HELTEC AUTOMATION

CubeCell

Getting Started Guide

V1.0 © by WASN.eu

Summary

Setup of development Environment

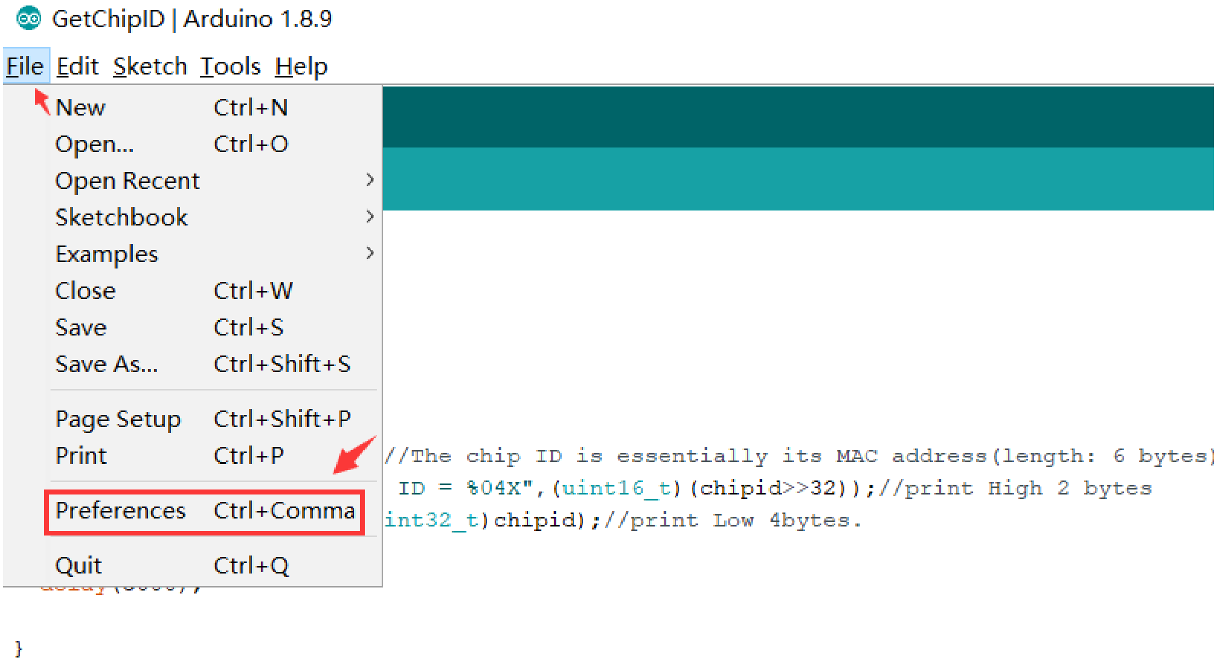
Download and Install Arduino IDE

Download and install the Arduino IDE from:

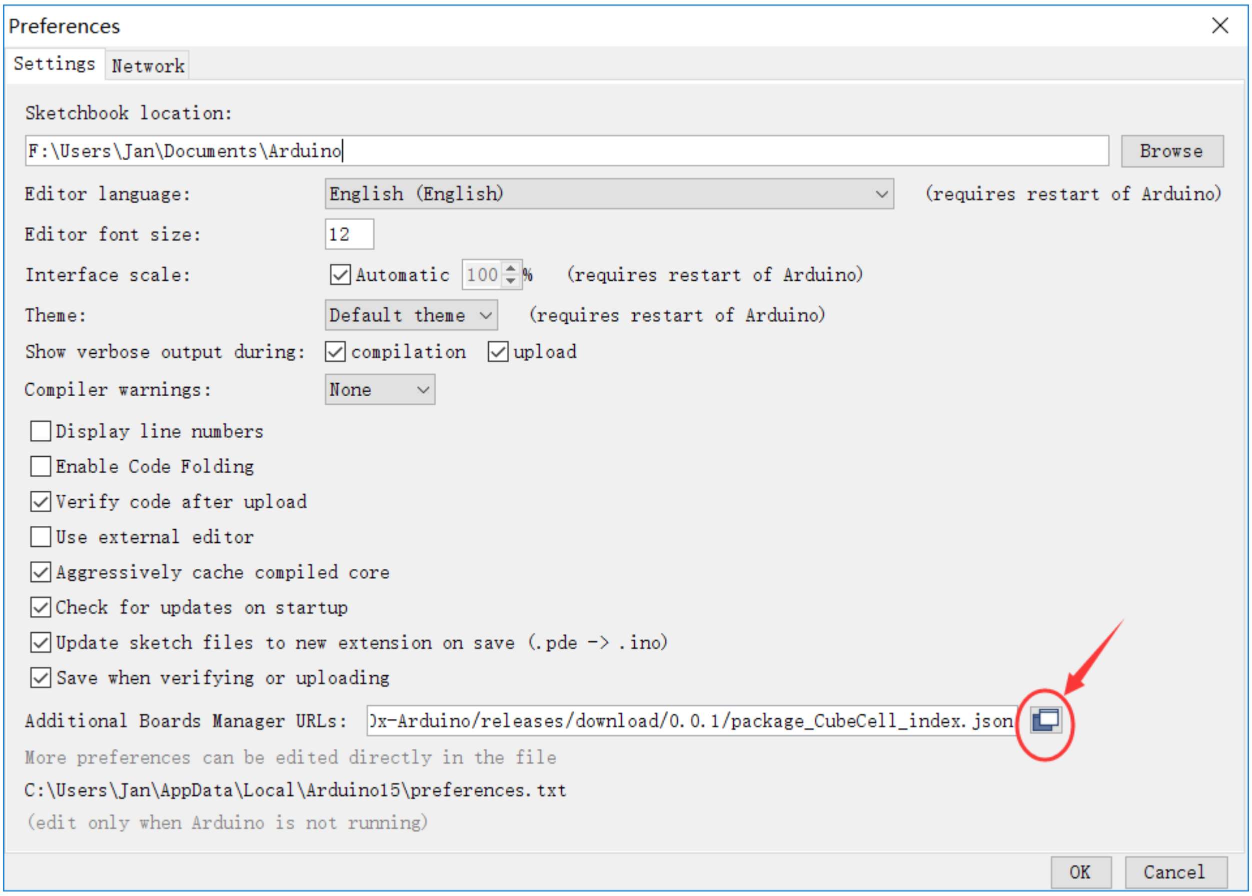
https://www.arduino.cc/en/Main/Software

Configure the Arduino IDE

- Open Arduino IDE
- click `File->Peferences->Settings`



```
//The chip ID is essentially its MAC address(length: 6 bytes)
 ID = %04X",(uint16_t)(chipid>>32));//print High 2 bytes
int32_t)chipid);//print Low 4bytes.

}
```



Sketchbook location:
F:\Users\Jan\Documents\Arduino

Editor language: English (English) (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☑ compilation ☑ upload

Compiler warnings: None

☐ Display line numbers

☐ Enable Code Folding

☑ Verify code after upload

☐ Use external editor

☑ Aggressively cache compiled core

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Additional Boards Manager URLs:)x-Arduino/releases/download/0.0.1/package_CubeCell_index.json

