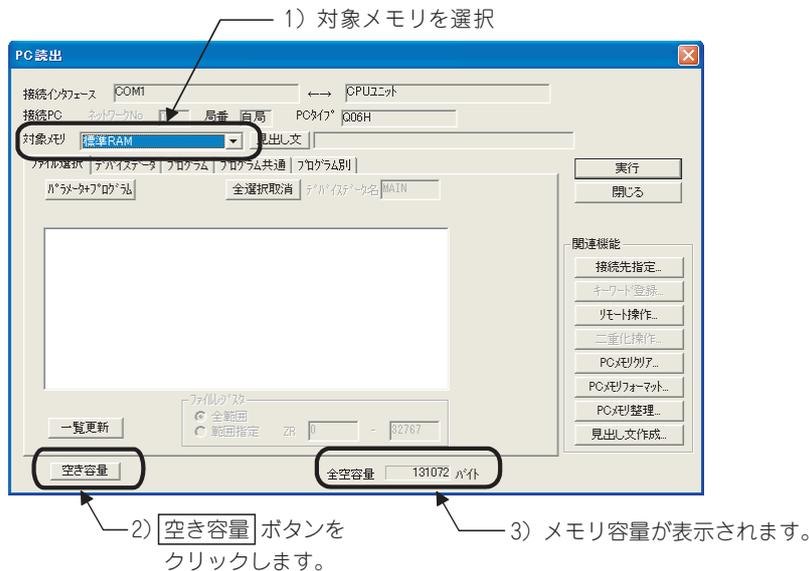


図 5.19 メモリ容量の確認手順

(3) 標準 RAM への書込み

GX Developer の [オンライン] → [PC 書込] で行います。

PC 書込画面の対象メモリで、“標準 RAM” を選択します。

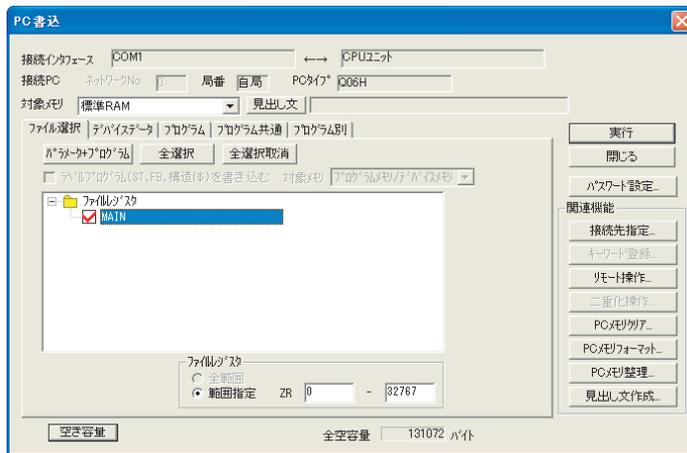


図 5.20 PC 書込画面

Point

ファイルサイズには最小単位があります。(5.4.4 項)
 占有するメモリ容量は、実際のファイルサイズよりも大きくなる場合があります。
 ファイル数が多くなると、占有するメモリ容量と実際のファイルサイズとの差は大きくなりますので注意してください。

5.2.5 メモリカードについて

(1) メモリカード

CPU ユニットのメモリを拡張する場合に使用します。
使用できるメモリカードには、下記があります。

- SRAM カード
- Flash カード
- ATA カード

(a) SRAM カード

シーケンスプログラムで SRAM カードに格納したファイルレジスタの書き込み／読出しが可能です。
SRAM カードは、下記の場合に使用します。

- ファイルレジスタの点数が、標準 RAM の容量よりも多い場合
- サンプリグトレース機能を使用する場合 (☞ 6.14 節)
- エラー履歴データを 17 点以上保存したい場合 (☞ 6.18.2 項)

SRAM カードをファイルレジスタとして使用すると、シーケンスプログラムでは、最大 1018k 点の書き込み／読出しができます。

(b) Flash カード

データの書き込みは GX Developer で行い、シーケンスプログラムで読み出して使用します。(シーケンスプログラムでは読出しのみ可能です。)

データの変更を行わない場合に使用します。

ファイルレジスタを最大 1018k 点分格納できます。

(c) ATA カード

PC ユーザデータ (汎用データ) で使用します。

シーケンスプログラムでファイルアクセス命令 (SP. FWRITE 命令など) を使用して、CSV 形式／バイナリ形式で ATA カードの PC ユーザデータにアクセスします。

(2) SRAM カード、ATA カードの使用前に

必ず GX Developer で SRAM カード、ATA カードのフォーマットを行ってください。

(a) フォーマットの実行

GX Developer の [オンライン] → [PC メモリフォーマット] で行います。

- SRAM カードをフォーマットする場合は、対象メモリに “メモリカード (RAM)” を選択して行います。
- ATA カードをフォーマットする場合は、対象メモリに “メモリカード (ROM)” を選択して行います。

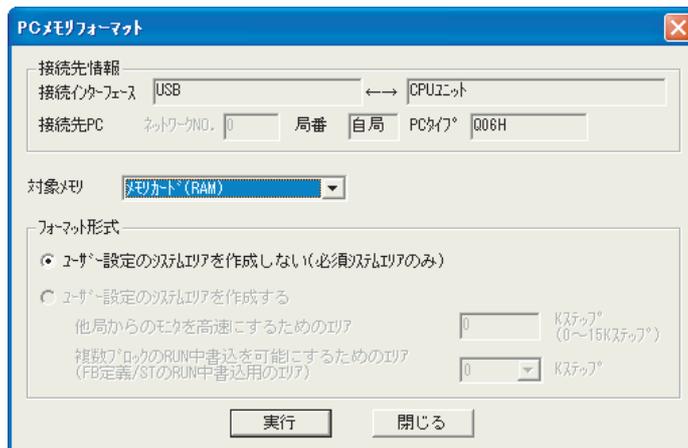


図 5.21 SRAM カード、ATA カードのフォーマット

Point

- ATA カードは、GX Developer 以外でフォーマットしないでください。
Microsoft® Windows® のフォーマット機能などによりフォーマットした場合は、CPU ユニットに装着して使用できなくなる場合があります。
- SRAM カードおよび ATA カードをフォーマットすると、メモリカード情報エリアが自動的に確保されるため、メモリカード情報エリア分容量が少なくなります。

備考

Flash カードは、フォーマットを行う必要はありません。

(b) フォーマット後のメモリ容量の確認

GX Developer の [オンライン] → [PC 読出] で確認できます。

- 1) PC 読出画面の対象メモリで、“メモリカード (RAM)” または “メモリカード (ROM)” を選択します。
- 2) **空き容量** ボタンをクリックします。
- 3) 全空容量欄にメモリ容量が表示されます。

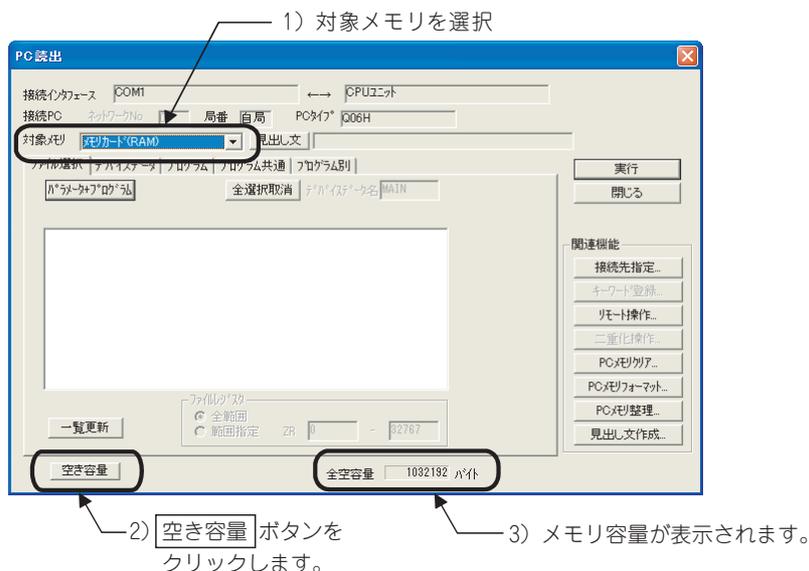


図 5.22 メモリ容量の確認手順

(3) メモリカードへの書込み

書込み前の操作と書込み方法の種類について説明します。

(a) SRAM カード、ATA カードへの書込み

GX Developer の [オンライン] → [PC 書込] で行います。

- SRAM カードの場合は、PC 書込画面の対象メモリで “メモリカード (RAM)” を選択します。
- ATA カードの場合は、PC 書込画面の対象メモリで “メモリカード (ROM)” を選択します。



図 5.23 PC 書込画面

(b) Flash カードへの書込み

Flash カードへの書込み方法には、下記の2種類があります。

- プログラムメモリのROM化による書込み (☞ 5.2.6 項 (1) (a))
- PC 書込 (フラッシュ ROM) による書込み (☞ 5.2.6 項 (1) (b))

Point

ファイルサイズには最小単位があります。(☞ 5.4.4 項)

占有するメモリ容量は、実際のファイルサイズよりも大きくなる場合があります。

ファイル数が多くなると、占有するメモリ容量と実際のファイルサイズとの差は大きくなりますので注意してください。

(4) メモリカードに格納したプログラムの使用方法

メモリカードに格納したプログラムの実行は、プログラムメモリにブートして行います。(☞ 5.2.7 項)

5.2.6 GX Developer による標準 ROM / Flash カードへの書込み

(1) 標準 ROM / Flash カードへの書込み方法と用途

標準 ROM / Flash カードにデータを書き込む方法には、図 5.24 のものがあります。

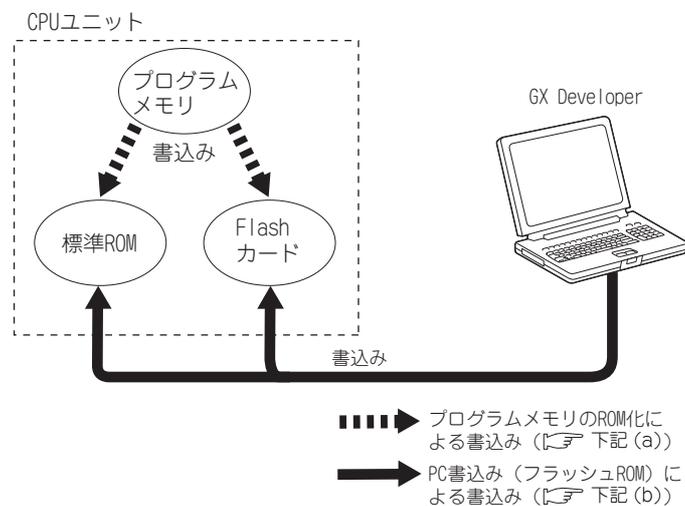


図 5.24 標準 ROM / Flash カードへの書込み方法の種類

(a) プログラムメモリの ROM 化による書込み

プログラムメモリの内容を、一括で Flash カードに書き込みます。

GX Developer の [オンライン] → [PC 書込 (フラッシュ ROM)] → [プログラムメモリの ROM 化] で行います。

プログラムメモリの ROM 化は、下記のような場合に使用します。

- プログラムメモリのプログラムでデバッグを行い、その状態のまま Flash カードにデータを書き込んでブート運転 (5.2.8 項) をする場合
- プログラムメモリのデータを、バッテリーバックアップなしで Flash カードに保存する場合

(b) PC 書込み (フラッシュ ROM) による書込み

GX Developer で指定したファイルを、一括で Flash カードに書き込みます。

GX Developer の [オンライン] → [PC 書込 (フラッシュ ROM)] → [PC 書込 (フラッシュ ROM)] で行います。

PC 書込み (フラッシュ ROM) は、下記のような場合に使用します。

- プログラムメモリに入りきれないパラメータ、デバイス初期値、デバイスコメントを、Flash カードに格納する場合
- ファイルレジスタを Flash カードに格納して使用する場合

(2) 標準 ROM / Flash カードへの書込み

書込み前の操作と書込み方法について説明します。

(a) 書込み前に

下記の点を確認してください。

1) 書き込むファイルの準備

標準 ROM / Flash カードにファイルを書き込むと、標準 ROM / Flash カードに格納されている全ファイルは自動的に削除されます。

格納するすべてのファイルをあらかじめ用意してから書込みを行ってください。

2) ブート運転をする場合

ブート運転を行う際に、パラメータを標準 ROM / Flash カードに格納する場合は、5.2.8 項に示すブートファイル設定を行ってください。

(b) 書込み手順

標準 ROM / Flash カードにファイルを書き込む手順について説明します。

1) GX Developer の [プログラムメモリの ROM 化] の手順

- GX Developer の [オンライン] → [PC 書込 (フラッシュ ROM)] → [プログラムメモリの ROM 化] を選択します。
- プログラムメモリの ROM 化画面が表示されます。

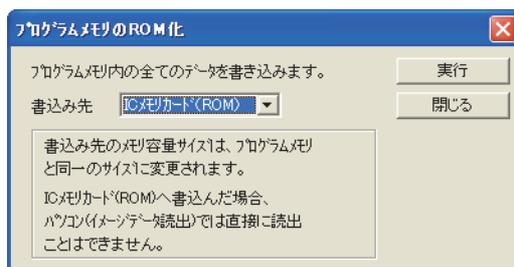


図 5.25 プログラムメモリの ROM 化画面

- 書き込み先を選択し、プログラムメモリのファイルを標準 ROM / Flash カードにコピーします。

Point

- プログラムメモリの ROM 化で書き込んだ場合、書き込み先の使用メモリ容量は、プログラムメモリで使用していた容量になります。
書き込み先のメモリ容量をフルに使用したい場合は、PC 書込 (フラッシュ ROM) で書き込んでください。
- Flash カードに、プログラムメモリに格納できないデータ (ファイルレジスタ) を書き込む場合は、PC 書込 (フラッシュ ROM) で書き込んでください。

2) GX Developer の [PC 書込 (フラッシュ ROM)] の手順

- GX Developer の [オンライン] → [PC 書込 (フラッシュ ROM)] → [PC 書込 (フラッシュ ROM)] を選択します。
- PC 書込 (フラッシュ ROM) 画面が表示されます。

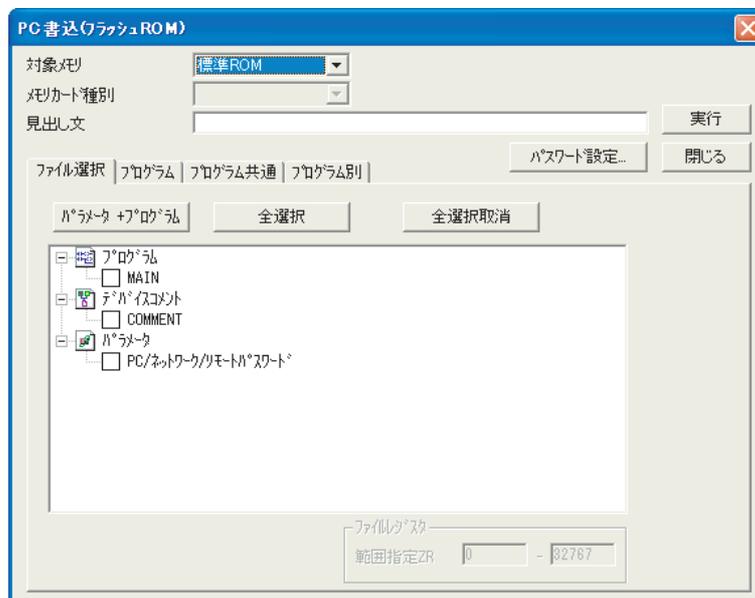


図 5.26 PC 書込 (フラッシュ ROM)

- 対象メモリを選択します。
- 書き込むファイルを選択し、標準 ROM / Flash カードにファイルを書き込みます。

(3) 標準 ROM / Flash カード内のファイルに対する追加・変更

追加・変更は、下記の方法で行ってください。(格納されているファイルに直接追加・変更はできません。)

(a) GX Developer の [プログラムメモリの ROM 化] を使用して書き込んだ場合

- GX Developer の [オンライン] → [PC 読出] で、プログラムメモリの全ファイルを読み出します。
- 読み出したファイルに対して、追加・変更を行います。
- 追加・変更を行ったファイルを、プログラムメモリに書き込みます。
- [オンライン] → [PC 書込 (フラッシュ ROM)] → [プログラムメモリの ROM 化] で、標準 ROM / Flash カードにファイルを書き込みます。

(b) GX Developer の [PC 書込 (フラッシュ ROM)] を使用して書き込んだ場合

- GX Developer の [オンライン] → [PC 読出] で、標準 ROM / Flash カードの全ファイルを読み出します。
- 読み出したファイルに対して、追加・変更を行います。
- 追加・変更を行ったファイルを、[オンライン] → [PC 書込 (フラッシュ ROM)] → [PC 書込 (フラッシュ ROM)] で、標準 ROM / Flash カードに書き込みます。

(4) 注意事項

(a) GX Developer の通信タイムチェック時間の設定について

標準 ROM / Flash カードにファイルを書き込む場合、処理に時間がかかるため、GX Developer の通信タイムチェック時間は 60 秒以上に設定してください。

通信タイムチェック時間が短いと、GX Developer がタイムアウトになることがあります。



図 5.27 通信タイムチェック時間設定

(b) CC-Link 経由の他局の GX Developer から書き込みを行う場合

標準 ROM / Flash カードにファイルを書き込むときは、処理に時間がかかるため、CC-Link の CPU 監視時間設定 (SW000A) を、60 秒以上に設定してください。

(デフォルト値は 90 秒のため、デフォルト値で使用できます。)

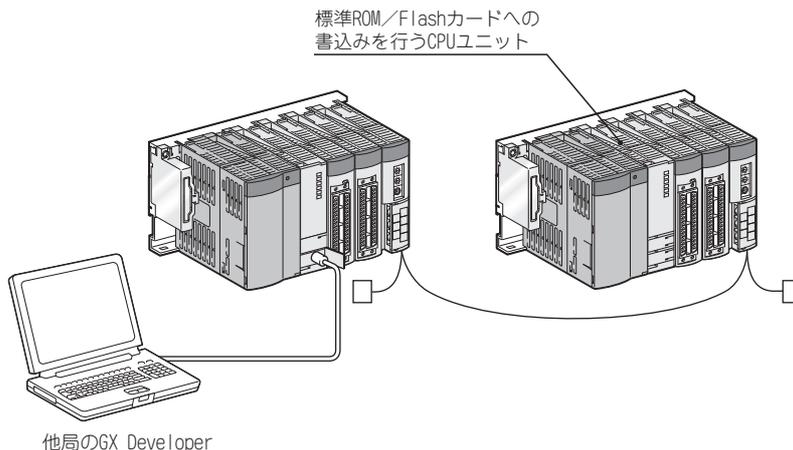


図 5.28 他局の GX Developer からの書き込み (CC-Link 経由)

備考

通信タイムチェック時間の設定および CPU 監視時間の設定については、下記マニュアルを参照してください。

☞ GX Developer Version8 オペレーティングマニュアル

(c) PC 書込 (フラッシュ ROM) に要する時間

PC 書込 (フラッシュ ROM) による書込みでは、標準 ROM / Flash カードの全容量分を書き込みます。このため、Flash カードにステップ数の小さいプログラムの書込みを行っても、Flash カードの全容量の書込みを行うため、完了までに時間がかかります。

(Q2MEM-4MBF に RS-232 インタフェースを使用し、通信速度が 115.2kbps の場合、約 14 分かかります。)

Flash カードへの書込みを行う場合は、伝送速度を速くするか、USB を使用してください。また他局から PC 書込 (フラッシュ ROM) を行うと、通信時間がかかります。

(d) RUN 中での書込み

下記の場合、RUN 中に PC 書込 (フラッシュ ROM) を行うとエラーになり、CPU ユニットが停止することがあります。STOP 状態にしてから PC 書込 (フラッシュ ROM) を実行してください。

- 1) シーケンスプログラムで、Flash カードのファイルレジスタを使用している。
- 2) PC パラメータでファイルレジスタを “使用しない” に設定し、シーケンスプログラムでファイルレジスタを使用している。

(e) PC 書込 (フラッシュ ROM) 中の書込み / 読出し

PC 書込 (フラッシュ ROM) を実行中は、他のユニットからの書込み / 読出しができません。このため、他のユニットでタイムアウトになることがあります。

(f) STOP 状態で PC 書込 (フラッシュ ROM) を行っている場合

書込み中は RUN 状態にしないでください。

PC 書込 (フラッシュ ROM) 中は正常に RUN できません。

完了後に RUN にしてください。

5.2.7 メモリカード→標準 ROM 全データ自動書込み 注 5.2

(1) メモリカード→標準 ROM 全データ自動書込み（以下、標準 ROM への自動書込みと略します）とは

あらかじめメモリカードに書き込んでおいたパラメータやプログラムを、標準 ROM に自動的に書き込む機能です。

標準 ROM への自動書込みでは、図 5.29 のようにプログラム、パラメータなどをメモリカードからプログラムメモリにブートし、ブートされたプログラム、パラメータなどをプログラムメモリから標準 ROM に書き込みます。

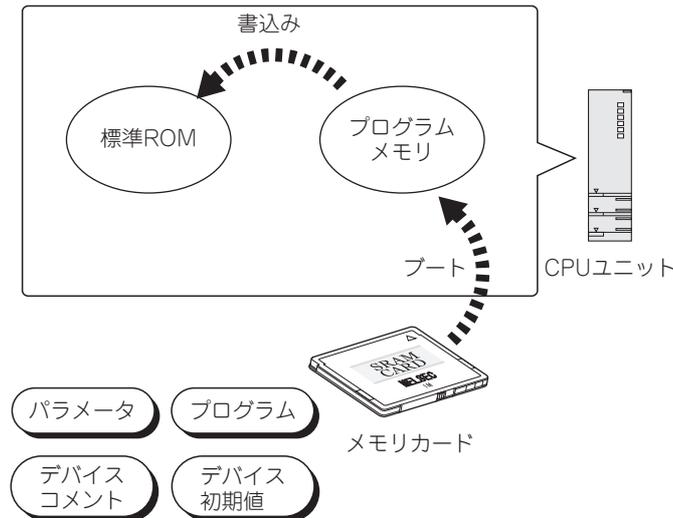


図 5.29 標準 ROM への自動書込み

(2) 標準 ROM への自動書込みの用途

標準 ROM への自動書込みを使用すると、メモリカードにあらかじめ書き込んだプログラム、パラメータなどを、GX Developer（パソコン）を使用することなく、標準 ROM への書込みができます。このため、下記のような場合に使用すると便利です。

- 同じパラメータやプログラムを、複数台の CPU ユニットに書き込みたい場合
- 遠くの現場で同じ環境を実現したい場合



注 5.2

High performance

ハイパフォーマンスモデル QCPU で標準 ROM への自動書込みをする場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。（付 2.2）

設定したメモリカードを標準 ROM への自動書込みに対応していないハイパフォーマンスモデル QCPU に装着した場合は、標準 ROM からのブート運転になります。

(3) 標準 ROM への自動書込みの実行手順

標準 ROM への自動書込みは、下記手順で行います。

(a) GX Developer での設定

- 1) PC パラメータのブートファイル設定で “プログラムメモリをクリアする” と “メモリカード→標準 ROM 全データ自動書込” をチェックします。
ブートファイル設定でブートするプログラム、パラメータなどを設定します。
(転送元は “標準 ROM” に設定します。)

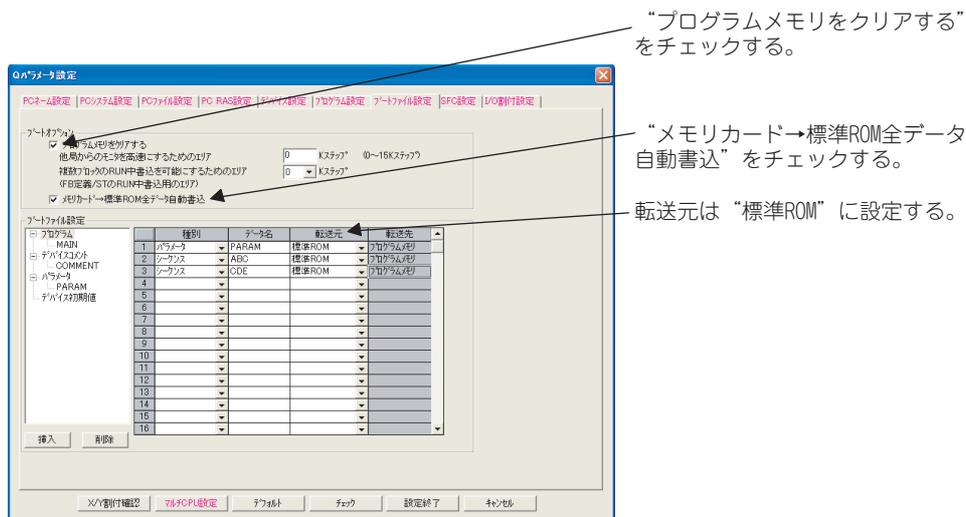


図 5.30 メモリカード→標準 ROM 全データ自動書込み設定

- 2) 設定したパラメータとブートするプログラムをメモリカードに格納します。

(b) CPU ユニットでの操作

- 1) CPU ユニットの電源を OFF にします。
- 2) CPU ユニットにパラメータとブートするプログラムを格納したメモリカードを装着します。
- 3) CPU ユニットのディップスイッチのパラメータ有効ドライブを装着したメモリカードに設定します。
 - SRAM カード装着時 …… SW2 : ON, SW3 : OFF
 - Flash / ATA カード装着時 …… SW2 : OFF, SW3 : ON

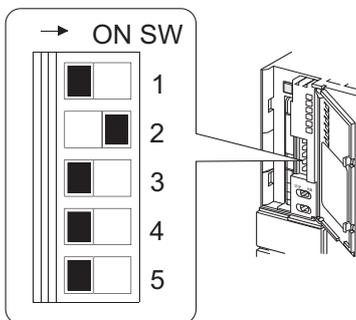


図 5.31 SRAM カード装着時のディップスイッチ

- 4) CPUユニットの電源を ON します。
メモリカードで指定されたファイルをプログラムメモリにブートします。
ブート完了後、プログラムメモリの内容を標準 ROM に書き込みます。
- 5) 標準 ROM への自動書き込みが完了すると、BOOT LED と ERR.LED が点滅し、CPU ユニットは停止エラー状態となります。
- 6) CPU ユニットの電源を OFF します。
- 7) メモリカードを抜き取り、CPU ユニットのディップスイッチのパラメータ有効ドライブを標準 ROM に設定します。
 - 標準 ROM …… SW2 : ON, SW3 : ON

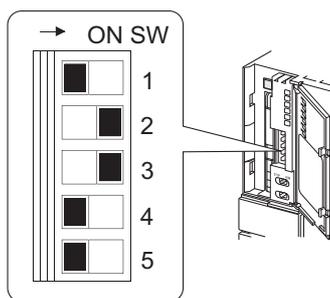


図 5.32 ディップスイッチのパラメータ有効ドライブを標準 ROM に設定した場合

- 8) CPU ユニットの電源を ON すると、標準 ROM からプログラムメモリへのブートを行い、実際の運転ができます。

(4) 注意事項

標準 ROM への自動書き込みを行う場合の注意事項を示します。

(a) 同一ファイル名のファイルがプログラムメモリに存在する場合

メモリカードからブートするファイルと同一ファイル名がプログラムメモリに存在する場合は、メモリカードのデータで上書きします。

また、メモリカードからブートするファイルと同一ファイル名がプログラムメモリに存在しない場合は、プログラムメモリに追加します。

このとき、プログラムメモリの容量をオーバーすると“FILE SET ERROR”（エラーコード：2401）となります。

(b) ブート時のプログラムメモリのクリアについて

メモリカードからプログラムメモリへのブート時、プログラムメモリのクリアを行ってからブートするか、プログラムメモリをクリアせずにブートするかの選択ができます。

標準 ROM への自動書き込みを行う場合、プログラムメモリのクリアを行ってからブートするように設定すると、ブート時プログラムメモリの容量オーバーになることを防止できます。

(c) 本機能使用時のディップスイッチのパラメータ有効ドライブ設定について

“メモリカード→標準 ROM 全データ自動書込”設定は、ディップスイッチのパラメータ有効ドライブ設定がメモリカードのときのみ有効になります。

このため、標準 ROM への自動書き込みが完了し、実際の運転に入るときにブートファイル設定の“メモリカード→標準 ROM 全データ自動書込”のチェックをはずす必要はありません。

5.2.8 標準 ROM / メモリカードのプログラムの実行（ブート運転）

本項では、標準 ROM / メモリカードに格納したプログラムの演算を行うための方法について説明します。

(1) 標準 ROM / メモリカードのプログラムを実行する方法

標準 ROM / メモリカードに格納したプログラムの演算は、プログラムメモリにブートして行います。

ブートするには PC パラメータのブートファイル設定を設定してください。（ 本項 (3), (4)）

ブートファイル設定で指定したプログラムが、CPU ユニットの電源 OFF → ON またはリセット解除時に、標準 ROM / メモリカードからプログラムメモリにブートされます。

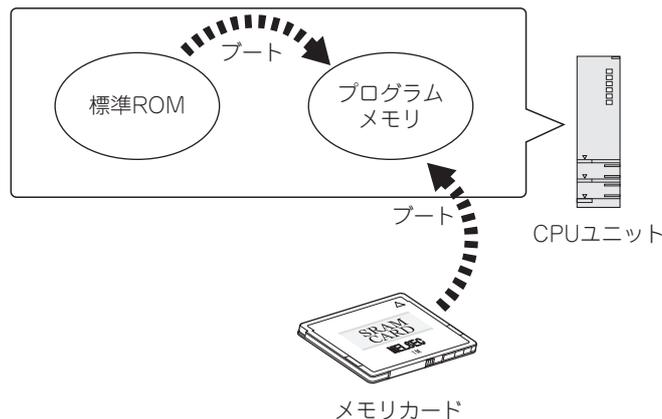


図 5.33 ブート運転

(2) ブート可能なファイル，転送元および転送先

ブート可能なファイル，転送元および転送先の組合せを表 5.7 に示します。

表 5.7 ブート可能なファイル，転送元および転送先の組合せ

ファイル種別	転送元		転送先
	標準 ROM	メモリカード	
パラメータ* 1	○	○	プログラムメモリ
シーケンスプログラム	○	○	
デバイスコメント	○	○	
デバイス初期値	○	○	

○：ブート可，×：ブート不可

* 1：インテリジェント機能ユニットパラメータも含まれます。

(3) ブート運転を行うまでの手順

作成したファイルを標準 ROM / メモリカードに格納し、ブート運転を行うまでの手順を下記に示します。

(a) プログラムの作成

プログラムを作成します。

(b) ブートファイル設定

PCパラメータのブートファイル設定で、プログラムメモリにブートするファイル（ファイル種別、ファイル名、転送元）を設定します。



図 5.34 ブートファイル設定

(c) ブート運転をするためのハードウェア設定

ディップスイッチのパラメータ有効ドライブを、パラメータを格納するメモリに設定します。ブート運転を行うときは、標準 ROM またはメモリカードをパラメータの格納先にしてください。
 (☞ 本項 (6)(a))

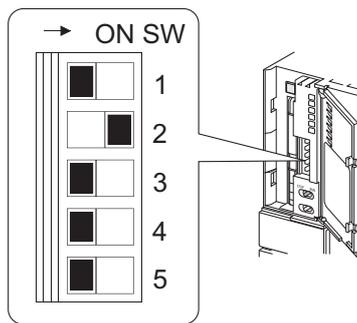


図 5.35 ディップスイッチのパラメータ有効ドライブを SRAM カードに設定した場合

5.2 ハイトパフォーマンスモデル QCPU、プロセス CPU、二重化 CPU で取り扱うメモリ
 5.2.8 標準 ROM / メモリカードのプログラムの実行 (ブート運転)

(d) メモリカードの装着（ブート元がメモリカードの場合）

CPU ユニットにメモリカードを装着します。

(e) 標準 ROM またはメモリカードの書込み

ディップスイッチのパラメータ有効ドライブに設定したメモリにパラメータを書き込みます。

また、(b) のブートファイル設定で設定したファイルを GX Developer で転送元メモリに書き込みます。

(f) プログラムの実行

RESET/L.CLR スイッチでリセットを行います。

指定メモリからのブートが完了すると、BOOT LED が点灯します。

(g) ブート完了の確認

ブートが正常に完了しているかは、下記により確認できます。

- BOOT LED が点灯する
- 特殊リレー（SM660）が ON する
- GX Developer の [オンライン] → [PC 照合] で、転送元のメモリに書き込んだデータと、プログラムメモリ内のデータの照合を行う

(4) ブート運転を中止する場合の操作

ブート運転を中止し、プログラムメモリに書き込まれたプログラム、パラメータで運転を実行するときは、GX Developer で下記の操作を行ってください。

- 1) ブートファイルを設定していないパラメータをプログラムメモリに書き込む。
- 2) CPU ユニットのディップスイッチのパラメータ有効ドライブを “プログラムメモリ” に設定する。
(SW2 : OFF, SW3 : OFF)

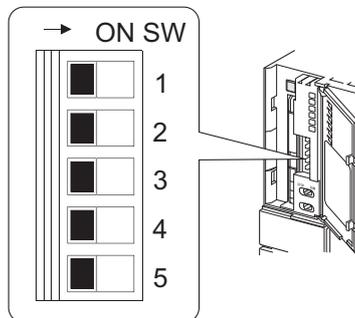


図 5.36 ディップスイッチのパラメータ有効ドライブをプログラムメモリに設定した場合

- 3) 電源の再投入、またはリセットする。

(5) RUN 中でのプログラムファイルの変更 注 5.3

(a) 変更方法

下記命令で行います。

- PLOADP 命令 (メモリカードからプログラムメモリへのプログラム転送)
- PUNLOADP 命令 (プログラムメモリからのプログラム削除)
- PSWAPP 命令 (プログラムメモリからのプログラム削除とメモリカードからプログラムメモリへのプログラム転送)

各命令の詳細については、下記マニュアルを参照してください。

 MELSEC-Q/L プログラミングマニュアル (共通命令編)

(b) プログラムファイルを変更した場合のプログラム設定について

RUN 中にプログラムファイルを変更しても、PC パラメータのプログラム設定は変更されません。

このため、CPU ユニートを STOP 状態にした場合は、PC パラメータのプログラム設定を RUN 中に変更した内容に修正 (プログラム名の追加, 変更, 削除) してください。

修正しない場合は、STOP 状態から RUN 状態に切り換えたときにエラーになります。

(6) ブート運転時の注意事項

(a) パラメータの格納先について

ブートファイル設定を行ったパラメータ (PC パラメータ) は、標準 ROM / メモリカードに格納してください。

プログラムメモリにパラメータを格納した場合、ディップスイッチのパラメータ有効ドライブをプログラムメモリに設定しても、CPU ユニートはプログラムメモリの PC パラメータのブートファイル設定を無視しません。( 5.2.10 項)



注 5.3

Redundant

二重化 CPU では、RUN 中でのプログラムファイルの変更はできません。

(b) ブート運転時の RUN 中書込みについて

1) SRAM カード, ATA カード

プログラムメモリのプログラムに RUN 中書込み (☞ 6.12 節) を行うと、ブート元の SRAM カード、ATA カードのプログラムに変更内容を反映できます。

2) 標準 ROM, Flash カード

プログラムメモリのプログラムに RUN 中書込みを行っても、ブート元の標準 ROM、Flash カードのプログラムに変更内容は反映されません。

このため、プログラムメモリに書き込んだプログラムと同じファイルを、PC 書込 (フラッシュ ROM) で標準 ROM、Flash カードに書き込んでください。(☞ 5.2.5 項、5.2.6 項)

(c) 設定可能な最大ブートファイル数について

PC パラメータのブートファイル設定で設定可能な最大ブートファイル数は、プログラムメモリに格納できるファイル数と同一にしてください。

ただし、下記設定を行った場合は、ブートファイルがそれぞれ 1 ファイル減少します。

- 見出し文を設定したとき
- 標準 ROM / メモリカードに格納されている PC パラメータ (ブートファイル設定を行ったもの) をブートしたとき

(d) ATA カード使用時のブート運転について

下記の状態でブート運転を行うと、ブート時に 1k ステップ (4k バイト) あたり最大 200ms かかることがあります。

- ATA カードからブートする場合
- ATA カードを装着した状態で、標準 ROM からブートする場合

(e) CPU ユニットの電源 OFF → ON またはリセット解除時にプログラムメモリの内容が変わる場合

プログラムメモリにシーケンスプログラムを書き込んで CPU ユニットの電源 OFF → ON またはリセット解除を行ったとき、プログラムメモリの内容が書き変わる場合にはブート運転になっていることが考えられます。

CPU ユニット前面の BOOT LED が点灯している場合はブート運転になっています。本項 (4) を参照し、ブート運転を中止してください。

(f) メモリカードからブートした場合の容量について

各メモリに格納されるファイルサイズ単位は、メモリカードとプログラムメモリとで異なります。

(☞ 5.4.4 項)

このため、メモリカードからプログラムメモリに転送されたファイルは、転送前と転送後でメモリ容量が変わりますので注意してください。

(g) メモリカードに書き込んだプログラムについて

メモリカードに書き込んだプログラム (ブートファイル設定で設定したプログラム) の PC タイプ (CPU ユニットの形名) とブートする CPU ユニットの形名は同一にしてください。

5.2.9 書き込んだファイルの詳細

CPUユニットに書き込まれた各ファイルには、GX Developerで作成時に設定したファイル名、ファイルサイズ、ファイルの書き込み日時などが付加されます。

GX Developerの[オンライン] → [PC読出]でファイルのモニタを行うと、各ファイルは次のように表示されます。

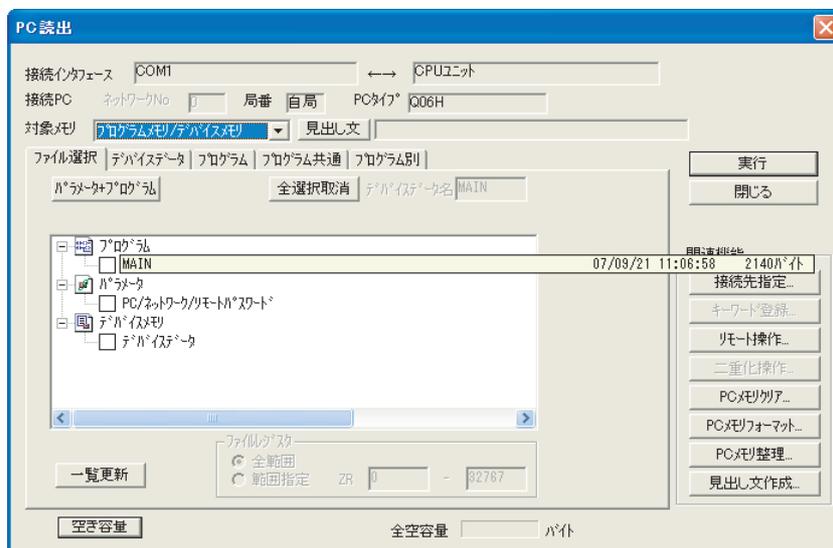


図 5.37 ファイル詳細の表示

(a) ファイル名

1) ファイル名の構成とファイルの指定

各ファイルは、ファイル名（最大半角8文字／全角4文字）と拡張子（半角3文字）で構成されています。

ファイル名は、すべて大文字で作成します。

拡張子は、ファイル作成時に設定した種別により自動的に付加されます。

2) ファイル名に使用できない文字

下記に示す Microsoft® Windows® の予約語は、ファイル名として使用できません。

COM1 ~ COM9, PRN, LPT1 ~ LPT9, NUL, AUX, CLOCK\$, CON

3) シーケンスプログラムでのファイル名の指定方法

半角小文字、半角大文字は区別するため、大文字で指定してください。

※ “ABC” と “abc” は、区別して取り扱います。

全角小文字、全角大文字は区別するため、大文字で指定してください。

※ “ABC” と “abc” は、区別して取り扱います。

(b) 日付, 時間

ファイルが CPU ユニットに書き込まれた日付と時間です。
GX Developer (パソコン) 側の時計に従って付加されます。

(c) サイズ

GX Developer から CPU ユニットに書き込みを行ったときのファイル容量をバイト単位で表示します。(最新データは, [オンライン] → [PC 読出] の **一覧更新** ボタンをクリックすると表示されます。)
ファイルレジスタを除くファイルには, ユーザで作成したファイル容量に最低 64 バイト (プログラムの場合は 136 バイト) が付加されます。(☞ 5.4.3 項)

5.2.10 有効にするパラメータの指定 (パラメータ有効ドライブ設定)

有効にするパラメータの格納ドライブ (メモリ) を CPU ユニット本体のディップスイッチで指定します。

(1) パラメータ有効ドライブの指定方法

ディップスイッチ SW2, SW3 の設定でパラメータ有効ドライブを指定します。

表 5.8 SW2, SW3 によるドライブ指定

SW2	SW3	パラメータの有効ドライブ
OFF	OFF	ドライブ 0 (プログラムメモリ)
ON	OFF	ドライブ 1 (メモリカード RAM)
OFF	ON	ドライブ 2 (メモリカード ROM)
ON	ON	ドライブ 4 (標準 ROM)

(2) 有効にするパラメータの決定タイミング

有効にするパラメータの決定タイミングを下記に示します。

- CPU ユニットの電源 OFF → ON 時
- リセット解除時

CPU ユニットは, 上記の決定タイミングに, SW2, SW3 で指定されたドライブのパラメータを有効にし, 有効になったパラメータ設定で動作します。

5.3 プログラムファイルの構成

プログラムファイルは、ファイルヘッダ、実行プログラム、RUN 中書込用確保ステップから構成されています。

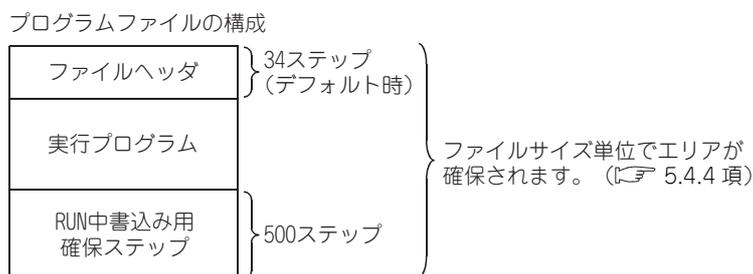


図 5.38 プログラムファイルの構成

(1) 各構成の詳細

CPU ユニットのプログラムメモリに格納した場合のプログラム容量は、上記 3 種類のエリアの合計になります。

(a) ファイルヘッダ

ファイル名、ファイルサイズ、ファイル作成日などが格納されているエリアです。

ファイルヘッダサイズは、PC パラメータのデバイス設定を変更することにより 25 ~ 35 ステップ (100 ~ 140 バイト) となります。(デフォルト時は 34 ステップ)

(b) 実行プログラム

作成したプログラムが格納されるエリアです。

(c) RUN 中書込用確保ステップ

GX Developer でステップ数が増加する RUN 中書込を行ったときに使用するエリアです。

RUN 中書込み完了後、RUN 中書込用確保ステップの残りステップ数が表示されます。

1) RUN 中書込用確保ステップ数のデフォルト

500 ステップ (2000 バイト) です。

2) RUN 中書込用確保ステップ数の変更

GX Developer の [オンライン] → [PC 書込] の ≪プログラム≫ タブから変更できます。

RUN 中書込み時にも設定できます。(☞ 6.12.1 項)

(2) GX Developer でのプログラム容量の表示

GX Developer のプログラミング時、図 5.39 のようにプログラム容量（ファイルヘッダ容量と作成したプログラムのステップ数の合計）がステップ数で表示されます。

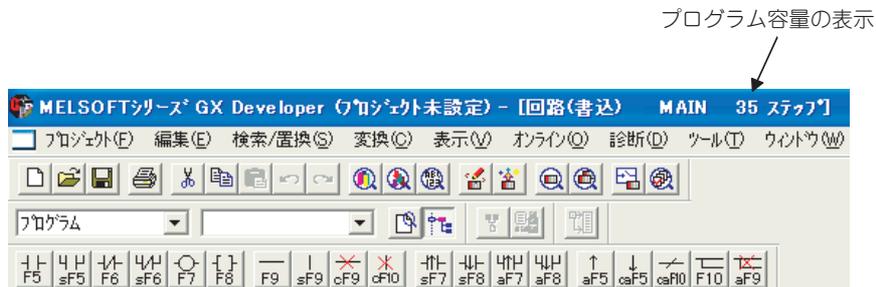
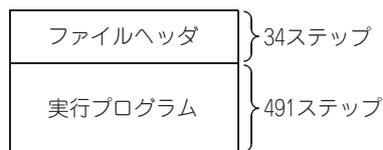


図 5.39 プログラム容量の表示

Point

- GX Developer でのプログラミング時に表示されるプログラム容量は、ファイルヘッダ+実行プログラムの容量であり、RUN 中書込用確保ステップの容量（500 ステップ）は含まれていません。

例 実行プログラム部分が 491 ステップのプログラムは、GX Developer 上での容量は以下の図のように表示されます。（ファイルヘッダは、34 ステップ固定）



GX Developer 上の表示：
34ステップ+491ステップ=525ステップとなります。

図 5.40 GX Developer 上でのファイルの状態

- プログラムメモリ上では、ファイルはファイルサイズ単位で格納されるため、GX Developer でのプログラミング時に表示されるプログラム容量と CPU ユニット上でのプログラムファイルの容量が異なる場合があります。（[図 5.4.3 項](#)）

5.4 GX Developer によるファイル操作および取扱い時の注意事項

5.4.1 ファイルの操作

プログラムメモリ、標準ROM、メモリカードに格納されているファイルは、GX Developer のオンライン操作により表 5.9 に示すファイル操作が可能です。

ただし GX Developer によるパスワード登録、CPU ユニットのシステムプロテクトスイッチの状態、CPU ユニットの RUN/STOP 状態により実行できるファイル操作が異なります。

表 5.9 GX Developer から実行できるファイル操作一覧

ファイルの操作	操作内容	操作可否* 1															
		ベーシックモデル QCPU			ハイパフォーマンスモデル QCPU				プロセス CPU				二重化 CPU				
		A	B	C	A	B	C	D	A	B	C	D	A	B	C	D	
PC 読出	対象メモリからファイルを読み出す。	○	△	○	○	△	○	○	○	△	○	○	○	△	○	○	
PC 書込	プログラムメモリ、SRAM カードにファイルを書き込む。	△	△	×	△	△	○	×	△	△	○	×	△	△	○	×	
	標準ROM にファイルを書き込む。	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	
PC 照合	対象メモリと GX Developer のファイルを照合する。	△	△	○	△	△	○	○	△	△	○	○	△	△	○	○	
プログラムメモリのROM化	プログラムメモリに格納されているファイルを一括でFlashカードに書き込む。	○	○	×	○	○	○	×	○	○	○	×	○	○	○	×	
	プログラムメモリに格納されているファイルを一括で標準ROMに書き込む。	○	○	×	○	○	○	×	○	○	○	×	○	○	○	×	
PC 書込 (フラッシュROM)	GX Developer から指定ファイルを一括でFlashカードに書き込む。	×	×	×	△	△	○	×	△	△	○	×	△	△	○	×	
	GX Developer から指定ファイルを一括で標準ROMに書き込む。	×	×	×	△	△	○	×	△	△	○	×	△	△	○	×	
PC データ削除	メモリ上に格納されているファイルを削除する。	△	△	×	△	△	×	×	△	△	×	×	△	△	×	×	
PC メモリフォーマット	メモリのフォーマットを行う。	○	○	×	○	○	×	×	○	○	×	×	○	○	×	×	
PC メモリ整理	メモリ上の配置が不連続になったファイルを再配置する。	○	○	×	○	○	×	×	○	○	×	×	○	○	×	×	
回路モードでのRUN中書込み	回路モードで変更した内容をプログラムメモリに書き込む。	△	△	○	△	△	○	×	△	△	○	×	△	△	○	×	

○：実行可、△：パスワードが一致のとき実行可、×：実行不可

* 1：操作可否の記号の内容を下記に示します。

- A：ファイルに書込禁止のパスワード設定時
- B：ファイルに読出／書込禁止のパスワード設定時
- C：CPUユニットがRUN状態のとき
- D：CPUユニットのシステムプロテクトスイッチがONのとき

5.4.2 ファイルの取扱い時の注意事項

(1) ファイル操作時の電源 OFF（リセットを含む）について

ファイル操作時に CPU ユニットの電源 OFF，または CPU ユニットをリセットした場合，各メモリのファイルは表 5.10 になります。

表 5.10 ファイル操作時に電源 OFF，または CPU ユニットをリセットした場合のファイルの状態

CPU ユニット	各メモリの状態
ベーシックモデル QCPU	各メモリのファイルは不定になります。
ハイパフォーマンスモデル QCPU プロセス CPU 二重化 CPU	各メモリのファイルは破壊されません。 (メモリカード使用時は，電源 OFF 中に CPU ユニットからメモリカードをはずさずに，そのままの状態電源 ON した場合を示します。)

Point

ファイル移動が発生する操作中にシーケンサの電源を OFF した場合，作業途中のデータを CPU ユニットの内部メモリに保持します。
保持しているデータは，電源 OFF → ON 時に復帰します。
内部メモリのデータを保持するためには，バッテリーによるバックアップが必要です。

(2) 複数の GX Developer から同一ファイルへの同時書込みについて

書込み中のファイルに対して他の GX Developer からアクセスはできません。

また，アクセス中のファイルに対して他の GX Developer から書込みできません。

複数の GX Developer から同一ファイルに書込みする場合は，1 つの GX Developer の処理が完了してから，次の GX Developer の処理を行うようにしてください。

(3) 複数の GX Developer から異なるファイルへの同時アクセスについて

同一 CPU ユニットの異なるファイルに対して他の GX Developer から同時にアクセスできるのは，最大 10 箇所です。

5.4.3 ファイルのメモリ容量

CPUユニットで使用するファイルは、種類によりファイルサイズが異なります。

なお、ファイルをメモリエリアに書き込む場合、書き込むCPUユニットとメモリエリアにより、格納する容量の単位が異なります。(☞ 5.4.4 項)

(1) ベーシックモデル QCPU 使用時

プログラムメモリ、標準RAM、標準ROMを使用する場合は、各ファイルのサイズの概算を、表 5.11 により算出してください。

表 5.11 ファイルのメモリ容量の算出 (ベーシックモデル QCPU)

機能	概略ファイル容量 (単位: バイト)																							
ドライブ見出し文	64																							
パラメータ	デフォルト: 522 (パラメータの設定により増加する。) 参考 ブート設定 → 94 ・ CC-Link IE コントローラネットワーク設定あり → 最大 7214 増加 ・ MELSECNET/H 設定あり → 最大 6180 増加 ・ Ethernet 設定あり → 最大 922 増加 ・ CC-Link 設定あり → 最大下表の容量増加 (下表の値は / ユニットあたりの増加量を示しています。) <table border="1" style="margin-left: 20px;"> <thead> <tr> <th rowspan="2">CC-Link 設定</th> <th colspan="3">モード設定</th> </tr> <tr> <th>Ver.1 モード</th> <th>Ver.2 モード</th> <th>Ver.2 追加モード</th> </tr> </thead> <tbody> <tr> <td>1 枚目</td> <td>550 バイト</td> <td>572 バイト</td> <td>624 バイト</td> </tr> <tr> <td>2 枚目 ~ 4 枚目</td> <td>536 バイト</td> <td>558 バイト</td> <td>610 バイト</td> </tr> <tr> <td>5 枚目</td> <td>550 バイト</td> <td>566 バイト</td> <td>618 バイト</td> </tr> <tr> <td>6 枚目 ~ 8 枚目</td> <td>536 バイト</td> <td>558 バイト</td> <td>610 バイト</td> </tr> </tbody> </table> ・ リモートパスワード設定あり → $64 + 20 + (\text{対象ユニット数} \times 10)$, 最大 164 増加	CC-Link 設定	モード設定			Ver.1 モード	Ver.2 モード	Ver.2 追加モード	1 枚目	550 バイト	572 バイト	624 バイト	2 枚目 ~ 4 枚目	536 バイト	558 バイト	610 バイト	5 枚目	550 バイト	566 バイト	618 バイト	6 枚目 ~ 8 枚目	536 バイト	558 バイト	610 バイト
CC-Link 設定	モード設定																							
	Ver.1 モード	Ver.2 モード	Ver.2 追加モード																					
1 枚目	550 バイト	572 バイト	624 バイト																					
2 枚目 ~ 4 枚目	536 バイト	558 バイト	610 バイト																					
5 枚目	550 バイト	566 バイト	618 バイト																					
6 枚目 ~ 8 枚目	536 バイト	558 バイト	610 バイト																					
シーケンスプログラム	$136 * 1 + (4 \times ((\text{ステップ数}) + (\text{RUN 中書込用確保ステップ数})))$																							
デバイスコメント	$74 + (\text{各デバイスのコメントデータサイズ合計})$ ・ $1 \text{ デバイスのコメントデータサイズ} = 10 + 10250 \times a + 40 \times b$ ・ a: $((\text{デバイス点数}) / 256)$ の商 ・ b: $((\text{デバイス点数}) / 256)$ の余り																							
ファイルレジスタ	$2 \times (\text{ファイルレジスタ点数})$																							
デバイス初期値	$66 + 44 \times n + 2 \times (\text{デバイス初期値で設定しているデバイス点数の合計})$ ・ n: デバイス初期値の設定数																							
インテリパラメータ	$68 + (24 \times \text{設定台数}) + \text{各ユーティリティごとのパラメータサイズ}$																							
ユーザ設定エリア	フォーマット時の設定値 (0 ~ 3k)																							
複数ブロック RUN 中書込み設定	フォーマット時の設定値 (0/1.25k/2.5k)																							

* 1: 136 はデフォルト値です。(パラメータの設定により増減します。)

備考

メモリ容量の計算例については、5.4.4 項を参照してください。

(2) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU 使用時

プログラムメモリ, 標準 RAM, 標準 ROM, メモリカードを使用する場合は, 各ファイルのサイズの概算を, 表 5.12 により算出してください。

表 5.12 ファイルのメモリ容量の算出 (ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU)

機能	概略ファイル容量 (単位: バイト)																															
ドライブ見出し文	64																															
パラメータ	<p>デフォルト (パラメータの設定により増加する。)</p> <table border="1"> <thead> <tr> <th>CPU ユニット</th> <th>ファイル容量</th> </tr> </thead> <tbody> <tr> <td>ハイパフォーマンスモデル QCPU</td> <td>564</td> </tr> <tr> <td>プロセス CPU</td> <td>558</td> </tr> <tr> <td>二重化 CPU</td> <td>764</td> </tr> </tbody> </table> <p>参 考</p> <p>ブート設定 → 70 + (18 × (ファイル数))</p> <ul style="list-style-type: none"> • CC-Link IE コントローラネットワーク設定あり → 最大 7214 / ユニット増加 • MELSECNET/H 設定あり → 最大 6180 / ユニット増加 • Ethernet 設定あり → 最大 922 / ユニット増加 • CC-Link 設定あり → 最大下表の容量増加 (下表の値は / ユニットあたりの増加量を示しています。) <table border="1"> <thead> <tr> <th rowspan="2">CC-Link 設定</th> <th colspan="3">モード設定</th> </tr> <tr> <th>Ver.1 モード</th> <th>Ver.2 モード</th> <th>Ver.2 追加モード</th> </tr> </thead> <tbody> <tr> <td>1 枚目</td> <td>550 バイト</td> <td>572 バイト</td> <td>624 バイト</td> </tr> <tr> <td>2 枚目 ~ 4 枚目</td> <td>536 バイト</td> <td>558 バイト</td> <td>610 バイト</td> </tr> <tr> <td>5 枚目</td> <td>550 バイト</td> <td>566 バイト</td> <td>618 バイト</td> </tr> <tr> <td>6 枚目 ~ 8 枚目</td> <td>536 バイト</td> <td>558 バイト</td> <td>610 バイト</td> </tr> </tbody> </table> <p>• リモートパスワード設定あり → 64 + 20 + (対象ユニット数 × 10), 最大 164 増加</p>	CPU ユニット	ファイル容量	ハイパフォーマンスモデル QCPU	564	プロセス CPU	558	二重化 CPU	764	CC-Link 設定	モード設定			Ver.1 モード	Ver.2 モード	Ver.2 追加モード	1 枚目	550 バイト	572 バイト	624 バイト	2 枚目 ~ 4 枚目	536 バイト	558 バイト	610 バイト	5 枚目	550 バイト	566 バイト	618 バイト	6 枚目 ~ 8 枚目	536 バイト	558 バイト	610 バイト
CPU ユニット	ファイル容量																															
ハイパフォーマンスモデル QCPU	564																															
プロセス CPU	558																															
二重化 CPU	764																															
CC-Link 設定	モード設定																															
	Ver.1 モード	Ver.2 モード	Ver.2 追加モード																													
1 枚目	550 バイト	572 バイト	624 バイト																													
2 枚目 ~ 4 枚目	536 バイト	558 バイト	610 バイト																													
5 枚目	550 バイト	566 バイト	618 バイト																													
6 枚目 ~ 8 枚目	536 バイト	558 バイト	610 バイト																													
シーケンスプログラム	$136 * 1 + (4 \times ((\text{ステップ数}) + (\text{RUN 中書込用確保ステップ数})))$																															
デバイスコメント	<p>74 + (各デバイスのコメントデータサイズ合計)</p> <ul style="list-style-type: none"> • 1 デバイスのコメントデータサイズ = $10 + 10250 \times a + 40 \times b$ • a : ((デバイス点数) / 256) の商 • b : ((デバイス点数) / 256) の余り 																															
デバイス初期値	<p>$66 + 44 \times n + 2 \times (\text{デバイス初期値で設定しているデバイス点数の合計})$</p> <p>n : デバイス初期値の設定数</p>																															
ユーザ設定エリア	フォーマット時の設定値 (0 ~ 15k)																															
複数ブロック RUN 中書込み設定	フォーマット時の設定値 (0/2k/4k)																															
ファイルレジスタ	$2 \times (\text{ファイルレジスタ点数})$																															
サンプリングトレースファイル	<p>$362 + (\text{ワードデバイス点数} + \text{ビットデバイス点数}) \times 12 + (N1 + N2 + N3 + \text{ワードデバイス点数} \times 2 + (\text{ビットデバイス点数} / 16) \times 2) \times \text{トレース回数 (トータル回数)} * 2$</p> <ul style="list-style-type: none"> • N1 ~ N3 は, トレース条件設定画面のトレース付加情報で設定している項目に応じて下記の値を加算します。 (☞ 6.14 節 (4)(b)) N1 : “時刻 (秒)” を設定時 “4” を加算 N2 : “ステップ No.” を設定時 “10” を加算 N3 : “プログラム名” を設定時 “8” を加算 																															
エラー履歴データ	$72 + 54 \times (\text{故障格納数})$																															

(次ページにつづく)

表 5.12 ファイルのメモリ容量の算出 (ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU) のつづき

機 能	概略ファイル容量 (単位: バイト)
ローカルデバイス	$72 + 6 \times (\text{設定デバイス種類}) + (2 \times ((M, V \text{ の合計点数}) / 16 + (D \text{ 点数}) + 18 \times (T, ST, C \text{ の合計点数}) / 16)) \times (\text{プログラム数})$ <ul style="list-style-type: none"> • M, V, D, T, ST, C は設定されている下記デバイスを示しています。 M: 内部リレー V: エッジリレー D: データレジスタ T: タイマ ST: 積算タイマ C: カウンタ

- * 1: 136 はデフォルト値です。(パラメータの設定により増減します。)
- * 2: ビットデバイス点数 / 16 は, 小数点以下は切り上げとなります。

備 考

メモリ容量の計算例については, 5.4.4 項を参照してください。

5.4.4 ファイルのサイズ単位

(1) ファイルサイズ単位とは

ファイルをメモリエリアに書き込む場合、書き込む CPU ユニットとメモリエリアにより、格納する容量の単位が異なります。

この単位をファイルサイズ単位と呼びます。

(a) メモリエリア別のファイルサイズ単位

書き込む CPU ユニットとメモリエリア別のファイルサイズ単位を示します。

表 5.13 CPU ユニットごとのファイルサイズ単位 (メモリエリア別)

CPU ユニット	メモリエリア	
	プログラムメモリ, 標準 ROM, Flash カード*1 のファイルサイズ単位	標準 RAM
Q00JCPU	1 ステップ / 4 バイト	—
Q00CPU, Q01CPU		4 バイト
Q02CPU, Q02HCPU, Q06HCPU	128 ステップ / 512 バイト*2	512 バイト
Q12HCPU	256 ステップ / 1024 バイト*2	1024 バイト*3
Q25HCPU	512 ステップ / 2048 バイト*2	
Q02PHCPU, Q06PHCPU	128 ステップ / 512 バイト	512 バイト
Q12PHCPU	256 ステップ / 1024 バイト*4	1024 バイト
Q25PHCPU	512 ステップ / 2048 バイト*4	
Q12PRHCPU	256 ステップ / 1024 バイト	
Q25PRHCPU	512 ステップ / 2048 バイト	

* 1: Flash カードのファイルサイズ単位は、GX Developer の [オンライン] → [プログラムメモリの ROM 化] で Flash カードに書き込む場合に適用されます。(☞ 5.2.6 項)

* 2: シリアル No. の上 5 桁が "04121" 以前のハイパフォーマンスモデル QCPU では、1024 ステップ / 4096 バイトになります。

* 3: シリアル No. の上 5 桁が "02091" 以前の Q12HCPU, Q25HCPU では、512 バイトになります。

* 4: シリアル No. の上 5 桁が "07031" 以前のプロセス CPU では、1024 ステップ / 4096 バイトになります。

(b) メモリカード別のファイルサイズ単位

表 5.14 ファイルサイズ単位 (メモリカード別)

種別	形名	ファイルサイズ単位 (クラスサイズ)
SRAM カード	Q2MEM-1MBSN, Q2MEM-1MBS	512 バイト
	Q2MEM-2MBSN, Q2MEM-2MBS	1024 バイト
	Q3MEM-4MBS	1024 バイト
	Q3MEM-8MBS	4096 バイト
Flash カード*1	Q2MEM-2MBF	1024 バイト
	Q2MEM-4MBF	1024 バイト
ATA カード	Q2MEM-8MBA	4096 バイト
	Q2MEM-16MBA	4096 バイト
	Q2MEM-32MBA	2048 バイト

* 1: Flash カードのファイルサイズ単位は、次の場合に適用されます。

- ・ GX Developer の [オンライン] → [PC 書込 (フラッシュ ROM)] で Flash カードに書き込む場合 (☞ 5.2.6 項)
- ・ GX Developer を利用して、CPU ユニットを経由せずに Flash カードに書き込む場合

(2) メモリ容量の計算例

プログラムメモリにパラメータとシーケンスプログラムを書き込んだ場合のメモリ容量の計算例を示します。

(a) 条件

- 1) 書き込み対象 CPU ユニット：Q25HCPU
- 2) 書き込みファイル

表 5.15 各ファイルの容量

ファイル名	ファイル容量* 1
PARAM.QPA (パラメータファイル)	564 バイト
MAIN.QPG (シーケンスプログラム)	525 ステップ / 2100 バイト* 2

* 1：ファイルのメモリ容量については、5.4.3 項を参照してください。

* 2：GX Developer で表示されるプログラム容量（ファイルヘッダ+実行プログラム）を示しています。（ 5.3 節）

- 3) RUN 中書込用確保ステップ：500 ステップ / 2000 バイト

(b) メモリ容量計算

メモリ容量の計算では、書き込み対象 CPU ユニットのファイルサイズ単位を基準に計算します。

本例の Q25HCPU では、ファイルサイズ単位は 512 ステップ / 2048 バイトとなります。（ 本項 (1)）

1) パラメータファイルの容量計算

プログラムメモリ上ではファイルサイズ単位で格納されるため、パラメータファイルの容量は 564 バイトですが、512 ステップ / 2048 バイトの容量を占有します。

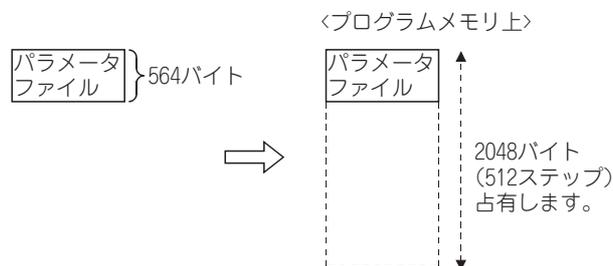


図 5.41 ファイルサイズ単位の占有（パラメータファイル）

2) プログラム容量の計算

プログラムの容量は、シーケンスプログラム容量 + RUN 中書込用確保ステップとなります。
 プログラムメモリ上ではファイルサイズ単位で格納するため、525 ステップ + 500 ステップ = 1025 ステップになりますが、1536 ステップ / 6144 バイトの容量を占有します。

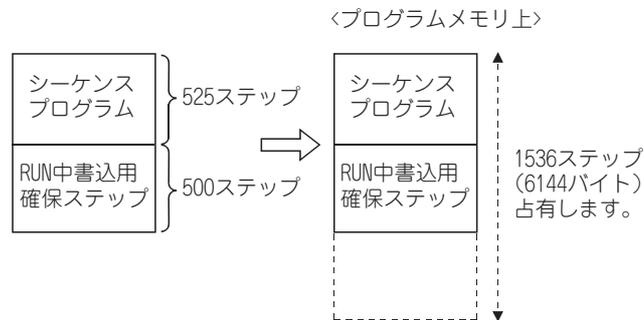


図 5.42 ファイルサイズ単位での占有 (プログラムファイル)

3) 計算結果

メモリ容量の計算結果は、以下のとおりです。

表 5.16 メモリ容量の計算結果

ファイル名	ファイル容量		メモリ容量
PARAM.QPA	564 バイト		512 ステップ (2048 バイト)
MAIN.QPG	シーケンスプログラム容量	525 ステップ	1536 ステップ (6144 バイト)
	RUN 中書込用確保ステップ	500 ステップ	
	合計	1025 ステップ	
メモリ容量合計			2048 ステップ (8192 バイト)

Point

下記に示す CPU ユニットは、ファイルサイズ単位が変更されています。

- シリアル No. の上 5 桁が “04122” 以降のハイパフォーマンスモデル QCPU
- シリアル No. の上 5 桁が “07032” 以降のプロセス CPU

このため、下記の点に注意してください。

- ① 上記に示したシリアル No. 以降の CPU ユニットで稼動していたファイルは、ファイルの容量により上記に示したシリアル No. より前の CPU ユニットに格納できない場合があります。
- ② ファイルを CPU ユニットから GX Developer に読出し、別の CPU ユニットに PC 書込みする場合、CPU ユニットと GX Developer のバージョンの組合せにより、制約があります。

表 5.17 ハイパフォーマンスモデル QCPU の場合

書込み先	書込み元			
	GX Developer Version 8		GX Developer Version 7	
	シリアル No. “04122” 以降の CPU の稼動ファイル	シリアル No. “04121” 以前の CPU の稼動ファイル	シリアル No. “04122” 以降の CPU の稼動ファイル	シリアル No. “04121” 以前の CPU の稼動ファイル
シリアル No. “04122” 以降	○	○	△ * 2	△ * 2
シリアル No. “04121” 以前	△ * 1	○	△ * 1 * 2	△ * 2

○：PC 書込み可能, △：制約事項あり

- * 1：ファイルサイズ単位が異なるため、ファイルの容量により CPU ユニットに格納できない場合があります。
- * 2：RUN 中書込用確保ステップ数を減らさないと、ファイルの容量により CPU ユニットに格納できない場合があります。

表 5.18 プロセス CPU の場合

書込み先	書込み元			
	GX Developer Version 8		GX Developer Version 7	
	シリアル No. “07032” 以降の CPU の稼動ファイル	シリアル No. “07031” 以前の CPU の稼動ファイル	シリアル No. “07032” 以降の CPU の稼動ファイル	シリアル No. “07031” 以前の CPU の稼動ファイル
シリアル No. “07032” 以降	○	○	△ * 2	△ * 2
シリアル No. “07031” 以前	△ * 1	○	△ * 1 * 2	△ * 2

○：PC 書込み可能, △：制約事項あり

- * 1：ファイルサイズ単位が異なるため、ファイルの容量により CPU ユニットに格納できない場合があります。
- * 2：RUN 中書込用確保ステップ数を減らさないと、ファイルの容量により CPU ユニットに格納できない場合があります。

第6章 機能

CPU ユニットの機能について説明します。

6.1 機能一覧

CPU ユニットの機能一覧を表 6.2 に示します。

表中の“CPU ユニット”欄に示す番号は、以下の CPU ユニットに対応しています。

表 6.1 CPU ユニットの機能一覧

番号	対応 CPU ユニット
1)	ベーシックモデル QCPU
2)	ハイパフォーマンスモデル QCPU
3)	プロセス CPU
4)	二重化 CPU

表 6.2 CPU ユニットの機能一覧

機能名	説明	CPU ユニット				参照
		1)	2)	3)	4)	
コンスタントスキャン	プログラムを一定間隔で実行させる機能です。	○	○	○	○	6.2 節
ラッチ機能	電源 OFF → ON, リセット操作時にデバイスのデータを保持する機能です。	○	○	○	○	6.3 節
STOP → RUN にしたときの出力状態の選択機能	CPU ユニットの STOP 状態から RUN 状態にしたときの出力 (Y) の状態 (STOP 前の出力 (Y) 状態を出力 / 出力 (Y) をクリア) を選択する機能です。	○	○	○	○	6.4 節
時計機能	CPU ユニット内部の時計データをシーケンスプログラムで読み出し、時間管理に使用する機能です。	○	○	○	○	6.5 節
リモート RUN/STOP	CPU ユニットの演算を外部から実行 / 停止する機能です。	○	○	○	○	6.6.1 項
リモート PAUSE	CPU ユニットの出力 (Y) を保持したまま、CPU ユニットの演算を外部から停止する機能です。	○	○	○	○	6.6.2 項
リモート RESET	CPU ユニットが STOP 状態のとき、CPU ユニットの外部からリセットする機能です。	○	○	○	○	6.6.3 項
リモートラッチクリア	CPU ユニットが STOP 状態のとき、CPU ユニットのラッチデータをクリアする機能です。	○	○	○	○	6.6.4 項
入力応答時間選択	Q シリーズ対応の入力ユニット、入出力混合ユニット、高速入力ユニット、割込みユニットの入力応答時間を選択する機能です。	○	○	○	○	6.7 節
エラー時の出力モード	Q シリーズ対応の出力ユニット、入出力混合ユニット、インテリジェント機能ユニット、割込みユニットへの出力を CPU ユニットが停止エラー時にクリアするか、保持するかを設定する機能です。	○	○	○	○	6.8 節
ハードウェアエラー時の CPU 動作モード設定	インテリジェント機能ユニット、割込みユニットのハードウェアエラーが発生したときに CPU ユニットの演算を停止させるか続行させるかを設定する機能です。	○	○	○	○	6.9 節
インテリジェント機能ユニットのスイッチ設定	インテリジェント機能ユニット、割込みユニットの各種設定を行う機能です。(設定内容は、各インテリジェント機能ユニット、割込みユニットのマニュアルを参照してください。)	○	○	○	○	6.10 節

○：使用可, △：一部制約あり, ×：使用不可

(次ページへつづく)

表 6.2 CPU ユニットの機能一覧 (つづき)

機能名	説明	CPU ユニット				参照
		1)	2)	3)	4)	
モニタ機能	GX Developer から CPU ユニットのプログラム、デバイスの状態を読み出す機能です。	○	○	○	○	6.11 節
モニタ条件の設定	CPU ユニットのモニタするタイミングをデバイスの条件やステップ No. で指定してモニタする機能です。	×	○	○	○	6.11.1 項
ローカルデバイスのモニタ・テスト	GX Developer で指定プログラムのローカルデバイスのモニタ・テストを行う機能です。	×	○	○	○	6.11.2 項
外部入出力の強制 ON/OFF	GX Developer から CPU ユニットの外部入出力を強制的に ON/OFF する機能です。	×	△ *4	○	○	6.11.3 項
RUN 中書き込み	CPU ユニットの RUN 中にプログラムを書き込む機能です。	△ *5	○	○	○	6.12 節
プログラム一覧モニタ	実行中のプログラムのスキャンタイム、実行状態を表示する機能です。	○	○	○	○	6.13.1 項
割り込みプログラム一覧モニタ	割り込みプログラムの実行回数を表示する機能です。	○	○	○	○	6.13.2 項
スキャンタイム測定	プログラムの任意ステップ間の実行時間を計測する機能です。	×	○	○	○	6.13.3 項
サンプリングトレース機能	指定デバイスデータを指定タイミングで連続して収集する機能です。	×	○	○	○	6.14 節
複数の GX Developer からのデバッグ機能	複数の GX Developer から同時にデバッグを行う機能です。	○	○	○	○	6.15 節
ウォッチドッグタイマ	CPU ユニットのハードウェア、プログラム異常などによる演算渋滞を監視する機能です。	○	○	○	○	6.16 節
自己診断機能	CPU ユニット自身で異常の有無の診断を行う機能です。	○	○	○	○	6.17 節
エラー履歴	自己診断結果をエラー履歴としてメモリに格納しておく機能です。	○	○	○	○	6.18 節
システムプロテクト	GX Developer, Ethernet ユニット, シリアルコミュニケーションユニットなどからのプログラム変更を防止する機能です。	△ *1	○	○	○	6.19 節
パスワード登録	GX Developer から CPU ユニットの各ファイルの書き込み/読出しを禁止する機能です。	○	○	○	○	6.19.1 項
リモートパスワード	シリアルコミュニケーションユニット, Ethernet ユニットによる外部からの不正アクセスを防止する機能です。	△ *4	△ *4	○	○	6.19.2 項
システム表示	GX Developer で、システム構成をモニタする機能です。	○	○	○	○	6.20 節
LED 表示	CPU ユニット前面の LED で CPU ユニットの動作状態を表示する機能です。	○	○	○	○	6.21 節
優先順位の設定	異常発生時に表示器データ (SD220 ~ 227) に格納されるエラーメッセージの優先順位を設定する機能です。 また、LED 表示を非表示に設定できます。	○	○	○	○	6.21.2 項
高速割り込み機能	割り込みポインタ I49 を使用して 0.2ms ~ 1.0ms 間隔の定周期割り込みによる割り込みプログラムを実行する機能です。	×	△ *3	×	×	6.22 節
インテリジェント機能ユニットからの割り込み	インテリジェント機能ユニットからの割り込み要求により、割り込みプログラムを実行する機能です。	△ *4	○	○	○	6.23 節
シリアルコミュニケーション機能	CPU ユニットの RS-232 インタフェースとパソコン/表示器などを RS-232 ケーブルで接続し、MC プロトコルによる通信を行う機能です。	△ *2	×	×	×	6.24 節
ユニットサービス間隔読出し	インテリジェント機能ユニット, ネットワークユニットまたは GX Developer のサービス間隔時間 (CPU ユニットのアクセス受付から次のアクセス受付までの時間) をモニタする機能です。	○	○	○	○	6.25.1 項

○：使用可，△：一部制約あり，×：使用不可

(次ページへつづく)

表 6.2 CPU ユニットの機能一覧 (つづき)

機能名	説明	CPU ユニット				参照
		1)	2)	3)	4)	
デバイス初期値	プログラムで使用するデータを、プログラムレスでデバイス、インテリジェント機能ユニット、特殊機能ユニットのバッファメモリに登録する機能です。	△ *4	○	○	○	6.26 節
メモリチェック機能	過度の電氣的ノイズなどにより CPU ユニットのメモリの内容が書き換わっていないかをチェックする機能です。	×	×	△ *4	○	6.27 節
オンラインユニット交換	オンライン中に Q シリーズの入力/出力ユニット、機能バージョン C のインテリジェント機能ユニット、MELSECNET/H リモート I/O 局または、増設ベースユニットに装着されている入力/出力ユニット、インテリジェント機能ユニットを交換する機能です。 電源を二重化している場合は、オンライン中に電源ユニットの交換も可能です。	×	×	○	○	QCPU ユーザーズ マニュアル (ハード ウェア設計・保守点検 編)
オートチューニング機能	オートチューニングは、PID 定数の初期設定を行うためのものです。 また、温度調節などの比較的応答の緩やかなプロセスで、S.PID、S.2PID 命令を使用したループにて使用できます。	×	×	○	○	QnPHCPU/ QnPRHCPU プログラ ミングマニュアル (プ ロセス制御命令編)
二重化システム機能	CPU ユニット、電源ユニット、ネットワークユニット、基本ベースユニットを二重化する機能です。	×	×	×	○	QnPRHCPU ユーザーズマニュアル (二重化システム編)
系切替え機能 (制御系と待機系の切替え)	制御系と待機系を切り替える (制御系を待機系にし、待機系を制御系に切り替える。) 機能です。 システム切替えとユーザ切替えの 2 種類があります。	×	×	×	○	
運転モードの変更	セパレートモードとバックアップモードを切り替える機能です。	×	×	×	○	
トラッキング転送機能	制御系と待機系でデータを共有するための機能です。(制御系のデータを待機系に転送する。) 制御系の故障/異常時に系切替えが発生しても、同一データで制御の継続が可能です。	×	×	×	○	
オンラインプログラム書込みの二重化追従機能	制御系に PC 書込み、RUN 中書込みで書込んだデータを待機系にも転送する機能です。	×	×	×	○	

○：使用可，△：一部制約あり，×：使用不可

- * 1：ベーシックモデル QCPU では、ディップスイッチによるシステムプロテクトはできません。
- * 2：Q00JCPU では使用できません。
- * 3：Q02CPU では使用できません。
- * 4：使用可否は、CPU ユニットのバージョンで異なります。(付 2)
- * 5：ファイルの RUN 書込みはできません。

6.2 コンスタントスキャン

(1) コンスタントスキャンとは

スキャンタイムは、シーケンスプログラムで使用している命令の実行／非実行により処理時間が異なるため、毎スキャン同一ではなく変化します。

コンスタントスキャンは、スキャンタイムを一定時間に保ちながらシーケンスプログラムを繰返し実行させる機能です。

(2) コンスタントスキャンの用途

I/O リフレッシュは、シーケンスプログラムの実行前に行っています。

コンスタントスキャン機能を使用することにより、シーケンスプログラムの実行時間が変化しても、I/O リフレッシュの間隔を一定にできます。

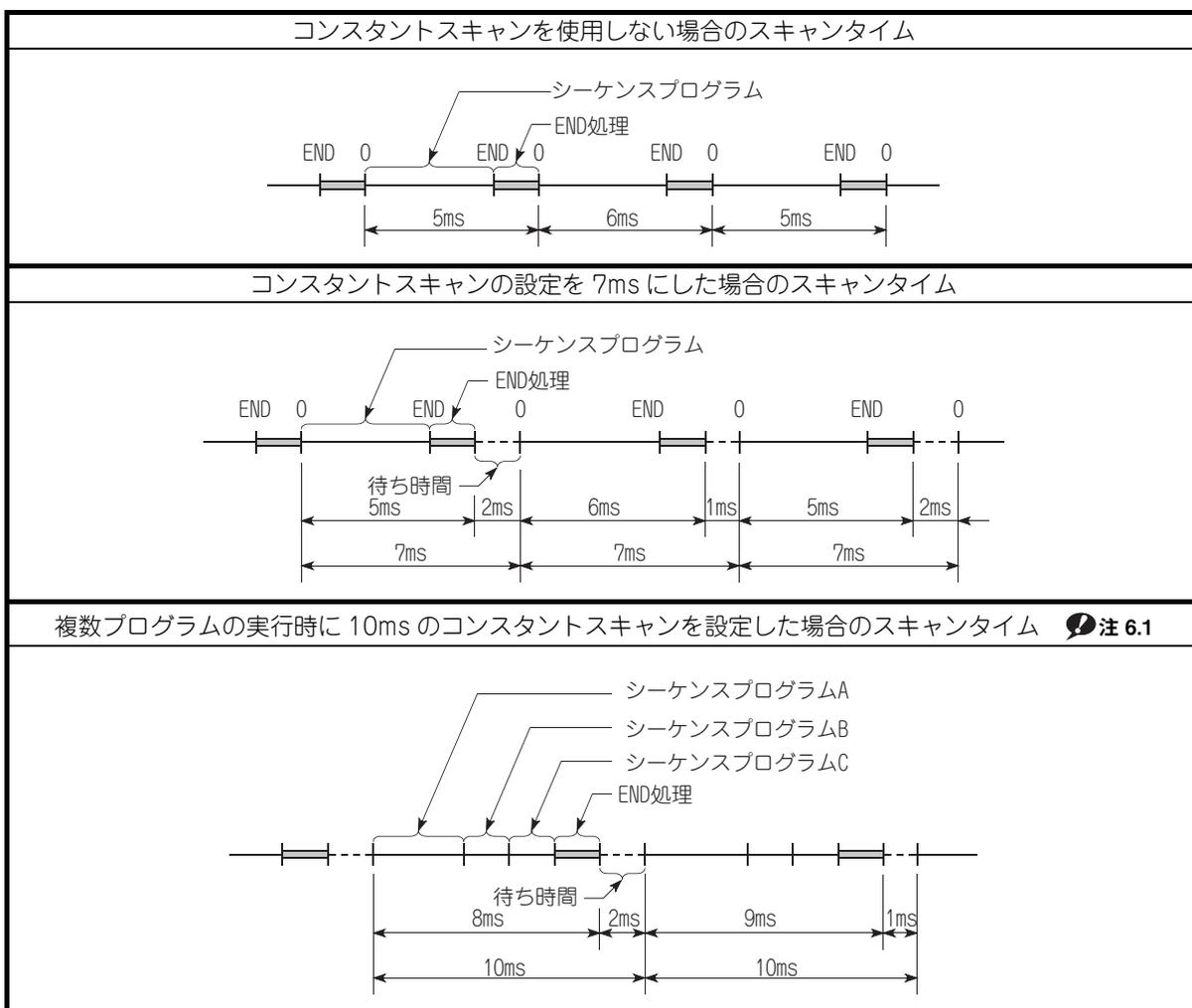


図 6.1 コンスタントスキャンの動作

注 6.1 Basic

ベーシックモデル QCPU では、複数プログラムが実行できないため、複数プログラム実行時のスキャンタイムを意識する必要はありません。

(3) コンスタントスキャン時間の設定

PC パラメータの PC RAS 設定で行います。
設定範囲は、CPU ユニットにより異なります。

- ベーシックモデル QCPU の場合：1 ~ 2000ms（設定単位は 1ms）
- ハイパフォーマンスモデル QCPU，プロセス CPU，二重化 CPU の場合：
0.5 ~ 2000ms（設定単位は 0.5ms）

コンスタントスキャンを実行しない場合は、コンスタントスキャン時間を “ブランク（空白）” にします。

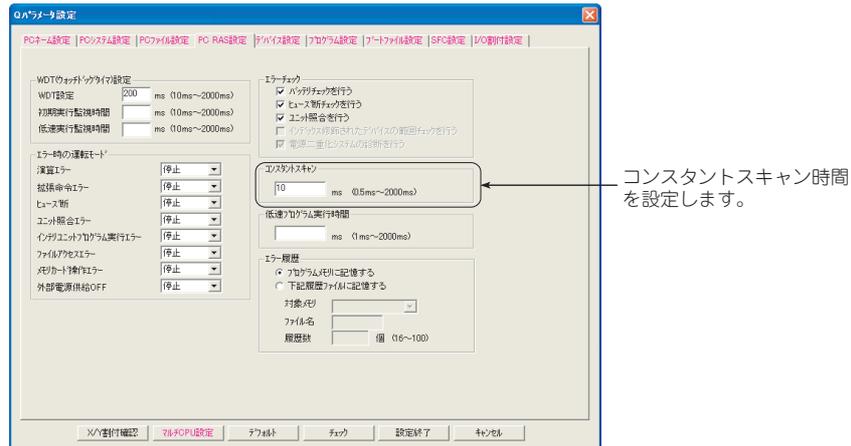


図 6.2 コンスタントスキャンを 10ms に設定した場合

(a) 設定時間の条件

コンスタントスキャンの設定時間は、下記の関係式を満たす値を設定してください。

$$\text{(WDT の設定時間)} > \text{(コンスタントスキャンの設定時間)} > \text{(シーケンスプログラムの最大スキャンタイム)}$$

シーケンスプログラムのスキャンタイムが、コンスタントスキャンの設定時間より長い場合、CPU ユニットは“PRG.TIME OVER”（エラーコード：5010）を検出します。

この場合は、コンスタントスキャンを無視してシーケンスプログラムのスキャンタイムで実行します。

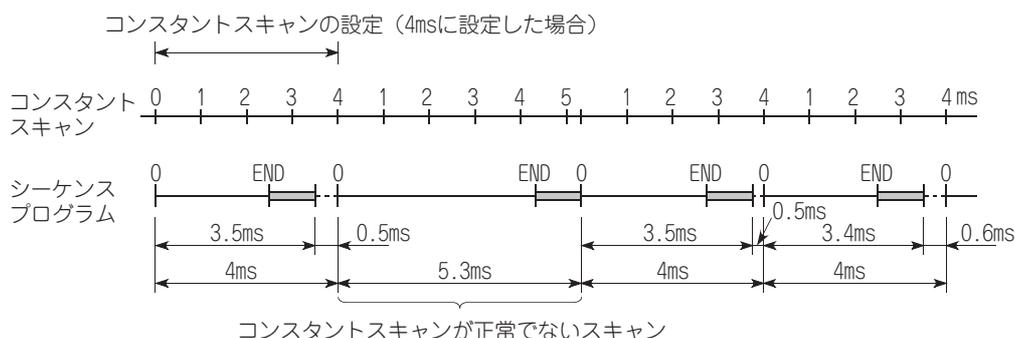


図 6.3 スキャンタイムがコンスタントスキャンより長い場合の動作

シーケンスプログラムのスキャンタイムが、WDT の設定時間よりも長い場合、CPU ユニットは“WDT ERROR”を検出します。

この場合は、プログラムの実行を停止します。

(4) END 処理実行から次のスキャンが始まるまでの待ち時間について

シーケンスプログラムの END 処理実行から、次のスキャンが始まるまでの待ち時間の間は、シーケンスプログラムの処理を中止します。

(a) 低速実行タイププログラムを実行している場合

下記に示す時間で低速実行タイププログラムの実行を中断します。

（コンスタントスキャン）－ 0.5ms

(b) 次のスキャンが始まるまでの待ち時間に割込み要因が発生した場合

下記のプログラムを実行します。

- 割込みプログラム
- 定周期実行タイププログラム

(5) コンスタントスキヤンの精度

コンスタントスキヤンの精度を、表 6.3 に示します。

表 6.3 CPU ユニットの機能一覧

CPU ユニット	モニタ無, ユーザ割込み無	モニタ有, ユーザ割込み無	モニタ無, ユーザ割込み有	モニタ有, ユーザ割込み有
Q00JCPU	0.20 ms	0.90 ms	割込みプログラムの実行時間 (10.1.2 項 (4) (a) 参照)	下記の合計時間 ① 左記 “モニタ有, ユーザ 割込み無” 欄に示す時間 ② 割込みプログラムの実行 時間の合計時間
Q00CPU	0.12 ms	0.60 ms		
Q01CPU	0.10 ms	0.50 ms		
Q02CPU	0.02 ms			
ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU	0.01 ms			

- ・モニタ有：GX Developer を接続し、モニタを行っている状態またはシリアルコミュニケーション機能により外部機器と通信を行っている状態を示します。
- ・モニタ無：GX Developer およびシリアルコミュニケーション機能による通信を行っていない状態を示します。

ただし、下記の場合、コンスタントスキヤンが延びる可能性があります。

(a) 低速実行タイププログラムを実行している場合

0.5ms の待ち時間があります。

- ・1つの命令の最大処理時間が 0.5ms 以内の場合
コンスタントスキヤンの誤差は、表 6.3 に示すコンスタントスキヤンの精度と同様です。
- ・1つの命令の最大処理時間が 0.5ms を超える場合
コンスタントスキヤンタイムが、0.5ms を超えた時間だけずれる場合があります。
また、“PRG, TIME OVER” (エラーコード：5010) を検出します。

1つの命令の最大処理時間については、下記マニュアルを参照してください。

 MELSEC-Q/L プログラミングマニュアル (共通命令編)

(b) 割込み／定周期実行タイププログラムを実行している場合

割込み／定周期実行タイププログラム実行中は、割込み禁止です。

このため、割込み／定周期実行タイププログラム実行中にコンスタントスキヤンタイムに達しても、コンスタントスキヤンを終了できません。

また、割込み／定周期実行タイププログラムを実行している場合は、割込み／定周期実行タイププログラムの実行時間だけコンスタントスキヤンの設定時間を超えて、スキヤンタイムが延びる場合があります。

(c) コンスタントスキヤンの終了間際にサービス処理が行われる場合

コンスタントスキヤンの終了間際にサービス処理が行われると、コンスタントスキヤンの設定時間を超えて、スキヤンタイムが延びる場合があります。

6.3 ラッチ機能

(1) ラッチ機能とは

下記の場合に、CPU ユニットの各デバイスの内容を保持しておく機能です。

- CPU ユニットの電源 OFF → ON 時
- リセット操作時
- 許容瞬停時間を超えた停電

ラッチ機能を使用しないと、CPU ユニットの各デバイスはデフォルト値（ビットデバイス：OFF，ワードデバイス：0）に戻ります。

(2) ラッチ機能の用途

ラッチは、連続した制御でデータの管理を行っているときに、CPU ユニットの電源 OFF → ON などでも、直前のデータを保持して制御を継続させる場合に使用できます。

(3) ラッチ使用時の演算

プログラムでの演算は、ラッチの有／無に関係なく同一です。

(4) ラッチ可能なデバイス

ラッチ可能なデバイスは下記のとおりです。

(デフォルトでは、ラッチリレーのみラッチが有効になっています。)

- ラッチリレー (L)
- リンクリレー (B)
- アナンシェータ (F)
- エッジリレー (V)
- タイマ (T)
- 積算タイマ (ST)
- カウンタ (C)
- データレジスタ (D)
- リンクレジスタ (W)

(5) ラッチ範囲設定

PC パラメータのデバイス設定で行います。

ラッチ範囲設定には、ラッチクリア操作が有効な範囲（ラッチ（1））の設定と無効な範囲（ラッチ（2））の設定の2種類があります。



図 6.4 ラッチ範囲設定

(6) ラッチ範囲のデバイスデータの保持方法とスキャンタイムへの影響

ラッチ範囲のデバイスにデータを書き込むと同時にラッチを行います。

ラッチのための処理は行わないので、ラッチによるスキャンタイムへの影響はありません。

(7) ラッチ範囲のデバイスデータのクリア

ラッチクリアを行った場合のデバイスの状態を表 6.4 に示します。

表 6.4 ラッチクリアを行った場合の状態

ラッチの有無	ラッチクリアによるクリア/保持
ラッチ範囲に指定していないデバイス	クリア
ラッチ(1)に設定したデバイス	クリア
ラッチ(2)に設定したデバイス	保持* 1

* 1：クリア方法については 3.7 節を参照してください。

Point

ファイルレジスタ (R, ZR) はラッチクリアを行ってもクリアされません。
 ファイルレジスタ (R, ZR) のクリアは、シーケンスプログラムまたは GX Developer によるデータクリア操作で行います。(参照 9.7.6 項 (3))

(8) 注意事項

(a) ローカルデバイス指定またはデバイス初期値指定を行っている場合

ラッチ指定したデバイスでも、ラッチされません。

(b) バッテリー使用について

ラッチ範囲のデバイス内容は、CPU ユニットに取り付けられているバッテリーで保持しています。

- CPU ユニットでブート運転を行う場合でも、ラッチを行う場合はバッテリーが必要です。
- シーケンサの電源が OFF 中に、バッテリーのコネクタを CPU ユニットのコネクタからはずすと、ラッチ範囲のデバイスの内容は保持されず、不定の値になりますので注意してください。

(c) 二重化 CPU のスタートモードがホットスタートモードの場合

ラッチ範囲に設定していないデータも保持されます。(インデックスレジスタおよびステップリレーなどの一部デバイスを除く。)

これらのデータをクリアする場合は、ラッチクリア操作 (図 3.7 節 (2) (b)) を行ってください。

6.4 STOP 状態→RUN 状態にしたときの出力 (Y) 状態の設定

(1) STOP 状態→RUN 状態にしたときの出力 (Y) 状態の設定とは

CPU ユニットは、RUN 状態などから STOP 状態にすると、RUN 状態の出力 (Y) をシーケンサ内部に記憶し、出力 (Y) をすべて OFF にします。

GX Developer のパラメータ設定で、STOP 状態→RUN 状態にしたときの状態を次の 2 つから選択できます。

- STOP 前の出力 (Y) 状態を出力する。
- 出力 (Y) をクリアする。

(2) 設定用途

保持回路などで、STOP 状態→RUN 状態にしたときに出力を前回の状態から再開するか、再開しないかを選択できます。



図 6.5 保持回路

- STOP 前の出力 (Y) 状態を出力に設定した場合

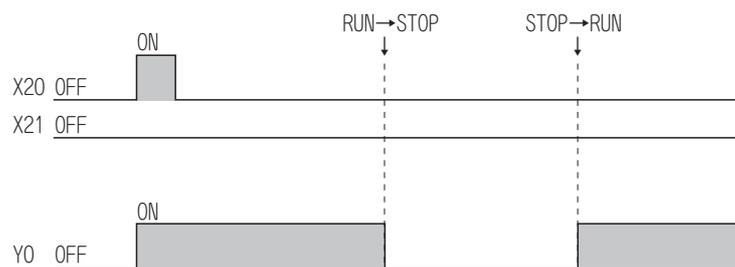


図 6.6 STOP 前の出力 (Y) 状態を出力に設定した場合のタイムチャート

- 出力 (Y) をクリアに設定した場合

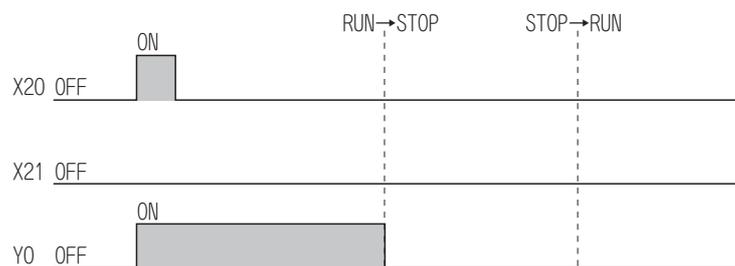


図 6.7 出力 (Y) をクリアに設定した場合のタイムチャート

(3) STOP 状態から RUN 状態にしたときの動作

(a) STOP 前の出力 (Y) 状態を出力 (デフォルト)

STOP 状態になる直前の出力 (Y) 状態を出力後、シーケンスプログラムの演算を行います。

(b) 出力 (Y) をクリア (出力は 1 スキャン後)

出力は OFF した状態になります。

出力 (Y) の出力は、シーケンスプログラムの演算実行後に行います。

STOP 状態のときに出力 (Y) を強制 ON した場合の動作は、本節 (5) を参照ください。

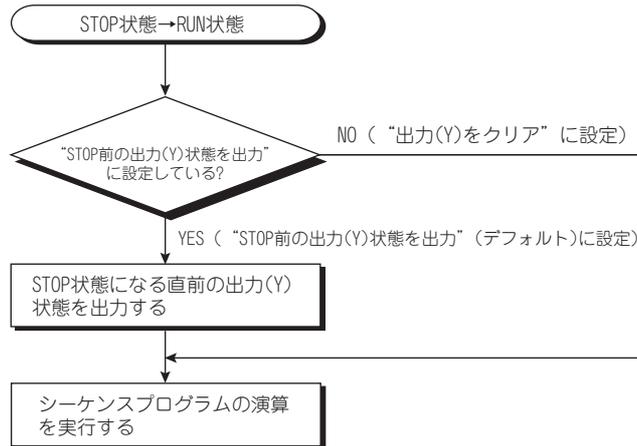


図 6.8 STOP 状態→RUN 状態にしたときの動作

(4) STOP 状態から RUN 状態にしたときの出力 (Y) 状態の設定

PC パラメータの PC システム設定で行います。



図 6.9 PC システム設定画面

(5) 注意事項

CPU ユニットが STOP 状態のとき出力 (Y) を強制 ON した後に、STOP 状態から RUN 状態にしたときの出力を表 6.5 に示します。

表 6.5 STOP 状態のときに出力 (Y) を強制 ON した後に、STOP 状態から RUN 状態にしたときの出力

STOP → RUN 時の出力モード	STOP 状態から RUN 状態にしたときの出力
STOP 前の出力 (Y) 状態を出力	STOP 前の状態を出力します。 STOP 前に出力 (Y) が OFF なら、ON 状態を保持しません。
出力 (Y) をクリア	ON 状態を保持します。

6.5 時計機能

(1) 時計機能とは

CPU ユニット内部の時計データをシーケンスプログラムで読出し、時間管理に使用する機能です。
時計データは、エラー履歴への日付の格納などで CPU ユニットのシステムが行う機能の時間管理などに使用します。

(2) 電源 OFF および瞬停時の時計動作

時計動作は、シーケンサの電源 OFF 中または許容瞬停時間を越えた停電発生時でも、CPU ユニットの内部のバッテリーにより継続しています。

(3) 時計データ

CPU ユニットの内部で使用する時計データで、表 6.6 に示すものがあります。

表 6.6 時計データの内容

データ名称	内容	
年	西暦で 4 桁* ¹ (1980 年～2079 年まで計測可能)	
月	1～12	
日	1～31 (閏年自動判別)	
時	0～23 (24 時間制)	
分	0～59	
秒	0～59	
曜日	0	日曜日
	1	月曜日
	2	火曜日
	3	水曜日
	4	木曜日
	5	金曜日
	6	土曜日
1/1000 秒* ²	0～999	

* 1：西暦は上 2 桁と下 2 桁に分けて SD213, SD210 に格納します。

* 2：拡張時計データの読出し命令 (S(P).DATERD) で読み出すことができます。

 MELSEC-Q/L プログラミングマニュアル (共通命令編)

(4) 時計データの変更と読出し

(a) 時計データの変更

時計データの変更方法には、GX Developer で行う方法とプログラムで行う方法の 2 種類があります。

1) GX Developer で行う方法

GX Developer の [オンライン] → [時計設定] で時計設定画面を表示し、時計データを変更します。

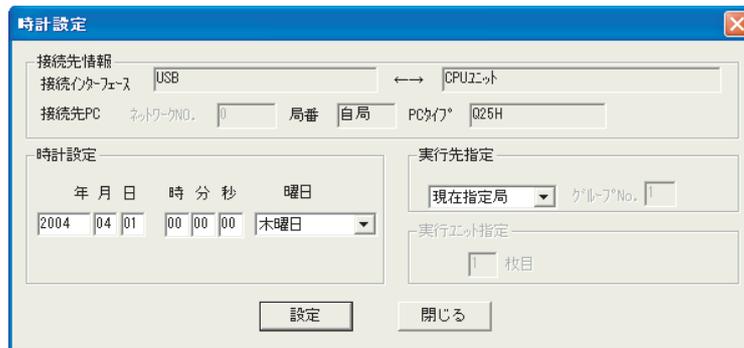


図 6.10 時計設定画面

2) プログラムで行う方法

時計データの書き込み命令 (DATEWR) により、時計データを変更します。
D0 ~ 6 に設定した時計データの書き込みプログラムを図 6.11 に示します。

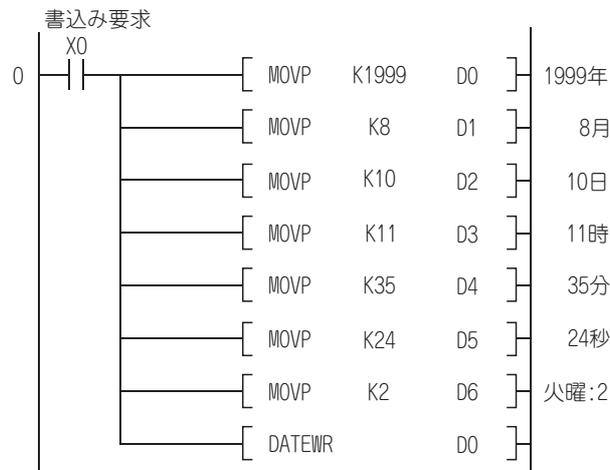


図 6.11 時計データを書き込むプログラム例

DATEWR 命令の詳細については、下記マニュアルを参照してください。

MELSEC-Q/L プログラミングマニュアル (共通命令編)

Point

- GX Developer またはプログラムで時計データを変更すると、1/1000 秒の時計は 0 にリセットされます。
- 時計データの年の最大設定値は、GX Developer では 2037 年です。

(b) 時計データの読出し

時計データをデータレジスタに読み出す場合は、プログラムで下記のいずれかの命令を使用してください。

- 時計データの読出し命令 (DATERD)
- 拡張時計データの読出し命令 (S(P).DATERD)

時計データの読出し命令 (DATERD) で読み出した時計データを、D10～16に格納するプログラム例を図6.12に示します。

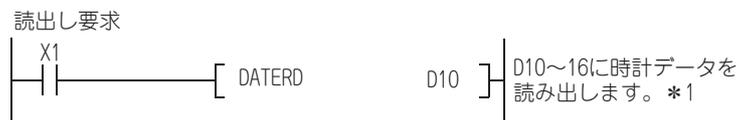


図 6.12 時計データを格納するプログラム

* 1 : D10～16に格納されるデータを図6.13に示します。

D10	2004	西暦4桁	} (☞ 本節(3))
D11	4	月	
D12	1	日	
D13	11	時	
D14	35	分	
D15	24	秒	
D16	2	曜日	

図 6.13 格納された時計データ

DATERD および S(P).DATERD 命令の詳細については、下記マニュアルを参照してください。

☞ MELSEC-Q/L プログラミングマニュアル (共通命令編)

Point

時計データの書込み／読出しは、特殊リレー (SM210～213) と特殊レジスタ (SD210～213) で行うこともできます。特殊リレー、特殊レジスタの詳細については、下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編)

(5) 注意事項

(a) 初めて使用する際の時計データ設定

時計データは、出荷時には設定されていません。

時計データは、エラー履歴など CPU ユニットのシステムおよびインテリジェント機能ユニットが使用します。

CPU ユニットを初めて使用する際は、必ず正確な時間を設定してください。

(b) 時計データの修正

時計データの一部を修正する場合でも、すべてのデータを再度 CPU ユニットに書き込んでください。

(c) 変更する時計データの範囲

変更する時計データは、本項 (3) に示す範囲で書き込んでください。

時刻としてありえないデータを CPU ユニットに書き込んだ場合は、時計機能が正常に動作しません。ただし、時刻としてありえないデータでも、本項 (3) に示す範囲内であればエラーになりません。

表 6.7 時計データに書き込むデータの例

	CPU ユニットへの書き込み動作	CPU ユニットの動作状態
2月30日	実行する	エラー検出されない
13月32日	実行しない	DATEWR 命令実行時：“OPERATION ERROR” (エラーコード：4100) SM210 ON 時：SM211 が ON

(d) 1/1000 秒の時計データの使用

1) 1/1000 秒の時計データが使用可能な機能

下記に示す命令でのみ使用可能です。

- S(P).DATERD
- S(P).DATE+
- S(P).DATE-

上記に示す命令以外では、1/1000 秒の時計データは使用できません。

(SM/SD による読出し、エラー履歴に格納するエラーの発生時刻、GX Developer による読出し、他ユニットの専用命令による読出しなど)

2) 時計データを変更した場合

GX Developer または命令 (他ユニットの専用命令も含む) により、時計データを変更すると、1/1000 の時計データは 0 にリセットされます。

(6) 時計データの精度

時計機能の精度は、周囲温度により異なります。

表 6.8 時計機能の精度

周囲温度 (°C)	精度 (日差, S)		
	ベーシックモデル QCPU	ハイパフォーマンスモデル QCPU, プロセス CPU	二重化 CPU
0	- 3.2 ~ + 5.27 (TYP. + 1.98)	- 3.18 ~ + 5.25 (TYP. + 2.12)	- 3.2 ~ + 5.27 (TYP. + 2.07)
+ 25	- 2.57 ~ + 5.27 (TYP. + 2.22)	- 3.93 ~ + 5.25 (TYP. + 1.9)	2.77 ~ + 5.27 (TYP. + 2.22)
+ 55	- 11.68 ~ + 3.65 (TYP. - 2.64)	- 14.69 ~ + 3.53 (TYP. - 3.67)	- 12.14 ~ + 3.65 (TYP. - 2.89)

(7) 時計データの比較

時計データを読み出してシーケンスプログラムで比較を行う場合は、時計データを時計データ読出し命令 (DATERD) で読み出してください。

時計データ読出し命令 (DATERD) では、年データを西暦 4 桁で読み出すので、そのまま比較命令で比較できます。

6.6 リモート操作

リモート操作とは、外部（GX Developer、MC プロトコルを使用した外部機器、CC-Link IE コントローラネットワークユニットまたは、MELSECNET/H ユニットのリンク専用命令、リモート接点など）から CPU ユニットの動作状態を変更する操作です。

リモート操作には次の 4 種類があります

- リモート RUN/STOP :  6.6.1 項
- リモート PAUSE :  6.6.2 項
- リモート RESET :  6.6.3 項
- リモートラッチクリア :  6.6.4 項

6.6.1 リモート RUN/STOP

(1) リモート RUN/STOP とは

CPU ユニットの RUN/STOP スイッチ（ベーシックモデル QCPU の場合は RUN/STOP/RESET スイッチ）を RUN の位置にしたままで、外部から CPU ユニットの RUN/STOP を行う操作です。

(2) リモート RUN/STOP の用途

次のような場合に使用すると便利です。

- CPU ユニットが手の届かない場所にあるとき
- 制御盤内の CPU ユニットの外部信号により RUN/STOP するとき

(3) リモート RUN/STOP 時の演算

リモート RUN/STOP を行った場合、プログラムの演算は下記ようになります。

(a) リモート STOP 時

プログラムを END 命令まで実行して、STOP 状態になります。

(b) リモート RUN 時

STOP 状態のときにリモート RUN を行うと再度 RUN 状態となり、プログラムをステップ 0 から実行します。

(4) リモート RUN/STOP の実行方法

リモート RUN/STOP の実行方法は、下記の 3 種類があります。

- RUN 接点による方法
- GX Developer, MC プロトコルを使用した外部機器による方法
- CC-Link IE コントローラネットワークユニットまたは、MELSECNET/H ユニットのリンク専用命令による方法

(a) RUN 接点による方法

PC パラメータの PC システム設定で RUN 接点を設定します。
設定可能なデバイス範囲は、CPU ユニットにより異なります。

- ベーシックモデル QCPU：入力 X0 ～ 7FF
- ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU：入力 X0 ～ 1FFF

設定した RUN 接点の ON/OFF により、リモート RUN/STOP を行います。

- RUN 接点が OFF の場合、CPU ユニットは RUN 状態になります。
- RUN 接点が ON の場合、CPU ユニットは STOP 状態になります。

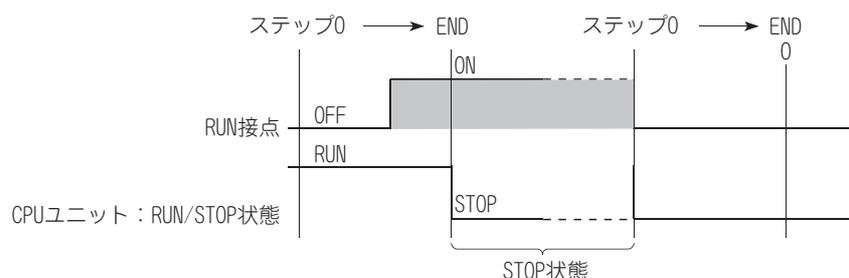


図 6.14 RUN 接点によるリモート RUN/STOP

(b) GX Developer, MC プロトコルを使用した外部機器などによる方法

GX Developer の [オンライン] → [リモート操作] で行います。

外部機器からリモート RUN/STOP を行う場合は、MC プロトコルのコマンドで行います。

🔗 MELSEC コミュニケーションプロトコルリファレンスマニュアル

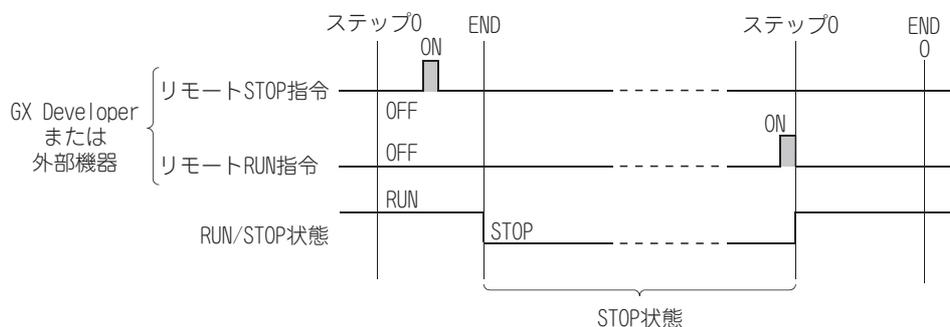


図 6.15 GX Developer, 外部機器によるリモート RUN/STOP

**(c) CC-Link IE コントローラネットワークユニットまたは MELSECNET/H ユニットの
リンク専用命令による方法**

CC-Link IE コントローラネットワークユニットまたは MELSECNET/H ユニットのリンク専用命令による
リモート RUN/STOP 操作で、CPU ユニットの RUN/STOP を行うことができます。

詳細については、下記マニュアルを参照してください。

 各ネットワークユニットのマニュアル

(5) 注意事項

CPU ユニットは、STOP 優先になっているため、次の点に注意してください。

(a) STOP 状態になるタイミング

CPU ユニットは、RUN 接点、GX Developer、MC プロトコルを使用した外部機器のいずれか 1 つから
リモート STOP を行うと STOP 状態になります。

