Smart-Coffee programming guide

Revision 1.5

Table of Contents:

MUST READ!

Safety during initial programming:

The Smart Coffee board can be powered via USB during programming with the coffee machine power switch OFF. This prevents the pump, solenoid, and element relays from turning on while programming.

Legal disclaimer:

Smart Coffee code ("Software") is provided "as is" without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Limitation of Liability:

The creator, contributors, and distributors of this Software shall in no event be held liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this Software, even if advised of the possibility of such damage.

Consumer Responsibility:

By using this Software, you acknowledge that the code is intended for educational and experimental purposes only. The Software is not intended to control espresso machines or any other equipment in a way that meets any safety, regulatory, or quality standards. If you choose to customize or tune the code to operate your espresso machine, you do so at your own risk and are fully responsible for any damage, harm, or inconsistencies that may result, including but not limited to damage to your espresso machine, harm to individuals, or subpar quality of the espresso produced.

Indemnification:

By using this Software, you agree to indemnify, defend, and hold harmless the creator, contributors, and distributors from any and all claims, liabilities, damages, and/or costs (including but not limited to attorney fees) arising from your use or misuse of the Software, your violation of this disclaimer, or your violation of any rights of a third party.

Modifications to Code:

If you modify the Software in any way, you agree to clearly indicate that the code has been modified and is not the original code supplied by the creator, contributors, or distributors.

By downloading or using this Software, you acknowledge that you have read this disclaimer, understand its terms, and agree to be bound by them.

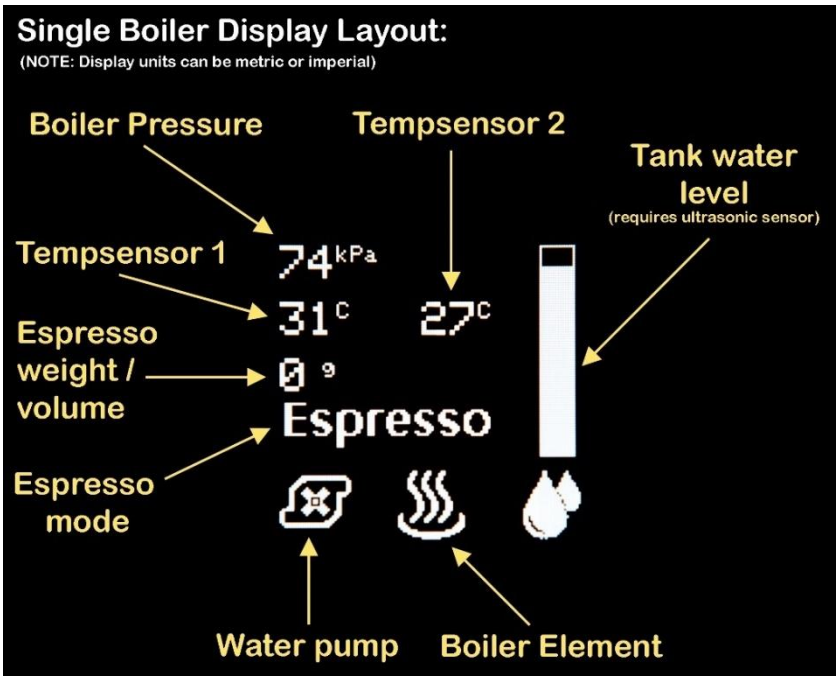
Espresso mode

Espresso mode changes the behaviour of the machine. Espresso mode can be turned on/off by the user pressing a button connected to the “ESW” pin and GND on the PCB. When the display shows an “Espresso” icon, espresso mode is ON. To learn more about the display icons click here

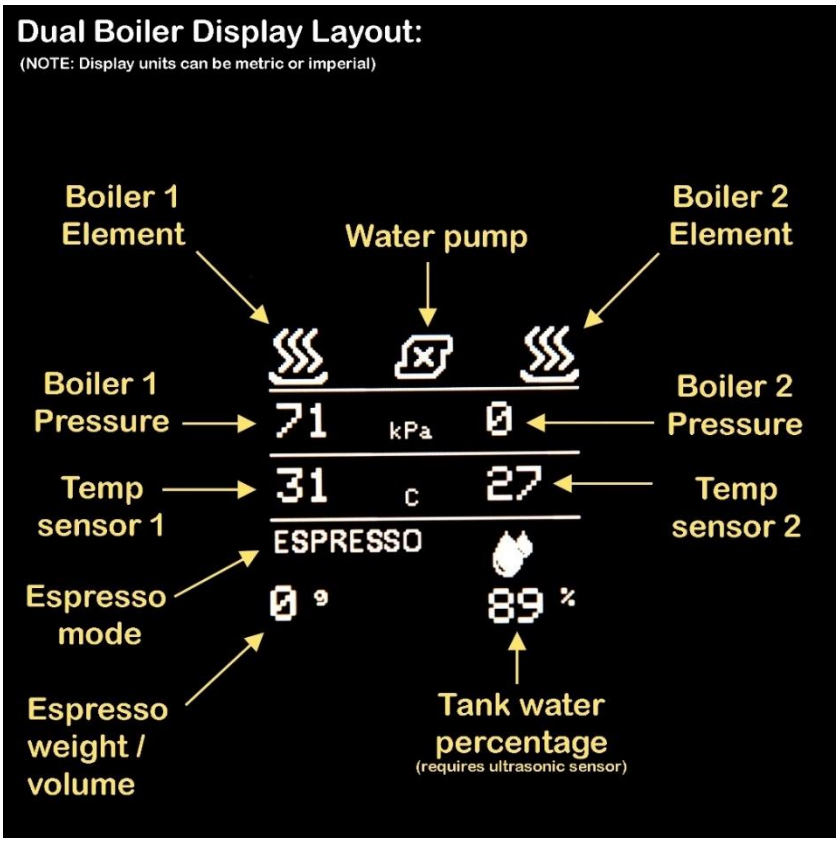
While espresso mode is on and the user presses the brew switch, the machine will perform a pre-infusion cycle (if enabled), then run the pump until the programmed amount of water is dispensed. The target for the dispensed about of water can be a timer, volumetric (using a flow meter), or gravimetric (using a loadcell).

Display layout and icons

There are 2 display modes. They are “Single Boiler” and “Dual Boiler”.
Single boiler mode is best suited for machines with a single boiler.
To enable the single boiler display layout, set DEFAULT_DUAL_BOILER to **false**

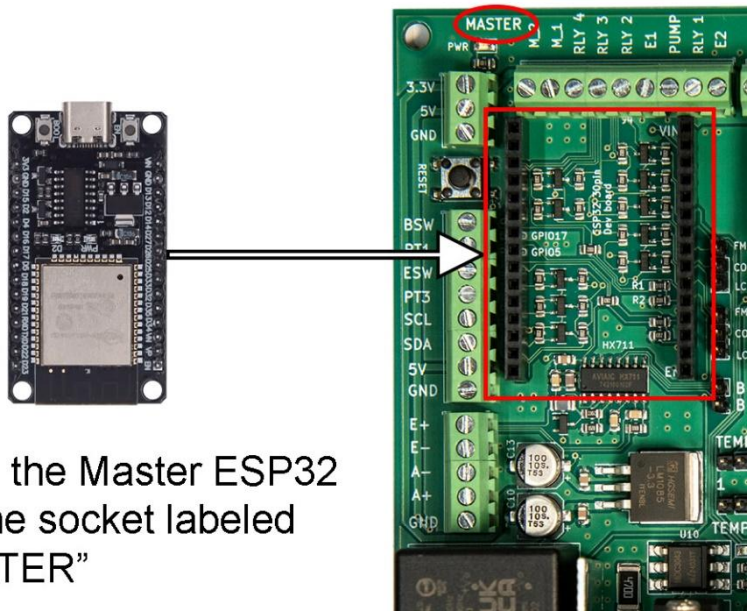


For dual boiler machines set DEFAULT_DUAL_BOILER to **true**

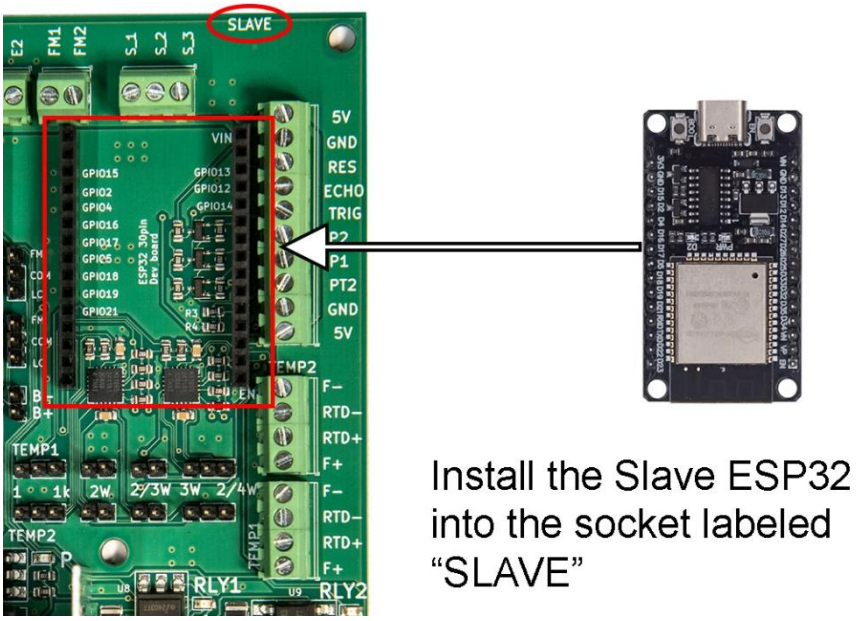


Installing the ESP32s onto the Board

Upload the Smart – Coffee Master code with your programmed settings to an ESP32 and install it into the socket labelled “**MASTER**” as shown below



Upload the Smart – Coffee **Slave** code to an ESP32 and install it into the socket labelled “**SLAVE**” as shown below



Quick Start Guide:

Step 1: Install Arduino IDE

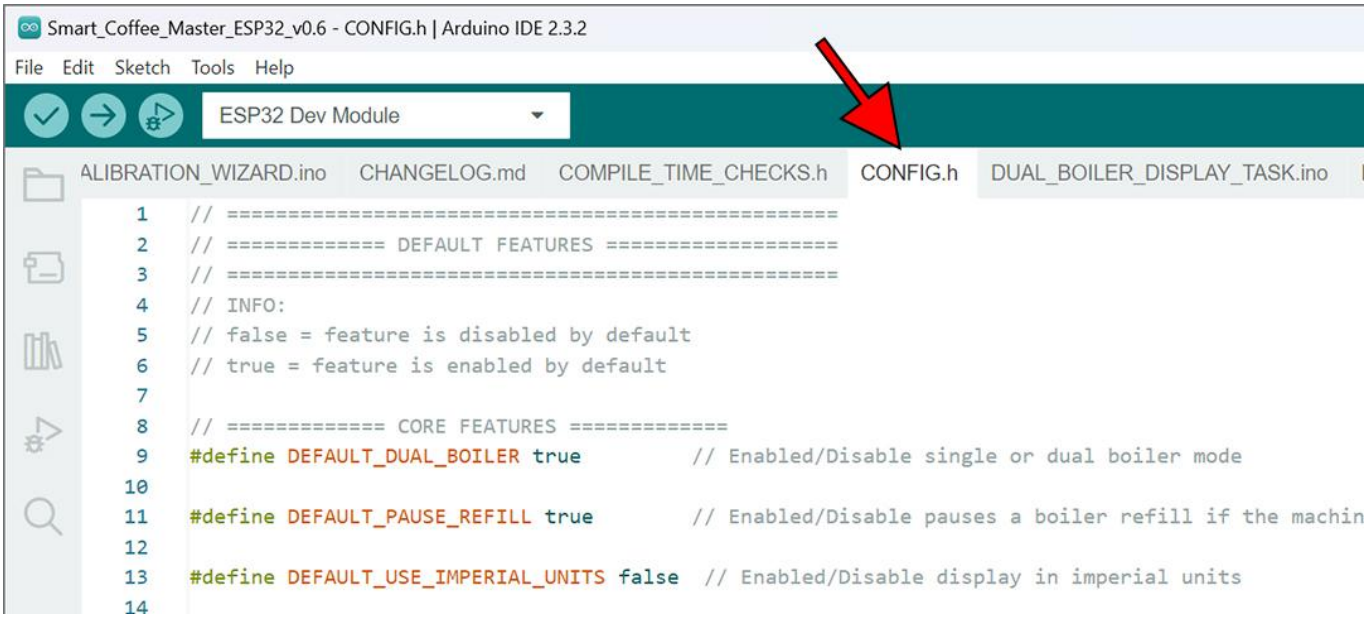
1. Download and install Arduino IDE (Free from https://www.arduino.cc/en/software)

2. Download and install the required libraries. See for more info click here

Step 2: Download Smart-Coffee

Download the latest version of Smart-Coffee and open the “Smart Coffee Master” folder. Open the program by double clicking on Smart_Coffee_Master_ESP32_Vx.x.ino

Then navigate to the “CONFIG.h” tab on the top navigation bar.



