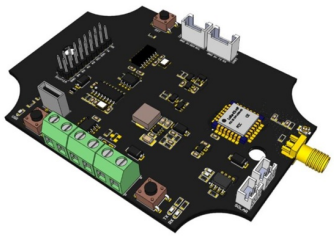


seeed studio Wio-E5 CAN FD Development Kit User Guide

2022/7/5 13:34
LoRa E5 – Longan Docs
Wio-E5

Contents

Wio-E5 CAN FD Development Kit

Wio-E5 CAN FD Development Kit is a compact development toolset for you to unlock the powerful performance of the Wio-E5 STM32WLE5JC as well as CAN FD and RS485 communication. We also provide a simple waterproof case and solar charging interface, you can easily make a sensor node placed outdoors.

Wio-E5 Dev Board embedded with Wio-E5 STM32WLE5JC Module that supports LoRaWAN protocol on global frequency band. It leads out full GPIOs of Wio-E5 supporting various data protocols and interfaces including RS-485, Grove, male/female headers, etc. It would be a perfect choice for fast testing and rapid prototyping of your Long Range IoT projects. Wio-E5 Dev Board embedded with Wio-E5 STM32WLE5JC Module, which is the world-first combo of Long Range RF and MCU chip into one single tiny chip and is FCC and CE certified. It is powered by ARM Cortex-M4 core and Semtech SX126X Long Range chip, supports both LoRaWAN and Long Range protocol on the worldwide frequency and (G)FSK, BPSK, (G)MSK, and Long Range modulations. Learn more about Wio-E5 here.

Wio-E5 Dev Board has a long-distance transmission range of Wio-E5 up to 10km in an open area.

The sleep current of Wio-E5 modules on board is as low as 2.1 uA(WOR mode). It is designed with industrial standards with a wide working temperature at -40 ℃ ~ 85℃, high sensitivity between -116.5 dBm ~ -136 dBm,

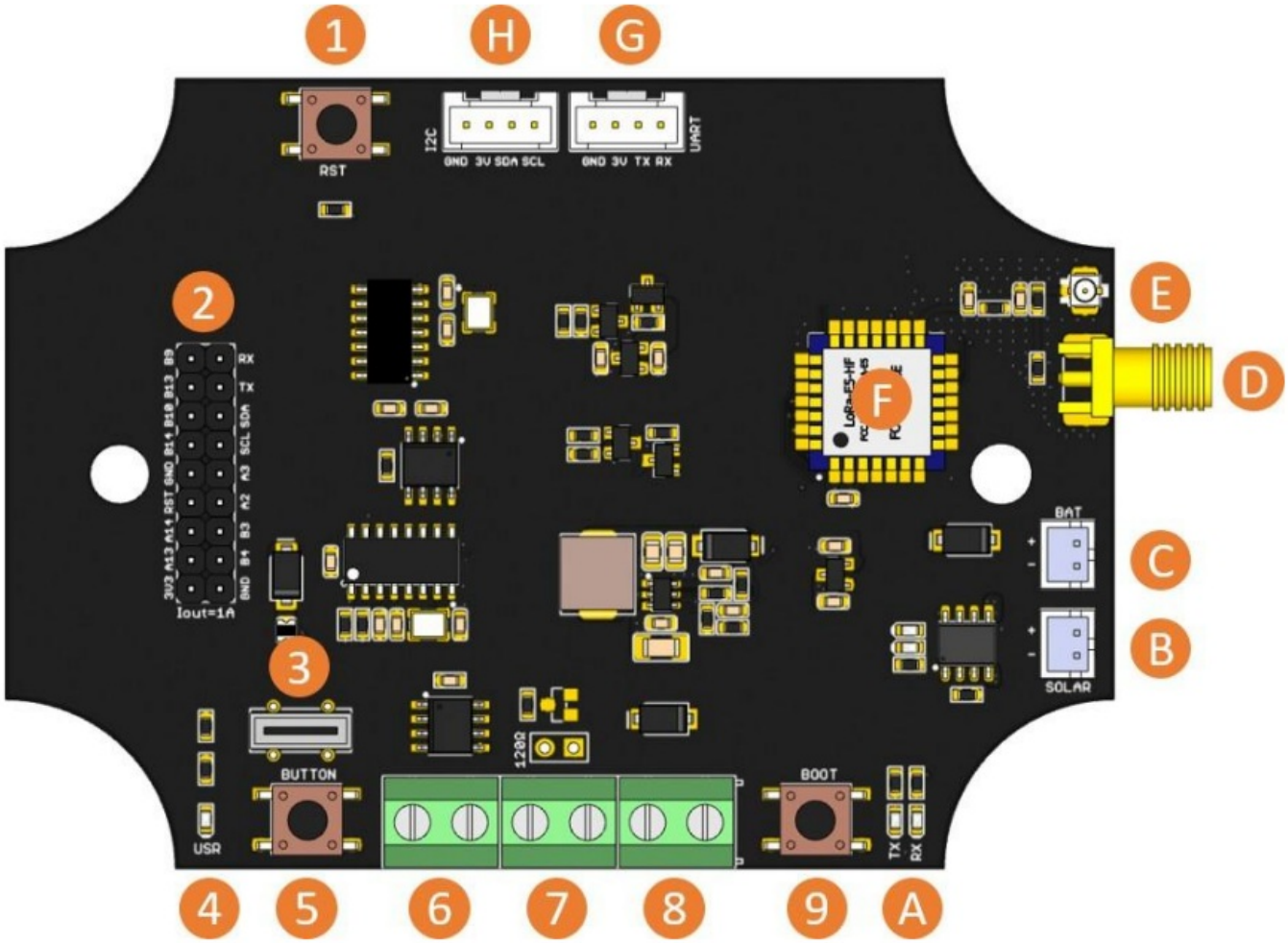
and power output up to +20.8dBm at 3.3V. Wio-E5 Dev Board also has rich interfaces. Developed to unlock the full functionality of the Wio-E5 module, Wio-E5 Dev Board has led out full 28 pins of Wio-E5 and provides with rich interfaces including Grove connectors, RS-485 terminal, male/female pin headers for you to connect sensors and modules with different connectors and data protocols, saving your time on wire soldering. You could also easily power the board by connecting the battery holder with 2 AA batteries, enabling temporary use when lacking an external power source. It is a userfriendly board for easy testing and rapid prototyping. Since Wio-E5 is a LoRaWAN chip with an MCU, there are three main ways to utilize the Wio-E5 Dev Board:

1. Connect Wio-E5 Dev Board to PC via USB and control by AT commands There is a built-in USB to UART function on board, you could just simply connect the Wio-E5 Dev Board to your PC with a USB type C cable, and use serial communication software to send AT commands and read data from the board
2. Connect Wio-E5 Dev Board to another mainboard via UART and control by AT commands For example, connect Wio-E5 Dev Board to Seeeduino XIAO and the Expansion Board via UART, and send AT commands and read data from Seeeduino XIAO through Arduino IDE serial monitor.
3. Develop user application by using SDK Develop your own Long Range development board with MCU function by using STM32Cube Programmer, which is the SDK officially provided by STMicroelectronics. To download this SDK resource, please find the resources in learning and document down below. With all the outstanding features listed above, the Wio-E5 Dev Board will be a superior choice for IoT device development, testing, prototyping, and applications in long-distance, ultra-low power consumption IoT scenarios like smart agriculture, smart office, and smart industry. If you are unfamiliar with Long Range and LoRaWAN technology, please check out this blog LoRapedia in detail.

