

GR_ROSE向けのMicroPython 使い方

2021年12月25日 更新

MicroPython for GR_ROSE

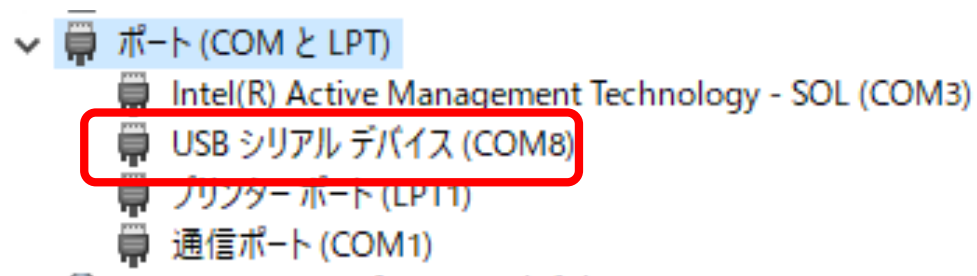
- MicroPythonのSTM32の実装をGR_ROSE(RX65N)に移植したものです。
- 移植した機能はSTM32向けの実装(pyboard)にほぼ準拠しています。
 - 使い方の詳細はpyboardのマニュアル(<https://docs.micropython.org/en/latest/>)を参照してください。
 - ただし、USB, CAN, WDT...は未実装です。
 - Pyboardの一部のモジュールのパラメータが実装できていないものがあります。詳細はソースコードを参照ください。
 - MicroPythonのESP8266およびESP32向けの実装とはモジュールが異なります。

用意するもの

- Windows 10が動作するPC
 - ターミナルソフトウェア(ここではTera Termを使用します。)
- GR_ROSE
- マイクロUSBケーブル

使い方

- GR-ROSEのUSBマイクロコネクタとWindow 10 PCのUSBコネクタを接続します。
- デバイスマネージャのポートでUSBシリアルデバイスとして認識されます。

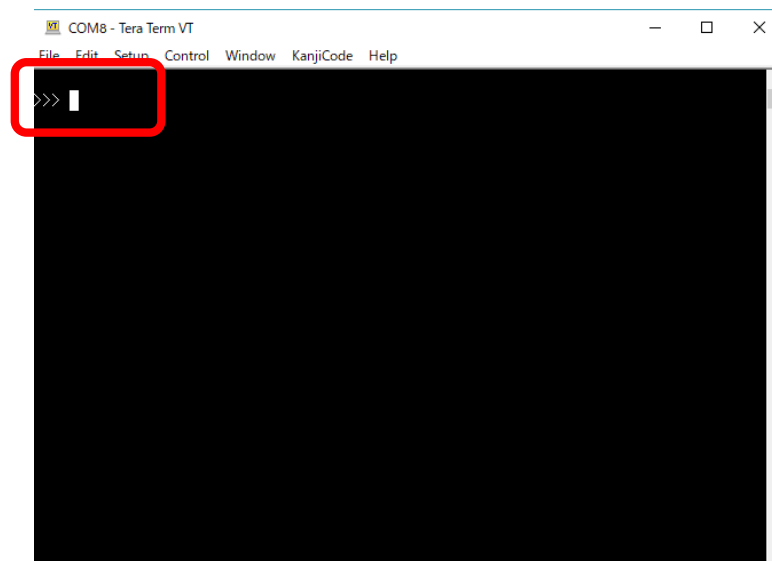
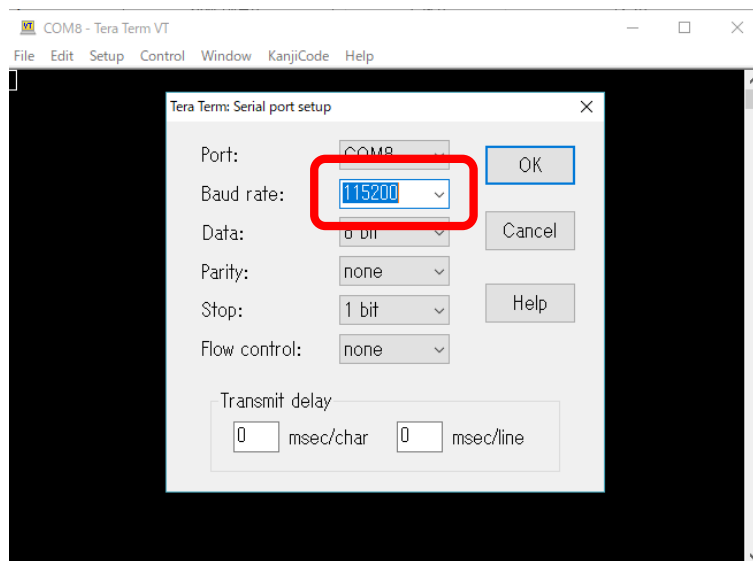
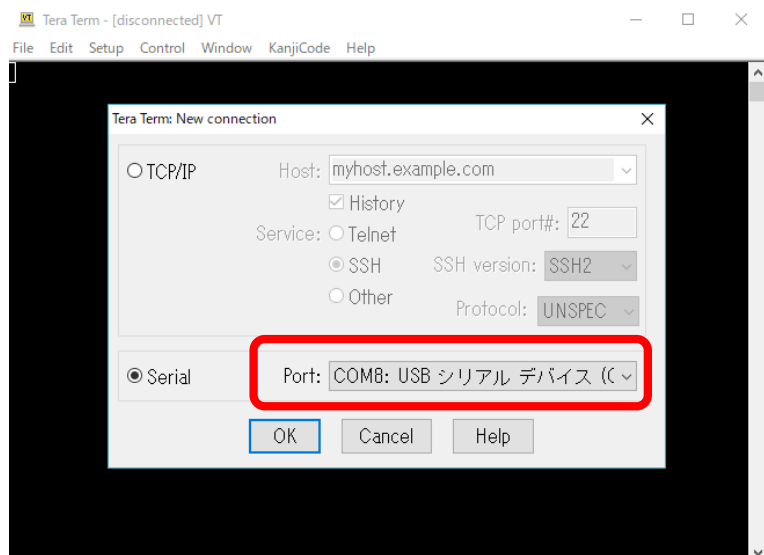


