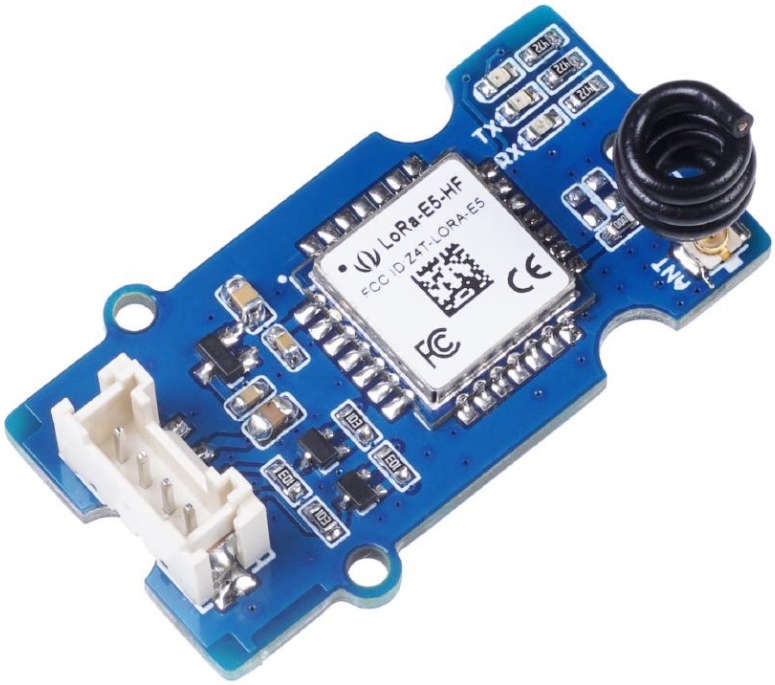
Grove - LoRa-E5



Grove LoRa-E5 embedded with LoRa-E5 STM32WLE5JC, powered by ARM Cortex M4 ultra-low-power MCU core and LoRa SX126x, is a wireless radio module supporting LoRa and LoRaWAN protocol on the EU868 & US915 frequency and (G)FSK, BPSK, (G)MSK, LoRa modulations. Grove - LoRa-E5 can endow your development boards' strong features of ultra-long transmitting range by easily plug and play with Grove connector on board.

As an upgrade of our old version - Grove - LoRa Radio - powered by RFM95 ultra-long-range Transceiver Module, Grove LoRa-E5 embedded with LoRa-E5 STM32WLE5JC Module is a high-performance and easy-to-use wireless radio LoRa module supporting LoRaWAN protocol.

LoRa-E5 LoRaWAN STM32WLE5JC module is the major functional part integrated into Grove - LoRa-E5. It is a LoRaWAN module that embedded with ARM Cortex M4 ultra-low-power MCU core and LoRa SX126x, as the world-first combo of LoRa RF and MCU chip into one single tiny module, it supports (G)FSK, BPSK, (G)MSK, and LoRa modulations, and is FCC, CE certified. (Learn more about LoRa-E5 from LoRa-E5 wiki)

More comparison between the LoRa-E5 and RFM95 chip:

LoRa-E5 (STM32WLE5JC)

RFM95 and RFM95W

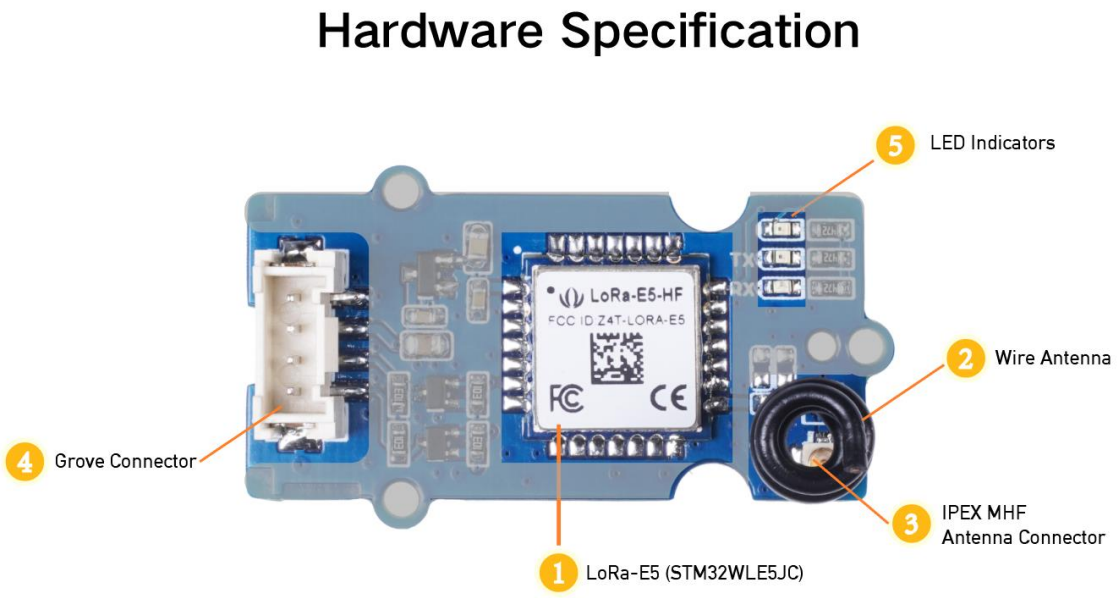
Core	32-bit Arm Cortex-M4 CPU, up to 48MHz	NONE
LoRaWAN stack	Built-in with AT Command Firmware; Program with STM32Cube MCU Package	NONE
Package	12*12mm, 28 pins SMD	16*16mm, 16 pins SMD
Interfaces	UART*3, I2C*1, ADC(12-bit)*1, SPI*1, GPIO*6	SPI*1, DIO*6
Sensitivity	-116.5dBm(SF5), -121.5dBm(SF7), -136dBm(SF12)	-111dBm ~ -148dBm
Modulation	LoRa, (G)FSK, (G)MSK and BPSK	LoRa, (G)FSK, (G)MSK and OOK
Certificate	FCC and CE (EU868/US915)	NONE
Power Supply	1.8V ~ 3.6V	1.8V ~ 3.7V
RF Output Power	up to +20.8 dBm at 3.3V	up to +20 dBm

By connecting Grove - LoRa-E5 to your development boards, your devices are able to communicate with and control LoRa-E5 conveniently by AT command through UART connection. Grove LoRa-E5 will be a superior choice for IoT device development, testing, and long-distance, ultra-low power consumption IoT scenarios like smart agriculture, smart office, and smart industry. It is designed with industrial standards with a wide working temperature at -40℃ ~ 85℃, high sensitivity between -116.5 dBm and -136 dBm, and power output between 10 dBm and 22 dBm.

