**HOW TO USE THE HEATING CONTROLLER**

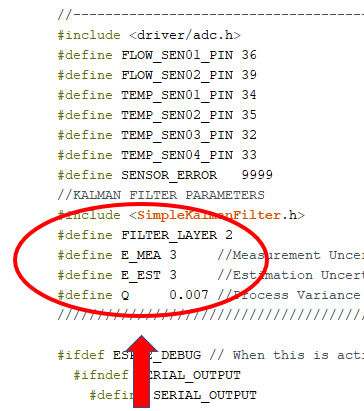
1. **4-20mA reading – temperature sensors, flow sensors**

Go into

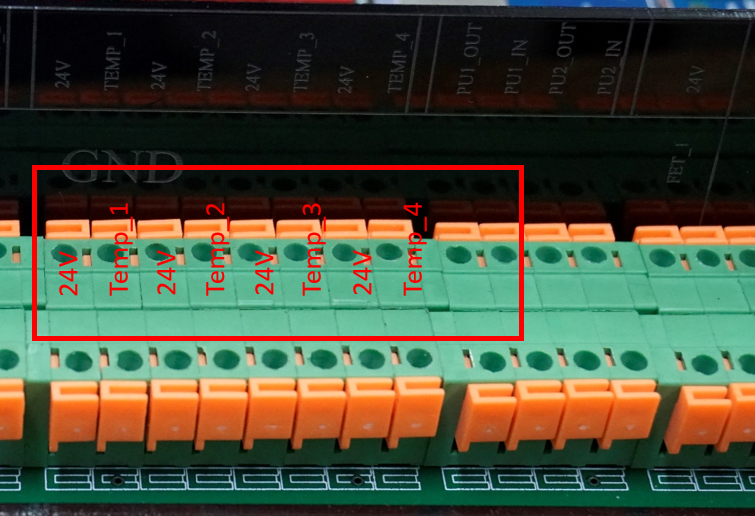
…\Heating Controller\_2019\Embedded Code\Heat\_Controller\separate\_components\ESP32\Standalone\_4-20mA\_ADC

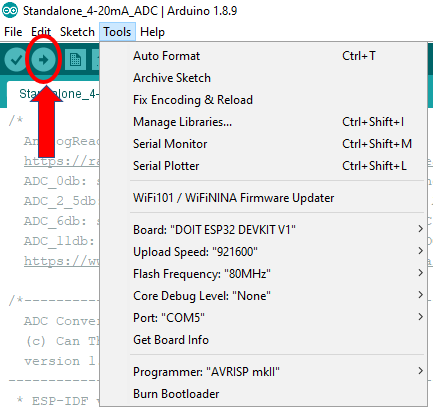
Open: Standalone\_4-20mA\_ADC.ino

You can change Kalman Filter layer and other parameter of the filter by changing these setting in config.h:

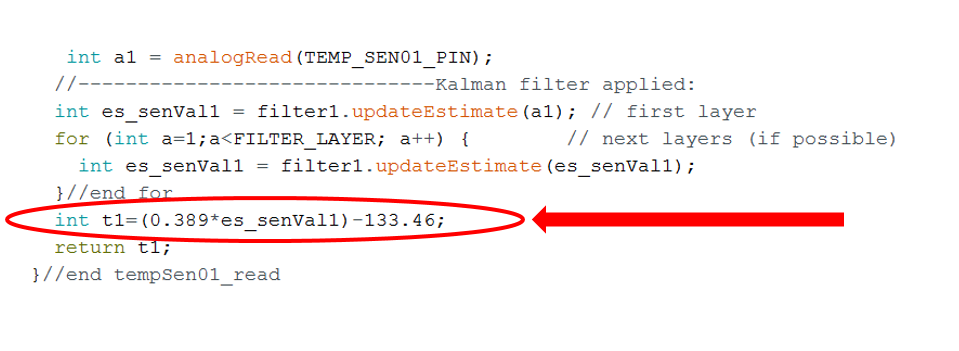


As you can see, the pins of the sensors are defined in config.h, you can follow the picture below to make the connection: (4-20mA protocol)

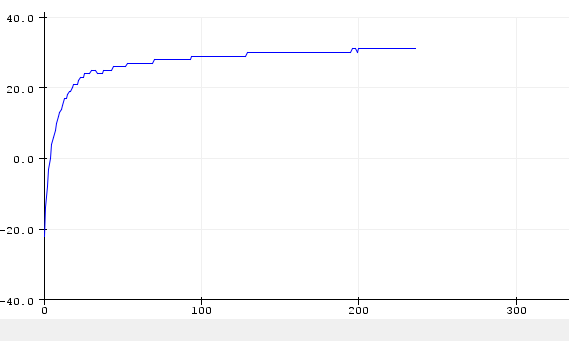


Compile and upload it (remember to turn off all bit switches before uploading)

Depend on the real result, you can calibrate the temperature output of the system using the function below, with es\_senval is the adc value 🡪 made this change to the main folder too



This is the desired result: (10s – 20s to stabilize)