(b) リモート STOP 後に再度 RUN 状態にする場合

リモート STOP で CPU ユニットの STOP 状態にしたあと、再度 RUN 状態にする場合は、最初にリモート
STOP を行った順序でリモート RUN を行ってください。

(c) 二重化 CPU 使用時

1) リモート操作画面にて両系指定を行っていない場合

リモート RUN/STOP は、接続先指定などで指定している系に対してのみ行います。

2) リモート操作画面にて両系指定を行っている場合

リモート RUN/STOP は、制御系および待機系の両方に行います。

ただし、両系指定によるリモート操作では、二重化 CPU がバックアップモードの場合のみ可能です。

Point 

-
- RUN/STOP 状態は、下記のとおりです。
 - RUN 状態：シーケンスプログラムのステップ 0 ～ END/FEND 命令までの演算を繰返し実行している状態です。
 - STOP 状態：シーケンスプログラムの演算を停止している状態で、出力 (Y) は全点 OFF になります。
 - CPU ユニットのリセットを行うと、リセット後は RUN/STOP スイッチ（ベーシックモデル QCPU の場合は RUN/
STOP/RESET スイッチ）で設定されている状態になります。
-

6.6.2 リモート PAUSE

(1) リモート PAUSE とは

CPU ユニットの RUN/STOP スイッチ（ベーシックモデル QCPU の場合は RUN/STOP/RESET スイッチ）を RUN の位置にしたままで、外部から CPU ユニットの PAUSE 状態にする操作です。PAUSE 状態とは、全出力 (Y) の ON/OFF 状態を保持したまま CPU ユニットの演算を停止している状態です。

(2) リモート PAUSE の用途

プロセス制御などで、CPU ユニットの RUN 状態のとき ON していた出力 (Y) を、STOP 状態にしても ON のまま保持したい場合に使用できます。

(3) リモート PAUSE の実行方法

リモート PAUSE の実行方法には、下記の 2 種類があります。

- PAUSE 接点による方法
- GX Developer, MC プロトコルを使用した外部機器による方法

(a) PAUSE 接点による方法

PC パラメータの PC システム設定で PAUSE 接点を設定します。設定可能なデバイス範囲は、CPU ユニットのタイプにより異なります。

- ベーシックモデル QCPU : 入力 X0 ~ 7FF
- ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU : 入力 X0 ~ 1FFF

- 1) PAUSE 接点と PAUSE 許可コイル (SM206) が共に ON したスキンの END 処理実行時に、PAUSE 接点 (SM204) が ON します。PAUSE 接点が ON した次のスキンを END 命令まで実行すると、PAUSE 状態になり演算を停止します。
- 2) PAUSE 接点を OFF にするか、SM206 を OFF にすると PAUSE 状態が解除され、再度ステップ 0 からシーケンスプログラムの演算を行います。

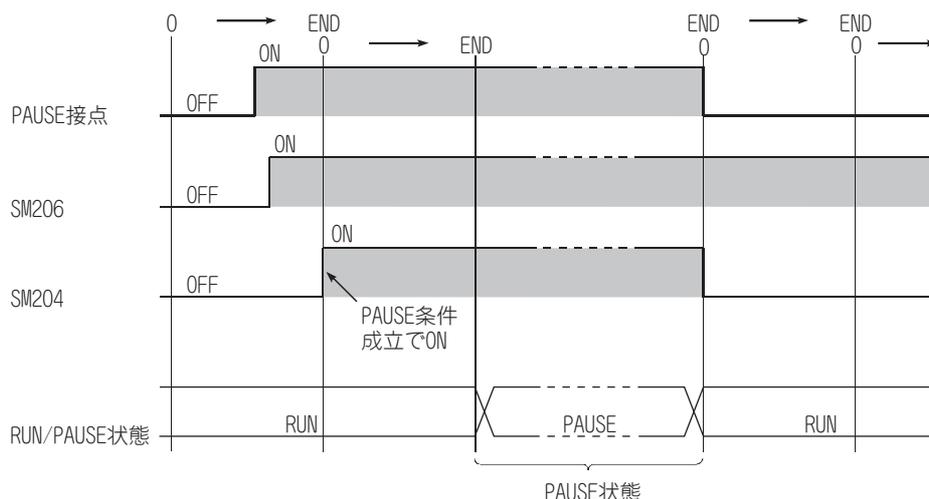


図 6.16 PAUSE 接点によるリモート PAUSE

Point

PAUSE 接点を設定する場合は、RUN 接点も設定してください。
PAUSE 接点のみの設定はできません。

(b) GX Developer, MC プロトコルを使用した外部機器による方法

GX Developer の [オンライン] → [リモート操作] で行います。

MC プロトコルを使用した外部機器からの場合、MC プロトコルのコマンドで行います。

📖 MELSEC コミュニケーションプロトコルリファレンスマニュアル

- 1) リモート PAUSE 指令が入ったスキャンの END 処理実行時に、PAUSE 接点 (SM204) が ON します。PAUSE 接点が ON した次のスキャンを END 処理まで実行すると PAUSE 状態になり、演算を停止します。
- 2) リモート RUN 指令が入ると、再度ステップ 0 からシーケンスプログラムの演算を行います。

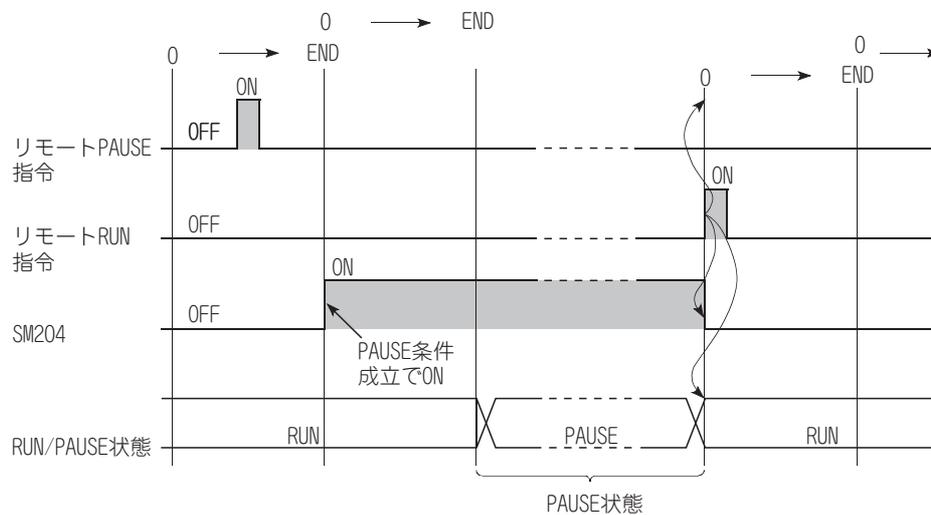


図 6.17 GX Developer, 外部機器によるリモート PAUSE

(4) 注意事項

(a) 強制的に ON または OFF 状態にしておく場合

PAUSE 状態で強制的に ON または OFF 状態にしておく場合は、PAUSE 接点 (SM204) でインタロックをとってください。

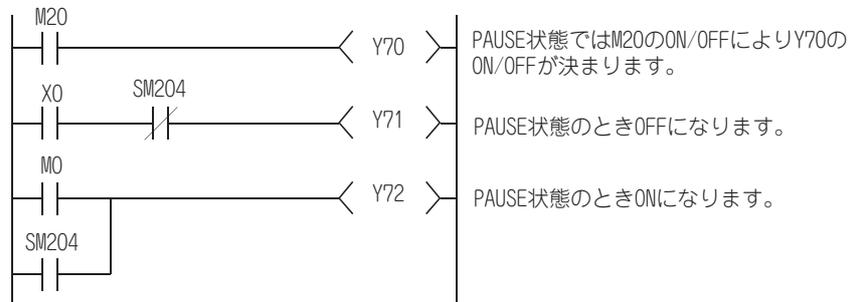


図 6.18 PAUSE 状態で出力 (Y) を強制的に ON または OFF 状態にするためのプログラム例

(b) 二重化 CPU 使用時

1) リモート操作画面にて両系指定を行っていない場合

接続先指定などで指定している系に対してのみ行います。

2) リモート操作画面にて両系指定を行っている場合

制御系および待機系の両方に行います。

ただし、両系指定によるリモート操作では、二重化 CPU がバックアップモードの場合のみ可能です。

6.6.3 リモート RESET (リモートリセット)

(1) リモート RESET とは

CPU ユニットが STOP 状態のときに、外部からの操作により CPU ユニットをリセットする操作です。
また、CPU ユニットの RUN/STOP スイッチ（ベーシックモデル QCPU の場合は RUN/STOP/RESET スイッチ）が RUN の位置にあっても、自己診断機能で検出できるエラーの発生により CPU ユニットが停止しているときはリセット可能です。

(2) リモート RESET の用途

CPU ユニットが手の届かない場所にある状態でエラーが発生したときに、遠隔操作で CPU ユニットをリセットできます。

(3) リモート RESET の実行方法

GX Developer、MC プロトコルを使用した外部機器などからの操作でのみ行うことが可能です。

- GX Developer の [オンライン] → [リモート操作] で行います。
- MC プロトコルを使用した外部機器の場合、MC プロトコルのコマンドで行います。

👉 MELSEC コミュニケーションプロトコルリファレンスマニュアル

Point

リモート RESET を行うには、あらかじめ PC パラメータの PC システム設定でリモートリセットを“許可する”に設定し、CPU ユニットに PC パラメータを書き込んでおきます。
設定していない場合は、リモート RESET ができません。

チェックをすると
リモートリセットを
許可します。



図 6.19 リモート RESET に必要な設定

(4) 注意事項

(a) RUN 状態時のリモート RESET

CPU ユニットが RUN 状態のときには、リモート RESET によるリセットはできません。
リモート STOP 操作などで CPU ユニットの状態を STOP 状態にしてからリモート RESET を行ってください。

(b) リセット処理完了後の状態

リモート RESET を行った CPU ユニットでリセット処理が完了すると、CPU ユニットは RUN/STOP スイッチ（ベーシックモデル QCPU の場合は RUN/STOP/RESET スイッチ）で設定されている運転状態になります。

スイッチが STOP の位置にある場合は STOP 状態になり、RUN の位置にある場合は RUN 状態になります。

Point

- CPU ユニットがエラーにより停止しているときにリモート RESET を行うと、リセット処理が完了で CPU ユニットは RUN/STOP スイッチ（ベーシックモデル QCPU の場合は RUN/STOP/RESET スイッチ）で設定されている運転状態になりますので注意してください。
- GX Developer でリモート RESET を行っても CPU ユニットの状態が変わらない場合は、PC パラメータの PC システム設定でリモートリセットが“許可する”になっているかを確認してください。
設定されていない場合、GX Developer のリモート処理が完了しても、CPU ユニットのリセット処理は行われません。

(c) ノイズにより異常が発生した場合

ノイズにより CPU ユニットに異常が発生した場合は、リモート RESET によるリセットができないことがありますので注意してください。

リモート RESET でリセットができない場合は、RUN/STOP スイッチ（ベーシックモデル QCPU の場合は RUN/STOP/RESET スイッチ）でリセットを行うか、CPU ユニットの電源の再立上げを行ってください。

(d) 二重化 CPU 使用時

バックアップモード時のリモートリセットは制御系に対して行い、リモートリセットを行うと両系がリセットされます。

セパレートモードおよびデバッグモード時は、GX Developer の接続先指定などで指定した系にのみリモートリセットを行います。

1) バックアップモード時のリモートリセット対象

制御系に対して行ってください。(制御系に対してリモートリセットを行うと、両系がリセットされます。) 待機系に対してリモートリセットを行うとエラー (エラーコード: 4240H) になります。

2) バックアップモード時に制御系と待機系の RUN/STOP 状態が異なる場合

制御系が STOP の状態でリモートリセットを行う場合は、待機系も STOP 状態にしてください。

制御系が STOP、待機系が RUN の状態でリモートリセットを行うと系切替えが発生します。

また、制御系が RUN 状態、待機系が STOP 状態でリモートリセットを行うと、待機系のみリセットされます。

3) バックアップモード時に待機系で WDT エラーが発生している場合

制御系に対してリモートリセットを行っても待機系はリセットされません。

この場合、下記に示す経路 (トラッキングケーブルを経由しない通信経路) でリモートリセットを行ってください。

- ・パソコン (GX Developer) を、待機系の二重化 CPU に直接接続する
- ・待機系のユニット経由でリモートリセットを行う (MC プロトコルによるリモートリセットなど)

4) バックアップモード時に別経路からリモート操作をしている場合

リモート操作を行っている CPU ユニットに、他の GX Developer からリモート操作はできません。

図 6.20 のように、制御系および待機系に対して別経路からリモート操作をしている場合、制御系に対してリモートリセットを行っても待機系はリセットしない場合があります。

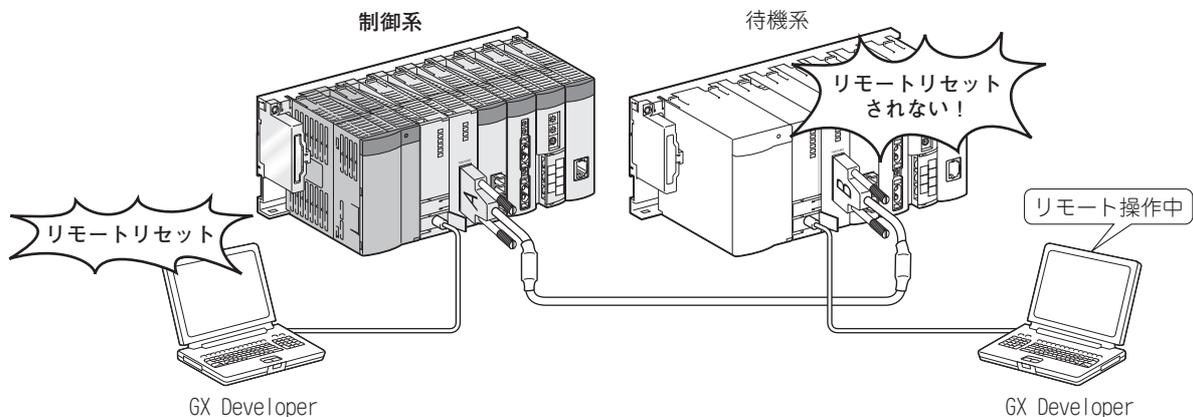


図 6.20 別経路からリモート操作をしている場合

待機系に対してリモート操作を行っている場合は、待機系へのリモート操作を解除してから、制御系に対してリモートリセットを行ってください。

6.6.4 リモートラッチクリア

(1) リモートラッチクリアとは

CPU ユニットが STOP 状態のときに、ラッチされているデバイスデータを GX Developer などによりリセットする操作です。

(2) リモートラッチクリアの用途

下記に示す場合、リモート RUN/STOP と組み合わせて行うと便利です。

- CPU ユニットが手の届かない場所にある場合
- 制御盤内の CPU ユニットの外部からラッチクリアする場合

(3) リモートラッチクリアの実行方法

GX Developer、MC プロトコルを使用した外部機器などからの操作でのみ可能です。

- GX Developer の操作は、[オンライン] → [リモート操作] で行います。
- MC プロトコルを使用した外部機器での制御は、MC プロトコルのコマンドで行います。

 MELSEC コミュニケーションプロトコルリファレンスマニュアル

リモートラッチクリアは下記の手順で行います。

- 1) リモート STOP で CPU ユニットの STOP 状態にします。
- 2) リモートラッチクリア操作で CPU ユニットのラッチクリアします。
- 3) リモートラッチクリア完了後、RUN 状態にする場合はリモート RUN を行います。

(4) 注意事項

(a) RUN 状態でのラッチクリア

CPU ユニットが RUN 状態のときには、ラッチクリアはできません。

(b) ラッチクリア操作が有効 / 無効になる範囲

PC パラメータのデバイス設定で設定するデバイスのラッチ範囲には、ラッチクリア操作が有効になる範囲と無効になる範囲があります。

リモートラッチクリア操作では、“ラッチ (1)” の範囲に設定されているデバイス範囲のみリセットされます。

ラッチクリア無効に設定されているデバイスのリセット方法については、3.7 節 (2) (c) を参照してください。

(c) リモートラッチクリア実行時にリセットされるデバイス

リモートラッチクリア実行時は、ラッチされていないデバイスもリセットされます。

(d) 二重化 CPU 使用時

1) リモート操作画面にて両系指定を行っていない場合

接続先指定などで指定している系に対してのみ行います。

2) リモート操作画面にて両系指定を行っている場合

制御系および待機系の両方に行います。

ただし、両系指定によるリモート操作では、二重化 CPU がバックアップモードの場合のみ可能です。

6.6.5 リモート操作と CPU ユニットの RUN/STOP 状態との関係

(1) リモート操作と CPU ユニットの RUN/STOP 状態との関係

リモート操作と CPU ユニットの RUN/STOP 状態の組合せによる CPU ユニットの動作状態は表 6.9 のようになります。

表 6.9 RUN/STOP 状態とリモート操作の関係

RUN/STOP 状態	リモート操作				
	RUN * 1	STOP	PAUSE * 2	RESET * 3	ラッチクリア
RUN	RUN	STOP	PAUSE	操作不可 * 4	操作不可 * 4
STOP	STOP	STOP	STOP	RESET * 5	ラッチクリア

* 1：RUN 接点で行う場合は、PC パラメータの PC システム設定で“RUN-PAUSE 接点”の設定が必要です。

* 2：PAUSE 接点で行う場合は、PC パラメータの PC システム設定で“RUN-PAUSE 接点”の設定が必要です。さらに、リモート PAUSE 許可コイル (SM206) を ON にしておく必要があります。

* 3：PC パラメータの PC システム設定で“リモートリセット”を“許可する”の設定が必要です。

* 4：CPU ユニットのリモート STOP 操作により STOP 状態にした場合は、リモート RESET またはリモートラッチクリアが可能です。

* 5：CPU ユニットのエラーで停止している場合も含まれます。

(2) 同一の GX Developer からのリモート操作

同一の GX Developer でリモート操作を行う場合は、あとから実行したリモート操作の状態になります。

(3) 複数の GX Developer からのリモート操作

1 つの GX Developer でリモート操作を行っている CPU ユニットに対して、他局経由で他の GX Developer からリモート操作を行うことはできません。

また、他の GX Developer からリモート操作を行う場合は、リモート操作を行っている GX Developer でリモート RUN を行い、リモート操作を解除してください。

例えば、1 つの GX Developer でリモート PAUSE を行っている場合、他の GX Developer からリモート STOP / リモート RUN を行っても、PAUSE 状態のままになります。

リモート PAUSE を行っている GX Developer でリモート RUN を行い、リモート操作を解除すると、他の GX Developer からリモート操作を行うことができます。

6.7 Q シリーズ対応ユニットの入力応答時間選択 (I/O 応答時間)

(1) 入力応答時間選択とは

Q シリーズ対応ユニットの入力応答時間を、ユニット単位で変更する機能です。
入力応答時間の変更が可能なユニットと設定可能な時間は、表 6.10 のとおりです。

表 6.10 入力応答時間を変更可能なユニット

ファイル種別	種別	設定可能な時間
入力ユニット	入力	1ms, 5ms, 10ms, 20ms, 70ms (デフォルト: 10ms)
入出力混合ユニット	入出力混合	
高速入力ユニット	高速入力	0.1ms, 0.2ms, 0.4ms, 0.6ms, 1ms
割込みユニット	割込み	(デフォルト: 0.2ms)

上記に示した Q シリーズ対応のユニットは、設定された入力応答時間で外部入力の取込みを行います。

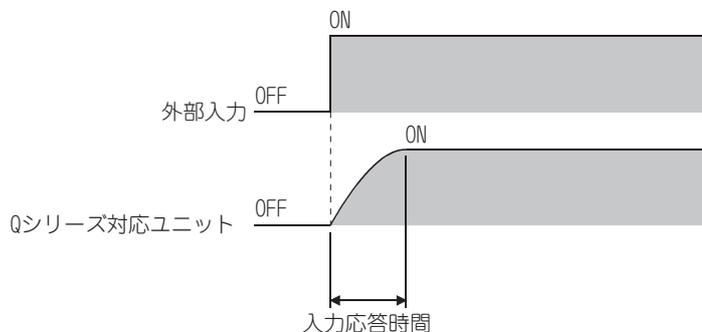


図 6.21 入力応答時間

(2) 入力応答時間の設定

PC パラメータの I/O 割付設定で行います。

- 1) I/O 割付設定を行います。
- 2) **詳細設定** ボタンを選択します。
- 3) 入力応答時間を設定します。

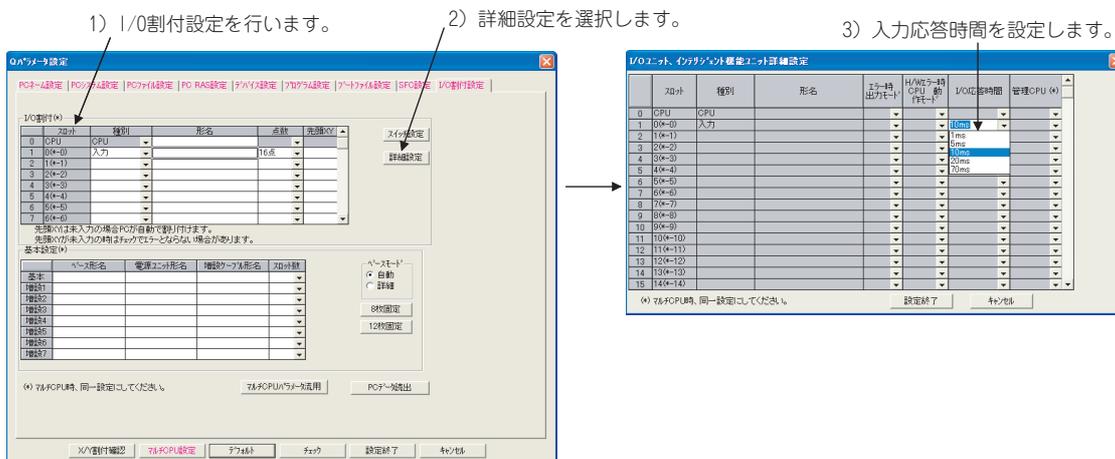


図 6.22 入力応答時間の設定

(3) 注意事項

(a) 設定可能な GX Developer のバージョンに制約があるユニットについて

下記のユニットの入力応答時間を変更する場合には、以下の GX Developer のバージョンを使用してください。

- 高速入力ユニット : GX Developer Version 5 (SW5D5C-GPPW) 以降
- 割込みユニット : GX Developer Version 6 (SW6D5C-GPPW) 以降

上記のバージョンより前の GX Developer を使用した場合は、入力応答時間のデフォルト値で動作します。

(b) 入力応答時間を短くした場合

入力応答時間を短く設定すると、ノイズなどに弱くなります。
入力応答時間は、使用環境を考慮して設定してください。

(c) AnS/A シリーズ対応のユニット使用時

入力応答時間は変更できません。

AnS/A シリーズ対応の入力ユニットまたは割込みユニットのスロットに、入力応答時間の設定を行った場合は無処理になります。

(d) 設定が有効になるタイミング

入力応答時間の設定は次の場合に有効になります。

- CPU ユニットの電源 OFF → ON 時
- リセット解除時

6.8 エラー時出力モード設定

(1) エラー時出力モード設定とは

Q シリーズ対応の出力ユニット、入出力混合ユニット、インテリジェント機能ユニット、割込みユニットへの出力を、CPU ユニットが停止エラー時にクリアするか、保持するかの設定です。

(2) エラー時出力モード設定

PC パラメータの I/O 割付設定で行います。

- 1) I/O 割付設定を行います。
- 2) **詳細設定** ボタンを選択します。
- 3) エラー時出力モードを設定するスロットで、“クリア”か“保持”を選択します。
(デフォルトは“クリア”)

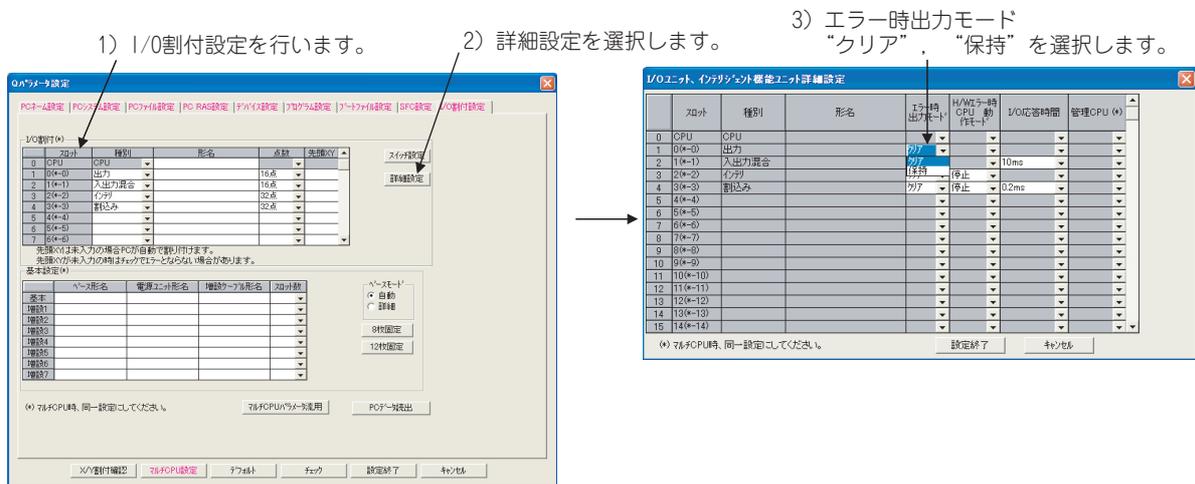


図 6.23 エラー時出力モード設定

(3) 注意事項

エラー時出力モード設定は、次の場合に有効になります。

- CPU ユニットの電源 OFF → ON 時
- リセット解除時

エラー時出力モード設定を変更後、上記動作を行わない場合は“PARAMETER ERROR”（エラーコード：3000）になります。

6.9 ハードウェアエラー時の CPU 動作モード設定

(1) ハードウェアエラー時の CPU 動作モード設定とは

インテリジェント機能ユニット、割込みユニットでハードウェアエラーが発生したときに、CPU ユニットの演算を停止させるか、続行させるかの設定です。

(2) ハードウェアエラー時の CPU 動作モード設定

PC パラメータの I/O 割付設定で行います。

- 1) I/O 割付設定を行います。
- 2) **詳細設定** ボタンを選択します。
- 3) ハードウェアエラー時の CPU 動作モードを設定するスロットのハードウェアエラー時 CPU 動作モードを設定します。(デフォルトは“停止”)

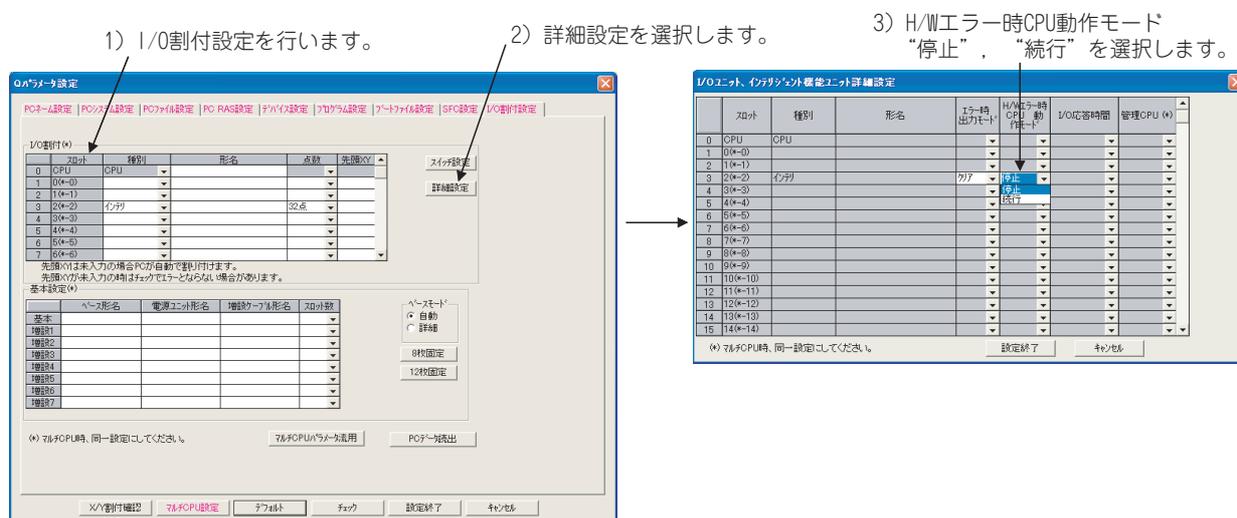


図 6.24 ハードウェアエラー時の CPU 動作モード設定

(3) 注意事項

ハードウェアエラー時の CPU 動作モード設定は、次の場合に有効になります。

- CPU ユニットの電源 OFF → ON 時
- リセット解除時

6.10 インテリジェント機能ユニットのスイッチ設定

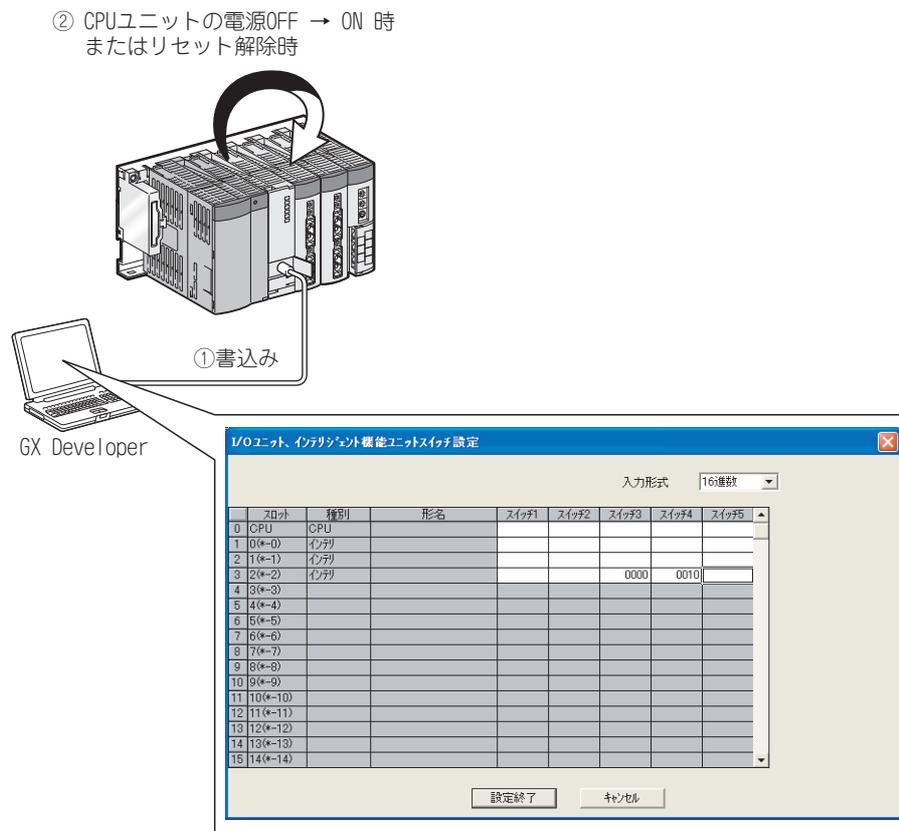
(1) インテリジェント機能ユニットのスイッチ設定とは

Q シリーズ対応のインテリジェント機能ユニット、割込みユニットのスイッチの内容を GX Developer で設定します。

(2) スイッチ設定が書き込まれるタイミング

下記のタイミングで CPU ユニットから各インテリジェント機能ユニット、割込みユニットに書き込まれます。

- CPU ユニットの電源 OFF → ON 時
- リセット解除時



インテリジェント機能ユニットのスイッチ設定

図 6.25 スイッチ設定がユニットに書き込まれるタイミング

(3) インテリジェント機能ユニット，割込みユニットのスイッチ設定

PCパラメータのI/O割付設定で行います。

- 1) I/O割付設定を行います。
- 2) **スイッチ設定** ボタンを選択します。
- 3) ユニットのスイッチ設定を行います。
 - 1) I/O割付設定を行います。
 - 2) スイッチ設定を選択します。

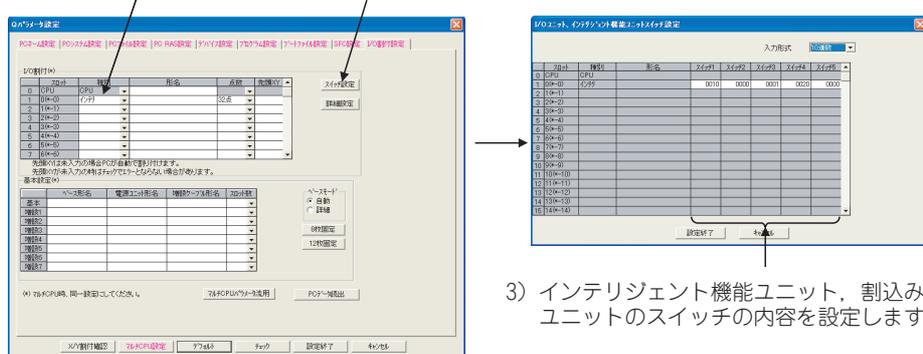


図 6.26 スイッチ設定

(4) 注意事項

(a) AnS/A シリーズ対応のユニット使用時 **注 6.2**

AnS/A シリーズ対応の特殊機能ユニットに対してスイッチ設定はできません。
数値を入力しても、その設定は無視されます。

(b) 各ユニットのスイッチ設定内容

使用するインテリジェント機能ユニット，割込みユニットのマニュアルを参照してください。

(c) GX Developer のバージョンによる設定の違い

割込みユニットのスイッチ設定の種別は，下記に設定してください。

- GX Developer Version 6 (SW6D5C-GPPW) 以降の場合：“割込み”
- GX Developer Version 5 (SW5D5C-GPPW) 以前の場合：“インテリ”

(d) 設定が有効になるタイミング

インテリジェント機能ユニット，割込みユニットの設定は次の場合に有効になります。

- CPU ユニットの電源 OFF → ON 時
- リセット解除時



注 6.2

Basic

Process

Redundant

ベーシックモデル QCPU，プロセス CPU，二重化 CPU では，AnS/A シリーズ対応の入力ユニットまたは割込みユニットは使用できません。

6.11 モニタ機能

(1) モニタ機能とは

GX Developer から CPU ユニットのプログラム、デバイス、インテリジェント機能ユニットの状態を読み出す機能です。

表 6.11 モニタ機能一覧と使用可否

ファイル種別	CPU ユニットごとの使用可否				参 照
	ベーシック モデル QCPU	ハイパフォーマンス モデル QCPU	プロセス CPU	二重化 CPU	
回路モニタ	○	○	○	○	GX Developer Version 8 オペレーティング マニュアル
デバイス/バッファメモリ一括モニタ	○	○	○	○	
デバイス登録モニタ	○	○	○	○	
デバイステスト	○	○	○	○	
回路登録モニタ	○	○	○	○	
モニタ条件の設定	×	○	○	○	6.11.1 項
ローカルデバイスのモニタ・テスト	×	○	○	○	6.11.2 項
外部入出力の強制 ON/OFF	△* 1	○	○	○	6.11.3 項

○：使用可，△：一部制約あり，×：使用不可

* 1：ベーシックモデル QCPU では、デバイステストによる強制 ON/OFF のみ可能です。

(2) モニタ要求の処理タイミングと表示データ

CPU ユニットは、GX Developer からのモニタ要求の処理を END 処理で行っています。
このため GX Developer では、CPU ユニットの END 処理時のデータを表示します。

(3) モニタ条件を設定したモニタ

デバッグを行うときに GX Developer でモニタ条件を設定すると、CPU ユニットの演算状態を指定した条件でモニタできます。

また、モニタ停止条件を設定すると、指定した条件でモニタ状態を保持できます。

(4) ローカルデバイスのモニタ

複数のプログラムを実行してローカルデバイスを使用している場合、各プログラムのローカルデバイスのデータをモニタすることもできます。

6.11.1 モニタ条件の設定 注 6.3

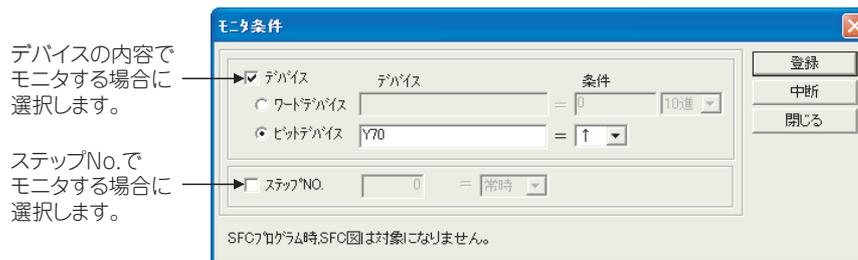
モニタ条件の設定は、指定した条件でモニタを行う場合に設定します。

(1) 回路モニタ時のモニタの実行条件設定

GX Developer でモニタモードにします。

[オンライン] → [モニタ] → [モニタ条件設定] で下記ウィンドウを表示します。

Y70 の立上りでモニタする場合は次のように設定します。



デバイスの内容で
モニタする場合に
選択します。

ステップNo.で
モニタする場合に
選択します。



条件が成立すると
モニタを行います。
(条件成立時の演算
状態を表示します。)

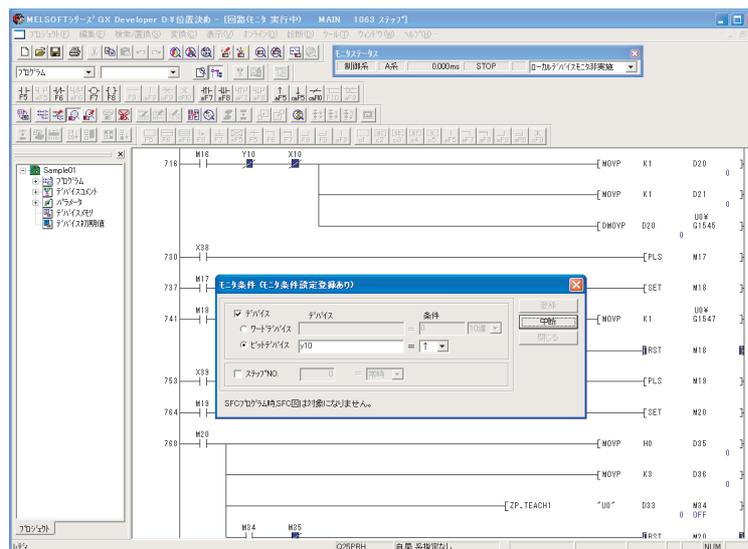


図 6.27 モニタ条件設定画面

(a) ステップ No. のみ指定した場合

モニタデータの収集タイミングは、指定ステップの実行直前の状態が指定状態となったときです。指定できる実行条件を下記に示します。

- 指定したステップの演算が非実行状態から実行状態になったとき：< ↑ >
- 指定したステップの演算が実行状態から非実行状態になったとき：< ↓ >
- 指定したステップの演算が実行中のみ常時：< ON >
- 指定したステップの演算が非実行中のみ常時：< OFF >
- 指定したステップの演算の状態に関係なく常時：< 常時 >



注 6.3

Basic

ベーシックモデル QCPU では、モニタ条件を設定したモニタはできません。

- AND/OR ブロックの途中のステップをモニタ条件として指定した場合、モニタデータの収集タイミングは、ブロック内の LD 命令から指定ステップ実行直前の状態が指定状態となったときです。このため指定したステップの回路により、モニタのタイミングが異なります。実行条件としてステップ 2 が ON でモニタする場合（ステップ No. [2] = < ON >）は次のようになります。

- ステップ 2 が AND 命令で接続されているとき
図 6.28 の X0 と X1 が共に ON の場合、モニタの実行条件が成立します。

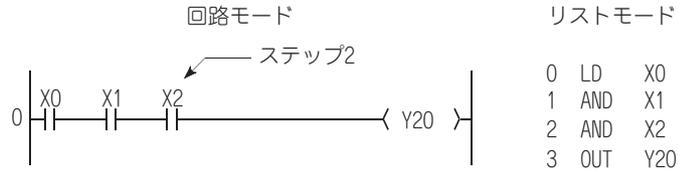


図 6.28 ステップ 2 が AND 命令で接続されている場合

- ステップ 2 が AND/OR ブロックの途中に接続されているとき
図 6.29 の X1 が ON で実行条件が成立します。（X0 の ON/OFF は、実行条件の成立には無関係です。）

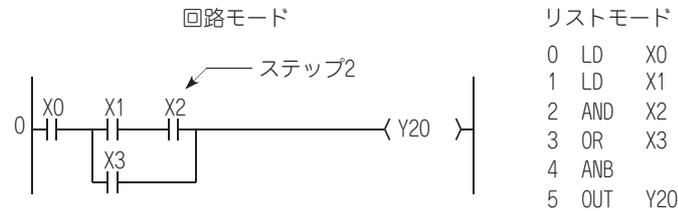


図 6.29 ステップ 2 が AND/OR ブロックの途中に接続されている場合

- 詳細条件としてステップ No. にステップ 0 以外の回路ブロック先頭を指定した場合は、実行直前の命令の実行状態が指定状態になったときにモニタデータの収集を行います。下記回路で（ステップ No. [2] = < ON >）を指定した場合は、OUT Y10 が ON でモニタデータの収集を行います。

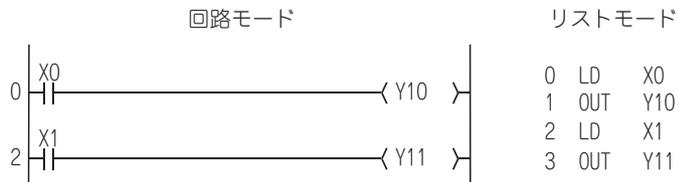


図 6.30 ステップ No. にステップ 0 以外の回路ブロック先頭を指定した場合

- ステップ No. に “0” を指定した場所は、条件を必ず “常時” に設定してください。

(b) デバイスのみ指定した場合

ワードデバイスまたはビットデバイスを指定できます。

1) ワードデバイスを指定した場合

指定ワードデバイスの現在値が指定値となったときに、モニタデータの収集を開始します。
現在値は 10 進数または 16 進数で指定します。

2) ビットデバイスを指定した場合

指定ビットデバイスの実行状態が指定状態となったときに、モニタデータの収集を開始します。
実行条件には立上り時または立下り時を指定できます。

(c) ステップ No. およびデバイスを指定した場合

指定ステップの実行直前の状態および指定ビットデバイス（ワードデバイス）の状態（現在値）が指定状態となったときに、モニタデータの収集を開始します。

Point

実行条件としてステップ 100 の立上りと、D1 = 5（ステップ No. [100] = < ↑ >、ワードデバイス [D1] = [K5]）を指定した場合は、ステップ 100 の立上りと D1 = 5 のときモニタの実行条件が成立します。



図 6.31 ステップ 100 の立上りと D1 = 5 を指定した場合

また、GX Developer のモニタ間隔は、GX Developer の処理速度に依存します。
モニタの実行条件が、GX Developer のモニタ間隔より短い間隔で成立した場合、GX Developer のモニタタイミング時にモニタの実行条件が成立しているとモニタを行います。

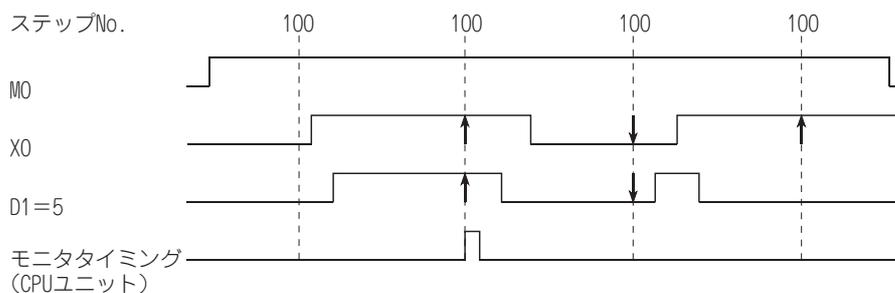


図 6.32 GX Developer のモニタタイミング

(2) モニタの停止条件の設定

設定は [オンライン] → [モニタ] → [モニタ停止条件設定] でモニタ停止条件画面を表示して行います。
Y71 の立上りでモニタを停止する場合は図 6.33 のように設定します。

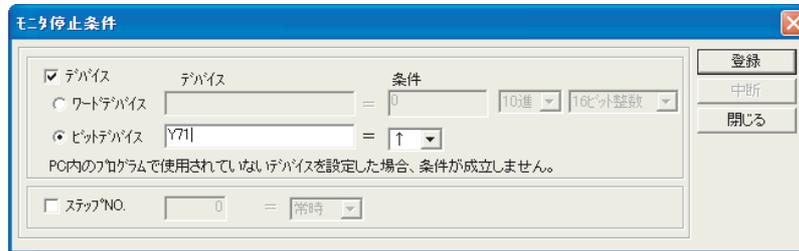


図 6.33 モニタ停止条件画面

(a) デバイスを指定した場合

ワードデバイスまたはビットデバイスを指定できます。

1) ワードデバイスを指定した場合

モニタ停止のタイミングは、指定ワードデバイスの現在値が指定値となったときになります。
現在値の指定は、10 進数 / 16 進数と 16 ビット整数 / 32 ビット整数 / 実数が選択できます。

2) ビットデバイスを指定した場合

モニタ停止のタイミングは、指定ビットデバイスの実行状態が指定状態となったときです。
実行条件には立上り時または立下り時を指定できます。

(b) “ステップ No.” を指定した場合

モニタ停止のタイミングは、モニタ条件で指定したステップ No. の実行状態が指定状態となったときです。
指定できる実行条件を下記に示します。

- 指定したステップの演算が非実行状態から実行状態になったとき：< ↑ >
- 指定したステップの演算が実行状態から非実行状態になったとき：< ↓ >
- 指定したステップの演算が実行中のみ常時：< ON >
- 指定したステップの演算が非実行中のみ常時：< OFF >
- 指定したステップの演算が状態に関係なく常時：< 常時 >

“ステップ No.” を指定しない場合、モニタ停止のタイミングは、CPU ユニットの END 処理後になります。

(3) 注意事項

(a) モニタを行う CPU ユニットのファイル

モニタ条件を設定してモニタする場合、GX Developer 上で表示しているファイルをモニタします。
GX Developer の [オンライン] → [PC 読出] を実行し、モニタを行う CPU ユニットのファイル名と GX Developer 上のファイル名を一致させてください。

(b) ファイルレジスタの設定がない場合

ファイルレジスタの設定がない場合に、ファイルレジスタをモニタすると 0 が表示されます。

(c) モニタ時の CPU ユニットと GX Developer のデバイス割付け

モニタ時は、CPU ユニットと GX Developer のデバイス割付けを一致させて行ってください。

(d) インテリジェント機能ユニットのバッファメモリをモニタする場合

インテリジェント機能ユニットのバッファメモリをモニタする場合は、FROM/TO 命令実行時と同様にスキャンタイムが延びます。

(e) 複数の GX Developer で同時にモニタを実行する場合

複数の GX Developer で同時にモニタを実行する場合、以下の注意事項があります。

- プログラムメモリのフォーマットまたは PC パラメータのブートファイル設定時に、他局用モニタファイル 1 つにつき 1k ステップずつシステムエリアを増やすことにより、高速モニタが可能になります。他局用モニタファイルは、最大 15 局まで設定可能ですが、その分プログラム領域が減少します。
- モニタ条件およびモニタ停止条件を設定した場合は、1 人のみモニタ可能です。

(f) モニタ停止条件が設定可能な状態

モニタ停止条件の設定は、回路モニタでのみ設定できます。

(g) 条件に同一デバイスを指定する場合

モニタ条件およびモニタ停止条件として、同一デバイスを指定する場合は “ON” または “OFF” を指定してください。

(h) モニタ条件でステップ No. を指定した場合

モニタ条件で “ステップ No.” を指定した場合、下記のように指定ステップの命令を実行しないときはモニタ条件が成立しません。

- CJ 命令、SCJ 命令、JMP 命令により設定したステップを飛ばしたとき
- 設定ステップが END 命令で、プログラム中に FEND 命令が存在し、END 命令を実行しないとき

(i) モニタ条件登録中

モニタ条件登録中に CPU ユニットのリセットを行わないでください。

(j) モニタ条件を設定したモニタを実行している場合

モニタ条件を設定したモニタを実行している場合、同一パソコン上の他アプリケーションでは、モニタ実行と同一経路でのオンライン交信機能を使用することができません。他のアプリケーションの例を下記に示します。

- GX Developer
- MX Component を使用したアプリケーション
- MX Sheet

他のアプリケーションのオンライン交信機能を、モニタ実行と同一経路で実行した場合、下記のような現象が発生する可能性があります。

- オンライン機能に対して、CPU ユニットからの応答が返らない。(CPU とのオンライン通信機能がタイムアウトする。)
- オンライン機能に対して、CPU ユニットでエラー (エラーコード：4109) が検出される。
- モニタ条件を設定したモニタにおいて、CPU ユニット内部でモニタ条件が成立しても、モニタ結果が更新されない。

6.11.2 ローカルデバイスのモニタ・テスト 注 6.4

GX Developer でモニタしているプログラムのローカルデバイス（ 9.13.2 項）の内容を確認しながらデバッグすることができます。

(1) ローカルデバイスのモニタ

ローカルデバイスを D0 ～ 99 に設定し、CPU ユニットでプログラム名が “A” , “B” , “C” の 3 本のプログラムを実行している場合の動作を表 6.12 に示します。

（プログラムの実行順序は、A → B → C → (END 処理) → A → B… とします。）

表 6.12 3本のプログラム実行時に表示されるデータ

設 定	モニタするデバイス	
	D0 (ローカルデバイス)	D100 (グローバルデバイス)
ローカルデバイスのモニタ設定時	指定したプログラムの D0 (各プログラム用のローカルデバイス) の値をモニタする。* 1	指定したプログラム実行後の D100 の値をモニタする。
ローカルデバイスのモニタ非設定時	プログラム “C” 実行後の D0 の値をモニタする。	プログラム “C” 実行後の D100 の値をモニタする。

* 1：プログラム設定のファイル使用方法設定の “ローカルデバイス” で “使用しない” に設定している場合、指定したプログラム実行後の D0 の値をモニタします。

注 6.4 **Basic**

ベーシックモデル QCPU では、グローバルデバイスとローカルデバイスの区別がないため、GX Developer でローカルデバイスのモニタ設定を行う必要はありません。

例えば、ローカルデバイスのモニタ設定をし、プログラム名が“B”のプログラムを表示しモニタすると、プログラム“B”のローカルデバイスのモニタができます。

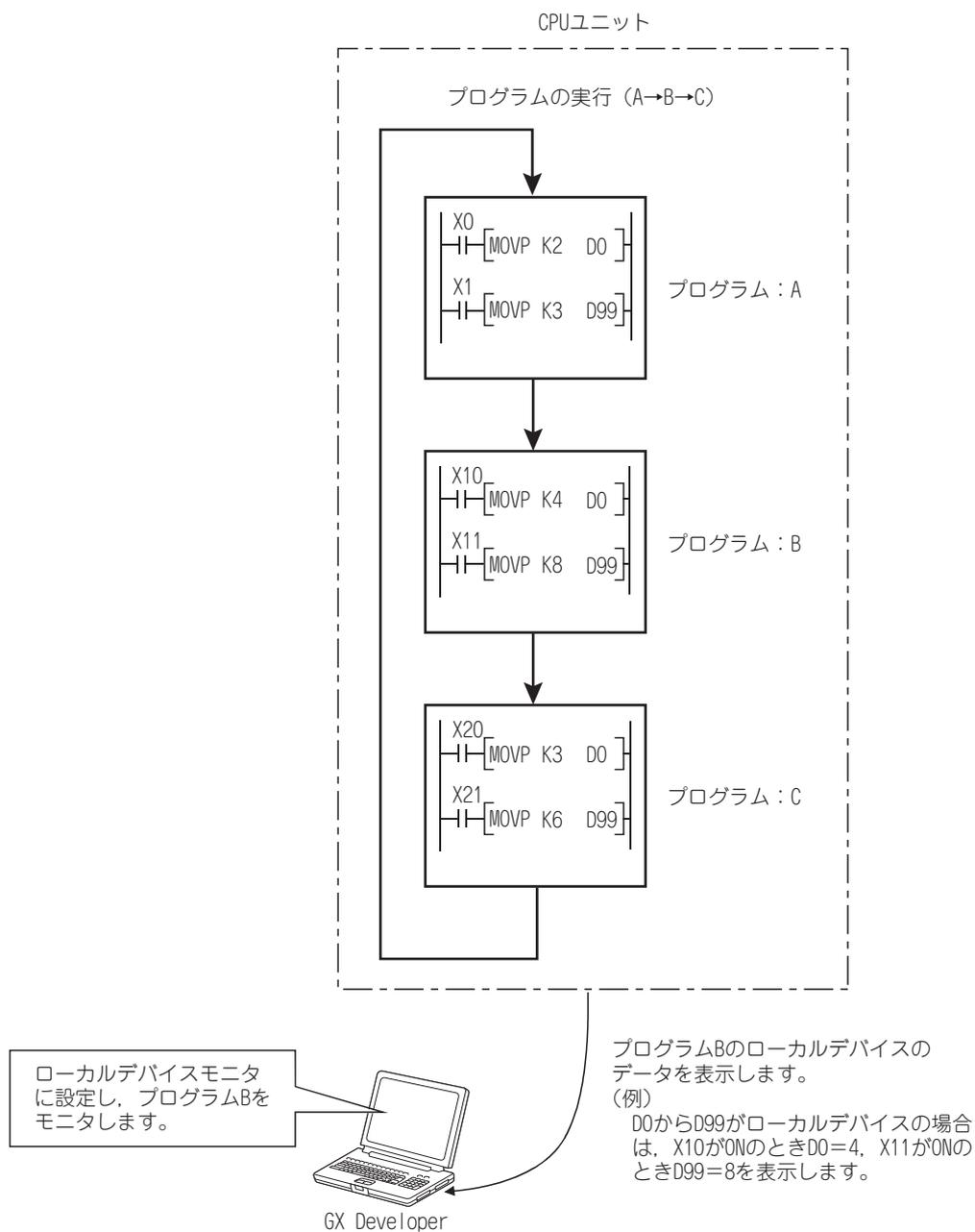


図 6.34 ローカルデバイスのモニタ例

(2) ローカルデバイスのモニタ手順

ローカルデバイスのモニタは、下記手順で行います。

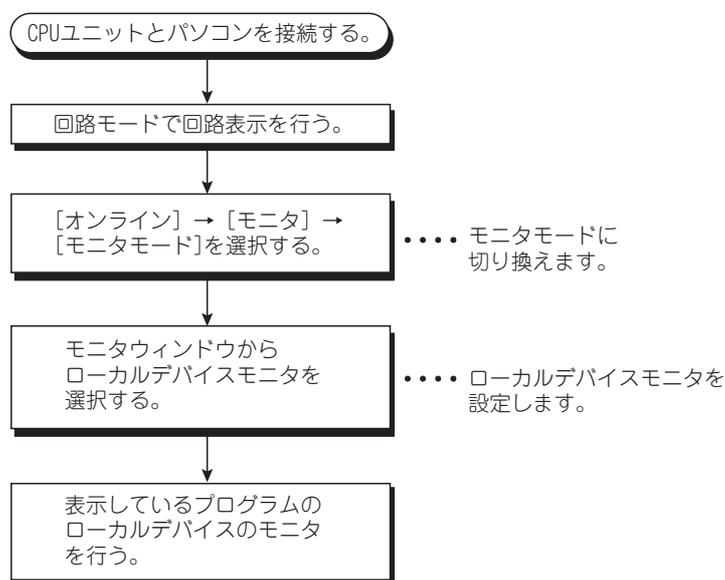


図 6.35 ローカルデバイスのモニタ手順フロー

(3) 注意事項

(a) 1つのGX Developerでモニタ・テスト可能なローカルデバイス

1つのGX Developerでは、1つのプログラムのローカルデバイスのモニタ・テストのみ可能です。複数のプログラムのローカルデバイスを同時にモニタ・テストすることはできません。

(b) モニタ・テスト可能なプログラム数

CPUユニットのRS-232、シリアルコミュニケーションユニットなどに接続した複数のGX Developerから、同時に16プログラムのローカルデバイスのモニタ・テストが可能です。

(c) 待機タイププログラムのローカルデバイスのモニタ

待機タイププログラムのローカルデバイスのモニタを行うと、ローカルデバイスデータの退避と復帰を行いますので、スキャンタイムが延びます。(参照 9.13.2 項)

(d) 定周期実行タイププログラムのローカルデバイスモニタ

定周期実行プログラムをローカルデバイスモニタした場合、デバイスデータを取得せずに0が表示されます。

6.11.3 外部入出力の強制 ON/OFF 注 6.5

GX Developer の [オンライン] → [デバッグ] → [強制入出力登録 / 解除] により、外部入出力を強制 ON/OFF できます。

ON/OFF 登録した情報は、GX Developer の操作により解除します。



図 6.36 強制入出力登録 / 解除画面

(1) 強制 ON/OFF 時の動作

強制 ON/OFF では、強制 ON（強制 ON 登録）、強制 OFF（強制 OFF 登録）、強制 ON/OFF の解除（登録解除）ができます。

強制 ON、強制 OFF、強制 ON/OFF の解除操作時の動作を表 6.13 に示します。

表 6.13 強制 ON/OFF 時の動作

操 作	入力 (X) の動作	出力 (Y) の動作
解除（未操作）時	外部入力によりシーケンスプログラムの演算を行います。	シーケンスプログラムの演算結果を外部へ出力します。* 1
強制 ON 時	強制 ON 状態で、シーケンスプログラムの演算を行います。	シーケンスプログラムの演算結果に関係なく、“ON” を外部へ出力します。* 1
強制 OFF 時	強制 OFF 状態で、シーケンスプログラムの演算を行います。	シーケンスプログラムの演算結果に関係なく、“OFF” を外部へ出力します。* 1

* 1：二重化 CPU のバックアップモードでは、待機系の出力 (Y) の強制 ON/OFF は行いません。

注 6.5

Basic

High performance

ベーシックモデル QCPU では、外部入出力の強制 ON/OFF は使用できません。
 ハイパフォーマンスモデル QCPU で外部入出力の強制 ON/OFF を使用する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。（ 付 2.2）

強制 ON/OFF を行った場合の動作を図 6.37 に示します。

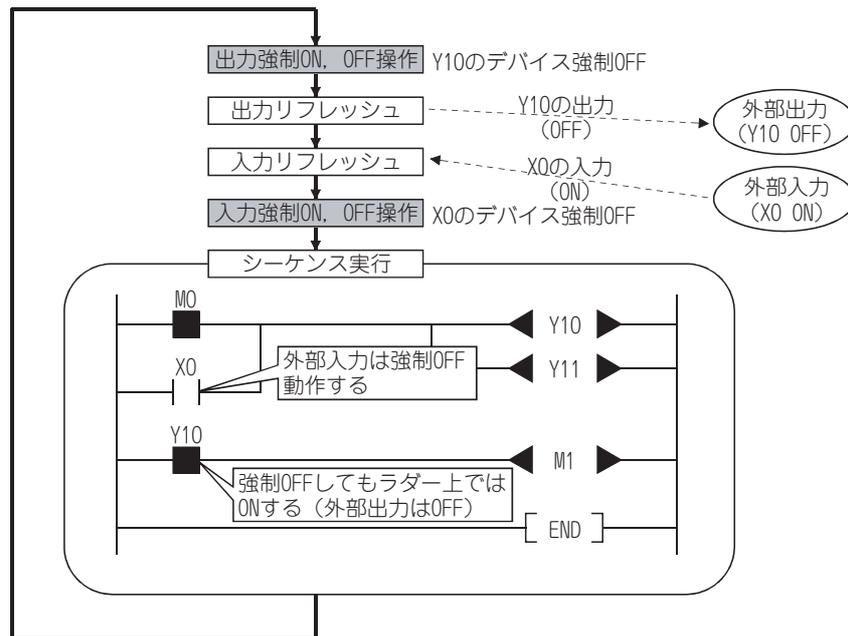


図 6.37 強制 ON/OFF 時の動作

(2) 仕様説明

(a) 強制 ON/OFF が可能な CPU ユニットの状態

CPU ユニットの RUN/STOP 状態に関係なく、強制 ON/OFF できます。
ただし、停止エラー時は入力のみ強制 ON/OFF できます。
出力はデバイス Y へのみ出力します。

Point

二重化 CPU で系切替えが発生した場合は、新制御系は系切替え発生前の制御系に登録した強制 ON/OFF 登録内容で、強制 ON/OFF を継続します。

(b) 登録可能なデバイス

CPU ユニットの入出力デバイス点数分を登録できます。

(c) 強制 ON/OFF の対象となる入出力

下記の入出力が対象となります。

- ベースユニット上に装着されているユニットの入力 (X)、出力 (Y)
- CC-Link IE コントローラネットワークユニットまたは、MELSECNET/H ユニットの LX/LY のリフレッシュ先の CPU ユニットの入出力 (X/Y)
- CC-Link の RX/RV のリフレッシュ先の CPU ユニットの入出力 (X/Y)

上記リフレッシュ範囲外のデバイス（例：空きスロット）を強制 ON/OFF した場合は、CPU ユニットのデバイスメモリのみ ON/OFF し、外部へは出力しません。

マルチ CPU システムでは、管理ユニットの入出力が強制 ON/OFF の対象になります。
 管理外ユニットの入出力を強制 ON/OFF 登録しても、他号機 CPU の入出力デバイスおよび他号機 CPU が管理するユニットの入出力は ON/OFF しません。(自号機 CPU の管理する入出力デバイスが ON/OFF します。)

(d) ON/OFF 登録情報の解除

GX Developer からの操作により、ON/OFF 登録した情報を解除できます。
 ON/OFF 登録した情報が解除されると、強制 ON/OFF したデバイスは下記のようにになります。

表 6.14 ON/OFF 登録情報解除後のデバイスの状態

強制 ON/OFF デバイス		シーケンスプログラムで ON/OFF している	シーケンスプログラムで ON/OFF していない
入 力	ベースユニット上に装着されているユニットからの入力	ユニットから取り込んだ ON/OFF 状態になる。	
	CC-Link IE コントローラネットワーク または MELSECNET/H の LX のリフレッシュ先の CPU ユニットの入力	CC-Link IE コントローラネットワーク、MELSECNET/H からリフレッシュされた ON/OFF 状態になる。	
	CC-Link の RX のリフレッシュ先の CPU ユニットの入力	CC-Link からリフレッシュされた ON/OFF 状態になる。	
	上記以外の入力 (リフレッシュ範囲外)	シーケンスプログラムでの演算結果が反映される。	強制 ON/OFF された状態を保持する。
出 力	ベースユニット上に装着されているユニットからの出力	シーケンスプログラムでの演算結果を出力する。	登録していた ON/OFF 状態を保持する。
	CC-Link IE コントローラネットワークまたは MELSECNET/H の LY のリフレッシュ先の CPU ユニットの出力		
	CC-Link の RY のリフレッシュ先の CPU ユニットの出力		
	上記以外の出力 (リフレッシュ範囲外)	CPU ユニットの出力は、シーケンスプログラムでの演算結果になる。(外部へは出力されない。)	強制 ON/OFF された状態を保持する。

強制 ON/OFF 設定は、次の操作でクリアされます。

- CPU ユニットの電源 OFF → ON
- RUN/STOP/RESET スイッチによるリセット
- リモートリセット操作によるリセット

(e) 外部入出力の強制 ON/OFF タイミング

外部入出力の強制 ON/OFF タイミングを表 6.15 に示します。

表 6.15 強制 ON/OFF のタイミング

リフレッシュエリア	入 力	出 力
ベースユニット上に装着されているユニットの入出力	<ul style="list-style-type: none"> • END 処理（入力リフレッシュ時） • COM 命令実行時（入力リフレッシュ時） • ダイレクトアクセス入力(DX)を使用した命令実行時 (LD,LDI,AND,ANI,OR,ORI,LDP,LDF,ANDP,ANDF,ORP,ORF) • RFS 命令実行時, MTR 命令実行時 • システム割込みで使用する命令実行時 (UDCNT1,UDCNT2,SPD) 	<ul style="list-style-type: none"> • END 処理（出力リフレッシュ時） • COM 命令実行時（出力リフレッシュ時） • ダイレクトアクセス出力(DY)を使用した命令実行時 (OUT,SET,DELTA,RST,PLS,PLF,FF,MC,SFT) • RFS 命令実行時, MTR 命令実行時 • システム割込みで使用する命令実行時 (PLSY,PWM)
CC-Link IE コントローラネットワークまたは MELSECNET/H の LX, LY のリフレッシュ先の CPU ユニットの入出力	<ul style="list-style-type: none"> • END 処理（CC-Link IE コントローラネットワーク, MELSECNET/H リフレッシュ時） • COM 命令実行時 • ZCOM 命令実行時 	
CC-Link の RX, RY のリフレッシュ先の CPU ユニットの入出力	<ul style="list-style-type: none"> • END 処理（自動リフレッシュ時） • COM 命令実行時（自動リフレッシュ時） • ZCOM 命令実行時（自動リフレッシュ時） 	

(f) 登録可能なデバイス数

強制 ON/OFF をあわせて合計 32 個のデバイスを登録できます。

(g) シーケンスプログラムで出力 Y の接点を使用している場合

シーケンスプログラムの動作が優先されます。

(h) 強制 ON/OFF 実行状況の確認

下記の方法で確認可能です。

- GX Developer の登録状況読出
- MODE LED の緑点滅（デバイスが 1 つでも登録されている場合は、MODE LED が点滅します。）
- SD840（デバッグ機能使用状況）の 1 ビット目の ON

(i) 複数の GX Developer からの強制 ON/OFF

ネットワークなどに接続されている複数の GX Developer から同一の CPU ユニットに対して強制 ON/OFF の登録ができます。

ただし、同一のデバイスに強制 ON/OFF を登録した場合、あとに登録された ON/OFF 状態になります。先に強制 ON/OFF を実行した GX Developer は、CPU ユニットの ON/OFF 情報と異なる ON/OFF 情報を表示することがあります。

複数の GX Developer により強制 ON/OFF を行っている場合は、“登録状況読出” ボタンにより最新のデータにしてから強制 ON/OFF を実行してください。

(3) 操作手順

操作手順を下記に示します。

- 指定デバイスを強制 ON/OFF 登録します。
GX Developer の [オンライン] → [デバッグ] → [強制入出力登録／解除]
- 強制入出力登録／解除設定画面でデバイスを設定後, “強制 ON 登録” / “強制 OFF 登録” を選択することにより, 指定したデバイスの強制 ON / 強制 OFF ができます。

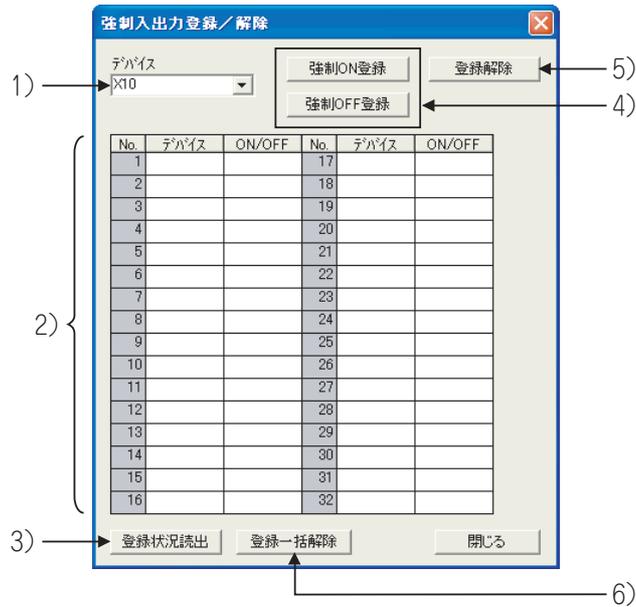


図 6.38 強制入出力登録／解除画面

表 6.16 強制入出力登録／解除画面の項目

No.	設定項目名	機能説明
1)	デバイス	強制 ON/OFF, または強制 ON/OFF を解除する入出力番号を入力する。
2)	登録状況表示エリア	登録されている強制入出力登録状況を表示する。
3)	登録状況読出	CPU ユニットから読み出した登録状況を表示する。
4)	強制 ON/OFF 登録	設定したデバイスの強制 ON/OFF を登録する。
5)	登録解除	登録したデバイスの強制 ON/OFF を解除する。
6)	登録一括解除	登録されているすべての強制入出力登録を解除する。

(4) 二重化 CPU 使用時の注意事項

(a) 登録／解除を行う系について

二重化 CPU 使用時に強制入出力の登録／解除を行う場合は、制御系に対して行ってください。(制御系および待機系個別に登録／解除はできません。)

系切替え後は、新制御系(系切替えで待機系から制御系になったシステム)に対して登録／解除を行ってください。

待機系に登録／解除を行った場合は、エラー(エラーコード:4240H)になります。

(b) セパレートモード時の電源 OFF → ON またはリセットについて

セパレートモード時に制御系の電源 OFF またはリセットを行うと、強制入出力の登録が解除されます。

登録解除後の動作は、表 6.13 の“解除(未操作)時”に示す動作と同様になります。

ただし、MELSECNET/H リモート I/O 局上のユニットの出力は、下記のようにになります。

1) 電源 OFF またはリセット中

制御系の電源 OFF またはリセット時の出力状態を保持します。

2) 電源 OFF → ON またはリセット解除時

シーケンスプログラムでの演算結果を出力します。(強制入出力解除時と同様)

6.12 CPU ユニットが RUN 中のプログラム書込み

CPU ユニットが RUN 中のプログラム書込みには、下記の 2 種類があります。

- 回路モードでの RUN 中書込み : 6.12.1 項
- ファイルの RUN 中書込み : 6.12.2 項

また、ポインタを使用して RUN 中書込みすることもできます。(6.15.2 項)

Point

二重化 CPU 使用時の RUN 中書込み（制御系と待機系への書込み順、RUN 中書込み実行時のトラッキング転送の実行可否など）については、下記マニュアルを参照してください。

QnPRHCPU ユーザーズマニュアル（二重化システム編）

6.12.1 回路モードでの RUN 中書込み

(1) 回路モードでの RUN 中書込みとは

CPU ユニットの RUN 中にプログラムの書込みを行う機能です。

回路モードでの RUN 中書込みを使用すると、CPU ユニットのプログラムの演算を止めずにプログラムの変更ができます。

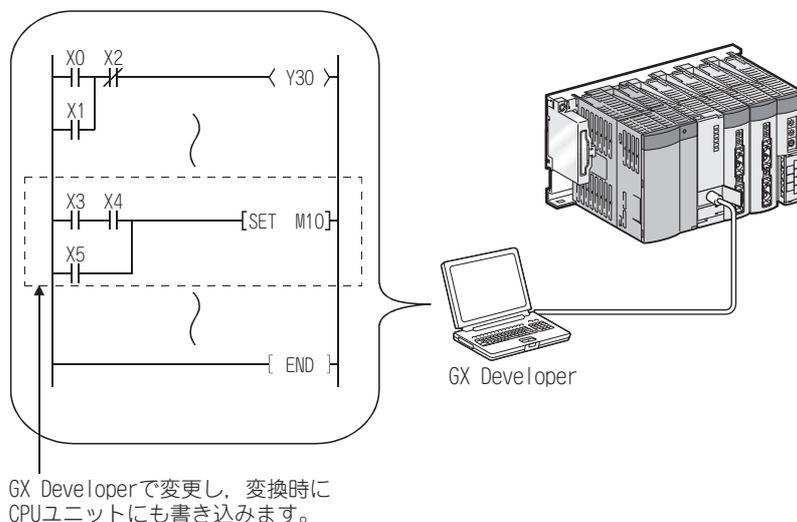


図 6.39 回路モードでの RUN 中書込みの概略

また、ネットワークの他局に接続した GX Developer からでもプログラムの RUN 中書き込みができます。

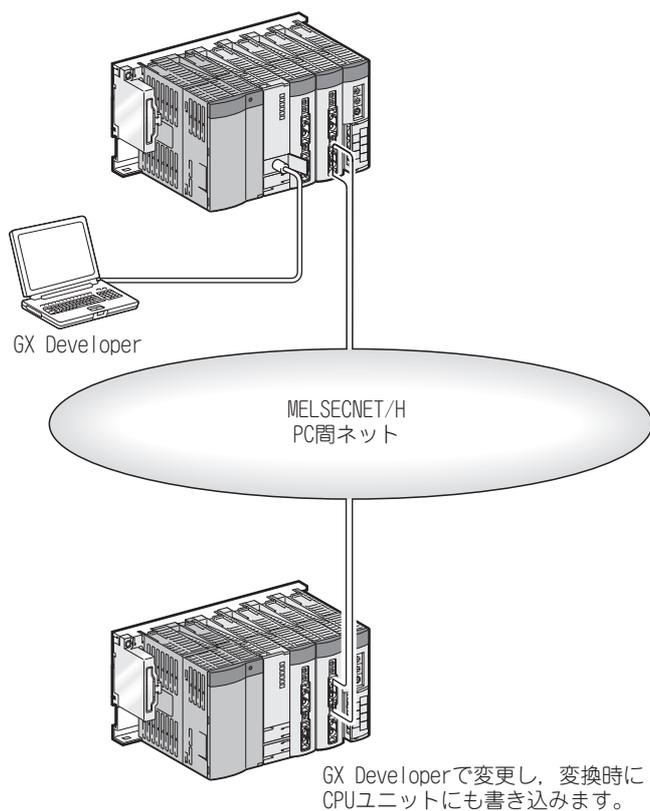


図 6.40 ネットワーク経由の RUN 中書き込みの概略

(2) RUN 中書き込みが可能なメモリ

RUN 中書き込みが可能なメモリは、プログラムメモリです。

(3) 一度に RUN 中書き込み可能なステップ数

最大 512 ステップです。

(4) 低速実行タイププログラムの RUN 中書込みの実行タイミング 注 6.6

低速実行タイププログラムの実行がすべて完了した次のスキャンの END 処理時で行います。
また、低速実行タイププログラムの RUN 中書込み中は、低速実行タイププログラムの実行を中断します。

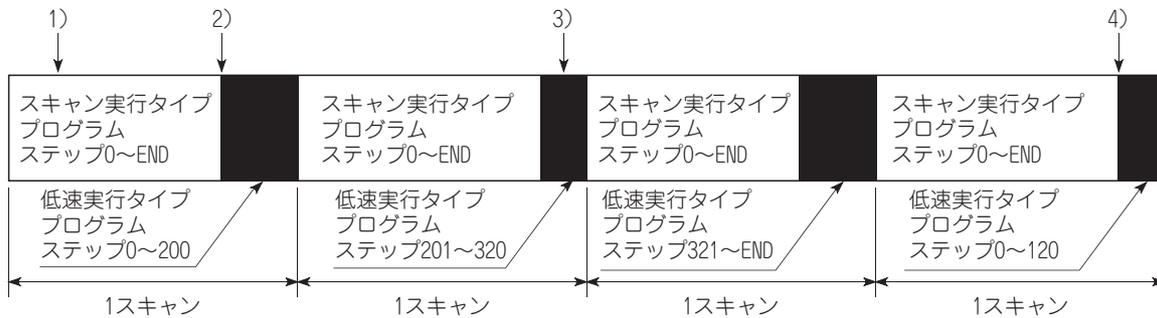


図 6.41 低速タイププログラム実行時の RUN 中書込み処理

- 1) : スキャン実行タイププログラムの RUN 中書込み指令
- 2) : スキャン実行タイププログラムの RUN 中書込み実行
- 3) : 低速実行タイププログラムの RUN 中書込み指令
- 4) : 低速実行タイププログラムの RUN 中書込み実行

(5) PLOADP, PUNLOADP, PSWAPP 命令実行中の RUN 中書込み 注 6.7

実行中の命令の処理が完了するまで、RUN 中書込みの処理が待たされます。
また RUN 中書込み実行中に各命令を実行すると、RUN 中書込みの処理が完了するまで各命令の実行が待たされます。

注 6.6

Basic **Redundant**

ベーシックモデル QCPU, 二重化 CPU では、低速実行タイププログラムが使用できません。

注 6.7

Basic **Redundant**

ベーシックモデル QCPU, 二重化 CPU では、PLOADP, PUNLOADP, PSWAPP 命令による RUN 中でのプログラムファイルの変更はできません。

(6) RUN 中書込用確保ステップの変更

プログラムファイルには、プログラムファイルの容量が変化する RUN 中書込みに対応するために、RUN 中書込用確保ステップがあります。

RUN 中書込用確保ステップを変更する場合の注意事項について説明します。

(a) プログラムファイルの容量

作成したプログラムの容量に RUN 中書込用確保ステップを加算した値になります。

(b) プログラムファイルの容量が確保した容量より増加する場合

RUN 中書込みの際、プログラムファイルの容量が確保している容量（RUN 中書込用確保ステップを含む容量）を超える場合、ユーザ用メモリエリアに空きエリアがあれば、RUN 中書込み時に RUN 中書込用確保ステップの再設定が可能です。

(c) スキャンタイムの延び時間

RUN 中書込み時に RUN 中書込用確保ステップを再設定した場合、スキャンタイムが延びます。

スキャンタイムの延び時間については、10.1.3 項を参照してください。

(7) RUN 中書込み時に正常動作しない命令について

6.12.3 項 (2) を参照してください。

6.12.2 ファイルの RUN 中書込み 注 6.8

(1) ファイルの RUN 中書込み機能とは

CPU ユニットの RUN 中に、GX Developer のオンライン操作により、表 6.17 のファイルを CPU ユニットに一括で書き込む機能です。

表 6.17 CPU ユニットの RUN 中に書き込み可能なファイル

ファイル種別	CPU ユニット内蔵メモリ			メモリカード (RAM)	メモリカード (ROM)	
	プログラムメモリ	標準 RAM	標準 ROM	SRAM カード	Flash カード	ATA カード
パラメータ	×	×	×	×	×	×
インテリジェント機能 ユニットパラメータ	×	×	×	×	×	×
プログラム	○	×	○	○	×	○
デバイスコメント	○	×	△ ^{*1}	△ ^{*1}	×	△ ^{*1}
デバイス初期値	×	×	×	×	×	×
ファイルレジスタ	×	△ ^{*1}	×	△ ^{*1}	×	×
ローカルデバイス	×	×	×	×	×	×
サンプリングトレース ファイル	×	○	×	○	×	×
PC ユーザデータ	×	×	○	×	×	○

○：書き込み可能、△：一部制約あり、×：書き込み不可

* 1：シーケンスプログラムでアクセス中でなければ書き込み可能です。

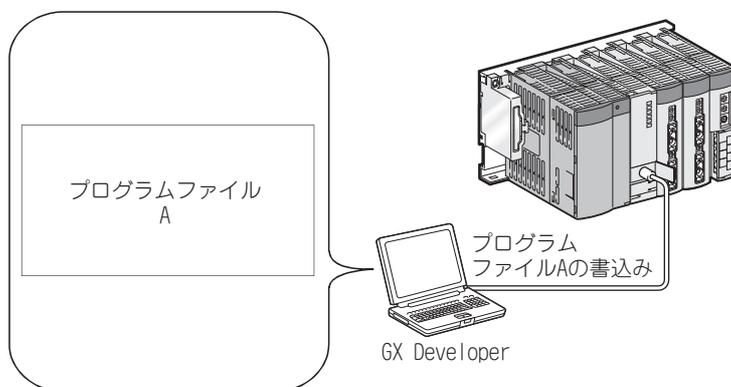


図 6.42 ファイルの RUN 中書込みの概略



注 6.8

Basic

ベーシックモデル QCPU では、ファイルの RUN 中書込みはできません。

(2) ファイルの RUN 中書込みの実行可否

ファイルの RUN 中書込みの実行可否を 表 6.18 に示します。

表 6.18 ファイルの RUN 中書込みの実行可否

書き込むプログラムファイル以上の空きエリア		ファイルの RUN 中書込み
プログラムメモリ	メモリカード	
有	無/有	実行可
無	有	実行可 * 1
無	無	実行不可

* 1：GX Developer のオプションメニューで RUN 中書込み時の命令動作設定で “立下り命令を実行しない” に設定した場合、書込みプログラムファイル内に立下り命令が含まれているとファイルの RUN 中書込みはできません。ただし、書込みプログラムが SFC プログラムの場合は、書込みプログラムファイル内に立下り命令が含まれていてもファイルの RUN 書込みができます。
(GX Developer のオプションメニューで RUN 中書込み時の命令動作設定で “立下り命令を実行しない” に設定した場合の動作については、6.12.3 項を参照してください。)

(3) スキャンタイムの延び時間

プログラムファイルの RUN 中書込みを行うと、スキャンタイムが延びます。
スキャンタイムの延び時間については、10.1.3 項を参照してください。

(4) シーケンスプログラムの命令でファイルにアクセスした場合

ファイルの RUN 中書込み実行中は、プログラムの命令からファイルにアクセスができませんので注意してください。

ファイルの RUN 中書込み実行中にファイルにアクセスした命令は非実行になります。

対象となる命令には、以下の命令があります。

- PLOADP 命令
- PUNLOADP 命令
- PSWAPP 命令

(5) 複数箇所からのファイルの RUN 中書込み

ファイルの RUN 中書込みは、複数箇所から同時に実行しないでください。
実行した場合、プログラムファイルの内容が消える可能性があります。

(6) SFC プログラムのファイルの RUN 中書込み 注 6.9

SFC プログラムのファイルの RUN 中書込みはできません。

(7) ファイルの RUN 中書込み時に正常動作しない命令について

6.12.3 項 (2) を参照してください。



注 6.9

High performance

Process

ハイパフォーマンスモデル QCPU、プロセス CPU で、SFC プログラムのファイルの RUN 中書込みを実行する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。(付 2.2、付 2.3)

6.12.3 RUN 中書込み時の注意事項

RUN 中書込み時の注意事項を下記に示します。

(1) ブート運転中の RUN 中書込み

ブート運転中に RUN 中書込みを実行した場合、ブート元プログラムの状態は使用する CPU ユニットとブート元メモリにより異なります。

表 6.19 RUN 中書込みを行った場合のブート元プログラムの状態

ブート元メモリ		ブート元プログラムの状態			
		ベーシックモデル QCPU	ハイパフォーマンス モデル QCPU	プロセス CPU	二重化 CPU
標準 ROM		変更されない* 2	変更されない* 2	変更されない* 2	変更される* 1* 3
メモリカード	・SRAM カード ・ATA カード	—	変更される* 1	変更される* 1	変更される* 1
	Flash カード	—	変更されない* 2	変更されない* 2	変更される* 1* 3

* 1：ブート元のプログラムは、下図のメッセージボックスが表示されたときに“はい”を選択すると変更されます。このとき、RUN 中書込みの完了までに時間がかかります。



“いいえ”を選択するとブート元プログラムは変更されません。

RUN 中書込み後に CPU ユニットの電源を OFF、または CPU ユニットをリセットする前に、プログラムメモリの内容を標準 ROM / メモリカードに書き込んでください。

* 2：RUN 中書込み後に CPU ユニットの電源を OFF、または CPU ユニットをリセットする前に、プログラムメモリの内容を標準 ROM / メモリカードに書き込んでください。

* 3：ブート元メモリが標準 ROM / Flash カードの場合、ブート元のファイルはすべて消去され、プログラムメモリ上のファイルに置き換わります。

PC パラメータのブートファイル設定で、ブート元メモリ内のファイルはすべてプログラムメモリに転送するように設定してください。

(2) RUN 中書き込み時に正常動作しない命令について

RUN 中書き込みを行ったとき正常に動作しない命令には、下記の3種類があります。

- 立下がり命令
- 立上がり命令
- SCJ 命令
- STMR 命令

(a) 立下がり命令

RUN 中書き込みの書き込み範囲内に立下がり命令がある場合は、RUN 中書き込み完了時に立下がり命令の実行条件 (ON → OFF) が成立していなくても立下がり命令が実行されます。

RUN 中書き込み時に立下がり命令が実行されないようにする方法については、ポイントを参照してください。

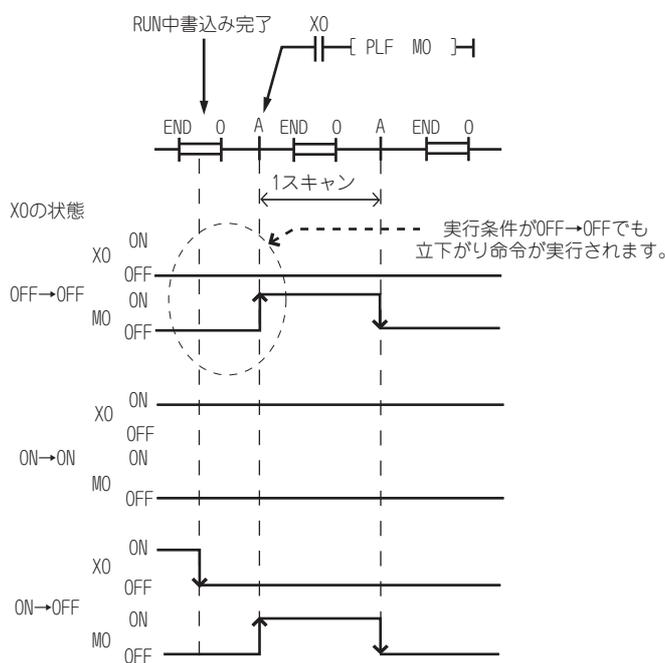


図 6.43 立下がり命令の動作

対象の立下がり命令は、LDF、ANDF、ORF、MEF、PLF、FCALLP、EFCALLP 命令です。

(b) 立上がり命令

RUN 中書き込みの書き込み範囲内に立上がり命令がある場合は、RUN 中書き込み完了時に立上がり命令の実行条件 (OFF → ON) が成立していても立上がり命令が実行されません。

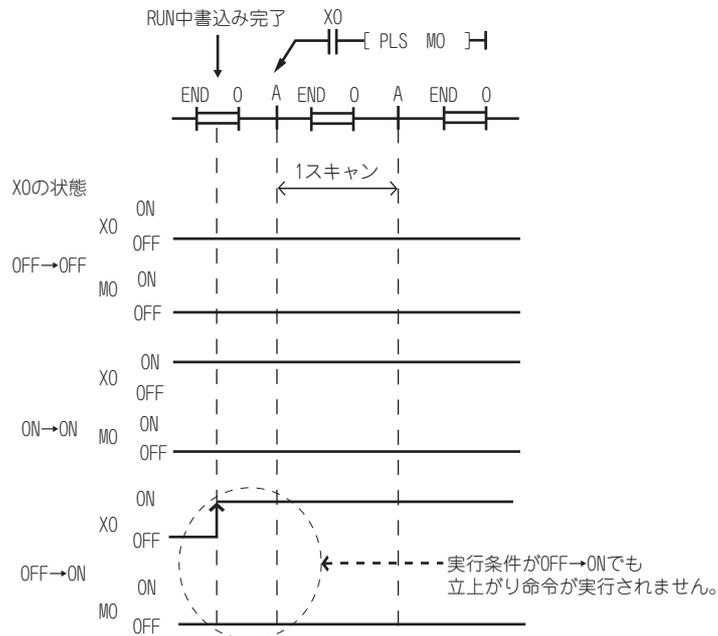


図 6.44 立上がり命令の動作

対象の立上がり命令は、PLS、□P 命令です。

(c) SCJ 命令

RUN 中書き込みの書き込み範囲内に SCJ 命令がある場合は、RUN 中書き込み完了時に SCJ 命令の実行条件が ON していると、1 スキャン待たずに指定ポインタへのジャンプを行います。

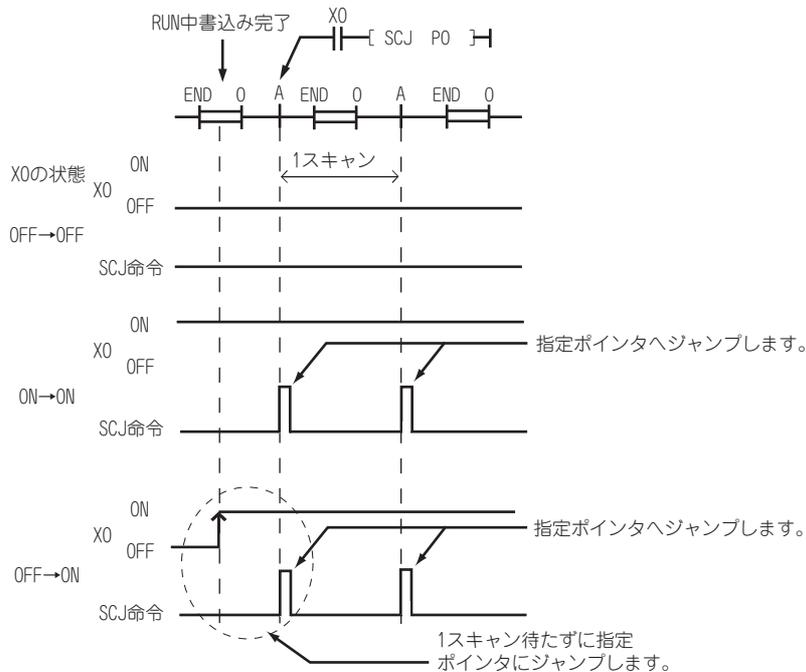


図 6.45 SCJ 命令の動作

(d) STMR 命令

RUN 中書き込みの書き込み範囲内に STMR 命令がある場合は、STMR 命令が動作するので注意してください。

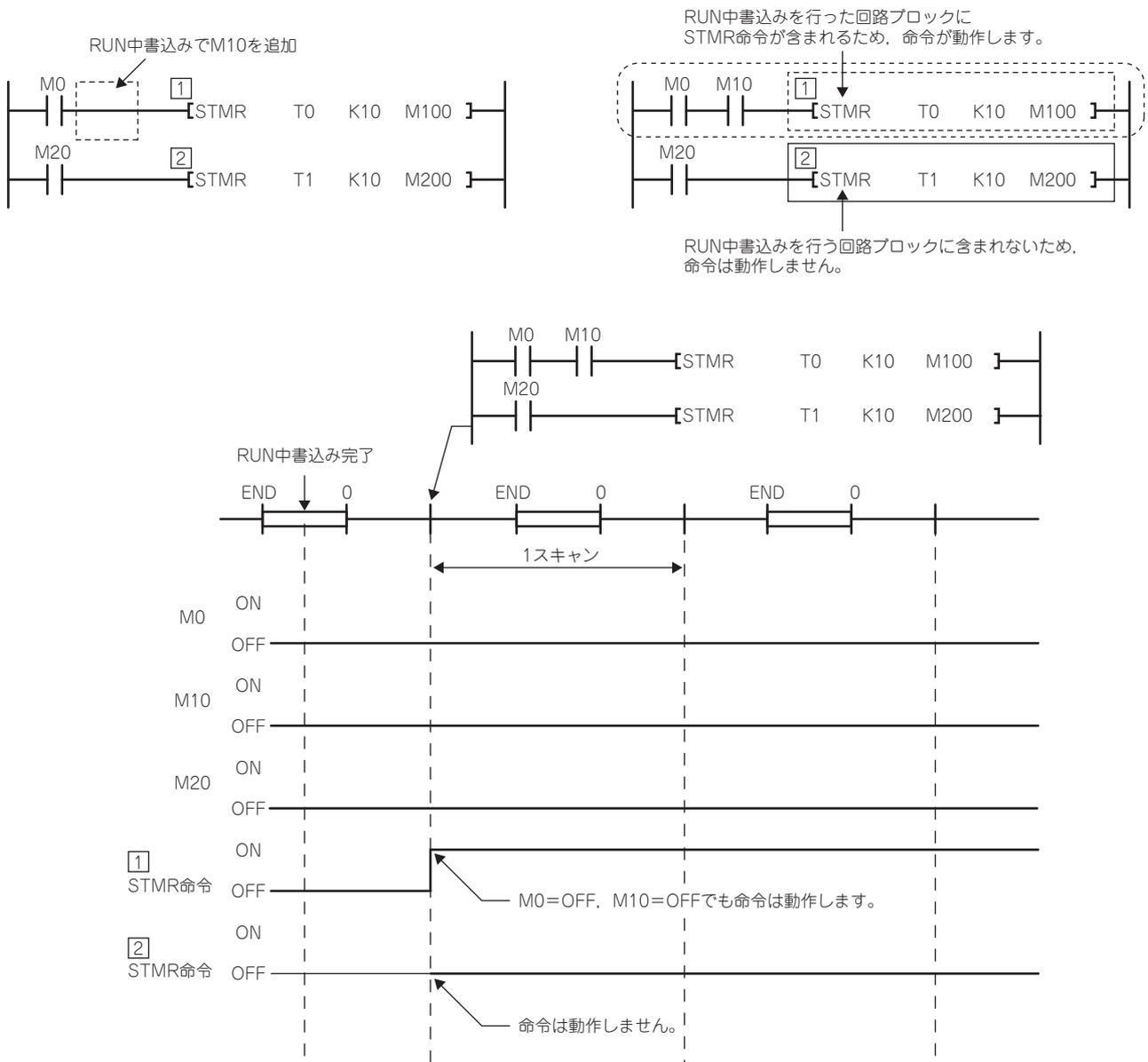


図 6.46 STMR 命令の動作

RUN 中書き込みを行ったとき、立下がり命令の実行条件 (ON → OFF) が成立していなくても、立下がり命令が実行される現象は、GX Developer のオプションメニューで “RUN 中書き込み / ファイルの RUN 中書き込み時の命令動作設定” を行うことにより回避できます。 **注 6.10**
 デフォルトでは、実行条件が OFF の立下がり命令を実行します。

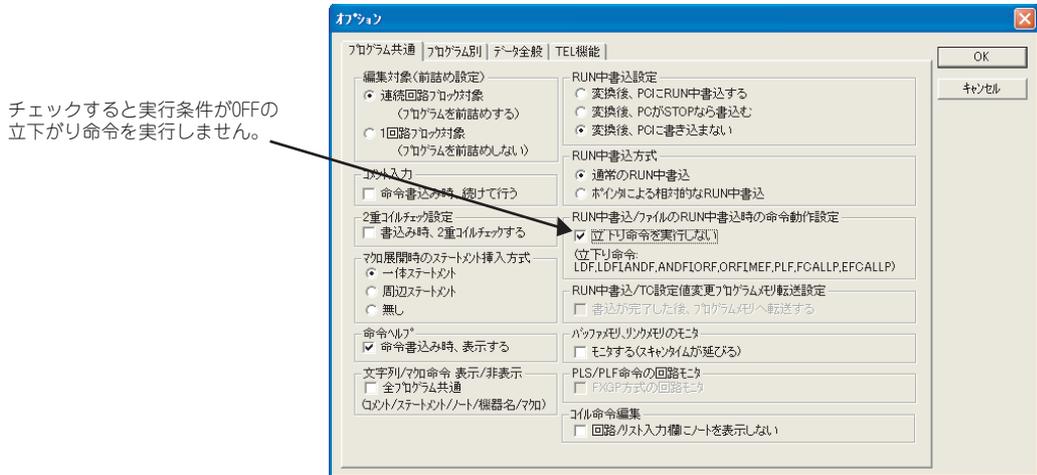


図 6.47 GX Developer のオプションメニュー画面

GX Developer の “RUN 中書き込み / ファイルの RUN 中書き込み時の命令動作設定” による立下がり命令の動作を図 6.48 に示します。

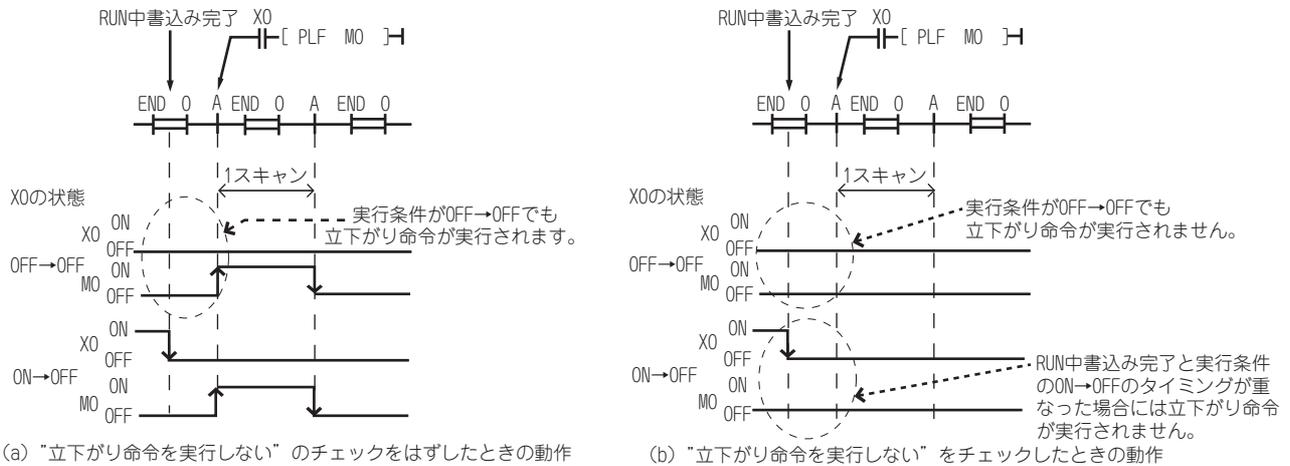


図 6.48 立下がり命令の動作比較

注 6.10 Basic High performance Process Redundant

ベーシックモデル QCPU では、RUN 中書き込み時の立下がり命令の実行 / 非実行選択を使用できません。
 ハイパフォーマンスモデル QCPU、プロセス CPU、二重化 CPU で RUN 中書き込み時の立下がり命令の実行 / 非実行選択を使用する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。(付 2)

6.13 実行時間計測

(1) 実行時間計測とは

実行中のプログラムの処理時間を表示する機能です。

(2) 実行時間計測の用途と種類

システムの調整時に、各プログラムの処理時間が全体のスキャンタイムに及ぼす影響を知るために使用できます。

実行時間計測には、下記の3種類があります。

- プログラム一覧モニタ :  6.13.1 項
- 割り込みプログラム一覧モニタ :  6.13.2 項
- スキャンタイム測定 :  6.13.3 項

6.13.1 プログラム一覧モニタ

(1) プログラム一覧モニタとは

プログラム別のスキャンタイム、実行回数および項目別の処理時間を表示します。

(2) プログラム一覧モニタの実行

GX Developer の [オンライン] → [モニタ] → [プログラム一覧モニタ] でプログラム一覧モニタ画面を表示して行います。* 1

プログラム一覧モニタの実行例を図 6.49 に示します。



図 6.49 プログラム一覧モニタ

* 1 : 定周期実行タイププログラム実行時は、定周期実行タイププログラムのスキャンタイムは表示されません。スキャンタイムの欄には“ - ”が表示されます。

(a) 全体のスキヤンタイム

PC パラメータの“PC RAS 設定”の WDT (ウォッチドッグタイマ) 設定で設定した監視時間, および CPU ユニットが実行しているプログラムの実行タイプ別のスキヤンタイム合計時間を表示します。

1) 監視時間

それぞれのプログラムの監視時間を表示します。

スキヤンタイム時間が, この時間を超えると CPU ユニットで“WDT ERROR”になります。

2) スキヤンタイム合計

“スキヤン実行分のスキヤンタイム詳細”の各項目の合計時間を表示します。

コンスタントスキヤン時間を設定した場合は, コンスタントスキヤン時間を表示します。

(b) スキヤン実行分のスキヤンタイム詳細

スキヤンタイムの詳細を表示します。

1) プログラム分

スキヤン実行タイププログラムの実行時間の合計を表示します。

2) END 処理時間

END 処理時間を表示します。

3) 低速プログラム分

- ・ハイパフォーマンスモデル QCPU, プロセス CPU の場合

低速実行タイププログラム実行時間, またはコンスタントスキヤン時間設定時にはコンスタントスキヤン時間と低速実行タイププログラムの実行時間の合計を表示します。

- ・ベーシックモデル QCPU, 二重化 CPU の場合

低速プログラムは使用できないため, “0.000” が表示されています。

4) コンスタント待ち

コンスタントスキヤン時間設定時, コンスタントスキヤンの待ち時間を表示します。

(c) 各プログラム実行状態

PC パラメータのプログラム設定で指定したプログラムの実行状態を表示します。

1) プログラム

PC パラメータで設定した順番にプログラム名を表示します。

2) 実行

PC パラメータで設定したプログラムの実行タイプを表示します。

3) スキヤンタイム

実際のスキヤンタイム (現在値) を表示します。

プログラム停止 (待機) 状態では, スキヤンタイムは 0.000ms と表示します。

4) 実行回数

計測を開始した時点をも 0 回として, モニタ時まで実行した回数を表示します。

実行回数は最大 65535 回まで表示し, 65536 回目の計測で 0 に戻ります。

プログラム停止時も実行回数は保持されます。

備考

POFF 命令を実行すると、非実行処理を 1 スキャン行います。実行回数の表示は、非実行処理分の実行回数を加算しない値になります。

POFF 命令の詳細については、下記マニュアルを参照してください。

☞ MELSEC-Q/L プログラミングマニュアル（共通命令編）

(3) プログラムの起動 注 6.11

図 6.49 の画面上（☞ 本項(2)）の「プログラム起動」ボタンをクリックすると、プログラム起動画面が表示されます。

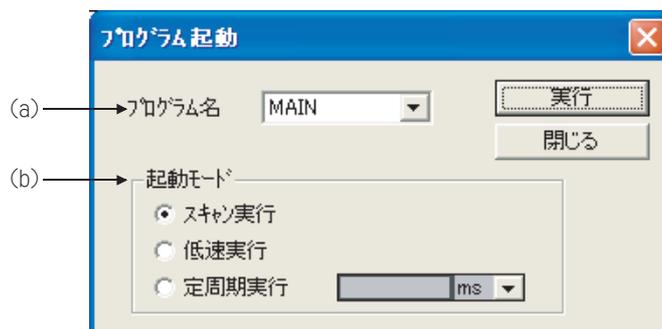


図 6.50 プログラム起動画面

(a) プログラム名

PC パラメータのプログラム設定で設定したプログラムを選択します。
プログラム名は任意に入力できません。

(b) 起動モード

待機タイププログラムをどのプログラムとして起動するかを設定できます。

- “スキャン実行”：スキャン実行タイププログラム
- “低速実行”：低速実行タイププログラム 注 6.12
- “定周期実行”：定周期実行タイププログラム

定周期実行のデフォルトは、PC パラメータのプログラム設定で設定した値が表示されます。
単位は ms, s を選択できます。

注 6.11 **Basic**

ベーシックモデル QCPU では、プログラム一覧モニター画面でプログラムの起動はできません。

注 6.12 **Redundant**

二重化 CPU では、低速実行タイププログラムが使用できないため、起動モードで“低速実行”は選択できません。

(4) プログラムの停止 注 6.13

図 6.49 の画面上 ( 本項 (2)) の **プログラム停止** ボタンをクリックすると、プログラム停止画面を表示します。

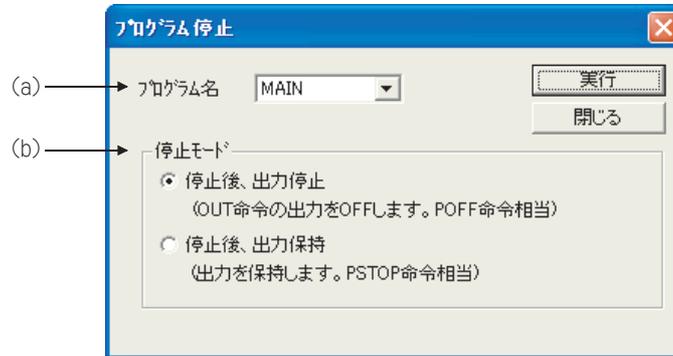


図 6.51 プログラム停止画面

(a) プログラム名

PC パラメータのプログラム設定で設定したプログラムを選択します。
プログラム名は任意に入力できません。

(b) 停止モード

下記プログラムに対して、“停止後、出力停止”を実行すると、次のように動作します。

- スキャン実行タイププログラム
次のスキャンで出力を OFF (非実行処理) します。
その次のスキャン以降待機状態になります。(POFF 命令実行時と同一動作になります。)
- 低速実行タイププログラム
低速実行タイプの実行を中断し、次のスキャンで出力を OFF します。
その次のスキャン以降待機タイププログラムになります。(POFF 命令実行時と同一動作になります。)
- 待機タイププログラム
無処理となります。(POFF 命令実行時と同一動作になります。)
このため、“実行回数”も + 1 されません。

Point

“停止後、出力停止”を実行しても、命令によっては出力が OFF しない場合があります。
詳細については下記マニュアルの POFF 命令の項目を参照してください。
 MELSEC-Q/L プログラミングマニュアル (共通命令編)

注 6.13 **Basic**

ベーシックモデル QCPU では、プログラム一覧モニタ画面でプログラムの停止はできません。

6.13.2 割込みプログラム一覧モニタ

(1) 割込みプログラム一覧モニタとは

割込みプログラムの実行回数を表示する機能です。
割込みプログラムの実行状態確認時に使用します。

(2) 割込みプログラム一覧モニタの実行

GX Developer の [オンライン] → [モニタ] → [割込みプログラム一覧] で、割込みプログラム一覧モニタウィンドウを表示して行います。

割込みプログラム一覧モニタの実行例を図 6.52 に示します。

割込みポイント	実行回数	共通コメント
I28	3243	100ms割込み
I29	8109	40ms割込み
I30	16219	20ms割込み
I31	0	
I32	0	
I33	0	
I34	0	
I35	0	
I36	0	
I37	0	
I38	0	
I39	0	
I40	0	
I41	0	
I42	0	
I43	0	

図 6.52 割込みプログラム一覧モニタ

(a) 実行回数

割込みプログラムを実行した回数を表示します。

CPU ユニットを RUN 状態にすると計測を開始し、最大 65536 回計測すると 0 回に戻ります。

(b) 共通コメント

割込みポイントに作成されているデバイスコメントを表示します。

6.13.3 スキャンタイム測定 注 6.14

(1) スキャンタイム測定とは

回路モニタ実行時プログラムの任意区間の処理時間を表示する機能です。
サブルーチンプログラムや、割込みプログラム内の時間も測定できます。

(2) スキャンタイム測定範囲指定

スキャンタイムの測定範囲指定には、下記の 2 種類の方法があります。

- 回路モニタ画面で指定する
- スキャンタイム測定画面で指定する

(3) サブルーチンプログラムのコール命令が存在する場合

スキャンタイムの測定範囲内にサブルーチンプログラムのコール命令（CALL 命令）が存在する場合は、サブルーチンプログラム処理も含めた時間になります。

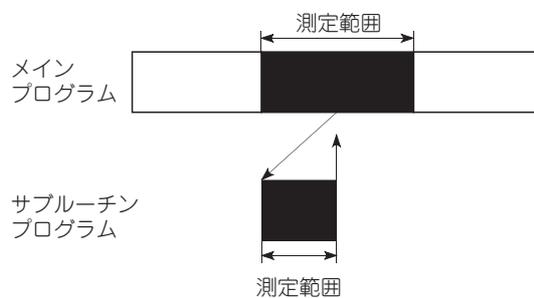


図 6.53 測定範囲にサブルーチンプログラムが含まれる場合

(4) スキャンタイム測定範囲内に割込み／定周期実行タイププログラムが実行された場合

割込み／定周期実行タイププログラムの実行時間が加算されます。



注 6.14 **Basic**

ベーシックモデル QCPU では、スキャンタイム測定が使用できません。

(5) スキャンタイム測定の実行

スキャンタイム測定は下記手順で行います。

- GX Developer でスキャンタイム測定を行う回路の先頭を表示後、モニタモードにします。
- [オンライン] → [モニタ] → [スキャンタイム測定] でスキャンタイム測定画面を表示します。



- 開始ステップ、終了ステップを指定して、**開始** ボタンをクリックします。

例 開始ステップ：52，終了ステップ：105 の場合



図 6.54 スキャンタイム測定画面

備考

モニタモードで、スキャンタイム測定範囲を指定してからスキャンタイム実行画面を表示させると、範囲を選択した部分が開始ステップ、終了ステップに設定されます。スキャンタイム測定範囲の指定は、「Shift」キー＋クリックで行います。(指定した部分が反転します。)

