

GR-CITRUSとMIKANを試す

自己紹介:

神戸市から来ました。ものづくり大好きおじさん

職業は、ものづくりを子供達に教えるものづくり体験講師

初めてGR-CITRUS IoTワークショップに参加(2016/11/19)

mrubyに触れてその使い易さに感動！！

なにか作ってみようと久しぶりに奮闘

2016.12.17

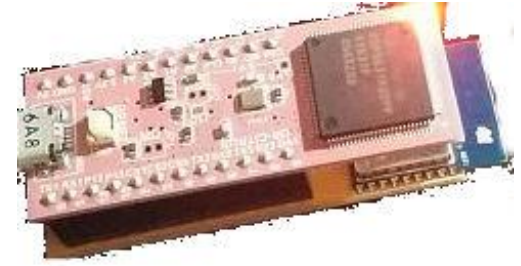
奥山 悦雄

GR-CITRUSとMIKANについて

- ①マイコン:RX631(R5F5631FDDFP)
- ②ROM:2MB(USB MSC firmwareを使うと960KB)
- ③RAM:256KB
- ④メインクロック:96MHz(12MHz×8)
- ⑤サブクロック:32.768KHz
- ⑥供給電圧:USB供給5V 基板上に3.3Vのレギュレータを内蔵
- ⑦動作電圧:3.3V

