ELV-LW-Base
Development board
for LoRaWAN

ELV-BM-TRX1

Getting started

Version 1.1

Author: JK

Status of document: DRAFT

Confidentiality level: PUBLIC

Table of Contents

1. Introduction

This document provides information on how to get started with your own firmware projects for the ELV-LW-Base development board.

It includes information about the development environment and the required programming adapter as well as the necessary steps to use LoRaWAN and to display transmitted data using a community-based solution.

2. Software

The software used here is based on the LoRaWAN End Node Project of the *STM32Cube MCU Package*[1] *for STM32WL series* by ST Microelectronics. This Package contains, among other things, a driver for the radio module, a LoRaWAN v1.0.3 implementation, some basic safety aspects and many examples of how to use the integrated peripherals.

For a first overview of the structure of this project, *AN5406*[2] *- How to build a LoRa application with STM32CubeWL* can be consulted.

To get started it is recommended to use the STM32CubeIDE[3], which is the Integrated Development Environment for STM32 microcontroller. This allows easy import and use of the attached example project.

Note: The example project was developed with version 1.7.0 of this IDE.

[1] https://www.st.com/en/embedded-software/stm32cubewl.html
[2] https://www.st.com/resource/en/application_note/an5406-how-to-build-a-lora-application-with-stm32cubewl-stmicroelectronics.pdf
[3] https://www.st.com/en/development-tools/stm32cubeide.html

2.1. Import

Once you have installed the IDE and downloaded the project, you can import it.

Open the IDE, navigate to *File->Import* and select *General->Existing Projects into Workspace* in the appearing window as shown in Figure 1**.**

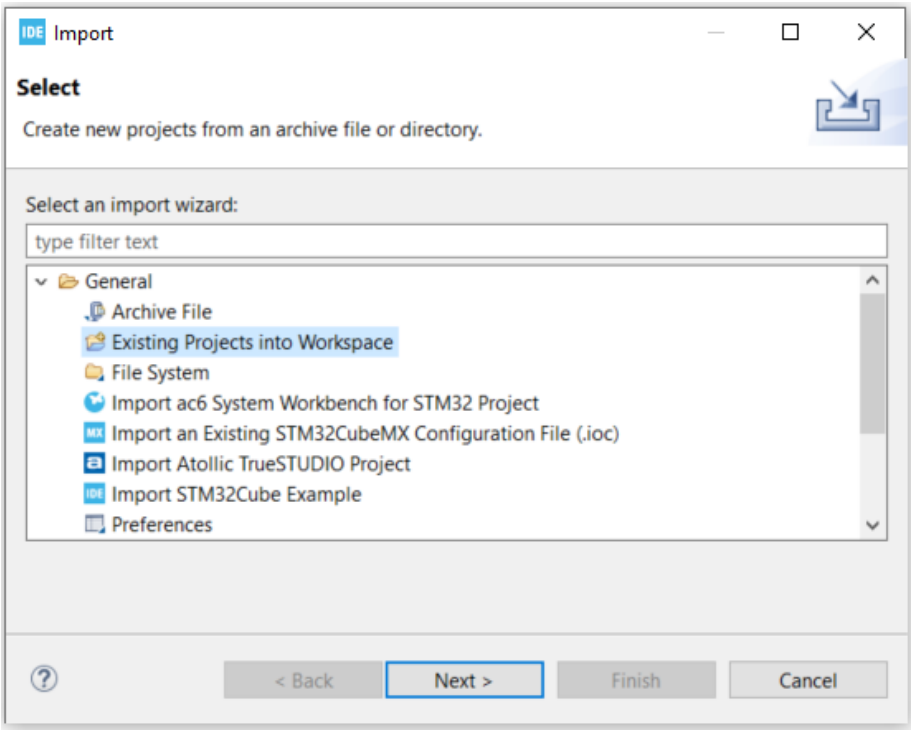


Figure 1: STM32CubeIDE import dialog

In the succeeding window you can specify either an archive file or the already unpacked project directory. After that the chosen project should be listed under *Projects:* and you can proceed with *Finish*.