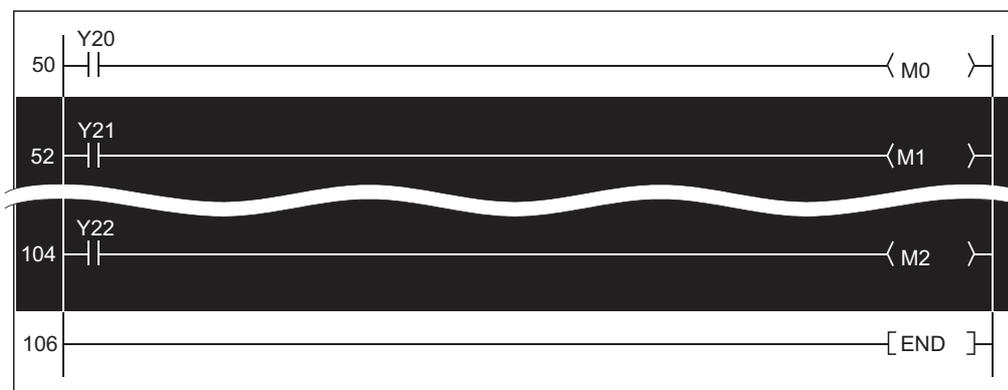


図 6.55 測定範囲の指定

(6) 注意事項

(a) 測定範囲の設定時

開始ステップ < 終了ステップ となるように設定してください。

(b) 計測時間の最小単位

計測時間の最小単位は 0.100ms です。

計測時間が 0.100ms 未満は、0.000ms を表示します。

(c) FOR ~ NEXT 命令間を指定した場合

指定ステップ間を 1 回実行した時間が計測されます。

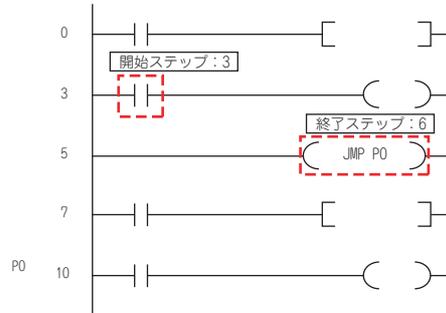
(d) スキャンタイムが測定できない場合

プログラムファイル間をまたぐスキャンタイムの測定はできません。

また、下記の場合、スキャンタイム測定画面でスキャンタイムを更新しません。

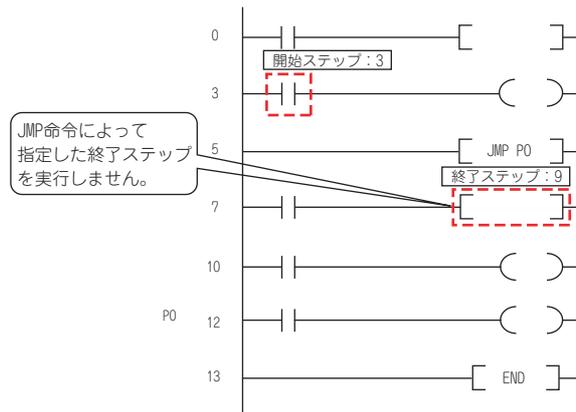
- 分岐命令を終了ステップに指定した場合

例 JMP 命令を終了ステップに指定したとき。



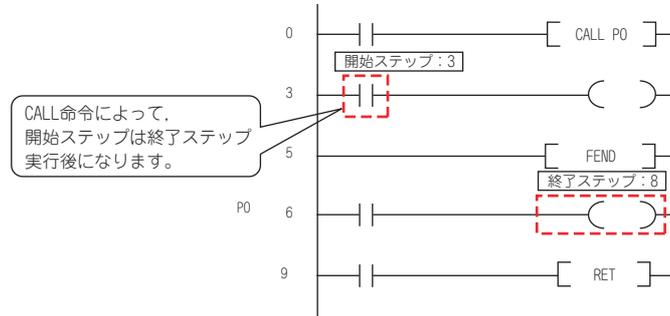
- 開始ステップのみ実行した場合

例 JMP 命令によって、指定した終了ステップを実行しないとき。



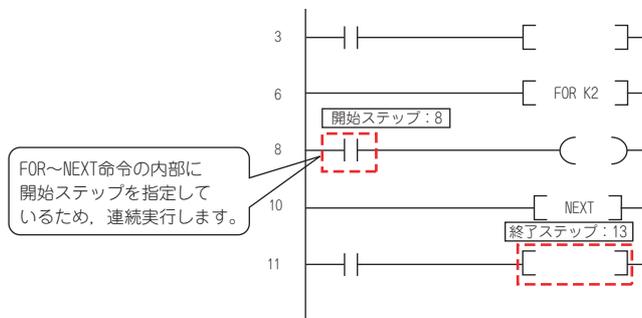
- 終了ステップ、開始ステップの順で実行した場合

例 CALL 命令の次ステップに開始ステップを指定し、CALL 命令で実行されるサブルーチンプログラム内に終了ステップを指定したとき。



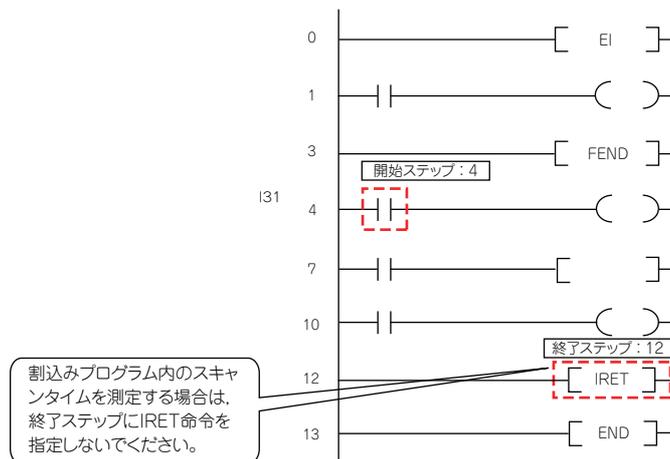
- 開始ステップを連続実行した場合

例 FOR ~ NEXT 命令間に開始ステップのみを指定したとき。



- 終了ステップに IRET 命令、FEND 命令、BREAK 命令、RET 命令を指定した場合

例 I31 による割り込みプログラムで終了ステップに IRET 命令を指定したとき



6.14 サンプルングトレース機能 注 6.15

(1) サンプルングトレース機能とは

指定したタイミングで CPU ユニットの指定デバイスの内容を一定間隔（サンプルング周期）でサンプルングし、トレース結果をサンプルングトレースファイルに格納する機能です。

(2) サンプルングトレースの用途

デバッグ時にプログラムで使用しているデバイスの内容の変化を、指定したタイミングで確認できます。またトリガ条件成立時のデバイス内容の取込みも行います。

(3) サンプルングトレースファイル

サンプルングトレースファイルには、サンプルングトレースを行うためのトレース設定、トレース結果を格納します。

Point

- サンプルングトレースファイルは、標準 RAM または SRAM カードにのみ格納できます。( 5.2.1 項 (2))
 - 標準 RAM にサンプルングトレースファイルを格納する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。( 付 2)
-



注 6.15 **Basic**

ベーシックモデル QCPU では、サンプルングトレース機能が使用できません。

(4) サンプリングトレースの動作

(a) CPU ユニットの動作

GX Developer でトレースを開始すると、設定したトレース回数分のトレースを行います。

サンプリングトレース用のエリアは最大 60k バイトです

トレース回数は、サンプリングトレース用のエリアのバイト数を、設定したデバイスのバイト数 ($N1 + N2 + N3 + \text{ワードデバイス点数} \times 2 + (\text{ビットデバイス点数} \div 16) \times 2$) で割った値になります。^{*1 *2}

* 1：計算式中の“ビットデバイス点数 $\div 16$ ”は、小数点以下は切り上げて計算してください。

* 2：N1 ~ N3 は、トレース条件設定のトレース付加情報で設定している項目に応じて、下記の値を加算します。

- ・ N1：時刻を設定時“4”を加算
- ・ N2：ステップ No. を設定時“10”を加算
- ・ N3：プログラム名を設定時“8”を加算

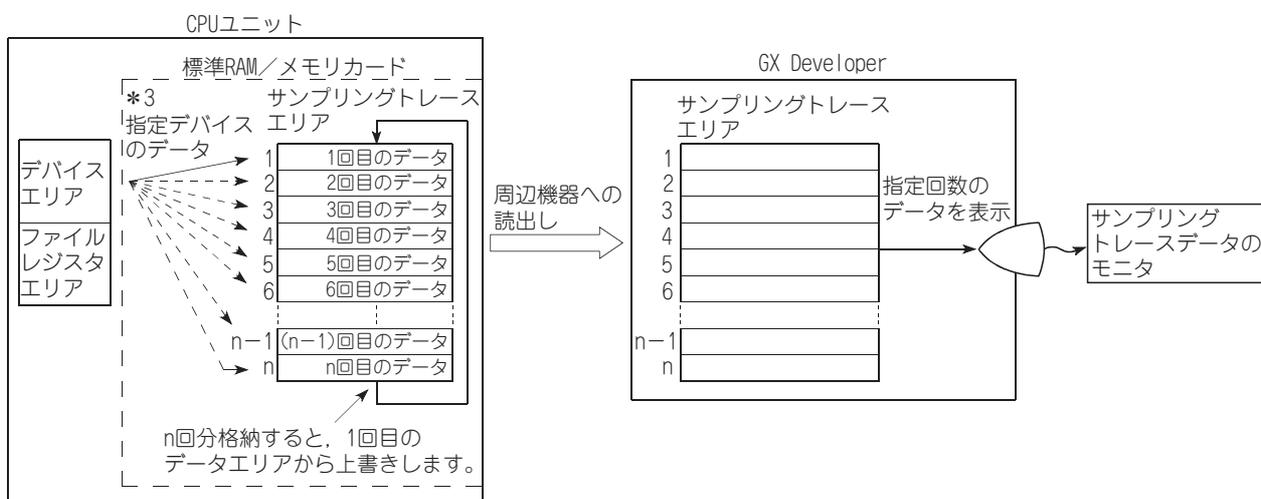


図 6.56 サンプリングトレースの動作

* 3：トリガポイントが実行されると指定回数サンプリング後、サンプリングトレースエリアのデータをラッチします。

(b) 特殊リレーの動作

1) サンプルングトレース実行時

表 6.20 に示す特殊リレーで、サンプルングトレースの実行状態を確認することができます。

表 6.20 サンプルングトレースの実行状態

番 号	名 称	内 容
SM800	トレース準備	GX Developer で設定した “トレース設定” を CPU ユニットに書き込むと ON します。 本リレーの ON により、サンプルングトレースを実行できるかが確認できます。
SM801	トレース開始	トレースを開始すると、ON します。
SM802	トレース実行中	トレース実行中に ON します。 本リレーの ON により、サンプルングトレースを実行しているかが確認できます。
SM803	トレーストリガ	本リレーの OFF → ON でトレースのトリガが ON になります。
SM804	トレーストリガ後	下記のいずれか 1 つが成立すると、ON します。 <ul style="list-style-type: none"> • GX Developer からのトリガ実行 • TRACE 命令実行 • SM803 の ON • 詳細設定 (デバイス, ステップ) 本リレーの ON により、トリガ条件が成立しているかが確認できます。
SM805	トレース完了	トレース完了で ON します。
SM826	トレースエラー	トレース実行中にエラーがあった場合に ON します。

サンプルングトレースの特殊リレーの動作を図 6.57 に示します。

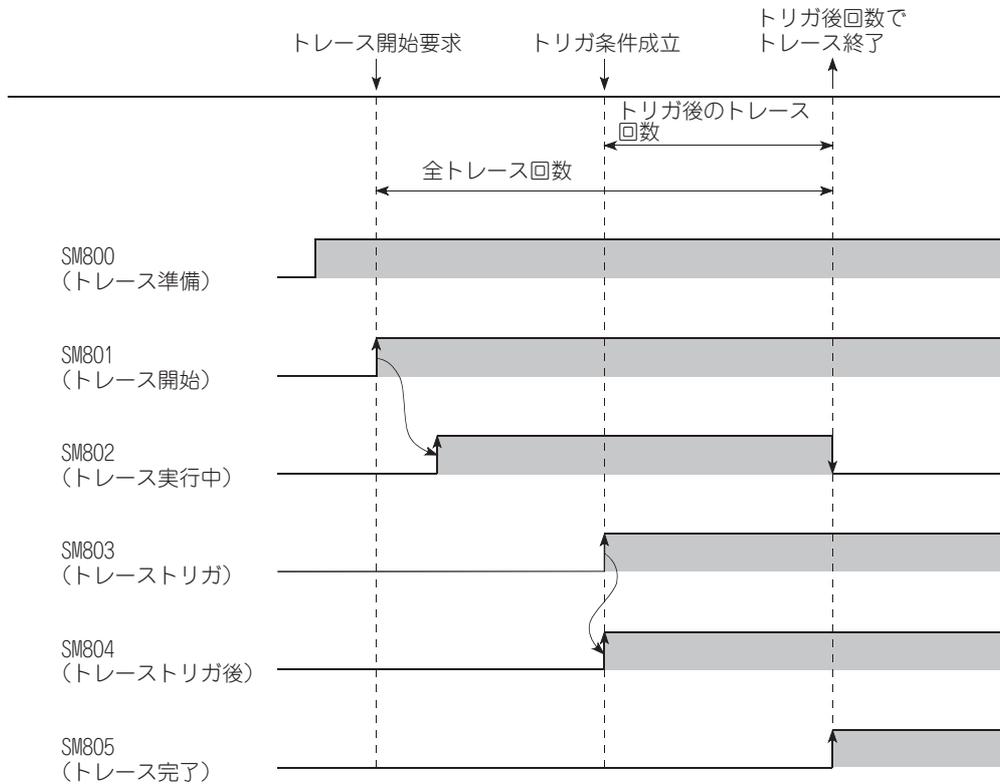


図 6.57 特殊リレーの動作 (サンプルングトレース実行時)

2) サンプリングトレースの中断時の動作

サンプリングトレース中に SM801 (トレース開始) が OFF すると、サンプリングトレースを中断します。サンプリングトレースを中断すると、トレース回数はクリアされます。

SM801 を再度 ON すると、サンプリングトレースを再開します。

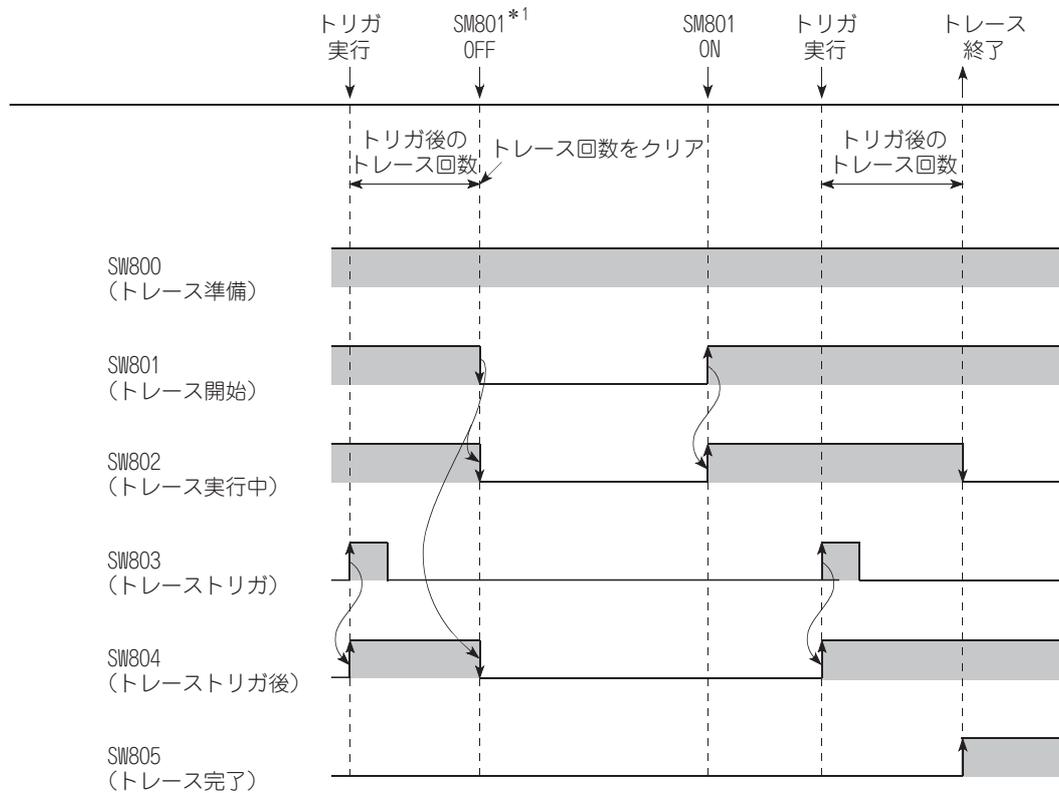


図 6.58 特殊リレーの動作 (サンプリングトレース中断時)

* 1 : GX Developer からトレースを中断させた場合でも、SM800 は OFF しません。

(5) 操作手順

GX Developer の [オンライン] → [トレース] → [サンプルングトレース] から行います。
サンプルングトレースの操作手順には、下記の 2 種類があります。

- “ウィザード設定/実行” から行う方法
(☞ GX Developer Version 8 オペレーティングマニュアル)
- “個別設定/実行” から行う方法
(☞ 本項 (5) (a))

(a) トレースデータ (設定 + 結果) 格納先, トレース実行方法の設定

トレースデータ (設定 + 結果) 格納先とトレース実行方法の設定は、下記画面から行います。

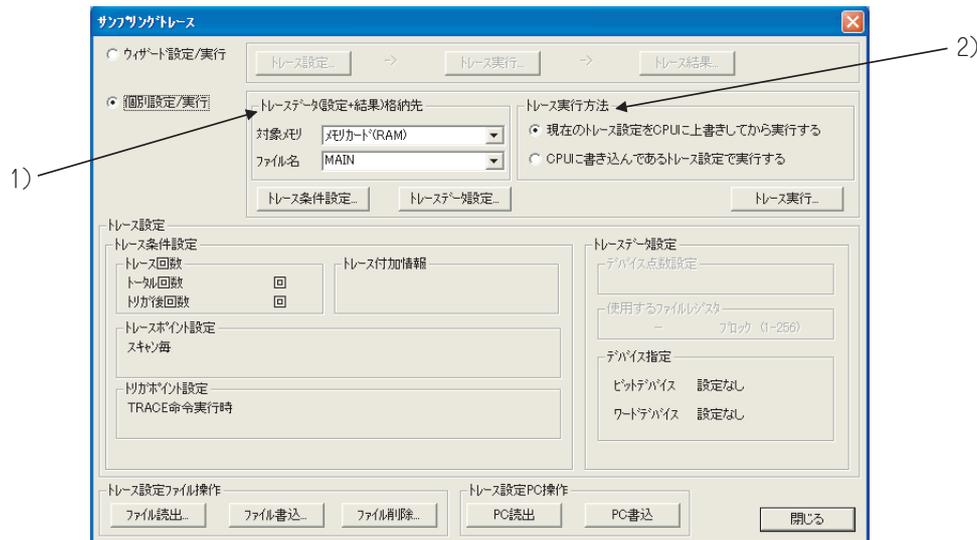


図 6.59 サンプルングトレース画面

1) トレースデータ (設定 + 結果) 格納先

トレースデータを格納する対象メモリ、およびトレース条件を書き込むファイル名を設定します。
対象メモリは、“標準 RAM” または “メモリカード (RAM)” のいずれかを選択します。
トレース結果は、本設定で設定した対象メモリに、指定したファイル名で格納されます。

2) トレース実行方法

トレース実行方法は、下記から選択します。

- 現在のトレース設定を CPU ユニットに上書きしてから実行する：
既存のサンプルングトレースファイルに、トレース設定を上書きします。
- CPU ユニットに書き込んであるトレース設定で実行する：
トレースデータ (設定 + 結果) 格納先で指定したサンプルングトレースファイルのトレース設定で実行します。

(b) トレース条件設定

トレース条件設定は、図 6.59 の **トレース条件設定** ボタンをクリックして行います。
トレース条件設定では、下記の項目を設定します。

- トレース回数（トータル回数、トリガ後回数）
- トレースポイント設定
- トリガポイント設定
- トレース付加情報
- トレースを自動開始するか、しないかの設定

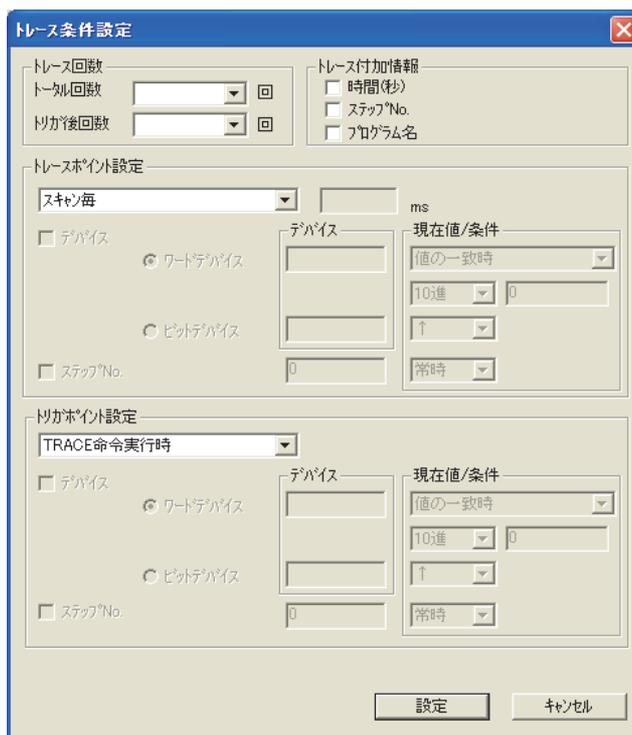


図 6.60 トレース条件設定画面

1) トレース回数

トレース回数設定には、“トータル回数”と“トリガ後回数”があります。

- トータル回数：トレース開始から終了までのサンプリングトレース実行回数を設定します。
- トリガ後回数：トリガ実行からトレース終了までのサンプリングトレース実行回数を設定します。

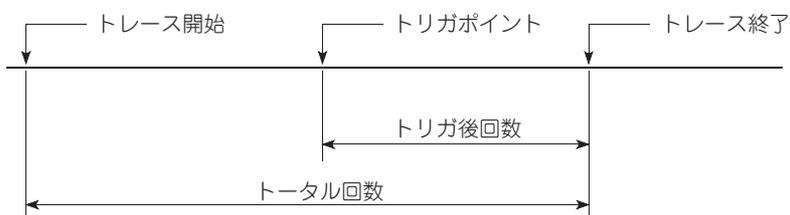


図 6.61 トータル回数およびトリガ後回数の関係

各回数の設定範囲を次に示します。

(トリガ後回数) ≤ (トータル回数) ≤ (8192 回)

2) トレースポイント設定

トレースポイント設定は、トレースデータを収集するタイミングを設定します。
表 6.21 の項目の中から 1 つを選択します。

表 6.21 トレースポイント設定

項目	内容													
スキャン毎	1 スキャンの END 処理ごとにデータの収集を行います。													
時間毎	指定した時間ごとにデータの収集を行います。													
詳細設定	<p>トレースポイントのデバイスおよびステップ番号を設定します。 設定可能なデバイスを以下に示します。</p> <ul style="list-style-type: none"> ビットデバイス：X(DX), Y(DY), M, L, F, SM, V, B, SB, T(接点), ST(接点), C(接点), FX, FY, J□≠X, J□≠Y, J□≠B, J□≠SB, BL□≠S ワードデバイス：T(現在値), ST(現在値), C(現在値), D, SD, W, SW, R, Z, ZR, FD, U□≠G, J□≠W, J□≠SW <p>上記デバイスに対して、以下の修飾が可能です。</p> <ul style="list-style-type: none"> ビットデバイスの桁指定 ワードデバイスのビット No. 指定 ワードデバイスの間接指定 インデックス修飾 <p>設定したデバイスが条件を満たすと収集を行います。条件の設定項目を下記に示します。</p> <table border="1"> <thead> <tr> <th>デバイス</th> <th>条件式</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td rowspan="2">ビットデバイス</td> <td>↑</td> <td>指定したデバイス/ラベルの立上り時にデータを収集します。</td> </tr> <tr> <td>↓</td> <td>指定したデバイス/ラベルの立下り時にデータを収集します。</td> </tr> <tr> <td rowspan="2">ワードデバイス</td> <td>値の一致時</td> <td>指定したデバイス/ラベルの現在値が条件値になったときにデータを収集します。</td> </tr> <tr> <td>値の変更時</td> <td>指定したデバイス/ラベルの現在値が変化したときにデータを収集します。</td> </tr> </tbody> </table> <p>“ステップ No.” 選択時のトレースデータの収集タイミングは、モニタ条件設定時と同様です。 (☞ 6.11.1 項)</p>	デバイス	条件式	内容	ビットデバイス	↑	指定したデバイス/ラベルの立上り時にデータを収集します。	↓	指定したデバイス/ラベルの立下り時にデータを収集します。	ワードデバイス	値の一致時	指定したデバイス/ラベルの現在値が条件値になったときにデータを収集します。	値の変更時	指定したデバイス/ラベルの現在値が変化したときにデータを収集します。
デバイス	条件式	内容												
ビットデバイス	↑	指定したデバイス/ラベルの立上り時にデータを収集します。												
	↓	指定したデバイス/ラベルの立下り時にデータを収集します。												
ワードデバイス	値の一致時	指定したデバイス/ラベルの現在値が条件値になったときにデータを収集します。												
	値の変更時	指定したデバイス/ラベルの現在値が変化したときにデータを収集します。												

3) トレース付加情報

トレース付加情報は、トレースごとに付加する情報を設定します。
次の中から複数選択できます。(選択しなくても可)

- 時刻：トレースを実行した時刻を格納します。
- ステップ No.：トレースを実行したステップ No. を格納します。
- プログラム名：トレースを実行したプログラム名を格納します。

4) トリガポイント設定

トリガポイント設定は、トリガを実行するポイントを設定します。

表 6.22 の項目の中から 1 つを選択します。

表 6.22 トリガポイント設定

項目	内容													
TRACE 命令実行時	TRACE 命令実行時をトリガとします。													
GX Developer からのトリガ操作時	GX Developer からのトリガ実行時をトリガとします。													
詳細設定	<p>トレースポイントのデバイスおよびステップ番号を設定します。 設定可能なデバイスを以下に示します。</p> <ul style="list-style-type: none"> ビットデバイス：X(DX), Y(DY), M, L, F, SM, V, B, SB, T(接点), ST(接点), C(接点), FX, FY ワードデバイス：T(現在値), ST(現在値), C(現在値), D, SD, W, SW, R, ZR <p>上記デバイスに対して、以下の修飾が可能です。</p> <ul style="list-style-type: none"> ワードデバイスのビット No. 指定 <p>なお、間接指定したデバイスは設定できません。</p> <p>設定したデバイスが条件を満たしたタイミングをトリガポイントにします。条件の設定項目を下記に示します。</p> <table border="1"> <thead> <tr> <th>デバイス</th> <th>条件式</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td rowspan="2">ビットデバイス</td> <td>↑</td> <td>指定したデバイス／ラベルの立上り時をトリガとします。</td> </tr> <tr> <td>↓</td> <td>指定したデバイス／ラベルの立下り時をトリガとします。</td> </tr> <tr> <td rowspan="2">ワードデバイス</td> <td>値の一致時</td> <td>指定したデバイス／ラベルの現在値が条件値になったときをトリガとします。</td> </tr> <tr> <td>デバイスへの書き込み時</td> <td>指定したデバイス／ラベルに値が書き込まれたときをトリガとします。</td> </tr> </tbody> </table> <p>“ステップ No.” 選択時のトレースデータの収集タイミングは、モニタ条件設定時と同様です。 (☞ 6.11.1 項)</p>	デバイス	条件式	内容	ビットデバイス	↑	指定したデバイス／ラベルの立上り時をトリガとします。	↓	指定したデバイス／ラベルの立下り時をトリガとします。	ワードデバイス	値の一致時	指定したデバイス／ラベルの現在値が条件値になったときをトリガとします。	デバイスへの書き込み時	指定したデバイス／ラベルに値が書き込まれたときをトリガとします。
デバイス	条件式	内容												
ビットデバイス	↑	指定したデバイス／ラベルの立上り時をトリガとします。												
	↓	指定したデバイス／ラベルの立下り時をトリガとします。												
ワードデバイス	値の一致時	指定したデバイス／ラベルの現在値が条件値になったときをトリガとします。												
	デバイスへの書き込み時	指定したデバイス／ラベルに値が書き込まれたときをトリガとします。												

(c) トレースデータ設定

トレースデータ設定は、図 6.59 の **トレースデータ設定** ボタンをクリックして、行います。
トレースデータに使用できるデバイスを、表 6.23 に示します。



図 6.62 トレースデータ設定画面

表 6.23 トレースデータに使用できるデバイス

項目	内容
ビットデバイス	下記のビットデバイスを最大 50 点設定できます。 X, DX, Y, DY, M, L, F, SM, V, B, SB, T(接点), T(コイル), ST(接点), ST(コイル), C(接点), C(コイル), J□¥X, J□¥Y, J□¥B, J□¥SB, BL□¥S
ワードデバイス	下記のワードデバイスを最大 50 点設定できます。 T(現在値), ST(現在値), C(現在値), D, SD, W, SW, R, Z, ZR, U□¥G, J□¥W, J□¥SW 上記デバイスに対して、以下の修飾が可能です。 <ul style="list-style-type: none"> • ビットデバイスの桁指定 • ワードデバイスのビット No. 指定 • インデックス修飾 <p>なお、間接指定したデバイスは設定できません。</p>

(d) トレース条件設定, トレースデータ設定の書込み

作成したトレース条件設定, トレースデータ設定を, サンプリグトレースファイルとして “トレースデータ (設定 + 結果) 格納先” で設定した対象メモリに書き込みます。

サンプリグトレースファイルの書込みは, 図 6.59 の **PC書込** ボタンをクリックして行います。

Point

サンプリグトレースファイルをメモリカード (SRAM カード) に格納する場合は, ファイル名を変えることにより, 複数のサンプリグトレースファイルを格納できます。
標準 RAM の場合, 格納可能なサンプリグトレースファイルは 1 つのみです。
複数のサンプリグトレースファイルを使用する場合は, メモリカード (SRAM カード) を使用してください。

(e) サンプリグトレースの実行

図 6.59 の **トレース実行** ボタンをクリックすると, サンプリグトレース実行画面が表示されます。

表 6.24 に示す項目を選択して **実行** ボタンをクリックして実行します。

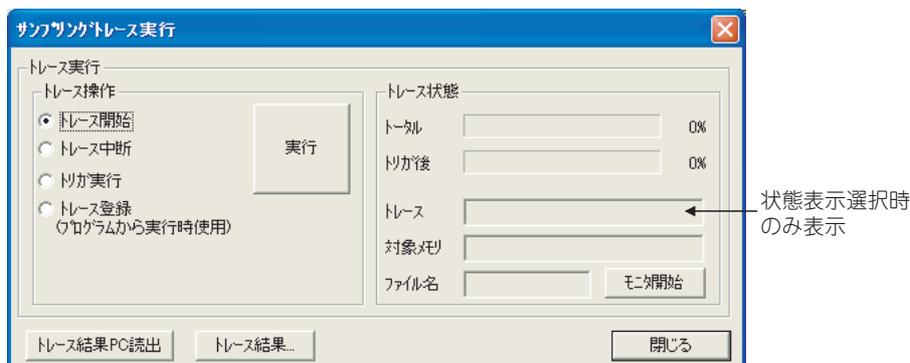


図 6.63 サンプリグトレース実行画面

表 6.24 トレースポイント設定

操作項目	内容
トレース開始	トレースを開始し, トレース回数のカウントを始めます。
トレース中断	トレースを中断し, カウントしたトレース回数, トリガ後のトレース回数をクリアします。 (トレースを再開する場合, 再度 “トレース開始” を選択してください。)
トリガ実行	トリガを実行し, トリガ後のトレース回数のカウントを始めます。 設定したトリガ後回数までカウントした時点でトレースを終了します。
トレース登録 (プログラムから実行時使用)	プログラムから実行時にトレースの登録を行います。

(f) トレース結果の表示

トレース結果を、CPU ユニットから呼び出して表示します。

- 1) 図 6.63 の **トレース結果PC読出** ボタンをクリックして、CPU ユニットからトレース結果を読み出します。
- 2) 図 6.63 の **トレース結果** ボタンをクリックして読み出したトレース結果を表示します。

トレース結果は、サンプリング周期ごとのビットデバイスの ON/OFF 状態、ワードデバイスの現在値を表示します。

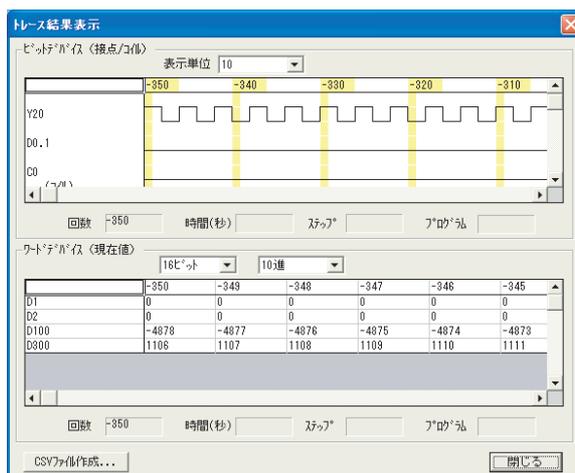


図 6.64 トレース結果

Point

トリガポイント設定で設定した条件（トリガ条件）成立で指定されたデバイスの内容を取り込みます。このためスキャン毎にサンプリングを行い、周辺のトリガ操作でサンプリングを終了させる場合は、トリガ条件成立とサンプリングが同一タイミングになるため、2回データの収集を行います。

トリガ条件成立時のデータ サンプリングによるデータ

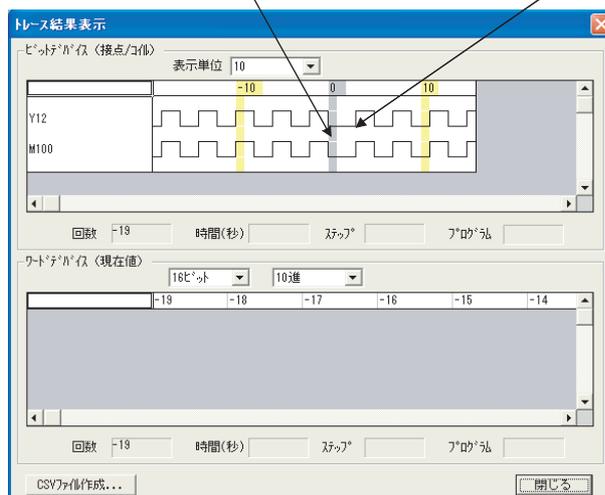


図 6.65 トレース結果

(6) トレース実行状態のクリア方法

トレース実行状態は、RESET/L.CLR スイッチによるラッチクリアまたはリモートラッチクリアでクリアされます。(図 6.6.4 項)

ラッチクリア後、再度サンプリングトレースを行う場合は“トレース開始”または“トレース登録”を選択してから、実行してください。

(7) 注意事項

(a) サンプリングトレースの実施可能箇所

ネットワーク上の他局、またはシリアルコミュニケーションユニットからもサンプリングトレースを実行することができます。

ただし、複数の箇所から同時に実施することはできません。

CPU ユニットで一度に実行できるのは 1 箇所からのみです。

(b) トレース設定の保持とクリアについて

CPU ユニットに登録したトレースの設定（サンプリングトレースファイル）はラッチしています。

CPU ユニットの電源 OFF → ON またはリセット操作を行っても、登録時のトレース設定で再度サンプリングトレースを実行できます。

ただし、前回のトレース結果は読み出せません。

また、下記 1) ~ 4) の場合は、サンプリングトレースのトリガ条件が成立していても、トリガ条件として認識しないため、ラッチしているトレース設定がクリアされます。(SM800 (トレース準備) が OFF します。)

GX Developer からトレース設定を再登録してください。

- 1) 対象メモリで“標準 RAM”を選択時、標準 RAM 内のローカルデバイスのサイズが変化する設定 * 1 を行い、パラメータを CPU ユニットに書き込んだ後、下記操作を行った場合

- CPU ユニットの電源 OFF → ON
- リセット操作
- STOP 状態から RUN 状態にした

* 1：新規にローカルデバイスを作成した場合も含まれます。

- 2) 対象メモリで“標準 RAM”を選択時、サンプリングトレースファイルが破損している状態で、下記操作を行った場合

- CPU ユニットの電源 OFF → ON
- リセット操作

Point

ローカルデバイスのサイズが変化する設定時に、トレース結果を残しておきたい場合は、下記の手順でトレース結果をパソコンに保存できます。

- ① 図 6.6.3 の **トレース結果PC読出** ボタンをクリックして、トレース結果をパソコンに読み出します。
- ② 図 6.6.3 の **トレース結果** ボタンをクリックしてトレース結果を表示します。
- ③ 図 6.6.4 の **CSVファイル作成** ボタンをクリックしてトレース結果を CSV 形式で保存します。

- 3) 対象メモリで“メモリカード (RAM)” を選択時、サンプリングトレースファイルが登録されている SRAM カードが装着されていない状態で、下記操作を行った場合
 - CPU ユニットの電源 OFF → ON
 - リセット操作
- 4) 対象メモリで“メモリカード (RAM)” を選択時、サンプリングトレースファイルが破損している状態で、下記操作を行った場合
 - CPU ユニットの電源 OFF → ON
 - リセット操作

(c) STOP 状態時のトレース結果の読出しについて

CPU ユニットの STOP 中は、トレース結果の読出しができません。
トレース結果の読出しは、RUN 中に行ってください。

(d) トリガ条件成立時のサンプリングトレースの登録について

サンプリングトレースの登録時に、トリガポイントで設定したトリガ条件が成立していないようにしてください。
トリガ条件が成立していた場合は、サンプリングトレースの設定を登録することができません。

(e) トレース条件の詳細設定で指定デバイスにファイルレジスタを指定した場合

トレースポイント設定、トリガポイント設定の詳細設定で、指定デバイスにファイルレジスタを指定した場合は、トレース登録後、ファイルレジスタファイルおよびファイルレジスタのプロック No. を変更しないでください。
トレースデータが正常に収集できなくなる恐れがあります。

(f) トレースポイント設定について

トレースポイント設定を、時間毎に設定した場合、割り込み処理として実行するためサンプリング間隔と一回のサンプリング処理時間に注意してください。
一回のサンプリング処理時間が長いと、“WDT ERROR” が発生することがあります。

(g) サンプリングトレース中にサンプリングトレースを実行した場合

先に実行したサンプリングトレースは正常に続きます。
後から実行したサンプリングトレースは実行することができません。

(h) RUN 中書込みの実行について

サンプリングトレースと RUN 中書込みを同時に実行する場合は、下記のとおりになります。

1) RUN 中書込み実行中にサンプリングトレースを実行した場合

- トレースポイントまたはトリガポイントをステップ No. で指定している：
RUN 中書込みは正常に完了しますが、サンプリングトレースは実行されません。
- トレースポイントとトリガポイントをステップ No. 以外で指定している：
RUN 中書込み、サンプリングトレースともに実行します。

2) サンプリングトレース実行中に RUN 中書込みを実行した場合

- トレースポイントまたはトリガポイントをステップ No. で指定している：
サンプリングトレースは中断され、RUN 中書込みが完了します。
- トレースポイントとトリガポイントをステップ No. 以外で指定している：
RUN 中書込み、サンプリングトレースともに実行します。

6.15 複数の GX Developer からのデバッグ機能

(1) 複数の GX Developer からのデバッグ機能とは

CPU ユニット、シリアルコミュニケーションユニットなどに接続した複数の GX Developer からデバッグを実行する機能です。

工程別、機能別などにファイルを分割した場合、複数の GX Developer から別ファイルのデバッグを行う場合に使用できます。

(2) 機能説明

複数の GX Developer からのデバッグ機能で実行できる組合せを表 6.25 に示します。

表 6.25 デバッグ機能の組合せ

実行中の機能	後から実行した機能			
	モニタ	RUN 中書込み	実行時間計測	サンプリングトレース
モニタ	○* 1	○* 2	○	○
RUN 中書込み	○* 2	×* 3	×	×
スキャン測定	○	○	×	○
サンプリングトレース	○	×	○	×

○：同時に実行可能、×：1つの GX Developer からのみ実行可能

* 1：モニタ条件の設定（ 6.11.1 項）は1つの GX Developer からのみ有効のため、他の GX Developer ではモニタ条件の設定ができません。

* 2：モニタ条件を設定したモニタと RUN 中書込みの同時実行はできません。

* 3：複数の GX Developer から1つのファイルに RUN 中書込みする方法については、6.15.2 項を参照してください。

6.15.1 複数の GX Developer から同時にモニタする機能

(1) 複数の GX Developer から同時にモニタする機能とは

CPU ユニット、シリアルコミュニケーションユニットなどに接続した複数の GX Developer から同時にモニタを実行できます。

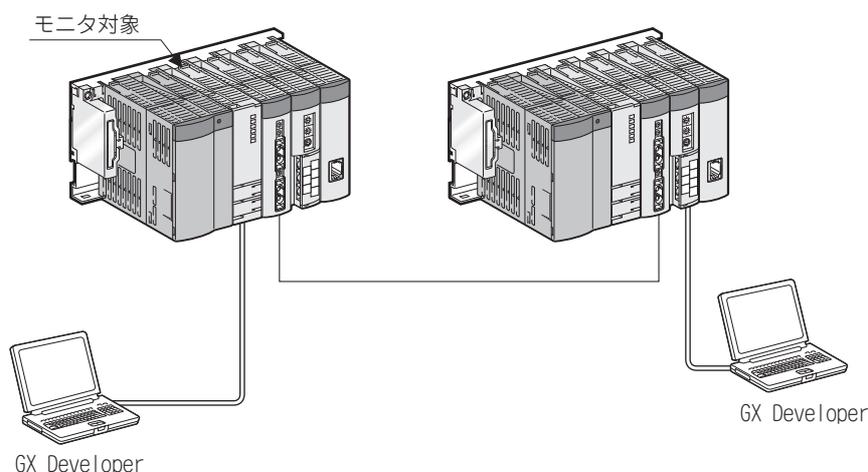


図 6.66 同時モニタ

ユーザ設定のシステムエリアを作成することにより、複数の GX Developer からの高速モニタができます。（自局用モニタファイルの設定は不要です。）

(2) 複数の GX Developer から同時にモニタする場合の設定

下記の手順でユーザ設定のシステムファイルを作成します。

- GX Developerの[オンライン]→[PCメモリフォーマット]を選択すると、図6.67の画面が表示されます。
- 対象メモリで“プログラムメモリ/デバイスメモリ”を選択します。
- フォーマット形式で“ユーザ設定のシステムファイルを作成する”に設定します。
- システムエリアのステップ数（1kステップ単位）を設定します。

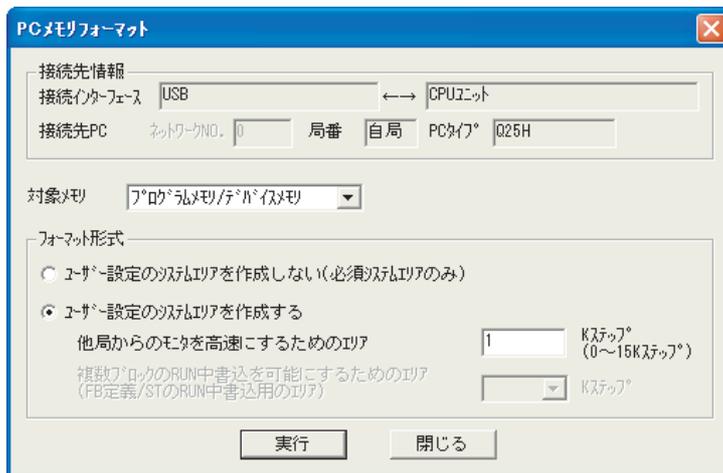


図 6.67 システムエリアの設定（1k ステップに設定した場合）

システムエリアとして設定可能な最大ステップ数は、表 6.26 のとおりです。
他局からのモニタファイル 1 つにつき 1k ステップだけ対応します。

表 6.26 システムエリアとして設定可能な最大ステップ数

CPU ユニット	設定可能な最大ステップ数	他局からのモニタ用システムエリア
ベーシックモデル QCPU	最大 3k ステップ	最大 3 個
ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU	最大 15k ステップ	最大 15 個

(3) 注意事項

(a) モニタ条件の設定について

モニタ条件設定は、1箇所からのみ可能です。  注 6.16

(b) システムエリアの設定要否について

ユーザ設定のシステムエリアを作成しなくても他局からの同時モニタはできますが、モニタ速度は遅くなります。

システムエリアはプログラムメモリに設定するため、システムエリアを設定した分だけプログラムの格納エリアが減少します。

(c) 設定によって高速化できるモニタ数

CPU ユニット 1 台に対して同時に高速化できるモニタ数は、“ユーザ設定のシステムエリア数 (k ステップ数) + 1” となります。

例えば、ユーザ設定のシステムエリアを 15k ステップ作成した場合は、CPU ユニット 1 台に対して同時に 16 箇所からのモニタを高速化できます。

6.15.2 複数の GX Developer から RUN 中書込みする機能 注 6.17

(1) 複数の GX Developer から RUN 中書込みする機能とは

複数の GX Developer から 1 つのファイルまたは別のファイルに RUN 中書き込みする機能です。

- 1 つのファイルへの RUN 中書き込み：
 - “ポインタを使用した相対的な RUN 中書き込み” になります。
- 別のファイルへの RUN 中書き込み：
 - “ポインタを使用した相対的な RUN 中書き込み” を使用しなくても実行できます。

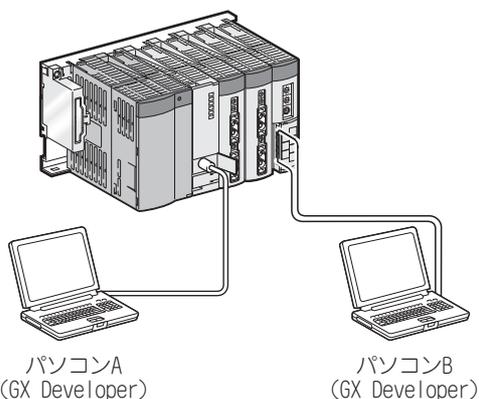


図 6.68 複数の GX Developer からの RUN 中書き込み

注 6.16 Basic

ベーシックモデル QCPU には、モニタ条件を設定したモニタが使用できません。

注 6.17 Basic

ベーシックモデル QCPU で複数の GX Developer から 1 つのファイルに RUN 中書き込みする場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。(付 2.1)

(2) 1つのファイルに RUN 中書込みする場合の操作手順

GX Developer の [ツール] → [オプション] の《プログラム共通》タブを選択します。
あらかじめ RUN 中書込み用のポインタを設定しておいてください。

(a) RUN 中書込設定および RUN 中書込方式の設定

RUN 中書込設定および RUN 中書込方式を下記のとおり設定します。

- 1) RUN 中書込設定で “変換後、PC に RUN 中書込する” を選択します。
- 2) RUN 中書込方式で “通常の RUN 中書込” または “ポインタによる相対的な RUN 中書込” を選択します。

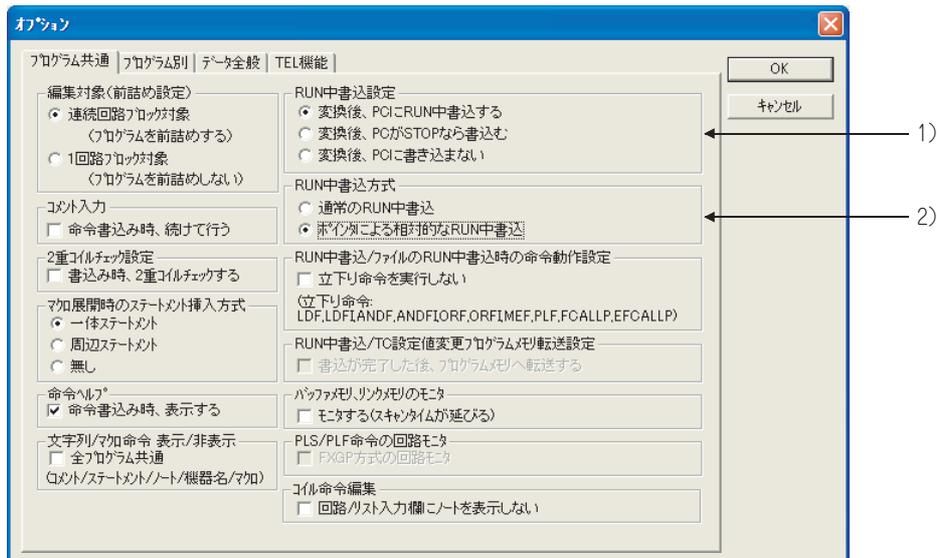


図 6.69 オプション画面

(b) RUN 中書込みの実行

指定されたポインタの回路を表示し、変更した回路を RUN 中書き込みます。

(3) 注意事項

複数の GX Developer から RUN 中書込みする場合の注意事項は、通常の RUN 中書込み時の注意事項と同様です。(☞ 6.12.3 項)

6.16 ウォッチドッグタイマ (WDT)

(1) ウォッチドッグタイマ (WDT) とは

CPU ユニットのハードウェアやシーケンスプログラムの異常を検出するための CPU ユニット内部のタイマです。

(2) ウォッチドッグタイマの設定とリセット

(a) ウォッチドッグタイマの設定

ウォッチドッグタイマの設定時間は、PC パラメータの PC RAS 設定で変更できます。

デフォルト値は、200ms に設定されています。

設定範囲は、10～2000ms (10ms 単位) です。

(b) ウォッチドッグタイマのリセット

CPU ユニットは、END 処理実行中にウォッチドッグタイマをリセットします。

- CPU ユニットが正常に動作し、ウォッチドッグタイマの設定値以内に END/FEND 命令を実行している場合には、ウォッチドッグタイマがタイムアップすることはありません。
- CPU ユニットのハードウェア異常や割込み/定周期実行タイププログラムの実行などでシーケンスプログラムのスキャンタイムが延び、ウォッチドッグタイマの設定値以内に END/FEND 命令を実行できなかった場合には、ウォッチドッグタイマがタイムアップします。

(3) ウォッチドッグタイマがタイムアップした場合

“WDT ERROR” となり、CPU ユニットは下記のようになります。

- 1) 出力をすべて OFF にします。
- 2) 前面の RUN LED が消灯し、ERR.LED が点滅します。
- 3) SM1 が ON し、SD0 にエラーコード 5000, 5001 が格納されます。

(4) 注意事項

(a) ウォッチドッグタイマの誤差

0～10ms の範囲内で誤差が生じます。

ウォッチドッグタイマを設定する場合は、上記の誤差を考慮した値にしてください。

(b) FOR ~ NEXT 命令でプログラムを繰り返し実行する際のウォッチドッグタイマのリセットについて

ウォッチドッグタイマは、シーケンスプログラムで WDT 命令を実行することによりリセットすることもできます。

FOR 命令と NEXT 命令でプログラムを繰り返し実行している場合にウォッチドッグタイマがタイムアップするときは、WDT 命令でウォッチドッグタイマをリセットします。

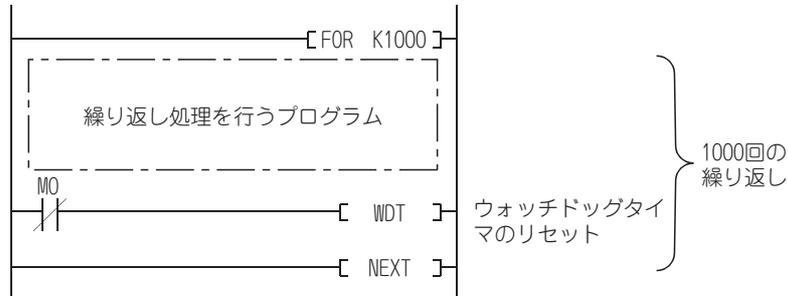


図 6.70 FOR ~ NEXT 命令実行時のウォッチドッグタイマのリセット

(c) WDT 命令を使用した際のスキャンタイムについて

シーケンスプログラムでウォッチドッグタイマをリセットしても、スキャンタイムの値はリセットしません。スキャンタイムは、END 命令まで計測した値になります。

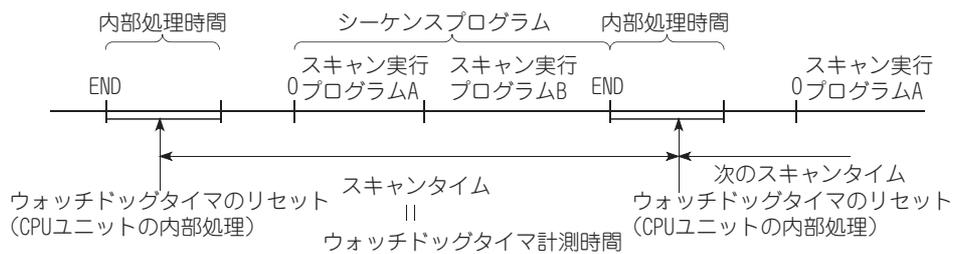


図 6.71 ウォッチドッグタイマのリセット

Point

- スキャンタイムは、CPU ユニットがシーケンスプログラムをステップ 0 から演算を実行し、再度同一ファイル名のシーケンスプログラムのステップ 0 を実行するまでの時間です。スキャンタイムは毎スキャン同一ではなく、下記により異なります。
 - 使用している命令の実行/非実行
 - 割込みプログラム、定周期実行タイププログラムの実行/非実行
- 毎スキャン同一のスキャンタイムで実行させる場合は、コンスタントスキャン機能 (6.2 節) を使用してください。

6.17 自己診断機能

(1) 自己診断機能とは

CPU ユニット自身で異常の有無の診断を行う機能です。
CPU ユニットの誤動作を防ぐとともに予防保全を目的としています。

(2) 自己診断のタイミング

CPU ユニットの電源投入時または CPU ユニットの RUN / STOP 中に異常が発生した場合、自己診断機能により異常を検出してエラーを表示し、CPU ユニットの演算停止などを行います。

ただし、異常の発生状態や実行する命令によっては、自己診断機能による異常検出ができない場合があります。自己診断による演算停止が行われない場合も、システム全体が安全側に働くよう、シーケンサの外部で安全回路を設けてください。

(3) 異常の確認

(a) LED の点灯

CPU ユニットは異常を検出した場合、ERR.LED の点灯などを行います。

(b) 異常内容の格納先と確認

CPU ユニットは異常を検出すると、特殊リレー (SM0, SM1) が ON し、異常の内容 (エラーコード) を特殊レジスタ (SD0) に格納します。

異常を複数検出したときは、最新エラーコードが SD0 に格納されます。

特殊リレー、特殊レジスタをプログラム上で使用して、シーケンサまたは機械系のインタロックにしてください。

Point

- 二重化 CPU の場合、他系で発生した異常の内容が特殊リレー (SM1610 ~ 1626)、および特殊レジスタ (SD1610 ~ 1636) に格納されます。
ただし下記の場合は、他系の異常内容は格納されません。
 - ・ 他系が電源 OFF、リセット、ハードウェア故障中
 - ・ “WDT ERROR” (エラーコード: 5000, 5001) 発生中
 - ・ トラッキングケーブル異常 (未装着, 断線, 故障など)
- CPU ユニットの状態を表すエラーは、特殊リレー (SM0, SM1) および特殊レジスタ (SD0 ~ 26) に異常の内容は格納されません。
また、ERR.LED も点灯しません。
エラー内容は、エラー履歴 (6.18 節) に格納されます。
 - ・ 待機系→制御系切替え発生 … “CONTROL EXE” (エラーコード: 6200)
 - ・ 制御系→待機系切替え発生 … “STANDBY” (エラーコード: 6210)

(4) 故障の履歴の確認

最新エラーコードは、GX Developer の [診断] → [PC 診断] の “エラー履歴” で確認できます。

エラー履歴は、シーケンサの電源を OFF してもバッテリーによりバックアップしています。

(5) 異常検出時の CPU ユニットの動作

(a) 異常検出時のモード

自己診断により異常を検出した場合、CPU ユニットの動作には、下記に示す 2 種類のモードがあります。

1) CPU ユニットの演算を停止するモード

異常を検出した時点で演算を停止し、PC パラメータの I/O 割付設定の詳細設定で “エラー時の出力モード” を “クリア” に設定したユニットの外部出力をすべて OFF します。(デバイスメモリの出力 (Y) は保持します。)

ただし、“エラー時の出力モード” を “保持” に設定したユニットは、外部出力が保持されます。(デバイスメモリ上の出力 (Y) は保持します。)

2) CPU ユニットの演算を続行するモード

異常を検出すると異常が発生したプログラム (命令) 以外のプログラムを実行します。

(b) 演算の続行／停止が選択可能なエラー

次のエラーは、演算の “続行／停止” の選択ができます。

1) PC パラメータの PC RAS 設定にて選択可能なエラー

- 演算エラー (SFC プログラム含む)
- 拡張命令エラー (将来拡張用の設定)
- ヒューズ断
- ユニット照合エラー
- インテリユニットプログラム実行エラー
- ファイルアクセスエラー  注 6.18
- メモリカード操作エラー  注 6.18
- 外部電源供給 OFF (将来拡張用の設定)  注 6.18

例えば、ユニット照合エラーを “続行” に設定した場合、エラーが発生するとエラー前の入出力番号にて演算を続行します。

エラーの詳細については、自己診断一覧を参照してください。( 本項)

2) PC パラメータの I/O 割付設定の詳細設定にて選択可能なエラー

- インテリジェント機能ユニットエラー

(6) エラーチェックの選択

次のエラーチェックは PC パラメータの PC RAS 設定により、エラーチェックを “行う／行わない” の選択ができます。(パラメータのデフォルトは、すべてチェックを “行う” に設定されています。)

- 1) バッテリチェック
- 2) ヒューズ断チェック
- 3) ユニット照合

注 6.18 **Basic**

ベーシックモデル QCPU では、ファイルアクセスエラー、メモリカード操作エラー、外部電源供給 OFF は設定できません。

(7) 自己診断一覧

CPU ユニットが行う自己診断の一覧を示します。

表 6.27 に示す “エラーメッセージ” 欄のエラーメッセージは、GX Developer の [診断] → [PC 診断] 確認できます。

表 6.27 自己診断一覧

診断内容	エラーメッセージ	診断タイミング	CPU ユニットの状態	LED の状態		ベーシックモデル QCPU	ハイパフォーマンスモデル QCPU	プロセス CPU	二重化 CPU	
				RUN	ERR.					
CPU 異常	MAIN CPU DOWN	・常時	停止	消灯	点滅	○	○	○	○	
END 命令実行せず	END NOT EXECUTE	・END 命令実行時	停止	消灯	点滅	○	○	○	○	
SFC プログラム実行エラー	SFCP. END ERROR	・SFC プログラム実行時	停止	消灯	点滅	○	×	○	×	
RAM チェック	RAM ERROR	・電源 ON 時およびリセット時	停止	消灯	点滅	○	○	○	○	
トラッキングメモリおよびトラッキングハードウェア異常	TRK. CIR. ERROR	・電源 ON 時およびリセット時 ・稼動中	停止	消灯	点滅	×	×	×	○	
演算回路チェック	OPE.CIRCUIT ERR.	・電源 ON 時およびリセット時 ・END 命令実行時	停止	消灯	点滅	○	○	○	○	
ヒューズ断*1*2	FUSE BREAK OFF	・常時	停止/ 続行	消灯/ 点灯	点滅/ 点灯	○	○	○	○	
I/O 割込みエラー	I/O INT. ERROR	・割込み発生時	停止	消灯	点滅	○	○	○	○	
ハードウェア異常	インテリジェント機能ユニットエラー*1	・電源 ON 時およびリセット時 ・FROM/TO 命令実行時 ・インテリジェント機能ユニット専用命令実行時 ・END 命令実行時	停止/ 続行	消灯/ 点灯	点滅/ 点灯	○	○	○	○	
	コントロールバスエラー	・電源 ON 時 ・END 処理実行時 ・FROM/TO 命令実行時 ・インテリジェント機能ユニット専用命令実行時 ・常時	停止	消灯	点滅	○	○	○	○	
	瞬停発生	AC/DC DOWN	・常時	続行	点灯	消灯	○	○	○	
	二重化ベースユニットの電源電圧低下	SINGLE PS. DOWN	・常時	続行	点灯	点灯	×	○*3	○*3	○
	二重化電源ユニット異常	SINGLE PS. ERROR	・常時	続行	点灯	点灯	×	○*3	○*3	○
	バッテリー低下*2	BATTERY ERROR	・常時	続行	点灯	BAT. ALM LED 点灯	○	○	○	○

○：自己診断を行う、×：自己診断を行わない

(次ページへつづく)

表 6.27 自己診断一覧 (つづき)

診断内容	エラー メッセージ	診断タイミング	CPU ユニット の状態	LED の状態		ベーシック モデル QCPU	ハイパ フォーマン スモデル QCPU	プロセス CPU	二重化 CPU
				RUN	ERR.				
ユニット照合 *1*2	UNIT VERIFY ERR.	• END 命令実行時	停止/ 続行	消灯/ 点灯	点滅/ 点灯	○	○	○	○
ベース割付けエラー	BASE LAY ERROR	• 電源 ON 時または リセット時	停止	消灯	点滅	○	×	×	○
インテリジェント 機能ユニット 割付けエラー	SP.UNIT LAY ERR.	• 電源 ON 時または リセット時 • STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○
インテリジェント プログラム実行 エラー*1	SP.UNIT ERROR	• FROM/TO 命令実 行時	停止/ 続行	消灯/ 点灯	点滅/ 点灯	○	○*4	○	○
インテリジェント 機能ユニット バージョンエラー	SP.UNIT VER.ERR	• 電源 ON 時または リセット時	停止	消灯	点滅	○*4	×	○	○
パラメータなし	MISSING PARA.	• 電源 ON 時または リセット時 • STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○
ブートエラー	BOOT ERROR	• 電源 ON 時または リセット時	停止	消灯	点滅	○	○	○	○
メモリカード操作 エラー*1	ICM.OPE. ERROR	• メモリカード着脱 時	停止/ 続行	消灯/ 点灯	点滅/ 点灯	×	○	○	○
ファイルセット エラー	FILE SET ERROR	• 電源 ON 時または リセット時 • PC 書込み時	停止	消灯	点滅	○	○	○	○
ファイルアクセス エラー*1	FILE OPE. ERROR	• 命令実行時	停止/ 続行	消灯/ 点灯	点滅/ 点灯	×	○	○	○
命令実行不能	CAN'T EXE.PRG.	• 電源 ON 時または リセット時 • STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○
パラメータ設定 チェック	PARAMETER ERROR	• 電源 ON 時または リセット時 • STOPからRUNに 切換え時 • PC 書込み時	停止	消灯	点滅	○	○	○	○
リンクパラメータ エラー	LINK PARA.ERROR	• 電源 ON 時または リセット時 • STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○
SFC パラメータ エラー	SFC PARA.ERROR	• STOPからRUNに 切換え時 • PC 書込み時	停止	消灯	点滅	○	○	○	○
インテリパラメータ エラー	SP.PARA. ERROR	• 電源 ON 時または リセット時	停止	消灯	点滅	○	○	○	○

○：自己診断を行う， ×：自己診断を行わない

(次ページへつづく)

表 6.27 自己診断一覧 (つづき)

診断内容	エラー メッセージ	診断タイミング	CPU ユニット の状態	LED の状態		ベーシック モデル QCPU	ハイパ フォー マン スモ デル QCPU	プロセ ス CPU	二重 化 CPU	
				RUN	ERR.					
パスワード異常	REMOTE PASS.ERR	・電源 ON 時または リセット時 ・STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○	
命令コードチェック	INSTRUCT. CODE ERR	・電源 ON 時または リセット時 ・STOPからRUNに 切換え時 ・命令実行時	停止	消灯	点滅	○	○	○	○	
END 命令なし	MISSING END INS.	・電源 ON 時または リセット時 ・STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○	
ポインタ設定エラー	CAN'T SET(P)	・電源 ON 時または リセット時 ・STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○	
	CAN'T SET(I)	・電源 ON 時または リセット時 ・STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○	
プログラム異常	演算エラー * 1 * 3	OPERATION ERROR	・命令実行時	停止/ 続行	消灯/ 点灯	点滅/ 点灯	○	○	○	○
	FOR ~ NEXT 命令 構成エラー	FOR NEXT ERROR	・命令実行時	停止	消灯	点滅	○	○	○	○
	CALL ~ RET 命令 構成エラー	CAN'T EXECUTE(P)	・命令実行時	停止	消灯	点滅	○	○	○	○
	割込みプログラム エラー	CAN'T EXECUTE(I)	・命令実行時	停止	消灯	点滅	○	○	○	○
	命令実行不能	INST. FORMAT ERR.	・命令実行時	停止	消灯	点滅	○	○	○	○
	SFC プログラム 構成エラー	SFCP.CODE ERROR	・STOPからRUNに 切換え時	停止	消灯	点滅	×	○	○	○
	SFC ブロック 構成エラー	CAN'T SET(BL)	・STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○
	SFC ステップ 構成エラー	CAN'T SET(S)	・STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○
	SFC 実行エラー	SFC EXE. ERROR	・STOPからRUNに 切換え時	停止	消灯	点滅	○	×	×	×
	SFC 構文エラー	SFCP. FORMAT ERR.	・STOPからRUNに 切換え時	停止	消灯	点滅	○	○	○	○
	SFC 演算チェック エラー* 1	SFCP.OPE. ERROR	・命令実行時	停止/ 続行	消灯/ 点灯	点滅/ 点灯	×	○	○	○
	SFC プログラム 実行エラー	SFCP.EXE. ERROR	・STOPからRUNに 切換え時	続行	点灯	点灯	×	○	○	○
	SFC ブロック 実行エラー	BLOCK EXE.ERROR	・命令実行時	停止	消灯	点滅	○	○	○	○
	SFC ステップ 実行エラー	STEP EXE.ERROR	・命令実行時	停止	消灯	点滅	○	○	○	○

○：自己診断を行う， ×：自己診断を行わない

(次ページへつづき)

表 6.27 自己診断一覧 (つづき)

診断内容	エラーメッセージ	診断タイミング	CPU ユニット の状態	LED の状態		ベーシック モデル QCPU	ハイパ フォーマン スモデル QCPU	プロセス CPU	二重化 CPU	
				RUN	ERR.					
CPU 異常	演算渋滞監視	WDT ERROR	・常時	停止	消灯	点滅	○	○	○	○
	プログラム タイムオーバー	PRG.TIME OVER	・常時	続行	点灯	点灯	○	○	○	○
二重化 システム異常	プログラム、 パラメータ、 ディップスイッチ 不一致	FILE DIFF.	・常時 ・電源 ON 時または リセット時 ・トラッキングケー ブル装着時 ・バックアップモー ド変更時 ・RUN 中書き込み ・系切替え時 ・両系 RUN へ移行 時	停止	消灯	点滅	×	×	×	○
	運転状態、キース スイッチ不一致	OPE.MODE DIFF.	・電源 ON 時または リセット時 ・常時	続行/停 止	消灯 /点 灯	点滅/ 点灯	×	×	×	○
	ユニット装着構成 不一致	UNIT LAY. DIFF.	・常時 ・電源 ON 時または リセット時 ・トラッキングケー ブル装着時 ・運転モード変更時	停止	消灯	点滅	×	×	×	○
	メモ리카ード装着 状態不一致	CARD TYPE DIFF.	・電源 ON 時または リセット時	停止	消灯	点滅	×	×	×	○
	現モード時機能実 行不可	CAN'T EXE. MODE	・常時	続行	点灯	点灯	×	×	×	○
	両系ファイル 同一性異常、パラ メータ有効ドライ ブ同一性異常	CPU MODE DIFF.	・電源 ON 時または リセット時 ・END 命令実行時 ・トラッキングケー ブル装着時	停止	消灯	点滅	×	×	×	○
	トラッキングデー タ通信エラー	TRK.TRANS. ERR.	・常時	続行	点灯	点灯	×	×	×	○
	トラッキング容量 超過エラー	TRK.SIZE ERROR	・END 命令実行時	続行	点灯	点灯	×	×	×	○
	トラッキングケー ブル異常、トラッ キング転送用ハー ドウェア故障	TRK.CABLE ERR.	・電源 ON 時または リセット時	停止	消灯	点滅	×	×	×	○
	トラッキングケー ブル未接続・故障、 トラッキング転送 用ハードウェア故 障	TRK. DISCONNECT	・常時	続行	点灯	点灯	×	×	×	○
	トラッキング転送 初期エラー	TRK.INIT. ERROR	・電源 ON 時または リセット時	停止	消灯	点滅	×	×	×	○
	待機系→制御系切 替え発生*5	CONTROL EXE.	・常時	続行	点灯	消灯	×	×	×	○
	制御系→待機系切 替え発生*5	STANDBY	・常時	続行	点灯	消灯	×	×	×	○

○：自己診断を行う， ×：自己診断を行わない

(次ページへつづく)

表 6.27 自己診断一覧 (つづき)

診断内容	エラーメッセージ	診断タイミング	CPUユニットの状態	LEDの状態		ベーシックモデル QCPU	ハイパフォーマンスモデル QCPU	プロセス CPU	二重化 CPU	
				RUN	ERR.					
二重化システム異常	系切替えエラー	CAN'T SWITCH	・系切替え実行時	続行	点灯	点灯	×	×	×	○
	待機系未起動/停止エラー	STANDBY SYS.DOWN	・常時	続行	点灯	点灯	×	×	×	○
	制御系未起動/停止エラー	CONTROL SYS.DOWN	・常時	停止	消灯	点滅	×	×	×	○
	プログラムメモリクリア実施	PRG.MEM. CLEAR	・プログラムメモリコピー機能実行時	停止	消灯	点滅	×	×	×	○
	メモリコピー機能実施	MEM.COPY EXE.	・メモリコピー機能実施時	続行	点灯	点灯	×	×	×	○
	トラッキング設定パラメータエラー	TRK.PARA. ERROR	・電源 ON 時またはリセット時	停止	消灯	点滅	×	×	×	○
	マルチ CPU システム構成不可	CPU LAY ERROR	・電源 ON 時またはリセット時	停止	消灯	点滅	×	×	×	○
マルチ CPU システム異常	他号機 CPU 重度異常	MULTI CPU DOWN	・常時 ・電源 ON 時またはリセット時	停止	消灯	点滅	○	○	○	×
	マルチ CPU 実行エラー	MULTI EXE.ERROR	・電源 ON 時またはリセット時	停止	消灯	点滅	○	○	○	×
	マルチ CPU 整合性異常	CPU LAY.ERROR	・電源 ON 時またはリセット時	停止	消灯	点滅	○	○	○	×
	他号機 CPU 軽度異常	MULTI CPU ERROR	・常時	続行	点灯	点灯	○	○	○	×
アナンシェータチェック	F****	・命令実行時	続行	点灯	USER LED 点灯	○	○	○	○	
CHK 命令チェック	<CHK>ERR ****_****	・命令実行時	続行	点灯	USER LED 点灯	×	○	○	○	
ブート OK	BOOT OK	・電源 ON 時またはリセット時	停止	消灯	点滅	×	○	○	○	

○：自己診断を行う， ×：自己診断を行わない

- * 1：GX Developer によるパラメータ設定で“続行”に設定可能です。(デフォルトは“停止”です。)
- * 2：GX Developer によるパラメータ設定で“チェックしない”に設定可能です。(デフォルトは“チェックする”です。)
- * 3：マルチ CPU システム構成時は、すべての CPU ユニットのシリアル No. の上 5 桁が“07032”以降の場合に 1 号機の CPU ユニットでのみ検出します。
- * 4：機能バージョン B 以降の CPU ユニットが対象です。
- * 5：CPU ユニットの状態を表す診断のため、GX Developer の PC 診断画面の“現在のエラー”欄にエラーメッセージは表示されません。
本エラーは、エラー履歴欄にのみ表示します。

Point

二重化 CPU の場合、他系で発生した異常の内容が特殊リレー (SM1610 ~ 1626)、および特殊レジスタ (SD1610 ~ 1636) に格納されます。

ただし下記の場合は、他系の異常内容は格納されません。

- ・他系が電源 OFF、リセット、ハードウェア故障中
- ・“WDT ERROR” (エラーコード：5000, 5001) 発生中
- ・トラッキングケーブル異常 (未装着, 断線, 故障など)

6.17.1 エラー発生による割込み 注 6.19

CPU ユニットは、エラー発生により対象となる割込みポイントの割込みプログラムを実行できます。

(1) PC RAS 設定で “ 続行 ” / “ 停止 ” に設定可能なエラーによる割込み

PC パラメータの PC RAS 設定で演算を “ 続行 ” / “ 停止 ” に設定できるエラーは、“ 続行 ” に設定したエラーのみ実行します。

“ 停止 ” に設定したエラーは、停止エラー全体の割込みプログラム (I32) を実行します。

(2) 割込みポイントと対応するエラー 注 6.20

割込みポイントと対応するエラーを図 6.72 に示します。

I32	停止エラー全体 ^{*1}
I33	SINGLE PS.DOWN ^{*2}
I34	UNIT VERIFY ERR. FUZE BREAK OFF SP.UNIT ERROR MULTI CPU ERROR
I35	OPERATION ERROR SFCP OPE.ERROR SFCP EXE.ERROR EX.POWER OFF
I36	ICM.OPE.ERROR FILE OPE.ERROR
I37	アキ
I38	PRG.TIME OVER
I39	CHK命令 アナンシェータ検出
I40	CAN'T SWITCH
I41	STANDBY

エラー発生時の運転モードが続行のエラー、または “ 停止 / 続行 ” の選択ができるエラーで “ 続行 ” に設定されているエラー

図 6.72 エラー発生割込みによる割込みポイント一覧

- * 1：下記の重度エラーが発生したときは、I32 の割込みプログラムは実行されません。
 - ・ MAIN CPU DOWN
 - ・ END NOT EXECUTE
 - ・ RAM ERROR
 - ・ OPE CIRCUIT ERR.
- * 2：ハイパフォーマンスモデル QCPU またはプロセス CPU でマルチ CPU システム構成時は、1 号機の CPU ユニットでのみ使用可能です。

注 6.19 **Basic**

ベーシックモデル QCPU では、エラー発生時の割込みポイントがないため、CPU ユニット自身のエラー発生による割込みはできません。

注 6.20 **High performance** **Process**

ハイパフォーマンスモデル QCPU、プロセス CPU では、割込みポイント I40、I41 は使用できません。また、I33 はシリアル No. の上 5 桁が “07032” 以降が対象です。

Point

割込みポインタ I32 ~ 41 は、電源立上げ時 / CPU ユニットのリセット時、実行禁止状態になっています。I32 ~ 41 を使用する場合は、IMASK 命令および EI 命令で実行許可状態にしてください。IMASK 命令と EI 命令の詳細については、下記マニュアルを参照してください。
 MELSEC-Q/L プログラミングマニュアル (共通命令編)

(3) 二重化 CPU 使用時の注意事項

(a) 割込みポインタ I41 の割込みプログラムを使用するときの注意事項

割込みポインタ I41 は、制御系→待機系切替え発生時の割込みポインタです。

I41 の割込みプログラムは、制御系→待機系への系切替え後に新待機系（系切替えで制御系から待機系になったシステム）で実行するため、下記の点に注意してください。

1) 特殊リレーの制御系判別フラグ・待機系判別フラグ (SM1515, SM1516)

制御系判別フラグ・待機系判別フラグは、待機系 (SM1515 : OFF, SM1516 : ON) になります。

2) トラッキング対象のデバイスを変更する場合

I41 の割込みプログラムでトラッキング対象のデバイスを変更するようにプログラミングしている場合、新待機系のデバイスは新制御系（系切替えで待機系から制御系になったシステム）のトラッキング転送により上書きされます。

I41 の割込みプログラムでトラッキング転送のデバイスを変更する場合は、下記のようにしてください。

- 二重化パラメータのトラッキング設定 ( 8.2 節 (2)) で、変更を行うデバイスをトラッキング転送の対象からはずす
- トラッキング対象のデバイスの内容を、別のデバイスに移して変更を加えるようにプログラミングする

3) 基本ベースユニットに装着されているユニットおよびネットワークユニットでの出力

I41 の割込みプログラムは新待機系で実行するため、下記に示す出力は行いません。

- 基本ベースユニットに装着されているユニットへの出力 (Y)
- MELSECNET/H リモート I/O ネットへの出力 (Y)
- MELSECNET/H リモート I/O ネット, MELSECNET/H PC 間ネットの他局へのリンクリレー (B), リンクレジスタ (W) の送信

6.17.2 エラー発生による LED 表示

エラーが発生時は、CPU ユニットの前面の LED が点灯 / 点滅します。( 6.21 節)

6.17.3 エラーの解除

CPU ユニットは、プログラムの演算を続行するエラーに限り、プログラムでエラーの解除操作を行うことができます。

(1) エラー解除手順

エラーの解除は、次の手順で行います。

- エラーの要因を解消します。
- 特殊レジスタ SD50 に解除するエラーコードを格納します。
- 特殊リレー SM50 を OFF → ON します。
- 対象エラーが解除されます。

(a) 複数エラーが発生しているときのエラー解除手順

最後に発生したエラー（特殊レジスタの SD0 に格納されているエラー）を解除すると、エラー情報の特殊リレー／レジスタ（SM0, SM1, SM5, SM16, SD0 ~ 26）の内容がクリアされるため、解除されていないエラーの情報を特殊リレー／レジスタから取得することができません。

解除されていないエラーは、エラー履歴（ 6.18 節）から過去に発生したエラーを取得し、解除してください。

(2) エラー解除後の状態

エラー解除で CPU ユニットの復帰した場合、エラーに関係する特殊リレー、特殊レジスタおよび LED はエラー発生前の状態に戻ります。

エラー解除を行ったあとに再度同じエラーが発生した場合、エラー履歴に再登録されます。

(3) アナンシェータの解除

複数検出したアナンシェータの解除は、最初に検出した F 番号のみ解除されます。（ 9.2.5 項）

Point

- 解除するエラーコードを SD50 に格納してエラー解除を行った場合、下 1 桁のコード番号は無視されます。

例 エラーコード 2410, 2411, 2412 が発生した場合、SD50 に 2412 を格納してエラーを解除するとエラーコード 2410, 2411 も解除されます。

- CPU ユニット以外の要因で発生しているエラーは、特殊リレー(SM50) および特殊レジスタ (SD50) によりエラー解除を行っても、エラー要因の解消はできません。

例 “SP. UNIT DOWN” は Q バス上で発生したエラーのため、特殊リレー (SM50)、特殊レジスタ (SD50) によりエラー解除を行っても、エラー要因は解消できません。

エラー要因を解消するには、下記マニュアルを参照してください。

 QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）

6.18 エラー履歴

CPU ユニットは、自己診断機能により検出した結果に検出時刻を付けて、エラー履歴としてメモリに格納します。エラー履歴は、GX Developer の [診断] → [PC 診断] で確認できます。

Point

検出時刻は、CPU ユニットの内蔵時計の時間を使用しますので、必ず最初に正確な時刻を設定してください。
(☞ 6.5 節)

6.18.1 ベーシックモデル QCPU

(1) 格納エリア

最新の故障 16 点分が、ラッチされているベーシックモデル QCPU のエラー履歴格納メモリに格納されます。

(2) 格納データ

CPU ユニットの電源 ON 中に同一エラーが複数回発生した場合、エラー履歴格納メモリへのデータ格納は 1 回のみ行います。

(3) エラー履歴のクリア方法

エラー履歴格納メモリのクリアは、GX Developer の [診断] → [PC 診断] の **履歴クリア** ボタンで行います。

エラー履歴クリアを行うと、ベーシックモデル QCPU のエラー履歴格納メモリのデータをすべてクリアします。

6.18.2 ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU

(1) 格納エリア

最新の故障 16 点分は、ラッチされている CPU ユニットのエラー履歴格納メモリに格納されます。

17 点以上格納する場合は、PC パラメータの PC RAS 設定によりメモ리카ード内のファイルへ格納できます。

表 6.28 エラー履歴ファイルの格納エリア

格納エリア	格納可能数
内蔵メモリおよび設定したメモ리카ード内のファイル	最大 100 個 (変更可) * 1

* 1 : 格納可能数を越えた場合、一番古い履歴を消して最新の履歴を格納します。

(2) 格納データ

下記の場合、メモ리카ードの履歴ファイルの内容をクリアしてから、CPU ユニットのエラー履歴格納メモリの 16 点分データを履歴ファイルへ転送します。

- パラメータの履歴ファイルの履歴数を途中で変更した場合
- 装着したメモ리카ードの履歴数がパラメータに設定されている履歴数と異なっている場合

Point

パラメータで設定したエラー履歴ファイルがメモ리카ードに存在していても、CPU ユニットはエラーになりません。この場合、CPU ユニットは発生した故障を CPU ユニットのエラー履歴格納メモリにのみ格納します。

(3) エラー履歴のクリア方法

エラー履歴格納メモリ/エラー履歴ファイルのクリアは、GX Developer の [診断] → [PC 診断] の

履歴クリア ボタンで行います。

エラー履歴クリアを行うと CPU ユニットのエラー履歴格納メモリのデータ、メモ리카ードのエラー履歴ファイルのデータをすべてクリアします。

6.19 システムプロテクト 注 6.21

CPU ユニットは、設計者以外の第3者からの GX Developer、シリアルコミュニケーションユニットなどによるプログラム変更に対していくつかの保護機能（システムプロテクト）を持っています。

表 6.29 システムプロテクトの種類

プロテクト対象	プロテクト有効ファイル	プロテクト内容	方法	有効タイミング	参照
CPU 全体* 1	すべてのファイル (デバイスにも有効)	GX Developer などからリモート操作による CPU ユニットの動作指示を一括禁止します。	CPU ユニットのディップスイッチ SW1 を ON にします。	常時	—
メモリカード単位* 1	すべてのファイル	メモリカードに対するライトプロテクトを行い、書き込みを禁止します。	メモリカードのライトプロテクトスイッチを ON にします。	常時	—
ファイル単位	<ul style="list-style-type: none"> プログラム デバイスコメント デバイス初期値 	ファイルごとに属性を変更します。 <ul style="list-style-type: none"> 読出／書込禁止 (プログラムの読出し、書き込みなどの操作) 書込禁止 (プログラムの書き込み、テストなどの書き込み処理がかかわる操作) 	パスワード登録で行います。	常時	6.19.1 項

* 1：ベーシックモデル QCPU では、メモリカードが使用できないため、CPU 全体およびメモリカード単位のシステムプロテクトはできません。

Point

CPU ユニットのディップスイッチ SW1 を ON し、システムプロテクトを行っている状態でも、PC パラメータ、CPU ユニットのディップスイッチで設定している下記機能は実行します。  注 6.21

- 標準 ROM、メモリカードからのブート
- 標準 ROM への自動書き込み

注 6.21 Basic

ベーシックモデル QCPU には、ディップスイッチによるシステムプロテクトはできません。

6.19.1 パスワード登録

パスワード登録は、GX Developer による CPU ユニット内のプログラム、デバイスコメントなどのデータの書込み／読出しを禁止するためのものです。

(1) パスワードの有効範囲

書込み／読出し禁止の範囲は、指定したメモリ（プログラムメモリ／標準 ROM /メモ리카ード）のプログラムファイル、デバイスコメントファイル、デバイス初期値ファイルが対象となります。  **注 6.22**

(2) パスワード登録により禁止できる操作

下記の 2 種類があります。

- ファイルの書込み／読出し
- ファイルの書込み

パスワード登録されている場合には、同じパスワードを入力しない限り、GX Developer からのファイルの操作は行えません。

(3) 設定方法

GX Developer の [オンライン] → [パスワード登録], または PC 書込画面の パスワード設定 ボタンをクリックして行います。

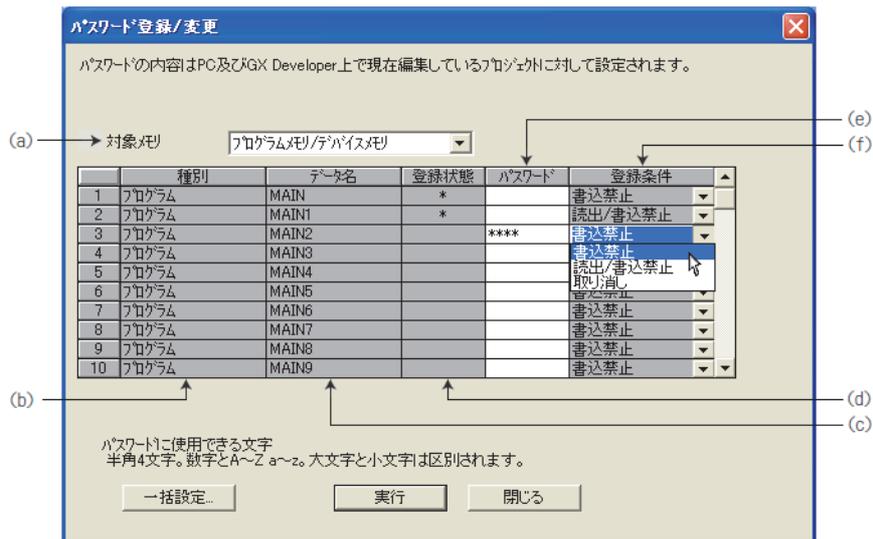


図 6.73 パスワード登録／変更画面

注 6.22 Basic

ベーシックモデル QCPU のパスワード有効範囲は、プログラムメモリのプログラムファイル、デバイスコメントファイル、デバイス初期値が対象となります。
標準 ROM、メモ리카ードは対象となりません。

(a) 対象メモリ

パスワードを登録するファイルが格納されているメモリを指定します。

(b) 種別

対象メモリに格納されているファイルの種別が表示されます。

(c) データ名

対象メモリに格納されているファイル名が表示されます。

(d) 登録状態

パスワードが登録されている場合は “*” が表示されます。

(e) パスワード

新規に登録するパスワードまたは現在設定されているパスワードを設定します。

パスワードを指定すると登録条件の設定ができます。

(f) 登録条件**1) 書込禁止**

パスワードを指定したファイルへの書込みを禁止します。(読出しはできます。)

2) 読出／書込禁止

パスワードを指定したファイルの書込み／読出しを禁止します。

3) 取り消し

設定したパスワードを取り消します。(現在設定されているパスワードの入力が必要です。)

(4) 注意事項

ファイルに登録したパスワードは、ファイルから読み出すことはできません。

登録したパスワードを忘れると、下記以外のファイルの操作ができなくなります。

(a) ベーシックモデル QCPU の場合

- プログラムメモリ : PC メモリフォーマット
- 標準 ROM : プログラムメモリの ROM 化

(b) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU の場合

- プログラムメモリ／メモリカード : PC メモリフォーマット
- 標準 ROM : 一括書込み

登録したパスワードは、書類などに記録して保管しておいてください。

6.19.2 リモートパスワード 注 6.23

(1) リモートパスワードとは

遠隔地のユーザから CPU ユニットへの不正なアクセスを防止するための機能です。

リモートパスワードが設定されていると、遠隔地のユーザから CPU ユニットへのアクセスがあった場合に、リモートパスワードのチェックを行います。

(2) 設定可能ユニットおよび設定枚数

リモートパスワードが設定可能なユニットとその設定枚数を表 6.30 に示します。

表 6.30 リモートパスワードが設定可能なユニットおよび設定枚数

設定可能なユニット	最大設定枚数
Ethernet ユニット	4 枚
シリアルコミュニケーションユニット	8 枚 * 1
モデムインタフェースユニット	

* 1 : Version 6 および 7 の GX Developer を使用した場合は、4 枚になります。

Point

- 上表の最大設定枚数は、リモートパスワードの設定枚数であり、CPU ユニットを使用したシステムでの最大装着枚数ではありません。
CPU ユニットを使用したシステムでの最大装着枚数については、下記マニュアルを参照してください。
 QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）
- 各ユニットでのリモートパスワードの詳細については、下記マニュアルを参照してください。
 各ユニットのマニュアル



注 6.23

Basic

High performance

ベーシックモデル QCPU、ハイパフォーマンスモデル QCPU でリモートパスワードを使用する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。（ 付 2.1、付 2.2）

(3) リモートパスワードの設定から反映までの流れ

リモートパスワード設定（ 本項 (5)）を設定して CPU ユニットに書き込みます。
 リモートパスワードは、CPU ユニットの電源 OFF → ON またはリセット解除時に、対象のユニット（ 本項 (2)）にリモートパスワードが転送されます。

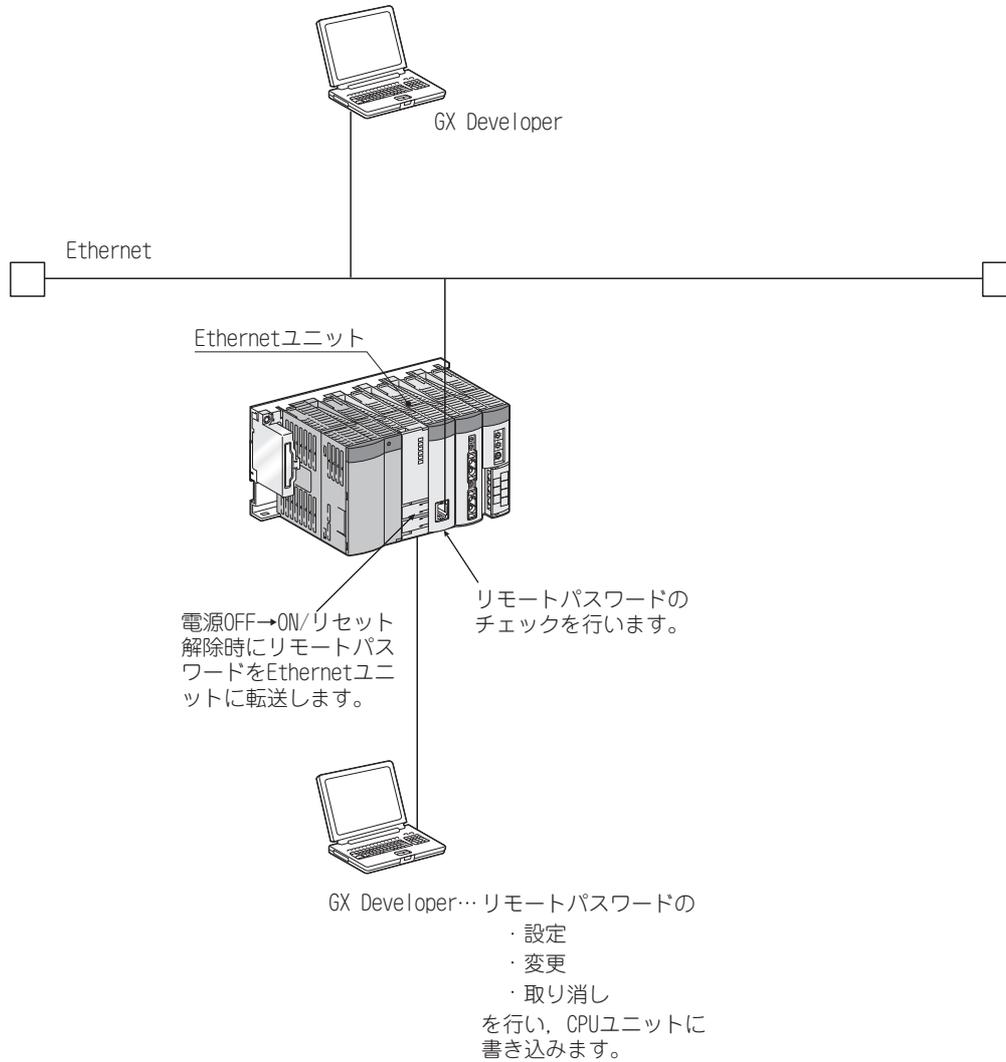


図 6.74 リモートパスワードの概略

(4) リモートパスワードのロック／アンロック処理

モデム経由でシリアルコミュニケーションユニット，または Ethernet 経由で Ethernet ユニットのリモートパスワードのアンロックを行います。

リモートパスワードが一致した場合は，CPU ユニットへのアクセスが可能になります。

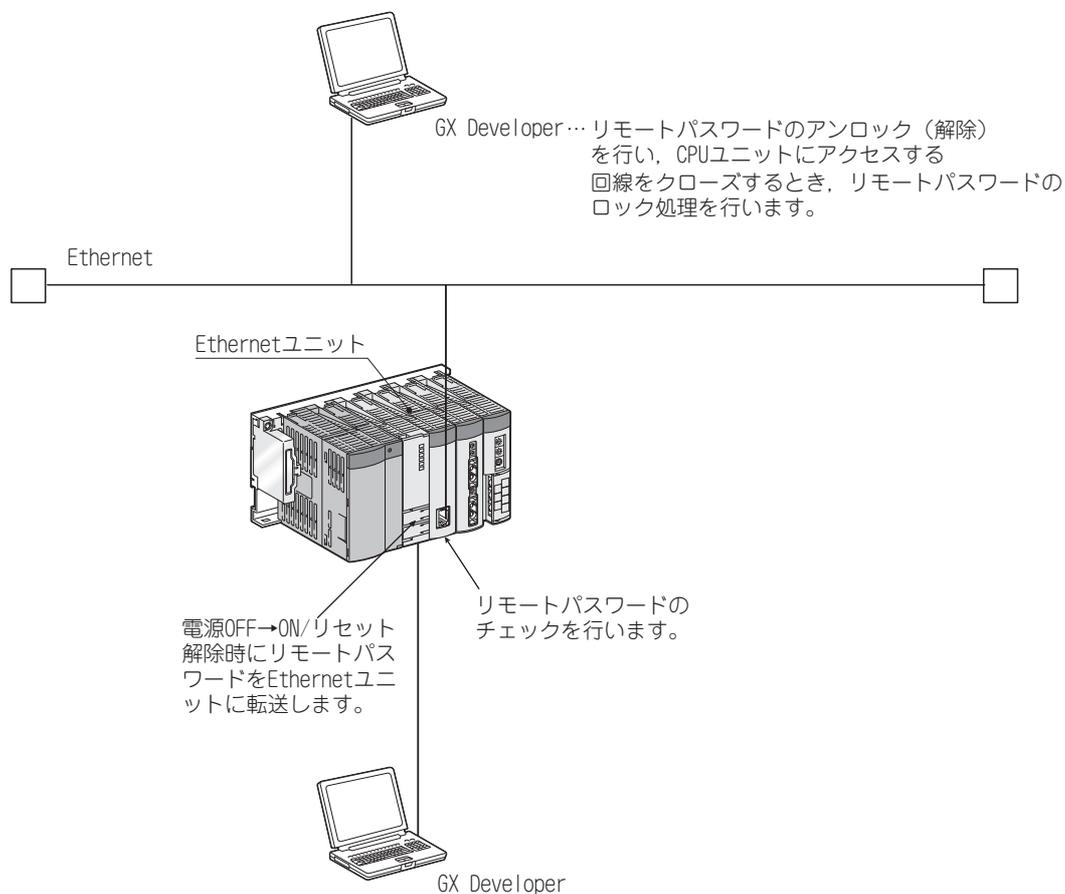


図 6.75 Ethernet ユニットでリモートパスワードのアンロック，ロック処理を行う場合の概略

(5) リモートパスワードの設定, 変更, 取り消し手順

(a) リモートパスワードの設定

- GX Developer のプロジェクトデータ一覧で, [パラメータ] → [リモートパスワード] を選択して行います。

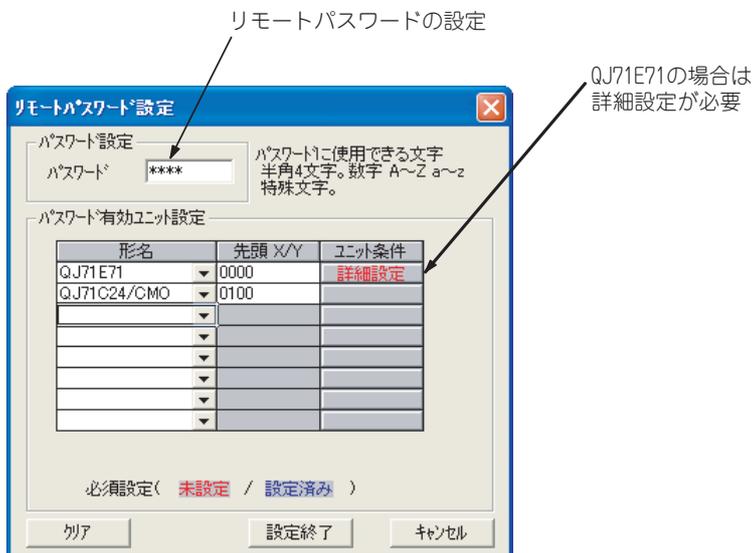


図 6.76 リモートパスワード設定画面

表 6.31 リモートパスワード設定画面の設定項目

項目		設定内容	設定範囲/選択肢
パスワード設定		リモートパスワード入力	4文字以内(英・数字, 特殊記号)
パスワード有効 ユニット設定	形名	形名を選択	<ul style="list-style-type: none"> • QJ71E71 • QJ71C24/CMO
	先頭 X/Y	ユニットの先頭アドレスを設定	<ul style="list-style-type: none"> • ベーシックモデル QCPU 使用時: 0000H ~ 03E0H • ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU 使用時: 0000H ~ 0FE0H
詳細設定		-	-
ユーザ用コネクション No. 有効設定		ユーザ用コネクション No. を設定	コネクション No.1 ~ コネクション No.16
システム用コネク ション有効設定	自動オープン UDP ポート	リモートパスワード有効ポートを チェック	-
	FTP 交信ポート (TCP/IP)		
	GX Developer 交信ポート (TCP/IP)		
	GX Developer 交信ポート (UDP/IP)		
HTTP ポート			

- GX Developer を CPU ユニットに接続します。
設定したリモートパスワードを CPU ユニットに書き込みます。
マルチ CPU システム構成時は, リモートパスワードを設定するユニットの管理 CPU に書き込みます。

Point

- リモートパスワードを設定した場合には、パラメータはプログラムメモリ（ドライブ0）に格納してください。
注 6.24
パラメータをプログラムメモリ（ドライブ0）以外に格納した場合は、リモートパスワード機能が正常に動作しません。
- ブート運転を行う場合には、パラメータファイルを標準 ROM またはメモリカードに格納し、PC パラメータのブートファイル設定でパラメータファイルがプログラムメモリに転送されるように設定してください。**注 6.24**
このとき、ディップスイッチのパラメータ有効ドライブは、パラメータファイルを格納した標準 ROM またはメモリカードにしてください。（ディップスイッチのパラメータ有効ドライブをプログラムメモリにすると、ブート運転が正常に動作しません。）

(b) リモートパスワードの変更

リモートパスワード設定画面でパスワードを変更して、変更したパスワードを CPU ユニットに書き込みます。

(c) リモートパスワードの削除

- リモートパスワード設定画面の **クリア** ボタンをクリックして、設定されているリモートパスワードを削除します。
- GX Developer でリモートパスワードを書き込みます。



注 6.24 Basic

ベーシックモデル QCPU では、パラメータは必ずプログラムメモリに格納されます。

6.20 GX Developer による CPU ユニットのシステム表示

CPU ユニットに GX Developer を接続し、システムモニタにより、ベースユニットに装着されているユニットの下記項目が確認できます。

- ・ 装着状態
- ・ 動作状態
- ・ ユニット詳細情報
- ・ 製品情報

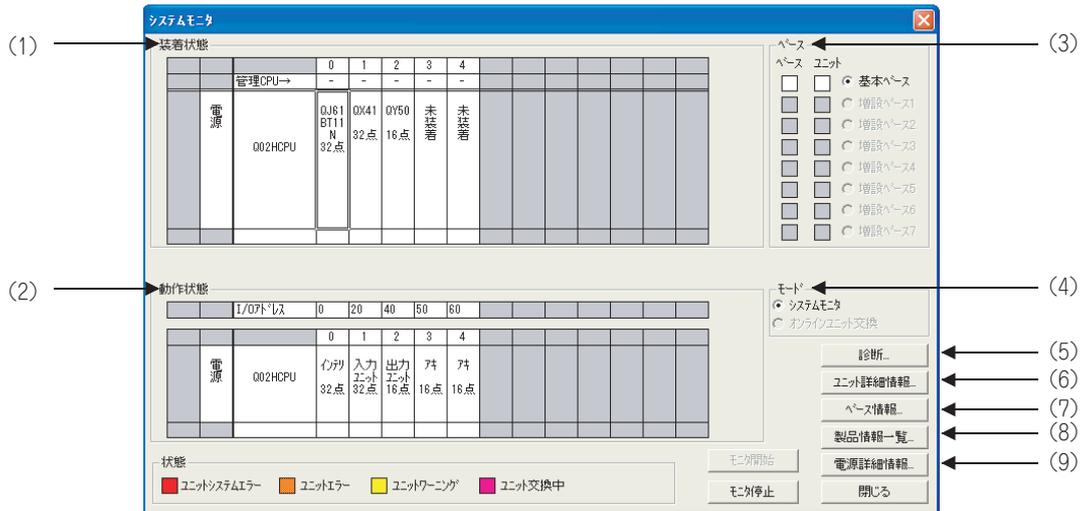


図 6.77 システムモニタ画面

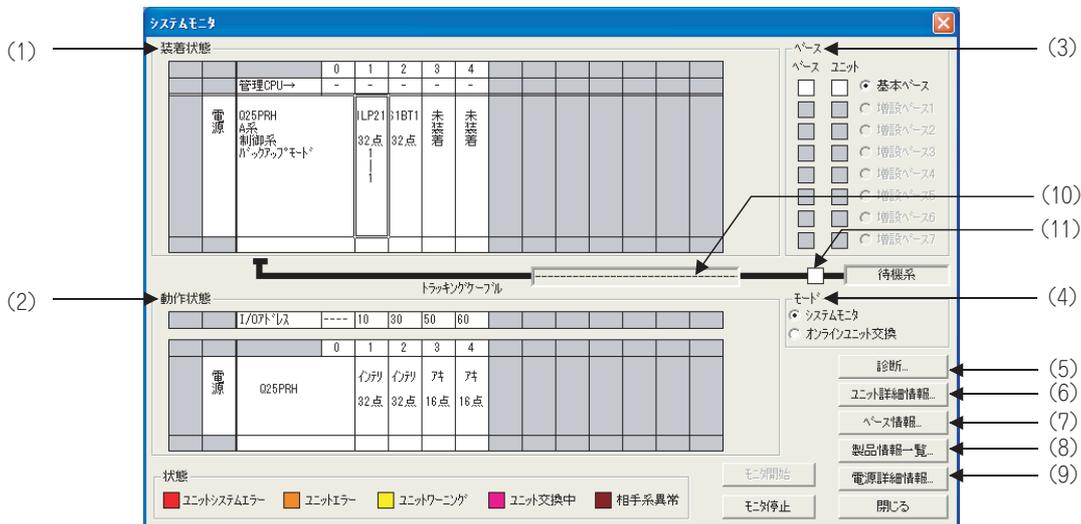


図 6.78 システムモニタ画面 (二重化 CPU の場合)

(1) 装着状態

選択されているベースユニットに装着されているユニットの下記の情報が確認できます。

- 管理 CPU（ベーシックモデル QCPU は除く）
- ユニット形名 * 1
- 点数

ユニットの装着されていないスロットは，“未装着”が表示されます。

二重化ベースユニット使用時は，電源ユニットの装着状態も表示します。（ベーシックモデル QCPU では表示されません。）

* 1：ユニットが装着されていても，PC パラメータの I/O 割付設定で “空き” を設定しているスロットは表示されません。

(2) 動作状態

選択されているベースユニットの各スロットの下記の情報が確認できます。

- 入出力番号
- ユニット種別
- 点数

PC パラメータの I/O 割付設定と実装状態が異なっている場合には，空き 0 点，割付エラーが表示されます。

PC パラメータの I/O 割付設定を実装状態に合わせてください。

二重化ベースユニット使用時は，電源ユニットの動作状態も表示します。（ベーシックモデル QCPU では表示されません。）

(3) ベース

使用しているベースユニットと装着されているユニットの状態が確認できます。

ユニット欄の色は，異常なユニットが 1 つでも存在すると，その異常なユニットの状態の色になります。

(4) モード 注 6.25

オンラインユニット交換実行時に使用します。

オンラインユニット交換については，下記マニュアルを参照してください。

 QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）

 オンラインユニット交換に対応したユニットのマニュアル

(5) 診断

選択されたユニットの状態，エラーの確認を行う場合に使用します。



注 6.25

Basic

High performance

ベーシックモデル QCPU，ハイパフォーマンスモデル QCPU では，オンラインユニット交換ができないため，モードの選択はできません。

(6) ユニット詳細情報

選択したユニットの詳細情報の確認を行う場合に使用します。

インテリジェント機能ユニットの詳細情報について、各インテリジェント機能ユニットのマニュアルを参照してください。

(7) ベース情報

ベース情報では、“全体情報”と“ベース情報”の確認ができます。

(a) 全体情報

使用しているベースユニット数、ベースユニットに装着されているユニット数の確認ができます。

(b) ベース情報

選択されているベースユニットのベースユニット名、スロット数、ベースタイプ、ベースユニットに装着されているユニット数の確認ができます。

(8) 製品情報一覧

装着されているCPUユニット、入出力ユニット、インテリジェント機能ユニットの個別情報（種別・シリーズ・形名・点数・先頭I/O・管理CPU・シリアルNo.・機能バージョン）の確認ができます。

スロット	種別	シリーズ	形名	点数	先頭I/O	管理CPU	シリアルNo	Ver
CPU	CPU	Q	Q06HCPU	-	-	-	0904200000000000	B
0-0	インテリ	Q	QJ71E71-100	32点	0000	-	0904200000000000	D
0-1	インテリ	Q	QJ71C24N	32点	0020	-	0606200000000000	B
0-2	-	-	空き	-	-	-	-	-

図 6.79 製品情報一覧

(9) 電源詳細情報

電源ユニットの ON/OFF 状態、異常の有無、瞬停回数を表示します。

電源詳細情報は、二重化ベースユニットおよび電源詳細情報をサポートしている電源ユニットを使用時に使用可能です。

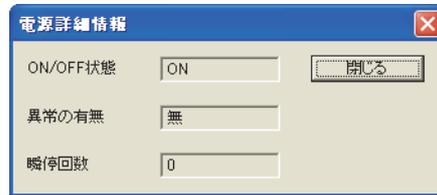


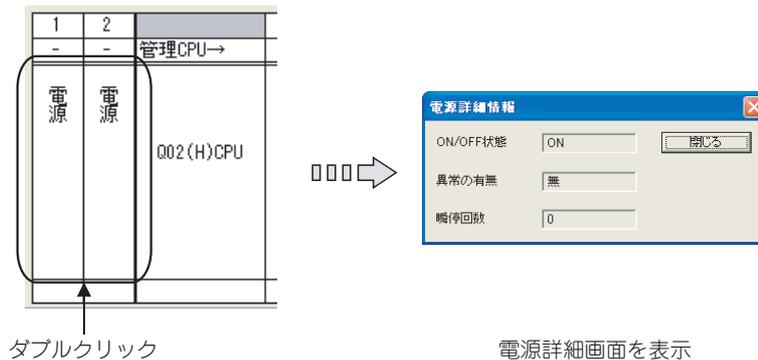
図 6.80 電源詳細画面

表 6.32 電源詳細画面の表示内容

項目	内容
ON/OFF 状態	二重化電源ユニットへの入力電源状態を表示します。
異常の有無	二重化電源ユニットの故障（異常）の有無を表示します。
瞬停回数	<ul style="list-style-type: none"> 二重化基本ベースユニットの二重化電源ユニットを選択時 二重化基本ベースユニットに装着されている二重化電源ユニットの瞬停回数を表示します。 (表示範囲：0～65535) 二重化増設ベースユニットの二重化電源ユニットを選択時 瞬停回数はカウントされず，“-”と表示します。

Point

- マルチ CPU システム構成時の電源詳細情報の表示は、以下の場合のみ可能です。
 - GX Developer を 1 号機に接続した場合
 - すべての CPU ユニットのシリアル No. の上 5 桁が “07032” 以降の場合
- 電源詳細画面は、装着状態の電源ユニット部分をダブルクリックすることによっても表示可能です。



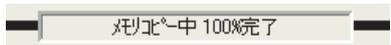
(10)メモリコピー状態

制御系から待機系へのメモリコピー機能の実施状態を表示します。

- 通常運転時



- 制御系から待機系へのメモリコピー時



- トラッキングケーブル異常時



図 6.81 メモリコピー状態

(11)他系状態

他系の状態を表示します。

- 他系正常時



- 他系異常発生時



図 6.82 他系状態

ただし、二重化 CPU がデバッグモードの場合は、他系で異常が発生中でも正常として表示します。(異常発生の表示にはなりません。)

備考

GX Developer のシステムモニタの詳細については、下記マニュアルを参照してください。

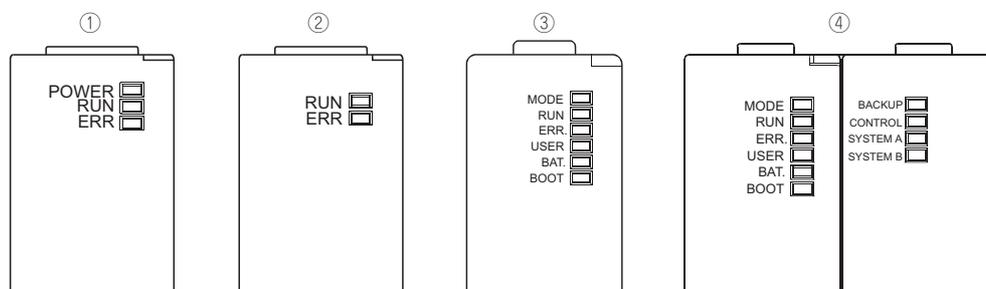
☞ GX Developer Version 8 オペレーティングマニュアル

6.21 LED の表示

CPU ユニットの前面の LED により、CPU ユニットの動作状態が確認できます。

各 LED の表示内容については、下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）



- ①：ベーシックモデルQCPU (Q00JCPU)
- ②：ベーシックモデルQCPU (Q00CPU, Q01CPU)
- ③：ハイパフォーマンスモデルQCPU, プロセスCPU
- ④：二重化CPU

図 6.83 CPU ユニット前面の LED

6.21.1 LED の消灯方法

(1) LED の消灯方法

(a) ベーシックモデル QCPU の場合

点灯している ERR. LED は、エラーの原因を解消したあと、特殊リレー SM50、特殊レジスタ SD50 を操作してエラーを解除すると消灯します。(リセット操作は除きます。)

(b) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU の場合

点灯している LED は、次の操作で消灯できます。(リセット操作は除きます。)

表 6.33 LED の消灯方法

消灯方法	対象 LED			
	ERR.	USER	BAT.	BOOT
エラーの原因を解消した後、LEDR 命令を実行する。	○	○	○	×
エラーの原因を解消した後、特殊リレー SM50、特殊レジスタ SD50 を操作してエラーを解除する。(運転続行エラーに限る)* ¹	○	○	○	×
特殊リレー SM202、特殊レジスタ SD202 を操作して LED を消灯する。* ¹	×	○	×	○

○：有効 ×：無効

*¹：特殊リレー、特殊レジスタの内容について

- SM50 … OFF → ON したときに、SD50 に格納したエラーコードのエラー解除を行います。
- SD50 … エラー解除するエラーコードを格納します。
エラーコードの詳細については、下記マニュアルを参照してください。
☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編)
- SM202 … OFF → ON したときに、SD202 の各ビットに対応した LED を消灯します。
- SD202 … 消灯する LED を指定します。

