

MKY36 Evaluation Board

EB-36

DATA SHEET

HLS Hi-speed
Link
System

MKY デバイス評価基板

EB-36 用サンプルソースコード説明

MKY36 の評価基板である EB-36 を動作させるためのサンプルソースについて説明します。

このサンプルコードは動作を保証するものではありません。PC のターミナルソフトと接続し、MKY36 のレジスタへのアクセスを可能にしたものです。実際の制御ソフトウェアについてはお客様の方でご準備いただき、お試しいただくようお願いいたします。

MKY36 のレジスタなどの説明については、「MKY36 ユーザーズマニュアル」に記載されています。まずはこれらの資料を熟読いただけますようお願いいたします。

目 次

1. 概要	1-1
2. NUCLEO 基板の準備.....	2-1
2.1. ハードウェアの準備.....	2-1
2.2. サンプルコードのダウンロード.....	2-1
2.2.1. 電源供給確認.....	2-1
2.2.2. USB ケーブルの接続	2-2
2.2.3. STM32CubeIDE の起動	2-2
2.2.4. Workspace の指定.....	2-2
2.2.5. プロジェクトの読み込み.....	2-4
2.2.6. ビルド.....	2-5
2.2.7. ダウンロード.....	2-6
2.2.8. 実行確認	2-7
3. 提供ソースコード構成.....	3-1
4. ターミナルソフト仕様.....	4-1
4.1. 通信仕様.....	4-1
4.2. コマンド仕様	4-1
4.2.1. HELP コマンド	4-2
4.2.2. MD コマンド:メモリー内容の表示.....	4-3
4.2.3. MM コマンド:連続するアドレスのメモリー内容の表示と変更	4-3
4.2.4. MW コマンド:連続するアドレスのメモリー内容を変更	4-4
4.2.5. REG コマンド:MKY36 の指定されたレジスタの状態の表示と変更.....	4-4
4.2.6. DIN コマンド:一つのアドレスのメモリー内容を表示.....	4-5
4.2.7. DOUT コマンド:一つのアドレスのメモリー内容を変更	4-5
4.2.8. CTRL コマンド:コントロールワードの状態の表示	4-6
4.2.9. DRC コマンド:DRC(Data Renewal Check)エリアの状態の表示	4-6
4.2.10. DO コマンド:Do エリアの状態の表示.....	4-7
4.2.11. DI コマンド:Di エリアの状態の表示.....	4-7

1. 概要

EB-36 は、株式会社ステップテクニカの HLS デバイス MKY36 を評価していただくために準備したものです。

MKY36 を制御するためには、ST Microelectronics 社が販売している NUCLEO 基板を購入いただき、ここでソフトウェア開発を行っていただく必要があります。

弊社では、この初期設定を簡略化できるようサンプルプログラムを提供させていただきます。

なお、サンプルプログラムとしては、NUCLEO-F446ZE を対象として作成したものをご提供させていただきます。

これ以外の NUCLEO 基板につきまして、サンプルコードを解析していただき、ポータリングをお願いします。

ご注意：NUCLEO 基板は、お使いの前に改造・設定変更が必要な場合があります。詳しくは、データシートをご覧ください、ご対応をお願いします。

ST Microelectronics 社の STM32 用統合開発環境は STM32CubeIDE となっております。

このソフトウェアにつきましては、ST Microelectronics 社のサイトから入手してください。

この資料では、Windows 版の開発環境での説明となります。

サンプルコード作成時のバージョンは 1.16.1 です。これ以外のバージョンをお使いになる場合は、それぞれのバージョンに合わせてポータリングを行ってください。

2. NUCLEO 基板の準備

接続できる NUCLEO 基板は、STM32 Nucleo-144 development board の分類に属するものです。Morpho Header と呼ばれる拡張コネクタを利用して接続する仕様となっています。

サンプルコードは、NUCLEO-F446ZE を前提に作成しております。

以降、特に断りのない限り NUCLEO-F446ZE を NUCLEO 基板とさせていただきます。

この章の作業は、NUCLEO 基板と EB-36 ユニットの接続しない状態で行うことが条件となっています。

お使いになる PC と NUCLEO 基板だけで作業を進めてください。

ご注意：EB-36 ユニットに取り付ける前に、NUCLEO 基板の IO 設定を行う必要があります。このため以下の手順でサンプルコードを先に書き込むようにしてください。

