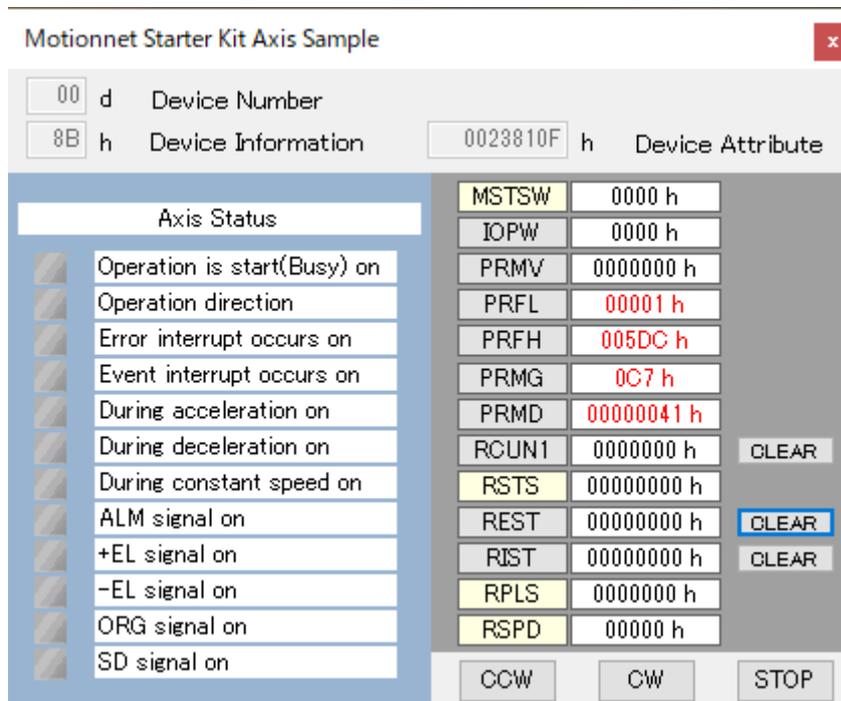


Motionnet スターターキット  
G9001A-EV, G9002A\_G9103C-EV  
取扱説明書  
AXIS サンプルプログラム



## 目次

1. はじめに .....	1
1.1 本書の取扱い .....	1
1.2 注意事項 .....	1
1.3 お願い .....	1
2. 紹介 .....	2
2.1 動作環境 .....	2
2.2 動作モード .....	2
2.3 使用したプログラミング言語 .....	3
2.4 注意 .....	3
3. サンプルプログラムの構成 .....	4
3.1 フォルダ構成 .....	4
3.2 ファイル構成 .....	4
3.3 デバイスドライバーのインストール .....	5
4. C#でのプロジェクト起動 .....	5
5. 動作説明 .....	6
5.1 プログラムの起動 .....	6
5.2 デバイス情報の表示 .....	6
5.3 ステータス情報の表示 .....	7
5.4 レジスタ情報の表示 .....	8
5.5 動作ボタン .....	9
5.5.1 CCW .....	9
5.5.2 CW .....	9
5.5.3 STOP .....	9
5.5.4 CLEAR .....	9
5.6 エラー発生時の表示 .....	9
6. ソースコード説明 .....	10
6.1 FTDI へのアクセス関数 .....	10
6.2 センターデバイス G9001A へのアクセス関数 .....	10
6.3 ローカルデバイス G9103C への初期設定 .....	11
6.4 ローカルデバイス G9103C からレジスタ内容の読出し .....	12
6.5 CCW 動作 .....	12
6.6 CW 動作 .....	13
6.7 STOP 動作 .....	13
6.8 CLEAR 動作 .....	13

# 1. はじめに

本取扱説明書は Motionnet スターターキット AXIS サンプルソフトウェア(MNET-STK\_AXIS\_Sample.exe)の仕様、機能、接続方法及び使用方法等を記載しています。

本製品を使用していただくため、必ず本書をお読みいただき、保管してください。

## 1.1 本書の取扱い

- ① 本書の全部又は一部を無断で転載することは、著作権法によって禁止されています。
- ② 本書の内容については、性能や品質の向上に伴い、将来予告なく変更することがあります。
- ③ 本書の内容については、万全を期しておりますが、万一不可解な点や誤り、並びに記載もれ等お気付きの点がありましたら、弊社営業担当へ連絡をお願いいたします。

## 1.2 注意事項

本書は、製品に含まれる機能詳細を説明するものであり、お客様の特定目的に適合することを保証するものではありません。

また、本書に記載されている応用例、回路図等は参考用です。

機器・装置の機能や安全性を確認の上、使用してください。

## 1.3 お願い

本製品は、原則として、次のいずれかの用途には、使用しないでください。使用する場合は、必ず弊社営業担当へ連絡をお願いいたします。

- ① 原子力設備、電力やガス等の供給システム、交通機関、車両設備、各種安全装置、医療機器等の高い信頼性と安全性が必要とされる設備
- ② 人命や財産に直接、危険を及ぼす可能性がある設備
- ③ カタログ、取扱説明書等に記載のない条件や環境での使用

本製品の故障により、人命や財産に重大な損害を及ぼす可能性のある用途では、冗長設計等により、高い信頼性と安全性を確保して、使用してください。

## 2. 紹介

本書は制御基板を操作する AXIS サンプルソフトウェア(MNET-STK\_AXIS\_Sample.exe)の取扱説明書です。

本ソフトウェアは、Motionnet スターターキット(G9001A-EV、G9002A\_G9103C-EV)を利用することでセンターデバイス G9001A、ローカルデバイス G9103C を使用した Motionnet 通信仕様及びモーター制御機能を学習することができます。