Features

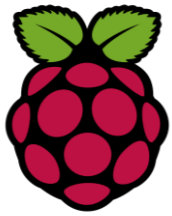
- LoRa-E5 (STM32WLE5JC) embedded
- Support LoRaWAN protocol on EU868/US915 frequency band
- Ultra-long transmitting range up to 10km (Ideal value in open space)
- Easy control by AT command via UART connection
- Rapid prototyping with plug-and-play Grove interfaces
- Ultra-low power consumption and high performance

Harware Overview



1. LoRa-E5 STM32WLE5JC (Datasheet)
2. MHF IPEX Connector
3. Wire Antenna
4. Grove Connector
5. LED Indicators

Platform Supported

Arduino	Raspberry Pi
	

Specification

General Parameters	
Voltage Supply:	3.3V - 5V
Power Output:	Up to +20 dBm at 3.3V
Working Frequency	868/915MHz
Protocol	LoRaWAN
Sensitivity	-116.5dBm ~ -136dBm
Modulation	LoRa, (G)FSK, (G)MSK and BPSK
Current	Only 60uA in sleep mode
Size	20*40mm
Working Temperature	-40℃ ~ 85℃

Part List:
Grove - LoRa-E5 PCBA *1
Grove Universal Cable *1

Application

- Works for LoRaWAN sensor nodes and any wireless communication application
- IoT device testing and development

Getting Started

Preparations

Here is a demo showing you how to connect TTN (The Things Network) and Seeeduino XIAO module via Grove - LoRa-E5 module. These modules are able to collect temperature and humidity parameters from the environment and send them back to TTN. The flashing LED lights on the Seeeduino Xiao indicate the status of the temperature and humidity sensor as connecting to TTN cloud.

Attention
Please ensure the consistent of the frequency band among the end nodes, gateway, and TTN configuration you are using by following this instruction. The frequency plan this demo applied is for **EU868**.

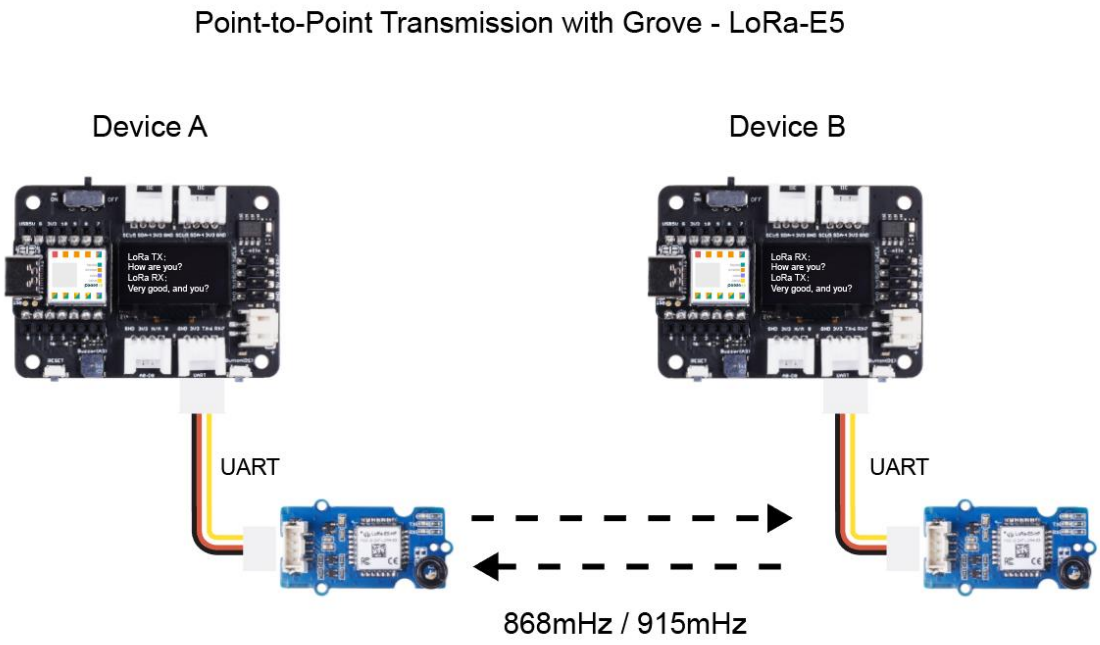
Hardware Required

Seeeduino XIAO	Grove - LoRa-E5	Seeeduino XIAO Expansion Board	Grove - Temperature & Humidity Sensor (DHT11)
			

Notes
If this is your first time using Seeeduino XIAO, please refer to Seeeduino XIAO's wiki. If this is your first time to use Arduino, Arduino's website is a great resource for you to start your Arduino journey.

Hardware Connection

- **Step 1.** Connect the LoRa-E5 module directly to the "UART" slot.
- **Step 2.** Put DH11 into the "A0/D0" socket. As shown below.
- **Step 3.** Download the code, please refer to the software part.



Software Preparation

Notes
If this is the first time you work with Arduino, we strongly recommend you to see Getting Started with Arduino before the start. Click to learn about detail about how to install an Arduino Library

Download Library

- **Step 1.** Install the u8g2 library
- **Step 2.** Install the DHT sensor library

Software Code

Download the example; copy the code stick onto the Aruino IDE and then upload it.

```
#include <Arduino.h>
#include <U8x8lib.h>
#include "DHT.h"
#define DHTPIN 0 // what pin we're connected to
// Uncomment whatever type you're using!
#define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT22 // DHT 22 (AM2302)
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
DHT dht(DHTPIN, DHTTYPE);
U8X8_SSD1306_128X64_NONAME_HW_I2C u8x8(/* reset=*/U8X8_PIN_NONE);
// U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(/* clock=*/ SCL, /* data=*/ SDA, /*
reset=*/ U8X8_PIN_NONE); // OLEDs without Reset of the Display
static char recv_buf[512];
static bool is_exist = false;
static bool is_join = false;
static int led = 0;
static int at_send_check_response(char *p_ack, int timeout_ms, char *p_cmd, ...)
{
int ch;
int num = 0;
int index = 0;
int startMillis = 0;
va_list args;
memset(recv_buf, 0, sizeof(recv_buf));
va_start(args, p_cmd);
Serial1.printf(p_cmd, args);
Serial.printf(p_cmd, args);
va_end(args);
delay(200);
startMillis = millis();
if (p_ack == NULL)
{
return 0;
}
do
{
while (Serial1.available() > 0)
{
ch = Serial1.read();
recv_buf[index++] = ch;
Serial.print((char)ch);
```

```
delay(2);
}
if (strstr(recv_buf, p_ack) != NULL)
{
return 1;
}
} while (millis() - startMillis < timeout_ms);
return 0;
}
static void recv_prase(char *p_msg)
{
if (p_msg == NULL)
{
return;
}
char *p_start = NULL;
int data = 0;
int rssi = 0;
int snr = 0;
p_start = strstr(p_msg, "RX");
if (p_start && (1 == sscanf(p_start, "RX: \"%d\"\r\n", &data)))
{
Serial.println(data);
u8x8.setCursor(2, 4);
u8x8.print("led :");
led = !!data;
u8x8.print(led);
if (led)
{
digitalWrite(LED_BUILTIN, LOW);
}
else
{
digitalWrite(LED_BUILTIN, HIGH);
}
}
p_start = strstr(p_msg, "RSSI");
if (p_start && (1 == sscanf(p_start, "RSSI %d,", &rssi)))
{
u8x8.setCursor(0, 6);
u8x8.print(" ");
u8x8.setCursor(2, 6);
u8x8.print("rssi:");
u8x8.print(rssi);
}
p_start = strstr(p_msg, "SNR");
```