2.1. ハードウェアの準備

NUCLEO 基板を使用するまえに何点かの改造が必要な点があります。

「EB-36 データシート」をご参考に改造を実施してください。

2.2. サンプルコードのダウンロード

サンプルコードは、プロジェクト一式で供給されていますが、この中にビルド後のダウンロード可能なファイルが含まれています。

まず、ZIP 圧縮されたファイルを PC 上のフォルダに解凍してください。

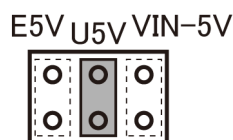
ここでは次のフォルダに解凍したということで以降の説明を行います。

C:\¥source¥EB36¥Project

2.2.1. 電源供給確認

NUCLEO 基板単体で動作させる場合、電源供給は接続する USB ケーブルからとなります。

NUCLEO 基板上の JP3 の設定が“U5V”になっているかを確認してください。



もしも、“E5V”や“VIN-5V”であった場合はショートバーを“U5V”に付け替えてください。

ご注意：ここでは NUCLEO 基板単体で動作させることを想定して説明しております。EB-36 に取り付けられた状態では JP3 の設定は異なります。

2.2.2. USB ケーブルの接続

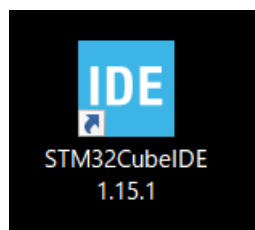
お使いになる PC と USB ケーブルを使い NUCLEO 基板と接続します。
NUCLEO 基板は CN1 の MicroB コネクタに接続します。

正しく接続されると、LD4(赤)と LD6(緑)の LED が点灯します。

この時 PC 側では“NOD_F446ZE”という名前のボリュームとしてエクスプローラーにマウントされると思います。

2.2.3. STM32CubeIDE の起動

ST Microelectronics 社の STM32 プロセッサ用の開発環境 STM32CubeIDE をインストールしている PC より、アプリケーションを起動します。インストール時にデスクトップにアイコンを作成していれば、このアイコンをクリックすれば起動できます。