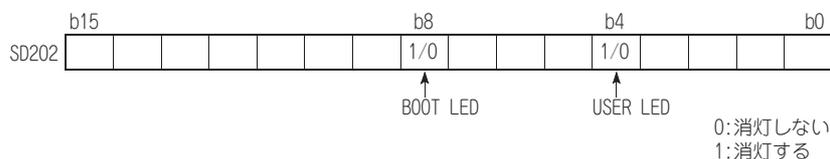


図 6.84 特殊レジスタ SD202 のビット構成

各 LED を消灯する場合の設定は、次のとおりです。

- BOOT LED, USER LED を共に消灯する場合：SD202 = 110H
- BOOT LED のみ消灯する場合：SD202 = 100H
- USER LED のみ消灯する場合：SD202 = 10H

(2) ERR. LED, USER LED, BAT. LED を点灯させない方法

ERR. LED, USER LED, BAT. LED の表示には、優先順位があります。(☞ 6.21.2 項)

優先順位の中から各 LED の対象要因番号を削除すると、要因番号のエラーが発生しても LED は点灯しません。

6.21.2 LED 表示の優先順位

異常発生時、表示器データ（SD220～227）に格納されるエラーメッセージの優先順位について説明します。

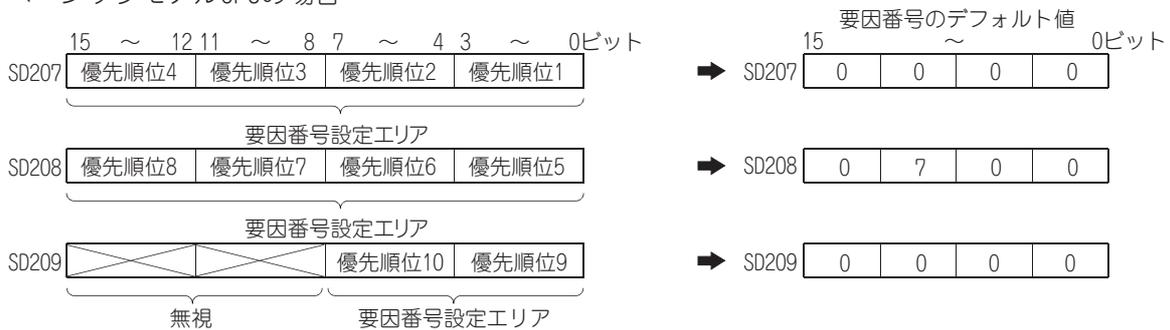
(1) 表示するエラーメッセージと優先順位

表示する要因が複数発生した場合には、次の条件で表示を行います。

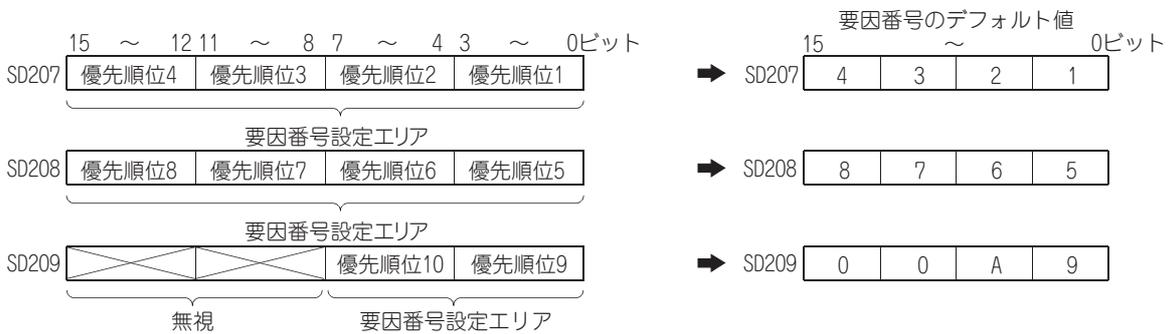
- ・ 停止エラーは、無条件に表示器データ（SD220～227）に設定されます。
- ・ 運転続行エラーは、本項に示す優先順位の要因番号に従って表示します。
優先順位は、変更することが可能です。（特殊レジスタ SD207～209 で設定します。）
- ・ 同一優先順位のエラーが発生した場合は、早く検出された方が表示されます。

優先順位は特殊レジスタ SD207～209 で、次のように設定します。

ベーシックモデルCPUの場合



ハイパフォーマンスモデルQCPU、プロセスCPUの場合



二重化CPUの場合

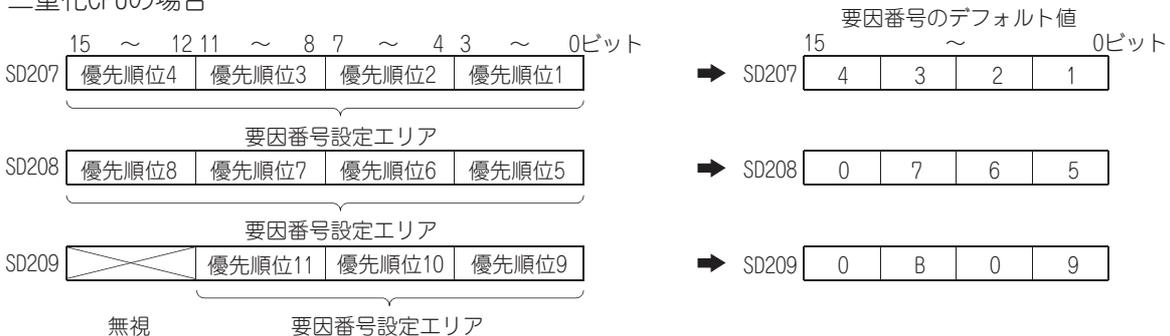


図 6.85 優先順位に関する特殊レジスタとビット構成

(2) 優先順位と要因番号

特殊レジスタ SD207 ~ 209 に設定する要因番号の内容と優先順位は、次のとおりです。

表 6.34 要因番号と優先順位一覧

優先順位	要因番号*1 (16進数)	表示するエラーメッセージ	備考
1	1	<ul style="list-style-type: none"> AC/DC DOWN SINGLE PS.DOWN *2 SINGLE PS.ERROR *2 	<ul style="list-style-type: none"> 電源断 二重化ベースユニットの電源電圧低下 二重化電源ユニット異常
2	2	<ul style="list-style-type: none"> UNIT VERIFY ERR. FUSE BREAK OFF SP.UNIT ERROR SP.UNIT DOWN 	<ul style="list-style-type: none"> 入出力ユニット照合エラー ヒューズ断 インテリジェント機能ユニット照合エラー
3	3	<ul style="list-style-type: none"> OPERATION ERROR SFCP OPE.ERROR SFCP EXE.ERROR 	<ul style="list-style-type: none"> 演算エラー SFC 命令演算エラー SFC プログラム実行エラー
4	4	<ul style="list-style-type: none"> ICM.OPE.ERROR FILE OPE.ERROR OPE. MODE DIFF. *3 CAN'T EXE.MODE *3 TRK.TRANS.ERR. *3 TRK.SIZE ERROR *3 TRK.DISCONNECT *3 	<ul style="list-style-type: none"> メモリカード操作エラー ファイルアクセスエラー 運転状態、キースイッチ不一致 現モード時機能実行不可 トラッキングデータ通信エラー トラッキング容量超過エラー トラッキングケーブル未接続・故障、トラッキング転送用ハードウェア故障
5	5	<ul style="list-style-type: none"> PRG.TIME OVER MULTI CPU ERROR 	<ul style="list-style-type: none"> コンスタントスキャン設定時間オーバー 低速実行監視時間オーバー マルチ CPU システム構成時の他号機異常
6	6	CHK 命令 *4	—
7	7	アナンシェータ	—
8	8	—	—
9	9	BATTERY ERROR	—
10	A	—	—
11	B	<ul style="list-style-type: none"> CAN'T SWITCH *3 STANDBY SYS.DOWN *3 MEM.COPY EXE *3 	<ul style="list-style-type: none"> 系切替えエラー 待機系未起動/停止エラー メモリコピー機能実施

* 1 : ベーシックモデル QCPU では、要因番号 7 (アナンシェータ) のみ設定可能です。

* 2 : ハイパフォーマンスモデル QCPU, プロセス CPU での対応は、CPU ユニットのバージョンを確認してください。

(付 2)

* 3 : ベーシックモデル QCPU, ハイパフォーマンスモデル QCPU, プロセス CPU では表示されません。

* 4 : ベーシックモデル QCPU では、CHK 命令は使用できません。

Point

- エラー発生時にLEDを消灯のままにする場合は、SD207～209の該当する要因番号が格納されている要因番号設定エリア（各々4ビット）を0にしてください。

例 ヒューズ断エラーを検出したとき、ERR.LEDを消灯のままにするには要因番号が“2”の要因番号設定エリアを0にします。



図 6.86 SD207～209 に格納された要因番号

要因番号“2”が設定されていないため、ヒューズ断を検出してもERR.LEDは消灯したままになります。このときに要因番号“2”のその他のエラー（入出力ユニット照合エラー、インテリジェント機能ユニット照合エラー）を検出してもERR.LEDは消灯したままになります。

- 要因番号設定エリアに0を設定（LEDを点灯しない設定）しても、SM0（診断エラーフラグ）のON、SM1（自己診断エラーフラグ）のONおよびSD0（診断エラーレジスタ）へのエラーコードの格納は行います。

6.22 高速割り込み機能 注 6.26

高速割り込みポインタ (I49) を使用して割り込みプログラムを作成すると、0.2ms ~ 1.0ms 間隔の高速な定周期割り込みによるプログラム実行ができます。

また、高速割り込みプログラムの実行前後では、パラメータで設定した範囲の入出力信号、およびインテリジェント機能ユニットのバッファメモリのリフレッシュを行うことで、入出力の応答性を向上します。

これにより、精密な位置検出などの高精度な制御が可能です。

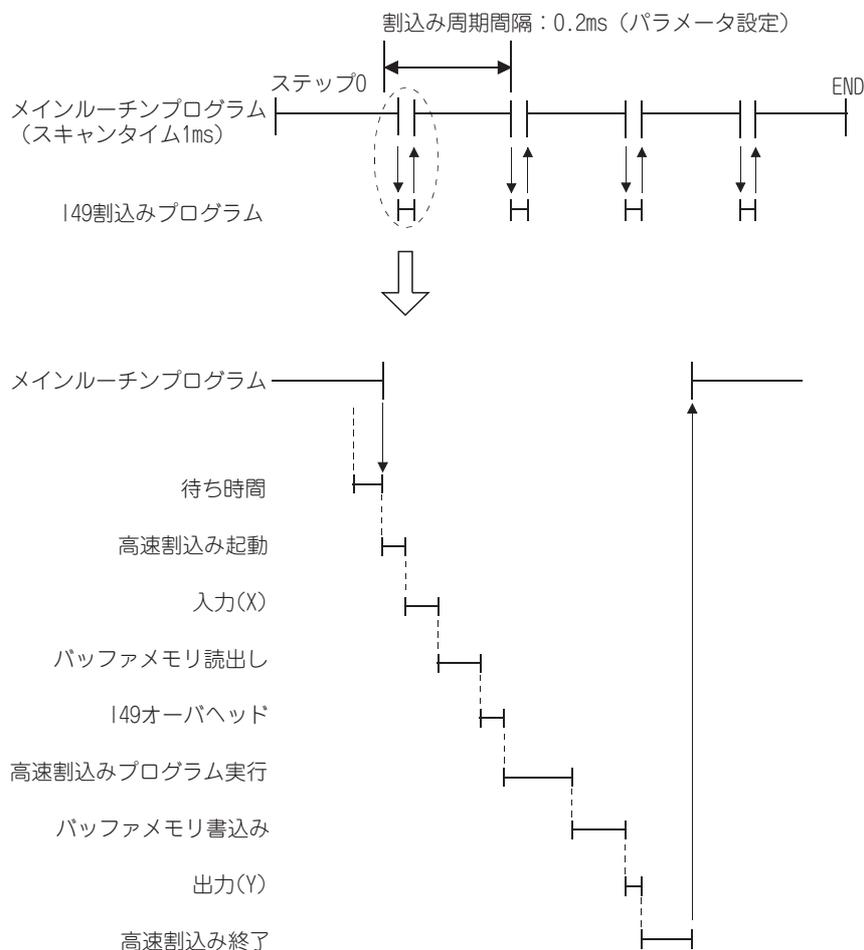


図 6.87 高速割り込みタイムチャート

高速割り込み機能は、以下の機能で構成されます。

- 高速割り込みプログラム実行機能： 6.22.1 項
- 高速 I/O リフレッシュ/高速バッファ転送機能： 6.22.2 項



注 6.26

Basic

High performance

Process

Redundant

ベーシックモデル QCPU、Q02CPU、プロセス CPU、二重化 CPU では、高速割り込み機能は使用できません。ハイパフォーマンスモデル QCPU で高速割り込み機能を使用する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。(付 2.2)

6.22.1 高速割り込みプログラム実行機能

高速割り込みポインタ (I49) に従って割り込みプログラムを実行する機能です。

(1) 設定方法

PC パラメータの PC システム設定で “システム割り込み設定” の **高速割り込み設定** ボタンをクリックして行います。

0.2 ~ 1.0ms の範囲で “高速割り込み I49 定周期間隔” を設定します。

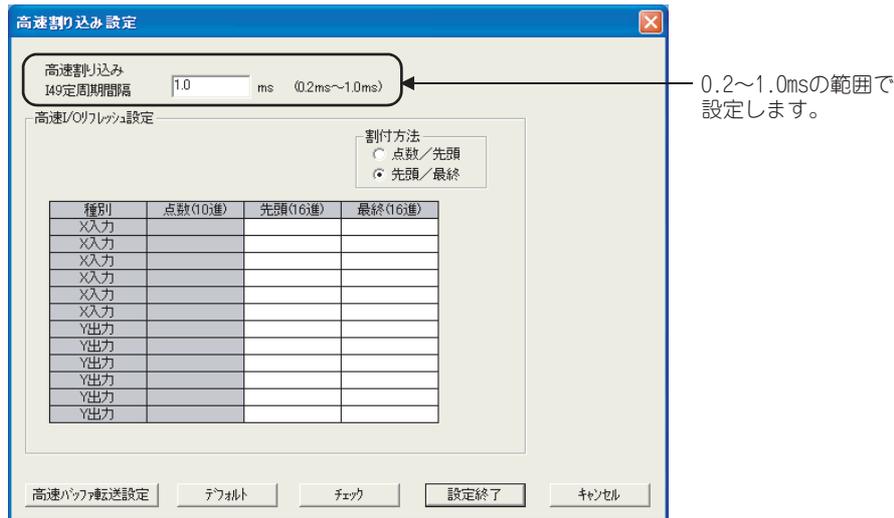


図 6.88 高速割り込み設定

(2) 注意事項

(a) 割り込み禁止中の高速割り込みについて

高速割り込みプログラムは、割り込み禁止中は実行されません。

割り込み禁止解除にて実行されます。

割り込み禁止により高速割り込みの起動が待たされる項目については、6.22.4 項 (3) を参照してください。

(b) 高速割り込みが無視される場合について

設定した割り込み周期間隔以上に割り込み禁止の期間が続いた場合は、高速割り込みが無視されることがあります。

割り込み禁止中に高速割り込みが 2 回発生するタイミングのときに、高速割り込みが 1 回無視されます。

(c) 高速割り込みプログラムの実行可否について

本機能は、下記の条件をすべて満たしている場合に実行します。

- EI 命令中
- CPU ユニットが RUN 状態
- IMASK 命令で高速割り込みポインタ (I49) がマスクされていない状態 (デフォルトでは、マスクされていません。)

IMASK 命令および EI 命令については、下記マニュアルを参照してください。

📖 MELSEC-Q/L プログラミングマニュアル (共通命令編)

6.2.2.2 高速 I/O リフレッシュ／高速バッファ転送機能

高速 I/O リフレッシュは、割り込み周期間隔で入出力ユニット、インテリジェント機能ユニットと CPU ユニット間で入出力信号を更新する機能です。

また、高速バッファ転送は、割り込み周期間隔でインテリジェント機能ユニットのバッファメモリデータと CPU ユニットのデバイスデータ間を更新する機能です。

(1) 設定方法

高速割り込み設定画面で下記を設定します。

- “高速割り込み 149 定周期間隔”
- “高速 I/O リフレッシュ設定”
- “高速バッファ転送設定”

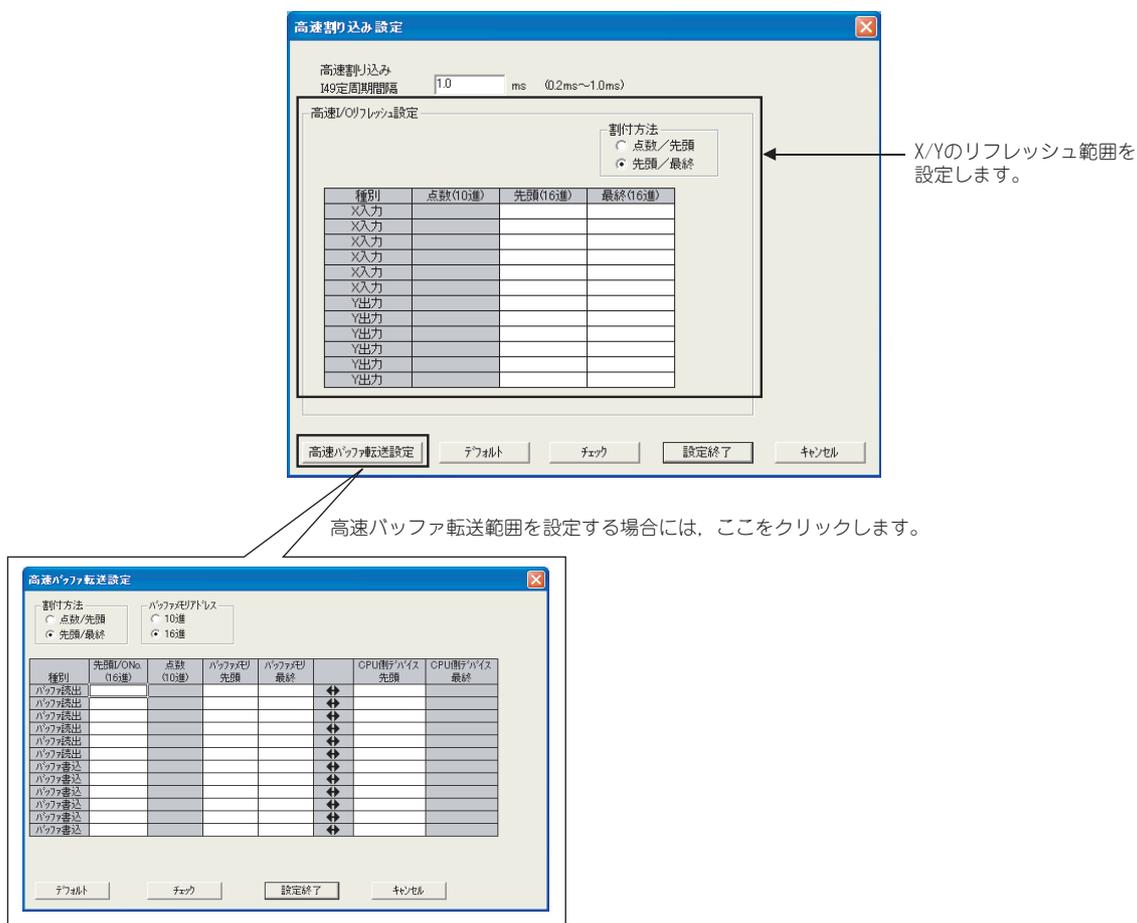


図 6.89 高速割り込み設定画面

表 6.35 高速 I/O リフレッシュ設定および高速バッファ転送設定

項目	設定項目	設定内容	制約事項	設定数
高速 I/O リフレッシュ 設定	点数 (10 進)	転送ビット数 (16 ~ 4096)	・入出力ユニット, インテリジェント 機能ユニットのみ ・16 の倍数指定のみ* 1	X 入力/Y 出力 それぞれ 6 個まで
	先頭 (16 進)	先頭デバイス No. (X0 ~ 0FF0/Y0 ~ 0FF0)		
	最終 (16 進)	最終デバイス No. (X000F ~ 0FFF/Y000F ~ 0FFF)		
高速バッファ転送 設定	先頭 I/O No.(16 進)	先頭 I/O No. ÷ 10H (0 ~ FFH)	インテリジェント機能ユニットのみ* 2	読入/書込 それぞれ 6 個まで
	点数 (10 進)	転送ワード数 (1, 2 ~ FFFE _H)	・インテリジェント機能ユニットのみ ・偶数アドレス, 偶数ワード指定のみ* 3	
	バッファメモリ先頭	先頭アドレス (0 ~ FFFF _H)		
	バッファメモリ最終	先頭アドレス (FFFF _H ~ FFFF _H)	D,W,R,ZR のみ	
	CPU 側デバイス先頭	先頭デバイス No.		
	CPU 側デバイス最終	先頭デバイス No.		

* 1 : 先頭デバイス No. および転送ビット数ともに 16 の倍数のみ設定可能です。

* 2 : AnS/A シリーズ対応の増設ベースユニット (QA1S5 □ B, QA1S6 □ B, QA1S6ADP+A1S5 □ B/A1S6 □ B, QA6 □ B, QA6ADP+A5 □ B/A6 □ B) は接続できないため, AnS/A シリーズ対応の特殊機能ユニットは対象外です。

接続すると, "PARAMETER ERROR" (エラーコード: 3006) になります。

また, インテリジェント機能ユニット装着チェック, バッファメモリ容量チェックでエラーの場合も, "PARAMETER ERROR" (エラーコード: 3006) になります。

* 3 : 転送ワード数が 1 の指定のみ, 奇数アドレスも可能です。

Point

本機能の対象とするユニットは, 基本ベースユニットに装着することをお奨めします。
(基本ベースユニットは, 増設ベースユニットよりもユニットへのアクセス時間が早いいためです。)

(2) 高速 I/O リフレッシュ/高速バッファ転送機能の実行可否

本機能は, 下記の条件をすべて満たしている場合に実行します。

- ・ EI 命令中
- ・ CPU ユニットが RUN 状態
- ・ IMASK 命令で高速割り込みポインタ (I49) がマスクされていない状態
(デフォルトでは, マスクされていません。)

IMASK 命令および EI 命令については, 下記マニュアルを参照してください。

📖 MELSEC-Q/L プログラミングマニュアル (共通命令編)

6.22.3 処理時間

高速割込み機能の起動から終了までの間の各処理時間を示します。

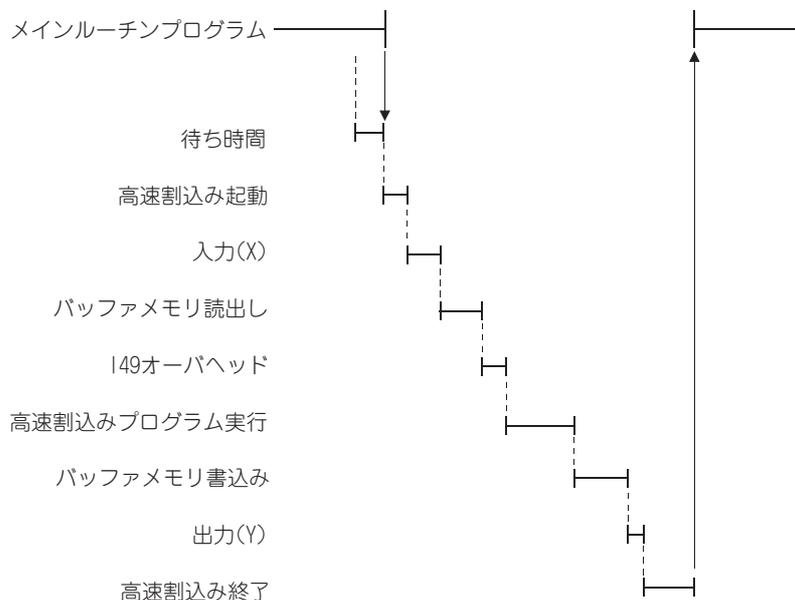


図 6.90 高速割込み機能の起動から終了までの処理

表 6.36 高速 I/O リフレッシュ, 高速バッファ転送に関連する処理時間

処理項目	処理時間
待ち時間	最大 37.5 μ s または 37.5 μ s 以上の命令処理時間, CC-Link IE コントローラネットワーク, MELSECNET/H, CC-Link, インテリジェント機能ユニットを増設ベースユニットに装着した場合は, 最大 40 μ s
高速割込み起動+高速割込み終了	22 μ s
X 入力	基本ベースユニット時: 時間 = $0.14 \times (X \text{ 総点数}) + 0.65 \times (\text{設定数}) + 0.85$ 増設ベースユニット時: 時間 = $0.21 \times (X \text{ 総点数}) + 0.65 \times (\text{設定数}) + 0.85$ (計算例) 基本ベースユニット装着で設定数 1, X 点数 16 点の場合, 3.74 μ s
バッファメモリ読出し	基本ベースユニット時: <ul style="list-style-type: none"> 16 ワード以下のとき…時間 = $0.47 \times (\text{転送総ワード数}) + 2.85 \times (\text{設定数}) + 0.95$ 16 ワードを超えるとき…時間 = $0.5 \times (\text{転送総ワード数}) + 0.95$ 増設ベースユニット時: <ul style="list-style-type: none"> 16 ワード以下のとき…時間 = $1.07 \times (\text{転送総ワード数}) + 2.85 \times (\text{設定数}) + 0.95$ 16 ワードを超えるとき…時間 = $1.1 \times (\text{転送総ワード数}) + 0.95$ (計算例) 基本ベースユニット装着で設定数 1, 2 ワードの場合, 4.74 μ s
I49 のオーバーヘッド	41 μ s
高速割込みプログラム実行	ユーザ作成の割込みプログラムによります。
バッファメモリ書込み	基本ベースユニット時: <ul style="list-style-type: none"> 16 ワード以下のとき…時間 = $0.47 \times (\text{転送総ワード数}) + 2.65 \times (\text{設定数}) + 0.95$ 16 ワードを超えるとき…時間 = $0.55 \times (\text{転送総ワード数}) + 0.95$ 増設ベースユニット時: <ul style="list-style-type: none"> 16 ワード以下のとき…時間 = $1.07 \times (\text{転送総ワード数}) + 2.65 \times (\text{設定数}) + 0.95$ 16 ワードを超えるとき…時間 = $1.15 \times (\text{転送総ワード数}) + 0.95$ (計算例) 基本ベースユニット装着で設定数 1, 2 ワードの場合, 4.54 μ s
Y 出力	基本ベースユニット時: 時間 = $0.13 \times (Y \text{ 総点数}) + 1.55$ 増設ベースユニット時: 時間 = $0.2 \times (Y \text{ 総点数}) + 1.55$ (計算例) 基本ベースユニット装着で設定数 1, Y 点数 16 点の場合, 3.63 μ s

6.22.4 制約事項

高速割込み機能を実行する場合の制約事項について説明します。

誤って実行した場合には、“WDT ERROR”が発生したり、高速割り込みが設定周期間隔で実行できなくなることがあります。

制約事項は、下記の4種類があります。

- (1) 高速割込み設定をした場合にすべて使用不可とする項目
- (2) 高速割込み内のみで使用不可とする項目
- (3) 割り込み禁止により高速割込みの起動が待たされる項目
- (4) (1)～(3)以外に注意する項目

また、1回の割り込みプログラムに要する時間は、割り込み周期間隔の設定時間を超えないようにしてください。

(1回の割り込みプログラムに要する時間が割り込み周期間隔の設定時間を超えた場合は、高速割込みの動作を保証できなくなります。)

(1) 高速割込み設定をした場合にすべて使用不可とする項目

表 6.37 高速割込み設定をした場合にすべて使用不可とする項目

項目	制約事項	使用した場合
ベースユニット	QA1S5 □ B, QA1S6 □ B, QA1S6ADP+A1S5 □ B/A1S6 □ B, QA6 □ B, QA6ADP+A5 □ B/A6 □ B は接続不可	“PARAMETER ERROR”を検出します。
マルチ CPU システム	マルチ CPU システムは構成不可	GX Developer のパラメータ設定時にチェックします。
命令	PR/PRC, UDCNT1/2, PLSY, PWM, SPD, PLOADP/PUNLOADP/PSWAPP 命令は実行不可	左記命令を実行せず、エラーを検出します。
命令	高速割込み周期以上に処理時間がかかる命令は使用不可	命令実行中は割り込み禁止のため、高速割込みが設定周期で実行できなくなります。
プログラミングユニット	プログラミングユニット接続不可	命令サーチの応答が遅くなります。 または、プログラミングユニット側で交信エラーとなる場合があります。
SFC	下記2つの SFC 機能は実行不可 ・ SM90～99, SD90～99 による、SFC の移行監視チェック機能 ・ 定時実行ブロック機能	左記機能は実行されず無視されます。
サンプリングトレース	時間毎のサンプリングトレースは使用不可 (スキャン毎、詳細条件実行時は可)	サンプリングトレースが実行されず無視されます。 (トレースデータ読出し時、トレース結果が読み出されない場合があります。)
割り込みプログラム (10～48, 150～255), 定周期プログラム	割り込みプログラム (10～48, 150～255), 定周期プログラムは実行不可	多重割り込み禁止のため、割り込みプログラム、定周期プログラム実行中は、高速割込みが設定周期で実行できなくなります。
RUN 中書込み (回路モードでの RUN 中書込み、ファイルの RUN 中書込み)	RUN 中書込みは実行不可	ファイルの RUN 中書込み実行中は割り込み禁止のため、その間高速割込みの起動が遅れるので、高速割込みが設定周期で実行できなくなり、下記時間がかかります。 ・ 回路モードでの RUN 中書込みで最大 102 μs, ・ ファイルの RUN 中書込みで最大 300ms

(次ページにつづく)

表 6.37 高速割込み設定をした場合にすべて使用不可とする項目 (つづき)

項目	制約事項	使用した場合
プログラム名と同じファイルレジスタ	プログラム名と同じファイルレジスタは使用不可	プログラム名と同じファイルレジスタを切換え時は割込み禁止のため、高速割込みが設定周期で実行できなくなり、下記時間がかかります。 <ul style="list-style-type: none"> 標準 RAM で $410 \mu s$, SRAM カードで $400 \mu s + 100 \mu s \times$ プログラムファイル数分
ローカルデバイス	ローカルデバイスは使用不可	ローカルデバイスは切換え時に割込み禁止のため、高速割込みが設定周期で実行できなくなり、下記時間がかかります。 <ul style="list-style-type: none"> 標準 RAM で $390 \mu s + 170 \mu s \times n$, SRAM カードで $390 \mu s + 950 \mu s \times n$ (n: プログラムファイル数)
CPU ユニットにアクセスするインテリジェント機能ユニットのコマンド	QJ71C24, QJ71E71 などの CPU ユニットにアクセスするインテリジェント機能ユニットから、CPU アクセスコマンド発行は実行不可	CPU アクセスコマンド発行時は割込み禁止となるため、その間高速割込みの起動が遅れ、高速割込みが設定周期で実行できなくなります。 <ul style="list-style-type: none"> N 点読出/書込: $(0.07 \times N + 34) \mu s$ N 点ランダム読出/書込: $(0.07 \times N + 101) \mu s$
他局経由モニタ	自局モニタ中に CC-Link IE コントローラネットワーク, MELSECNET/H, QJ71C24 などのインテリジェント機能ユニット経由モニタは実行不可	自局モニタ要求とインテリジェント機能ユニット経由モニタ要求が重なると、割込み禁止の処理時間が延びるため、その間高速割込みの起動が遅れ ($102 \mu s$)、高速割込みが設定周期で実行できなくなります。
割込みカウンタ	割込みポイント I49 に対応した割込みカウンタ使用不可	割込みカウンタ設定があっても、I49 に対する設定は無視され、高速割込み I49 は通常に実行されます。 (他の割込みポイントは、割込みプログラムが実行されず、割込みカウンタが実行されます。)

(2) 高速割込み内のみで使用不可とする項目

表 6.38 高速割込み内のみで使用不可とする項目

項目	制約事項	使用した場合
デバイスコメント	高速割込みプログラム内でプログラム名と同じデバイスコメントは退避/復帰しない	高速割込みプログラムのデバイスコメントが書き変わります。
インデックスレジスタ	高速割込みシーケンス内では、インデックスレジスタは退避/復帰しない	高速割込みプログラムのインデックスレジスタが書き変わります。
アクセス実行フラグ (SM390)	高速割込みプログラムでアクセス実行フラグ SM390 は退避/復帰しない	高速割込みプログラムの SM390 の値が書き変わります。
強制 ON/OFF	高速 X/Y リフレッシュエリアは強制 ON/OFF 不可	高速割込み内で、実行されず無視されます。 (タイムアウトエラーにはなりません。)
モニタ条件の設定	高速割込みプログラム内は指定不可	正常に実行されません。 (タイムアウトエラーにはなりません。)
実行時間計測	高速割込みプログラム内は指定不可	実行されず無視されます。 (タイムアウトエラーにはなりません。)

(3) 割込み禁止により高速割込みの起動が待たされる項目

表 6.39 割込み禁止により高速割込みの起動が待たされる項目

項目	注意事項
命令	命令実行中は、割込み禁止です。
リンクリフレッシュ	リフレッシュ（バスアクセス）中は、割込み禁止です。 CC-Link IE コントローラネットワーク、MELSECNET/H、CC-Link、インテリジェント機能ユニットのリフレッシュでは、各ユニットを基本ベースユニット装着時で最大 37.5 μ s、増設ベースユニット装着時で最大 40 μ s 待たされます。
複数プログラム実行	複数プログラム実行時、プログラム切替中は割込み禁止です。30 μ s 待たされます。 高速割込み機能設定時は、プログラム本数 1 本を推奨いたします。
モニタ	回路モニタ、デバイス一括モニタ、デバイス登録モニタは下記の時間待たされます。 (0.096 × デバイス点数 + 20) μ s
AC DOWN 時	最大 20ms 高速割込み起動が待たされます。

(4) (1) ~ (3) 以外に注意する項目

(a) PC パラメータの割込プログラム／定周期プログラム設定について

“高速実行する” は、高速割込み機能に対しては無効です。

(b) 高速バッファ転送時について

設定範囲外のファイルレジスタ（最大点数を超える範囲）を使用した場合は、エラーにはならず範囲外への転送を実施しません。（他のデバイスの内容を壊すことはありません。）

(c) プログラム作成上の注意事項

他の割込みプログラムの場合と同様の注意事項があります。（ 2.2.3 項）

6.23 インテリジェント機能ユニットからの割り込み 注 6.27

CPU ユニットは、インテリジェント機能ユニットからの割り込み要求により、割り込みプログラム (I□) を実行できます。

例えば、シリアルコミュニケーションユニットでは、下記データ交信機能実行時に割り込みプログラムでデータの受信処理を行うことができます。

- 無手順プロトコルによる交信時のデータ受信
- 双方向プロトコルによる交信時のデータ受信

割り込みプログラムでデータの受信処理を行うことにより、CPU ユニットへの受信データの取込みを早くすることができます。

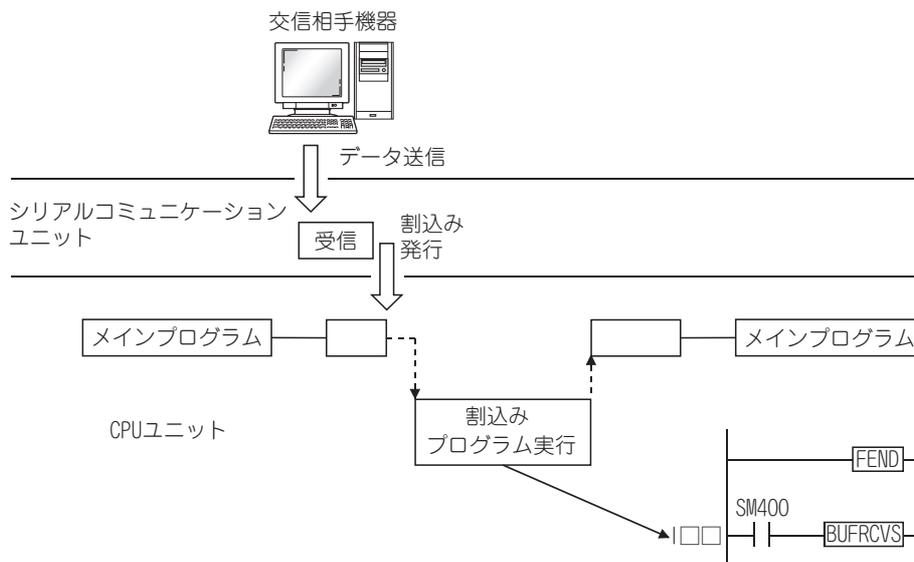


図 6.91 シリアルコミュニケーションユニットからの割り込み

(1) インテリジェント機能ユニットからの割り込みの設定

PC パラメータの PC システム設定のインテリジェント機能ユニット設定の割込ポイント設定の “インテリジェント機能ユニット割込ポイント設定” を設定します。

また、インテリジェント機能ユニットでの “インテリジェント機能ユニットパラメータ” の設定が必要です。インテリジェント機能ユニットからの割り込みで割り込みプログラムを実行する場合は、下記マニュアルを参照してください。

使用するインテリジェント機能ユニットのマニュアル

備考

インテリジェント機能ユニットからの割り込み時に、使用可能な割り込みポイントの番号については、9.11 節を参照してください。

注 6.27 Basic

ベーシックモデル QCPU で、インテリジェント機能ユニットからの割り込みを使用する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。(付 2.1)

6.24 シリアルコミュニケーション機能 注 6.28

(1) シリアルコミュニケーション機能とは

CPU ユニットの RS-232 インタフェースとパソコン、表示器などを RS-232 ケーブルで接続し、MC プロトコル*¹により通信を行う機能です。

シリアルコミュニケーション機能の仕様、機能、各種設定について説明します。

* 1：MC プロトコルとは、MELSEC コミュニケーションプロトコルの略称です。

MELSEC コミュニケーションプロトコルは、Q シリーズシーケンサ（シリアルコミュニケーションユニット、Ethernet ユニットなど）の通信手順に従い、相手機器から CPU ユニットへアクセスするための通信方式の名前です。MC プロトコルについては、下記マニュアルを参照してください。

 MELSEC コミュニケーションプロトコルリファレンスマニュアル

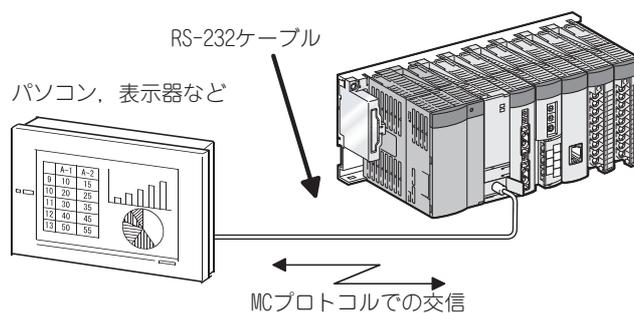


図 6.92 パソコン、表示器との通信

Point

- パソコン、表示器などからシリアルコミュニケーション機能で通信できるのは、パソコン、表示器などと接続している CPU ユニットのみです。
CC-Link IE コントローラネットワーク、MELSECNET/H、Ethernet、CC-Link の他局との通信はできません。
- GX Developer、GX Configurator と CPU ユニットの接続には、シリアルコミュニケーション機能を使用しません。

 注 6.28 **Basic** **High performance** **Process** **Redundant**

Q00JCPU、ハイパフォーマンスモデル QCPU、プロセス CPU、二重化 CPU では、シリアルコミュニケーション機能は使用できません。

(2) 仕様

(a) 伝送仕様

CPU ユニットのシリアルコミュニケーション機能で使用する RS-232 の伝送仕様を表 6.40 に示します。パソコン/表示器などの仕様が表 6.40 に合っていることを確認後、CPU ユニットのシリアルコミュニケーション機能を使用してください。

表 6.40 シリアルコミュニケーション機能の伝送仕様

項目	デフォルト	設定範囲
通信方式	全二重通信	—
同期方式	調歩同期方式	—
伝送速度* 1	19.2kbps	9.6kbps, 19.2kbps, 38.4kbps, 57.6kbps, 115.2kbps
データ形式	<ul style="list-style-type: none"> • スタートビット：1 • データビット：8 • パリティビット：奇数 • ストップビット：1 	—
MC プロトコル形式* 2 (自動判別)	<ul style="list-style-type: none"> • 形式 4 (ASCII) • 形式 5 (バイナリ) 	—
フレーム* 2	<ul style="list-style-type: none"> • QnA 互換 3C フレーム • QnA 互換 4C フレーム 	—
伝送制御	DTR/DSR 制御	—
サムチェック* 1	有	有, 無
伝送ウェイト時間* 1	ウェイト無	ウェイト無, 10ms ~ 150ms (10ms 単位)
RUN 中書込設定* 1	許可しない	許可する, 許可しない
延長距離	15m	—

* 1 : GX Developer の PC パラメータ設定で設定できます。

* 2 : MC プロトコル形式とフレームの関係を表 6.41 に示します。

表 6.41 MC プロトコル形式とフレームの関係

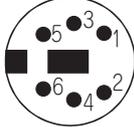
機能		形式 4	形式 5
ASCII コードでの交信	QnA 互換 3C フレーム	○	×
	QnA 互換 4C フレーム	○	×
バイナリコードでの交信	QnA 互換 4C フレーム	×	○

○ : 使用可, × : 使用不可

(b) RS-232 コネクタ仕様

CPU ユニットの RS-232 コネクタの仕様を表 6.42 に示します。

表 6.42 RS-232 コネクタ仕様

外 観	ピン番号	信号名	信号名称
 Mini-Din 6ピン (メス)	1	RD(RXD)	受信データ
	2	SD(TXD)	送信データ
	3	SG	信号グラウンド
	4	—	—
	5	DR(DSR)	データセットレディ
	6	ER(DTR)	データターミナルレディ

(c) RS-232 ケーブル

パソコン/表示器などと CPU ユニットの接続は、下記 RS-232 ケーブルが使用できます。

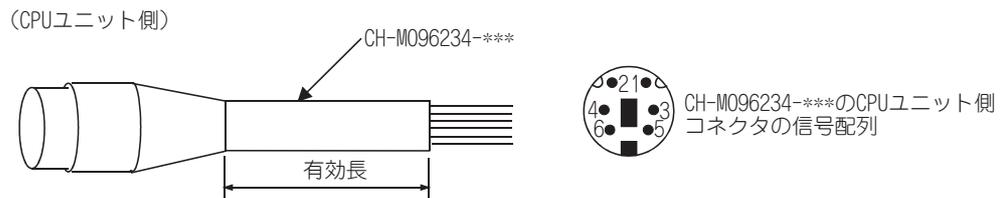
- QC30R2 (ケーブル長：3m)
- CH-M096234-*** (株式会社チューガイ製)

片側：ミニ DIN コネクタ付、片側：コネクタ無のケーブル

*** はケーブル長を示し、0.1m 単位で 15m まで指定可能

連絡先：株式会社チューガイ

www.chu-gai.co.jp



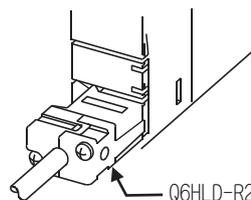
ピン番号	1	2	3	4	5	6	金属シェル
信号名	RD	SD	SG	—	DR	ER	
線心	赤	黒	緑・白	—	黄	茶	シールド

図 6.93 RS-232 ケーブルの有効長と信号配列

Point

RS-232 ケーブルを CPU ユニットの固定する場合は、RS-232 コネクタ抜け防止用ホルダ (Q6HLD-R2) の使用を推奨します。Q6HLD-R2 については、下記マニュアルを参照してください。

📄 Q6HLD-R2 形 RS-232 コネクタ抜け防止用ホルダユーザーズマニュアル



(3) 機能

シリアルコミュニケーション機能では、表 6.43 の MC プロトコルのコマンドが実行できます。

表 6.43 シリアルコミュニケーション機能がサポートする MC プロトコルコマンド一覧

機能		コマンド	処理内容	処理点数
デバイスメモリ	一括読出し	ビット単位	0401(00□1)	ビットデバイスを1点単位で読み出す。 ASCII: 3584点, BIN: 7168点
		ワード単位	0401(00□0)	ビットデバイスを16点単位で読み出す。 ワードデバイスを1点単位で読み出す。 480ワード(7680点) 480点
	一括書き込み*1	ビット単位	1401(00□1)	ビットデバイスを1点単位で書き込む。 ASCII: 3584点, BIN: 7168点
		ワード単位	1401(00□0)	ビットデバイスを16点単位で書き込む。 ワードデバイスを1点単位で書き込む。 480ワード(7680点) 480点
	ランダム読出し*2*3	ワード単位	0403(00□0)	ビットデバイスを16点、32点単位で、デバイス・デバイス番号をランダムに指定して読み出す。 ワードデバイスを1点、2点単位で、デバイス・デバイス番号をランダムに指定して読み出す。 96点
	テスト*1 (ランダム書き込み)	ビット単位	1402(00□1)	ビットデバイスを1点単位で、デバイス・デバイス番号をランダムに指定してセット、リセットする。 94点
		ワード単位*2	1402(00□0)	ビットデバイスを16点、32点単位で、デバイス・デバイス番号をランダムに指定してセット、リセットする。 ワードデバイスを1点、2点単位で、デバイス・デバイス番号をランダムに指定して書き込む。 *5
	モニタ登録*2,*3,*4	ワード単位	0801(00□0)	モニタするビットデバイスを16点、32点単位で登録する。 96点
				モニタするワードデバイスを1点、2点単位で登録する。 96点
	モニタ	ワード単位	0802(00□0)	モニタ登録したデバイスをモニタする。 モニタ登録点数分

*1: RUN 中に書き込む場合は、“RUN 中書込設定”を“許可する”に設定します。

*2: TS, TC, SS, SC, CS, CC のデバイスは、ワード単位の指定はできません。モニタ登録で指定した場合は、モニタ実行時にエラー(4032H)になります。

*3: モニタ条件指定は使用できません。

*4: モニタ登録を複数の相手機器から実行しないでください。実行時は最後のモニタ登録が有効になります。

*5: 処理点数は下式の範囲内で設定します。

$$(\text{ワードアクセス点数}) \times 12 + (\text{ダブルワードアクセス点数}) \times 14 \leq 960$$

・ビットデバイスは、ワードアクセス時1点が16ビット分に、ダブルワードアクセス時1点が32ビット分になります。

・ワードデバイスは、ワードアクセス時1点が1ワード分に、ダブルワードアクセス時1点が2ワード分になります。

(4) アクセス可能デバイス

シリアルコミュニケーション機能でアクセス可能なデバイスを表 6.44 に示します。

表 6.44 シリアルコミュニケーション機能でアクセス可能なデバイス

分類	デバイス	デバイスコード* 1		デバイス番号範囲		
		ASCII	バイナリ			
内部システム デバイス	ファンクション入力	—	—	(アクセス不可)	16 進	
	ファンクション出力	—	—		16 進	
	ファンクションレジスタ	—	—		10 進	
	特殊リレー	SM	91 _H		10 進	
	特殊レジスタ	SD	A9 _H		10 進	
内部ユーザ デバイス	入力	X *	9C _H	アクセス先の CPU ユニットが持つ デバイス番号の範囲で指定で きます。 ただし、ローカルデバイスには アクセスできません。	16 進	
	出力	Y *	9D _H		16 進	
	内部リレー	M *	90 _H		10 進	
	ラッチリレー	L *	92 _H		10 進	
	アナンシェータ	F *	93 _H		10 進	
	エッジリレー	V *	94 _H		10 進	
	リンクリレー	B *	A0 _H		16 進	
	データレジスタ	D *	A8 _H		10 進	
	リンクレジスタ	W *	B4 _H		16 進	
	タイマ	接点	TS		C1 _H	10 進
		コイル	TC		C0 _H	
		現在値	TN		C2 _H	
	積算 タイマ	接点	SS		C7 _H	10 進
		コイル	SC		C6 _H	
		現在値	SN		C8 _H	
	カウンタ	接点	CS		C4 _H	10 進
		コイル	CC		C3 _H	
		現在値	CN		C5 _H	
		リンク特殊リレー	SB		A1 _H	16 進
		リンク特殊レジスタ	SW		B5 _H	16 進
	ステップリレー	S *	98 _H	10 進		
	ダイレクト入力* 2	DX	A2 _H	16 進		
	ダイレクト出力* 2	DY	A3 _H	16 進		
インデックスレジ スタ	インデックスレジスタ	Z *	CC _H	アクセス先の CPU ユニットが持つ デバイス番号の範囲で指定で きます。	10 進	
ファイルレジスタ * 3	ファイルレジスタ	R *	AF _H		10 進	
		ZR	B0 _H		16 進	

* 1 : デバイスコードは、MC プロトコルの伝文内で指定するコードです。ASCII コードでデータ送信時は、2 キャラクタでデバイスコードを指定します。デバイス文字が 1 キャラクタの場合は、デバイス文字のあとに「*」（ASCII コード：2A_H）またはスペース（ASCII コード：20_H）を付加します。

* 2 : DX/DY1000 以降にはアクセスできません。DX/DY1000 以降にアクセスしたい場合は、X/Y デバイスを使用してください。

* 3 : Q00JCPU では、使用できません。

(5) 伝送仕様の設定

シリアルコミュニケーション機能の伝送速度、サムチェック、伝送ウェイト時間、RUN 中書込設定は、PC パラメータのシリアルコミュニケーション設定で行います。

- シリアルコミュニケーション機能でパソコン、表示器などと通信する場合は、“シリアルコミュニケーション機能を使用する” に設定します。
- 伝送速度、サムチェック、伝送ウェイト時間、RUN 中書込設定の設定を行います。

シリアルコミュニケーション機能使用時にクリックします。

伝送速度、サムチェック、伝送ウェイト時間、RUN中書込設定の設定を行います。



図 6.94 シリアルコミュニケーション設定画面

(6) 注意事項

(a) 表示器などとの通信中に GX Developer に接続を切り換える場合について

シリアルコミュニケーション機能によるパソコン、表示器などとの通信中に GX Developer に接続を切り換えることができます。

ただし、シリアルコミュニケーション機能により通信していたパソコン、表示器などは、通信エラーになります。

CPU ユニットに再接続した場合のパソコン、表示器などの立上げ方法については、使用する機器のマニュアルを参照してください。

(b) 接続先指定画面で設定した伝送速度について

“シリアルコミュニケーション機能を使用する” に設定すると、GX Developer の接続先指定画面で設定した伝送速度は、無視されます。

(c) 通信異常時について

下記のいずれかの条件が該当する場合は、応答が返らないため通信できません。該当する条件を見直してください。

- ・ シリアルコミュニケーション機能を使用しない設定になっている
- ・ 異なる伝送速度およびデータ形式で通信を行っている
- ・ 送信するフレームに正しい始端／終端がない、または誤っている
 - ・ 3C フレーム形式 4：ENQ/CR + LF
 - ・ 4C フレーム形式 4：ENQ/CR + LF
 - ・ 4C フレーム形式 5 * 1：DLE+STX/DLE + ETX

* 1：“サムチェックあり”を設定している場合は、サムチェックコードまで含みます。

- ・ 送信するフレームのフレーム識別番号が不正である
- ・ 送信バイト数がヘッダ部サイズを満たしていない

(7) シリアルコミュニケーション機能による発信時のエラーコード

シリアルコミュニケーション機能による発信時にエラーが発生したとき、CPU ユニットから外部機器に送信されるエラーコード、エラー内容およびその処置を表 6.45 に示します。

表 6.45 CPU ユニットから外部機器に送信されるエラーコード一覧

エラーコード (16進)	エラー項目	エラー内容	処置方法
4000H ～ 4FFFH	—	CPU ユニットが検出したエラー（シリアルコミュニケーション機能以外で発生したエラー）	QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）を参照し処置する。
7153H	フレーム長エラー	受信伝文の長さが許容範囲を超えている。	送信伝文を見直す。 アクセス点数が許容範囲内の伝文構成にする。
7155H	モニタ未登録エラー	モニタ登録をする前にモニタ要求をした。	事前にモニタしたいデバイスを登録してからモニタ要求をする。
7164H	要求内容エラー	要求内容またはデバイス指定方法に誤りがある。	相手機器の送信伝文／要求内容を確認し、修正して再度発信する。
7167H	RUN 中不可	RUN 中書き込み可否の設定時に書き込みのコマンドを指定した。	<ul style="list-style-type: none"> • RUN 中書き込み可能に設定変更して再度発信を行う。 • CPU ユニットの STOP にして再度発信する。
7168H		RUN 中に実行できないコマンドを指定した。	CPU ユニットの STOP にして再度発信する。
716DH	モニタ登録エラー	QnA 互換 3C / 4C フレームでモニタ登録していない。	モニタ登録をやりなおす。
7E40H	コマンドエラー	存在しないコマンド、またはサブコマンドを指定した。	相手機器の送信伝文を確認、修正して再度発信する。
7E41H	データ長エラー	ランダム書き込み／読出し時発信可能点数を越えて指定した。	相手機器の送信伝文を確認、修正して再度発信する。
7E42H	データ数エラー	要求点数がコマンドに対して範囲を越えている。	相手機器の送信伝文を確認、修正して再度発信する。
7E43H	デバイスエラー	存在しないデバイスを指定した。 該当コマンドで指定できないデバイスを指定した。	相手機器の送信伝文を確認、修正して再度発信する。
7E47H	連続要求エラー	応答伝文を返す前に次の要求を受信した。	相手機器から連続要求を出さない。 タイム 1 の監視時間を相手機器側のタイムアウト時間と合わせる。
7E4FH	デバイス点数エラー	アクセス点数が正しくない。	相手機器の送信伝文を確認し、修正して再度発信を行う。
7E5FH	要求先ユニット I/O 番号エラー	要求先ユニット I/O 番号に誤りがある。	データ送信先のユニット I/O 番号を修正する。
7E64H	登録点数範囲エラー	登録点数（ワード／ビット）が範囲外である。	登録点数（ワード／ビット）の設定値を修正する。
7F01H	バッファフルエラー	受信データ処理が完了する前に次のデータを受信した。	相手機器とハンドシェイクをとるなどにより送信間隔をあげる。
7F21H	受信ヘッダ部エラー	コマンド（フレーム）部分に指定の誤りがある。 バイナリに変換できない ASCII コードを受信した。	相手機器の送信伝文を確認、修正して再度発信する。
7F22H	コマンドエラー	存在しないコマンド、またはデバイスを指定した。 リモートパスワード長に誤りがある。	相手機器の送信伝文を確認、修正して再度発信する。
7F23H	MC プロトコル伝文エラー	キャラクタ部のあとのデータ（ETX、CR-LF など）がない、または誤った指定をした。	相手機器の送信伝文を確認、修正して再度発信する。
7F24H	サムチェックエラー	計算したサムチェックと受信したサムチェックが合わない。	相手機器側のサムチェックを見直す。
7F67H	オーバーランエラー	CPU ユニットが受信処理を完了する前に次のデータを受信した。	通信速度を下げた再度発信する。 CPU ユニットで瞬停が発生していないか確認する。 (CPU ユニットの場、特殊レジスタ SD53 で確認可) 瞬停が発生しているときは、その要因をなくす。
7F69H	パリティエラー	<ul style="list-style-type: none"> • パリティビットの設定が合っていない。 • 相手機器の電源 ON/OFF により、回線に乱れが生じた。 • 回線上にノイズが発生している。 	<ul style="list-style-type: none"> • CPU ユニットと相手機器の設定を合わせる。 • ノイズ対策する。
7F6AH	バッファフルエラー	OS の受信バッファがオーバーフローし、受信データを読み飛ばした。	DTR 制御を行い、バッファフルにならないよう発信を行う。

エラーコード (16進)	エラー項目	エラー内容	処置方法
F□□□H	—	MELSECNET/Hネットワークシステムが検出したエラー	相手機器の送信伝文を確認し、修正して再度交信する。 (局番が指定されている可能性があります。 MELSECNET/H, Ethernet 経由の他局との交信はできません。)

6.25 サービス処理

サービス処理とは、GX Developer および外部機器との通信処理です。

サービス処理で実行される処理には、下記があります。

- インテリジェント機能ユニット経由での通信（ネットワークユニットとのリフレッシュは含みません。）
- USB ケーブル、RS-232 ケーブル経由での通信（GX Developer や GOT などとの通信）

6.25.1 ユニットサービス間隔読出し

ユニットサービス間隔は、モニタ、テスト、プログラムの書込み／読出しなどのトランジェント要求の間隔を示します。

CPU ユニットは、インテリジェント機能ユニット、ネットワークユニットまたは GX Developer のサービス間隔時間（サービス受付から次のサービス受付までの時間）をモニタできます。

これにより、外部から CPU ユニットに対してどのくらいの頻度でアクセスが発生しているかが分かります。

(1) 読出し方法

表 6.46 に示す特殊リレーおよび特殊レジスタを操作します。

表 6.46 ユニットサービス間隔時間を読み出す特殊リレーおよび特殊レジスタ

番号	名称	内容
SM551	ユニットサービス間隔読出し	OFF → ON にすると、特殊レジスタ SD550 で指定したインテリジェント機能ユニットのユニットサービス間隔時間を SD551 ~ 552 に読み出します。(ON: 読出し, OFF: 無処理)
SD550	サービス間隔測定ユニット	ユニットサービス間隔時間を測定するユニットの入出力番号を設定します。 CPU ユニットの RS-232 または USB インタフェースに接続する周辺機器の入出力番号は FFFFH にします。
SD551, SD552	サービス間隔時間	SM551 を ON したときに、SD550 で指定したユニットからのサービス間隔時間を格納します。 • SD551: 1ms 単位 (0 ~ 65535 の範囲) • SD552: 100 μ s 単位 (0 ~ 900 の範囲, 100 μ s ごとに格納) (例) ユニットサービス間隔時間が 123.4ms の場合は、SD551=123, SD552=400

(2) プログラム例

X/Y160 のインテリジェント機能ユニットの、ユニットサービス間隔時間を読み出す場合のプログラム例を図 6.95 に示します。



図 6.95 ユニットサービス間隔時間読出しプログラム例

Point

- ネットワークユニットからのサイクリック通信でのアクセス間隔は格納されません。
- ネットワーク上の他局の GX Developer からアクセスしているサービス間隔時間を読み出すには、ネットワークユニットの入出力番号を設定してください。

6.26 デバイス初期値 注 6.29

(1) デバイス初期値とは

プログラムで使用するデータをプログラムレスでデバイス、インテリジェント機能ユニットのバッファメモリに登録する機能です。

(2) デバイス初期値の用途

デバイス初期値を使用すると、イニシャル処理プログラムによるデバイスへのデータの設定プログラムを省略できます。

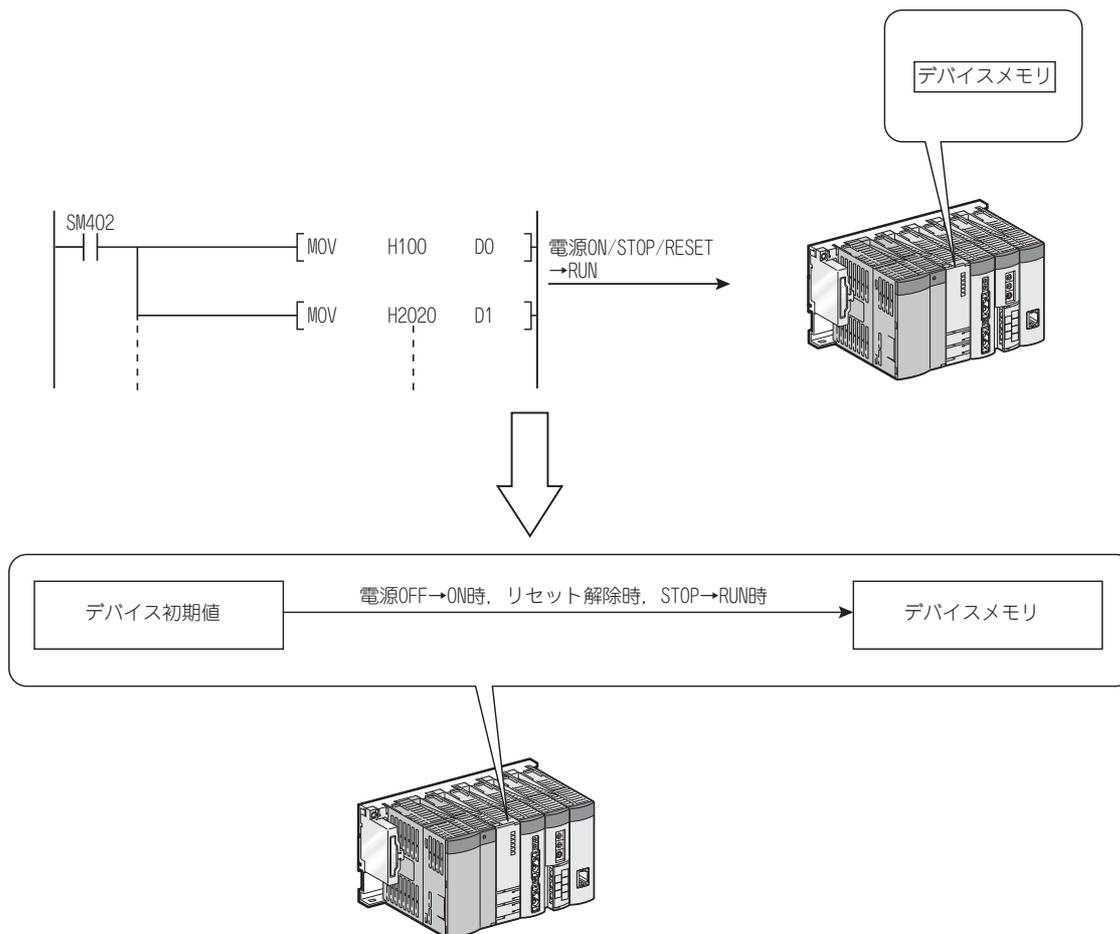


図 6.96 イニシャルプログラムによるデータのセット

注 6.29 **Basic**

ベーシックモデル QCPU でデバイス初期値を使用する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。(付 2.1)

(3) デバイス初期値が指定デバイスに書き込まれるタイミング

CPUユニットは、電源OFF→ON時、リセット解除時、STOP→RUN時に、指定されたデバイス初期値ファイルのデータを指定デバイス、インテリジェント機能ユニットのバッファメモリに書き込みます。

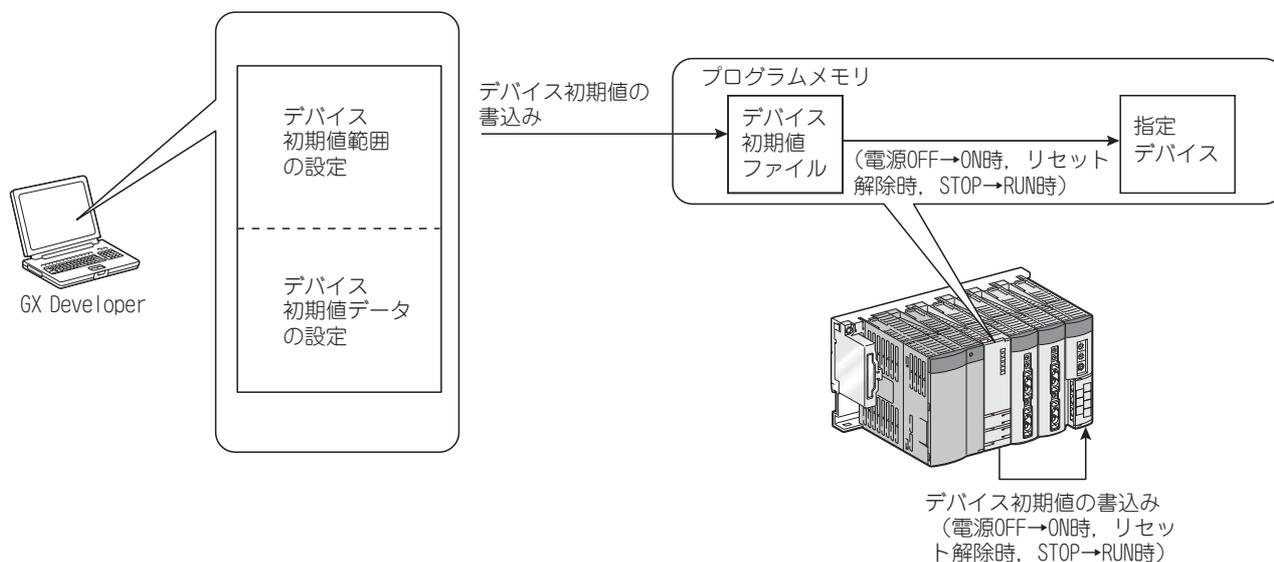


図 6.97 デバイス初期値書き込みまでの流れ

(4) 使用可能なデバイス* 1

デバイス初期値で使用可能なデバイスを下記に示します。

- タイマの現在値 (T)
- リンク特殊レジスタ (SW)
- 積算タイマの現在値 (ST)
- ファイルレジスタ (R)
- カウンタの現在値 (C)
- ファイルレジスタ (ZR)
- データレジスタ (D)
- インテリジェント機能ユニットデバイス (U□ ¥ G□)
- 特殊レジスタ (SD)
- リンクダイレクトデバイス (J□ ¥ W□, J□ ¥ SW□)
- リンクレジスタ (W)

* 1：使用可能な範囲については、9.1 節を参照してください。

(5) デバイス初期値を使用する場合の手順と設定

あらかじめ GX Developer でデバイス初期値データを作成し、CPU ユニットのプログラムメモリ、標準 ROM またはメモリカードに、デバイス初期値ファイルとして格納しておく必要があります。

- GX Developer のプロジェクトデータ一覧で、デバイス初期値のデータを追加します。
デバイス初期値範囲設定画面が表示されるので、デバイス初期値の範囲を設定します。
設定可能な点数は、1 つの範囲設定につき、最大 8000 点です。
- GX Developer のプロジェクトデータ一覧で、デバイスメモリのデータを追加します。
デバイスメモリ画面が表示されるので、上記で設定したデバイス初期値の範囲に、デバイス初期値データを設定します。

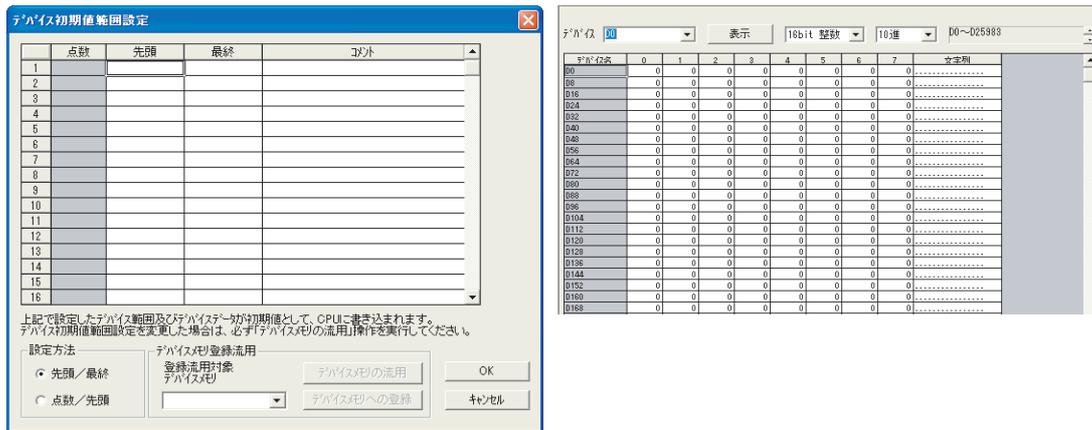


図 6.98 デバイス初期値範囲設定画面，デバイスメモリ画面

Point

デバイス初期値範囲設定を変更した場合は、必ず“デバイスマモリ登録流用”を実行してください。
デバイスマモリ登録流用の詳細については、下記マニュアルを参照してください。

☞ GX Developer Version 8 オペレーティングマニュアル

- PC パラメータの PC ファイル設定で “デバイス初期値” を設定します。

1) ベーシックモデル QCPU の場合

デバイス初期値を “使用する” に設定します。

2) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU の場合

デバイス初期値ファイルを格納するメモリとそのファイル名を設定します。

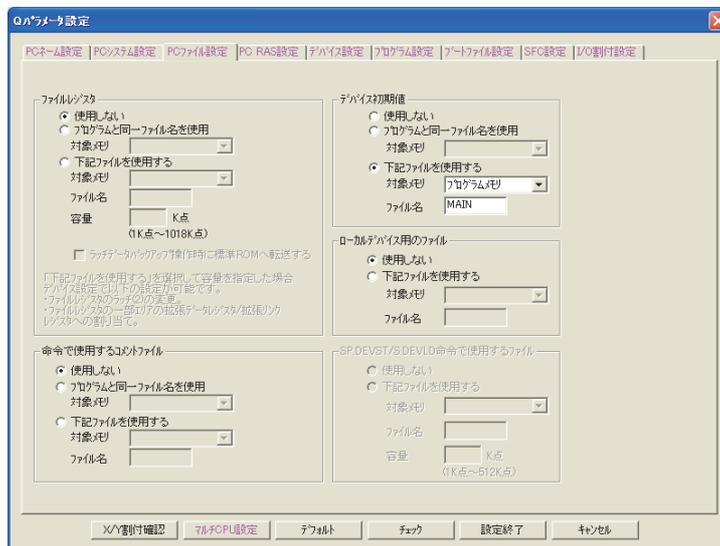


図 6.99 PC ファイル設定画面 (ハイパフォーマンスモデル QCPU の場合)

- 設定したデバイス初期値およびパラメータを CPU ユニットに書き込みます。

(6) デバイス初期値使用時の注意事項

(a) デバイス初期値とラッチ範囲が重複している場合

デバイス初期値を優先します。このためラッチ範囲のデータも、電源 OFF → ON 時にデバイス初期値データに書き換わります。

(b) STOP → RUN 時に設定したくないエリアについて

デバイス初期値は、STOP → RUN 時にも反映されます。

STOP → RUN 時に設定したくないエリア（電源 OFF → ON 時に設定しプログラムで変化させるデータ）には、デバイス初期値を使用できません。

メインルーチンプログラムで MOV 命令などにより、指定デバイスに初期値を設定するプログラムを作成してください。

またインテリジェント機能ユニットの場合は、TO 命令でバッファメモリに書き込みます。

(c) ユニット同期設定が必要なデバイスについて

デバイス初期値範囲設定において下記デバイスを指定する場合は、PC パラメータの PC システム設定で “ユニット同期設定” を設定してください。

ユニット同期設定を行わなかった場合は、対象ユニットにデバイス初期値が正常に設定されないことがあります。

- インテリジェント機能ユニットデバイス (U□¥G□)
- リンクダイレクトデバイス (J□¥W□, J□¥SW□)

備考

デバイス初期値範囲の設定、デバイス初期値データの設定操作およびデバイス初期値の CPU ユニットへの書き込み操作の詳細については、下記マニュアルを参照してください。

 GX Developer Version 8 オペレーティングマニュアル

6.27 メモリチェック機能 注 6.30

過度の電氣的ノイズなどにより CPU ユニットのメモリの内容が書き換わっていないかをチェックする機能です。

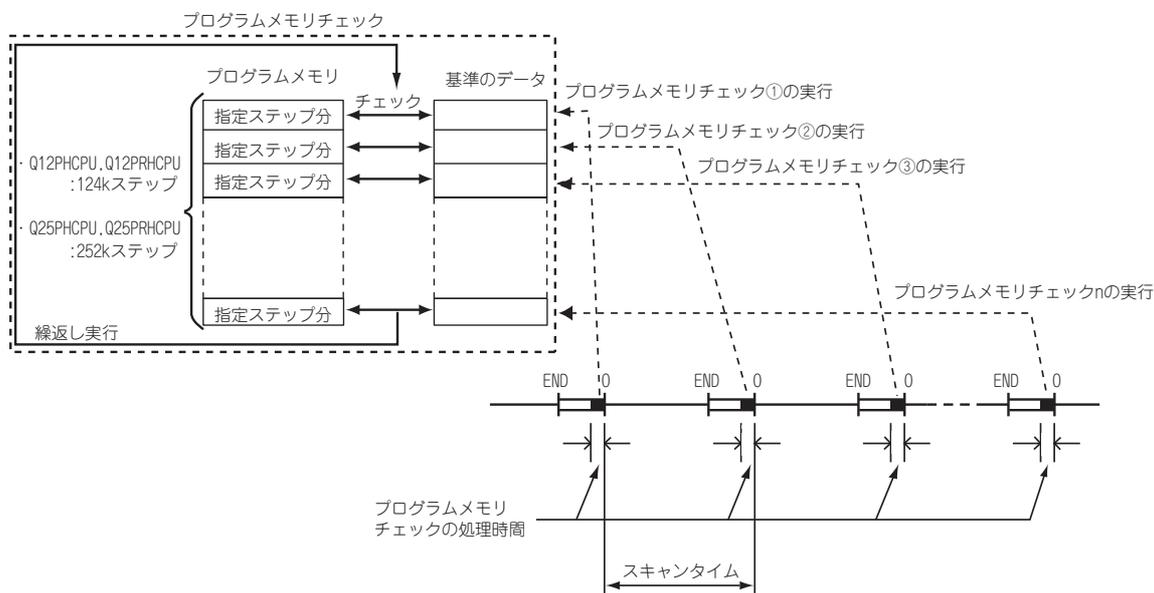


図 6.100 メモリチェックの概要

(1) メモリチェックの対象

CPU ユニットのプログラムメモリに書き込んだときのプログラム、パラメータなどのデータを基準にして、実行中のプログラムメモリのデータが基準のデータと一致しているかを STOP → RUN 切換え時と END 処理時にチェックします。

基準のデータとプログラムメモリのデータが異なる場合は、プログラムメモリのデータが書き換わっていると判断し、停止エラーの“RAM ERROR”（エラーコード：1106）になります。



注 6.30

Basic

High performance

Process

ベーシックモデル QCPU、ハイパフォーマンスモデル QCPU では、メモリチェック機能は使用できません。プロセス CPU でメモリチェック機能を使用する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。（付 2.3）

(2) メモリチェックを行うための設定

- PC パラメータの PC RAS 設定 (2) で, “プログラムメモリのチェックを行う” にチェックします。
- “1 回にチェックする容量” を設定します。

プログラムメモリチェックを行う場合の設定



設定範囲
・ Q12PHCPU, Q12PRHCPU: 1~496
(256ステップ~124kステップ)
・ Q25PHCPU, Q25PRHCPU: 1~1008
(256ステップ~252kステップ)

図 6.101 PC RAS 設定 (2) 画面

二重化 CPU では, 制御系と待機系ともメモリチェックを行います。
ただし, バックアップモードの待機系では, 1 回にチェックする容量の設定に関係なく “1 (256 ステップ)” 固定で行います。

(3) 実行タイミング

下記のタイミングで行います。

- STOP → RUN 切換え時: プログラムメモリの全エリアのチェックを行います。
- END 処理時: 1 回にチェックする容量 × 256 (ステップ) ずつチェックを行います。

(4) 処理時間

(a) STOP → RUN 切換え時

下記に示す処理時間分だけスキャンタイムが延びます。

- Q12PHCPU, Q12PRHCPU : 434 ms
- Q25PHCPU, Q25PRHCPU : 882 ms

(b) RUN 中の END 処理時

下記に示す処理時間分だけスキャンタイムが延びます。

$$(\text{スキャンタイムの延び時間}) = 3.5 \times \frac{(\text{1回にチェックする容量}) \times 256}{1024} \text{ (ms)}$$

PC パラメータの PC RAS 設定 (2) で, 1 回にチェックする容量はスキャンタイムの延び時間を考慮して設定してください。

例えば, 1 回にチェックする容量を “4” に設定した場合の処理時間は,
 $3.5 \times 4 \times 256 \div 1024 = 3.5 \text{ (ms)}$ になります。

(5) 注意事項

(a) エラー検出の最大遅れ時間

プログラムメモリのデータの書き換えが発生してからデータの書き換えを検出するまでには、下式に示す最大遅れ時間があります。

$$\text{エラー検出の最大遅れ時間} = \frac{(\text{最大プログラムメモリ容量})^{*1}}{(\text{1回にチェックする容量}) \times 256} \times (\text{スキャンタイム}) \text{ (ms)}$$

* 1 : 各 CPU ユニットの最大プログラムメモリ容量を下記に示します。

- ・ Q12PHCPU, Q12PRHCPU : 124k ステップ (124×1024 ステップ)
- ・ Q25PHCPU, Q25PRHCPU : 252k ステップ (252×1024 ステップ)

“RAM ERROR” を検出するまではシーケンスプログラムが実行されるため、エラー検出前に他のエラーを検出するか、プログラムメモリ化けによる異常動作が起きる可能性があります。

例えば、Q12PHCPU で 1 回にチェックする容量を “4” に設定して、スキャンタイムが 10ms のときの最大遅れ時間は、 $(124 \times 1024) \div (4 \times 256) \times 10 = 1240$ (ms) になります。

プログラムメモリのフォーマット時に、ユーザ設定のシステムエリアを設定した場合は、設定した容量分プログラムメモリ容量が減少します。

ユーザ設定のシステムエリア、プログラムメモリ容量の確認方法については、5.2.2 項を参照ください。

(b) メモリチェック実行時に使用できない命令

下記命令を実行すると、“OPERATION ERROR” (エラーコード : 4105) になります。

- ・ PLOADP 命令
- ・ PUNLOADP 命令
- ・ PSWAPP 命令

(c) PC パラメータの有効条件

メモリチェックの実行/非実行は、PC パラメータを CPU ユニットに書き込んだ後、下記操作を行うと有効になります。

- ・ CPU ユニットの電源の再立上げ
- ・ CPU ユニットのリセット

(d) COM 命令実行時のメモリチェック

COM 命令を実行したときには、メモリチェックは行いません。

第7章 インテリジェント機能ユニットとの交信

(1) インテリジェント機能ユニットとは

インテリジェント機能ユニットは、入出力ユニットで処理できないアナログ量、高速パルスなどを CPU ユニットで取り扱うためのユニットです。

例えば、アナログ量は、インテリジェント機能ユニットのアナログ/デジタル変換ユニットにより、デジタル値に変換して使用します。

(2) インテリジェント機能ユニットとの交信

インテリジェント機能ユニットには、外部から取り込んだデータおよび外部へ出力するためのデータを格納しておくメモリ（バッファメモリ）があります。

CPU ユニットは、インテリジェント機能ユニットのバッファメモリからデータの書込み/読出しを行います。

7.1 CPU ユニットからインテリジェント機能ユニットとの交信

CPU ユニットとインテリジェント機能ユニットの交信方法と交信タイミングを下記に示します。

表 7.1 インテリジェント機能ユニットとの交信方法と交信タイミング

インテリジェント機能ユニットとの交信方法	交信タイミング					参 照	
	電源 OFF → ON	リセット 解除時	STOP → RUN * 2	命令実行	END 処理		
GX Configurator	初期設定	○	○	○	—	—	7.1.1 項
	自動リフレッシュ設定	—	—	—	—	○	
デバイス初期値	○	○	○	—	—	—	7.1.2 項
FROM/TO 命令 * 1	—	—	—	○	—	—	7.1.3 項
インテリジェント機能ユニットデバイス * 1	—	—	—	○	—	—	7.1.4 項
インテリジェント機能ユニット専用命令 * 1	—	—	—	○	—	—	7.1.5 項

○：実行する， —：実行しない

* 1：インテリジェント機能ユニットデバイス， FROM/TO 命令， インテリジェント機能ユニット専用命令を使用したプログラムを示しています。

* 2：RUN/STOP スイッチ（ベーシックモデル QCPU の場合は， RUN/STOP/RESET スイッチ）を STOP → RUN（RUN LED が点滅） → STOP → RUN とした場合を示しています。

Point

- インテリジェント機能ユニットとの交信に使用するデータ（初期設定など）は， CPU ユニットに格納します。格納先については， 5.1.1 項， 5.2.1 項を参照してください。
- GX Configurator で設定した初期設定を反映するには， CPU ユニットの電源 OFF → ON またはリセット解除を行ってください。

7.1.1 GX Configurator による初期設定, 自動リフレッシュ設定

GX Developer にインテリジェント機能ユニット対応の GX Configurator をアドインして初期設定, 自動リフレッシュ設定を行います。

初期設定, 自動リフレッシュ設定を行うと, インテリジェント機能ユニットとの通信プログラムを作成することなく, データの書込み/読出しができます。

(1) GX Configurator の起動

GX Developer の [ツール] → [インテリジェント機能ユニットユーティリティ] → [起動] を選択して起動します。

(2) GX Configurator による設定

A/D 変換ユニット (Q64AD) の初期設定, 自動リフレッシュ設定を行う場合の例で説明します。

(a) 初期設定

Q64AD の初期設定には, 下記に示す 4 種類があります。

- A/D 変換許可/禁止設定
- サンプリング/平均処理指定
- 時間平均/回数平均指定
- 平均時間/平均回数指定

Q64AD の初期設定は, 図 7.1 に示す GX Configurator の初期設定画面で行います。



図 7.1 初期設定画面

設定した初期設定データは, CPU ユニットのインテリジェント機能ユニットパラメータに格納されます。

(b) 自動リフレッシュ設定

自動リフレッシュ設定では、下記に示すデータを格納する際の、CPU ユニット側のデバイスを設定します。

- Q64AD のデジタル出力
- Q64AD の最大値／最小値
- エラーコード

Q64AD の自動リフレッシュ設定は、図 7.2 に示す GX Configurator の自動リフレッシュ設定画面で行います。



図 7.2 自動リフレッシュ設定画面

設定した自動リフレッシュ設定データは、CPU ユニットのインテリジェント機能ユニットパラメータに格納されます。

(3) パラメータ設定個数の制限について

GX Configurator で設定するパラメータ（初期設定、自動リフレッシュ設定）には、設定できる項目数に制限があります。

複数のインテリジェント機能ユニットを装着した場合、全インテリジェント機能ユニットのパラメータ設定個数の合計が表 7.2 の最大パラメータ設定個数を超えないように、GX Configurator の設定を行います。

表 7.2 GX Configurator で設定できるパラメータ設定個数

CPU ユニット	パラメータ設定個数	
	初期設定	自動リフレッシュ設定
ベーシックモデル QCPU, ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU, MELSECNET/H リモート I/O 局	512	256

パラメータ設定個数は、自動リフレッシュ画面の 1 行を 1 個と数えます。



この1行で設定個数1個と数えます。
空欄は個数に数えません。
この設定画面の全設定項目を加算し、
他のインテリジェント機能ユニット
の個数と合計します。

図 7.3 パラメータ設定個数の考え方

(4) 注意事項

AnS/A シリーズ対応の特殊機能ユニットでは、GX Configurator を使用した設定はできません。

備考

.....
GX Configurator の詳細については、使用するインテリジェント機能ユニットのマニュアルを参照してください。
.....

7.1.2 デバイス初期値による初期設定

(1) デバイス初期値

デバイス初期値（ 6.26 節）の使用により、インテリジェント機能ユニットの初期設定をプログラムレスで行うことができます。

設定されたデバイス初期値は、CPU ユニットの電源 OFF → ON 時、リセット解除時、STOP → RUN 時に、CPU ユニットからインテリジェント機能ユニットに書き込まれます。

(2) デバイス初期値の設定

GX Developer で以下の設定を行います。

- デバイスメモリにデバイス初期値として使用するインテリジェント機能ユニットデバイス（ 9.5.1 項）のデータを設定します。
- デバイス初期値設定で、デバイス初期値として使用するインテリジェント機能ユニットデバイスの範囲を指定します。

7.1.3 FROM/TO 命令による交信

FROM 命令は、インテリジェント機能ユニットのバッファメモリから読出したデータを指定したデバイスに格納します。

TO 命令は、指定したデバイスのデータをインテリジェント機能ユニットのバッファメモリに書き込みます。

備考

- FROM/TO 命令の詳細については、下記マニュアルを参照してください。
 MELSEC-Q/L プログラミングマニュアル（共通命令編）
- インテリジェント機能ユニットのバッファメモリの詳細については、下記マニュアルを参照してください。
 使用するインテリジェント機能ユニットのマニュアル

7.1.4 インテリジェント機能ユニットデバイスによる交信

(1) インテリジェント機能ユニットデバイス

インテリジェント機能ユニットデバイス（ 9.5.1 項）は、インテリジェント機能ユニットのバッファメモリを CPU ユニットのデバイスとして表現したものです。

インテリジェント機能ユニットのバッファメモリに格納されているデータを、デバイスメモリと同様にシーケンス命令で扱うことができます。

例えば、入出力番号が X/Y20 ~ 2F のインテリジェント機能ユニットのバッファメモリアドレス 0 に “100” を書き込む場合は、次のようにプログラミングします。



図 7.4 インテリジェント機能ユニットデバイスの使用方法

(2) FROM/TO 命令との相違点

インテリジェント機能ユニットデバイスは、CPU ユニットのデバイスとして扱うことができるため、インテリジェント機能ユニットから読出しとデータの加工を 1 命令で行います。

例えば、インテリジェント機能ユニットからの読出しとデータの加算を行い、結果を D2 へ格納する処理は、下記のようにプログラミングします。

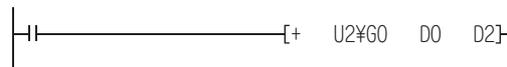


図 7.5 インテリジェント機能ユニットデバイスの使用例

これにより、プログラム全体のステップ数を減らすことができます。

命令処理時間は、命令の実行時間にインテリジェント機能ユニットとのアクセス時間を加算した時間になります。

インテリジェント機能ユニットデバイスは、命令実行ごとにインテリジェント機能ユニットへのアクセスを行います。シーケンスプログラム内でインテリジェント機能ユニットデバイスを複数使用し、バッファメモリのデータを書込み／読出しする場合は、FROM/TO 命令を使用してプログラムの 1 箇所を書込み／読出しを行ってください。

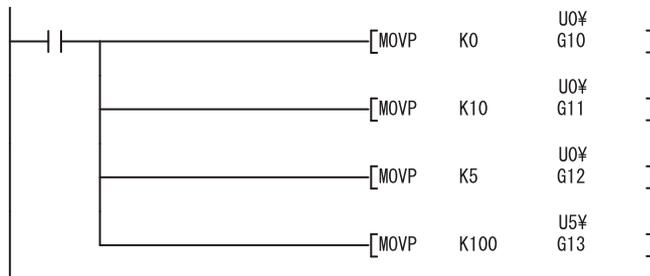


図 7.6 インテリジェント機能ユニットデバイスを複数使用して書き込む場合

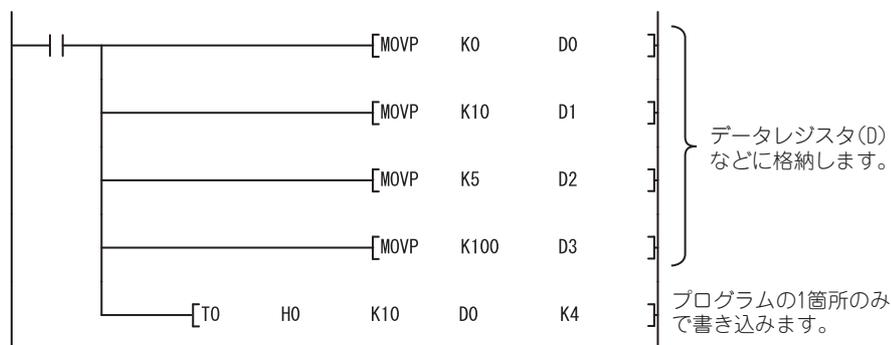


図 7.7 TO 命令を使用してプログラムの 1 箇所を書き込む場合

7.1.5 インテリジェント機能ユニット専用命令による交信

(1) インテリジェント機能ユニット専用命令とは

インテリジェント機能ユニット専用命令は、インテリジェント機能ユニットの機能を使用する場合に、プログラミングを容易にするための命令です。

(a) シリアルコミュニケーションユニット用専用命令 (OUTPUT 命令) での例

OUTPUT 命令を使用すると、シリアルコミュニケーションユニットのバッファメモリアドレスを意識することなく、無手順プロトコルにより相手機器との通信を行うことができます。

[OUTPUT 命令による送信]

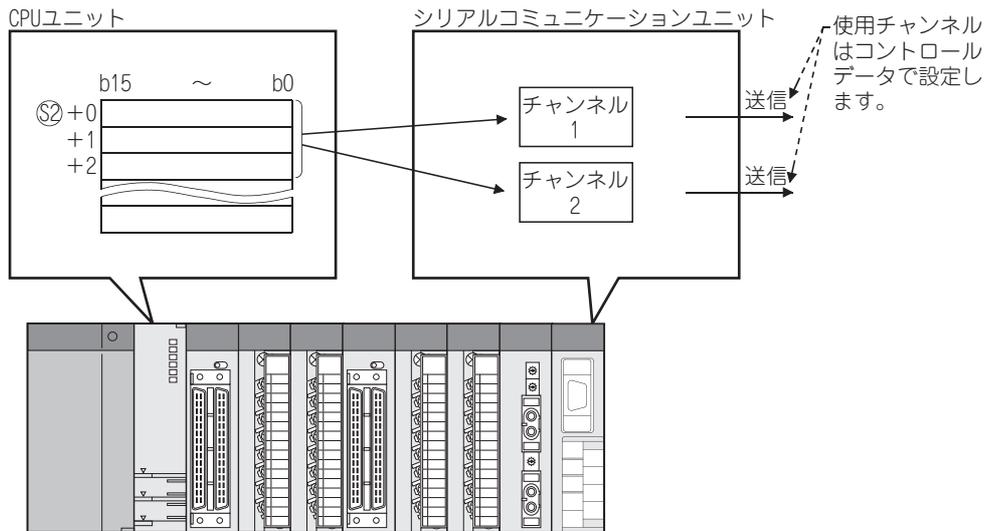


図 7.8 OUTPUT 命令を使用した相手機種との交信

(2) インテリジェント機能ユニット専用命令の処理

インテリジェント機能ユニット専用命令によっては、完了デバイスを指定できます。

この完了デバイスは、命令の実行完了で1スキャンONします。

同一インテリジェント機能ユニットに複数のインテリジェント機能ユニット専用命令を使用する場合は、完了デバイスのONで次のインテリジェント機能ユニット専用命令を実行してください。

(3) 注意事項

(a) 完了デバイスが ON する前に RUN から STOP にした場合

インテリジェント機能ユニット専用命令を実行し、完了デバイスが ON する前に CPU ユニットの RUN から STOP にすると、次に RUN した 1 スキャン後に完了デバイスが ON します。

(b) 使用可能な範囲について

MELSECNET/H のリモート I/O 局に装着したインテリジェント機能ユニットに対して、インテリジェント機能ユニット専用命令は実行できません。

基本ベースユニット、増設ベースユニットのインテリジェント機能ユニットに対してのみ実行できます。

備考

インテリジェント機能ユニット専用命令、完了デバイスについては、下記マニュアルを参照してください。

 使用するインテリジェント機能ユニットのマニュアル

7.2 AnS/A シリーズ対応の特殊機能ユニットにアクセスする場合

注 7.1

(1) 特殊機能ユニットへのアクセス高速化に対する影響

Q シリーズ CPU ユニットは、処理が高速化されたためスキャンタイムが短縮されています。

特殊機能ユニットに対して、短いスキャンで頻繁に FROM/TO 命令を実行した場合は、対象となる特殊機能ユニットの処理が正常に完了しなくなる恐れがあります。

(2) 特殊機能ユニットへのアクセスの高速化に対する対策

- 1) CPU ユニットのタイマ、コンスタントスキャンなどを利用し、FROM/TO 命令の実行間隔を処理時間・変換時間に合わせてください。
- 2) SM415 (2n ms クロック)、SD415 (2n ms クロック設定) で実行間隔を調整してください。SD415 の初期値は“30”に設定されており、SM415 を FROM/TO 命令のインタロックに使用すると 120ms ごとに FROM/TO 命令が実行されます。



図 7.9 対策プログラム例

Point

SM415 のクロックを変更する場合は、SD415 に変更値を格納してください。

SM415、SD415 の詳細については、下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編)



注 7.1

Basic

Process

Redundant

ベーシックモデル QCPU、プロセス CPU、二重化 CPU では、AnS/A シリーズ対応の特殊機能ユニットは使用できません。

第 8 章 パラメータ

本章では、シーケンサシステム構築時に設定するパラメータについて説明します。

(1) パラメータの種類

CPU ユニットのパラメータには、下記のものがあります。

- PC パラメータ (☞ 8.1 節)
シーケンサを単独で使用する際に設定します。
- 二重化パラメータ (☞ 8.2 節)
二重化 CPU を使用して、二重化システムを構築時に設定します。
- ネットワークパラメータ (☞ 8.3 節)
CC-Link IE コントローラネットワークユニット、MELSECNET/H ユニット、Ethernet ユニット、
CC-Link ユニットのシーケンサと組み合わせて使用する際に設定します。
- リモートパスワード (☞ 8.4 節)
Ethernet ユニット、シリアルコミュニケーションユニット、モデムインタフェースユニットのリモートパ
スワード機能使用時に設定します。

(2) パラメータの設定方法

GX Developer で設定します。

設定操作については、下記マニュアルを参照してください。

☞ GX Developer Version 8 オペレーティングマニュアル

Point

GX Developer でグレー表示（選択不可）の設定項目は、未対応のため設定できません。

備考

本章の表に示すパラメータ No. は、パラメータ設定でエラーが発生したときに特殊レジスタ（SD16～26）に格納されます。

パラメータ No. により、パラメータエラーの箇所を特定してください。

特殊レジスタ（SD16～26）の一覧については、下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）

パラメータの反映手順については第 11 章を参照してください。

8.1 PC パラメータ

PC パラメータの設定画面と各パラメータの内容について説明します。

8.1.1 ベーシックモデル QCPU

(1) PC ネーム設定

使用する CPU ユニットのラベル, コメントを設定します。

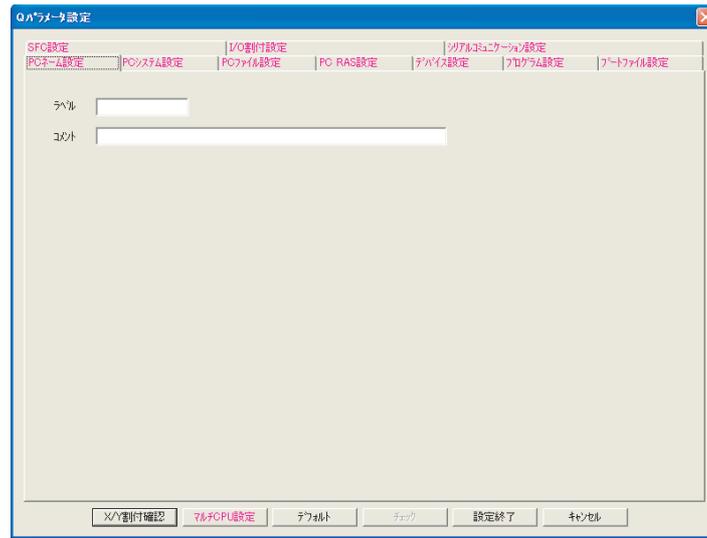


図 8.1 PC ネーム設定

表 8.1 PC ネーム設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
ラベル	0000 _H	CPU ユニットのラベル (名称, 用途) を設定します。	半角 10 文字以下	設定なし	—
コメント	0001 _H	CPU ユニットのラベルのコメントを設定します。	半角 64 文字以下	設定なし	—

(2) PC システム設定

CPU ユニットを使用する場合に必要な設定を行います。
デフォルト値のままでも制御できます。



図 8.2 PC システム設定

表 8.2 PC システム設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
タイマ 時限設定	低 速	低速タイマ/高速タイマの時限 を設定します。	1ms ~ 1000ms (1ms 単位)	100ms	9.2.10 項
	高 速		0.1ms ~ 100.0ms (0.1ms 単位)	10.0ms	9.2.10 項
RUN- PAUSE 接点	RUN	CPU ユニットの RUN/PAUSE を制御する接点を設定します。 PAUSE 接点のみの設定はできま せん。(RUN 接点、RUN 接点+ PAUSE 接点の設定は可。)	X0 ~ 7FF	設定なし	6.6.1 項
	PAUSE				6.6.2 項
リモートリセット	1002 _H	GX Developer からのリモート リセット操作の許可/禁止を設 定します。	許可する/許可しない	許可しない	6.6.3 項
STOP → RUN 時の出力 モード	1003 _H	STOP 状態から RUN 状態に切 換えたときの出力 (Y) 状態を設 定します。	STOP 前の出力 (Y) 状態を出力/出力 (Y) をクリア (出力は 1 スキャン後)	STOP 前の出力 (Y) 状態を出力	6.4 節
インテリジェント機能ユ ニット設定 (割り込みポイ ンタ設定)	100A _H	割り込みポインタ (I50 ~ 127) の 割付け、インテリジェント機能 ユニットの先頭 I/O No.、先頭 SI No. を設定します。	<ul style="list-style-type: none"> 先頭 I/O No. 先頭 SI No. I50 ~ 127 	設定なし	6.23 節
ユニット同期設定	100C _H	CPU ユニットの立上りをインテ リジェント機能ユニットの立上 りに同期させるか、同期させな いかを設定します。	インテリジェント機能ユニットの立上 がりを同期する/同期しない	インテリジェント 機能ユニットの立 上がりを同期する	—

(次ページへつづく)

表 8.2 PC システム設定一覧 (つづき)

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
空きスロット点数	1007H	基本ベースユニット/増設ベースユニットの空きスロット点数を設定します。	【Q00JCPU】 0点, 16点, 32点, 64点, 128点, 256点 【Q00CPU, Q01CPU】 0点, 16点, 32点, 64点, 128点, 256点, 512点, 1024点	16点	4.2.2 項
システム 割り込み 設定	割り込みカウンタ 先頭 No.	割り込みカウンタの先頭 No. を設定します。	C0 ~ 13184 (カウンタ設定点数 -128 まで設定可)	設定なし	9.2.11 項 (4)
	In 定周期間隔 (n : 28 ~ 31)	割り込みポインタ (128 ~ 31) の実行間隔を設定します。	2ms ~ 1000ms (1ms 単位)	<ul style="list-style-type: none"> • I28 : 100ms • I29 : 40ms • I30 : 20ms • I31 : 10ms 	9.10 節
割込プログラム/定周期プログラム設定 * 1	1008H	割り込みプログラムの高速実行を行うか、行わないかを設定します。	高速実行しない/高速実行する	高速実行しない	2.2.3 項

* 1 : ベーシックモデル QCPU では、定周期実行タイププログラムは実行できません。

(3) PC ファイル設定

CPU ユニットで使用するファイルを設定します。



図 8.3 PC ファイル設定

表 8.3 PC ファイル設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
デバイス初期値	1102H	CPU ユニットで使用するデバイス初期値のファイルを設定します。	使用しない/使用する	使用しない	6.26 節

(4) PC RAS 設定

RAS 機能のための各種設定を設定します。



図 8.4 PC RAS 設定

表 8.4 PC RAS 設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
WDT (ウォッチドッグタイマ) 設定	3000 _H	CPU ユニットのウォッチドッグタイマの時間を設定します。	10ms ~ 2000ms (10ms 単位)	200ms	6.16 節
エラー時の運転モード	演算エラー	エラーを検出したときの CPU ユニットの動作モードを設定します。	停止 / 続行	停止	6.17 節
	拡張命令エラー* 1				
	ヒューズ断				
	ユニット照合エラー				
	インテリユニットプログラム実行エラー				
エラーチェック	バッテリーチェックを行う	指定エラーを検出するか、検出しないかを設定します。	チェックを行う / 行わない	チェックを行う	6.17 節
	ヒューズ断チェックを行う				
	ユニット照合を行う				
コンスタントスキャン	3003 _H	コンスタントスキャン時間を設定します。	1ms ~ 2000ms (1ms 単位)	設定なし	6.2 節

* 1：拡張命令エラーは、将来拡張用の設定です。

(5) デバイス設定

デバイスごとに使用点数、ラッチ範囲を設定します。

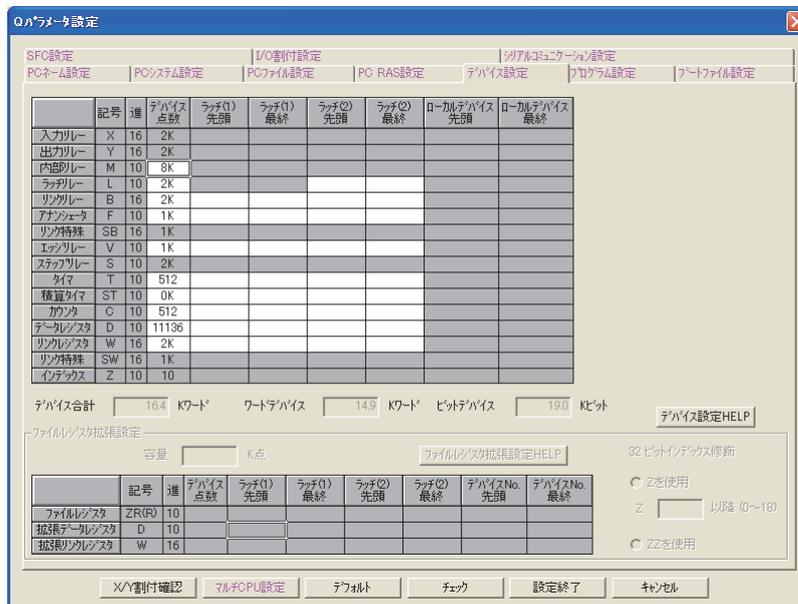


図 8.5 デバイス設定

表 8.5 デバイス設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
デバイス点数* 1	2000 _H	システムに合わせてデバイスの使用点数を設定します。	X(2k点), Y(2k点), S(2k点), SB(1k点), SW(1k点)は固定。上記点数分(1.5kワード)を含み合計 16.4kワードの範囲で設定可能。 1デバイス：最大 32k点 (ビットデバイスの合計点数に制約はありません。)	<ul style="list-style-type: none"> • X : 2k点 • Y : 2k点 • M : 8k点 • L : 2k点 • B : 2k点 • F : 1k点 • SB : 1k点 • V : 1k点 • S : 2k点 • T : 512点 • ST : 0k点 • C : 512点 • D : 11136点 • W : 2k点 • SW : 1k点 	9.1節, 9.2節
ラッチ(1)先頭/最終 (ラッチクリア有効)	2001 _H	リモートラッチクリア操作でクリア可能にするラッチ範囲(先頭デバイス番号/最終デバイス番号)を設定します。	B, F, V, T, ST, C, D, Wの各デバイスを1範囲のみ設定。	設定なし	3.7節, 6.3節
ラッチ(2)先頭/最終 (ラッチクリア無効)	2002 _H	リモートラッチクリア操作でクリア不可にするラッチ範囲(先頭デバイス番号/最終デバイス番号)を設定します。	L, B, F, V, T, ST, C, D, Wの各デバイスを1範囲のみ設定。	設定なし	3.7節, 6.3節

* 1 : デバイス点数を変更する場合は、ネットワークユニットのリフレッシュ範囲やインテリジェントユニットの自動リフレッシュ範囲がデバイス点数の範囲外とならないようにする必要があります。
該当デバイス範囲を超えた場合は、他のデバイスにデータが書き込まれたり、エラーになることがあります。

(6) ブートファイル設定

標準 ROM からのブートを行うか、行わないかを設定します。

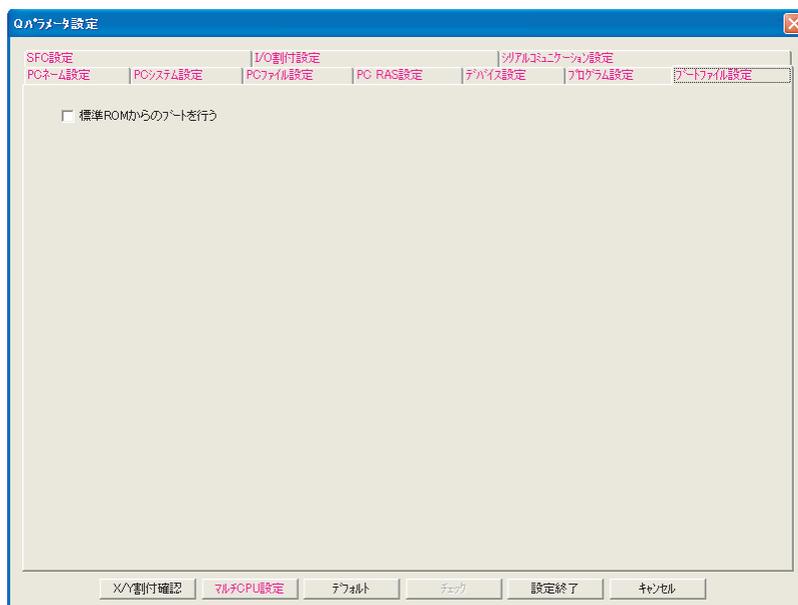


図 8.6 ブートファイル設定

表 8.6 ブートファイル設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
ブートファイル設定	—	標準 ROM からのブートを行うか、行わないかを設定します。	ブートを行わない／ブートを行う	ブートを行わない	5.1.5 項

(7) SFC 設定

SFC プログラム使用時の SFC プログラム起動モード， 起動条件， ブロック停止時の出力モードを設定します。



図 8.7 SFC 設定

表 8.7 SFC 設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
SFC プログラム起動モード	8002 _H	SFC プログラム使用時の SFC プログラム起動モード， 起動条件， ブロック停止時の出力モードを設定します。	MELSEC-Q/L/QnA プログラミングマニュアル（SFC 編）を参照してください。	イニシャルスタート	—
起動条件	8003 _H			ブロック 0 を自動起動する	
ブロック停止時の出力モード	8006 _H			OFF する	

(8) I/O 割付設定

システムの名ユニットの装着状態を設定します。



図 8.8 I/O 割付設定

表 8.8 I/O 割付設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
I/O 割付	種別	装着しているユニットの種別を設定します。	CPU2 号機～3 号機：n 号機/空き (CPU ユニットの装着しないスロットは“CPU (空き)”を設定する。) 空き、入力、高速入力、出力、インテリ、入出力混合、割込み	設定なし	4.2 節
	形名	装着しているユニットの形名を設定します。(ユーザのメモ。CPU ユニットの使用しない。)	半角 16 文字		
	点数	各スロットの点数を設定します。	【Q00JCPU】 0 点、16 点、32 点、48 点、64 点、128 点、256 点 【Q00CPU、Q01CPU】 0 点、16 点、32 点、48 点、64 点、128 点、256 点、512 点、1024 点		
	先頭 XY (先頭入出力番号)	各スロットの先頭入出力番号を設定します。	【Q00JCPU】 0 _H ～F0 _H 【Q00CPU、Q01CPU】 0 _H ～3F0 _H		

(次ページへつづく)

表 8.8 I/O 割付設定一覧 (つづき)

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照	
基本設定	ベース形名	使用している基本ベースユニット、増設ベースユニットの形名を設定します。(ユーザのメモ。CPU ユニツは使用しない。)	半角 16 文字	設定なし	4.1.2 項	
	電源ユニット形名	基本ベースユニット、増設ベースユニットに装着している電源ユニットの形名を設定します。(ユーザのメモ。CPU ユニツは使用しない。)	半角 16 文字			
	増設ケーブル形名	増設ケーブル形名を設定します。(ユーザのメモ。CPU ユニツは使用しない。)	半角 16 文字			
	スロット数	基本ベースユニット、増設ベースユニットのスロット数を設定します。 スロット数の設定は、すべてのベースユニットに行います。	2, 3, 5, 8, 10, 12			
スイッチ設定	0407 _H	インテリジェント機能ユニットの各種スイッチを設定します。	使用するインテリジェント機能ユニットのマニュアル参照	設定なし	6.10 節	
詳細設定	エラー時出力モード	0403 _H	管理 CPU が停止エラーになったとき出力をクリアするか/保持するかを設定します。	クリア/保持	クリア	6.8 節
	H/W エラー時 CPU 動作モード	4004 _H	インテリジェント機能ユニットのハードウェア異常時、管理 CPU の運転を停止させるか、続行させるかを設定します。	停止/続行	停止	6.9 節
	I/O 応答時間	0405 _H	入力ユニット、高速入力ユニット、入出力混合ユニット、割込みユニットの応答時間を設定します。	<ul style="list-style-type: none"> 入力、入出力混合：1ms, 5ms, 10ms, 20ms, 70ms 高速入力、割込み：0.1ms, 0.2ms, 0.4ms, 0.6ms, 1ms 	<ul style="list-style-type: none"> 入力、入出力混合：10ms 高速入力、割込み：0.2ms 	6.7 節
	管理 CPU	0406 _H	入出力ユニット、インテリジェント機能ユニットの管理 CPU を設定します。	1号機, 2号機, 3号機	1号機	QCPU ユーザーズマニュアル (マルチ CPU システム編)

(9) シリアルコミュニケーション設定 注 8.1

CPU ユニットのシリアルコミュニケーション機能を使用したときの伝送速度、サムチェック、伝送ウェイト時間、RUN 中書込の可否を設定します。



図 8.9 シリアルコミュニケーション設定

表 8.9 シリアルコミュニケーション設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
シリアルコミュニケーション機能を使用する	100E _H	シリアルコミュニケーション機能を使用するとき、チェックマークを付けます。	シリアルコミュニケーション機能を使用しない/使用する	シリアルコミュニケーション機能を使用しない	6.24 節
伝送速度		相手機器とデータ交信を行うための伝送速度を設定します。	9.6kbps/19.2kbps /38.4kbps/57.6kbps /115.2kbps	19.2kbps	
サムチェック		シリアルコミュニケーション機能によるデータ交信時、送信伝文、受信伝文にサムチェックコードを付加するかしないかを、相手機器の仕様に合わせ設定します。	なし/あり	あり	
伝送ウェイト時間		相手機器がデータを送信後、すぐにデータを受信できないときのベーシックモデル QCPU 側の送信待ち時間を設定します。	ウェイトなし / 10ms ~ 150ms (10ms 単位)	ウェイトなし	
RUN 中書込設定		相手機器から CPU ユニットへデータを書き込むとき、CPU ユニットが RUN 中でも書き込むか、書き込まないかを設定します。	許可しない/許可する	許可しない	