3. Application

To make it easy to get started with the ELV-LW-Base development board, an application directory, which contains some basic implementations, has been added to the underlying project.

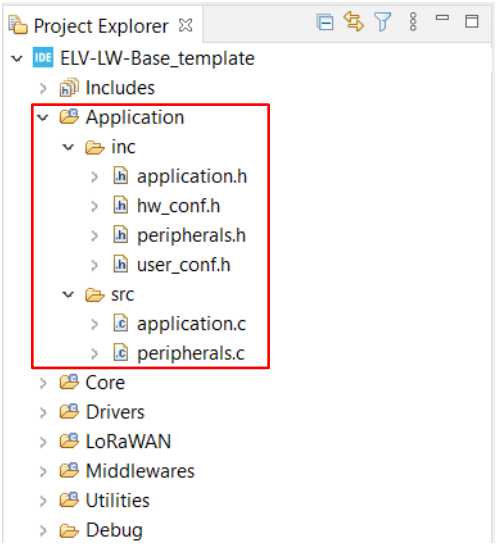
Figure 2 shows this directory unfolded and exposes its contents:

hw_conf.h: contains pin definitions

user_conf.h: used to set LoRaWAN joining parameters
- Device EUI
- Join EUI
- App Key

peripherals. *: some implementations for the LEDs and the user button

application. *: a minimal application whose flow chart can be taken from Figure 3
- Joining (including retry on failure)
- Send data (on button press and cyclic)



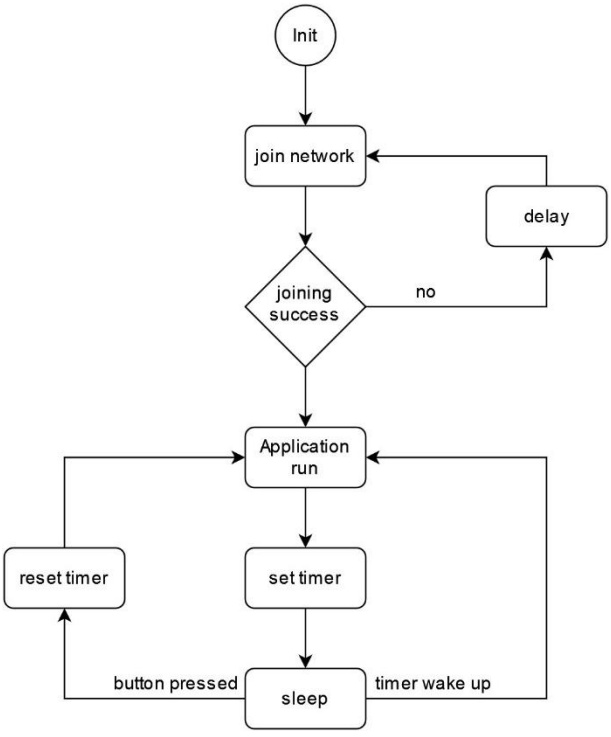


Figure 3: Firmware template flow chart

3.1. Commissioning

First of all, you will need to get LoRaWAN joining parameters in order to connect your device to a LoRaWAN System. These are:

- LoRaWAN Device EUI
- LoRaWAN Join EUI (App EUI in LoRaWAN version < 1.1.0)
- LoRaWAN App Key

In this example we are using The Things Network as network operator and therefore we get the needed parameters from there.
To do so you have to create an account[4] and sign in to the console they are offering. On the following site there is a choice between gateways and applications. If you need to register your own gateway you should do that first.
At the top of the page (Figure 4) you have to enter a gateway ID and your gateway EUI. You can find the EUI in the documentation of the gateway.
Now the frequency plan must be set according to the region EU868. To do so scroll down, find *frequency plan* and set it to *Europe 863-870 MHz (SF9 for RX2 - recommended)*.