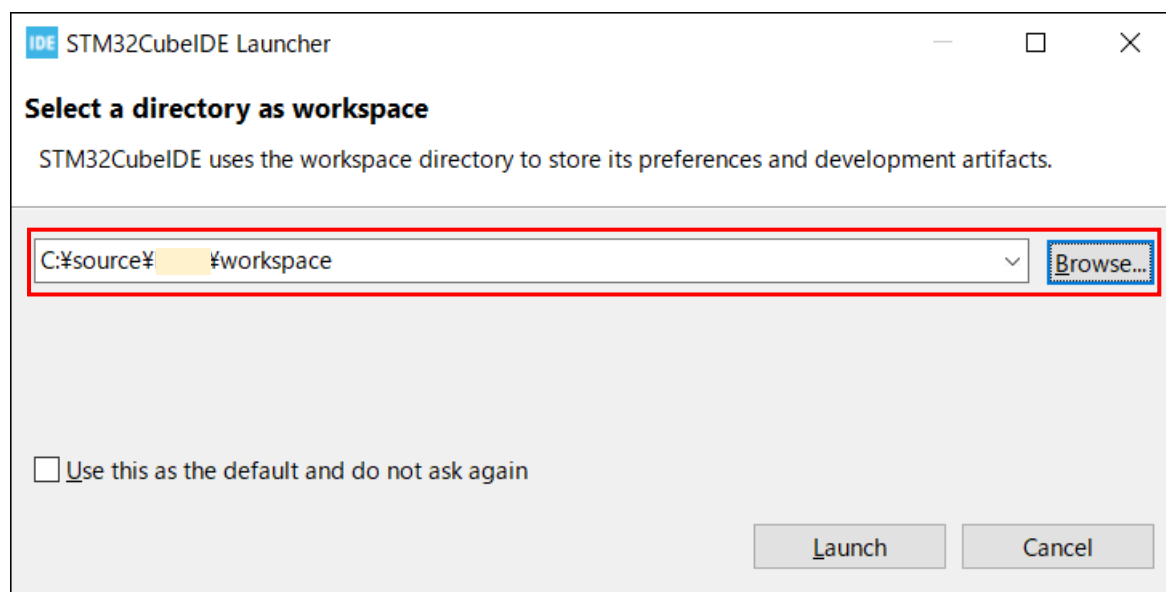


2.2.4. Workspace の指定

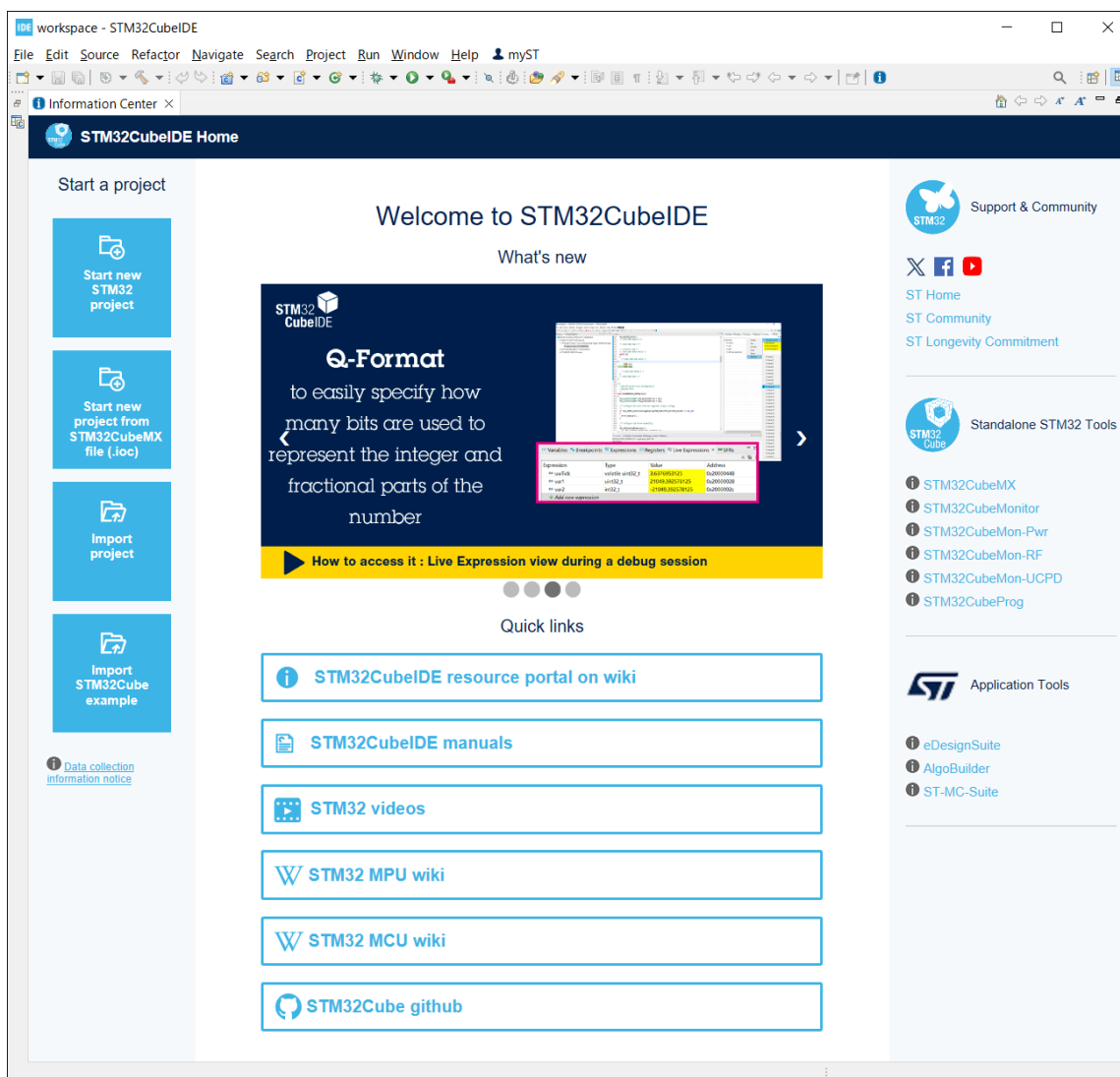
起動時に作業場所を選択するよう指示されます。初めて使われる場合は、[Browse...]ボタンを押し、作業環境を指定してください。2度目以降であれば選択項目に含まれているはずです。

