5G Networks Meetup

1st Edition

How to make a cellular IoT Device?

Tour of the Ericsson 5G Campus

Network, drinks & Pizza

31

Date: 29-10-2019
Time: 17.50u
Location: Ericssonstraat 2, Rijen

How to make a cellular IoT device?

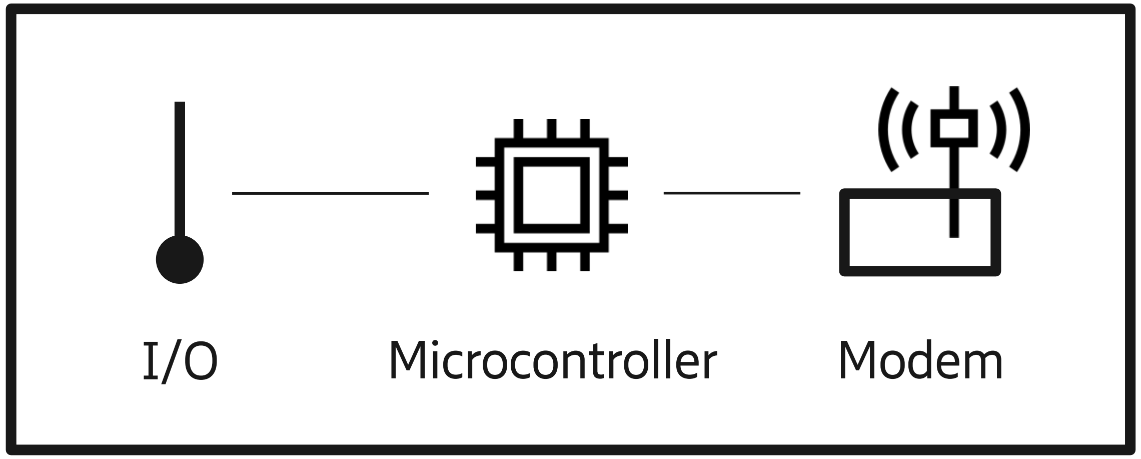
- Agenda
  - Introduction to IoT devices
  - HW setup of DHT11 sensor, ESP32 microcontroller
  - SIM7000E 4G modem module
  - "Semi live" coding

IoT

Internet of things

What is an IoT device?

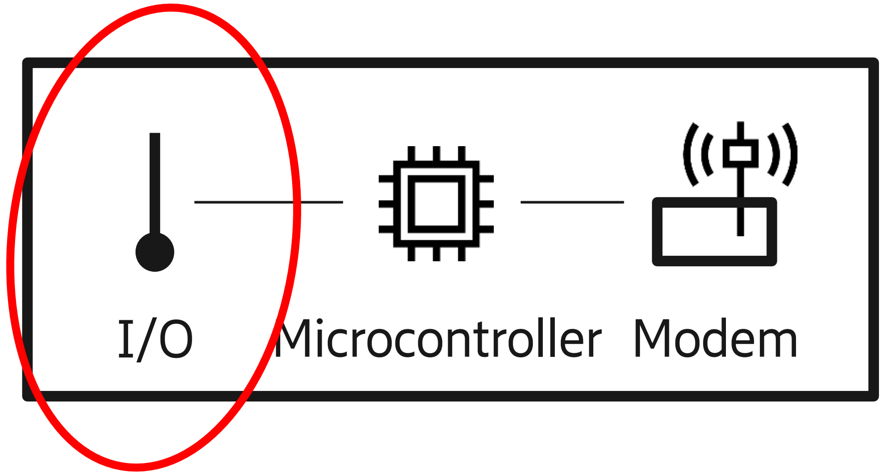
- A "thing" connected to Internet (and thus able to send or receive data)
- Consist of 3 parts (typically)
  - Input or Output device (e.g. sensor or actuator)
  - Microcontroller
  - Modem



Input or Output device

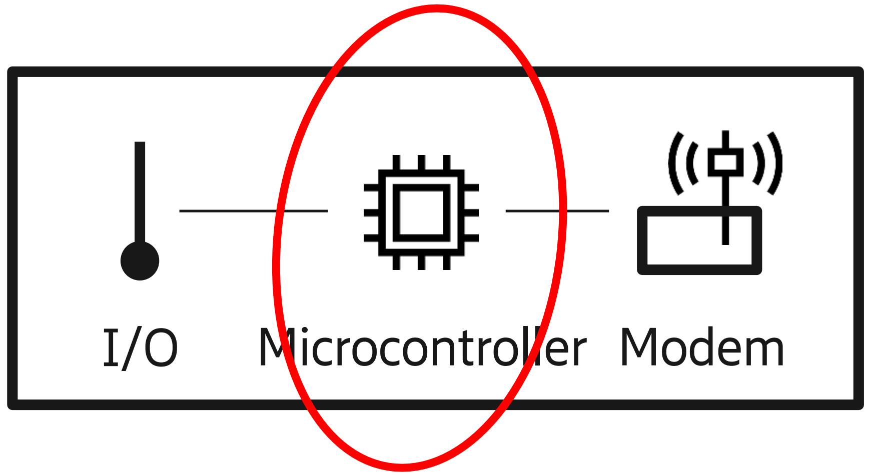
- Input sensors, can be anything...
  - Environmental (Temperature, Humidity, Light, Sound)
  - Location (GPS)
  - Motion (Acceleration, Gyroscope)
  - And many, many more...

- Output devices, can be anything...
  - Environmental (Heater, Humidifier)
  - Motion (Motor, Linear actuator)
  - Relays (Light switch)
  - And many, many more...



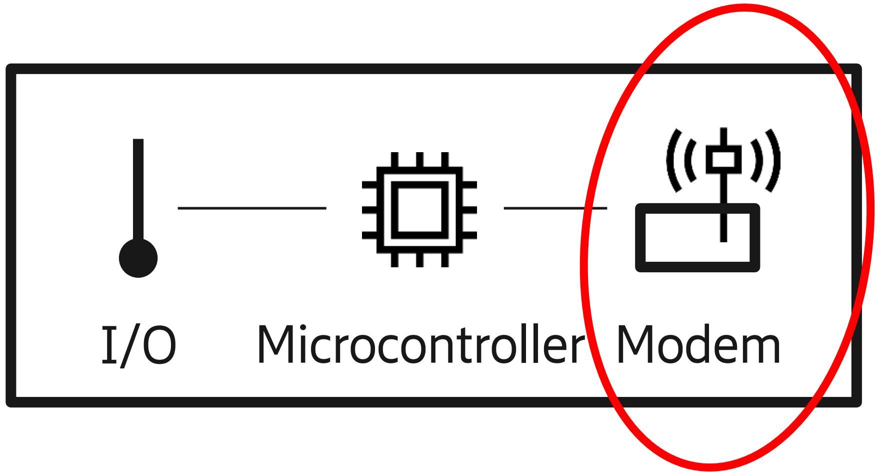
Micro controllers

- Are the brains of the IoT device
- Are programmable
- Consist of:
  - CPU
  - Memory
  - Peripherals



Modem

- A IoT device need some form of communication towards internet
  - Short range
    - Ethernet
    - Bluethooth (BLE)
    - Wifi
    - Zigbee
    - IR
  - Long range
    - Lora
    - Mobile/Cellular (2G/3G/4G/5G...)