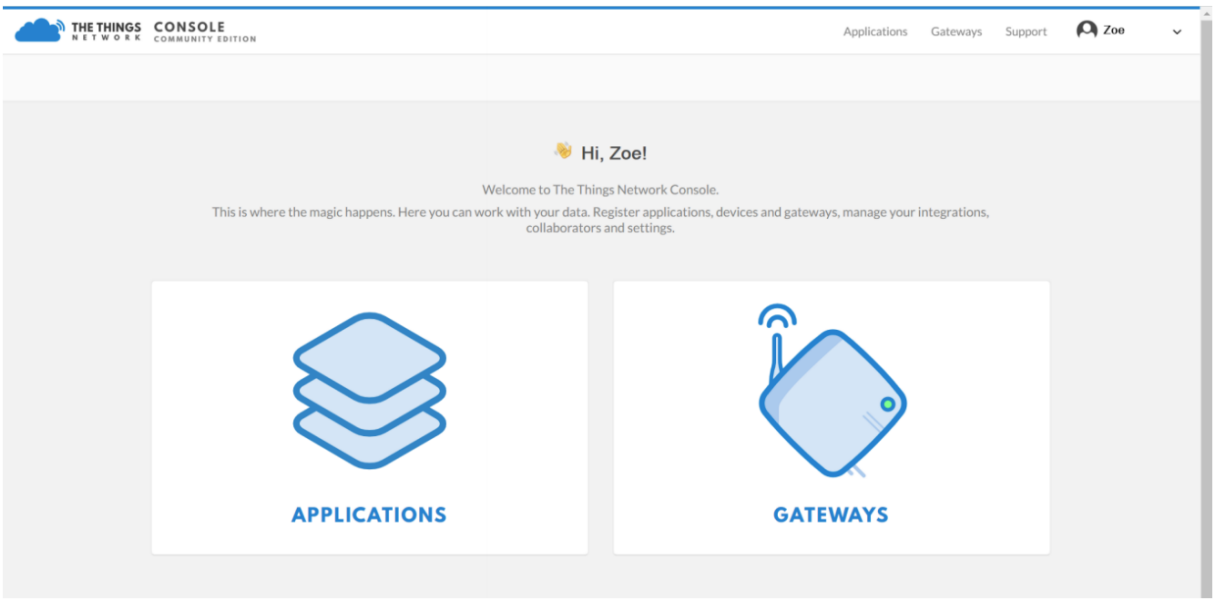
```
if (p_start && (1 == sscanf(p_start, "SNR %d", &snr)))
{
u8x8.setCursor(0, 7);
u8x8.print(" ");
u8x8.setCursor(2, 7);
u8x8.print("snr :");
u8x8.print(snr);
}
}
void setup(void)
{
u8x8.begin();
u8x8.setFlipMode(1);
u8x8.setFont(u8x8_font_chroma48medium8_r);
Serial.begin(115200);
pinMode(LED_BUILTIN, OUTPUT);
digitalWrite(LED_BUILTIN, HIGH);
Serial1.begin(9600);
Serial.print("E5 LORAWAN TEST\r\n");
u8x8.setCursor(0, 0);
if (at_send_check_response("+AT: OK", 100, "AT\r\n"))
{
is_exist = true;
at_send_check_response("+ID: AppEui", 1000, "AT+ID\r\n");
at_send_check_response("+MODE: LWOTAA", 1000, "AT+MODE=LWOTAA\r\n");
at_send_check_response("+DR: EU868", 1000, "AT+DR=EU868\r\n");
at_send_check_response("+CH: NUM", 1000, "AT+CH=NUM,0-2\r\n");
at_send_check_response("+KEY: APPKEY", 1000,
"AT+KEY=APPKEY,\"2B7E151628AED2A6ABF7158809CF4F3C\"\r\n");
at_send_check_response("+CLASS: C", 1000, "AT+CLASS=A\r\n");
at_send_check_response("+PORT: 8", 1000, "AT+PORT=8\r\n");
delay(200);
u8x8.setCursor(5, 0);
u8x8.print("LoRaWAN");
is_join = true;
}
else
{
is_exist = false;
Serial.print("No E5 module found.\r\n");
u8x8.setCursor(0, 1);
u8x8.print("unfound E5 !");
}
dht.begin();
u8x8.setCursor(0, 2);
u8x8.setCursor(2, 2);
```

```
u8x8.print("temp:");
u8x8.setCursor(2, 3);
u8x8.print("humi:");
u8x8.setCursor(2, 4);
u8x8.print("led :");
u8x8.print(led);
}
void loop(void)
{
float temp = 0;
float humi = 0;
temp = dht.readTemperature();
humi = dht.readHumidity();
Serial.print("Humidity: ");
Serial.print(humi);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(temp);
Serial.println(" *C");
u8x8.setCursor(0, 2);
u8x8.print(" ");
u8x8.setCursor(2, 2);
u8x8.print("temp:");
u8x8.print(temp);
u8x8.setCursor(2, 3);
u8x8.print("humi:");
u8x8.print(humi);
if (is_exist)
{
int ret = 0;
if (is_join)
{
ret = at_send_check_response("+JOIN: Network joined", 12000, "AT+JOIN\r\n");
if (ret)
{
is_join = false;
}
else
{
at_send_check_response("+ID: AppEui", 1000, "AT+ID\r\n");
Serial.print("JOIN failed!\r\n\r\n");
delay(5000);
}
}
else
{
```

```
char cmd[128];
sprintf(cmd, "AT+CMSGHEX=\"%04X%04X\"\r\n", (int)temp, (int)humi);
ret = at_send_check_response("Done", 5000, cmd);
if (ret)
{
recv_prase(recv_buf);
}
else
{
Serial.print("Send failed!\r\n\r\n");
}
delay(5000);
}
}
else
{
delay(1000);
}
}
```