MicroPythonのインストール

- GR-ROSEのディップスイッチをプログラム書き込み用(P)に設定します。
- GR-ROSEをUSBケーブルでホストPCに接続します。
- GR-ROSEをリセットします。
- ドライブがマップされたら、MicroPythonのバイナリ (MPY_GR_ROSE_DD.bin)をドラック&ドロップして書き込みます。
- GR-ROSEのディップスイッチを実行モード(RUN)に設定します。
- GR-ROSEをリセットします。

使い方

- Tera Termを起動し、認識されたCOMポートを選択し、Setup – Serial portメニューを選択し、Baud rateで115200を選択し、Enterキーを押します。>>というMicroPython REPLプロンプトが表示されるはずです。
- このコンソールより、MicroPythonのプログラムが実行できます。



最初のサンプル Lチカ

- 最初のサンプルとして、GR-ROSEボード上のLEDを点灯してみます。
- 下記のプログラムを入力してみます。
- Enterキーを数回入力します。
- Ctrl-Cキーの入力でプログラムが終了します。

```
import pyb
while True:
    pyb.LED(1).toggle()
    pyb.delay(50)
```

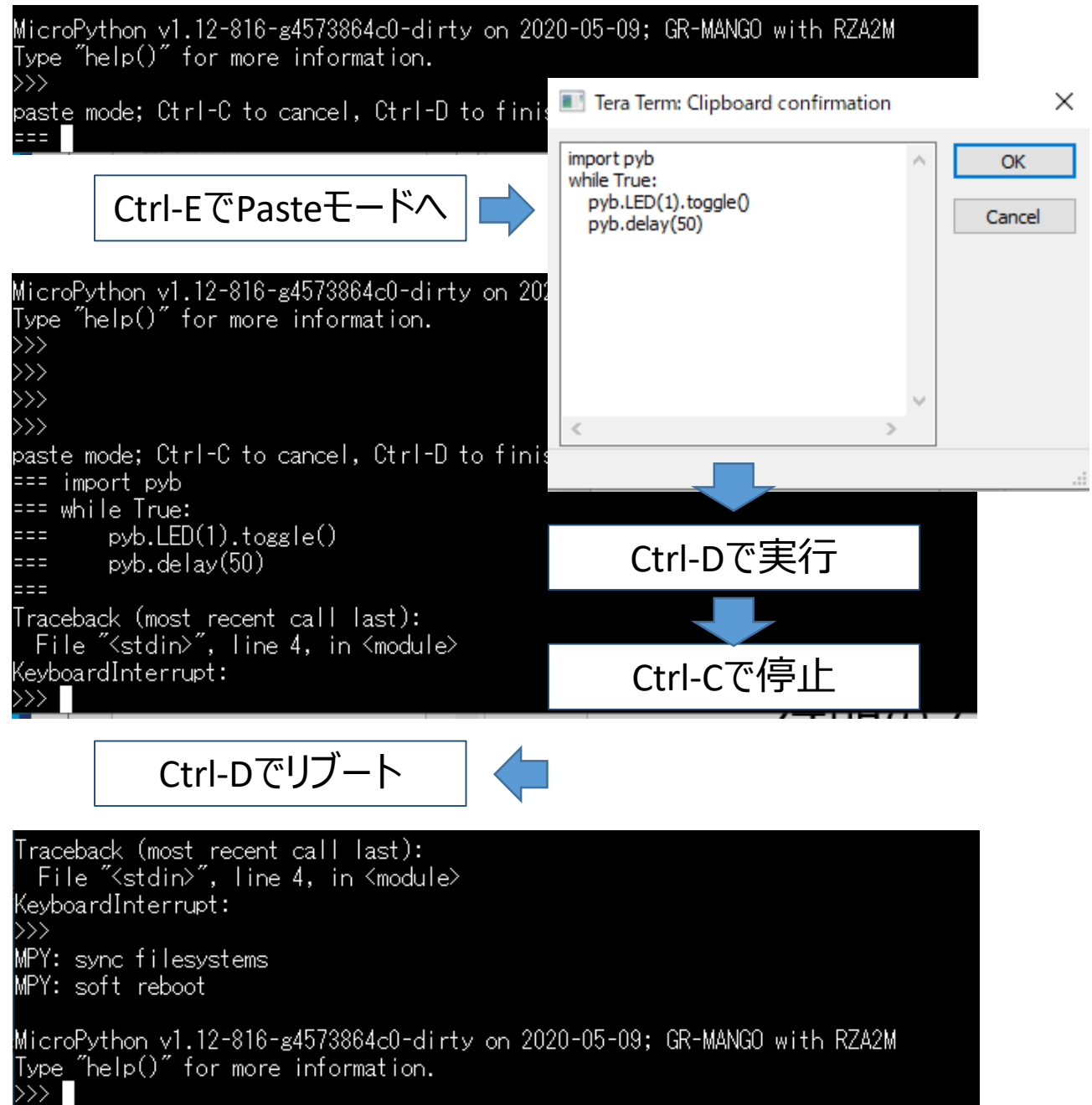


```
COM8 - Tera Term VT
File Edit Setup Control Window KanjiCode Help

>>> import pyb
>>> while True:
...     pyb.LED(1).toggle()
...     pyb.delay(50)
...
...
...
Traceback (most recent call last):
  File "<stdin>", line 3, in <module>
KeyboardInterrupt:
>>> █
```

