

คู่มือการทำงาน 3G module

1. คำนำ

คู่มือนี้เป็นการอธิบายการทำงาน 3G module (UC20-G) เบื้องต้น โดยเป็นการส่งข้อมูลผ่าน TCP protocol โดยสั่งงานและสื่อสารกับ Serial port ผ่าน Arduino Uno โดยในที่นี่เป็นเนื้อหาเกี่ยวกับคำสั่งที่ใช้สั่งการ Arduino และการสร้าง network flow ของ Node-red เท่านั้นส่วนการใช้งานอื่นๆของ 3G module จะทำการแนบลิ้งไว้ที่หน้าสุดท้าย

2. การใช้งาน

```
#include "TEE_UC20.h"  
#include "SoftwareSerial.h"  
#include <AltSoftSerial.h>  
#include "internet.h"  
#include "tcp.h"
```

- ทำการ include library ตามรูป “TEE_UC20.h” คือ library ที่เอาไว้สื่อสารระหว่าง UC20 กับ Uno ผู้ใช้สามารถกำหนดได้ว่าจะให้สื่อสารผ่าน Hardware/ Software Serial โดยเลือกใช้ library SoftwareSerial.h / AltSoftSerial “internet.h” ไว้เพื่อเชื่อมต่อ internet “tcp.h” เอาไว้รับส่งข้อมูลผ่าน TCP protocol

```
INTERNET net;  
TCP tcp;  
int cnt;  
String nodeName = "3G Module-test";  
String payload;
```

- ประกาศตัวแปรโดย cnt เอาไว้นับจำนวนครั้งที่ส่งข้อมูลไปยัง server ส่วน payload เอาไว้รวมข้อมูลที่จะส่งขึ้น server ในรูปแบบ String

```
const long interval = 1800000; //  
unsigned long previousMillis = 0;
```

- Interval เอาไว้กำหนดว่าครั้งต่อไปใช้เวลาเท่าไรมีหน่วยเป็น ms (millisecond)

```
#define APN "internet"  
#define USER ""  
#define PASS ""
```

- APN USER และ PASS ขึ้นอยู่กับแต่ละผู้ให้บริการเครือข่าย

```
AltSoftSerial mySerial;  
  
void setup()  
{  
  Serial.begin(9600);  
  gsm.begin(&mySerial, 9600);  
  gsm.Event_debug = debug;  
  cnt = 0;  
  Serial.println(F("UC20"));  
  gsm.PowerOn();  
  while (gsm.WaitReady()) {}  
  
  Serial.print(F("GetOperator --> "));  
  Serial.println(gsm.GetOperator());  
  Serial.print(F("SignalQuality --> "));  
  Serial.println(gsm.SignalQuality());  
}
```

- Serial ตั้งค่าไว้ที่ 9600 Hz “gsm.begin” ทำงานเหมือน Serial.begin แต่เป็นของ 3G Module “gsm.PowerOn()” เป็นการสั่งเปิด module “GetOperator” คือการแสดงค่าว่าเชื่อมต่อกับเครือข่ายอะไร “SignalQuality” คือการสั่งแสดงค่าความแรงของสัญญาณคลื่น

```

Serial.println(F("Disconnect net"));
net.DisConnect();
Serial.println(F("Set APN and Password"));
net.Configure(APN, USER, PASS);
Serial.println(F("Connect net"));
net.Connect();
Serial.println(F("Show My IP"));
Serial.println(net.GetIP());
Serial.print(F("GetOperator --> "));
Serial.println(gsm.GetOperator());
Serial.print(F("SignalQuality --> "));
Serial.println(gsm.SignalQuality());
Serial.println(tcp.CheckConnection());
open_tcp();
}

```

- “DisConnect” เป็นการสั่งตัดการเชื่อมต่อกับ internet “Configure” คือการเซตค่าการเชื่อมต่อใหม่ “Connect” เป็นสั่งให้เชื่อมต่อ internet โดยการใช้ค่าในคำสั่ง Configure “open_tcp” เป็นการสั่งให้เป็น port การทำงานของ tcp

```

void loop()
{
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval)
    {
        send_tcp();
        previousMillis = currentMillis;
    }

    if (tcp.ReceiveAvailable() == true) {
        read_tcp();
    }

    delay(200);
}

```

- “send_tcp” คือการส่งข้อมูลไปยัง port ที่สั่งเปิดโดย “open_tcp” “read_tcp” เป็นการอ่านข้อมูลในกรณีที่ server ส่งข้อมูลกลับมา

```

void open_tcp()
{
    Serial.println();
    Serial.println(F("Connect Server"));
    bool ret = tcp.Open("139.59.242.154", "5007");
}

```

- “tcp.Open” เป็นการสั่งเปิด port ที่จะส่งไปยัง URL ที่กำหนด

```

void send_tcp()
{
    if (tcp.StartSend())
    {
        cnt++;
        payload = String(nodeName + ", "
                        + cnt);
        Serial.println("TCP sent");
        tcp.print(payload);
        tcp.StopSend();
        Serial.println("Stop");
        // Serial.println("TCP Close");
        // tcp.Close();
    }
}

