LORA / LORAWAN TUTORIAL 56

OUI, OUI-36, EUI-64, DevEUI, AppEUI, JoinEUI

70B3D57EDF0B9BBA

ABE8F8F0F0B9BBAE

INTRO

- In this tutorial I will explain what OUI, OUI-36, EUI-64, DevEUI, AppEUI and JoinEUI exactly are.

PRESENTATION

- This presentation can be found at:
  https://www.mobilefish.com/download/lora/lora_part56.pdf

- All my LoRa/LoRaWAN tutorials and presentations can be found at:
  https://www.mobilefish.com/developer/lorawan/lorawan_quickguide_tutorial.html

- In this video when V2 is mentioned, V2 refers to The Things Network
  and when V3 is mentioned, V3 refers to The Things Stack Community Edition.

OUI

- An Organizationally Unique Identifier (OUI) is a 24-bit number that uniquely identifies a vendor, manufacturer, or organization.

- The IEEE Standards Association Registration Authority assigns OUI values. https://standards.ieee.org/products-services/regauth/index.html

- An OUI example in a three octet sequence:

octet 0 **octet 1** **octet 2**

F4-BD-9E

OUI registered to Cisco Systems, Inc

octet 0 in bits

11110100

F **4**

OUI

octet 0 octet 1 octet 2

F4-BD-9E

octet 0 in bits

11110100

X M

The least and second least significant bits of octet 0 are designated the M bit and X bit, respectively.

OUI

octet 0 octet 1 octet 2

F4-BD-9E

octet 0 in bits

11110100

X M

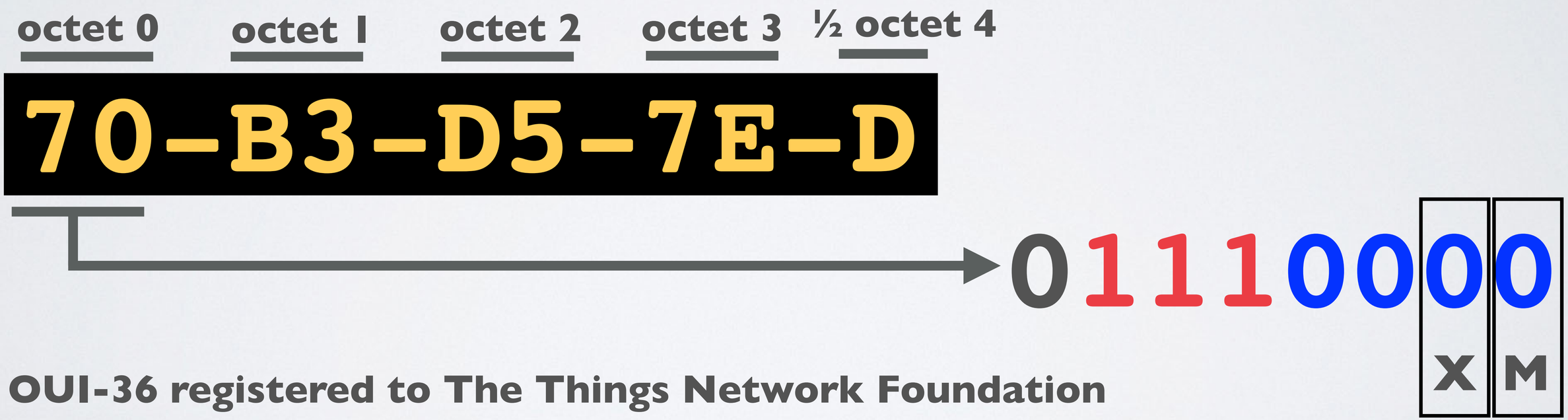
In octet 0, both the X and M bit have the value 0

X bit is also called U/L bit
(U)niversal administered = 0
(L)ocal administered = 1

M bit is also called I/G bit
(I)ndividual address = 0
(G)roup address = 1

OUI-36

- The OUI-36 is a 36-bit (four-and-one-half-octet) sequence.
- The least and second least significant bits of octet 0 are designated the M bit and X bit, respectively. In this example both the M and X bit have the value 0.



OUI-36 registered to The Things Network Foundation

OUI-36

- An OUI-36 is created by the IEEE Registration Authority by concatenating 12 bits to a 24-bit IEEE-reserved base OUI, after octet 2.

OUI (24-bits) **12 bits**

70-B3-D5-7E-D

octet 0 **octet 1** **octet 2**

- An assignee of an OUI-36 (for example The Things Network Foundation) shall not truncate the OUI-36 to use as an OUI (24-bits) because the IEEE RA will use the base OUI to assign OUI-36 values to multiple organizations.