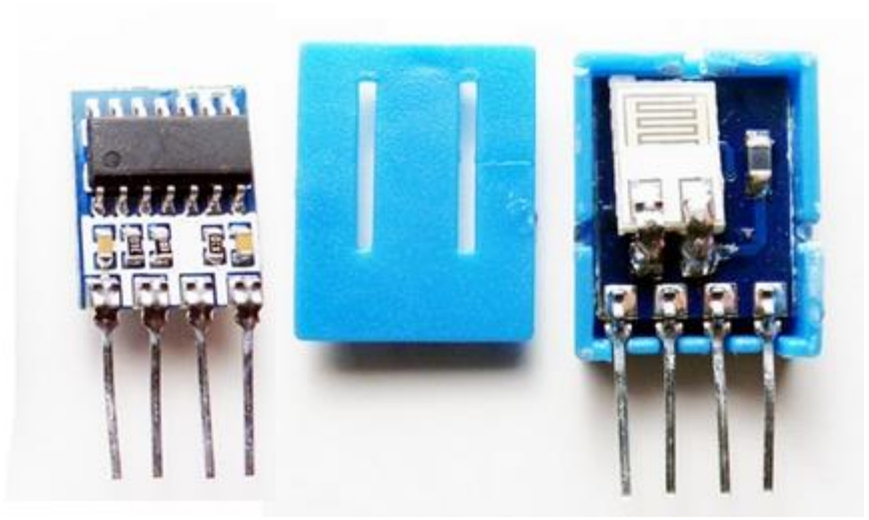
HW used in today's example

DHT11
Temperature /Humidity
sensor

ESP32
Microcontroller

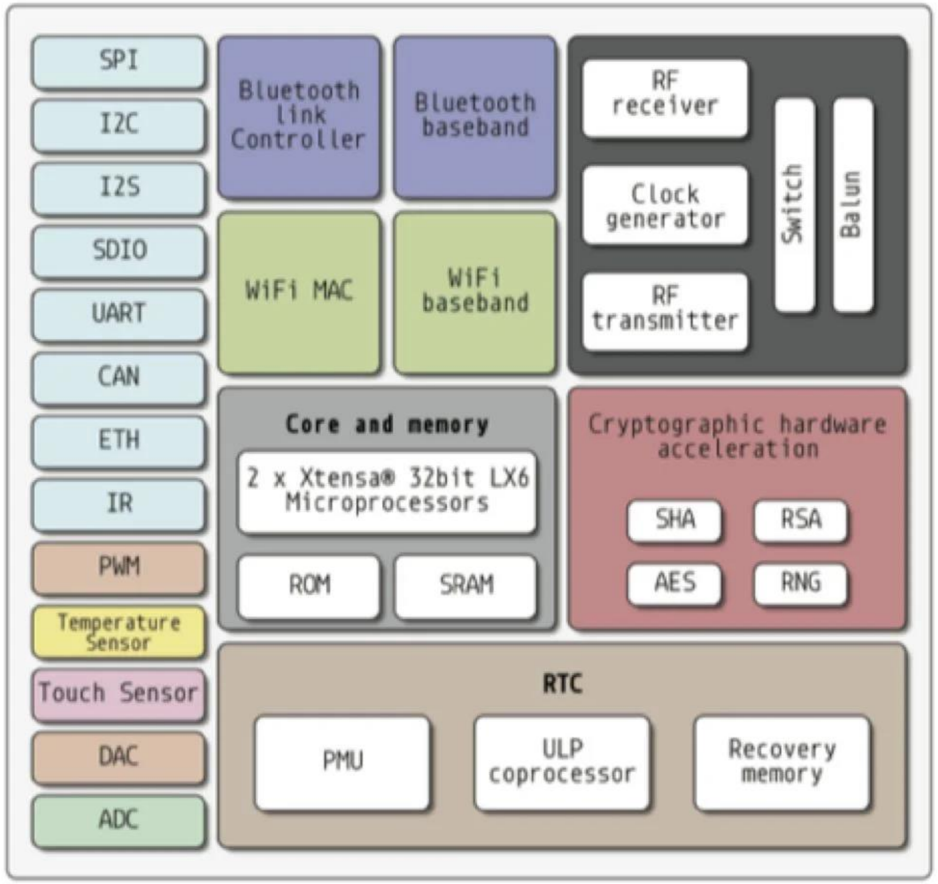
SIM7000E
4G Modem

DHT11 Sensor



- Temperature range 0 - 50°C (+/- 2°)
- Humidity range 20 - 80% (+/- 5%)
- Sample frequency 1Hz
- Supply voltage 3-5v
- Supply current 2,5mA