```

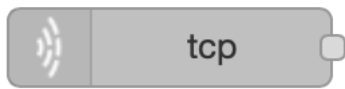
- “tcp.print” เป็นคำสั่งส่งข้อมูลไปยัง server “tcp.StopSend()” เป็นการสั่งให้ module ลสิ้นสุดการส่งข้อมูลหลังจากที่ส่งทุกข้อมูลครบแล้วในแต่ละครั้ง

```

void read_tcp() {
    int len = tcp.ReadBuffer();
    Serial.println(len);
    while (len)
    {
        if (gsm.available())
        {
            Serial.write(gsm.read());
            len--;
        }
    }
    Serial.println();
    Serial.println("Finish");
}

```

3. ด้าน Server (Node-red)



-สร้าง input node tcp



- ตั้งค่าให้รอข้อมูลจากทาง port 5007



- สร้าง function node เพื่อแปลงข้อมูลที่รับมาให้เป็น Json

```
let str = msg.payload
let data = str.slice(0).split(',');

if (data.length == 2){
    msg.payload = {
        Name : data[0],
        Count : parseInt(data[1])
    }
}

return msg;
```

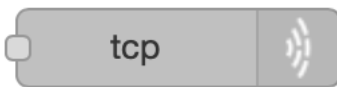
- ในการส่งข้อความมาที่ server จาก arduino ได้ส่งมาสองอย่างจึงต้องแยกออกจากกันโดยคำสั่ง slice และ split



- สร้าง function node เพื่อสร้าง string แล้วส่งกลับไปให้ arduino

```
let str = msg.payload
buf = str.Count
msg.payload = "Send back " + buf.toString()
return msg;
```

- กำหนดค่าในฟังก์ชันตามรูป



- สร้าง output node tcp เพื่อส่งข้อความกลับไปให้ arduino

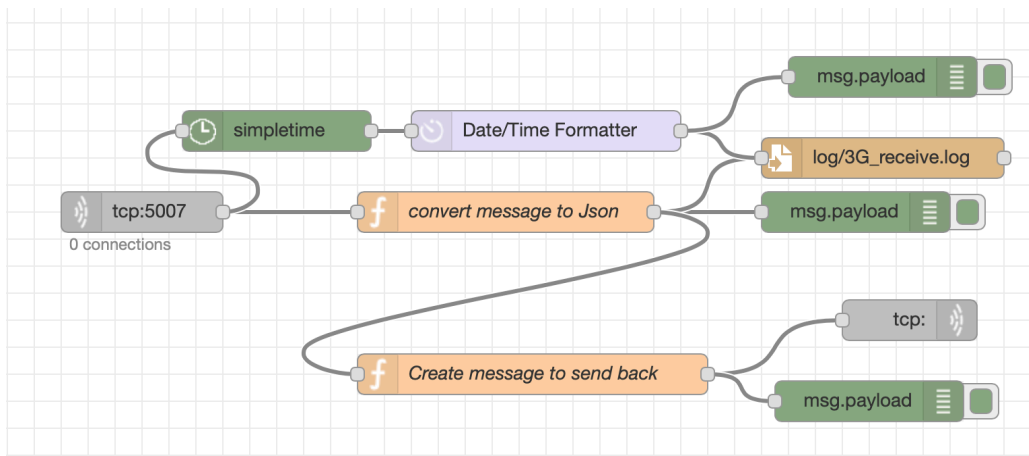
⦿ Type

Reply to TCP

- คลิกที่ node แล้วตั้งค่าเป็น Reply to TCP



- สร้าง file node เพื่อบันทึกข้อมูลทุกครั้งที่มีการรับข้อมูล



- หน้าตาเมื่อเชื่อมทุก node เข้าด้วยกัน

4. Output ที่ควรออกมาที่ Serial monitor เมื่อรัน โปรแกรม

```
UC20
Power OFF
Power ON

+CFUN: 1

+CPIN: READY

+QUSIM: 1

+QIND: SMS DONE

+QIND: PB DONE

UC20 Ready...

Close Echo
OK
```

- ข้อมูลที่แสดงที่ Serial monitor เมื่อคำสั่งตั้งแต่เริ่มใช้งานจนถึง gsm.PowerOn()

```
GetOperator --> "AIS",6
SignalQuality --> 99
Disconnect net
OK
Set APN and Password
OK
Connect net
OK
Show My IP
OK
10.6.152.239
GetOperator --> "AIS",6
SignalQuality --> 24

OK

OK
0

Connect Server

OK

+QIOPEN: 0,0
Connect Server Success
send raedy
```

- ข้อมูลที่แสดงที่ Serial monitor เมื่อรันคำสั่งตั้งแต่ gsm.GetOperator จนถึง open_tcp

```
TCP sent
Stop
0

Finish
11
Send back 1
Finish
```

- ข้อมูลที่แสดงที่ Serial monitor เมื่อส่งข้อมูลไปที่ server แล้ว serverตอบกลับมา เป็น ความยาวของ String และ ข้อความใน String

5. Reference

https://www.thaieasyelec.com/downloads/ETEE059/Development_Guide_for_3G_Shield_and_3G_Module_using_Arduino_TH_20160726.pdf

https://www.thaieasyelec.com/downloads/ETEE059/ETEE059_3G_Shield_User_Manual_TH_20160321.pdf