注 8.1

Basic

Q00JCPU では、シリアルコミュニケーション機能を使用できません。

(10)X/Y 割付確認

I/O 割付け, Ethernet/CC IE/MELSECNET 設定, CC-Link 設定で設定した内容の確認を行います。



図 8.10 X/Y 割付確認

表 8.10 X/Y 割付確認一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
X/Y 割付確認	—	I/O 割付け, Ethernet/CC IE/MELSECNET 設定, CC-Link 設定で設定した内容の確認ができます。	—	—	—

(11) マルチ CPU 設定

マルチ CPU システムを構築するための設定を行います。



図 8.11 マルチ CPU 設定

表 8.11 マルチ CPU 設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
CPU 台数	0E00H	マルチ CPU システムで使用する CPU ユニットの台数を設定します。	1 台～ 3 台	1 台	
動作モード	0E01H	2 号機～ 3 号機の CPU ユニットが停止エラーになったときのマルチ CPU システムの動作を設定します。1 号機が停止エラーになったとき、マルチ CPU システムは停止します。(固定)	n 号機のエラーで全号機停止する / 停止しない	n 号機のエラーで全号機停止する	
グループ外の入出力設定	0E04H	グループ外の入力状態を取り込む	グループ外の入力状態を取り込まない / グループ外の入力状態を取り込む	グループ外の入力状態を取り込まない	QCPU ユーザーズマニュアル (マルチ CPU システム編)
		グループ外の実出力状態を取り込む	グループ外の実出力状態を取り込まない / グループ外の実出力状態を取り込む	グループ外の実出力状態を取り込まない	
リフレッシュ設定	E002H E003H	マルチ CPU システムの各 CPU ユニット間で、自動リフレッシュによりデータの書き込み / 読出しを行うデバイスと点数を設定します。	【各 CPU 送信範囲】 1 号機 : 0 ~ 320 点 (2 点単位) 2, 3 号機 : 0 ~ 2048 点 (2 点単位) / 1 台 最大 4416 点 / 1 システム 【CPU 側デバイス】 B, M, Y, D, R, ZR 指定デバイス番号から送信範囲で設定した点数分のデバイスを占有する。 送信範囲 1 点で B, M, Y は 16 点を占有。 送信範囲 1 点で D, W, R, ZR は 1 点を占有。	設定なし	

8.1.2 ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU

(1) PC ネーム設定

使用する CPU ユニットのラベル, コメントを設定します。

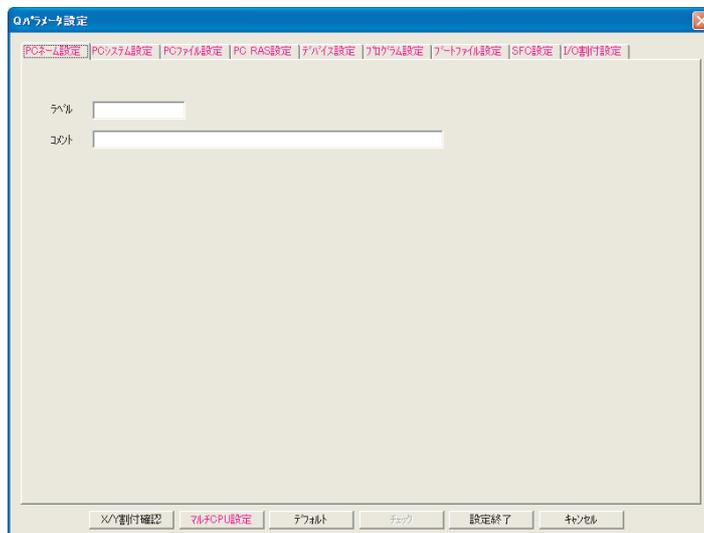


図 8.12 PC ネーム設定

表 8.12 PC ネーム設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
ラベル	0000 _H	CPU ユニットのラベル (名称, 用途) を設定します。	半角 10 文字以下	設定なし	—
コメント	0001 _H	CPU ユニットのラベルのコメントを設定します。	半角 64 文字以下	設定なし	—

(2) PC システム設定

CPU ユニットを使用する場合に必要な設定を行います。

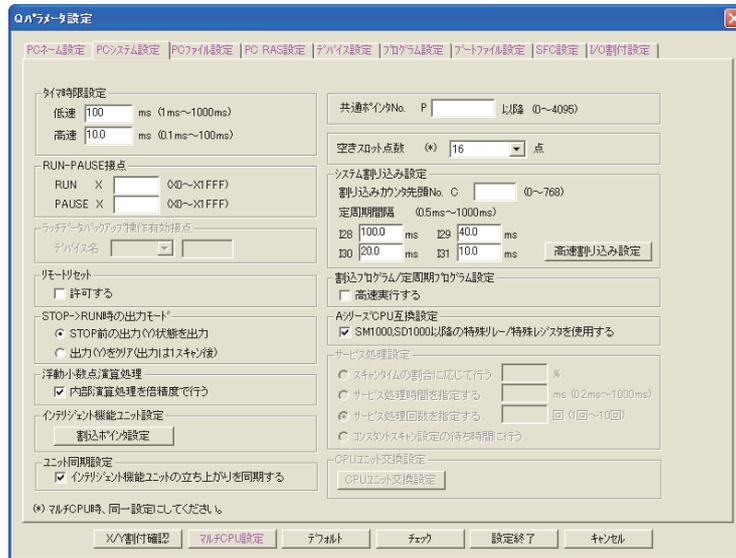


図 8.13 PC システム設定

表 8.13 PC システム設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照	
タイマ 時限設定	1000 _H	低速タイマ/高速タイマの時限を設定します。	1ms ~ 1000ms (1ms 単位)	100ms	9.2.10 項	
			0.01ms ~ 100.0ms (0.01ms 単位)	10.0ms	9.2.10 項	
RUN- PAUSE 接点	1001 _H	CPU ユニットの RUN/PAUSE を制御する接点を設定します。PAUSE 接点のみの設定はできません。	X0 ~ 1FFF	設定なし	6.6.1 項	
					6.6.2 項	
リモトリセット	1002 _H	GX Developer からのリモトリセット操作の許可/禁止を設定します。	許可する/許可しない	許可しない	6.6.3 項	
STOP → RUN 時の 出力モード	1003 _H	STOP 状態から RUN 状態に切換えたときの出力 (Y) 状態を設定します。	STOP 前の出力 (Y) 状態を出力/出力 (Y) をクリア (出力は 1 スキャン後)	STOP 前の出力 (Y) 状態を出力	6.4 節	
インテリジェント機能 ユニット設定 (割 込ポイント設定)	100A _H	割り込みポイント (I50 ~ 255) の割り付け、インテリジェント機能ユニットの先頭 I/O No.、先頭 SI No. を設定します。	<ul style="list-style-type: none"> 先頭 I/O No. 先頭 SI No. I50 ~ 255 	設定なし	6.23 節	
ユニット同期設定	100C _H	CPU ユニットの立上りをインテリジェント機能ユニットの立上りに同期させるか、同期させないかを設定します。	インテリジェント機能ユニットの立上りを同期する/同期しない	インテリジェント機能ユニットの立上りを同期する	—	
共通ポイント No.	1005 _H	共通ポイントとして使用するポイントの先頭 No. を設定します。	P0 ~ 4095	設定なし	9.9.2 項	
空きスロット点数	1007 _H	基本ベースユニット/増設ベースユニットの空きスロット点数を設定します。	0 点, 16 点, 32 点, 64 点, 128 点, 256 点, 512 点, 1024 点	16 点	4.2.2 項	
システム 割り込み 設定	In 定周期 間隔 (n : 28 ~ 31)	1008 _H	割り込みポイント (I28 ~ 31) の実行間隔を設定します。	0.5ms ~ 1000ms (0.5ms 単位)	<ul style="list-style-type: none"> I28:100.0ms I29:40.0ms I30:20.0ms I31:10.0ms 	9.10 節
割込プログラム/定 周期プログラム設定	1008 _H	割り込みプログラム/定周期プログラムの高速実行を行うか、行わないかを設定します。	高速実行しない/高速実行する	高速実行しない	2.2.3 項, 2.3.5 項	

(3) PC ファイル設定

CPU ユニットで使用する各種ファイルを設定します。

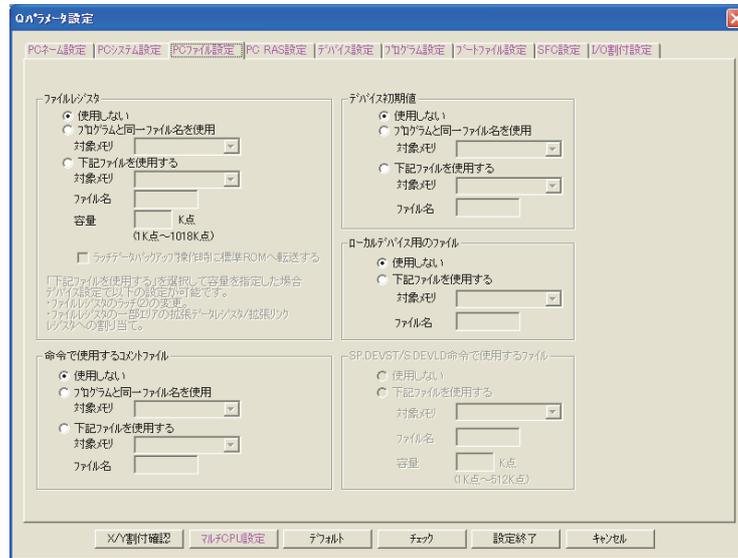


図 8.14 PC ファイル設定

表 8.14 PC ファイル設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
ファイルレジスタ	1100 _H	プログラムで使用するファイルレジスタのファイルを設定します。	<ul style="list-style-type: none"> 使用しない プログラムと同一ファイル名を使用 下記ファイルを使用する 	使用しない	9.7 節
命令で使用するコメントファイル	1101 _H	プログラムで使用するデバイスコメントのファイルを設定します。	<ul style="list-style-type: none"> 使用しない プログラムと同一ファイル名を使用 下記ファイルを使用する 	使用しない	—
デバイス初期値	1102 _H	CPU ユニットで使用するデバイス初期値のファイルを設定します。	<ul style="list-style-type: none"> 使用しない プログラムと同一ファイル名を使用 下記ファイルを使用する 	使用しない	6.26 節
ローカルデバイス用ファイル	1103 _H	プログラムで使用するローカルデバイスのファイルを設定します。	<ul style="list-style-type: none"> 使用しない 下記ファイルを使用する 	使用しない	9.13.2 項

(4) PC RAS 設定 (PC RAS 設定 (1) * 1)

RAS 機能のための各種設定を設定します。

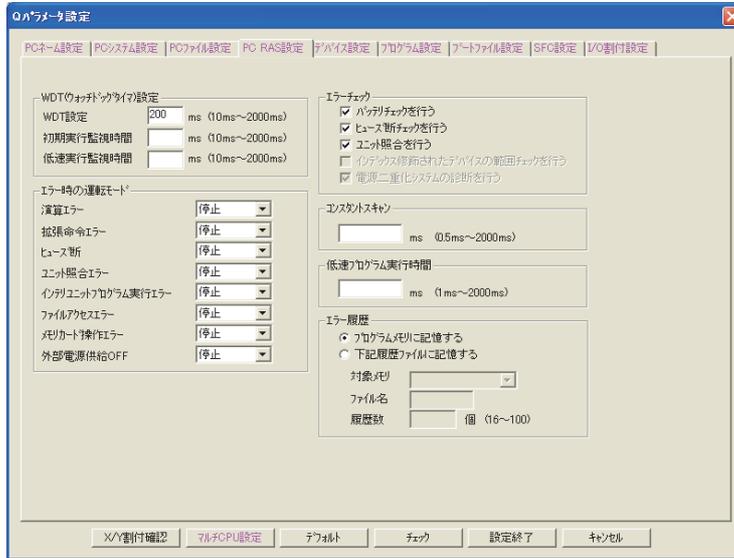


図 8.15 PC RAS 設定

表 8.15 PC RAS 設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
WDT(ウォッチドッグタイマ)設定	WDT 設定	CPU ユニットのウォッチドッグタイマの時間を設定します。	10ms ~ 2000ms (10ms 単位)	200ms	6.16 節
	初期実行監視時間	初期実行タイププログラム使用時のウォッチドッグタイマの時間を設定します。	10ms ~ 2000ms (10ms 単位)	設定なし	2.3.1 項
エラー時の運転モード	演算エラー	エラーを検出したときの CPU ユニットの動作モードを設定します。	停止/続行	停止	6.17 節
	拡張命令エラー * 2				
	ヒューズ断				
	ユニット照合エラー				
	インテリユニットプログラム実行エラー				
	ファイルアクセスエラー				
	メモ리카ード操作エラー				
	外部電源供給 OFF * 2				
エラーチェック	バッテリーチェックを行う	指定エラーを検出するか、検出しないかを設定します。	チェックを行う/行わない	チェックを行う	6.17 節
	ヒューズ断チェックを行う				
	ユニット照合を行う				
コンスタントスキャン	3003 _H	コンスタントスキャン時間を設定します。	0.5ms ~ 2000ms (0.5ms 単位)	設定なし	6.2 節
低速プログラム実行時間 * 3	3006 _H	毎スキャンの低速プログラムの実行時間を設定します。	1ms ~ 2000ms	設定なし	2.3.3 項
エラー履歴	3005 _H	CPU ユニットのエラー履歴の格納先を設定します。	プログラムメモリに記憶する/下記履歴ファイルに記憶する	プログラムメモリに記憶する	6.18 節

* 1 : プロセス CPU または二重化 CPU の PC パラメータ設定時、GX Developer の PC パラメータ画面には “PC RAS 設定 (1)” と表示されます。

* 2 : 拡張命令エラー、外部電源供給 OFF は、将来拡張用の設定です。

* 3 : 二重化 CPU 使用時は設定できません。

(5) PC RAS 設定 (2)  注 8.2

プロセス CPU, 二重化 CPU での RAS 機能の各種設定を設定します。



図 8.16 PC RAS 設定 (2)

表 8.16 PC RAS 設定 (2) 一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
メモリ チェック	3008 _H	プログラムメモリ（システムエリアを除く）のユーザエリアに対して、データが破損していないかをチェックする／しないを設定します。	プログラムメモリチェックを行わない／行う	プログラムメモリチェックを行わない	6.27 節
		“プログラムメモリチェックを行う”に設定した場合は、チェックを行うステップ数を設定します。	Q12PHCPU, Q12PRHCPU : 1 ~ 496 点 (256 ~ 124k ステップ) Q25PHCPU, Q25PRHCPU : 1 ~ 1008 点 (256 ~ 252k ステップ)	1 x 256 ステップ	

Point 

プログラムメモリのエリアのすべてをチェックする場合、チェック対象のステップ数は下記の方法で算出します。

- PC 読出画面で、プログラムメモリの全空容量を確認します。
全空容量の確認は、プログラムメモリをフォーマットし、ファイルを書き込んでいない状態で確認してください。( 5.2.2 項)
- 下記の計算式から、チェック対象のステップ数を算出します。

$$\text{チェック対象ステップ数} = \frac{\text{全空容量 (バイト)}}{4}$$

 注 8.2

ハイパフォーマンスモデル QCPU では、メモリチェック機能は使用できません。
プロセス CPU でメモリチェック機能を使用する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。( 付 2.3)

(6) デバイス設定

デバイスごとに使用点数、ラッチ範囲、ローカルデバイス範囲を設定します。

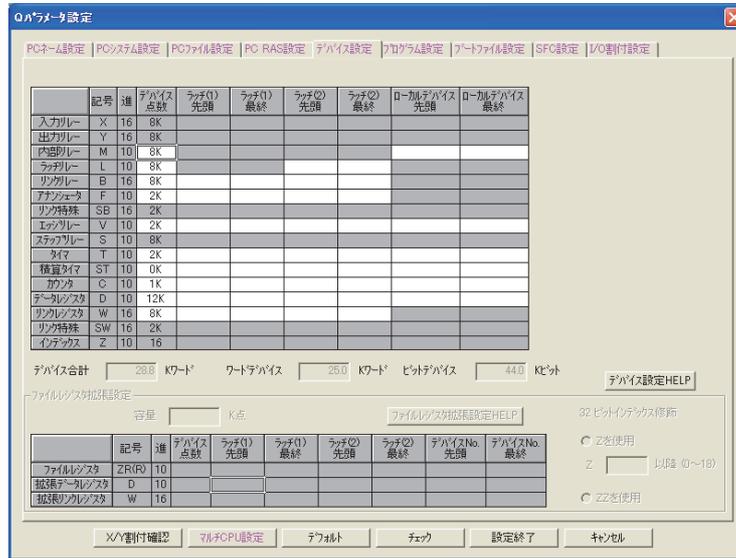


図 8.17 デバイス設定

表 8.17 デバイス設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
デバイス点数 *1	2000 _H	システムに合わせてデバイスの使用点数を設定します。	X(8k点), Y(8k点), S(8k点), SB(2k点), SW(2k点)は固定。 上記点数分(3.7kワード)を含み合計29kワードの範囲で設定可能。 1デバイス:最大32k点 ビットデバイスの合計:最大64k点	<ul style="list-style-type: none"> • X:8k点 • Y:8k点 • M:8k点 • L:8k点 • B:8k点 • F:2k点 • SB:2k点 • V:2k点 • S:8k点 • T:2k点 • ST:0k点 • C:1k点 • D:12k点 • W:8k点 • SW:2k点 	9.1 節
ラッチ(1)先頭/最終 (ラッチクリア有効)	2001 _H	RESET/L.CLRスイッチ、リモートラッチクリア操作でクリア可能にするラッチ範囲(先頭デバイス番号/最終デバイス番号)を設定します。	B, F, V, T, ST, C, D, Wの各デバイスを1範囲のみ設定。	設定なし	3.4 節, 6.3 節
ラッチ(2)先頭/最終 (ラッチクリア無効)	2002 _H	RESET/L.CLRスイッチ、リモートラッチクリア操作でクリア不可にするラッチ範囲(先頭デバイス番号/最終デバイス番号)を設定します。	L, B, F, V, T, ST, C, D, Wの各デバイスを1範囲のみ設定。	設定なし	3.4 節, 6.3 節
ローカルデバイス 先頭/最終	2003 _H	ローカルデバイスで使用するデバイス範囲(先頭デバイス番号/最終デバイス番号)を設定します。	M, V, T, ST, C, Dの各デバイスを1範囲のみ設定。	設定なし	9.13.2 項

* 1: デバイス点数を変更する場合は、ネットワークユニットのリフレッシュ範囲やインテリジェントユニットの自動リフレッシュ範囲がデバイス点数の範囲外とならないようにする必要があります。
該当デバイス範囲を超えた場合は、他のデバイスにデータが書き込まれたり、エラーになることがあります。

(7) プログラム設定

CPU ユニットに複数のプログラムを書き込む場合、プログラムのファイル名と実行タイプ（実行条件）を設定します。

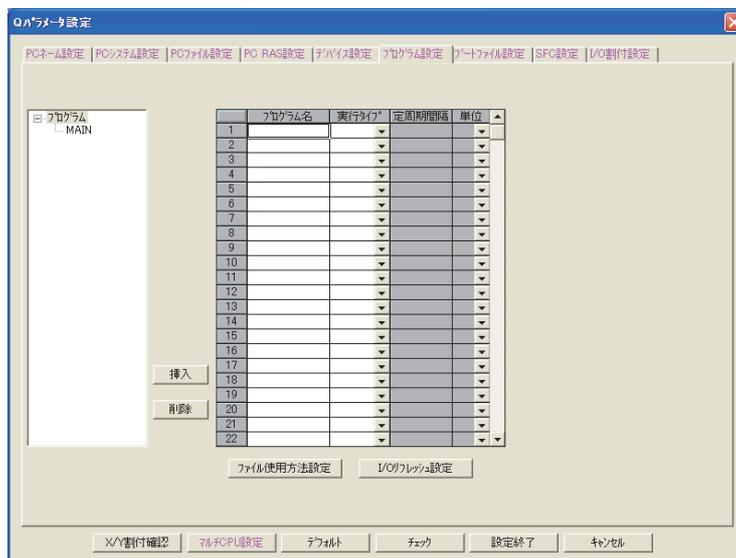


図 8.18 プログラム設定

表 8.18 プログラム設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
プログラム設定	7000 _H	CPU ユニットに複数のプログラムを書き込む場合、プログラムのファイル名と実行タイプ（実行条件）を設定します。 また定周期間隔（定周期実行タイププログラムの実行間隔）も設定します。	プログラム名 実行タイプ（定周期実行選択時の定周期間隔） ファイル使用方法設定 I/O リフレッシュ設定	設定なし	2.3 節

(8) ブートファイル設定

ブート運転、標準 ROM への自動書込みを行うための設定を設定します。

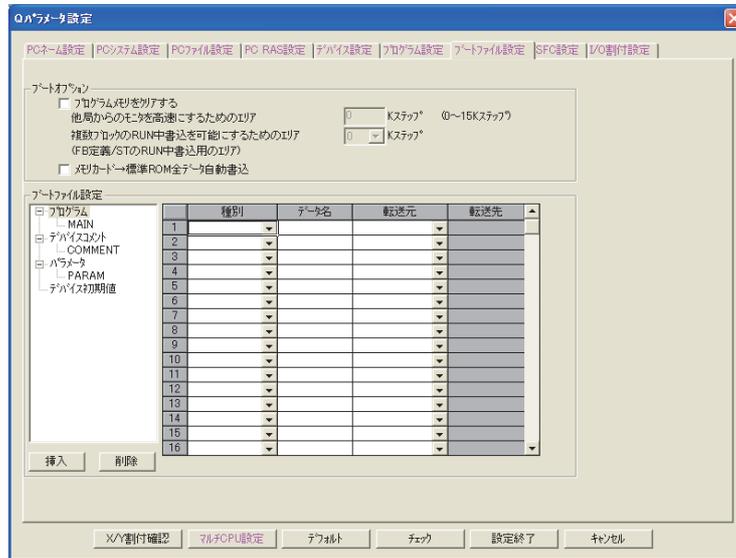


図 8.19 ブートファイル設定

表 8.19 ブートファイル設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
ブートオプション	7000H	ブート時にプログラムメモリをクリアするか、クリアしないかを設定します。	ブート時にプログラムメモリをクリアしない/ブート時にプログラムメモリをクリアする	ブート時にプログラムメモリをクリアしない	5.2.7 項、 5.2.8 項
		ブート時に、メモリカードのデータを標準 ROM に自動で書き込むか、書き込まないかを設定します。	メモリカード→標準 ROM 全データ自動書込をしない/する	メモリカード→標準 ROM 全データ自動書込をしない	
ブートファイル設定		ブート運転するファイルの種類、データ名、転送元ドライブを設定します。	種別、データ名、転送元ドライブ（転送先ドライブはプログラムメモリに自動設定されます。）	設定なし	

(9) SFC 設定

SFC プログラム使用時の SFC プログラム起動モード、起動条件、ブロック停止時の出力モードを設定します。



図 8.20 SFC 設定

表 8.20 SFC 設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
SFC プログラム起動モード	8002 _H	SFC プログラム使用時の SFC プログラム起動モード、起動条件、ブロック停止時の出力モードを設定します。	MELSEC-Q/L/QnA プログラミングマニュアル (SFC 編) を参照してください。	イニシャルスタート	—
起動条件	8003 _H			ブロック 0 を自動起動する	
ブロック停止時の出力モード	8006 _H			OFF する	

(10)I/O 割付設定

システム各ユニットの装着状態を設定します。



図 8.21 I/O 割付設定

表 8.21 I/O 割付設定一覧

項目		パラメータ No.	内容	設定範囲	デフォルト値	参照
I/O 割付	種別	0400 _H	装着しているユニットの種別を設定します。	CPU2号機～4号機：n号機／空き (CPUユニットを装着しないスロットは“CPU(空き)”を設定する。) 空き、入力、高速入力、出力、インテリ、入出力混合、割込み	設定なし	4.2.2 項
	形名		装着しているユニットの形名を設定します。(ユーザのメモ。CPUユニットは使用しない。)	半角 16 文字		
	点数		各スロットの点数を設定します。	0 点, 16 点, 32 点, 48 点, 64 点, 128 点, 256 点, 512 点, 1024 点		
	先頭 XY		各スロットの先頭入出力番号を設定します。	0 _H ~ FF _H		
基本設定	ベース形名	0401 _H	使用している基本ベースユニット、増設ベースユニットの形名を設定します。(ユーザのメモ。CPUユニットは使用しない。)	半角 16 文字	設定なし	4.1.2 項
	電源ユニット形名		基本ベースユニット、増設ベースユニットに装着している電源ユニットの形名を設定します。(ユーザのメモ。CPUユニットは使用しない。)	半角 16 文字		
	増設ケーブル形名		増設ケーブル形名を設定します。(ユーザのメモ。CPUユニットは使用しない。)	半角 16 文字		
	スロット数		基本ベースユニット、増設ベースユニットのスロット数を設定します。スロット数の設定は、すべてのベースユニットに行います。	2, 3, 5, 8, 10, 12		

(次ページへつづく)

表 8.9 I/O 割付設定一覧 (つづき)

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照	
スイッチ設定	0407 _H	インテリジェント機能ユニットの各種スイッチを設定します。	使用するインテリジェント機能ユニットのマニュアル参照	設定なし	6.10 節	
詳細設定	エラー時出力モード	0403 _H	管理 CPU が停止エラーになったとき出力をクリアするか/保持するかを設定します。	クリア/保持	クリア	6.8 節
	H/W エラー時 CPU 動作モード	4004 _H	インテリジェント機能ユニットのハードウェア異常時、管理 CPU の運転を停止させるか、続行させるかを設定します。	停止/続行	停止	6.9 節
	I/O 応答時間	0405 _H	入力ユニット、高速入力ユニット、入出力混合ユニット、割込みユニットの応答時間を設定します。	<ul style="list-style-type: none"> 入力, 入出力混合: 1ms, 5ms, 10ms, 20ms, 70ms 高速入力, 割込: 0.1ms, 0.2ms, 0.4ms, 0.6ms, 1ms 	<ul style="list-style-type: none"> 入力, 入出力混合: 10ms 高速入力, 割込: 0.2ms 	6.7 節
	管理 CPU * 1	0406 _H	入出力ユニット、インテリジェント機能ユニットの管理 CPU を設定します。	1号機, 2号機, 3号機, 4号機	1号機	QCPU ユーザーズマニュアル (マルチ CPU システム編)

* 1: 二重化 CPU 使用時は設定できません。

(11)X/Y 割付確認

I/O 割付け, Ethernet/CC IE/MELSECNET 設定, CC-Link 設定で設定した内容の確認を行います。



図 8.22 X/Y 割付確認

表 8.22 X/Y 割付確認一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
X/Y 割付確認	—	I/O 割付け, Ethernet/CC IE/MELSECNET 設定, CC-Link 設定で設定した内容を確認できます。	—	—	—

(12) マルチ CPU 設定 注 8.3

マルチ CPU システムを構築するための設定を行います。

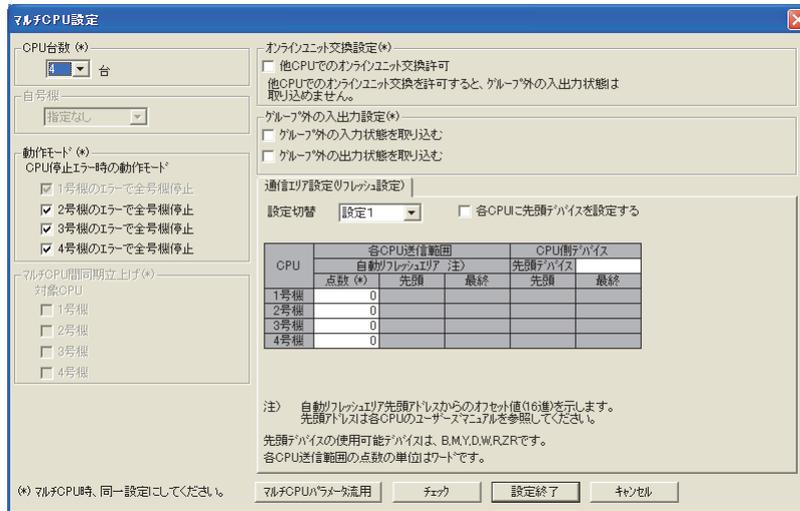


図 8.23 マルチ CPU 設定

表 8.23 マルチ CPU 設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
CPU 台数	0E00 _H	マルチ CPU システムで使用する CPU ユニットの台数を設定します。	1 台～ 4 台	1 台	QCPU ユーザーズマニュアル (マルチ CPU システム編)
自号機	E00C _H	マルチ CPU 設定パラメータを設定する号機 No. を設定します。(自号機 No. を設定します。)	1 号機～ 4 号機	設定なし	
動作モード	0E01 _H	2 号機～ 4 号機の CPU ユニットが停止エラーになったときのマルチ CPU システムの動作を設定します。 1 号機が停止エラーになったとき、マルチ CPU システムは停止します。(固定)	n 号機のエラーで全号機停止する / 停止しない	n 号機のエラーで全号機停止する	
マルチ CPU 間同期立上げ	E00B _H	マルチ CPU システムを構成する各 CPU ユニットの立ち上がり時間を同期させるかを設定します。	1 号機～ 4 号機	マルチ CPU 間の立ち上がりを全号機同期する	
オンラインユニット交換設定	E006 _H	マルチ CPU システムでオンラインユニット交換を許可する / 許可しないかを設定します。(許可するに設定した場合、グループ外の入出力状態は取り込みません。)	他 CPU ユニットでのオンラインユニット交換を許可する / 許可しない	<ul style="list-style-type: none"> ハイパフォーマンスモデル QCPU：許可しない プロセス CPU：許可する 	

(次ページへつづく)

注 8.3 **Redundant**

二重化 CPU では、マルチ CPU システムは構成できません。

表 8.23 マルチ CPU 設定一覧 (つづき)

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照	項目
グループ外の入出力設定	グループ外の入力状態を取り込む	0E04 _H	他号機が管理している入力ユニット、インテリジェント機能ユニットの入力状態を取り込むか/取り込まないかを設定します。	グループ外の入力状態を取り込まない/グループ外の入力状態を取り込む	グループ外の入力状態を取り込まない	
	グループ外の実出力状態を取り込む		他号機が管理している出力ユニットの実出力状態を取り込むか/取り込まないかを設定します。	グループ外の実出力状態を取り込まない/グループ外の実出力状態を取り込む	グループ外の実出力状態を取り込まない	
通信エリア設定 (リフレッシュ設定)	E002 _H E003 _H	マルチ CPU システムの各 CPU ユニット間で、自動リフレッシュによりデータの書き込み/読出しを行うデバイスと点数を設定します。		【各 CPU に先頭デバイスを設定する】 各 CPU に先頭デバイスを設定しない/各 CPU に先頭デバイスを設定する	各 CPU に先頭デバイスを設定しない	QCPU ユーザーズ マニュアル (マルチ CPU システム編)
				【各 CPU 送信範囲】 0 ~ 2048 点 (2 点単位) / 1 台 最大 8k 点 (8192 点) / 1 システム	設定なし	
				【CPU 側デバイス】 B, M, Y, D, W, R, ZR 指定デバイス番号から送信範囲で設定した点数分のデバイスを占有する。 送信範囲 1 点で B, M, Y は 16 点を占有。 送信範囲 1 点で D, W, R, ZR は 1 点を占有。		

8.2 二重化パラメータ 注 8.4

二重化パラメータの一覧と各パラメータの内容について説明します。

表 8.24 二重化パラメータ

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
二重化パラメータ	0D00H	二重化 CPU における動作モードおよびトラッキング転送設定を行います。	—	—	QnPRHCPU ユーザーズマニュアル (二重化システム編)

 QnPRHCPU ユーザーズマニュアル (二重化システム編)

(1) 動作モード設定

二重化システムの電源を ON した際の、二重化 CPU の動作を設定します。

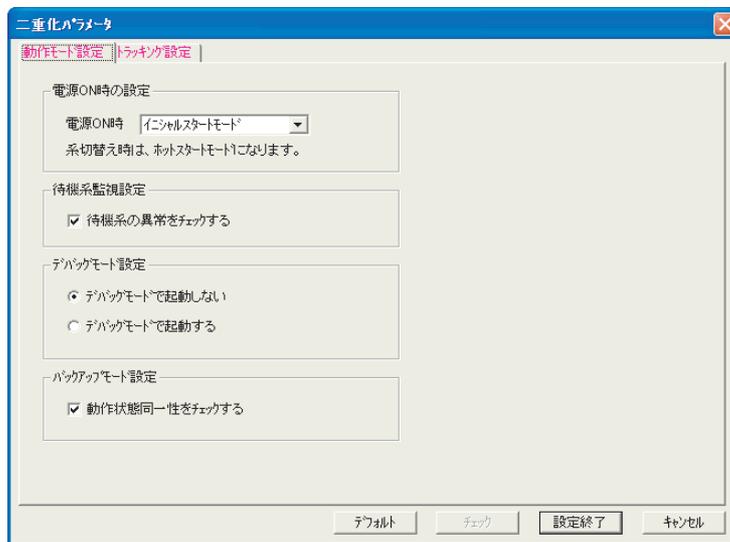


図 8.24 動作モード設定

表 8.25 動作モード設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
電源 ON 時の設定	D001H	電源 OFF → ON 時の二重化 CPU のスタートモードを設定します。	インイシャルスタートモード／ホットスタートモード	インイシャルスタートモード	QnPRHCPU ユーザーズマニュアル (二重化システム編)
待機系監視設定		待機系の異常を監視する／しないを設定します。	待機系の異常をチェックする／待機系の異常をチェックしない	待機系の異常をチェックする	
デバッグモード設定		二重化 CPU をデバッグモードで起動する／起動しないを設定します。	デバッグモードで起動しない／デバッグモードで起動する	デバッグモードで起動しない	
バックアップモード設定		二重化 CPU が、バックアップモードで運転時に、両系の動作状態の同一性をチェックする／しないを設定します。	動作状態同一性をチェックする／動作状態同一性をチェックしない	動作状態同一性をチェックする	



注 8.4

Basic

High performance

Process

ベーシックモデル QCPU, ハイパフォーマンスモデル QCPU, プロセス CPU では、二重化パラメータは設定できません。

(2) トラッキング設定

二重化 CPU のトラッキング転送機能に関する設定を行います。

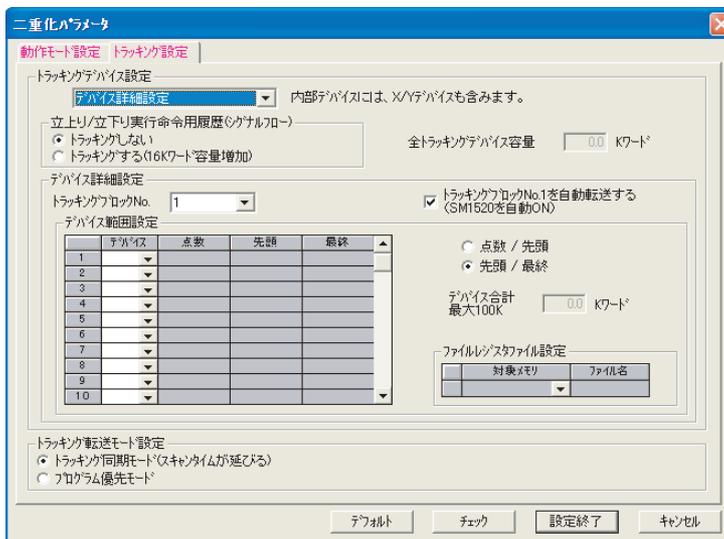


図 8.25 トラッキング設定

表 8.26 トラッキング設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
トラッキングデバイス設定		トラッキング転送するデバイスデータの範囲を、任意に設定する／しないを設定します。	内部デバイス一括設定／デバイス詳細設定	内部デバイス一括設定	
立上り／立下り実行命令履歴 (シグナルフロー)		シグナルフローメモリのトラッキングを実施する／実施しないを設定します。	トラッキングしない／トラッキングする	トラッキングしない	
デバイス詳細設定		トラッキング転送するデバイスをユーザが任意に設定する場合に、トラッキングするデバイスを設定します。	—	—	
トラッキングブロック No.	D003 _H	トラッキング転送するデバイスのブロック No. を選択します。ブロック No. を選択し、デバイス範囲設定で、トラッキングデバイスの設定を行います。	1 ~ 64	1	QnPRHCPU ユーザーズマニュアル (二重化システム編)
トラッキングブロック No. 1 を自動転送する (SM1520 を自動 ON)		トラッキングブロック No.1 を自動で転送する／しないを設定します。	自動転送する／自動転送しない	自動転送する	
デバイス範囲設定		トラッキング転送するデバイスと範囲を設定します。		—	
ファイルレジスタファイル設定		デバイス範囲設定でファイルレジスタを設定した場合、トラッキングを行うファイルレジスタの対象メモリ、ファイル名などを設定します。	QnPRHCPU ユーザーズマニュアルを参照してください。	—	
トラッキング転送モード設定	D002 _H	トラッキング転送を実施する際の、トラッキング転送の動作を設定します。	トラッキング同期モード／プログラム優先モード	トラッキング同期モード	

8.3 ネットワークパラメータ

ネットワークパラメータの一覧と各パラメータの内容について説明します。

■ 本節のパラメータ No. 欄に示す mn, **, M,N について

本節のパラメータ No. 欄に示す mn, **, M,N は、下記を示しています。

- mn：先頭 I/O No. ÷ 16 の値を示しています。
- **：任意の値
- N：何枚目のユニットであるか
- M：ネットワーク種別

表 8.27 CC-Link IE コントローラネットワーク, MELSECNET/H の場合 (☞ 本節 (1), (2))

M	ネットワーク種別
1H	CC IE Control (管理局) * 1, MELSECNET/H モード (管理局), MELSECNET/H 拡張モード (管理局), MELSECNET/10 モード (管理局)
2H	CC IE Control (通常局), MELSECNET/H モード (通常局), MELSECNET/H 拡張モード (通常局), MELSECNET/10 モード (通常局)
5H	MELSECNET/H (リモートマスタ局)
AH	MELSECNET/H (待機局)
BH	MELSECNET/H モード多重リモート I/O ネットマスタ局
DH	MELSECNET/H モード多重リモート I/O ネットサブマスタ局 (パラメータ設定がない場合)
EH	MELSECNET/H モード多重リモート I/O ネットサブマスタ局 (パラメータ設定がある場合)

* 1：ベーシックモデル QCPU では、CC IE Control (管理局) は設定できません。

表 8.28 CC-Link 設定の場合 (☞ 本節 (4))

M	ネットワーク種別
0H	マスタ局
1H	ローカル局
2H	待機マスタ局

(1) CC-Link IE コントローラネットワーク設定

CC-Link IE コントローラネットワークのネットワークパラメータを設定します。



図 8.26 Ethernet/CC IE/MELSECNET 枚数設定 (CC-Link IE コントローラネットワーク設定時)

表 8.29 CC-Link IE コントローラネットワーク設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
CC-Link IE コントローラネットワーク枚数設定	A000 _H	CC-Link IE コントローラネットワークのネットワークパラメータを設定します。	CC-Link IE コントローラネットワークのマニュアルを参照してください。	—	—
先頭 I/O No.	ANM0 _H				
ネットワーク No.					
総(子)局数					
局番					
グループ No.	0Amn _H				
モード	ANM0 _H				
リフレッシュパラメータ	ANM1 _H				
共通パラメータ	ANM2 _H				
局固有パラメータ	ANM3 _H				
グループ設定 * 1	D004 _H				
二重化設定 * 1	DA** _H				
リンク間転送	A002 _H				
ルーチングパラメータ	5003 _H				

* 1：二重化 CPU 使用時のみ設定可能です。

(2) MELSECNET/H 設定

MELSECNET/H のネットワークパラメータを設定します。



図 8.27 Ethernet/CC IE/MELSECNET 枚数設定 (MELSECNET/H 設定時)

表 8.30 MELSECNET/H 設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
MELSECNET/H 枚数設定	5000 _H	MELSECNET/H のネットワークパラメータを設定します。	Q 対応 MELSECNET/H のマニュアルを参照してください。	—	—
先頭 I/O No.	5NM0 _H				
ネットワーク No.					
総 (子) 局数					
グループ No.					
モード	5NM0 _H				
リフレッシュパラメータ	5NM1 _H				
共通パラメータ	5NM2 _H				
局固有パラメータ	5NM3 _H				
サブマスタ用パラメータ * 2	5NM5 _H				
共通パラメータ 2	5NMA _H				
局固有パラメータ 2	5NMB _H				
割込み設定					
グループ設定 * 1	D004 _H				
二重化設定 * 1	D5** _H				
他局アクセス時の有効ユニット	5001 _H				
リンク間転送	5002 _H				
ルーチングパラメータ	5003 _H				

* 1 : 二重化 CPU 使用時のみ設定可能です。

* 2 : プロセス CPU, 二重化 CPU 使用時のみ設定可能です。

(3) Ethernet 設定

Ethernet のネットワークパラメータを設定します。



図 8.28 Ethernet/CC IE/MELSECNET 枚数設定 (Ethernet 設定時)

表 8.31 Ethernet 設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
Ethernet 枚数設定	9000 _H	Ethernet のネットワークパラメータを設定します。	Q 対応 Ethernet のマニュアルを参照してください。	—	—
先頭 I/O No.	9N00 _H				
ネットワーク No.					
グループ No.					
局番					
動作設定					
イニシャル設定	9N01 _H				
オープン設定	9N02 _H				
ルータ中継パラメータ	9N03 _H				
局番 <-> IP 関連情報	9N05 _H				
FTP パラメータ	9N06 _H				
電子メール設定	9N07 _H				
報知設定	9N08 _H				
割込み設定	9N09 _H				
二重化設定 * 1	D9** _H				
他局アクセス時の有効ユニット	5001 _H				
ルーチングパラメータ	9N04 _H				
グループ設定 * 1	D004 _H				

* 1 : 二重化 CPU 使用時のみ設定可能です。

(4) CC-Link 設定

CC-Link のパラメータを設定します。



図 8.29 CC-Link 一覧設定

表 8.32 CC-Link 設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
ユニット枚数	C000 _H				
種別					
先頭 I/ONo.					
動作設定	CNM2 _H				
総接続台数					
リモート入力 (RX)	CNM1 _H	CC-Link のパラメータを設定します。	CC-Link のマニュアルを参照してください。	—	—
リモート出力 (RY)					
リモートレジスタ (RW _r)					
リモートレジスタ (RW _w)					
Ver.2 リモート入力 (RX) * 1					
Ver.2 リモート出力 (RY) * 1					
Ver.2 リモートレジスタ (RW _r) * 1					
Ver.2 リモートレジスタ (RW _w) * 1					
特殊リレー (SB)					
特殊レジスタ (SW)					
リトライ回数	CNM1 _H				
自動復列台数					
待機マスタ局番号					
CPU ダウン指定					
スキャンモード指定					
ディレイ時間設定					
局情報設定					
リモートデバイス局イニシャル設定					
割込み設定					

* 1 : ハイパフォーマンスモデル QCPU, プロセス CPU 使用時のみ設定可能です。

8.4 リモートパスワード

リモートパスワードに関するパラメータの一覧と各パラメータの内容について説明します。

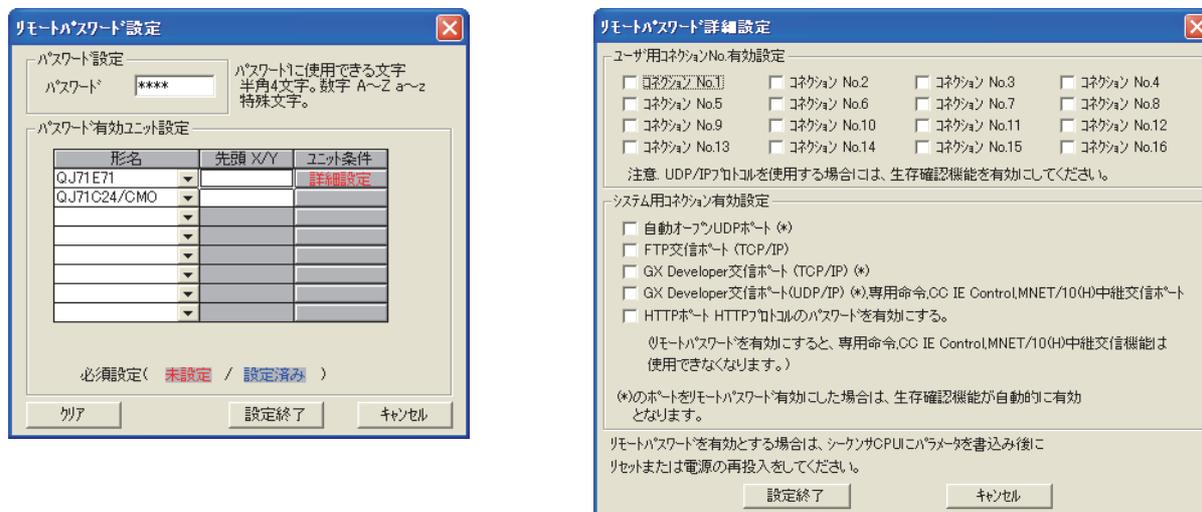


図 8.30 リモートパスワード設定画面

Ethernet ユニット、シリアルコミュニケーションユニット、モデムインタフェースユニットのリモートパスワードを設定します。

表 8.33 リモートパスワード設定一覧

項目	パラメータ No.	内容	設定範囲	デフォルト値	参照
パスワード設定	—	リモートパスワードを設定します。	4文字以内(英・数字、特殊記号)	—	
パスワード有効ユニット設定	形名	CPUユニットに設定されたリモートパスワードに対してチェックを行うユニット形名を選択します。	QJ71E71 QJ71C24/CMO	—	<ul style="list-style-type: none"> • QJ71E71 : Ethernet ユニットのマニュアル • QJ71C24 : シリアルコミュニケーションユニットのマニュアル • QJ71CMO : モデムインタフェースユニットのマニュアル
	先頭 XY	リモートパスワードのチェックを行うユニットの先頭アドレスを設定します。	<ul style="list-style-type: none"> • ベーシックモデル QCPU : 0000_H ~ 03E0_H • ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU : 0000_H ~ 0FE0_H 	—	
詳細設定	—	QJ71E71 におけるリモートパスワードの詳細を設定します。	—	—	
	ユーザ用コネクション No. 有効設定	ユーザ用コネクション No. を設定します。	コネクション No.1 ~ コネクション No.16	—	
	システム用コネクション有効設定	システム用コネクションのリモートパスワード有効ポートを指定します。	リモートパスワード有効ポートを指定する。 自動オープン UDP ポート FTP 通信ポート (TCP/IP) GX Developer 通信ポート (TCP/IP) GX Developer 通信ポート (UDP/IP), 専用命令, CC IE Control, MNET/10(H) 中継通信ポート HTTP ポート	—	

第9章 デバイスの説明

CPU ユニットで使用できるデバイスを説明します。

9.1 デバイス一覧

CPU ユニットで使用できるデバイスと使用範囲を表 9.1、表 9.2 に示します。

(1) ベーシックモデル QCPU の場合

表 9.1 デバイス一覧表

分類	種別	デバイス名	デフォルト値			パラメータ設定による設定範囲	参照
			点数	使用範囲			
内部ユーザデバイス	ビットデバイス	入力	2048点	X0 ~ 7FF	16進	16.4kワード以内で変更可*3	9.2.1項
		出力	2048点	Y0 ~ 7FF	16進		9.2.2項
		内部リレー	8192点	M0 ~ 8191	10進		9.2.3項
		ラッチリレー	2048点	L0 ~ 2047	10進		9.2.4項
		アナンシェータ	1024点	F0 ~ 1023	10進		9.2.5項
		エッジリレー	1024点	V0 ~ 1023	10進		9.2.6項
		ステップリレー	2048点	S0 ~ 127 / ブロック	10進		9.2.9項
		リンクリレー	2048点	B0 ~ 7FF	16進		9.2.7項
		リンク特殊リレー	1024点	SB0 ~ 3FF	16進		9.2.8項
	ワードデバイス	タイマ*1	512点	T0 ~ 511	10進	16.4kワード以内で変更可*3	9.2.10項
		積算タイマ*1	0点	(ST0 ~ 511)	10進		
		カウンタ*1	512点	C0 ~ 511	10進		9.2.11項
		データレジスタ	11136点	D0 ~ 11135	10進		9.2.12項
		リンクレジスタ	2048点	W0 ~ 7FF	16進		9.2.13項
リンク特殊レジスタ		1024点	SW0 ~ 3FF	16進	9.2.14項		
内部システムデバイス	ビットデバイス	ファンクション入力	16点	FX0 ~ F	16進	不可	9.3.1項
		ファンクション出力	16点	FY0 ~ F	16進		9.3.1項
		特殊リレー	1024点	SM0 ~ 1023	10進		9.3.2項
	ワードデバイス	ファンクションレジスタ	5点	FD0 ~ 4	10進		9.3.1項
		特殊レジスタ	1024点	SD0 ~ 1023	10進		9.3.3項
リンクダイレクトデバイス	ビットデバイス	リンク入力	8192点	Jn¥X0 ~ 1FFF	16進	不可	9.4節
		リンク出力	8192点	Jn¥Y0 ~ 1FFF	16進		
		リンクリレー	16384点	Jn¥B0 ~ 3FFF	16進		
		リンク特殊リレー	512点	Jn¥SB0 ~ 1FF	16進		
	ワードデバイス	リンクレジスタ	16384点	Jn¥W0 ~ 3FFF	16進		
		リンク特殊レジスタ	512点	Jn¥SW0 ~ 1FF	16進		
ユニットアクセスデバイス	ワードデバイス	インテリジェント機能ユニットデバイス	65536点	Un¥G0 ~ 65535*2	10進	不可	9.5節
インデックスレジスタ	ワードデバイス	インデックスレジスタ	10点	Z0 ~ 9	10進	不可	9.6節

(次ページにつづく)

表 9.1 デバイス一覧表 (つづき)

分類	種別	デバイス名	デフォルト値			パラメータ設定 による設定範囲	参照	
			点数	使用範囲				
ファイルレジスタ*5	ワードデバイス	ファイルレジスタ	64k点	• R0 ~ 32767 • ZR0 ~ 65535	10進	不可	9.7節	
ネスティング	—	ネスティング	15点	N0 ~ 14	10進	不可	9.8節	
ポインタ	—	ポインタ	300点	P0 ~ 299	10進	不可	9.9節	
		割込みポインタ	128点	I0 ~ 127	10進		9.10節	
その他	ビットデバイス	SFC ブロックデバイス	128点	BL0 ~ 127	10進	不可	9.11.1項	
	—	ネットワーク No. 指定デバイス	239点	J1 ~ 239	10進	不可	9.11.3項	
		I/O No. 指定デバイス	Q00JCPU	—	U0 ~ F		16進	9.11.4項
			Q00CPU, Q01CPU	—	U0 ~ 3F		16進	9.11.4項
	—	マクロ命令引数デバイス	—	VD0 ~ □	10進	不可	9.11.5項	
定数	—	10進定数	K-2147483648 ~ 2147483647				9.12.1項	
		16進定数	H0 ~ FFFFFFFF				9.12.2項	
		実数定数	E±1.17550 - 38 ~ E±3.40282 + 38				9.12.3項	
		文字列定数	"ABC" , "123" *4				9.12.4項	

* 1 : タイマ, 積算タイマ, カウンタは, 接点・コイルがビットデバイス, 現在値がワードデバイスになります。

* 2 : 実際に使用できる点数は, インテリジェント機能ユニットにより異なります。

バッファメモリの点数については, 使用するインテリジェント機能ユニットのマニュアルを参照してください。

* 3 : GX Developer の PC パラメータで変更可能です。

(入力, 出力, ステップリレー, リンク特殊リレー, リンク特殊レジスタを除く。)( 9.2 節)

* 4 : 文字列は \$MOV, STR, DST, VAL, DVAL, ESTR, EVAL 命令でのみ使用できます。

他の命令で文字列を使用することはできません。

* 5 : Q00JCPU には, 標準 RAM がいないためファイルレジスタは使用できません。

(2) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU の場合

表 9.2 デバイス一覧表

分類	種別	デバイス名	デフォルト値			パラメータ設定による設定範囲	参照
			点数	使用範囲			
内部ユーザデバイス	ビットデバイス	入力	8192点	X0 ~ 1FFF	16進	29kワード 以内で変更可 *3	9.2.1項
		出力	8192点	Y0 ~ 1FFF	16進		9.2.2項
		内部リレー	8192点	M0 ~ 8191	10進		9.2.3項
		ラッチリレー	8192点	L0 ~ 8191	10進		9.2.4項
		アナンシェータ	2048点	F0 ~ 2047	10進		9.2.5項
		エッジリレー	2048点	V0 ~ 2047	10進		9.2.6項
		ステップリレー	8192点	S0 ~ 511 / ブロック	10進		9.2.9項
		リンクリレー	8192点	B0 ~ 1FFF	16進		9.2.7項
	ワードデバイス	リンク特殊リレー	2048点	SB0 ~ 7FF	16進		9.2.8項
		タイマ*1	2048点	T0 ~ 2047	10進		9.2.10項
		積算タイマ*1	0点	(ST0 ~ 2047)	10進		
		カウンタ*1	1024点	C0 ~ 1023	10進		9.2.11項
		データレジスタ	12288点	D0 ~ 12287	10進		9.2.12項
		リンクレジスタ	8192点	W0 ~ 1FFF	16進		9.2.13項
リンク特殊レジスタ	2048点	SW0 ~ 7FF	16進	9.2.14項			
内部システムデバイス	ビットデバイス	ファンクション入力	16点	FX0 ~ F	16進	不可	9.3.1項
		ファンクション出力	16点	FY0 ~ F	16進		9.3.1項
		特殊リレー	2048点	SM0 ~ 2047	10進		9.3.2項
	ワードデバイス	ファンクションレジスタ	5点	FD0 ~ 4	10進		9.3.1項
		特殊レジスタ	2048点	SD0 ~ 2047	10進		9.3.3項
リンクダイレクトデバイス	ビットデバイス	リンク入力	8192点	Jn¥X0 ~ 1FFF	16進	不可	9.4節
		リンク出力	8192点	Jn¥Y0 ~ 1FFF	16進		
		リンクリレー	16384点	Jn¥B0 ~ 3FFF	16進		
		リンク特殊リレー	512点	Jn¥SB0 ~ 1FF	16進		
	ワードデバイス	リンクレジスタ	16384点	Jn¥W0 ~ 3FFF	16進		
		リンク特殊レジスタ	512点	Jn¥SW0 ~ 1FF	16進		
ユニットアクセスデバイス	ワードデバイス	インテリジェント機能ユニットデバイス	65536点	Un¥G0 ~ 65535 *2	10進	不可	9.5節
		マルチCPU間共有デバイス*4	4096点	U3En¥G0 ~ 4095	10進	可	
インデックスレジスタ	ワードデバイス	インデックスレジスタ	16点	Z0 ~ 15	10進	不可	9.6節
ファイルレジスタ	ワードデバイス	ファイルレジスタ	0点	—	—	0 ~ 1018k点	9.7節
ネスティング	—	ネスティング	15点	N0 ~ 14	10進	不可	9.8節
ポインタ	—	ポインタ	4096点	P0 ~ 4095	10進	不可	9.9節
		割込みポインタ	256点	I0 ~ 255	10進		9.10節

(次ページにつづく)

表 9.2 デバイス一覧表 (つづき)

分類	種別	デバイス名	デフォルト値			パラメータ設定による設定範囲	参照
			点数	使用範囲			
その他	ビットデバイス	SFC ブロックデバイス	320 点	BL0 ~ 319	10 進	不可	9.11.1 項
		SFC 移行デバイス	512 点	TR0 ~ 511	10 進		9.11.2 項
	—	ネットワーク No. 指定デバイス	255 点	J1 ~ 255	10 進		9.11.3 項
		I/O No. 指定デバイス	—	U0 ~ FF	16 進		9.11.4 項
		マクロ命令引数デバイス	—	VD0 ~ □	16 進		9.11.5 項
定数	—	10 進定数	K-2147483648 ~ 2147483647			9.12.1 項	
		16 進定数	H0 ~ FFFFFFFF			9.12.2 項	
		実数定数	単精度浮動小数点データ E±1.17549435 - 38 ~ E±3.40282347 + 38			9.12.3 項	
			倍精度浮動小数点データ E±2.2250738585072014 - 308 ~ E±1.7976931348623157 + 308			9.12.3 項	
		文字列定数	"ABC" , "123"			9.12.4 項	

- * 1: タイマ, 積算タイマ, カウンタは, 接点・コイルがビットデバイス, 現在値がワードデバイスになります。
- * 2: 実際に使用できる点数は, インテリジェント機能ユニット/特殊機能ユニットにより異なります。
バッファメモリの点数については, 使用するインテリジェント機能ユニット/特殊機能ユニットのマニュアルを参照してください。
- * 3: PC パラメータのデバイス設定で変更可能です。(入力, 出力, ステップリレー, リンク特殊リレー, リンク特殊レジスタを除く。)( 9.2 節)
- * 4: マルチ CPU システム構成時のみ使用できます。

9.2 内部ユーザデバイス

(1) 内部ユーザデバイスとは

ユーザの用途に合わせて使用できるデバイスです。

(2) 内部ユーザデバイスのデバイス点数

PC パラメータのデバイス設定で変更できます。ただし、入力(X)、出力(Y)、ステップリレー(S)、リンク特殊リレー(SB)、リンク特殊レジスタ(SW)の点数は変更できません。

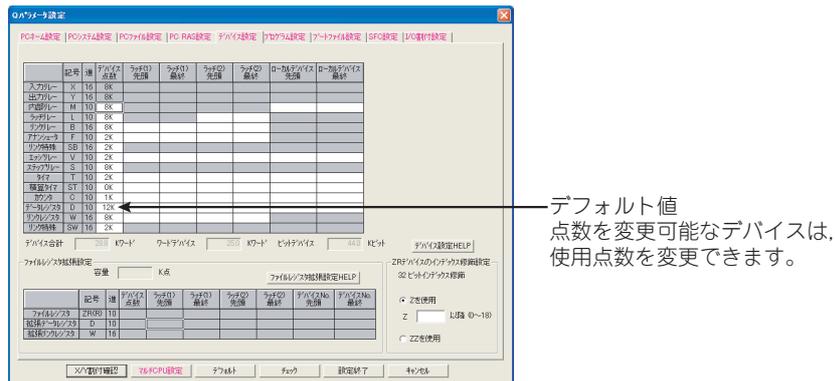


図 9.1 PC パラメータのデバイス設定

デバイスの点数を変更する場合は、以下に注意してください。

- 1 デバイスは、16 点単位で設定します。
- ビットデバイスの 1 デバイスの最大点数は、32k 点です。
- 内部ユーザデバイスの合計の最大点数は、各 CPU ユニットで異なります。
 - ベーシックモデル QCPU：16.4k ワード
 - ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU：29k ワード
- タイマ、積算タイマ、カウンタは、デバイス 1 点につきワードデバイス 1 点分、ビットデバイス 2 点分で計算します。

Point

- デバイス点数を変更する場合は、下記のリフレッシュ範囲がデバイス点数の範囲外とならないように注意してください。
 - ネットワークユニットのリフレッシュ範囲
 - インテリジェントユニットの自動リフレッシュ範囲

該当デバイス範囲を超えた場合は、他のデバイスにデータが書き込まれたり、エラーになることがあります。

- 内部リレー、ラッチリレー、アナンシェータ、エッジリレー、リンクリレー、リンク特殊リレー、ステップリレー、タイマ、積算タイマ、カウンタの合計点数は、最大 64k 点数になります。
- 内部ユーザデバイスの使用点数を変更してパラメータを PC 書込みした場合、デバイスのアドレスと格納値がずれることがあり、そのずれた値を使用する可能性があるため、変更前のパラメータで作成した下記のファイルはそのままでは使用できません。
 - シーケンスプログラム
 - SFC プログラム
 - ST プログラム

内部ユーザデバイスの使用点数を変更する場合、GX Developer で下記を行ってください。

[内部ユーザデバイス使用点数変更前]

CPU ユニットから使用するデバイス、各プログラムをすべて読み出す。

[内部ユーザデバイス使用点数変更後]

変更前に読み出したデバイス、各プログラムを CPU ユニットに書き込む。

デバイス、各プログラムの読出し/書込みについては、下記マニュアルを参照してください。

GX Developer Version 8 オペレーティングマニュアル

(3) メモリ容量の考え方

内部ユーザデバイスは下式を満足するように設定してください。

(ビットデバイス容量) + (タイマ, 積算タイマ, カウンタ容量) + (ワードデバイス容量) \leq 29k ワード

(a) ビットデバイスの場合

ビットデバイスは, 16点を1ワードとして計算します。

$$\text{(ビットデバイス容量)} = \frac{(X+Y+M+L+B+F+SB+V+S)}{16} \text{ (ワード)}$$

(b) タイマ (T), 積算タイマ (ST), カウンタ (C) の場合

タイマ (T), 積算タイマ (ST), カウンタ (C) は, 16点を18ワードとして計算します。

$$\text{(タイマ, 積算タイマ, カウンタ容量)} = \frac{(T+ST+C)}{16} \times 18 \text{ (ワード)}$$

(c) ワードデバイスの場合

データレジスタ (D), リンクレジスタ (W), リンク特殊レジスタ (SW) は, 16点を16ワードとして計算します。

$$\text{(ワードデバイス容量)} = \frac{(D+W+SW)}{16} \times 16 \text{ (ワード)}$$

(4) デバイス点数割付け例

デバイス点数割付け例を表 9.3 に示します。

表 9.3 は、付 3 に記載のデバイス点数割付けシートを使用しています。

表 9.3 デバイス点数割付け例

デバイス名	記号	進数	デバイス点数* ²		制限チェック			
			点数	番号	容量 (ワード)* ³	ビット点数* ²		
入力リレー* ¹	X	16	8k(8192) 点	X0000 ~ 1FFF	÷16	512 ワード	×1	8192 点
出力リレー* ¹	Y	16	8k(8192) 点	Y0000 ~ 1FFF	÷16	512 ワード	×1	8192 点
内部リレー	M	10	16k(16384) 点	M0 ~ 16383	÷16	1024 ワード	×1	16384 点
ラッチリレー	L	10	4k(4096) 点	L0 ~ 4095	÷16	256 ワード	×1	4096 点
リンクリレー	B	16	4k(4096) 点	B0000 ~ 0FFF	÷16	256 ワード	×1	4096 点
アナンシェータ	F	10	1k(1024) 点	F0 ~ 1023	÷16	64 ワード	×1	1024 点
リンク特殊リレー* ¹	SB	16	2k(2048) 点	SB0000 ~ 07FF	÷16	128 ワード	×1	2048 点
エッジリレー	V	10	1k(1024) 点	V0 ~ 1023	÷16	64 ワード	×1	1024 点
ステップリレー* ¹	S	10	8k(8192) 点	S0 ~ 8191	÷16	512 ワード	×1	8192 点
タイマ	T	10	2k(2048) 点	T0 ~ 2047	× $\frac{18}{16}$	2304 ワード	×2	4096 点
積算タイマ	ST	10	2k(2048) 点	ST0 ~ 2047	× $\frac{18}{16}$	2304 ワード	×2	4096 点
カウンタ	C	10	1k(1024) 点	C0 ~ 1023	× $\frac{18}{16}$	1152 ワード	×2	2048 点
データレジスタ	D	10	14k(14336) 点	D0 ~ 14335	×1	14336 ワード	—	
リンクレジスタ	W	16	4k(4096) 点	W0000 ~ 4095	×1	4096 ワード	—	
リンク特殊レジスタ* ¹	SW	16	2k(2048) 点	SW0000 ~ 07FF	×1	2048 ワード	—	
デバイス合計						29568 ワード (29696 ワード以下)		63488 点 (65536 点以下)

* 1 : システムで固定です。(変更不可)

* 2 : 1 デバイスの最大点数は 32k 点です。

* 3 : デバイス点数を、容量 (ワード) 欄に示す数字で乗算 (または除算) した数値を記入します。

9.2.1 入力 (X)

(1) 入力とは

入力は押ボタンスイッチ，切換えスイッチ，リミットスイッチ，デジタルスイッチなどの外部機器から，CPU ユニットに指令やデータを与えるためのものです。

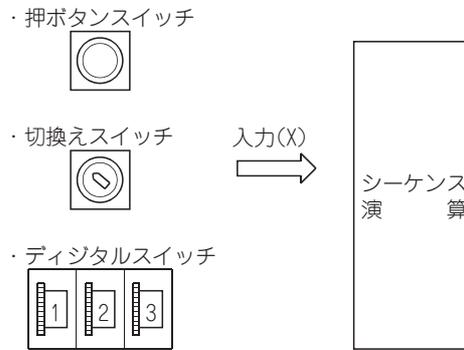


図 9.2 外部機器から CPU ユニットへの指令

(2) 入力 (X) の考え方

入力 1 点に対して仮想のリレー Xn を CPU ユニット内に内蔵していると考えます。プログラムではその Xn の a 接点，b 接点を使用します。

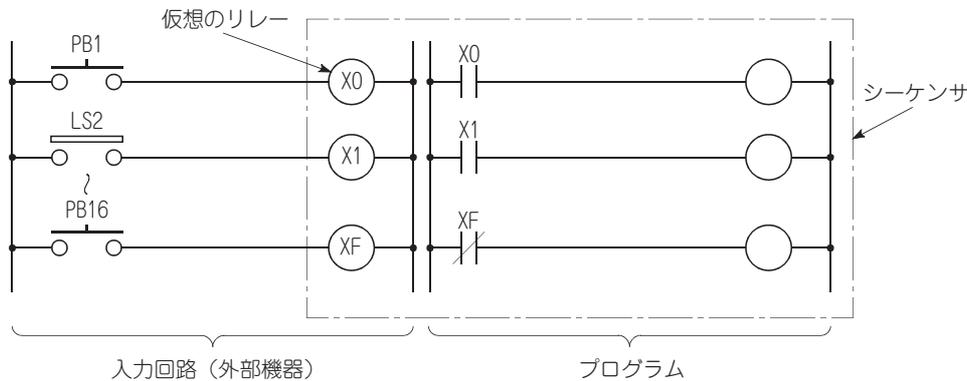


図 9.3 入力 (X) の考え方

(3) a 接点と b 接点の使用数

プログラム内での Xn の a 接点と b 接点の使用数は，プログラム容量の範囲内であれば制限はありません。

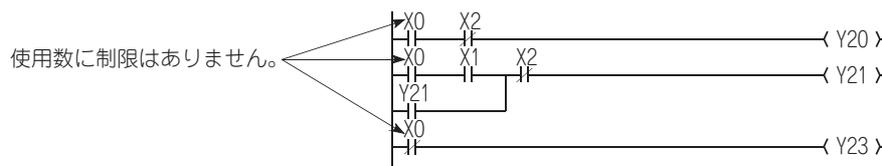


図 9.4 入力 (X) のプログラムでの使用

Point

- 作成したプログラムのデバッグ時、入力 (X) を次の方法で ON/OFF できます。
 - GX Developer のデバイステスト
 - OUT Xn 命令

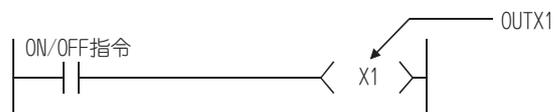


図 9.5 OUT Xn 命令による入力 (X) の ON/OFF

- 入力 (X) は次の場合にも使用できます。
 - CC-Link の RX のリフレッシュ先 (CPU ユニット側) デバイス
 - CC-Link IE コントローラネットワークまたは MELSECNET/H のリフレッシュ先 (CPU ユニット側) デバイス
-

9.2.2 出力 (Y)

(1) 出力とは

プログラムの制御結果を外部の信号灯, デジタル表示器, 電磁開閉器 (コンタクタ), ソレノイドなどへ出力するものです。

外部へ a 接点相当で取り出すことができます。

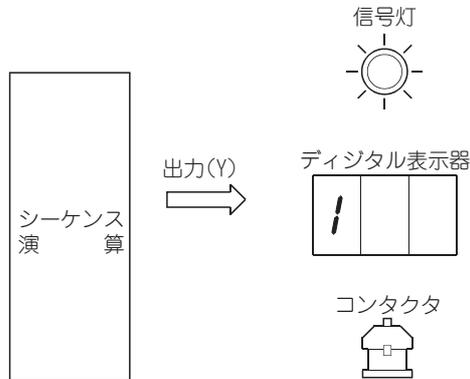


図 9.6 CPU ユニットから外部機器への出力

(2) a 接点と b 接点の使用数

プログラム内での出力 Yn の a 接点と b 接点の使用数は, プログラム容量の範囲内であれば制限はありません。

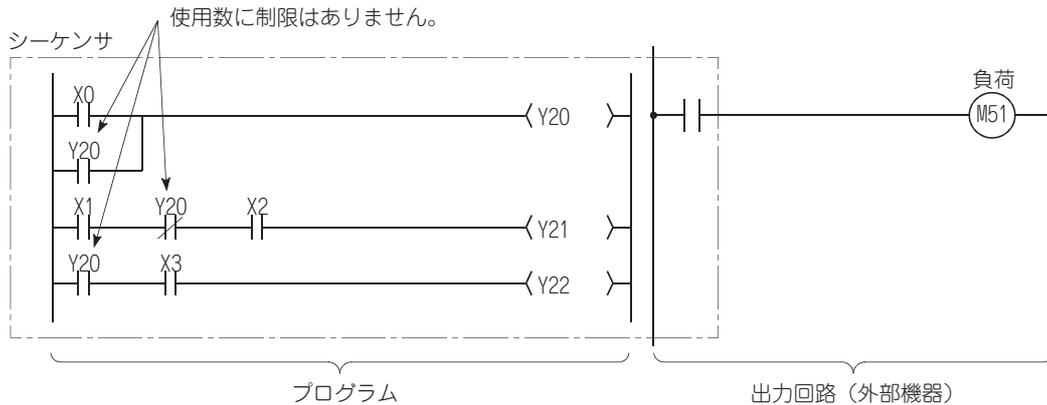


図 9.7 出力 (Y) のプログラムでの使用

(3) 内部リレー (M) の代用としての使用

入力ユニットを装着した領域, およびユニット未装着域に対応する出力 (Y) を, 内部リレー (M) の代わりにして使用できます。

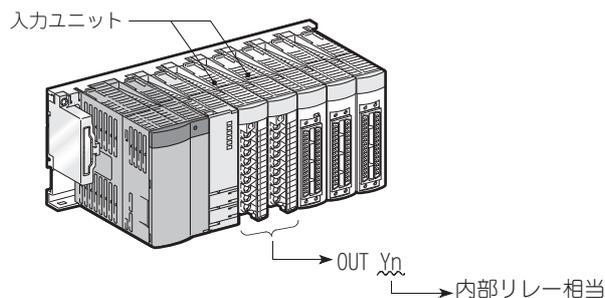


図 9.8 内部リレーの代用

9.2.3 内部リレー (M)

(1) 内部リレーとは

CPU ユニット内部で使用する補助リレーです。

下記の操作を行うと内部リレーは、すべて OFF します。

- CPU ユニットの電源 OFF → ON 時
- リセット操作時
- ラッチクリア時 (☞ 6.3 節)

(2) ラッチ (停電保持) の可否

内部リレーは、ラッチ (停電保持) できません。

(3) a 接点と b 接点の使用数

プログラム内での接点 (a 接点, b 接点) の使用数は、プログラム容量の範囲内であれば制限はありません。

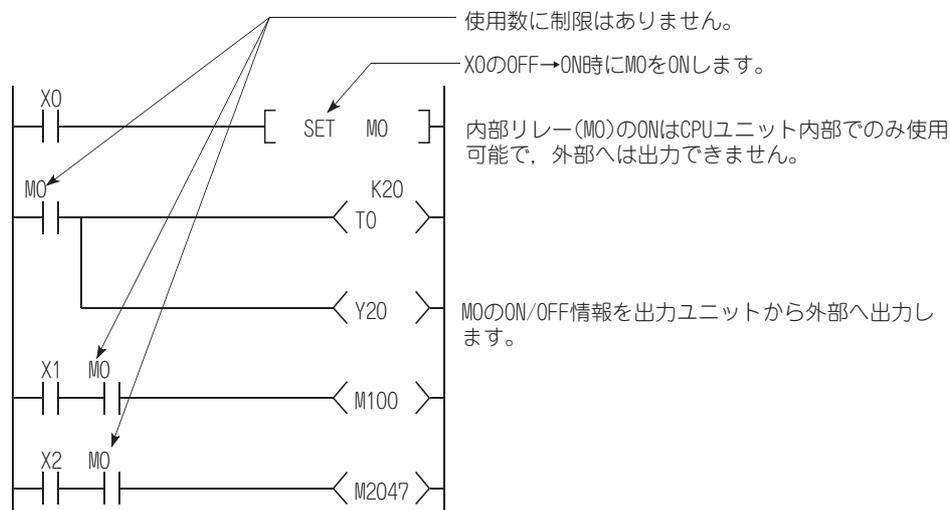


図 9.9 内部リレーのプログラムでの使用

(4) 外部への出力方法

シーケンスプログラムの演算結果を外部に出力する場合は、出力 (Y) を使用します。

Point

ラッチ (停電保持) が必要な場合は、ラッチリレー (L) を使用してください。(☞ 9.2.4 項)

9.2.4 ラッチリレー (L)

(1) ラッチリレーとは

CPU ユニット内部で使用するラッチ（停電保持）のできる補助リレーです。
 停電中には、ラッチリレーの内容を CPU ユニット本体のバッテリーにより、保持しています。
 下記の操作を行っても直前の演算結果（ON/OFF 情報）を保持します。

- CPU ユニットの電源 OFF → ON 時
- CPU ユニットのリセット操作時

(2) ラッチリレーのクリア

ラッチリレーは、ラッチクリア操作で OFF します。（ 3.7 節）

ただし、PC パラメータのデバイス設定で“ラッチ (2) 先頭/最終”に設定したラッチリレーは、RESET/ L.CLR スイッチ  **注 9.1** / リモートラッチクリアによるラッチクリアを行っても OFF できません。

(3) a 接点と b 接点の使用数

プログラム内での接点（a 接点、b 接点）の使用数は、プログラム容量の範囲内であれば制限はありません。

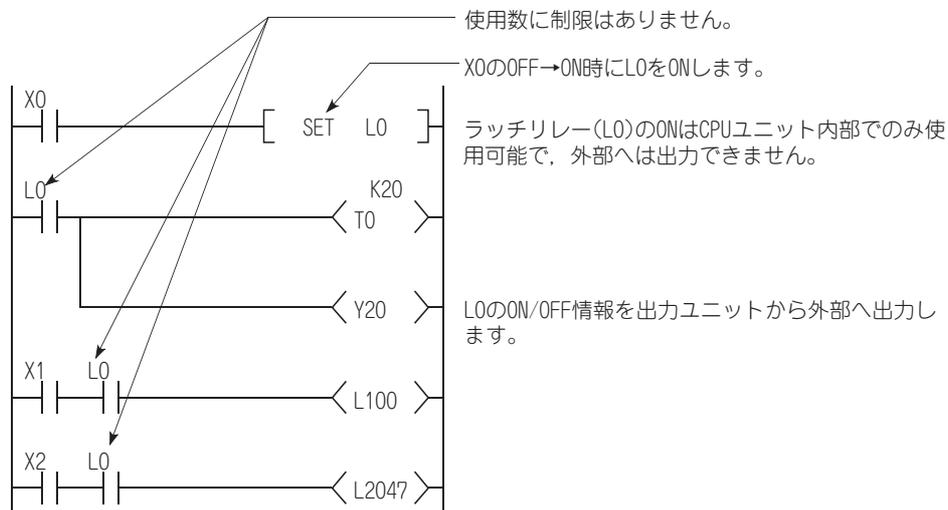


図 9.10 ラッチリレー



注 9.1

Basic

ベーシックモデル QCPU では、スイッチ操作によるラッチクリアはできません。

(4) 外部への出力方法

シーケンスプログラムの演算結果を外部に出力する場合は、出力 (Y) を使用します。

Point

- ラッチが必要ない場合は、内部リレー (M) を使用してください。
( 9.2.3 項)
 - ラッチクリアの無効範囲は、PC パラメータのデバイス設定で設定します。( 6.3 節)
-

9.2.5 アナンシェータ (F)

(1) アナンシェータとは

ユーザで作成した設備において異常・故障検出用のプログラムに使用すると便利な内部リレーです。

(2) アナンシェータ ON 時の特殊リレーと特殊レジスタ

アナンシェータを ON すると、特殊リレー (SM62) が ON し、特殊レジスタ (SD62 ~ 79) に ON したアナンシェータの個数と番号が格納されます。

- ・特殊リレー : SM62 . . . アナンシェータが1つでも ON すると ON します。
- ・特殊レジスタ : SD62 . . . 最初に ON したアナンシェータ番号を格納します。
- SD63 . . . ON しているアナンシェータの個数を格納します。
- SD64 ~ 79 . . . ON した順にアナンシェータ番号を格納します。
(SD62 と SD64 は同一アナンシェータ番号が格納されます。)

また SD62 に格納されているアナンシェータ番号は、エラー履歴格納エリアにも登録されます。

Point

ベーシックモデル QCPU では、シーケンサの電源 ON 中にエラー履歴格納エリアへ格納されるアナンシェータ番号は 1 つのみです。

(3) アナンシェータの用途

故障検出プログラムにアナンシェータを使用すると、特殊リレー (SM62) が ON したときに、特殊レジスタ (SD62 ~ 79) をモニタすることによって、設備の異常・故障の有無 (アナンシェータ番号) を確認できます。

例 アナンシェータ (F5) が ON したとき、外部に ON したアナンシェータ番号を出力するプログラム

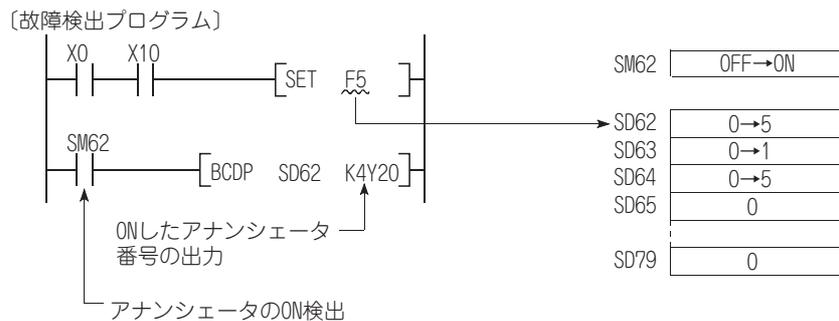


図 9.11 アナンシェータの ON 検出と格納

(4) a 接点と b 接点の使用数

プログラム内での接点 (a 接点, b 接点) の使用数は、プログラム容量の範囲内であれば制限はありません。

(5) アナンシェータの ON 方法と処理内容

(a) アナンシェータの ON 方法

下記に示す命令で行うことができます。

1) SET F_n 命令

SET F_n 命令は、入力条件の立上り時 (OFF → ON) のみアナンシェータを ON します。

入力条件が OFF してもアナンシェータは ON のまま保持されます。

アナンシェータを多く使用する場合は、OUT F_n 命令を使用するより、スキャンタイムを短くできます。

2) OUT F_n 命令

OUT F_n 命令を使用しアナンシェータの ON/OFF ができますが、毎スキャン処理を行うため、SET F_n 命令を使用するよりスキャンタイムが延びます。

また OUT F_n 命令でアナンシェータを OFF しても、RST F_n 命令 / LEDR 命令  注 9.2 / BKRST 命令の実行が必要のため、アナンシェータの ON は SET F_n 命令を使用してください。

Point

- アナンシェータを SET F_n 命令または OUT F_n 命令以外 (例えば MOV 命令) で ON した場合は、内部リレー (M) と同じ動作になります。
SM62 の ON および SD62, SD64 ~ 79 へのアナンシェータ番号の格納は行いません。
- 二重化 CPU でトラッキング転送している場合、アナンシェータを制御系から待機系へトラッキングしている場合、待機系においてアナンシェータは ON しません。
アナンシェータをトラッキングしている場合の処理については、下記マニュアルを参照してください。
 QnPRHCPU ユーザーズマニュアル (二重化システム編)



注 9.2

Basic

ベーシックモデル QCPU では、LEDR 命令が使用できません。

(b) アナシエータ ON 時の処理内容**1) 特殊レジスタ (SD62 ~ 79) の格納データ**

- ONしたアナシエータ番号がSD64～79に順次格納されます。
- SD64に格納されているアナシエータ番号が、SD62に格納されます。
- SD63の内容が+1されます。

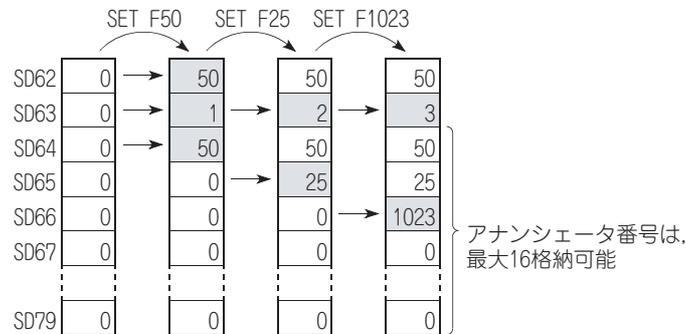


図 9.12 アナシエータ ON 時の処理内容

2) 本体での処理

- ベーシックモデル QCPU 使用時：
ユニット前面の ERR LED が点灯します。
- ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU 使用時：
ユニット前面の USER LED (赤色) が点灯します。

3) LED の点灯／消灯の選択

エラー発生時の表示優先順位を SD207 ~ 209 に設定することにより、アナシエータが ON した場合の USER.LED (ベーシックモデル QCPU では、ERR.LED) の点灯／消灯の選択ができます。

(☞ 6.21.2 項)

(6) アナシエータの OFF 方法と処理内容

(a) アナシエータの OFF 方法

下記に示す命令で行うことができます。

1) RST F_n 命令

SET F_n 命令で ON したアナシエータ番号を OFF する場合に使用します。

2) LEDR 命令 注 9.3

SD62,SD64 に格納されているアナシエータ番号を OFF する場合に使用します。

3) BKRST 命令

指定範囲内のアナシエータ番号を一括で OFF する場合に使用します。

4) OUT F_n 命令

アナシエータ番号の ON/OFF を同一の命令で行います。

ただし、OUT F_n 命令でアナシエータ番号を OFF しても、本項 (6)(b) のアナシエータ OFF 時の処理を行いません。

OUT F_n 命令でアナシエータを OFF した場合は、RST F_n 命令、LEDR 命令または BKRST 命令の実行が必要です。

- アナシエータ 5(F5) を OFF する場合



- 使用しているアナシエータをすべて OFF する場合



図 9.13 アナシエータを OFF するときの例

備考

各命令の詳細については、下記マニュアルを参照してください。

 MELSEC-Q/L プログラミングマニュアル (共通命令編)



注 9.3

Basic

ベーシックモデル QCPU では、LEDR 命令が使用できません。

(b) アナシエータ OFF 時の処理内容

1) LEDR 命令実行時の特殊レジスタ (SD62 ~ 79) の格納データ

- SD64 に格納されているアナシエータ番号を削除し, SD65 以降に格納されているアナシエータ番号が前づめされます。
- SD64 に格納されたアナシエータ番号が, SD62 に格納されます。
- SD63 の内容が -1 されます。
- SD63 が 0 になった場合は, SM62 を OFF します。

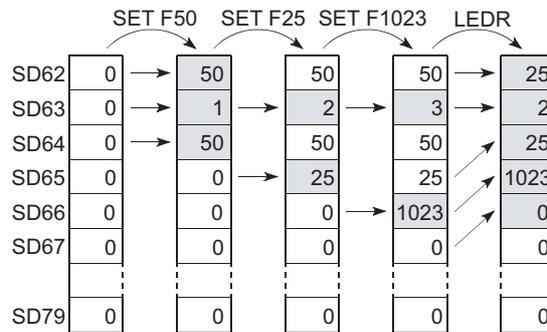


図 9.14 アナシエータ OFF 時の処理内容 (LEDR 命令実行時)

2) RST F 命令または BKRST 命令実行でアナシエータを OFF した場合の特殊レジスタ (SD62 ~ 79) の格納データ

- RST 命令 / BKRST 命令で指定したアナシエータ番号が削除され, 削除されたアナシエータ以降に格納されているアナシエータ番号が前づめされます。
- SD64 に格納されているアナシエータ番号を OFF した場合は, 新しく SD64 に格納されたアナシエータ番号が SD62 に格納されます。
- SD63 の内容が -1 されます。
- SD63 が 0 になった場合は, SM62 を OFF します。

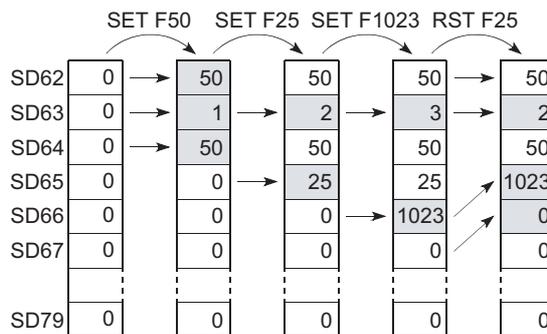


図 9.15 アナシエータ OFF 時の処理内容 (RST F 命令実行時)

3) LED の表示

SD64 ~ 79 のアナシエータ番号がすべて OFF すると, アナシエータの ON によって点灯した LED (本項 (5)(b)) は消灯します。

Point

アナシエータの ON, およびアナシエータより優先順位 (6.21.2 項) の高い演算を続行するエラーが同時に発生している場合に LEDR 命令を実行すると, LEDR 命令はアナシエータより優先順位の高いエラーの解除を行います。このため, LEDR 命令を実行してもアナシエータの OFF ができません。

LEDR 命令を使用してアナシエータの OFF を行う場合は, アナシエータより優先順位の高いエラー要因を除いてから LEDR 命令を実行してください。

9.2.6 エッジリレー (V)

(1) エッジリレーとは

回路ブロックの先頭からの接点の ON/OFF 情報を記憶するデバイスです。
接点でのみ使用できます。(コイルとしての使用はできません。)

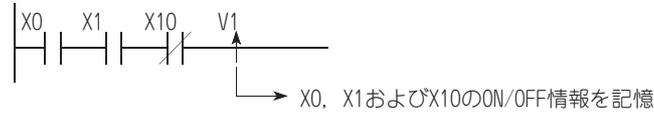
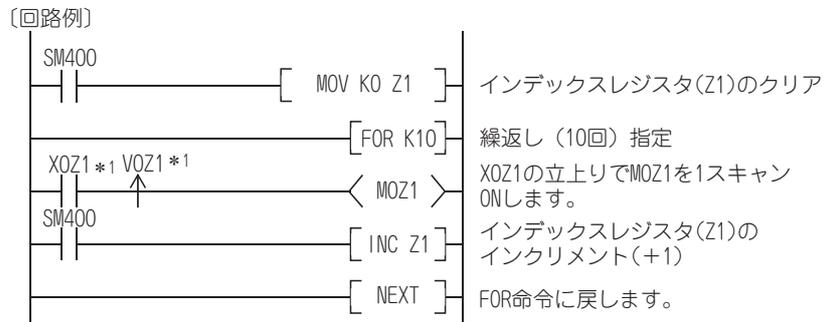


図 9.16 エッジリレー

(2) エッジリレーの用途

エッジリレーは、インデックス修飾を使用したプログラムで、立上り (OFF → ON) 検出を行って実行させる場合に使用します。



* 1 : XOZ1 の ON/OFF 情報を、エッジリレーの VOZ1 で記憶します。
例えば、X0 の ON/OFF 情報を V0 で記憶し、X1 の ON/OFF 情報を V1 で記憶していきます。

〔タイミングチャート〕

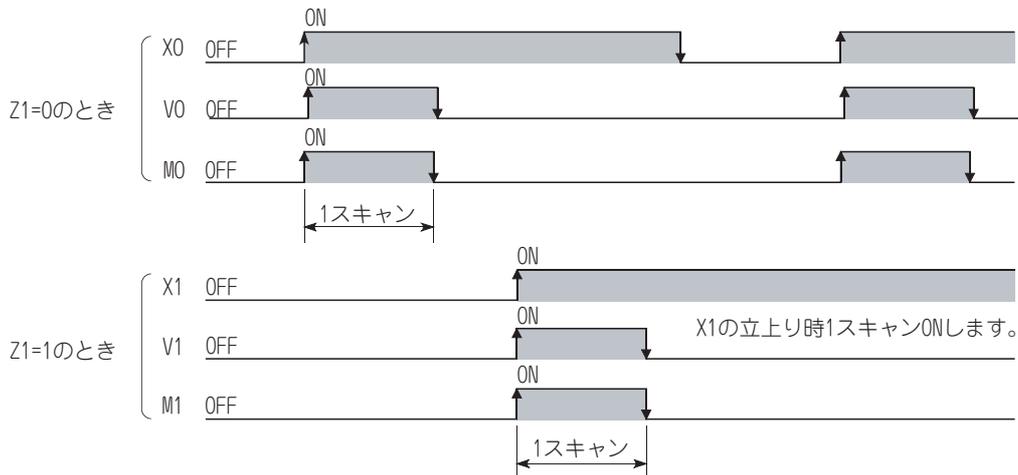


図 9.17 エッジリレーの使用例とタイミングチャート

(3) 注意事項

プログラムにおいて、同じ番号のエッジリレーは複数個使用できません。

9.2.7 リンクリレー (B)

(1) リンクリレーとは

MELSECNET/H ネットワークユニットなどのリンクリレー (LB) を CPU ユニットにリフレッシュする場合、または CPU ユニット内のデータを、MELSECNET/H ネットワークユニットなどのリンクリレー (LB) にリフレッシュする場合に使用する CPU ユニット側のリレーです。

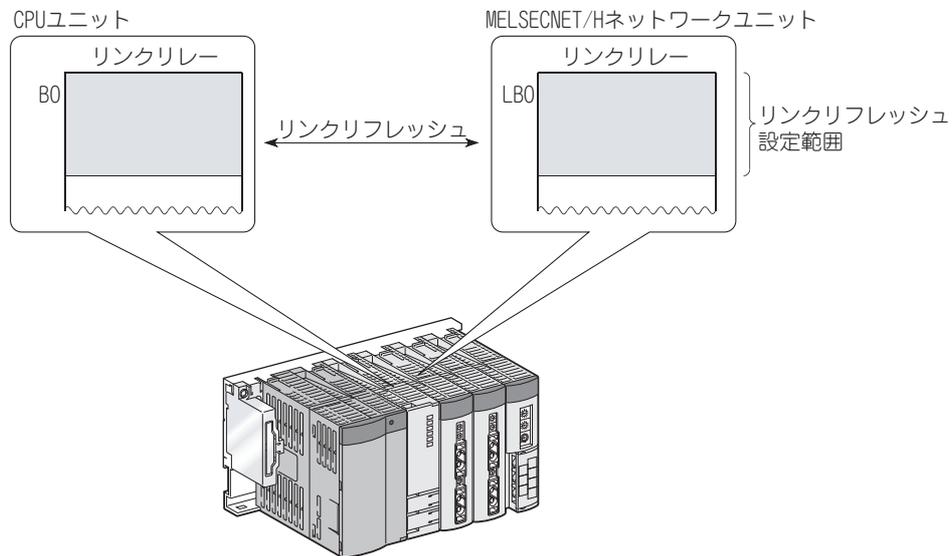


図 9.18 リンクリフレッシュ

(2) a 接点と b 接点の使用数

プログラム内での接点 (a 接点, b 接点) の使用数は、プログラム容量の範囲内であれば制限はありません。

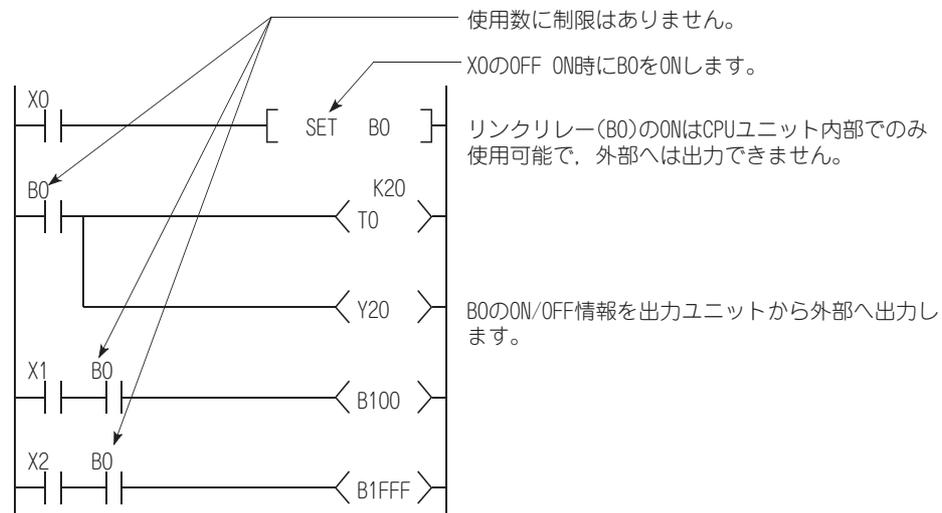


図 9.19 リンクリレー

(3) ネットワークシステムで使用する場合

ネットワークパラメータの設定が必要です。

ネットワークパラメータを設定していない (MELSECNET/H ネットワークシステムなどで使用しない) 範囲は、内部リレーまたはラッチリレーとして使用できます。

- リンクリレーでラッチを行わない範囲 …… 内部リレー相当
- リンクリレーでラッチを行う範囲 …………… ラッチリレー相当

Point

- CC-Link IE コントローラネットワークユニット内のリンクリレーは 32768 点ですが、CPU ユニットのリンクリレーのデフォルト値は、ベーシックモデル QCPU が 2048 点、ハイパフォーマンスモデル QCPU、プロセス CPU、二重化 CPU が 8192 点です。
- MELSECNET/H ネットワークユニット内のリンクリレーは 16384 点ですが、CPU ユニットのリンクリレーのデフォルト値は、ベーシックモデル QCPU が 2048 点、ハイパフォーマンスモデル QCPU、プロセス CPU、二重化 CPU が 8192 点です。

上記に示した点数以降のリンクリレーを使用する場合は、PC パラメータのデバイス設定でリンクリレーの点数を変更してください。

備考

ネットワークパラメータについては、下記マニュアルを参照してください。

 各ネットワークユニットのマニュアル

9.2.8 リンク特殊リレー (SB)

(1) リンク特殊リレーとは

CC-Link IE コントローラネットワークユニット, MELSECNET/H ネットワークユニットなどのインテリジェント機能ユニットの通信状態・異常検出を示すリレーです。

データリンク時に発生するさまざまな要因により ON/OFF 制御されます。

リンク特殊リレーをモニタすることにより, データリンクの通信状態・異常状態などを把握できます。

(2) リンク特殊リレー点数

リンク特殊リレーの点数は, CPU ユニットにより異なります。

表 9.4 CPU ユニットごとのリンク特殊リレーの点数

CPU ユニット	リンク特殊リレー点数																		
ベーシックモデル QCPU	CPU ユニットのリンク特殊リレーの点数は 1024 点 (SB0 ~ 3FF) です。CC-Link IE コントローラネットワークユニット, MELSECNET/H ネットワークユニットなどのリンク特殊リレーを持つインテリジェント機能ユニットは 512 点になります。																		
ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU	<p>CPU ユニットのリンク特殊リレーの点数は 2048 点 (SB0 ~ 7FF) です。CC-Link IE コントローラネットワークユニット, MELSECNET/H ネットワークユニットなどのリンク特殊リレーを持つインテリジェント機能ユニットは 512 点になります。</p> <p>リンク特殊リレーは, 下図のように割り付けることが可能です。</p> <div style="text-align: center;"> <p>リンク特殊リレー</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding: 2px;">SB0 └</td> <td style="padding: 2px;">1枚目の ネットワークユニット用</td> <td rowspan="4" style="font-size: 2em; vertical-align: middle;">} 2048点</td> </tr> <tr> <td style="padding: 2px;">SB1FF └</td> <td style="padding: 2px;">512点</td> </tr> <tr> <td style="padding: 2px;">SB200 └</td> <td style="padding: 2px;">2枚目の ネットワークユニット用</td> </tr> <tr> <td style="padding: 2px;">SB3FF └</td> <td style="padding: 2px;">512点</td> </tr> <tr> <td style="padding: 2px;">SB400 └</td> <td style="padding: 2px;">3枚目の ネットワークユニット用</td> <td rowspan="4" style="font-size: 2em; vertical-align: middle;">} 2048点</td> </tr> <tr> <td style="padding: 2px;">SB5FF └</td> <td style="padding: 2px;">512点</td> </tr> <tr> <td style="padding: 2px;">SB600 └</td> <td style="padding: 2px;">4枚目の ネットワークユニット用</td> </tr> <tr> <td style="padding: 2px;">SB7FF └</td> <td style="padding: 2px;">512点</td> </tr> </table> </div>	SB0 └	1枚目の ネットワークユニット用	} 2048点	SB1FF └	512点	SB200 └	2枚目の ネットワークユニット用	SB3FF └	512点	SB400 └	3枚目の ネットワークユニット用	} 2048点	SB5FF └	512点	SB600 └	4枚目の ネットワークユニット用	SB7FF └	512点
SB0 └	1枚目の ネットワークユニット用	} 2048点																	
SB1FF └	512点																		
SB200 └	2枚目の ネットワークユニット用																		
SB3FF └	512点																		
SB400 └	3枚目の ネットワークユニット用	} 2048点																	
SB5FF └	512点																		
SB600 └	4枚目の ネットワークユニット用																		
SB7FF └	512点																		

備考

リンク特殊リレーの詳細については, リンク特殊リレーを持つインテリジェント機能ユニットのマニュアルを参照してください。

9.2.9 ステップリレー (S)

SFC プログラム用のデバイスです。

Point

SFC プログラム専用のデバイスのため、シーケンスプログラムで内部リレーの代用として使用できません。
使用した場合は、SFC エラーが発生しシステムダウンすることがあります。

備考

ステップリレーの使用方法については、下記マニュアルを参照してください。

 MELSEC-Q/L/QnA プログラミングマニュアル (SFC 編)

9.2.10 タイマ (T)

(1) タイマとは

タイマのコイルが ON すると計測を開始し、現在値が設定値と同一の値になるとタイムアップとなり接点が ON します。

タイマは加算式です。

(2) タイマの種類

タイマには大きく分けて下記の 2 種類があります。

- 1) コイルが OFF したとき現在値が 0 になるタイマ
- 2) コイルが OFF しても現在値を保持する積算タイマ

さらに、1) には低速タイマと高速タイマがあり、2) には低速積算タイマと高速積算タイマがあります。

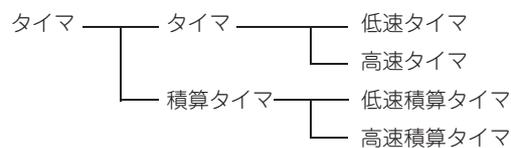


図 9.20 タイマの種類

(3) タイマの指定方法

低速タイマと高速タイマは同一デバイスで、タイマの指定（命令の書き方）で低速タイマまたは高速タイマになります。

例 OUT T0 を指定すると低速タイマになり、OUTH T0 を指定すると高速タイマになります。

低速積算タイマと高速積算タイマは同一デバイスで、タイマの指定（命令の書き方）で低速積算タイマまたは高速積算タイマになります。

例 OUT ST0 を指定すると低速積算タイマになり、OUTH ST0 を指定すると高速積算タイマになります。

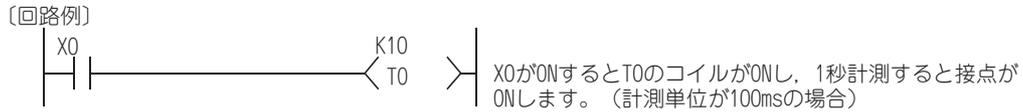
(4) 低速タイマ

(a) 低速タイマとは

計測単位が 1 ～ 1000ms のタイマです。

タイマのコイルが ON すると計測を開始し、タイムアップすると接点が ON します。

タイマのコイルが OFF すると現在値が 0 になり、接点も OFF します。



〔タイムチャート〕

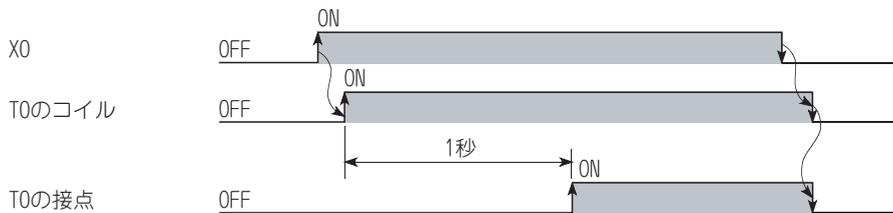


図 9.21 低速タイマの回路例とタイムチャート

(b) 計測単位の設定

PC パラメータの PC システム設定で行います。

デフォルト値が 100ms で、1ms 単位で変更できます。

(5) 高速タイマ

(a) 高速タイマとは

計測単位が 0.1 ～ 100ms のタイマです。

タイマのコイルが ON すると計測を開始し、タイムアップすると接点が ON します。

タイマのコイルが OFF すると現在値が 0 になり、接点も OFF します。



〔タイムチャート〕

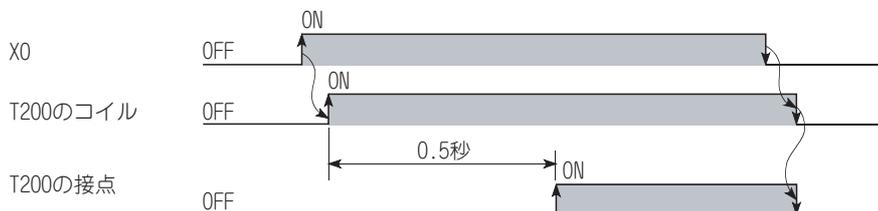


図 9.22 高速タイマの回路例とタイムチャート

(b) 計測単位の設定

PC パラメータの PC システム設定で行います。

デフォルト値が 10.0ms で、0.1ms 単位で変更できます。

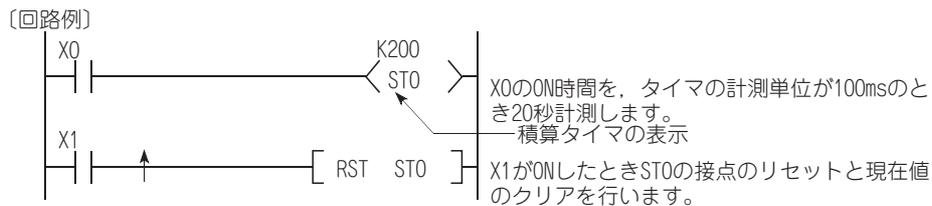
(6) 積算タイマ

(a) 積算タイマとは

コイルが ON している時間を計測するタイマです。
 タイマのコイルが ON すると計測を開始し、タイムアップすると接点が ON します。
 タイマのコイルが OFF になっても現在値、接点の ON/OFF 状態を保持します。
 再度コイルが ON すると、保持していた現在値から計測を再開します。

(b) 積算タイマのクリア

現在値のクリアと接点の OFF は、RST ST0 命令で行います。



〔タイムチャート〕

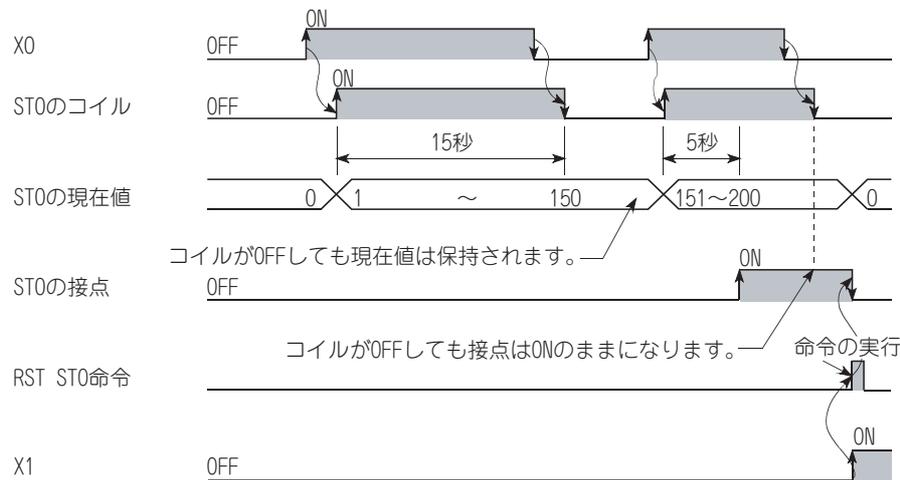


図 9.23 積算タイマの回路例とタイムチャート

(c) 計測単位の設定

積算タイマの計測単位設定は、低速タイマ、高速タイマと同一です。

- 低速積算タイマ：低速タイマ
- 高速積算タイマ：高速タイマ

Point

積算タイマを使用する場合は、PC パラメータのデバイス設定で積算タイマの使用点数を設定してください。

(7) タイマの処理方法と精度

(a) 処理方法

OUT T□□, OUT ST□□命令実行時にタイマのコイルの ON/OFF, 現在値の更新および接点の ON/OFF 処理を行います。

END 処理では, タイマの現在値の更新と接点の ON/OFF 処理を行いません。

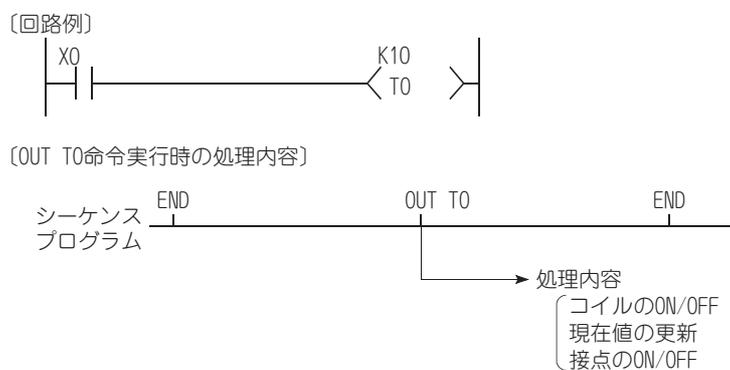


図 9.24 OUT T0 命令実行時の処理内容

(b) 精度

現在値は、END 命令での計数を OUT T₀、OUT ST₀ 命令実行時に加算します。

OUT T₀、OUT ST₀ 命令実行時タイマのコイルが OFF している場合は、現在値を更新しません。

タイマ時限設定=10ms, T₀の設定値=8 (10ms×8=80ms), スキャンタイム=25ms

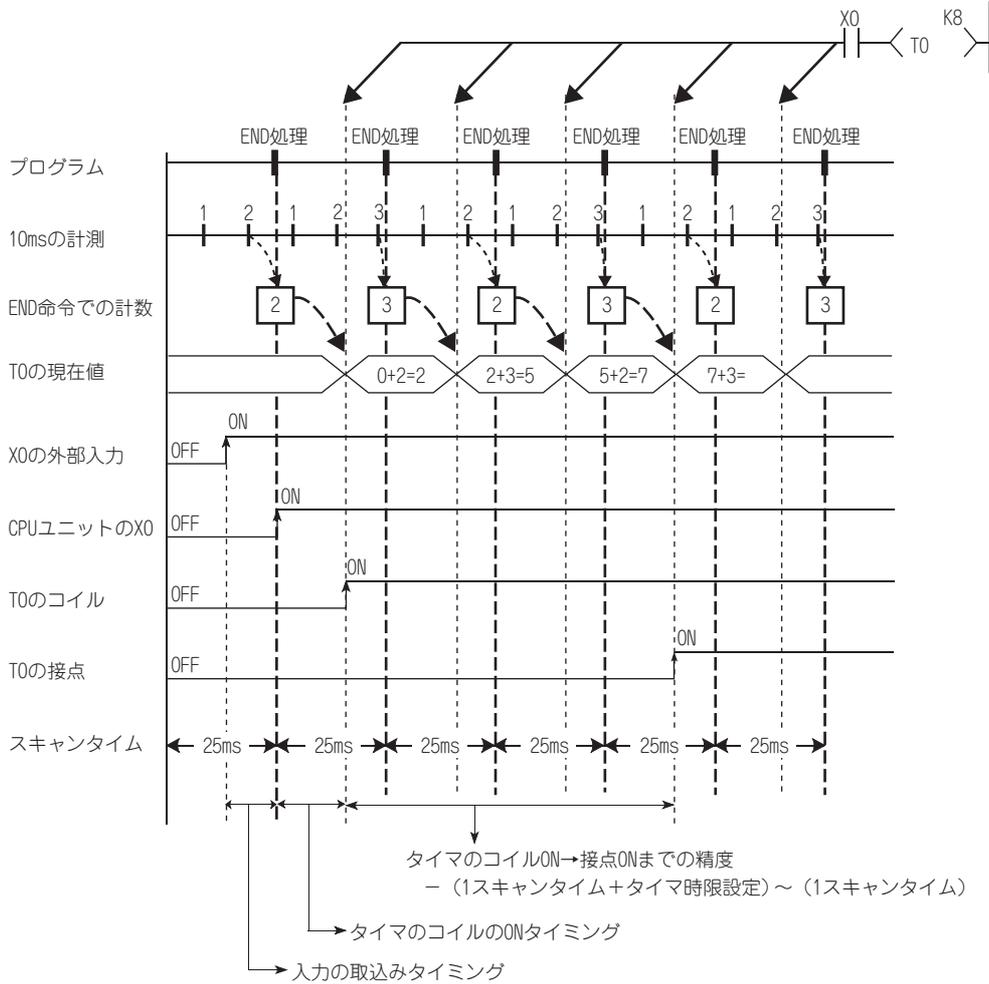


図 9.25 タイマの精度 (10ms の場合)