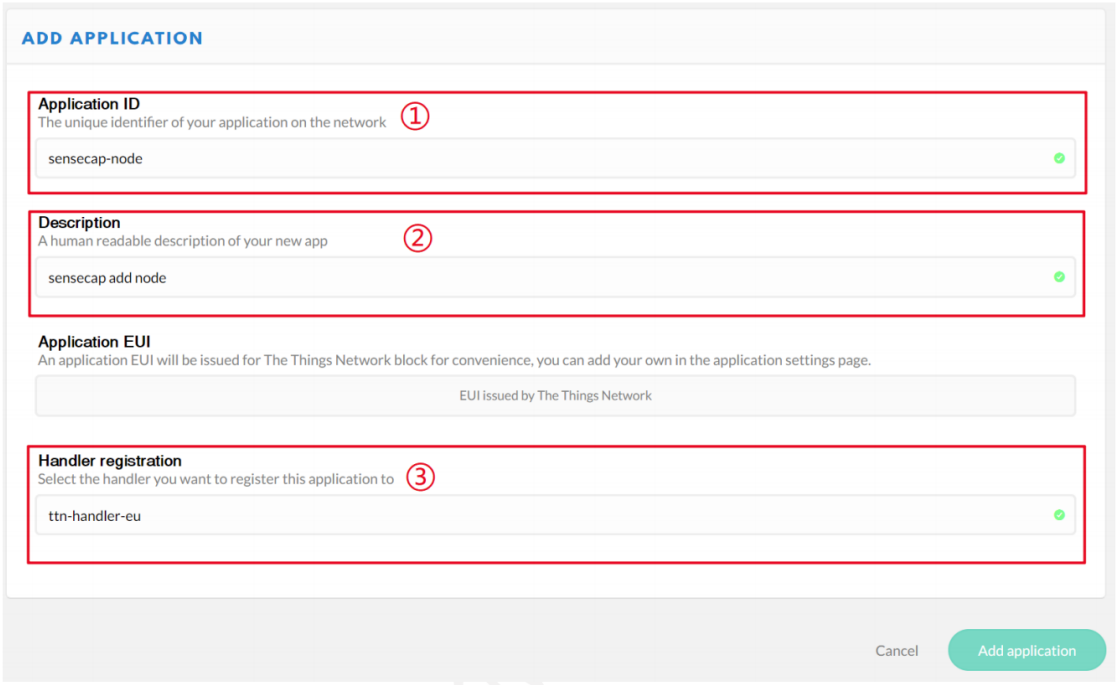
TTN Console Configuration Setup

- **Step 1**: Load into TTN website: https://www.thethingsnetwork.org and create your account, then access “Console” and first click on "APPLICATIONS".

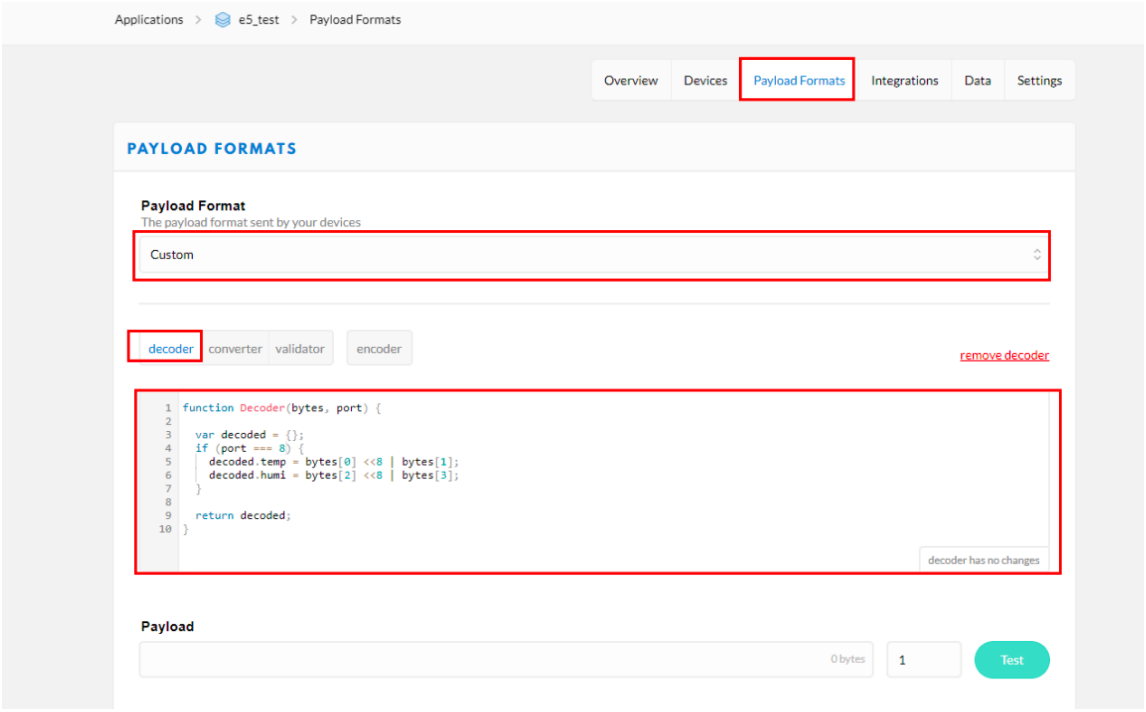


- **Step 2**: Add an Application:
- Application ID: Enter a unique name.

- Description: Enter a description.
- Handler registration: select the same handler as the gateway router.
- Select Add Application to continue.

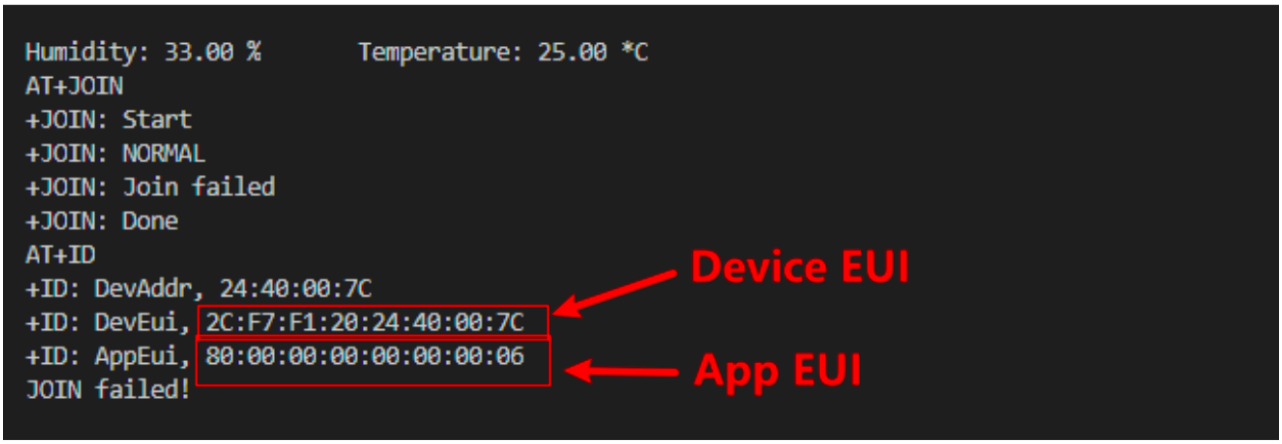


- **Step 3**: Add a decoding script to the application and save.