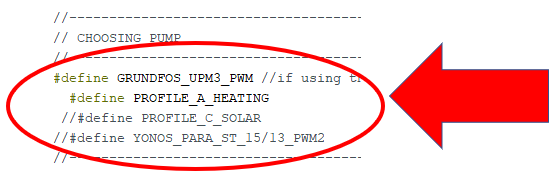
1. **PWM Pump**

Go into

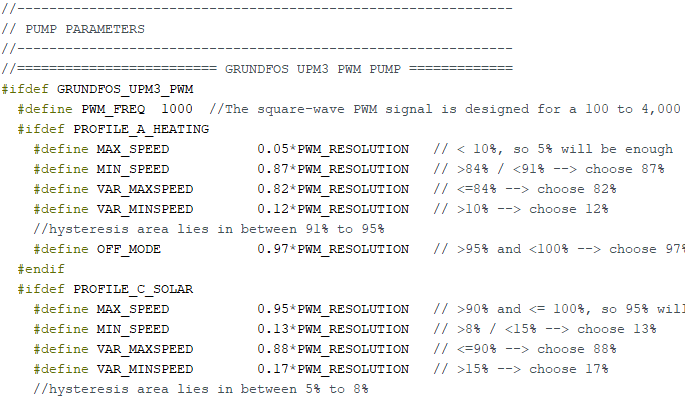
…\Heating Controller\_2019\Embedded Code\Heat\_Controller\separate\_components\ESP32\Standalone\_PWM\_Pump

Open: Standalone\_PWM\_Pump.ino

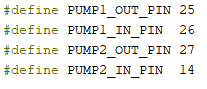
You can change the desire pump by changing the setting in config.h, only need to slash out the unwanted pump here, do not need to change anything below if you just want to change the pump.



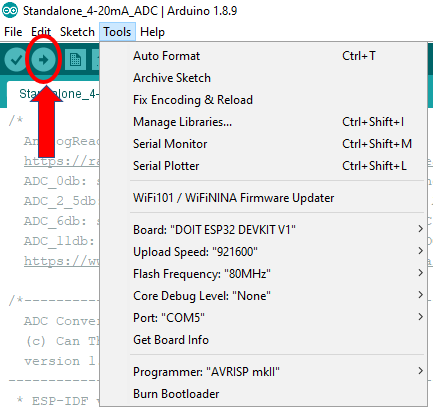
If you also want to change the parameter of the pump, you can change it in config.h. These parameters were pulled out from the datasheet of the pumps. Depend on real situation you can add or modify them



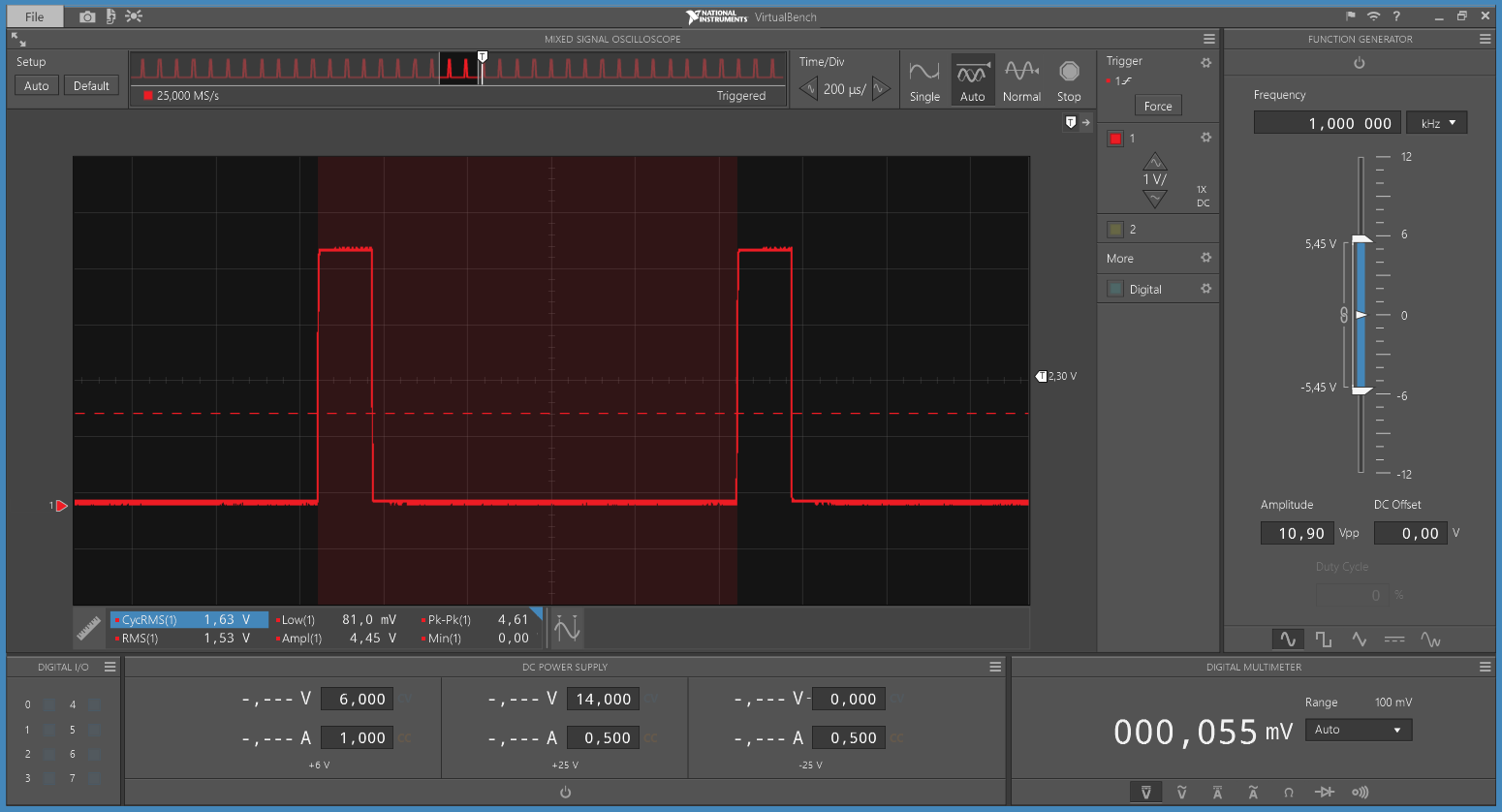
As you can see below, the pins of the pumps were defined. Connect the desired pump and start testing. Because we don’t have the pump so we can only manage to create the pwm signal, not on real pump