REPLの補足

- インデントのあるプログラムをCut&Pasteする際は、Ctrl-Eを押してから、Pasteすると便利です。
- Ctrl-Dでプログラムを実行します。
- Ctrl-Cでプログラムを停止します。
- さらにCtrl-Dでソフトウェアリブートします。



ストレージ

- GR_ROSEの実装では、ストレージとして、内蔵FlashとSDカードが使用できます。
- 内蔵フラッシュは、仮に0xffffa0000より256KB確保しています。
- 起動時に、”/flash”デフォルトのフォルダとなり、初期化後はboot.pyとmain.pyの2つのファイルが置かれています。
 - main.pyを更新することで起動時にプログラムを実行することができます。
- ~~起動時にSDカードが挿入されている場合には”/sd”フォルダがデフォルトのフォルダとなります。~~

```
MicroPython v1.12-816-g4573864c0-dirty on 2020-05-09; GR-MANGO with RZA2M
Type "help()" for more information.
>>> import uos
>>> uos.listdir("")
['boot.py', 'main.py']
>>> uos.listdir("/")
['flash']
```

Visual Studio Code + PyMakrによるプログラム編集

View – Command PaletteからPyMakrのメニューにアクセス

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** A list of files in the 'PYMAKR-SAMPLE' folder, including `blink.py`, `boot.py`, `i2c_scan.py`, `main.py`, `max7219.py`, `opt3001_sample.py`, `opt3001.py`, `pymakr.conf`, and `README.md`. A green box highlights this list with the text "PC上にフォルダを作成ソースファイル格納".
- EDITOR:** The `blink.py` file is open, showing the following code:

```
1 import pyb
2 while True:
3     pyb.LED(1).toggle()
4     pyb.delay(500)
5
6
```

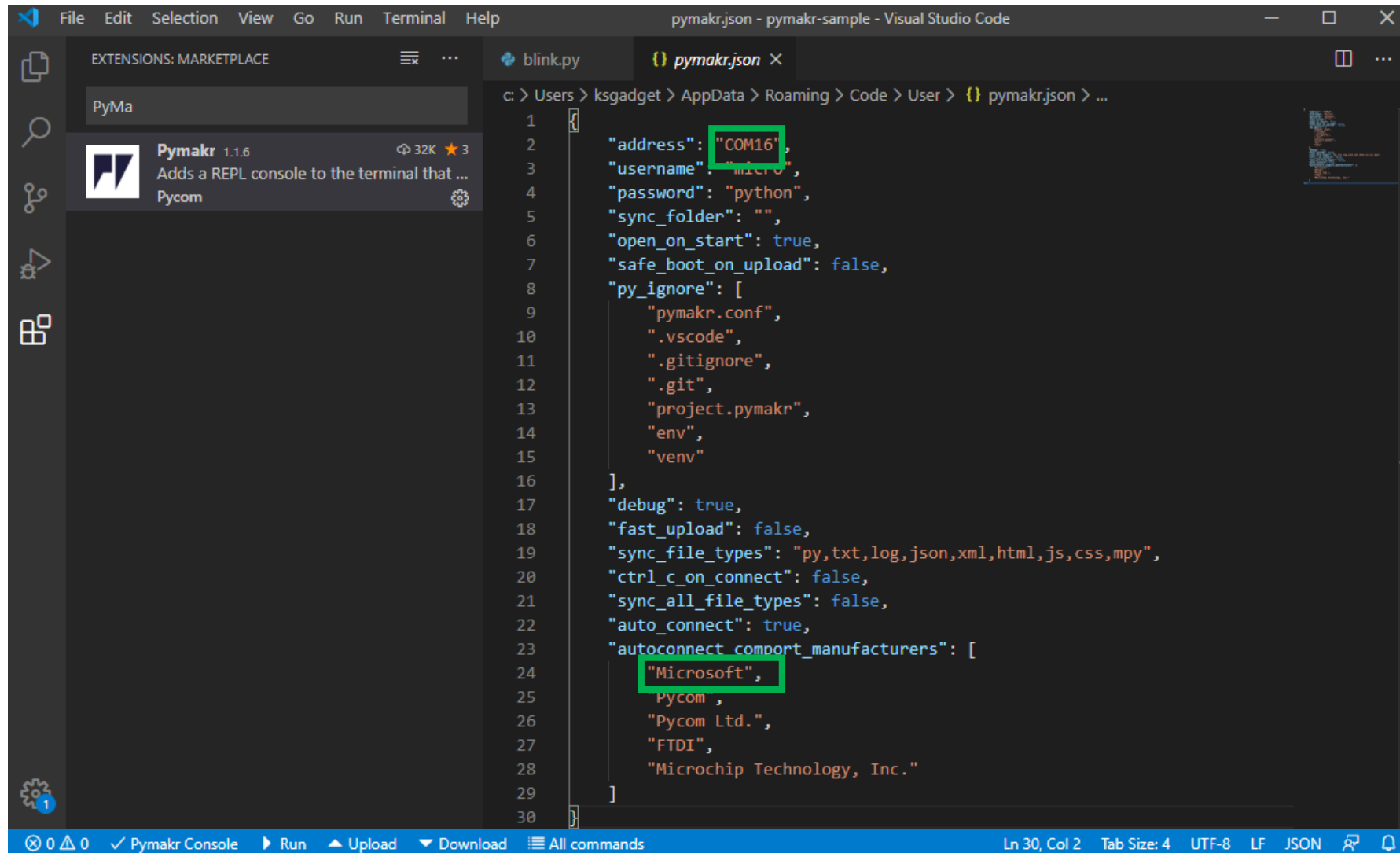
A green box highlights the editor with the text "PC上のフォルダにソースファイルを編集する".
- TERMINAL:** The terminal shows the output of the PyMakr console, including "AutoConnect enabled, ignoring 'address' setting (see Global Settings)", "Searching for PyCom boards on serial...", and "Connecting to COM16...". A green box highlights the terminal with the text "GR_ROSE上のMicroPythonのREPLに接続プログラムを実行する".
- COMMAND PALETTE:** The 'View' menu is open, showing the 'Command Palette' option. A green box highlights this menu with the text "PyMakrのメニュー。ファイル実行。アップロード、ダウンロードが可能。".

