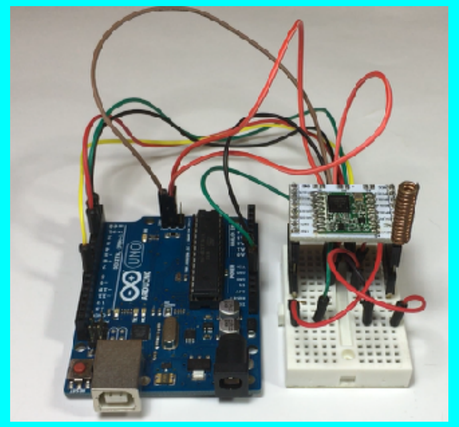
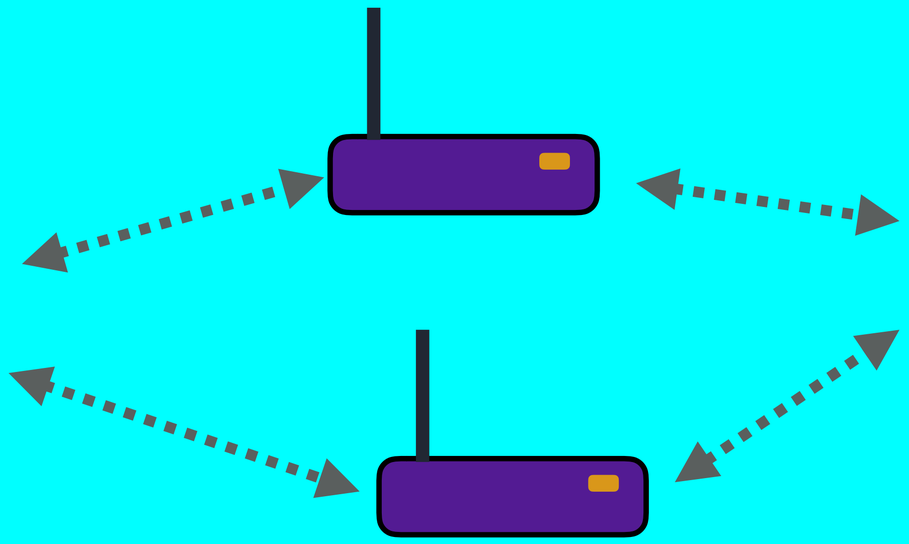
LORA / LORAWAN TUTORIAL 22

