
LoRaWAN-UNAL Documentation

Versión latest

22 de febrero de 2023

Contenido:

1. UNAL-LoRaWAN	1
2. Basic MAC	3
3. STM32L0 Discovery kit LoRa, Sigfox, low-power wireless	5

CAPÍTULO 1

UNAL-LoRaWAN

Esta documentación tiene como finalidad explicar el manejo de **Basic MAC**, haciendo uso de la tarjeta de desarrollo **B-L072Z-LRWAN1 LoRa®/Sigfox™ Discovery Kit**.

CAPÍTULO 2

Basic MAC

Basic MAC es una implementación portátil de la especificación LoRaWAN® de **LoRa Alliance®** en el lenguaje de programación C. Es una bifurcación de la biblioteca LMIC de IBM y admite múltiples regiones, que se pueden seleccionar en tiempo de compilación y/o ejecución. Puede manejar dispositivos de Clase A, Clase B y Clase C.¹

¹ Una traducción de Basic MAC getting started guide.

STM32L0 Discovery kit LoRa, Sigfox, low-power wireless

3.1 B-L072Z-LRWAN1

El kit de descubrimiento LoRa®/Sigfox™ B-L072Z-LRWAN1 es una herramienta de desarrollo para aprender y desarrollar soluciones basadas en las tecnologías LoRa®, Sigfox™ y FSK/OOK. El kit B-L072Z-LRWAN1 Discovery incluye una interfaz de herramienta de depuración embebida ST-LINK/V2-1, LEDs, pulsadores, antena, conectores Arduino™ Uno V3 y conector USB OTG en formato Micro-B.²

3.1.1 Preparación del entorno

Para hacer uso de [Basic MAC](#) es necesario tener instalado en terminal Linux las siguientes librerías, estas se requieren para realizar la compilación del firmware como también su instalación en la placa de desarrollo [B-L072Z-LRWAN1 LoRa®/Sigfox™ Discovery Kit](#).

La guía explicara los preparativos en dos de las principales distribuciones de Linux, [Ubuntu](#) y [Arch Linux](#)

Requisitos previos

- Es necesario instalar un compilador para arm, para ello usamos el paquete [GCC, the GNU Compiler Collection](#) y su variante para arm la cual es una cadena de herramientas de compilación cruzada.

En ubuntu para instalar se puede instalar directamente con el gestor de paquetes apt.:

```
sudo apt update && sudo apt-get install gcc-arm-none-eabi libnewlib-arm-none-eabi
```

En Arch Linux y sus derivadas se usa el gestor de pacman.

```
sudo pacman -Sy arm-none-eabi-gcc arm-none-eabi-newlib
```

- Un entorno con un intérprete de Python 3.8, para ello vamos a usar entornos virtuales y un control de versiones en python.

² Una traducción de [Product overview B-L072Z-LRWAN1](#).

En ubuntu se instalan las siguientes librerías.:

```
sudo apt-get install python3 python3-tk python3-tk-dev python3.8-venv
```

En Arch Linux

```
sudo pacman -Sy pyenv tk python
```

- OpenOCD como herramienta para instalar en el hardware real, es necesario usar la version 1.0.10 del 01/Nov/2020

En ubuntu (aun por verificar como se instala esa version exactamente)

```
sudo apt-get -y install openocd
```

En Arch Linux

```
sudo pacman -U https://archive.archlinux.org/packages/o/openocd/openocd-1%3A0.10.0-4-x86_64.pkg.tar.zst
```

``Nota: Editar el archivo /etc/pacman.conf y agregar la linea IgnorePkg = openocd`` mas información sobre pacman aquí.

- make y git para compilar y administrar una colección de aplicaciones y archivos desde el código fuente.

En Ubuntu

```
sudo apt-get install make git
```

En Arch Linux

```
sudo pacman -Sy make git
```

3.1.2 Construcción y carga del proyecto

1. Clonar el repositorio de GitHub de Basicmac y todos sus submódulos.:

```
git clone --recurse-submodules https://github.com/mkuyper/basicmac.git basicmac
```

2. Crear y activar un entorno virtual con Python dentro de la carpeta basicmac.

Para mejorar la gestion de las versiones de python la idea seria utilizar pyenv para cargar la version necesaria de python, pues los paquetes requeridos solo funcionan en python 3.8.*

Pasos:

- instalar pyenv y tk en arch: sudo pacman -Sy pyenv tk y en Debian/Ubuntu sudo apt install python-tk python3-tk tk-dev y revisar estas dos guías <https://www.devopsroles.com/install-pyenv/> <https://bgasparotto.com/install-pyenv-ubuntu-debian>
- instalar la version de python 3.8.16 con pyenv install 3.8.16 y
- dentro de la carpeta del proyecto ejecutar pyenv local 3.8.16 y pyenv exec python -m venv py3.8.16 y source py3.8.16/bin/activate

```
`bash cd basicmac python3 -m venv venv source venv/bin/activate `
```