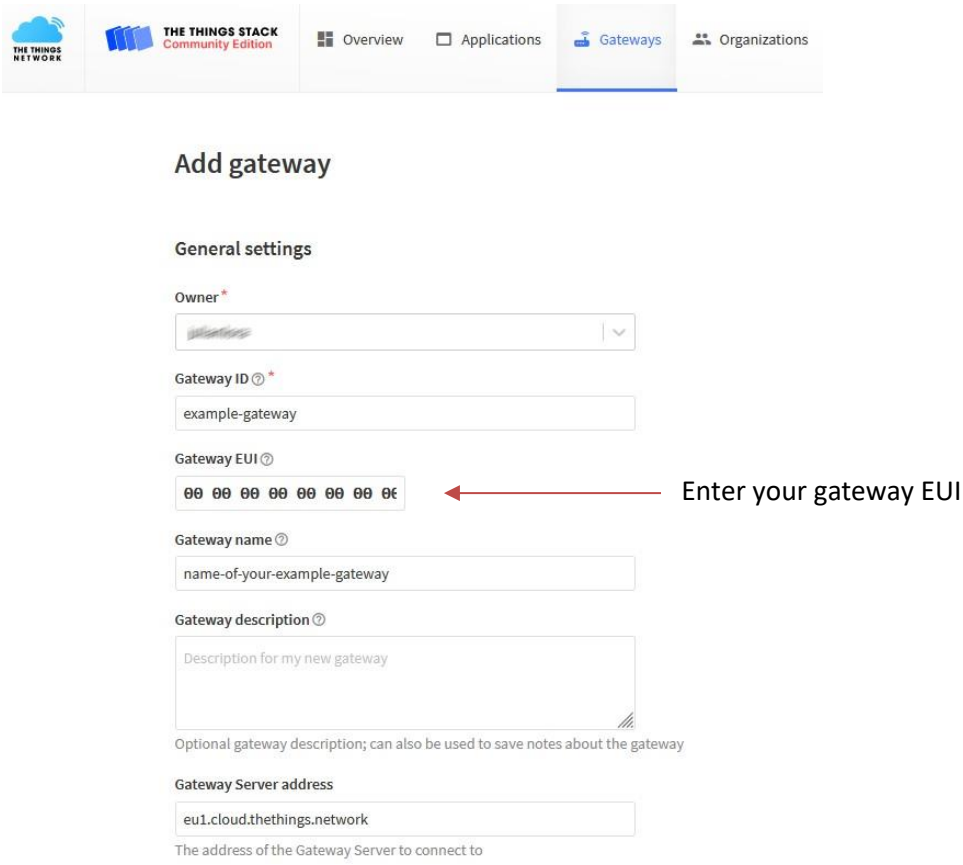


Figure 4: Adding a gateway to your TTN console

[4] https://account.thethingsnetwork.org/register

After adding your gateway, you can switch to the *Applications* tab for registering your first application. Choose *Add Application* and enter the required information. Then confirm with *create application* (Figure 5).

