

mruby/c 評価ボード
利用説明書

2018/8/20 rev 1.2
しまねソフト研究開発センター

ピンアサイン

ポート名	タイプ	デバイス割り当て	(参考) CY8CKIT-059 上のデバイス PSoC5 内蔵デバイス
P0-0	D-in	SW1	
P0-1	D-in	SW2	
P0-2	D-in	SW3 UP	C12 1u
P0-3	D-in	SW4 DOWN	C13 1u
P0-4	D-in	SW5 LEFT	C9 1u
P0-5	D-in	SW6 RIGHT	
P0-6			
P0-7			
P1-0		(N/C)	PROG_SWDIO
P1-1		(N/C)	PROG_SWDCLK
P1-2			
P1-3		(N/C)	P_SWO
P1-4			P_TDI JTAG[0] tdi
P1-5			JTAG[0] ntrst
P1-6			
P1-7			
P2-0			
P2-1	D-out	-----	LED1
P2-2	D-in	-----	SW1
P2-3			
P2-4			
P2-5			
P2-6			
P2-7			
P3-0	A-in	U1 ThermoSensor TI LM60BIZ	
P3-1			
P3-2			C7 1u
P3-3			
P3-4			
P3-5			
P3-6		(SP1+)	
P3-7		(SP1-)	
P12-0		I2C SCL	I2C[1] scl
P12-1		I2C SDA	I2C[1] sda
P12-2			SIO
P12-3			SIO
P12-4		UART RxD	I2C[0] scl
P12-5		UART TxD	I2C[0] sda
P12-6	D-in	-----	SIO RX (プログラマ経由ホストシリアル)
P12-7	D-out	-----	SIO TX (プログラマ経由ホストシリアル)
P15-0	D-out	LCD DB4 (SC2004CS)	
P15-1	D-out	LCD DB5	
P15-2	D-out	LCD DB6	C42 22p
P15-3	D-out	LCD DB7	C41 22p
P15-4	D-out	LCD E	C4 2200p
P15-5	D-out	LCD RS	
P15-6	USB	-----	USBIO dp (オンボード USB マイクロコネクタ用)
P15-7	USB	-----	USBIO dm (オンボード USB マイクロコネクタ用)

