# Windigipet 2025 – BoosterMonitor

31-10-2025 – Jens Krogsgaard, Copenhagen - Denmark

## Table of contents



# 1. Summary

In Windigipet, you can view the real-time values for your boosters – that's great – but it would also be nice if there were an easy way to see how these values develop over time.

It's actually only recently, with Windigipet version 2025.0b, that it has become possible to transfer booster data from an Ecos to Windigipet.

Since I have an Ecos, I naturally had to try it right away, and with Markus's helpful guidance, it went smoothly.

Soon, the desire arose to be able to save the values and display the development graphically. Since Windigipet 2025 also has the option to connect to MQTT, and I've had good experiences with this technology, I got the idea to create a small program that displays booster data from MQTT as line graphs.

I've tried to make the program as flexible and configurable as possible, so hopefully it can be used by other Windigipet 2025 users. You can choose between Danish, German, or English language, and you can configure from 1 to 6 graphs showing the current consumption of your boosters.

# 2. Prerequisites

It is assumed that you are using Windigipet 2025 and that you have defined one or more boosters in the Booster Management section. In addition, you must have established a connection to an MQTT broker or be willing to try setting one up.

If you already have Windigipet 2025 installed (demo, small, or premium), you can create an MQTT broker locally on your PC for free, or you can set it up in the cloud.

Read more about MQTT and Windigipet 2025:

- MQTT in Windigipet
- Windigipet 2025 – MQTT integration

Read more about Booster data in Windigipet

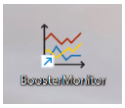
- Booster in Glesibild

# 3. Installation

The BoosterMonitor program can run on a Windows PC – typically the same one where Windigipet is running.

Download BoosterMonitor

The program is installed like a regular Windows application. An icon is created on the desktop from which the program can be launched.