Features

- Ultra-low power consumption and high performance
- Easy testing and rapid prototyping
- CAN FD communication
- RS485 Interface
- Global LoRaWAN and Long Range frequency plan supported
- Long-distance transmission range to 10km(ideal value in open area)

Hardware Overview



1. Reset button
2. 2×9 Header
3. Type-C USB
4. User LED
5. User Button
6. 485 Interface
7. CAN Interface
8. Power Input (5-28V)
9. Boot button

   A. CAN send/recv indicators

   B. Solar input

   C. Lipo battery input

   D/E. Antenna

   F. Wio E5 Module G/H.

   Grove connector

Specification

Item	Specifications
Voltage supply	5V/USB, 3.7V Lipo Battery, 4.5~28V DC input
Voltage output	3.3V
Power output	Up to 20.8dBm at 3.3V
Frequency	EU868 / US915 / AU915 / AS923 / KR920 / IN865
Protocol	LoRaWAN
Sensitivity	116.5dBm ~ -136dBm
Modulation	LoRa, (G)FSK, (G)MSK, BPSK
CAN 2.0 speed	Up to 1Mb/s
CAN FD speed	Up to 5Mb/s

Application

- A common Lora usage scenario, that is, building a Long Range sensor network. A large number of sensors in the industry are transmitted through CAN Bus. With the Wio-E5 CAN Bus Dev board you can use the on-board CAN Bus function to read the sensors, and send the data through Long Range .
- Since the circuit board can receive 5-28V input, the user can connect the circuit board to the OBD interface, get the data of the car and send it out through Long Range

Application Notes

1. **Factroy AT Firmare**

   Wio-E5 series has a built-in AT command firmware, which supports LoRaWAN Class A/B/C protocol and a wide frequency plan: EU868/US915/AU915/AS923/KR920/IN865. With this AT command firmware, developers can easily and quickly build their prototype or application. The AT command firmware contains a bootloader for DFU and the AT application. The “PB13/SPI_SCK/BOOT” pin is used to control Wio-E5 to stay in the bootloader or jump to the AT application. When PB13 is HIGH, the module will jump to AT application after reset, with a default baud rate of 9600. When PB13 is LOW (press the “Boot” button on Wio-E5 Dev Board or Wio-E5 mini), the module will stay in the bootloader, and keep transmitting “C” character every 1S at baud rate 115200.

   Attention

   Factory AT Firmware is programmed with RDP(Read Protection) Level 1, developers need to remove RDP first with STM32Cube Programmer. Note that regression RDP to level 0 will cause a flash memory mass to erase and the Factory AT Firmware can’t be restored again.

   The “PB13/SPI_SCK/BOOT” pin on the Wio-E5 module is just a normal GPIO, not the “BOOT0” pin of the MCU. This “PB13/SPI_SCK/BOOT” pin is used in the bootloader of the

   Factory AT firmware, to decide to jump to APP or stay in bootloader(for DFU). The real “BOOT0” pin doesn’t pinout to the module, so users need to be careful when developing the low-power applications.
2. **Clock Configuration**

   2.1 HSE

   32MHz TCXO

TCXO power supply: PB0-VDD_TCXO

2.2 LSE

32.768KHz crystal oscillator

3. RF Switch

   Wio-E5 module ONLY transmits through RFO_HP:

   Receive: PA4=1, PA5=0

   Transmit(high output power, SMPS mode): PA4=0, PA5=1

Getting Started

1. Quick start of AT Commands

   1.1 Preparation

   Step 1. Connect Wio-E5 Development Board to PC via a Type-C cable

   Step 2. Open a serial tool(eg. Arduino Serial Monitor), select the right COM port, set baudrate to 9600 and select Both NL & CR

   Step 3. Try to send “AT” and you will see the response.

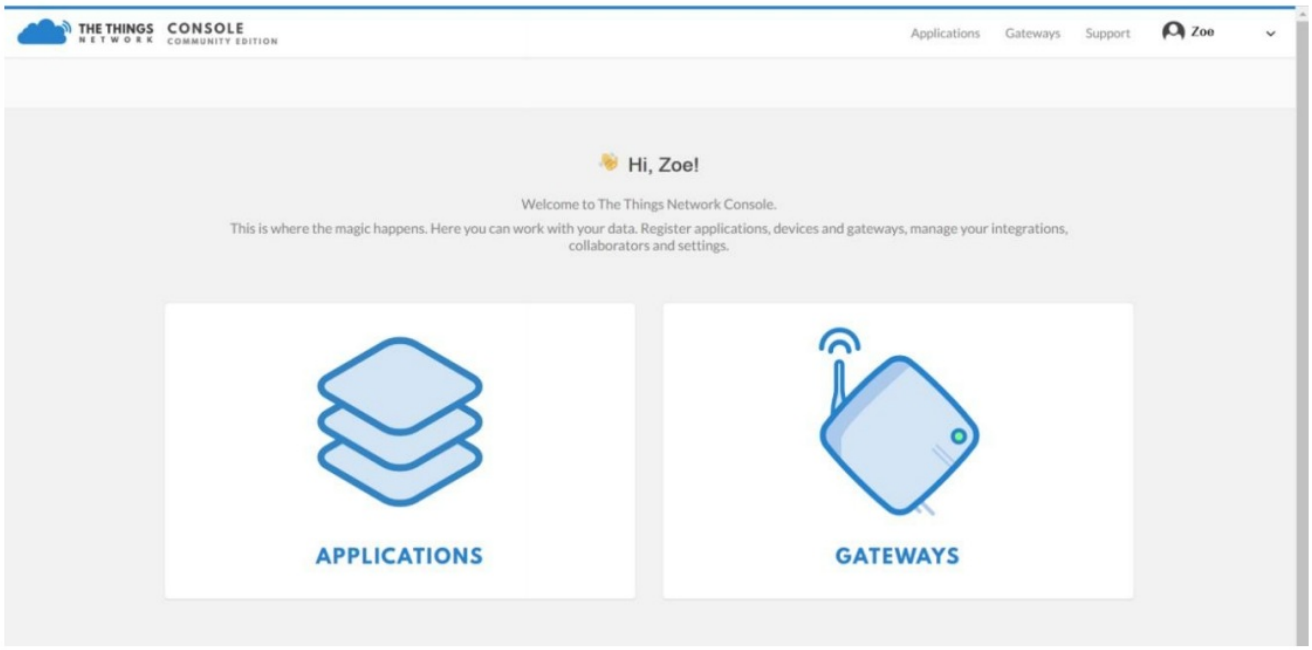
1.2 Basic AT Commands

- AT+ID // Read all, DevAddr(ABP), DevEui(OTAA), AppEui(OTAA)
- AT+ID=DevAddr // Read DevAddr
- AT+ID=DevEui // Read DevEui
- AT+ID=AppEui // Read AppEui
- AT+ID=DevAddr,”devaddr” // Set new DevAddr
- AT+ID=DevEui,”deveui” // Set new DevEui
- AT+ID=AppEui,”appeui” // Set new AppEui
- AT+KEY=APPKEY,”16 bytes length key” // Change application session key
- AT+DR=band // Change the Band Plans
- AT+DR=SCHEME // Check current band
- AT+CH=NUM, 0-7 // Enable channel 0~7
- AT+MODE=”mode” // Select work mode: LWOTAA, LWABP or TEST