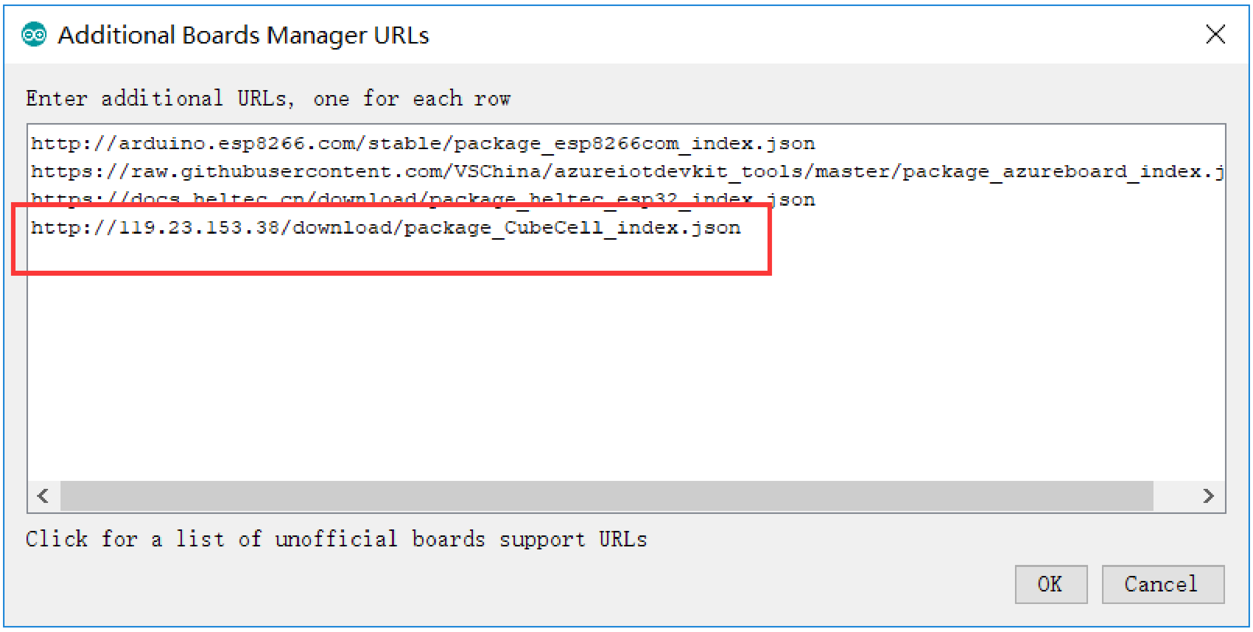
More preferences can be edited directly in the file
C:\Users\Jan\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)

Input last ASR650x board manager URL:

https://docs.heltec.cn/download/package_CubeCell_index.json

or

http://119.23.153.38/download/package_CubeCell_index.json

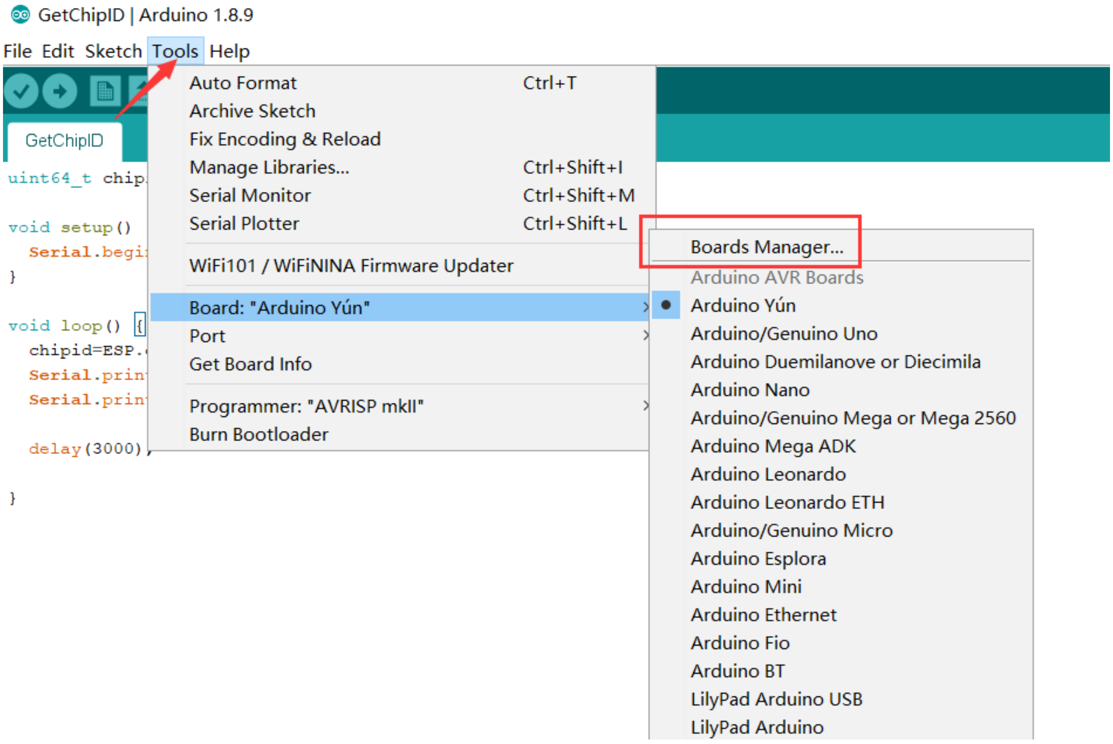


Enter additional URLs, one for each row

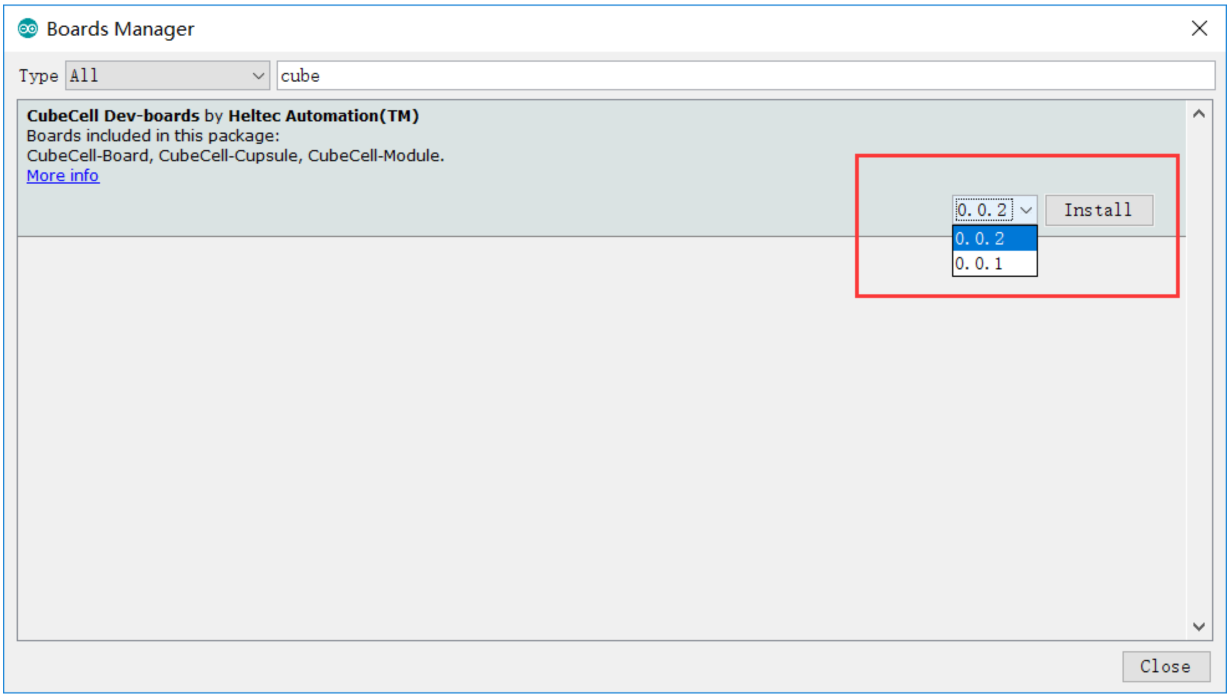
```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
https://raw.githubusercontent.com/VSChina/azureiotdevkit_tools/master/package_azureboard_index.j
https://docs.heltec.cn/download/package_heltec_esp32_index.json
http://119.23.153.38/download/package_CubeCell_index.json
```