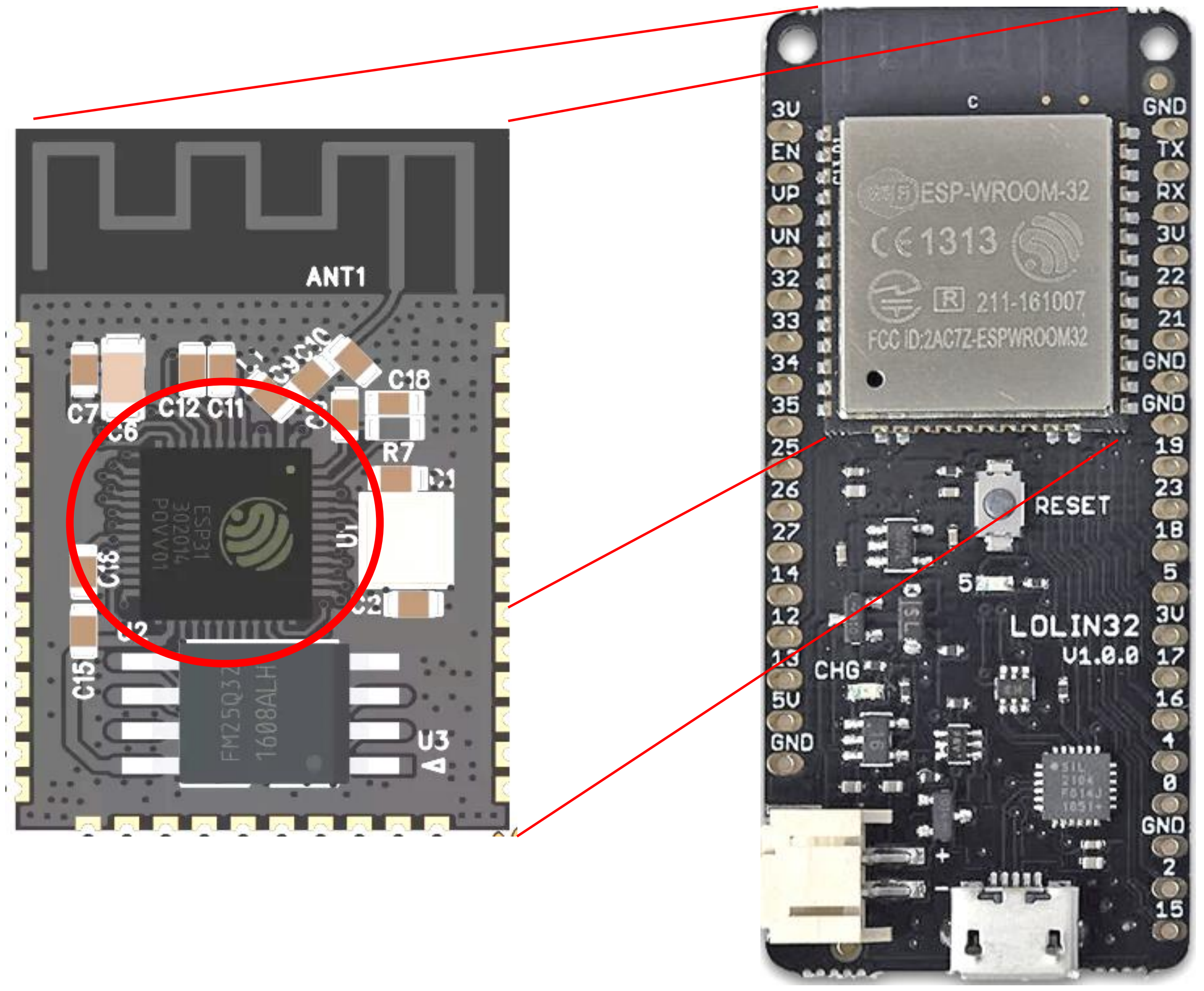
ESP32 microcontroller



— 2 × 32bit CPU's running on 160Mhz
— 448 KB ROM / 520 KB SRAM
— WIFI
— Bluetooth
— 34 × programmable GPIOs
— 12-bit SAR ADC up to 18 channels
— 2 × 8-bit DAC
— 10 × touch sensors
— 2 × I²C
— 3 × UART
...and many more

Supported by Arduino framework

ESP32 chip is inside a module (WROOM-32)...
and put onto a development board (lolin32)...



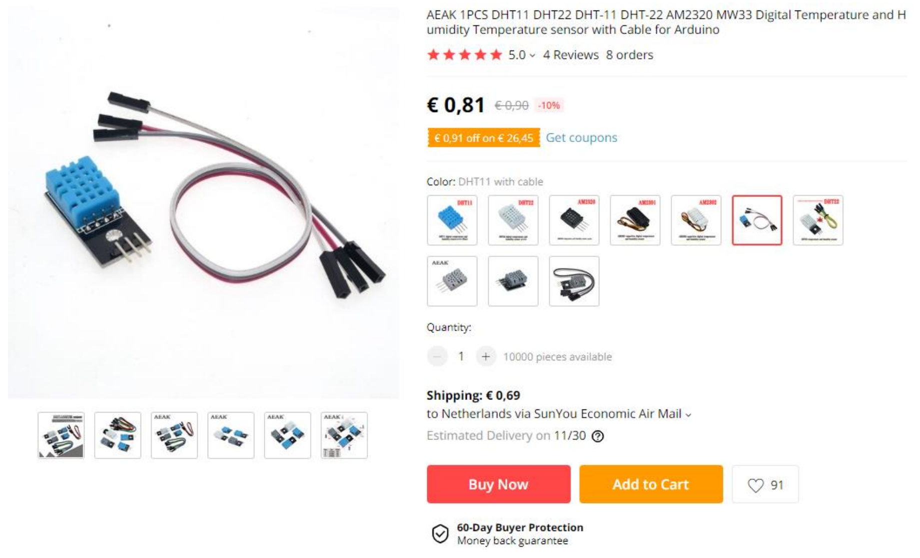
SIM7000E

- E version for european bands FDD-LTE B3/B8/B20/B28
- GPRS/EDGE 900/1800
- LTE CATM1: 375kbpsD DL/ 300kbps UL
- LTE NB-IoT: 66kbps DL / 34kbps UL
- EDGE: 237kbps DL/ 237 kbps UL
- GPRS: 86kbps DL / 86kbps UL
- GNSS (GPS) receiver
- SMS
- Low power (up to 7uA)
- TCP/IP, UDP, HTTP, MQTT, FTP
- AT command controlled

...and many more

SIM7000E modem is put on a development board (BK-7000)...

Where to buy?



AEAK 1PCS DHT11 DHT22 DHT-11 DHT-22 AM2320 MW33 Digital Temperature and Humidity Temperature sensor with Cable for Arduino

★★★★★ 5.0 4 Reviews 8 orders

€ 0,81 ~~€ 0,90~~ -10%

€ 0,91 off on € 26,45 Get coupons

Color: DHT11 with cable

Quantity:

1 10000 pieces available

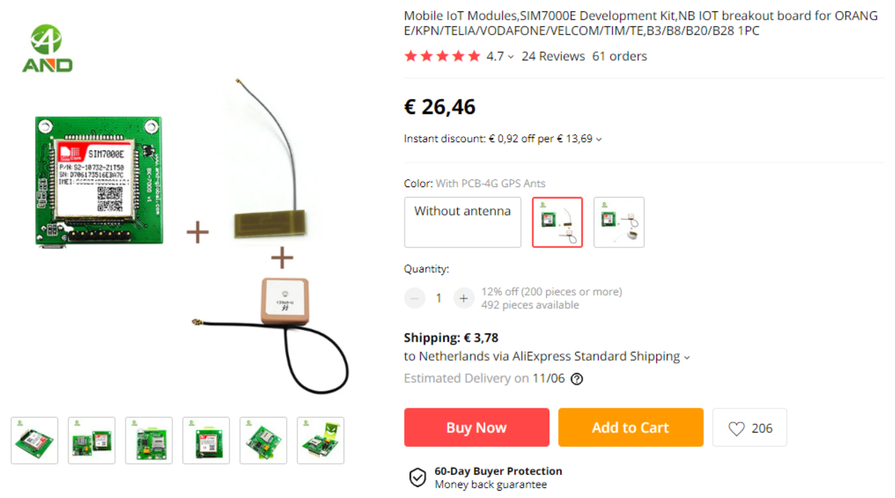
Shipping: € 0,69

to Netherlands via SunYou Economic Air Mail

Estimated Delivery on 11/30

60-Day Buyer Protection

Money back guarantee



Mobile IoT Modules,SIM7000E Development Kit,NB IOT breakout board for ORANGE/KPN/TELIA/VODAFONE/VELCOM/TIM/TE,B3/B8/B20/B28 1PC