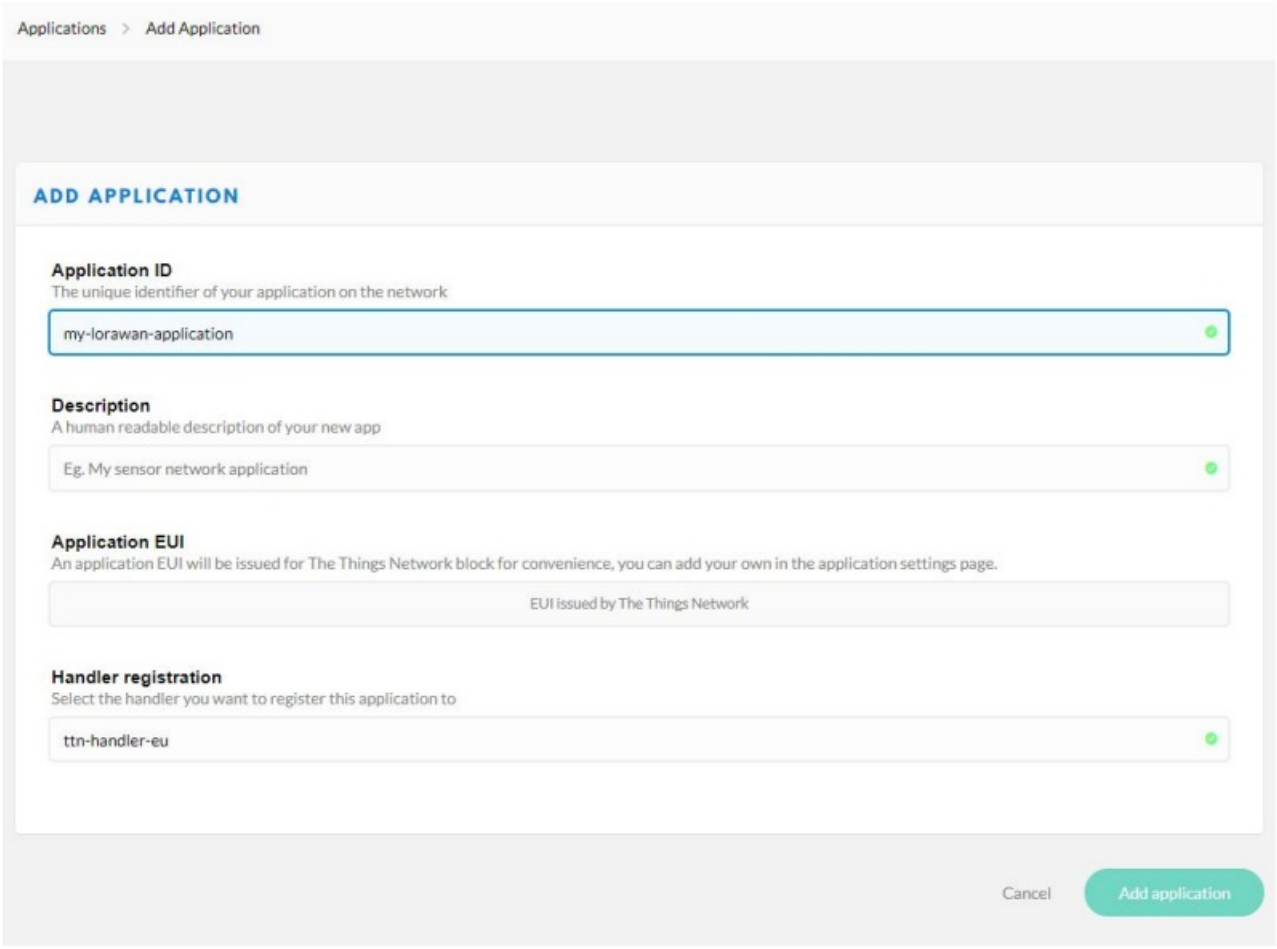
- AT+JOIN // Send JOIN request
- AT+MSG=”Data to send” // Use to send string format frame which is no need to be confirmed by the server
- AT+CMSG=”Data to send” // Use to send string format frame which must be confirmed by the server
- AT+MSGHEX=”xx xx xx xx” // Use to send hex format frame which is no need to be confirmed by the server
- AT+CMSGHEX=”xx xx xx xx” // Use to send hex format frame which must be confirmed by the server

1.3 Connect and send Data to TTN

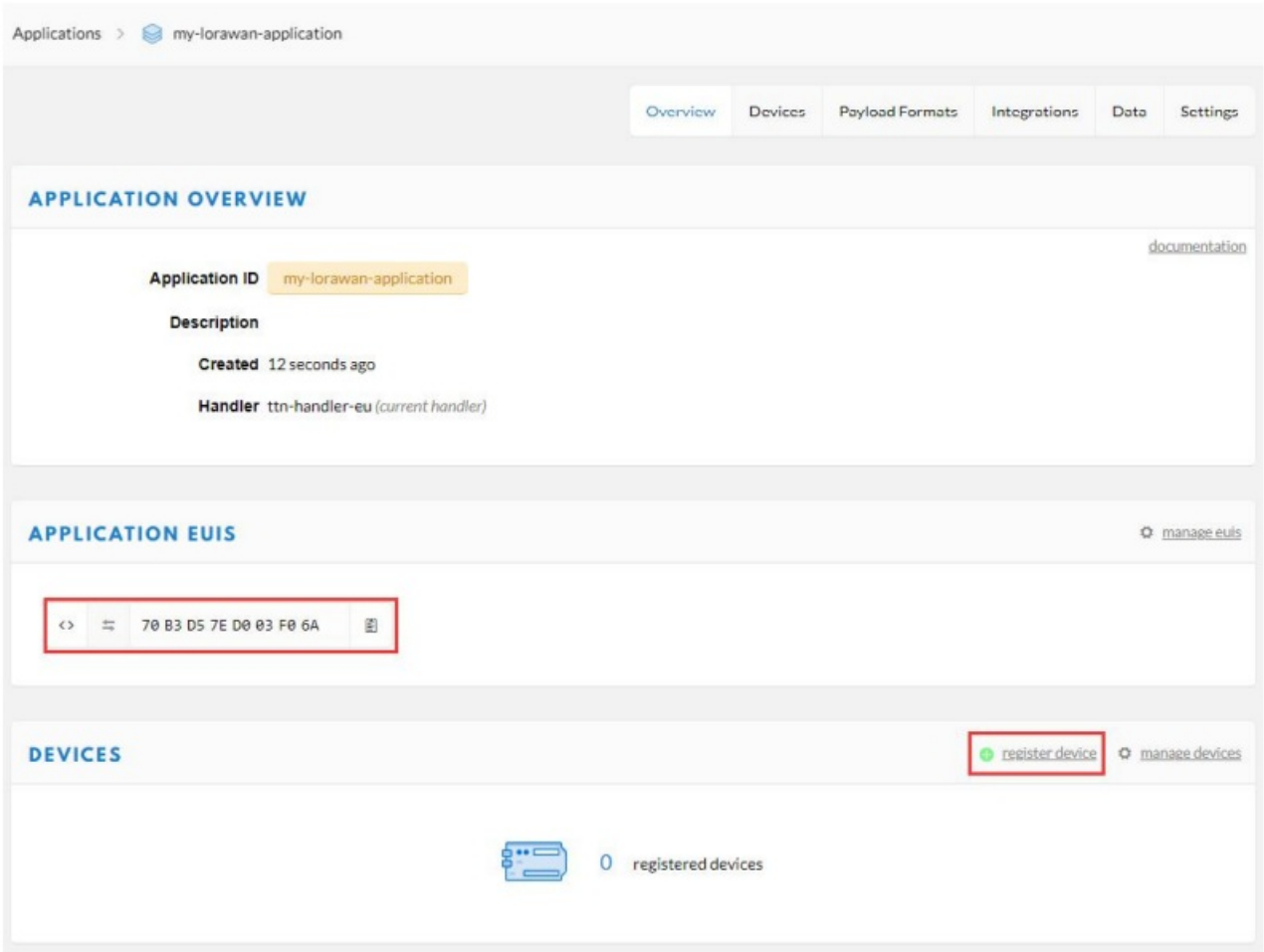
Step 1: Load into TTN website: **https://www.thethingsnetwork.org** and create your account, then access “Console” and first click on “APPLICATIONS”



Step 2: Add an Application



- Step3: Copy APPLICATION EUIS the to TTN and click “register device” button to add your device



- Step4: Send AT command AT + I D = DevEui to get your Device EUI, send AT command AT + KEY = APPKEY , ” 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 0 0 1 1 2 2 3 3 4 4 5 5 6 6 ” to set the App Key, and

send AT command

finally fill all these EUIs and Key to the page to register your device

to set the App EUI,

AT + I D = A p p E u i , ” A P P L I C A T I O N E U I S y o u c o p y j u s t now”