Step 3: Enabling / Disabling Features and settings

Beguin programming the core features you want enabled or disabled. If the feature ends with **false**, the feature is **disabled**. And inversely, if the feature ends with **true**, the feature is **enabled**.

```
// ============= CORE FEATURES =============
#define DEFAULT_DUAL_BOILER false

#define DEFAULT_PAUSE_REFILL true

#define DEFAULT_USE_IMPERIAL_UNITS false

#define DEFAULT_STARTUP_LOGO true

#define DEFAULT_SERIAL_PRINTING false
```

Example of enabling a feature by typing “true”

```
// ============= CORE FEATURES =============
#define DEFAULT_DUAL_BOILER false

#define DEFAULT_PAUSE_REFILL false

#define DEFAULT_USE_IMPERIAL_UNITS false

#define DEFAULT_STARTUP_LOGO true

#define DEFAULT_SERIAL_PRINTING false
```

Example of disabling a feature by typing “false”

1. **Single / Dual Boiler:**
   Single boiler machines, set DEFAULT_DUAL_BOILER to **false**
   OR
   Dual boiler machines, set DEFAULT_DUAL_BOILER to **true**

2. **Pause Refill:**
   If you want to pause a boiler refill during brewing an espresso shot,
   set DEFAULT_PAUSE_REFILL to **true**

3. **Metric or Imperial display units:**
   By default, the display shows metric units in Celsius, Grams, and kPa. If you prefer the display to show imperial units (Fahrenheit, Ounces, PSI) set DEFAULT_USE_IMPERIAL_UNITS to **true**

4. **Startup Logo:**
   Startup Logo. By default, the startup logo will be displayed on bootup. If you’d prefer to not see the logo, you can disable it by setting DEFAULT_STARTUP_LOGO to **false**

5. **Serial Printing:**
   Normally serial printing is off since it can be enabled later by USB commands when required.

6. **Pre-Infusion settings:**
   If you would like to use pre-infusion cycle for the start of each espresso shot, set DEFAULT_PRE_INFUSION to **true**
   To learn more about customizing the pre-infusion cycle settings click here

7. **Brew and Espresso Switch Logic:**

   There are 2 switches which serve different functions. The brew switch turns the pump on/off to brew coffee. The second switch toggles “Espresso mode” on/off. To learn more about espresso mode click here

   You can program the switch behavior to toggle on/off with each press, OR only be on while the switch in pressed. To learn more click here

8. **Pressure Transducer 1:**

   By default, pressure transducer 1 is enabled with DEFAULT_TRANSDUCER_1 set to **true**

   Calibrating the pressure transducer min, max and kPa rating can be performed later using the Calibration Wizard.

   Setting Pressure kPa target:
   Set your pressure target in kPa by programing the “DEFAULT_B1_KPA” value. For example “DEFAULT_B1_KPA 50” for a pressure target of 50kPa

   Target Pressure or Temperature:
   By default, the target for boiler 1 is pressure kPa. However, if you would rather the target be temperature then set “DEFAULT_TARGET_TEMP_B1” to **true.**
   If you are targeting temprature, set “DEFAULT_B1_TEMP” value. For example “DEFAULT_B1_TEMP 90” for a 90 celsius target.

   Safety Pressure Limits:
   To learn more about setting min and max pressure limits click here

9. **Pressure Transducer 2:**
   For single boiler machines, skip this step

   If you have enabled “Dual_Boiler” you can configure pressure transducer 2 using the same procedure as **Pressure Transducer 1** in the prevous paragraph.

10. **Temperature Sensor 1:**
    To enable or disable temperature sensor 1. Set DEFAULT_TEMP_SENSOR_1 to **true** or **false.**

    Temprature Safety Limits:
    If you would like to enable minimum and maximum temperature safety limits, set “DEFAULT_TEMP_1_SAFE” to **true**
    Then program “DEFAULT_TEMP_1_MIN” and “DEFAULT_TEMP_1_MAX” with your preferred limits in Celsius units.

11. **Temperature Sensor 2:**
    Same procedure applies to temperature sensor 2 as described in the prevous paragraph.

12. **Espresso Shot Target:**
    There are 3 options avaible to dispense a set amout of water for making espresso (only applies if espresso mode is on).
    1. Espresso Timer
    2. Espresso by weight (Gravimetric which uses the loadcell)
    3. Espresso by volume (Volumetric which uses a flow meter)

    Only 1 option can be enabled at a time, so choose wisely.
    Alternatively, you can leave all 3 options disabled which means the pump will run until the user stops the pump.

    - To enable Timer shots, set DEFAULT_ESPRESSO_TIMER to **true**
      Then set your timer duration DEFAULT_SHOT_TIMER in miliseconds
      (For example, DEFAULT_SHOT_TIMER 5000 for a 5second timer)

    - To enable gravimetric shots, set DEFAULT_ESPRESSO_WEIGHT to **true**
      Then set your shot output DEFAULT_SHOT_GRAM in grams
      (for example, DEFAULT_SHOT_GRAM 40 for a 40 gram shot)
      Lastly, set your weight offset to counteract overshooting your target.
      For example, DEFAULT_WEIGHT_OFFSET 5 to shut the pump off 5 grams early before the target.

    - To enable volumetric shots, set DEFAULT_ESPRESSO_VOLUME to **true**
      Then set your desired shot output DEFAULT_SHOT_ML in mililiters
      (for example, DEFAULT_SHOT_ML 40 for a 40militers shot target)

13. **Flow Meter:**
    To enable flow meter 1, set DEFAULT_FLOW_METER to **true**.

    Then program the pulses per litre value for DEFAULT_FM1_HZ
    (the pulses per litre value should be provided by the manifacture of the flow meter)

    Flow Meter 2:
    To enable flow meter 2, set DEFAULT_FLOW_METER_2 to **true**.

    Then program the pulses per litre value for DEFAULT_FM2_HZ
    (the pulses per litre value should be provided by the manifacture of the flow meter)

    You will likely need to calibrate the pulses per litre value for accurate results. Calibration should be done later when the machine is up and running. To learn more about calibration click here

14. **Load Cell:**
    To enable the load cell, set DEFAULT_LOAD_CELL to **true**.

    You will need to calibrate the load cell scale factor to achieve accurate weight measurements. Calibration should be done later when the machine is up and running later.

15. **Boiler Probe:**
    By default, boiler 1 water level detection probe is enabled.
    By default, boiler 2 water level detection probe is disabled.

    To enable/disable boiler 1 probe, set DEFAULT_B1_WTR_PROBE to **true** or **false**

    To enable/disable boiler 2 probe, set DEFAULT_B2_WTR_PROBE to **true** or **false**

    If you know the touch sensitivity tigger threshold value for boiler 1 probe, you can program DEFAULT_B1_REFILL_TRIG with your value.
    And for boiler 2, DEFAULT_B2_REFILL_TRIG with your value.

    Otherwise, you use the calibration wizard to find the correct value once the machine is up and running later.

16. **Boiler Refill timer:**
    (If you have disabled both boiler probes, skip this step)

    The boiler refill timer will run the pump and any enabled relays for a fixed cycle time to refill the boiler to a desired level.

    For boiler 1:
    Set “DEFAULT_B1_REFILL_TIMER” with your desired refill time in miliseconds.
    For example, “DEFAULT_B1_REFILL_TIMER 2000” for a 2 second refill timer.

    For boiler 2:
    Set “DEFAULT_B2_REFILL_TIMER” with your desired refill time in miliseconds.
    For example, “DEFAULT_B2_REFILL_TIMER 2000” for a 2 second refill timer.

17. **Reservoir refill timer:**
    For most machines, this setting will be left as 0. However, if you want to trigger a refill timer to refill a water reservoir on the machine, click here for more information.

18. **Ultrasonic Sensor:**
    To enable the ultrasonic sensor, set DEFAULT_ULTRASONIC_SENSOR to **true**

    Then program your distance values DEFAULT_RES_FULL and DEFAULT_RES_EMPTY
    to accurately measure the amount of water in the reservoir. For more information, click here

19. **Water Reservoir level detection:**
    If your machine has a water reservoir which requires manual refilling. You can enable 1 of 3 options which will put the machine into a standby state until the reservoir has been refill by the user. This prevents the machine operating when the reservoir is empty.

    Enable upto 1 of the following options:

- Set DEFAULT_RES_SW to **true** for a on/off switch (incl fload switch)
- Set DEFAULT_WTR_RES_PROBE to **true** for a touch sensitive water reservoir probe
- Set DEFAULT_US_LVL_DETECT to **true** for using the ultrasonic distance sensor

    For more information click here

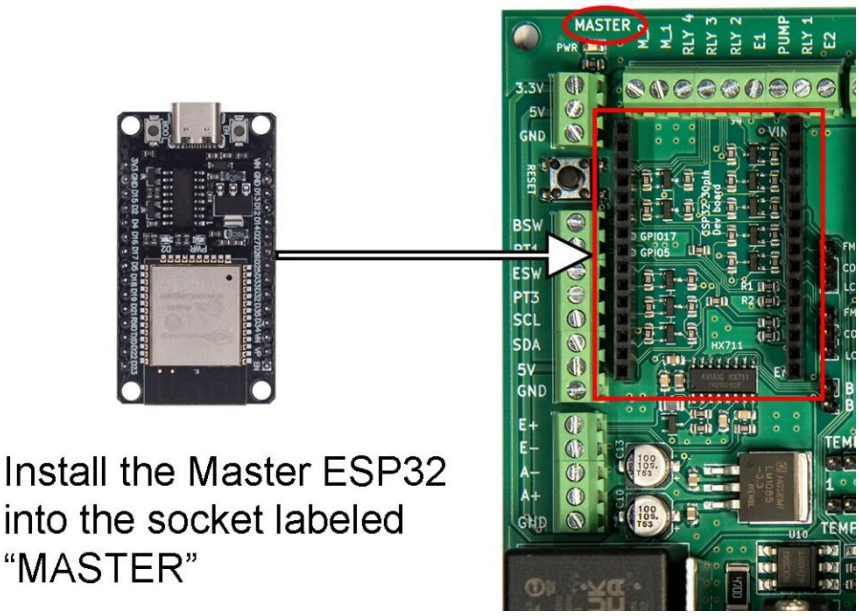
20. **Relay Configuration:**
    The relays can be programmed to turn on at various stages during operation. This allows the user to configure the relay outputs to turn ON or OFF solenoid valves, pumps, elements, or any other hardware connected to the relay.

    By default,:
    When Boiler 1 requires refilling, the pump and Relay 1 will turn on.
    When Boiler 2 requires refilling, the pump and Relay 2 will turn on.
    When Brewing, the pump relay will turn on.
    To learn more about configuring the relays, click here

Step 4: Uploading the code to the Master ESP32

1. At this point, you have completed the bulk of the programming and are ready to upload the master code to the Master ESP32.
2. Compile the code and upload to the Master ESP32.
   For more information on uploading, click here

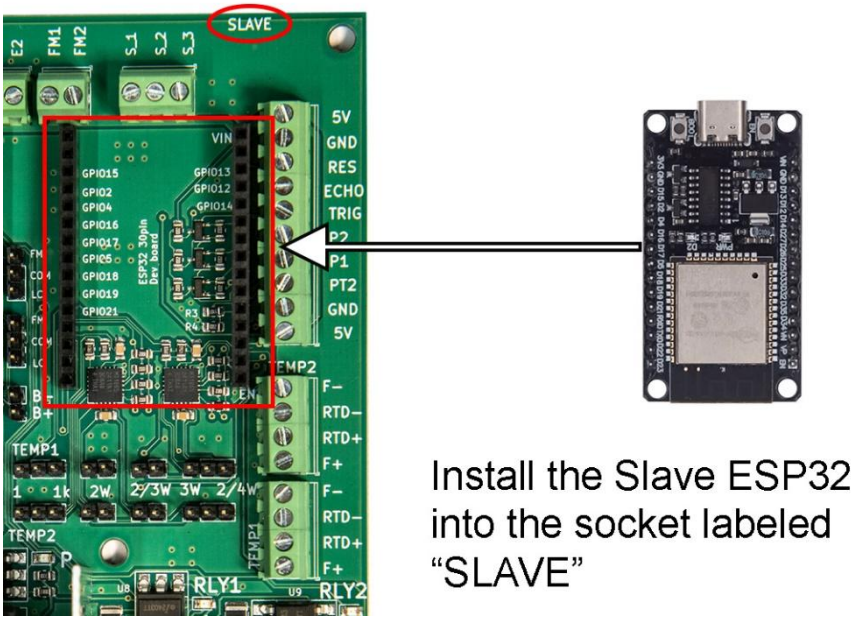
3. Install the ESP32 into the master socket on the board



Step 5: Uploading the code to the Slave ESP32

1. Next open the folder for the Slave ESP32 and open **Smart_Coffee_Slave_ESP32_Vx.x.ino** in Arduino IDE. Then compile and upload the code to the Slave ESP32.
   For more information on uploading, click here

2. Install the ESP32 into the master socket on the board



3. At this point, you should have the Smart Coffee hardware, sensors and ESP32s installed onto the coffee machine and are now ready to beguin calibrating the sensors in the next step.

Step6: Calibrating Boiler Probes

It is critical that the boiler probe/s detect when a refill is required. Otherwise, the boiler/s may run out of water and damage the machine! Or continually refill even if the boiler is full.

1. **Disconnect Power:**
   Leave the coffee machine disconnected from mains power. This prevents the pump, elements and relays from operating while testing/calibrating takes place.

2. **Connect USB Power:**
   Power up the Smart Coffee circuit board by connecting a USB cable from a PC to the Master ESP32. Using Arduino IDE, connect to the ESP32 and open up the serial monitor. To learn more click here

3. **Start the Calibration Wizard for the boiler probes:**
   To start the calibration wizard for **Boiler 1** prob, send the command "**CALIBRATE_B1_PROBE**" via the USB serial connection. Then follow the onscreen prompts.

   To start the calibration wizard for **Boiler 2** prob, send the command "**CALIBRATE_B2_PROBE**" via the USB serial connection.

   Alternatively, if you want to manually calibrate Boiler 1 and Boiler 2 probes, follow the instructions by clicking here

Step 8: Calibrate pressure transducers

It is important to calibrate the pressure transducer/s. Otherwise the Smart Coffee circuit board won't be able to accurately measure the boiler pressure. This can lead to under or over pressurization of the boilers.