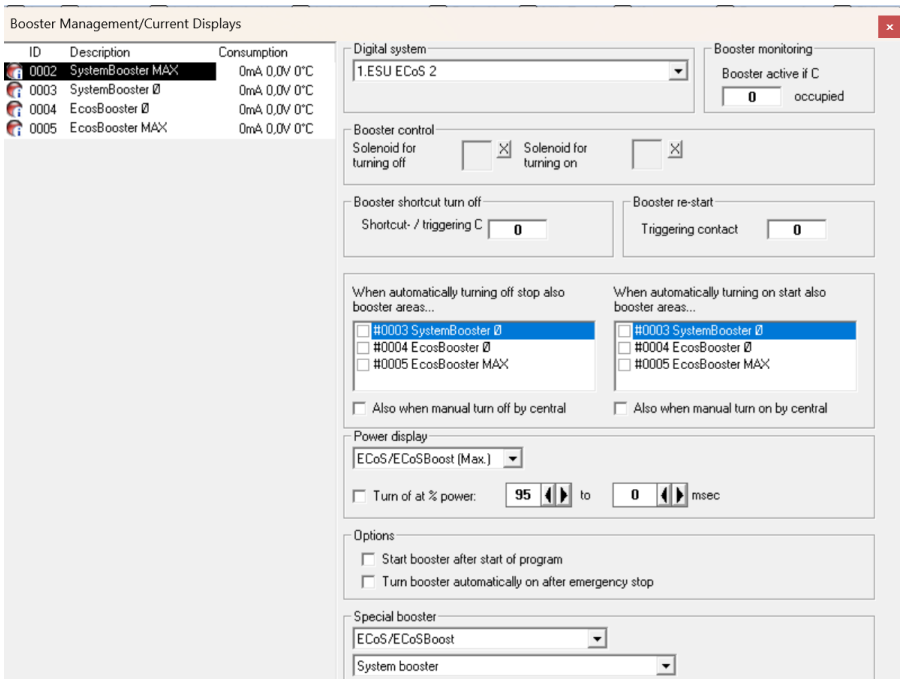



# 4. Prepare Windigipet 2025

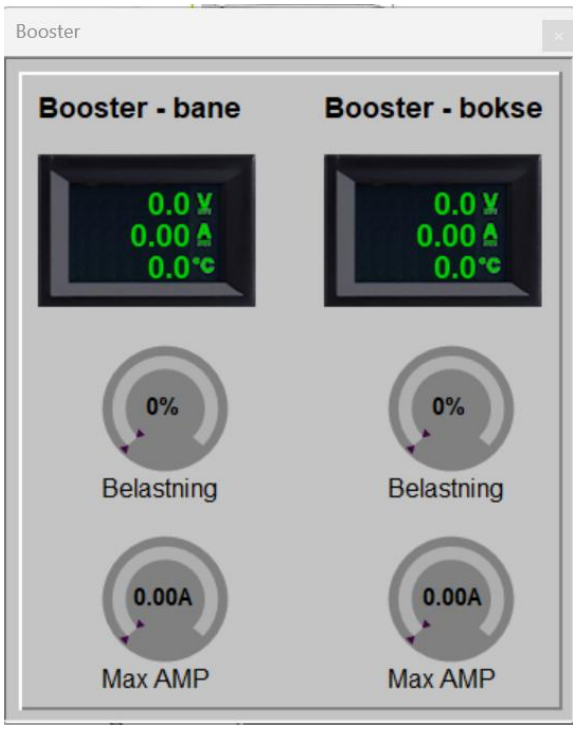
## 4.1. Boosterdata in Windigipet

In this document, you can read in detail how to transfer booster data to Windigipet and how it is displayed in a suitable instrument on the screen. Booster in Glesibild

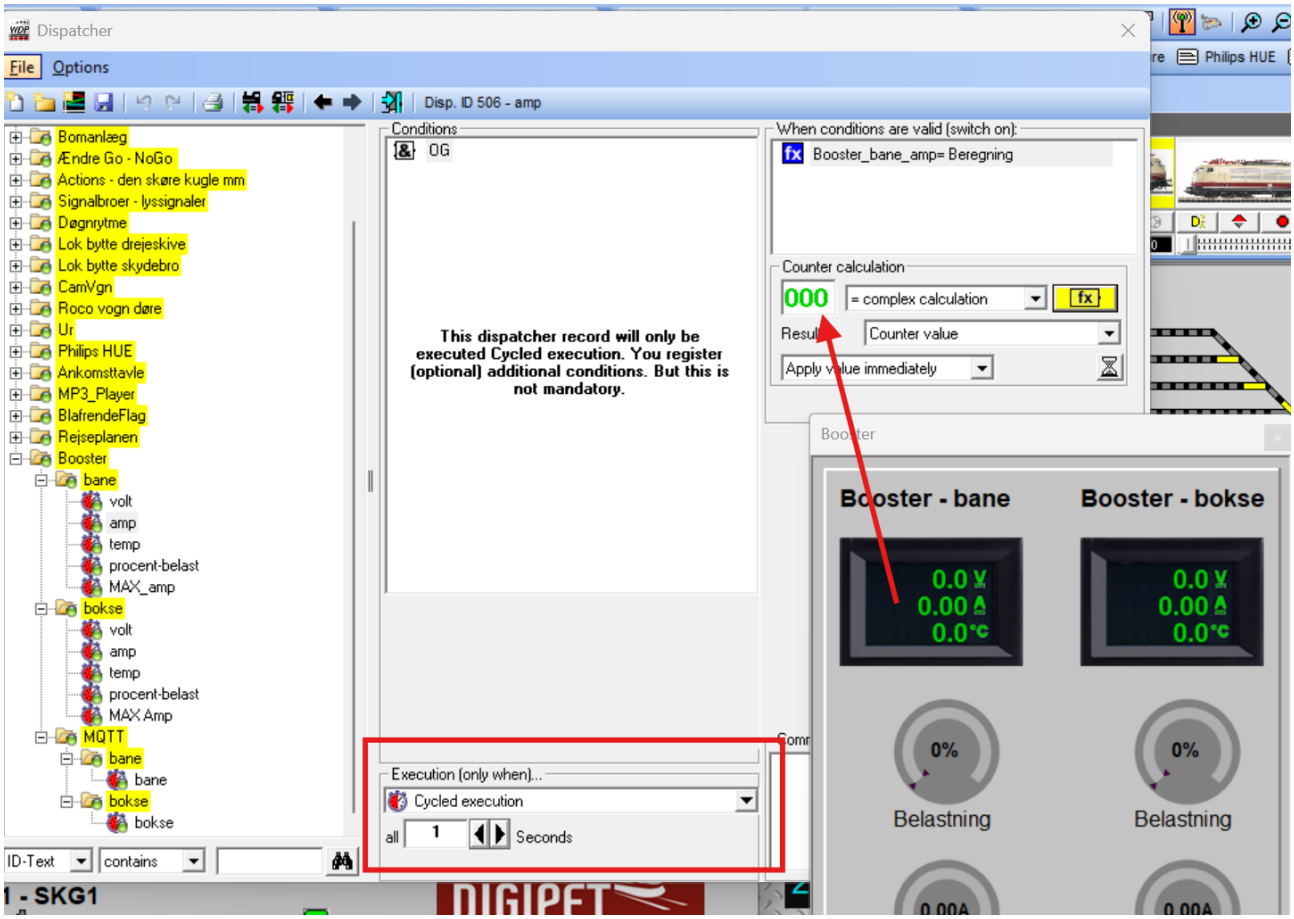
Step 1 – Define the boosters in Booster Management:



Step 2 – Draw instruments in the Track Diagram Editor:

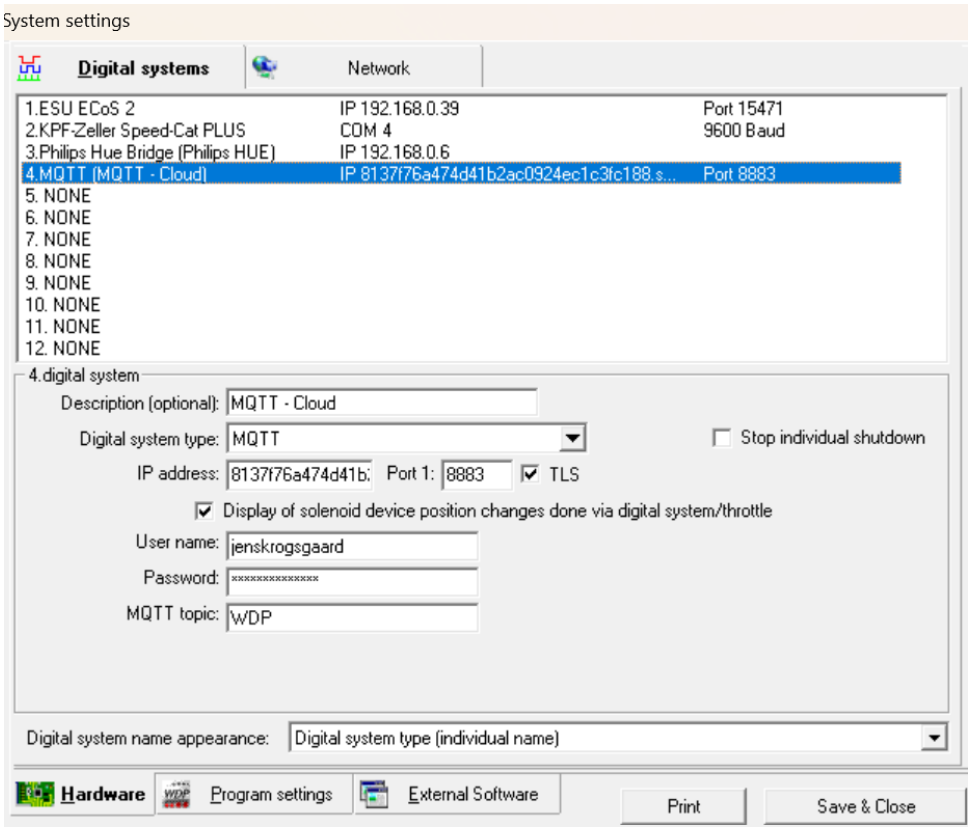



Step 3 - Transfer values to the instruments:



## 4.2. Send Boosterdata to MQTT

The data must now be sent to MQTT. If you haven't already done so, you now need to create a connection to an MQTT broker. See the links earlier in this document for reference:

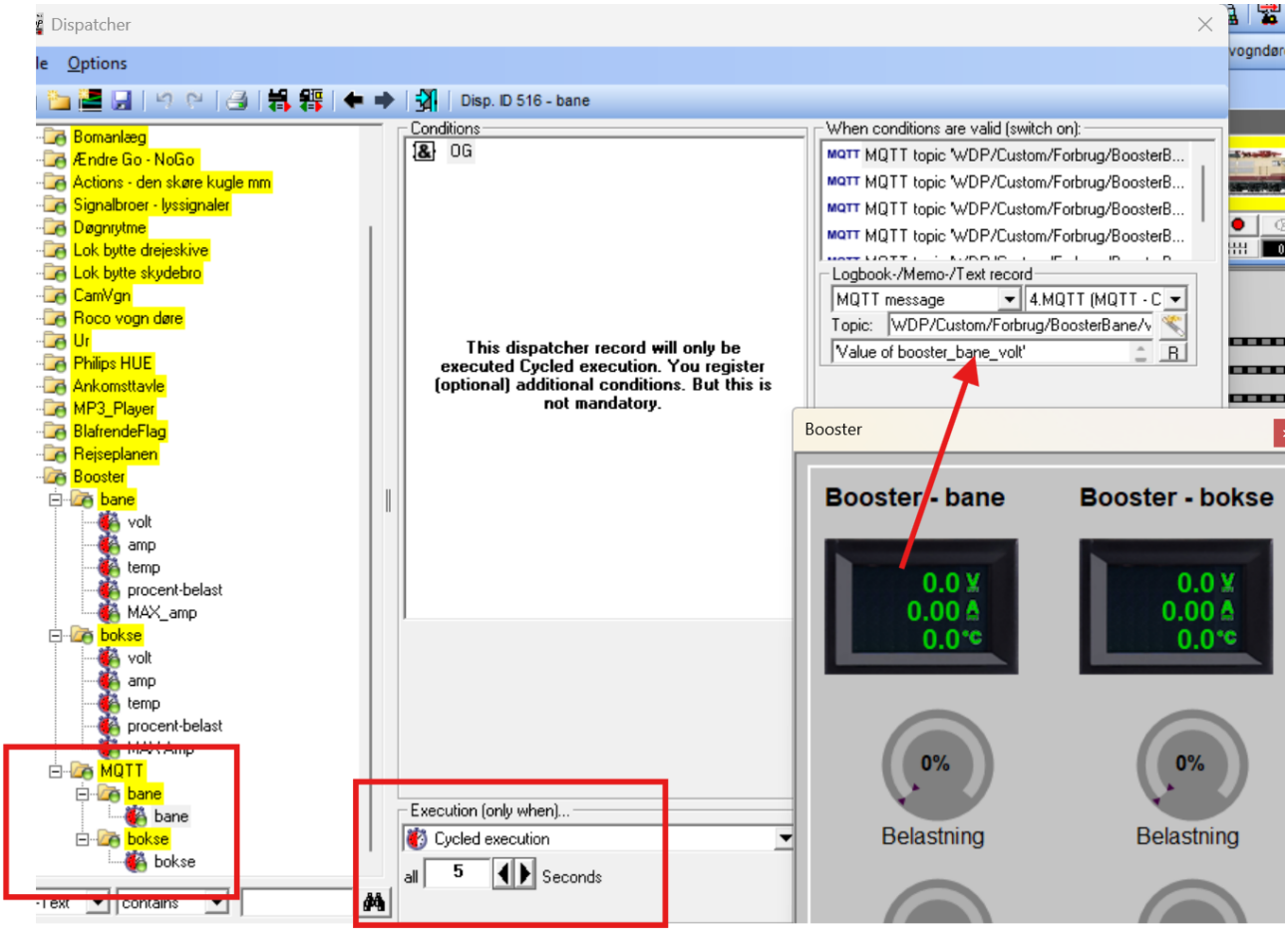



In Dispatcher, we already transfer the booster values from Booster Management to the instruments – the gauges – that we have drawn in the Track Diagram Editor. We now need to take these values and send them to our MQTT broker at a fixed interval – for example, every 5 seconds.