Tx : A T+ID=D evEui

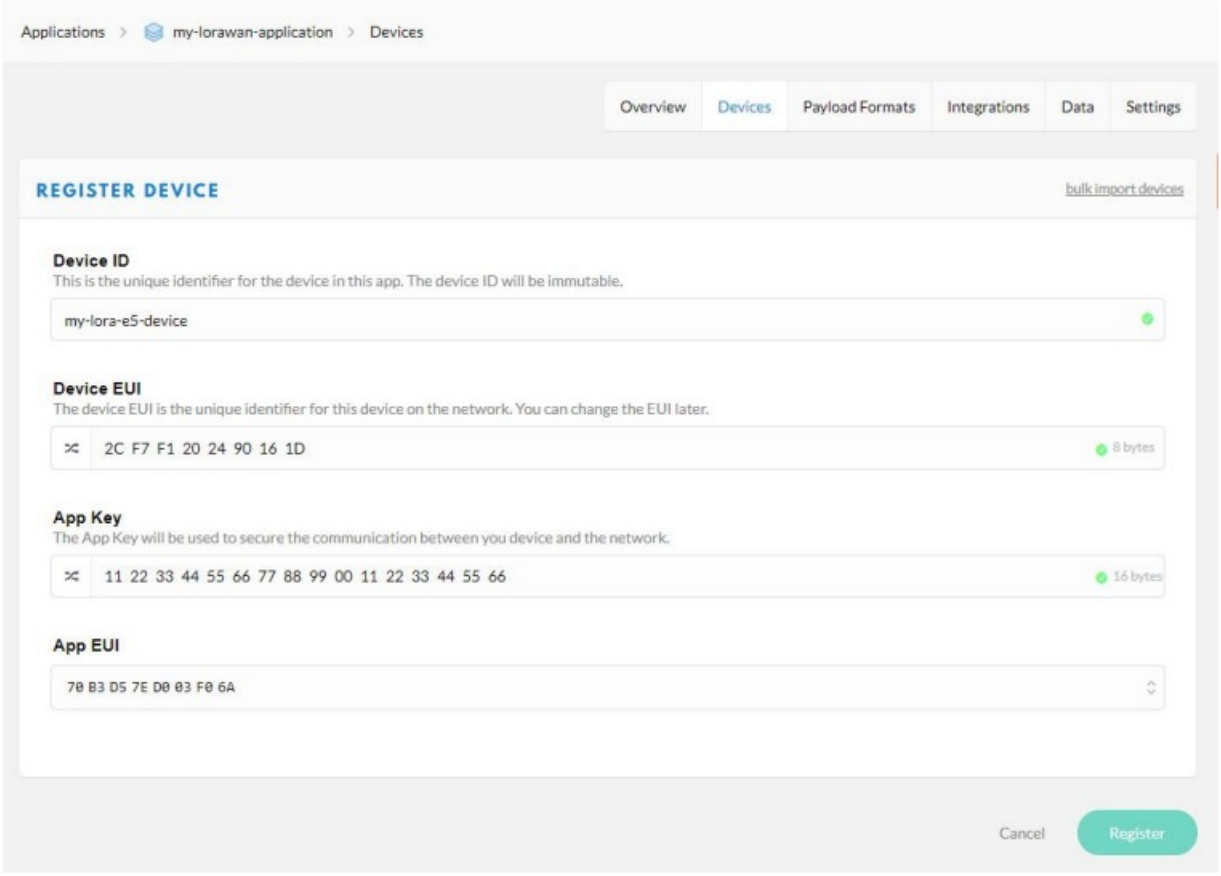
Rx : + I D : D e v E u i , 2 C:F7:F 1:20:2 4:90:1 6:1D

Tx : A T + KEY = A P P K E Y , ” 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 0 0 1 1 2 2 3 3 4 4 5 5 6 6 ”

Rx : + KEY : A P P K E Y 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 0 0 1 1 2 2 3 3 4 4 5 5 6 6

Tx : A T + ID = A p p E u i , ” 7 0 B 3 D 5 7 E D 0 0 3 F 0 6 A ”

Rx : + ID : A p p E u i , 7 0:B3:D 5:7E:D 0:03:F 0:6A



- Step 5: Register your LoRaWAN Gateway 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 6: Type the following AT Commmand to connect to TTN

  T x : AT +ID

  R x : + I D : D e v A d d r , 24 :90:16 :1D

  + I D : D e v E u i , 2C :F7:F1 :20:24 :90:16 :1D

  + I D : A p p E u i , 70 :B3:D5 :7E:D0 :03:F0 :6A

  T x : A T + D R = E U 8 6 8

  R x : + D R : E U 8 6 8

  T x : A T + C H = N U M , 0 – 2

  R x : + C H : N U M , 0 -2

  / / I f y o u a r e u s i n g U S 9 1 5 FSB2

  / / T x : AT +DR=US 915

  / / R x : + D R : US 915

  / / T x : A T + C H = N U M , 8 -15

/ / R x : + C H : N U M , 8 – 15

T x : A T + M O D E = L W O T A A

R x : + M O D E : L W O T A A

T x : AT +JOIN

R x : + JO I N : St art

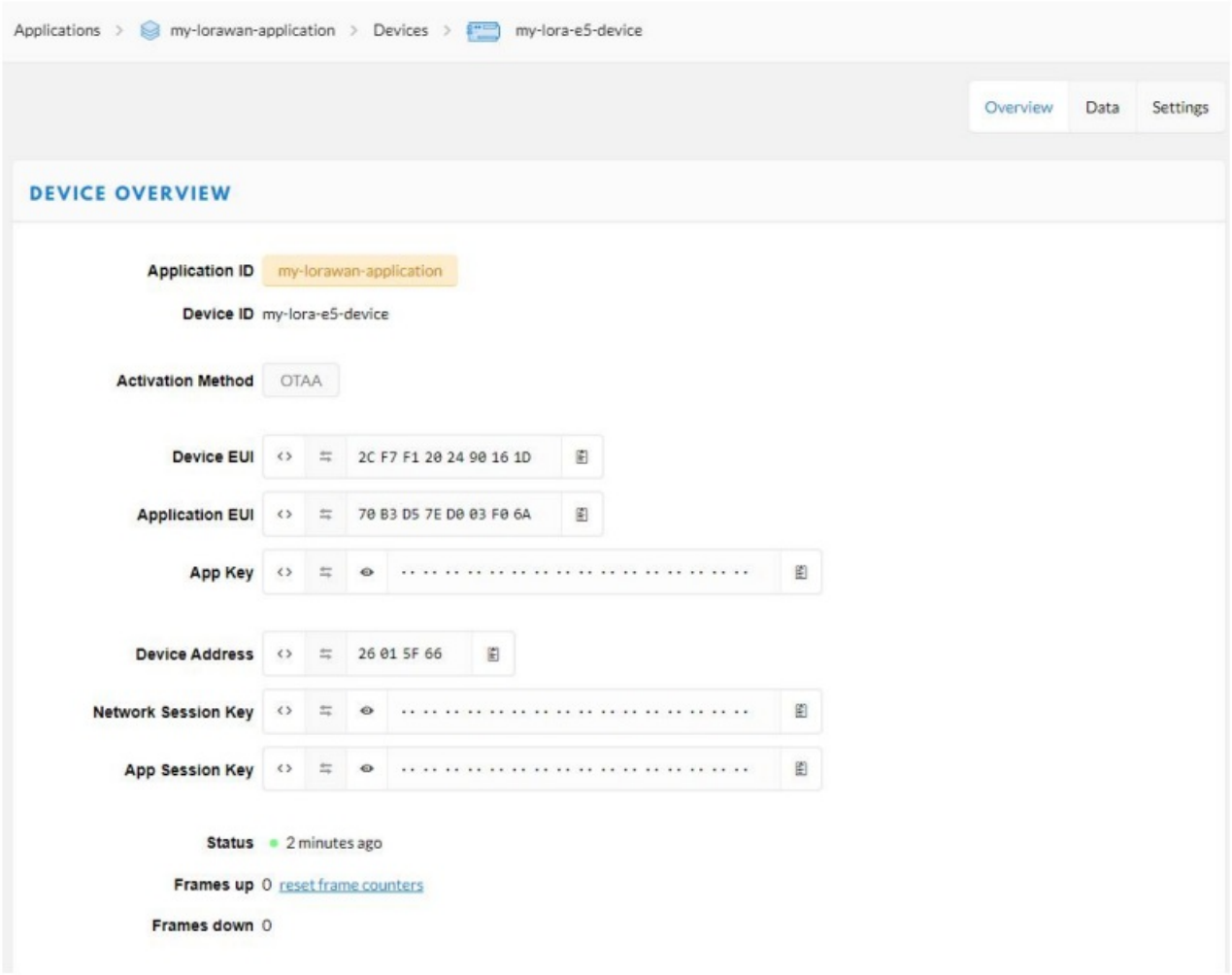
+ J OI N : NO RMAL

+ J OI N : N e t w o r k jo ined

+ J O I N : N e t I D 0 0 0 0 1 3 D e v A d d r 26 :01:5F :66

+ J O I N : Done

If you see in your serial console, congratulations, your device have + J O I N : N e t w o r k j oined already connect to TTN! You can also check your device status at the “overview” page.