★★★★★ 4.7 24 Reviews 61 orders

€ 26,46

Instant discount: € 0,92 off per € 13,69

Color: With PCB-4G GPS Ants

Without antenna

Quantity:

1 12% off (200 pieces or more) 492 pieces available

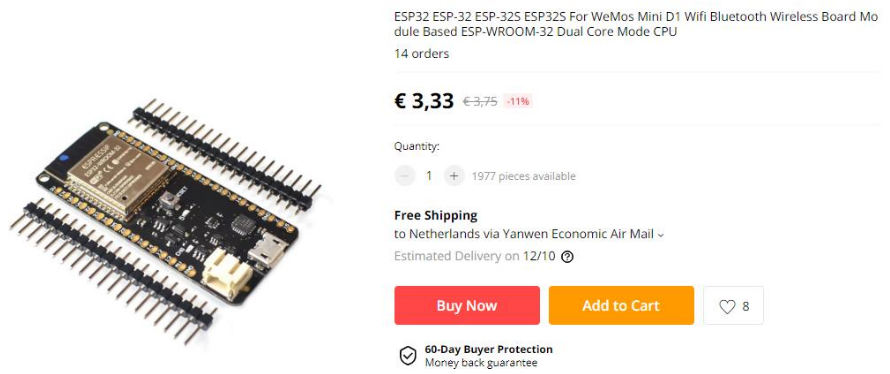
Shipping: € 3,78

to Netherlands via AliExpress Standard Shipping

Estimated Delivery on 11/06

60-Day Buyer Protection

Money back guarantee



ESP32 ESP-32 ESP-32S ESP32S For WeMos Mini D1 Wifi Bluetooth Wireless Board Module Based ESP-WROOM-32 Dual Core Mode CPU

14 orders

€ 3,33 ~~€ 3,75~~ -11%

Quantity:

1 1977 pieces available

Free Shipping

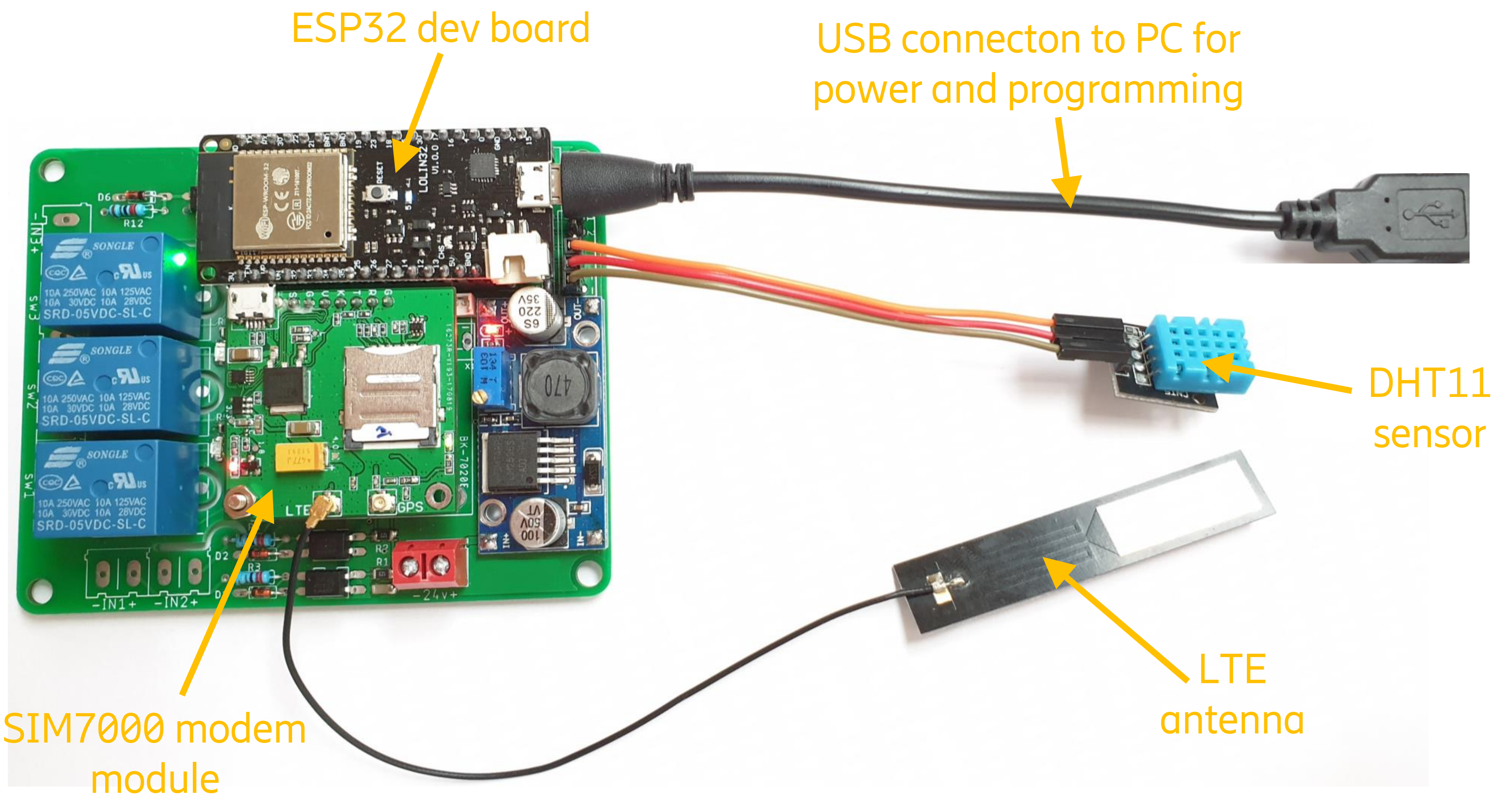
to Netherlands via Yanwen Economic Air Mail