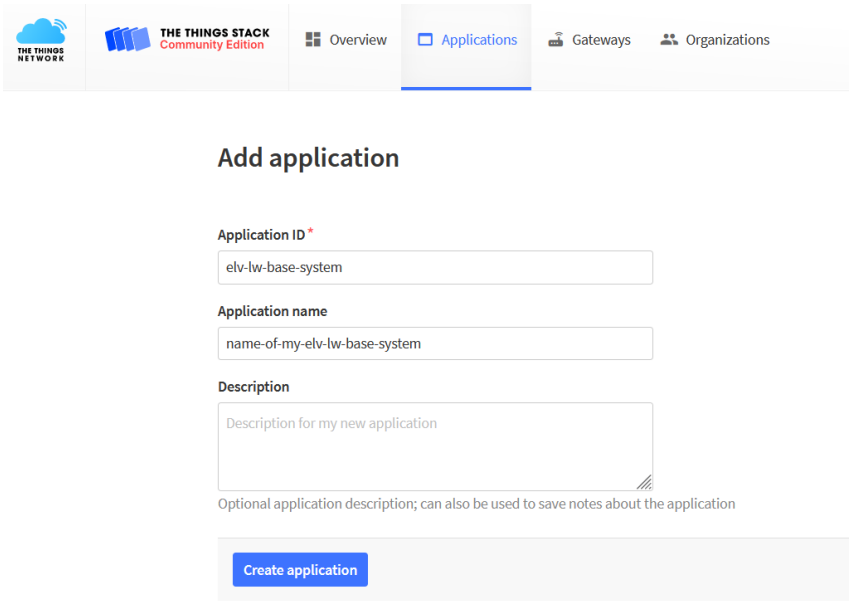


Figure 5: Adding an application to your TTN console

After that you are ready to create your first LoRaWAN device. Search for the *End devices* tab on the left side and there click on *Add end device.* Switch the type of registration to *Manually* and fill in the information according to Figure 6.
Note: Of course, the data attached to the ELV-LW-Base can also be entered here.

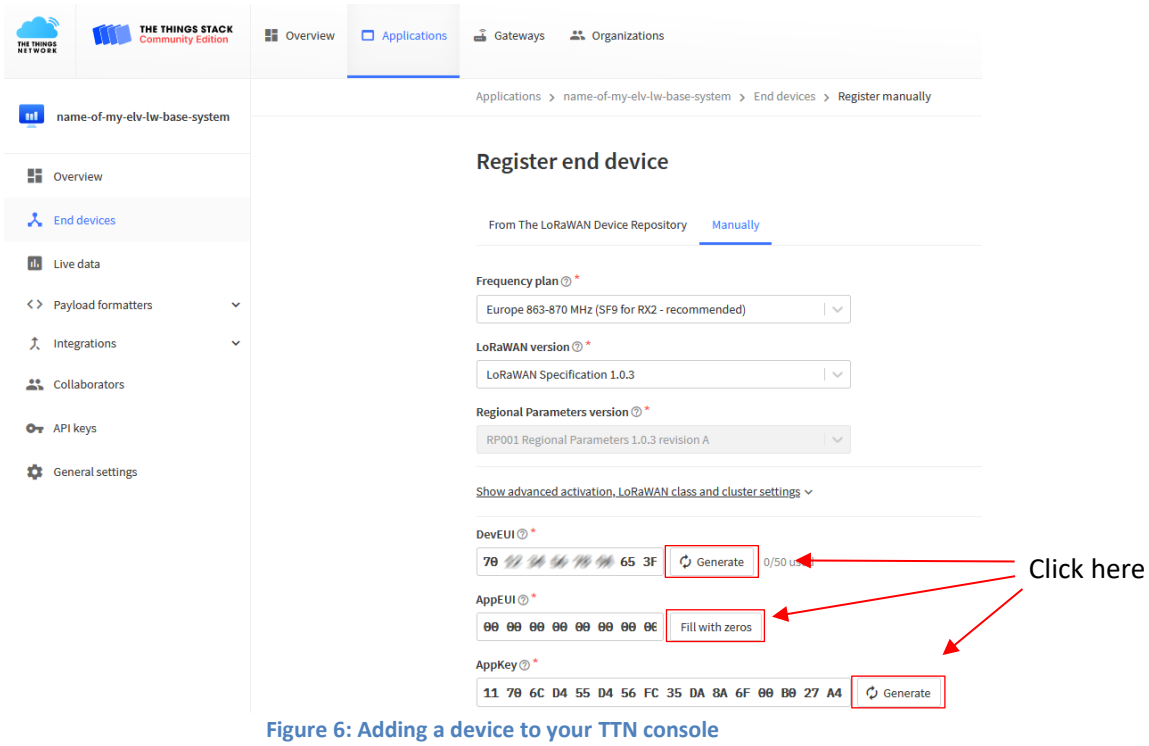


Figure 6: Adding a device to your TTN console

The last step is adding the data to your firmware. Open the project and in the application folder navigate to user_conf.h and enter your generated device EUI, JoinEUI (AppEUI) and your App Key.
Note: Do not use the data displayed here for your application.

```
28 /*
29  *  Definitions ------------------------------------------------------------
30  */
31
32 #define USER_CONF_LORAWAN_DEVICE_EUI            { 0x70, [redacted], 0x65, 0x3F }
33 #define USER_CONF_LORAWAN_JOIN_EUI              { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }
34 #define USER_CONF_LORAWAN_APP_KEY               11,70,6C,D4,55,D4,56,FC,35,DA,8A,6F,00,B0,27,A4
35
```

Figure 7: Customizing LoRaWAN joining parameter

After you have plugged in the USB cable you can connect a compatible programming adapter (e.g. ST-Link/V2), an appropriate cable adapter like Olimex Arm-JTAG-20-10 and load the firmware into the target (Figure 11).