入力 (X) を取り込んで出力するまでのタイマの応答精度は、最大で「2 スキャンタイム+タイマの時限設定」になります。

(8) タイマ使用時の注意事項

(a) 同一タイマの使用について

1 スキャン中に OUT T□□ で同一タイマを複数記述しないでください。

同一タイマを複数記述すると、各 OUT T□□ 命令実行時にタイマの現在値を更新するため、正常に計測されません。

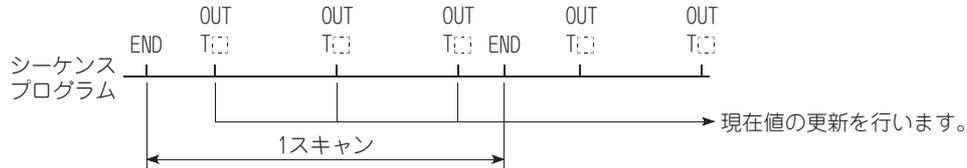


図 9.26 同一タイマ使用時

(b) タイマを毎スキャン実行しない場合

タイマ（例：T1）のコイルが ON 中に OUT T1 命令を、CJ 命令などで飛ばさないでください。

OUT T□□ 命令が飛ばされていると、タイマの現在値は更新されません。

また、サブルーチンプログラム内にタイマが存在する場合は、タイマ（例：T1）のコイルが ON 中に、OUT T1 命令を含むサブルーチンコールを必ず毎スキャン 1 回のみ実行してください。

サブルーチンコールを毎スキャン実行しないと、タイマの現在値は更新されません。

(c) タイマを使用できないプログラムについて

割込みプログラム／定周期実行タイププログラムでは、タイマは使用できません。

(d) 設定値が 0 の場合

OUT T□□ 命令実行時に接点が ON します。

(e) タイマ設定値とタイマ時限設定

タイマは、下記の条件を満たすように設定してください。



図 9.27 タイマ設定値とタイマ時限設定の関係

「タイマ設定値 < スキャンタイム + タイマ時限設定」になっていると、コイルが ON するタイミングにより、コイルと接点が同時に ON する場合があります。

条件を満たしていない場合は、タイマ時限設定を小さくし、条件を満たすようにしてください。

例 低速タイマから高速タイマに変更し、タイマ時限設定を小さくする例（スキャンタイムは 20ms とします）

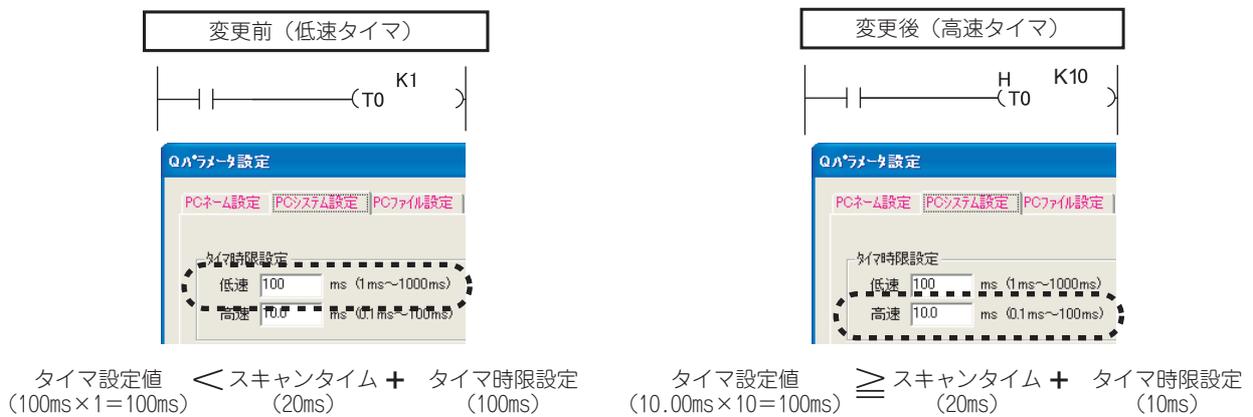


図 9.28 タイマ時限設定を小さくする例

「タイマ設定値 < スキャンタイム + タイマ時限設定」の場合に、コイルと接点が同時に ON する例を示します。

例 タイマ設定値が 1(1×100ms)、スキャンタイムが 20ms、タイマ時限設定が 100ms の場合
 「END 命令での係数 ≥ タイマ設定値」となった次のスキャンでタイマ (T0) のコイルが ON すると、タイマ起動時に「タイマ現在値 = タイマ設定値」となるため、コイルと接点が同時に ON します。

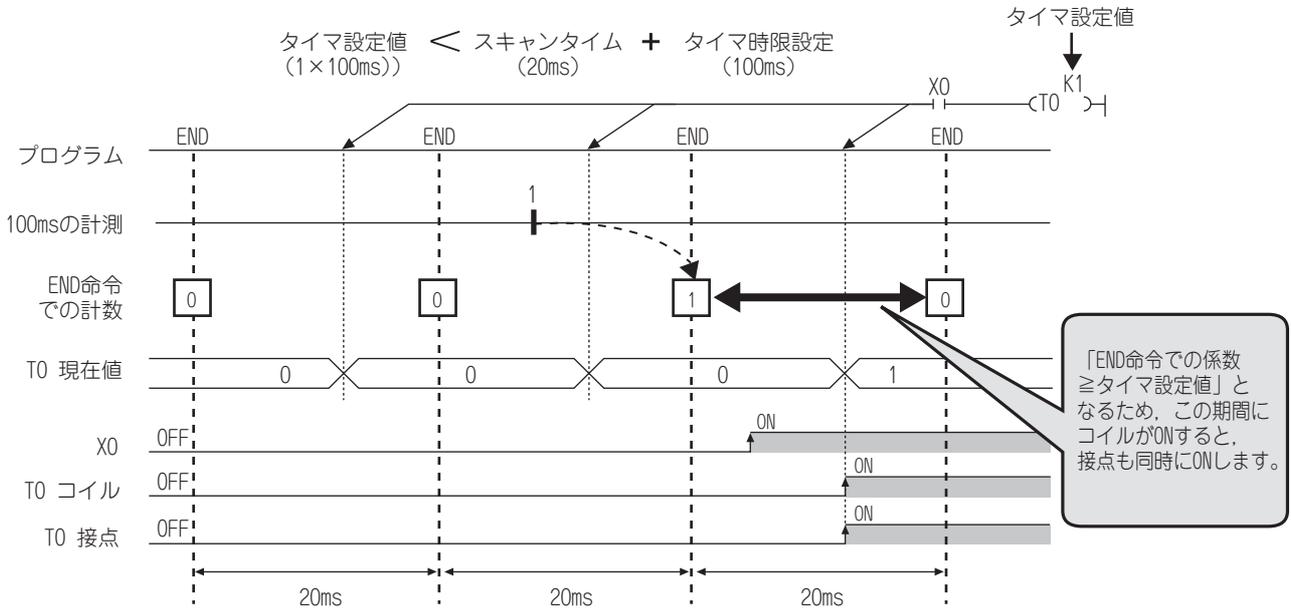


図 9.29 タイマ設定値が 1 の場合にコイルと接点が同時に ON するとき

例 タイマ設定値が 2(2×100ms)、スキャンタイムが 110ms、タイマ時限設定が 100ms の場合
 「END 命令での係数 ≥ タイマ設定値」となった次のスキャンでタイマ (T0) のコイルが ON すると、タイマ起動時に「タイマ現在値 = タイマ設定値」となるため、コイルと接点が同時に ON します。

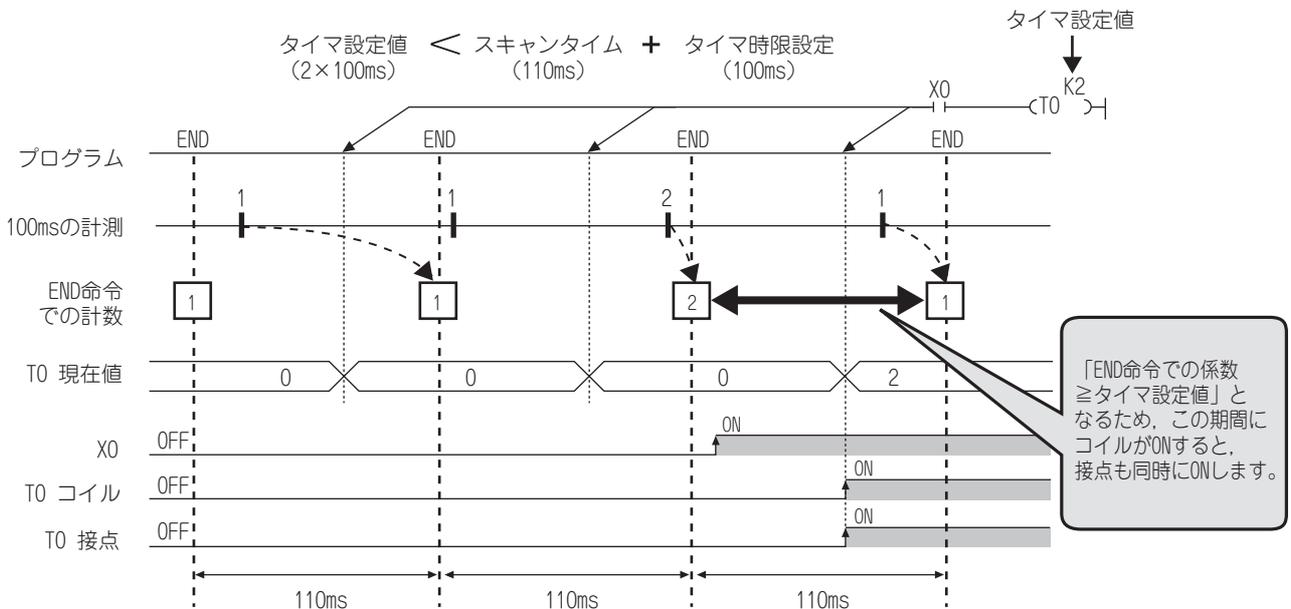


図 9.30 タイマ設定値が 2 の場合にコイルと接点が同時に ON するとき

(f) タイムアップ後に設定値を変更した場合

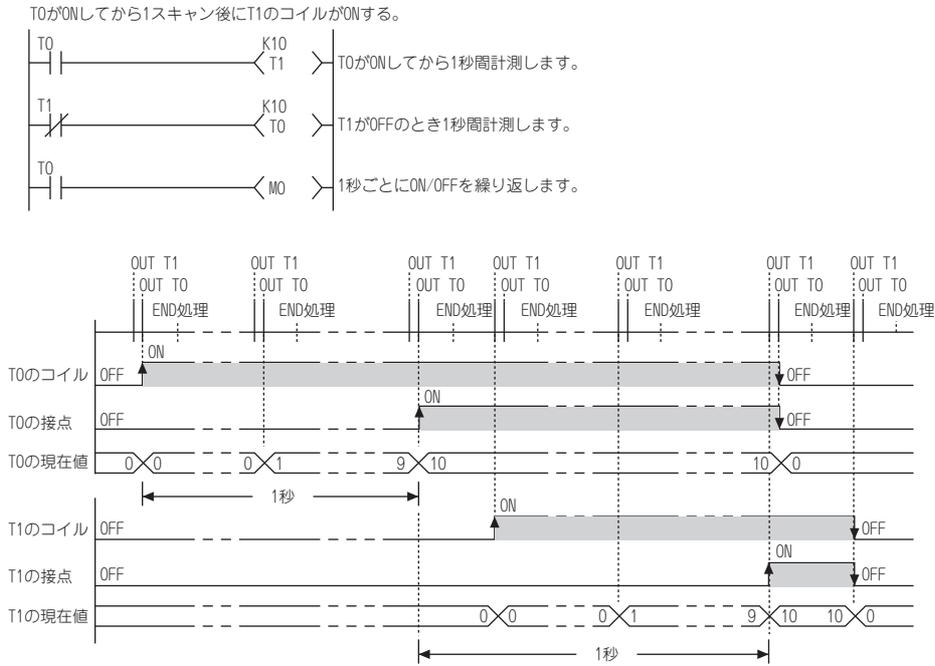
タイマがタイムアップ後に、設定値を現在の値より大きい値に変更しても、タイマはタイムアップしたまま動作しません。

(g) 複数のタイマを使用する場合

各 OUT T[]命令実行時にタイマの現在値を更新するため、複数のタイマを使用する場合は、回路の順序に注意してください。

例えば、タイマを2個使用して ON/OFF 回路を作成する場合、図 9.31 に示す回路のようにしてください。

[正しい回路例]



[間違った回路例]

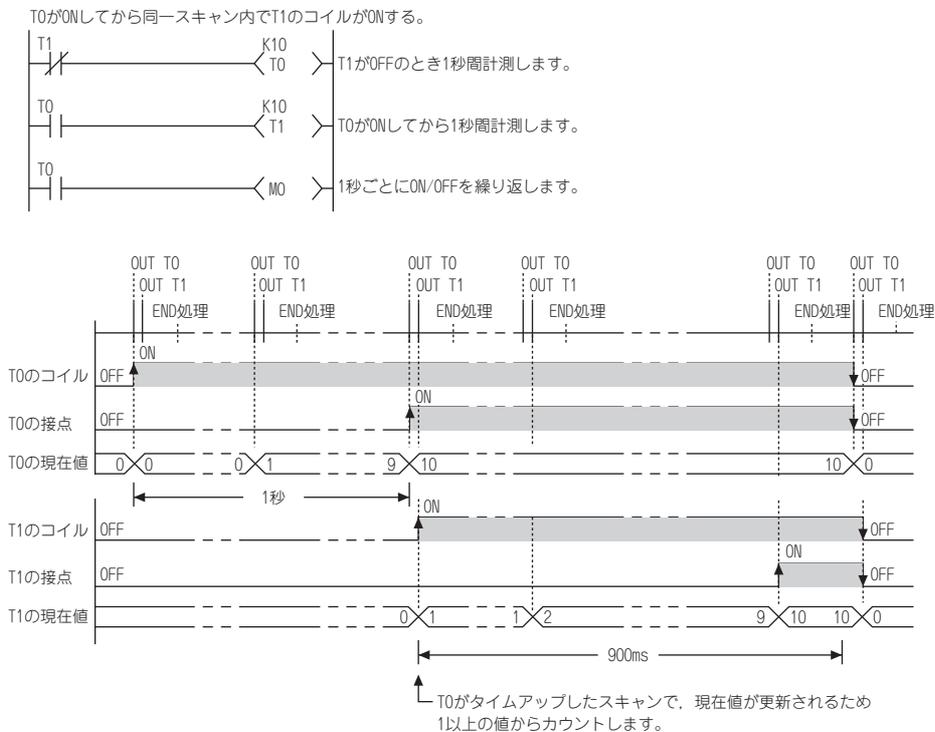


図 9.31 タイマを2個使用した ON/OFF 回路

9.2 内部ユーザデバイス
9.2.10 タイマ(T)

9.2.11 カウンタ (C)

(1) カウンタとは

シーケンスプログラムで入力条件の立上り回数をカウントするデバイスです。
カウント値と設定値が同一になるとカウントアップし、接点が ON します。
カウンタは加算式です。

(2) カウンタの種類

下記の 2 種類があります。

- ・シーケンスプログラムで入力条件の立上り回数をカウントするカウンタ
- ・割込み要因の発生回数をカウントする割込みカウンタ

(3) カウント処理

(a) OUT C□命令の実行時

カウンタのコイルの ON/OFF、現在値の更新（カウント値 + 1）および接点の ON を行います。
END 処理では、カウンタの現在値の更新と接点の ON は行いません。

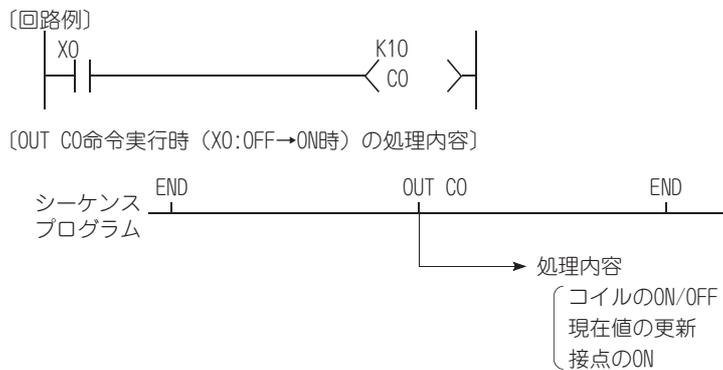


図 9.32 OUT C□命令の実行と処理内容

(b) 現在値の更新（カウント値 + 1）

OUT C□命令の立上り時（OFF → ON）に行います。

OUT C□命令が OFF、ON → ON および ON → OFF 時には、現在値を更新しません。

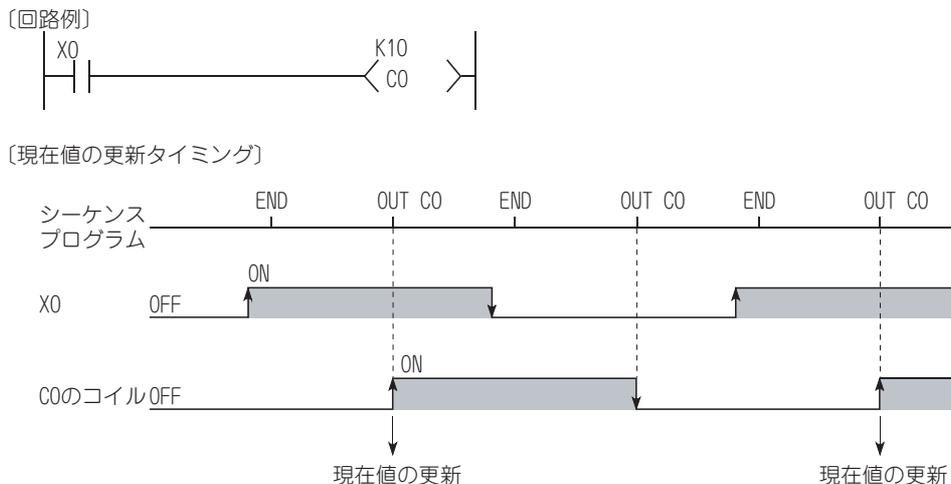


図 9.33 現在値の更新タイミング

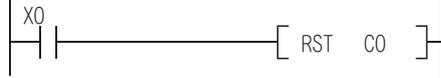
(c) カウンタのリセット

カウンタの現在値は、OUT C₀命令が OFF してもクリアされません。

カウンタの現在値のクリア（リセット）と接点の OFF は、RST C₀命令で行います。

RST C₀命令を実行した時点でカウンタ値はクリアされ、接点も OFF します。

〔回路例〕



〔カウンタのリセットタイミング〕

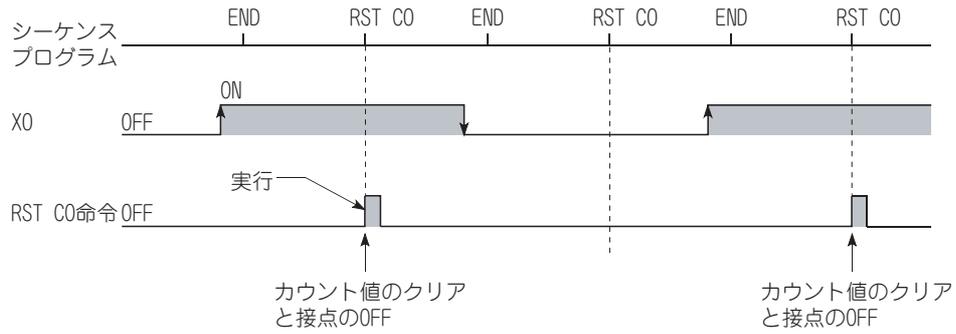


図 9.34 カウンタのリセット

1) カウンタリセット時の注意事項

RST C \square 命令を実行すると C \square のコイルも OFF します。

RST C \square 命令実行後も OUT C \square 命令の実行条件が ON している場合は、OUT C \square 命令実行時に C \square のコイルを ON し、現在値を更新（カウント値 + 1）します。

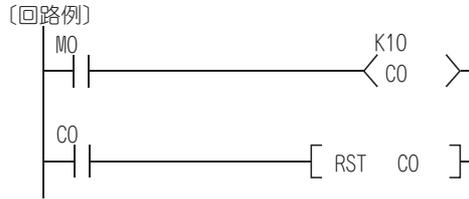


図 9.35 カウンタリセットの回路例

上記の回路例では、M0 が OFF → ON で C0 のコイルが ON し、現在値が更新されます。

C0 がカウントアップすると C0 の接点が ON し、RST C0 命令の実行により C0 の現在値がクリアされます。

このとき、C0 のコイルも OFF します。

次のスキャンで M0 が ON している場合は、OUT C0 命令実行時に C0 のコイルが OFF → ON となるため、現在値を更新してしまいます。（現在値が 1 になります。）

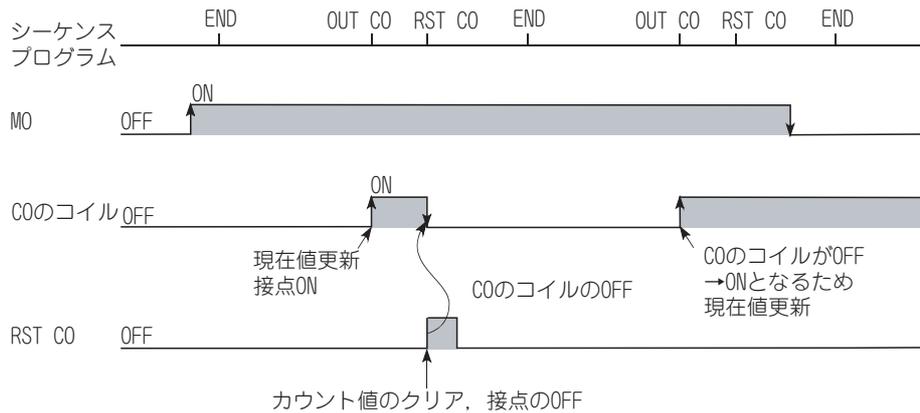


図 9.36 現在値更新のタイミング

上記の対応として、RST C0 命令の実行条件に OUT C0 命令の実行条件の b 接点を挿入し、OUT C0 命令の実行条件 (M0) が ON の間は C0 のコイルが OFF しないようにすることを推奨します。

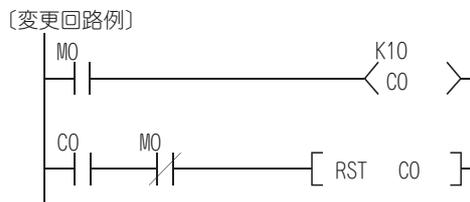


図 9.37 カウンタリセットの回路例 (推奨例)

(d) カウンタの最大計数速度

カウンタは、入力条件の ON/OFF 時間が同一 OUT C 命令の実行間隔より大きい場合のみカウントできません。

カウンタの最大計数速度は、下式で算出します。

$$\text{最大計数速度 } C_{\text{max}} = \frac{n}{100} \times \frac{1}{T} \text{ [回/S]}$$

n : デューティ (%) * 1
T : OUT C 命令の実行間隔 (sec)

* 1 : デューティ (n) は、カウント入力信号の ON/OFF 時間の比率をパーセント (%) で表したものです。

- ・ T1 ≥ T2 のとき $n = \frac{T2}{T1 + T2} \times 100\%$
- ・ T1 < T2 のとき $n = \frac{T1}{T1 + T2} \times 100\%$

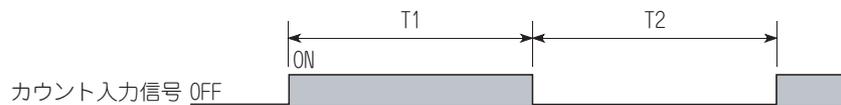


図 9.38 デューティ比

Point

1 スキャン中に複数のカウンタを記述し、最大計数速度を上げることができます。
このとき、カウンタの入力信号は、ダイレクトアクセス入力 (DX) (3.8.2 項) を使用します。

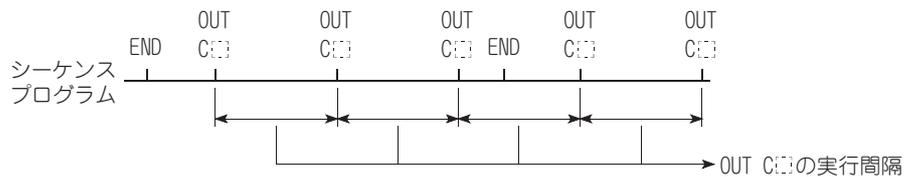


図 9.39 最大計数速度を上げる回路例

(4) 割込みカウンタ

(a) 割込みカウンタとは

割込みの要因発生回数をカウントするデバイスです。

(b) カウント処理

1) 割込み発生時

割込みカウンタは、割込みが発生したときにカウンタの現在値の更新を行います。

割込みカウンタでカウントを行う場合には、割込みカウンタを使用したプログラムを作成する必要はありません。

2) 割込みカウンタのカウントについて

割込みカウンタは、設定値が指定されていてもカウントアップしません。

割込みカウンタを制御に使用する場合は、比較命令 (=, <= など) で設定値と比較して内部リレー (M) などを ON/OFF させてください。

I0 の割込み入力が 10 回 ON したとき、M0 を ON させる場合は、図 9.40 のようなプログラムを作成します。(I0 に対応する割込みカウンタを C300 と仮定します。)

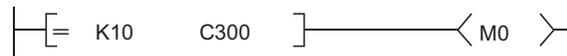


図 9.40 割込みカウンタを制御に使用する場合の回路例

(c) 割込みカウンタの設定

PC パラメータの PC システム設定で “割込みカウンタ先頭 No.” を設定します。

設定されたカウンタ番号から表 9.5 に示す点数が割込みカウンタになります。

表 9.5 CPU ユニットごとの割込みカウントの点数

ベーシックモデル QCPU	ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU
<p>設定されたカウンタ番号から 128 点が割込みカウンタになります。割込みカウンタ先頭 No. を C300 に設定した場合は、C300 ~ 427 が割込みカウンタになります。</p>	<p>設定されたカウンタ番号から 256 点が割込みカウンタになります。割込みカウンタ先頭 No. を C300 に設定した場合は、C300 ~ 555 が割込みカウンタになります。</p>

割込みカウンタを使用する場合は、メインルーチンプログラムで EI 命令により、割込み許可状態にしてください。

(5) 注意事項

(a) 割込みカウンタと割込みプログラムの実行について

1つの割込みポインタで割込みカウンタによるカウントと割込みプログラムの実行を行うことはできません。PCパラメータのPCシステム設定で割込みカウンタの設定を行うと、割込みプログラムを実行できません。

(b) カウント処理が待たされる処理について

下記に示す各処理中に割込みが発生した場合は、各処理の実行が終了するまでカウント処理が待たされます。

- シーケンスプログラムの各々の命令を実行中
- 割込みプログラム実行中
- 定周期実行タイププログラムの実行中

各処理の実行が終了した時点でカウント処理を行います。

ただし、各処理中に再度同一の割込みが発生した場合は、1回のみカウントを行います。

(c) 割込みカウンタの最大計数速度について

下記に示す処理時間のうち一番長い処理により決まります。

- プログラム中で使用している命令のうち、処理時間が一番長い命令
- 割込みプログラムの処理時間
- 定周期実行タイププログラムの処理時間

(d) 割込みカウンタを多点多数使用した場合について

シーケンスプログラムの処理時間が長くなり“WDT ERROR”となることがあります。

この場合は、割込みカウンタの点数を減らすか、入力パルス信号の計数速度を遅くしてください。

(e) 割込みカウンタのリセットについて

メインルーチンプログラムのRST C_n命令で行ってください。

(f) カウント値の読出しについて

割込みカウンタのカウント値は、シーケンスプログラムのMOV命令で読み出すことができます。

9.2.12 データレジスタ (D)

(1) データレジスタとは

数値データ (-32768 ~ 32767 または 0000H ~ FFFFH) を格納できるメモリです。

(2) データレジスタのビット構成

(a) ビット構成と読み出しおよび書き込み単位

データレジスタは、1 点 16 ビットで構成され、16 ビット単位で読み出しおよび書き込みができます。

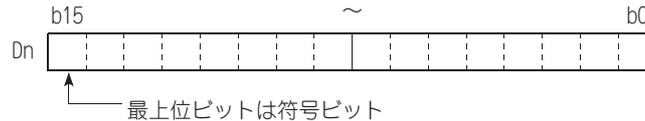


図 9.41 データレジスタのビット構成

Point

データレジスタは符号付きで扱います。

HEX(16 進数) の場合は 0000H ~ FFFFH を格納可能ですが、最上位のビットが符号ビットとなるため、指定可能な数値の範囲は、-32768 ~ 32767 となります。

(b) 32 ビット命令でデータレジスタを使用するとき

連続した 2 点のデータレジスタ (Dn と Dn+1) が処理対象になります。

シーケンスプログラムで指定しているデータレジスタ番号 (Dn) が下位 16 ビット、シーケンスプログラムで指定しているデータレジスタ番号 + 1 のデータレジスタが上位 16 ビットになります。

例 DMOV 命令で、D12 を指定した場合は、D12 が下位 16 ビット、D13 が上位 16 ビットになります。

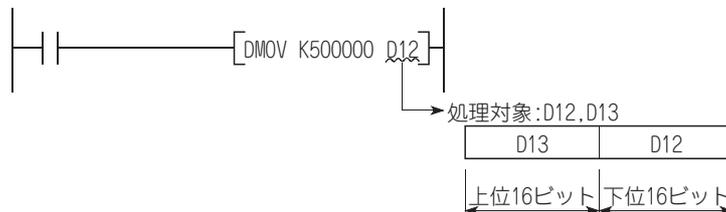


図 9.42 32 ビット命令によるデータの転送と格納先

データレジスタ 2 点では、-2147483648 ~ 2147483647 または 0H ~ FFFFFFFFH のデータを格納できます。(32 ビット構成時の最上位ビットは符号ビットになります。)

(3) 格納データの保持について

データレジスタに格納したデータは、ほかのデータを格納するまで保持されます。

ただし、CPU ユニットの電源 OFF、またはリセットを行うと初期化されます。

9.2.13 リンクレジスタ (W)

(1) リンクレジスタとは

MELSECNET/H ネットワークユニットなどインテリジェント機能ユニットのリンクレジスタ (LW) のデータを、CPU ユニットへリフレッシュする場合の CPU ユニット側のメモリです。

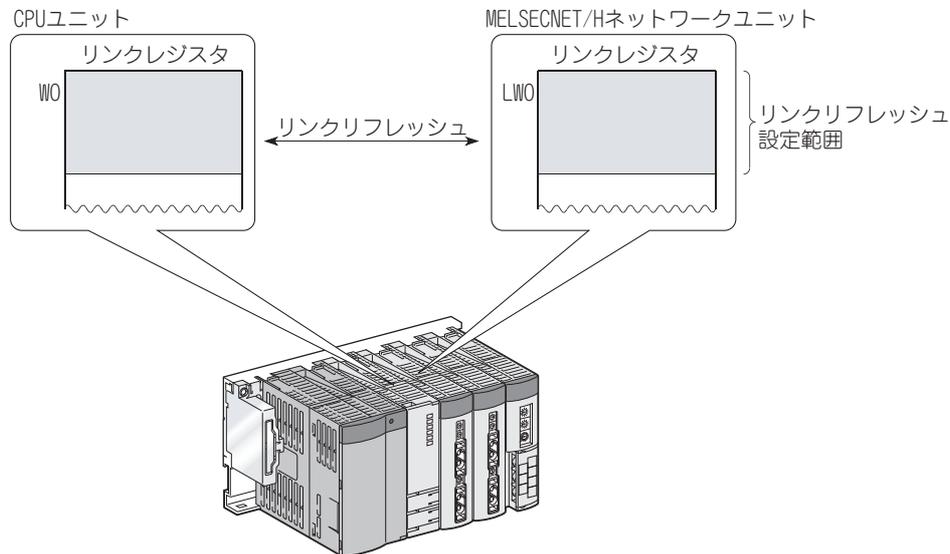


図 9.43 リンクリフレッシュ

リンクレジスタには、数値データ (-32768 ~ 32767 または 0000H ~ FFFFH) を格納できます。

(2) リンクレジスタのビット構成

(a) ビット構成と読出しおよび書込み単位

リンクレジスタは 1 点 16 ビットで構成され、16 ビット単位で読出しおよび書込みができます。



図 9.44 リンクレジスタのビット構成

Point

リンクレジスタは符号付きで扱います。HEX (16 進数) の場合は 0000H ~ FFFFH を格納可能ですが、最上位のビットが符号ビットとなるため、指定可能な数値の範囲は、-32768 ~ 32767 となります。

(b) 32ビット命令でリンクレジスタを使用するとき

連続した2点のリンクレジスタ (W_n と W_{n+1}) が処理対象になります。

シーケンスプログラムで指定しているリンクレジスタ番号 (W_n) が下位 16 ビット、シーケンスプログラムで指定しているリンクレジスタ番号 + 1 のリンクレジスタが上位 16 ビットになります。

例 DMOV 命令で、W12 を指定した場合は、W12 が下位 16 ビット、W13 が上位 16 ビットになります。

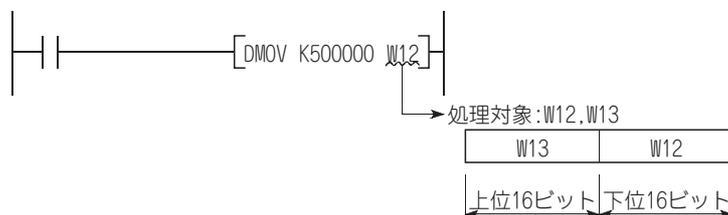


図 9.45 32ビット命令によるデータの転送と格納先

リンクレジスタ 2 点では、-2147483648 ~ 2147483647 または 0H ~ FFFFFFFFH のデータを格納できます。(32 ビット構成時の最上位ビットは符号ビットになります。)

(3) 格納データの保持について

リンクレジスタに格納したデータは、ほかのデータを格納するまで保持されます。

ただし、シーケンサの電源 OFF、またはリセットを行うと初期化されます。

Point

- CC-Link IE コントローラネットワークユニット内のリンクレジスタは 131072 点ですが、CPU ユニット内のリンクレジスタのデフォルト値は、ベーシックモデル QCPU が 2048 点、ハイパフォーマンスモデル QCPU、プロセス CPU、二重化 CPU が 8192 点です。
- MELSECNET/H ネットワークユニット内のリンクレジスタは 16384 点ですが、CPU ユニット内のリンクレジスタのデフォルト値は、ベーシックモデル QCPU が 2048 点、ハイパフォーマンスモデル QCPU、プロセス CPU、二重化 CPU が 8192 点です。

上記に示した点数以降のリンクレジスタを使用する場合は、PC パラメータのデバイス設定でリンクレジスタ点数を変更またはファイルレジスタを使用してください。

(4) ネットワークシステムで使用する場合

ネットワークパラメータの設定が必要です。

ネットワークパラメータで設定されていない範囲は、データレジスタの代用として使用できます。

備考

ネットワークパラメータについては、下記マニュアルを参照してください。

☞ 各ネットワークユニットのマニュアル

9.2.14 リンク特殊レジスタ (SW)

(1) リンク特殊レジスタとは

CC-Link IE コントローラネットワークユニット, MELSECNET/H ネットワークユニットなどのインテリジェント機能ユニットの通信状態・異常内容を格納するレジスタです。

データリンク時の情報が数値で格納されるため, リンク特殊レジスタをモニタすることにより, 異常箇所および原因を調べることができます。

(2) リンク特殊レジスタ点数

表 9.6 CPU ユニットごとのリンク特殊レジスタの点数

CPU ユニット	リンク特殊レジスタ点数									
ベーシックモデル QCPU	CPU ユニットのリンク特殊レジスタの点数は 1024 点 (SW0 ~ 3FF) です。CC-Link IE コントローラネットワークユニット, MELSECNET/H ネットワークユニットなどの, リンク特殊レジスタを持つインテリジェント機能ユニットは 512 点になります。									
ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU	<p>CPU ユニットのリンク特殊レジスタの点数は 2048 点 (SW0 ~ 7FF) です。CC-Link IE コントローラネットワークユニット, MELSECNET/H ネットワークユニットなどの, リンク特殊レジスタを持つインテリジェント機能ユニットごとに 512 点。</p> <p>リンク特殊レジスタは, 下図のように割り付けることが可能です。</p> <div style="text-align: center;"> <p>リンク特殊レジスタ</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding: 2px;">SW0 ┆ SW1FF</td> <td style="padding: 2px;">1枚目の ネットワークユニット用</td> <td rowspan="4" style="padding: 2px;">} 512点</td> </tr> <tr> <td style="padding: 2px;">SW200 ┆ SW3FF</td> <td style="padding: 2px;">2枚目の ネットワークユニット用</td> </tr> <tr> <td style="padding: 2px;">SW400 ┆ SW5FF</td> <td style="padding: 2px;">3枚目の ネットワークユニット用</td> </tr> <tr> <td style="padding: 2px;">SW600 ┆ SW7FF</td> <td style="padding: 2px;">4枚目の ネットワークユニット用</td> </tr> </table> <p style="margin-left: 100px;">} 2048点</p> </div>	SW0 ┆ SW1FF	1枚目の ネットワークユニット用	} 512点	SW200 ┆ SW3FF	2枚目の ネットワークユニット用	SW400 ┆ SW5FF	3枚目の ネットワークユニット用	SW600 ┆ SW7FF	4枚目の ネットワークユニット用
SW0 ┆ SW1FF	1枚目の ネットワークユニット用	} 512点								
SW200 ┆ SW3FF	2枚目の ネットワークユニット用									
SW400 ┆ SW5FF	3枚目の ネットワークユニット用									
SW600 ┆ SW7FF	4枚目の ネットワークユニット用									

備考

使用できるリンク特殊レジスタの詳細については, リンク特殊レジスタを持つインテリジェント機能ユニットのマニュアルを参照してください。

9.3 内部システムデバイス

システム用のデバイスです。

内部システムデバイスの割付け／容量は固定で、変更はできません。

9.3.1 ファンクションデバイス (FX, FY, FD)

(1) ファンクションデバイスとは

ファンクションデバイスは、引数付きサブルーチンプログラムで使用するデバイスです。

ファンクションデバイスは引数付きサブルーチンプログラムとその呼び出し元の間でデータの書き込み／読出しを行います。

例 サブルーチンプログラムで FX0, FY1, FD2 を使用している場合に、サブルーチンプログラムコール命令で X0, M0, D0 を指定すると、X0 の ON/OFF データが FX0, FY1 の ON/OFF データが M0, D0 のデータが FD2 に引き渡されます。

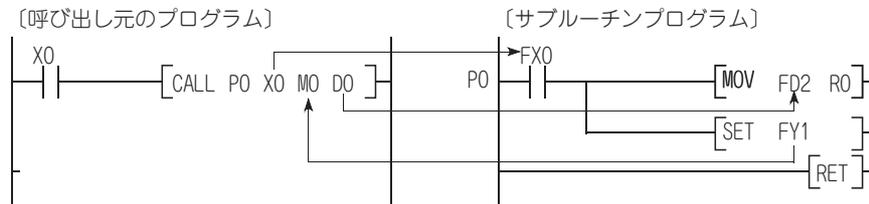


図 9.46 ファンクションデバイスの使用例

(2) ファンクションデバイスの用途

ファンクションデバイスをサブルーチンプログラムで使用すると、各呼び出し元で使用するデバイスを決めることができるため、同一サブルーチンプログラムを使用しても、ほかのサブルーチンプログラムの呼び出し元を意識することなく使用できます。

(3) ファンクションデバイスの種類

下記の 3 種類があります。

- ファンクション入力 (FX)
- ファンクション出力 (FY)
- ファンクションレジスタ (FD)

(a) ファンクション入力 (FX)

- ファンクション入力は、サブルーチンプログラムに ON/OFF データを引き渡す場合に使用します。
- サブルーチンプログラムでは、引数付きサブルーチンコール命令で指定されたビットデータを取り込み、演算に使用します。
- CPU ユニットのビットデータ指定デバイスがすべて使用できます。

(b) ファンクション出力 (FY)

- ファンクション出力は、サブルーチンプログラムでの演算結果（ON/OFF データ）を呼び出し元に引き渡す場合に使用します。
- 引数付きサブルーチンプログラムで指定されたデバイスに演算結果が格納されます。
- CPU ユニットの入力（X,DX）を除くビットデータ指定デバイスが使用できます。

(c) ファンクションレジスタ (FD)

- ファンクションレジスタは、サブルーチンプログラムとその呼び出し元の間でデータの書込み／読出しに使用します。
- ファンクションレジスタのインプット／アウトプット条件は、CPU ユニットが自動判別します。ソースデータの場合は、サブルーチンプログラムのインプットデータになります。ディスティネーションデータの場合は、サブルーチンプログラムからのアウトプットデータになります。
- ファンクションレジスタ 1 点で最大 4 ワードを占有します。ただし、使用するワード数は、サブルーチンプログラムでの命令により異なります。

① 1 ワードの命令の場合は、1 ワードのみ使用します。



図 9.47 ファンクションレジスタ 1 点で 1 ワード占有する場合

② 2 ワードの命令の場合は、2 ワードを使用します。



図 9.48 ファンクションレジスタ 1 点で 2 ワード占有する場合

③ 32 ビットの乗除算のディスティネーションの場合は、4 ワードを使用します。

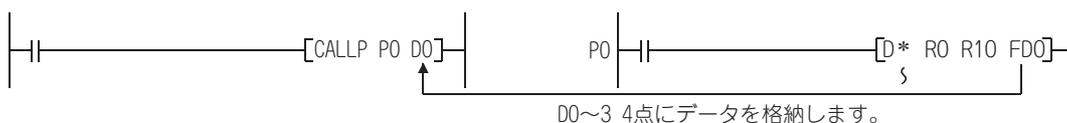


図 9.49 ファンクションレジスタ 1 点で 4 ワード占有する場合

- CPU ユニットのワードデータ指定デバイスが使用できます。

Point

引数付きサブルーチンプログラム内では、ファンクションレジスタが使用しているデバイスを使用しないでください。使用すると、呼び出し元にファンクションレジスタの値が正常に引き渡されなくなります。

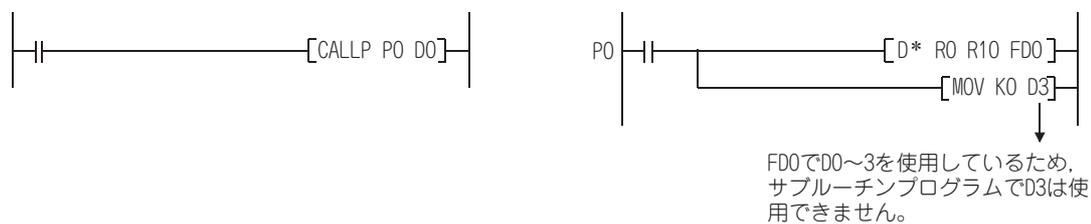


図 9.50 引数付きサブルーチンプログラムでデバイスを使用できない回路例

備考

ファンクションデバイスの使用方法については、下記マニュアルを参照してください。

☞ MELSEC-Q/L プログラミングマニュアル（共通命令編）

9.3.2 特殊リレー (SM)

(1) 特殊リレーとは

特殊リレーは、シーケンサ内部で仕様が決まっている内部リレーで、CPU ユニットの状態が格納されます。

(2) 使用可能な特殊リレーの分類

使用可能な特殊リレーの分類を表 9.7 に示します。

表 9.7 特殊リレーの分類一覧

分類	特殊リレー	CPU ユニット			
		ベーシックモデル QCPU	ハイパフォーマンス モデル QCPU	プロセス CPU	二重化 CPU
診断情報	SM0 ~ 99	○	○	○	○
	SM100 ~ 199	×	○	○	○
シリアルコミュニケーション機能	SM100 ~ 129	○	×	×	×
システム情報	SM200 ~ 399	○	○	○	○
システムクロック/ システムカウンタ	SM400 ~ 499	○	○	○	○
スキャン情報・I/O リフレッシュ	SM500 ~ 599	○	○	○	○
ドライブ情報	SM600 ~ 699	○	○	○	○
命令関連	SM700 ~ 799	○	○	○	○
デバッグ	SM800 ~ 899	×	○	○	○
ラッチエリア	SM900 ~ 999	×	○	○	○
A → Q/QnA 変換対応	SM1000 ~ 1255 * 1	×	○	○	×
プロセス制御命令	SM1500 ~ 1509	×	×	○	○
二重化対応 (自系 CPU 情報)	SM1510 ~ 1599	×	×	×	○
二重化対応 (他系 CPU 情報)	SM1600 ~ 1699	×	×	×	○
二重化対応 (トラッキング情報)	SM1700 ~ 1779	×	×	×	○
二重化電源ユニット情報	SM1780 ~ 1799	×	○	○	○

○：使用可能な特殊リレーが存在する， ×：使用可能な特殊リレーが存在しない

* 1：PC パラメータの PC システム設定で “SM1000, SD1000 以降の特殊リレー／特殊レジスタを使用する” に設定した場合のみ有効です。

備考

特殊リレーの詳細については、下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編)

9.3.3 特殊レジスタ (SD)

(1) 特殊レジスタとは

特殊レジスタは、シーケンサ内部で仕様が決まっている内部レジスタで、CPU ユニットの状態（故障診断・システム情報など）が格納されます。

(2) 使用可能な特殊レジスタの分類

使用可能な特殊レジスタの分類を表 9.8 に示します。

表 9.8 特殊レジスタの分類一覧

分類	特殊レジスタ	CPU ユニット			
		ベーシックモデル QCPU	ハイパフォーマンスモデル QCPU	プロセス CPU	二重化 CPU
診断情報	SD0 ~ 99	○	○	○	○
	SD100 ~ 199	×	○	○	○
シリアルコミュニケーション機能	SD100 ~ 129	○	×	×	×
ヒューズ断ユニット	SD130 ~ 149	○	×	×	×
	SD1300 ~ 1399	×	○	○	○
入出力ユニット照合	SD150 ~ 199	○	×	×	×
	SD1400 ~ 1499	×	○	○	○
システム情報	SD200 ~ 399	○	○	○	○
システムクロック/ システムカウンタ	SD400 ~ 499	○	○	○	○
スキャン情報	SD500 ~ 599	○	○	○	○
ドライブ情報	SD600 ~ 699	○	○	○	○
命令関連	SD700 ~ 799	○	○	○	○
デバッグ	SD800 ~ 899	×	○	×	×
ラッチエリア	SD900 ~ 929	×	○	×	×
二重化 CPU 情報（停電保持情報）	SD952	×	×	×	○
A → Q/QnA 変換対応	SD1000 ~ 1255 * 1	×	○	○	×
プロセス制御命令	SD1500 ~ 1509	×	×	○	○
二重化対応（自系 CPU 情報）	SD1510 ~ 1599	×	×	×	○
二重化対応（他系 CPU 情報）	SD1600 ~ 1699	×	×	×	○
二重化対応（トラッキング情報）	SD1700 ~ 1779	×	×	×	○
二重化電源ユニット情報	SD1780 ~ 1799	×	○	○	○

○：使用可能な特殊レジスタが存在する、×：使用可能な特殊レジスタが存在しない

* 1：PC パラメータの PC システム設定で“SM1000, SD1000 以降の特殊リレー／特殊レジスタを使用する”に設定した場合のみ有効です。

備考

特殊レジスタの詳細については、下記マニュアルを参照してください。
 QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）

9.4 リンクダイレクトデバイス

(1) リンクダイレクトデバイスとは

CC-Link IE コントローラネットワークまたは MELSECNET /H のネットワークユニット内のリンクデバイスを直接アクセスするデバイスです。

CPU ユニットのリンクリフレッシュに関係なく、シーケンスプログラムにより CC-Link IE コントローラネットワークまたは MELSECNET/H のネットワークユニット内のリンクデバイスを直接読み書きできます。

(2) 指定方法と使用例

(a) 指定方法

ネットワーク No. とデバイス番号で指定します。

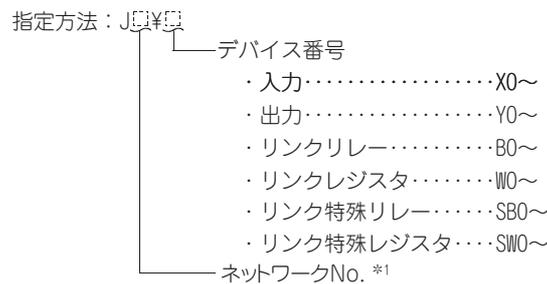


図 9.51 指定方法

* 1 : ベーシックモデル QCPU … No.1 ~ 239
 ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU … No.1 ~ 255

(b) 使用例

ネットワーク No.2 のリンクレジスタ 10(W10) の場合は, “J2¥W10” になります。

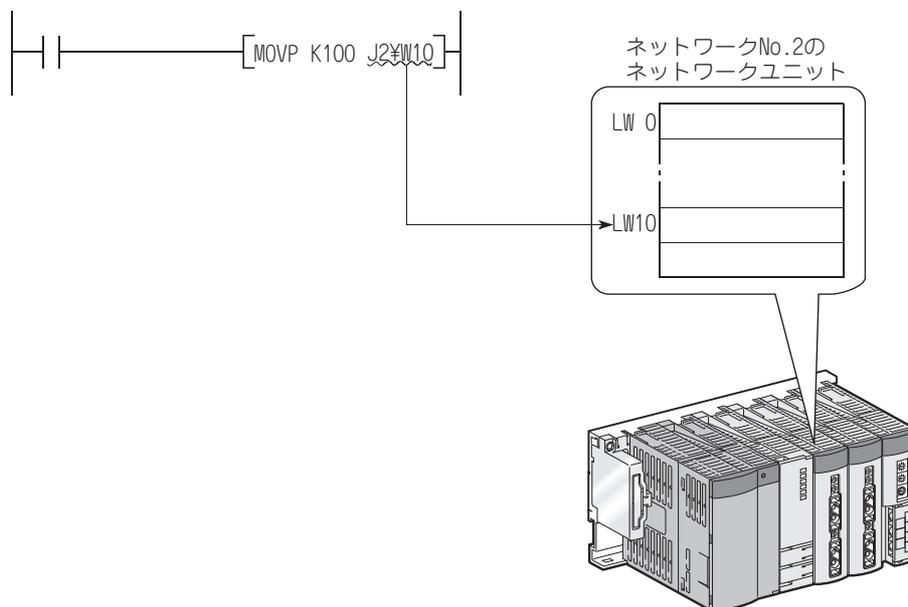


図 9.52 使用例

ビットデバイス (X, Y, B, SB) の場合, 桁指定を行います。

例 J1¥K1X0, J10¥K4B0

9.4 リンクダイレクトデバイス

(3) 指定範囲

ネットワークリフレッシュパラメータで設定していない範囲のリンクデバイスも指定できます。

(a) 書込みの場合

- ネットワークパラメータの共通パラメータで送信範囲に設定したリンクデバイス範囲内で、ネットワークリフレッシュパラメータのリフレッシュ範囲外に書込みを行ってください。

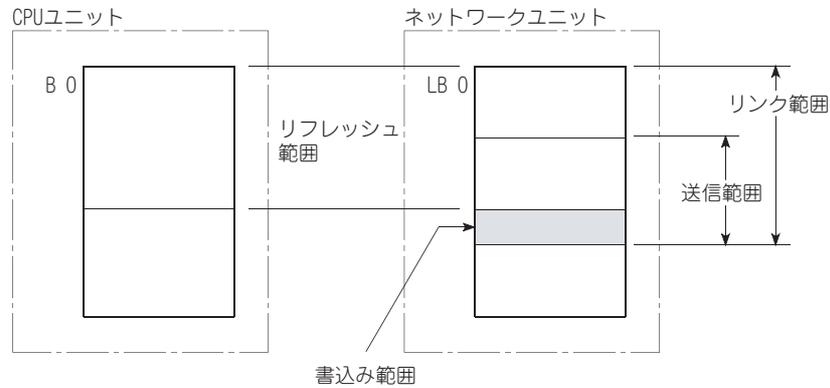


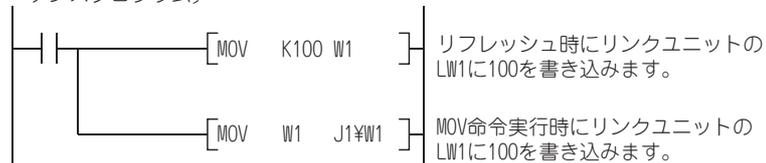
図 9.53 リンクダイレクトデバイスの書込み範囲

- リフレッシュパラメータでリフレッシュ範囲に設定したリンクデバイス範囲にも書込みはできますが、リフレッシュ時にリンクユニットのリンクデバイスのデータが書き換わります。リンクダイレクトデバイスによる書込み時には、リフレッシュパラメータで設定されている CPU ユニット側の該当デバイスにも同一データを書き込んでください。

【リフレッシュパラメータの設定】

- ① ネットワーク No. : 1
- ② CPU ユニット (W0 ~ 3F) ↔ ネットワークユニット (LW0 ~ 3F)

〔シーケンスプログラム〕



〔書込みタイミング〕

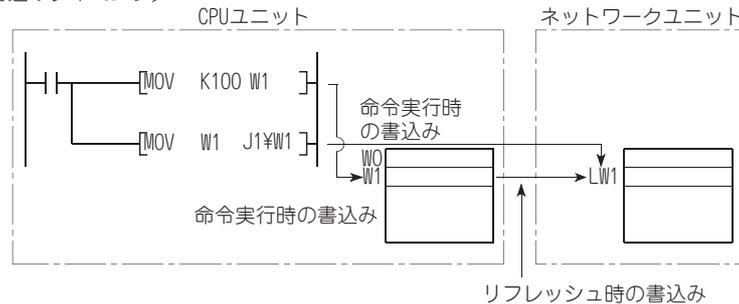


図 9.54 リフレッシュ範囲に設定したリンクデバイスへの書込み

- 他局の書込み範囲にリンクダイレクトデバイスによる書込みを行った場合は、他局からのデータ受信時に、受信したデータに書き換わります。

(b) 読出しの場合

ネットワークユニットのリンクデバイス範囲の読出しができます。

Point

リンクダイレクトデバイスで書込み／読出しのできるネットワークユニットは、1つのネットワーク No. で1台のみです。

同一ネットワーク No. で2台以上のネットワークユニットを装着している場合は、スロット No. の若いネットワークユニットがリンクダイレクトデバイスによる書込み／読出しの対象になります。

注 9.4

例えば、ネットワーク No.1 で局番1と局番2のネットワークユニットを図9.55のように装着している場合は、局番2のネットワークユニットがリンクダイレクトデバイスの対象になります。

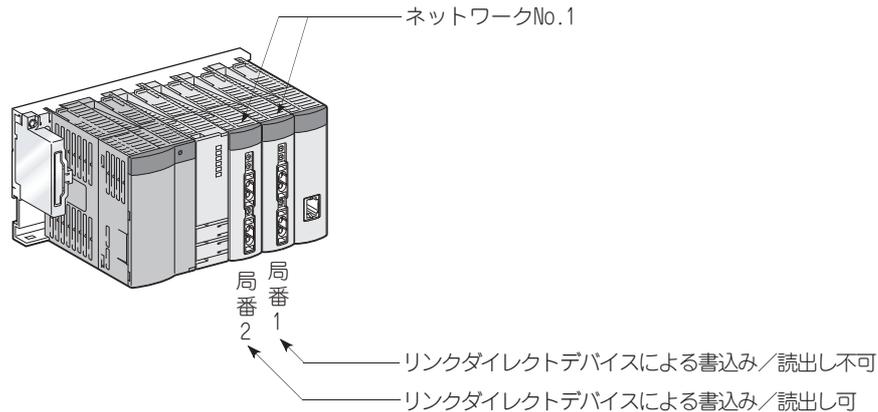


図9.55 同一ネットワーク No. で2台以上のネットワークユニットを装着している場合

注 9.4 Basic

ベーシックモデル QCPU では、同一ネットワーク No. で2台以上のネットワークユニットは装着できません。

(4) リンクダイレクトデバイスとリンクリフレッシュの相違点

表 9.9 リンクダイレクトデバイスとリンクリフレッシュの相違点一覧表

項 目		リンクダイレクトデバイス	リンクリフレッシュ
プログラムでの 表記方法	リンクリレー	J□¥K4B0～	B0～
	リンクレジスタ	J□¥W0～	W0～
	リンク特殊リレー	J□¥K4SB0～	SB0～
	リンク特殊レジスタ	J□¥SW0～	SW0～
ステップ数		2 ステップ	1 ステップ
ネットワークユニットとのアクセス範囲		J□¥□0～3FFF	リフレッシュパラメータで設定した範囲
アクセスデータの保証範囲		ワード（16ビット）単位	

備 考

ネットワークパラメータ、共通パラメータ、ネットワークリフレッシュパラメータについては、下記マニュアルを参照してください。

- 詳細説明：
 - ☞ 各ネットワークユニットのネットワークマニュアル
- 設定方法：
 - ☞ GX Developer Version 8 オペレーティングマニュアル

9.5 ユニットアクセスデバイス

9.5.1 インテリジェント機能ユニットデバイス

(1) インテリジェント機能ユニットデバイスとは

基本ベースユニットおよび増設ベースユニットに装着されているインテリジェント機能ユニットのバッファメモリに、CPU ユニットから直接アクセスするデバイスです。

(2) 指定方法と使用例

(a) 指定方法

インテリジェント機能ユニットの入出力番号とバッファメモリアドレスで指定します。

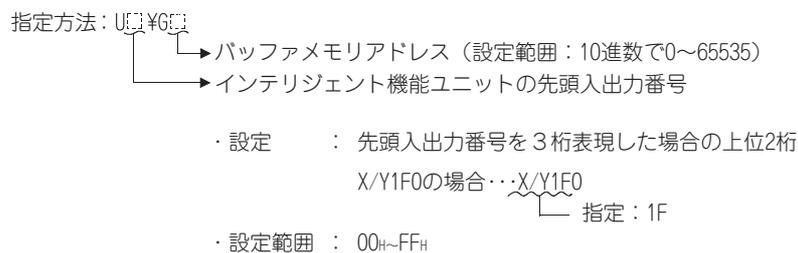


図 9.56 指定方法

(b) 使用例

入出力番号 020 に装着されている Q64AD 形アナログ/デジタル変換ユニット (X/Y020 ~ 02F) の CH.1 ~ 4 のデジタル出力値を、D0 ~ 3 に格納する場合は図 9.57 のように指定します。

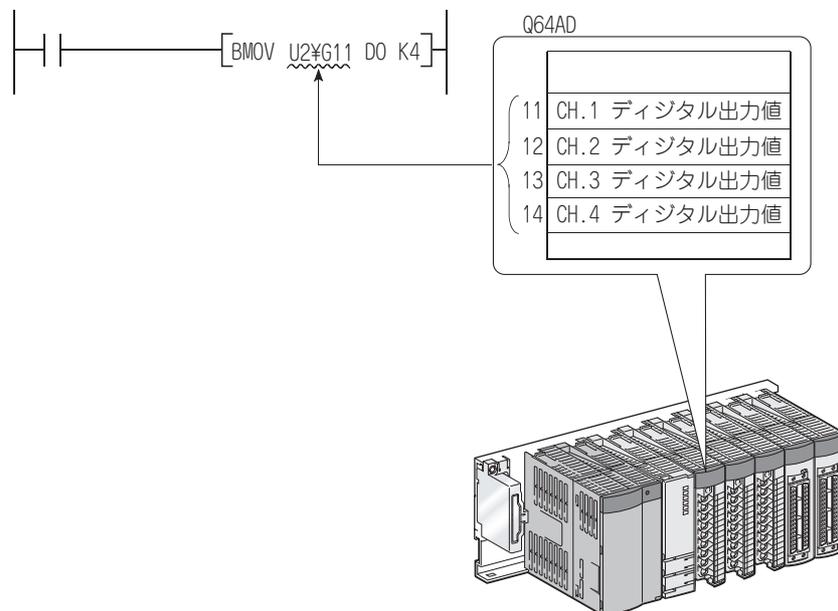


図 9.57 使用例

Point

インテリジェント機能ユニットデバイスを使用すると、バッファメモリに対してデバイスコメントを付けることができます。

☞ GX Developer Version 8 オペレーティングマニュアル

(3) 処理速度

インテリジェント機能ユニットデバイスによる処理速度は次のようになります。

- インテリジェント機能ユニットデバイスによる書込み／読出しは、FROM/TO 命令での書込み／読出しより処理速度が多少高速になります。

例 “MOV U2¥G11 D0” の場合

- インテリジェント機能ユニット  **注9.5** のバッファメモリからの読出しと別の処理を1つの命令で行う場合は、FROM/TO 命令での処理速度と命令の処理速度の合計値を目安にしてください。

例 “+ U2¥G11 D0 D10” の場合



注 9.5

Basic

Process

Redundant

ベーシックモデル QCPU, プロセス CPU, 二重化 CPU では、AnS/A シリーズ対応の特殊機能ユニットが使用できません。

Point

シーケンスプログラム内でインテリジェント機能ユニットデバイスを2回以上使用し、バッファメモリのデータを書込み／読出しする場合は、FROM/TO 命令を使用してプログラムの1箇所での書き込み／読出しを行うようにすると、処理速度を早くできます。

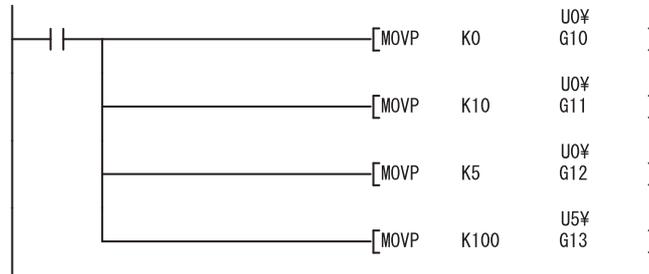


図 9.58 インテリジェント機能ユニットデバイスを複数使用して書き込む場合

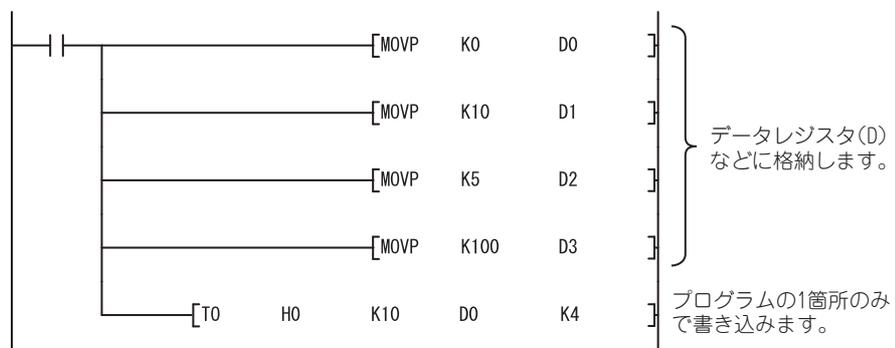


図 9.59 TO 命令を使用してプログラムの1箇所での書き込む場合

備考

- 1) バッファメモリのアドレス、用途については、使用するインテリジェント機能ユニットのマニュアルを参照してください。
- 2) FROM/TO 命令については、下記マニュアルを参照してください。
 MELSEC-Q/L プログラミングマニュアル（共通命令編）

9.5.2 マルチ CPU 間共有デバイス 注 9.6

(1) マルチ CPU 間共有デバイスとは

マルチ CPU システムの各 CPU ユニットの CPU 共有メモリにアクセスするデバイスです。

(2) マルチ CPU 間共有デバイスの特長

- CPU 共有メモリへの書込み (S.TO/TO 命令) / 読出し命令 (FROM 命令) よりも高速で転送でき、プログラムのステップ数を少なくできます。* 1
- マルチ CPU 間共有デバイスを使用することにより、ビット操作が可能です。
- マルチ CPU 間共有デバイスに対しデバイスコメントを設定することにより、プログラムの可読性が向上します。
- CPU 共有メモリ上の情報を命令の引数として直接指定できるため、インタロック用のデバイスなどが不要になります。

* 1 : ハイパフォーマンスモデル QCPU, プロセス CPU では、マルチ CPU 間共有デバイス (U3En¥G□) を使用した命令で自号機の CPU 共有メモリにデータを書き込むことはできません。

(3) 指定方法

マルチ CPU 間共有デバイスは、CPU ユニットの入出力番号と CPU 共有メモリのアドレスで指定します。

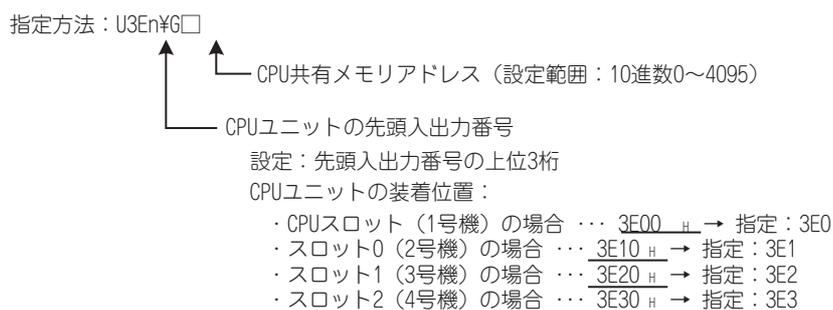


図 9.60 指定方法

備考

マルチ CPU 間共有デバイスの詳細については、下記マニュアルを参照してください。
 QCPU ユーザーズマニュアル (マルチ CPU システム編)

注 9.6

Redundant

二重化 CPU では、マルチ CPU 間共有デバイスは使用できません。

9.6 インデックスレジスタ (Z)

9.6.1 インデックスレジスタ (Z)

(1) インデックスレジスタとは

シーケンスプログラムで使用するデバイスの間接設定（インデックス修飾）に使用するデバイスです。インデックス修飾は、インデックスレジスタ 1 点を使用します。

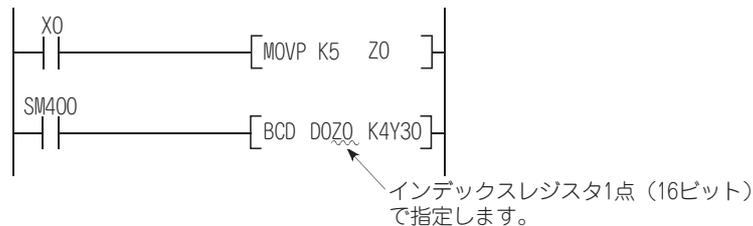


図 9.61 インデックスレジスタ

インデックスレジスタの点数は下記のとおりです。

- ベーシックモデル QCPU : Z0 ~ 9 (10 点)
- ハイパフォーマンス QCPU, プロセス CPU, 二重化 CPU : Z0 ~ 15 (16 点)

(2) インデックスレジスタのビット構成

(a) ビット構成と書込みおよび読出し単位

1 点 16 ビットで構成され、16 ビット単位で書込みおよび読出しができます。



図 9.62 インデックスレジスタのビット構成

Point

インデックスレジスタは符号付きで扱います。

HEX (16 進数) の場合は 0000H ~ FFFFH を格納可能ですが、最上位のビットが符号ビットとなるため、指定可能な数値の範囲は -32768 ~ 32767 となります。

(b) 32ビット命令でインデックスレジスタを使用する場合

Zn と Zn + 1 が処理対象になります。

指定しているインデックスレジスタ番号 (Zn) が下位 16ビット、指定しているインデックスレジスタ番号 + 1 のインデックスレジスタが上位 16ビットになります。

例 DMOV 命令で Z2 を指定した場合は、Z2 が下位 16ビット、Z3 が上位 16ビットになります。

(32ビット構成時の最上位ビットは、符号ビットになります。)

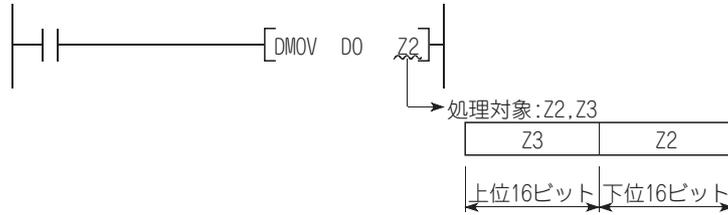


図 9.63 32ビット命令によるデータの転送と格納先

備考

インデックスレジスタを使用したインデックス修飾の詳細内容および注意事項については、下記マニュアルを参照してください。

 MELSEC-Q/L プログラミングマニュアル (共通命令編)

9.6.2 スキャン実行タイププログラムと低速実行タイププログラムの切換え時の処理 注 9.7

CPU ユニットは、スキャン実行タイププログラムと低速実行タイププログラムの切換え時に、インデックスレジスタの内容の退避（保護）と復帰を行っています。

(1) スキャン実行タイププログラムから低速実行タイププログラムへの切換え時

スキャン実行タイププログラムのインデックスレジスタの値を退避し、低速実行タイププログラムのインデックスレジスタの値を復帰させます。

(2) 低速実行タイププログラムからスキャン実行タイププログラムへの切換え時

低速実行タイププログラムのインデックスレジスタの値を退避し、スキャン実行タイププログラムのインデックスレジスタの値を復帰させます

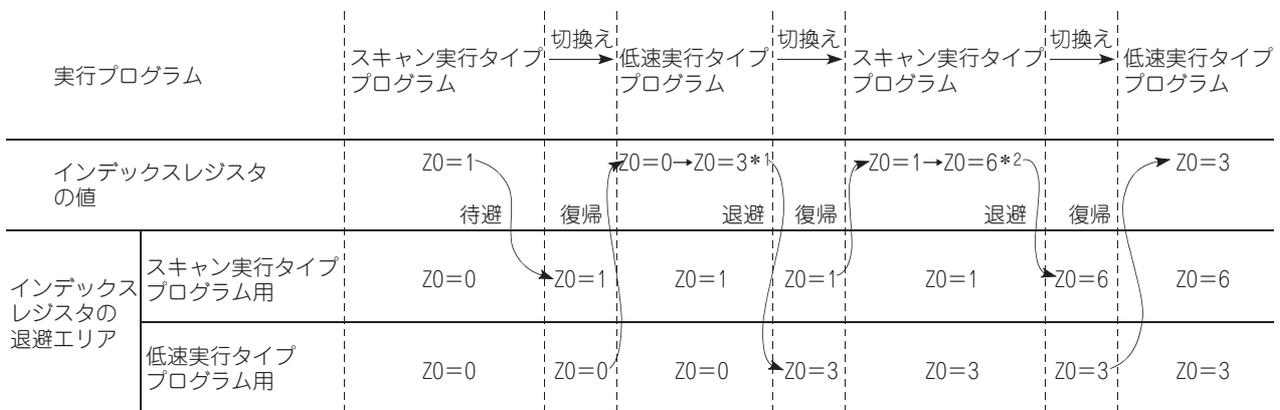


図 9.64 スキャン / 低速実行タイププログラム切換え時のインデックスレジスタの退避と復帰

- * 1：低速実行タイププログラムで Z0 を 3 に変更します。
- * 2：スキャン実行タイププログラムで Z0 を 6 に変更します。

(3) インデックスレジスタの引渡し

スキャン実行タイププログラムと低速実行タイププログラム間でインデックスレジスタの引渡しを行う場合は、ワードデバイスを使用してください。

9.6 インデックスレジスタ(Z)
9.6.2 スキャン実行タイププログラムと低速実行タイププログラムの切換え時の処理

 注 9.7 **Basic** **Redundant**

ベーシックモデル QCPU、二重化 CPU では、低速実行タイププログラムは使用できないため、低速実行タイププログラムに切り換えることができません。

9.6.3 スキャン実行タイプ／低速実行タイププログラム→割込み／定周期実行タイププログラムへの切換え時の処理 注 9.8

CPU ユニットは、スキャン実行タイププログラムから割込み／定周期実行タイププログラムへの切換え時に下記の処理を行っています。

- ・インデックスレジスタの内容を退避（保護）・復帰
- ・ファイルレジスタのブロック No. の退避（保護）・復帰

(1) 退避（保護）・復帰の設定

PC パラメータの PC システム設定で退避（保護）・復帰するかどうかの設定ができます。

割込み／定周期実行タイププログラムでインデックスレジスタへの書き込みを行わない場合は、“割込プログラム／定周期プログラム設定”を“高速実行する”に設定します。

この設定により、プログラムの切換えが早くなります。

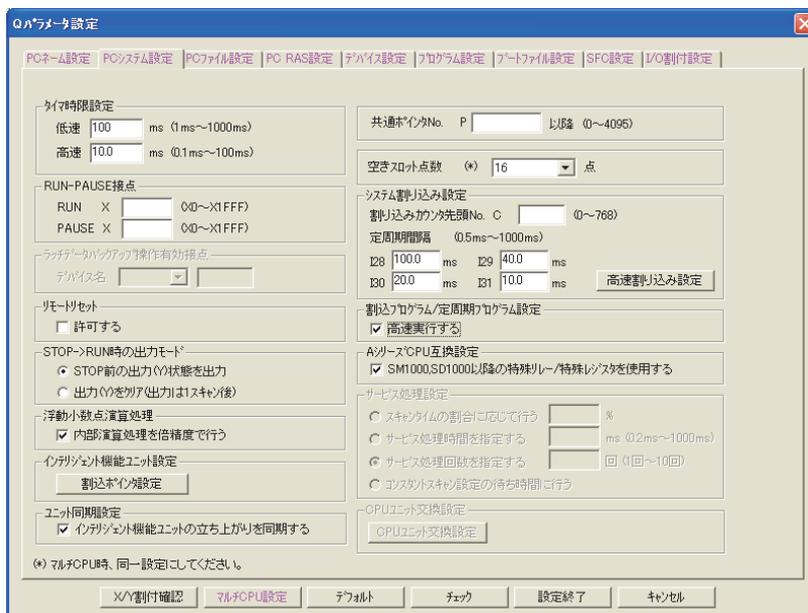


図 9.65 PC システム設定

注 9.8

Basic Redundant

ベーシックモデル QCPU では、低速実行タイププログラムおよび定周期実行タイププログラムは使用できません。
“スキャン実行タイプ／低速実行タイププログラム”は“メインルーチン／サブルーチンプログラム”，“割込み／定周期実行タイププログラム”は“割込みプログラム”に置き換えて読んでください。
二重化 CPU では、低速実行タイププログラムは使用できません。
“スキャン実行タイプ／低速実行タイププログラム”は“スキャン実行タイププログラム”に置き換えて読んでください。

(2) インデックスレジスタの処理

(a) “高速実行する” を選択しない場合

- 1) スキャン実行タイププログラムから割込み／定周期実行タイププログラムへの切換え時
 スキャン実行タイププログラムのインデックスレジスタの値を退避し、割込み／定周期実行タイププログラムに引き渡します。
- 2) 割込み／定周期実行タイププログラムからスキャン実行タイププログラムへの切換え時
 退避したインデックスレジスタの値を復帰します。

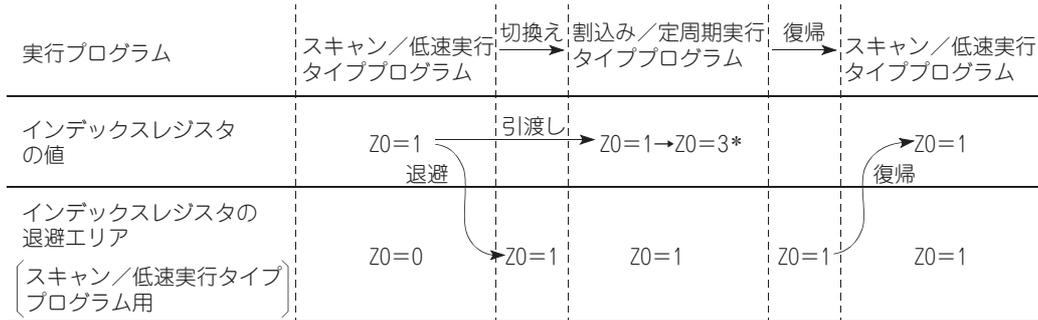


図 9.66 インデックスレジスタの退避(保護)・復帰 (“高速実行する” を選択しない場合)

* 1: 割込みプログラムで Z0 を 3 に変更します。

Point

割込み／定周期実行タイププログラムからスキャン実行タイププログラムへのインデックスレジスタの値の引渡しを行う場合は、ワードデバイスを使用してください。

9.6 インデックスレジスタ(Z)
 9.6.3 スキャン実行タイプ／低速実行タイププログラム→割込み／定周期実行タイププログラムへの切換え時の処理

(b) “高速実行する” を選択した場合

1) スキャン実行タイプ／低速実行タイププログラムから割り込み／定周期実行タイププログラムへの切換え時

インデックスレジスタの退避・復帰をしません。

2) 割り込み／定周期実行タイププログラムからスキャン実行タイプ／低速実行タイププログラムへの切換え時

割り込み／定周期実行タイププログラムでインデックスレジスタにデータの書き込みを行うと、スキャン実行タイプ／低速実行タイププログラムで使用していたインデックスレジスタの値が壊されます。

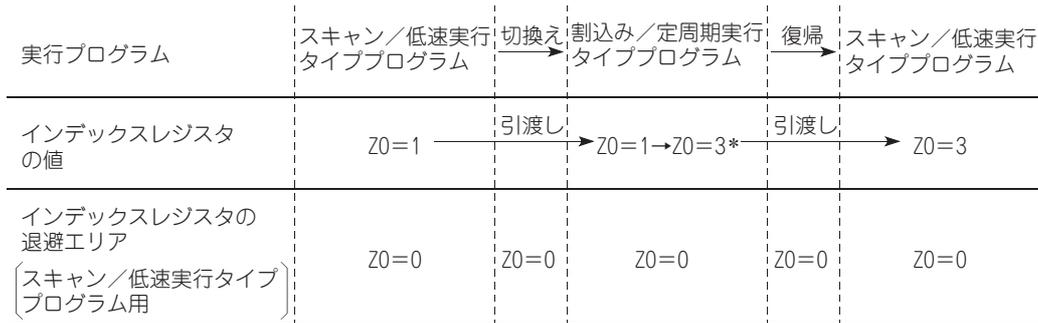


図 9.67 インデックスレジスタの退避 (保護) ・復帰 (“高速実行する” を選択した場合)

* 1: 割り込みプログラムで Z0 を 3 に変更します。

割り込み／定周期実行タイププログラムでインデックスレジスタにデータを書き込む場合は、ZPUSH 命令／ZPOP 命令で、インデックスレジスタの退避・復帰を行ってください。

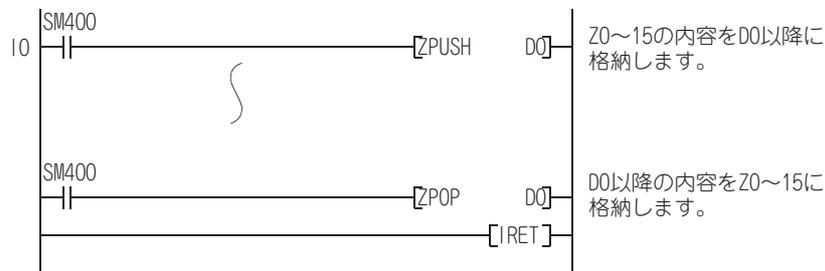


図 9.68 割り込み／定周期実行タイププログラムでインデックスレジスタにデータを書き込む場合

(3) ファイルレジスタのブロックNo.の処理

(a) スキャン実行タイプ／低速実行タイププログラムから割込み／定周期実行タイププログラムに切換え時

スキャン実行タイプ／低速実行タイププログラムでのファイルレジスタのブロック No. を退避し、割込み／定周期実行タイププログラムに引き渡します。

(b) 割込み／定周期実行タイププログラムからスキャン実行タイプ／低速実行タイププログラムへの切換え時

退避したファイルレジスタのブロック No. を復帰します。

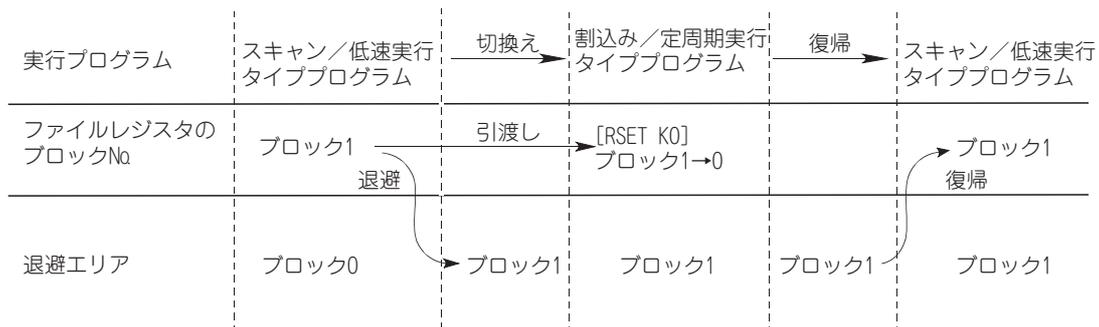


図 9.69 ファイルレジスタのブロックNo.の退避 (保護) ・復帰

9.6 インデックスレジスタ (Z)
9.6.3 スキャン実行タイプ／低速実行タイププログラム→割込み／定周期実行タイププログラムへの切換え時の処理

9.7 ファイルレジスタ (R) 注 9.9

(1) ファイルレジスタとは

データレジスタの拡張用のデバイスです。
データレジスタと同一の処理速度で使用できます。

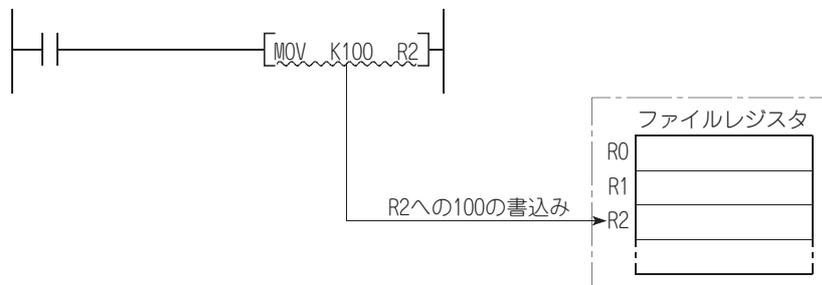


図 9.70 ファイルレジスタへの書き込み

(2) ファイルレジスタのビット構成

(a) ビット構成と書き込みおよび読出し単位

ファイルレジスタは 1 点 16 ビットで構成され、16 ビット単位で書き込みおよび読出しができます。

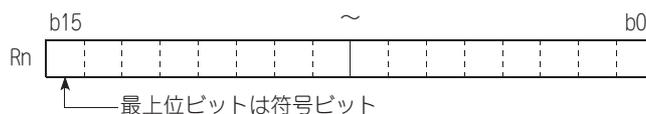


図 9.71 ファイルレジスタのビット構成

(b) 32 ビット命令でファイルレジスタを使用する場合

R_n と R_{n+1} が処理対象になります。

シーケンスプログラムで指定しているファイルレジスタ番号 (R_n) が下位 16 ビット、シーケンスプログラムで指定しているファイルレジスタ番号 + 1 のファイルレジスタが上位 16 ビットになります。

例えば、DMOV 命令で、 R_2 を指定した場合は、 R_2 が下位 16 ビット、 R_3 が上位 16 ビットになります。

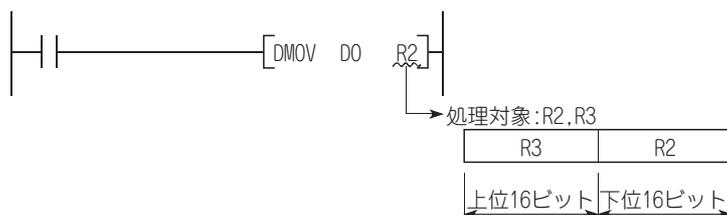


図 9.72 32 ビット命令によるデータの転送と格納先

ファイルレジスタ 2 点では、-2147483648 ~ 2147483647 または 0H ~ FFFFFFFFH のデータを格納できます。(32 ビット構成時の最上位ビットは符号ビットになります。)



注 9.9

Basic

Q00JCPU には、ファイルレジスタはありません。

(3) ファイルレジスタのクリア

ファイルレジスタの内容は、CPU ユニットの電源 OFF、リセット操作を行っても保持されます。
(ラッチクリアを行っても“クリア”できません。)

ファイルレジスタの内容を“クリア”する場合は、シーケンスプログラムまたは GX Developer によるデータクリア操作で行ってください。

(a) シーケンスプログラムでクリアする場合

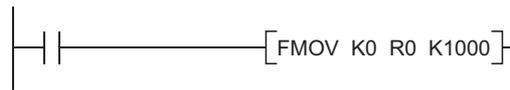


図 9.73 ファイルレジスタ R0 ~ 999 をクリアする例

(b) GX Developer でクリアする場合

GX Developer の [オンライン] → [PC メモリクリア] で “ファイルレジスタオールクリア” を選択してデータクリアしてください。

9.7.1 ファイルレジスタの格納場所

ファイルレジスタを格納するメモリを下記に示します。

- ・ベーシックモデル QCPU :
標準 RAM
- ・ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU :
標準 RAM, SRAM カード, Flash カード

9.7.2 ファイルレジスタの容量

(1) 標準 RAM 使用時

標準 RAM には、最大で下記点数のファイルレジスタが格納できます。

ただし、標準 RAM をファイルレジスタ以外の用途で使用する場合、その分使用できる点数は少なくなります。

(5.1.1 項, 5.2.1 項)

表 9.10 ファイルレジスタの容量

CPU ユニット		点 数
ベーシックモデル QCPU	Q00CPU, Q01CPU	64k 点
	Q02CPU	32k 点
ハイパフォーマンスモデル QCPU * 1	Q02HCPU, Q06HCPU	64k 点
	Q12HCPU, Q25HCPU	128k 点
プロセス CPU	Q02PHCPU, Q06PHCPU	64k 点
	Q12PHCPU, Q25PHCPU	128k 点
二重化 CPU	Q12PRHCPU, Q25PRHCPU	128k 点

* 1 : CPU ユニットのバージョンにより標準 RAM の容量が異なるため、使用できる点数が異なります。(付 2.2)

(2) SRAM カード使用時

- 1つのファイルは最大 1018k 点まで格納できます。
- 1ブロック 32k ワード単位のため最大 32 ブロックまで格納できます。

ただし、拡張できる点数、ブロック数は、メモ리카ードに格納されているプログラム、デバイスコメントなどの容量により異なります。

(3) Flash カード使用時

- 1つのファイルは最大 1018k 点まで格納できます。
- 1ブロック 32k ワード単位のため最大 32 ブロックまで格納できます。

ただし、拡張できる点数、ブロック数は、使用するメモ리카ードの容量とメモ리카ードに格納されているプログラム、デバイスコメントなどの容量により異なります。

備 考

.....
CPU ユニットで使用できるメモ리카ードについては、5.2.1 項を参照してください。
.....

9.7.3 格納先メモリによるアクセス方法の違い

ファイルレジスタのアクセス方法は、各メモリにより違いがあります。

表 9.11 メモリによるファイルレジスタのアクセス方法の違い

アクセス方法		標準 RAM	SRAM カード	Flash カード
プログラムの書込み		○	○	×
プログラムの読出し		○	○	○
デバイスメモリの PC 書込み		○	○	×
デバイスメモリの PC 読出し		○	○	○
データの 変更方法	GX Developer のオンラインによるテスト操作	○	○	×
	GX Developer による PC 書込み	○	○	×
	GX Developer による PC 書込み (フラッシュ ROM)	×	×	○
	シリアルコミュニケーションユニットによる一括書込み	○	○	×
	GOT1000 シリーズからのデバイス書込み	○	○	×
	GOT1000 シリーズからのランダム書込みコマンド	○	○	×

9.7.4 ファイルレジスタの登録手順 注 9.10

ファイルレジスタを使用する場合は、下記手順で CPU ユニットにファイルレジスタのファイルを登録します。

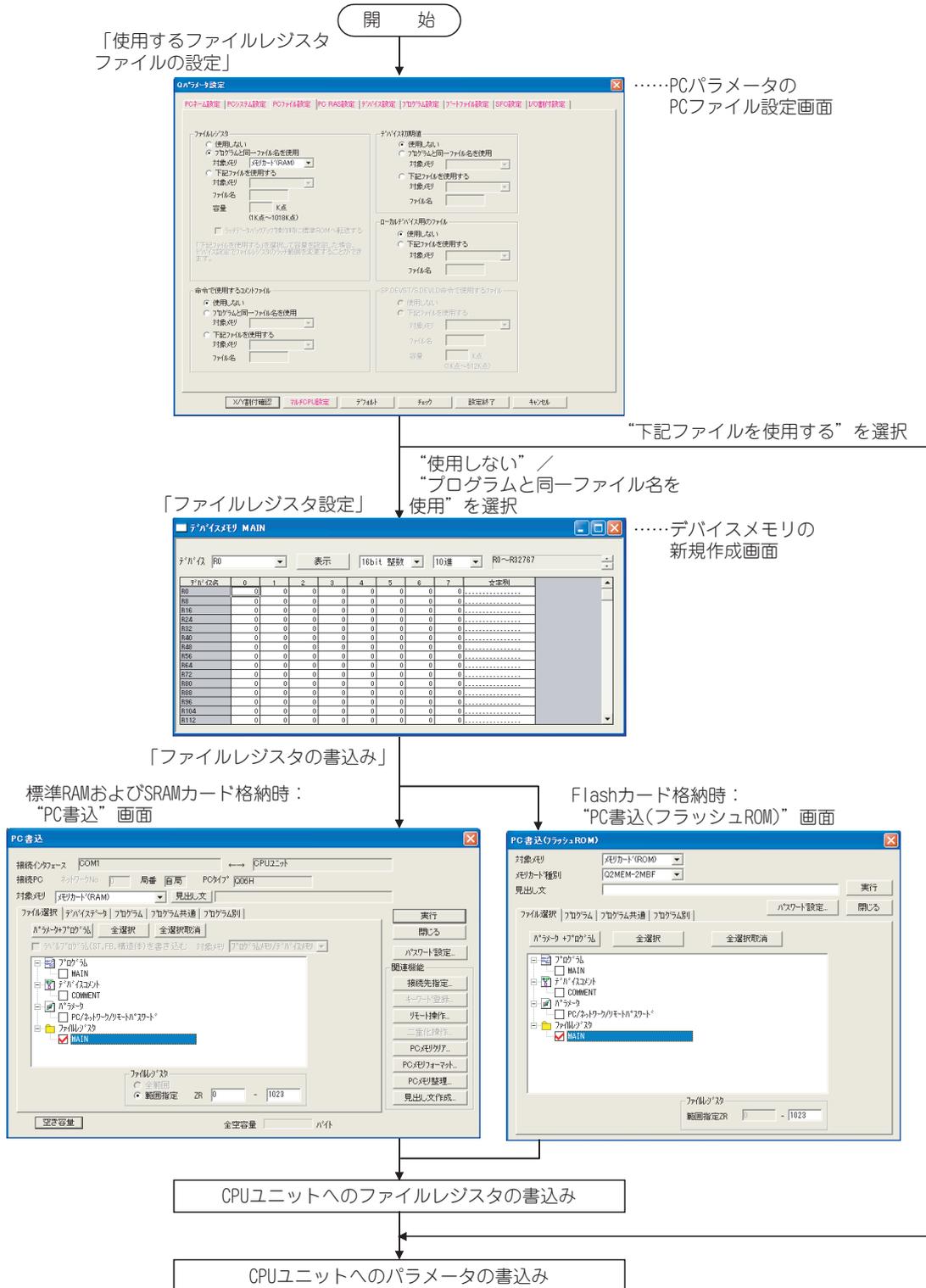


図 9.74 ファイルレジスタの登録手順

9.7 ファイルレジスタ (R)
9.7.4 ファイルレジスタの登録手順

注 9.10 Basic

ベーシックモデル QCPU では、標準 RAM に自動で登録されるためファイルレジスタの登録をする必要はありません。

(1) 使用ファイルレジスタの設定

シーケンスプログラムで標準 RAM またはメモリカード内のどのファイルレジスタを使用するかを、PC パラメータの PC ファイル設定で設定をします。



図 9.75 ファイルレジスタの設定

(a) 使用しない

次の場合に設定します。

- ファイルレジスタを使用しないとき
- シーケンスプログラムで使用するファイルレジスタを指定するとき（使用するファイルレジスタは QDRSET 命令で指定します。）

(b) プログラムと同一ファイル名を使用

シーケンスプログラムと同一ファイル名のファイルレジスタを実行する場合に設定します。

1) プログラムが切り換わった場合

自動でファイルレジスタもプログラムと同一ファイル名に切り換わります。

ファイルレジスタを実行しているプログラムのみで使用するローカルデバイスとして使用する場合に便利です。

例 プログラム A～C と同一ファイル名のファイルレジスタ A～C がある場合は、下記のようになります。

- プログラム A 実行時：ファイルレジスタ A をアクセス
- プログラム B 実行時：ファイルレジスタ B をアクセス
- プログラム C 実行時：ファイルレジスタ C をアクセス

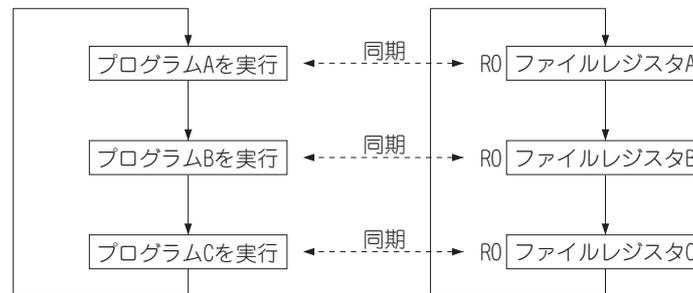


図 9.76 プログラムが切り換わった場合のファイルレジスタ

2) ファイルレジスタ点数の設定

使用できるファイルレジスタ点数は、GX Developer の [オンライン] → [PC 書込] で設定します。

Point

- 標準 RAM に作成できるファイルレジスタは、1 ファイルのみです。
複数ファイル作成したい場合は、SRAM カード、Flash カードを使用してください。
- 命令によっては、プログラムごとに設定したファイルレジスタを指定できません。
詳細については、下記マニュアルの各命令の使用可能デバイスを参照してください。
☞ MELSEC-Q/L プログラミングマニュアル（共通命令編）

(c) 下記ファイルを使用する

1 つのファイルレジスタを、実行する全プログラムで共通に使用する場合に設定します。

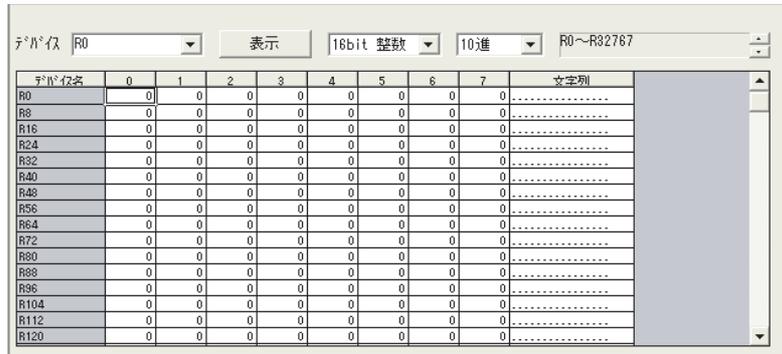
ファイルレジスタで使用する“対象メモリ”、“ファイル名”、“容量”を設定し、CPU ユニットに書き込むと、パラメータで指定されたファイルレジスタのファイルを作成します。

容量を設定しない場合は、次のようになります。

- 指定ドライブに格納されている指定ファイル名のファイルレジスタファイルを使用します。（容量は、格納されているファイルレジスタファイルの容量になります。）
- 指定ドライブに指定ファイル名のファイルレジスタファイルが存在しない場合は、“PARAMETER ERROR”（エラーコード：3002）になります。
- ATA カードを使用する場合、対象メモリにメモリカード (ROM) は設定できません。（ATA カードにファイルレジスタは格納できません。）
対象メモリにメモリカード (ROM) を設定し、CPU ユニットに書き込むと “PARAMETER ERROR”（エラーコード：3002）になります。

(2) ファイルレジスタ設定

デバイスメモリの新規作成で指定ファイル名のファイルレジスタの設定を行います。



デバイス名	0	1	2	3	4	5	6	7	文字列
R0	0	0	0	0	0	0	0	0
R8	0	0	0	0	0	0	0	0
R16	0	0	0	0	0	0	0	0
R24	0	0	0	0	0	0	0	0
R32	0	0	0	0	0	0	0	0
R40	0	0	0	0	0	0	0	0
R48	0	0	0	0	0	0	0	0
R56	0	0	0	0	0	0	0	0
R64	0	0	0	0	0	0	0	0
R72	0	0	0	0	0	0	0	0
R80	0	0	0	0	0	0	0	0
R88	0	0	0	0	0	0	0	0
R96	0	0	0	0	0	0	0	0
R104	0	0	0	0	0	0	0	0
R112	0	0	0	0	0	0	0	0
R120	0	0	0	0	0	0	0	0

図 9.77 デバイスメモリ画面

(a) デバイス

Rn（上記の場合は R0）を設定し、表示 ボタンをクリックすると、ファイルレジスタの一覧が表示されます。

(b) データの設定

ファイルレジスタに設定するデータを作成します。

ファイルレジスタの容量のみを設定する場合は、データの作成は不要です。

(3) ファイルレジスタファイルの CPU ユニットへの登録

PC パラメータの PC ファイル設定で、下記を選択した場合は、CPU ユニットへの登録の前に、ファイルレジスタ設定（デバイスメモリの作成）が必要です。

- 使用しない（シーケンスプログラムで指定するとき）（ 本項(1)(a)）
- プログラムと同一ファイル名を使用（ 本項(1)(b)）

CPU ユニットへの登録は、GX Developer の下記の操作で行います。

(a) 標準 RAM および SRAM カード格納時

[オンライン] → [PC 書込] で “PC 書込” 画面で行います。

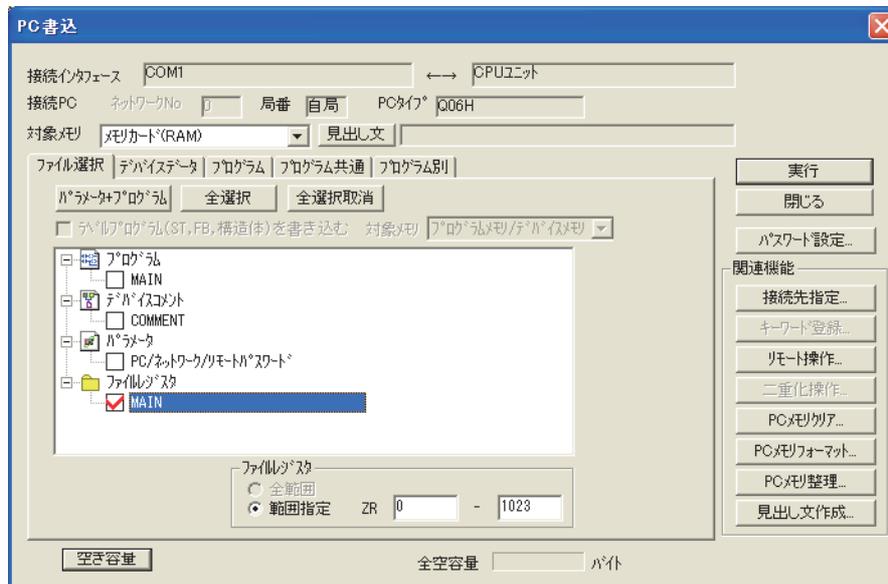


図 9.78 “PC 書込” 画面

- “対象メモリ” で、ファイルレジスタファイルを格納するメモリを“標準RAM”または“メモリカード (RAM)” から選択します。
- ファイルレジスタファイルを格納するメモリを選択すると、設定したファイルレジスタのファイル名が表示されます。
- ファイルレジスタファイルを選択します。
- “実行” ボタンをクリックすると、CPU ユニットの指定メモリに、指定点数のファイルレジスタファイルが登録されます。

Point

ファイルレジスタ容量は、1点単位で設定できます。ただし、ファイルとしては256点単位の容量を確保するため、ファイルレジスタの設定がZR0からの指定でない場合でも、ZR0から最終番号までのファイルが作成されます。例えば、ファイルレジスタの書き込み範囲をZR1000～1791に指定すると、ZR0～1791までのファイルが作成されず。その場合、ZR0～ZR999は不定なデータになるため、書き込み時はZR0から指定してください。また、ファイルレジスタ容量のチェックは、1k点単位で行うため、ファイルレジスタ容量はR0から1k点単位で設定するようにしてください。

(b) Flash カード格納時

[オンライン] → [PC 書込 (フラッシュ ROM)] → [PC 書込 (フラッシュ ROM)] で “PC 書込 (フラッシュ ROM)” 画面で行います。

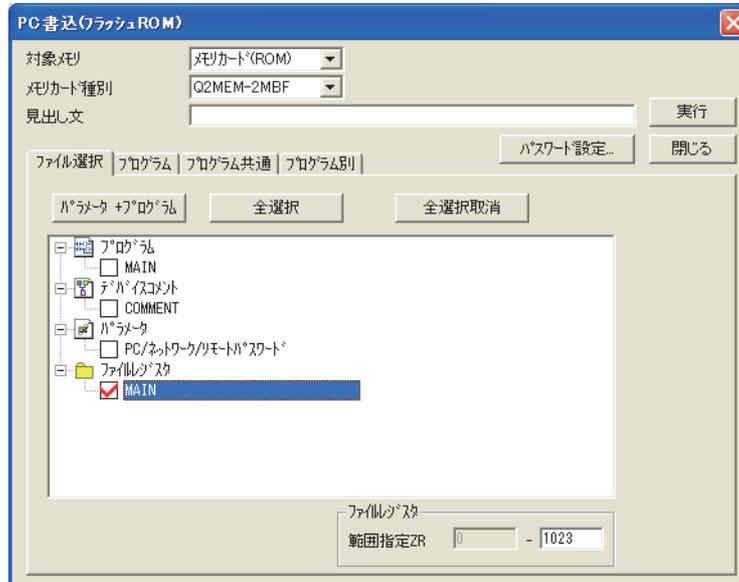


図 9.79 “PC 書込 (フラッシュ ROM)” 画面

- “対象メモリ” で、ファイルレジスタファイルを格納するメモリに “メモリカード(ROM)” を選択します。
- ファイルレジスタファイルを格納するメモリを選択すると、設定したファイルレジスタのファイル名が表示されます。
- “メモリカード種別” を選択します。
- ファイルレジスタファイルを選択します。
- “実行” ボタンをクリックすると、CPU ユニットの指定メモリに、指定点数のファイルレジスタファイルが登録されます。

9.7.5 ファイルレジスタの指定方法

(1) ブロック切換え方式

使用しているファイルレジスタ点数を 32k 点 (R0 ~ 32767) 単位で区切って指定する方式です。複数ブロックを使用している場合は、RSET 命令で使用するブロック No. に切り換えて指定します。各ブロックとも R0 ~ 32767 で指定します。

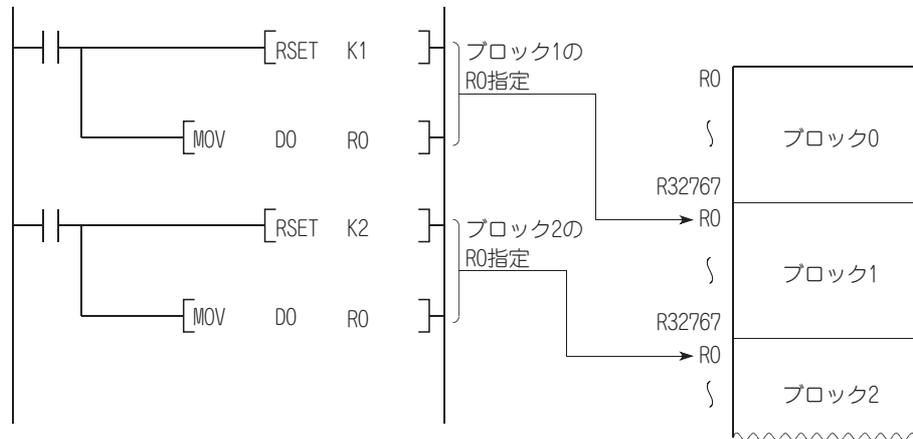


図 9.80 ブロック切換え方式

(2) 連番アクセス方式

32k 点を超えるファイルレジスタを連続したデバイス番号で指定する方式です。複数ブロックのファイルレジスタを連続したファイルレジスタとして使用できます。デバイス名は“ZR”を使用します。

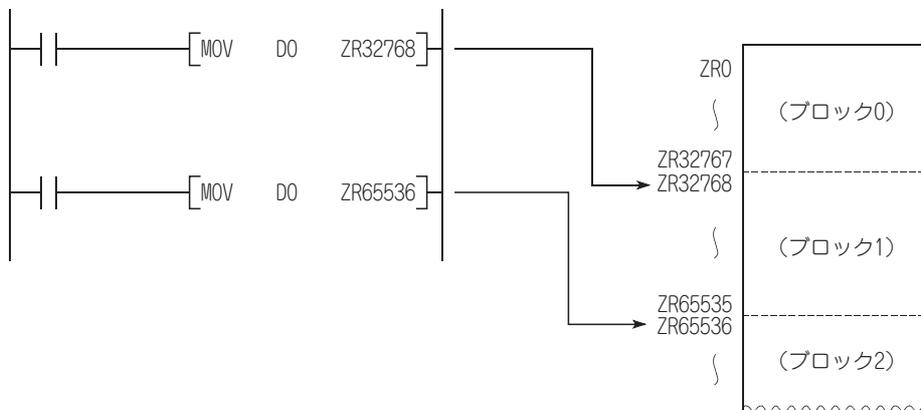


図 9.81 連番アクセス方式

Point

指定可能なブロック No. および ZR デバイスの点数は、下記により異なります。

- ファイルレジスタの格納場所 (☞ 9.7.1 項)
- ファイルレジスタの容量 (☞ 9.7.2 項)

9.7.6 ファイルレジスタ使用時の注意事項

(1) ベーシックモデル QCPU 使用時

64k 点以上のファイルレジスタ番号への書き込み／読出しを行ってもエラーにはなりません。
ただし、ファイルレジスタからの読出しを行うと不定なデータが格納されるので注意してください。

Point

ファイルレジスタファイル (MAIN.QDR) は、削除できません。
ただし、ファイルレジスタの内容は削除可能です。(☞ 9.7 節 (3))

(2) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU 使用時

(a) 未登録／登録した容量以上のファイルレジスタ番号使用時

1) ファイルレジスタのファイルが未登録の場合

ファイルレジスタへの書き込み／読出しを行ってもエラーにはなりません。
ただし、ファイルレジスタからの読出しを行うと下記ようになります。

- 標準 RAM の場合は不定なデータが格納されます。
- メモリカードの場合は “0H” が格納されます。

2) 登録した容量 (点数) 以上のファイルレジスタ番号への書き込み／読出し

ファイルレジスタ番号への書き込み／読出しを行ってもエラーにはなりません。
ただし、ファイルレジスタからの読出しを行うと下記ようになります。

- 標準 RAM の場合は不定なデータが格納されます。
- メモリカードの場合は “0H” が格納されます。

(b) ファイルレジスタ容量のチェック

ファイルレジスタに書き込み／読出しする場合は、CPU ユニットに設定した容量 (点数) 内で書き込み／読出しできるように、ファイルレジスタ容量のチェックを行ってください。

1) ファイルレジスタ容量の確認方法

ファイルレジスタ容量格納レジスタ (SD647) により確認できます。* 1
SD647 には、1k 点単位のファイルレジスタ容量が格納されます。

* 1: ファイルレジスタのファイルを切り換えると、切り換わったファイルのファイルレジスタ容量が SD647 に格納されます。

Point

ファイルレジスタ容量を 1k 点で割った場合の余りは切り捨てられます。
使用範囲のチェックを確実にするため、ファイルレジスタの設定は、1k 点 (1024 点) 単位にしてください。

2) チェックを行うタイミング

- ・ファイルレジスタを使用しているプログラムでは、ステップ0でファイルレジスタ容量をチェックしてください。
- ・ファイルレジスタのファイル切換え命令（QDRSET）実行後にファイルレジスタ容量をチェックしてください。
- ・ファイルレジスタのブロック切換え命令（RSET）は、切換え後のブロックに1k点以上容量があるか確認して実行してください。

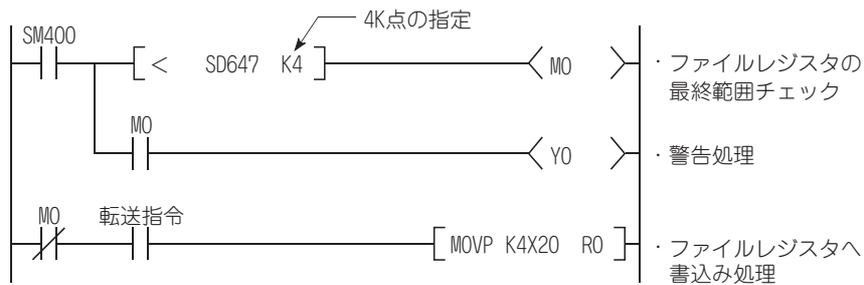
$$\text{〔ファイルレジスタ容量〕} > \text{〔}32\text{k点} \times \text{（切換えブロックNo）} + 1\text{k点} \text{〕}$$

3) ファイルレジスタ容量の確認手順

- ・各シーケンスプログラムで使用しているファイルレジスタの容量を明確にします。
- ・シーケンスプログラムでSD647にセットされているファイルレジスタの全容量から、使用する点数以上のファイルレジスタ容量があるかチェックします。

