

50×18mm の極小 Ruby ボードです。

システム概要

マイコンには、**mruby**が実装されており、**Serial 1**を使って**3Gシールド**とやり取りができる**Ruby**のクラスが用意されています。

基板はArduino Pro Miniとピン互換、GR-SAKURAと機能互換となっています。GR-SAKURA互換基板として使用するには、GR-SAKURAファームを書き込む必要があります。

赤文字ピン番は5Vトレラント

ダウンロードすべきファイルが何で、ダウンロード先はどこかを示したファイルをダウンロードさせることもできる。その場合は、さらにシステムがフレキシブルになる。

プログラム書き換えの仕組み

インターネット・クラウド
DropBoxファイル
公開サーバ上ファイル

インターネット上のサーバにapacheなどを入れておき、http/httpsでプログラムが取得できるようにしておく。

下記のサイトからのファイル取得が可能なが確認できている。

- ・DropBox (httpsによる取得)
- ・Yahoo ジオシティーズ (httpによる取得)

mrubyボードには、httpを用いて指定ファイルが取得できるrubyクラスが実装されている。

実行中のプログラム
自分自身を終了させて、ダウンロードしたプログラムを呼び出す

ダウンロードしたプログラム → ダウンロードしたプログラムの呼び出し → ダウンロードしたプログラムの実行

mruby VM が動いているので、アプリ部分の書き換えができる。欠点としては、高速な処理は苦手という点である。



現在は、これだけだが、将来的にサーボクラスと SPI クラスを実装したい。



分かりにくいですが、ダウンロードするプログラムが存在するかどうかをループして待つ Ruby プログラム例です。

rubyプログラム

geocitiesからrubyプログラムをダウンロードするプログラム例です。

```

while(true)do
  #http://www.geocities.jp/cansatkainan01/からプログラムをダウンロードする
  x = Net.getrbl("www.geocities.jp"/cansatkainan01/sensor-a.mrb", 80)

  #x-0だったら取得に成功している
  if(x == 0)then
    #sensor-a.mrbの実行セット
    Sys.setrun("sensor-a.mrb")
    delay(100) #100ms待つ

    #Success load sensor-a.mrb #cansatkainanとツイートする
    mes = "Success load sensor-a.mrb #cansatkainan"
    res = Net.httpost("arduino-tweet.appspot.com", "update", 80, "", #Token + mes)

    #今走っているプログラムを終了させる
    Sys.exit()
  end

  #Dat = Net.date()
  #Tim = Net.time()
  #Lat = Net.lat(60000)
  #Lng = Net.lng(60000)

  #時間待ち。予定の時間になるまで持っている
  while(millis() < #toTime)do
    #ファイルローダーを呼び出すべきか調べる
    fileloader()
    delay(1) #1ms待ちます
  end

  #次の動作時間をセットする
  nextTime()

  #LEDを点滅させる
  led(#Sw)
  #Sw = 1 - #Sw
end
  