- Step 7: Type the following AT Command to send data to TTN

  / / s e n d s t r i n g ” H E L L O ” t o T T N

  T x : A T + M S G = H E L L O

  R x : + M S G : St art

  + M S G : FP ENDING

  + M S G : R X W I N 2 , R S S I – 1 1 2 , S N R – 1.0

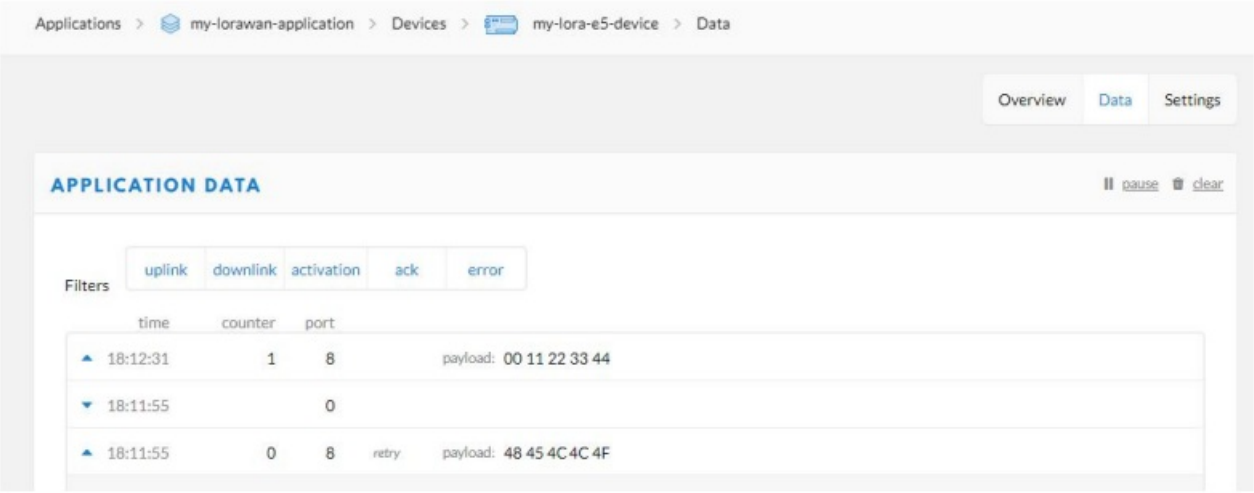
  + M S G : Done

  / / s e n d h e x ” 0 0 1 1 2 2 3 3 44″

  T x : A T + M S G H E X = ” 0 0 1 1 2 2 3 3 44″

  R x : + M S G H E X : St art

  + M S G H E X : Done



- Step 8: For more information about AT Commands, please refer to Wio-E5 AT Command Specification

2. Develop with STM32Cube MCU Package

This section is for Wio-E5 Mini or Wio-E5 Dev Board, aiming at creating a LoRaWAN End Node with STM32Cube MCU Package for STM32WL series(SDK), to join and send data to LoRaWAN Network.

Attention

Please read Erase Factory AT Firmware section first, as if we need to erase the Factory AT Firmware before we program with SDK. After erasing the Factory AT Firmware it CANNOT be recovered.

2.1 Preparations

Softwares:

- Install STM32CubeIDE(to compilation and debug)
- Install STM32CubeProgrammer(to program STM32 devices)
- Download and extract STM32Cube MCU Package for STM32WL series(SDK)

Hardwares:

- LoRaWAN Gateway connected to LoRaWAN Network Server(e.g. TTN)
- Prepare an USB TypeC cable and a ST-LINK. Connect the TypeC cable to the TypeC port for
- power and serial communication, connect the ST-LINK to the SWD pins like this:

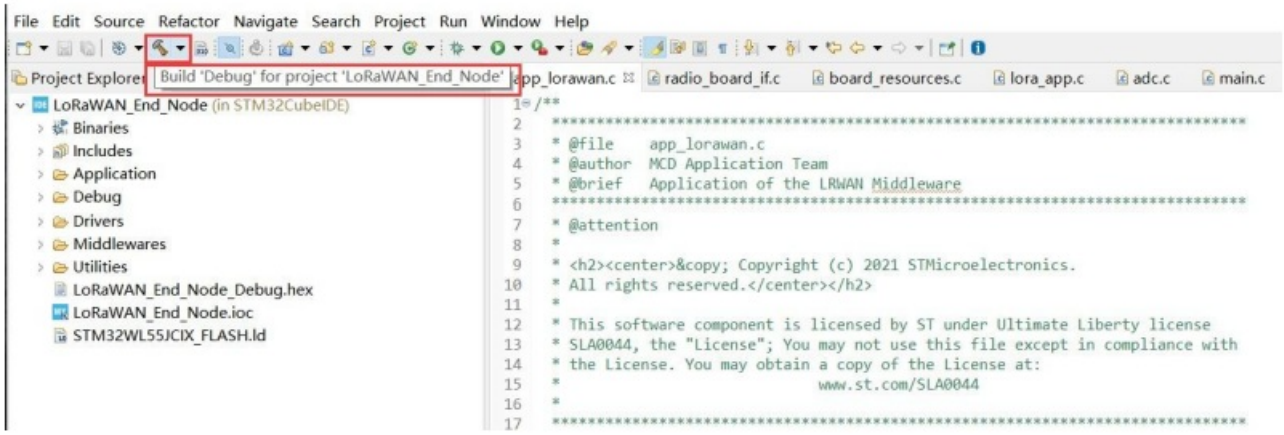
2.2 GPIO Configuration Overview

As the hardware design of Wio-E5 series is a bit different with NUCLEO-WL55JC, the official STM32WL55JC development board from ST, developers need to reconfigure some gpios, to adapt the SDK example to Wio-E5 series. We have already reconfigured gpios in this example, but we think it is nessary to point out the difference.

SDK Example Label	GPIO of NUCLEOWL55JC	GPIO of Wio-E5 Mini and Wio-E5 Dev Boa rd
RF_CTRL1	PC4	PA4
RF_CTRL2	PC5	PA5
RF_CTRL3	PC3	None
BUT1	PA0	PB13 (Boot Button)
BUT2	PA1	None
BUT3	PC6	None
LED1	PB15	None
LED2	PB9	PB5
LED3	PB11	None
DBG1	PB12	PA0 (D0 Button)
DBG2	PB13	PB10
DBG3	PB14	PB3
DBG4	PB10	PB4
Usart	Usart2(PA2/PA3)	Usart1(PB6/PB7)