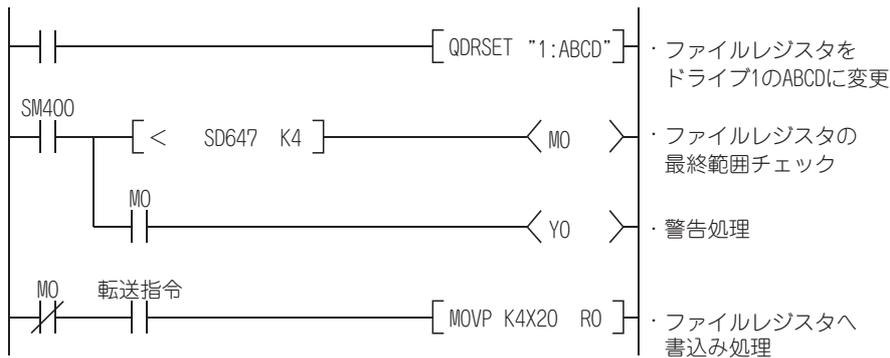
【プログラム例1】

各プログラムの先頭でファイルレジスタの使用範囲をチェックする場合



【プログラム例2】

QDRSET 命令実行後にファイルレジスタの使用範囲をチェックする場合



9.7 ファイルレジスタ(R)
9.7.6 ファイルレジスタ使用時の注意事項

【プログラム例 3】
ブロック切り替えを行う場合

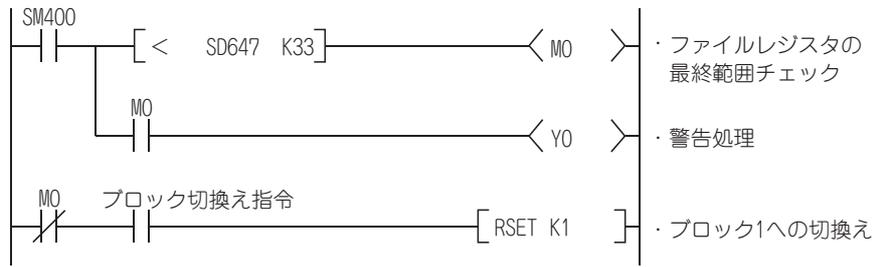


図 9.82 ファイルレジスタ確認のプログラム例

(c) 1018k 点を超えるファイルレジスタが格納されている SRAM カードを使用する場合

シリアル No. の上 5 桁が “16021” 以降のハイパフォーマンスモデル QCPU, プロセス CPU, および二重化 CPU では, 2M バイトのエリアのファイルレジスタまで使用できます。パラメータ, 命令などで 1018k 点を超えたファイルレジスタを指定しても, 停止エラーは発生しません。

(3) ファイルレジスタファイルの削除

不要になったファイルレジスタファイルの削除は, GX Developer の [オンライン] → [PC データ削除] で行います。

(4) CPU ユニットのバージョンによるファイルレジスタの処理時間の違い **注 9.11**

シリアル No. の上 5 桁が “02092” 以降のハイパフォーマンスモデル QCPU では, 標準 RAM へのアクセス命令で連番アクセス方式 (ZR □) でファイルレジスタを指定すると, “02091” 以前のハイパフォーマンスモデル QCPU より 1 命令あたりの処理時間が延びます。

- Q02CPU : 平均 1.1 μ s
- QnHCPU : 平均 0.65 μ s

MOV 命令を使用した場合の処理時間を表 9.12 に示します。

表 9.12 ファイルレジスタの指定方法による処理時間の違い

命令	Q12HCPU		Q02CPU	
	“02092” 以降	“02091” 以前	“02092” 以降	“02091” 以前
MOV K0 R0	0.11	0.11	0.26	0.26
MOV K0 ZR0	3.55	2.88	7.71	6.64

(単位 : μ s)

注 9.11 **Basic** **Process** **Redundant**

ベーシックモデル QCPU, プロセス CPU, 二重化 CPU では, CPU ユニットのバージョンによるファイルレジスタの処理時間の違いはありません。

9.8 ネスティング (N)

(1) ネスティングとは

マスタコントロール命令 (MC 命令, MCR 命令) で使用し, 動作条件を入れ子構造でプログラミングするためのデバイスです。

(2) マスタコントロール命令での指定方法

マスタコントロール命令は, 回路の共通母線の開閉により, 効率のよい回路切換えのシーケンスプログラムを作成するための命令です。

ネスティングは, 入れ子構造の外側から若番 (N0 から N14 の順) で指定します。

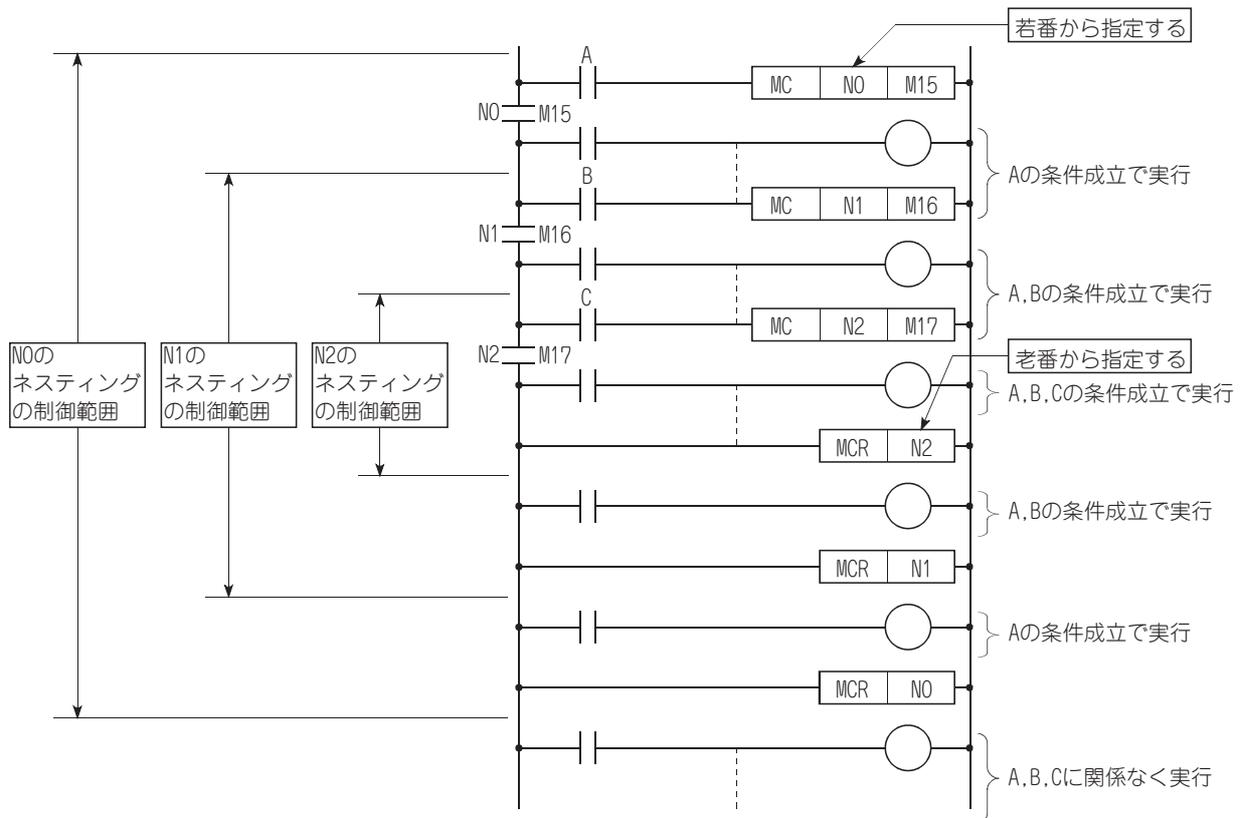


図 9.83 ネスティングを使用したプログラム例

備考

ネスティングの使用方法については, 下記マニュアルを参照してください。

☞ MELSEC-Q/L プログラミングマニュアル (共通命令編)

9.9 ポインタ (P)

(1) ポインタとは

ポインタ分岐命令 (CJ, SCJ, JMP) やサブルーチンコール命令 (CALL など) で使用するデバイスです。

(2) ポインタの用途

ポインタの用途には、下記のものがあります。

- ポインタ分岐命令 (CJ, SCJ, JMP) の飛び先指定とラベル (飛び先先頭の指定)
- サブルーチンコール命令 (CALL, CALLP) の呼び出し先とラベル (サブルーチンプログラム先頭の指定)

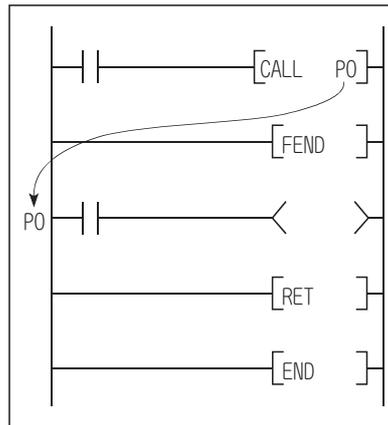


図 9.84 ポインタを使用したプログラム

(3) ポインタの種類

(a) ベーシックモデル QCPU の場合

ベーシックモデル QCPU は、複数のプログラムが実行できないため、ローカルポインタ、共通ポインタの区別がありません。

(b) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU の場合

ポインタには、下記の 2 種類があります。

- ローカルポインタ (☞ 9.9.1 項) :
各プログラムで独立して使用するポインタです。
- 共通ポインタ (☞ 9.9.2 項) :
実行しているすべてのプログラムが、サブルーチンコール命令で呼び出しできるポインタです。

(4) 使用可能なポインタの点数

- ベーシックモデル QCPU : 300 点
- ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU : 4096 点

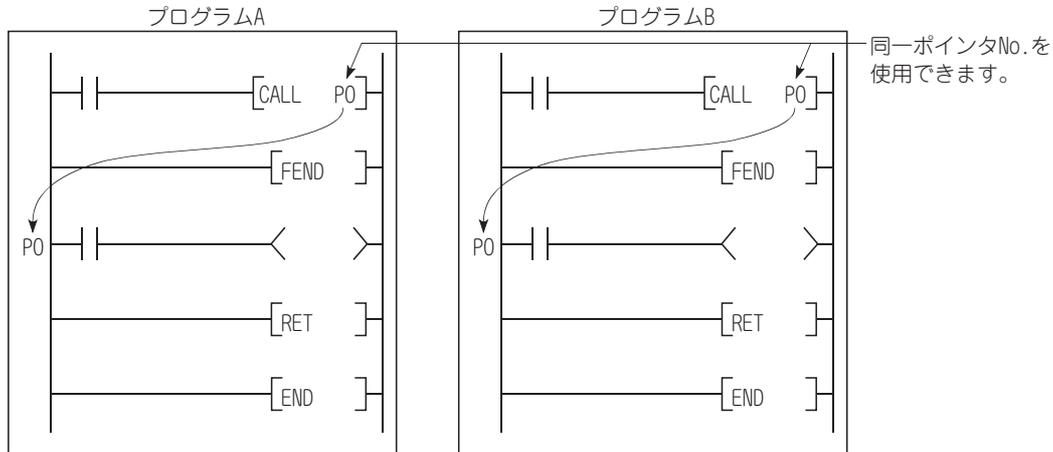
備考

ポインタ分岐命令、サブルーチンコール命令については、下記マニュアルを参照してください。
☞ MELSEC-Q/L プログラミングマニュアル (共通命令編)

9.9.1 ローカルポインタ 注 9.12

(1) ローカルポインタとは

各プログラム内でのジャンプ、サブルーチンプログラムの呼び出しに使用するポインタです。
各プログラムで同一ポインタ No. を使用できます。

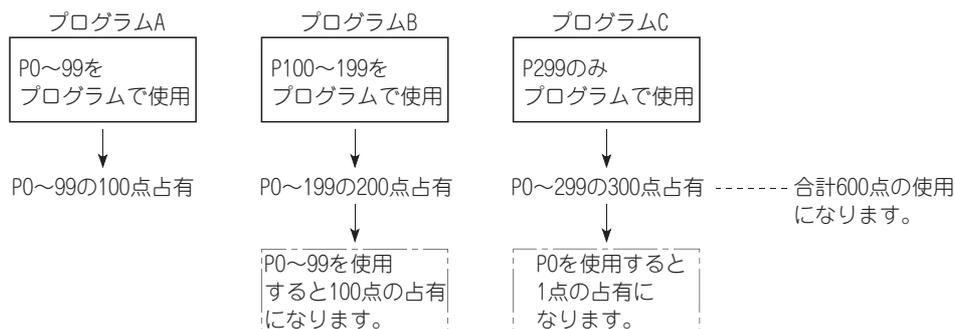


(2) ローカルポインタ点数の考え方

ローカルポインタは、プログラムメモリに格納されているすべてのプログラムで分割して使用します。
ローカルポインタ点数は、P0 から使用しているローカルポインタの最大 No. までになります。(使用点数は、CPU ユニットのシステムが算出します。)

例えば、P99 のみを使用している場合、P0 ~ P99 の 100 点の使用になります。
複数のプログラムでポインタを使用する場合は、各プログラムでポインタは P0 から順に使用してください。

例 下記のようにポインタを使用すると、合計 600 点になります。



注 9.12 Basic

ベーシックモデル QCPU では、ローカルポインタ、共通ポインタの区別はありません。

(3) ローカルポインタ使用時の注意事項

(a) ローカルポインタが記述されているプログラムについて

他のプログラムからジャンプできません。

ローカルポインタが記述されているプログラムファイルのサブルーチンプログラムは、他プログラムから ECALL 命令で呼び出します。

(b) ローカルポインタの合計点数について

各々のプログラムで使用しているポインタの合計点数が 4096 点を超えると “ポインタ構成エラー” (エラーコード：4020) になります。

9.9.2 共通ポインタ 注 9.13

(1) 共通ポインタとは

実行しているすべてのプログラムからサブルーチンプログラムを呼び出すためのポインタです。

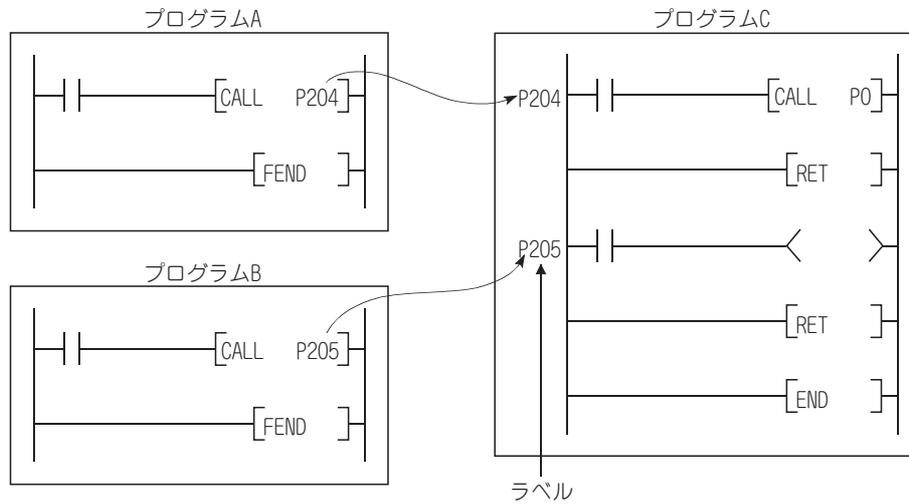


図 9.87 他のプログラムのポインタを呼出し (共通ポインタ)

注 9.13 **Basic**

ベーシックモデル QCPU では、ローカルポインタ、共通ポインタの区別はありません。

(2) 共通ポインタの使用範囲

PC パラメータの PC システム設定で共通ポインタの先頭番号を設定します。

設定したポインタ番号から P4095 までが共通ポインタになります。

ただし、パラメータで設定できる共通ポインタの先頭番号は、ローカルポインタの使用合計点数以降です。

共通ポインタの
先頭番号を設定



図 9.88 共通ポインタの設定画面

例えば、プログラム A で 100 点、プログラム B で 100 点、プログラム C で 200 点の合計 400 点をローカルポインタで使用している場合は、P400 以降を共通ポインタに設定できます。

(3) 注意事項

- 1) 同一ポインタ No. をラベルとして使用できません。
使用すると，“ポインタ構成エラー”（エラーコード：4021）になります。
- 2) 各々のプログラムで使用しているローカルポインタの合計点数が共通ポインタの先頭番号を超えると “ポインタ構成エラー”（エラーコード：4020）になります。

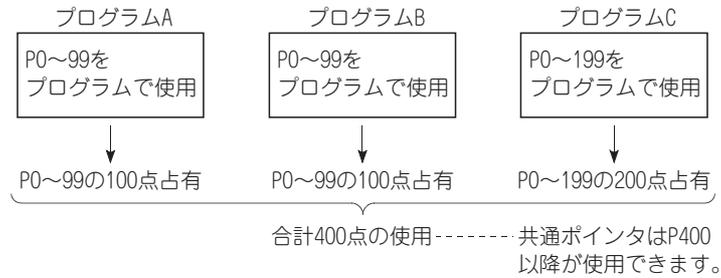


図 9.89 共通ポインタの使用範囲の考え方

Point

ポインタ分岐命令では、他のプログラムの共通ポインタに飛ばすことはできません。
共通ポインタは、サブルーチンコール命令でのみ使用してください。

9.10 割込みポインタ (I)

(1) 割込みポインタとは

割込みプログラムの先頭にラベルとして使用するデバイスです。
すべてのプログラムで使用可能です。

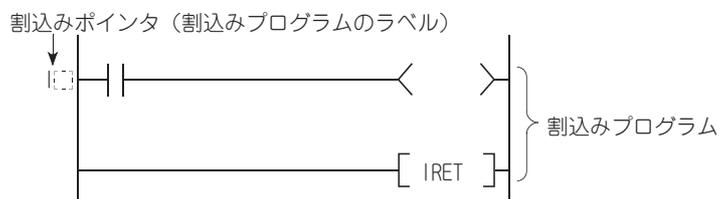


図 9.90 割込みポインタ

(2) 使用可能な割込みポインタの点数

- ベーシックモデル QCPU : 128 点 (10 ~ 127)
- ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU : 256 点 (10 ~ 255)

(3) 割込み要因

使用可能な割込みポイントの割込み要因を、表 9.13 に示します。

表 9.13 割込み要因の分類

割込み要因	割込み ポイント 番号	内 容	対応 CPU ユニット			
			ベーシック モデル QCPU	ハイパ フォーマン スモデル QCPU	プロセス CPU	二重化 CPU
割込みユニット* ¹ による割込み* ⁷	10 ~ 15	割込みユニットからの割込み入力	○	○	○	○
シーケンス起動発生ユニットによる 割込み	116 ~ 27	CPU ユニットへ、割込み起動をかけることが 可能な AnS/A シリーズ対応の特殊機能ユニッ ト* ⁶ からの割込み	×	○	×	×
内部タイマによる割込み	128 ~ 31	CPU ユニットの内部タイマによる定周期 割込み	○	○	○	○
	149		×	△* ²	×	×
エラー発生割込み* ³	132 ~ 39	シーケンスプログラムの演算を続行する エラー発生による割込み	×	○	○	○
	140, 141		×	×	×	○
インテリジェント機能ユニット割込み	150 ~ 255 * ⁴	インテリジェント機能ユニット* ⁵ からの 割込み	○	○	○	○

* 1：使用可能な割込みユニットについては、下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）

* 2：Q02CPU では、割込みポイント 149（高速割込み機能専用ポイント）は使用できません。

* 3：PC パラメータの PC RAS 設定において、エラー時の運転モードが設定可能なエラーについては、設定が“続行”となっている場合のみ割込みを実行します。

* 4：ベーシックモデル QCPU は、150 ~ 127 です。

* 5：該当するユニットには、シリアルコミュニケーションユニット、MELSECNET/H ユニット、Ethernet ユニット、高速カウンタユニットなどがあります。

詳細については、各ユニットのマニュアルを参照してください。

* 6：該当するユニットには、インテリジェントコミュニケーションユニットがあります。

詳細については、各ユニットのマニュアルを参照してください。

* 7：ハイパフォーマンスモデル QCPU で割込みユニットを使用する場合は、CPU ユニットおよび GX Developer のバージョンを確認してください。（☞ 付 2.2）

Point

インテリジェント機能ユニット割込み（☞ 6.23 節）を使用する場合は、PC パラメータの PC システム設定でインテリジェント機能ユニット設定（割込みポイント設定）が必要です。（☞ 8.1.1 項 (2), ☞ 8.1.2 項 (2)）

9.10.1 割込みポイント番号と割込み要因の一覧

割込みポイント番号と割込み要因の一覧を、下記に示します。

(1) ベーシックモデル QCPU

表 9.14 割込みポイント番号と割込み要因一覧表 (ベーシックモデル QCPU)

I No.	割込み要因	優先順位
I0	割込みユニット (QI60) による割込み	1 点目
I1		2 点目
I2		3 点目
I3		4 点目
I4		5 点目
I5		6 点目
I6		7 点目
I7		8 点目
I8		9 点目
I9		10 点目
I10		11 点目
I11		12 点目
I12		13 点目
I13		14 点目
I14		15 点目
I15		16 点目
I16 ~ 27	アキ	—
I28	内部タイマによる割込み * 1	100ms
I29		40ms
I30		20ms
I31		10ms
I32 ~ 49	アキ	—
I50 ~ 127 * 2 * 3	インテリジェント機能ユニット / 割込みユニット (QI60) による割込み	パラメータでどのインテリジェント機能ユニット / 割込みユニット (QI60) が使用するかを設定する
		21 ~ 98

* 1 : 内部タイマの時限は、デフォルト値を示しています。

PC パラメータの PC システム設定で 2ms ~ 1000ms までを 1ms 単位で変更できます。

* 2 : インテリジェント機能ユニット割込みを使用する場合は、PC パラメータの PC システム設定でインテリジェント機能ユニット設定 (割込みポイント設定) が必要です。(インテリジェント機能ユニットからの割込みについては 6.23 節を参照してください。)

* 3 : I50 ~ 127 の優先順位は I50 が最も高く (優先順位 21)、I127 が最も低く (優先順位 98) になります。

(2) ハイパフォーマンスモデル QCPU

(a) Qシリーズ割込みユニット装着時

表 9.15 割込みポイント番号と割込み要因一覧表 (ハイパフォーマンスモデル QCPU)

I No.	割込み要因	優先順位	I No.	割込み要因	優先順位		
10	割込みユニット (QI60) による割込み	2 ~ 223	132 * 5	停止するエラー全般	1		
11			133	SINGLE PS. DOWN * 3 * 4			
12			134	UNIT VERIFY ERR., FUSE BREAK OFF, SP. UNIT ERROR, MULTI CPU ERROR			
13				135		OPERATION ERROR, SFCP OPE. ERROR, SFCP EXE. ERROR, EX.POWER OFF	
14						136	ICM. OPE ERROR, FILE OPE. ERROR
15							137
16			138				
17				139			
18						140 ~ 48	
19							149
110			150 ~ 255				
111				パラメータでどのインテリジェント機能ユニットが使用するかを設定する			
112						2 ~ 223	
113							内部タイマによる割込み * 2
114			100ms				
115				40ms			
116	20ms						
117		10ms					
118			224				
119			225				
120	226						
121	227						
122	228						
123	229						
124	230						
125	231						
126	232						
127	233						
128	234						
129	235						
130	236						
131	237						
	238						
	239						
	240						
	241						
	242						
	243						
	244						
	245						
	246						
	247						
	248						
	249						
	250						
	251						
	252						
	253						
	254						
	255						

9.10 割込みポイント (I)
9.10.1 割込みポイント番号と割込み要因の一覧

- * 1 : 1 枚目～ 12 枚目は、ベースユニットに装着されているシーケンス起動発生ユニットの内、ハイパフォーマンスモデル QCPU に近いユニットを 1 枚目として順に割り付けます。
- * 2 : 内部タイマの時限は、デフォルト値を示しています。
PC パラメータの PC システム設定で 0.5ms ～ 1000ms までを 0.5ms 単位で変更できます。
- * 3 : シリアル No. の上 5 桁が “07032” 以降が対象です。
- * 4 : マルチ CPU システム構成時は、すべての CPU ユニットのシリアル No. の上 5 桁が “07032” 以降の場合に 1 号機の CPU ユニットでのみ使用可能です。
- * 5 : エラー発生割込みで “I32 (停止するエラー全般)” は、エラー発生時 I32 の処理を行ったあと、ハイパフォーマンスモデル QCPU は停止します。
- * 6 : I32 ～ 48 は、電源立上げ/ハイパフォーマンスモデル QCPU のリセットを行うと、実行禁止状態 (DI) になっています。
I32 ～ 48 を使用する場合は、IMASK 命令で割込み許可状態にしてください。
IMASK 命令については、下記マニュアルを参照してください。
 MELSEC-Q/L プログラミングマニュアル (共通命令編)
- * 7 : 内部タイマの時限は、PC パラメータの PC システム設定のシステム割込み設定の高速割り込み設定で設定します。
設定範囲は、0.2ms ～ 1.0ms の間を 0.1ms 単位で設定します。
- * 8 : インテリジェント機能ユニット割込みを使用する場合は、PC パラメータの PC システム設定でインテリジェント機能ユニット設定 (割込みポインタ設定) が必要です。(インテリジェント機能ユニットからの割込みについては 6.23 節を参照してください。)
- * 9 : I50 ～ 255 の優先順位は I50 が最も高く (優先順位 2)、I255 が最も低く (優先順位 223) になります。
- * 10 : PC パラメータで I49 を設定した場合、他の割込みプログラム (I0 ～ 48, I50 ～ 255)、定周期実行タイププログラムは実行しないでください。
定周期実行タイププログラムなどを実行すると I49 の割込みプログラムは、設定した割込み周期間隔で実行できなくなります。

(b) A シリーズ割込みユニット装着時

表 9.16 割込みポイント番号と割込み要因一覧表 (ハイパフォーマンスモデル QCPU)

I No.	割込み要因	優先順位	I No.	割込み要因	優先順位		
10	割込みユニット (A1SI61) による割込み	1 点目	220	132 * 5	1		
11		2 点目	221	133		停止するエラー全般 SINGLE PS. DOWN * 3 * 4	
12		3 点目	222	134		UNIT VERIFY ERR., FUSE BREAK OFF, SP. UNIT ERROR, MULTI CPU ERROR	
13		4 点目	223				
14		5 点目	224	135		OPERATION ERROR, SFCP OPE. ERROR, SFCP EXE. ERROR, EX.POWER OFF	
15		6 点目	225				
16		7 点目	226	136		ICM. OPE ERROR, FILE OPE. ERROR	
17		8 点目	227				
18		9 点目	228	137		ア キ	
19		10 点目	229				
110		11 点目	230	138		PRG. TIME OVER	
111		12 点目	231				
112		13 点目	232	139		CHK 命令実行, アナンシェータ検出	
113		14 点目	233				
114		15 点目	234	140 ~ 48		ア キ	
115		16 点目	235				
116	シーケンス起動 発生ユニット による割込み * 1	1 枚目	208	149	内部タイマ による割込み	0.2ms ~ 1ms * 7	* 10
117		2 枚目	209	150 ~ 255	インテリジエント機能ユニット 割込み * 8 * 9	パラメータでどのインテリジエント機能ユニット が使用するかを設定する	2 ~ 207
118		3 枚目	210				
119		4 枚目	211				
120		5 枚目	212				
121		6 枚目	213				
122		7 枚目	214				
123		8 枚目	215				
124		9 枚目	216				
125		10 枚目	217				
126		11 枚目	218				
127		12 枚目	219				
128	内部タイマに による割込み * 2	100ms	239				
129		40ms	238				
130		20ms	237				
131		10ms	236				

9.10 割込みポイント (I)
9.10.1 割込みポイント番号と割込み要因の一覧

- * 1 : 1 枚目～ 12 枚目は、ベースユニットに装着されているシーケンス起動発生ユニットの内、ハイパフォーマンスモデル QCPU に近いユニットを 1 枚目として順に割り付けます。
- * 2 : 内部タイマの時限は、デフォルト値を示しています。
PC パラメータの PC システム設定で 0.5ms ～ 1000ms までを 0.5ms 単位で変更できます。
- * 3 : シリアル No. の上 5 桁が “07032” 以降が対象です。
- * 4 : マルチ CPU システム構成時は、すべての CPU ユニットのシリアル No. の上 5 桁が “07032” 以降の場合に 1 号機の CPU ユニットでのみ使用可能です。
- * 5 : エラー発生割込みで “I32 (停止するエラー全般)” は、エラー発生時 I32 の処理を行ったあと、ハイパフォーマンスモデル QCPU は停止します。
- * 6 : I32 ～ 48 は、電源立上げ/ハイパフォーマンスモデル QCPU のリセットを行うと、実行禁止状態 (DI) になっています。
I32 ～ 48 を使用する場合は、IMASK 命令で割込み許可状態にしてください。
IMASK 命令については、下記マニュアルを参照してください。
☞ MELSEC-Q/L プログラミングマニュアル (共通命令編)
- * 7 : 内部タイマの時限は、PC パラメータの PC システム設定のシステム割込み設定の高速割込み設定で設定します。
設定範囲は、0.2ms ～ 1.0ms の間を 0.1ms 単位で設定します。
- * 8 : インテリジェント機能ユニット割込みを使用する場合は、PC パラメータの PC システム設定でインテリジェント機能ユニット設定 (割込みポイント設定) が必要です。(インテリジェント機能ユニットからの割込みについては 6.23 節を参照してください。)
- * 9 : I50 ～ 255 の優先順位は I50 が最も高く (優先順位 2)、I255 が最も低く (優先順位 207) になります。
- * 10 : PC パラメータで I49 を設定した場合、他の割込みプログラム (I0 ～ 48, I50 ～ 255)、定周期実行タイププログラムは実行しないでください。
定周期実行タイププログラムなどを実行すると I49 の割込みプログラムは、設定した割込み周期間隔で実行できなくなります。

(3) プロセス CPU

表 9.17 割り込みポイント番号と割り込み要因一覧表 (プロセス CPU)

I No.	割り込み要因		優先順位	I No.	割り込み要因		優先順位
10	割り込みユニット (QI60) による 割り込み	1 点目	208	132 * 4	エラー発生 割り込み * 4 * 5	停止するエラー全般	1
11		2 点目	209	133		SINGLE PS. DOWN * 2 * 3	
12		3 点目	210	134		UNIT VERIFY ERR., FUSE BREAK OFF, SP. UNIT ERROR, MULTI CPU ERROR	
13		4 点目	211				
14		5 点目	212				
15		6 点目	213				
16		7 点目	214	135		OPERATION ERROR, SFCP OPE. ERROR, SFCP EXE. ERROR, EX.POWER OFF	
17		8 点目	215				
18		9 点目	216	136		ICM. OPE ERROR, FILE OPE. ERROR	
19		10 点目	217				
110		11 点目	218	137		ア キ	
111		12 点目	219	138		PRG. TIME OVER	
112		13 点目	220	139		CHK 命令実行, アナンシェータ検出	
113		14 点目	221				
114		15 点目	222				
115		16 点目	223	140 ~ 49		-	
116 ~ 127	-	ア キ	-	150 ~ 255	インテリジェント機能ユニット 割り込み * 6 * 7	パラメータでどのインテリジェント機能ユニットが使用するかを設定する	2 ~ 207
128	内部タイマによる 割り込み * 1	100ms	227				
129		40ms	226				
130		20ms	225				
131		10ms	224				

- * 1 : 内部タイマの時限は、デフォルト値を示しています。
PC パラメータの PC システム設定で 0.5ms ~ 1000ms までを 0.5ms 単位で変更できます。
- * 2 : シリアル No. の上 5 桁が “07032” 以降が対象です。
- * 3 : マルチ CPU システム構成時は、すべての CPU ユニットのシリアル No. の上 5 桁が “07032” 以降の場合に 1 号機の CPU ユニットでのみ使用可能です。
- * 4 : エラー発生割り込みで “132 (停止するエラー全般)” は、エラー発生時 132 の処理を行ったあと、プロセス CPU は停止します。
- * 5 : 132 ~ 48 は、電源立上げ/プロセス CPU のリセットを行うと、実行禁止状態 (DI) になっています。
132 ~ 48 を使用する場合は、IMASK 命令で割り込み許可状態にしてください。
IMASK 命令については、下記マニュアルを参照してください。
☞ MELSEC-Q/L プログラミングマニュアル (共通命令編)
- * 6 : インテリジェント機能ユニット割り込みを使用する場合は、PC パラメータの PC システム設定でインテリジェント機能ユニット設定 (割り込みポイント設定) が必要です。(インテリジェント機能ユニットからの割り込みについては 6.23 節を参照してください。)
- * 7 : 150 ~ 255 の優先順位は 150 が最も高く (優先順位 2)、1255 が最も低く (優先順位 207) になります。

9.10 割り込みポイント (I)
9.10.1 割り込みポイント番号と割り込み要因の一覧

(4) 二重化 CPU

表 9.18 割込みポイント番号と割込み要因一覧表 (二重化 CPU)

I No.	割込み要因		優先順位	I No.	割込み要因		優先順位
I10	割込みユニット (QI60) による 割込み	1 点目	208	I32 * 2	エラー発生 割込み * 2 * 3	停止するエラー全般	1
I11		2 点目	209	I33		SINGLE PS. DOWN	
I12		3 点目	210	I34		UNIT VERIFY ERR., FUSE BREAK OFF, SP. UNIT ERROR	
I13		4 点目	211				
I14		5 点目	212				
I15		6 点目	213				
I16		7 点目	214	I35		OPERATION ERROR, SFCP OPE. ERROR, SFCP EXE. ERROR, EX.POWER OFF	
I17		8 点目	215				
I18		9 点目	216	I36		ICM. OPE ERROR, FILE OPE. ERROR	
I19		10 点目	217				
I110		11 点目	218				
I111		12 点目	219	I37		ア キ	
I112		13 点目	220	I38		PRG. TIME OVER	
I113		14 点目	221	I39		CHK 命令実行, アナンシェータ検出	
I114		15 点目	222				
I115		16 点目	223	I40		CAN'T SWITCH	
I116 ~ I127	-	ア キ	-	I41	STANDBY	-	
				I42 ~ I49	-		ア キ
I128 ~ I131	内部タイマによる割込み * 1	100ms	227	I150 ~ I255	インテリジェント機能ユニット 割込み * 4 * 5	パラメータでどのインテリジェント機能ユニットが使用するかを設定する	2 ~ 207
		40ms	226				
		20ms	225				
		10ms	224				

* 1: 内部タイマの时限は、デフォルト値を示しています。

PC パラメータの PC システム設定で 0.5ms ~ 1000ms までを 0.5ms 単位で変更できます。

* 2: エラー発生割込みで "I32 (停止するエラー全般)" は、エラー発生時 I32 の処理を行ったあと、二重化 CPU は停止します。

* 3: I32 ~ 48 は、電源上げ/二重化 CPU のリセットを行うと、実行禁止状態 (DI) になっています。

I32 ~ 48 を使用する場合は、IMASK 命令で割込み許可状態にしてください。

IMASK 命令については、下記マニュアルを参照してください。

 MELSEC-Q/L プログラミングマニュアル (共通命令編)

* 4: インテリジェント機能ユニット割込みを使用する場合は、PC パラメータの PC システム設定でインテリジェント機能ユニット設定 (割込みポイント設定) が必要です。(インテリジェント機能ユニットからの割込みについては 6.23 節を参照してください。)

* 5: I150 ~ 255 の優先順位は I150 が最も高く (優先順位 2)、I255 が最も低く (優先順位 207) になります。

9.11 その他のデバイス

9.11.1 SFC ブロックデバイス (BL)

SFC プログラムの指定ブロックが活性しているかのチェックに使用するデバイスです。

☞ MELSEC-Q/L/QnA プログラミングマニュアル (SFC 編)

9.11.2 SFC 移行デバイス (TR) 注 9.14

SFC プログラムの指定ブロックの指定移行条件が強制移行に指定されているかのチェック用デバイスです。

☞ MELSEC-Q/L/QnA プログラミングマニュアル (SFC 編)

9.11.3 ネットワーク No. 指定デバイス (J)

(1) ネットワーク No. 指定デバイスとは

リンク専用命令でネットワーク No. を指定する場合に使用するデバイスです。

(2) 指定方法

リンク専用命令で図 9.91 のように指定します。

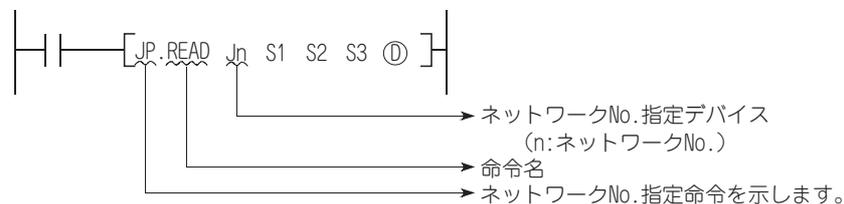


図 9.91 ネットワークNo.指定デバイスの指定方法

備考

リンク専用命令の詳細については、下記マニュアルを参照してください。

☞ 各ネットワークユニットのマニュアル

注 9.14 Basic

ベーシックモデル QCPU では、SFC 移行デバイス (TR) は使用できません。

9.11.4 I/O No. 指定デバイス (U)

(1) I/O No. 指定デバイスとは

インテリジェント機能ユニット専用命令で I/O No. を指定する場合に使用するデバイスです。

(2) 指定方法

インテリジェント機能ユニット専用命令で図 9.92 のように指定します。

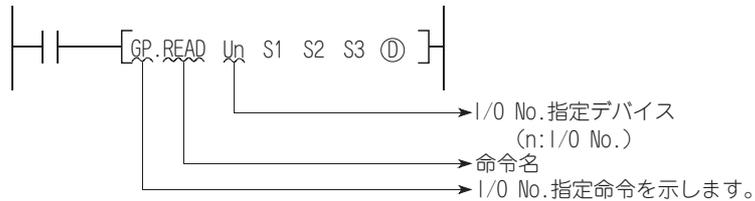


図 9.92 I/O No.指定デバイスの指定方法

備考

インテリジェント機能ユニット専用命令の詳細については、下記のマニュアルを参照してください。

☞ 使用するインテリジェント機能ユニットのマニュアル

9.11.5 マクロ命令引数デバイス (VD)

(1) マクロ命令引数デバイスとは

マクロ登録回路で使用するデバイスです。

マクロ登録回路で VD を使用すると、マクロ命令使用時に指定したデバイスに変換されます。

(2) 指定方法

GX Developer のマクロ登録時にマクロ命令として登録する回路で使用しているデバイスのうち、VD にするデバイスを指定します。

シーケンスプログラムでマクロ命令を使用する場合は、マクロ登録回路で使用しているマクロ命令引数デバイスの若番順に対応するデバイスに指定します。

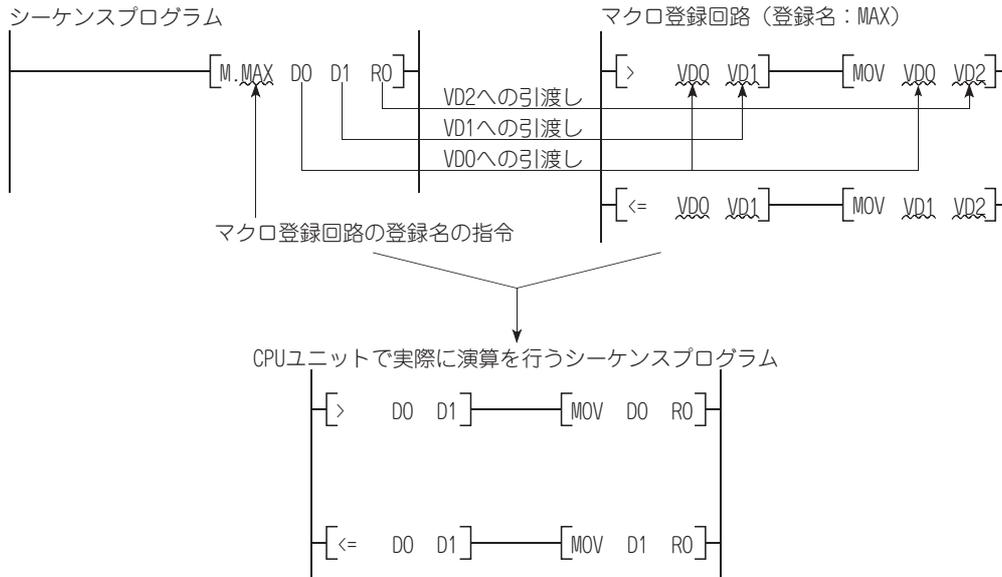


図 9.93 マクロ命令引数デバイスの指定

Point

- マクロ命令引数デバイスは、1つのマクロ登録回路で VD0 ~ 9 が使用できます。
- GX Developer の読出モードでプログラムを表示しているときは、マクロ命令形式での表示もできます。(表示の切換えは [表示] → [マクロ命令形式表示] で行います。)



図 9.94 マクロ命令の表示切換え設定

9.12 定 数

9.12.1 10 進定数 (K)

(1) 10 進定数とは

シーケンスプログラムで 10 進数データを指定するデバイスです。

シーケンスプログラムでは、K□□□ (例：K1234) で指定します。

CPU ユニット内部には 2 進数 (BIN) で格納されます。(☞ 2.4.1 項)

(2) 指定範囲

10 進定数での指定範囲は、次のようになります。

- ワードデータ (16 ビット) 使用時 …… K-32768 ~ 32767
- 2 ワードデータ (32 ビット) 使用時 …… K-2147483648 ~ 2147483647

Point

最上位ビットは符号ビットになります。

9.12.2 16 進定数 (H)

(1) 16 進定数とは

シーケンスプログラムで 16 進数または BCD データを指定するデバイスです。

(BCD でデータ指定時は、16 進数の各桁を 0 ~ 9 で指定します。)

シーケンスプログラムでは、H□□□ (例：H1234) で指定します。(☞ 2.4.2 項)

(2) 指定範囲

16 進定数での設定範囲は、次のようになります。

- ワードデータ (16 ビット) 使用時 …… H0 ~ FFFF (BCD データのときは H0 ~ 9999)
- 2 ワードデータ (32 ビット) 使用時 …… H0 ~ FFFFFFFF (BCD データのときは H0 ~ 99999999)

9.12.3 実数 (E) 注 9.15

(1) 実数とは

シーケンスプログラムで実数を指定するデバイスです。

シーケンスプログラムでは、E□□□(例：E1.234)で指定します。(付 2.4.4 項)

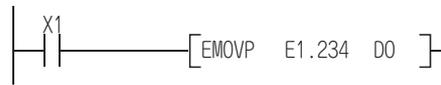


図 9.95 実数の指定

(2) 指定範囲

(a) 実数の設定範囲

$-2^{128} < \text{デバイス} \leq -2^{-126}$, 0 , $2^{-126} \leq \text{デバイス} < 2^{128}$

(b) オーバーフローおよびアンダーフロー時の動作

演算時にオーバーフローおよびアンダーフローが発生した場合の動作を表 9.19 に示します。

表 9.19 オーバーフローおよびアンダーフロー時の動作

CPU ユニット	オーバーフロー時の動作	アンダーフロー時の動作
ベーシックモデル QCPU	"OPERATION ERROR" (エラーコード：4100)	"OPERATION ERROR" (エラーコード：4100)
ハイパフォーマンスモデル QCPU		
プロセス CPU		エラーは発生せずに 0 になる *1
二重化 CPU		

* 1：シリアル No. の上 5 桁が "07031" 以前のプロセス CPU では、"OPERATION ERROR" (エラーコード：4100) となります。

(c) 特殊な値 *1 を入力したときの動作

入力データが特殊な値で演算を行った場合は、"OPERATION ERROR" (エラーコード：4140) になります。(ハイパフォーマンスモデル QCPU では、内部演算を単精度に設定している場合のみエラーが発生します。)

* 1：特殊な値とは、-0、非正規化数、非数、 $\pm \infty$ です。

(3) 指定方法

シーケンスプログラムでは、実数を下記の表現で指定できます。

- ・ ノーマル表現 … 設定する数値をそのまま指定します。

例 10.2345 は、E10.2345 で指定します。

- ・ 指数表現 … 設定する数値を (数値) $\times 10^n$ で指定します。

例 1234 は、E1.234 + 3 で指定します。* 1

* 1：E1.234 + 3 の + 3 は 10^3 を示しています。

注 9.15 **Basic**

ベーシックモデル QCPU で実数演算機能を使用する場合には、CPU ユニットおよび GX Developer のバージョンを確認してください。(付 2.1)

9.12.4 文字列（ “ ” ） 注 9.16

(1) 文字列定数とは

シーケンスプログラムで文字列を指定するデバイスです。

“ ” で囲まれた半角文字（例：“ABCD1234”）で指定します。

(2) 使用できる文字

文字列には、シフト JIS コードを使用できます。

CPU ユニットでは大文字と小文字の区別を行っています。

(3) 指定文字数

文字列は、指定文字から NUL コード (00H) までが単位となります。

ただし、文字列を使用した命令（\$MOV 命令など）で指定できる文字列は最大 32 文字までです。

注 9.16 **Basic**

ベーシックモデル QCPU では、\$MOV, STR, DST, VAL, DVAL, ESTR, EVAL 命令でのみ文字列が使用可能です。

9.13 デバイスの便利な使い方 注 9.17

CPUユニットで複数のプログラムを実行させる場合、内部ユーザデバイスのローカルデバイス指定により、各々のプログラムを独立化できます。

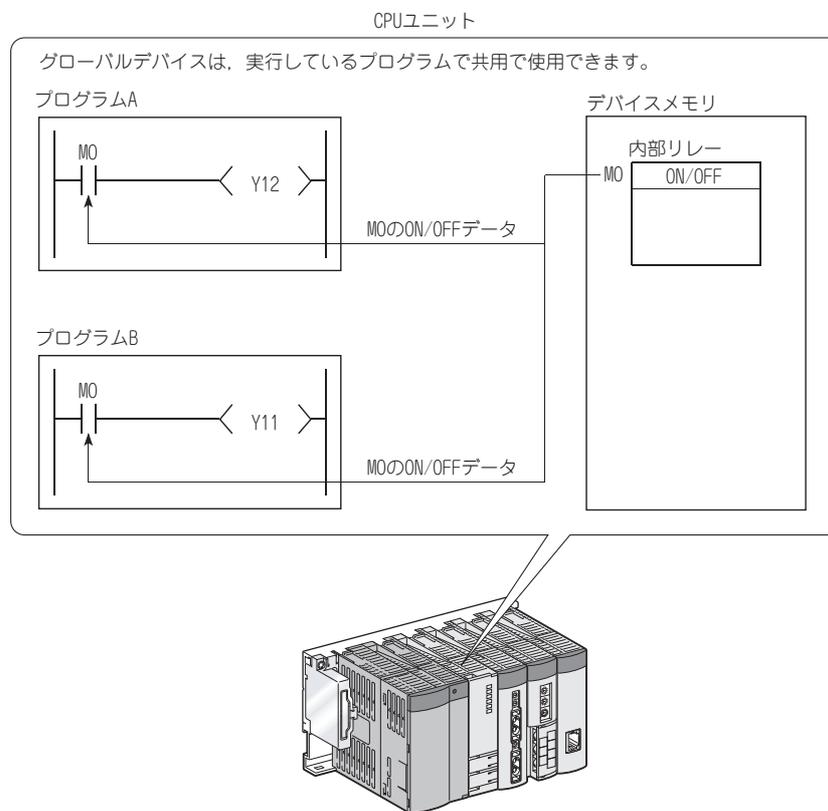
CPUユニットの各デバイスは、下記の2種類に分類されます。

- 実行している複数のプログラムで共用できるグローバルデバイス
- 各プログラムごとに独立したデバイスとして使用できるローカルデバイス

9.13.1 グローバルデバイス

CPUユニットで実行しているプログラムが共用で使用できるデバイスです。

グローバルデバイスデータは、CPUユニットのデバイスメモリに格納され、全プログラムで同一データを使用できます。



注 9.17 Basic

ベーシックモデル QCPU では、ローカルデバイス、グローバルデバイスの区別はありません。

- ローカルデバイス（ 9.13.2 項）の設定を行っていないデバイスは、すべてグローバルデバイスです。
- 複数のプログラムを実行させる場合は、あらかじめ全プログラムで共用する範囲、各々のプログラムが単独で使用する範囲（ 9.13.2 項）の取決めが必要です。



図 9.97 デバイスの使用範囲の取決め

9.13.2 ローカルデバイス

プログラムごとに独立して使用できるデバイスです。

使用すると、複数の独立したプログラムを実行する場合に、他のプログラムを意識することなくプログラミングできます。

ただし、ローカルデバイスデータは、標準RAMとメモリカード（SRAM）にのみ格納できます。

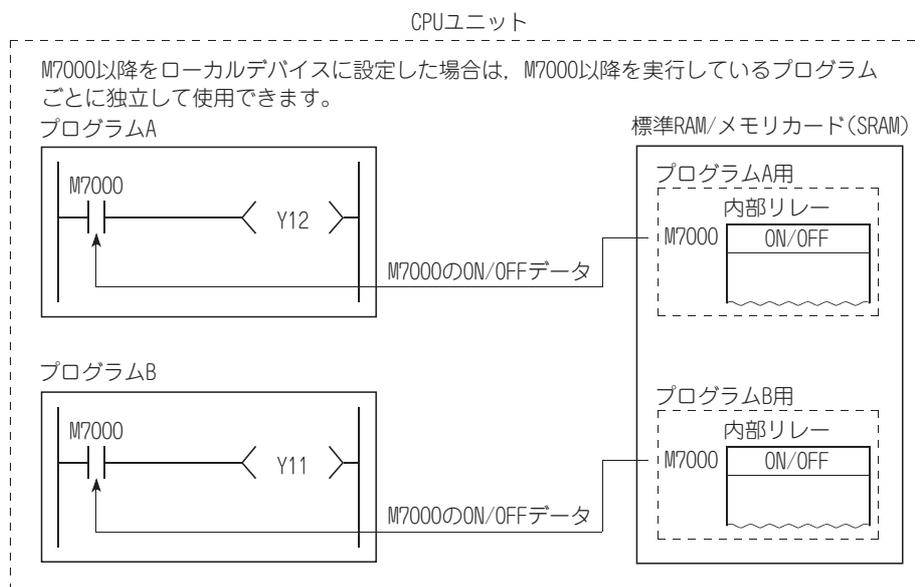


図 9.98 ローカルデバイスの使用

(1) ローカルデバイスとして使用できるデバイス

ローカルデバイスとして使用できるデバイスは、下記のとおりです。

- 内部リレー (M)
- エッジリレー (V)
- タイマ (T, ST)
- カウンタ (C)
- データレジスタ (D)

(2) ローカルデバイスファイルの退避と復帰

ローカルデバイスを使用しているプログラムは、プログラム実行後に標準 RAM またはメモリカード (SRAM) のローカルデバイスファイルのデータと、CPU ユニットのデバイスメモリのデータの交換を行います。このため、データの交換時間だけスキャンタイムが延びます。

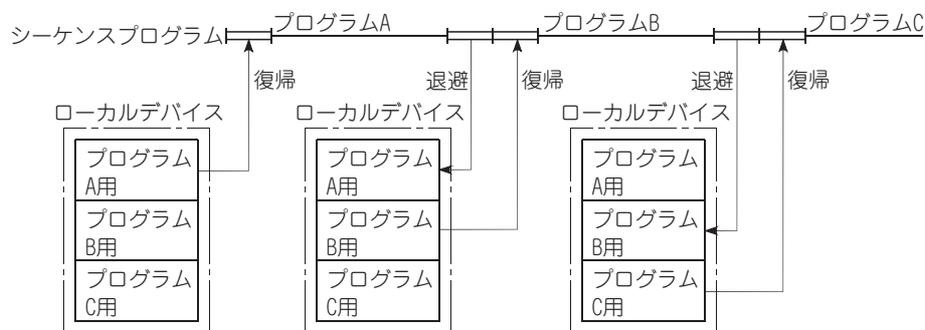


図 9.99 ローカルデバイスファイルの退避・復帰

備考

1. 命令によっては、ローカルデバイスを指定できない命令があります。詳細については、下記マニュアルの各命令の使用可能デバイスを参照してください。
 MELSEC-Q/L プログラミングマニュアル (共通命令編)
2. ローカルデバイスで使用するデバイスのワード数の考え方は、9.2 節を参照してください。

(3) ローカルデバイス使用時の設定

(a) ローカルデバイスとして使用するデバイスの範囲を設定

PC パラメータのデバイス設定でローカルデバイスとして使用する範囲を設定します。



図 9.100 デバイス設定

ただし、ローカルデバイスに設定した範囲のデバイスは、すべてのプログラムで共通であり、プログラムごとに設定範囲を変更できません。

例えば、M0 ~ 100 をローカルデバイスに設定した場合は、ローカルデバイスを使用するプログラムで M0 ~ 100 がローカルデバイスになります。

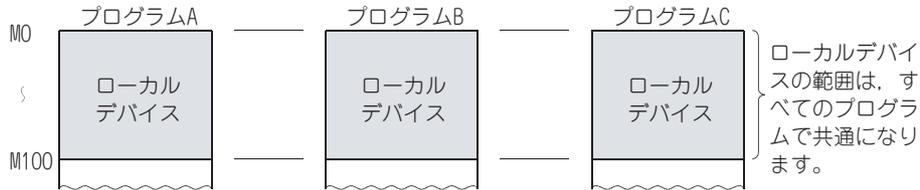


図 9.101 ローカルデバイスの範囲

(b) ローカルデバイスを格納するドライブとファイル名の設定

ローカルデバイスとして使用するデバイスの範囲を設定後、PCパラメータのPCファイル設定でローカルデバイス用のファイルを格納する対象メモリとファイル名を設定します。

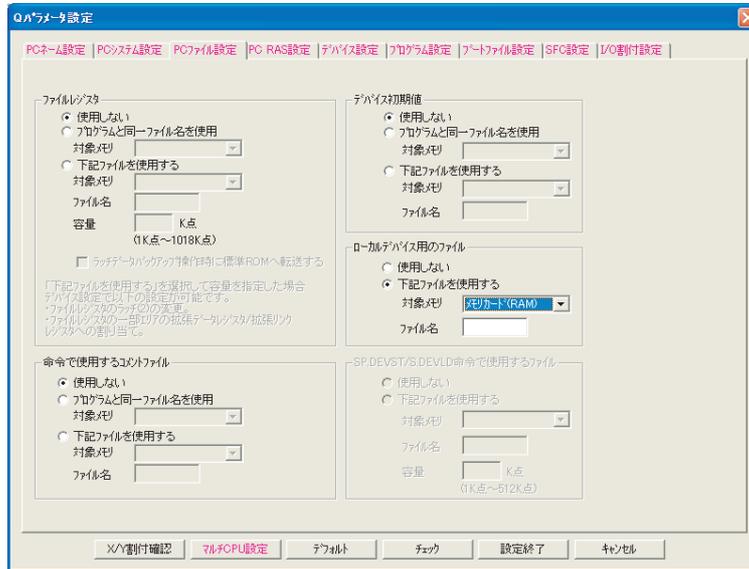


図 9.102 PC ファイル設定

(c) 設定した内容の書き込み

(a), (b) で設定した内容を、CPU ユニットに書き込みます。
GX Developer の [オンライン] → [PC 書込] で行います。

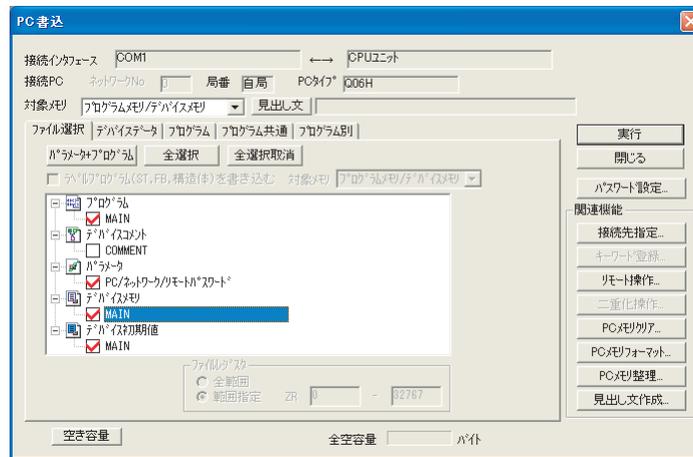


図 9.103 デバイスメモリの書き込み

9.13 デバイスの便利な使い方
9.13.2 ローカルデバイス

- 標準 RAM にサンプリングトレースファイルを格納している状態で、標準 RAM 内のローカルデバイスのサイズが変化する設定を行うと、サンプリングトレースファイルがクリアされます。トレース結果を残しておきたい場合は、下記の手順でトレース結果をパソコンに保存できます。
 - ① サンプリングトレース実行画面の **「トレース結果PC読出」** ボタンで、トレース結果をパソコンに読み出します。(参照 6.14 節 (5) (e))
 - ② **「トレース結果」** ボタンをクリックしてトレース結果を表示します。
 - ③ **「CSVファイル作成」** ボタンをクリックしてトレース結果を CSV 形式で保存します。
- ローカルデバイスの設定を行っていないデバイスは、すべてグローバルデバイスです。

(4) ローカルデバイス使用有無の設定 (プログラム単位での使用)

プログラム単位でのローカルデバイス使用可否の設定は、ローカルデバイスの使用可否をプログラム単位で設定し、スキャンタイムを短縮できる機能です。

また、ローカルデバイスを使用しないプログラムは、復帰・退避させるエリアを作成しないため、ローカルデバイスファイルのサイズを減らすこともできます。

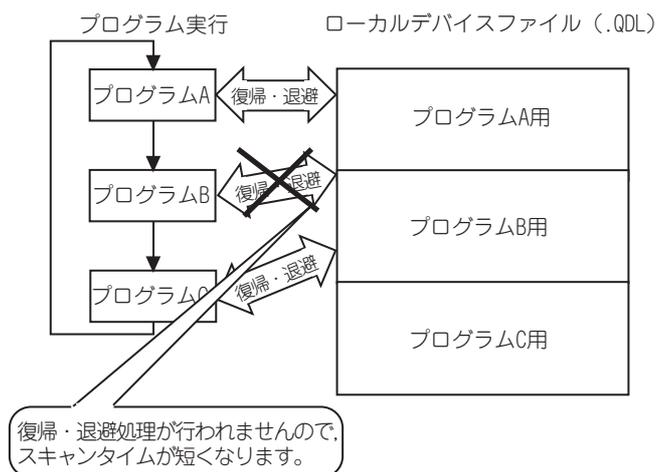


図 9.104 ローカルデバイスファイルの退避エリア構成

(a) 設定方法

本項 (3) の設定に追加して、下記の設定を行います。

PC パラメータのプログラム設定で “ファイル使用方法設定” を選択し、ローカルデバイスを使用するプログラムを設定します。

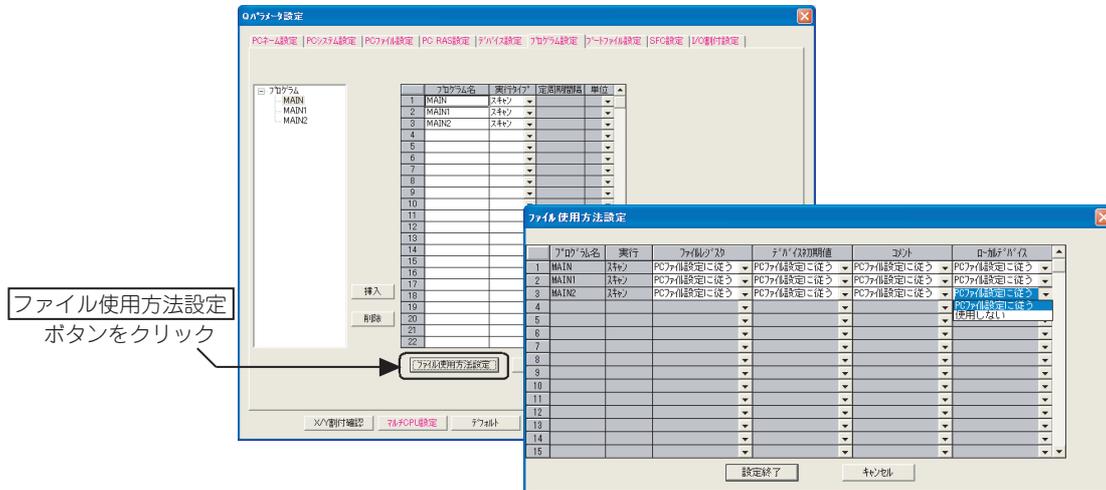


図 9.105 ファイル使用方法設定画面

(b) 注意事項

1) ローカルデバイスの変更

ローカルデバイスを “使用しない” に設定したプログラムでは、ローカルデバイスを参照、変更しないでください。

ローカルデバイスを使用しないプログラムでローカルデバイスを変更しても変更した値は保持されません。

2) ローカルデバイスファイルの作成条件

ローカルデバイスファイルを作成するかどうかは、PC パラメータの設定により決定します。

ローカルデバイスファイルの作成条件を表 9.20 に示します。

表 9.20 ローカルデバイスファイルの作成条件

PC パラメータ設定			ファイル作成 有無	エラー検出
PC ファイル設定	デバイス設定* 1	ファイル使用方法設定		
設定有り	設定有り	PC ファイル設定に従う	○	—
		使用しない	○	—
	設定無し	PC ファイル設定に従う	×	—
		使用しない	×	—
設定無し	設定有り	PC ファイル設定に従う	×	“FILE SET ERROR” (エラーコード：2400)
		使用しない	×	—
	設定無し	PC ファイル設定に従う	×	—
		使用しない	×	—

○：作成する，×：作成しない

* 1：デバイス設定でのローカルデバイスの範囲設定を示します。

9.13 デバイスの便利な使い方
9.13.2 ローカルデバイス

(5) サブルーチンプログラム格納先ファイルのローカルデバイスの使用

サブルーチンプログラム実行時、サブルーチンプログラムの格納先ファイルのローカルデバイスを使用できます。

サブルーチンプログラムの格納先ファイルのローカルデバイスの使用は、SM776 の ON/OFF で設定します。

表 9.21 特殊リレー（SM776）の ON/OFF によるローカルデバイスの切換え

SM776	動作
OFF	サブルーチンプログラムのコール元のファイルのローカルデバイスで演算を行う。
ON	サブルーチンプログラムの格納されているファイルのローカルデバイスで演算を行う。

(a) SM776:OFF 時の動作

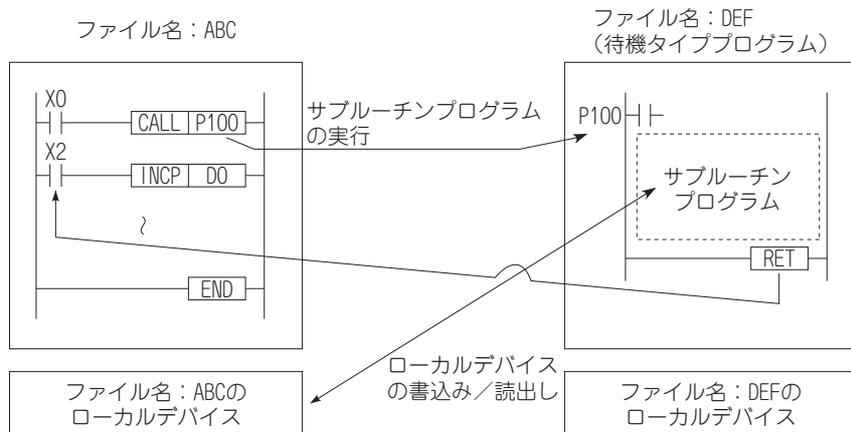


図 9.106 SM776 が OFF 時

(b) SM776:ON 時の動作

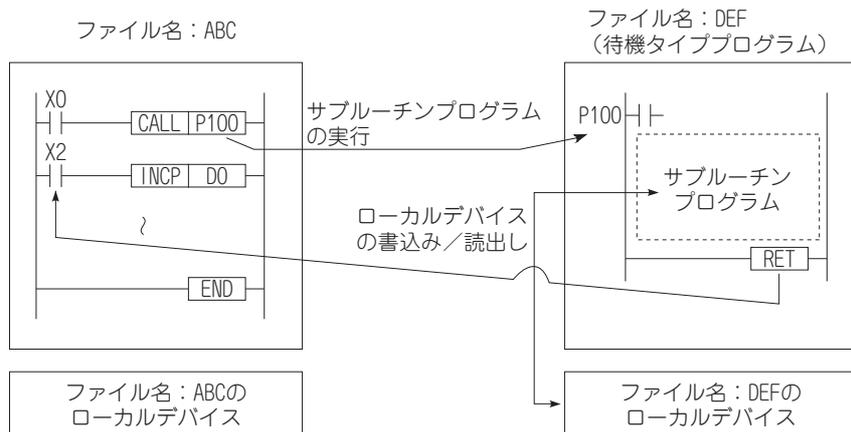


図 9.107 SM776 が ON 時

(c) 注意事項

- SM776 が ON の場合は、サブルーチンプログラムのコール時にローカルデバイスデータの読出しを行い、RET 命令実行後にローカルデバイスデータの退避を行います。
このため SM776 が ON の場合は、サブルーチンプログラムを 1 回実行するとスキャンタイムが延びます。
- SM776 の ON/OFF 設定は、CPU ユニット単位になります。
ファイル単位での設定はできません。
- SM776 の ON/OFF をシーケンスプログラムの実行中に変更した場合は、変更後の情報で制御を行います。

備考

SM776 の詳細については、下記マニュアルを参照してください。
☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編)

(6) 割り込み／定周期実行タイププログラム実行時のローカルデバイス

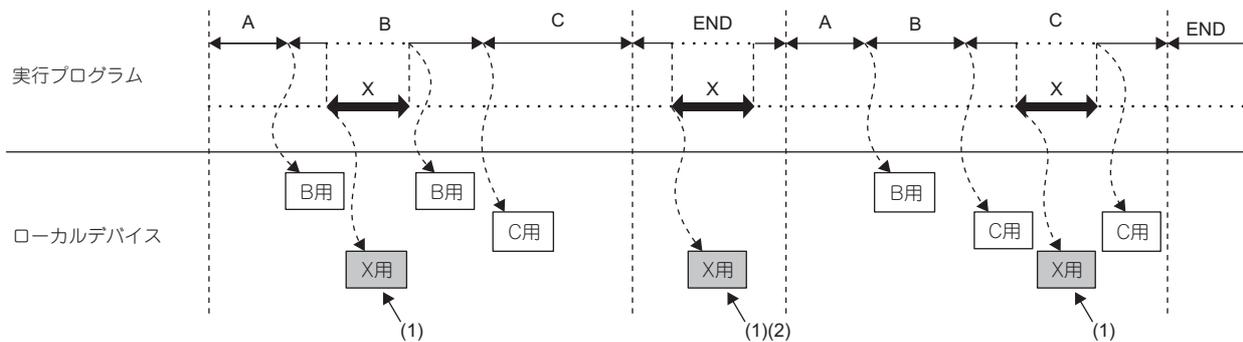
割り込み／定周期実行タイププログラムでローカルデバイスを使用する場合、SM777（割り込みプログラムにおけるローカルデバイスの可／不可設定）を ON に設定してください。SM777 を OFF に設定した場合は、正しく動作しません。* 1

* 1 ローカルデバイスに設定したインデックスレジスタは、SM777 の ON/OFF にかかわらず、割り込み／定周期実行タイププログラム実行前に実行していたプログラムのローカルデバイスを使用します。

例 下記の設定で SM777 を ON した場合の動作

表 9.22 各プログラムの実行タイプとローカルデバイス使用有無

プログラム名	実行タイプ	ローカルデバイス使用有無
A	スキャン	使用しない
B	スキャン	使用する
C	スキャン	使用する
X	定周期	使用する



(1) プログラム X のローカルデバイスを使用します。

(2) END 処理中に割り込み／定周期実行タイププログラムを実行すると、END 処理前に実行していたプログラム C のローカルデバイスの退避・読出しを実行するため、END 処理時間がその分延びます。

図 9.108 SM777 を ON した場合の動作

(a) 注意事項

- SM777 が ON の場合は、割込み／定周期実行タイププログラムの実行前にローカルデバイスデータの読み出しを行い、IRET 命令実行後にローカルデバイスデータの退避を行います。
このため SM777 が ON の場合は、割込み／定周期実行タイププログラムを 1 回実行するとスキャンタイムが延びます。
- SM777 の ON/OFF 設定は、CPU ユニット単位での設定になります。
ファイル単位での設定はできません。
- ローカルデバイスモニタ実行時、対象のローカルデバイスに切り替えてモニタを行います。したがって、SM777 が OFF の場合、切替え処理直後に割込みが発生し、ローカルデバイスをアクセスすると、ローカルデバイスモニタ対象先のローカルデバイスを使用します。(割込み発生前に実行していたプログラム (END 直前のプログラム) のローカルデバイスにはアクセスしません。)

備考

SM777 の詳細については、下記マニュアルを参照してください。
 QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編)

(7) ローカルデバイスデータのクリア

ローカルデバイスのデータは、下記のいずれかの方法でクリアします。

- CPU ユニットの電源 OFF → ON またはリセット時
- CPU ユニットの STOP 状態から RUN 状態に変化したとき

GX Developer などからローカルデバイスデータのクリアを行うことはできません。

第 10 章 CPU ユニットの処理時間

本章では、CPU ユニットの処理時間について説明します。

10.1 スキャンタイム

CPU ユニットのスキャンタイムの構成と処理時間について説明します。

10.1.1 スキャンタイムの構成

CPU ユニットのスキャンタイムの構成を示します。

CPU ユニットの RUN 状態時に下記の処理を順次行います。

(1) ベーシックモデル QCPU のスキャンタイムの構成

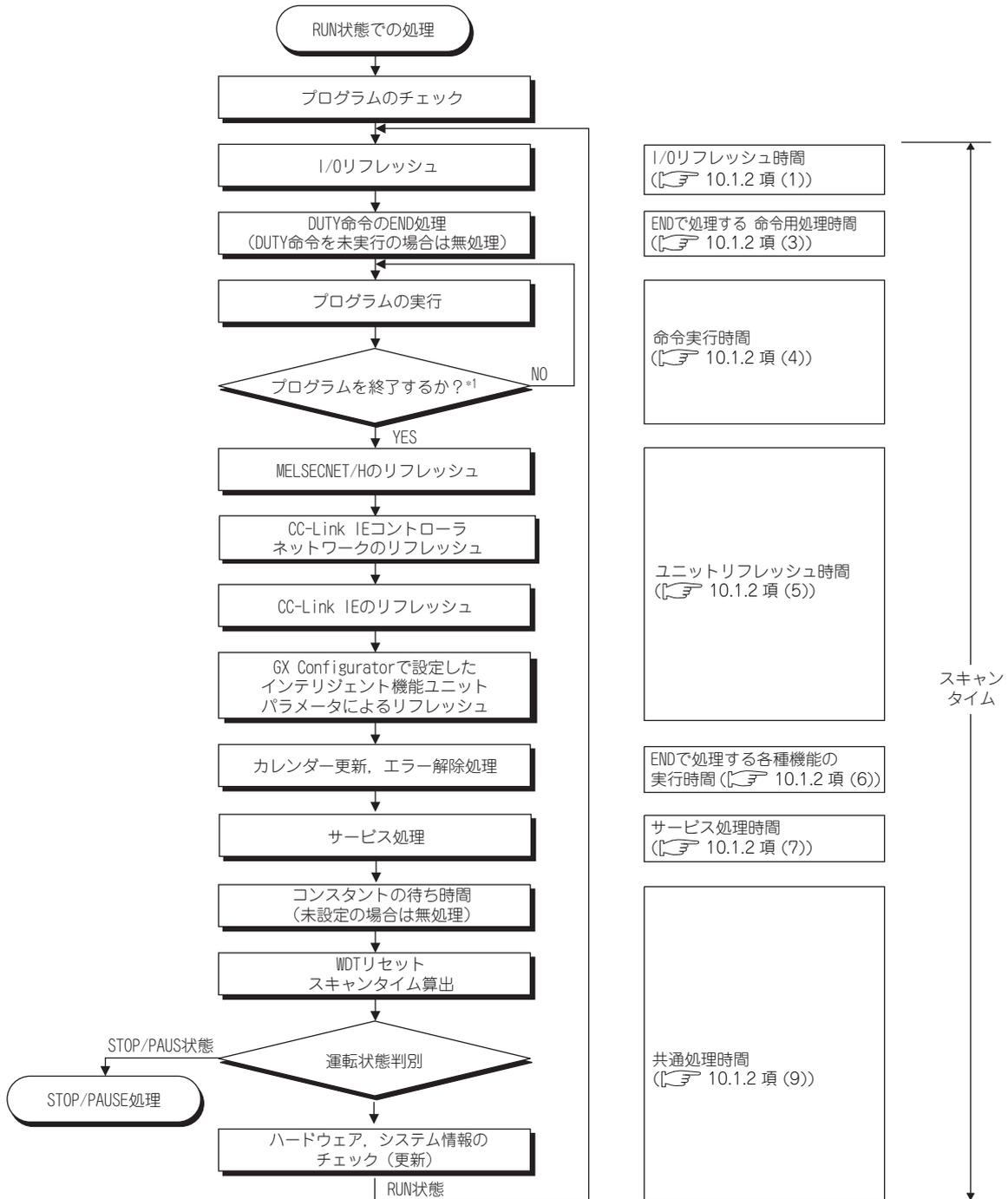


図 10.1 ベーシックモデル QCPU のスキャンタイムの構成

* 1 : プログラムの終了とは, END,GOEND,FEND,STOP 命令を実行したタイミングを示します。

10.1 スキャンタイム
10.1.1 スキャンタイムの構成

(2) ハイパフォーマンスモデル QCPU, プロセス CPU のスキャンタイムの構成

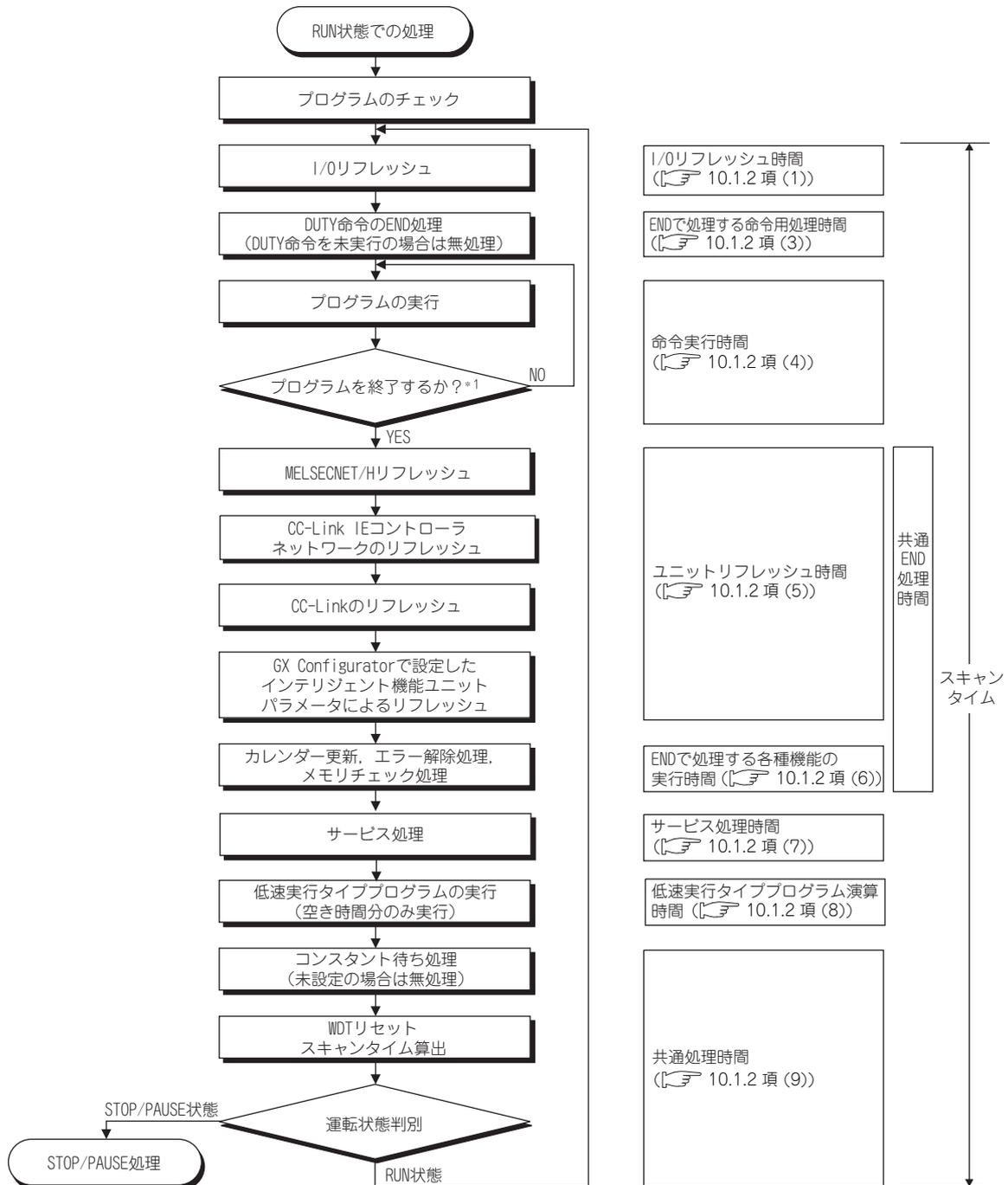


図 10.2 ハイパフォーマンスモデル QCPU, プロセス CPU のスキャンタイムの構成

* 1: プログラムの終了とは, END, GOEND, FEND, STOP 命令を実行したタイミングを示します。

(3) 二重化 CPU のスキャンタイムの構成

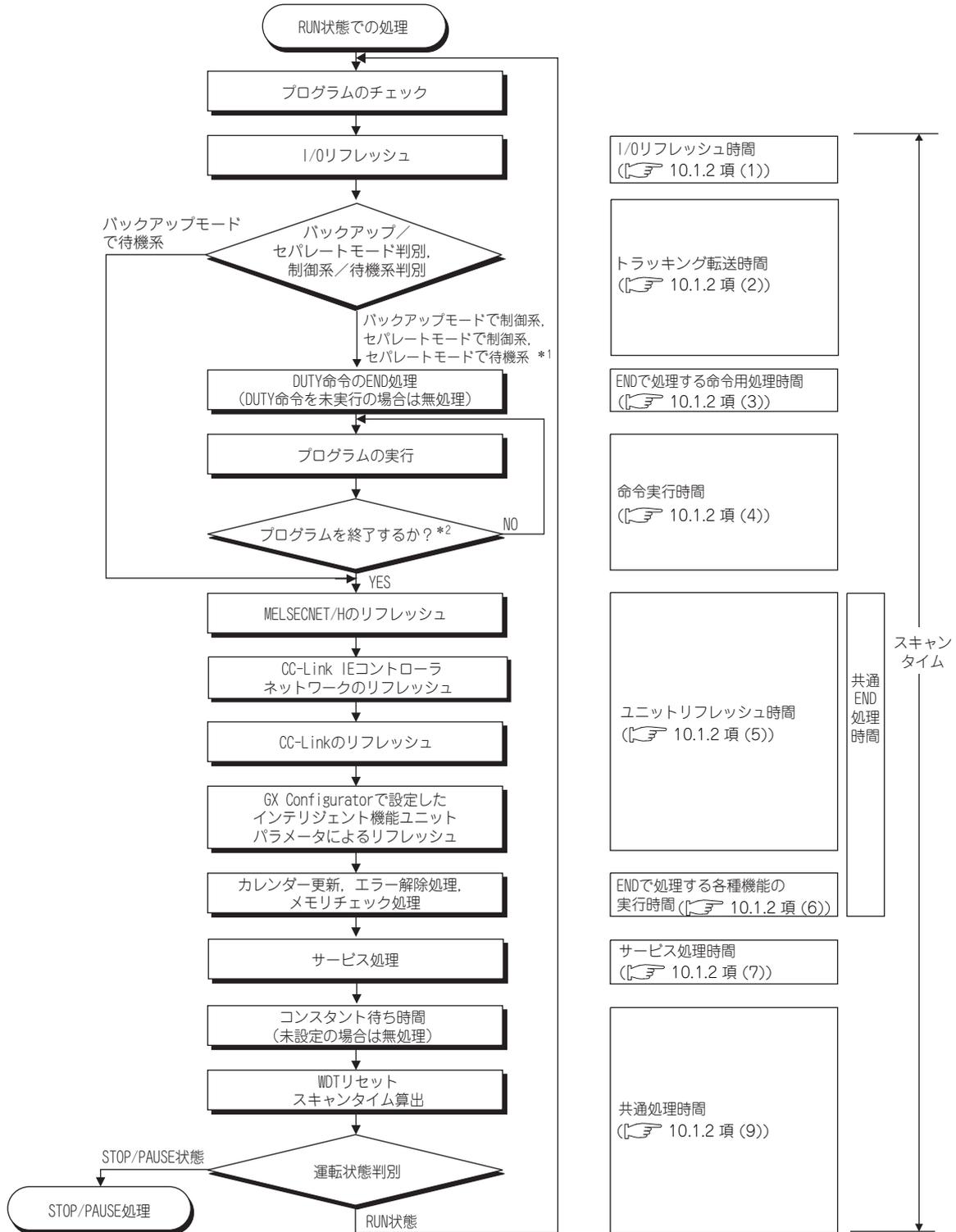


図 10.3 二重化 CPU のスキャンタイムの構成

- * 1 : バックアップモードからセパレートモードに変更したとき、待機系 (RUN LED 点滅) では、プログラムの実行はできません。
- * 2 : プログラムの終了とは、END,GOEND,FEND,STOP 命令を実行したタイミングを示します

(4) スキャンタイムの確認方法

CPU ユニットは、スキャンタイムの現在値、最小値、最大値を計測しています。

特殊レジスタ (SD520, SD521, SD524 ~ 527) をモニターすることにより、スキャンタイムを確認できます。

格納される各スキャンタイムの精度は、 $\pm 0.1\text{ms}$ です。

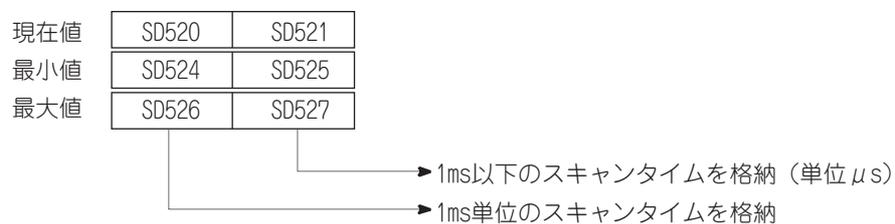


図 10.4 スキャンタイムの格納場所

例 SD520 が 3, SD521 が 400 の場合、スキャンタイムは 3.4ms になります。

10.1.2 スキャンタイムに関連する要素の処理時間

本項では、10.1.1 項に示した処理および実行時間の算出方法について説明します。

(1) I/O リフレッシュ時間

I/O リフレッシュ時間は、基本ベースユニット、増設ベースユニットに装着されている下記ユニットとの、入出力データのリフレッシュ時間です。

- 入力ユニット
- 出力ユニット
- インテリジェント機能ユニット

■ 算出方法

I/O リフレッシュ時間は、下式で算出します。

$$(I/O \text{ リフレッシュ時間}) = (\text{入力点数} / 16) \times N1 + (\text{出力点数} / 16) \times N2$$

N1, N2 は表 10.1 を参照してください。

表 10.1 I/O リフレッシュ時間

CPU ユニット	Q3 □ B, Q3 □ SB, Q3 □ RB, Q3 □ DB		Q5 □ B, Q6 □ B, Q6 □ RB		Q6 □ WRB		QA1S5 □ B, QA1S6 □ B		QA6 □ B, QA6ADP + A5 □ B, QA6ADP + A6 □ B	
	N1	N2	N1	N2	N1	N2	N1	N2	N1	N2
Q00JCPU	2.05 μs	1.25 μs	2.95 μs	2.20 μs	—	—	—	—	—	—
Q00CPU	2.00 μs	1.20 μs	2.75 μs	2.05 μs						
Q01CPU	1.95 μs	1.15 μs	2.70 μs	2.00 μs						
Q02CPU	2.2 μs	1.3 μs	2.9 μs	2.1 μs	—	—	4.9 μs	3.9 μs	5.6 μs	4.8 μs
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	1.7 μs	1.3 μs	2.4 μs	2.1 μs	—	—	4.3 μs	3.9 μs	5.0 μs	4.8 μs
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU							—	—	—	—
Q12PRHCPU, Q25PRHCPU							2.4 μs	2.1 μs	—	—

備考

使用可能なベースユニットは、CPU ユニットごとに異なります。

☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編)

(2) トラッキング処理時間

二重化 CPU のトラッキング機能使用時に要する時間です。
トラッキング処理時間については、下記マニュアルを参照してください。
 QnPRHCPU ユーザーズマニュアル（二重化システム編）

(3) DUTY 命令の END 処理時間

DUTY 命令を使用しているときの END 処理の延び時間です。
DUTY 命令で指定されているユーザタイミグクロック（SM420～424, SM430～434）を、END 処理で ON/OFF します。  **注 10.1**

表 10.2 DUTY 命令の END 処理時間

CPU ユニット	END 処理の延び時間	
	設定数 1 の場合	設定数 5 の場合
Q00JCPU	0.15 ms	0.21 ms
Q00CPU	0.14 ms	0.19 ms
Q01CPU	0.12 ms	0.16 ms
Q02CPU	0.02 ms	0.02 ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.01 ms	0.01 ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

(4) 命令実行時間

命令実行時間は、CPU ユニットで実行するプログラムで使用している各命令の処理時間の合計です。

備考

各命令の処理時間については、下記マニュアルを参照してください。
 MELSEC-Q/L プログラミングマニュアル（共通命令編）

注 10.1 **Basic**

ベーシックモデル QCPU では、SM430～434 が使用できません。

(a) 割込み／定周期実行タイププログラム実行時のオーバーヘッド時間

命令実行時間算出時は、本項 (4) の命令実行時間に下記に示すオーバーヘッド時間を加算してください。

割込みプログラムには、起動前オーバーヘッド時間と、プログラム終了オーバーヘッド時間があります。

表 10.3 割込みプログラムの起動前のオーバーヘッド時間

CPU ユニット	定周期割込み (I28 ~ 31)		QI60 からの割込み (I0 ~ 15) * 1 / インテリジェント機能ユニットからの割込み (I50 ~ 127)	
	高速起動なし	高速起動あり	高速起動なし	高速起動あり
Q00JCPU	175 μ s	150 μ s	350 μ s	325 μ s
Q00CPU	145 μ s	125 μ s	285 μ s	265 μ s
Q01CPU	135 μ s	120 μ s	270 μ s	255 μ s
Q02CPU	190 μ s	85 μ s	205 μ s	100 μ s
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	85 μ s	40 μ s	90 μ s	45 μ s
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU	85 μ s	40 μ s	90 μ s	45 μ s
Q12PRHCPU, Q25PRHCPU (増設ベースユニット未接続時)	85 μ s	40 μ s	90 μ s	45 μ s
Q12PRHCPU, Q25PRHCPU (増設ベースユニット接続時)	85 μ s	40 μ s	1090 μ s * 2	1045 μ s * 2

* 1 : QI60 が基本ベースユニット上のスロット 0 に装着されている場合の値を示しています。

* 2 : 増設ベースユニット接続時に QI60 は使用できません。表内の数値はインテリジェント機能ユニットからの割込み処理時のオーバーヘッド時間です。

表 10.4 割込みプログラムの終了オーバーヘッド時間

CPU ユニット	高速起動なし	高速起動あり
Q00JCPU	175 μ s	150 μ s
Q00CPU	145 μ s	125 μ s
Q01CPU	135 μ s	120 μ s
Q02CPU	180 μ s	75 μ s
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	80 μ s	35 μ s
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

表 10.5 定周期実行タイププログラムのオーバーヘッド時間

CPU ユニット	高速起動なし	高速起動あり
Q02CPU	380 μ s	230 μ s
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	165 μ s	100 μ s
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

1) 割込みプログラム内のローカルデバイスを使用可能にした場合のオーバーヘッド時間

SM777（割込みプログラムにおけるローカルデバイスの可／不可設定）を ON にした場合、表 10.6、表 10.7 に示す時間が表 10.3 と表 10.4 のオーバーヘッド時間に加算されます。

表中の n, N1, N2, N3 は、以下を表します。

- n：ローカルデバイス点数（単位：kワード）
- N1：ローカルデバイスを指定したデバイス種別の数
- N2：ローカルデバイスを指定したワードデバイス点数
- N3：ローカルデバイスを指定したビットデバイス点数

表 10.6 標準 RAM 内のローカルデバイスファイル使用時

CPU ユニット	割込みプログラム起動前 (表 10.3) の オーバーヘッド時間への加算値	割込みプログラム終了 (表 10.4) の オーバーヘッド時間への加算値
Q02CPU	$(0.35 \times n + 0.05) \times 10^3 \mu s$	$(0.35 \times n + 0.05) \times 10^3 \mu s$
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	$(0.15 \times n + 0.03) \times 10^3 \mu s$	$(0.15 \times n + 0.03) \times 10^3 \mu s$
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

表 10.7 SRAM カード内のローカルデバイスファイル使用時

CPU ユニット	割込みプログラム起動前 (表 10.3) の オーバーヘッド時間への加算値	割込みプログラム終了 (表 10.4) の オーバーヘッド時間への加算値
Q02CPU	$(1.15 \times n + 0.30) \times 10^3 \mu s$	$(1.15 \times n + 0.30) \times 10^3 \mu s$
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	$(0.85 \times n + 0.15) \times 10^3 \mu s$	$(0.85 \times n + 0.15) \times 10^3 \mu s$
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

(5) ユニットリフレッシュ時間

ユニットリフレッシュ時間は、CC-Link IE コントローラネットワーク、MELSECNET/H および CC-Link などによるリフレッシュ時間の合計です。

(a) CC-Link IE コントローラネットワークのリフレッシュ

CC-Link IE コントローラネットワークユニットのリンクデバイスと、CPU ユニットのデバイス間でデータをリフレッシュする時間です。

(b) MELSECNET/H のリフレッシュ

MELSECNET/H ネットワークユニットのリンクデバイスと、CPU ユニットのデバイス間でデータをリフレッシュする時間です。

(c) CC-Link の自動リフレッシュ

CC-Link のマスタ・ローカルユニットと、CPU ユニット間でデータをリフレッシュする時間です。

備考

各リフレッシュ時間については、下記マニュアルを参照してください。

☞ 各ネットワークユニットのマニュアル

(d) インテリジェント機能ユニットの自動リフレッシュ

インテリジェントユニットのバッファメモリと、CPUユニットのデバイス間でデータをリフレッシュする時間です。

自動リフレッシュ設定は、インテリジェント機能ユニット用ユーティリティパッケージ（GX Configurator）で行います。

■ 算出方法

インテリ自動リフレッシュ時間は、下記計算式により計算します。

$$(\text{リフレッシュ時間}) = \text{KN1} + \text{KN2} \times (\text{リフレッシュ点数})$$

KN1, KN2 は表 10.8, 表 10.9 の値を使用します。

表 10.8 基本ベースユニットにインテリジェントユニットを装着時

CPU ユニット	KN1	KN2
Q00JCPU	115 μ s	55 μ s
Q00CPU	91 μ s	46 μ s
Q01CPU	85 μ s	41 μ s
Q02CPU	53 μ s	13 μ s
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	27 μ s	6 μ s
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

表 10.9 増設ベースユニットにインテリジェントユニットを装着時

CPU ユニット	KN1	KN2
Q00JCPU	120 μ s	56 μ s
Q00CPU	92 μ s	48 μ s
Q01CPU	86 μ s	43 μ s
Q02CPU	61 μ s	15 μ s
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	29 μ s	8 μ s
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

例 アナログ→デジタル変換ユニット（Q64AD）に対する自動リフレッシュ点数が4点の場合（Q01CPUの基本ベースユニットに装着時）

$$0.249(\text{ms}) = 0.085 + 0.041 \times 4$$

(6) END で処理する各種機能の実行時間

カレンダー更新, エラー解除の処理, メモリチェック処理による END 処理の延び時間です。

(a) カレンダー更新処理時間

時計データセット要求 (SM210 が OFF から ON に変化), または時計データ読出し要求 (SM213 が ON) を行った際に, END 処理で時計データを変更/読み出すための時間です。

表 10.10 カレンダー更新処理時間

CPU ユニット	END 処理の延び時間	
	時計データセット要求時	時計データ読出し要求あり
Q00JCPU	1.25 ms	0.04 ms
Q00CPU	0.99 ms	0.03 ms
Q01CPU	0.98 ms	0.02 ms
Q02CPU	0.26 ms	0.01 ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.11 ms	0.005 ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

(b) エラー解除処理時間

SM50 (エラー解除) の立上り (OFF から ON に変化) の際に, SD50 に格納されている続行エラーのエラーを解除する時間です。

表 10.11 エラー解除処理時間

CPU ユニット	END 処理の延び時間	
	アナンシェータの解除	その他のエラー解除
Q00JCPU	2.1 ms	2.0 ms
Q00CPU	1.75 ms	1.7 ms
Q01CPU	1.45 ms	1.35 ms
Q02CPU	1.15 ms	0.41 ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.84 ms	0.21 ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

(c) メモリチェック処理時間

PC パラメータの PC RAS 設定 (2) で “プログラムメモリのチェックを行う” に設定している場合の時間です。

【メモリチェックに必要な時間】

$$= \frac{1 \text{ スキャンごとのチェック容量 (ステップ)} * 1}{1024} \times 3.5 \text{ ms}$$

* 1 : PC パラメータの PC RAS 設定 (2) で設定したステップを示します。(☞ 8.1.2 項 (5))

10.1 スキャンタイム
10.1.2 スキャンタイムに関連する要素の処理時間

(7) サービス処理時間

サービス処理とは、GX Developer および外部機器との通信処理です。

(a) ベーシックモデル QCPU の場合

GX Developer でモニタを行う場合の処理時間を表 10.12 に示します。

表 10.12 GX Developer によるモニタ処理時間

機能	自局 CPU ユニットの RS-232 に接続時			他局接続時* 4		
	Q00JCPU	Q00CPU	Q01CPU	Q00JCPU	Q00CPU	Q01CPU
プログラムの PC 読出* 1	1.6 ms	1.3 ms	1.2 ms	2.3 ms	1.9 ms	1.8 ms
デバイスモニタ* 2	1.2 ms	1.0 ms	0.9 ms	2.4 ms	2.0 ms	1.9 ms
RUN 中書込み* 3	1.0 ms	1.0 ms	1.0 ms	1.9 ms	1.6 ms	1.5 ms

* 1：プログラムメモリから 8k ステップのプログラムを読み出すときの時間です。

* 2：登録モニタで 32 点を設定したときの時間です。

* 3：100 ステップの回路を追加したときの時間です。

* 4：MELSECNET/H, Ethernet, CC-Link, シリアルコミュニケーションユニット経由でアクセスしたときの時間です。

シリアルコミュニケーションユニット、および Ethernet ユニットとの通信時間については、下記マニュアルを参照してください。

 MELSEC コミュニケーションプロトコルリファレンスマニュアル

(b) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU の場合

GX Developer でモニタ、プログラム読出し、モニタ条件の設定を行う場合の処理時間を表 10.13、表 10.14 に示します。

表 10.13 デバイス登録モニタ、プログラム読出しでの処理時間

CPU ユニット	処理時間
Q02CPU	0.017 ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.011 ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU	
Q12PRHCPU, Q25PRHCPU	

* 1：サービス処理回数を 1 回に設定した場合の処理時間です。

表 10.14 モニタ条件を設定した場合

CPU ユニット	処理時間	
	指定ステップ一致	指定デバイス一致
Q02CPU	0.05 ms	0.01 ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.03 ms	0.01 ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

(8) 低速プログラム演算時間

CPU ユニットで実行する低速実行タイププログラムで使用している各命令の処理時間の合計です。

備考

各命令の処理時間については、下記マニュアルを参照してください。

☞ MELSEC-Q/L プログラミングマニュアル（共通命令編）

(9) 共通処理時間

システムで処理する CPU ユニットの共通処理時間を表 10.15 に示します。

表 10.15 共通処理時間

CPU ユニット	共通処理時間
Q00JCPU	0.66 ms
Q00CPU	0.60 ms
Q01CPU	0.52 ms
Q02CPU	0.40 ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.16 ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU	
Q12PRHCPU, Q25PRHCPU	0.5 ms

10.1.3 スキャンタイムを延ばす要因

本項に示す機能または操作を行う場合は、10.1.2 項で算出した値に本項で示す値を加算してください。

(1) サンプリグトレース

サンプリグトレース (図 6.14 節) を実行すると、表 10.16 に示す処理時間が発生します。

表 10.16 ビットデバイスに内部リレー 50 点、ワードデバイスにデータレジスタ 50 点をサンプリグトレースデータとして設定した場合の処理時間

CPU ユニット	処理時間	
	標準 RAM	メモリカード
Q02CPU	0.16 ms	0.24 ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.06 ms	0.12 ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU,		
Q12PRHCPU, Q25PRHCPU		

(2) ローカルデバイスの使用

ローカルデバイスを使用すると、表 10.17 に示す処理時間が発生します。

表中の n はプログラム実行本数を表します。

表 10.17 ローカルデバイス設定した場合の処理時間

CPU ユニット	処理時間	
	標準 RAM	メモリカード
Q02CPU	$3.52 \times n \times 10^3 \mu s * 1$	$10.73 \times n \times 10^3 \mu s * 1$
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	$1.54 \times n \times 10^3 \mu s * 1$	$8.16 \times n \times 10^3 \mu s * 1$
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU,		
Q12PRHCPU, Q25PRHCPU		

* 1 : ローカルデバイスが 10k 点時の処理時間となります。

(a) サブルーチンプログラム内のローカルデバイスを使用可能にした場合のスキャンタイムの延び時間

SM776 (CALL 時におけるローカルデバイスの可/不可設定) を ON にした場合、サブルーチンプログラム
の CALL1 回あたり、表 10.18、表 10.19 に示す処理時間が発生します。

表中の n はローカルデバイス点数 (単位:k ワード) を表します。

表 10.18 標準 RAM 内のローカルデバイスファイル使用時

CPU ユニット	同一ファイル内のサブルーチン コール時のスキャンタイム延び時間	別ファイル内のサブルーチン コール時のスキャンタイム延び時間
Q02CPU	$(0.35 \times n + 0.05) \times 10^3 \mu s$	$(0.70 \times n + 0.10) \times 10^3 \mu s$
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	$(0.15 \times n + 0.03) \times 10^3 \mu s$	$(0.30 \times n + 0.05) \times 10^3 \mu s$
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU,		
Q12PRHCPU, Q25PRHCPU		

表 10.19 SRAM カード内のローカルデバイスファイル使用時

CPU ユニット	同一ファイル内のサブルーチン コール時のスキャンタイム延び時間	別ファイル内のサブルーチン コール時のスキャンタイム延び時間
Q02CPU	$(1.15 \times n + 0.30) \times 10^3 \mu s$	$(2.30 \times n + 0.60) \times 10^3 \mu s$
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	$(0.85 \times n + 0.15) \times 10^3 \mu s$	$(1.65 \times n + 0.30) \times 10^3 \mu s$
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU,		
Q12PRHCPU, Q25PRHCPU		

(3) 複数プログラムの実行

複数のプログラムを実行すると、表 10.20 に示す処理時間が発生します。

表 10.20 複数プログラム実行時の各プログラムの処理時間

CPU ユニット	処理時間
Q02CPU	$0.08 \times n$ ms * 1
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	$0.03 \times n$ ms * 1
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU	
Q12PRHCPU, Q25PRHCPU	

* 1 : n はプログラムファイル数

(4) メモリカードの着脱

メモリカードの着脱を行うと、メモリカードの着脱を行った 1 スキャンでのみ表 10.21 に示す処理時間が発生します。

表 10.21 メモリカード着脱時の処理時間

CPU ユニット	処理時間	
	メモリカード挿入時	メモリカード抜取り時
Q02CPU	0.16 ms	0.10 ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.08 ms	0.04 ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

(5) ファイルレジスタの使用

PC パラメータの PC ファイル設定で “プログラムと同一ファイルを使用” を選択している場合、表 10.22 に示す処理時間が発生します。

“下記ファイルを使用する” を選択している場合は、スキャンタイムは延びません。

表 10.22 ファイルレジスタ使用時の処理時間

CPU ユニット	処理時間	
	標準 RAM	メモリカード
Q02CPU	1.03 ms	$1.14 \times n$ ms * 1
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.41 ms	$0.50 \times n$ ms * 1
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

* 1 : n はプログラムファイル数

(6) RUN 中書込みの実行

RUN 中書込みを行うと、下記に示す処理時間が発生します。

(a) 回路モードでの RUN 中書込み時

回路モードでの RUN 中書込みを行うと、表 10.23 に示す処理時間が発生します。

表 10.23 回路モードでの RUN 中書込みを行ったときの延び時間

CPU ユニット	RUN 中書込用確保ステップ	
	変更無	再設定
Q00JCPU	最大 2.1 ms	最大 30 ms
Q00CPU	最大 1.7 ms	最大 26 ms
Q01CPU	最大 1.7 ms	最大 36 ms
Q02CPU	最大 4 ms (最大 10 ms * 1)	最大 30 ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	最大 2 ms (最大 4 ms * 1)	最大 90 ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

* 1 : RUN 中書込みの書込み範囲内にローカルポインタがある場合の値を示しています。

(b) ファイルの RUN 中書込み時

ファイルの RUN 中書込みを行うと、スキャンタイムが表 10.24 に示す処理時間が発生します。

ATA カード使用時は、30k ステップで 1.25s スキャンタイムが延びます。

表 10.24 ファイルの RUN 中書込み時の処理時間

CPU ユニット	プログラムメモリに空きエリアが確保できる場合	メモ리카ードに空きエリアが確保できる場合 (ATA カードを除く)
Q02CPU	最大 90 ms (最大 200 ms * 2)	最大 150 ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	最大 350 ms (最大 650 ms * 2)	最大 650 ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU		
Q12PRHCPU, Q25PRHCPU		

* 2 : SFC プログラムの場合の値を示しています。

(7) グループ外の出力状態を取込み

マルチ CPU システム構成時、PC パラメータのマルチ CPU 設定で “グループ外の出力状態を取り込む” に設定した場合は、スキャンタイムが延びます。

チェックをすると、スキャンタイムが延びます。

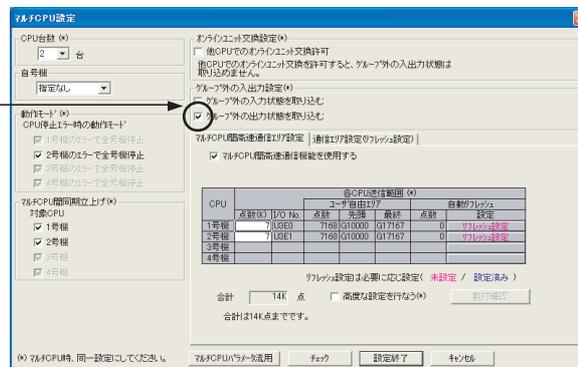


図 10.5 マルチ CPU 設定画面

10.1 スキャンタイム
10.1.3 スキャンタイムを延ばす要因

(8) ベーシックモデル QCPU 使用時にのみスキャンタイムが延びる機能

下記に示す機能を使用すると、スキャンタイムが延びます。

- ・ システムモニタ
- ・ バッテリチェック
- ・ ヒューズ断チェック
- ・ ユニット照合
- ・ 一般データ処理

増設ベースユニット 1 段にインテリジェント機能ユニットを計 12 枚装着し、システムモニタを実行した場合のスキャンタイムの延びを示します。

表 10.25 システムモニタ使用時のスキャンタイムの延び (インテリジェント機能ユニットを計 12 枚装着時)

CPU ユニット	スキャンタイムの延び
Q00JCPU	0.036ms
Q00CPU	0.015ms
Q01CPU	0.011ms

(9) メモリチェック機能の実行

CPU ユニットでメモリチェック機能を実行すると、メモリチェック機能の処理時間分スキャンタイムが延びます。メモリチェック機能の処理時間については、6.27 節 (4) を参照ください。

(10) スキャンタイム測定

GX Developer でスキャンタイム測定 (☞ 6.13.3 項) を実行すると、表 10.26 に示す処理時間が発生します。

表 10.26 GX Developer で CPU ユニットのスキャンタイム測定を実行した場合の処理時間

CPU ユニット	処理時間
Q02CPU	$120.0 + 16.5 \times \text{分岐命令数} \mu\text{s} * 1$
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	$52.0 + 7.0 \times \text{分岐命令数} \mu\text{s} * 1$
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU	
Q12PRHCPU, Q25PRHCPU	

* 1 : 分岐命令数は、スキャンタイム測定の開始位置から終了位置までの間で実行される下記ポインタ分岐命令、サブルーチンコール命令の数を示しています。

- ・ ポインタ分岐命令 : CJ, SCJ, JMP
- ・ サブルーチンコール命令 : CALL(P), FCALL(P), ECALL(P), EFCALL(P), XCALL(P), RET

10.1.4 設定変更でスキャンタイムを短くできる要因

パラメータ設定を本項に示すように変更するとスキャンタイムを短縮できます。

(1) A シリーズ CPU 互換設定 (☞ 8.1.2 項 (2)) 注 10.2

PC パラメータの PC システム設定で、“SM1000, SD1000 以降の特殊リレー／特殊レジスタを使用しない” に設定する（設定画面のチェックをはずす）と、表 10.27 に示す処理時間分のスキャンタイムを短縮できます。

表 10.27 A シリーズ CPU 互換設定による処理時間の違い

CPU ユニット	処理時間
Q02CPU	0.07ms
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.03ms
Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU	



図 10.6 A シリーズ CPU 互換設定

Point

“SM1000, SD1000 以降の特殊リレー／特殊レジスタを使用しない” に設定した場合は、下記に示す A シリーズ互換特殊リレー／特殊レジスタを Q シリーズ専用特殊リレー／特殊レジスタに置き換えてください。

- A シリーズ互換特殊リレー (SM1000 ~ 1299)
→ Q シリーズ専用特殊リレー (SM0 ~ 999)
- A シリーズ互換特殊レジスタ (SD1000 ~ 1299)
→ Q シリーズ専用特殊レジスタ (SD0 ~ 999)

注 10.2 Basic Process Redundant

ベーシックモデル QCPU, プロセス CPU, 二重化 CPU では、A シリーズ CPU 互換設定はできません。

(2) 浮動小数点演算処理 (☞ 8.1.2 項 (2)) 注 10.3

PC パラメータの PC システム設定で、“内部演算処理を倍精度で行わない”に設定する（設定画面のチェックをはずす）と、浮動小数点を使用した命令の演算処理に要する時間が短縮されます。



図 10.7 浮動小数点演算処理

備考

浮動小数点を使用した命令の演算処理時間については、下記マニュアルを参照してください。
☞ QCPU プログラミングマニュアル（共通命令編）



注 10.3

Basic

Process

Redundant

ベーシックモデル QCPU、プロセス CPU、二重化 CPU では、浮動小数点演算処理の設定はできません。

(3) ファイル使用方法設定 (👉 8.1.2 項 (7)) 注 10.4

ファイルレジスタ、デバイス初期値、またはデバイスコメントを使用しないプログラムでは、ファイル使用方法設定で“使用しない”に設定することにより、プログラムのオーバヘッド時間が短縮されます。

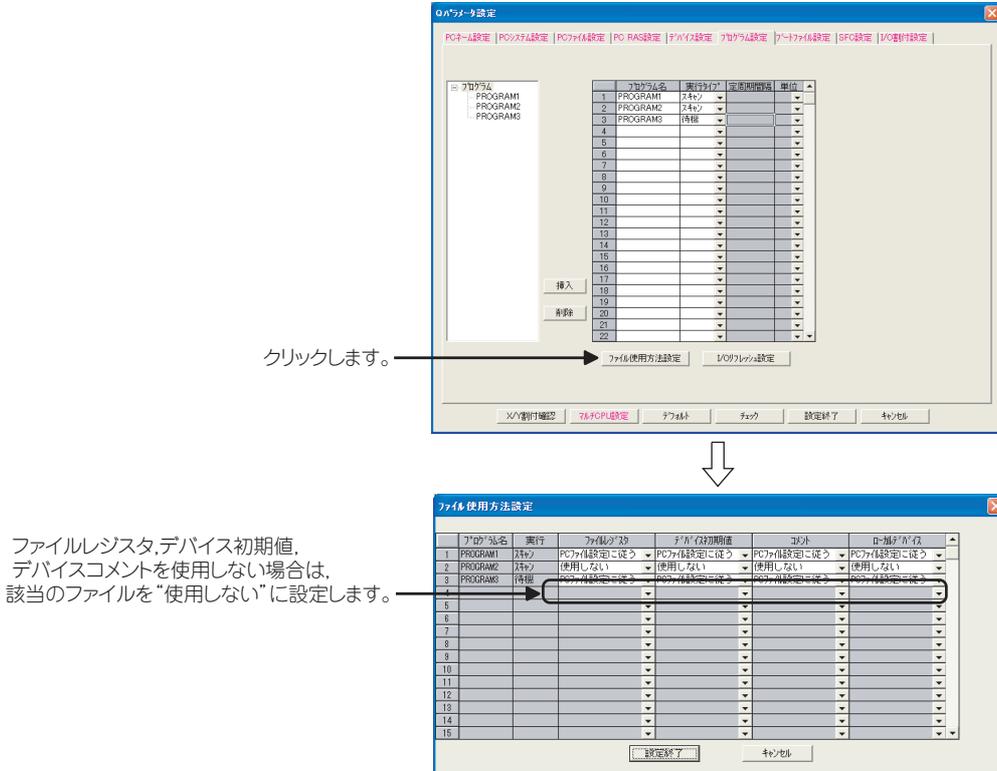


図 10.8 ファイル使用方法設定画面

Point

PC パラメータの PC ファイル設定で、“プログラムと同一ファイル名を使用”に設定している場合にのみ有効です。

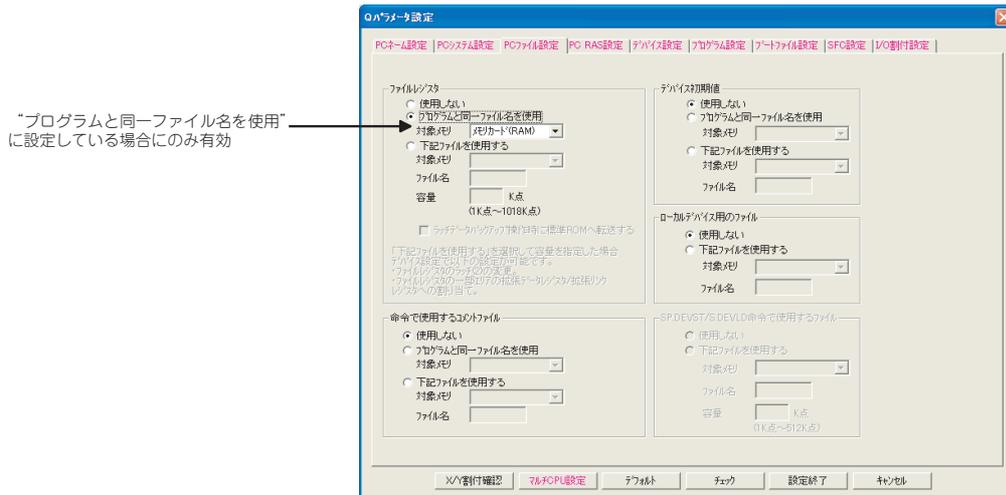


図 10.9 PC ファイル設定

10.1 スキャンタイム
10.1.4 設定変更でスキャンタイムを短くできる要因

注 10.4 Basic

ベーシックモデル QCPU では、ファイル使用方法設定は使用できません。

第 11 章 プログラムを CPU ユニットに書き込むまでの手順

GX Developer で作成したプログラムを CPU ユニットに書き込むまでの手順について説明します。

Point

GOT やネットワークユニットを経由してプログラミングツールと CPU ユニットの通信を行う場合、PC タイプが異なる状態でも接続できてしまうことがあります。
異なる形名で接続した場合、データの書込み／読出しが正常にできないことがあります。通信を行う際は、必ず PC タイプを確認してください。

