

Modbus Master

Example

o implement the Modbus master station, press button A to trigger the reading of the hold register data of the slave station address 4 and display it on the screen

Button A wasPressed

Label label1 show

Read holding registers slave address 4 starting address 10 register qty 3 signed False timeout

Setup

Init UART id 2 TX 17 RX 16 baudrate 9600 data bits 8 stop bits 1 parity None ctrl pin None

```
from m5stack import *
from m5ui import *
from uiflow import *
from modbus.master.uModBusSerial import uModBusSerial

setScreenColor(0x222222)

title0 = M5Title(title="Master Read Holding Register", x=60, fgcolor=0xFFFFFF, bgcolor=0xff0078)
label0 = M5TextBox(16, 212, "Press A Send Data", lcd.FONT_Default, 0xFFFFFF, rotate=0)
label1 = M5TextBox(15, 69, "Text", lcd.FONT_DejaVu24, 0xe2ff00, rotate=0)

def buttonA_wasPressed():
    # global params
    label1.setText(str(modbus.read_holding_registers(4, 10, 3, False, timeout=)))
    pass
btnA.wasPressed(buttonA_wasPressed)

modbus = uModBusSerial(2, tx=17, rx=16, baudrate=9600, data_bits=8, stop_bits=1, parity=None, ctrl_pin=
```

API

Init UART id 1 TX 0 RX 0 baudrate 9600 data bits 8 stop bits 1 parity None ctrl pin None

```
modbus = uModBusSerial(1, tx=0, rx=0, baudrate=9600, data_bits=8, stop_bits=1, parity=None, ctrl_pin=No
```

o UART initialization configuration

Read coils slave address 1 starting address 1 coil qty 0 timeout 2000

```
print((str('address:') + str((modbus.read_coils(1, 1, 0, timeout=2000)))))
```

- Read coil status

Read discrete inputs slave address starting address input qty timeout

```
print((str('address:') + str((modbus.read_discrete_inputs(1, 1, 0, timeout=2000)))))
```

- Read discrete inputs

Read holding registers slave address starting address register qty signed timeout

```
print((str('address:') + str((modbus.read_holding_registers(1, 1, 0, True, timeout=2000)))))
```

- Read holding registers

Read input registers slave address starting address register qty signed timeout

```
print((str('address:') + str((modbus.read_input_registers(1, 1, 0, True, timeout=2000)))))
```

- Read input registers

Write multiple coils slave address starting address output value timeout

```
print((str('write address:') + str((modbus.write_multiple_registers(1, 1, 0, True, timeout=2000)))))
```

- Write multiple coils

Write multiple register slave address starting address register value signed timeout

```
print((str('write address:') + str((modbus.write_multiple_coils(1, 1, 0, timeout=2000)))))
```

- Write multiple registers

Write single coil slave address output address output timeout

```
print((str('write address:') + str((modbus.write_single_coil(1, 1, 0xFF00, timeout=2000)))))
```

- Write single coil

Write single register slave address register address register value signed timeout

```
print((str('write address:') + str(modbus.write_single_register(1, 1, 0, True, timeout=2000))))
```

- Write single register