2.3 Build the LoRaWAN End Node Example

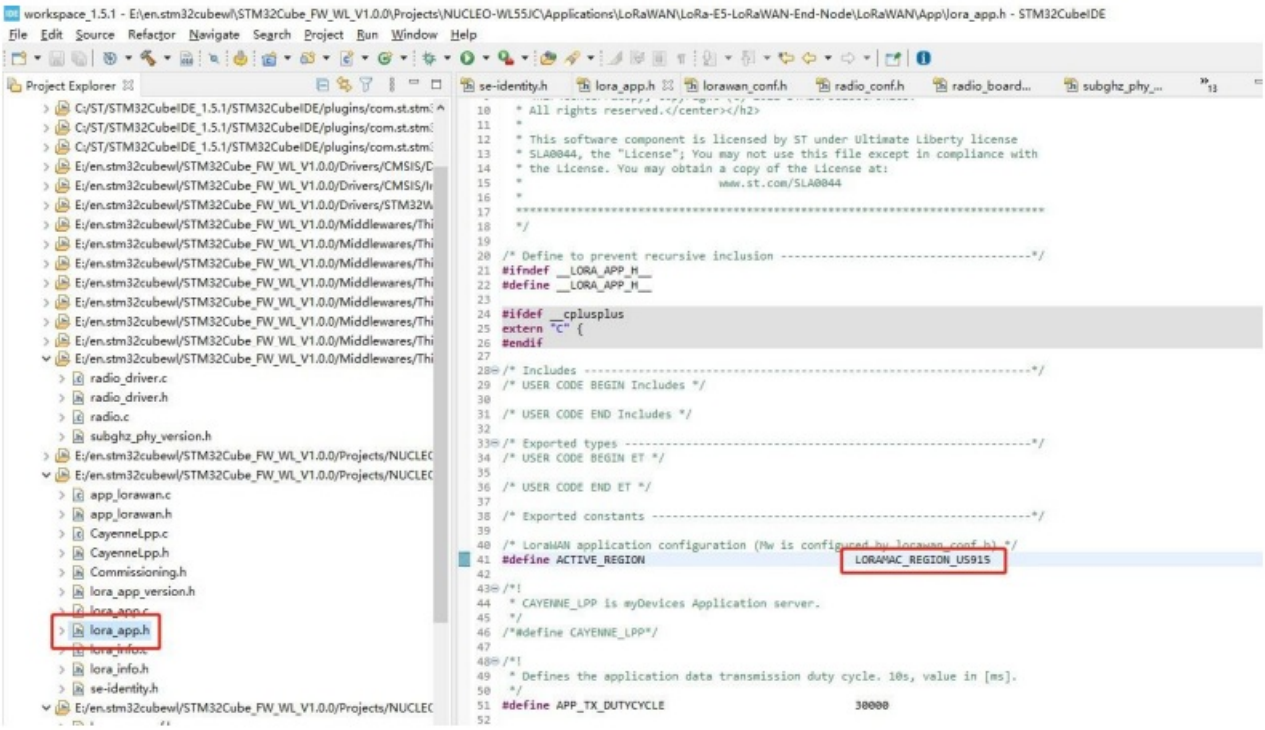
- Download and copy this repo to your SDK folder en. stm32c ubewl \STM32 Cube_F W_WL_V 1.0.0 \Proje cts\NU CLEOW L 5 5 J C \ A p p l i c a t i o n s \ L o R a W A N and replace the origin en. stm32c ubewl \STM32 Cube_F W_WL_V 1.0.0 \Proje cts\NU CLEO- WL5 5JC\Ap plica tions\ LoRaWA N\LoRa WAN_E nd_Nod e folder
- Open the L o R a W A N _ E n d _ N o d e example with S T M 3 2 C u b e I D E , by double click file LoR aWAN_E nd_No de\STM 32Cube IDE\.p rojec t
- Click for this example, it should work without any errors BuildD ebug



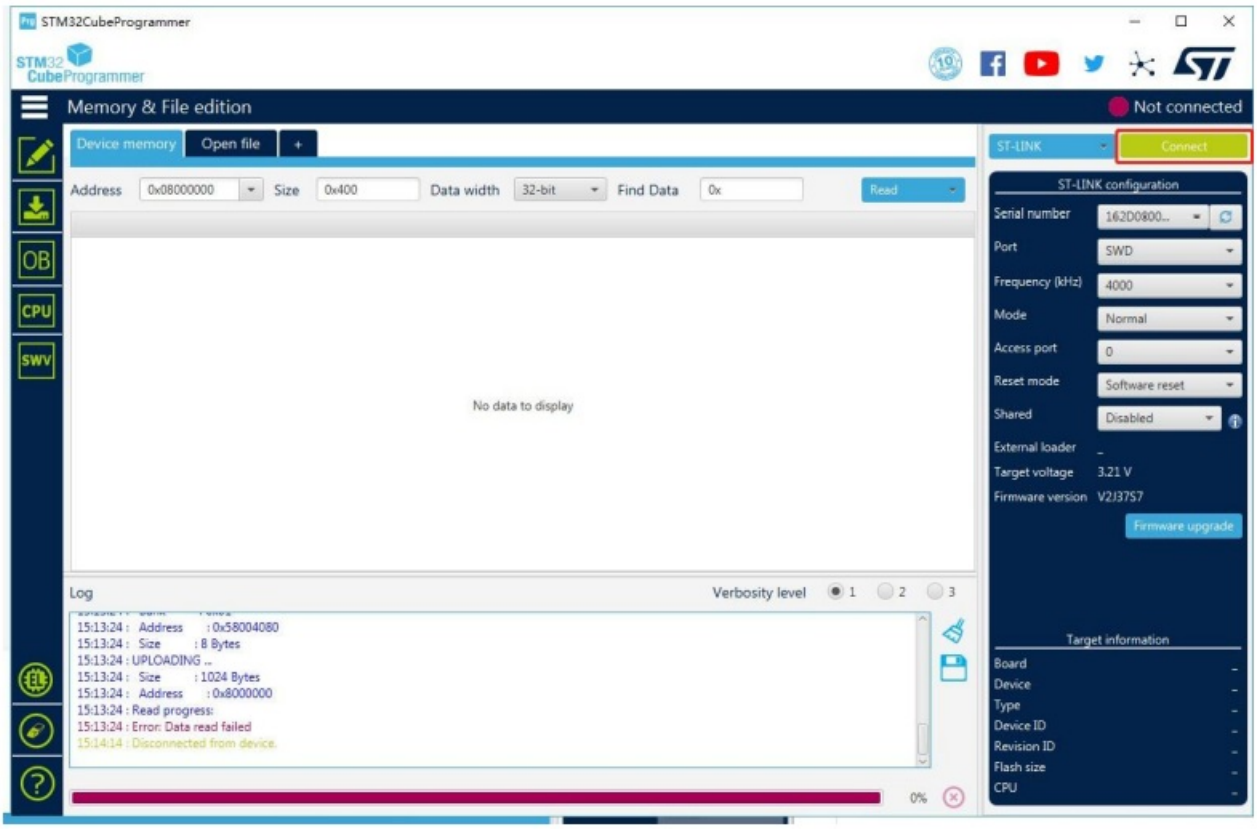
2.4 Modify your Device EUI, Application EUI, Application KEY and your LoRawan Region

Please follow the guide here to setup your TTN application, get your Application EUI and copy it to the macro definition L O R A W A N _ J O I N _ E U I in example, my Application EUI is