別途弊社の取扱説明書(下記に記載)と併せてご覧ください。

(x は版数)

	取扱説明書名【概要】	文書ファイル名	対象ソフト ファイル名	文書番号
ハードウェア 取扱説明書	Motionnetスターターキット 取扱説明書 (ハードウェア)	MotionnetStarterKit_ HardwareManual_VerxJ.p df	—	TA600036- JPx/x
	Motionnetスターターキット 取扱説明書 (簡易版)	MotionnetStarterKit_ SimpleManual_VerxJE.pdf	—	TA600035- JPx/x
アプリケーション ソフトウェア 取扱説明書	Motionnetスターターキット 取扱説明書 (アプリケーションソフトウェ ア) 【全レジスタの表示】	Motionnet Starter Kit_ ApplicationManual_VerxJ. pdf	MotionnetStarterKit_A pplication_VxxxJE.zip	TA600037- JPx/x
I/Oサンプル ソフトウェア 取扱説明書	Motionnetスターターキット I/Oサンプルソフトウェア 取扱説明書	Motionnet Starter Kit_IO_ SampleManual_VerxJ.pdf	MotionnetStarterKit_IO _Sample_Vxxx.zip	TA600048- JPx/x
Axisサンプル ソフトウェア 取扱説明書	Motionnetスターターキット AXISサンプルソフトウェア 取扱説明書	Motionnet Starter Kit_AXIS_ SampleManual_VerxJ.pdf	MotionnetStarterKit_A XIS_Sample_Vxxx.zip	TA600049- JPx/x (本書)
参考資料	G9001A/G9002A ユーザーズマニュアル			DA70109-4/x
	G9103C ユーザーズマニュアル			DA70143-1/x

アプリケーションソフト及び関係資料は、NPM ウェブサイトよりダウンロードしてください。

### 2.1 動作環境

本ソフトウェアは、Windows7、および Windows10(共に 32 bit と 64 bit)での動作確認を行っています。

(上記以外の OS については動作確認を行っておりません。)

また動作中に OS がスリープモードへ移行しないように省電力設定を変更してください。

### 2.2 動作モード

センターデバイス G9001A を USB から 8 ビットパラレルバス I/F モードで制御しています。

センターデバイス G9001A とローカルデバイス G9103C は Motionnet ケーブルにて接続し、制御します。

---

## 2.3 使用したプログラミング言語

マイクロソフト社の以下の製品を使用しています。

Microsoft Visual Studio Express 2013 for Windows Desktop (無償版)

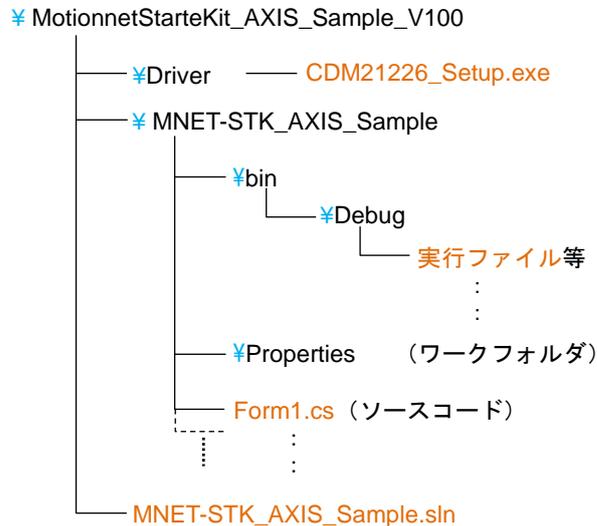
## 2.4 注意

- ・ "Microsoft Visual C#" の使用法などに関しては、お答えすることはできません。
- ・ FTDI 社製の製品の用法などに関しては、お答えすることはできません。
- ・ 本サンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承ください。

## 3. サンプルプログラムの構成

### 3.1 フォルダ構成

圧縮ファイル(MotionnetStarterKit\_AXIS\_Sample\_V100.zip)を解凍するとサンプルプログラムは下記のようなフォルダ構成になっています。



### 3.2 ファイル構成

< MotionnetStarterKit\_AXIS\_Sample\_V100 フォルダ内 >

MNET-STK\_AXIS\_Sample.sln ..... ソリューションファイル

< MotionnetStarterKit\_AXIS\_Sample\_V100Driver フォルダ内 >

CDM21226\_Setup.exe ..... デバイスドライバのインストーラ (FTDI 社製)

< MotionnetStarterKit\_AXIS\_Sample\_V100MNET-STK\_Aim\_Sample フォルダ内 >

Form1.cs	.....	ソースコード
clsFTDI.cs	.....	FTDI アクセス関数
accessMNET_Axis.cs	.....	G9001A,G9103C アクセス関数
FTD2XX_NET.dll	.....	FTDI ライブラリ
FTD2XX_NET.xml	.....	FTDI XML ドキュメント
*.bmp	.....	画像データ
その他		

< MotionnetStarterKit\_AXIS\_Sample\_V100MNET-STK\_AXIS\_SamplebinDebug フォルダ内 >

MNET-STK_AXIS_Sample.exe	.....	実行ファイル
FTD2XX_NET.dll	.....	FTDI ライブラリ (実行時に必須)
FTD2XX_NET.xml	.....	FTDI XML ドキュメント (実行時には不要)
その他	.....	ワークファイル類 (実行時には不要)