OUI-36

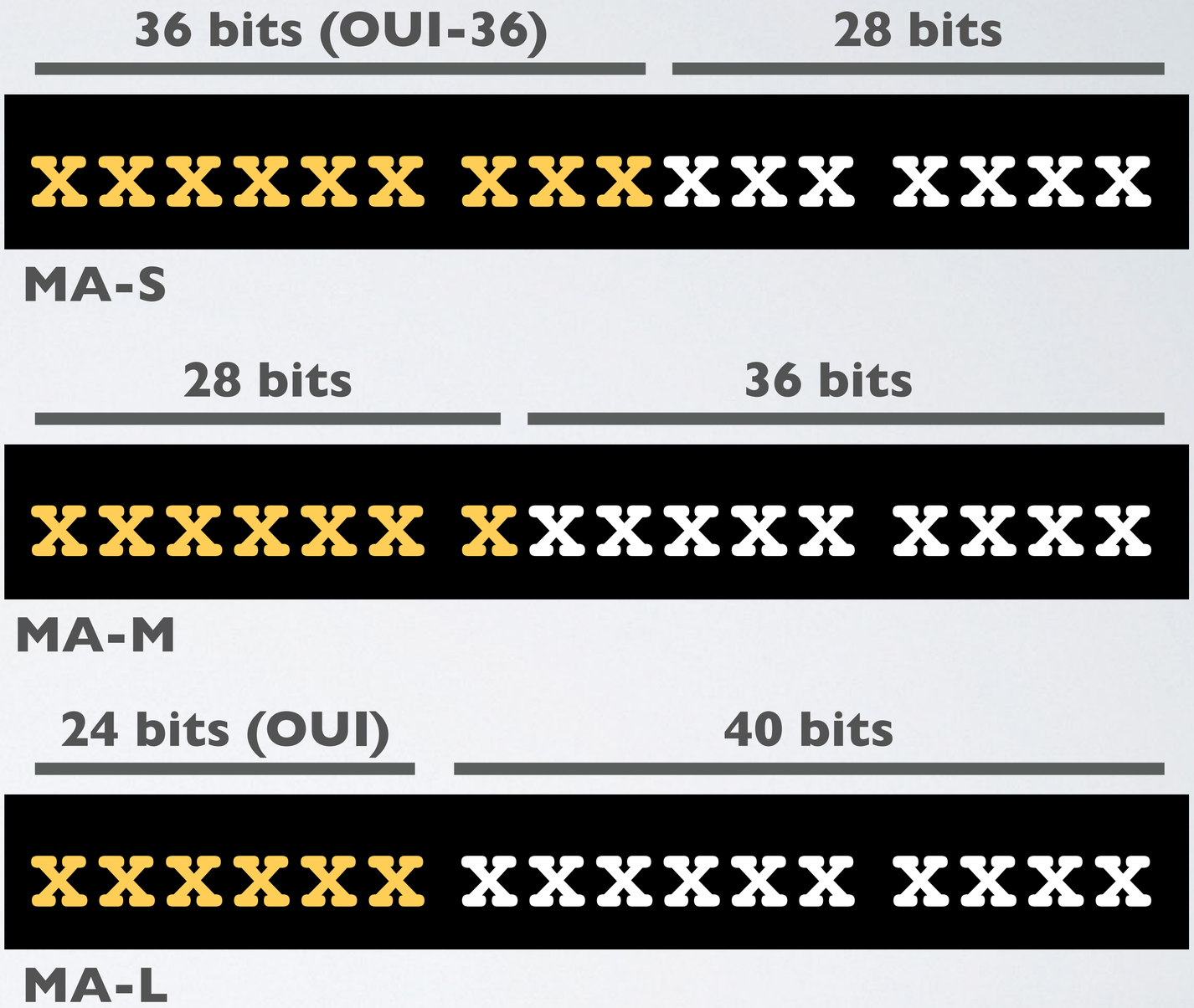
OUI (24-bits)	12 bits	
70-B3-D5-	7E-D	OUI-36 registered to The Things Network Foundation
70-B3-D5-	45-D	OUI-36 registered to Sensapex Oy
70-B3-D5-	CB-E	OUI-36 registered to Ensure Solutions BV
70-B3-D5-	12-0	OUI-36 registered to GSP Sprachtechnologie GmbH

MA-S / MA-M / MA-L

- The IEEE Registration Authority divides the addresses in 3 different size blocks:
  MAC Address Block **S**mall:
  MA-S
  2^{28} = 268,435,456 EUI-64 addresses

- **MA**C Address Block **M**edium:
  MA-M
  28 bits identifier, no OUI
  2^{36} = 68,719,476,736 EUI-64 addresses

- **MA**C Address Block **L**arge:
  MA-L
  2^{40} = 1,099,511,627,776 EUI-64 addresses



MA-S / MA-M / MA-L

- Companies or Organizations can purchase a MAC address block with its own unique OUI (MA-L), OUI-36 (MA-S) or 28 bits identifier (MA-M).