Estimated Delivery on 12/10

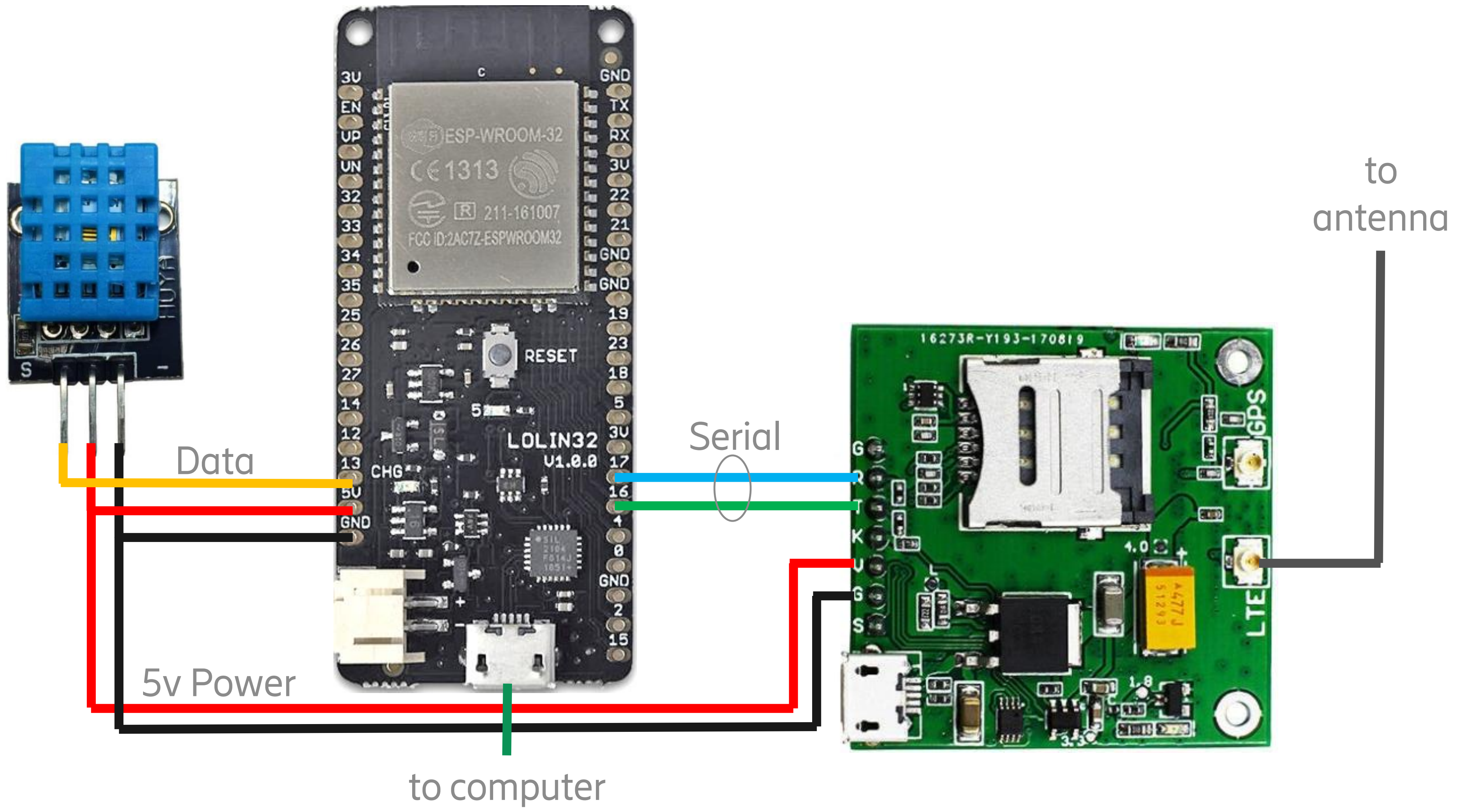
60-Day Buyer Protection

Money back guarantee

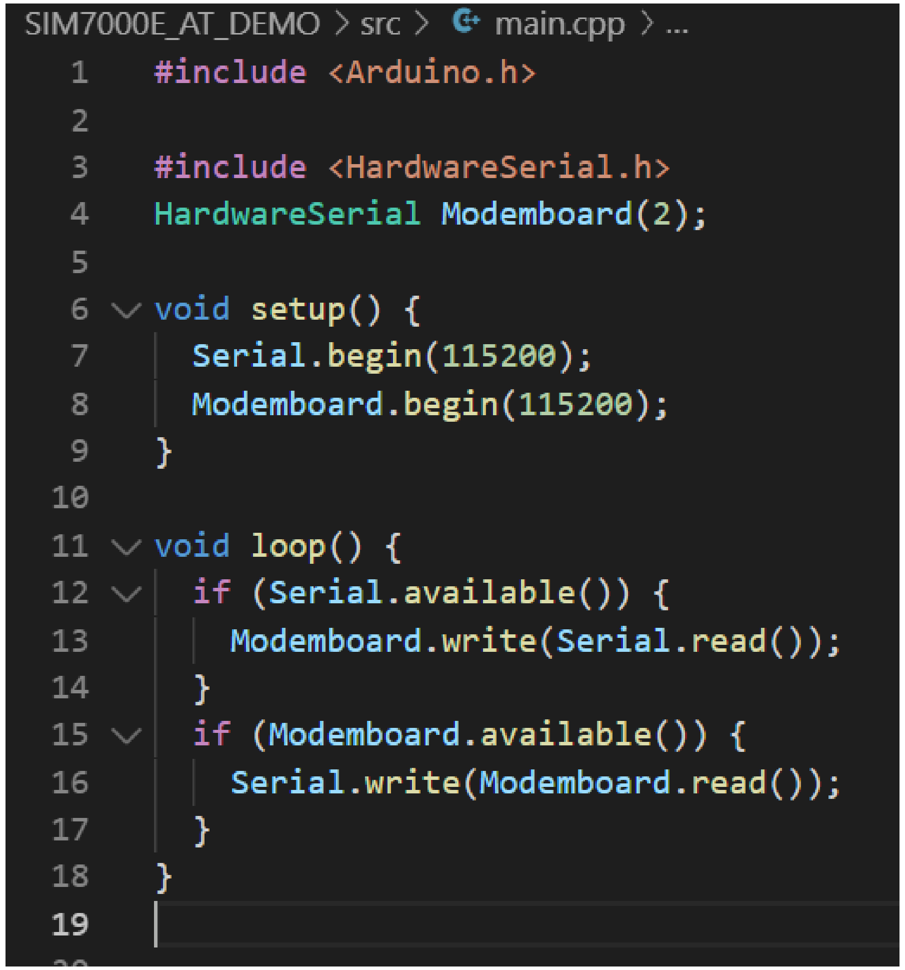
Physical HW setup...



Schematic HW setup...



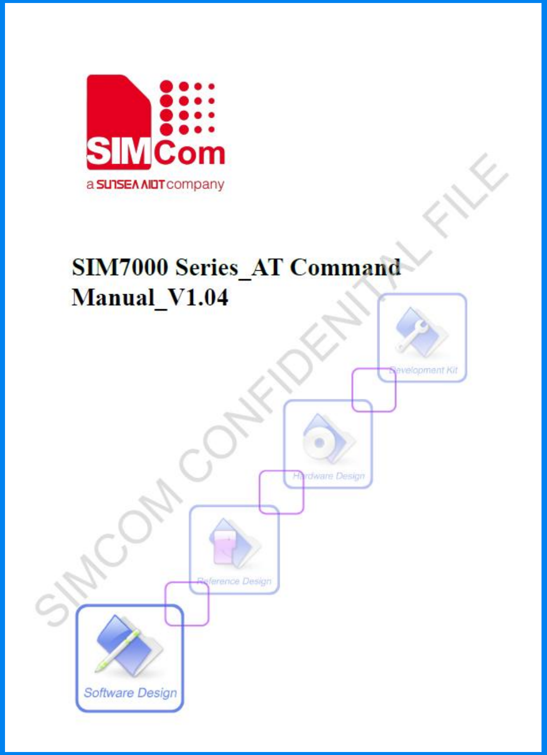
Moving to Visual Studio Code...