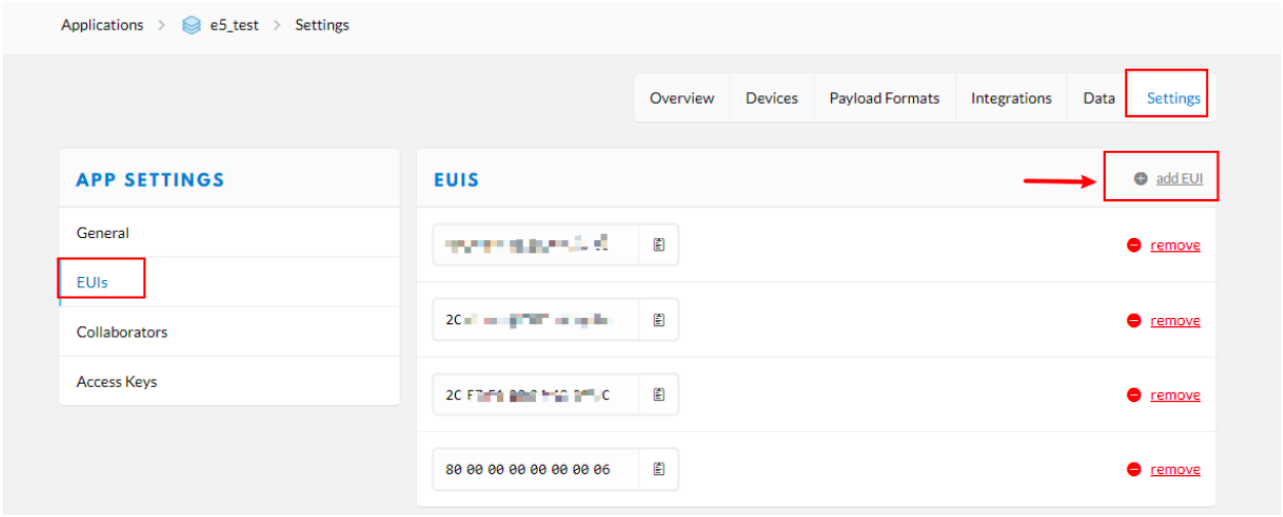


```
function Decoder(bytes, port) {

var decoded = {};
if (port === 8) {
decoded.temp = bytes[0] <<8 | bytes[1];
decoded.humi = bytes[2] <<8 | bytes[3];
}

return decoded;
}
```

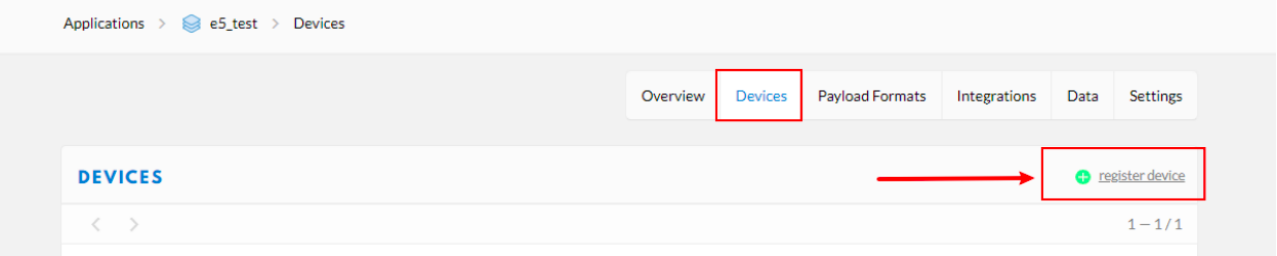
- **Step 4**: Add Grove - LoRa-E5 device to the TTN Console
- Run the downloaded program module, view the DEVEUI and APP EUI of the LoRa-E5 module through the serial port.



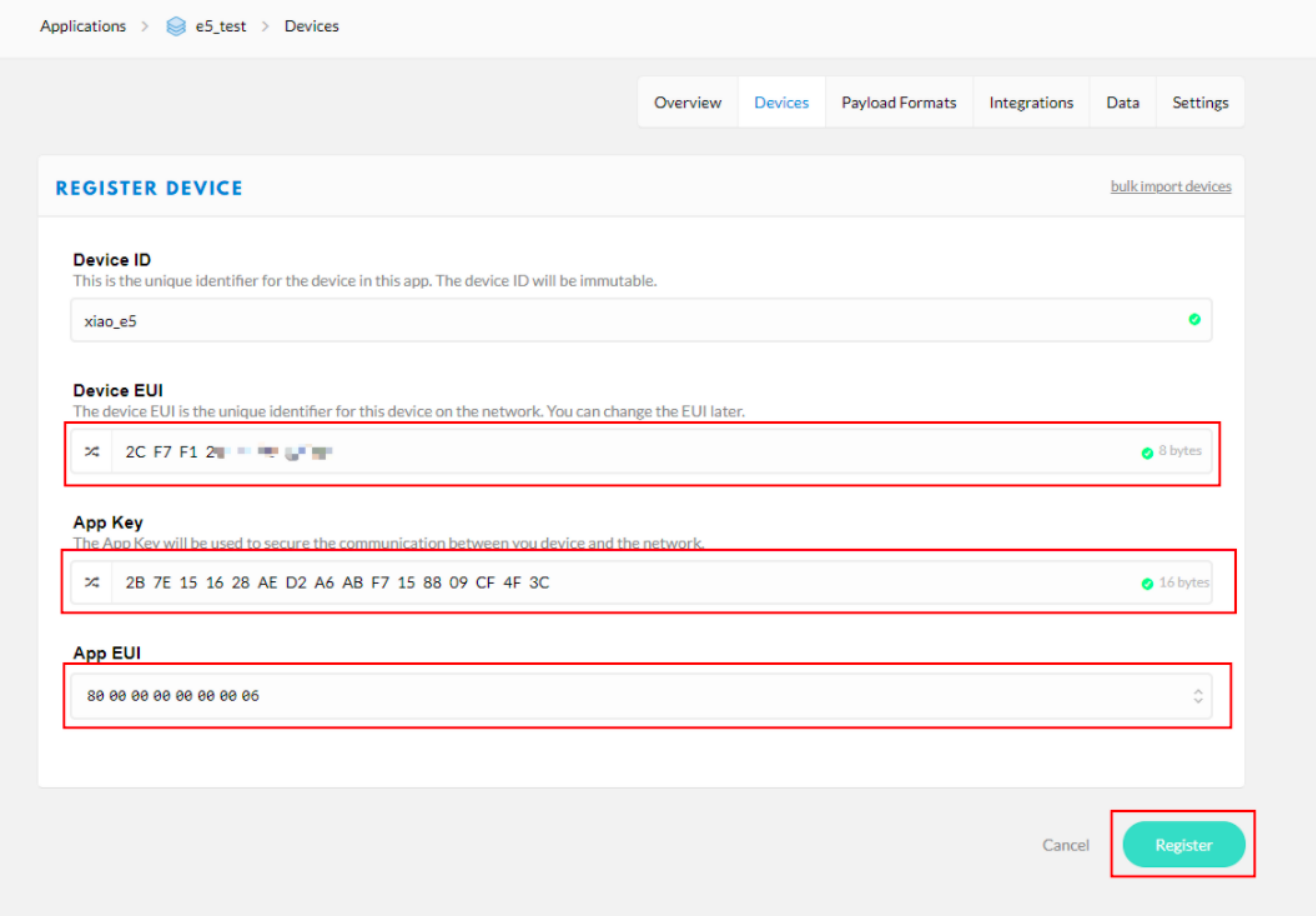
1. Add these two EUIs to the application.



