

FAULHABER

プログラミングマニュアル

MC 5010

MC 5004

MCS

```
35 :StartOfProgram
36 LET i = i + 1
37 LET a = 6
38 IF a < 5 THEN
39   LET c = 7
40 ELSE
41   LET c = 123456789
42   GOSUB GosubExample
43 ENDIF
44 IF (i % 10) = 0 THEN
45   LET j = j + 1
46 ENDIF
47 FOR k = (8 + 5) TO 15
48   LET l = k
49 NEXT k
50 REM Check Timer Abort
51 IF t = 1 THEN
52   SAVE i, j
53   DI_EVT
54   END
55 ENDIF
```

```
IF (i % 10) = 0 THEN
  LET j = j + 1
  SAVE i, j
ENDIF
FOR k = (8 + 5) TO 15
  LET l = k
```

JP

WE CREATE MOTION

インプリント

バージョン:

15-04-2016

Copyright

Fritz Faulhaber GmbH & Co. KG

Daimlerstr.23 / 25 · 71101 Schönaich

翻訳を含む転載禁止。

本マニュアルの全部または一部を、Dr. Fritz Faulhaber GmbH & Co. KGの明示的な書面による同意なしに、情報システムへ複製、再生、保存、あるいは他の形式に加工または転送することは禁止されています。

本マニュアルは細心の注意を払って作成されています。

ただし、Dr. Fritz Faulhaber GmbH & Co. KGは本マニュアルの誤記および誤記によって発生した結果に対して何ら責任を負わないものとします。同様に、機器の不適切な使用による直接的損失および結果的損害に関しても何ら責任を負わないものとします。

本ソフトウェアを使用する際には、安全工学および干渉抑制に関連する規制ならびに本マニュアルに記載された要件に留意し遵守する必要があります。

仕様は予告なしに変更されることがあります。

最新版のテクニカルマニュアルは、以下のFAULHABERのインターネットサイトから入手できます：
www.faulhaber.com

目次

1	本マニュアルについて	4
1.1	本マニュアルの目的	4
1.2	関連文書	4
1.3	略語一覧	4
1.4	本マニュアルで使用する記号とマーク	5
2	はじめに	6
3	プログラミング言語の特性	7
3.1	コマンドセット	8
3.2	演算子と特殊文字	10
3.3	プログラムの作成方法	11
4	Motion Managerを使用したシーケンスプログラムの開発	12
4.1	プログラムの編集	13
4.2	プログラムのコントローラへのロードと実行	14
4.3	プログラムデバッグ	15
5	シーケンスプログラムの制御	16
5.1	インターフェース経由の制御	16
5.2	シーケンスプログラムの自動的な起動	17
5.3	シーケンスプログラムとのデータ交換	17
6	プログラム例	18
6.1	ステップシーケンス	18
6.2	イベント処理	19

1 本マニュアルについて

1.1 本マニュアルの目的

本マニュアルは、FAULHABER Motion Managerを使用して、モーションコントローラおよびモーションコントロールシステムファミリーV3.0のシーケンスプログラムのプログラミングについて説明します。

本マニュアルの読者はプログラミングの知識があるソフトウェア開発者および駆動機器のテクノロジープロジェクトのエンジニアが対象です。

本マニュアルの全ての情報は、標準バージョンの駆動機器を基本に説明しています。顧客固有のバージョンについては添付書類を参照してください。

1.2 関連文書

FAULHABER製品のコミッショニングおよび運転時の操作に関する追加情報は、以下のマニュアルを参照してください。

マニュアル	説明
Motion Manager 6	FAULHABER Motion Manager PCソフトウェア取扱説明書
クイックスタートガイド	FAULHABERモーションコントローラのコミッショニングおよび運転の最初の手順の説明
機能マニュアル	駆動機器の運転モードおよび機能の説明

マニュアルは以下のサイトからpdf形式でダウンロードできます：www.faulhaber.com/manuals/

1.3 略語一覧

略語	意味
BASIC	初心者向け汎用記号命令コード
EEPROM	消去可能なプログラマブルROM
Sxx	符号付（正の数と負の数）データ型、ビットサイズxx
Uxx	符号なし（正の数）データ型、ビットサイズxx

1.4 本マニュアルで使用する記号とマーク

注意!