"Microsoft Visual C#" がインストールされていなくても、MNET-STK\_AXIS\_Sample.exe (実行ファイル)を実行することで AXIS サンプルプログラムが起動します。

### 3.3 デバイスドライバーのインストール

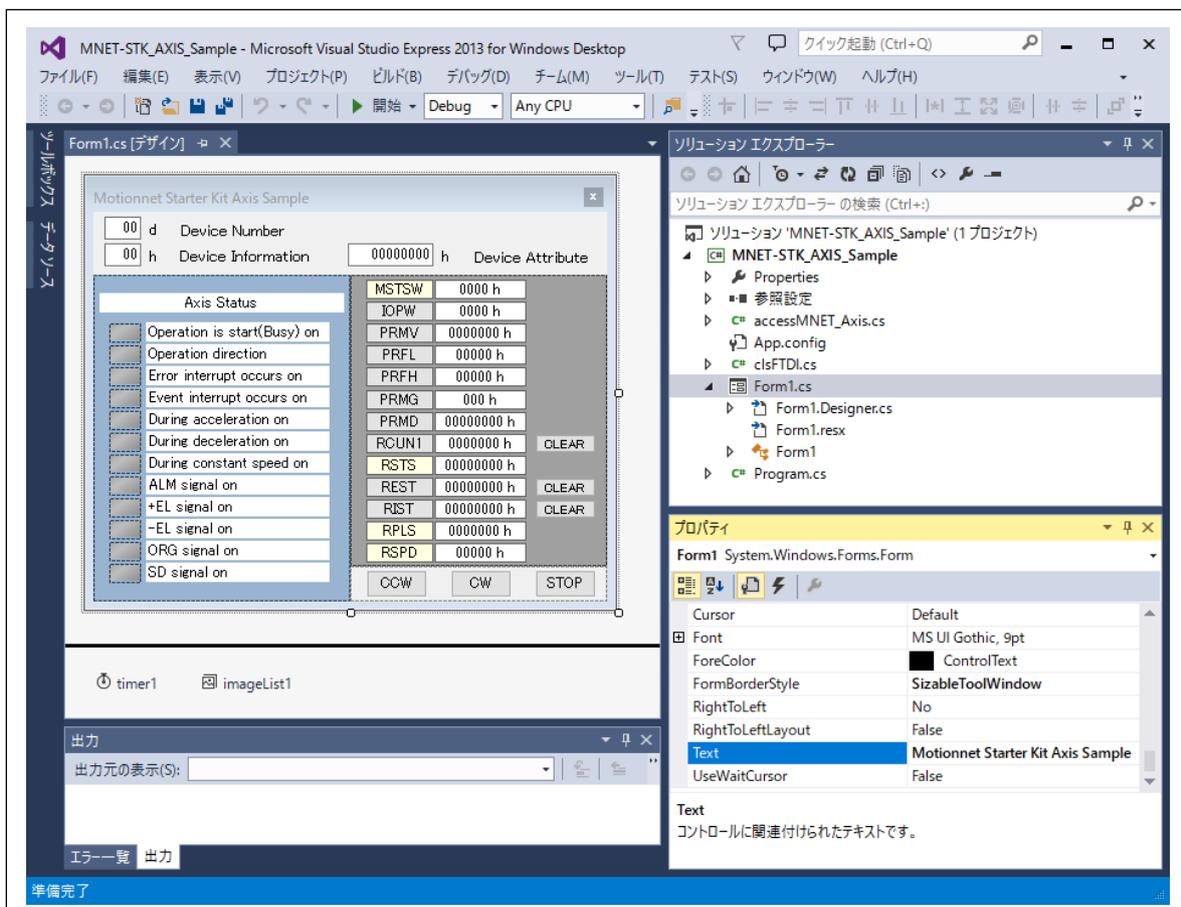
「CDM21226\_Setup.exe」をダブルクリックしてインストーラを起動し、画面の指示に従ってインストールを完了させてください。ただし、既にインストール済の場合、再度インストールする必要はありません。



注：FTDI 社の Web サイト(<http://www.ftdichip.com/Drivers/D2XX.htm>)に最新版のデバイスドライバーがある場合、そちらをダウンロードしてご利用ください。

## 4. C#でのプロジェクト起動

Motionnet スターターキット(G9001A-EV、G9002A\_G9103C-EV)がパソコンに接続されていることを確認してください。  
"Microsoft Visual C#" がインストールされていることを確認し、MNET-STK\_AXIS\_Sample.sln「ソリューションファイル」をダブルクリックしてください。