PyMakrのメニュー。ファイル実行。アップロード、ダウンロードが可能。

Visual Studio Code + PyMakrのインストール

- Visual Studio Codeのインストール
 - <https://visualstudio.microsoft.com/ja/>
- Nodejsのインストール (6.9.5 以降) たぶん、最新版で問題ない
 - <https://nodejs.org/en/>
- Visual Studio CodeにPyMakerプラグインのインストール
 - 左下の拡張メニューアイコンをクリックし、EXTENSIONS:MARKETPLACEの検索欄にPyMakrと入力し、インストール
- PyMakerプラグインにGR_ROSEのCOMポートなどを登録
 - View – Command Paletteメニューで、PyMakr – Global settingsを選択し、pymakr.jsonファイル中の”address”: ”xxxx”のxxxxにCOMポートを設定

PyMakr – Global settings



PyMakr Tips

- PyMakrのファイルは、以下のフォルダに置かれている。
 - C:¥Users¥ユーザ名¥.vscode¥extensions¥pycom.pymakr-1.1.6
- PyMakrの設定fileは、以下のフォルダに置かれている。
 - C:¥Users¥ユーザ名¥.AppData¥Roaming¥Code¥User
- Windows環境でCOMポート接続が断続する場合には、C:¥Users¥ユーザ名¥.vscode¥extensions¥pycom.pymakr-1.1.6¥lib(and ¥src)¥connections¥pyserial.jsの139行目ぐらいからのコメントアウトする。
- Uploadに失敗する場合、config.jsでupload_chunk_sizeを512から256に設定する。

```
sendPing(cb) {  
  //if (process.platform == 'win32') {  
  // avoid MCU waiting in bootloader on hardware restart by setting both dtr and rts high  
  // this.stream.set({ rts: true });  
  //}  
  // not implemented  
  if (this.dtr_supported) {  
    this.stream.set({ dtr: true }, function (err) {  
      if (cb) {  
        cb(err);  
        return err ? false : true;  
      }  
    });  
  } else {  
    cb();  
    return true;  
  }  
}
```

CN3_R	RESET
CN3_M	FINED
CN3_G	GND
CN3_V	3.3V

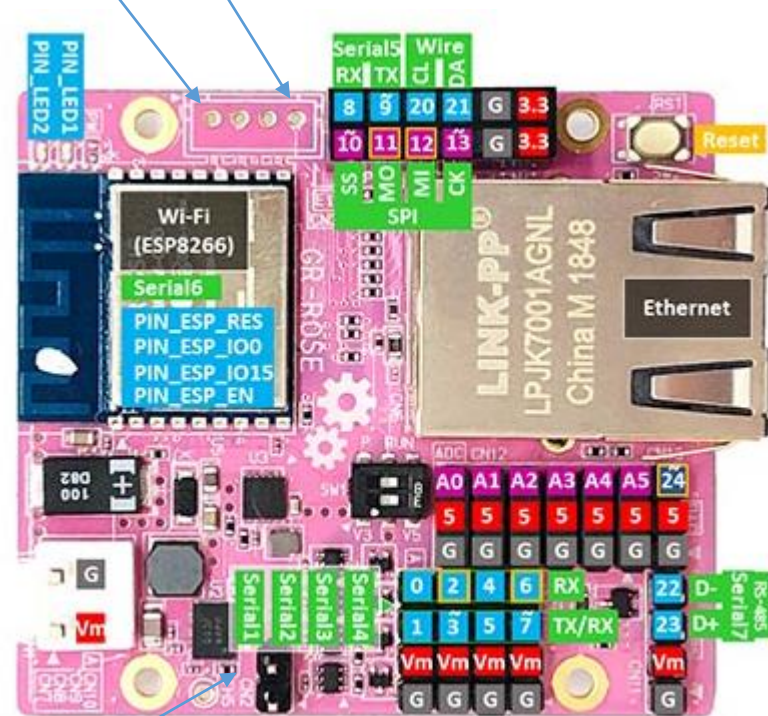
CN4_RX(8)	P30
CN4_TX(9)	P26
CN4_CL(20)	P52
CN4_DA(21)	P50
CN4_GND	GND
CN4_3V3	3.3V