SIM7000E_AT_DEMO > src > main.cpp > ...

```cpp
#include <Arduino.h>

#include <HardwareSerial.h>
HardwareSerial Modemboard(2);

void setup() {
  Serial.begin(115200);
  Modemboard.begin(115200);
}

void loop() {
  if (Serial.available()) {
    Modemboard.write(Serial.read());
  }
  if (Modemboard.available()) {
    Serial.write(Modemboard.read());
  }
}

```

Use the following documents as reference:



SIMCom
a SUNSEA AIOT company

SIM7000 Series_AT Command Manual_V1.04

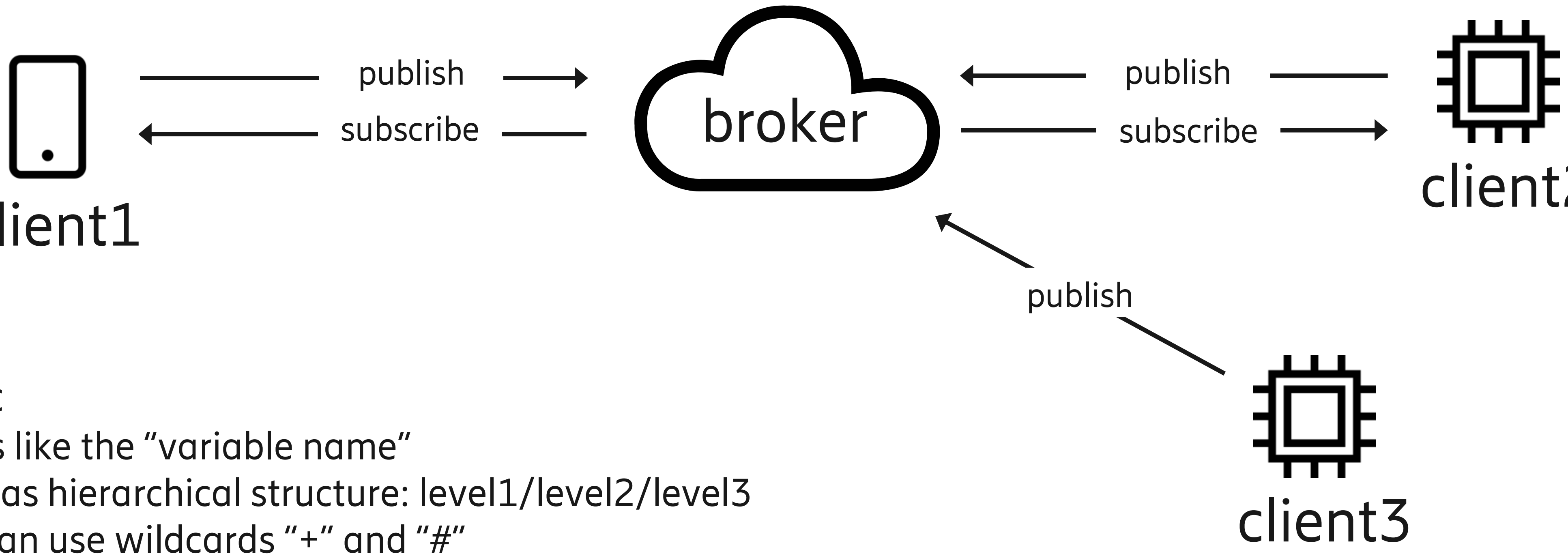
SIM7000 Series_MQTT_Application Note

Version:1.01
Release Date:January 23, 2019

Simplify Communication.