During calibration, it is necessary to have a trustworthy pressure gauge connected to the boiler. Which you will reference several times during the Calibration Wizard.

1. **Connect USB Power:**

   Power up the Smart Coffee circuit board by connecting a USB cable from a PC to the Master ESP32. Using Arduino IDE, connect to the ESP32 and open the serial monitor.

2. **Start the Calibration Wizard:**
   To start the calibration wizard for **Pressure Transducer 1**, send the command "**CALIBRATE_PT1**" via the USB serial connection. Then follow the onscreen prompts.

   To start the calibration wizard for **Pressure Transducer 2**, send the command "**CALIBRATE_PT2**" via the USB serial connection. Then follow the onscreen prompts.

3. **Switch on the Power:**
   When prompted by the Calibration Wizard turn on the mains power to the machine. During calibration, the boiler element will need to be turned on to build up pressure.

   Alternatively, if you want to manually calibrate the pressure transducer, follow the instructions here

Step 9: Calibrate other enabled sensors

1. **Reservoir Probe:**

   If you have enabled the Reservoir Probe, use the Calibration Wizard by sending the command "**CALIBRATE_RES_PROBE**" via USB serial connection. Then follow the onscreen prompts.

   To manually calibrate the reservoir probe, follow the instructions here

2. **Load Cell:**

   If you have enabled the Load Cell, use the Calibration Wizard by sending the command "**CALIBRATE_LOAD_CELL**" via USB serial connection. Then follow the onscreen prompts.

3. **Flow Meter:**
   If you have enabled a Flow Meter, follow the instructions to calculate the true pulses per litre value by clicking here

Step 10: Make Coffee!

At this point, your machine should be fully operational. Make yourself a nice cup of coffee, you've earned it!

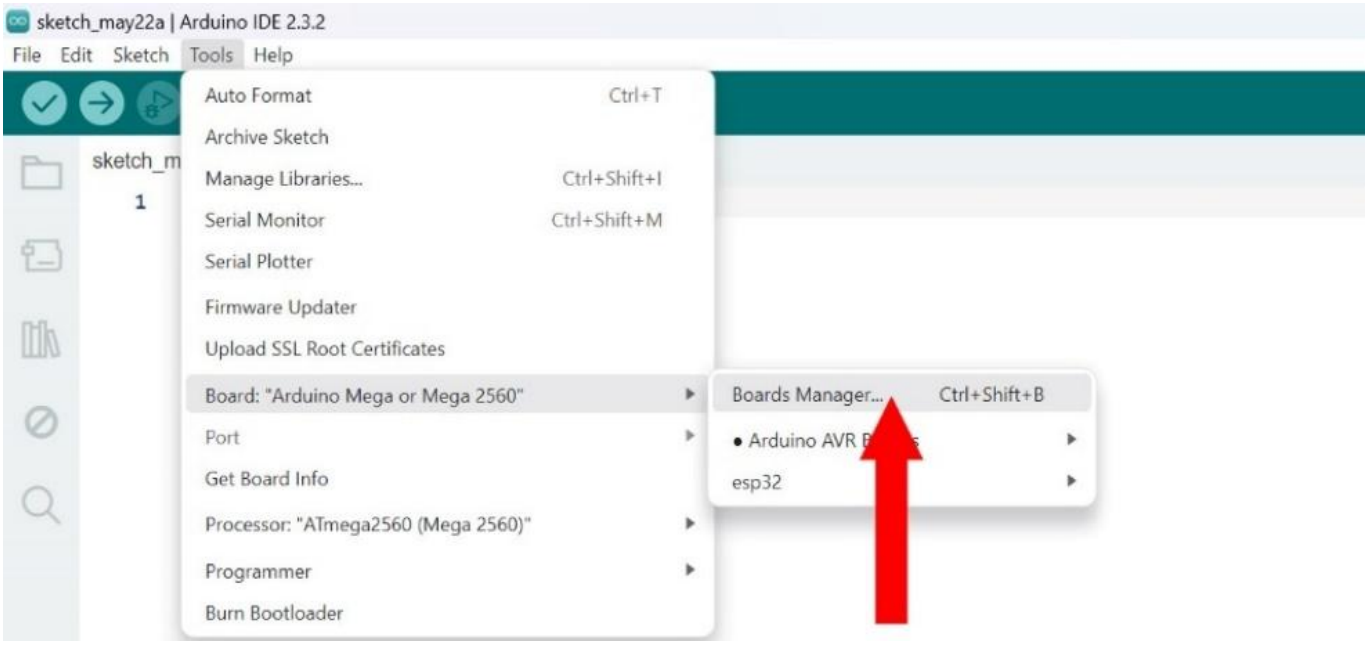
How to connect your PC to an ESP32 module

1. Download and install Arduino IDE (it's free) from https://www.arduino.cc/en/software

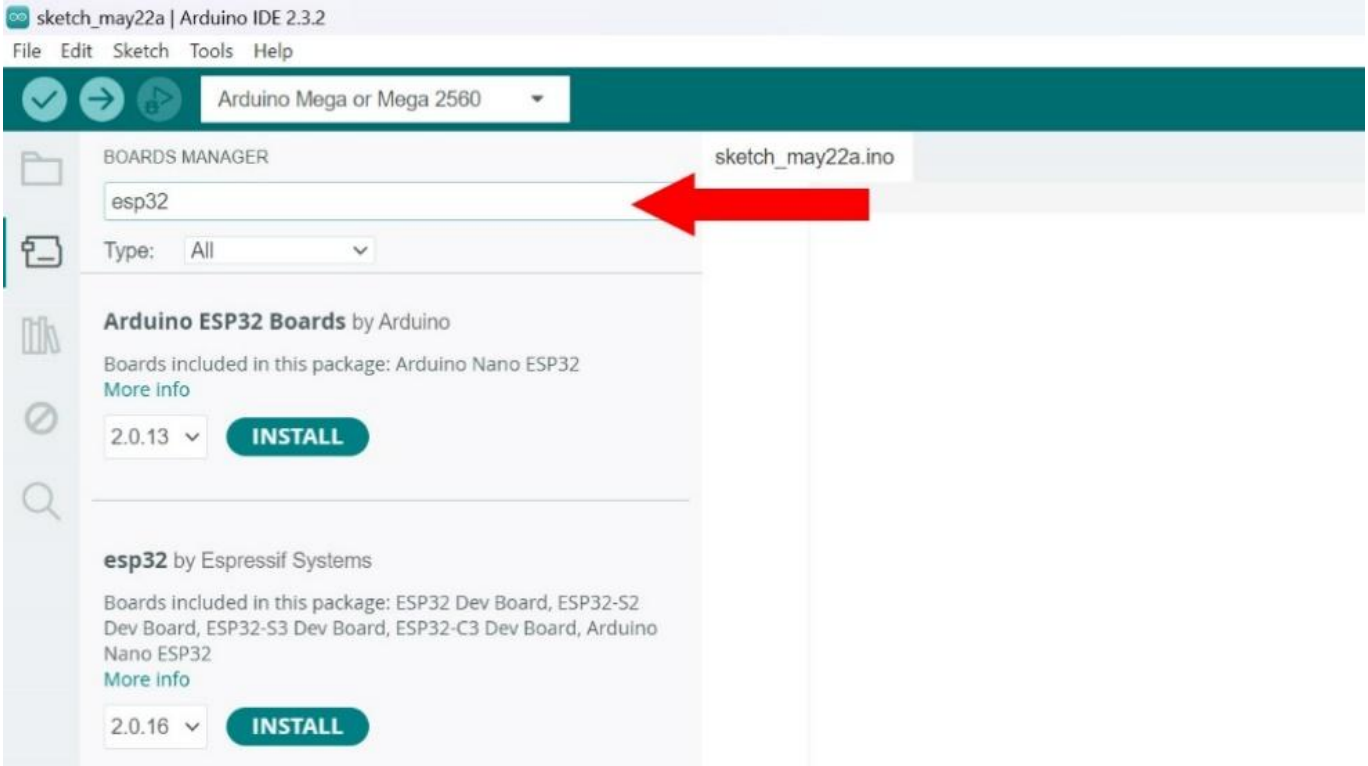
2. During installation, you maybe prompted to install FTDI and/or CP210x drivers. Please allow installation of these drivers.

3. Open Arduino IDE

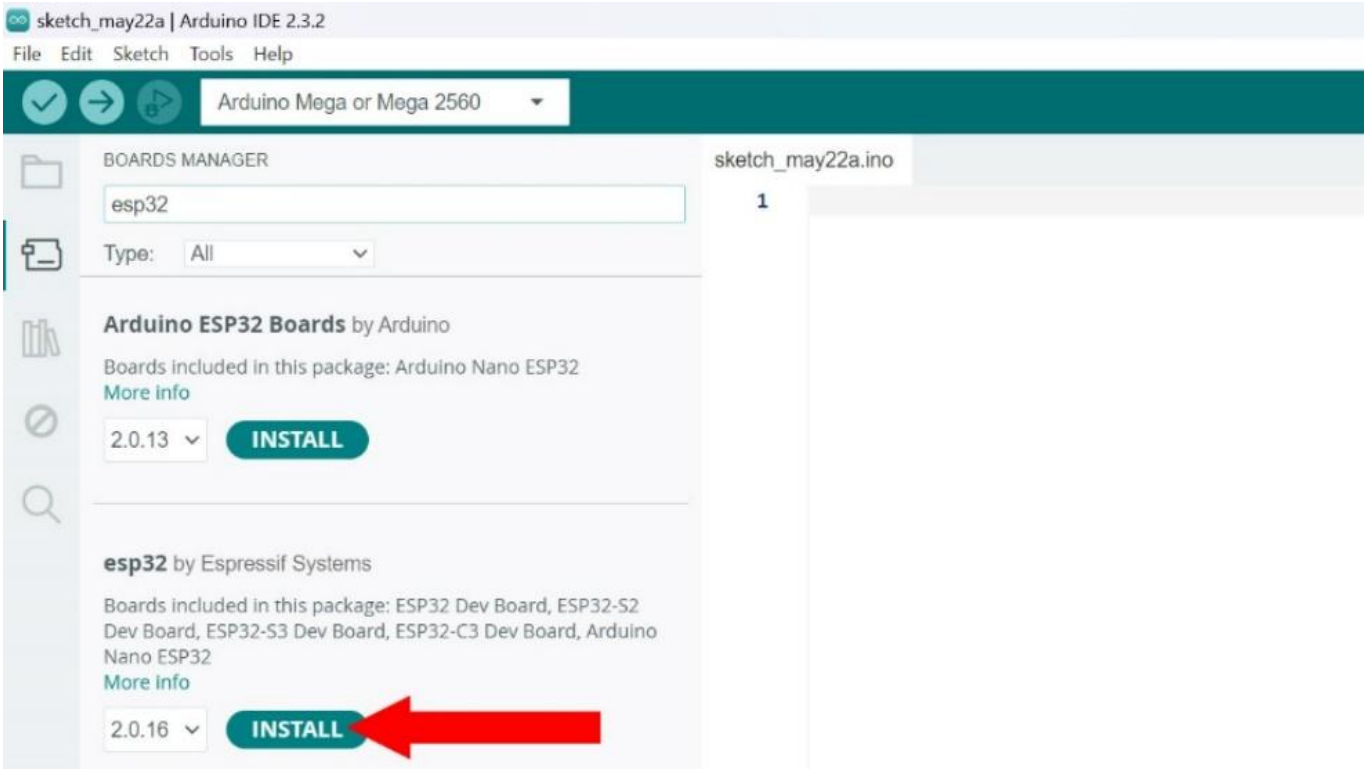
4. Select the Board Manager icon on the left OR select: Tools > Board > Boards Manager from the tool bar.