CN4_SS(10)	PE4
CN4_MO(11)	PE6
CN4_MI(12)	PE7
CN4_CK(13)	PE5
CN4_GND	GND
CN4.3V3	3.3V

LED1	PA0
LED2	PA1

SER6_TX	P23
SER6_RX	P25
ESP_RES	P17
ESP_IO0	P27
ESP_IO15	P31
ESP_EN	P24

CN1_GND	GND
CN1_VM	VM



CN12C1_A5	PD7
CN12C2_A4	PD6
CN12C3_A3	PD5
CN12C4_A2	PD4
CN12C5_A1	PD3
CN12C6_A0	PD2
CN13_GND	GND
CN13_5V	5V
CN13_DA(24)	P05

CN11_D-	PC6
CN11_D+	PC7
CN11_VM	VM
CN11_GND	GND
DIR	PC5

CN2_5V	5V
CN2_VSYSA	VSYSA

CN7_TX(0)	P21
CN7_RX(1)	P20
CN7_VM	VM
CN7_GND	GND
2 wire	P22 - 0

CN8_TX(2)	P12
CN8_RX(3)	P13
CN8_VM	VM
CN8_GND	GND
2 wire	P14-0

CN9_TX(4)	PC2
CN9_RX(5)	PC3
CN9_VM	VM
CN9_GND	GND
2 wire	PC4-0

CN10_TX(6)	P33
CN10_RX(7)	P32
CN10_VM	VM
CN10_GND	GND
2 wire	P34-0

Pin Assign Information

Available Pins

Pin Name	CPU Pin	Pin Name	CPU Pin	Pin Name	CPU Pin
SER1_TX	P20	DAC	P05	ETH_MDIO	PA3
SER1_RX	P21	A0	PD2	ETH_TXEN	PB4
SER1_SEL	P22	A1	PD3	ETH_TXD0	PB5
SER2_TX	P13	A2	PD4	ETH_TXD1	PB6
SER2_RX	P12	A3	PD5	ETH_RXD0	PB1
SER2_SEL	P14	A4	PD6	ETH_RXD1	PB0
SER3_TX	PC3	A5	PD7	ETH_RXER	PB3
SER3_RX	PC2	ACC_SCL	P52	ETH_CRS	PB7
SER3_SEL	PC4	ACC_SDA	P50	ETH_CLK	PB2
SER4_TX	P32	ACC_INT	P07	LED1	PA0
SER4_RX	P33	ESP_RES	P17	LED2	PA1
SER4_SEL	P34	ESP_IO0	P27	A_WIRE_SCL	PE2
SPI_SS	PE4	ESP_IO15	P31	A_WIRE_SDA	PE1
SPI_MO	PE6	ESP_EN	P24	A_ESP_CK	PC5
SPI_MI	PE7	SER6_RX	P25	A_ESP_MI	PC7
SPI_CK	PE5	SER6_TX	P23	A_ESP_MO	PC6
SER5_RX	P30	SER7_TX	PC7	A_ESP_SS	PC4
SER5_TX	P26	SER7_RX	PC6	A_ESP_IO0	P15
WIRE_CL	P52	SER7_DIR	PC5		
WIRE_DA	P50	ETH_MDC	PA4		

Pin interrupt

Pin Name	CPU Pin	INT
SER1_TX	P20	IRQ8
SER1_RX	P21	IRQ9
SER2_TX	P13	IRQ3
SER2_RX	P12	IRQ2
SPI_MO	PE6	IRQ6
SPI_MI	PE7	IRQ7
SPI_CK	PE5	IRQ5
SER5_RX	P30	IRQ0
DAC	P05	IRQ13
A0	PD7	IRQ7
A1	PD6	IRQ6
A2	PD5	IRQ5
A3	PD4	IRQ4
A4	PD3	IRQ3
A5	PD2	IRQ2
SER7_TX	PC7	IRQ14
SER7_RX	PC6	IRQ13

PWM pin

Pin Name	CPU Pin	PWM
SER1_TX	P20	MTIOC1A
SER4_TX	P32	MTIOC0C
SPI_CK	PE5	MTIOC4C
SER7_TX	PC7	MTIOC3A
SER7_RX	PC6	MTIOC3C
LED1	PA0	MTIOC4A

Module

Pyboard pyb modules	GR Rose pyb(rxb) modules	Description
Accel	未実装	加速度センサー
ADC	ADC	AD変換
CAN	未実装	CAN (controller area network communication bus)
DAC	DAC	DA変換
ExtInt	ExtInt	I/Oピンによる外部割込み
I2C	I2C	I2C (a two-wire serial protocol)
LCD	未実装	LCD制御
LED	LED	LEDオブジェクト
Pin	Pin	I/Oピン
PinAF	未実装	ピン周辺機能
RTC	RTC	リアルタイマー
Servo	Servo	サーボ(PWM)
SPI	SPI	SPI (a master-driven serial protocol)
Switch	Switch	スイッチ
Timer	Timer	タイマー
TimerChannel	同等機能なし	タイマー向けチャンネル設定
UART	UART	シリアル通信
USB_HID	未実装	USB Human Interface Device (HID)
USB_VCP	未実装	USB仮想COMポート

サンプルコード

- ファイルアクセス
- ピン割り込み
- Timer (ソフトウェア)
- I2C – デバイススキャン
- I2C – SH30 温度湿度センサー
- I2C – OMRON 2SMPB-02E MEMS絶対圧センサ
- ネットワーク

サンプル ファイルアクセス

- デフォルトの/flashフォルダのmain.pyの内容を表示します。

```
f = open('main.py', 'r')
f.read()
f.close()
```

```
MicroPython v1.12-816-g4573864c0-dirty on 2020-05-09; GR-MANGO with RZA2M
Type "help()" for more information.
>>> f = open('main.py', 'r')
>>> f.read()
'# main.py -- put your code here!¥r¥n'
>>> f.close()
```