ここではサンプルプロジェクトを解凍したフォルダと同じところに“workspace”フォルダを作成して、そこを指定することとします。

C:\source\EB36\workspace

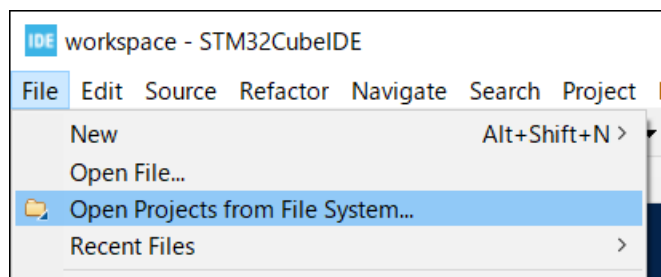


[Launch]ボタンを押すとアプリケーションが起動し、トップ画面が表示されます。

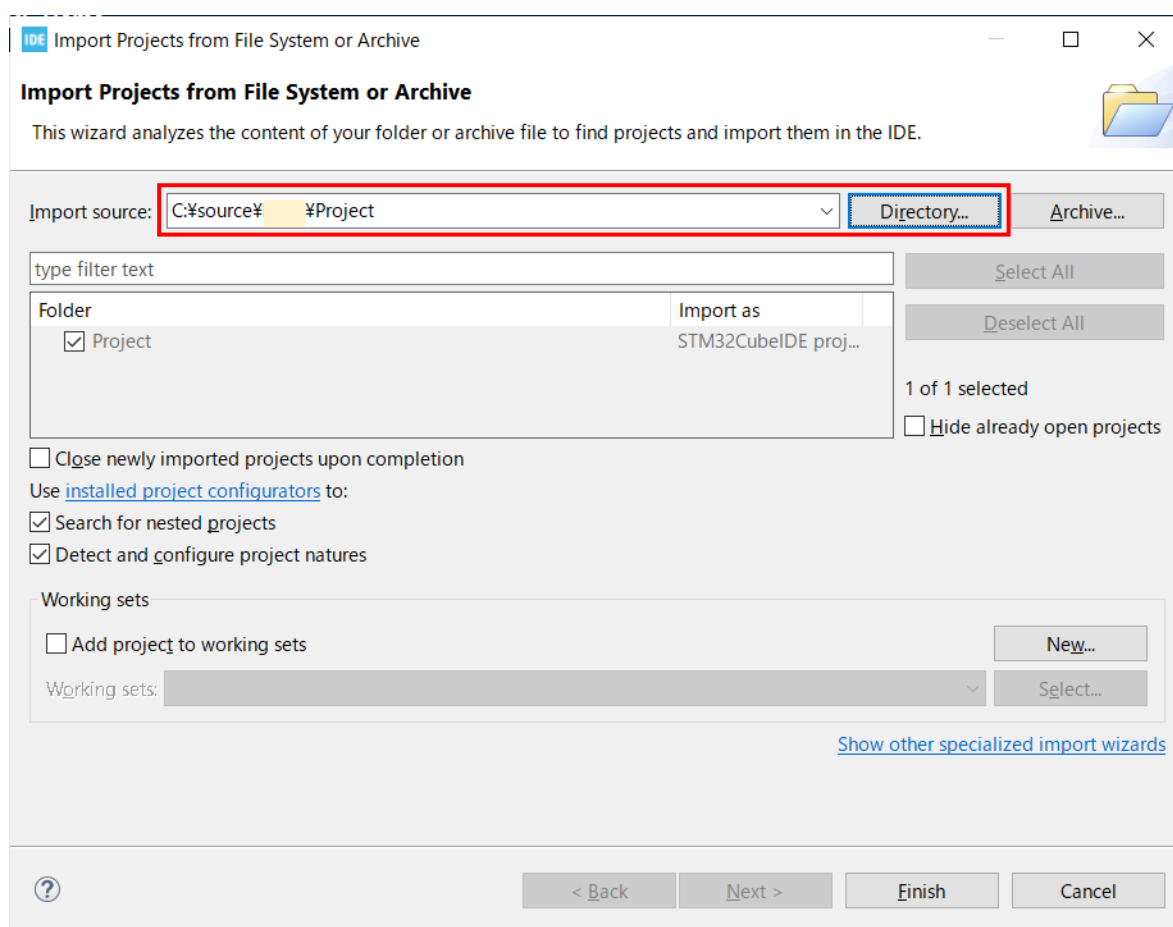


2.2.5. プロジェクトの読み込み

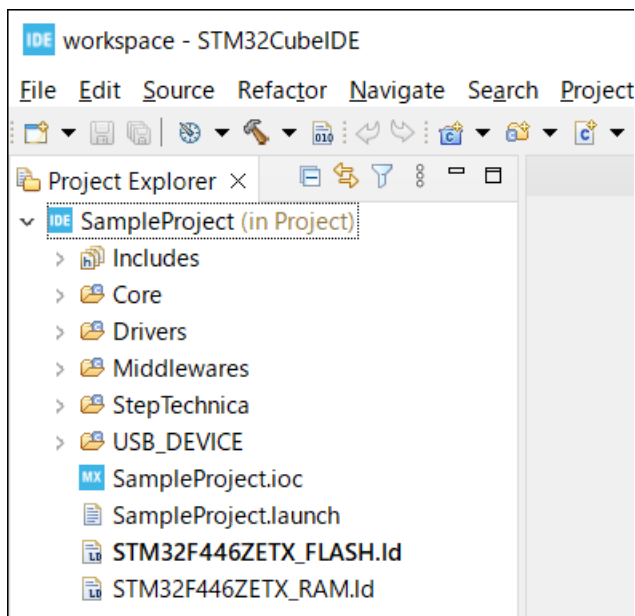
“File”メニューにある“Open Projects from File System...”を選びます。