5. In the search box type "esp32"

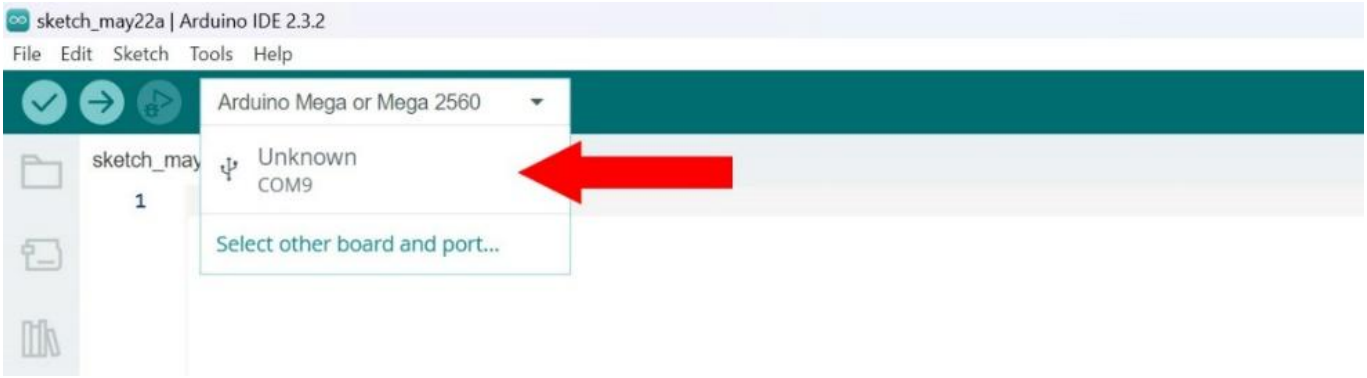


6. Install “esp32 by Espressif Systems”

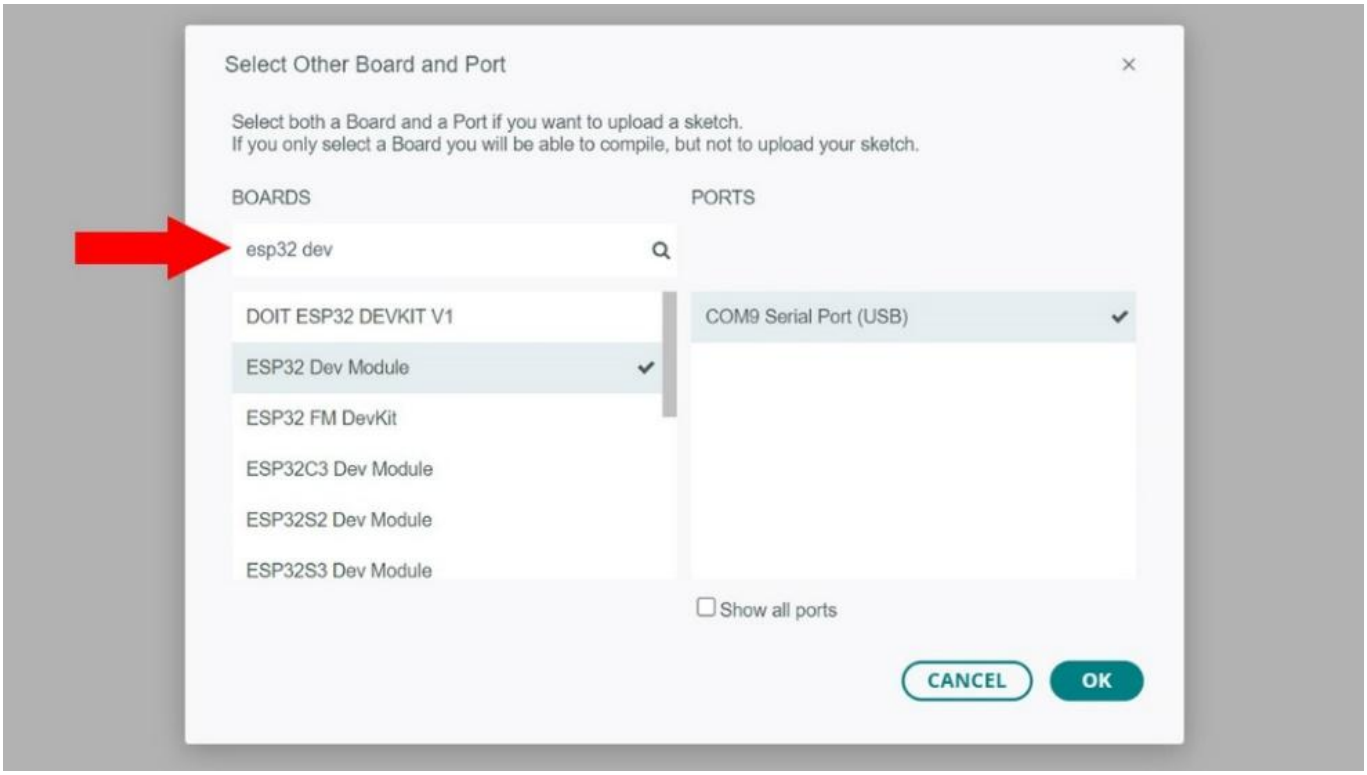


7. Connect the **Master ESP32** to a PC using a USB cable. And click on the board selection box.

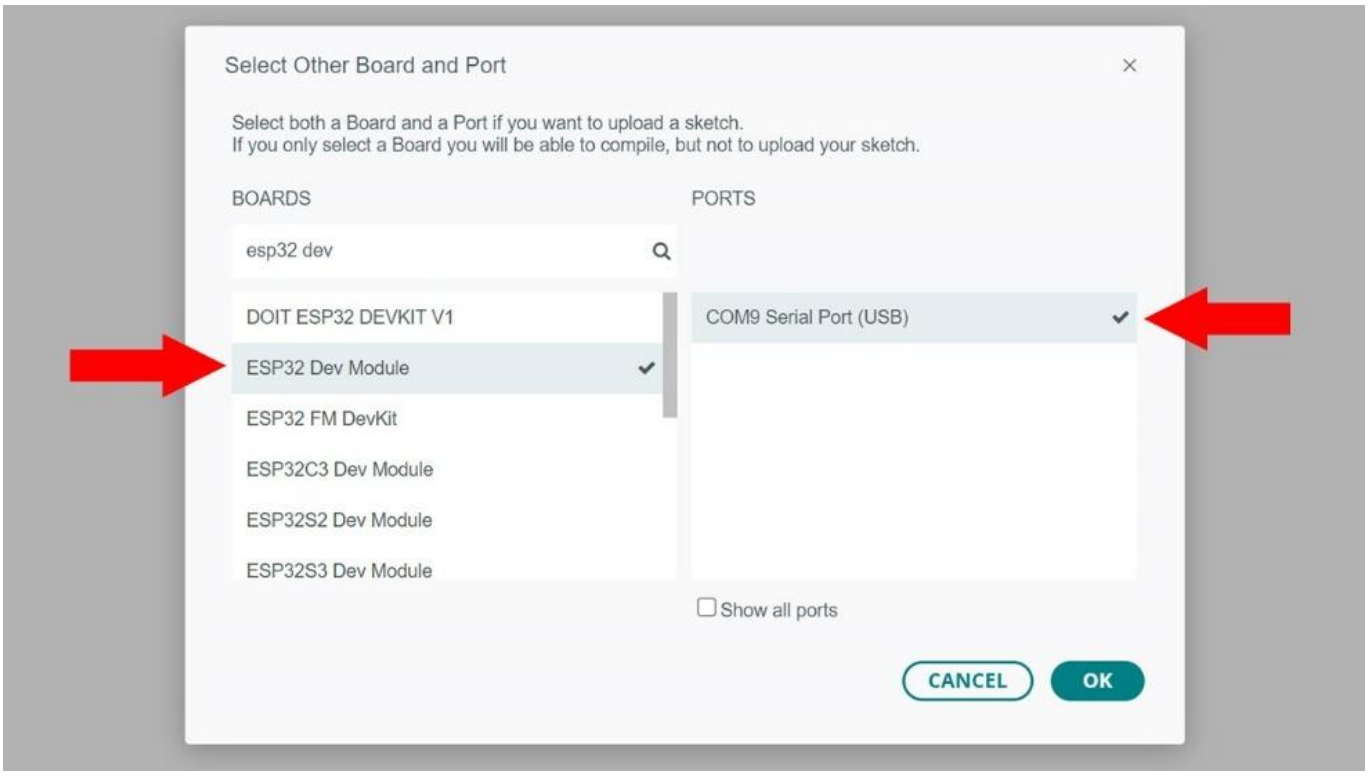
8. You should see a known/unknown USB com device in the menu. Click on it.



9. In the search box, type “esp32 dev”

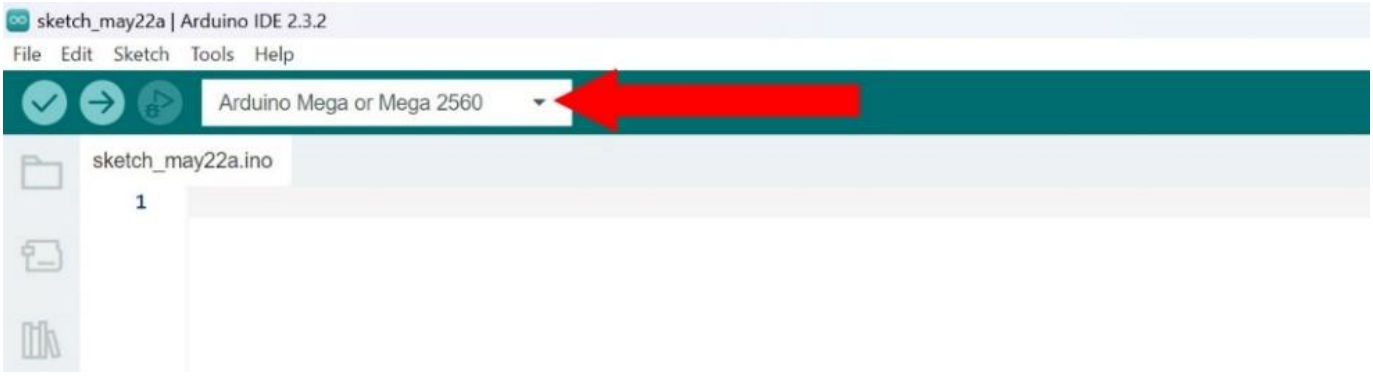


10. Select “ESP32 Dev Module” on the lefthand side. Make sure the COM serial port device on the righthand side is selected and press OK. You are now connected to the ESP32

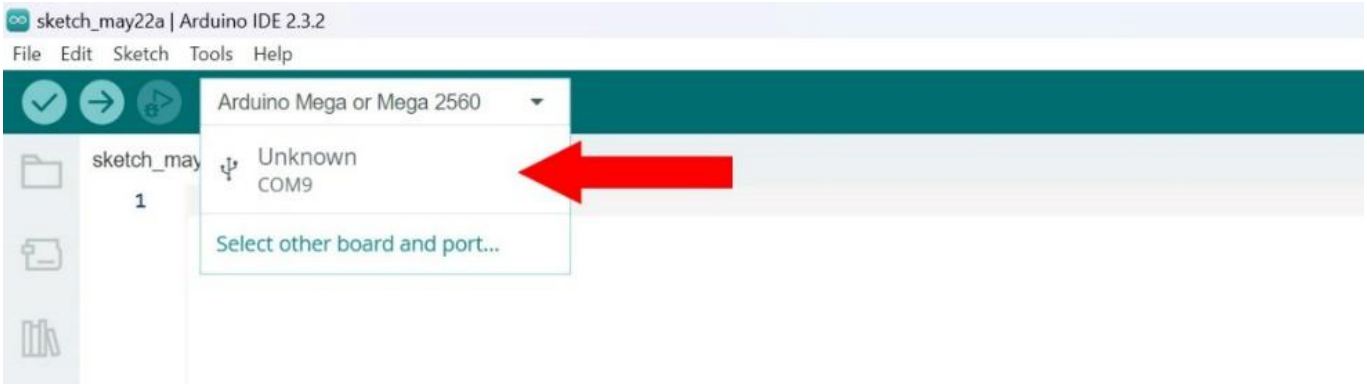


Serial Monitor Connection

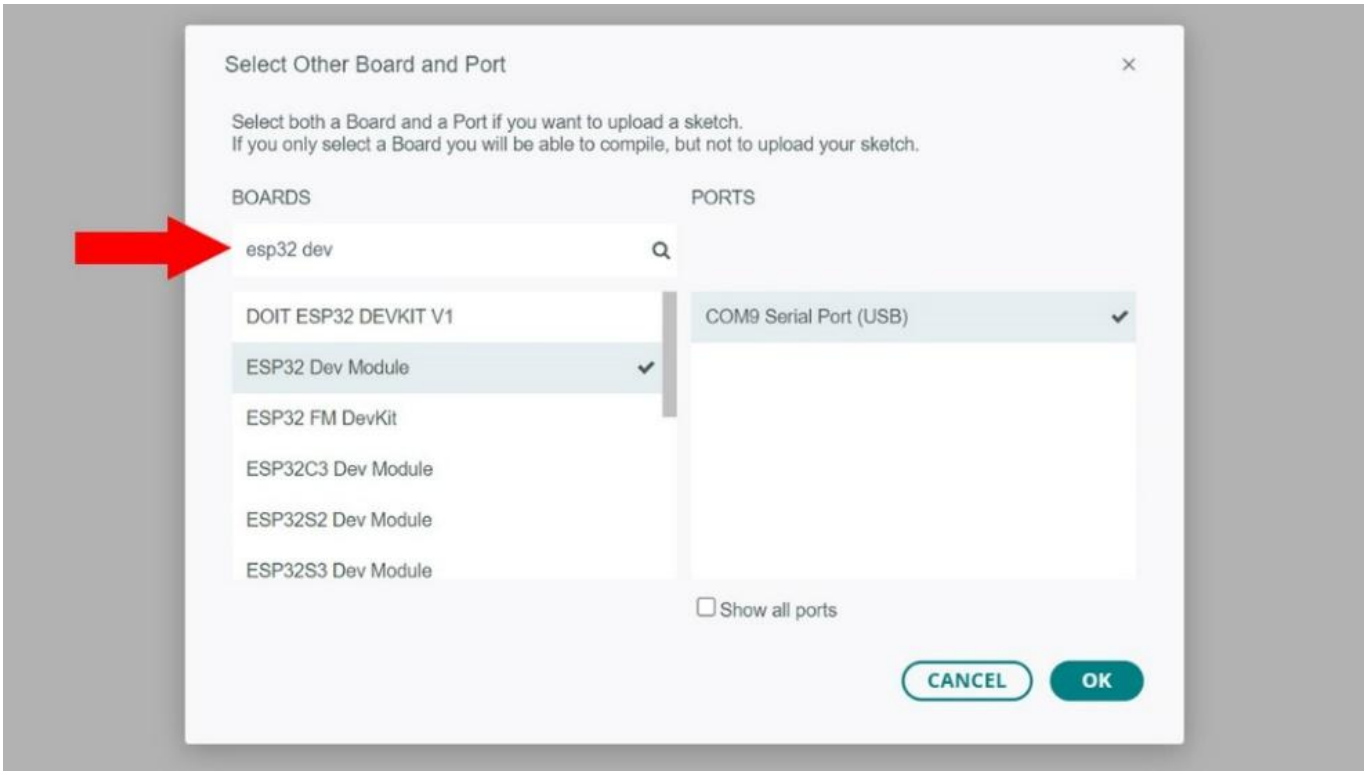
1. Connect the ESP32 via USB to your PC. Then click on the dropdown menu as indicated below