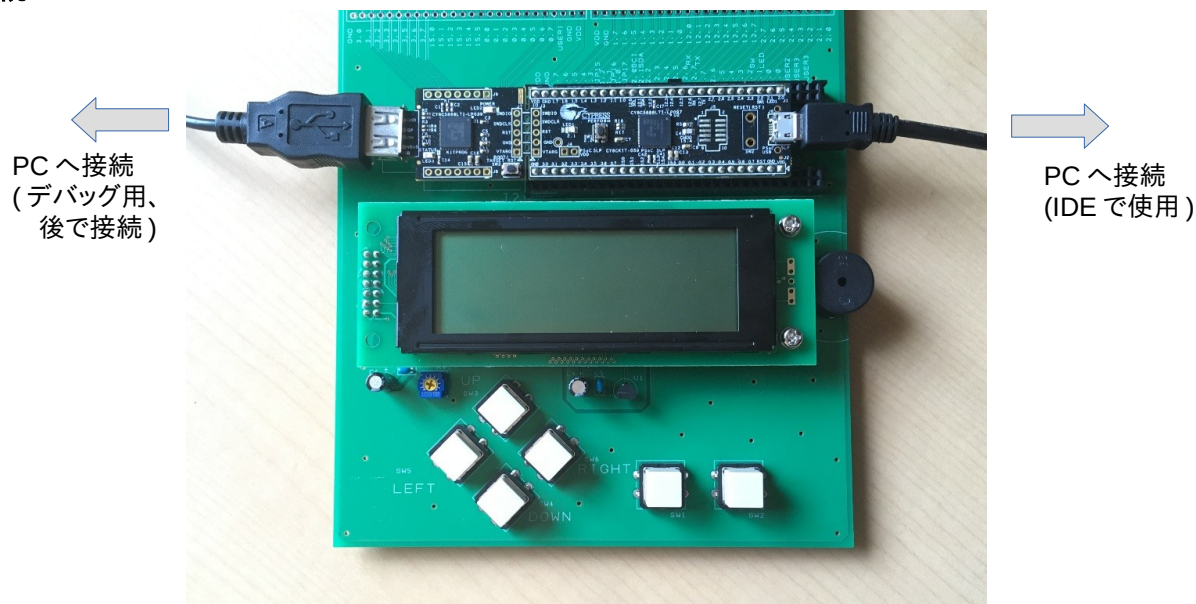
mruby/c IDE の利用

提供された機能のみを使ってアプリケーションを構築する方法です。

統合開発環境、IDE を利用します。

あらかじめファームが書き込まれた本体を使用します。ファームを壊してしまった等の場合は、後述する方法でファームを書き込みます。

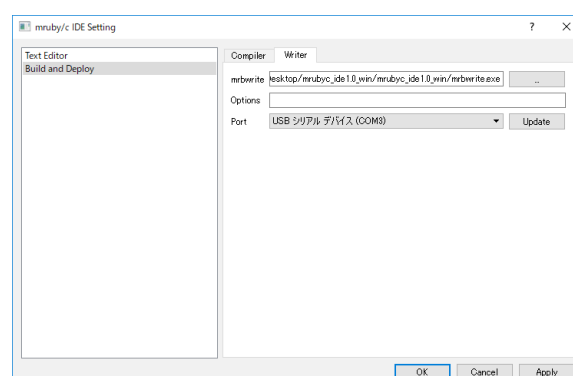
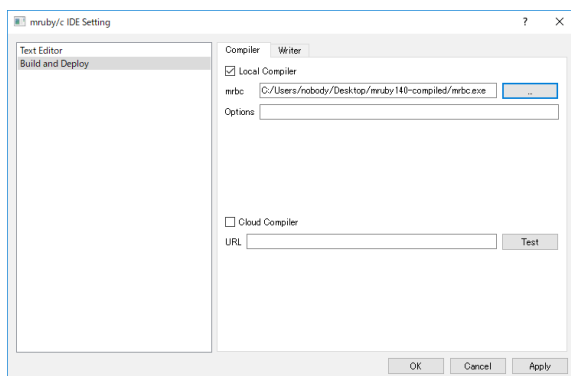
接続



IDE 起動と設定

Windows の例で説明しますが、Mac や UNIX も同等です。

1. mrbyc-ide.exe を実行します。
2. File > Settings を選び、設定ダイアログを表示します。
3. 左ペインの Build and Deploy をクリックします。
4. 右ペインの Compiler タブで、Local Compiler にチェックが入っていることを確認します
5. mrbc 欄の[...]ボタンをクリックし、mrbc.exe のパスを指定します。
6. 右ペインの Writer タブに移り、同様に mrbwrite.exe のパスを指定します。
7. Port 欄に、基板右側のマイクロ USB 端子側の認識ポートを指定します。

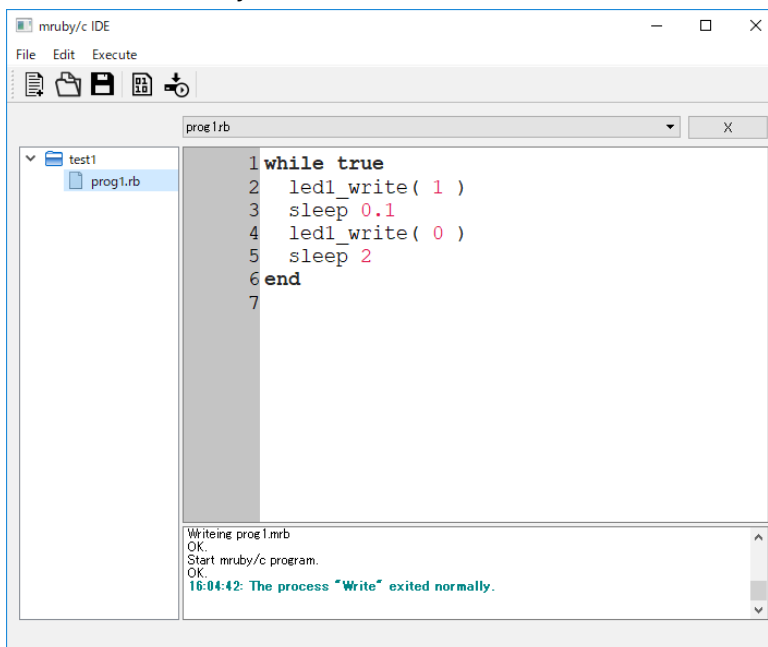


デバッグ用コンソールの起動(任意)

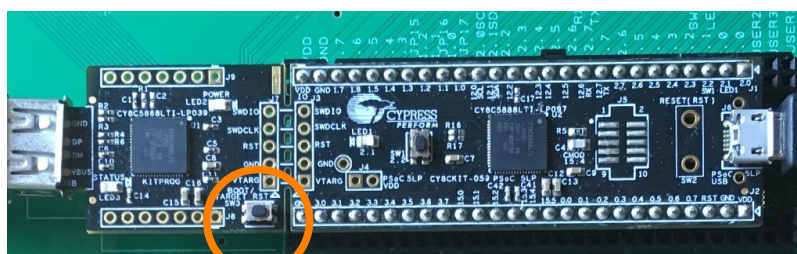
基板左側の USB 端子から、デバッグ用メッセージが表示されますので、開発時は接続しておくくと便利です。任意のターミナルソフトを利用して、ボーレート 57600bps で接続します。

