

RBoard 仕様書

～～

小倉氏

目次

ボード仕様.....	3
概要	3
入出力仕様.....	3
電氣的仕様.....	4
プログラム書き込み	5
スモウルビー	5
mruby/c IDE.....	5
初期設定	5
書き込み	5
ハードプログラム更新.....	7
プログラム例	8
関数一覧	9
Digital(class)	9
new(initial)	9
write(pin, status)	9
read(pin)	9
interrupt(pin)	10
ADC(class)	10
new(pin).....	10
start.....	10
read.....	10
stop	11
スリープ.....	11
sleep(sec)	11
sleep_ms(m_sec)	11
hibernate	12
sleep_set_interval(sec)	12
sleep_until_nexttime	12
I2C(class).....	12
new	13
read(addr, data_addr, number)	13
write(addr, data_addr, data1...)	13
status	13

clear_status.....	14
UART(class)	14
new(pin).....	14
read(num)	14
write(txt)	15
gets	15
puts(txt)	15
clear_tx_buffer	15
clear_rx_buffer	16
その他	16
puts(String)	16
leds_write(num)	16
SW()	16

ボード仕様

概要

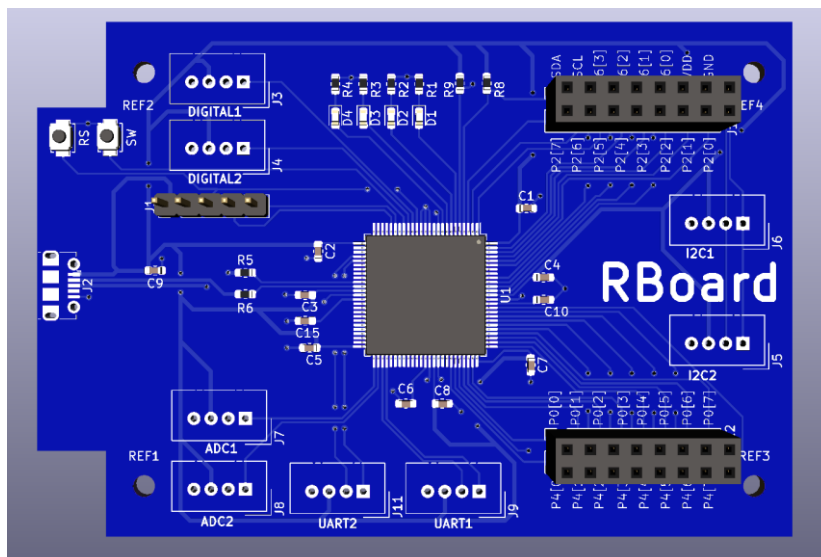
GROVE ポート搭載により多種多様なセンサを搭載可能になるため、回路設計の手間が無くなり初心者にもやさしい。

Mruby/c と呼ばれるマイコン向け Ruby 言語によるプログラミングが可能であるため、プログラミング初心者でも容易にプログラムが可能である。

疑似的なマルチタスク処理対応。

入出力仕様

外観図：



インターフェース名	数	対応クラス等	備考
Digital	2	Digital	デジタル入出力
ADC	2	ADC	アナログ入力
I2C	2	I2C	
UART	2	UART	
SWD	1		PG 書き込み用

USB	1	puts	mruby/c 書き込み用 電源供給
SW	1	SW	オンボードスイッチ
LED	4	leds_write	オンボード LED
P0	8		
P2	8		
P4	8		
P6	4		
SCL,SDA	1	I2C	

電氣的仕様

入力電圧	3.3~5V(推奨 5V)
ポート電圧	3.3V
ハイバネート電流	0.2uA
動作電流	10mA

プログラム書き込み

スモウルビー

対応予定

mruby/c IDE

初期設定

1. [公式サイト](#)よりダウンロード
2. mrubyc-ide.exe を実行。
3. File > Settings を選択、設定ダイアログを表示。
4. Build and Deploy をクリック。
5. Compiler タブで、LocalCompiler にチェックが入っていることを確認
6. mrbc 欄の[...]ボタンをクリックし、mrbc.exe のパスを指定。
7. Writer タブに移り、mrbcwrite.exe のパスを指定。
8. Port 欄に、基板の COM ポートを指定。

書き込み

1. File > New > Project を選択、Setup New Project ダイアログを表示。
2. プロジェクト名と、保存場所を指定。
3. File > New > File を選択、Setup Add New File ダイアログを表示。
4. ファイル名を指定。(例：main.rb)
5. 作成した Ruby プログラムを編集。
6. 編集後、Execute > Write を選択。
7. プログラムの書き込み待ち状態になるので、基板上のリセットスイッチを押下。

待ち状態

```
10:45:17: Starting: "Local Compile"  
C:/Users/164/Downloads/201809041043236698b4b6ce0/mrbc.exe main.rb  
10:45:17: The process "Local Compile" exited normally.  
10:45:17: Starting: "Write"  
C:/Users/164/Downloads/mruby_ide1.02_win/mruby_ide1.01_win/mrbwrite.exe -I COM16 main.mrb  
Start connection.  
....
```