OUI-36 **28 bits**

70B3D57ED0040620

OUI-36 registered to The Things Network Foundation

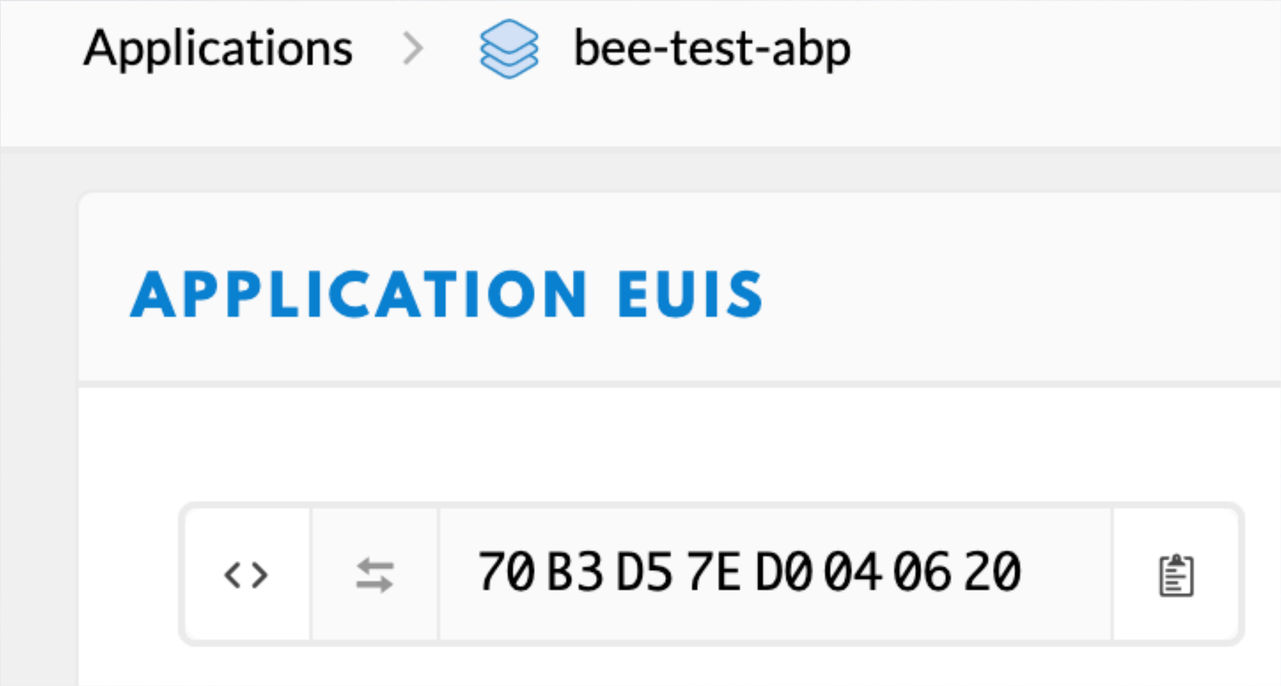
- A company can use the purchased OUI-36 to create their own unique EUI-64 addresses for purposes where unique 64 bits are needed.

- In this example The Things Network Foundation must assign additional 28 bits (aka extension identifier) to create a EUI-64 bit address.

- TTN Foundation can freely set these 28 bits.

MA-S / MA-M / MA-L

- The previously shown EUI-64 address is an AppEUI generated by the Things Network console (V2).



SEARCH MA-S / MA-M / MA-L PUBLIC LISTINGS

- You can search the public IEEE Registration Authority master listings (MA-L, MA-M and MA-L) for the company names and addresses and which OUI, OUI-36 or unique 28 bits identifiers they are registered to:
  https://regauth.standards.ieee.org/standards-ra-web/pub/view.html

- Alternative use the Wireshark OUI Lookup Tool
  https://www.wireshark.org/tools/oui-lookup.html
  Search for: 70-B3-D5-7E-D0-04-06-20
  Search for: 70-B3-D5-7E-D0
  Search for: the things network foundation

SEARCH MA-S / MA-M / MA-L PUBLIC LISTINGS

- MAC Address Block Large (MA-L)
  Search: Microchip Technology Inc.
  00-04-A3 (hex)
  OUI: **00-04-A3**

- MAC Address Block Medium (MA-M)
  Search: Honeywell
  30-09-F9 (hex)
  C00000-**C**FFFFF
  28 bits identifier: **30-09-F9-C**