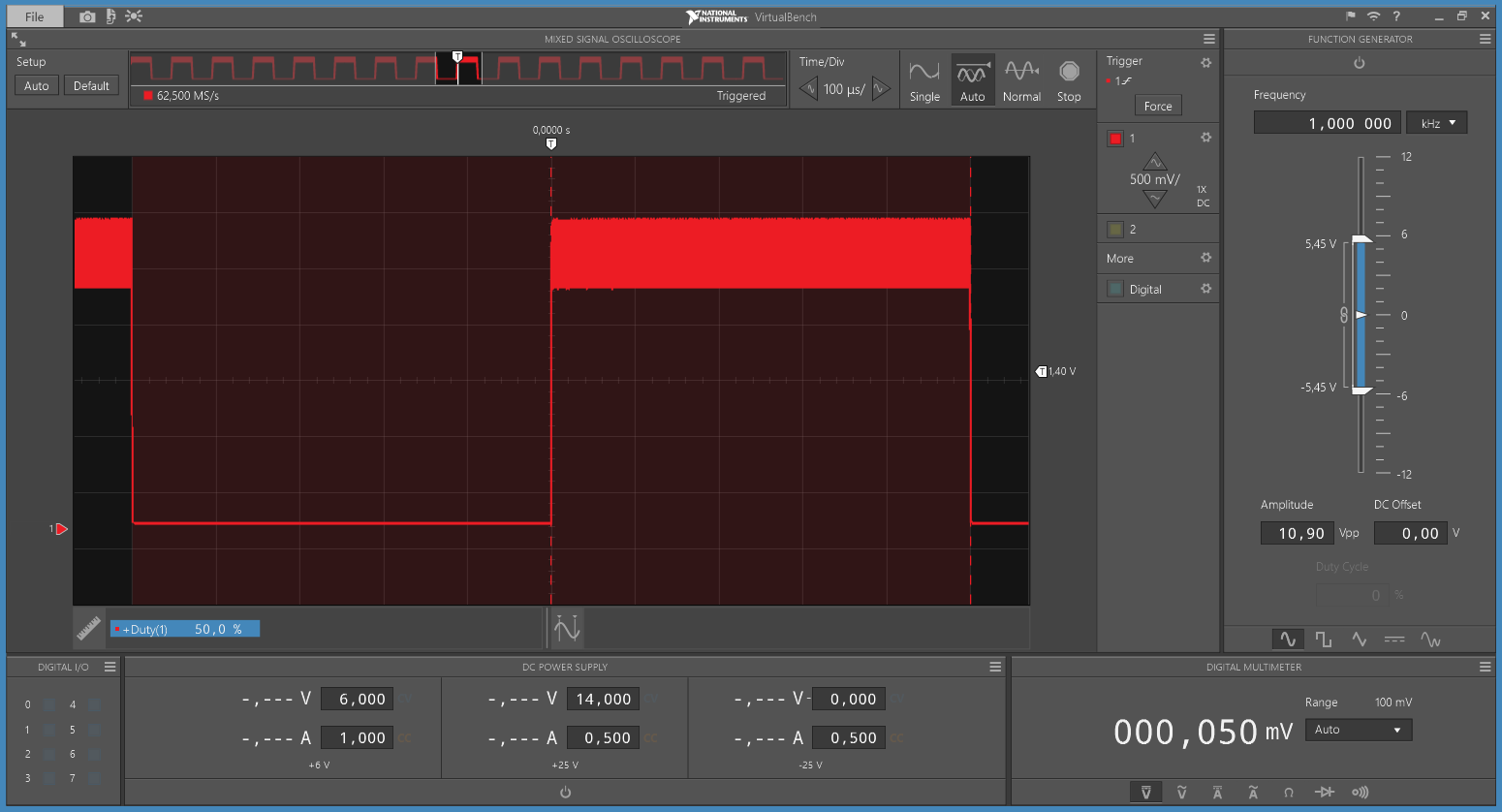


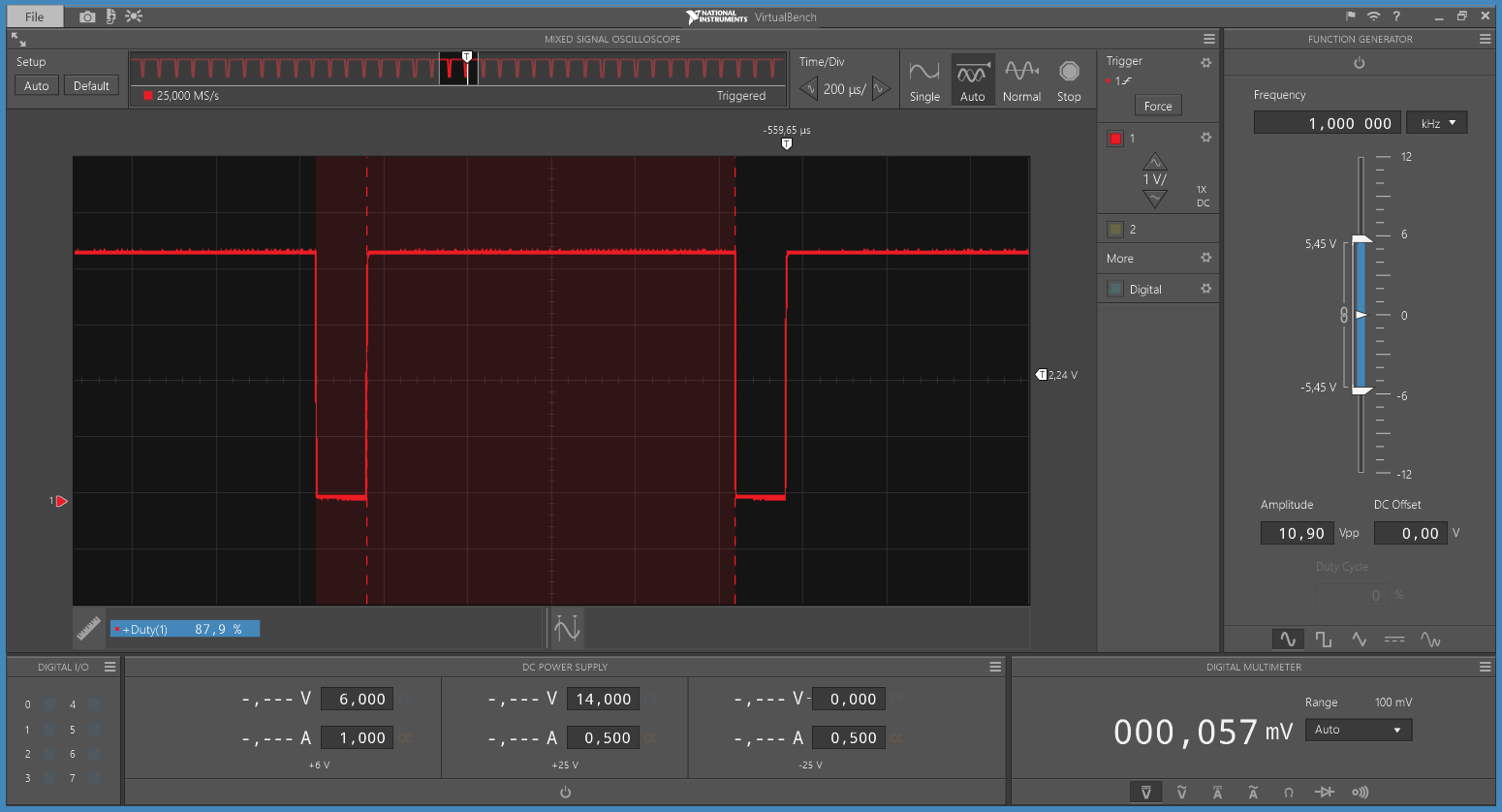
Hình mũi tên chỉa vô mấy cái port của bơm (2 bơm, mỗi bơm 2 chân in-out)

Compile and upload it (remember to turn off all bit switches before uploading)

