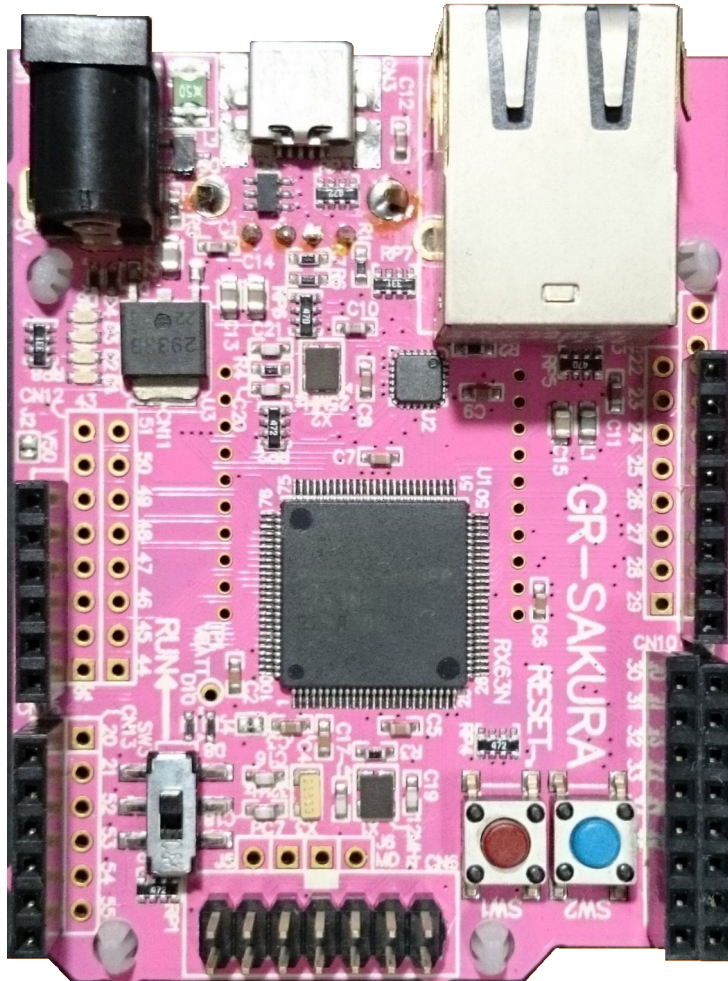


SakuRubyボード

説明資料 ver1.3

Wakayama. rb
たろサ

SakuRubyボード Ver. SakuraJam



特 徴

- ・mrubyを実装したGR-SAKURAです。オブジェクト指向スクリプト言語Rubyを用いてプログラミングできます。作成したプログラムはシリアル経由で書き換えることができます。

ソフト仕様

カーネルクラス

```
pinMode (pin, mode)
digitalRead (pin)
digitalWrite (pin, value)
analogRead (number)
pwm (pin, value)
pwmHz (value)
analogDac (value)
delay (value)
millis ()
micros ()
led (sw)
```

システムクラス

```
System. exit ()
System. setrun (filename)
System. version (r)
System. push (address, buf, length)
System. pop (address, length)
System. fileload ()
System. reset ()
```

ファイルクラス

```
MemFile. open (number, filename [, mode])
MemFile. close (number)
MemFile. read (number)
MemFile. write (number, buf, len)
MemFile. seek (number, byte)
MemFile. copy (src, dst [, mode])
```

シリアルクラス

```
Serial. begin (number, bps)
Serial. setDefault (number)
Serial. print (number, string)
Serial. println (number, string)
Serial. read (number)
Serial. write (number, buf, len)
Serial. available (number)
Serial. end (number)
```

I2Cクラス

```
I2c. sdascl (sda, scl)
I2c. write (id, address, data)
I2c. read (id, addressL, addressH)
I2c. begin (id)
I2c. lwrite (address)
I2c. end ()
I2c. request (id, n)
I2c. lread ()
I2c. freq (Hz)
```

サーボクラス

```
Servo. attach (ch, pin [, min, max])
Servo. write (ch, angle)
Servo. us (ch, us)
Servo. read (ch)
Servo. attached (ch)
Servo. detach (ch)
```

ソフト仕様

コマンドはこんな感じです。

リアルタイムクロッククラス

Rtc.begin()

Rtc.setDateTime(Year, Month, Day, Hour, Minute, Second)

Rtc.getDateTime()

SDカードクラス

SD.open(number, filename[, mode])

SD.close(number)

SD.read(number)

SD.write(number, buf, len)

SD.seek(number, byte)

SD.exists(filename)

SD.mkdir(dirname)

SD.remove(filename)