- MAC Address Block Small (MA-S)
  Search: The Things Network Foundation
  70-B3-D5 (hex)
  7ED000-**7ED**FFF
  OUI-36: **70-B3-D5-7E-D**

IEEE REGISTRATION FEES FOR MA-S / MA-M / MA-L

The IEEE registration fees for the MAC address block Small, Medium and Large.
Source: https://standards.ieee.org/products-services/regauth/oui36/index.html
(June 2021)

	MA-S (US $)	MA-M (US $)	MA-L (US $)
One Time Fee			
Publicly registered Company name & address will be displayed on the IEEE public listing	780	1855	3085
Contract fee Optional - For those that need a signed, stamped hard copy of a contract to release payment	200	200	200
Optional Recurring Fee			
Yearly confidentiality renewal fee Company name & address will **not** be displayed on the IEEE public listing	1260	2340	3565

MORE INFORMATION ABOUT OUI AND EUI

- For more information about OUI and EUI: https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/tutorials/eui.pdf

MAC ADDRESS

- A **M**edia **A**ccess **C**ontrol address (MAC address) is a unique identifier assigned to a **network card**. These addresses identifies each node on a network.

- MAC addresses are primarily assigned by device manufacturers and are embedded in the hardware. Depending on the manufacturer, these MAC addresses can be changed.

- A MAC address includes a manufacturer's organizationally unique identifier (OUI) and an extension identifier which can be set freely by the device manufacturer.

OUI **EXTENSION ID**

186590 12AF1A

Network card
48 bit MAC address
OUI registered to Apple Inc

EUI

- The **E**xtended **U**nique **I**dentifier (EUI) is used to identify **other devices and software.**

- Just like a MAC address, an EUI address includes a manufacturer's organizationally unique identifier (OUI or OUI-36) and an extension identifier which can be set freely by the device manufacturer.

OUI-36	EXTENSION ID
70B3D57ED	0040620

TTN generated AppEUI (EUI-64)
OUI-36 registered to The Things Network Foundation

GENERATE LOCAL ADMINISTERED EUI-64

- The whole point of an EUI address (same applies to MAC) is that they are guaranteed to be unique.
- If users start using random numbers to generate an EUI address it is not going to be unique anymore. The self generated EUI may clash with someone else's.
- As mentioned earlier you can generate a local administered EUI-64 with a very low probability of collision.
- Procedure to create a local administered EUI-64:
  - Create a random 64 bit number: **79-AE-0B-C5-66-3B-F3-A7**
  - Make sure in octet 0 the X-bit = 1 and the M-bit = 0

GENERATE LOCAL ADMINISTERED EUI-64

Incorrect local administered EUI-64

79-AE-0B-C5-66-3B-F3-A7

→ 01111001

X M

Correct local administered EUI-64

76-AE-0B-C5-66-3B-F3-A7

→ 01111010

X M

GENERATE LOCAL ADMINISTERED EUI-64

Local administered EUI-64 (X-bit = 1, M-bit = 0)
N must be 2, 6, A or E

XN-XX-XX-XX-XX-XX-XX-XX

~~0 = 0000~~	~~5 = 0101~~	A = 1010
~~1 = 0001~~	6 = 0110	~~B = 1011~~
2 = 0010	~~7 = 0111~~	~~C = 1100~~
~~3 = 0011~~	~~8 = 1000~~	~~D = 1101~~
~~4 = 0100~~	~~9 = 1001~~	E = 1110
		~~F = 1111~~

PROVISIONING A DEVICE

- The term "provisioning a device" means to evolve a device to a state in which it can be handed off to an end-user for their specific use in a functional manner.

- In a LoRaWAN context "*provisioning a device*" refers to storing essential data such as DevEUI, JoinEUI, AppKey or NwkKey on a LoRaWAN 1.1 device.

For example:
The Things Node is a LoRaWAN 1.0.x device. Its is a waterproof matchbox size device with a movement, light, and temperature sensor. The device has a LED and a button.

Out-of-the box this device is provisioned with a DevEUI and AppEUI.

LoRaWAN 1.0.x
The Things Stack Community Edition

LORAWAN 1.0.X: DEVEUI, APPEUI, APPKEY

- In LoRaWAN 1.0.x the following values are important: DevEUI, AppEUI and AppKey.
- The DevEUI is an EUI-64 address that uniquely identifies the end-device.
- The AppEUI is an EUI-64 address that uniquely identifies the entity able to process the JoinReq frame.
- The purpose of the DevEUI and AppEUI are explained in detail in tutorial 21.