- **Step 5**: Register Device: enter the registered device page

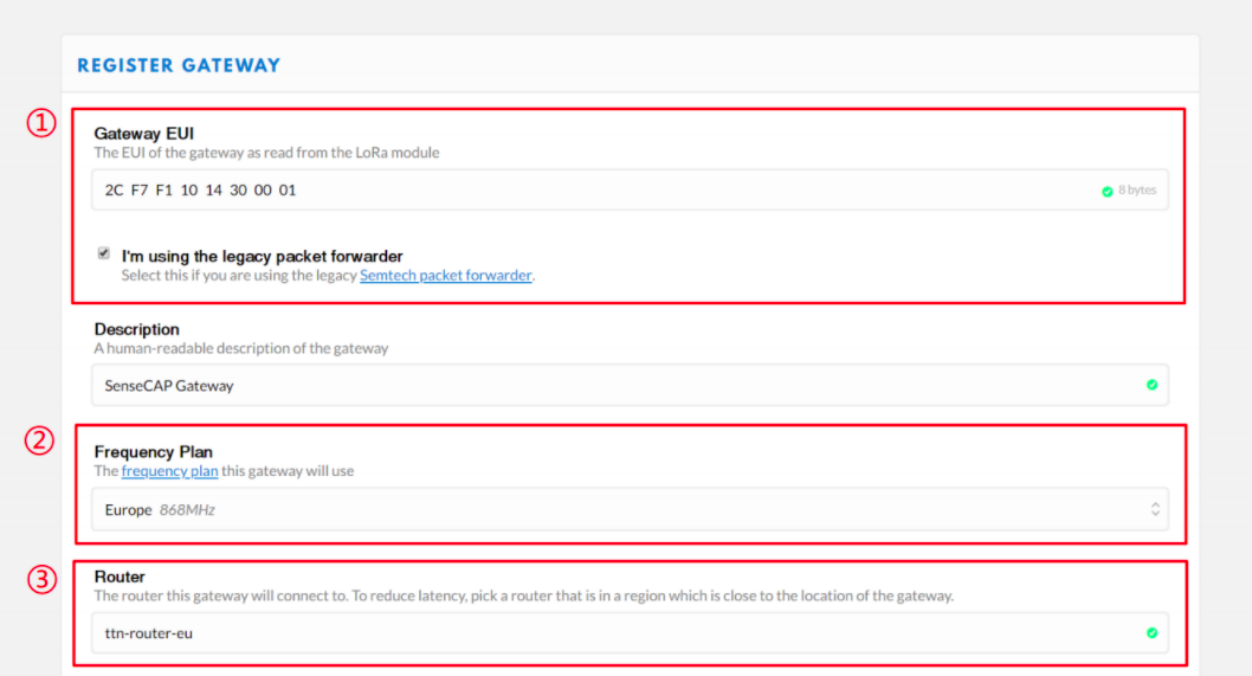


1. Device ID: Enter a unique name.
2. Device EUI: Select the E5 dev EUI.
3. APP KEY: Use this APPkey 2B7E151628AED2A6ABF7158809CF4F3C
4. App EUI: Select the E5 App EUI.



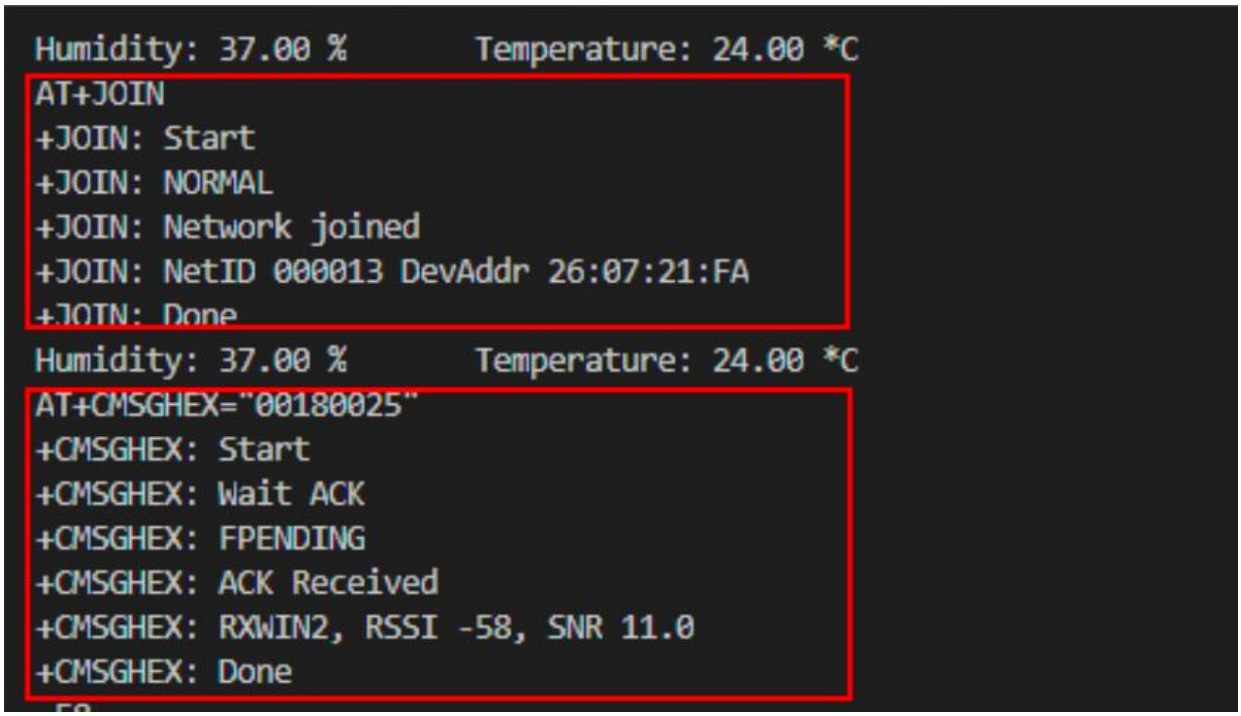
- **Step 6**: Gateway Registration on TTN Console