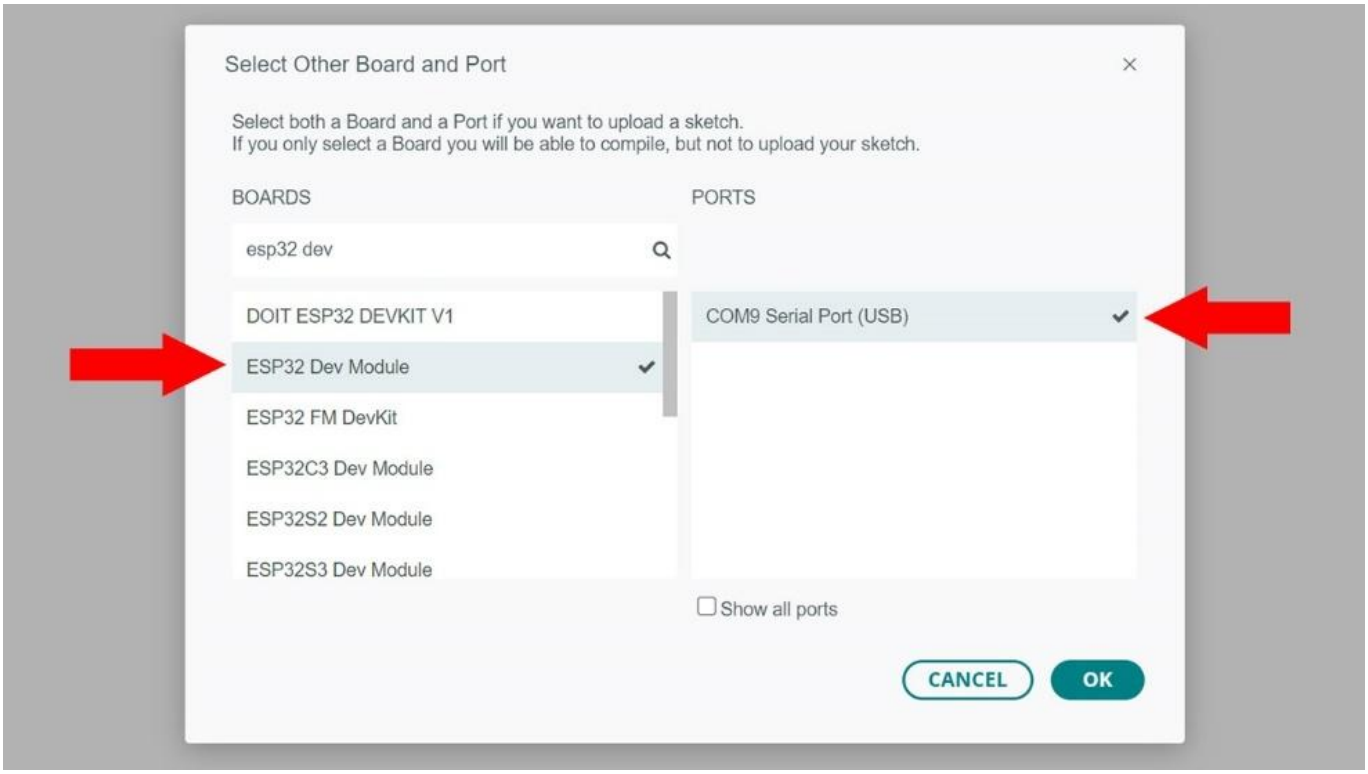
2. You should see a known/unknown USB com device in the menu. Click on it.



3. In the search box, type “esp32 dev”



4. Select “ESP32 Dev Module” on the lefthand side (if you cannot find it, click here to learn more on installing the ESP32 library). Make sure the COM serial port device on the righthand side is selected and press OK. You are now connected to the ESP32



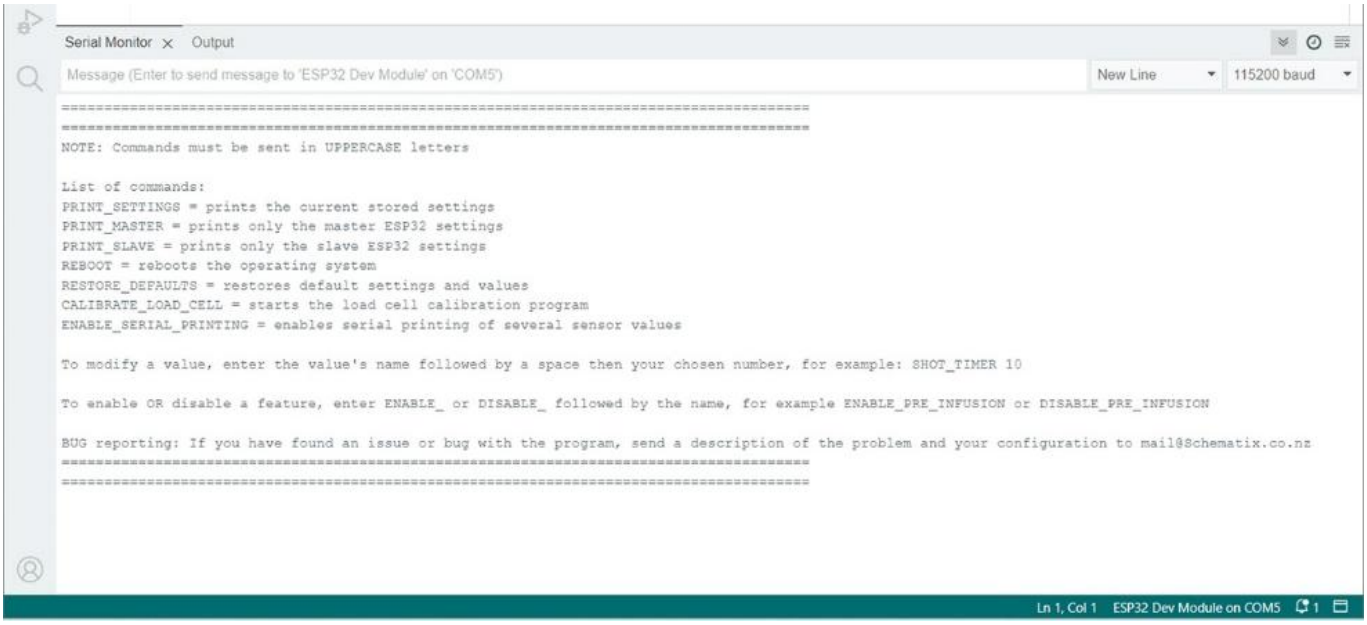
1. Open the serial monitor from the tool bar.

2. In the lower right corner, make sure the baud rate is set to 115200

3. Type your commands in the “message box” and send them by pressing the enter key on your keyboard.

Note: Commands are case sensitive and must be sent in UPPERCASE.

4. Test the serial connection by sending command “INFO” (without quotation marks). You should see text printed in the serial monitor which confirms the ESP32 has been successfully connected.



Programming the code

Programming default values within the source code:

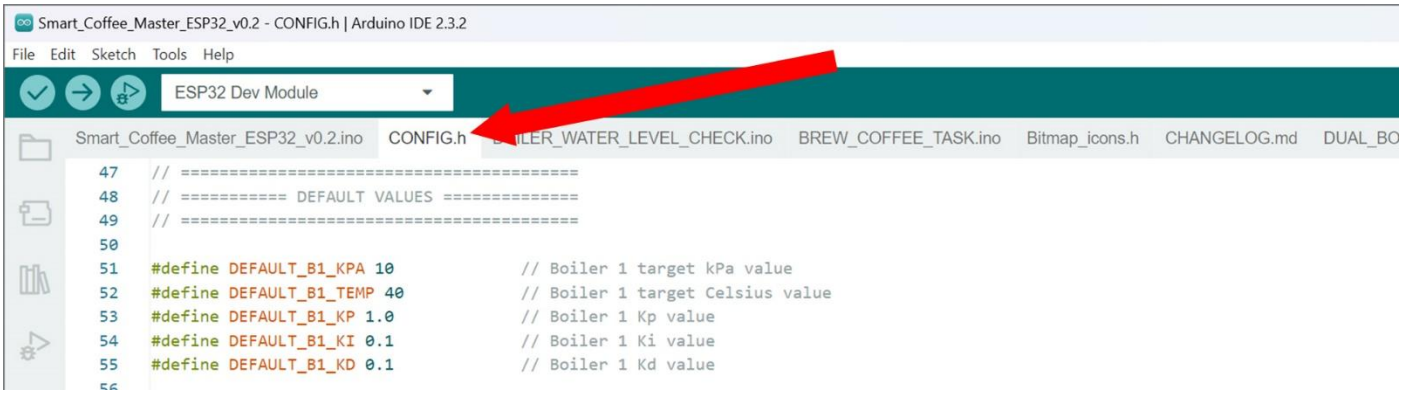
There are 2 methods for programming Smart Coffee software, they are:

- Programming the default values and features in the source code **before** uploading it to the ESP32
- and the other method is changing values and features via serial input commands **after** uploading the code.

For initial programming, ideally, you'll want to program the default values and feature within the source code to match your selection of sensors, and desired features. This will get the bulk of the programming done.

Then the fine tuning can be performed by updating values via serial input commands.

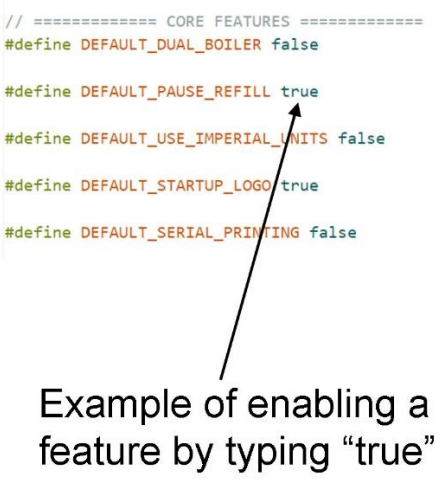
Opening the Smart Coffee Master software in Arduino IDE (free and available from https://www.arduino.cc/en/software) , you'll see several pages at the top navigation bar. The default values and features that are user programmable are contained in the "CONFIG.h" page

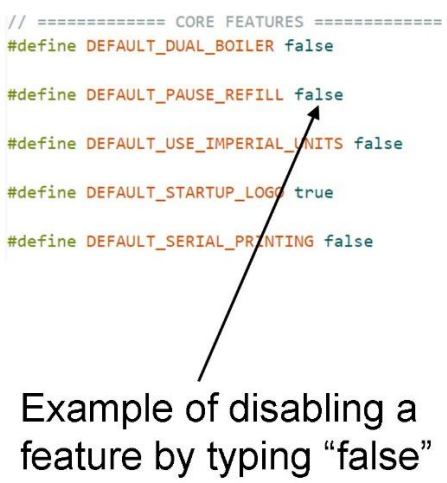


Default Features:

There are 2 categories of default values. They are **values** which program how the machine works, and **features** which turn features on or off. Under "default features" you'll find several options with **false** or **true** at the end of each feature.

If the feature ends with **false**, the feature is **disabled**. And inversely, if the feature ends with **true**, the feature is **enabled**. To **enable** a feature, simply type "true" after the feature you wish to enable. To **disable** a feature, type "false" after the feature you with to disable.





The default features are used by the ESP32 when no settings are found in its memory OR when the user sends the command “RESTORE_DEFAULTS”.

NOTE: This means that changing a default value or feature, re-compiling and uploading to an ESP32 won’t necessarily overwrite the settings stored in the flash memory. To overwrite the settings in the flash memory with the default values in the code. You must send the “RESTORE_DEFAULTS” command via serial input.

Programming via USB:

- With the **Master ESP32** connected to your PC. Send commands via the Arduino IDE serial monitor. Click here to learn more

- Commands are **case-sensitive** and must be sent in UPPERCASE characters.

- To print the current settings on the ESP32, send “PRINT_SETTINGS” (without quotation marks) the list of settings will be printed in the serial monitor.

True / False:

Some settings will have values, while others will have “false” or “true” next to them. “**True**” means the feature is **enabled**. And inversely, “**false**” means the feature is **disabled**.

How to Enable / Disable a feature via USB:

To **enable** a feature, start by typing “ENABLE_” (without quotation marks) followed by the name of the feature. For example, “ENABLE_PRE_INFUSION”.

To **disable** a feature, start by typing “DISABLE_” (without quotation marks) followed by the name of the feature. For example, “DISABLE_PRE_INFUSION”.

How to program a value via USB:

To change the value of a setting, start by typing the name of the setting exactly as displayed, followed by a space, and then your new value.
For example, “B1_REFILL_TIMER 5000”

List of commands:

- **INFO** prints a help section
- **PRINT_SETTINGS** prints the currently stored settings.
- **PRINT_SLAVE** prints only the slave ESP32 settings.
- **REBOOT** reboots the operating system.
- **RESTORE_DEFAULTS** restores default settings and values.
- **CALIBRATE_LOAD_CELL** starts the load cell calibration program.
- **CALIBRATE_B1_PROBE** starts the boiler probe calibration program.
- **CALIBRATE_B2_PROBE** starts the boiler probe calibration program.
- **ENABLE_SERIAL_PRINTING** enables serial printing of several sensor values.

Settings and Features Guide

Single / Dual Boiler:

INFO: This feature changes the display layout and enables/disables Boiler water level probe 2. When DUAL_BOILER is enabled, the screen can print dual pressure, dual temperature, and display the on/off state of each boiler element.

For **single** boiler machines, “DUAL_BOILER” should be disabled by sending the **command** “DISABLE_ DUAL_BOILER”.

For **dual** boiler machines, “DUAL_BOILER” should be enabled by sending the **command** “ENABLE_ DUAL_BOILER”.

