[LoRaWAN] Configuración

> No usar una Raspberry Pi para hacer la configuración de la placa LoRaWAN, ya que el software necesario para flashear la placa no soporta la arquitectura de la Raspberry Pi

Preparación del entorno

> Para la configuración de LoRaWAN, usaremos un dispositivo HTCC-AB02A

- Instalamos la extensión "*PlatformIO IDE*" para Visual Studio Code.
- Abrimos la extensión y hacemos click en "Open", "Project Examples" y "Install Embedded Platform".
  - Buscamos la plataforma "*Heltec CubeCell*", la seleccionamos y hacemos click en "Install".
- Volvemos al menú "Project Examples", seleccionamos el ejemplo "LoRa/LoRaWan/LoRaWan" y hacemos click en "Import".
- Se habrá generado un entorno de trabajo. Los dos ficheros que nos interesan son "*platformio.ini*" y "*src/LoRaWan.ino*".

platformio.ini

- Eliminamos todas las configuraciones excepto la correspondiente a "*cubecell_board*", y la editamos de forma que quede lo siguiente:

```
[env:cubecell_board]
platform = heltec-cubecell
framework = arduino
board = cubecell_board
monitor_speed = 115200
upload_port = <port>
board_build.arduino.lorawan.region = EU868
```

- Conectamos la placa LoRaWAN a un puerto USB y hacemos click en "Devices". Se mostrará en la terminal una lista con los dispositivos conectados y su puerto. En nuestro caso, la descripción debería ser algo similar a "*Silicon Labs CP210x USB...*". Sustituímos el parámetro "*<port>*" en "*platformio.ini*" por el puerto donde está conectada la placa.

  > Si no vemos el dispositivo, es probable que el motivo sea que estamos usando Windows, y no tenemos los drivers instalados.

LoRaWan.ino

Este fichero contiene el código que flashearemos a la placa, y que esta ejecutará continuamente, siempre que disponga de alimentación. Por ahora, este código prepara un paquete con 4 bytes (0x00, 0x01, 0x02, 0x03) y los intenta retransmitir.

Registro del dispositivo en TheThingsNetwork

- Accedemos a la página de administración de TTN e iniciamos sesión (o creamos una cuenta).
- Hacemos click en "Add application", rellenamos los datos (el ID lo usaremos más adelante) y hacemos click en "Create application".
- Hacemos click en "Register end device":
  - **Brand:** *HelTec AutoMation*
  - **Model:** *HTCC-AB02A(Class A OTAA)*
  - **Hardware Ver.:** *Unknown ver.*
  - **Firmware Ver.:** *1.0*
  - **Profile (Region):** *EU_863_870*
- Seleccionamos el plan de frecuencia "Europe 868.1 MHz"

- Rellenamos el campo "*JoinEUI*" con ceros, y hacemos click en confirmar
- Hacemos click en "Generate" para los campos "*DevEUI*" y "*AppKey*"
- Rellenamos el campo "*End device ID*" con un identificador que usaremos más adelante.
- Hacemos click en "Register new device"

Prueba de funcionamiento

- Editamos el fichero "*LoRaWan.ino*" para añadir las claves del dispositivo registrado, disponibles en el menú "Device overview" de la consola de TNN.
    - **Línea 14:** la modificamos para usar el "*DevEUI*" de nuestro dispositivo.

      ```
      // EJEMPLO
      uint8_t devEui[] = { 0x22, 0x32, 0x33, 0x00, 0x00, 0x88, 0x88, 0x02 };
      ```

    - **Línea 16:** la modificamos para usar el "*AppEUI*" de nuestro dispositivo.

      ```
      //EJEMPLO
      uint8_t appKey[] = { 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88,
      0x88, 0x66, 0x01 };
      ```

- Desde la extensión:
    - Hacemos click en "Build", para compilar el programa. Si todo está correcto, veremos un mensaje "*SUCESS*".
    - Hacemos click en "Upload", para flashear el programa en la placa. Si todo está correcto, veremos un mensaje "*SUCESS*". Una vez hecho esto, la placa ya estará ejecutando el programa siempre que tenga alimentación.
    - Hacemos click en "Monitor". Podremos ver en la terminal cada vez que se transmita un paquete, mediante un mensaje "*confirmed uplink sending ...*". Es posible que al principio no muestre nada, ya que se tiene que realizar primero la conexión al gateway.
    - Desde la consola de TNN, en el menú "Live data", podremos ver todos los mensajes recibidos y enviados. En concreto, por cada transmisión tenemos tres mensajes:
        - "*Successfully processed data message*"
        - "*Forward uplink data message*"
        - "*Schedule data downlink for transmission on Gateway server*"

Envío de datos

- Editamos el fichero "*LoRaWan.ino*" para eliminar esa transmisión de prueba, de forma que el código hará que el módulo se una a la red, y se quede a la espera.
    - Nos quedará un código como este, pero con las claves de nuestro dispositivo.
    - Desde la extensión hacemos click en "Build" para compilar el programa, y cuando termine hacemos click en "Upload" para flashearlo en la placa.
- Para enviar los datos a la red, será necesario primero enviarlos al módulo mediante un comando AT a través de su puerto serie. Para ello, creamos un fichero "*pub.py*" con este código.

Recepción de datos

- Cada vez que TNN recibe un mensaje, codifica el payload en formato Base64, lo empaqueta en un objeto JSON, y lo reenvía a todas las integraciones disponibles. De todas ellas, nos interesa la integración MQTT, que automáticamente publica en un tópico todos los mensajes que recibe, con el formato anteriormente mencionado. Desde el menú "Integrations > MQTT" podremos ver la información para conectarnos al broker:
    - **Dirección:** "*eu1.cloud.thethings.network*"
    - **Puerto:** "*1883*"
    - **Usuario:** dependerá del ID que hayamos asignado al crear la aplicación.
    - **Contraseña:** haciendo click en "Generate new API key" se generará una clave única.
- Los datos se publican en varios tópicos, pero el que nos interesa sigue la forma "*v3/{application id}@{tenant id}/devices/{device id}/up*":
    - **{application id}:** será el ID que hayamos asignado a la aplicación.
    - **{tenant id}:** "*tnn*"
    - **{device id}:** será el ID que hayamos asignado al dispositivo.
- Por último, para obtener los datos, es necesario conectarse al broker, suscribirse al tópico, y procesar los datos cada vez que se reciba un mensaje.
    - Creamos un fichero "*sub.py*" con este código.
    - Creamos un directorio "*templates*", y dentro creamos un fichero "*dashboard_secciones.html*" con este código.

Ejecución y visualización

- Conectamos la placa LoRaWAN a la Raspberry Pi que tenga los sensores, y ejecutamos el código "*pub.py*" para comenzar a enviar los datos a la red de TNN.
- Desde cualquier otro dispositivo, ejecutamos el código "*sub.py*" para conectarnos al broker MQTT de TNN y leer los datos recibidos.

- Abrimos una ventana del navegador y accedemos a *http://localhost:8080* para visualizar una gráfica con los datos recogidos de los sensores.

  > ⚠ Para visualizar correctamente la gráfica con los datos de los sensores, debemos disponer de conexión a internet desde el dispositivor suscriptor. Esto se debe a que el HTML de la interfaz web emplea un script descargado de internet para el renderizado de las gráficas.