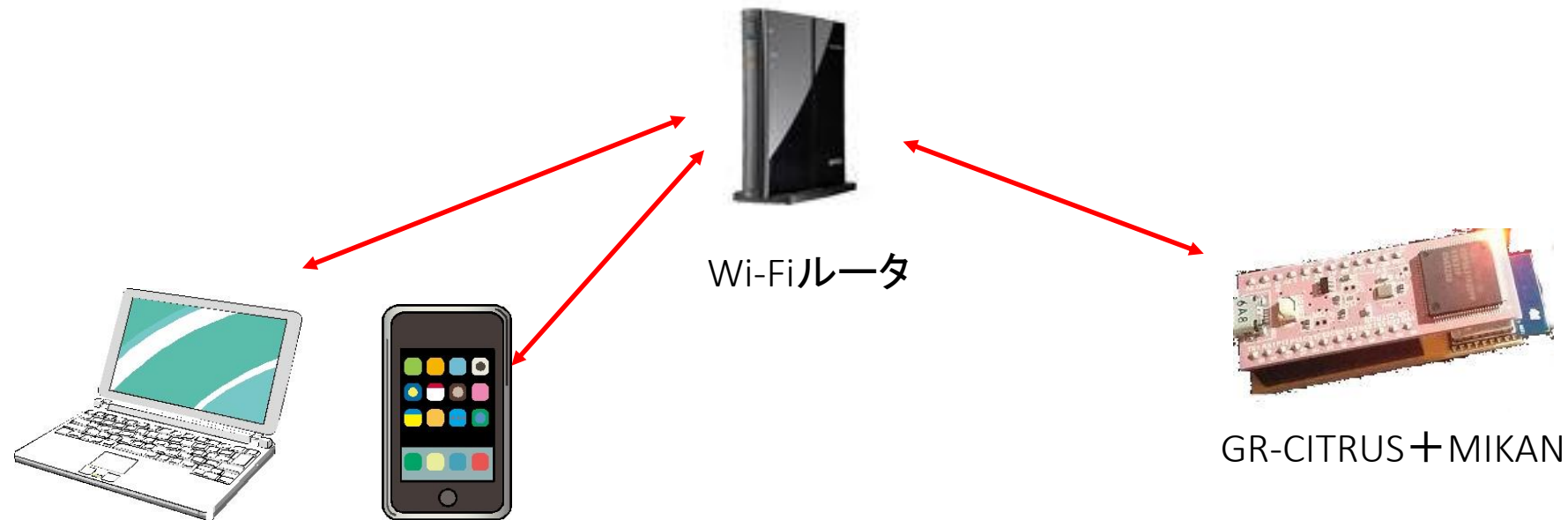
【Wi-Fiモジュール】

Wi-Fiモジュールはおなじみの「ESP8266」を搭載。また、マイクロSDソケットも併設。両モジュールをドッキングすると上記のようにスタックで非常にコンパクト

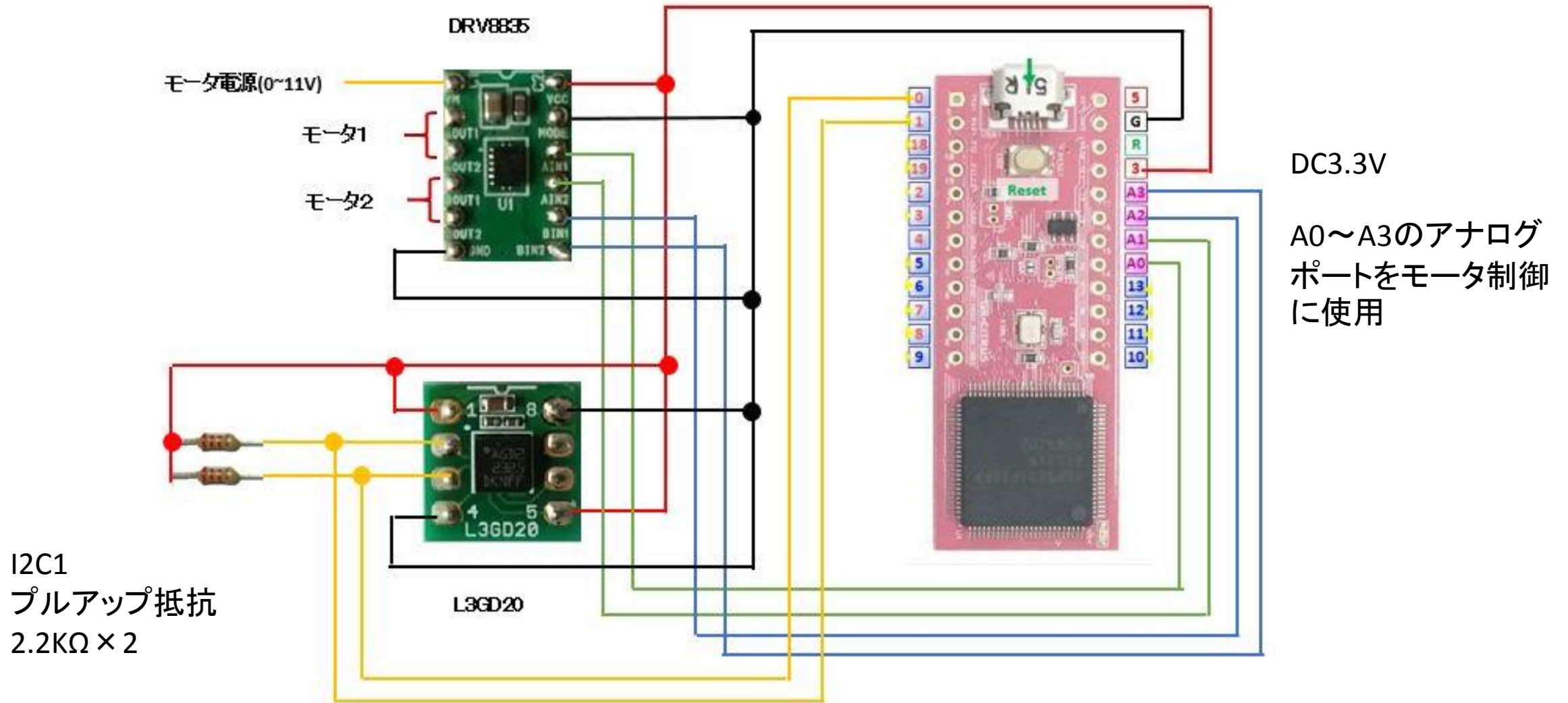


製作のために考慮したこと

- mrubyを使ってGR-CITRUSで動くものをつくる。
- MIKANを使ってGR-CITRUSを端末(PC&Android携帯)側からWi-Fi (UDP)で遠隔操作
- 端末側からの要求によりセンサからのデータをWi-Fiで端末側に送信