プログラムの編集と書き込み

1. File > New > Project を選び、Setup New Project ダイアログを表示します。
2. 任意のプロジェクト名と、保存場所を指定します。
3. File > New > File を選び、Setup Add New File ダイアログを表示します。
4. 任意のファイル名を指定します。
5. 右上のプログラムペインで、Ruby プログラムを編集します。



6. 編集が終わったら、Execute > Write を選びます。
7. プログラムの書き込み待ち状態になるので、基板上的のリセットスイッチを押下します。
8. プログラムが書き込まれ、動作開始します。



リセットスイッチ

利用できる機能 mruby/c IDE 編

オンボードスイッチ

マイコンドータ基板上のプッシュスイッチです。SW1 のシルクがあります。
負論理で、0(押された)、1(押されていない)が、返ります。

```
sw1 = sw1_read()
```

LED

マイコンドータ基板上の青色 LED です。LED1 のシルクがあります。
正論理で、1(点灯)、0(消灯)です。

```
led1_write(1)
```

キャラクタ LCD 表示器

4 行×20 文字の LCD パネルです。
アルファベットと数字、記号などが表示できます。詳しくは、HD44780 のデータシート等を参照してください。

```
lcd_location( row, column )  
lcd_putc( char_code )  
lcd_puts( "String" )  
lcd_clear()
```

キーパッド

メイン基板上の 6 個のタクトスイッチです。
ビットマップで全てのスイッチの押下状態を同時に確認できます。
負論理で、ビットは基板上シルクの SWx を参照してください。

```
key = keypad_read()
```

温度センサー (A/D 変換)

メイン基板上に、温度センサー LM60 が搭載されており、温度の測定ができます。
LM60 は、アナログ値を出力しますので、A/D 変換を行います。

A/D コンバータ仕様

- 12 ビット
- 0V - 2.048V
- 100000sps

```
adc_start_convert()          # A/D 変換開始  
adc_wait_end_conversion()    # A/D 変換終了待ち  
val = adc_get_result16()     # 値読み込み(温度変換前、整数値)
```

PWM

メイン基板上に、圧電スピーカーが搭載されており、PWM を使って音を出すことができます。

PWM 仕様

- カウンタ 16bit
- クロック 120kHz

```
pwm_write_period( n )    # n = 0~65535
pwm_write_compare( n )
```

I2C

I2C インターフェースを扱うことができます。

参照：<https://github.com/mruby/c/dev/tree/master/PSoC5LP>

I2C 仕様

- 5V
- 100kbps
- マスタモードのみ
- ピンアサイン 12.0 SCL 12.1 SDA

```
i2c = I2C.new()

# write
i2c.write( i2c_address, device_register, data1, data2, ... )
i2c.write( i2c_address, device_register, "String" )

# read
s = i2c.read( i2c_address, device_register, length )
```

UART

シリアルインターフェースを扱うことができます。

PSoC5LP 開発環境の仕様により、ボーレートの変更がソフトウェアでできません。標準で、19200bps 固定です。

変更するためには、サイプレス社の開発環境 PSoC Creator を使って、画面上で書き換える必要があります。

現在の仕様では、読み込み (get, read) 時にデータが揃っていない場合、ブロックせずに nil を返します。

UART 仕様

- 5V
- 19200bps
- ピンアサイン 12.5 TxD 12.4 RxD

```
uart = UART.new()

# String read / write
s = uart.gets()
uart.puts("STRING\r\n")

# Binary read / write
s = uart.read(n) # read n bytes.
uart.write("STRING\r\n")

# Flush buffer
uart.clear_tx_buffer()
uart.clear_rx_buffer()
```

Mutex

複数プログラムの同期用に、Mutex を使う事ができます。

```
$mutex1 = Mutex.new
$mutex1.lock()
$mutex1.unlock()
$mutex1.try_lock()
```

その他のメソッド

mruby/c ランタイムスケジューラには、以下のメソッドを用意しています。

```
sleep(n) # 実行一時停止。秒単位。1 以下(0.1 等)も可能。
sleep_ms(n) # 実行一時停止。ミリ秒単位。
relinquish() # 他のタスクに実行を譲る。
```

ファームの書き込み 改造

ファームウェアがなんらかの原因で壊れてしまったとき、または、UART 数やボーレート変更など、改造が必要となったときに、ファームウェアの書き込みが必要になります。

手順

1. サイプレスのホームページから、PSoC Creator をダウンロードし、インストールします。
2. ITOC のホームページから、ファームウェアをダウンロードし、展開します。
3. ファームウェアを PSoC Creator で開きます。
4. 必要に応じて、回路図等を変更します。
5. メニューから、Build > Build Design 01 を選び、コンパイルを行います。
6. ターゲットボードの左側の USB (カードエッジ側) を PC と接続します。
7. メニューから Debug > Program を選び、書き込みます。