/flash及び/sdファイルシステムへの読み書きは、標準的には、MicroPythonのファイル入出力ライブラリを呼び出すことで行っています。

サンプルピン割り込み – スイッチ検出

- Switch(PD7)を押すと、print(“intr”)が実行されます。
- IRQxを割り当て可能なピンのみで実行できます。

```
from pyb import Pin, ExtInt
callback = lambda e: print("intr")
ext = ExtInt(Pin(Pin.cpu.PD7, Pin.IN, Pin.PULL_UP),
ExtInt.IRQ_RISING, Pin.PULL_UP, callback)
```

サンプル Timer – タイマー呼び出し

- ソフトウェアタイマ機能で、2 秒ごとに、print(2)を呼び出します。

```
from machine import Timer
tim = Timer(-1)
tim.init(period=2000, mode=Timer.PERIODIC,
callback=lambda t:print(2))
```

```
>>> from machine import Timer
>>> tim = Timer(-1)
>>> tim.init(period=2000, mode=Timer.PERIODIC, callback=lambda t:print(2))
>>> 2
2
2
2
2
2
```

サンプル I2C – SH30 温度湿度センサー

- 温度湿度を表示します。
 - sh30.pyモジュールを/flashに(PyMakr経由などで)Upload後、下記のコードを実行します。

```
from sht30 import SHT30

sensor = SHT30(scl_pin=machine.Pin.cpu.P52,
sda_pin=machine.Pin.cpu.P50, i2c_address=0x44)

temperature, humidity = sensor.measure()

print('Temperature:', temperature, '°C, RH:',
humidity, '%')
```

machine pinは、pyb.Pin としても定義できます
machine.Pin.cpu.P52 -> pyb.Pin('WIRE_CL')
machine.Pin.cpu.P50 -> pyb.Pin('WIRE_DA')

```
[1/9] Writing file blink.py (0kb)
[2/9] Writing file boot.py (0kb)
[3/9] Writing file i2c_scan.py (0kb)
[4/9] Writing file main.py (0kb)
[5/9] Writing file max7219.py (4kb)
[6/9] Writing file opt3001.py (0kb)
[7/9] Writing file opt3001_sample.py (0kb)
[8/9] Writing file sht30.py (7kb)
[9/9] Writing file sht30_sample.py (0kb)
Upload done, resetting board...
OK
MicroPython v1.12-816-g4573864c0-dirty on 2020-05-09; GR-MANGO with RZA2M
Type "help()" for more information.
>>> █
```

```
>>> Running selected lines
>>>
>> Temperature: 30.86672 °C, RH: 43.53094 %
>
```

上記はGR-MANGOの出力例

サンプル ネットワーク – HTTPアクセス

- ソケットモジュールを使用して、<http://micropython.org>にアクセスします。

```
import network
net=network.LAN()
net.ifconfig()
net.active(True)
net.ifconfig("dhcp")
net.ifconfig()
import usocket as socket
s = socket.socket()
addr = socket.getaddrinfo('micropython.org', 80)[0][-1]
s.connect(addr)
s.send(b'GET / HTTP/1.1\r\nHost:
micropython.org\r\n\r\n')
data = s.recv(1000)
s.close()
data
```

```
>>> import network
>>> net=network.Ethernet()
>>> net.ifconfig()
('0.0.0.0', '0.0.0.0', '0.0.0.0', '0.0.0.0')
>>> net.active(True)
>>> net.ifconfig("dhcp")
>>> net.ifconfig()
('192.168.0.47', '255.255.255.0', '192.168.0.1', '192.168.0.1')
>>> import usocket as socket
>>> s = socket.socket()
>>> addr = socket.getaddrinfo('micropython.org', 80)[0][-1]
>>> s.connect(addr)
>>> s.send(b'GET / HTTP/1.1\r\nHost: micropython.org\r\n\r\n')
41
>>> data = s.recv(1000)
>>> s.close()
>>> data
b'HTTP/1.1 200 OK\r\nServer: nginx/1.12.2\r\nDate: Sat, 12 Jan 2019 03:27:43 GMT
\r\nContent-Type: text/html; charset=utf-8\r\nContent-Length: 16839\r\nConnectio
n: keep-alive\r\nVary: Accept-Encoding\r\nX-Frame-Options: SAMEORIGIN\r\n\r\n<!D
OCTYPE html>\r\n\r\n<html lang="en">\r\n  <head>\r\n    <meta charset="utf-8">\r\n
<meta http-equiv="X-UA-Compatible" content="IE=edge">\r\n    <meta name="viewport"
content="width=device-width, initial-scale=1">\r\n    <!-- The above 3 meta tags
*must* come first in the head -->\r\n\r\n    <link rel="icon" href="/static/img/favi
con.ico'
```