Important note: See pictures for how to connect the adapter properly and double check the orientation of the plug. A wrong connection of the adapter could damage the board.

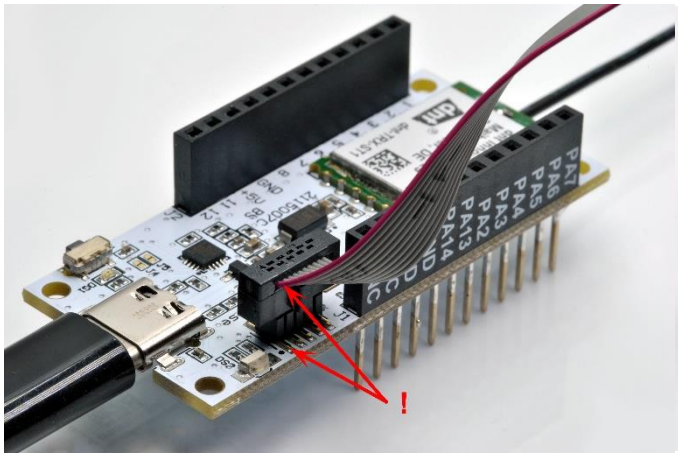


Figure 9: Correct alignment of the plug

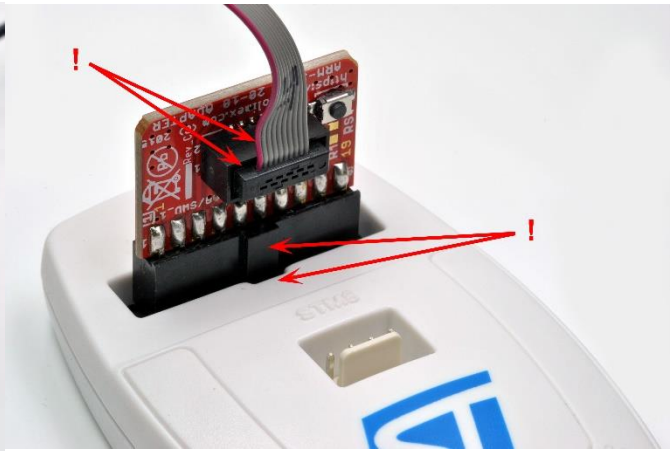


Figure 8: ST-Link V2 with Olimex adapter

Figure 10: ELV-LW-Base completely connected

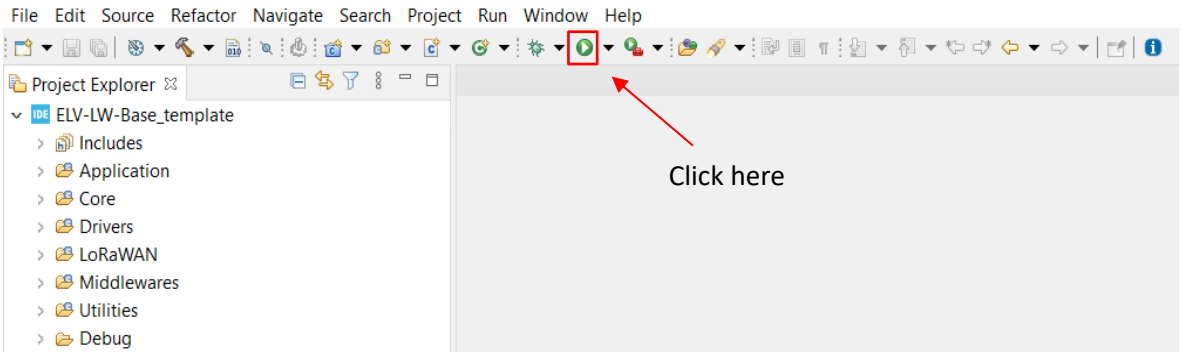


Figure 11: Load the firmware into the target

On startup the green LED will flash for a short time to show that the device is active and starts to try joining the LoRaWAN network. If a gateway connected to TTN is within the range of your device a *Accept join-request* message will appear in the TTN console and the device starts working.