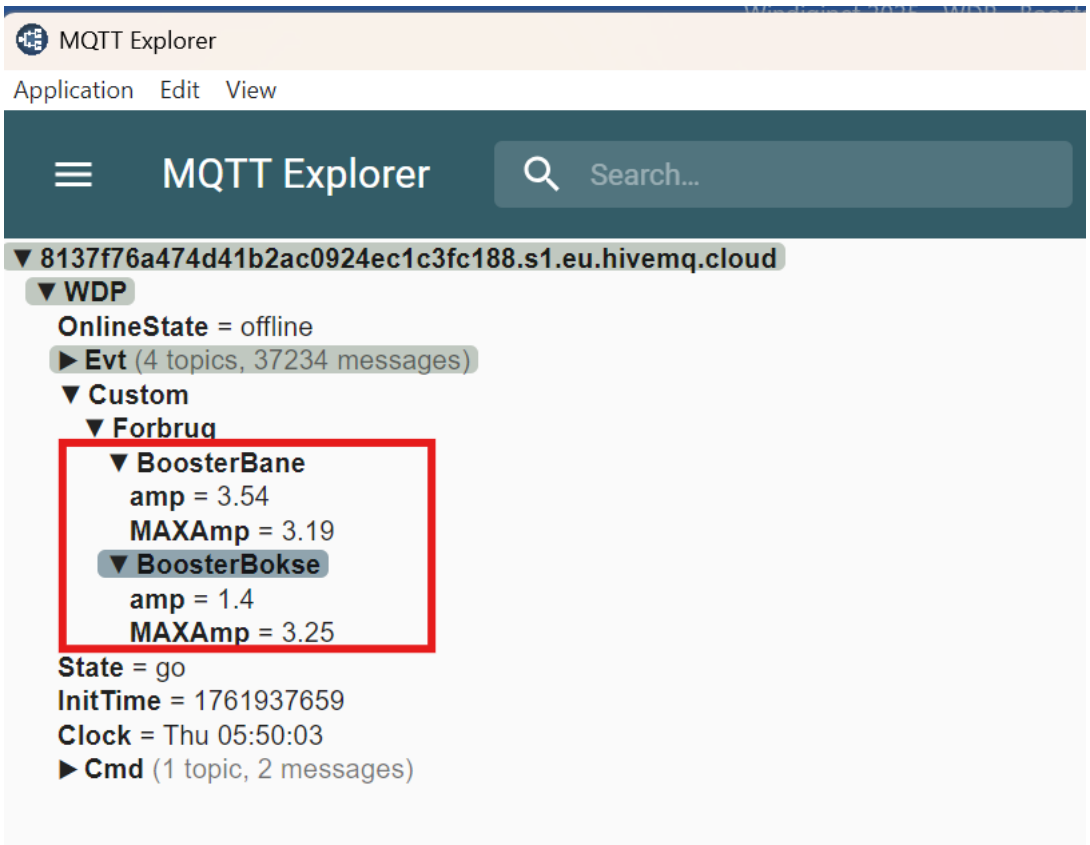
It is important here to specify a Topic (address) in MQTT where the data will be delivered.

Typically, the Topic starts with: WDP/Custom/ – followed by a unique identifier. Here's an example from my Danish setup :) :

WDP/Custom/Forbrug/BoosterBane/volt

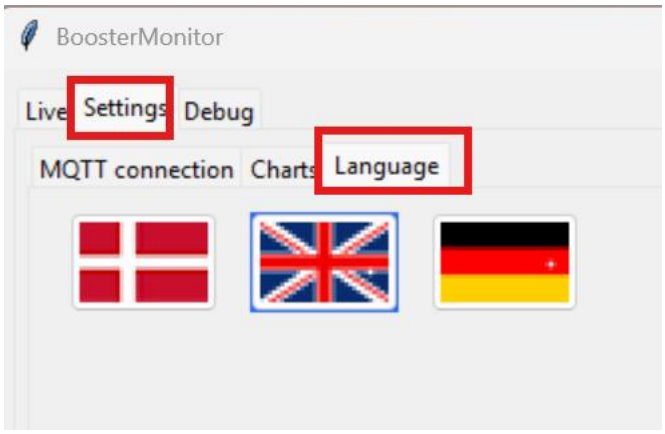



Once we have the above working, we should be able to see the booster values in an MQTT explorer – for example, like this:

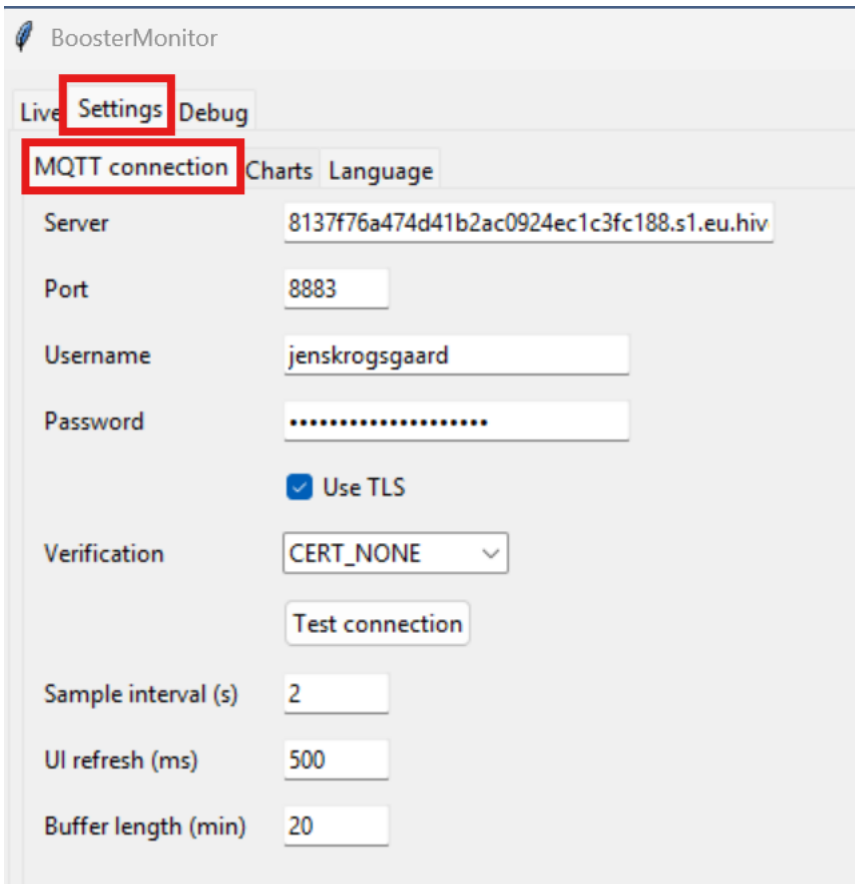


# 5. Configure BoosterMonitor

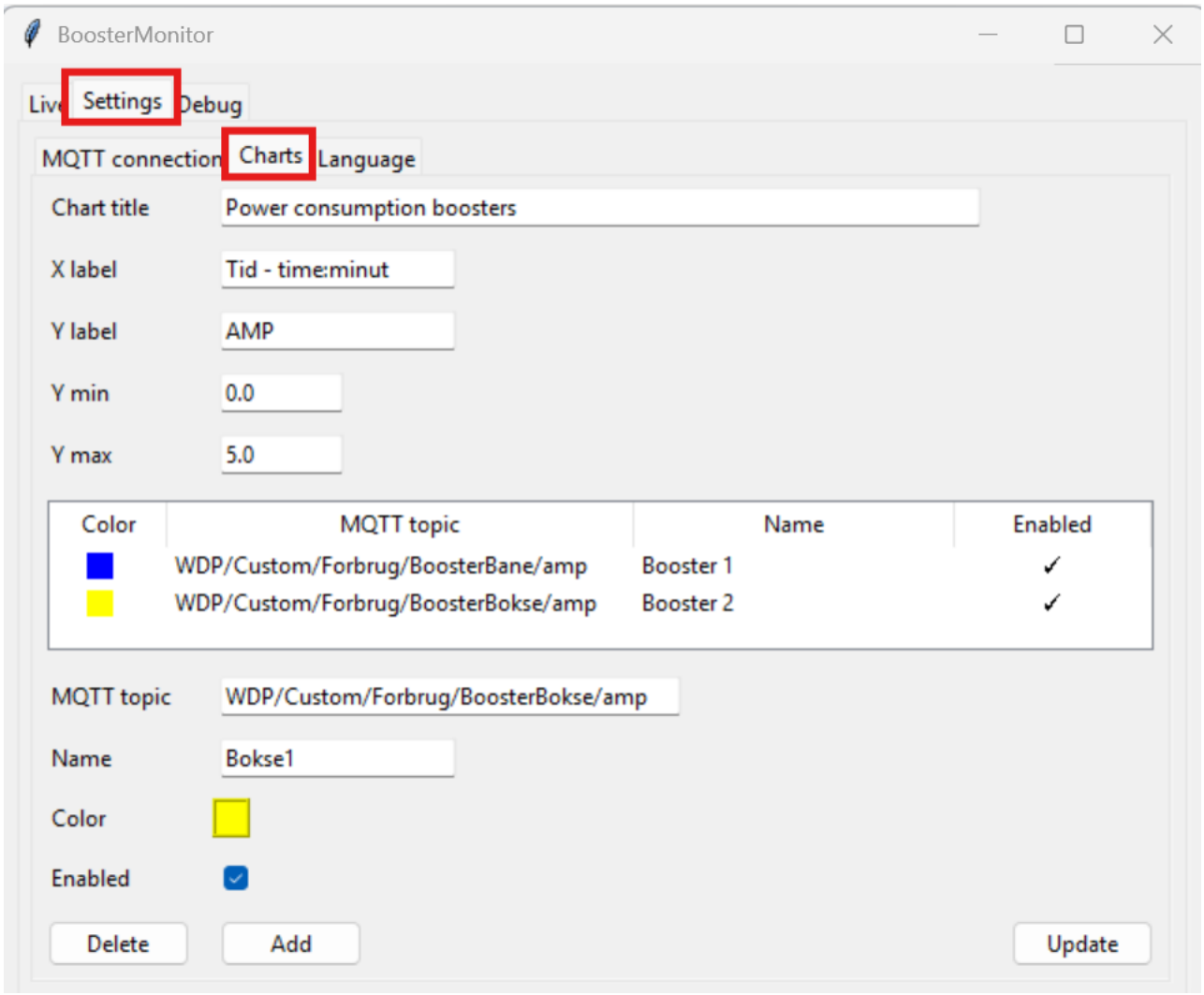
Start by selecting the language as follows:




Then select your MQTT connection – corresponding to the one you defined in Windigipet:

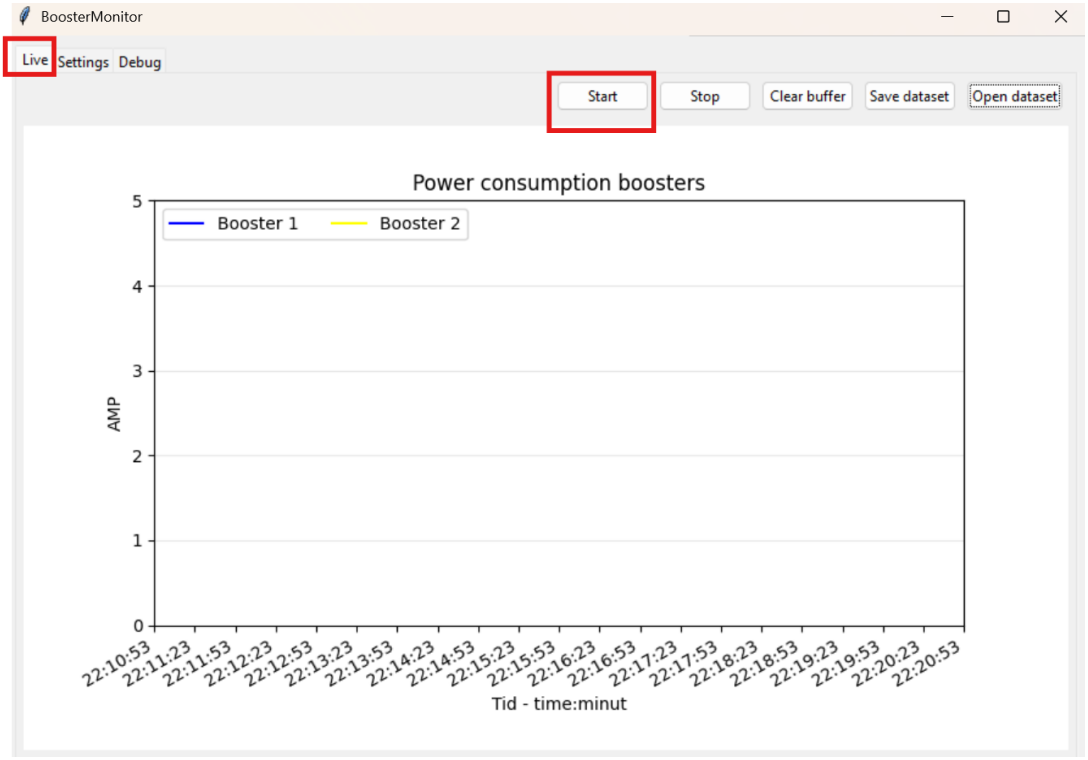


We now need to define up to six line graphs. The starting point is the Topics we have defined in the MQTT broker. We need to choose a color and a name for each graph:

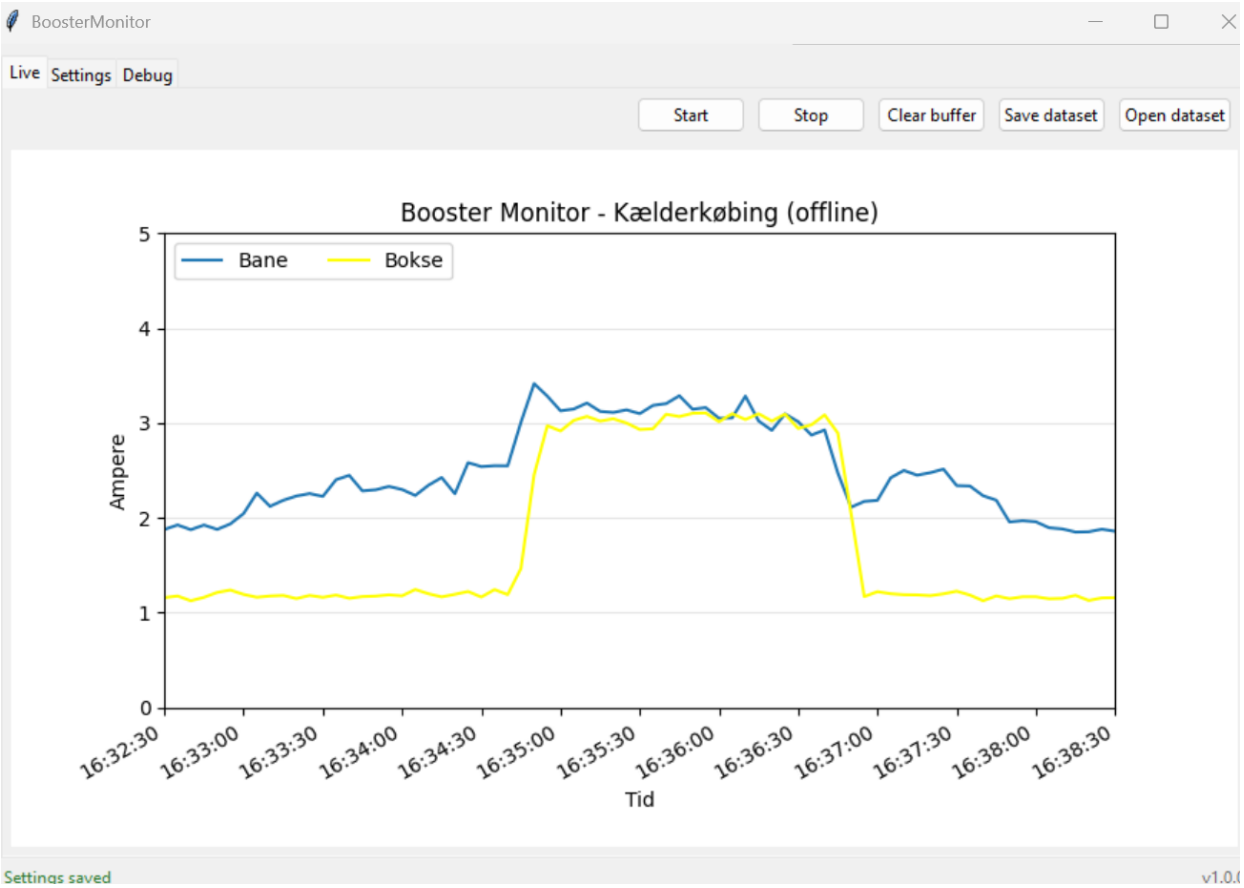


# 6. Start data collection

Press the Start button – and the data will now be transferred to the program:




The data will now begin to appear from left to right – for example, like this:



You can stop the data transfer by pressing **Stop**. The data can be cleared from the display by pressing **Clear Buffer**. You can save a dataset for later analysis by pressing **Save Dataset**, and likewise, you can load a saved dataset by pressing **Open Dataset**.

In the Debug window, you can view various log messages as well as any error messages. These can be copied and saved for later use.