機器が損傷する危険

- ▶ 上記の状況を回避するための措置



運転に対する理解を深めるヒントや適切な指示の情報

- ✓ 必要な操作の前提条件
- ▶ 単発の操作
- 1. 必要な操作の最初の手順
 - ↳ 手順の結果
- 2. 必要な操作の次の手順
 - ↳ 操作の結果

2 はじめに

シーケンスプログラムはFAULHABER Motion Managerによってコントローラに転送され、コントローラで直接実行することができます。これにより、監視コントローラを使用しないスタンドアロン動作が可能になります。また、さらに小さなプログラムシーケンスを半自律的に実行できます。

シーケンスプログラムはBASICプログラミング言語でプログラム化され、FAULHABER固有の拡張機能を備えています。

最大で約2000行のコードを持つ8つのユーザプログラムを利用できます。オプションで、起動時に1つのプログラムを自動的に起動することもできます。

3 プログラミング言語の特性

- 制限されたコマンドセットおよびFAULHABER固有の拡張機能を持つBASICインタプリタ
- 全てのプログラムで共通して使用される合計26個の32ビットグローバル変数（a~z）のセット
- 恒常的な保存とロードが可能な変数
- 行番号はなくラベルにジャンプ
- 行の先頭に配置されコロンの開始されるジャンプラベル行
- 大文字と小文字の区別（大文字が通例のコマンド、小文字が通例の変数）
- オブジェクトディクショナリでのオブジェクトへの読み込み／書き込みアクセス
- 正常なプログラム実行時にイベントに応答する能力
- 時間測定および待機ループのためのタイマ
- 数値演算、比較演算、およびビット演算子
- 16進数で表現される値の特殊文字\$
- 1つのプログラムの最大長: 4Kバイト
- 全てのプログラムの最大長: 16Kバイト

3.1 コマンドセット

表1: 標準のBASICコマンドセット

コマンド	機能	例
END	終了プログラム	END
GOTO	指定したラベルにジャンプする。 以下の構文では使用できません: <ul style="list-style-type: none"> IF THEN ELSE ENDIF GOSUB RETURN FOR TO NEXT 	GOTO Start
GOSUB RETURN	指定されたラベルのサブプログラムにジャンプする。 実行後、呼び出し元の位置に戻る。 サブプログラムからのGOTOジャンプは実行できません。	GOSUB Step1 :Step1 RETURN
FOR TO NEXT	ループのプログラミング。 FORループからの条件付きGOTOジャンプは実行できません。	FOR i = 1 TO 10 NEXT I
IF THEN ELSE END IF	ブランチのプログラミング。 IF命令からのGOTOジャンプは実行できません。	IF a > 3 THEN b = 1 ELSE b = 0 ENDIF
REM	コメント。 行の先頭に配置して対象の行の終了まで適用。	REM Comment
IF THEN GOTO IF THEN GOSUB	条件付きジャンプ、またはサブプログラムへの分岐。 ENDIFなしの行で使用。 以下の構文では使用できません: <ul style="list-style-type: none"> IF THEN ELSE ENDIF GOSUB RETURN FOR TO NEXT 	IF z=1 THEN GOSUB Step1
IF THEN EXIT FOR IF THEN EXIT EVT	FORループまたはイベントルーチンからのジャンプ。 ENDIFなしの行で使用。 以下の構文では使用できません: <ul style="list-style-type: none"> IF THEN ELSE ENDIF GOSUB RETURN 	FOR a = 1 TO 5 IF x = 1 THEN EXIT FOR NEXT a
IF THEN EXIT GOSUB	サブプログラムからのジャンプ。 ENDIFなしの行で使用。	:Sub1 IF x = 1 THEN EXIT GOSUB