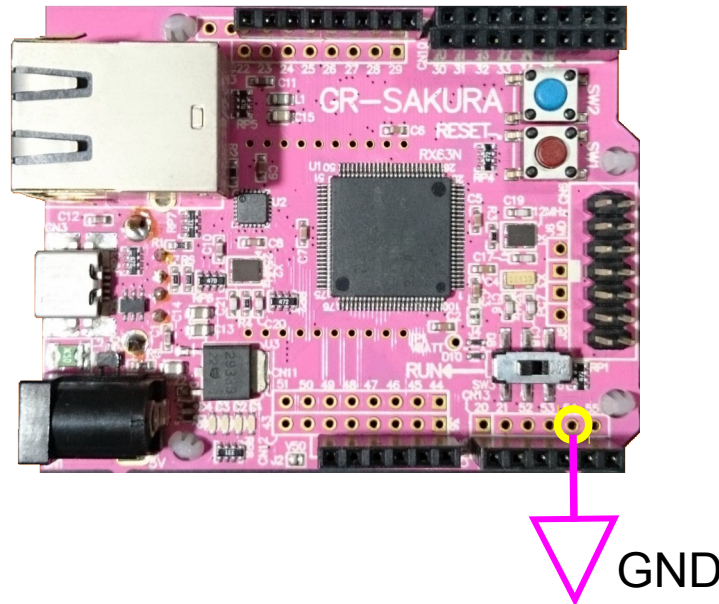
SD.rename(oldfilename, newfilename)

SD.rmdir(dirname)

コマンドモードについては、
WRBB_BOARD4.pdfを参考にしてください。

電源ONで即実行する方法

電源をONすると、直ぐにコマンドモードが起動します。改行を入力するとメニューが表示されます。電源ONで即プログラムを実行したい場合は下記の場所をGNDに落としてください。



印の部分をつなぐと、電源ONでプログラムが走ります。

内部にwrbb.xmlファイルがあり、Startタグで開始プログラム名が書かれているときには、該当プログラムを実行します。無い場合はmain.mrbが実行されます。main.mrbが無い場合は、コマンドモードになります。

Rubyプログラム実行の仕組みについては、
WRBB_BOARD4.pdfを参考にしてください。

mrbbファイルの作成方法については、
WRBB_BOARD4.pdfを参考にしてください。

クラスとメソッドについては、
WRBB_BOARD4.pdfを参考にしてください。

メソッドの説明

SDカードクラス

ファイルのオープン SD. open(number, filename[, mode])

ファイルをオープンします。

number: ファイル番号 0 または 1

filename: ファイル名(8.3形式)

mode: 0:Read, 1:Append, 2:New Create

戻り値

成功: 番号, 失敗: -1

※同時に開けるファイルは2つまでに限定しています。

ファイルのクローズ SD. close(number)

ファイルをクローズします。

number: クローズするファイル番号 0 または 1

ファイルの読み出し位置に移動 SD. seek(number, byte)

Openしたファイルの読み出し位置に移動します。

number: ファイル番号 0 または 1

byte: seekするバイト数(-1)でファイルの最後に移動する

戻り値

成功: 1, 失敗: 0

メソッドの説明

SDカードクラス

Openしたファイルからの読み込み SD.read(number)

Openしたファイルから1バイト読み込みます。
number: ファイル番号 0 または 1

戻り値
0x00~0xFFが返る。ファイルの最後だったら-1が返る。

Openしたファイルにバイナリデータを書き込む SD.write(number, buf, len)

Openしたファイルにバイナリデータを書き込みます。
number: ファイル番号 0 または 1
buf: 書き込むデータ
len: 書き込むデータサイズ

戻り値
実際に書いたバイト数

ファイルが存在するかどうか調べる: SD.exists(filename)

ファイルが存在するかどうか調べます。
filename: 調べるファイル名

戻り値
存在する: 1, 存在しない: 0

メソッドの説明

SDカードクラス

ディレクトリを作成する: `SD.mkdir(dirname)`

ディレクトリを作成する。
dirname: 作成するディレクトリ名

戻り値
成功: 1, 失敗: 0

ファイルを削除します: `SD.remove(filename)`

ファイルを削除します。
filename: 削除するファイル名

戻り値
成功: 1, 失敗: 0

ファイル名を変更します: `SD.rename(oldfilename, newfilename)`

ファイルを削除します。
oldfilename: 旧ファイル名
newfilename: 新しいファイル名

戻り値
成功: 1, 失敗: 0

メソッドの説明

SDカードクラス

ディレクトリを削除します: `SD.rmdir(dirname)`

ディレクトリを削除します。
dirname: 削除するディレクトリ名

戻り値

成功: 1, 失敗: 0

メソッドの説明

SDカードクラス

使用例

```
SD.open(0, 'sample.txt', 2)
  SD.write(0, 'Hello mruby World', 17)
  data = 0x30.chr + 0x31.chr + 0.chr + 0x32.chr + 0x33.chr
  Serial.write(0, data, 5)
SD.close(0)

Serial.begin(0, 115200)      #USBシリアル通信の初期化

SD.open(0, 'sample.txt', 0)
while(true)do
  c = SD.read(0)
  if(c < 0)then
    break
  end
  Serial.write(0, c.chr, 1)
end
SD.close(0)
System.exit()
```