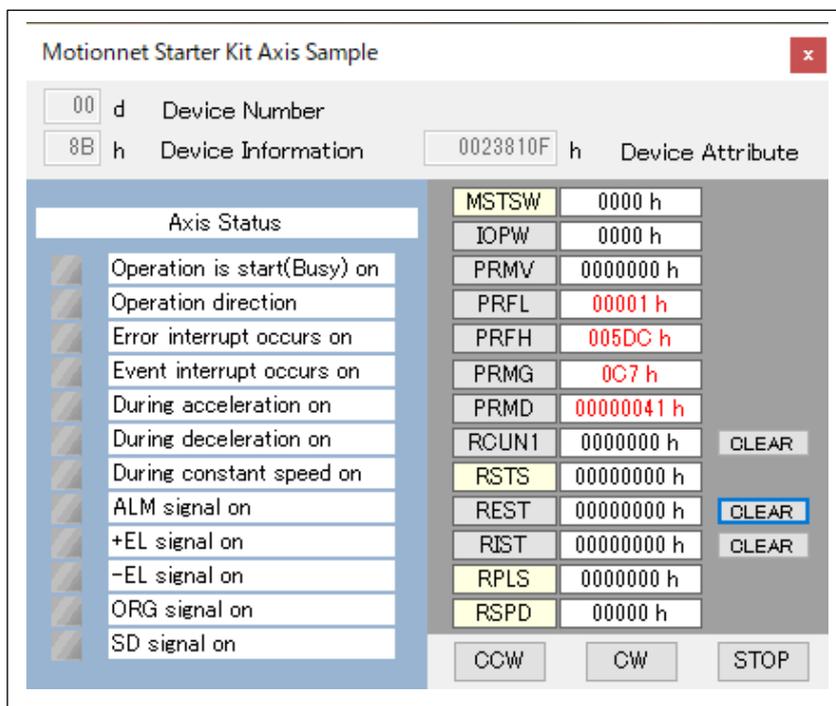


マイクロソフト製品のインストールに関しては、マイクロソフトの Web サイトを参照ください。  
プロジェクトのビルドやデバッグについても、その操作方法はマイクロソフトの Web サイトを参照ください。

## 5. 動作説明

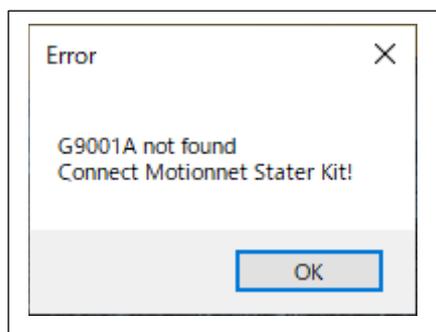
### 5.1 プログラムの起動

デバッグを開始すると、以下の画面のソフトウェアが起動します。

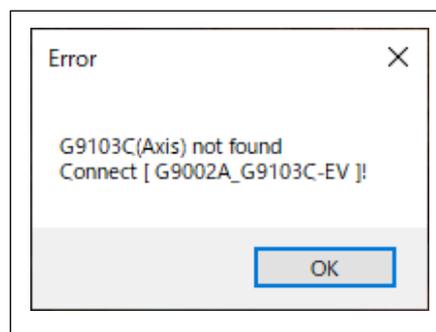


Motionnet スターターキット(G9001A-EV, G9002A\_G9103C-EV)がパソコンに接続されていない場合には、次のエラー画面を表示します。

G9001A-EV が未接続



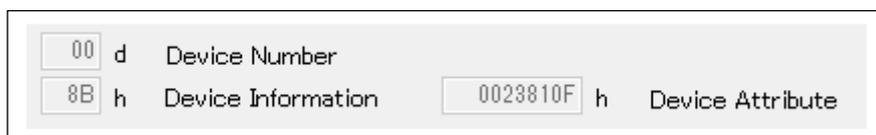
G9001A-EV は接続されているが G9002A\_G9103C-EV が未接続