Sensor / Boiler groups:

Sensors are grouped together with their respective boiler and settings.

Group 1:

- Boiler water level probe 1
- Pressure transducer 1 (aka PT1)
- Temperature sensor 1 (aka Temp 1)
- Element 1 relay

Group 2:

- Boiler water level probe 2
- Pressure transducer 2 (aka PT2)
- Temperature sensor 2 (aka Temp 2)
- Element 2 relay

Switch Configuration:

Depending on your switch style and configuration, you can program the behaviour of the switch logic. For example, if you are using a momentary push button style switch for the espresso mode switch. You will likely want to **enable** “TOGGLE_ES_SW”. When enabled, espresso mode will cycle ON or OFF with each press of the switch. Inversely, when “TOGGLE_ES_SW” is disabled, espresso mode is only active when the switch is pressed. This allows for use of both latching and momentary switch types to be used.

For most e61 style machines, you can use the factory momentary switch actuated by the group head lever for the Brew Switch. Then add an additional momentary push button switch for Espresso Mode. For this configuration, you’d **disable** “TOGGLE_BREW_SW”, and **enable** “TOGGLE_ES_SW”.

To enable toggle mode for the Brew Switch, send the **command** “ENABLE_ TOGGLE_BREW_SW” (to disable send “DISABLE_ TOGGLE_BREW_SW”)

To enable toggle mode for the Espresso Mode Switch, send the **command** “ENABLE_ TOGGLE_ES_SW” (to disable send “DISABLE_ TOGGLE_ES_SW”)

Relay Configuration:

First let's break down the relay controls into categories to understand their functions. There is a total of 7 relay controls. 2 relays are dedicated for controlling up to 2 boiler elements. The other 5 relays are user programmable that can be configured to turn on **OR** off to operate various hardware on the machine during each mode of operation.

There are 3 categories for the relays. Which are, "**High Current**", "**Low Current**" and "**External**".

High current relays (upto 10A, 100~240VAC):

- E1 (Element 1)
- E2 (Element 2)
- P (Pump)

Low current relays (upto 1A, 100~240VAC):

- RLY1 (Relay 1)
- RLY2 (Relay 2)

External (5VDC, 50ma output):

- E1 (Element 1)
- E2 (Element 2)
- P (Pump)
- RLY1 (Relay 1)
- RLY2 (Relay 2)
- RLY3 (Relay 3)
- RLY3 (Relay 4)

The "**High Currant**" relays are rated up to 10A each (**PLEASE NOTE** the combined maximum current input for the circuit board is 10A! if higher current is required, please use external relays). These relays can supply power to Element 1, Element 2 and the water pump.

The "**Low Current**" relays are rated upto 1A each. These are designed to switch solenoids valves which are frequently used during a boiler refill. By default, RLY1 (Relay 1) will only turn on when refilling boiler 1. And similarly, RLY2 (Relay 2) will only turn on when refilling boiler 2.

The "**External**" relay outputs are used to control external Solid-State-Relays. All 7 relays have external control outputs. The output voltage is 5VDC, and upto 50ma current per output.

IMPORTANT!!! Mechanical relays should not be connected directly to the PCB. The inductive kick-back from the coil can permanently damage the electronics on the PCB. When connecting external relays, please use Solid-State-Relays only.

Programming The Relays:

Programming each relay to turn ON or OFF during each state of the machine **must be done before uploading the code** to the master ESP32 (these settings cannot be changed via serial input).

Open the Smart-Coffee Master code in IDE. Navigate to the "CONFIG.h" tab and scroll down to the bottom, where you will find a section called "RELAY CONFIG".

In this section, you'll find several groups called "BOILER 1 REFILL", "BOILER 2 REFILL", "RESERVOIR REFILL", "DURING BREWING", and "END OF BREWING". Here you can program what relays will be ON during each state of the machine.

BOILER 1 REFILL:
When boiler 1 requires refilling, any relays that are uncommented will be turned ON during the refill. After the end of the refill, the relay will switch OFF (provide no other functions keep it on).

BOILER 2 REFILL:
When boiler 2 requires refilling, any relays that are uncommented will be turned ON during the refill. After the end of the refill, the relay will switch OFF (provide no other functions keep it on).

RESERVOIR REFILL:
When the reservoir requires refilling, any relays that are uncommented will be turned ON during the refill. After the end of the refill, the relay will switch OFF (provide no other functions keep it on). Most machines that are fitted with a reservoir and require the user to manually refill the reservoir will have all the relays OFF.

DURING BREWING:
During the brewing process, any relays that are uncommented will be turned ON during the brew process. After brewing, the relay will switch OFF (provide no other functions keep it on).

Example:

For example, let's assume to refill boiler 2, we require "RELAY3" and "RELAY4" to turn on. To achieve this, uncomment "RELAY3_ON_B2_REFILL" and "RELAY4_ON_B2_REFILL" by removing the leading forward slashes "//" as shown below.

Example with only Pump relay ON during boiler 2 refill:

```
// ========= BOILER 2 REFILL =========
#define PUMP_ON_B2_REFILL
//#define RELAY1_ON_B2_REFILL
//#define RELAY2_ON_B2_REFILL
//#define RELAY3_ON_B2_REFILL
//#define RELAY4_ON_B2_REFILL
```

Example with Pump, Relay 3, and Relay 4 ON during boiler 2 refill:

```
// ========= BOILER 2 REFILL =========
#define PUMP_ON_B2_REFILL
//#define RELAY1_ON_B2_REFILL
//#define RELAY2_ON_B2_REFILL
#define RELAY3_ON_B2_REFILL
#define RELAY4_ON_B2_REFILL
```

Serial Printing:

During setup, it's necessary to print several sensor values in the serial monitor for calibration. To enable serial printing, send the **command** "ENABLE_SERIAL_PRINTING"
(or "DISABLE_ SERIAL_PRINTING" to disable)

Metric / Imperial units:

By default, the units shown on the display are metric (Celsius, kPa, Grams, Millilitres).

To change the display to show imperial units (Fahrenheit, PSI, Ounces, fluid Ounces) send the **command** "ENABLE_IMPERIAL_UNITS" (to disable send "DISABLE_IMPERIAL_UNITS")

NOTE: This feature **ONLY** changes the values shown on the display. It does not change the units that operate the machine. For example, your target boiler pressure is set in kPa, whether "Imperial units" is enabled or disabled.

Boiler water level probe

INFO: There are 2 Boiler water level probes that are touch-sensitive to detect water inside the boiler/s. Each probe can be programmed with a touch sensitive value that determines when a refill is performed. Setting a threshold value is relatively simple, however, it's critical to set a proper value that accurately detects when a refill is required. Otherwise, the boiler may run out of water and damage the boiler/element! Or continually refill even if the boiler is full. Boiler probes can be calibrated manually or using the calibration wizard.

Some boilers do not have a water level detection probe, since they are automatically refilled whenever coffee is brewed. In that case, you can disable one or both boiler probes to suit your hardware.

Enabling / Disabling the boiler probes:
To enable the **boiler 1** water level detection probe, send the **command** "ENABLE_ B1_WTR_PROBE" (or to disable, send "DISABLE_ B1_WTR_PROBE)

To enable the **boiler 2** water level detection probe, send the **command** "ENABLE_ B2_WTR_PROBE" (or to disable, send "DISABLE_ B2_WTR_PROBE)

Programming refill trigger sensitivity:
To set a boiler probe touch sensitive value, send the **command** "B1_REFILL_TRIG" (or "B2_REFILL_TRIG" for boiler probe 2) followed by a space and then your value. For example, "B1_REFILL_TRIG 20".

BOILER PROBE CALIBRATION WIZARD:
Both boiler probes can be calibrated using the calibration wizard when connected to a PC.
To begin calibration wizard, send the command:

"CALIBRATE_B1_PROBE" for **boiler probe 1**.

Or

"CALIBRATE_B2_PROBE" for **boiler probe 2**.

Then follow the on-screen prompts.

Tips:
Before beginning the calibration process, back off the compression nut that retains the boiler probe. This will allow you to lift the probe above the water level inside the boiler during the calibration wizard.

If your boiler probe isn't height adjustable, you may have to find an alternative method for raising and lowering the water level, or the probe itself.

MANUAL BOILER PROBE CALIBRATION PROCEDURE:

Step 1: Before proceeding, for safety the boiler should be close to room temperature and have no pressure inside. This can be done by leaving the main power switch OFF while running the Smart-Coffee PCB from USB power. This prevents the boiler element or pump from turning on, while programming takes place.

Step 2: The boiler must be filled with water to your desired fill level. If the water level is low, or empty. You may have to use your creativity to manually refill the boiler. Alternatively, the Calibration Wizard can activate whatever combination of relays you've enabled to refill boiler 1 and/or 2

Step 3: Enable serial printing by sending the command "ENABLE_SERIAL_PRINTING". Once serial printing is enabled, the ESP32 will print out the boiler probe values.

Step 4: To set the B1_REFILL_TRIG / B2_REFILL_TRIG value, first ensure water is NOT contacting the probe by lifting the probe above the boiler water level (often this can be achieved by backing off the locking nut on the probe which allows the probe to easily be raised or lowered without removing it from the boiler). In this example, let's assume the value is 50 (take note of this value for later).

Step 5: Next, the probe should be lowered back down until it's touching the water inside the boiler. Let's assume the value has dropped to 10.

Step 6: Setting the probe value somewhere in the middle of these 2 values should provide reliable water level detection.
In this example, the B1_REFILL_TRIG value would be set to 30. Any value below 30 should trigger a boiler refill. Send **command** "B1_REFILL_TRIG" or "B2_REFILL_TRIG" followed by a space then your value. For example, "B1_REFILL_TRIG 30".

Verify your settings by testing if raising and lowering the boiler probe correctly detects when a refill is required BEFORE attempting to heat the boiler. Failure to set a value that accurately triggers a refill may cause the boiler to overfill or run dry which can damage the machine.

Boiler refill timer:

Once the boiler probe detects a boiler needs to be refilled, the pump and any enabled relays will be turned ON to refill boiler 1 and/or 2.

Boiler1:
The duration of a boiler refill can be customized by sending the **command** "B1_REFILL_TIMER" followed by a space then your timer value in milliseconds. For example: "B1_REFILL_TIMER 5000" for a 5 second refill timer

Boiler2:
The duration of a boiler refill can be customized by sending the **command** "B2_REFILL_TIMER" followed by a space then your timer value in milliseconds. For example: "B2_REFILL_TIMER 5000" for a 5 second refill timer

Reservoir refill timer:

The reservoir refill timer begins counting down after the reservoir has been refilled. This is only useful for a very limited number of machines that feature a reservoir and are plumbed into a mains water supply. Any relays that have been programmed to turn on when a reservoir refill is required will kept on until the end of the "RES_REFILL_TIMER".

To change the reservoir refill timer, send the **command** "RES_REFILL_TIMER" followed by a space then your timer value in milliseconds. For example: "RES_REFILL_TIMER 5000" for a 5 second refill timer

Pause Boiler refill while brewing:

INFO: If a boiler refill occurs while an espresso shot is being made, often the espresso shot quality will be compromised. This is especially true if you're using flow meters to dispense a set amount of water for an espresso shot when a refill occurs. Because the water used to refill the boiler will be counted towards the espresso shot.

To avoid this problem, you can enable a feature called Pause-Refill. As the name suggests, if a boiler refill is triggered while an espresso is being produced, the refill will be paused until the brew switch is OFF

To enable this feature, send the **command** "ENABLE_PAUSE_REFILL" (or to disable, send "DISABLE_ PAUSE_REFILL)

Reservoir water detection:

INFO: If the machine has a water reservoir, there are 3 options for detecting water:

- Touch sensitive probe
- Ultrasonic Sensor
- Float switch

Note: Only one option can be enabled at a time.

ULTRASONIC SENSOR LEVEL DETECTION:

For more information on using the ultrasonic sensor for low water level detection, click here

RESERVOIR FLOAT SWITCH:

The float switch should be connected to the input labelled "RES" on the PCB and ground. Shorting the "RES" pin to ground activates a "Refill Reservoir" state which turns off all elements and pump on the machine until the reservoir is refilled.

The float switch can be enabled by sending the **command** "ENABLE_ RES_SW" (to **disable** send "DISABLE_RES_SW").

INVERT SWITCH LOGIC:

If the switch logic is inverted (for example, the reservoir is full of water, but the machine thinks the tank is empty) the logic can be inverted by sending "ENABLE_RES_SW_NC" (to **disable** send "DISABLE_RES_SW_NC").

RESERVOIR TOUCH-SENSITIVE PROBE:

The metal probe that is fitted inside the water reservoir should be connected to the input header labelled "RES".

To **enable** the Reservoir probe, send "ENABLE_WTR_RES_PROBE" (to **disable** send "DISABLE_WTR_RES_PROBE")

To accurately detect water inside the reservoir, the RES_PROBE value must be set correctly.

Step 1: Start with an empty water reservoir.

Step 2: Enable serial printing by sending the command “ENABLE_SERIAL_PRINTING”. Once serial printing is enabled, the ESP32 will print out the reservoir probe value. For this example, let's assume the value is 60.

Step 3: Next, the reservoir should be filled with water. Let's assume the value is now 10.

Step 4: Setting the RES_PROBE value somewhere in the middle of these 2 values should provide reliable water detection. In this example, the RES_PROBE value would be set to 35. Any value below 35 should trigger a "Refill water reservoir" state. Send **command** the “RES_PROBE” followed by a space then your value. For example, “RES_PROBE 35”

Reservoir Refill Timer:

Once a “Refill Reservoir” state is triggered by one of the detection methods, you can program the Reservoir refill timer to control the relays on the machine. For example, some machine may require a solenoid value to be opened and remain open for a period to refill a reservoir. The refill timer countdown will begin once the “Refill Reservoir” state is cleared.