次のダイアログが表示されますので、“Import source:”で解凍したフォルダを指定します。

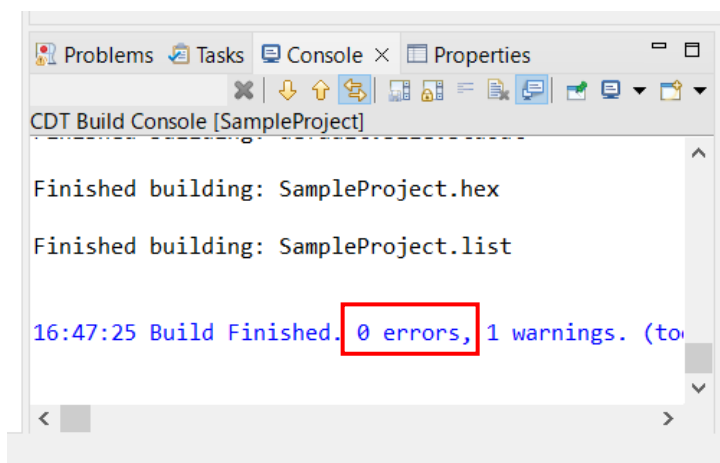
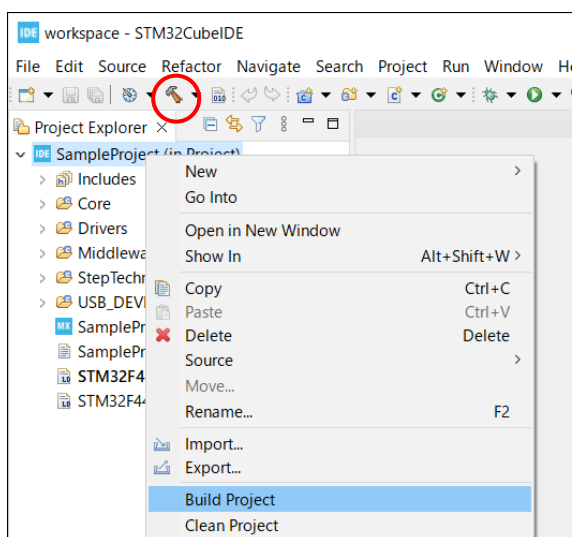