Network-HTTPS access

- ソケットモジュールを使用して、<https://micropython.org> にアクセスします。

```
import network
net=network.LAN()
net.active(True)
net.ifconfig("dhcp")
net.ifconfig()
import socket
import ssl
s = socket.socket()
addr = socket.getaddrinfo("micropython.org", 443)[0][-1]
s.connect(addr)
ss = ssl.wrap_socket(s)
ss.write(b"GET / HTTP/1.0\r\n\r\n")
data=ss.read(4096)
s.close()
print(data)
```

```
>>> import network
>>> net=network.LAN()
>>> net.active(True)
>>> net.ifconfig("dhcp")
>>> net.ifconfig()
('192.168.0.73', '255.255.255.0', '192.168.0.1', '192.168.0.1')
>>> import socket
>>> import ssl
>>> s = socket.socket()
>>> addr = socket.getaddrinfo("micropython.org", 443)[0][-1]
>>> s.connect(addr)
>>> ss = ssl.wrap_socket(s)
>>> ss.write(b"GET / HTTP/1.0\r\n\r\n")
18
>>> data=ss.read(4096)
>>> s.close()
>>> print(data)
b'HTTP/1.1 200 OK\r\nServer: nginx/1.10.3\r\nDate: Sun, 24 May 2020 04:09:00 GMT\r\nContent-Type: text/html\r\nContent-Length: 11482\r\nLast-Modified: Fri, 20 May 2016 09:54:37 GMT\r\nConnection: close\r\nVary: Accept-Encoding\r\nETag: "573ededd-2cda"\r\nStrict-Transport-Security: max-age=15768000\r\nAccept-Ranges: bytes\r\n\r\n<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"\n"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">\n\n<html>\n\n<head>\n<title>Damien P. George</title>\n<link rel="shortcut icon" type="image/x-icon" href="/favicon.ico'
```

Network-HTTP access via ESP8266

- ESP8266用のソケットモジュールを使用して、<http://micropython.org>にアクセスします。
- WIFIの場合、インポートするモジュールはsocketではなく、wsocketとなります(独自仕様)。

```
import network
# esp = network.ESP8266()
esp = network.WIFI()
esp.connect("xxxxxx", "xxxxxxxxxx")
esp.ifconfig()

import wsocket as socket
s = socket.socket()
addr = socket.getaddrinfo('www.micropython.org',
80)[0][-1]
s.connect(addr)
s.send(b"GET / HTTP/1.0\r\n\r\n")
data=s.recv(8192)
s.close()
print(data)
```

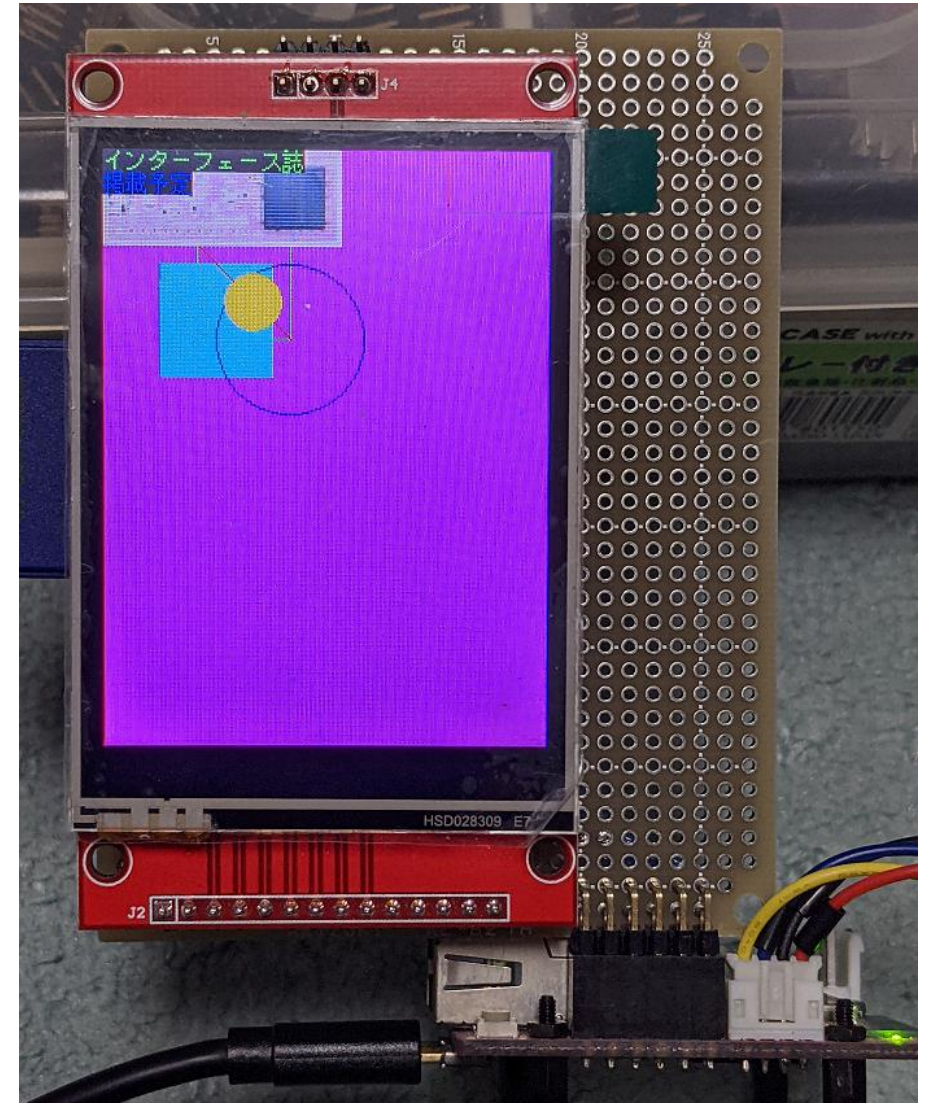
```
>>> import network
>>> esp = network.ESP8266()
AT ver=1.6.2.0(Apr 13 2018 11:10:59)
SDK ver=2.2.1(6ab97e9)
>>> esp.connect(' ', ' ')
>>> esp.ifconfig()
('192.168.0.75', '192.168.0.1', '255.255.255.0')
>>> import wsocket as socket
>>> s = socket.socket()
>>> addr = socket.getaddrinfo('www.micropython.org', 80)[0][-1]
>>> s.connect(addr)
>>> s.send(b"GET / HTTP/1.0\r\n\r\n")
18
>>> data=s.recv(8192)
>>> s.close()
>>> print(data)
b'HTTP/1.1 200 OK\r\nServer: nginx/1.10.3\r\nDate: Sun, 24 May 2020 04:00:12 GMT
\r\nContent-Type: text/html\r\nContent-Length: 54\r\nLast-Modified: Sat, 04 Oct
2014 21:54:13 GMT\r\nConnection: close\r\nVary: Accept-Encoding\r\nETag: "54306c
85-36"\r\nAccept-Ranges: bytes\r\n\r\nServer down for maintenance.\n\nPlease che
ck back soon.\n'
```