[OK]ボタンを押すことにより、ソフトウェアが終了します。

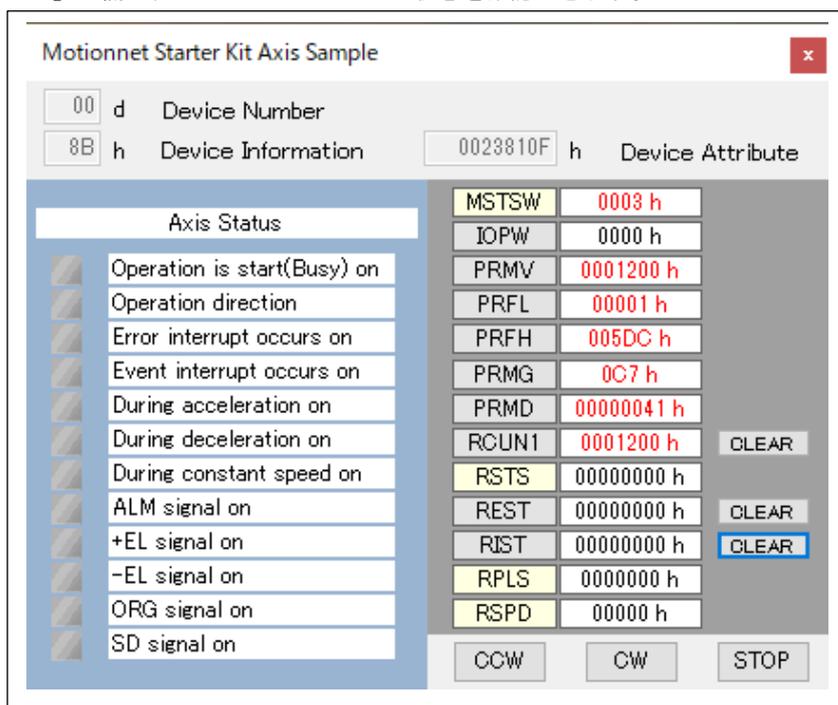
### 5.2 デバイス情報の表示

接続されている G9002A\_G9103C-EV ボード上の「G9103C」のデバイス番号、デバイス情報、デバイス属性データを表示します。



## 5.3 ステータス情報の表示

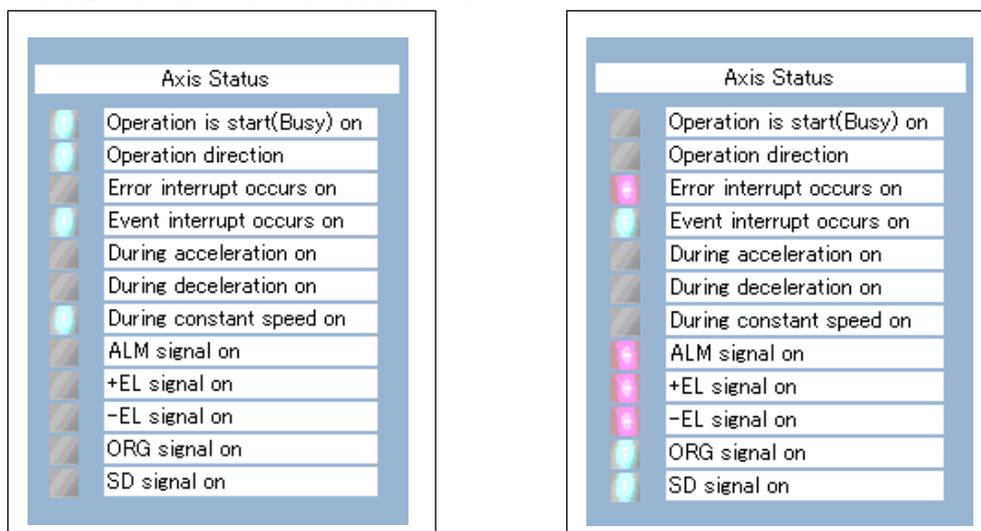
「Axis Status」の欄で、G9103Cのいくつかの状態を確認できます。



詳細は次のとおりです。

表記	内容
Operation is start(Busy) on	動作中に "1" になります。(MSTSW.SBSY) [青]
Operation direction	CW 動作で "0"、CCW 動作で "1" になります。(RSTS.SDIR) [青]
Error interrupt occurs on	エラー割込みが発生したときに "1" になります。(MSTSW.SERR) [赤]
Event interrupt occurs on	イベント割込みが発生したときに "1" になります。(MSTSW.SEVT) [青]
During acceleration on	加速中に "1" になります。(SSTSW.CND = 0x08) [青]
During deceleration on	減速中に "1" になります。(SSTSW.CND = 0x0A) [青]
During constant speed on	定速中に "1" になります。(SSTSW.CND = 0x06, 0x07, 0x09) [青]
ALM signal on	ALM 入力が ON の時に "1" になります。(SSTSW.SALM) [赤]
+EL signal on	+EL 入力が ON の時に "1" になります。(SSTSW.SPEL) [赤]
-EL signal on	-EL 入力が ON の時に "1" になります。(SSTSW.SMEL) [赤]
ORG signal on	ORG 入力が ON の時に "1" になります。(SSTSW.SORG) [青]
SD signal on	SD 入力が ON の時に "1" になります。(SSTSW.SSD) [青]

状態が "1" になると、各項目横の表示色が変化します。

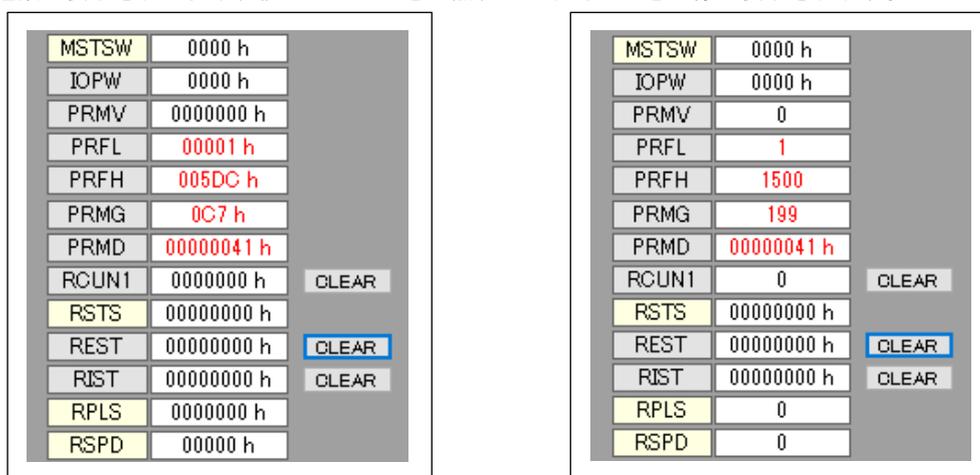


## 5.4 レジスタ情報の表示

G9103C のステータス、汎用入出力データと、いくつかのレジスタの内容が表示されています。詳細は次のとおりです。

表記	内容
MSTSW	メインステータス。
IOPW	汎用入出力データ。
PRMV	位置決め量。
PRFL	FL 速度。
PRFH	FH 速度。
PRMG	速度倍率。
RCUN1	カウンター1。
RSTS	拡張ステータス。
REST	エラー割込み要因レジスタ。
RIST	イベント割込み要因レジスタ。
RPLS	位置決めカウンター。
RSPD	現在速度モニター。