Click for a list of unofficial boards support URLs

- Click `Tools->Board:->Boards Manager...`
- search `Heltec cubecell` in the new pop-up dialog
- select the latest `releases` and click `install`



```
uint64_t chip

void setup()
  Serial.begi
}

void loop() {
  chipid=ESP.
  Serial.prin
  Serial.prin

  delay(3000)

}
```



CubeCell Dev-boards by **Heltec Automation(TM)**
Boards included in this package:
CubeCell-Board, CubeCell-Cupsule, CubeCell-Module.
More info

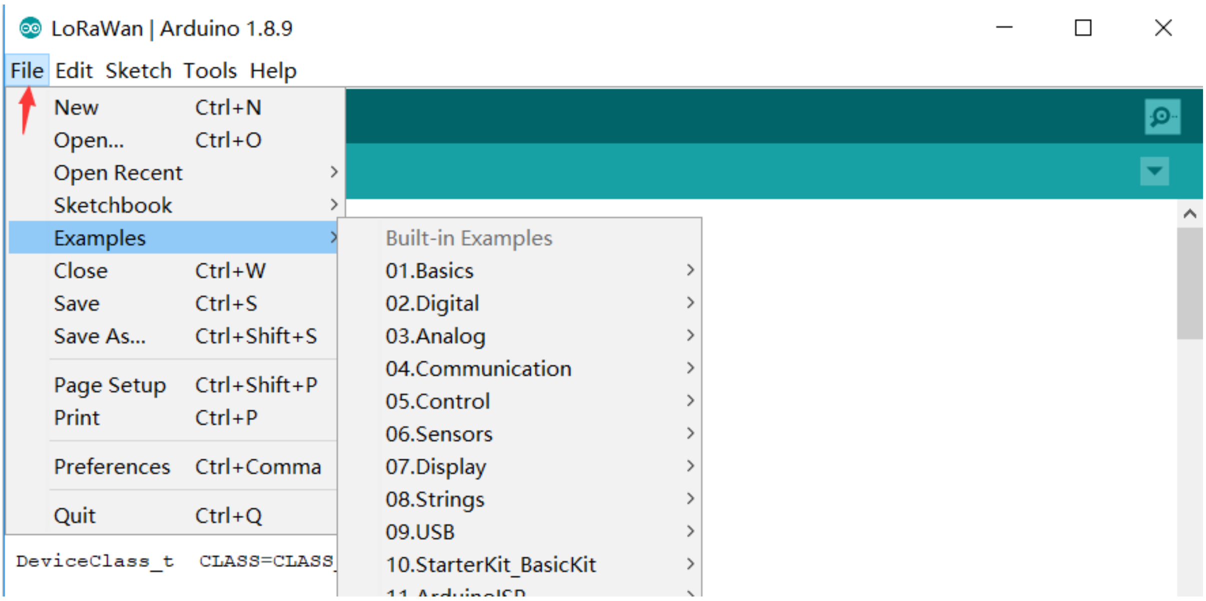
The source code of Heltec ASR650x series (ASR6501&ASR6502) framework is also available here:

https://github.com/HelTecAutomation/ASR650x-Arduino

Loading Firmware

Select the example:

File -> examples -> LoRa -> LoRaWan_Sensors -> LoRaWan_Multisensor



```
DeviceClass_t  CLASS=CLASS_
```

This sketch allows you to use a number of sensors. All you have to do is to set the number to a 1 beside the sensor you would like to use
(**only 1 selection is possible, selecting more will result in errors**).

Selecting AUTO_SCAN the Firmware will try to identify the connected I2C Sensor by its address. AUTO_SCAN takes its time, it is not very battery friendly.
Please select the sensor for saving battery power,
for maximum flexibility choose AUTO_SCAN.