To set a probe threshold value, send the **command** “RES_REFILL_TIME” or “RES_REFILL_TIME” followed by a space and then your value in seconds. For example, “RES_REFILL_TIME 10”.

To configure what relays are on when a “Refill Reservoir” state is triggered, see section “**Relay Configuration”.**

Ultrasonic sensor:

INFO: The ultrasonic sensor should be connected to the TRIG and ECHO, 5V, and GND pins on the PCB. The signal output from the ultrasonic sensor should be “PWM”. Models such as A02YY PWM Ultrasonic, and HC-SR04 are compatible.

To **enable** the ultrasonic sensor, send the **command** “ENABLE_ULTRASONIC_SENSOR”
(to **disable** send “DISABLE_ULTRASONIC_SENSOR”).

For the ultrasonic sensor to accurately measure the water level in the reservoir, you must measure (in millimeters) the distance from the face of the sensor to the bottom of the reservoir when it’s empty. Then send the empty distance by sending the **command** “RES_EMPTY” followed by a space and your measurement. For example, if the distance is 310mm, you’d send “RES_EMPTY 310”

Then measure the distance (in millimeters) from the face of the sensor to the water level when the reservoir is full. Then send the full distance by sending the **command** “RES_FULL” followed by a space and your measurement. For example, if the distance is 40mm, you’d send “RES_FULL 40”

Low water level detection:
Besides the ultrasonic sensor being used to measure the water level, it can also be used to activate a “Refill Reservoir” state which turns off all elements and pump on the machine, until the reservoir is refilled.

To enable this feature, send the **command** "ENABLE_US_LVL_DETECT" (to **disable** send "DISABLE_US_LVL_DETECT").
Then set your desired minimum water percentage that triggers a "Refill Reservoir" by sending the **command** "RES_LOW" followed by a space and then your value. For example, to set the minimum water percentage at 10%, send "RES_LOW 10"

Startup Logo:

- To disable the Smart Coffee logo on startup, send the **command** "DISABLE_STARTUP_LOGO".
- Or to enable, send the **command** "ENABLE_STARTUP_LOGO".

Pressure Transducer:

- **INFO:** 2 pressure transducer inputs that can be enabled or disabled depending on the desired configuration.

PRESSURE TRANSDUCER 1:

- To enable Pressure Transducer 1, send the **command** "ENABLE_ TRANSDUCER_1"
- To disable Pressure Transducer 1, send the **command** "DISABLE_ TRANSDUCER_1"

PRESSURE TRANSDUCER 2:

- To enable Pressure Transducer 2, send the **command** "ENABLE_ TRANSDUCER_2"
- To disable Pressure Transducer 2, send the **command** "DISABLE _ TRANSDUCER_2"

The signal from the pressure transducer should output a voltage range between 0.5 ~ 5v (the maximum voltage rating for the signal input is 5v)

Each pressure transducer has 3 programable settings which are:

- Maximum pressure reading of the pressure transducer (Setting name: PT1_KPA and PT2_KPA)
- Voltage output at 0 pressure (Setting name: PT1_MIN_V and PT2_MIN_V)
- Voltage output at maximum pressure (Setting name: PT1_MAX_V and PT2_MAX_V)

ESP32's have a maximum input voltage of 3.3v on the input pins. The signal from the pressure transducer goes through a voltage divider to lower the to a safe level for the ESP32 to read.
When programming and min and max voltage for a pressure transducer signal, divide the voltage by 1.51.
For example,
4.5V / 1.51 = 2.98V
and
0.5V / 1.51 = 0.33V

Example:

For most applications, a pressure transducer that is rated for up to 200kPa and outputs a signal voltage of 0.5~4.5v is desirable. And would be programmed as follows:
PT1_KPA 200
PT1_MIN_V 0.33 (voltage output after voltage divider 0.5V / 1.51 = 0.33V)
PT1_MAX_V 2.98 (voltage output after voltage divider 4.5V / 1.51 = 2.98V)

Calibration:

Set safety limits for pressure:

INFO: Pressure safety limits allow the user to set minimum and maximum kPa pressure limits. If the pressure goes outside these limits, safe-mode will be activated. Which disables the machine until the error is cleared by rebooting the machine.

PRESSURE TRANSDUCER 1:

- To enable pressure safety limits for Transducer 1,
  send the **command** "ENABLE_ TRANSDUCER_1_KPA_SAFE".

- To disable pressure safety limits for Transducer 1,
  send the **command** "DISABLE_ TRANSDUCER_1_KPA_SAFE".

- To set your desired minimum pressure limit (in kPa units) by sending **command** B1_MIN_KPA followed by a space and your desired limit. For example "B1_MIN_KPA -5".
  NOTE: It is recommended to set the Min kPa value in the negative range to avoid false triggers when there is no pressure in the boiler.
  Default value: -5

- Then set your desired maximum pressure limit (in kPa units) by sending **command** B1_MAX_KPA followed by a space and your desired limit. For example "B1_MAX_KPA 100".

PRESSURE TRANSDUCER 2:

- To enable pressure safety limits for Transducer 2,
  send the **command** "ENABLE_ TRANSDUCER_2_KPA_SAFE".

- To disable pressure safety limits for Transducer 2,
  send the **command** "DISABLE_ TRANSDUCER_2_KPA_SAFE".

- To set your desired minimum pressure limit (in kPa units) by sending **command** B2_MIN_KPA followed by a space and your desired limit. For example, "B2_MIN_KPA -5".
  NOTE: It is recommended to set the Min kPa value in the negative range to avoid false triggers when there is no pressure in the boiler.
  Default value: -5

- Then set your desired maximum pressure limit (in kPa units) by sending **command** B2_MAX_KPA followed by a space and your desired limit. For example, "B2_MAX_KPA 100".

Temperature sensor:

- **INFO:** 2 temperature sensors can be enabled or disabled depending on the desired configuration. The temperature sensors are multi-purpose. They can measure any temperature on the machine and shown on the display. Or be used to control one or both boiler elements to target a desired temperature (See "Target temperature or pressure for each boiler" for more info).

TEMP SENSOR 1:

- To enable Temp Sensor 1, send the **command** "ENABLE_TEMP_SENSOR_1".
- To disable Temp Sensor 1, send the **command** "DISABLE_TEMP_SENSOR_1".

TEMP SENSOR 2:

- To enable Temp Sensor 2, send the **command** "ENABLE_TEMP_SENSOR_2".
- To disable Temp Sensor 2, send the **command** "DISABLE_TEMP_SENSOR_2".

Set safety limits for temperature:

TEMP SENSOR 1:

- **INFO:** Temperature safety limits allow the user to define a minimum and maximum temperature threshold. Once the temperature goes outside of these limits, safe-mode will be activated. Which disables the machine until the error is cleared by rebooting the machine.

- To enable temperature safety limits for Temp sensor 1, send the **command** "ENABLE_ TEMP_1_SAFE".
- To disable temperature safety limits for Temp sensor 1, send the **command** "DISABLE_ TEMP_1_SAFE".

- To set your desired minimum and maximum temperature limits (in Celsius units) by sending the **command** TEMP_1_MIN followed by a space and your desired limit. For example "TEMP_1_MIN 10".
- The same format applies for setting the maximum temp limit, For example: "TEMP_1_MAX 90"

TEMP SENSOR 2:

- To enable temperature safety limits for Temp sensor 2, send **command** "ENABLE_ TEMP_2_SAFE".
- To disable temperature safety limits for Temp sensor 2, send **command** "DISABLE_ TEMP_2_SAFE".

- To set your desired minimum and maximum temperature limits (in Celsius units) by sending **command** TEMP_2_MIN followed by a space and your desired limit. For example, "TEMP_2_MIN 10".
- The same format applies for setting the maximum temp limit, For example: "TEMP_2_MAX 90".

Target temperature or pressure for each boiler:

- **INFO:** The Target can be programmed as a temperature or pressure value for each boiler. By default, each boiler target is set as pressure (kPa). 2 sensor groups that correspond to their respective boiler/s.
  Group 1: Temp sensor 1, Pressure Transducer 1, and Element 1 (relay output)
  Group 2: Temp sensor 2, Pressure Transducer 2, and Element 2 (relay output)

BOILER 1:

- To set the target as **temperature** (Celsius), for boiler 1, send the **command** "ENABLE_ TARGET_TEMP_B1".
  OR
- To set the target as **pressure** (kPa), for boiler 1, send the **command** "DISABLE_ TARGET_TEMP_B1".

BOILER 2:

- To set the target as **temperature** (Celsius), for boiler 2 send the **command** "ENABLE_ TARGET_TEMP_B2".
  OR
- To set the target as **pressure** (kPa), for boiler 2, send the **command** "DISABLE_ TARGET_TEMP_B2".

Setting a target pressure or temperature:

BOILER 1:

- To set a target **pressure** (kPa) for Boiler 1, send the **command** "B1_KPA" followed by a space and then your desired kPa pressure value. *For example: "B1_KPA 100"*
  OR
- To set a target temperature target (Celsius) for Boiler 1, send the **command** "B1_TEMP" followed by a space and then your desired kPa pressure value. *For example: "B1_TEMP 100"*

BOILER 2:

- To set a target pressure (kPa) for Boiler 2, send the **command** "B2_KPA" followed by a space and then your desired kPa pressure value. *For example: "B2_KPA 100"*
  OR
- To set a target temperature target (Celsius) for Boiler 2, send the **command** "B2_TEMP" followed by a space and then your desired kPa pressure value. *For example: "B2_TEMP 100"*

PWM or PID element control:

- **INFO:** There are 2 options for controlling each element. They are PID and PWM. By default, PID is enabled. PID offers programmable Kp, Ki, and Kd values. PID is good for heating systems with low inertia and faster response, such as thermal blocks. For high inertia and slow response systems (which describes most water boilers) you will likely find PWM provides a more stable solution

BOILER 1:

- To enable PID for boiler 1, send the **command** "ENABLE_ PID_CONTROL_B1"
- To enable PWM for boiler 1, send the **command** "DISABLE_ PID_CONTROL_B1"

BOILER 2:

- To enable PID for boiler 2, send the **command** "ENABLE_ PID_CONTROL_B2"
- To enable PWM for boiler 2, send the **command** "DISABLE_ PID_CONTROL_B2"

Setting PID values:

BOILER 1:

- To set the Kp value, send the **command** "B1_KP" followed by a space, then your desired value. *For example: "B1_KP 8"*
- To set the Ki value, send the **command** "B1_KI" followed by a space, then your desired value. *For example: "B1_KI 5"*
- To set the Kd value, send the **command** "B1_KD" followed by a space, then your desired value. *For example: "B1_KD 3"*

BOILER 2:

- To set the Kp value, send the **command** "B2_KP" followed by a space, then your desired value. *For example: "B2_KP 8"*
- To set the Ki value, send the **command** "B2_KI" followed by a space, then your desired value. *For example: "B2_KI 5"*
- To set the Kd value, send the **command** "B2_KD" followed by a space, then your desired value. *For example: "B2_KD 3"*

Setting PWM values:

Tips for tuning PWM settings:
While several values can be tuned for PWM, most settings can be left on their default values. B1_MAX_OT and B1_MIN_OT are used for most of the tuning process. Start by tuning these values then move on to tuning B1_STABLE_PWM to stabilize the pressure/temperature at the target value.

Note: These instructions also apply to B2_PWR, B2_MAX_OT, B2_MIN_OT, B2_PWM, B2_STABLE_PWM, and B2_STABLE_TH

B1_MAX_OT and B1_MIN_OT

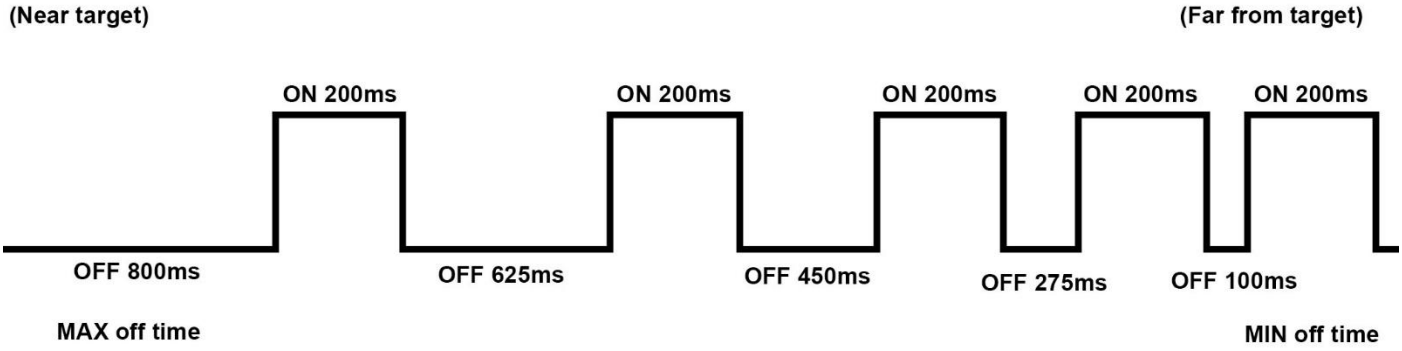
- To change the B1_MAX_OT value, send **command** "B1_MAX_OT" followed by a space then your value. *For example "B1_MAX_OT 800"*

- To change the B1_MIN_OT value, send **command** "B1_MIN_OT" followed by a space then your value. *For example "B1_MIN_OT 100"*

- **B1_MAX_OT** stands for "Boiler 1, maximum off-time". As the pressure/temperature approaches the target, the delay between each pulse of power is increased which effectively reduces the power the element receives.
  The lower this value is, will mean the system is more aggressive in terms of gaining pressure/temperature when **near** the target.
  Default value: 800

- **B1_MIN_OT** stands for "Boiler 1, minimum off-time". When the pressure/temperature is far from the target, this value determines the minimum off-time. The lower this value is, the shorter the delay between pulses of power, which makes the system more aggressive in terms of gaining pressure/temperature when **far** from the target.
  Default value: 100

The MAX and MIN OFF values progressively increase the OFF duration as the pressure/temperature nears the target (as shown below).