www.simcom.com

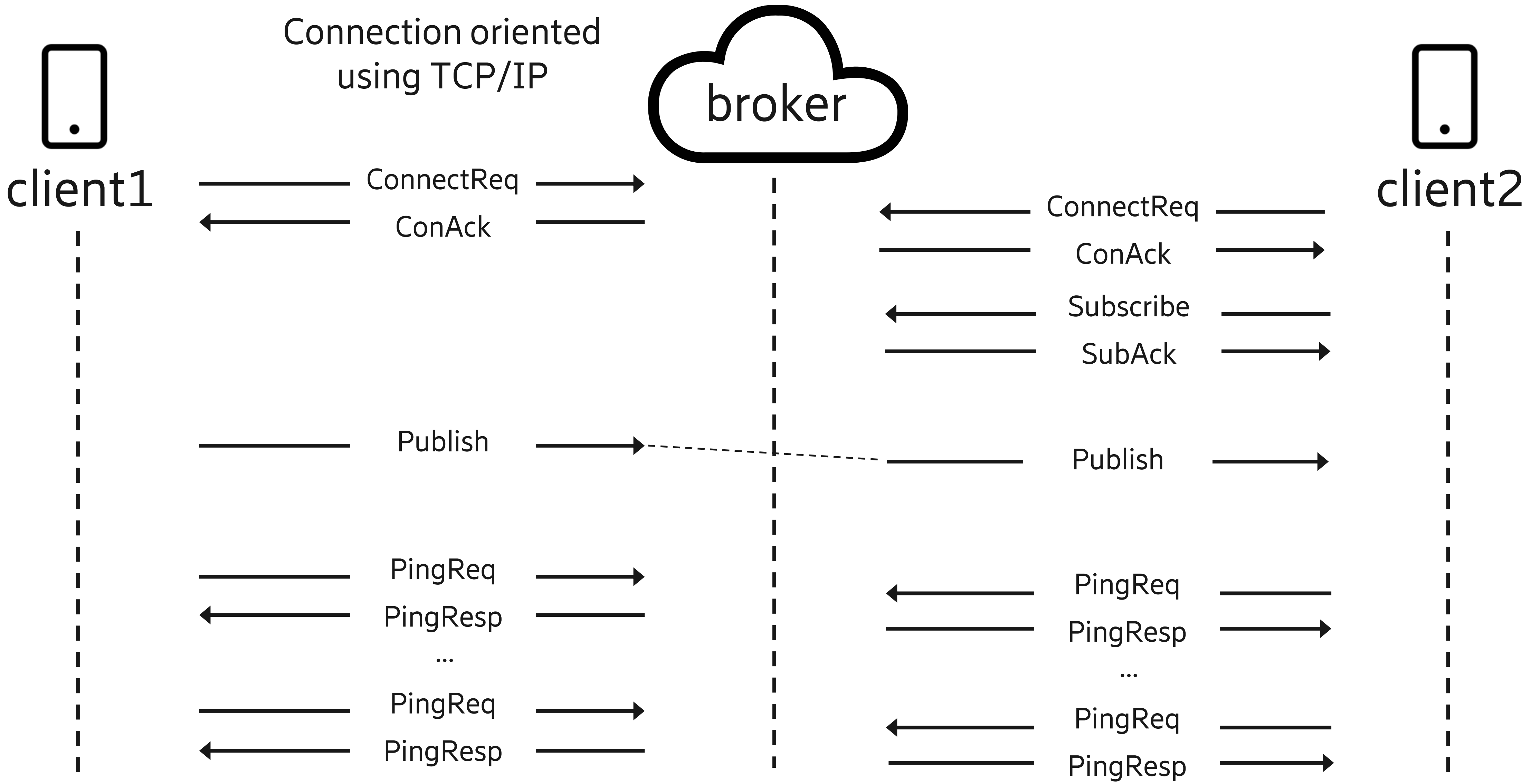
MQTT -1

- Central server is the MQTT broker
- Devices are MQTT Clients

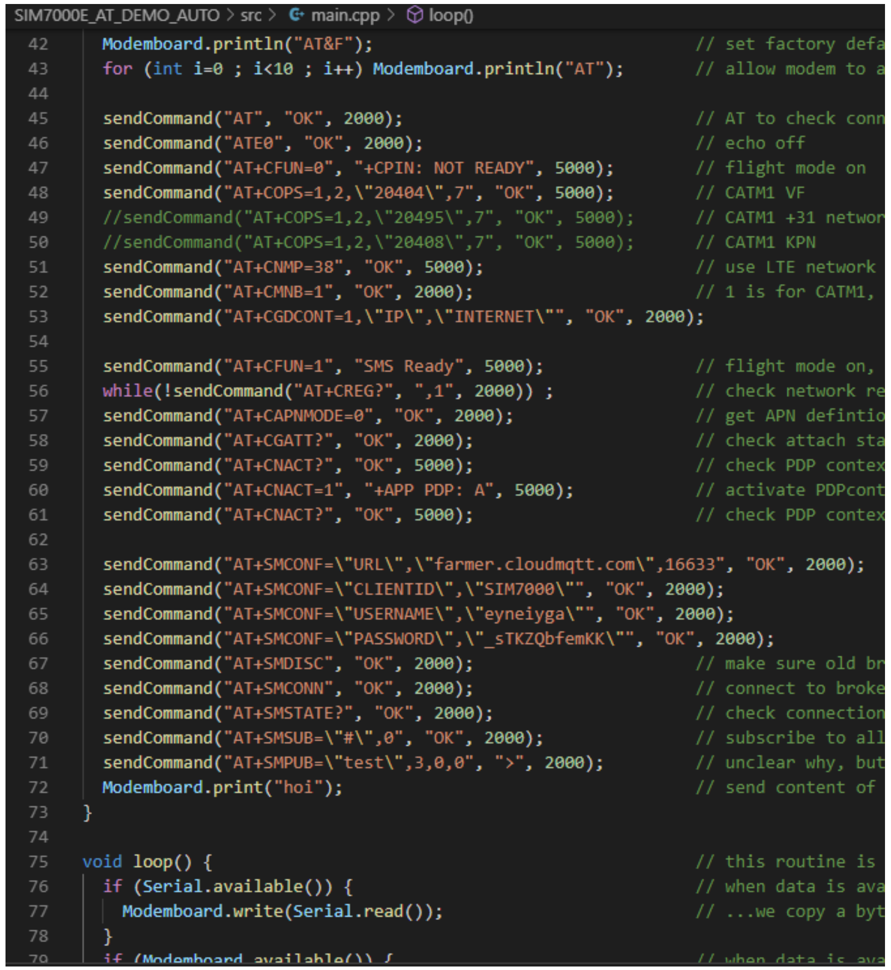


- Topic
  - is like the “variable name”
  - has hierarchical structure: level1/level2/level3
  - can use wildcards “+” and “#”
- Message
  - is like the “variable value”

MQTT -2


Connection oriented using TCP/IP
broker
client1
client2
ConnectReq
ConAck
ConnectReq
ConAck
Subscribe
SubAck
Publish
Publish
PingReq
PingResp
...
PingReq
PingResp
PingReq
PingResp
...
PingReq
PingResp

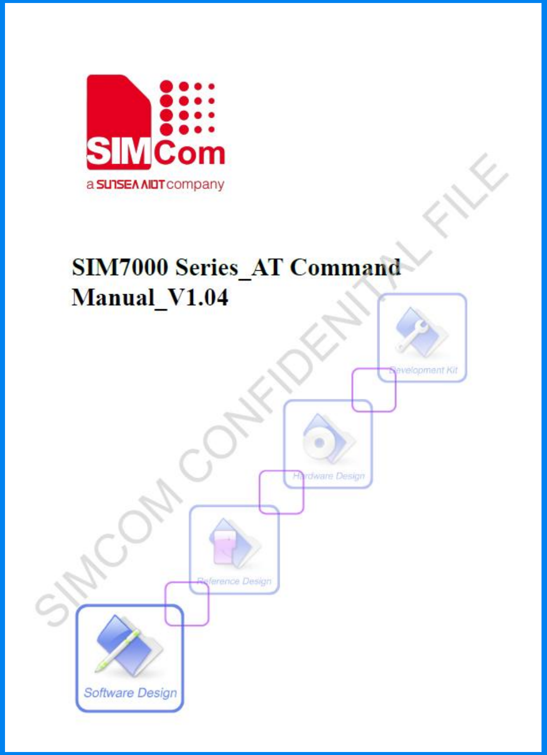
Moving to Visual Studio Code...