SPI LCDの表示

- GR-ROSEのPMODコネクタにSPI LCD(KMRTM24024SPI)を接続します。

```
from machine import Pin
from pyb import LCDSPI, FONT

cs=Pin("SPI_SS") # LCDのCSとPMD Pin1
dout=Pin("SPI_MO") #LCDのMOSIとPMD Pin2
din=Pin("SPI_MI") # LCDのMISOとPMD Pin3
clk=Pin("SPI_CK") # LCDのCLKとPMD Pin4
reset=Pin(Pin.cpu.P26) #LCDのRESETとPMD Pin8
rs=Pin(Pin.cpu.P30) # LCDのRSとPMD Pin7
lcd_id=LCDSPI.M_KMRTM24024SPI
lcd=LCDSPI(lcd_id=lcd_id, font_id=4, spi_id=2, baud=15000000,
cs=cs, clk=clk, dout=dout, rs=rs, reset=reset, din=din)
lcd.clear(lcd.Pink)
font=FONT(4)
print(font)
lcd.box(50, 50, 100, 100, lcd.Green)
lcd.box_fill(30, 60, 90, 120, lcd.Cyan)
lcd.line(50, 50, 100, 100, lcd.Red)
lcd.circle(100, 100, 40, lcd.Blue)
lcd.circle_fill(80, 80, 15, lcd.Yellow)
# lcd.disp_bmp_file(0, 0, "citrus16.bmp")
lcd.disp_jpeg_file(0, 0, "citrus24.jpg")
lcd.fcol(lcd.Green)
lcd.pututf8("インターフェース誌¥r¥n")
lcd.fcol(lcd.Blue)
lcd.pututf8("掲載予定¥r¥n")
```



GC Memory Information

- Display GC Memory information.

```
import micropython  
micropython.mem_info()
```

```
MicroPython v1.12-571-gfaae6b130-dirty on 2020-05-24; GR-ROSE with RX65N  
Type "help()" for more information.  
>>> import micropython  
>>> micropython.mem_info()  
stack: 960 out of 23552  
GC: total: 250368, used: 1552, free: 248816  
No. of 1-blocks: 21, 2-blocks: 8, max blk sz: 40, max free sz: 15541
```

おしまい

- 使い方の詳細は、MicroPythonのドキュメントを参照してください。
 - <https://docs.micropython.org/en/latest/>
- 移植したソースコードは以下のGithubのrenesas(昔はrx)ブランチに置かれています。
 - <https://github.com/ksekimoto/micropython>
 - `git clone https://github.com/ksekimoto/micropython -b renesas`
 - ビルド済のバイナリファイルは、release(昔はrx_release) フォルダ以下に格納しています。

その他、制限事項など。

- ビルド方法は、Github上のreadme.mdファイルに記載する予定です。
 - Boards¥GR_ROSE_DDにMBED USBドライブコピー用のビルド定義ファイル
 - Boards¥GR_ROSEフォルダがJ-linkデバッグ向けのビルド定義ファイル
- 内蔵フラッシュドライブに不整合が発生した場合には、起動時にSwitch(PinD7)を3秒押して、boot.py, main.pyを再作成してください。
- シリアル通信の処理にバグがあります。4Kバイトの受信バッファを超える通信で問題が発生する可能性があります。
- タイマーは、STM32とはMicroPythonのモジュールの実装が大幅に異なります。詳細はソースファイルをご確認ください。
- 各クラスのパラメータはオリジナルから機能を大幅に省略している場合があります。
- 各クラスのprintメソッドはオリジナルから機能を大幅に省略している場合があります。

See the book 「GR-ROSE」ではじめる電子工作 (Japanese)



<http://www.kohgakusha.co.jp/books/detail/978-4-7775-2084-8>

Backupスライド

プログラム編集、実行環境

ツール	インストール	使い方など
Visual Studio Code + PyMakr (Windows / Linux / Mac(?))	拡張メニューでPyMakrと入力してインストール。Nodejs 6.9.5以降がインストールされていること。設定方法は、Pymakr > Global settingメニューで使用するCOMポートを設定する。	メニューから、Pythonのプログラムの実行、アップロード、ダウンロードが可能
uPyCraft v.1.1 (Windows環境) 	下記リンクより、実行ファイルをダウンロードして実行する。 https://randomnerdtutorials.com/uPyCraftWindows	メニューから、Pythonのプログラムの実行、アップロード、ダウンロードが可能 https://randomnerdtutorials.com/install-upycraft-ide-windows-pc-instructions/
MU エディタ (Windows/Linux) 	Githubからソースをダウンロードして、GR_ROSEのUSBを認識するように変更 (https://github.com/ksekimoto/mu/tree/pyboard)	Microbitのエディタとして利用されている。 https://codewith.mu/