すべて 16 進数で表示されており、値がゼロのときは黒、ゼロ以外のときは赤で表示されます。



レジスタの値が表示されている部分をマウス右クリックすることで、10 進数と 16 進数の状態に切り替えることができます。

ただしビット単位で意味を持つレジスタ（RSTS など）は 16 進数に固定され、切り替えることはできません。

## 5.5 動作ボタン

### 5.5.1 CCW

クリックすると、CCW 方向への動作が開始され、パルスを 4608 回出力(モーター軸 2 回転)して停止します。  
動作開始後、約 1 秒で 1pps から 1500pps まで加速し、しばらく 1500pps で動作した後、約 1 秒かけて減速停止します。

### 5.5.2 CW

クリックすると、CW 方向への動作が開始されます。動作内容は CCW と同様です。

### 5.5.3 STOP

動作中にクリックすると、減速停止します。

### 5.5.4 CLEAR

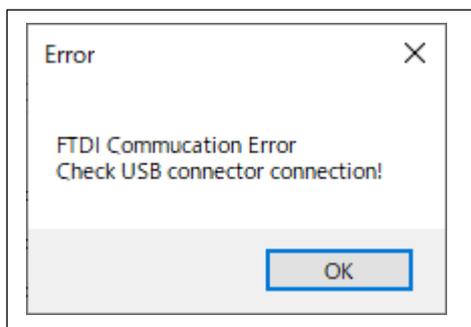
RCUN1 (カウンター1), REST (エラー割込み要因レジスタ), RIST (イベント割込み要因レジスタ) の状態をクリアできます。

各レジスタ横のボタンで、対応するレジスタの値をゼロクリアします。

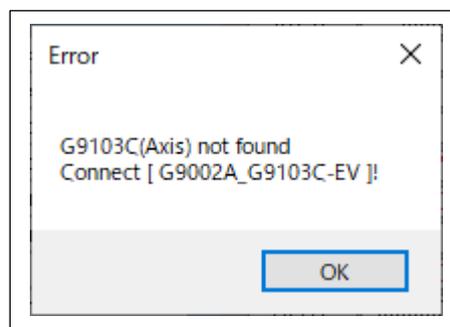
## 5.6 エラー発生時の表示

ソフトウェアが起動中に G9001A-EV の USB ケーブルが外れた場合、G9002A\_G9103C-EV の電源が OFF した場合、あるいは Motionnet ケーブルが外れた場合には次のエラー画面を表示します。

G9001A-EV の USB ケーブルが外れた



G9002A\_G9103C-EV の電源が OFF または Motionnet ケーブルが外れた



[OK]ボタンを押すことにより、ソフトウェアが終了します。

## 6. ソースコード説明

ソースファイルは「Form1.cs」です。お客様が試したい動作に修正することで、操作手順の確認を行ってみてください。

### 6.1 FTDI へのアクセス関数

FTDI への Open アクセスは「Form1\_Load」関数に記述されております。

接続されたボードが G9001A-EV であることの確認を行う関数

LSI_Open (string description)	
description	G9001A-EV の製品説明データが正しければアクセスを開始。

FTDI への Close アクセスは「Form1\_FormClosed」関数に記述されております。

G9001A-EV とのアクセスを中止する関数

LSI_Close()	
	G9001A-EV のアクセスを中止。

### 6.2 センターデバイス G9001A へのアクセス関数

センターデバイスへのアクセスは「Form1\_Load」関数に記述されております。

センターデバイス G9001A 内部をリセットする関数

MNETCenterComWriteSoftwareReset ()	
	G9001A をリセット(command = 0x0100)

センターデバイス G9001A から全デバイスへのシステム通信を開始する関数

MNETCenterComdWriteSystemComm()	
	全デバイスへのシステム通信(command = 0x1000)

センターデバイス G9001A から全デバイスへのサイクリック通信を開始する関数

MNETCenterComdWriteCyclicComm()	
	全デバイスへのサイクリック通信(command = 0x3000)

センターデバイス G9001A のステータス情報を取得する関数

MNETCenterReadSts(ref Status, ref IntStatus)	
Status	G9001A のステータス情報を取得
IntStatus	G9001A の割込みステータス情報を取得

ローカルデバイスサーチアクセスは「Device\_Search」関数に記述されております。

アドレスマップへのアクセスでデバイス情報を取得する関数

MNETCenterReadInf(dno, ref DevInf_D)	
dno	デバイス番号
DevInf_D	指定したデバイス情報データを取得し、保持するバッファ

コマンドアクセスでデバイス属性情報を読み出す関数

MNETCenterComdReadDevAttribute(dno)	
dno	デバイス番号(command = 0x1300 + dno)

コマンドアクセスでデバイス属性情報を取得する関数

MNETCenterComdGetDevAttribute(dno, ref Attr_D)	
dno	デバイス番号(command = 0x1300 + dno)
Attr_D	指定したデバイス属性情報データを取得し、保持するバッファ (データ受信用 FIFO よりデータの読出しを行う)

ローカルデバイスステータス読出しアクセスは「Axis\_StatusRead」関数に記述されています。

コマンドアクセスでポートデータを読出して取得する関数

MNETCenterComdReadPortD(dno, 0/1, ref portrd[x], ref portrd[y]);	
dno	デバイス番号(command = 0x6400 + dno × 2)
0/1	0: ポート 1,0 指定 1: ポート 3,2 指定
portrd[]	指定したデバイスのポートデータを取得し、保持するバッファ(入出力バッファよりデータの読み出しを行う)