Note that the accept message does not necessarily mean that the joining was successful as the transmission of the response to the device may have been interfered with. Therefore, the success/failure will be indicated by the LEDs green/red a few seconds after the joining has started.

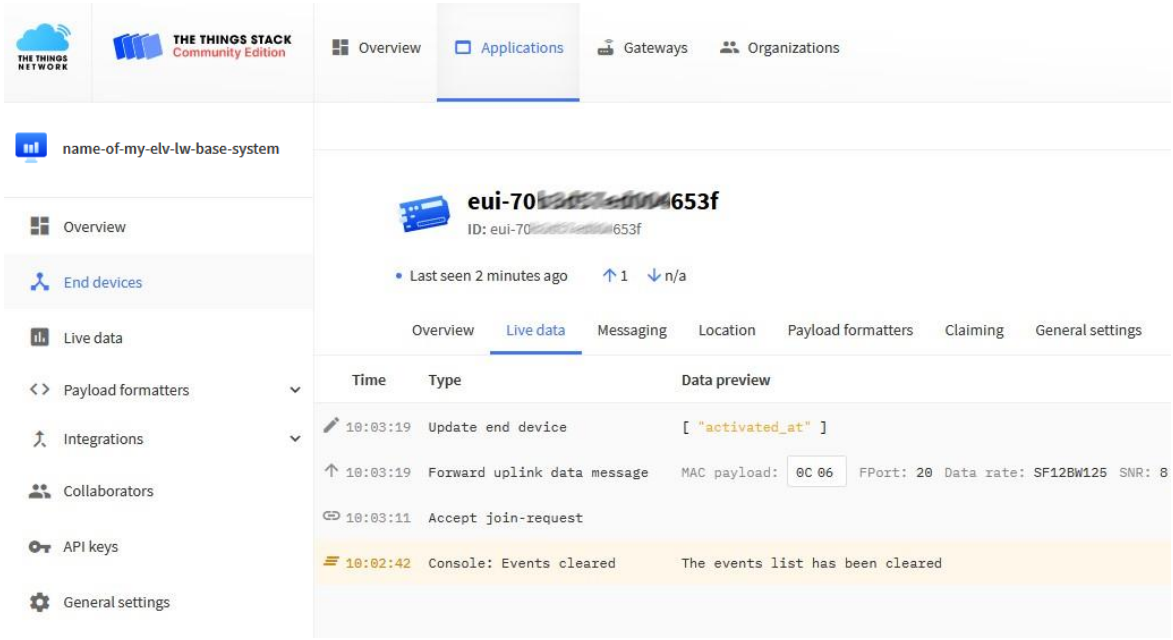


Figure 12: First messages in the TTN console

If you have set up a serial connection (Chapter Serial Output) you should now also see the first messages in the serial terminal (Figure 13).
Note: If the lines do not start flush left you may need to change the terminals newline setting to CRLF.