実態配線図



製作した制御基板

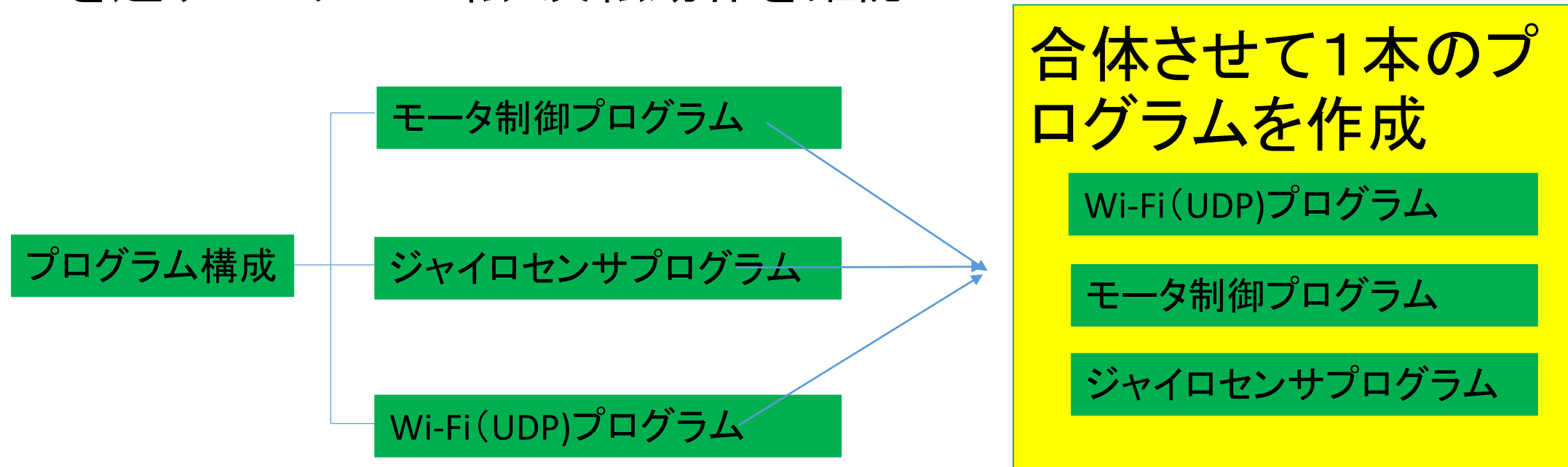
- ・47mm×72mmのユニバーサル基板に実装した制御基板



mrubyによるプログラム作成

- 個々のプログラムの動作を確認して合体させるようにした。

例えばモータ制御プログラムではPCからUSBシリアルを介してコマンドを送りモータの正転・反転動作を確認



mrubyは文法をあまり意識せずに作成できる生産性の高い良い言語である。

モータ制御のしくみ

- 動作はUSBシリアルでPCからGR-CITRUSにコマンドを送るとそれに合わせてモータ制御モジュールDRV8835をコントロールするものです。
- PCから送るコマンドは5種あり
 - ”s” : 全モータストップ
 - ”f” : 前進 (両モータ共前進)
 - ”b” : 後退 (両モータ共後退)
 - ”l” : 左回転 (モータ1後退、モータ2前進)
 - ”r” : 右回転 (モータ1前進、モータ2後退)
- USBシリアルのボーレートは230400bps
- GR-CITRUSのシリアルSerial4を使用し、TX/RXはそれぞれピン12/11に接続

