

目次

Modbus のプログラミング (pylibmodbus)	1
使用方法	1
API	2

Modbus のプログラミング (pylibmodbus)

libmodbus の Python binding “pylibmodbus”¹⁾を利用し、簡単に Modbus 機器と通信するプログラムを作ることができます。

- [libmodbus](#) - A Modbus library for Linux, Mac OS X, FreeBSD, QNX and Win32
- [pylibmodbus - Github \(original\)](#)
- [pylibmodbus - Github \(fork\)](#)

使用方法

ipython3 でインタラクティブに試すことができます。
実行例を下記に示します。

```
root@plum:~# ipython3
Python 3.4.3 (default, Oct 14 2015, 21:23:51)
Type "copyright", "credits" or "license" for more information.

IPython 1.2.1 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: import pylibmodbus

In [2]: pylibmodbus.ModbusRtu?
Type:          type
String Form:<class 'pylibmodbus.modbus_rtu.ModbusRtu'>
File:          /usr/lib/python3/dist-packages/pylibmodbus/modbus_rtu.py
Docstring:     <no docstring>
Constructor information:
  Definition:pylibmodbus.ModbusRtu(self, device='/dev/ttyS0', baud=19200,
parity='N', data_bit=8, stop_bit=1)

In [3]: rtu=pylibmodbus.ModbusRtu('/dev/ttyUSB0', baud=19200)

In [4]: rtu.connect()

In [5]: rtu.set_slave(1)

In [6]: rtu.read_input_bit(0)
Out[6]: 1
```

- [1] pylibmodbus モジュールを読み込みます。
- [2] Modbus RTU のインスタンスを初期化するコンストラクタの情報を確認しています。

- [3] Modbus RTU のインスタンスを初期化します。
- [4] 接続を行います。
- [5] Slave 機器のアドレスを設定します。
- [6] ファンクションコード2 (Read Input Status) で、アドレス 10001 を読み込みます。

API

libmodbus の API Reference は、[ここ](#)にあります。

Python の API もほぼ同じ内容となっています。一部、使いやすいようメソッドを追加しています。

Python	Arguments	libmodbus function	Note
Context			
ModbusRtu	コン ス ト ラ ク タ device="/dev/ttyS0", baud=19200, parity="N", data_bit=8, stop_bit=1	modbus_new_rtu	Create a libmodbus context for RTU
Connection			
connect		modbus_connect	Establish a connection
clonse		modbus_close	Close a connection
Common			
set_slave	slave	modbus_set_slave	Set slave ID
Client (Read data)			
read_bits	addr, nb	modbus_read_bits	Modbus function code 0x01 (read coil status)
read_bit	addr		□
read_input_bits	addr, nb	modbus_read_input_bits	Modbus function code 0x02 (read input status)
read_input_bit	addr		□
read_registers	addr, nb	modbus_read_registers	Modbus function code 0x03 (read holding registers)

Python	Arguments	libmodbus function	Note
read_input_registers	addr, nb	modbus_read_input_registers	Modbus function code 0x04 (read input registers)
Client (Write data)			
write_bit	addr, status	modbus_write_bit	Modbus function code 0x05 (force single coil)
write_register	addr, value	modbus_write_register	Modbus function code 0x06 (preset single register)
write_bits	addr, nb, data	modbus_write_bits	Modbus function code 0x0F (force multiple coils)
write_registers	addr, data	modbus_write_registers	Modbus function code 0x10 (preset multiple registers)
Client (Write and Read)			
write_and_read_registers	write_addr, data, read_addr, read_nb	modbus_write_and_read_registers	Modbus function code 0x17 (write/read registers)
Timeout			
set_response_timeout	seconds	modbus_set_response_timeout	set the timeout interval used to wait for a response
get_response_timeout		modbus_get_response_timeout	return the timeout interval used to wait for a response

1)

v2.6.6α1 からインストールしています

From:
<https://ma-tech.centurysys.jp/> - MA-X/MA-S/MA-E/IP-K Developers' WiKi

Permanent link:
https://ma-tech.centurysys.jp/doku.php?id=mae3xx_devel:modbus_programming:start

Last update: **2018/12/29 19:22**

Last
update: 2018/12/29 mae3xx_devel:modbus_programming:start https://ma-tech.centurysys.jp/doku.php?id=mae3xx_devel:modbus_programming:start
19:22