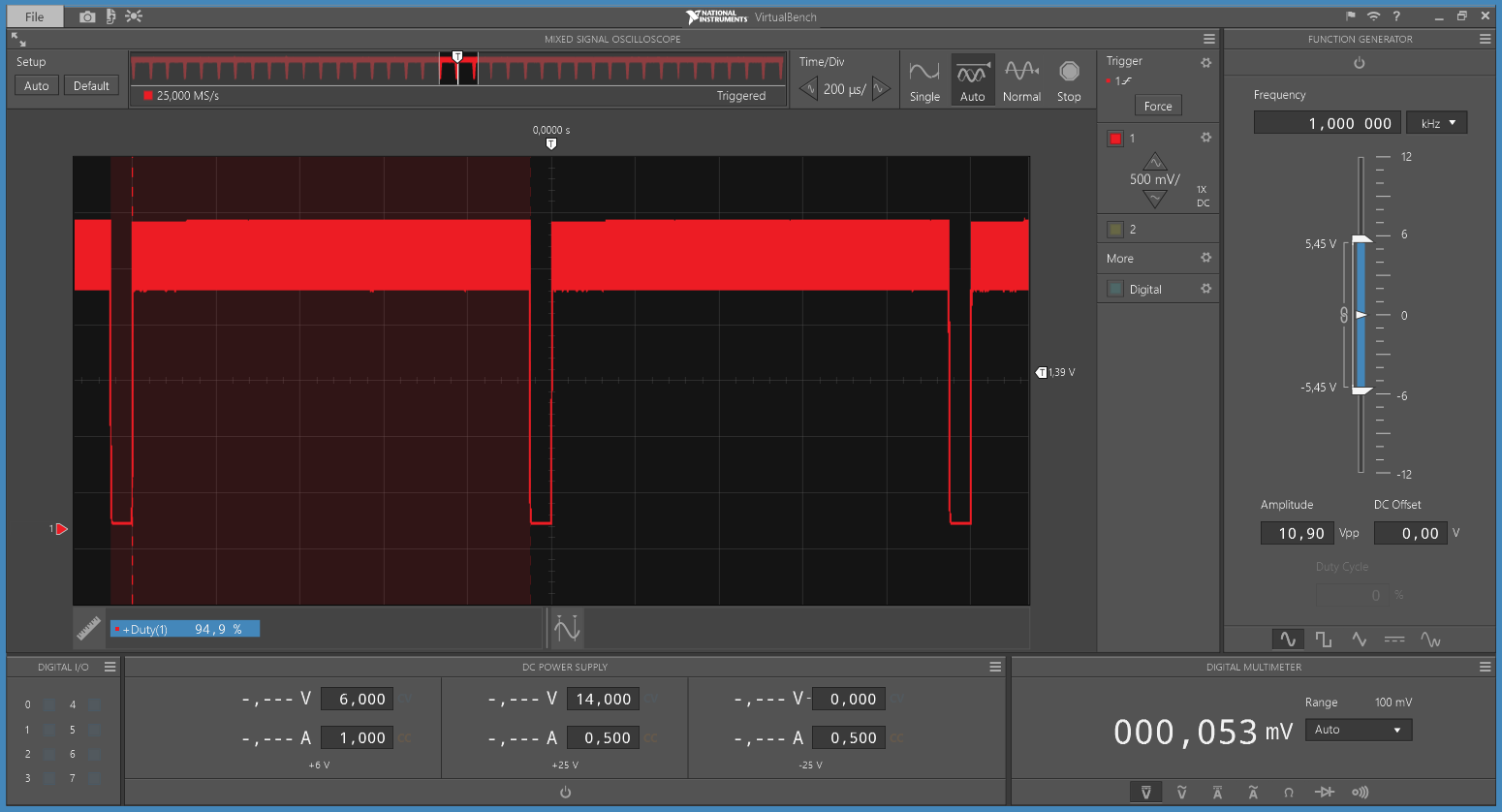
This is the output signal of pump1 start at minimum speed (pump Grunfos profile C – 13% pwm) and increase its speed to maximum (pump Grunfos profile C – 88% pwm):