[Finish]ボタンを押すとプロジェクトが取り込まれます。

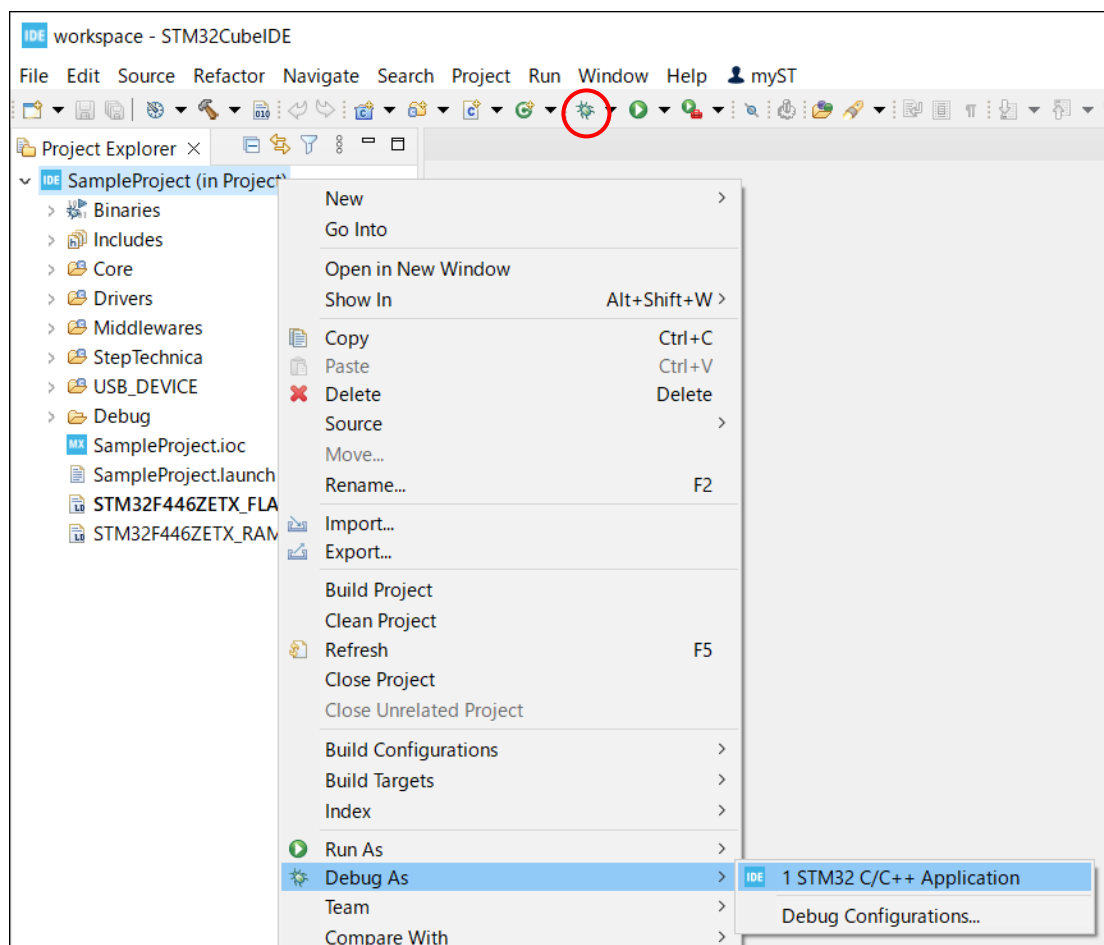


2.2.6. ビルド

サンプルプロジェクトはビルド済みですが、ここで一度ビルドしてエラーがないことを確認します。



2.2.7. ダウンロード

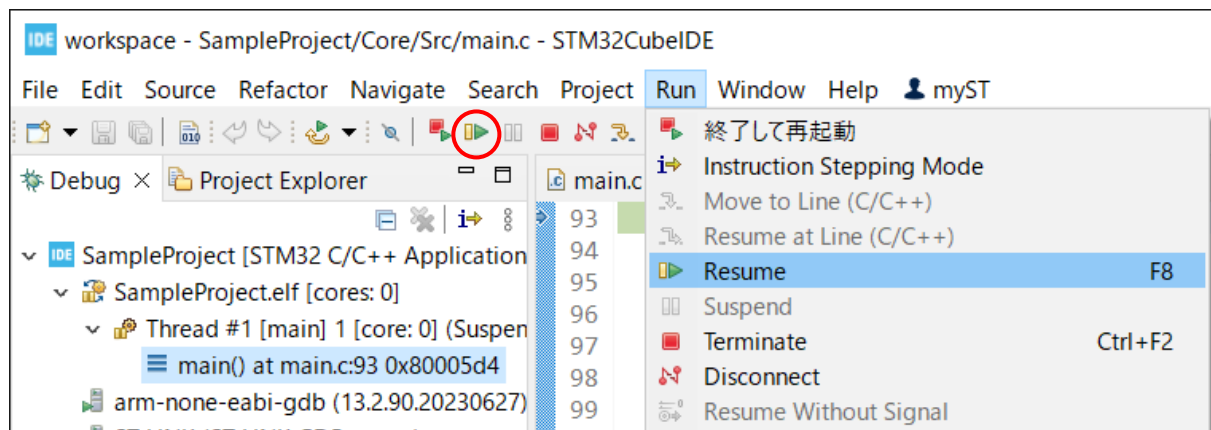


```

main.c x
93  HAL_Init();
94
95  /* USER CODE BEGIN Init */
96
97  /* USER CODE END Init */
98
99  /* Configure the system clock */
100 SystemClock_Config();
101
102 /* USER CODE BEGIN SysInit */
103
104 /* USER CODE END SysInit */
105
106 /* Initialize all configured peripherals */
107 MX_GPIO_Init();
108 MX_FMC_Init();
109 MX_SPI2_Init();
110 MX_SPI4_Init();
111 MX_USB_DEVICE_Init();
112 MX_UART4_Init();
113 MX_TIM1_Init();
114 MX_TIM4_Init();
115 /* USER CODE BEGIN 2 */
116     HAL_TIM_Base_Start_IT(&htim1);
117     HAL_TIM_Base_Start_IT(&htim4);

```

2.2.8. 実行確認



3. 提供ソースコード構成

提供するソースファイルは、プロジェクトフォルダに“StepTechnica”の名称でフォルダを作成し、そこにまとめております。

StepTechnica

└─ DebugMonitor

- | └─ dbTerm.c ターミナル処理プログラム
- | └─ dbTerm.h ターミナル処理ヘッダーファイル

└─ HLSDevice

- | └─ HLS.c HLS処理プログラム
- | └─ HLS.h HLS処理ヘッダーファイル
- | └─ MKYDevice.h MKY36アクセス関数用ヘッダーファイル

└─ IO

- | └─ charDef.h 文字コード定義ヘッダーファイル
- | └─ ctrlUART4.c UART4制御プログラム
- | └─ ctrlUART4.h UART4制御ヘッダーファイル
- | └─ IB36config.c IB36基板制御プログラム
- | └─ IB36config.h IB36基板制御ヘッダーファイル
- | └─ Utility.c 補助関数プログラム
- | └─ Utility.h 補助関数ヘッダーファイル

└─ MKY36

- | └─ dbTerm36.c ターミナル処理固有プログラム
- | └─ MKY36.c MKY36アクセス関数プログラム
- | └─ MKY36.h MKY36アクセス関数ヘッダーファイル

└─ VCOM

- | └─ ctrlVCOM.c USB(仮想COMポート)制御プログラム
- | └─ ctrlVCOM.h USB(仮想COMポート)制御ヘッダーファイル
- | └─ VPortTerm.c USB(仮想COMポート)ターミナル処理プログラム
- | └─ VPortTerm.h USB(仮想COMポート)ターミナル処理ヘッダーファイル

これ以外に自動生成されるソースファイルも一部(main.c や割込み処理ソースなど)編集しておりますが、その内容につきましてはそれぞれのソースファイルをご確認ください。

4. ターミナルソフト仕様

サンプルプログラムには UART4 を使ったターミナルコマンドが組み込まれています。

CN22 に信号が割り当てられていますが、電気的には TTL レベルとなっているため、市販の USB 変換基板などを使って PC と接続していただくと、このターミナルソフトを使用して、MKY36 を制御することが可能となります。

4.1. 通信仕様

シリアル通信の仕様としては次表のようになっています。

お使いになるターミナルソフトではこれに合わせて環境設定を行ってください。

項目	値
スピード	115200
データ長	8bit
パリティ	None
ストップビット	1bit
フロー制御	none
改行コード	[CR]+[LF]