備考

CPU ユニットの立上げ手順については、下記マニュアルを参照してください。

☞ QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）

11.1 ベーシックモデル QCPU

11.1.1 プログラムを作成する場合の検討事項

プログラムを作成する場合、各々のプログラムで使用するプログラム容量、使用デバイス点数、ファイル名などをあらかじめ決めておく必要があります。

(1) プログラム容量の検討

プログラムとパラメータの合計容量が、使用する CPU ユニットで実行できるプログラム容量を越えていないか検討します。（☞ 5.4.3 項）

各 CPU ユニットで実行できるプログラム容量を表 11.1 に示します。

表 11.1 ベーシックモデル QCPU のプログラム容量

CPU ユニット	プログラム容量
Q00JCPU	8k ステップ (32k バイト)
Q00CPU	8k ステップ (32k バイト)
Q01CPU	14k ステップ (56k バイト)

(2) 使用デバイスの用途と点数の設定

プログラムで使用するデバイスの用途と点数を設定します。（☞ 第 9 章）

(3) デバイス初期値の設定

ベーシックモデル QCPU のデバイスメモリおよびインテリジェント機能ユニットのバッファメモリに初期値として必要なデータを設定します。（☞ 6.26 節）

(4) ブート運転の検討

プログラムを標準 ROM に格納した場合は、ブート運転してプログラムの実行を行います。

ブート運転を行う場合は、PC パラメータのブートファイル設定を設定してください。

（☞ 5.1.5 項, 11.1.4 項）

11.1.2 ハードウェアチェック

作成したプログラムを書き込む前に、ハードウェアのチェックを行います。
下記手順で、☐ は CPU ユニット側での操作項目を示しています。

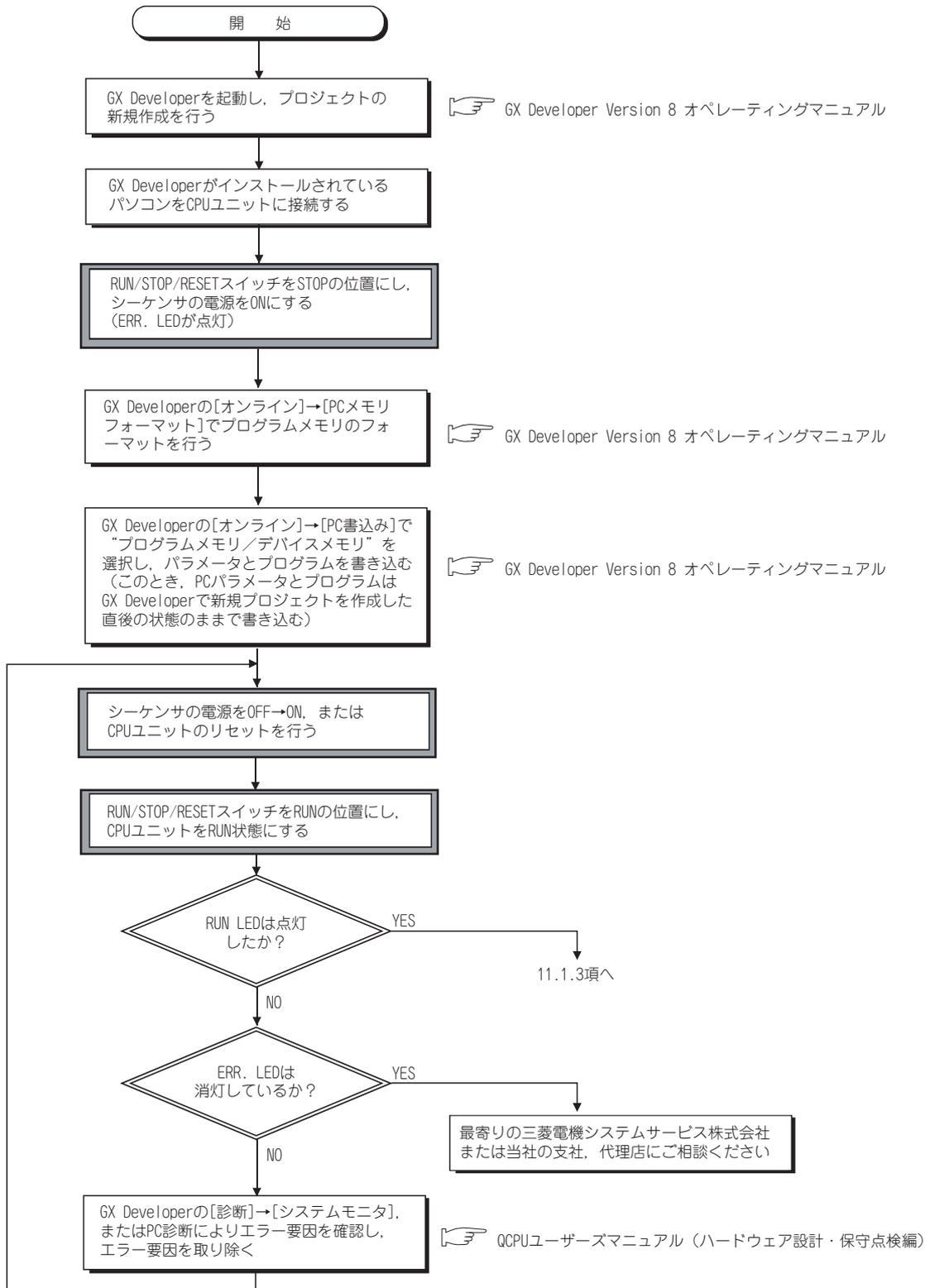


図 11.1 ハードウェアチェックのフロー

11.1 ベーシックモジュール QCPU
11.1.2 ハードウェアチェック

備考

CPU ユニットの設置、装着手順については、下記マニュアルを参照してください。

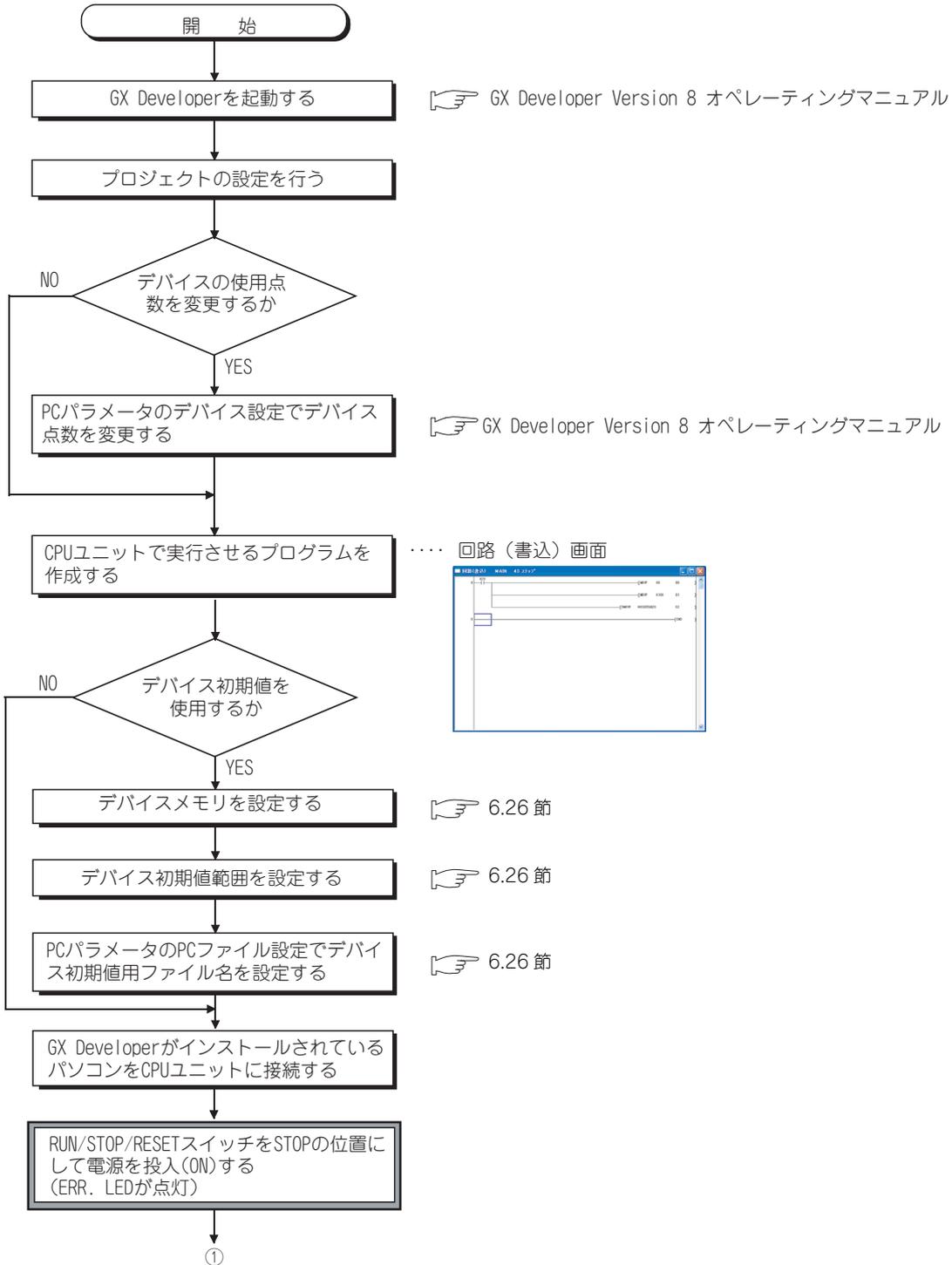
☞ QCPU ユーザーズマニュアル（ハードウェア設計・保守点検編）

11.1.3 プログラムを書き込むまでの手順

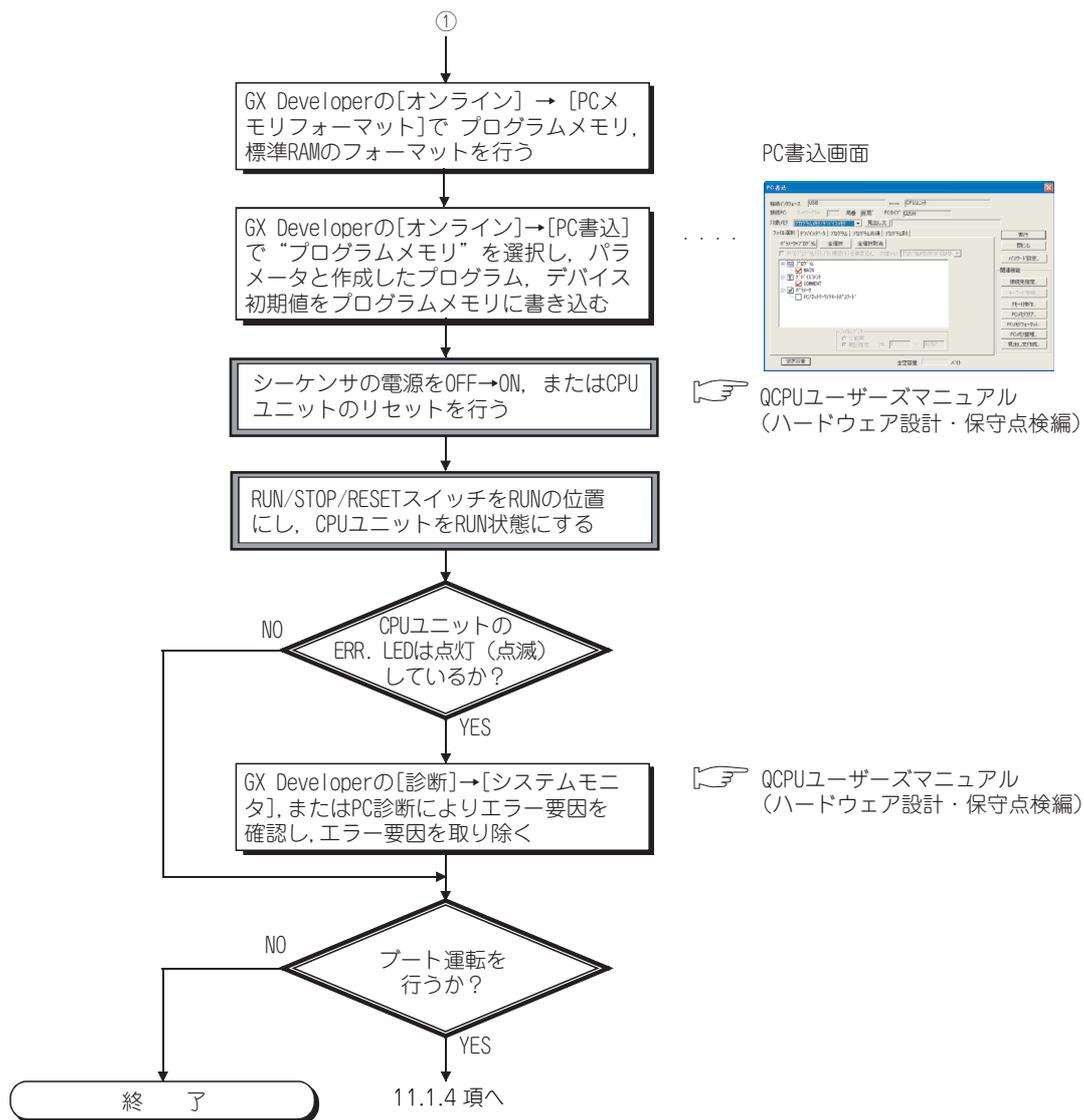
プログラムメモリ (☞ 5.1.2 項) にプログラムを書き込む手順を下記に示します。

標準 ROM にプログラムを格納し、ブート運転を行う場合は、本項の手順を実施した後に 11.1.4 項に示す手順を実施してください。

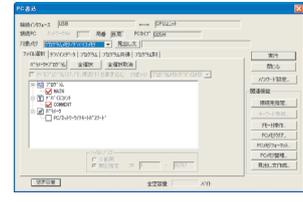
下記手順で、☐ は CPU ユニット側での操作項目を示しています。



11.1 ベーシックモデル QCPU
11.1.3 プログラムを書き込むまでの手順



PC書込画面



☞ QCPUユーザーズマニュアル (ハードウェア設計・保守点検編)

☞ QCPUユーザーズマニュアル (ハードウェア設計・保守点検編)

図 11.2 1本のプログラムを書き込むまでのフロー

11.1.4 ブート運転の手順

ブート運転の手順を下記に示します。

下記手順で、□ は CPU ユニット側での操作項目を示しています。

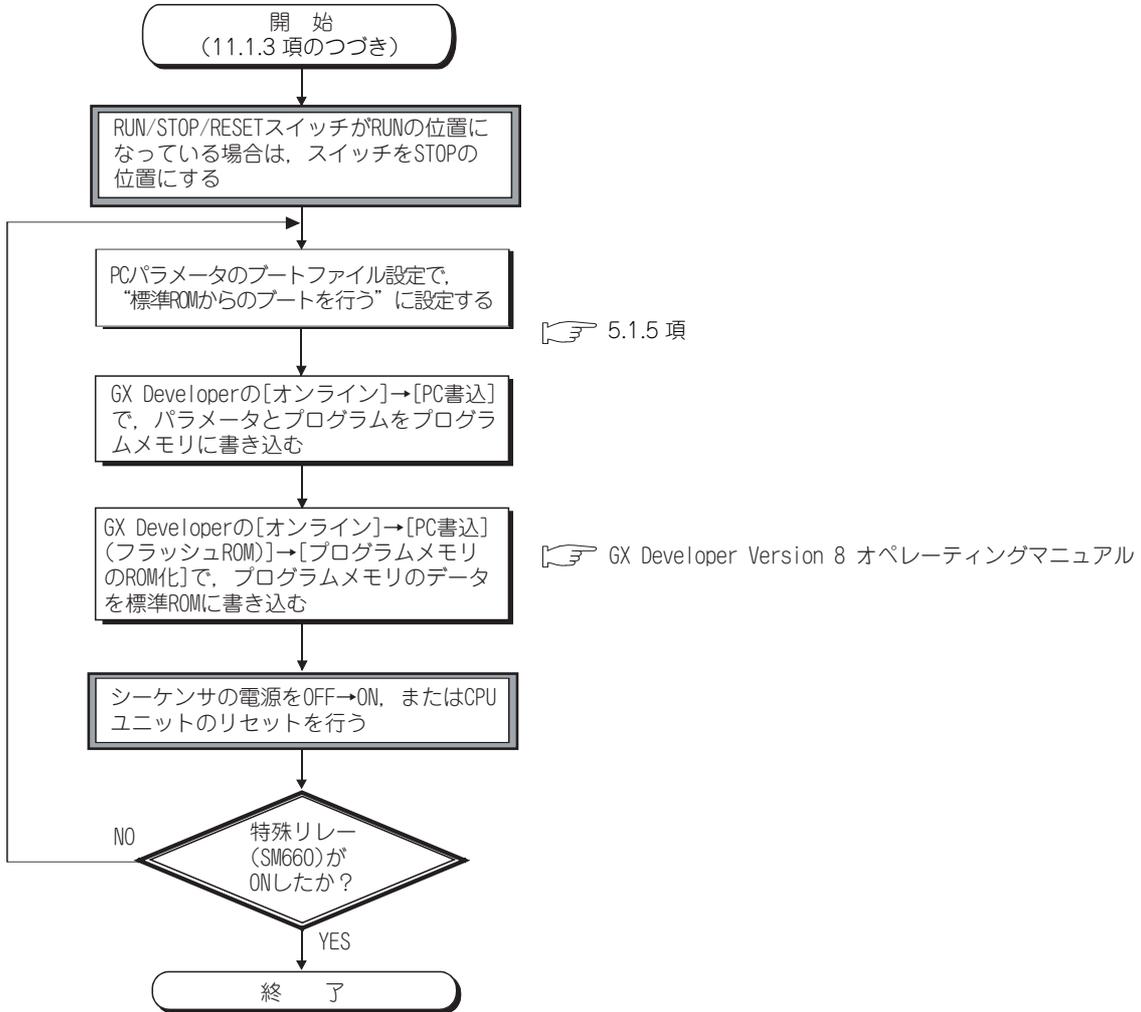


図 11.3 ブート運転のフロー

11.2 ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU

11.2.1 プログラムを作成する場合の検討事項

プログラムを作成する場合、各々のプログラムで使用するプログラム容量、使用デバイス点数、ファイル名などをあらかじめ決めておく必要があります。

(1) プログラム容量の検討

プログラムとパラメータの合計容量が、使用する CPU ユニットで実行できるプログラム容量を越えていないか検討します。(☞ 5.4.3 項)

各 CPU ユニットで実行できるプログラム容量を表 11.2 に示します。

表 11.2 プログラム容量

CPU ユニット	プログラム容量
Q02CPU, Q02HCPU, Q02PHCPU	28k ステップ (112k バイト)
Q06HCPU, Q06PHCPU	60k ステップ (240k バイト)
Q12HCPU, Q12PHCPU, Q12PRHCPU	124k ステップ (496k バイト)
Q25HCPU, Q25PHCPU, Q25PRHCPU	252k ステップ (1008k バイト)

パラメータの格納場所は、プログラムメモリ／標準 ROM／メモリカードから選択できます。

プログラムのみで上記プログラム容量が必要な場合は、パラメータを標準 ROM／メモリカードに格納してください。

(2) プログラムを構造化する単位を決定

プログラムを複数作成する場合は、プログラムを構造化する単位（工程別、機能別）を決定します。

(3) 作成するプログラムの実行条件の設定

複数のプログラムを実行する場合は、プログラムごとに実行条件を設定します。(☞ 2.3 節)

設定されていない場合は、プログラムを実行できません。

(4) 使用デバイスの用途と点数の設定

プログラムで使用するデバイスの用途と点数を設定します。(☞ 第 9 章)

(5) デバイス初期値の設定

デバイスメモリおよびインテリジェント機能ユニットのバッファメモリに初期値として必要なデータを設定します。(☞ 6.26 節)

(6) ブート運転の設定

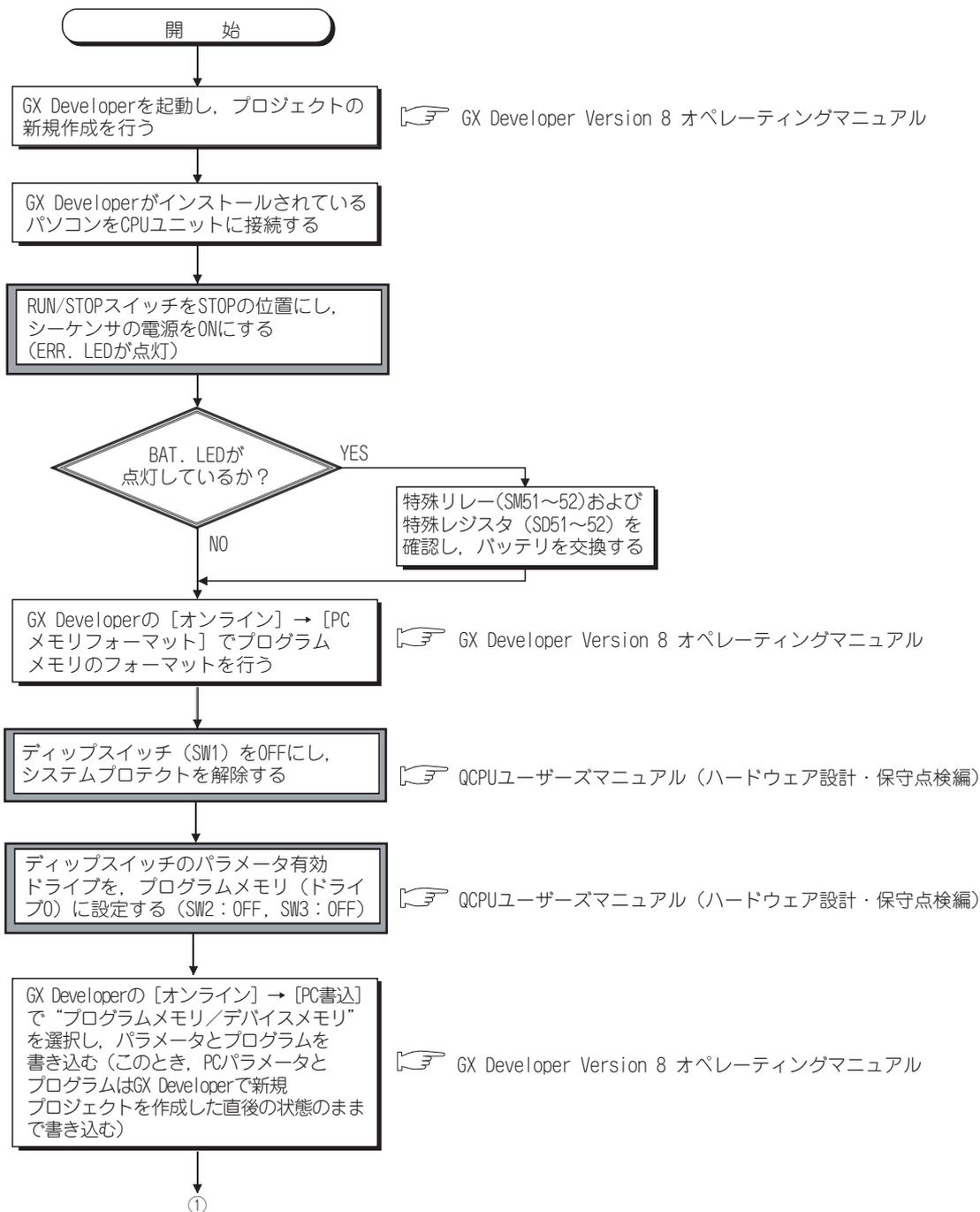
プログラムを標準 ROM／メモリカードに格納した場合は、ブート運転してプログラムの実行を行います。

ブート運転を行う場合は、PC パラメータのブートファイル設定を設定してください。

(☞ 5.2.8 項, 11.2.5 項)

11.2.2 ハードウェアチェック

作成したプログラムを書き込む前に、ハードウェアのチェックを行います。
下記手順で、☐ は CPU ユニット側での操作項目を示しています。



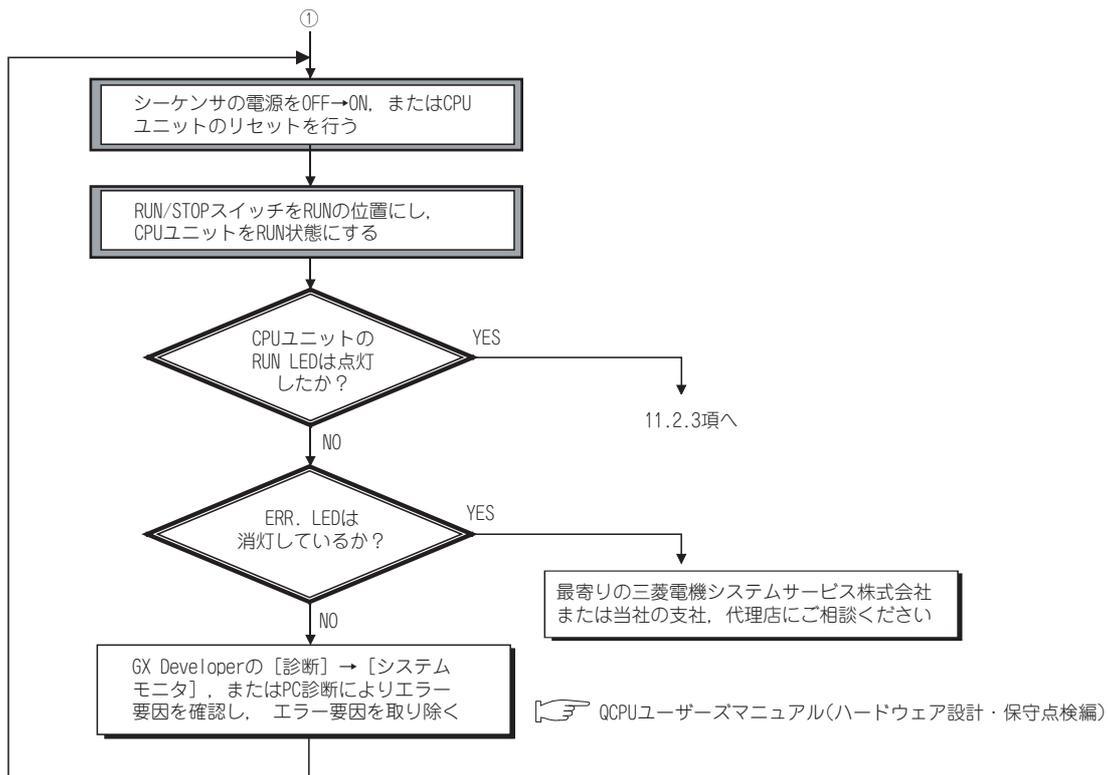


図 11.4 ハードウェアチェックのフロー

備考

CPU ユニットの設置, 装着手順については, 下記マニュアルを参照してください。

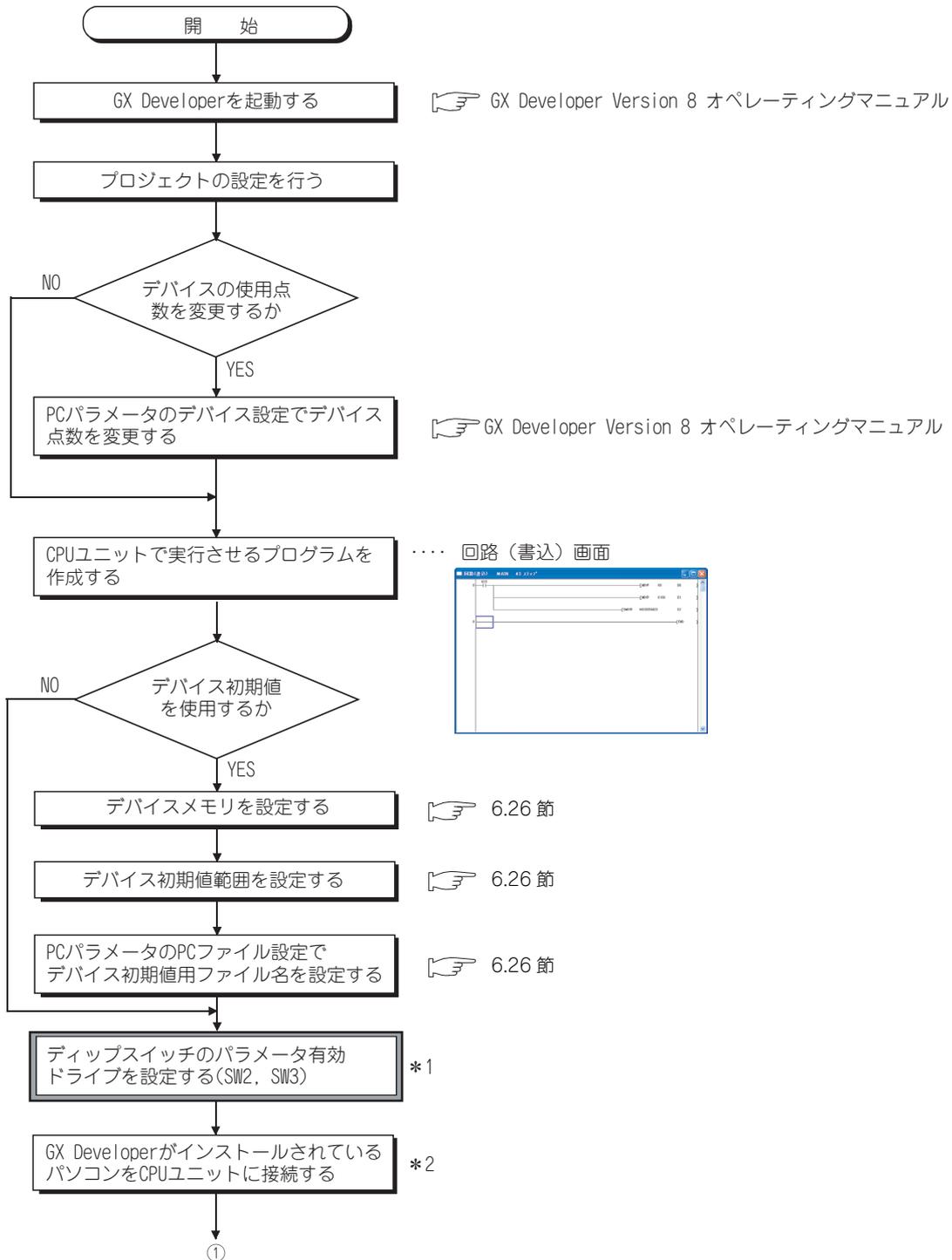
☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編)

11.2.3 1 本のプログラムを書き込むまでの手順

プログラムメモリ (☞ 5.2.2 項) にプログラムを書き込む手順を下記に示します。

標準 ROM / メモリカードにプログラムを格納し、ブート運転を行う場合は、本項の手順を実施した後に 11.2.5 項に示す手順を実施してください。

下記手順で、☐ は CPU ユニット側での操作項目を示しています。



11
11.2 ハイパフォーマンスモデル QCPU、プロセス CPU、二重化 CPU
11.2.3 1 本のプログラムを書き込むまでの手順

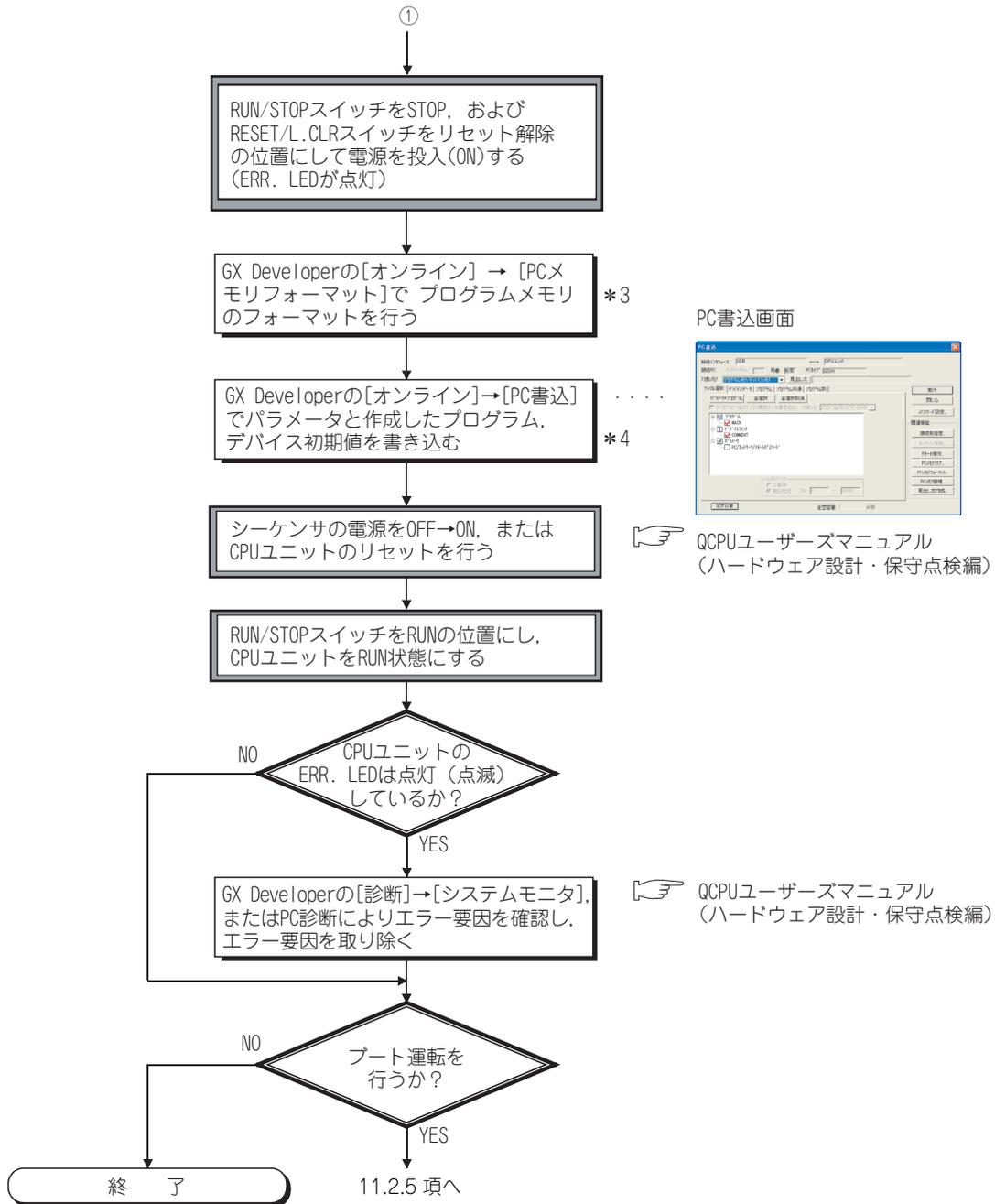


図 11.5 1本のプログラムを書き込むまでのフロー

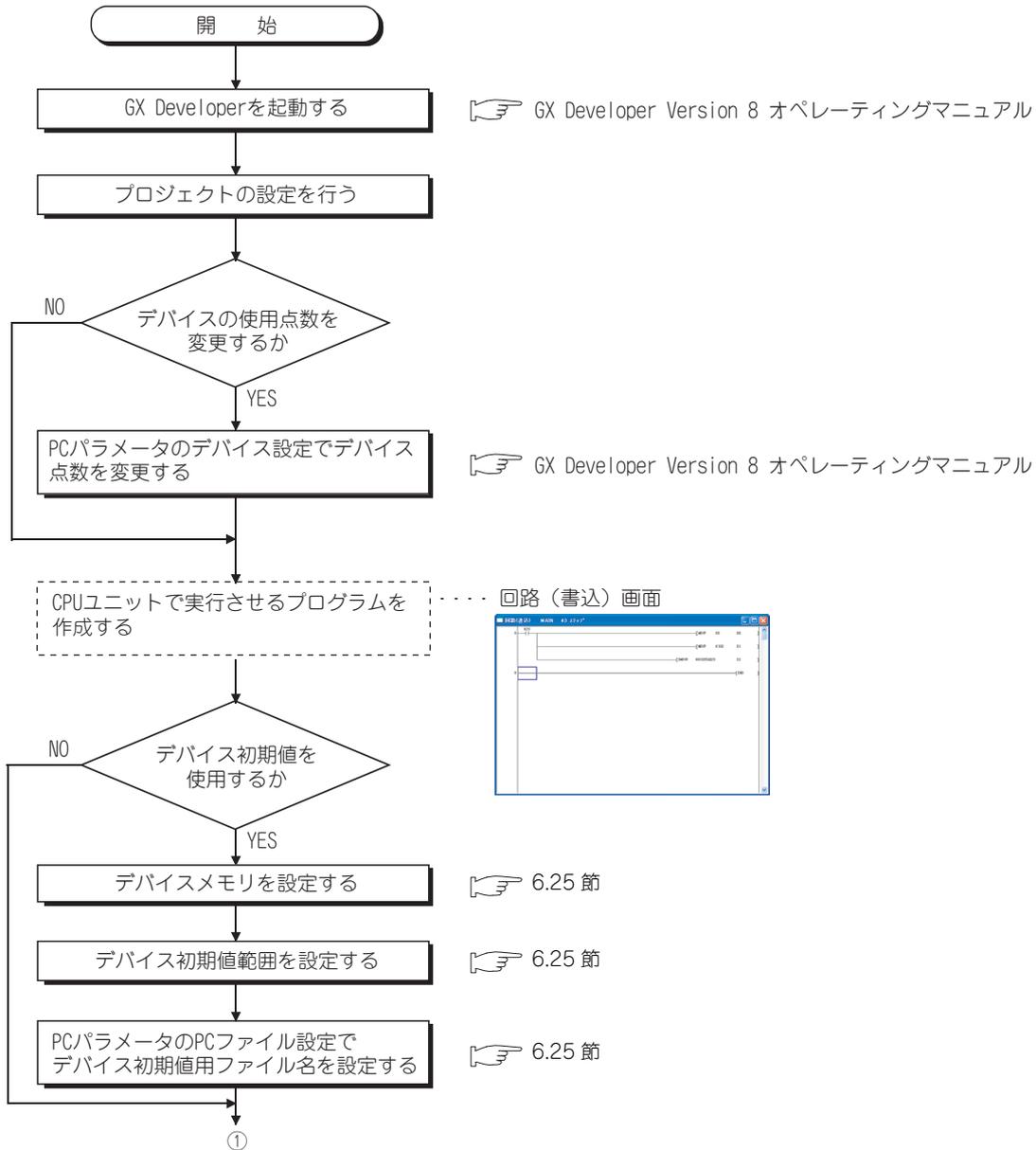
- * 1: ディップスイッチのパラメータ有効ドライブは、デフォルトではプログラムメモリ (ドライブ 0) になっています。(SW2: OFF, SW3: ON) パラメータを標準 ROM やメモリカードに格納する場合は、ディップスイッチのパラメータ有効ドライブを変更してください。
- * 2: 二重化 CPU 使用時の手順については、下記マニュアルを参照してください。
☞ QnPRHCPU ユーザーズマニュアル (二重化システム編)
- * 3: 標準 RAM またはメモリカード (Flash カードは除く) にファイルレジスタやデバイス初期値を格納する場合は、標準 RAM とメモリカードのフォーマットも行ってください。
- * 4: 各データは、下記に示すメモリに書き込んでください。
 - ・プログラム・・・プログラムメモリ
 - ・パラメータ・・・ディップスイッチのパラメータ有効ドライブに設定したメモリ
 - ・デバイス初期値・・・PC パラメータの PC ファイル設定で指定したメモリ

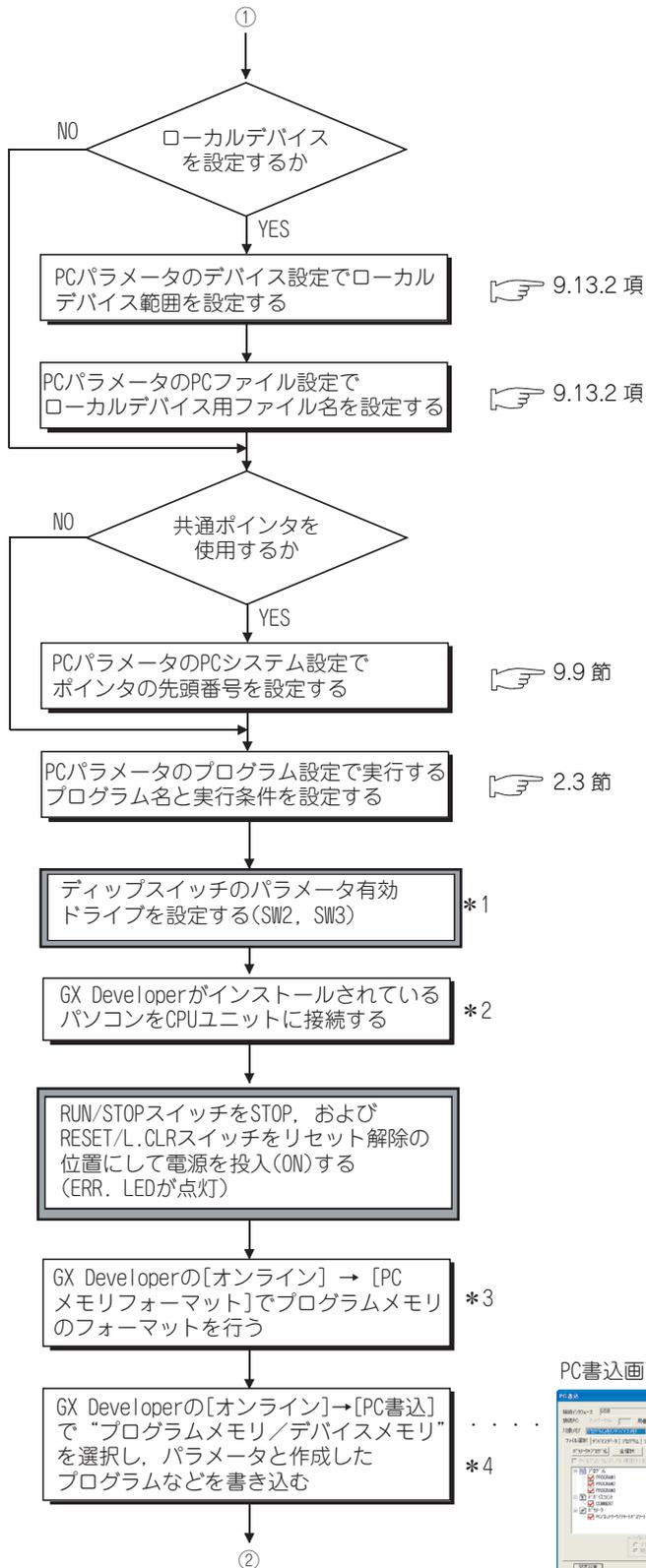
11.2.4 複数のプログラムを書き込むまでの手順

プログラムメモリ (☞ 5.2.2 項) にプログラムを書き込む手順を下記に示します。

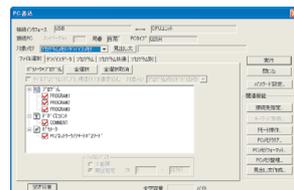
標準 ROM / メモリカードにプログラムを格納し、ブート運転を行う場合は、本項の手順を実施した後に 11.2.5 項に示す手順を実施してください。

下記手順で、☐ は CPU ユニット側での操作項目を示しています。





PC書込画面



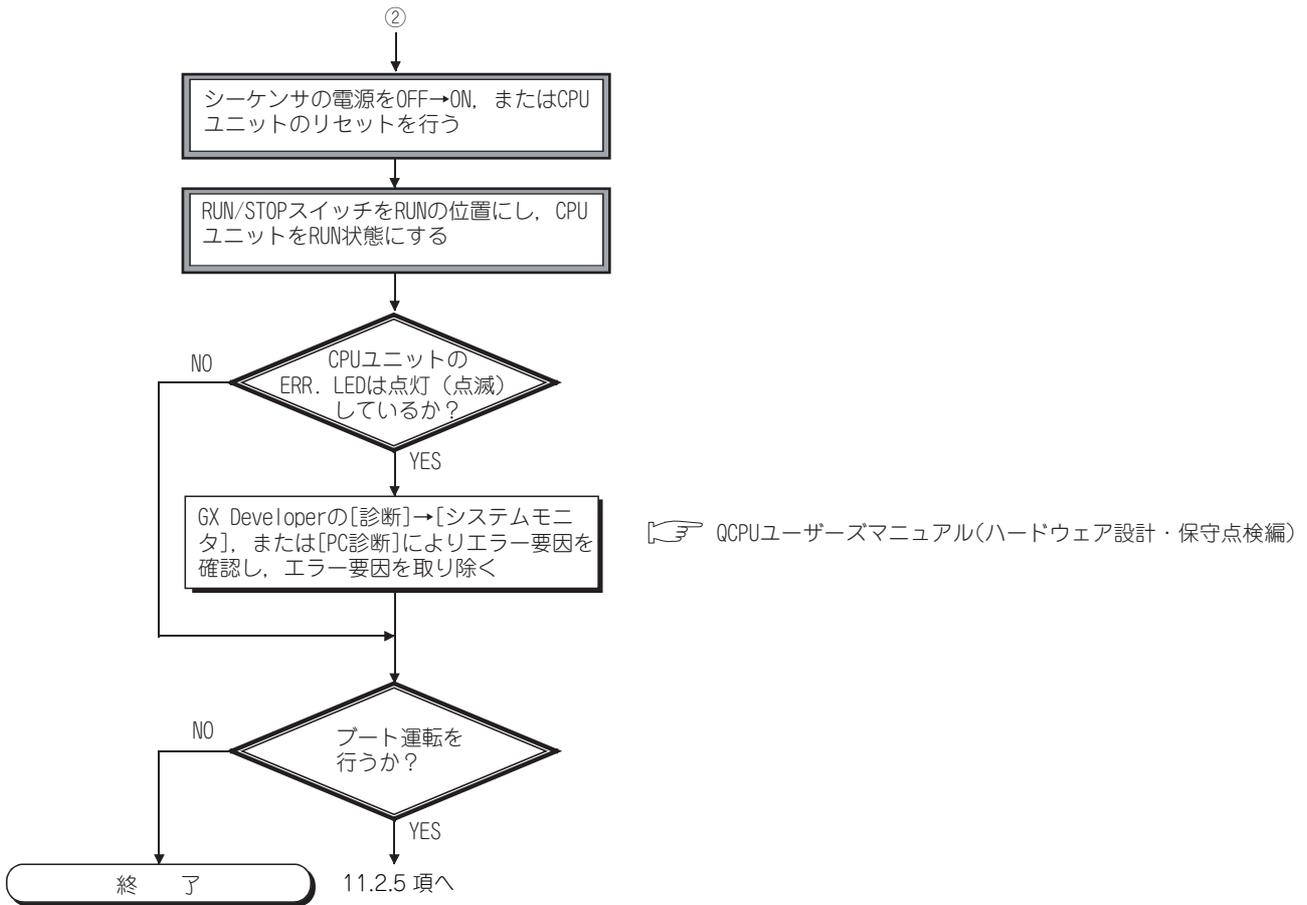


図 11.6 複数のプログラムを書き込むまでのフロー

- * 1: ディップスイッチのパラメータ有効ドライブは、デフォルトではプログラムメモリ (ドライブ 0) になっています。(SW2: OFF, SW3: ON) パラメータを標準 ROM やメモリカードに格納する場合は、ディップスイッチのパラメータ有効ドライブを変更してください。
- * 2: 二重化 CPU 使用時の手順については、下記マニュアルを参照してください。
☞ QnPRH CPU ユーザーズマニュアル (二重化システム編)
- * 3: 標準 RAM またはメモリカード (Flash カードは除く) にファイルレジスタやデバイス初期値を格納する場合は、標準 RAM とメモリカードのフォーマットも行ってください。
- * 4: 各データは、下記に示すメモリに書き込んでください。
 - ・プログラム・・・プログラムメモリ
 - ・パラメータ・・・ディップスイッチのパラメータ有効ドライブに設定したメモリ
 - ・デバイス初期値・・・PC パラメータの PC ファイル設定で指定したメモリ

11.2 ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU
11.2.4 複数のプログラムを書き込むまでの手順

11.2.5 ブート運転の手順

ブート運転の手順を下記に示します。

下記手順で、 は CPU ユニット側での操作項目を示しています。

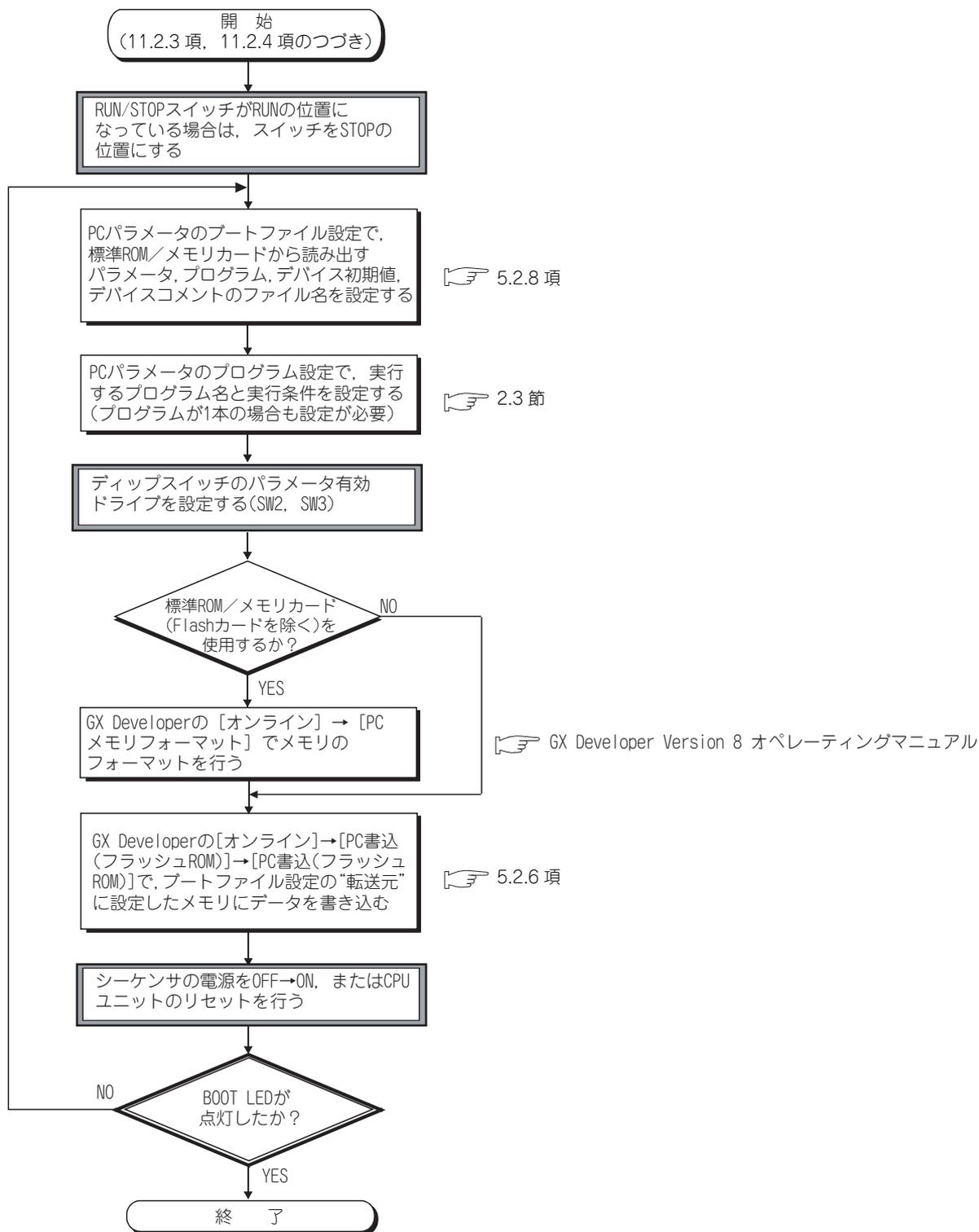


図 11.7 ブート運転のフロー

付 録

付 1 パラメータ No. 一覧

パラメータ No. は、パラメータ設定でエラーが発生したときに特殊レジスタ (SD16 ~ SD26) に格納されます。
パラメータ項目とパラメータ No. の対応一覧を表付 .1 に示します。

表中のパラメータ No. 欄に示す mn, **, M, N の説明については、8.3 節を参照してください。

表付 .1 パラメータ No. 一覧

項 目	パラメータ No.	参 照	
ラベル	0000 _H	8.1 節	
コメント	0001 _H		
I/O 割付	種別	0400 _H	4.2.2 項, 8.1 節
	形名		
	点数		
	先頭 XY(先頭入出力番号)		
基本設定	ベース形名	0401 _H	4.2.1 項, 8.1 節
	電源ユニット形名		
	増設ケーブル形名		
	スロット数		
詳細設定	エラー時出力モード	0403 _H	6.8 節, 8.1 節
	I/O 応答時間	0405 _H	6.7 節, 8.1 節
	管理 CPU	0406 _H	8.1 節, QCPU ユーザーズマニュアル (マルチ CPU システム編)
スイッチ設定	0407 _H	6.10 節, 8.1 節	
グループ No.	05mn _H	8.3 節	
	0Amn _H	8.3 節	
二重化パラメータ	0D00 _H	8.2 節, QCPU ユーザーズマニュアル (二重化システム編)	
CPU 台数	0E00 _H	8.1 節, QCPU ユーザーズマニュアル (マルチ CPU システム編)	
動作モード	0E01 _H		
グループ外の入出力設定	グループ外の入力状態を取り込む		0E04 _H
	グループ外出力状態を取り込む		
タイマ時限設定	低 速	1000 _H	8.1 節
	高 速		
RUN-PAUSE 接点	RUN	1001 _H	6.6.1 項, 8.1 節
	PAUSE		6.6.2 項, 8.1 節
リモートリセット	1002 _H	6.6.3 項, 8.1 節	
STOP → RUN 時の出力モード	1003 _H	6.4 節, 8.1 節	
浮動小数点演算処理	1004 _H	2.4.4 項, 8.1.2 項	
共通ポインタ No	1005 _H	9.9.2 項, 8.1 節	
空きスロット点数	1007 _H	4.1.1 項, 8.1 節	
割込プログラム/定周期プログラム設定	1008 _H	2.2.3 項, 2.3.5 項, 8.1 節	
システム割り込み設定		割り込みカウンタ先頭 No.	8.1 節
	In 定周期間隔 (n : 28 ~ 31)		
インテリジェント機能ユニット設定 (割込みポインタ設定)	100A _H	9.11 節, 8.1 節	
ユニット同期設定	100C _H	8.1 節	
A シリーズ CPU 互換設定	100D _H	8.1 節, 10.1.4 項	

(次ページへつづく)

表付.1 パラメータ No. 一覧 (つづき)

項 目		パラメータ No.	参 照
シリアルコミュニケーション機能を使用する		100E _H	6.24 節, 8.1 節
伝送速度			
サムチェック			
伝送ウェイト時間			
RUN 中書込設定			
システム割り込み設定 (高速割り込み設定)	X 入力	100F _H	6.22 節, 8.1 節
	Y 出力	1010 _H	
	バッファ読出	1011 _H	
	バッファ書込	1012 _H	
ファイルレジスタ		1100 _H	9.7 節, 8.1 節
命令で使用するコメントファイル		1101 _H	8.1 節
デバイス初期値		1102 _H	6.26 節, 8.1 節
ローカルデバイス用ファイル		1103 _H	9.13.2 項, 8.1 節
デバイス点数		2000 _H	9.1 節, 8.1 節
ラッチ (1) 先頭/最終		2001 _H	3.7 節, 6.3 節, 8.1 節
ラッチ (2) 先頭/最終		2002 _H	
ローカルデバイス先頭/最終		2003 _H	9.13.2 項, 8.1 節
WDT (ウォッチドッグ タイマ) 設定	WDT 設定	3000 _H	6.16 節, 8.1 節
	初期実行監視時間		2.3.1 項, 8.1 節
エラーチェック	バッテリーチェックを行う	3001 _H	6.17 節, 8.1 節
	ヒューズ断チェックを行う		
	ユニット照合を行う		
エラー時の運転モード	演算エラー	3002 _H	6.17 節, 8.1 節
	拡張命令エラー		
	ヒューズ断		
	ユニット照合エラー		
	インテリユニットプログラム実行エラー		
	ファイルアクセスエラー		
	メモリカード操作エラー		
外部電源供給 OFF			
コンスタントスキャン		3003 _H	6.2 節, 8.1 節
エラー履歴		3005 _H	6.18 節, 8.1 節
低速プログラム実行時間		3006 _H	2.3.3 項, 8.1 節
メモリチェック	プログラムメモリチェックを行う	3008 _H	6.27 節, 8.1 節
詳細設定	H/W エラー時 CPU 動作モード	4004 _H	6.9 節, 8.1 節
MELSECNET/H 枚数設定		5000 _H	8.3 節
他局アクセス時の有効ユニット		5001 _H	
リンク間転送		5002 _H	
ルーチングパラメータ		5003 _H	
先頭 I/O No.		5NM0 _H	
ネットワーク No.			
総 (子) 局数			
モード		5NM0 _H	

(次ページへつづく)

表 付 .1 パラメータ No. 一覧 (つづき)

項 目		パラメータ No.	参 照
リフレッシュパラメータ		5NM1 _H	8.3 節
共通パラメータ		5NM2 _H	
局固有パラメータ		5NM3 _H	
サブマスタ用パラメータ		5NM5 _H	
共通パラメータ 2		5NMA _H	
局固有パラメータ 2		5NMB _H	
割込み設定			
プログラム設定		7000 _H	2.3 節, 8.1 節
ブートオプション	プログラムメモリをクリアする	7000 _H	5.2.8 項, 8.1 節
	メモリカード→標準 ROM 全データ自動書込		
ブートファイル設定			
SFC プログラム起動モード		8002 _H	8.1 節
起動条件		8003 _H	
ブロック停止時の出力モード		8006 _H	
Ethernet 枚数設定		9000 _H	8.3 節
先頭 I/O No.		9N00 _H	
ネットワーク No.			
グループ No.			
局番			
動作設定			
イニシャル設定		9N01 _H	
オープン設定		9N02 _H	
ルータ中継パラメータ		9N03 _H	
局番 <-> IP 関連情報		9N05 _H	
FTP パラメータ		9N06 _H	
電子メール設定		9N07 _H	
	報知設定	9N08 _H	
割込み設定		9N09 _H	
ルーチングパラメータ		9N04 _H	
CC-Link IE コントローラネットワーク枚数設定		A000 _H	8.3 節
リンク間転送		A002 _H	
先頭 I/O No.		ANM0 _H	
ネットワーク No.			
総 (子) 局数			
局番			
モード		ANM0 _H	
リフレッシュパラメータ		ANM1 _H	
共通パラメータ		ANM2 _H	
局固有パラメータ		ANM3 _H	

(次ページへつづく)

表付 .1 パラメータ No. 一覧 (つづき)

項 目	パラメータ No.	参 照
ユニット枚数	C000 _H	
リモート入力 (RX)	CNM1 _H	8.3 節
リモート出力 (RY)		
リモートレジスタ (RWr)		
リモートレジスタ (RWw)		
Ver.2 リモート入力 (RX)		
Ver.2 リモート出力 (RY)		
Ver.2 リモートレジスタ (RWr)		
Ver.2 リモートレジスタ (RWw)		
特殊リレー (SB)		
特殊レジスタ (SW)		
先頭 I/ONo.	CNM2 _H	
動作設定		
総接続台数		
リトライ回数	CNM2 _H	8.3 節
自動復列台数		
待機マスタ局番号		
CPU ダウン指定		
スキャンモード指定		
ディレイ時間設定		
局情報設定		
リモートデバイス局イニシャル設定		
イニシャル設定		
割込み設定		
電源 ON 時の設定	D001 _H	
待機系監視設定		
デバッグモード設定		
バックアップモード設定	D002 _H	
トラッキング転送モード設定		
トラッキングデバイス設定	D003 _H	8.2 節, QnPRHCPU ユーザーズ マニュアル (二重化システム編)
立上り/立下り実行命令履歴 (シグナルフロー)		
デバイス詳細設定		
トラッキングブロック No.		
トラッキングブロック No.1 を自動転送する (SM1520 を自動 ON)		
デバイス範囲設定		
ファイルレジスタファイル設定		
グループ設定	D004 _H	8.3 節
二重化設定	D5** _H	
	D9** _H	
	DA** _H	
通信エリア設定 (リフレッシュ設定)	E002 _H E003 _H	8.1 節, QCPU ユーザーズマニ ュアル (マルチ CPU システム編)
オンラインユニット交換設定	E006 _H	

付 2 バージョンアップによる機能の追加・変更

CPU ユニットは、バージョンアップにより機能の追加、仕様変更を行っています。
使用できる機能、仕様は、機能バージョン/シリアル No. により、異なります。

付 2.1 ベーシックモデル QCPU の機能アップ

(1) 仕様比較

表付 .2 仕様比較

仕 様		CPU ユニットのシリアル No. の上 5 桁	
		機能バージョン A	機能バージョン B
		“04121” 以前	“04122” 以降
標準 RAM 容量	Q00JCPU	×	
	Q00CPU	64k バイト	128k バイト
	Q01CPU	64k バイト	128k バイト
CPU 共有メモリ	Q00JCPU	×	
	Q00CPU	×	○
	Q01CPU	×	○

○：使用可能/対応可， ×：使用不可/未対応

付

付 2 バージョンアップによる機能の追加・変更
付 2.1 ベーシックモデル QCPU の機能アップ

(2) CPU ユニットおよび GX Developer のバージョンによる追加機能の使用可否

表付 .3 CPU ユニットおよび GX Developer のバージョンによる追加機能の使用可否

追加機能	対応機能バージョン	対応シリアル No. の上 5 桁	対応 GX Developer
ファンクションブロック (☞ GX Developer Version 8 オペレーティングマニュアル (ファンクションブロック編))	A	"04121" 以前	Version 8.00A 以降
ストラクチャードテキスト (ST) 言語 (☞ MELSEC-Q/L プログラミングマニュアル (ストラクチャードテキスト編))			
MELSAP3 (☞ MELSEC-Q/L/QnA プログラミングマニュアル (SFC 編))	B	"04122" 以降	
PID 演算機能* 1 (☞ MELSEC-Q/L/QnA プログラミングマニュアル (PID 制御命令編))			
実数演算機能* 1 (☞ 9.12.3 項)			
インテリジェント機能ユニット割込み (☞ 6.23 節)			
デバイス初期値自動設定機能 (☞ 6.26 節)			
リモートパスワード設定機能 (☞ 6.19.2 項)			
電子メールパラメータ (☞ 電子メール機能に対応したユニットのマニュアル)			
ポインタを使用した RUN 中書込 (☞ 6.15.2 項)			
ファイルレジスタ R の大容量化* 2 (☞ 9.7 節)			
マルチ CPU システム対応 (☞ QCPU ユーザーズマニュアル (マルチ CPU システム編))* 2			
複数ブロック RUN 中書込 (☞ 6.15.2 項)	"06112" 以降	Version 8.03D 以降	
CC-Link リモートネット追加モード対応 (☞ MELSEC-Q CC-Link システムマスタ・ローカルユニットユーザーズマニュアル)			
CC-Link IE コントローラネットワーク対応 (☞ MELSEC-Q CC-Link IE コントローラネットワークリファレンスマニュアル)	—	—	Version 8.68W 以降

— : 機能バージョン, シリアル No., GX Developer に関係しない機能

* 1 : Version 8 の GX Developer で実装された CPU ユニットの命令は, Version 7 以前の GX Developer で読み出すと「命令コード異常」として GX Developer で処理されます。

* 2 : Q00JCPU では未対応です。

(3) ベーシックモデル QCPU での相違点

表付 .4 ベーシックモデル QCPU での相違点

項 目		Q00JCPU	Q00CPU	Q01CPU
CPU ユニット		CPU ユニット, 電源ユニット, 基本ベースユニット (5 スロット) 一体型	CPU ユニット単体	
基本ベースユニット/スリムタイプ基本 ベースユニット		不要	要	
増設ベースユニット		接続可 ただし、スリムタイプ基本ベースユニット使用時は、接続不可		
増設段数		最大 2 段	最大 4 段 (スリムタイプ基本ベースユニット使用時は、 接続不可)	
装着可能ユニット数		16 枚	24 枚	
電源ユニット				
基本ベースユニット		不要	要	
スリムタイプ基本ベースユニット		不要	要	
増設ベース ユニット	Q52B, Q55B	不要		
	Q63B, Q65B, Q68B, Q68RB, Q612B	要		
増設ケーブル		QC05B, QC06B, QC12B, QC30B, QC50B, QC100B		
メモリカードインタフェース		無		
外部インタフェース	RS-232	有 (伝送速度 : 9.6kbps, 19.2kbps, 38.4kbps, 57.6kbps, 115.2kbps)		
	USB	無		
処理速度 (シーケンス命令)	LD X0	0.20 μ s	0.16 μ s	0.10 μ s
	MOV D0 D1	0.70 μ s	0.56 μ s	0.35 μ s
プログラム容量* 1		8k ステップ (32k バイト)	8k ステップ (32k バイト)	14k ステップ (56k バイト)
メモ リ 容 量	プログラムメモリ	58k バイト	94k バイト	
	標準 RAM	—	128k バイト* 2	
	標準 ROM	58k バイト	94k バイト	
	CPU 共有メモリ* 3	無	1k バイト (ユーザ自由エリア 320 ワード)	
標準 ROM の書き込み回数		最大 10 万回		
デバイスメモリ容量		16.4k ワードの範囲内でデバイス点数の変更が可能		
入出力デバイス点数 (リモート入出力を含む)		2048 点		
入出力点数		256 点	1024 点	
ファイルレジスタ		無	有	
シリアルコミュニケーション機能		無	有 (CPU ユニットの RS-232 インタフェースを使用)	

○ : 使用可能/対応可, × : 使用不可, — : GX Developer に関係しない機能

* 1 : プログラム容量の 1 ステップは、4 バイトです。

* 2 : 機能バージョン A では、64k バイトです。

* 3 : 機能バージョン B で追加されたメモリです。

CPU 共有メモリはラッチされません。

CPU ユニットの電源 OFF → ON またはリセット操作を行うと、CPU 共有メモリがクリアされます。

付 2.2 ハイパフォーマンスモデル QCPU の機能アップ

(1) 仕様比較

表付 .5 仕様比較

仕 様		CPU ユニットのシリアル No. の上 5 桁					
		機能バージョン A		機能バージョン B			
		“02091” 以前	“02092” 以降	“02112” 以降	“03051” 以降	“04012” 以降	“16021” 以降
標準 RAM 容量	Q02CPU	64k バイト					
	Q02HCPU	64k バイト				128k バイト	
	Q06HCPU	64k バイト				128k バイト	
	Q12HCPU	64k バイト	256k バイト				
	Q25HCPU	64k バイト	256k バイト				
CPU 共有メモリ		×	×	○	○	○	○
SRAM カードのバッテリーの長寿命化		×	×	×	×	○	○
2M バイトの SRAM カード対応		×	×	×	×	○	○
4M バイトの SRAM カード対応		×	×	×	×	×	○

○：使用可能／対応可， ×：使用不可／未対応

(2) CPU ユニットおよび GX Developer のバージョンによる追加機能の使用可否

表付 .6 CPU ユニットおよび GX Developer のバージョンによる追加機能の使用可否

追加機能	対応 機能バージョン	対応シリアル No. の上 5 桁	対応 GX Developer		
標準 ROM への自動書き込み機能 (☞ 5.2.7 項)	A	“02092” 以降	Version 6 以降		
外部入出力の強制 ON/OFF 機能 (☞ 6.11.3 項)					
リモートパスワード設定機能 (☞ 6.19.2 項)					
MELSECNET/H リモート I/O ネットワーク対応 (☞ 4.2.1 項)					
割込みユニット対応 (☞ 9.10 節)	B	“02122” 以降	Version 7 以降		
プログラミングユニット対応 (☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編))					
マルチ CPU システム対応 (☞ QCPU ユーザーズマニュアル (マルチ CPU システム編))				“03051” 以降	Version 7.10L 以降
マルチ CPU システムのパソコン CPU ユニット対応 (☞ QCPU ユーザーズマニュアル (マルチ CPU システム編))				“04012” 以降	Version 8 以降
高速割込み機能 (☞ 6.22 節)					—
専用命令のユニット指定のインデックス修飾対応 (☞ 専用命令を使用できるインテリジェント機能ユニットのマニュアル)				—	
COM 命令のリフレッシュ項目の選択 (☞ MELSEC-Q/L プログラミングマニュアル (共通命令編))				“04122” 以降	Version 8 以降
SFC プログラムのファイルの RUN 中書き込み (☞ 6.12.2 項)					
ファイルのサイズ単位 (☞ 5.4.4 項)					

(次ページにつづく)

表 付 .6 CPU ユニットおよび GX Developer のバージョンによる追加機能の使用可否 (つづき)

追加機能	対応機能バージョン	対応シリアル No. の上 5 桁	対応 GX Developer
CC-Link リモートネット追加モード対応 (☞ MELSEC-Q CC-Link システムマスタ・ローカルユニットユーザズマニュアル)	B	"05032" 以降	Version 8.03D 以降
不完全微分 PID 演算機能 (☞ MELSEC-Q/L/QnA プログラミングマニュアル (PID 制御命令編))			—
浮動小数点比較命令の高速化			—
SFC 活性ステップコメントの読出し対応 (☞ MELSEC-Q/L/QnA プログラミングマニュアル (SFC 編))		"07012" 以降	—
電源二重化システムのエラー検出 (☞ 6.20 節)		"07032" 以降	Version 8.23Z 以降
1/1000 秒単位の時計データ対応 (☞ 6.5 節)			—
サンプリングトレースファイルの標準 RAM 格納 (☞ 6.14 節)			Version 8.23Z 以降
マルチ CPU システム構成時のリフレッシュデバイスの個別設定 (☞ QCPU ユーザズマニュアル (マルチ CPU システム編))			Version 8.27D 以降
RUN 中書き込み時の立下がり命令の実行/非実行選択 (☞ 6.12.3 項)		"07092" 以降	Version 8.27D 以降
CC-Link サイクリックデータ局単位ブロック保証機能対応 (☞ MELSEC-Q CC-Link システムマスタ・ローカルユニットユーザズマニュアル)		"08032" 以降	Version 8.32J 以降
CC-Link パラメータ 8 枚設定対応 (☞ MELSEC-Q CC-Link システムマスタ・ローカルユニットユーザズマニュアル)		"09012" 以降	Version 8.45X 以降
CC-Link IE コントローラネットワーク対応 (☞ MELSEC-Q CC-Link IE コントローラネットワークリファレンスマニュアル)			
ATA カード変更対応 (☞ QCPU ユーザズマニュアル (ハードウェア設計・保守点検編))			
4M バイトの SRAM カード対応 (☞ QCPU ユーザズマニュアル (ハードウェア設計・保守点検編))			
		"16021" 以降	—

— : GX Developer に関係しない機能

付

付 2 バージョンアップによる機能の追加・変更
付 2.2 ハイパフォーマンスモデル QCPU の機能アップ

付 2.3 プロセス CPU の機能アップ

(1) CPU ユニットおよび GX Developer のバージョンによる追加機能の使用可否

表付 .7 CPU ユニットおよび GX Developer のバージョンによる追加機能の使用可否

追加機能	対応機能バージョン	対応シリアル No. の上 5 桁	対応 GX Developer	
専用命令のユニット指定のインデックス修飾対応 (☞ 専用命令を使用できるインテリジェント機能ユニットのマニュアル)	C	“07032” 以降	—	
COM 命令のリフレッシュ項目の選択 (☞ MELSEC-Q/L プログラミングマニュアル (共通命令編))			—	
SFC プログラムのファイルの RUN 中書込み (☞ 6.12.2 項)			Version 8 (Version 8.22Y 以前)	
ファイルのサイズ単位 (☞ 5.4.4 項)			Version 8.23Z 以降	
CC-Link リモートネット追加モード対応 (☞ MELSEC-Q CC-Link システムマスタ・ローカルユニットユーザズマニュアル)			—	
メモリチェック機能 (☞ 6.26 節)			Version 8.23Z 以降	
SFC 活性ステップコメントの読出し対応 (☞ MELSEC-Q/L/QnA プログラミングマニュアル (SFC 編))			—	
電源二重化システムのエラー検出 (☞ 6.20 節)			Version 8.23Z 以降	
1/1000 秒単位の時計データ対応 (☞ 6.5 節)			—	
サンプリングトレースファイルの標準 RAM 格納 (☞ 6.14 節)			Version 8.23Z 以降	
マルチ CPU システム構成時のリフレッシュデバイスの個別設定 (☞ QCPU ユーザズマニュアル (マルチ CPU システム編))			Version 8.23Z 以降	
RUN 中書込み時の立下がり命令の実行/非実行選択 (☞ 6.12.3 項)			“07092” 以降	Version 8.27D 以降
CC-Link サイクリックデータ局単位ブロック保証機能対応 (☞ MELSEC-Q CC-Link システムマスタ・ローカルユニットユーザズマニュアル)			“08032” 以降	Version 8.32J 以降
CC-Link パラメータ 8 枚設定対応 (☞ MELSEC-Q CC-Link システムマスタ・ローカルユニットユーザズマニュアル)			“10042” 以降	Version 8.68W 以降
4M バイトの SRAM カード対応 (☞ QCPU ユーザズマニュアル (ハードウェア設計・保守点検編))			“16021” 以降	—

— : GX Developer に関係しない機能

付 2.4 二重化 CPU の機能アップ

(1) CPU ユニットおよび GX Developer のバージョンによる追加機能の使用可否

表付 .8 CPU ユニットおよび GX Developer のバージョンによる追加機能の使用可否

追加機能	対応機能バージョン	対応シリアル No. の上 5 桁	対応 GX Developer
SFC 活性ステップコメントの読出し対応 (☞ MELSEC-Q/L/QnA プログラミングマニュアル (SFC 編))	D	"07032" 以降	—
1/1000 秒単位の時計データ対応 (☞ 6.5 節)			Version 8.23Z 以降
サンプリングトレースファイルの標準 RAM 格納 (☞ 6.14 節)		"07092" 以降	Version 8.27D 以降
RUN 中書き込み時の立下がり命令の実行/非実行選択 (☞ 6.12.3 項)		"09012" 以降	Version 8.45X 以降
増設ベースユニット対応 (☞ QnPRHCPU ユーザーズマニュアル (二重化システム編))		"09102" 以降	Version 8.58L 以降
CC-Link パラメータ 8 枚設定対応 (☞ MELSEC-Q CC-Link システムマスタ・ローカルユニットユーザーズマニュアル)		"10042" 以降	Version 8.68W 以降
4M バイトの SRAM カード対応 (☞ QCPU ユーザーズマニュアル (ハードウェア設計・保守点検編))		"16021" 以降	—

— : GX Developer に関係しない機能

付

付 2 バージョンアップによる機能の追加・変更
付 2.4 二重化 CPU の機能アップ

付 3 デバイス点数割付けシート

(1) ベーシックモデル QCPU 用

表付 .9 デバイス点数割付けシート

デバイス名	記号	進数	デバイス点数* ²		制限チェック			
			点数	番号	容量 (ワード)* ³	ビット点数* ²		
入力リレー* ¹	X	16	2k(2048) 点	X0000 ~ 07FF	÷16	128 ワード	×1	2048 点
出力リレー* ¹	Y	16	2k(2048) 点	Y0000 ~ 07FF	÷16	128 ワード	×1	2048 点
内部リレー	M	10	k() 点	M0 ~	÷16	ワード	×1	点
ラッチリレー	L	10	k() 点	L0 ~	÷16	ワード	×1	点
リンクリレー	B	16	k() 点	B0000 ~	÷16	ワード	×1	点
アナンシェータ	F	10	k() 点	F0 ~	÷16	ワード	×1	点
リンク特殊リレー* ¹	SB	16	1k(1024) 点	SB0000 ~ 03FF	÷16	64 ワード	×1	1024 点
エッジリレー	V	10	k() 点	V0 ~	÷16	ワード	×1	点
ステップリレー* ¹	S	10	2k(2048) 点	S0 ~ 2047	÷16	128 ワード	×1	2048 点
タイマ	T	10	k() 点	T0 ~	× $\frac{18}{16}$	ワード	×2	点
積算タイマ	ST	10	k() 点	ST0 ~	× $\frac{18}{16}$	ワード	×2	点
カウンタ	C	10	k() 点	C0 ~	× $\frac{18}{16}$	ワード	×2	点
データレジスタ	D	10	k() 点	D0 ~	×1	ワード		—
リンクレジスタ	W	16	k() 点	W0000 ~	×1	ワード		—
リンク特殊レジスタ* ¹	SW	16	1k(1024) 点	SW0000 ~ 03FF	×1	1024 ワード		—
デバイス合計						ワード (16704 ワード以下)		点

* 1 : システムで固定です。(変更不可)

* 2 : 1 デバイスの最大点数は 32k 点です。

* 3 : デバイス点数を、容量 (ワード) 欄に示す数字で乗算 (または除算) した数値を記入します。

(2) ハイパフォーマンスモデル QCPU, プロセス CPU, 二重化 CPU 用

表付 .10 デバイス点数割付けシート

デバイス名	記号	進数	デバイス点数*2		制限チェック			
			点数	番号	容量 (ワード)*3		ビット点数*2	
入力リレー*1	X	16	8k(8192)点	X0000 ~ 1FFF	÷16	512ワード	×1	8192点
出力リレー*1	Y	16	8k(8192)点	Y0000 ~ 1FFF	÷16	512ワード	×1	8192点
内部リレー	M	10	k()点	M0 ~	÷16	ワード	×1	点
ラッチリレー	L	10	k()点	L0 ~	÷16	ワード	×1	点
リンクリレー	B	16	k()点	B0000 ~	÷16	ワード	×1	点
アナンシェータ	F	10	k()点	F0 ~	÷16	ワード	×1	点
リンク特殊リレー*1	SB	16	2k(2048)点	SB0000 ~ 07FF	÷16	128ワード	×1	2048点
エッジリレー	V	10	k()点	V0 ~	÷16	ワード	×1	点
ステップリレー*1	S	10	8k(8192)点	S0 ~ 8191	÷16	512ワード	×1	8192点
タイマ	T	10	k()点	T0 ~	$\times \frac{18}{16}$	ワード	×2	点
積算タイマ	ST	10	k()点	ST0 ~	$\times \frac{18}{16}$	ワード	×2	点
カウンタ	C	10	k()点	C0 ~	$\times \frac{18}{16}$	ワード	×2	点
データレジスタ	D	10	k()点	D0 ~	×1	ワード		—
リンクレジスタ	W	16	k()点	W0000 ~	×1	ワード		—
リンク特殊レジスタ*1	SW	16	2k(2048)点	SW0000 ~ 07FF	×1	2048ワード		—
デバイス合計						ワード (29696ワード以下)		点 (65536点以下)

* 1: システムで固定です。(変更不可)

* 2: 1 デバイスの最大点数は32k点です。

* 3: デバイス点数を、容量 (ワード) 欄に示す数字で乗算 (または除算) した数値を記入します。

付

付 3 デバイス点数割付けシート

索引

[0] ~ [9]

10 進定数 (K) 9-95
16 進定数 (H) 9-95

[A]

A5 □ B A-25
A6 □ B A-25
AnS シリーズ対応特殊機能ユニット 7-10
AnS シリーズ電源ユニット A-25
A 系 A-24
A シリーズ CPU 互換設定 付-1
A シリーズ電源ユニット A-25

[B]

B (リンクリレー) 9-20
BCD (2 進化 10 進数) 2-47
BIN (2 進数) 2-45
BL (SFC ブロックデバイス) 9-92
B 系 A-24

[C]

C (カウンタ) 9-33
CC-Link A-25
CPU ユニット A-24
C 言語コントローラユニット A-24

[D]

D (データレジスタ) 9-39
DX (ダイレクトアクセス入力) 3-13
DY (ダイレクトアクセス出力) 3-13

[E]

E (実数) 9-96
END 処理 3-3
Ethernet A-25

[F]

F (アナンシェータ) 9-14
FD (ファンクションレジスタ) 9-44
FROM/TO 命令による交信 7-5
FX (ファンクション入力) 9-43
FY (ファンクション出力) 9-44

[G]

GOT A-25
GX Configurator 7-2
GX Configurator による設定 7-2
GX Developer A-25

[H]

H (16 進定数) 9-95
H/W エラー時 CPU 動作モード 6-34
HEX (16 進数) 2-46

[I]

I (割込みポインタ) 9-83
I/O No. 指定デバイス (U) 9-93
I/O 応答時間 6-31
I/O リフレッシュ設定 2-17
I/O 割付け 4-8

[J]

J (ネットワーク No. 指定デバイス) 9-92
J □ ¥ □ (リンクダイレクトデバイス) 9-48

[K]

K (10 進定数) 9-95

[L]

L (ラッチリレー) 9-12
LED
~の消灯方法 6-118
~の表示 6-117
~の優先順位 6-119

[M]

M (内部リレー) 9-11
MELSECNET/H A-25

[N]

N (ネスティング) 9-76

[P]

P (ポインタ) 9-77
PAUSE 状態 3-4
PC RAS 設定 (1) 8-17
PC RAS 設定 (2) 8-18
PC 書込 (フラッシュ ROM) 5-28
PC ネーム設定 8-2,8-14
PC パラメータ 8-2
PC ファイル設定 8-4,8-16

[Q]

Q3 □ B A-24
Q3 □ DB A-25
Q3 □ RB A-25
Q3 □ SB A-24

Q5 □ B A-25
 Q6 □ B A-25
 Q6 □ RB A-25
 Q6 □ WRB A-25
 QA1S5 □ B A-25
 QA1S6 □ B A-25
 QA1S6ADP A-25
 QA1S6ADP+A1S5 □ B/A1S6 □ B A-25
 QA6 □ B A-25
 QA6ADP A-25
 QA6ADP+A5 □ B/A6 □ B A-25
 QI60 9-85
 Qn(H)CPU A-24
 QnHCPU A-24
 QnPHCPU A-24
 QnPRHCPU A-24
 Q シリーズ電源ユニット A-25

[R]

R (ファイルレジスタ) 9-63
 RUN 状態 3-4
 RUN 中書込み 6-53
 回路モード~ 6-53
 ファイルの~ 6-57
 RUN 中書込用確保ステップ 5-41

[S]

S (ステップリレー) 9-23
 SB (リンク特殊リレー) 9-22
 SD (特殊レジスタ) 9-47
 SFC 移行デバイス (TR) 9-92
 SFC 設定 8-8,8-22
 SFC ブロックデバイス (BL) 9-92
 SM (特殊リレー) 9-46
 ST (積算タイマ) 9-26
 STOP → RUN 時の出力モード 6-12
 STOP 状態 3-4
 SW (リンク特殊レジスタ) 9-42

[T]

T (タイマ) 9-24
 TR (SFC 移行デバイス) 9-92

[U]

U (I/O No. 指定デバイス) 9-93
 U □ ¥ G □ (インテリジェント機能ユニット
 デバイス) 9-52
 U3En ¥ G □ (マルチ CPU 間共有デバイス)
 9-55

[V]

V (エッジリレー) 9-19
 VD (マクロ命令引数デバイス) 9-94

[W]

W (リンクレジスタ) 9-40
 WDT (ウォッチドッグタイマ) 6-90
 WDT 設定 6-90

[X]

X (入力) 9-8
 X/Y 割付確認 8-12,8-24

[Y]

Y (出力) 9-10

[Z]

Z (インデックスレジスタ) 9-56
 ZR (ファイルレジスタの連番アクセス方式)
 9-72

[あ]

空きスロット点数 8-4,8-15, 付-1
 アナンシェータ (F)
 OFF 時の処理 9-18
 ON 時の処理 9-16

[い]

イニシャル処理 3-1
 インデックスレジスタ (Z)
 ~の退避 9-59
 ~の復帰 9-59
 インテリジェント機能ユニット
 ~からの割込み 6-130
 ~専用命令 7-8
 ~デバイス (U □ ¥ G □) 9-52
 ~のスイッチ設定 6-35
 インテリジェント機能ユニット設定
 8-3,8-15, 付-1

[う]

ウォッチドッグタイマ (WDT) 6-90

[え]

エッジリレー (V) 9-19
 エラー
 ~解除 6-101
 ~時の運転モード設定 8-5,8-17, 付-2
 ~時出力モード設定 6-33
 ~発生による LED 表示 6-100
 ~発生による割込み 6-99
 ~メッセージ 6-119
 エラー履歴 6-102

[か]

外部入出力の強制 ON/OFF	6-47
回路モード	2-2,6-53
カウンタ (C)	
~のリセット	9-34
カウント処理	9-33
書込み	
標準 RAM への~	5-8,5-21
標準 ROM への~	5-6,5-19
プログラムメモリへの~	5-5,5-18
メモリカードへの~	5-24
拡張子	5-2,5-14,5-39
格納可能データ	5-2

[き]

機能アップ	
二重化 CPU の~	付-11
ハイパフォーマンスモデル QCPU の~	付-8
プロセス CPU の~	付-10
ベーシックモデル QCPU の~	付-5
機能一覧	6-1
機能バージョンの確認方法	1-19
基本ベースユニット	A-24
強制 ON/OFF	6-47
共通ポインタ	9-80

[く]

クリア	
エラー履歴の~	6-102,6-103
カウント値の~	9-34
積算タイムの~	9-26
ファイルレジスタの~	9-64
ラッチリレーの~	9-12
リモートラッチの~	6-28
ローカルデバイスデータの~	9-108
グローバルデバイス	9-98

[こ]

高速 I/O リフレッシュ	6-124
高速タイマ (T)	9-25
高速バッファ転送	6-124
高速割込み	6-122
コンスタントスキャン	6-4

[さ]

サービス処理	6-140
サービス処理時間	10-13
サイズ (ファイル容量)	5-45
最大計数速度	9-36
サブルーチンプログラム	2-5
サンプリングトレース	6-73

[し]

システムプロテクト	6-104
-----------	-------

自己診断	6-92
システムモニタ	6-112
システム割り込み設定	8-4,8-15,付-1
実行時間計測	6-64
実行タイプの切換え	2-40
自動モード	4-1
自動リフレッシュ設定	7-2
ソフト JIS コード	2-51
瞬停	3-6
詳細モード	4-1
初期実行監視時間	2-21
初期実行タイププログラム	2-19
初期設定	7-2,7-5
処理時間	6-126
シリアル No. の確認方法	1-19
シリアルコミュニケーション機能	6-131
シリアルコミュニケーション設定	8-11

[す]

スイッチ設定	
インテリジェント機能ユニットの~	6-35
スキャン実行タイププログラム	2-22
スキャンタイム	2-20
スキャンタイム測定	6-69
スキャンタイムの精度	2-20,2-28
ステップリレー (S)	9-23
スリムタイプ基本ベースユニット	A-24
スリムタイプ電源ユニット	A-25

[せ]

制御系	A-24
-----	------

[そ]

増設ケーブル	A-25
増設ベースユニット	A-24

[た]

待機系	A-24
待機タイププログラム	2-31
タイマ (T)	
処理方法	9-27
精度	9-28
タイマ時限設定	8-3,8-15,付-1
ダイレクトアクセス出力	3-13
ダイレクトアクセス入力	3-13
ダイレクト方式	3-13
単精度浮動小数点データ	2-48

[て]

データのクリア処理	3-7
データレジスタ (D)	9-39
定周期実行タイププログラム	2-36
低速 END 処理	2-27
低速実行監視時間	2-30
低速実行タイププログラム	2-23

低速スキャンタイム	2-28
低速タイマ (T)	9-25
低速プログラム実行時間	2-26
手順	
ブート運転の～	5-35
プログラムを書き込むまでの～	5-10,5-27
デバイス一覧	9-1
デバイス初期値	6-141
デバイス設定	8-6,8-19
デバイス点数割付けシート	付-12
デューティ	9-36
電源二重化基本ベースユニット	A-24
電源二重化増設ベースユニット	A-24
電源二重化ベースユニット	A-24
電源ユニット	A-25

[と]

特殊リレー (SM)	9-46
特殊レジスタ (SD)	9-47
時計機能	6-14
時計データの変更	6-15
時計データの読出し	6-16
トラッキングケーブル	A-25

[な]

内部演算処理	2-50
内部システムデバイス	9-43
内部ユーザデバイス	9-5
内部リレー (M)	9-11

[に]

二重化 CPU	A-24
二重化増設ベースユニット	A-24
二重化電源ユニット	A-25
二重化パラメータ	8-27
二重化ベースユニット	A-24
入出力処理と応答遅れ	3-9
入出力番号	4-1
入出力番号の割付け	4-1
入力 (X)	9-8

[ね]

ネスティング	9-76
ネットワーク No. 指定デバイス (J)	9-92
ネットワークパラメータ	8-29

[は]

ハードウェアエラー時 CPU 動作モード設定	6-34
ハードウェアチェック	11-2,11-8
ハイパフォーマンスモデル QCPU	A-24
パスワード	6-105
パソコン CPU ユニット	A-24
バッテリー	A-25
パラメータ	

PC ～	8-2
二重化～	8-27
ネットワーク～	8-29
パラメータ有効ドライブ設定	5-40

[ひ]

標準 RAM	5-7,5-20
標準 ROM	5-6,5-19
標準 ROM 自動書込み	5-31

[ふ]

ブート運転	5-9,5-34
ブートファイル設定	5-9
ファイル	5-41
ファイル使用方法設定	2-16
ファイルのサイズ単位	5-48
ファイルのメモリ容量	5-45
ファイルレジスタ	
～アクセス方法	9-65
～のクリア処理	9-64
～ブロック切り換え方式	9-72
～連番アクセス方式	9-72
ファンクションデバイス (FX,FY,FD)	9-43
フォーマット	
ATA カードの～	5-23
SRAM カードの～	5-23
プログラムメモリの～	5-3,5-16
複数の GX Developer から RUN 中書込みする機能	6-88
複数の GX Developer から同時にモニタする機能	6-86
複数の GX Developer からのデバッグ機能	6-86
浮動小数点データ	2-48
プログラム一覧モニタ	6-64
プログラム設定	8-20
プログラムメモリ	5-3,5-16
プログラムメモリの ROM 化	5-26
プロセス CPU	A-24
プロテクト	
パスワード登録	6-105
リモートパスワード	6-107

[へ]

ベーシックモデル QCPU	A-24
ベースユニット	A-24

[ほ]

ポインタ	
共通～	9-80
ローカル～	9-78
割込み～	9-83

[ま]

マクロ命令引数デバイス (VD).....	9-94
マルチ CPU 間共有デバイス (U3En ¥ G □)	9-55
マルチ CPU 間高速基本ベースユニット.....	A-24
マルチ CPU 設定.....	8-25

[め]

メインルーチンプログラム.....	2-4
メモリカード→標準 ROM 全データ自動書込み	5-31
メモリ構成.....	5-1,5-13
メモリチェック機能.....	6-146
メモリ容量.....	5-45

[も]

モーション CPU.....	A-24
文字列.....	9-97
文字列データ.....	2-51
モニタ.....	6-37
モニタ条件の設定.....	6-38

[ゆ]

ユーザ設定のシステムエリア.....	5-4,5-17
ユニットサービス間隔時間.....	6-140
ユニットサービス間隔読出し.....	6-140
ユニット同期設定.....	8-3,8-15, 付-1

[ら]

ラッチ.....	6-8
ラッチ機能.....	6-8
ラッチクリア.....	3-8
ラッチクリア操作無効 (ラッチ (2) 先頭/最終)	3-8
ラッチクリア操作有効 (ラッチ (1) 先頭/最終)	3-8
ラッチリレー (L).....	9-12

[り]

リストモード*.....	2-2
リフレッシュ方式.....	3-10
リモート PAUSE.....	6-22
リモート RESET (リモートリセット).....	6-25
リモート RUN/STOP.....	6-19
リモート局の入出力番号.....	4-6
リモート操作.....	6-19
リモートパスワード*.....	6-107
リモートラッチクリア.....	6-28
リンクダイレクトデバイス (J □ ¥ □).....	9-48
リンク特殊リレー (SB).....	9-22
リンク特殊レジスタ (SW).....	9-42
リンクリレー (B).....	9-20
リンクレジスタ (W).....	9-40

[ろ]

ローカルデバイス.....	9-99
～のモニタ.....	6-44
データのクリア.....	9-108
ローカルポインタ.....	9-78

[わ]

割込み	
～カウンタ.....	9-37
～カウンタ先頭 No.	9-37
～プログラム.....	2-6
～プログラム一覧モニタ.....	6-68
～ポインタ (I).....	9-83
～ポインタ設定.....	8-3,8-15, 付-1
～ユニット.....	6-31
～要因一覧.....	9-85
インテリジェント機能ユニットからの～	6-130
エラー発生による～.....	9-84
高速～.....	6-122
割込プログラム/定周期プログラム設定	8-4,8-15, 付-1

保証について

ご使用に際しましては、以下の製品保証内容をご確認いただきますよう、よろしくお願いいたします。

1. 無償保証期間と無償保証範囲

無償保証期間中に、製品に当社側の責任による故障や瑕疵（以下併せて「故障」と呼びます）が発生した場合、当社はお買い上げいただきました販売店または当社サービス会社を通じて、無償で製品を修理させていただきます。ただし、国内および海外における出張修理が必要な場合は、技術者派遣に要する実費を申し受けます。また、故障ユニットの取替えに伴う現地再調整・試運転は当社責務外とさせていただきます。

【無償保証期間】

製品の無償保証期間は、お客様にてご購入後またはご指定場所に納入後 36 ヶ月とさせていただきます。ただし、当社製品出荷後の流通期間を最長 6 ヶ月として、製造から 42 ヶ月を無償保証期間の上限とさせていただきます。また、修理品の無償保証期間は、修理前の無償保証期間を超えて長くなることはありません。

【無償保証範囲】

- (1) 一次故障診断は、原則として貴社にて実施をお願い致します。ただし、貴社要請により当社、または当社サービス網がこの業務を有償にて代行することができます。この場合、故障原因が当社側にある場合は無償と致します。
- (2) 使用状態・使用方法、および使用環境などが、取扱説明書、ユーザーズマニュアル、製品本体注意ラベルなどに記載された条件・注意事項などに従った正常な状態で使用されている場合に限定させていただきます。
- (3) 無償保証期間内であっても、以下の場合には有償修理とさせていただきます。
 - ① お客様における不適切な保管や取扱い、不注意、過失などにより生じた故障およびお客様のハードウェアまたはソフトウェア設計内容に起因した故障。
 - ② お客様にて当社の了解なく製品に改造などの手を加えたことに起因する故障。
 - ③ 当社製品がお客様の機器に組み込まれて使用された場合、お客様の機器が受けている法的規制による安全装置または業界の通念上備えられているべきと判断される機能・構造などを備えていれば回避できたと認められる故障。
 - ④ 取扱説明書などに指定された消耗部品が正常に保守・交換されていれば防げたと認められる故障。
 - ⑤ 消耗部品（バッテリー、リレー、ヒューズなど）の交換。
 - ⑥ 火災、異常電圧などの不可抗力による外部要因および地震、雷、風水害などの天変地異による故障。
 - ⑦ 当社出荷当時の科学技術の水準では予見できなかった事由による故障。
 - ⑧ その他、当社の責任外の場合またはお客様が当社責任外と認めた故障。

2. 生産中止後の有償修理期間

- (1) 当社が有償にて製品修理を受け付けることができる期間は、その製品の生産中止後 7 年間です。生産中止に関しましては、当社テクニカルニュースなどにて報じさせていただきます。
- (2) 生産中止後の製品供給（補用品も含む）はできません。

3. 海外でのサービス

海外においては、当社の各地域 FA センターで修理受付をさせていただきます。ただし、各 FA センターでの修理条件などが異なる場合がありますのでご了承ください。

4. 機会損失、二次損失などへの保証責務の除外

無償保証期間の内外を問わず、以下については当社責務外とさせていただきます。

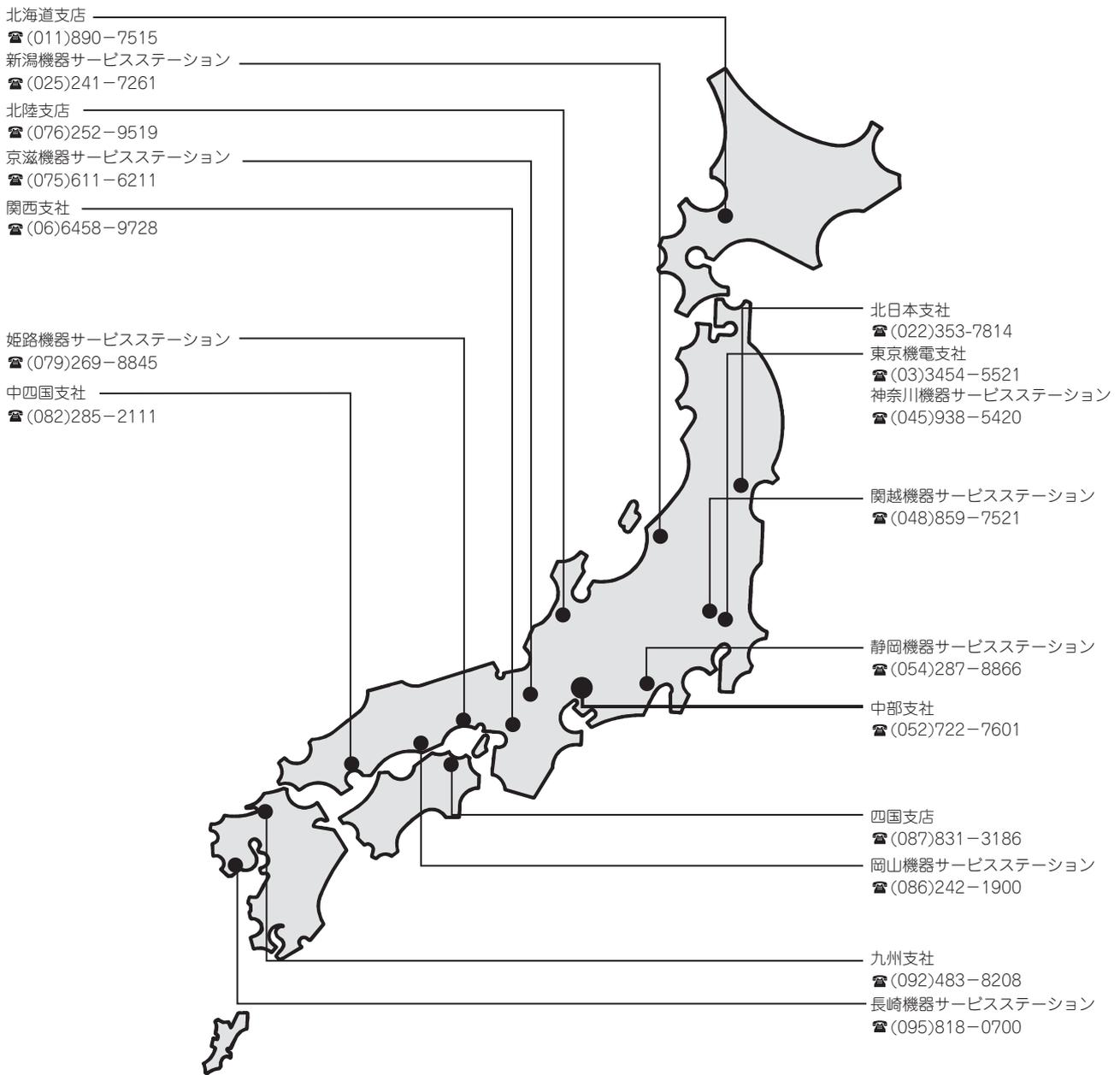
- (1) 当社の責に帰すことができない事由から生じた障害。
- (2) 当社製品の故障に起因するお客様での機会損失、逸失利益。
- (3) 当社の予見の有無を問わず特別の事情から生じた損害、二次損害、事故補償、当社製品以外への損傷。
- (4) お客様による交換作業、現地機械設備の再調整、立上げ試運転その他の業務に対する補償。

5. 製品仕様の変更

カタログ、マニュアルもしくは技術資料などに記載の仕様は、お断りなしに変更させていただく場合がありますので、あらかじめご承知おきください。

以 上

サービスネットワーク（三菱電機システムサービス株式会社）



Microsoft および Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

本文中における会社名、システム名、製品名などは、一般に各社の登録商標または商標です。

本文中で、商標記号 (™, ®) は明記していない場合があります。

三菱電機株式会社 〒100-8310 東京都千代田区丸の内2-7-3 (東京ビル)

お問い合わせは下記へどうぞ

本社機器営業部	〒110-0016	東京都台東区台東1-30-7 (秋葉原アイマークビル)	(03) 5812-1450
北海道支社	〒060-8693	札幌市中央区北二条西4-1 (北海道ビル)	(011) 212-3794
東北支社	〒980-0013	仙台市青葉区花京院1-1-20 (花京院スクエア)	(022) 216-4546
関東支社	〒330-6034	さいたま市中央区新都心11-2 (明治安田生命さいたま新都心ビル)	(048) 600-5835
新潟支店	〒950-8504	新潟市中央区東大通2-4-10 (日本生命ビル)	(025) 241-7227
神奈川支社	〒220-8118	横浜市西区みなとみらい2-2-1 (横浜ランドマークタワー)	(045) 224-2624
北陸支社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル)	(076) 233-5502
中部支社	〒450-6423	名古屋市中村区名駅3-28-12 (大名古屋ビルディング)	(052) 565-3314
豊田支店	〒471-0034	豊田市小坂本町1-5-10 (矢作豊田ビル)	(0565) 34-4112
静岡支店	〒422-8067	静岡市駿河区南町14-25 (エスパティオビル)	(054) 202-5630
関西支社	〒530-8206	大阪市北区大深町4-20 (グランフロント大阪タワーA)	(06) 6486-4122
中国支社	〒730-8657	広島市中区中町7-32 (ニッセイ広島ビル)	(082) 248-5348
四国支社	〒760-8654	高松市寿町1-1-8 (日本生命高松駅前ビル)	(087) 825-0055
九州支社	〒810-8686	福岡市中央区天神2-12-1 (天神ビル)	(092) 721-2247

三菱電機 FA
検索

www.MitsubishiElectric.co.jp/fa

メンバー登録無料!

インターネットによる情報サービス「三菱電機FAサイト」

三菱電機FAサイトでは、製品や事例などの技術情報に加え、トレーニングスクール情報や各種お問い合わせ窓口をご提供しています。また、メンバー登録いただくとマニュアルやCADデータ等のダウンロード、eラーニングなどの各種サービスをご利用いただけます。

三菱電機FA機器電話、FAX技術相談

●電話技術相談窓口 受付時間*1 月曜～金曜 9:00～19:00、土曜・日曜・祝日 9:00～17:00

対象機種	電話番号	対象機種	電話番号
自動窓口案内	052-712-2444	SCADA MC Works64	052-712-2962**2**6
エッジコンピューティング製品	産業用PC MELIPC Edgecross対応ソフトウェア (MTConnectデータコレクタを除く) 052-712-2370**2	MELSERVOシリーズ 位置決めユニット (MELSEC iQ-R/Q/L/AnSシリーズ) シンプルモーションユニット (MELSEC iQ-R/IQ-F/Q/Lシリーズ) モーションCPU (MELSEC iQ-R/Q/AnSシリーズ) センシングユニット (MR-MTシリーズ) シンプルモーションボード C言語コントローラ インタフェースユニット (Q173SCCF)/ポジションボード 052-712-6607	
シーケンサ	MELSEC iQ-R/Q/L/QnAS/AnSシーケンサ一般 052-711-5111 MELSEC iQ-F/FXシーケンサ全般 052-725-2271**3 ネットワークユニット/ シリアルコミュニケーションユニット 052-712-2578 MELSOFTシーケンサ プログラミングツール MELSOFT GXシリーズ 052-711-0037 MELSOFT統合 エンジニアリング環境 iQ Works (Navigator) 052-799-3591**2 iQ Sensor Solution MELSOFT通信支援 ソフトウェアツール MELSOFT MXシリーズ MELSEC/パソコンボード Q80BDシリーズなど 052-712-2370**2 C言語コントローラ MESインタフェースユニット/ 高速データロガーユニット 052-799-3592**2 プロセスCPU/二重化CPU (MELSEC-Qシリーズ) プロセスCPU/二重化機能 SIL2プロセスCPU (MELSEC iQ-Rシリーズ) MELSOFT PXシリーズ 052-712-2830**2**3 安全シーケンサ (MELSEC iQ-R/QSシリーズ) 安全コントローラ (MELSEC-WSシリーズ) 052-712-3079**2**3 電力計測ユニット/ 絶縁監視ユニット QEシリーズ/REシリーズ 052-719-4557**2**3	サーボ/位置決めユニット/ シンプルモーションユニット/ モーションコントローラ/ センシングユニット/ 組込み型サーボシステム コントローラ センサレスサーボ FR-E700EX/MM-GKR 052-722-2182 インバータ FREQROLシリーズ 052-722-2182 三相モータ 三相モータ225フレーム以下 0536-25-0900**2**4 産業用ロボット MELFAシリーズ 052-721-0100 電磁クラッチ・ブレーキ/テンションコントローラ 052-712-5430**5 データ収集アナライザ MELQIC IU1/IU2シリーズ 052-712-5440**5 低圧開閉器 MS-Tシリーズ/MS-Nシリーズ US-Nシリーズ 052-719-4170 低圧遮断器 ノーヒューズ遮断器/ 漏電遮断器/ MDUブレーカ/ 気中遮断器 (ACB) など 052-719-4559 電力管理用計器 電力量計/計器用変成器/ 指示電気計器/管理用計器/ タイムスイッチ 052-719-4556 省エネ支援機器 EcoServer/E-Energy/ 検針システム/ エネルギー計測ユニット/ B/NETなど 052-719-4557**2**3 小容量UPS (5kVA以下) FW-Sシリーズ/FW-Vシリーズ/ FW-Aシリーズ/FW-Fシリーズ 052-799-9489**2**6	
FAセンサ MELSENSOR	レーザ変位センサ ビジョンセンサ 052-799-9495**2		
表示器 GOT	GOT2000/1000シリーズ MELSOFT GTシリーズ 052-712-2417		

お問い合わせの際には、今一度電話番号をお確かめの上、お掛け間違いのないようお願い致します。
 *1: 春季・夏季・年末年始の休日を除く *2: 土曜・日曜・祝日を除く *3: 金曜は17:00まで *4: 月曜～木曜の9:00～17:00と金曜の9:00～16:30
 *5: 受付時間9:00～17:00 (土曜・日曜・祝日・当社休日を除く) *6: 月曜～金曜の9:00～17:00

●FAX技術相談窓口 受付時間 月曜～金曜 9:00～16:00 (祝日・当社休日を除く)

対象機種	FAX番号	対象機種	FAX番号
電力計測ユニット/絶縁監視ユニット (QEシリーズ/REシリーズ)	084-926-8340	低圧遮断器	084-926-8280
三相モータ225フレーム以下	0536-25-1258**7	電力管理用計器/省エネ支援機器/小容量UPS (5kVA以下)	084-926-8340
低圧開閉器	0574-61-1955		

三菱電機FAサイトの「仕様・機能に関するお問い合わせ」もご利用ください。
 ※7: 月曜～木曜の9:00～17:00と金曜の9:00～16:30 (祝日・当社休日を除く)

本マニュアルは、輸出する場合、経済産業省への役員取引許可申請は不要です。

SH(名)-080803-O(1904)MEE

形名: QNCPU-U-KP-J
 形名コード: 13JY95

2019年4月作成
 標準価格 4,000円

本マニュアルは、お断りなしに仕様を変更することがありますのでご了承ください。
 この標準価格には消費税は含まれておりません。ご購入の際には消費税が付加されますのでご承知置き願います。