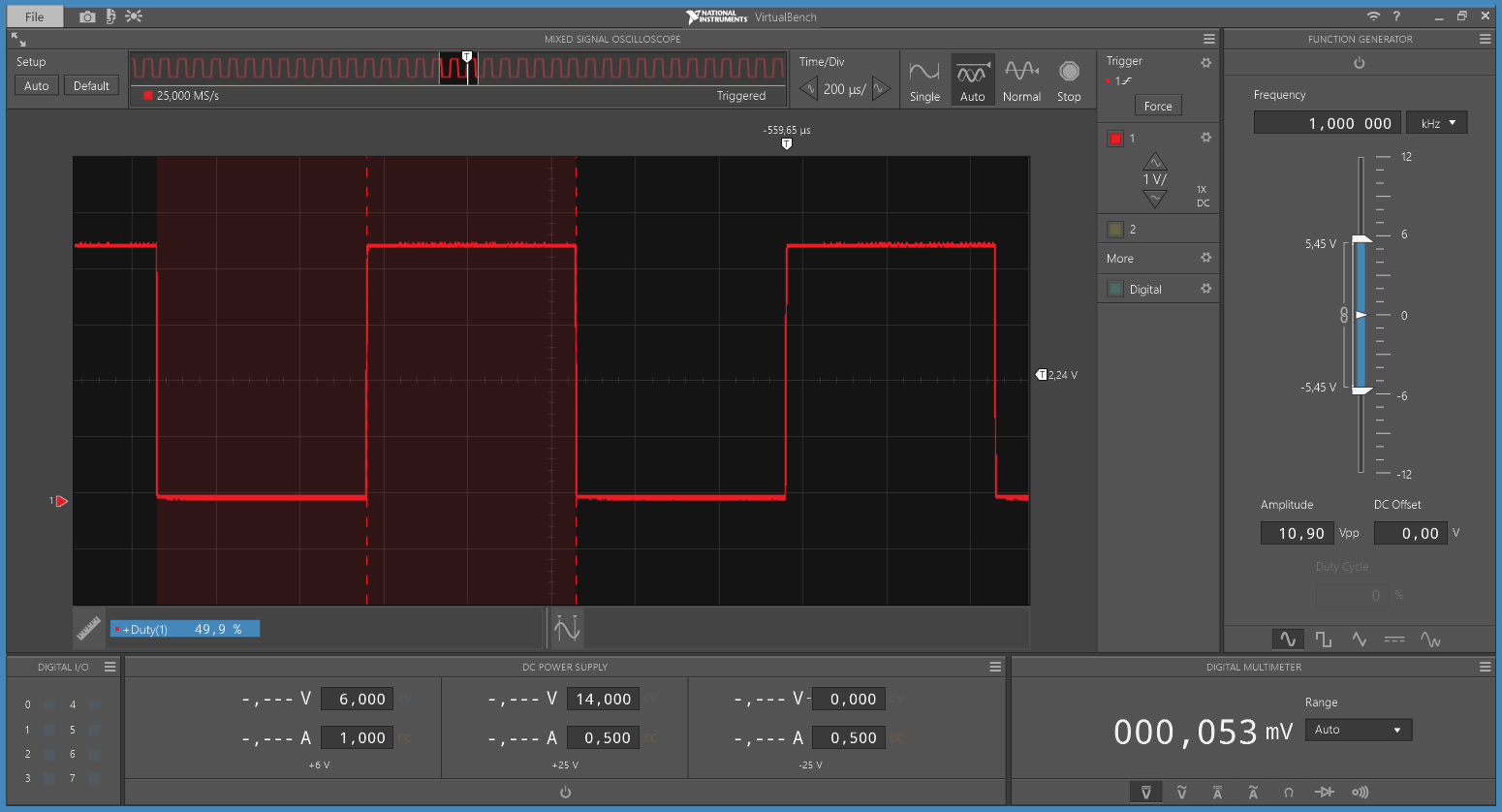


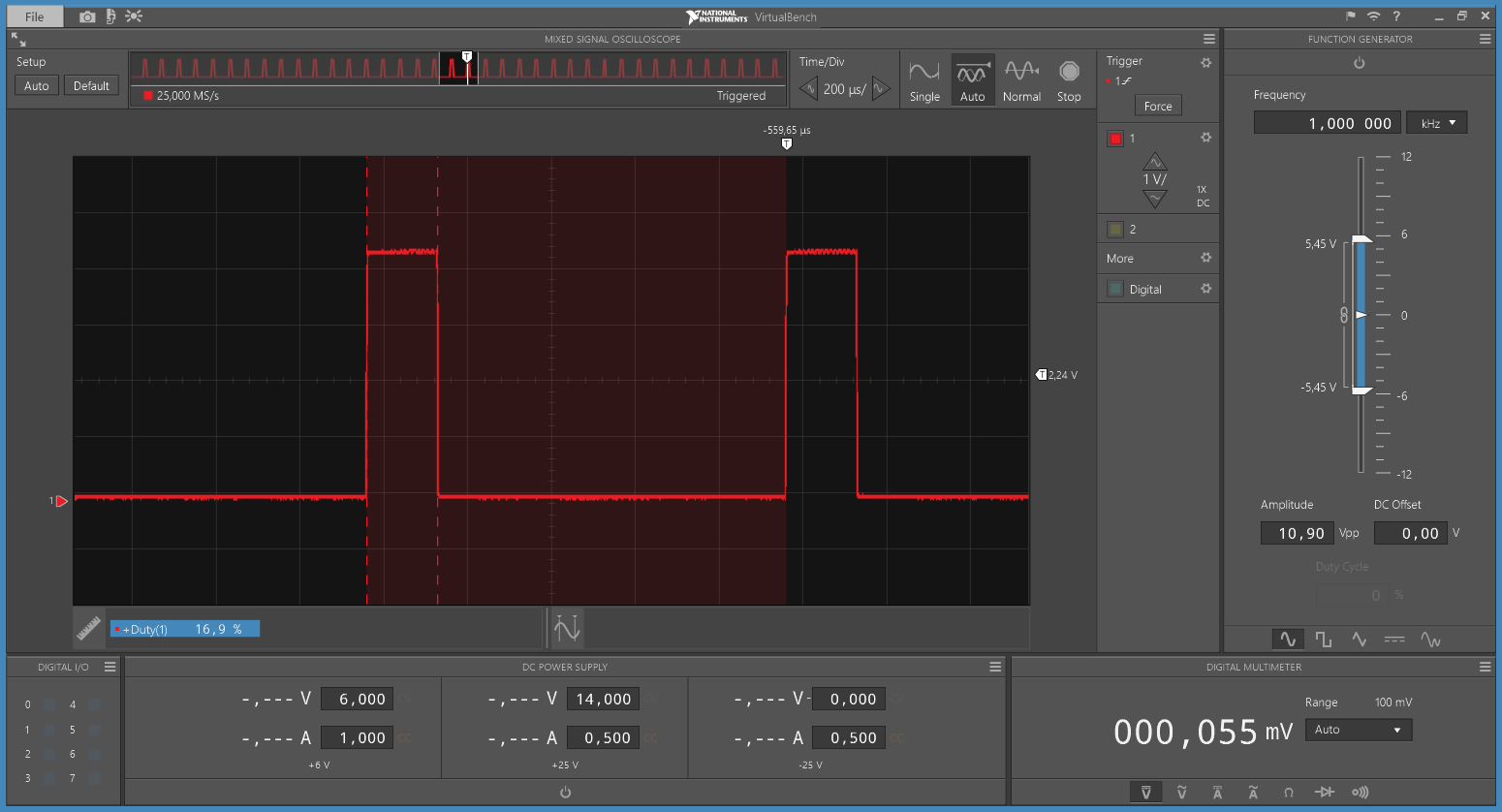




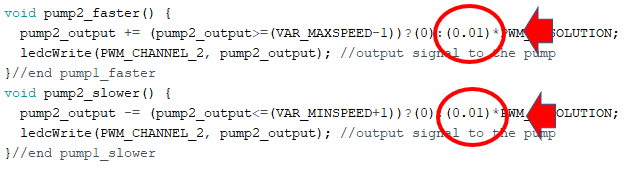
And this is the output signal of pump2 start at maximum speed (pump Grunfos profile C – 95% pwm) and decrease its speed to minimum (pump Grunfos profile C – 17% pwm):