コマンド	機能	例
	以下の構文では使用できません: <ul style="list-style-type: none"> ▪ IF THEN ELSE ENDIF ▪ GOSUB RETURN 	RETURN

表2: FAULHABERコマンドの拡張機能

コマンド	機能	例
SETOBJ	オブジェクトをオブジェクトディクショナリに書き込む。 構文: SETOBJ <Index>.<Subindex> = <variable or value>	SETOBJ \$6083.\$00 = 500
GETOBJ	オブジェクトディクショナリのオブジェクトを読み込む。 構文構文: <variable> = GETOBJ <Index>.<Subindex>	a = GETOBJ \$6083.\$00
DEF_EVT_VAR	イベントの発生時に、イベントステータスビットマスクの 値を返す変数を定義する。	DEF_EVT_VAR e
EN_EVT ¹⁾	オブジェクト0x2324.01（イベント処理）の変更に伴う機器 の状態によってトリガーされるイベントルーチンのアクティ ブ化。 構文: EN_EVT <bit mask>,<event mark>	EN_EVT \$ffffff, EvHandler ¹⁾
DI_EVT	並列で実行される処理の全てのイベントの非アクティブ化。 構文: DI_EVT	DI_EVT
RET_EVT	イベントルーチンから戻る。	: EvHandler RET_EVT
SAVE	1つまたはそれ以上の変数のEEPROMへの恒常的な保存 （コンマ区切りリスト）。	SAVE a, b, z
LOAD	前回保存した1つまたはそれ以上の変数のEEPROMからの ロード（コンマ区切りリスト）。	LOAD a, b, z
DEF_TIM_VAR	タイマとして使用する変数を定義する。 構文: DEF_TIM_VAR <variable1>	DEF_TIM_VAR t
START_TIM	ミリ秒単位の値で指定するタイマの開始（値が0の場合、 タイマの停止） 構文: START_TIM <変数2または値> 指定時間の経過後のタイマ変数は1、それ以外は0（タイマ の継続実行）。	START_TIM 3000 IF t = 1 THEN ENDIF

1) 1度に起動できるイベントハンドラは、1つのみです。

3.2 演算子と特殊文字

算術演算子	
加算	+
減算	-
乗算	*
除算	/
モジュロ (剰余)	%

論理演算子	
論理積 (A かつ B)	AND
論理和 (A または B)	OR
反転	NOT

比較演算子	
より大きい	>
より小さい	<
に等しい	=
に等しくない	<>
以上	>=
以下	<=

ビット演算子	
ビットごとのAND	&
ビットごとのOR	
ビットごとの反転	~

代入演算子	
代入演算子	=

特殊文字	意味
()	算術演算子に使用
,	EN_EVTおよびSAVE/LOADで使用
.	SETOBJ/GETOBJの区切り文字
\$	16進数
:	ジャンプラベル。行の先頭に配置

3.3 プログラムの作成方法

- シーケンスプログラムは常に、実行コード全体を包含するメインループを持つ一連のステップとして、構成する必要があります（6.1節、18ページ参照）。Waitループには、特定のイベントで制御される条件付きジャンプを含めることはできません。
- シーケンスプログラムは、FAULHABER Motion Managerで作成および編集されます。
- シーケンスプログラムをコントローラにダウンロードする前に、FAULHABER Motion Managerはジャンプラベルのアドレスや必要なメモリエリアを決定するために、前処理のステップを実行します。
- FAULHABER Motion Managerには、シーケンスプログラムの作成、編集、コントローラへの転送機能のほかに、プログラミングエラーの検証および修正機能（デバッグ機能）もあります。

4 Motion Managerを使用したシーケンスプログラムの開発

FAULHABER Motion ManagerのEditorウィンドウは、シーケンスプログラムの統合開発環境を提供します。開発環境には以下の機能があります:

- 構文の強調表示
- 最大で8つのユーザプログラムの表示と編集
- 機器のメモリーやPCメモリーからのユーザプログラムのロードと表示
- 各シーケンスプログラムの起動
- 起動状態のシーケンスプログラムの停止
- 起動状態のシーケンスプログラムの中断
- シングルステップの実行
- ブレークポイントの定義
- 現在のプログラム状態および現在のプログラム行の表示
- 変数の内容の監視および変更
- アクセスコードによる読み込み保護
- ユーザ独自のプログラムで使用するコードモジュール

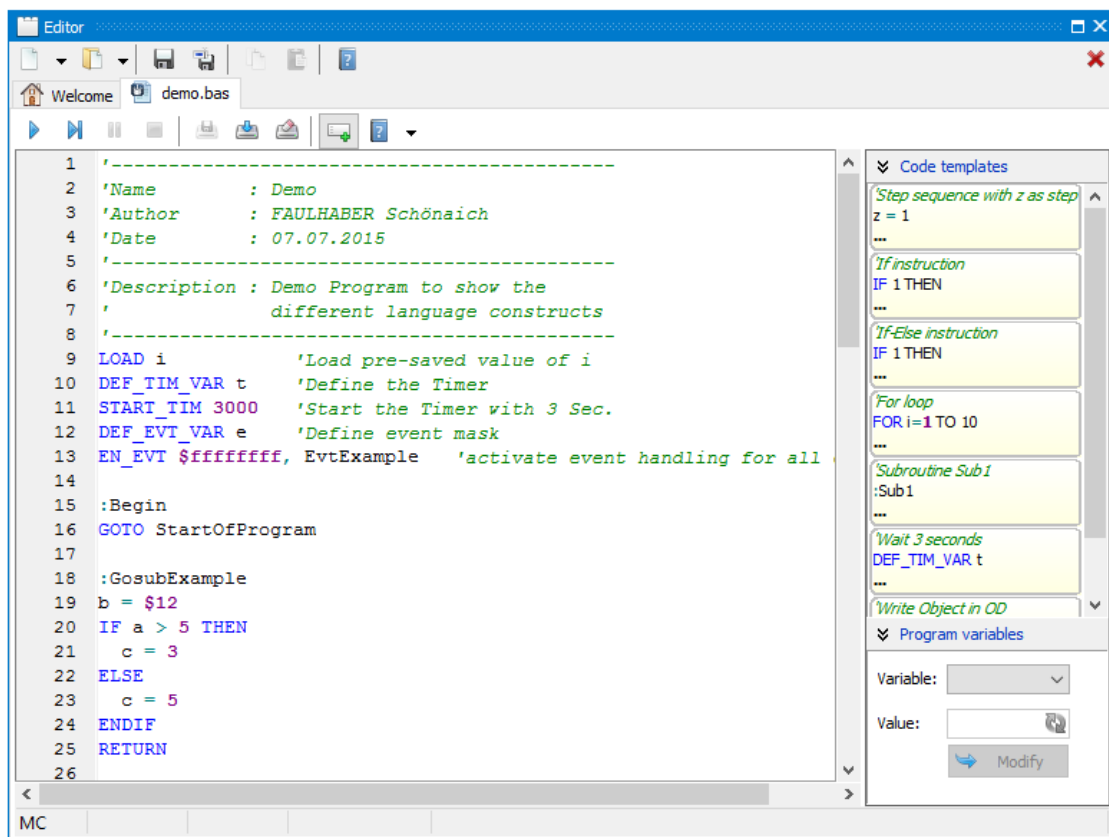











表3: 各ボタンの機能（編集環境）

ボタン	名前	機能
	実行	シーケンスプログラムをコントローラにダウンロードして実行します。
	ステップ	シーケンスプログラムをシングルステップで実行します。
	中断	シーケンスプログラムを中断します。
	停止	シーケンスプログラムを停止します。
	EEPROM保存	現在ロードされているシーケンスプログラムを、EEPROMに保存します。
	ダウンロード	シーケンスプログラムをコントローラにダウンロードし、EEPROMに保存します。
	削除	シーケンスプログラムをコントローラのEEPROMから削除します。
	Extras	コードテンプレートを表示し、各変数の内容を検証、変更する機能を提供します。