OTAA and Uplink Demonstration With The Things Network





THE THINGS NETWORK

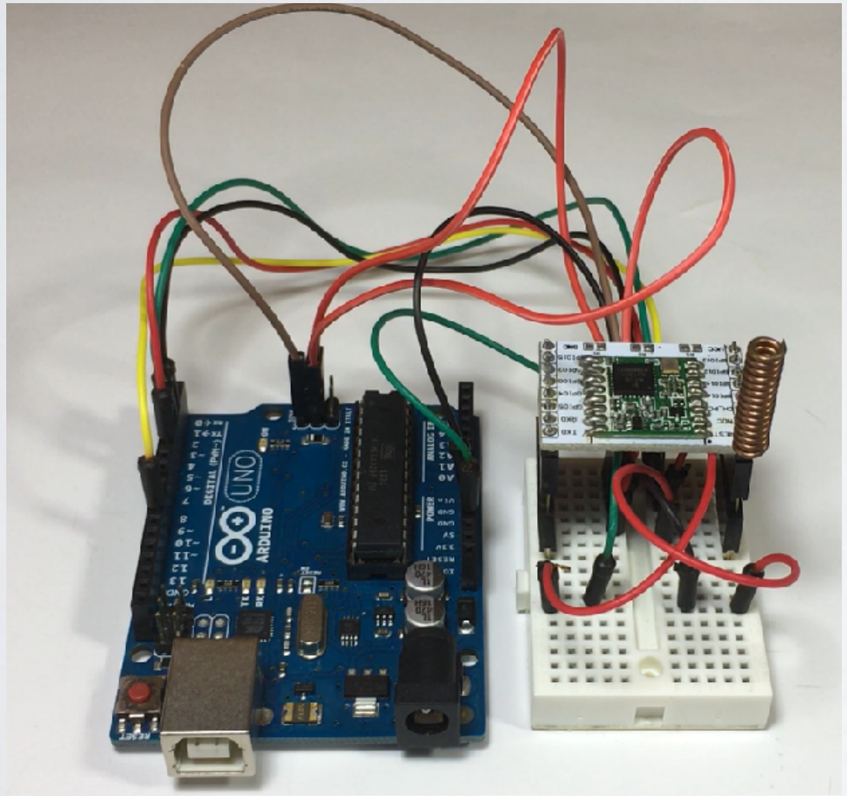
INTRO

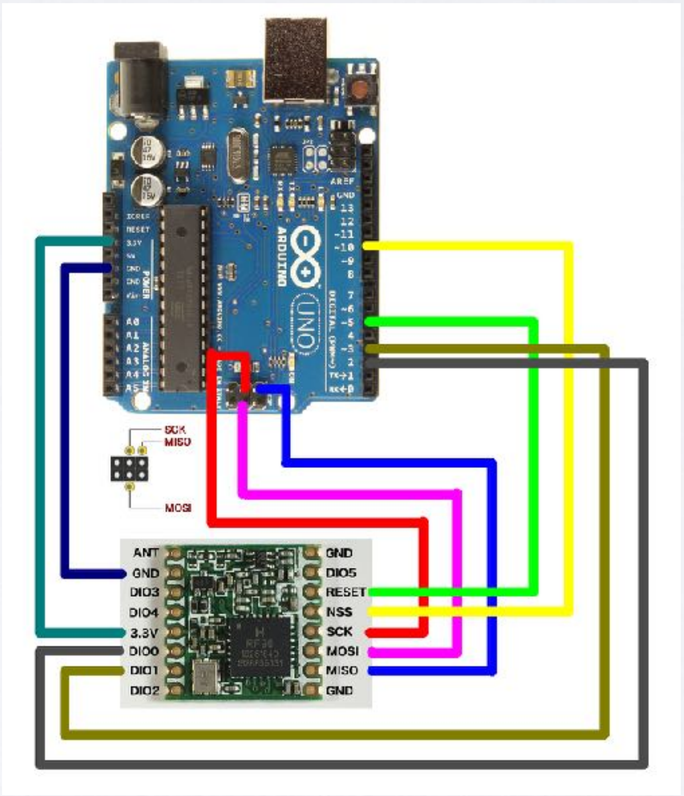
- In this tutorial I will demonstrate the Over-The-Air-Activation method.

- In the demonstration I will use my self build LoRa development board to send messages to The Things Network using the MCCI Arduino LMIC library.

DEMONSTRATION SETUP

- Self build LoRa development board, make sure an antenna is connected, see: https://www.mobilefish.com/developer/lorawan/lorawan_quickguide_build_lora_node_rfm95_arduino_uno.html





DEMONSTRATION

- The self build LoRa development board uses the following jumper wire connections between the HopeRF RFM95 LoRa transceiver module and the Arduino Uno.

HopeRF RFM95 LoRa transceiver module	Arduino Uno Pin		HopeRF RFM95 LoRa transceiver module	Arduino Uno Pin
ANT	-		GND	-
GND	GND		DIO5	-
DIO3	-		RESET	5
DIO4	-		NSS	10
3.3V	3.3V		SCK	13
DIO0	2		MOSI	11
DIO1	3		MISO	12
DIO2	-		GND	-

- Note: The HopeRF RFM95 LoRa transceiver module is compatible with SX1276.

DEMONSTRATION

- Make sure a LoRa gateway is in your area and your LoRa end device can send messages to that gateway. Use this map:
  https://www.thethingsnetwork.org/map

- Install the latest open source Arduino IDE. In this tutorial, version 1.8.7 is used.
  https://www.arduino.cc/en/Main/Software

- Now install the MCCI Arduino LMIC Library.
  - In the Arduino IDE, select menu Sketch | Include Library | Manage Libraries
  - In the search box enter: MCCI
  - Click the MCCI library.
  - Select the latest version and press the Install button.