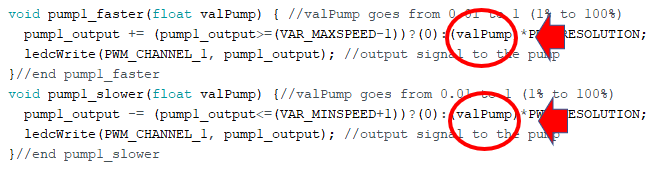




You can increase/decrease the interval of changing by editing these values in pump2:



Or keeping it as a variable like pump1:

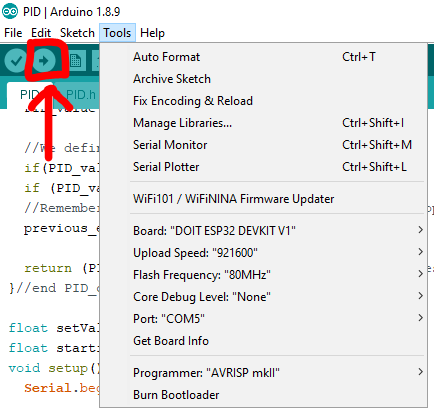
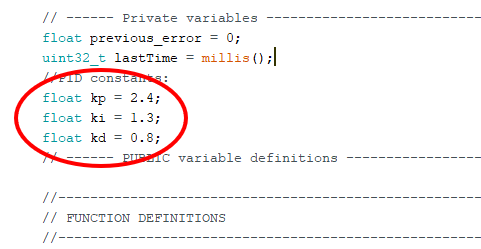