4.1 プログラムの編集

- **File - New - Motion Control file MC V3.x** コマンドを実行すると、新しいプログラムが作成されます。
- **File - Open**コマンドの実行後に、使用する.basファイルを選択すると、既存のプログラムがロードされます。
- 新しいファイルを作成すると、事前にファイルヘッダが挿入された空の.basファイルが生成されます。このファイルを編集し任意のメモリー位置に保存します。
- コントローラにダウンロードされた保存済みの各ファイルは、Node Explorerをダブルクリックしてアクティブなノードのシーケンスプログラムディレクトリから開くことができます。

宣言されたプログラムリンクとは関係なく、**Upload**ノードをダブルクリックして、コントローラに保存された全てのプログラムをロードする機能もあります。アップロードされたプログラムに、適切なファイルリンクが存在する場合は、ファイルの内容がNode Explorerに表示されます。それ以外の場合は、機器から読み取られたプログラムの内容が表示されます。

-  リンクされたファイルではなく、機器のメモリーの内容を表示する場合は、Node Explorerのファイルリンクを削除する必要があります（**Delete**キー）。次にファイルをダウンロードすると、Node Explorerでリンクが再確立されます。
- 『』（アポストロフィー）で開始されるコメント行は、文書化が目的です。コントローラにはダウンロードされません。
- Editorウインドウ（**Extras**ボタン）の右側に、ユーザのサポート用のコラムを表示することができます。このコラムは、コードのテンプレートが表示され、マウスで現在のプログラムにドラッグし、プログラムを容易に作成できます。

4.2 プログラムのコントローラへのロードと実行

- **Run**ボタンを押すと、完成したプログラムがコントローラにダウンロードされ、直ちに実行されます。
- このプログラムがコントローラに保存されていない新しいプログラムの場合は、1~8のプログラム番号を割り当てる必要があります。プログラムは、割り当てられたプログラム番号でコントローラに保存されます。
- すでにファイルを任意のディレクトリに保存している場合は、このファイルへのリンクが、Node Explorerのアクティブなノードのシーケンスプログラムの下に生成されます。割り当てられたプログラム番号（プログラム番号1を示すP1など）も表示されます。
- プログラムの実行を開始すると、Editorエリアは背景色が変わり、デバッグモードに切り替わります。デバッグモードでは、プログラムの編集はできません。Editorウインドウのステータス行には、次の情報が表示されます
 - 現在ロードされているプログラムの番号
 - 現在のプログラム行
 - プログラムの状態（実行中など）
- プログラムの実行中にエラーが発生した場合、プログラムの実行は中断され、最後に実行された行が強調表示されます。
- プログラム編集モードに戻るには、**Stop**ボタンを押してプログラムの実行を終了する必要があります。
- **Run**ボタンによってロードされたプログラムは、プログラム番号が割り当てられている場合のみ不揮発性メモリーに保存されます。その後の変更内容は、作業メモリーにのみ保存されます。変更内容をコントローラに恒常的に保存するには、**EEPROM Save**ボタンを押します。
- **Download**ボタンを押すと、プログラムを任意のプログラム番号で保存できます。
- **Delete**キーを押すと、選択されたプログラムが機器のメモリーから削除されます。
- 使用しないプログラムリンクは、Node Explorerで選択し**Delete**キーを押すかコンテキストメニューから削除できます。

4.3 プログラムデバッグ

シーケンスプログラムのトラブルシューティングには、以下のデバッグ機能を利用できます:

- 現在の実行位置でのプログラムの中断 (**Halt**ボタン) :
 - アクティブ行がエディタで強調表示されます。
 - 編集エリアは非アクティブ状態です。
 - **Halt**の後、**Run**によってプログラムを継続できます。また、**Step**でステップごとに実行することもできます。**Stop**を押すと、プログラムは編集モードに戻ります。
- プログラムの1ステップごとの実行 (**Step**ボタン) :
 - 次の行のプログラムのみが実行されます。
 - 新しいアクティブ行がエディタで強調表示されます。
 - 編集エリアは非アクティブ状態です。
 - **Step**の後、**Run**によってプログラムを継続できます。また**Step**でステップごとに実行することもできます。**Stop**を押すと、プログラムは編集モードに戻ります。
- ブレークポイントでのプログラムの中断 :
 - ブレークポイントは、ウインドウの左端にある行番号をクリックすると設定できます。
 - プログラムの実行は、この行で中断します。プログラムの実行は、**Run**または**Step**で継続できます。**Stop**を押すと、プログラムは編集モードに戻ります。
 - ウインドウの左端のブレークポイントをクリックすると、ブレークポイントが削除されます。いずれかの位置にブレークポイントが設定されていると、別の位置にブレークポイントを設定することができません。
 - ブレークポイントは、プログラムの開始前およびプログラムの実行中に設定できます。
- 変数の内容の監視と変更:
 - Editorの右端のExtras列が表示されている場合 (**Extras**ボタン)、各変数の内容を検証することができます。必要に応じて、プログラムの実行中または中断時に変更することができます。

5 シーケンスプログラムの制御

保存したシーケンスプログラムは、監視コンピュータによりインターフェース経由で起動することができます。または、コントローラのブート時に自動的に開始することもできます。

5.1 インターフェース経由の制御

シーケンスプログラムの実行は、オブジェクト0x3001を介して監視コンピュータにより制御、監視することができます。

表4: 現在の制御パラメータセット

インデックス	サブインデックス	名前	型	属性	意味
0x3001	0	エントリ数	U8	ro	オブジェクトエントリの数
	1	プログラム制御	U8	rw	0x3001.02または0x3002.00で起動されたシーケンスプログラムの制御: <ul style="list-style-type: none"> 1: 起動されたプログラムをEEPROMからのロード (ロード) 2: ロードされたプログラムの開始または継続 (実行) 3: 各プログラム行の実行 (ステップ) 4: 実行中のプログラムの中断 (ブレイク) 5: 実行中のプログラムの停止 (終了)
	2	プログラム番号	U8	rw	プログラム番号でのシーケンスプログラムの起動
	3	実行位置	U16	ro	現在実行中の行のアドレス
	4	プログラム実行状態	U8	ro	プログラムの現在の状態: <ul style="list-style-type: none"> 0: 動作なし (アイドル) 1: 現在、EEPROMからプログラムをロード中 (読み込み中) 2: 現在、EEPROMにプログラムを保存中 (保存中) 3: 現在、プログラムを削除中 (削除中) 4: 現在、プログラムを実行中 (実行中) 5: プログラムの中断 (中断済み)
	8	エラー状態	U8	ro	エラー状態: <ul style="list-style-type: none"> 0: エラーなし (エラーなし) 1: 構文エラー (構文解釈エラー) 2: EEPROMへのアクセスエラー (EEPROMアクセスエラー)

i 新しいプログラムをロードするには、現在実行中のプログラムを停止する必要があります。疑似コード:

- If 0x3001.04 = 4 (Running) or 0x3001.04 = 5 (Halted), then 0x3001.01 = 5 (Terminate)
- Wait until 0x3001.04 = 0 ("Idle")

プログラム番号1のシーケンスプログラムをロードし実行する例：

1. プログラム1を選択します：
 - 0x3001.02 = 1 (P1)
 2. プログラムをロードします：
 - 0x3001.01 = 1 (Load)
 - 0x3001.04 = 0になるまで待機します（読み込みをしません）。
 3. プログラムを実行します：
 - 0x3001.01 = 2 (Run)
- 👉 プログラム1がロードされ、実行されます。