LORAWAN 1.0.X: DEVEUI, APPEUI, APPKEY

- The combination DevEUI and AppEUI should be unique which means one or both these values must be unique.
- If your end device is provided with a **DevEUI**, you should use it.
- If your end device is not provided with a **DevEUI**, you should generate a local administered EUI-64 (X-bit=1, M-bit = 0).
- If your end device is provided with an **AppEUI**, you should use it.
- If your end device is not provided with an **AppEUI**, the TTS CE console recommends that you use an AppEUI consisting of all zeros.
- Ensure that you use the same AppEUI in your device as you enter in TTS CE console.

LORAWAN 1.0.X: DEVEUI, APPEUI, APPKEY

By pressing this button, the AppEUI is set to all zeros.

REGISTER END DEVICE (TTS COMMUNITY EDITION V3)

Register end device

From The LoRaWAN Device Repository | Manually

1 **Basic settings** End device ID's, Name and Description
2 Network layer settings Frequency plan, regional parameters, end device class and session keys.
3 Join settings Root keys, NetID and kek labels.

End device ID *
my-new-device

AppEUI *
·· ·· ·· ·· ·· ·· ·· ·· 00

DevEUI *
·· ·· ·· ·· ·· ·· ·· ··

End device name
My new end device

End device description
Description for my new end device

Optional end device description; can also be used to save notes about the end device

Network layer settings >

Register end device

From The LoRaWAN Device Repository | Manually

✓ Basic settings End device ID's, Name and Description
2 **Network layer settings** Frequency plan, regional parameters, end device class and session keys.
3 Join settings Root keys, NetID and kek labels.

Frequency plan *
Select...

LoRaWAN version *
MAC V1.0.2

Regional Parameters version *
Select...

LoRaWAN class capabilities
☐ Supports class B
☐ Supports class C

Advanced settings

< Basic settings | Join settings >

Register end device

From The LoRaWAN Device Repository | Manually

✓ Basic settings End device ID's, Name and Description
✓ Network layer settings Frequency plan, regional parameters, end device class and session keys.
3 **Join settings** Root keys, NetID and kek labels.

Root keys

AppKey *
·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ·· ··

Advanced settings

< Network layer settings | Add end device

LoRaWAN 1.0.x

LoRaWAN 1.1
The Things Stack Community Edition

LORAWAN 1.1: DEVEUI, JOINEUI, APPKEY, NWKKEY

- In LoRaWAN 1.1 the following values are important: DevEUI, JoinEUI, AppKey and NwkKey.
- The DevEUI is an EUI-64 address that uniquely identifies the end-device.
- The JoinEUI is an EUI-64 address that uniquely identifies the join server that can assist in the processing of the join procedure and session keys derivation.
- The purpose of the DevEUI and JoinEUI are explained in detail in tutorial 55.

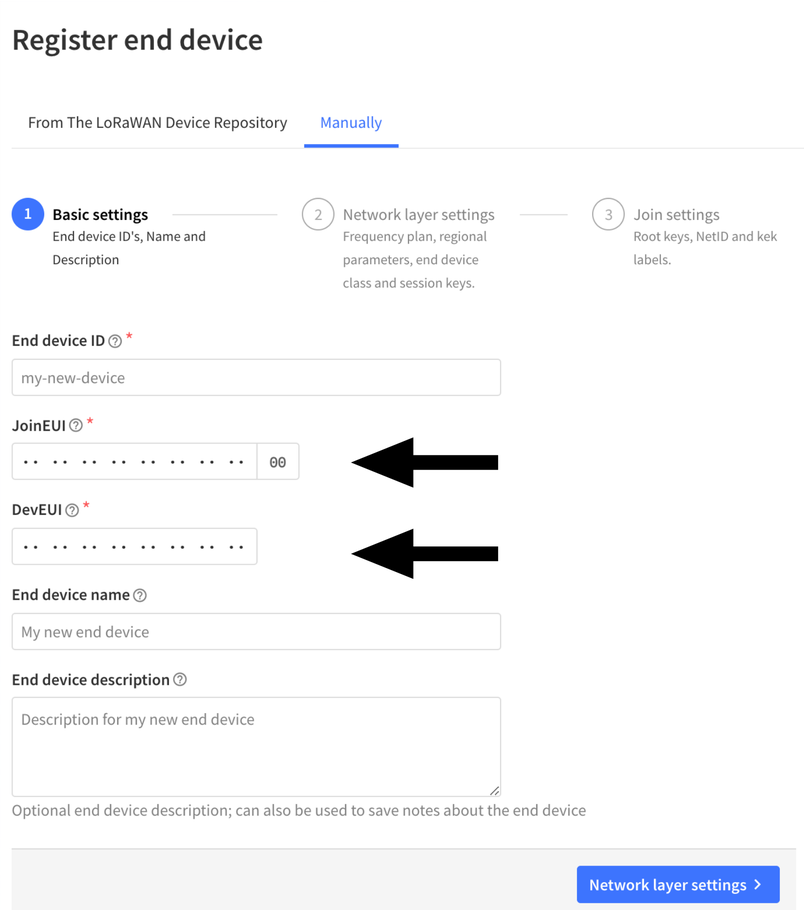
LORAWAN 1.1: DEVEUI, JOINEUI, APPKEY, NWKKEY