DEMONSTRATION

- The Arduino libraries are installed on these default locations:
  Windows: C:\Users\<username>\Documents\Arduino\libraries
  OSX: /Users/<username>/Documents/Arduino/libraries
  Linux: /home/<username>/arduino/sketchbook/libraries

- Configure the MCCI Arduino LMIC Library according to your situation.
  Edit file lmic_project_config.h. This file can be found at:
  ../libraries/MCCI_LoRaWAN_LMIC_library/project_config

DEMONSTRATION

- The changes I made to MY lmic_project_config.h file.
  Make changes according to YOUR situation.

```
// project-specific definitions
#define CFG_eu868 1
//#define CFG_us915 1
//#define CFG_au921 1
//#define CFG_as923 1
// #define LMIC_COUNTRY_CODE LMIC_COUNTRY_CODE_JP  /* for as923-JP */
//#define CFG_in866 1

#define CFG_sx1276_radio 1
//#define CFG_sx1272_radio 1

#define DISABLE_PING
#define DISABLE_BEACONS
#define LMIC_DEBUG_LEVEL 0
#define USE_IDEETRON_AES
```

DEMONSTRATION

- The Arduino Uno (ATmega328) has 32kBytes of flash memory for the bootloader + uploaded sketch. It is important to use these settings:

```
#define DISABLE_PING
#define DISABLE_BEACONS
#define LMIC_DEBUG_LEVEL 0
#define USE_IDEETRON_AES
```

  ....otherwise the compiled sketch does fits the Arduino Uno flash memory.

- There are more configuration settings which can be overridden by the lmic_project_config.h file, see:
  …/libraries/MCCI_LoRaWAN_LMIC_library/src/lmic/config.h
  But in this demonstration I do not override any of these other settings.

DEMONSTRATION

- Create an account on The Things Network (TTN)
  https://www.thethingsnetwork.org/

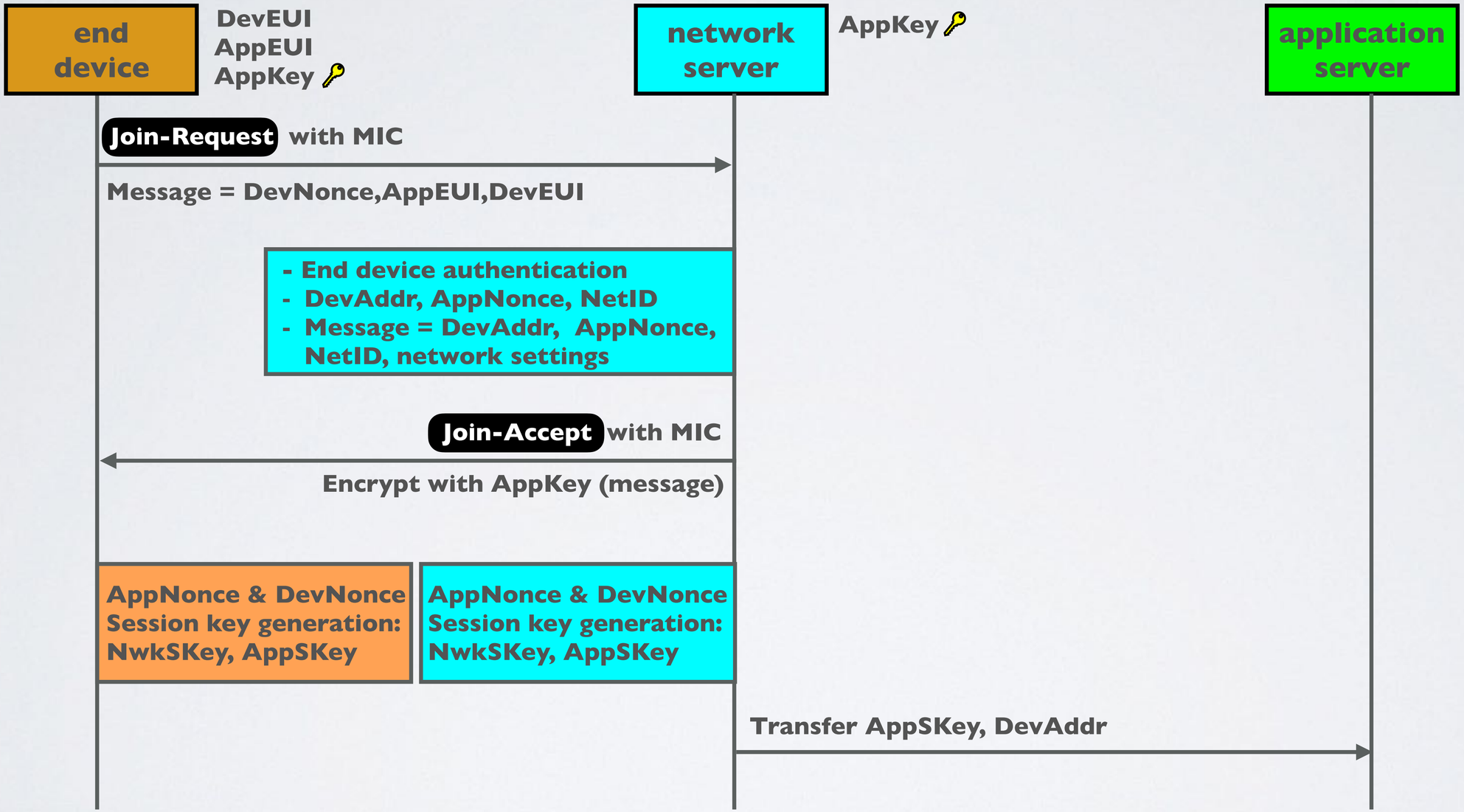
- On TTN, add an application:
  Application ID: youtube_demo_app
  Description: My YouTube LoRawan demo application