モータ制御プログラムの例

```
#!/ruby
```

```
@Uart = Serial.new(4, 230400)//#TX/RX:pin12/pin11, baud rate: 230400
```

```
def set_up()
```

```
  #//L Motor PinSET
```

```
  pinMode(14, 1)
```

```
  pinMode(15, 1)
```

```
  #//R Motor PinSET
```

```
  pinMode(16, 1)
```

```
  pinMode(17, 1)
```

```
End
```

```
def motor_stop()
```

```
  pwm(14, 0)
```

```
  pwm(15,0)
```

```
  pwm(16, 0)
```

```
  pwm(17, 0)
```

```
end
```

モータを制御する4本のコントロール信号はアナログ用のピン14～17に接続し、デジタル出力として使用

信号をPWMで出力し、モータ停止のため全て“0”を指定する。


```
def l_motorf()  
  pwm(14, 0)  
  pwm(15, 255)  
End
```

～r_motorb()まで定義する。

```
@Uart.println("setup")
```

```
  set_up()
```

```
  motor_stop()
```

```
  #//while (true) do
```

```
    2000.times do
```

```
      led 1
```

```
      #//コマンド解読
```

```
      if (@Uart.available())then #//シリアルデータを受け取ったたら
```

```
        readc = @Uart.read() #//データを読みに行く
```

```
        #//FF
```

```
        if(readc=="f")then
```

```
          l_motorf()
```

```
          r_motorf()
```

```
          @Uart.println("motor FF")
```