SIM7000E_AT_DEMO_AUTO > src > main.cpp > loop()

```
  Modemboard.println("AT&F");                                 // set factory defa
  for (int i=0 ; i<10 ; i++) Modemboard.println("AT");        // allow modem to a

  sendCommand("AT", "OK", 2000);                              // AT to check conn
  sendCommand("ATE0", "OK", 2000);                            // echo off
  sendCommand("AT+CFUN=0", "+CPIN: NOT READY", 5000);         // flight mode on
  sendCommand("AT+COPS=1,2,\"20404\",7", "OK", 5000);         // CATM1 VF
  //sendCommand("AT+COPS=1,2,\"20495\",7", "OK", 5000);       // CATM1 +31 networ
  //sendCommand("AT+COPS=1,2,\"20408\",7", "OK", 5000);       // CATM1 KPN
  sendCommand("AT+CNMP=38", "OK", 5000);                      // use LTE network
  sendCommand("AT+CMNB=1", "OK", 2000);                       // 1 is for CATM1,
  sendCommand("AT+CGDCONT=1,\"IP\",\"INTERNET\"", "OK", 2000);

  sendCommand("AT+CFUN=1", "SMS Ready", 5000);                // flight mode on,
  while(!sendCommand("AT+CREG?", ",1", 2000)) ;               // check network re
  sendCommand("AT+CAPNMODE=0", "OK", 2000);                   // get APN defintio
  sendCommand("AT+CGATT?", "OK", 2000);                       // check attach sta
  sendCommand("AT+CNACT?", "OK", 5000);                       // check PDP contex
  sendCommand("AT+CNACT=1", "+APP PDP: A", 5000);             // activate PDPcont
  sendCommand("AT+CNACT?", "OK", 5000);                       // check PDP contex

  sendCommand("AT+SMCONF=\"URL\",\"farmer.cloudmqtt.com\",16633", "OK", 2000);
  sendCommand("AT+SMCONF=\"CLIENTID\",\"SIM7000\"", "OK", 2000);
  sendCommand("AT+SMCONF=\"USERNAME\",\"eyneiyga\"", "OK", 2000);
  sendCommand("AT+SMCONF=\"PASSWORD\",\"_sTKZQbfemKK\"", "OK", 2000);
  sendCommand("AT+SMDISC", "OK", 2000);                       // make sure old br
  sendCommand("AT+SMCONN", "OK", 2000);                       // connect to broke
  sendCommand("AT+SMSTATE?", "OK", 2000);                     // check connection
  sendCommand("AT+SMSUB=\"#\",0", "OK", 2000);                // subscribe to all
  sendCommand("AT+SMPUB=\"test\",3,0,0", ">", 2000);          // unclear why, but
  Modemboard.print("hoi");                                    // send content of
}

void loop() {                                                 // this routine is
  if (Serial.available()) {                                   // when data is ava
    Modemboard.write(Serial.read());                          // ...we copy a byt
  }
  if (Modemboard.available()) {                               // when data is ava
```

Use the following documents as reference:



SIMCom
a SUNSEA AIOT company

SIM7000 Series_AT Command Manual_V1.04

Application

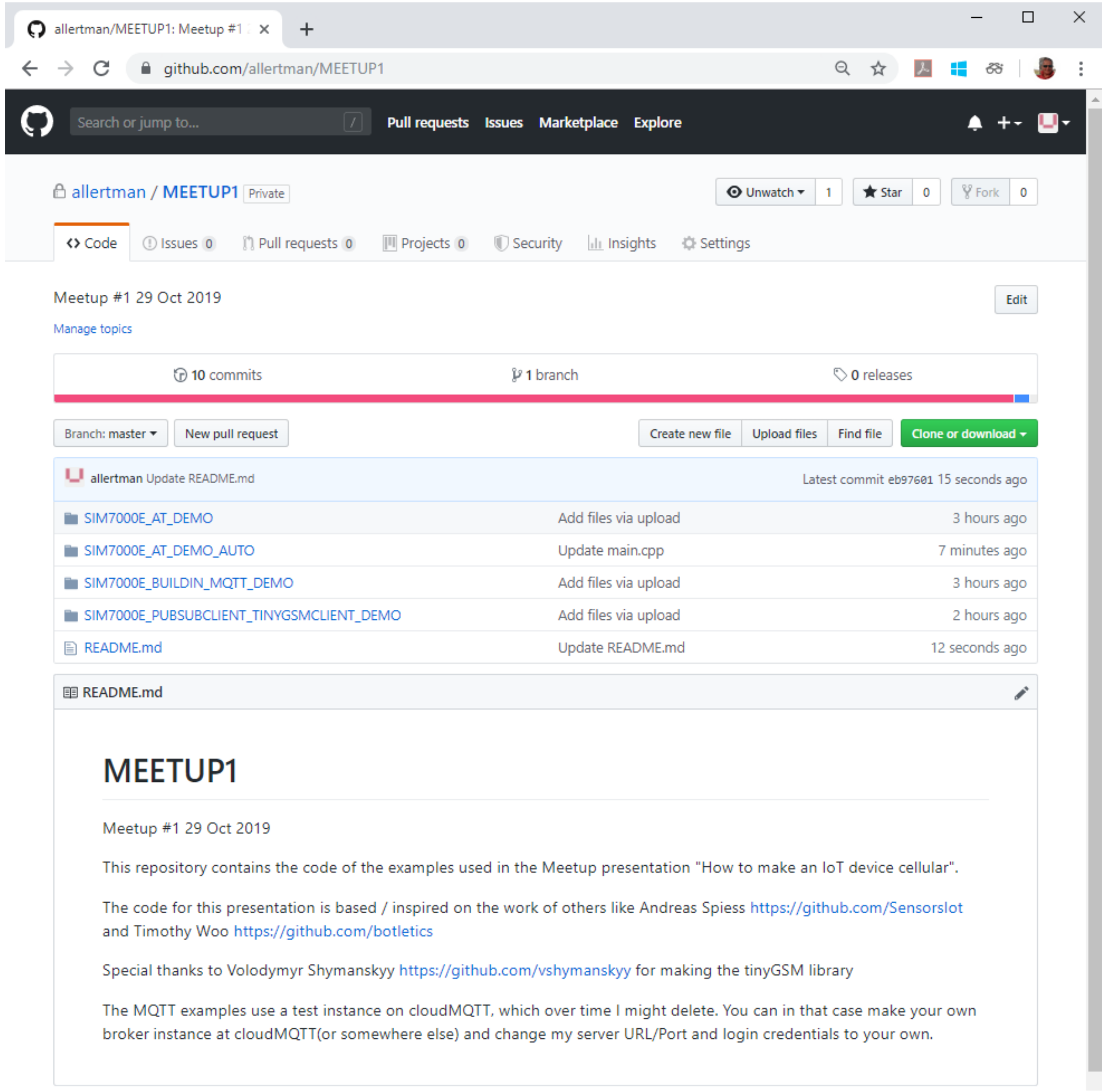
SIMCom
a SUNSEA AIOT company

SIM7000 Series_MQTT_Application Note

Version:1.01
Release Date:January 23, 2019

Simplify Communication.
www.simcom.com

https://github.com/allertman/MEETUP1



Meetup #1 29 Oct 2019

allertman Update README.md — Latest commit eb97601 15 seconds ago

SIM7000E_AT_DEMO	Add files via upload	3 hours ago
SIM7000E_AT_DEMO_AUTO	Update main.cpp	7 minutes ago
SIM7000E_BUILDIN_MQTT_DEMO	Add files via upload	3 hours ago
SIM7000E_PUBSUBCLIENT_TINYGSMCLIENT_DEMO	Add files via upload	2 hours ago
README.md	Update README.md	12 seconds ago

README.md

MEETUP1

Meetup #1 29 Oct 2019

This repository contains the code of the examples used in the Meetup presentation "How to make an IoT device cellular".

The code for this presentation is based / inspired on the work of others like Andreas Spiess https://github.com/Sensorslot and Timothy Woo https://github.com/botletics

Special thanks to Volodymyr Shymanskyy https://github.com/vshymanskyy for making the tinyGSM library

The MQTT examples use a test instance on cloudMQTT, which over time I might delete. You can in that case make your own broker instance at cloudMQTT(or somewhere else) and change my server URL/Port and login credentials to your own.