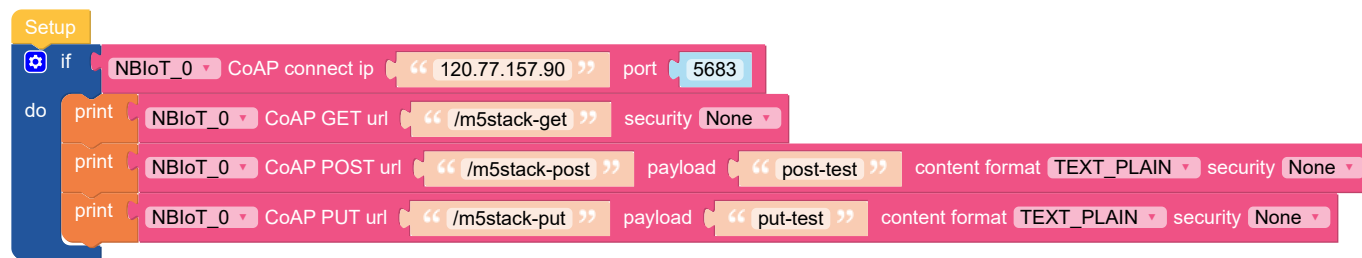


Unit NbloT

Example

CoAP test



```
from m5stack import *
from m5stack_ui import *
from uiflow import *
import unit

screen = M5Screen()
screen.clean_screen()
screen.set_screen_bg_color(0xFFFFFF)
NBloT_0 = unit.get(unit.NBIOT, unit.PORTA)

if NBloT_0.coap_connect('120.77.157.90', 5683):
    print(NBloT_0.coap_get('/m5stack-get'))
    print(NBloT_0.coap_post('/m5stack-post', 'post-test', content_format=0))
    print(NBloT_0.coap_put('/m5stack-put', 'put-test', content_format=0))
```

API

NBloT_0 Set core uart ID num 1

```
NBloT_0.uart_port_id(1)
```

◦ Configure Core UART Interface ID

NBloT_0 check GPRS network registration

```
print((str('GPRS:') + str((NBloT_0.get_gprs_network_registration()))))
```

◦ Check GPRS Network Registration Status

NBloT_0 check network registration

```
print((str('network registration:') + str((NBloT_0.get_network_registration()))))
```

- Check Network Registration Status

NBIoT_0 check single quality

```
print((str('single quality:') + str((NBIoT_0.get_single_quality()))))
```

- Check Signal Quality

NBIoT_0 check module status

```
print((str('status:') + str((NBIoT_0.check_status()))))
```

- Check Module Operating Status

NBIoT_0 CoAP connect ip " " port 5683

```
print((str('connect status:') + str((NBIoT_0.coap_connect('', 5683)))))
```

- Establish CoAP Connection to Specified IP and Port

NBIoT_0 CoAP destroy

```
print((str('destroy status:') + str((NBIoT_0.coap_destroy()))))
```

- Terminate CoAP Connection

NBIoT_0 CoAP GET url " " security None

```
print((str('get status:') + str((NBIoT_0.coap_get('')))))
```

- Initiate Unencrypted CoAP GET Request to Specified URL

NBIoT_0 CoAP POST url " " payload " " content format TEXT_PLAIN security None

```
print((str('post status:') + str((NBIoT_0.coap_post('', '', content_format=0)))))
```

- Send Plain Text POST Request to Specified URL

NBIoT_0 CoAP PUT url " " payload " " content format TEXT_PLAIN security None

```
print((str('put status:') + str((NBIoT_0.coap_put('', '', content_format=0)))))
```

- Send Plain Text PUT Request to Specified URL

NBIoT_0 MQTT check connection

```
print((str('is connection:') + str((NBIoT_0.mqtt_check_connection()))))
```

- Check MQTT Connection Status

NBIoT_0 MQTT connect server " " port 1883 client id " " username " " password " " keepalive 0

```
NBIoT_0.mqtt_connect('', 1883, '', '', '', 0)
```

- Configure MQTT Server Parameters (Address, Port, Client ID, Username, Password, Keepalive)
- server
- port
- client id
- username
- passwd
- keepalive

NBIoT_0 MQTT disconnect

```
NBIoT_0.mqtt_disconnect()
```

- Disconnect MQTT Connection

NBIoT_0 MQTT poll downlink message

```
NBIoT_0.mqtt_poll()
```

- Poll for Incoming MQTT Messages

NBIoT_0 MQTT publish topic " " payload " " QoS 0

```
NBIoT_0.mqtt_publish('', '', 0)
```

- Publish MQTT Message to Subscribed Topic
- topic: MQTT topic
- payload: Message content
- QoS: Quality of Service Level

NBiot_0 MQTT subscribe topic “ ” QoS 0

```
print((str('MQTT subscribe:') + str((NBiot_0.mqtt_subscribe('', unit_nbiot_mqtt_cb, 0)))))
```

- Subscribe to Topic with QoS Level

MQTT subscribe callback topic nb_topic msg nb_msg

```
def unit_nbiot_mqtt_cb(nb_mq_topic, nb_mq_payload):  
    global nb_topic, nb_msg  
    nb_topic = nb_mq_topic  
    nb_msg = nb_mq_payload  
    pass
```

- Set Callback Function for Subscribed Topic

NBiot_0 MQTT unsubscribe topic “ ”

```
NBiot_0.mqtt_unsubscribe('')
```

- Unsubscribe from Topic

NBiot_0 power down module

```
NBiot_0.poweroff()
```

- Power Off Module

NBiot_0 reset module

```
NBiot_0.reset()
```

- Reset Module

NBiot_0 Set command echo mode OFF

```
print((str('echo module:') + str((NBiot_0.set_command_echo_mode(0)))))
```

- Enable/Disable AT Command Echo