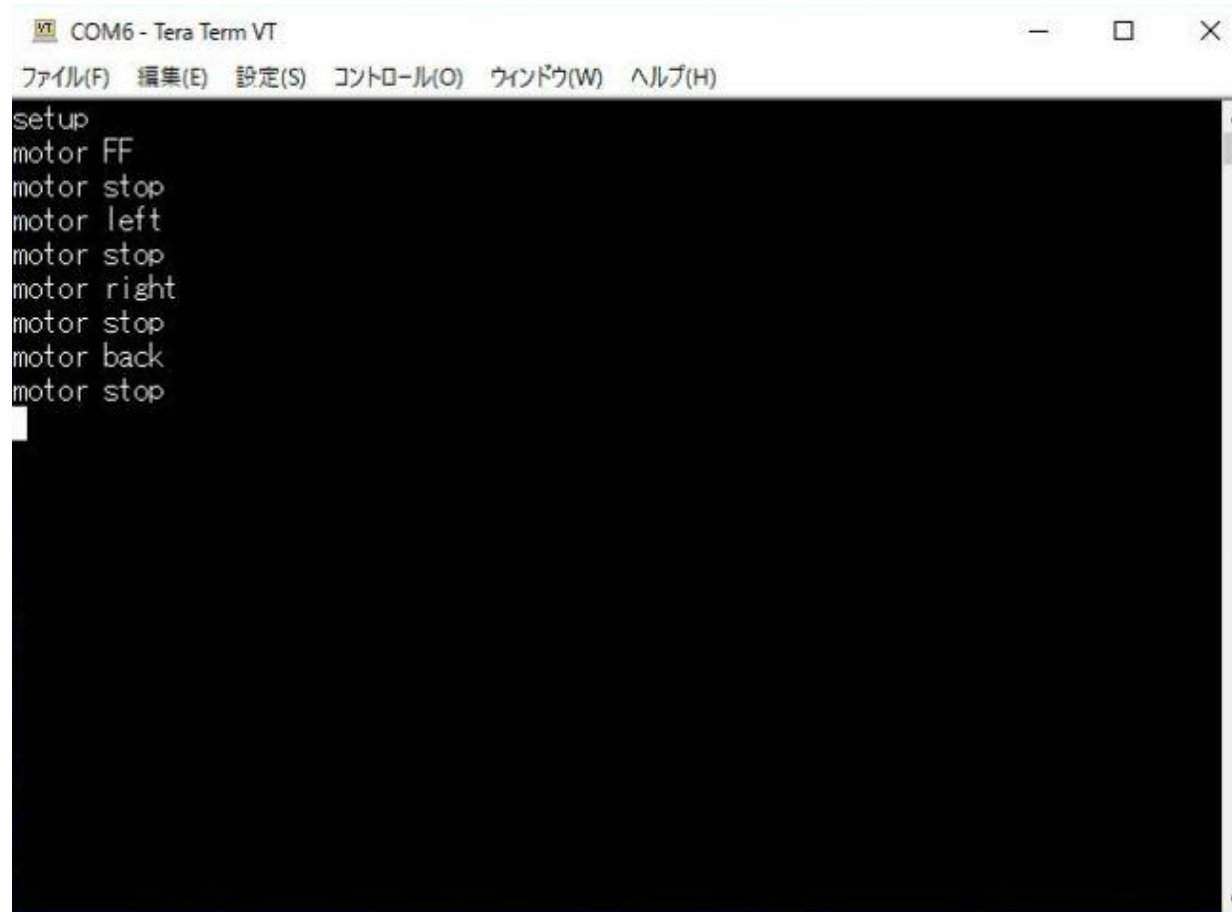
```
        end
```

Lモータを前進させるため14ピンを”0”，15ピンを255(最大値)に指定

“f”コマンドを受信したらLモータおよびRモータを前進させる。

モータ制御動作確認

TeraTermからコマンドを送って、GR-CITRUSがコマンドを受け取るとモータのSTATUS情報をTeraTermに返す



```
COM6 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
setup
motor FF
motor stop
motor left
motor stop
motor right
motor stop
motor back
motor stop
```

Wi-Fi (UDP) 接続について

#ESP8266を一度停止させる(リセットと同じ)

```
pinMode(5,1)
digitalWrite(5,0) # LOW:Disable
Delay 500
digitalWrite(5,1) # HIGH:Enable
  if( System.useWiFi() == 0)then
    @Usb.println "WiFi Card can't use."
    System.exit()
  end
@Usb.println "WiFi Ready"
@Usb.println "WiFi Get Version"
@Usb.println WiFi.version
@Usb.println "WiFi disconnect"
```

MIKANのEnableポートを500ms間
LOWにして一旦リセット

MIKANが使えるかチェック

MIKANが使用可能であればその
バージョンを表示して一旦接続を閉
じる

```
@Usb.println WiFi.connect("xxxxxxxxxx","yyyyyyyyyy")
```

```
@Usb.println "WiFi ipconfig"
```

```
@Usb.println WiFi.ipconfig
```

```
@Usb.println "Set wifi mode"
```

```
@Usb.println WiFi.setMode 3 #Station-Mode & SoftAPI-Mode
```

```
@Usb.println "Enable multiple connection"
```

```
@Usb.println WiFi.multiConnect 1
```

```
@Usb.println "Create a UDP transmission, for example, id is 4"
```

```
#UDP通信,WAMIKAN受信:5555, 送信: 5556
```

```
@Usb.println WiFi.udpOpen(4,"192.168.1.3",5555,5556)
```

～

```
if(array[0] == 0x65) then #// "e"
```

```
@Usb.println WiFi.cClose 4
```

```
@Usb.println WiFi.disconnect
```

```
break
```

xxxxxxxxxxにWi-FiルータのSSIDを入力
yyyyyyyyyyにWi-FiルータのKEYを入力

StationとSoftAPIモードを選択
マルチコネクション4接続までを選択

接続番号4で相手のIPアドレス
192.168.1.3に受信ポート5555, 送信
ポート5556でUDP通信開始

コマンド"e"を受信するとWi-Fi (UDP)
接続を閉じる

Wi-Fi(UDP)での受信

```
array = WiFi.recv 4 #受信データがない場合は array[0]に -1 が返ります
```

```
if(array[0] >= 0)then
```

```
  @Usb.println array[0].to_s
```

```
  if(array[0] == 0x66) then #//"f"
```

```
    WiFi.send 4, "MC11"
```

```
    l_motorf()
```

```
    r_motorf()
```

```
    @Usb.println("motor FF")
```

```
  end
```

```
  ~
```