1. **PID Calculation**

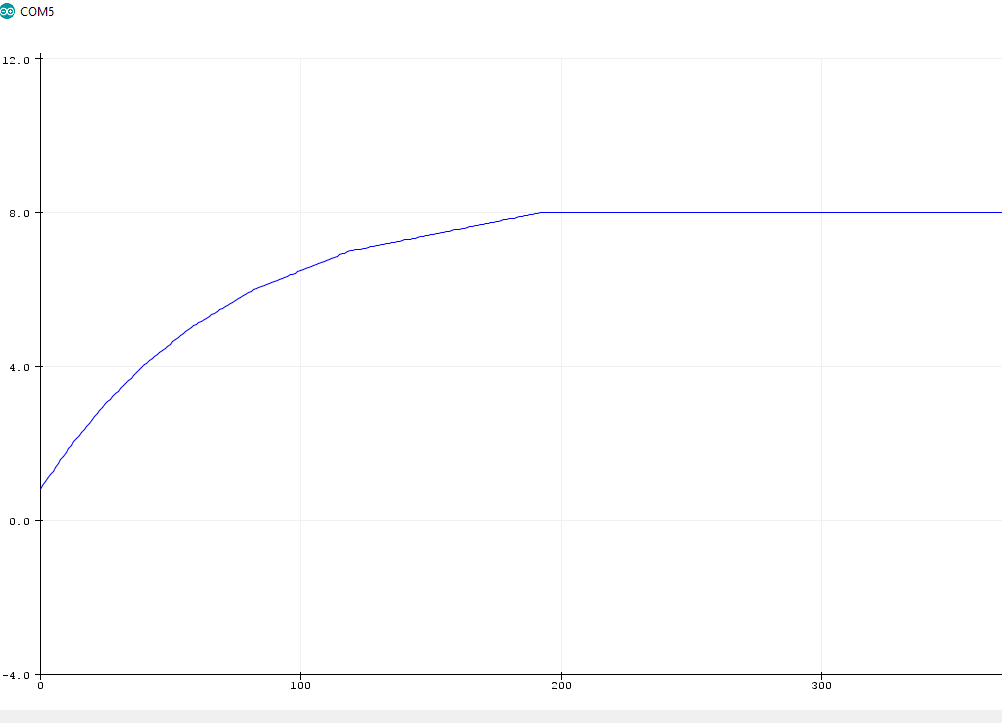
Go into …\Heating Controller\_2019\Embedded Code\Heat\_Controller\separate\_components\ESP32\PID

Open: PID.ino

Change Kp, Ki, Kd correspondingly and upload it to the esp32.



Open the Serial plotter (serial monitor) to see the result

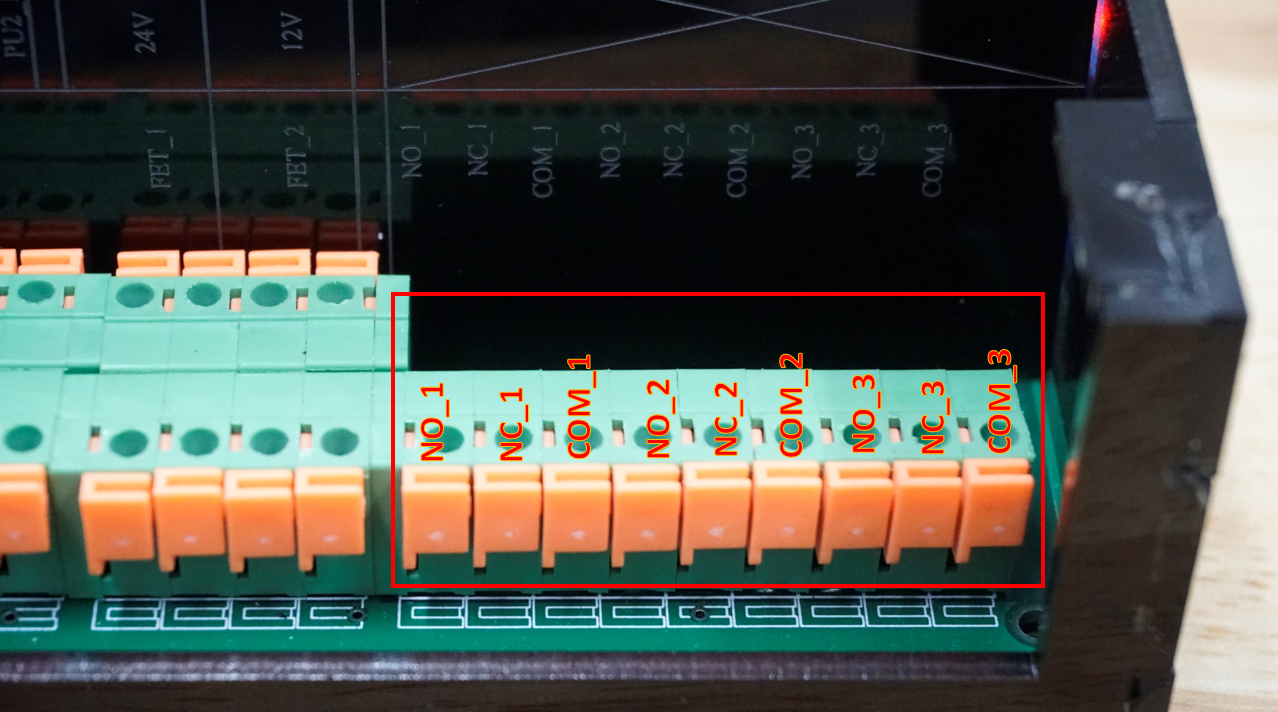


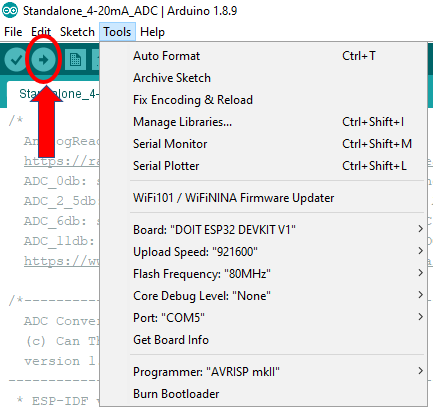
1. **Relay control with HC595**

Go into …\Heating Controller\_2019\Embedded Code\Heat\_Controller\separate\_components\ESP32\ Valve\_Relay\_HC595