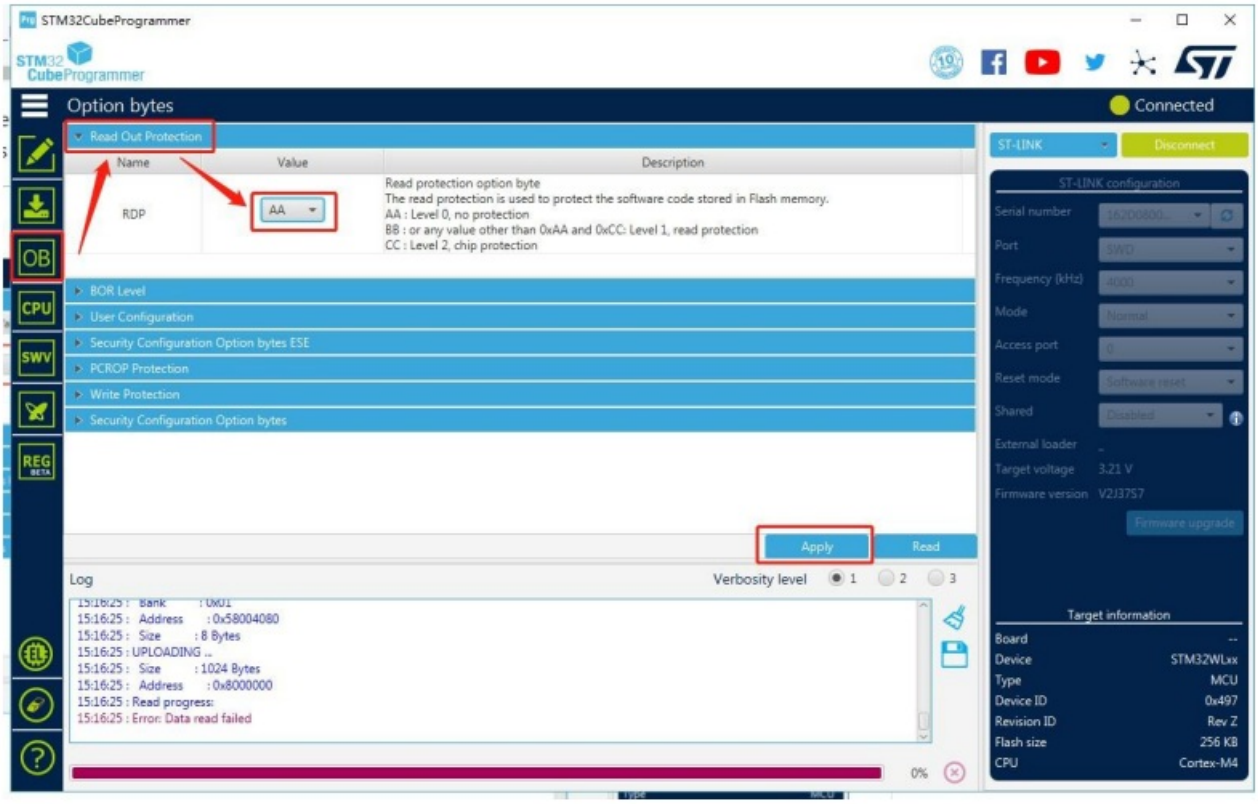
L o R a W A N / A p p / s e – i dentity.h , for 7 0 B 3 D 5 7 E D 0 0 3 F 0 6A : / / L o R a W A N / A p p / s e -id entity .h /*! * App / J o i n s e r v e r I E E E E U I (b i g en dian) */ # d e f i n e LO RAWAN_ JOIN_E UI 0 x D 5 , 0 x 7 E , 0 x D 0 , 0 x 0 3 , 0 x F 0 , 0 6 A } { 0 x 7 0 , 0 xB3, Also, you can modify your Device EUI and Application Key, by setting the macro definition LORAWAN _ DEVICE _ EUI and LORAWAN _ N W K _ K E Y in L o R a W A N / A p p / s e – i dentit y.h , don't L ORAWA N_DEVI CE_EUI forget to ensure andand in TTN console. are the same as the LORAWA N_NWK_ KEY Device EUI App Key
/ / L o R a W A N / A p p / s e -id entity .h /*! * end – dev ice IEEEE I (b i g en dian) */ # d e f i n e LO RAWAN_ DEVICE _EUI 0 x E 1 , 0 x 1 5 , 0 x 0 0 , 0 x 0 7 , 0 x 4 C , 0 x D 5 } { 0 x 0 0 , 0 x80, /*! * N e t w o r k r o o t key */ # d e f i n e L O R A W A N NWK _ K E Y 2B ,7E,15 ,16,28 ,AE,D2 ,A6,A B,F7,1 5,88,0 9,CF,4 F,3C The default LoRaWAN Region is E U 8 6 8 , you can modify it, by setting the macro definition in ACT IVE_RE GION L oRaWA N/App/ lora_a pp.h / / Lo RaWAN/ App/lo ra_app .h / * L o r a W A N a p p l i c a t i o n c o n f i g u r a t i o n (M w i s c o n f i g u r e d b y l orawa n_conf .h) */ / * A v a i l a b l e : L O R A M A C _ R E G I O N _ A S 9 2 3 , L O R A M A C _ R E G I O N _ A U 9 1 5 , L O R A M A C _ R E G I O N _ E U 8 6 8 , L O R A M A C _ R E G I O _ K R 9 2 0 , L O R A M A C _ R E G I O N _ I N 8 6 5 , L O R A M A C _ R E G I O N _ U S 9 1 5 , L O R A M A C _ R E G I O N _ R U 8 6 4 */ # d e f i n e AC TIVE_R EGION
L ORAMAC _REGI ON_EU8 68



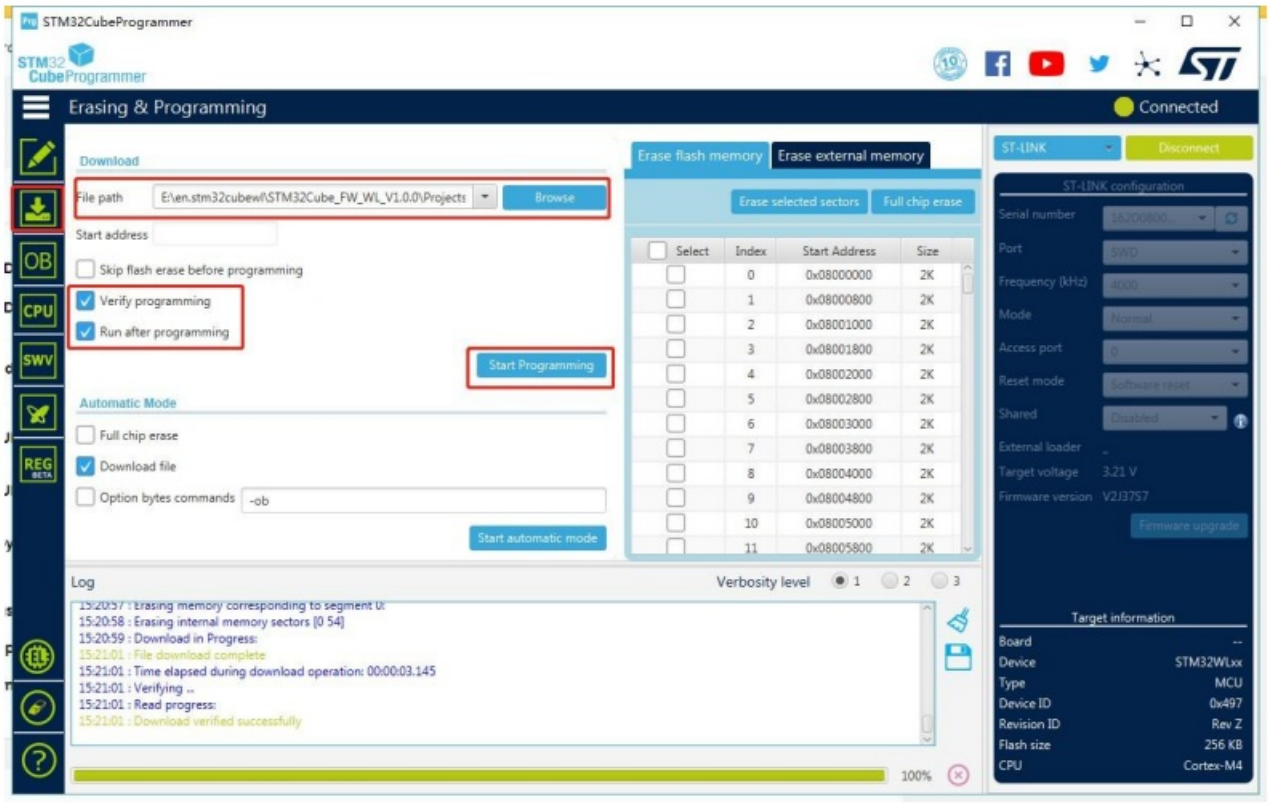
- After modification, please rebuild the example and program to your Wio-E5. Open S T M 3 2 Cube Programmer , connect ST-LINK to your PC, hold of your Device, RESET Button then click and release RESE T Butto n : Connect



- Make sure the Read Out Protection is AA , if it is shown as B B , select and click Apply :