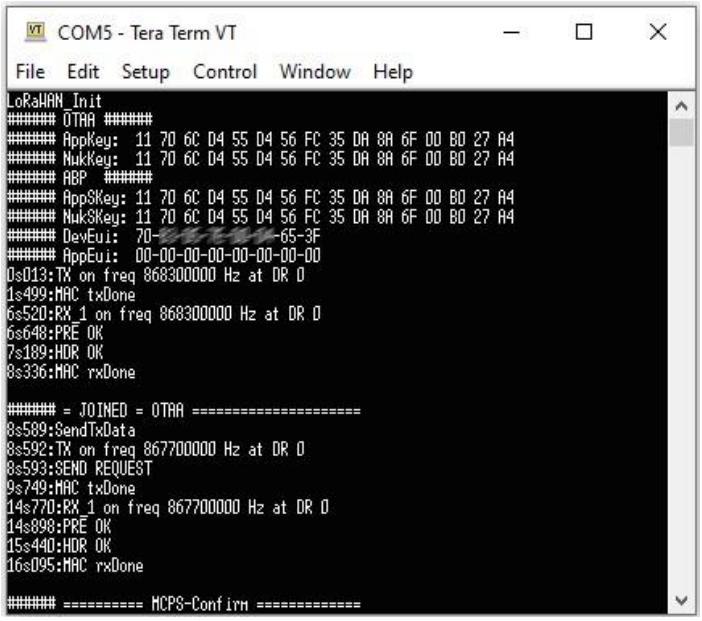


Figure 13: Serial output at startup

4. Serial Output

To see what the device is doing you have the possibility to track serial messages sent by your device on your development computer.

In this example the open-source tool Tera Term is used for this.

First you need to choose the serial port the adapter is connected to.

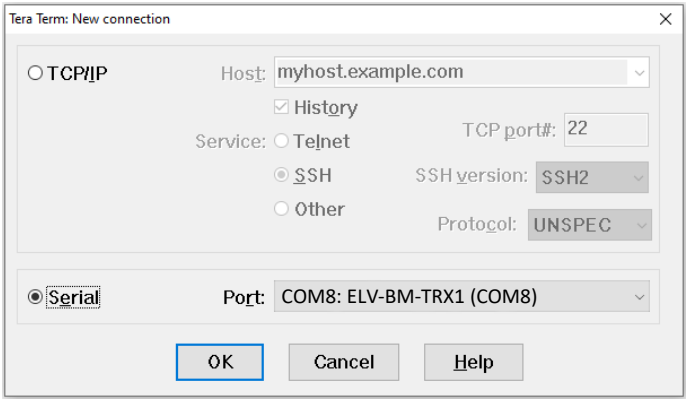


Figure 14: Connect to serial adapter

After this be sure that the baud rate is set to 115200. In Tera Term this can be found under *Setup->Serial port.*

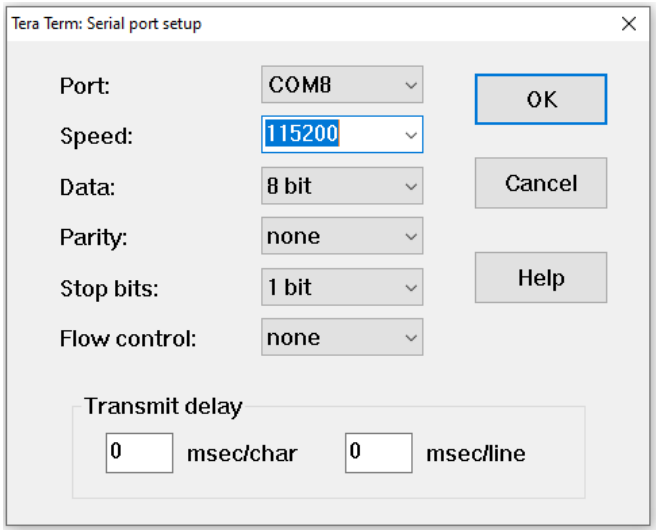


Figure 15: Choosing baud rate

5. Certification

The ELV-BM-TRX1 platform is CE approved and thus meets the requirements that apply in the specified applications. This condition is given if the module is operated with the specified firmware and this firmware is not changed in areas relevant for conformity.

This means that the module is basically equipped with a reference certification, which proves that operation within the legal requirements is possible. In order to maintain this, it is strongly recommended not to change the firmware with regard to the configured radio parameters. This means that you should not make any changes to the middleware in the *LoRaWAN* and *SubGHZ_Phy* directories.

I Appendix

A. List of figures

B. Change history

Version	Datum	Status	Description
1.0	08. Oct. 2021	Draft	First Edition
1.1	15. Aug. 2022	Draft	Renamed Base