8. プログラムが書き込まれ、動作開始します。

ハードプログラム更新

1. [PSoC Creator](#) をインストール
2. [GIT](#) より最新のハードプログラムをダウンロード
3. ダウンロードファイル解凍後、「mcmc.cywrk」を実行
4. ターゲットボードの SWD ポートへプログラマを接続
5. PSoC Creator -> Debug -> Program を実行
6. LED1 が点滅したら書き込み成功

プログラム例

hello_world.rb

#5 秒間隔で PC へ Hello World を送信

while true

puts "Hello World"

sleep 5

end

l_chika.rb

#LED 1 を 1 秒間隔で点滅させる

While true

leds_write(1)

sleep 1

leds_write(0)

sleep 1

end

関数一覧

Digital(class)

デジタル入出力用の GROVE ポートを制御するクラス

new(pin)

説明：

クラス生成

パラメータ：

pin : 0 か 1

戻り値：

なし

write(status)

説明：

指定したデジタルの値を対象ピンに書き込む。

パラメータ：

status : 0(OFF) か 1(ON)

戻り値：

なし

read()

説明：

デジタル値を対応ピンから読み込む。

パラメータ：

なし

戻り値：

0(OFF)か 1(ON)

interrupt(pin)

説明：

デジタル入力の割り込みを許可

パラメータ：

pin：ピン番号(1)

※2は不可

戻り値：

なし

ADC(class)

アナログデジタル変換用の GROVE ポートを制御するクラス

new(pin)

説明：

クラス生成

パラメータ：

pin：使用するポート(0,1)

戻り値：

なし

start

説明：

ADC を実行開始状態にする。

パラメータ：

なし

戻り値；

なし

read

説明：

センサーからアナログ値を取得する。

パラメータ：

なし

戻り値：

整数（入力電圧を 12bit 化した値【0~4096】）

stop

説明：

ADC を一時停止にする。

停止後に start メソッドを実行すると再度起動する。

パラメータ：

なし

戻り値：

なし

スリープ

sleep(sec)

説明：

待ち時間(秒)

パラメータ：

Sec：整数

戻り値：

なし

sleep_ms(m_sec)

説明：

待ち時間(ミリ秒)

パラメータ：

m_sec：整数

戻り値：

なし

hibernate

説明：

ハイバネートモードへ移行

Digital 等の外部割り込みにより復帰する

通常 sleep より省電力

パラメータ：

なし

戻り値：

なし

sleep_set_interval(sec)

説明：

待ち時間設定(秒)

sleep_until_nexttime とセット

パラメータ：

Sec：整数

戻り値：

なし

sleep_until_nexttime

説明：

sleep_set_interval にて設定したスリープ開始

通常 sleep より省電力

パラメータ：

なし

戻り値；

なし

I2C(class)

I2C を制御するクラス(I2C 用 GROVE ポートを含む)

new

説明；

クラス生成

パラメータ；

なし

戻り値；

なし

read(addr, data_addr, number)

説明；

スレーブのデータを取得

パラメータ；

Addr：スレーブのアドレス（センサー固有のアドレス：16 進数）

Data_addr：データの読み込み開始アドレス（16 進数）

Number：読み込む byte データの数（10 進数）

戻り値；

String

write(addr, data_addr, data1...)

説明；

スレーブにデータを書き込み

パラメータ；

Addr：スレーブのアドレス（16 進数）

data_addr：データを書き込み開始アドレス（16 進数）

data：書き込むデータたち（16 進数）

戻り値；

なし

status

説明；

ステータスを取得する

パラメータ；

なし
 戻り値：
 整数

1	読み取り転送完了。
2	書き込み転送完了。
4	転送処理中
8	転送が一時停止しました。
その他	エラー

clear_status

説明：
 ステータスフラグの初期化
 パラメータ：
 なし
 戻り値：
 なし

UART(class)

UART 用の GROVE ポートを制御するクラス

new(pin)

説明：
 クラス生成
 パラメータ：
 pin：使用するポート(0,1)
 戻り値：
 なし

read(num)

説明：
 文字列受信
 パラメータ：
 Num：受信文字数

戻り値：
文字列

write(txt)

説明：
文字列送信
パラメータ：
Txt：文字列
戻り値：
なし

gets

説明：
文字列受信(一行読み込み)
パラメータ：
なし
戻り値：
文字列

puts(txt)

説明：
文字列送信
パラメータ：
Txt：文字列
戻り値
なし

clear_tx_buffer

説明：
送信バッファのクリア
パラメータ：
なし
戻り値；

なし

clear_rx_buffer

説明；

受信バッファのクリア

パラメータ；

なし

戻り値；

なし

その他

puts(String)

説明；

microUSB から UART による文字送信(デバッグ用)

ポートを使用 57600;8;NOP;1;CTS;

パラメータ；

String：文字列

戻り値；

なし

leds_write(num)

説明；

ボードに標準搭載されている LED 4 つへ 10 進数を 2 進数に変換して 4 bit 出力を行う

パラメータ；

num：0 ~ 15

戻り値；

なし

SW()

説明；

ボードに標準搭載されているスイッチの状態を取得する。

パラメータ：

なし

戻り値：

0(OFF)か 1(ON)