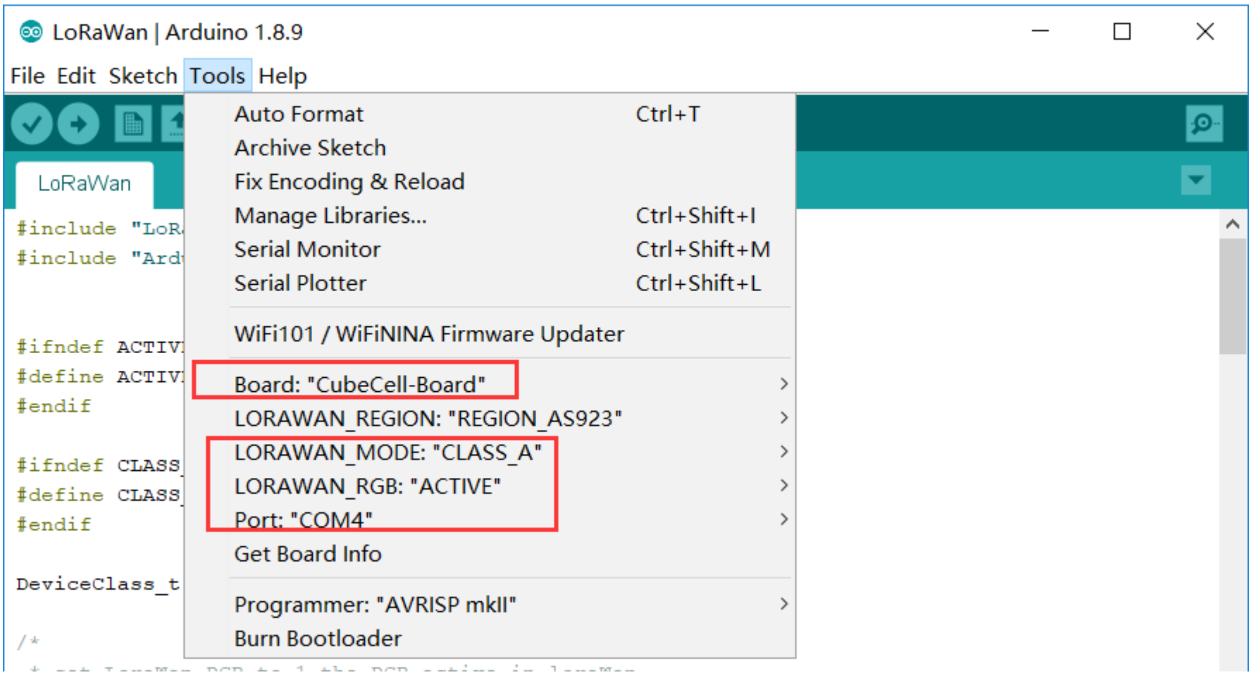
```
#define AUTO_SCAN  1
#define MJMCU_8128 0
#define BME_680    0 // wrong values
#define BME_280    0
#define CCS_811    0
#define BMP_180    0 // not tested
#define HDC_1080   0
#define BH_1750    0
#define One_Wire   0 // sensors not found
```

Login to TTN and create a new device. Get your keys from TTN and set them in the Sketch:

```
const char myDevEui[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
const char myAppEui[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
const char myAppKey[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0
```