4.2. コマンド仕様

コマンドとして用意しているものは次表の通りです。

Command	機能
HELP or ?	ヘルプメッセージ表示
MD	メモリーダンプ
MM	連続するアドレスのメモリー内容の表示と変更
MW	連続するアドレスのメモリー内容を変更
REG	MKY36 の指定されたレジスタの内容表示と変更
DIN	一つのアドレスのメモリー内容を表示
DOUT	一つのアドレスのメモリー内容を変更
CTRL	コントロールワードの状態の表示
DRC	DRC(Data Renewal Check)エリアの状態の表示
DO	Do エリアの状態の表示
DI	Di エリアの状態の表示

ソースコードとしては、StepTechnica¥MKY36¥dbTerm36.c の CommonCmdTbl[] にエントリーポイントを定義しています。

4.2.1. HELP コマンド

```
Usage :  
help or ? : show this messages  
md  [Format] [address] <[byte length]> : memory dump  
mm  [Format] [address]                  : memory mode  
mw  [Format] [address]                  : memory mode write  
reg <[Name]>                            : register modify  
din  [Format] [address]                 : direct read  
dout [Format] [address] [data]          : direct write  
ctrl <[Format]>                         : control area dump  
drc  <[Format]>                         : DRC area dump  
do   <[Format]>                         : DO area dump  
di   <[Format]>                         : DI area dump  
  
[Format] :  
    -b = byte      : 1byte  
    -w = word      : 2byte  
    -l = long      : 4byte  
    -d = long long : 8byte
```

4.2.2. MD コマンド:メモリー内容の表示

書式

md [Format] [address] <[byte length]>

説明

指定されたアドレスから [Format] で指定された書式でメモリー内容を表示します。

パラメーター

[Format]

表示する書式を指定します。

-b byte = 1byte 単位の 16 進数表示 (ASCII 表示付き)

-w word = 2byte 単位の 16 進数表示

-l long = 4byte 単位の 16 進数表示

-d long long = 8byte 単位の 16 進数表示

[address]

MKY36 エリアのオフセットアドレス値を 16 進数で指定します。

入力範囲:0x0000 ~ 0x07FE

<[byte length]> (“< >” は省略可能を意味する)

表示するバイト数を指定します。

省略した場合は 0x0100 と解釈します。

入力範囲:0x0000 ~ 0x07FF

4.2.3. MM コマンド:連続するアドレスのメモリー内容の表示と変更

書式

mm [Format] [address]

説明

指定されたアドレスのデータ表示と変更を行います。

コマンドが入力されると指定アドレスの内容を表示し、書き込みデータの入力待ちになります。

データ変更を行わず「リターン」キーを押すと次のアドレスに進みます。

‘-’ (ハイフン) キーを押すと前のアドレスに戻ります。

‘.’ (ピリオド) キーの入力で mm コマンド処理を終了します。

パラメーター

[Format]

アクセスするデータ幅を指定します。

-b byte = 1byte 単位

-w word = 2byte 単位

-l long = 4byte 単位

-d long long = 8byte 単位

[address]

MKY36 エリアのオフセットアドレス値を 16 進数で指定します。

入力範囲:0x0000 ~ 0x07FE

4.2.4. MW コマンド:連続するアドレスのメモリー内容を変更

書式

mw [Format] [address]

説明

指定アドレスから順にデータ書き込みを行います。
コマンド入力後、指定アドレスへの書き込みデータの入力待ちをします。(メモリー内容の表示は行いません。)
データ変更を行わず「リターン」キーを押すと次のアドレスに進みます。
‘-’ (ハイフン)キーを押すと前のアドレスに戻ります。
‘.’ (ピリオド)キーの入力で mm コマンド処理を終了します。

パラメーター

[Format]

アクセスするデータ幅を指定します。

-b byte = 1byte 単位
-w word = 2byte 単位
-l long = 4byte 単位
-d long long = 8byte 単位

[address]

MKY36 エリアのオフセットアドレス値を 16 進数で指定します。
入力範囲:0x0000 ~ 0x07FE

4.2.5. REG コマンド:MKY36 の指定されたレジスタの状態の表示と変更

書式

reg <[name]>

説明

引数で指定された MKY36 内部レジスタ内容の表示と変更を行います。
注意点としてすべてのレジスタが変更可能ではありません。
次のレジスタは、内容の表示だけとなります。
SSR, CCR
その他のレジスタについては、内容の表示と変更が可能です。
内部レジスタの詳細機能については「MKY36 ユーザーズマニュアル」を参照ください。
reg コマンドに続けてレジスタ名を入力することにより、指定されたレジスタ内容を表示しユーザーからの書き込みデータの入力待ちをします。
データ変更を行わない場合は「リターン」キーを押して下さい。

パラメーター

<[name]>

内容の表示・変更を行いたいレジスタ名を指定します。

SCR SSR HPR
INT0R INT1R C1CR C2CR
BCR CCR

省略した場合はパラメーター指定できるレジスタ名の一覧を表示します。

4.2.6. DIN コマンド:一つのアドレスのメモリー内容を表示

書式

din [Format] [address]