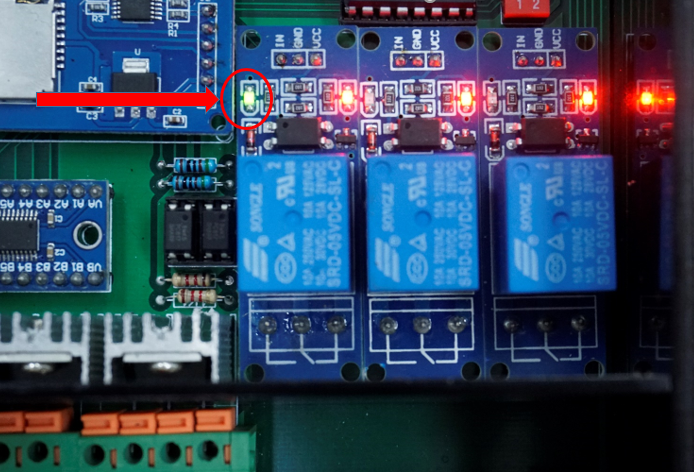
Open: Valve\_Relay\_HC595.ino

You can connect the 220V valve as shown:



Compile and upload it (remember to turn off all bit switches before uploading)

The relay should start turn on and off correspondingly.

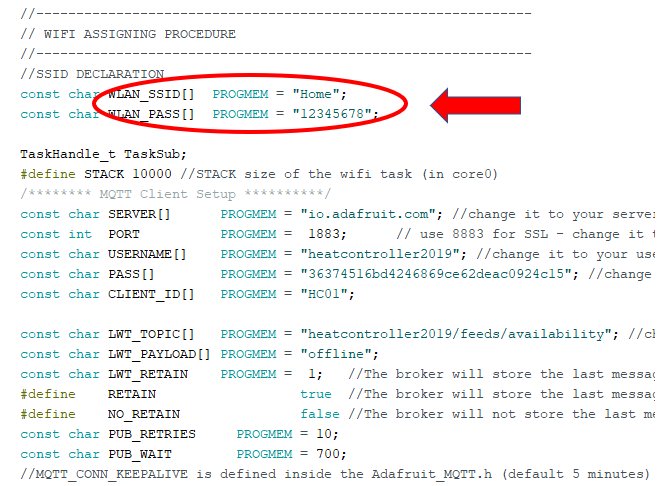


1. **Wifi connect with Adafruit MQTT broker**

Go into …\Heating Controller\_2019\Embedded Code\Heat\_Controller\separate\_components\ESP32\Wifi

Open: Wifi.ino

You can change the wifi SSID and password in the config.h. You can also change the server information here. This system is using MQTT protocol.

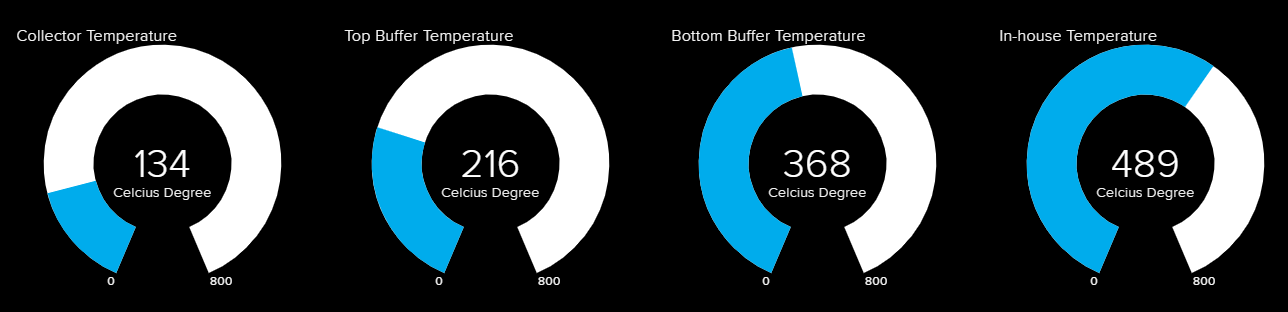


You can send data using the function:



With temp01 is your desired topic of the mqtt broker, 104 is your temperature value (int type), RETAIN option to keep the data even when your system is broken, and the last two are the message to debug when the message is sent or not.

If you connect successfully, data will be sent to the server:



In order to connect to the server, use the file **account.txt** in the main folder to get the login information. You can monitor the temps by using the link:

<https://io.adafruit.com/heatcontroller2019/dashboards/heat-controller-2019>

1. **Main folder – all function integrated**

Go into …\Heating Controller\_2019\Embedded Code\Heat\_Controller\separate\_components\ESP32\ESP32\_main

Open: ESP32\_main.ino

This is the integration of all objects above, but unfortunately, we cannot test all functions working together because we don’t have the devices. But in theory, the system can now work properly.