コマンドアクセスでポートデータを書込む関数

MNETCenterComd WritePortD(dno, 0/1, portx, porty);	
dno	デバイス番号(command = 0x5400 + dno × 2)
0/1	0: ポート 1,0 指定 1: ポート 3,2 指定
port	指定したデバイスに書込むポートデータ(入出力バッファへデータの書き込みを行う)

## 6.3 ローカルデバイス G9103C への初期設定

初期設定は「Axis\_InitSet」関数に記述されており、ソフトウェアの起動時に呼び出されます。

初期設定として以下の操作が行われています。

データ送信用 FIFO に wrcom, data を書き込み、コマンドアクセスで指定したデバイス書込む関数

MNETLocalRegisterWrite (dno, wrcom, data);	
dno	デバイス番号 (command = 0x4000 + dno)
wrcom	レジスタ書込みコマンド
data	レジスタ書込みデータ

操作	内容
wrcom = 0x9C(RENV1) data = 0x00000002	環境設定 1 を次のように設定。 出力パルス仕様(PMD = "010")を設定。 OUT 端子から負論理でパルスを出力、DIR 端子からプラス方向時に Low を出力。
wrcom = 0x9D(RENV2) data = 0x000000FF	環境設定 2 を次のように設定。 汎用ポート 0~7(P0M-PM7 = "FF")を出力ポートに設定。
wrcom = 0x9F(RENV4) data = 0x81000000	環境設定 4 を次のように設定。 COUNTER1 ~ COUNTER3 のラッチタイミング(LTM = "01")を ORG 信号 ON に設定。 REST および RIST レジスタの読出し自動リセット(ISMR = "1")を解除。  <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">           本ソフトウェアは定期的に全てのレジスタをリードし、表示しています。            REST および RIST もリードしているため、読出し自動リセット機能が働く場合、エラー発生時であっても定期的なリードでフラグが消えてしまい、目視確認ができない場合があります。これを防止するため、RENV2.MRST=1 としています。         </div>
wrcom = 0x AC(RIRQ) data = 0x00000C01	イベント割り込み要因を次のように設定。 正常停止時(IREN = "1") に設定。 ORG 信号 ON 時(IROL = "1") に設定。 SD 信号 ON 時(IRSD = "1") に設定。
wrcom = 0x B1(PRFL) data = 0x00000001	FL 速度を次のように設定。 プリレジスタ(PRFL)を"0x01"に設定。[1 pps]
wrcom = 0x B2(PRFH) data = 0x000005DC	FH 速度を次のように設定。 プリレジスタ(PRFH)を"0x5DC"に設定。[1500 pps]
wrcom = 0x B3(PRUR) data = 0x00000D07	加速レートを次のように設定。 プリレジスタ(PRUR)を"0xD07"に設定。[1 s]
wrcom = 0x B5(PRMG) data = 0x000000C7	倍率を次のように設定。 倍率(PRMG = "0xC7")を 1 倍に設定。[199]
wrcom = 0x B7(PRMD) data = 0x00000041	動作モードを次のように設定。 位置決めモード(MOD = "41")。 直線加減速(MSMD = "0")。 スローダウンポイント自動設定(MSDP = "0")。

## 6.4 ローカルデバイス G9103C からレジスタ内容の読出し

レジスタ内容の読出しは「Axis\_RegisterRead」関数に記述されております。

データ送信用 FIFO に rrcom を書き込み、コマンドアクセスで指定したデバイス書込む関数

MNETLocalAllRegisterRead (dno, ref rrcom, ref readflag, min, max, allflag);	
dno	デバイス番号(command = 0x4000 + dno)
rrcom	レジスタ読出しコマンド群
readflag	レジスタ読出しが必要("1")/不要("0")のフラグ群
min	レジスタ読出しコマンド群の最小値を設定
Max	レジスタ読出しコマンド群の最大値を設定
allflag	全てのレジスタ読出し("1")

データ受信用 FIFO からレジスタデータを読出す関数

MNETLocalAllRegisterGet (ref rrcom, ref readflag, min, max, allflag, ref rbuffer);	
rrcom	レジスタ読出しコマンド群
readflag	レジスタ読出しが必要("1")/不要("0")のフラグ群
min	レジスタ読出しコマンド群の最小値を設定
Max	レジスタ読出しコマンド群の最大値を設定
allflag	全てのレジスタ読出し("1")
rbuffer	レジスタ読出しデータを格納するバッファ

読出したレジスタ内容の表示は「AllDataDisp」、「DispReg」、「DispLed」、「SetLed」、「reg\_MouseClick」関数に記述されております。

## 6.5 CCW 動作

CCW 動作を行っているのは、ソースファイル内の「btn\_ccw\_Click」関数です。

関数内では以下の操作が行われています。

データ送信用 FIFO に wrcom, data を書き込み、コマンドアクセスで指定したデバイス書込む関数

MNETLocalRegisterWrite (dno, wrcom, data);	
dno	デバイス番号 (command = 0x4000 + dno)
wrcom	レジスタ書込みコマンド
data	レジスタ書込みデータ

データ送信用 FIFO に RCom を書き込み、コマンドアクセスで指定したデバイス書込む関数

MNETLocalWrite_COM (dno, RCom)	
dno	デバイス番号(command = 0x4000 + dno)
RCom	動作コマンド

操作	内容
wrcom = 0x B0(PRMV)	移動量を次のように設定。
data = 0xFFFFEE00	移動量として "FFFFEE00h"(-4608) を設定。
Rcom = 0x53(STAUD)	高速スタート 2 を実行。