説明

指定されたアドレスのデータを指定された書式で表示します。

パラメーター

[Format]

アクセスするデータ幅を指定します。

-b byte = 1byte 単位の 16 進数表示

-w word = 2byte 単位の 16 進数表示

-l long = 4byte 単位の 16 進数表示

-d long long = 8byte 単位の 16 進数表示

[address]

MKY36 エリアのオフセットアドレス値を 16 進数で指定します。

入力範囲:0x0000 ~ 0x07FE

4.2.7. DOUT コマンド:一つのアドレスのメモリー内容を変更

書式

dout [Format] [address] [data]

説明

指定されたアドレスに指定された初期の幅でデータを書き込みます。

パラメーター

[Format]

アクセスするデータ幅を指定します。

-b byte = 1byte 単位の 16 進数指定

-w word = 2byte 単位の 16 進数指定

-l long = 4byte 単位の 16 進数指定

-d long long = 8byte 単位の 16 進数指定

[address]

MKY36 エリアのオフセットアドレス値を 16 進数で指定します。

入力範囲:0x0000 ~ 0x07FE

[data]

書き込みデータを 16 進数で指定します。

4.2.8. CTRL コマンド:コントロールワードの状態の表示

書式

ctrl <[Format]>

説明

コントロールワードの状態を表示します。

全コントロールワードエリア(128 バイト)を [Format]で指定された書式で表示します。

パラメーター

<[Format]>

表示する書式を指定します。

省略した場合は“-b”と解釈します。

-b byte = 1byte 単位の 16 進数表示 (ASCII 表示付き)

-w word = 2byte 単位の 16 進数表示

-l long = 4byte 単位の 16 進数表示

-d long long = 8byte 単位の 16 進数表示

4.2.9. DRC コマンド:DRC(Data Renewal Check)エリアの状態の表示

書式

drc <[Format]>

説明

コントロールワードの状態を表示します。

全コントロールワードエリア(128 バイト)を [Format]で指定された書式で表示します。

パラメーター

<[Format]>

表示する書式を指定します。

省略した場合は“-b”と解釈します。

-b byte = 1byte 単位の 16 進数表示 (ASCII 表示付き)

-w word = 2byte 単位の 16 進数表示

-l long = 4byte 単位の 16 進数表示

-d long long = 8byte 単位の 16 進数表示

4.2.10. DO コマンド:Do エリアの状態の表示

書式

do <[Format]>

説明

コントロールワードの状態を表示します。

全コントロールワードエリア(128 バイト)を [Format]で指定された書式で表示します。

パラメーター

<[Format]>

表示する書式を指定します。

省略した場合は“-b”と解釈します。

-b byte = 1byte 単位の 16 進数表示 (ASCII 表示付き)

-w word = 2byte 単位の 16 進数表示

-l long = 4byte 単位の 16 進数表示

-d long long = 8byte 単位の 16 進数表示

4.2.11. DI コマンド:Di エリアの状態の表示

書式

di <[Format]>

説明

コントロールワードの状態を表示します。

全コントロールワードエリア(128 バイト)を [Format]で指定された書式で表示します。

パラメーター

<[Format]>

表示する書式を指定します。

省略した場合は“-b”と解釈します。

-b byte = 1byte 単位の 16 進数表示 (ASCII 表示付き)

-w word = 2byte 単位の 16 進数表示

-l long = 4byte 単位の 16 進数表示

-d long long = 8byte 単位の 16 進数表示

➤ 改訂履歴

Version	発行日	改訂内容
100	2025/3/25	初版発行

ご注意

- 1) 本資料に記載された内容は、将来予告なしに変更する場合があります。本製品をご使用になる際には、本資料が最新の版であるかをご確認ください。
- 2) 本資料において記載されている説明や回路例などの技術情報は、お客様が用途に応じて本製品を適切にご利用いただくための参考資料です。
- 3) 実際に本製品をご使用になる際には、基板上における本製品の周辺回路条件や環境を考慮の上、お客様の責任においてシステム全体を十分に評価し、お客様の目的に適合するようシステムを設計してください。当社は、お客様のシステムと本製品との適合可否に対する責任を負いません。
- 4) 本資料に記載された情報、製品および回路等の使用に起因する損害または特許権その他権利の侵害に関して、当社は一切その責任を負いません。
- 5) 本製品および本資料の情報や回路などをご使用になる際、当社は第三者の工業所有権、知的所有権およびその他権利に対する保証または実施権を許諾致しません。
- 6) 本製品は、人命に関わる装置用としては開発されておりません。人命に関わる用途への採用をご検討の際は、当社までご相談ください。
- 7) 本資料の一部または全部を、当社に無断で転載および複製することを禁じます。

➤ お問い合わせ先

株式会社ステップテクニカ
〒207-0021 東京都東大和市立野1丁目1-15
TEL 042-569-8577 / E-Mail: info@steptechnica.com