Arduino DIE compile settings

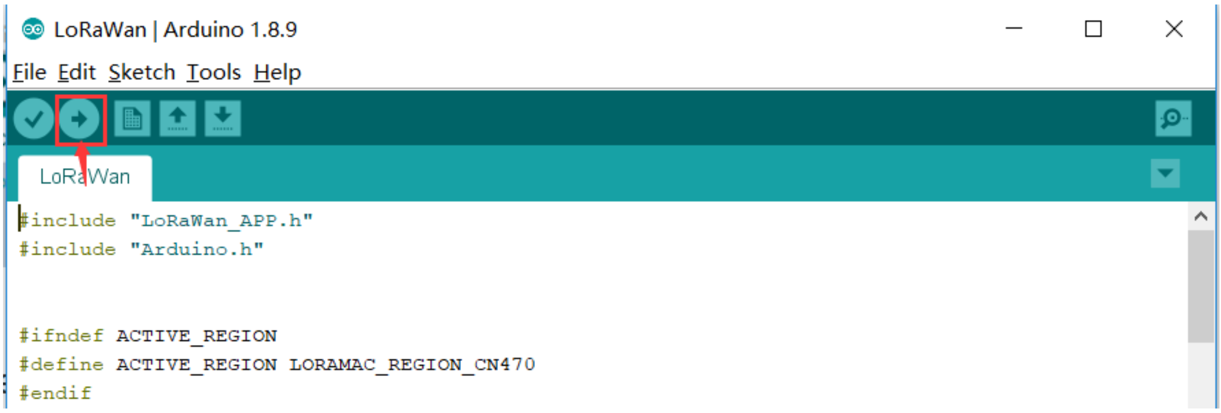
Correct select the board you had connected to computer and the specified serial port: e.g. CubeCell-Board, CubeCell-Module, CubeCell-Cupsule, COM-Port


LoRaWan | Arduino 1.8.9
File Edit Sketch Tools Help
LoRaWan
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader

Board: "CubeCell-Board"
LORAWAN_REGION: "REGION_AS923"
LORAWAN_MODE: "CLASS_A"
LORAWAN_RGB: "ACTIVE"
Port: "COM4"

```
#include "LoR
#include "Ard


#ifndef ACTIV
#define ACTIV
#endif

#ifndef CLASS
#define CLASS
#endif

DeviceClass_t

/*
```

Compile and upload it



```
#include "LoRaWan_APP.h"
#include "Arduino.h"


#ifndef ACTIVE_REGION
#define ACTIVE_REGION LORAMAC_REGION_CN470
#endif
```

The Things Network decoder

Login to TTN and define the decoder.

The decoder is found in the example directory.

ttn-decoder.js:

```
function bytesToFloat(by) {
  var bits = by[3]<<24 | by[2]<<16 | by[1]<<8 | by[0];
  var sign = (bits>>>31 === 0) ? 1.0 : -1.0;
  var e = bits>>>23 & 0xff;
  var m = (e === 0) ? (bits & 0x7fffff)<<1 : (bits & 0x7fffff) | 0x800000;
  var f = sign * m * Math.pow(2, e - 150);
  return f;
}

function Decoder(bytes, port) {

// Decode an uplink message from a buffer
// (array) of bytes to an object of fields.
var decoded = {};

if (port === 2) {
  var i = 1;
  sensor = bytes[0].toFixed(0);
  decoded.sensortype = sensor;

  if (sensor === "0") { // MJMCU-8128
    decoded.temperature = ((((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10) - 100).toFixed(1);
    decoded.humidity = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.pressure = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.lux = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.co2 = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
    decoded.tvoc = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
    decoded.battery = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
  }
  if (sensor === "1") { // BME680
    decoded.temperature = ((((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10) - 100).toFixed(1);
    decoded.humidity = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.pressure = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.gas = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.battery = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
  }
  if (sensor === "2") { // BME280
    decoded.temperature = ((((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10) - 100).toFixed(1);
    decoded.humidity = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.pressure = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.battery = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
  }
  if (sensor === "3") { // CCS811
    decoded.co2 = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
    decoded.tvoc = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
    decoded.battery = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
  }
  if (sensor === "4") { // HDC1080
    decoded.temperature = ((((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10) - 100).toFixed(1);
    decoded.humidity = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.battery = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
  }
  if (sensor === "5") { // BMP180
    decoded.temperature = ((((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10) - 100).toFixed(1);
    decoded.pressure = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.battery = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
  }
  if (sensor === "6") { // BH1750
    decoded.lux = (((bytes[i++] << 8) | bytes[i++]).toFixed(0) / 10).toFixed(1);
    decoded.battery = ((bytes[i++] << 8) | bytes[i++]).toFixed(0);
  }
}

return decoded;
}
```