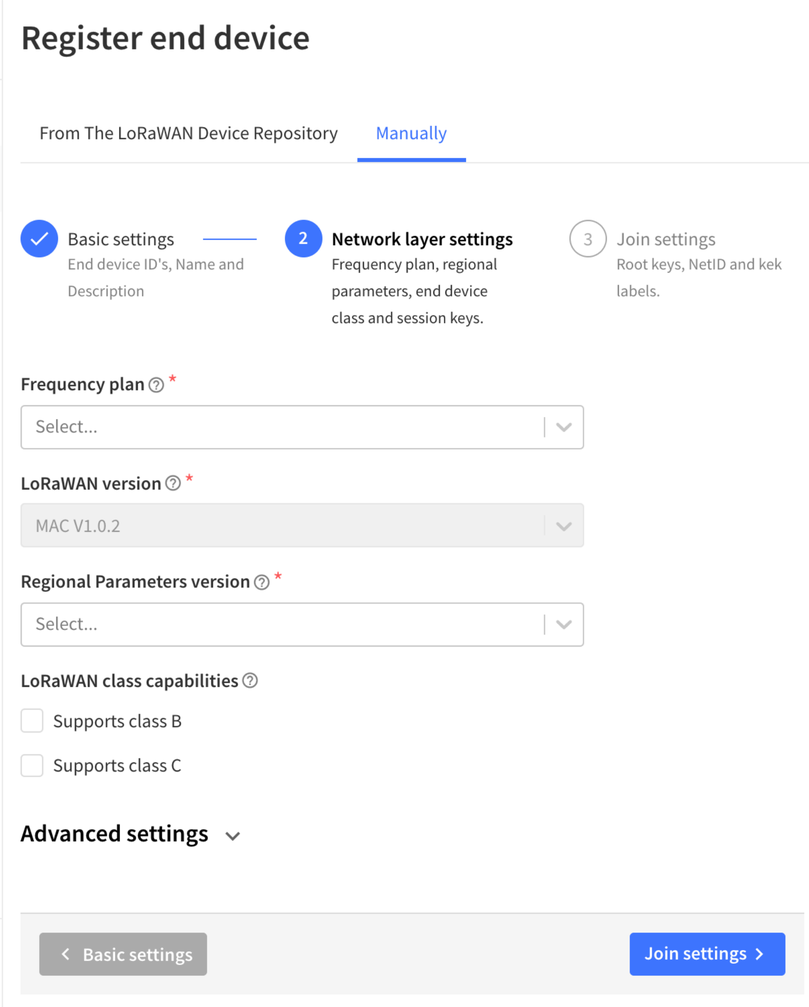
- If your end device is provided with a **DevEUI**, you should use it.
- If your end device is not provided with a **DevEUI**, you should generate a local administered EUI-64 (X-bit=1, M-bit = 0).
- If your end device is provided with a **JoinEUI**, you should use it.
- If your end device is not provided with a **JoinEUI**, the TTS CE console recommends that you use a JoinEUI consisting of all zeros.
- Ensure that you use the same JoinEUI in your device as you enter in TTS CE console.

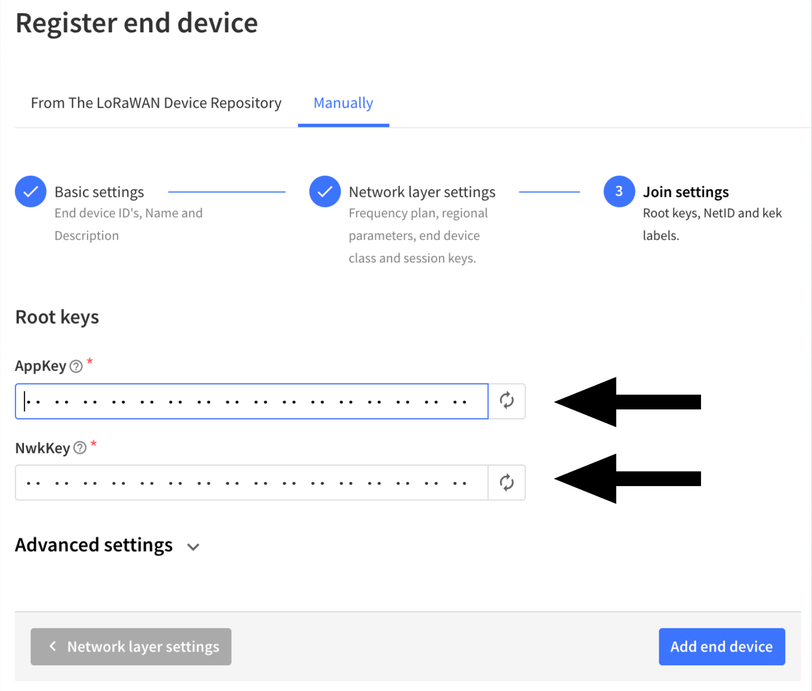
LORAWAN 1.1: DEVEUI, JOINEUI, APPKEY, NWKKEY

By pressing this button, the JoinEUI is set to all zeros.