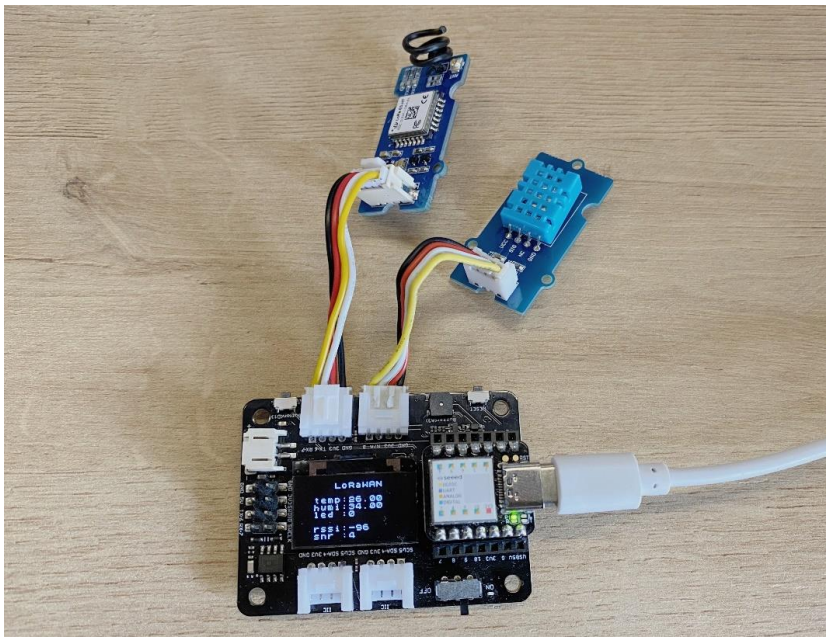
Please refer to the instruction shown in The Things Indoor Gateway wiki page: The Things Indoor Gateway Get Started with SenseCAP



- **Step 7**: Review Result

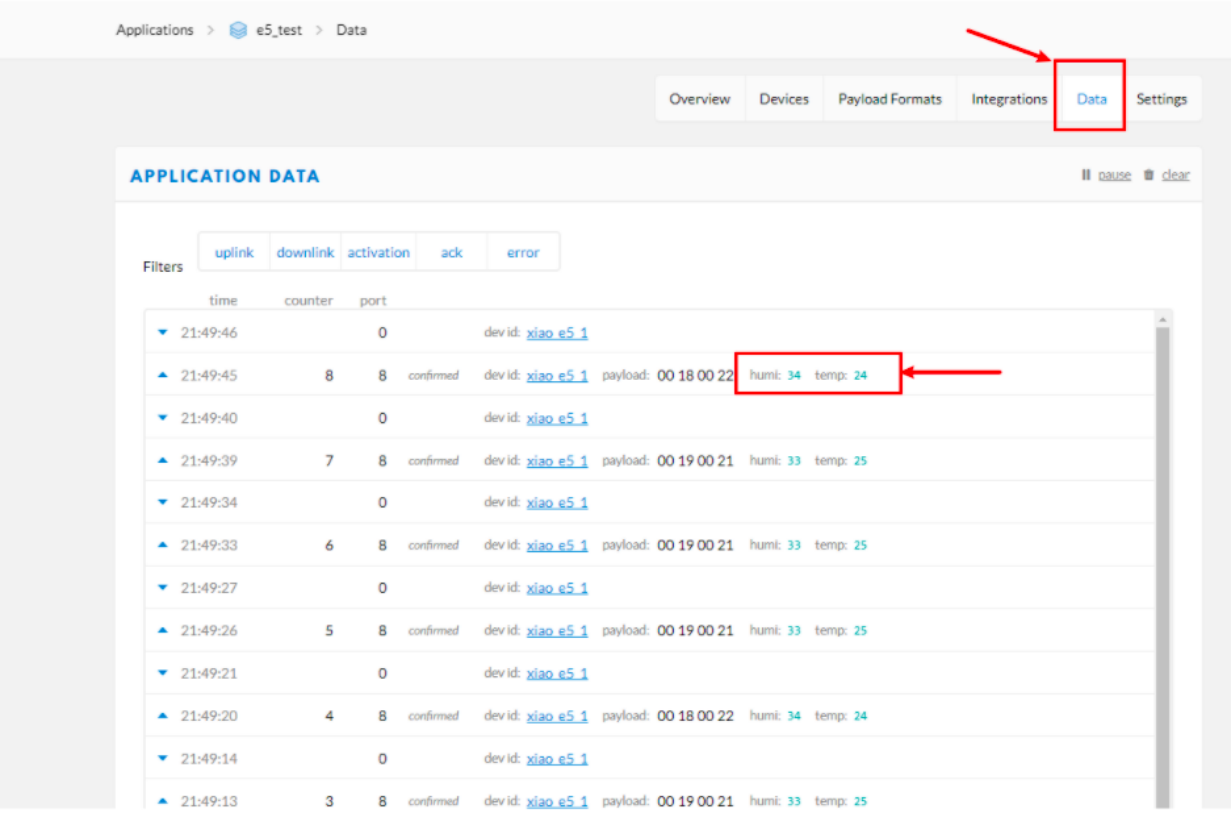
you can check the readings via the Serial Monitor.