```

mrblファイル名を指定してダウンロードしているダウンロード成功の場合そのmrblファイルが次に実行されるファイルとして登録される

exit()によりプログラムを終了させると登録されたmrblファイルが実行される

Http POSTを利用してtweetしている

実際の使用例は、和歌山県立海南高校の生徒が、この基板を用いて、疑似人工衛星の製作と成果を競い合う大会に出場したときの例です。

実際の使用例

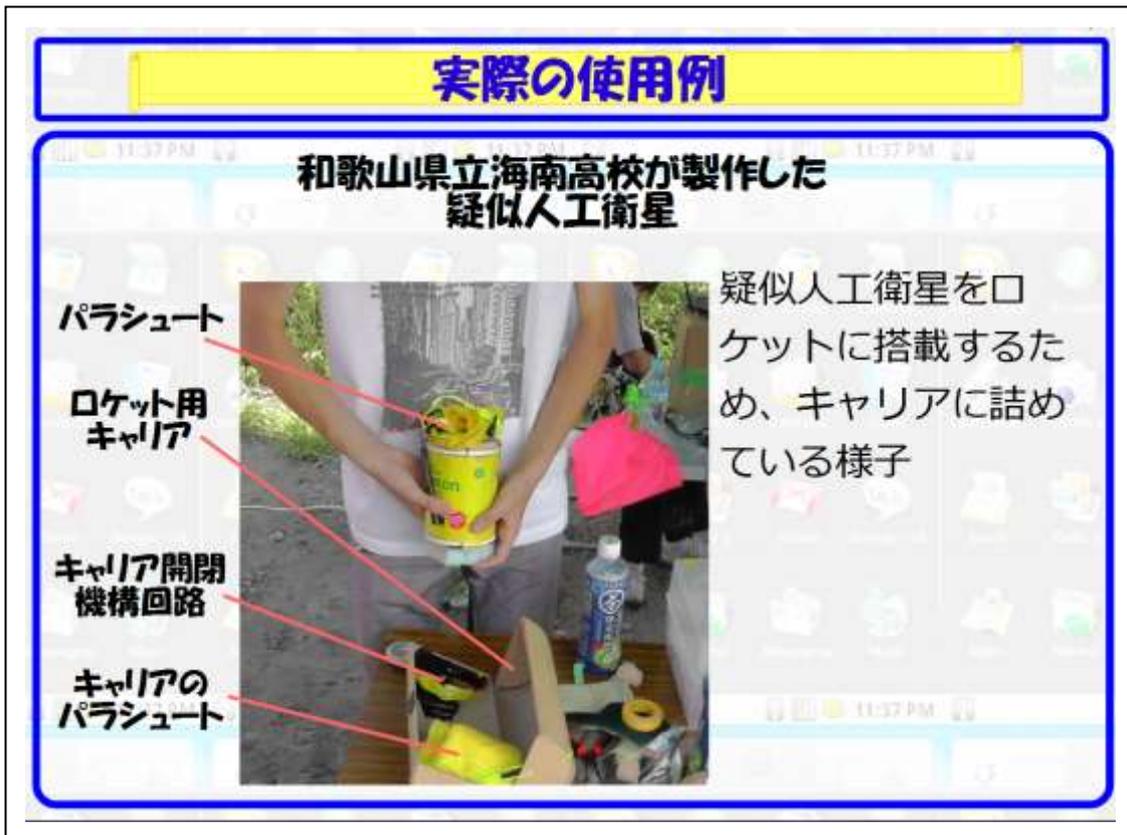
和歌山県立海南高校が製作した疑似人工衛星

- 加速度&ジャイロ センサA
- 加速度&ジャイロ センサB
- RubyボードA
- RubyボードB
- 3Gシールド
- ログ用SDメモリ X2
- バッテリーと bluetooth基板 は取り外し中
- 温度&気圧センサ

ネットワークアクセス用のRubyボード(A)と、センサ制御用のRubyボード(B)の2枚のボードを搭載している。

衝撃吸収用クッション

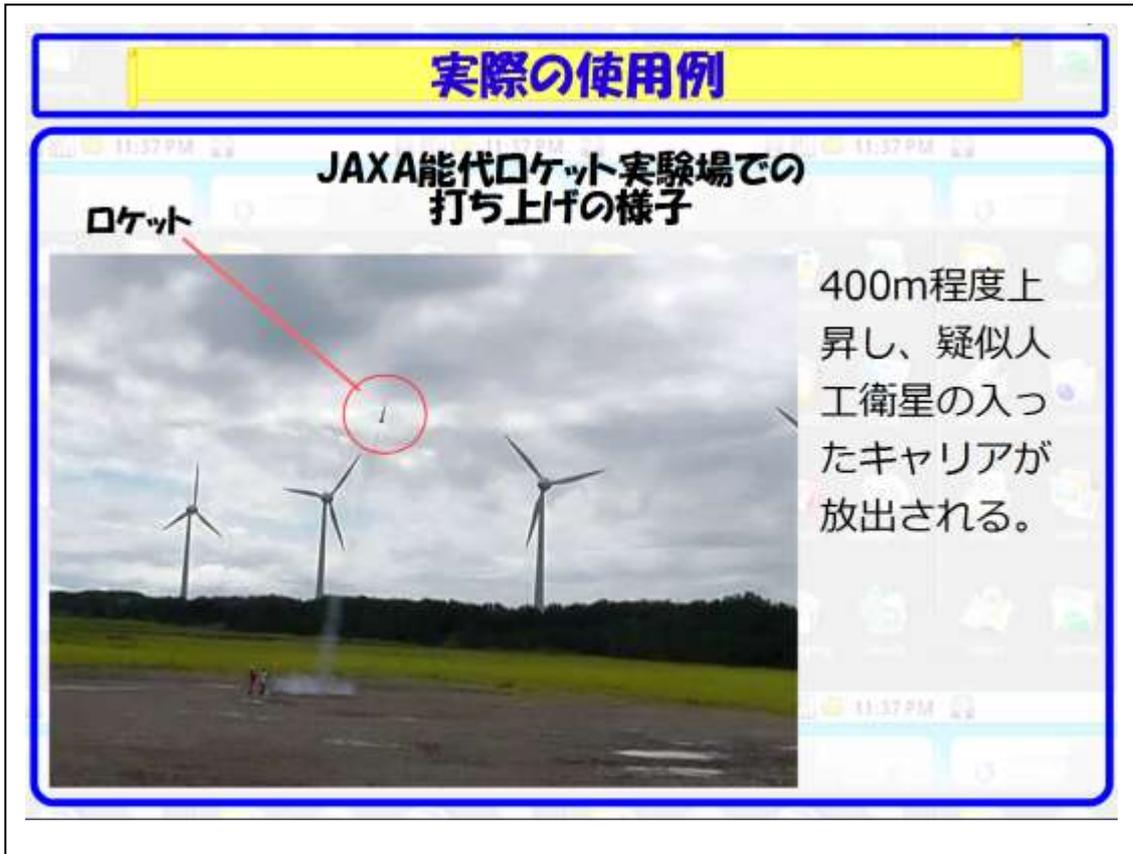
人工衛星のセンサ処理とネットワークの処理を 1 枚の Ruby ボードで行うことが難しかったので、2 枚の基板を用いて分散処理をさせています。



ロケットはパーツが支給されて、尾翼と先端コーンの組み立て、疑似人工衛星の搭載は生徒たちで行います。



打ちあがっている時点で、内部プログラムのダウンロードと書き換えは終了していました。



クラウドにあるサーバからのプログラム書き換えは成功しています。