スタートコマンドの実行により、約 1 秒で 1pps から 1500pps まで加速し、しばらく 1500pps で動作した後、約 1 秒かけて減速停止します。

動作中に出力されるパルス数は、マイナス方向に 4608 パルスです。

## 6.6 CW 動作

CW 動作を行っているのは、ソースファイル内の「btn\_cw\_Click」関数です。  
関数内では以下の操作が行われています。

データ送信用 FIFO に wrcom, data を書き込み、コマンドアクセスで指定したデバイスを書込む関数

MNETLocalRegisterWrite (dno, wrcom, data);	
dno	デバイス番号 (command = 0x4000 + dno)
wrcom	レジスタ書き込みコマンド
data	レジスタ書き込みデータ

データ送信用 FIFO に RCom を書き込み、コマンドアクセスで指定したデバイスを書込む関数

MNETLocalWrite_COM (dno, RCom)	
dno	デバイス番号 (command = 0x4000 + dno)
RCom	動作コマンド

操作	内容
wrcom = 0x B0(PRMV)	移動量を次のように設定。
data = 0x00001200	移動量として "1200h"(+4608)を設定。
Rcom = 0x53(STAUD)	高速スタート 2 を実行。

スタートコマンドの実行により、約 1 秒で 1pps から 1500pps まで加速し、しばらく 1500pps で動作した後、約 1 秒かけて減速停止します。

動作中に出力されるパルス数は、プラス方向に 4608 パルスです。

## 6.7 STOP 動作

STOP 動作を行っているのは、ソースファイル内の「Btn\_stop\_Click」関数です。  
関数内では以下の操作が行われています。

データ送信用 FIFO に RCom を書き込み、コマンドアクセスで指定したデバイスを書込む関数

MNETLocalWrite_COM(dno, RCom)	
dno	デバイス番号(command = 0x4000 + dno)
RCom	動作コマンド

操作	内容
Rcom = 0x4A(SDSTP)	減速停止を実行。

## 6.8 CLEAR 動作

カウンター1 クリア動作を行っているのは、ソースファイル内の「Btn\_rcun1\_clr\_Click」関数です。  
関数内では以下の操作が行われています。

データ送信用 FIFO に RCom を書き込み、コマンドアクセスで指定したデバイス書き込む関数

MNETLocalWrite_COM(dno, RCom)	
dno	デバイス番号(command = 0x4000 + dno)
RCom	動作コマンド

操作	内容
Rcom = 0x20(SDSTP)	カウンター1 のクリアを実行。

REST のクリア動作を行っているのは、ソースファイル内の「Btn\_rest\_clr\_Click」関数です。

関数内では以下の操作が行われています。

データ送信用 FIFO に Rcom と WriteD を書き込み、コマンドアクセスで指定したデバイスを書込む関数

MNETLocalRegisterWrite(dno, Rcom, WriteD)	
dno	デバイス番号(command = 0x4000 + dno)
RCom	動作コマンド
WriteD	レジスタ書き込みデータ

操作	内容
Rcom = 0xAD(WREST)	エラー割込みステータスデータの書き込みコマンド。
WriteD = (読出しデータ)	リセットするビットを"1"にした任意のデータの書き込みでリセット。

RIST のクリア動作を行っているのは、ソースファイル内の「Btn\_rist\_clr\_Click」関数です。

関数内では以下の操作が行われています。

データ送信用 FIFO に RCom と WriteD を書き込み、コマンドアクセスで指定したデバイス書き込む関数

MNETLocalRegisterWrite(dno, Rcom, WriteD)	
dno	デバイス番号 (command = 0x4000 + dno)
RCom	動作コマンド
WriteD	レジスタ書き込みデータ

操作	内容
Rcom = 0xAE(WRIST)	イベント割込みステータスデータの書き込みコマンド。
WriteD = (読出しデータ)	リセットするビットを"1"にした任意のデータの書き込みでリセット。

本ソフトウェアは、RENV4 レジスタの bit31(ISMR)を "1" に設定してあります。この場合の REST や RIST のクリアは、クリアしたいビット部分に "1" を書き込むことで行われます。

REST や RIST からリードした値をそのまま書込むと、すべてをクリアします。

弊社は、弊社ソフトウェアについて著作権を含む一切の知的所有権を保持します。弊社は、弊社ソフトウェアに関するいかなる権利もお客様に譲渡しません。お客様は、弊社の製品を使用する目的でのみ、現状有姿の弊社ソフトウェアを使用することができます。弊社は、弊社ソフトウェアの完全性、正確性、適用性、有用性、第三者知財の非侵害性を含め、明示たると黙示たるとを問わず何らの保証をいたしません。また、弊社ソフトウェアを使用したことで生じる損害（収入または利益の逸失を含む）について、一切の責任を負いません。お客様が、購入国以外で弊社ソフトウェアを使用する場合は、購入国と使用国の輸出管理法や規制を遵守する必要があります。

## 改訂履歴

版数	日付	内容
初版	2020年3月11日	新規作成

**NPM** 顧客「満足」から「感動」へ。  
日本パルスモーター株式会社

[www.pulsemotor.com](http://www.pulsemotor.com)

お問い合わせ

[www.pulsemotor.com/support](http://www.pulsemotor.com/support)

東京 電話 03(3813)8841 FAX 03(3813)8550

大阪 電話 06(6576)8330 FAX 06(6576)8335

お電話受付時間 平日 9:00～17:00