- On TTN, register a device:
  Device ID: youtube_demo_device
  Device EUI: F9C01FAA68E1D265 (Enter a **random value** consisting of 8 bytes)
  You can use this tool to create the device EUI:
  https://www.mobilefish.com/services/guid/guid.php

DEMONSTRATION

- Make sure the activation method is OTAA.
  The App Key is generated.

OTAA


end device
DevEUI
AppEUI
AppKey
network server
AppKey
application server
Join-Request with MIC
Message = DevNonce,AppEUI,DevEUI
- End device authentication
- DevAddr, AppNonce, NetID
- Message = DevAddr, AppNonce, NetID, network settings
Join-Accept with MIC
Encrypt with AppKey (message)
AppNonce & DevNonce
Session key generation:
NwkSKey, AppSKey
AppNonce & DevNonce
Session key generation:
NwkSKey, AppSKey
Transfer AppSKey, DevAddr

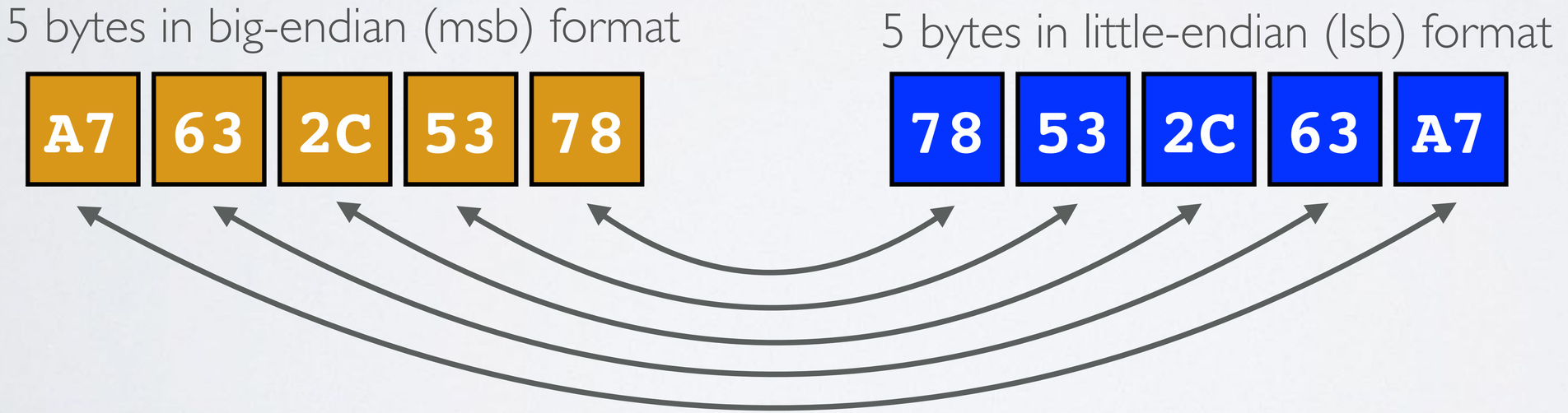
DEMONSTRATION

- In the Arduino IDE, select menu File | Examples | MCC LoRaWAN LMIC library and select the ttn-otaa sketch. Re-save the ttn-otaa sketch and call it ttn-otaa-mydemo.

- From The Things Network console copy YOUR DevEUI, AppEUI and AppKey to the ttn-otaa-mydemo sketch.
  Watch out:
  The DevEUI and AppEUI must be in little-endian format.
  The AppKey must be in big endian format.

BIG-ENDIAN VS LITTLE-ENDIAN FORMAT

- Big-endian and little-endian are terms that describe the order in which a sequence of bytes are stored in computer memory.
- Big-endian is an order in which the **m**ost **s**ignificant **b**it (msb) is stored first.
- Little-endian is an order in which the **l**east **s**ignificant **b**it (lsb) is stored first.



DEMONSTRATION

- Make the additional changes to the ttn-otaa-mydemo sketch.

```
// Pin mapping
const lmic_pinmap lmic_pins = {
  .nss = 10,
  .rxtx = LMIC_UNUSED_PIN,
  .rst = 5,
  .dio = {2, 3, LMIC_UNUSED_PIN},
};
```