REGISTER END DEVICE (TTS COMMUNITY EDITION V3)







LoRaWAN 1.1

WHEN TO USE LOCAL ADMINISTERED EUI-64

When registering a LoRaWAN 1.0.2 end device in the TTS CE console, the DevEUI and AppEUI are required.

1. First check if your device has the required AppEUI and DevEUI.
   Most commercially available end devices have these EUI numbers built-in.

2. If the end device has the DevEUI, use it.

3. If the end device does not have the DevEUI, check if the TTS CE console provides a method to generate the required DevEUI.
   As of June 2021 this method does not exist.

WHEN TO USE LOCAL ADMINISTERED EUI-64

3. For the DevEUI you can use your own local administered EUI-64.
   An online tool to generate your own local administered EUI:
   https://www.mobilefish.com/download/lora/lorawan_device.html

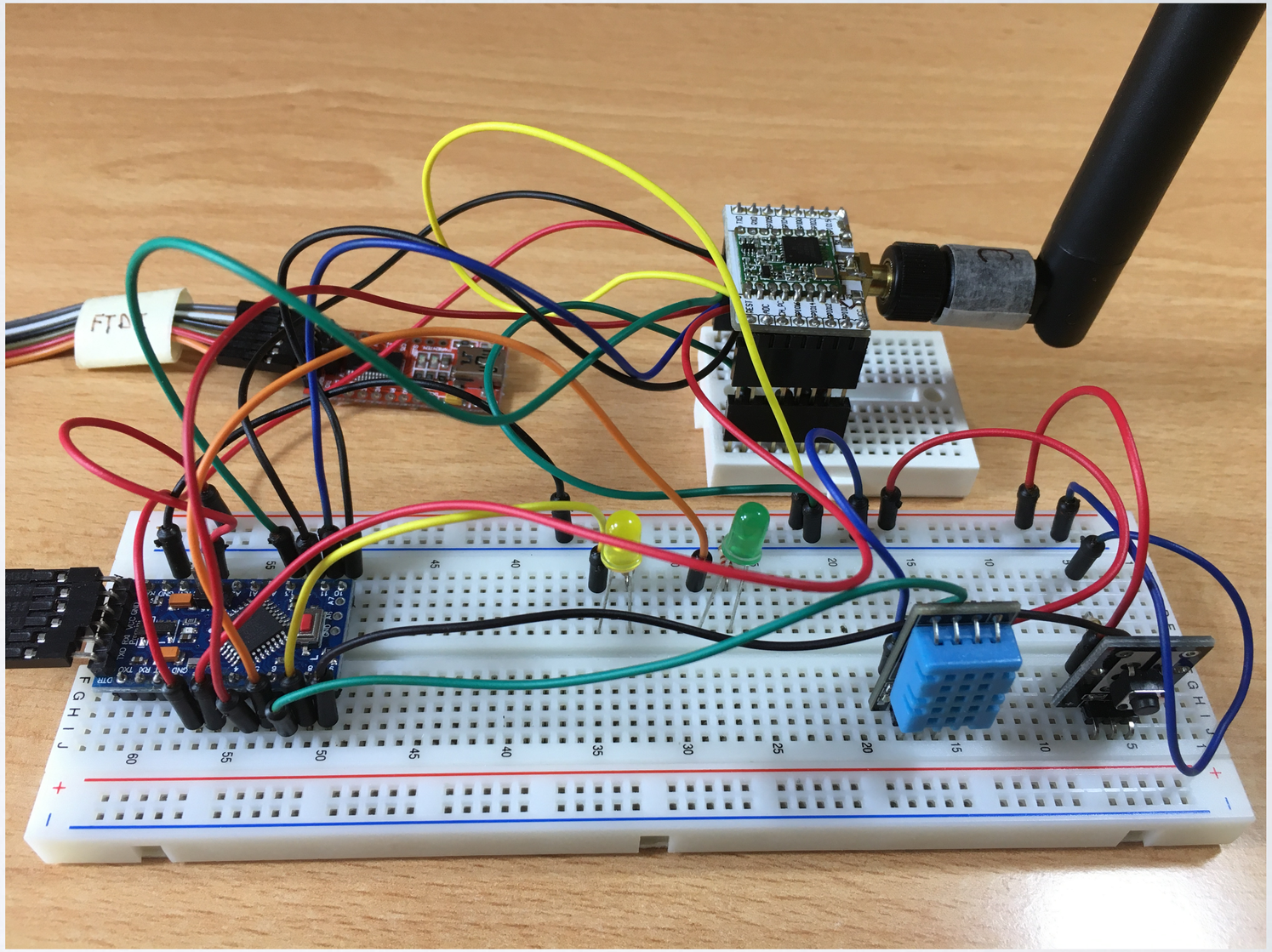
4. If the end device has the AppEUI, use it.