Temperature and Humidity Parameters

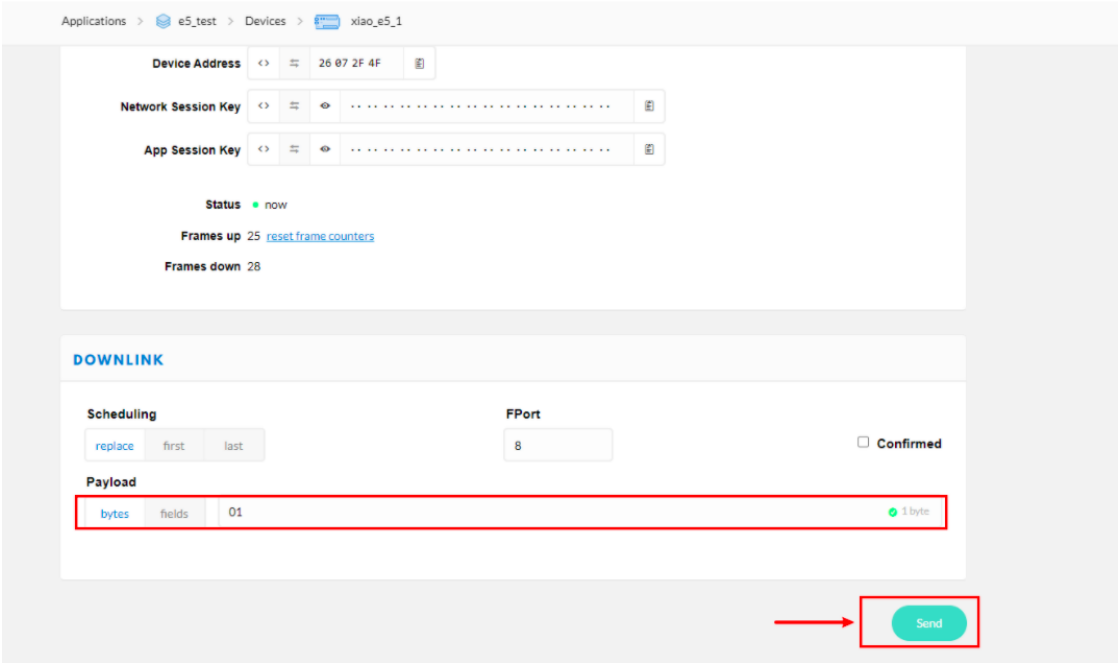
- **Step 1**: Enter the APPLICATION created in TTN, click on the data page to view the reported data

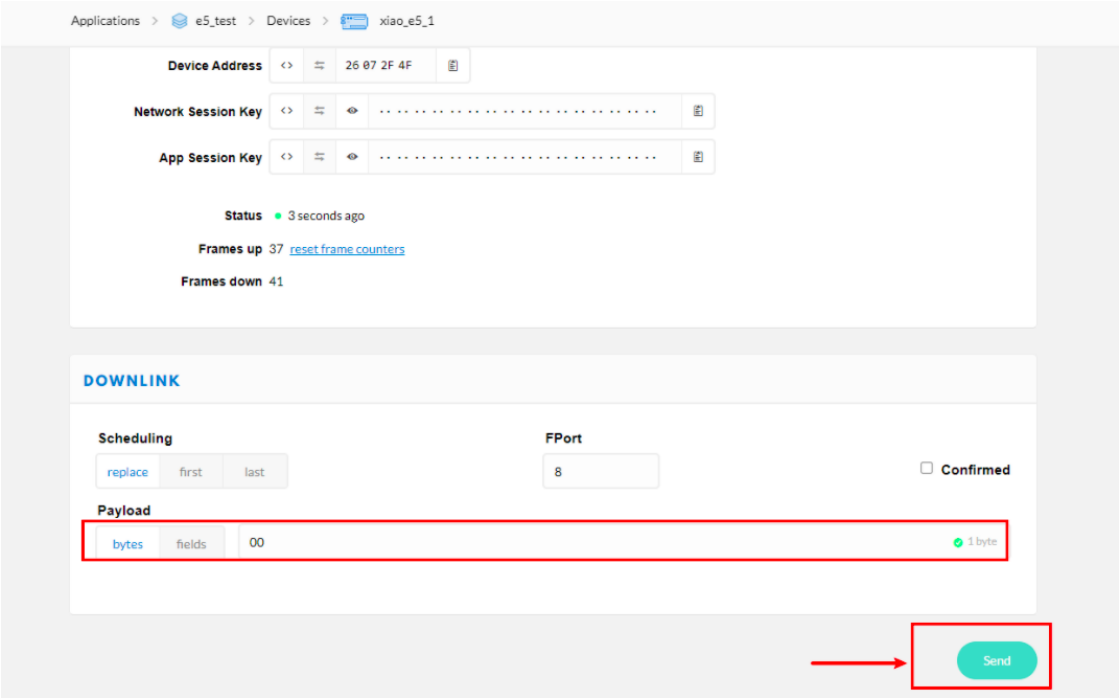


- **Step 2**: LED control

Enter the current device control page. Send the specified data in the “DOWNLINK” window.

Send "01" to turn on LED light; Send "00" to turn off:





Resources

Datasheet:

- Grove LoRa-E5 v1.0.brd
- Grove LoRa-E5 v1.0.pdf
- Grove LoRa-E5 v1.0.sch
- LoRa-E5 datasheet and specifications
- LoRa-E5 AT Command Specification
- STM32WLE5JC Datasheet

Certifications:

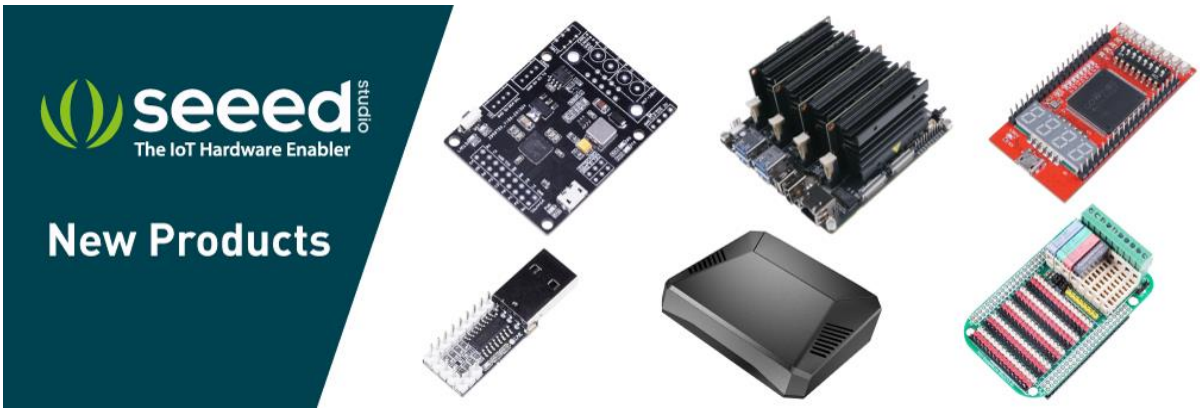
- LoRa-E5-HF Certification CE-VOC-RED
- LoRa-E5-HF FCC Certification -DSS
- LoRa-E5-HF FCC Certification -DTS

Relevant SDK:

- STM32Cube MCU Package for STM32WL series

Tech Support

Please submit any technical issue into our forum.



https://wiki.seeedstudio.com/Grove_LoRa_E5_New_Version/