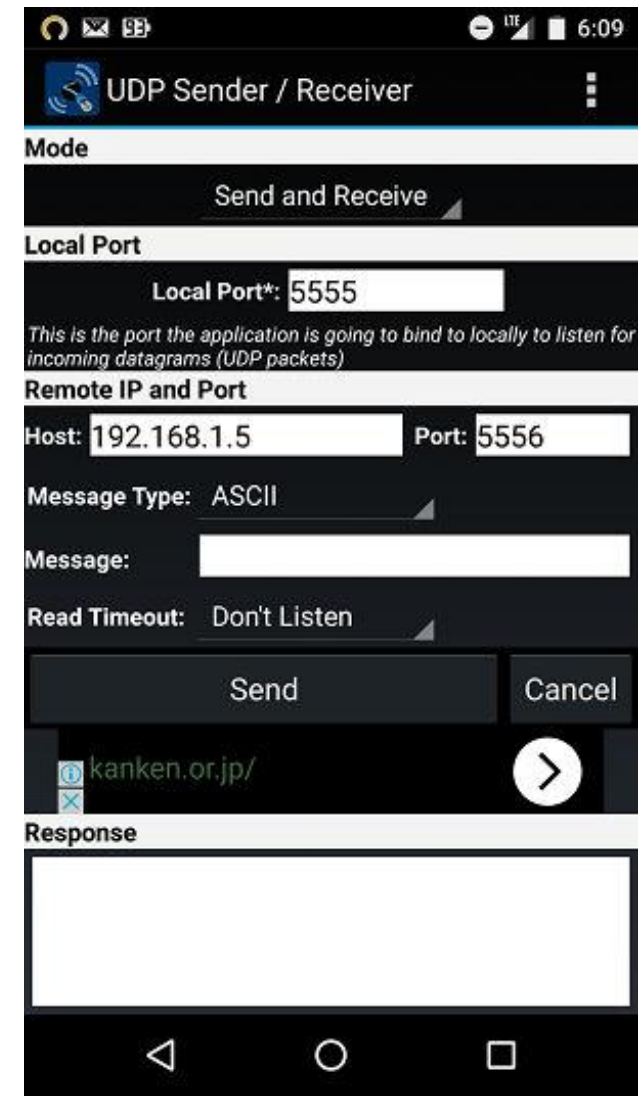
```
end
```

Wi-Fiの接続4を見に行き、もし受信データがあればその先頭array[0]を文字に変換する。

先頭の文字が0x66、コマンド“f”であれば端末側に“MC11”を返す。またモータ制御をL,Rモータ共前進させる。

Wi-Fi(UDP)の接続確認方法

- 接続を確認するためにAndroid携帯に「UDP Sender/Receiver」ソフトをインストール
- 右の画面のようにテキストボックス3か所に入力
 - ①Local port: 受信ポート5555
 - ②Host: GR-CITRUS側のIPアドレス
この例では“192.168.1.5”
 - ③Port: 送信ポート5556
- Messageテキストボックスに送信する文字列を入力してSendボタンを押すと文字列が送信される
- 相手から受信する時にはRead Timeoutを「Infinite」にすると下のResponse欄受信した文字列が表示されます。



ジャイロセンサのデータ送出

- ・ジャイロセンサから取得したデータを小数点以下1桁で四捨五入

```
v0 = (v0 * @L3GD20_dps).round(1)
```

```
v1 = (v1 * @L3GD20_dps).round(1)
```

```
v2 = (v2 * @L3GD20_dps).round(1)
```