HopeRF RFM95 LoRa transceiver module	Arduino Uno Pin		HopeRF RFM95 LoRa transceiver module	Arduino Uno Pin
ANT	-		GND	-
GND	GND		DIO5	-
DIO3	-		RESET	5
DIO4	-		NSS	10
3.3V	3.3V		SCK	13
DIO0	2		MOSI	11
DIO1	3		MISO	12
DIO2	-		GND	-

- In this sketch the message "Hello, world!" will be transmitted every 60 seconds, see variables: mydata[] and TX_INTERVAL.

- For this video the TX_INTERVAL is 60 seconds.

DEMONSTRATION

- Connect the self build LoRa development board to your computer using the USB cable.

- In the Arduino IDE, select menu Tools | Board and select: Arduino/Genuino Uno
  In the Arduino IDE, select menu Tools | Port and select: *your_port*

- Compile ttn-otaa-mydemo sketch.
  You should not see any errors.

- Upload the ttn-otaa-mydemo sketch to the Arduino Uno.
  You should not see any errors.

- In the Arduino IDE, select menu Tools | Serial Monitor
  Select baud rate: 9600

DEMONSTRATION

- Goto The Things Network console:
  - Select the app: youtube_demo_app
  - Select the registered device: youtube_demo_device
  - Select reset frame counters. Do this each time your end device is powered up.

- In the Device Overview screen, top right corner, select Data.

- Your device is being registered and after a few minutes the message "Hello, world!" (in hex: 48 65 6c 6c 6f 2c 20 77 6f 72 6c 64 21) is received by a gateway and send to the network server and displayed in the The Things Network console.

- The metadata displayed by TTN console, during the demonstration can be found here: https://www.mobilefish.com/download/lora/ttn-otaa-mydemo-data.txt

DEMONSTRATION

- Question:
  Do I comply with the ETSI duty cycles and The Things Network Fair Access Policy if I keep running the ttn-otaa-mydemo sketch for a day?

- This question will be answered in the next video.