5. If the end device does not have the AppEUI, use an AppEUI with all zeros.

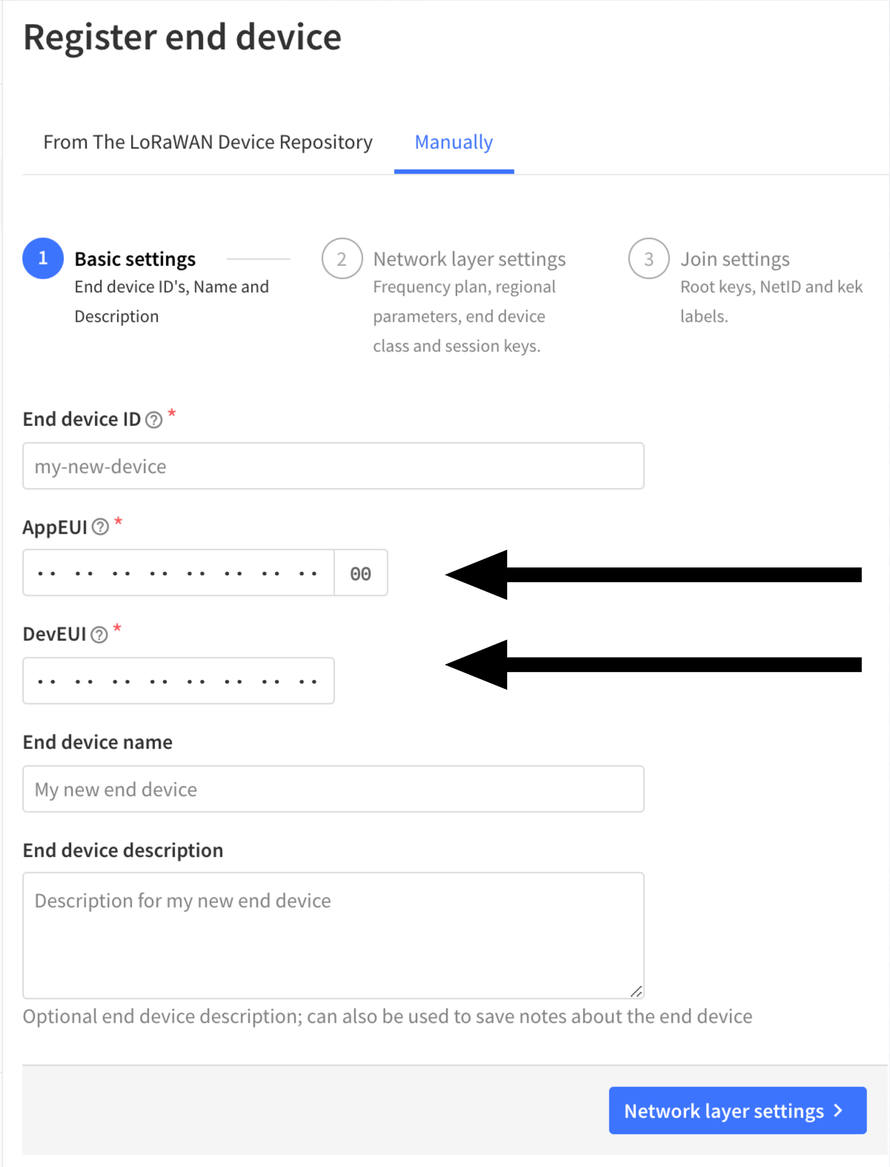
EXAMPLE USE LOCAL ADMINISTERED EUI-64

- As an example you build your own LoRaWAN 1.0.2 device for a personal project using an Arduino Pro Mini (ATmega328P, 3.3V, 8MHz) as the micro controller with the HopeRF RFM95W as the RF transceiver module.

- The HopeRF RFM95W has no build-in DevEUI or AppEUI.



EXAMPLE USE LOCAL ADMINISTERED EUI-64



AppEUI = 0000000000000000

DevEUI = CE169335114AAF5C