取得したデータv0～v2に角速度の分解能
@L3GD20:0.0175を掛けて角速度を求める。
そしてround(1)により小数点以下1桁に丸
める

- ・データ送出コマンド“m”を受け取るとヘッダ“MA”とスペース0x20を付け文字列に変換して送出

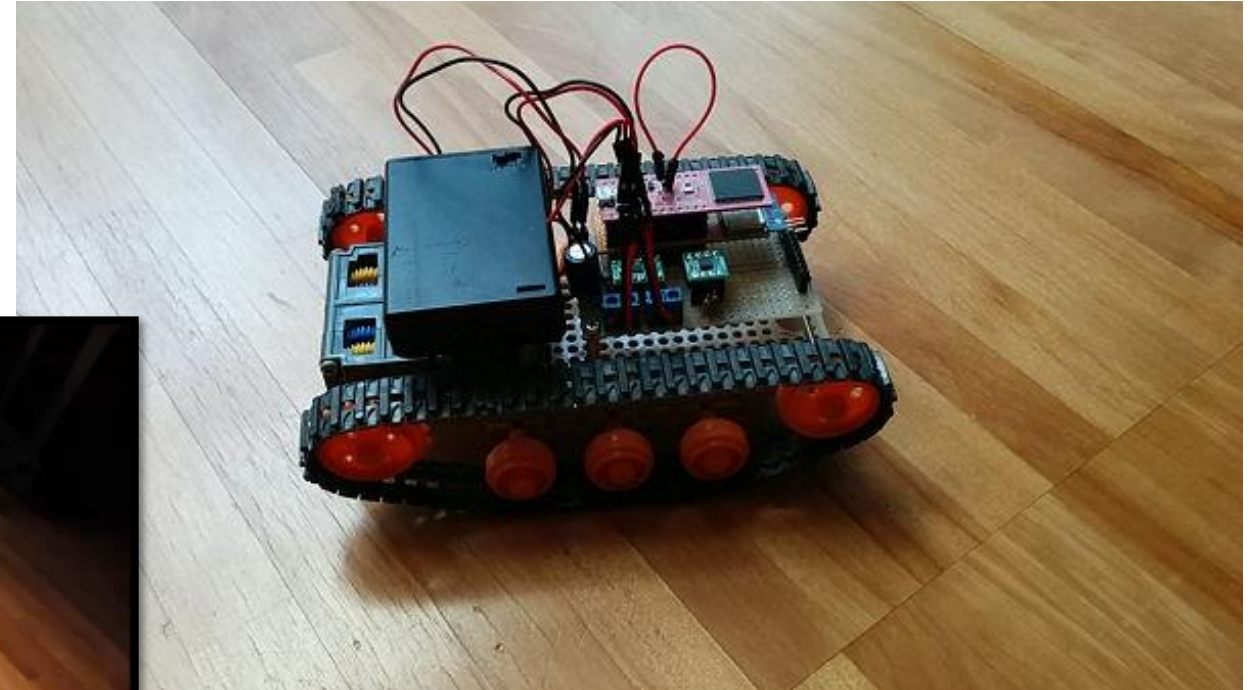
```
if(array[0] == 0x6D) then #//"m"
```

```
  @Usb.println WiFi.send(4, "MA" + 0x20.chr + v0.to_s + 0x20.chr + v1.to_s + 0x20.chr + v2.to_s + "¥r¥n").to_s
```

```
end
```


製作したもの

タミヤのタンク部品に制御基板を搭載



実際に動かした動画がこちら

最後に

- GR-CITRUSとMIKANを試してみて

mrubyという非常に使い易い言語と良く考えられたコンパクトな基板により素晴らしいものづくりができる環境が出来たと思います。

これからも機会があれば、この環境を使って子供達にもものづくりの楽しさを教えてあげたい！！

コンパクトな基板を設計された山本様、Rubicによるmrubyの開発環境を提供してくれたkimu shu様、がじえるねの皆様感謝！！

Web blog URL: <http://jf3rgskun.hatenablog.com/>