5.2 シーケンスプログラムを自動的に起動

オブジェクト0x3002.00を使用すると、プログラム番号の入力ができます。コントローラが起動すると、このプログラムは自動的に起動します。

表5: 自動で起動するプログラム番号

インデックス	サブインデックス	名前	型	属性	意味
0x3002	0	自動で起動するプログラム番号	U8	rw	自動的に起動されるシーケンスプログラムのプログラム番号。

i この機能は、Motion Managerの駆動機能ダイアログからも利用できます（**Device control - Sequence programs**）。

5.3 シーケンスプログラムとのデータ交換

プログラム変数a~zは、シーケンスプログラムと監視コンピュータ間のデータ交換に使用することもできます。オブジェクト0x3004.01は変数を選択、その値の読み込み／書き込みにはオブジェクト0x3004.02を使用します。

表6: 変数アクセス

インデックス	サブインデックス	名前	型	属性	意味
0x3004	0	エントリ数	U8	ro	オブジェクトエントリの数
	1	変数インデックス	U8	rw	変数インデックス 0 = a, 1 = b,...
	2	変数値	S32	rw	変数値

6 プログラム例

6.1 ステップシーケンス

以下にステップシーケンスの実装例を示します。メインループは連続して処理されます。現在のステップの条件を満たすと、直ちにステップカウンタ (z) が1つ増分され、次のステップに進み、さらに条件がテストされます。

```
REM Sequence of steps with z as the step counter
```

```
DEF_TIM_VAR t
```

```
z = 1
```

```
:MainLoop
```

```
IF z = 1 THEN GOSUB Z1
```

```
IF z = 2 THEN GOSUB Z2
```

```
IF z = 3 THEN GOSUB Z3
```

```
IF z = 4 THEN
```

```
    END
```

```
ENDIF
```

```
GOTO MainLoop
```

```
:Z1
```

```
`Step 1:
```

```
'ステップカウンタ-zが1つ増えます。
```

```
a = GETOBJ $6041.$00
```

```
IF a = $0023 THEN 'デバイスコントローラの状態 = "Switched On"
```

```
    z = 2
```

```
ENDIF
```

```
RETURN
```

```

:Z2
`Step 2: `(ここでは "Target Reached") タイマーが設定され、ステップカウンタ-zが1つ増えます。
a = GETOBJ $6041.$00
IF (a & $400) = $400 THEN
`ステータスワードにBit10をセット ("Target Reached")
  START_TIM 2000 `タイマーを2秒にセット
  z = 3
ENDIF
RETURN

:Z3
`Step 3:タイマーが切れた時,
`ステップカウンタ-zが1つ増えます。
IF t = 1 THEN `タイマーがダウンします。
  z = 4
ENDIF
RETURN

```

6.2 イベント処理

以下のプログラムの一部は、**Temperature warning limit reached**（温度警告制限に達しました）イベントに対するプログラムの反応を示しています。

```

DEF_EVT_VAR e `イベントマスクを定義する
EN_EVT $00030000, EvtOverTemp `過度な温度に対するイベント処理を有効にする
:EvtOverTemp
IF e & $00020000 THEN
  END
ELSE
  w = 1 `温度警告、変数wをセット
ENDIF
RET_EVT

```

〒140-0013

東京都 品川区 南大井 6-20-8

ユニゾ大森ビル 8F

新光電子株式会社

TEL. 03-6404-1003

FAX. 03-6404-1005

e-mail.motor-info@shinkoh-elecs.co.jp

www.shinkoh-faulhaber.jp

7000.05056, 15-04-2016

© DR.FRITZ FAULHABER GMBH & CO.KG

仕様は予告なしに変更されることがあります。

DR.FRITZ FAULHABER

GMBH & CO.KG