- Now, go to the E r a s i n g & P r o g r a m m i n g page, select your hex file path(my path is E:\ en.stm 32cub ewl\ST M32Cub e_FW_W L_V1. 0.0\Pr ojects \NUCLE OWL5 5JC\Ap plica tions\ LoRaWA N\LoRa WAN_E nd_Nod e\STM3 2CubeI DE\Lo RaWAN_ End_Node_Debug . hex), select the programming options as the following picture, then click S tart P r o g r a m m i n g ! Once the prgramming is finished,



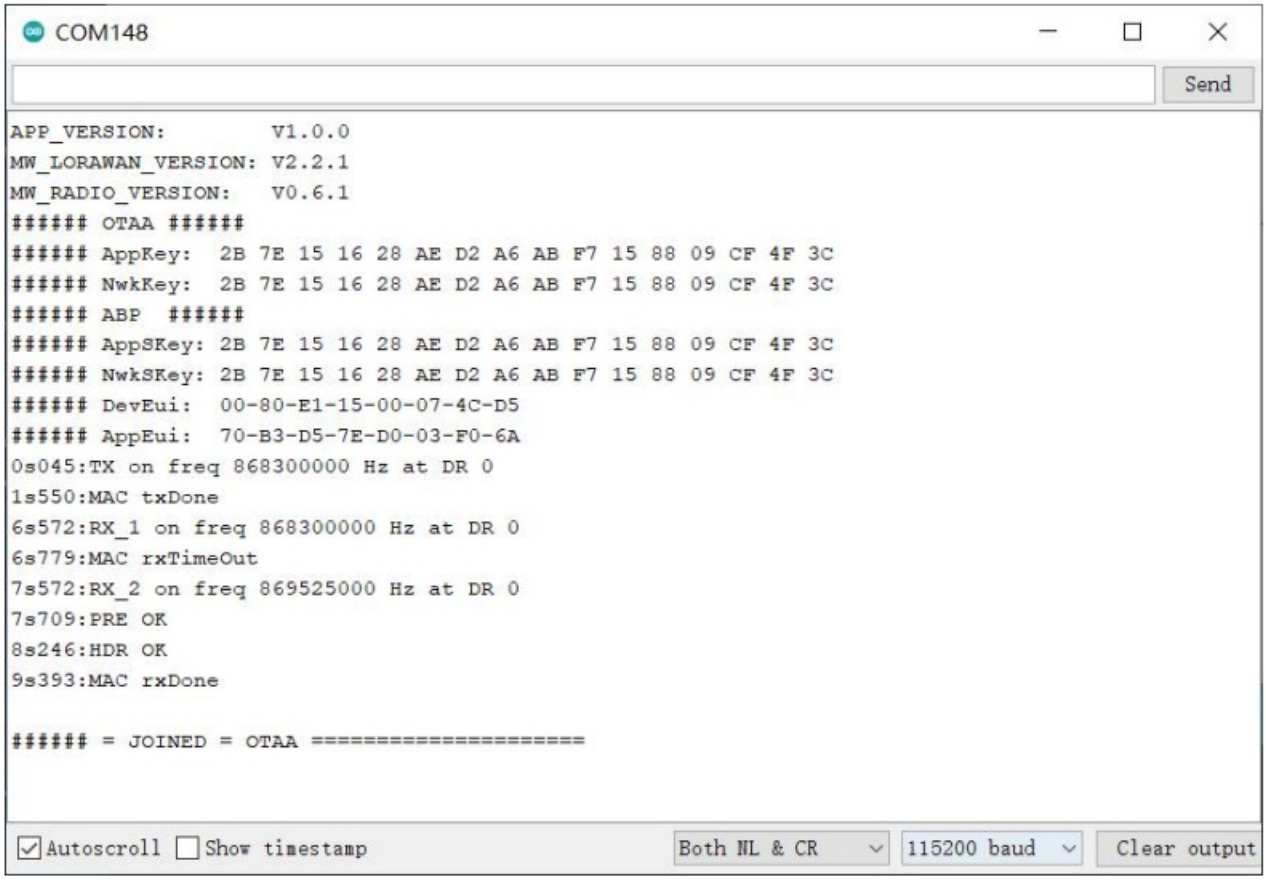
2.5 Connect to TTN

- If your LoRaWAN Gateway and TTN are setup, Wio-E5 will join successfully after reset! A comfirm LoRaWAN package will be sent to TTN every 30 seconds. The following log will come out from the serial port if the join is successful:

```
APP_VERSION:        V1.0.0
MW_LORAWAN_VERSION: V2.2.1
MW_RADIO_VERSION:   V0.6.1
###### OTAA ######
###### AppKey:  2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
###### NwkKey:  2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
###### ABP  ######
###### AppSKey: 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
###### NwkSKey: 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
###### DevEui:  00-80-E1-15-00-07-4C-D5
###### AppEui:  70-B3-D5-7E-D0-03-F0-6A
0s045:TX on freq 868100000 Hz at DR 0
1s550:MAC txDone
6s572:RX_1 on freq 868100000 Hz at DR 0
6s779:MAC rxTimeOut
7s572:RX_2 on freq 869525000 Hz at DR 0
7s709:PRE OK
8s246:HDR OK
9s393:MAC rxDone

###### = JOINED = OTAA ====================
30s068:temp= 25
30s068:VDDA= 254
30s069:TX on freq 868500000 Hz at DR 0
30s082:SEND REQUEST
31s728:MAC txDone
32s750:RX_1 on freq 868500000 Hz at DR 0
32s957:MAC rxTimeOut
33s706:RX_2 on freq 869525000 Hz at DR 3
33s744:PRE OK
33s815:HDR OK
33s897:MAC rxDone

###### ========== MCPS-Confirm =============
```



- Cheers! You have already connected Wio-E5 to LoRaWAN Network! Can’t wait to see you develop some wonderful LoRaWAN End Node applications!

Application Notes

- Wio-E5 only supports high power output mode, so you can’t use these macro defnitions in : rad io_boa rd_if .h # d e f i n e RB I_CONF _RFO / / or # d e f i n e RB I_CONF _RFO R BI_CO NF_RFO _LP_HP R BI_CO NF_RFO _LP
  More demos coming soon…

Resources
Wio-E5 CAN FD Development Board Datasheet:

- Schematics in PDF Format
- Schematics in Eagle Format
- CAN FD library
- MCP2517FD datasheet

  Wio-E5 Datasheet:
- Wio-E5 datasheet and specifications
- Wio-E5 AT Command Specification
- STM32WLE5JC Datasheet

  Wio-E5 Certifications:
- Wio-E5-HF Certification CE-VOC-RED
- Wio-E5-HF FCC Certification -DSS
- Wio-E5-HF FCC Certification -DTS

Relevant SDK:

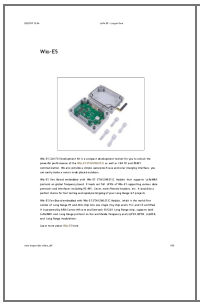
- STM32Cube MCU Package for STM32WL series

Tech Support

Please submit any technical issue into our forum.

docs.longan-labs.cc/lora_e5/

Documents / Resources

	seeed studio Wio-E5 CAN FD Development Kit [pdf] User Guide Wio-E5 CAN FD Development Kit, Wio-E5, CAN FD Development Kit, FD Development Kit, Development Kit

References

- LoRa E5 - Longan Docs
- Seeed Forum - Seeed Forum
- my.st.com
- my.st.com
- my.st.com
- Wio-E5 mini | Seeed Studio Wiki
- Wio-E5 mini | Seeed Studio Wiki
- Wio-E5 STM32WLE5JC Module | Seeed Studio Wiki
- The Things Indoor Gateway | Seeed Studio Wiki
- An introduction of LoRa® and LoRaWAN® technology
- Wio-E5 STM32WLE5JC lora module, embedded SX126X and MCU for LoRaWAN Wireless Sensor Network & IoT devices - EU868 & US915 - Seeed Studio
- The Things Network
- The Things Network