3. Instalar los módulos de Python requeridos por las herramientas de compilación y simulación, se pueden instalar usando pip. no actualizar pip

```
`bash pip install wheel pip install -r basicloader/requirements.txt pip install -r ./
requirements.txt pip install pyyaml `
```

4. Basicmac emplea un gestor de arranque, Basicloader, para cargar e iniciar el firmware real. Este cargador de arranque también aplica cualquier actualización de firmware y verifica la integridad del firmware actual antes de llamar al punto de entrada del firmware. Para crear este cargador usar el siguiente comando:

```
`bash cd basicloader VARIANT=us915 make ` El resultado del proceso de compilación es un archivo llamado
bootloader.hex.
```

5. Conectar la tarjeta al computador mediante USB, verifique que el dispositivo es reconocido como una unidad USB con nombre *DIS_L072Z*.
6. Después de compilar un proyecto, se puede cargar en un dispositivo. Tenga en cuenta que tanto el cargador de arranque como el firmware deben cargarse, aunque como el cargador de arranque rara vez cambia, generalmente solo se carga una vez. Para cargar el cargador de arranque y el firmware, respectivamente, use los objetivos *make loadbl* y *make load*.

```
`bash cd ../projects/ex-join VARIANT=us915 make loadbl VARIANT=us915 make load `
```

3.1.3 Personalización

El HAL para STM32 almacena información de personalización como EUI y claves para el funcionamiento de LoRaWAN en EEPROM. Si no se encuentra información de personalización válida en la EEPROM, la HAL creará un dispositivo EUI a partir de los registros de ID únicos de la MCU's y utilizará una clave de prueba y un EUI de unión fijos [[1]](<https://basicmac.io/guide/gettingstarted.html>):

- Device EUI: *FF-FF-FF-AA-xx-xx-xx-xx*
- Join EUI: *FF-FF-FF-BB-00-00-00-00*
- Device Key: *404142434445464748494A4B4C4D4E4F*

Los valores de Join EUI y Device Key son los mismos para todas las tarjetas, solo el Device EUI es único para cada tarjeta. Para ver el Device EUI utilizado por la tarjeta de desarrollo, se deben seguir los siguientes pasos:

1. Abrir una nueva terminal o salir del directorio anterior.
2. En el B-L072Z-LRWAN1, el firmware imprime información de depuración en el UART que está conectado a través de ST-LINK a la computadora host. En Linux, este dispositivo suele aparecer como */dev/ttyACM0*. Ejecutar el siguiente comando para lanzar el terminal serial minicom de la tarjeta:

```
`bash sudo minicom -D /dev/ttyACM0 -b 2000000 `
```

3. Presionar los dos botones de reset de la tarjeta al tiempo. En la terminal se visualizara la salida de depuración. En esta se muestra el Device EUI como se muestra en la siguiente imagen.

![minicom](./img/minicom.png)

Conexión a The Things Network (TTN)

1. Entrar a la sección de aplicaciones de la pagina de The Things Network, a traves del link: <https://nam1.cloud.thethings.network/console/applications>
2. Crear un nueva aplicación presionando el botón de *add application* y dandole un nombre y un ID.
3. Ir a la pestaña de *End devices* y crear un nuevo dispositivo en *add end device*, como se muestra en la siguiente imagen.

![registro 3](./img/Registro_3.png)

4. Registrar manualmente el dispositivo como se muestra en la siguiente imagen.

![registro 1](./img/Registro_1.png)

5. Ingresar los datos de Device EUI, Join EUI y Device Key como se muestra en la siguiente imagen. Estos datos fueron dados en la sección anterior, recordar que el Device EUI es único de la tarjeta y se obtuvo en el paso 3 de la sección de Personalizaron. Por ultimo, presionar el botón de *Register end device*.

![registro 2](./img/Registro_2.png)

Una vez terminados estos pasos, ya se podrán visualizar los datos enviados por la tarjeta a TTN desde la pestaña *Live data* de la aplicación creada, siempre y cuando se tenga cobertura de un Gateway conectado a la red de TTN.

Referencias

[[1] M. Kuyper. Basicmac, Getting Started, 2020](<https://basicmac.io/guide/index.html>)

[[2] ST. UM2115, User manual. Discovery kit for LoRaWAN™, Sigfox™, and LPWAN protocols with STM32L0](https://www.st.com/resource/en/user_manual/dm00329995-discovery-kit-for-lorawan-sigfox-and-lpwan-protocols-with-stm32l0-stmicroelectronics.pdf)

[[3] A crash course in LoRaWAN®, 2021](https://twacław.github.io/presentation_lora_lorawan/index.html)