- **B1_STABLE_PWM**
- To change the B1_STABLE_PWM value, send **command** "B1_STABLE_PWM" followed by a space then your value. *For example "B1_STABLE_PWM 2000"*
- **Default value: 2000 (2seconds)**

- **INFO:** Most of the settings change characteristics when the pressure/temperature is lower than the target. If we simply turn the element off when the pressure is above the target pressure,

the heating element won't turn on until the pressure/temperature is below the target. This is acceptable for fast response systems because they can recover quickly. However, most water boilers have several seconds of delay between the element tuning on, and the pressure recovering back up to the target. This behavior is described as an oscillation around the target.

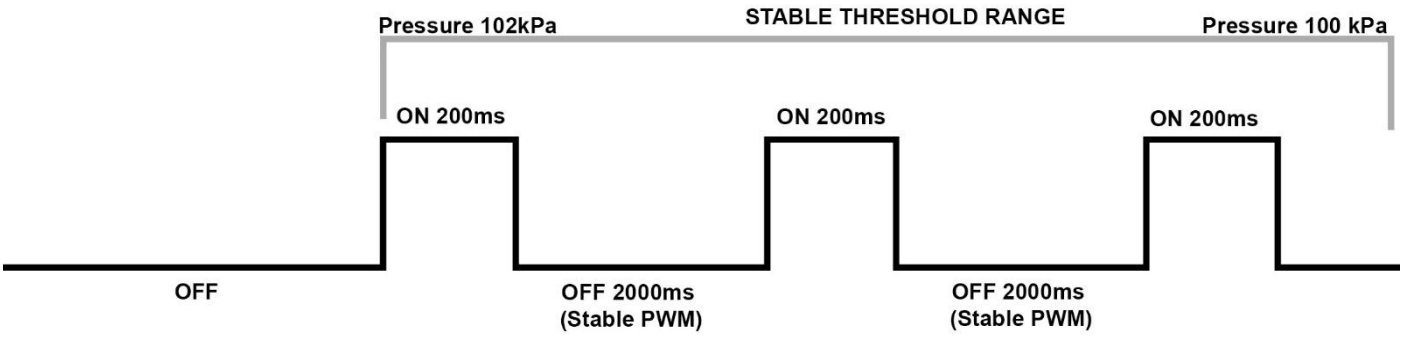
Oscillation can be easy to resolve by adding a slow fixed pulse of power to steady the pressure. This is called "B1_STABLE_PWM".
The B1_STABLE_PWM value sets the OFF period between pulses.

B1_STABLE_PWM will operate when the pressure is equal to or above the target pressure BUT less than the " B1_STABLE_TH" value.

========== ADVANCED PWM TUNING ==========

B1_STABLE_TH (Boiler 1 Stable Threshold) provides a narrow range above the target pressure/temperature so that the STABLE PWM can pulse the power on/off at fixed intervals.
For example, let's assume your target pressure is 100kPa, and your B1_STABLE_TH value is 2. This allows B1_STABLE_PWM to pulse the boiler element ON/OFF when the pressure is between 100 and 102 (as shown below).

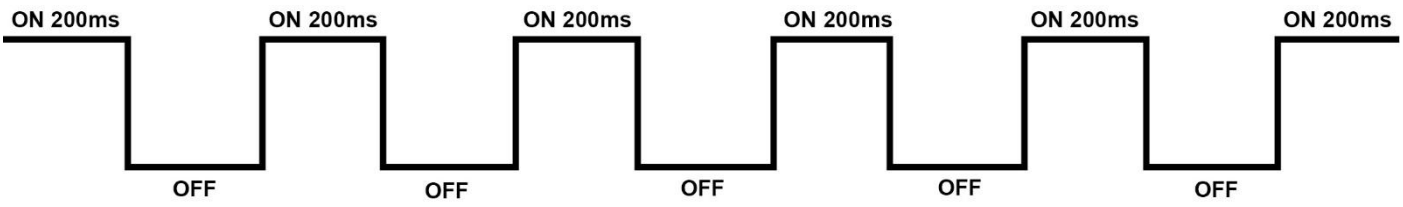
- To change the B1_STABLE_TH value, send **command** "B1_STABLE_TH" followed by a space then your value. *For example "B1_STABLE_TH 2"*
- **Default value: 2**



B1_PWR:

- To change the B1_PWR value, send **command** "B1_PWR" followed by a space then your value. *For example "B1_PWR 200"*
- **Default value: 200**

- **INFO:** B1_PWR value determines the Power ON duration for each pulse in milliseconds. For example, if B1_PWR is set to 200ms (0.2 seconds), then the boiler element will be on for 200ms per pulse.
  TIP: If your system is struggling to gain the last couple of kPa near the target pressure, then increasing this value should help.

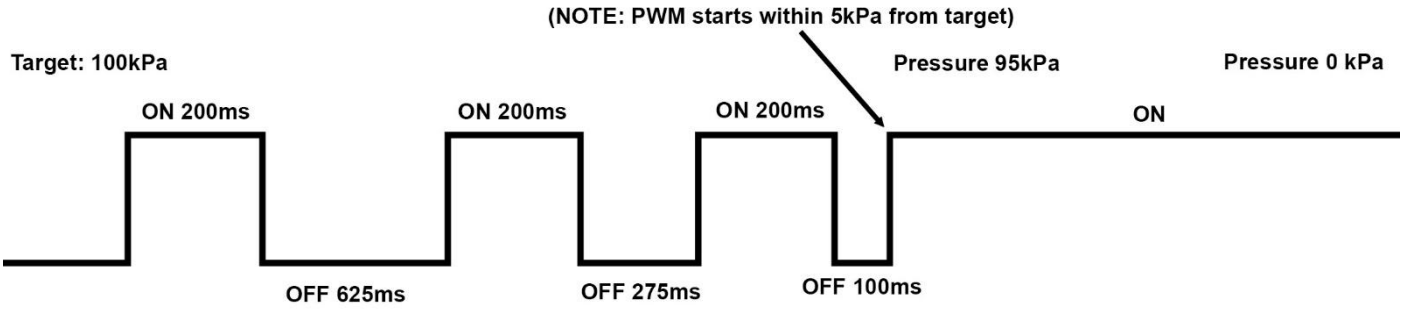


- **B1_PWM**
- To change the B1_PWM value, send **command** "B1_PWM" followed by a space then your value. *For example "B1_PWM 5"*
- **Default value: 5**

INFO: B1_PWM, this value determines when PWM will begin to pulse the power on and off to regulate the pressure/temperature as it nears the target.

TIP: If your system is overshooting above the target, then consider setting a higher PWM THRESHOLD value. This means PWM beguines earlier, which should help reduce overshooting issues.

For example, let us assume the target pressure is 100kPa, and a B1_PWM value has a value of 5. This means that when the pressure is within 5kPa of the target, the boiler element will be controlled by PWM (as shown below).



Espresso Shot Target:

The Espresso Shot Target stops the pump at the desired target to dispense the same volume of water for each espresso shot. To use the espresso shot target, the user must press the "Espresso mode" button to activate Espresso Mode. When espresso mode is activated, the display will show an icon that says "Espresso".

There are 3 options available for the espresso shot target:

- 1. Fixed countdown timer that shuts the pump off after a set amount of time has passed from pressing the brew button (for more info look under heading "Espresso Shot Timer")
- 2. Measured volume of water passed through a flow meter (for more info look under heading "Flow meter/s")
- 3. Gravimetric amount of water weighted by the loadcell (for more info look under heading "Load Cell:")

Espresso Shot Timer:

While espresso mode is ON, you can program an espresso timer that runs the pump for a fixed amount of time (in seconds) before switching the pump off.
To enable the espresso timer, send the **command** "ENABLE_ ESPRESSO_TIMER" (or to disable, send "DISABLE_ ESPRESSO_TIMER").

Then set your timer duration by sending **command** "SHOT_TIMER" followed by a space then your timer value in milliseconds. For example, "SHOT_TIMER 20000" for a 20 second timer

Load Cell:

INFO: The load cell is used for weighing the espresso shot.
To **enable** the load cell, send the **command** "ENABLE_LOAD_CELL" (to **disable** the load cell, send "DISABLE_LOAD_CELL").

With the load cell enabled and calibrated you can program the target espresso shot output in grams. Send the **command** "SHOT_GRAM" followed by a space and then your value. For example, "SHOT_GRAM 40" for a 40-gram target.

By default, the espresso weight target is enabled. However, if you would rather run the pump continually without a weight target while espresso mode is on, you can disable the weight target by sending the **command** "DISABLE_ ESPRESSO_WEIGHT (or to re-enable send "ENABLE_ ESPRESSO_WEIGHT")

CALIBRATION:
To **calibrate** the load cell, send the **command** "CALIBRATE_LOAD_CELL" to start the calibration wizard. Follow the prompts in the serial monitor by:

1. Remove all weights/cups from the load cell
2. Press enter
3. Place a known calibration weight on the load cell. For example, a 200gram weight
4. Enter the weight (in grams) in the serial monitor and press enter.
5. The wizard will calculate the new scale factor and save the value. Calibration is completed.

MANUALLY SET SCALE FACTOR:
To manually update the **scale factor**, send the **command** "SCALE_FACTOR" followed by a space then your number. For example, "SCALE_FACTOR 2389"

WEIGHT OFFSET:
While espresso mode is active, once the espresso weight reaches the shot gram target, the pump will turn off. Some machines may continue to produce espresso for a short period after the pump switches off which will result in more espresso being produced than intended. To combat this issue, you can set a weight offset which switches off the pump earlier.
For example, let's assume a shot gram target of 40, and the machine consistently produces a 46 gram espresso shot. We'd program a weight offset of 6.
To set a weight offset, send the **command** "WEIGHT_OFFSET" followed by a space then your value. For example, "WEIGHT_OFFSET 6".

Flow meter/s

INFO: When a flow meter is enabled. You can dispense a programmed volume of water for making espresso shots. The other option for dispensing espresso shots is by enabling and fitting a load cell. Load cell and flow meter features cannot be both enabled at the same time. So, choose wisely.

To enable the flow meter, send the **command** "ENABLE_FLOW_METER" (or to disable, send the command "DISABLE_FLOW_METER").

By default, the espresso volume target is enabled. However, if you would rather run the pump continually without a volume target while espresso mode is on, you can disable the volume target by sending the **command** "DISABLE_ ESPRESSO_VOLUME" (or to re-enable send "ENABLE_ ESPRESSO_VOLUME")

Note: For dual flow meter configurations, scroll down to "**DUAL FLOW METER SETUP**"

Tested flow meter values for reference:
Model: USN-HS06P = 120000 Hz per litre
Model: USN-HS06PS = 4400 Hz per litre

SINGLE FLOW METER CALIBRATION:
INFO: A single high-pressure flow meter should be installed in a location after all pressure regulator/relief valves to avoid measuring water that is ultimately returned to the reservoir.
Calibrating the flow meter can take a bit of trial and error. First, let's run through the calibration process.

1. Set your pulses per litre for the flow meter by sending the **command** "FM1_HZ" followed by a space and then your value. For example, "FM1_HZ 10"
   (The manufacturer of your flow meter should list the pulses per litre. This value should be used as a starting point to begin calibration)

2. Set your espresso shot ml target at a reasonably high value, such as 100ml by sending the **command** "SHOT_ML" followed by a space then your value. For example, "SHOT_ML 100".

3. Prime the system with water by running the pump until water flows out of the group head

4. Place a receptacle under the group head to capture the water.

5. Make sure "Espresso mode" is on

6. Use the coffee machine and run an espresso shot (no coffee in the basket needed). At the end of the shot and before resetting the brew lever, note the displayed value of water dispensed. For this example, let's assume the displayed water value is 102.8ml

7. Next, measure the dispensed water on a set of scales (in grams) and note down the value. For this example, the weighted water is 89 grams.

8. To calculate the new corrected pulses per ml value. Using a calculator, first enter the displayed value, then, divide that number by the actual amount of water dispensed in grams, then, multiply by your current pulses per litre value. In our example, the equation would look like this:

   102.8 (displayed output of water) / 89 (the actual amount of water dispensed) = 1.155
   1.155 x 6 (current FM1_HZ value) = 6.93 (Round this number to the nearest decimal place, in this example 6.9 is our new pulses per ml value for the flow meter)

9. Update your pulses per litre for the flow meter by sending the **command** “FM1_HZ” followed by a space and then your value. For example, “FM1_HZ 6.9”

10. It's a good idea to run a shot and then weigh the water to verify the new value is correct.

DUAL FLOW METER SETUP:
INFO: Two flow meters should be installed on the machine. One on the water inlet side of the pump (low pressure) and the other flow meter should be installed on the return line to the reservoir (also low pressure).

Why does the program have 2 flow meters to calculate water flow?
In most coffee machines there is an OPV (Over Pressure Valve) on the high-pressure side of the pump. This valve bleeds off water and returns it to the water reservoir.
If only a single flow meter was fitted before the pump, the amount of water dispensed out of the group head would be unknown because some water would have been returned to the reservoir.

Ultimately, the first flow meter counts how much water enters the system. While flow meter 2 subtracts any water returned to the reservoir.

To enable flow meter 2, send the **command** “ENABLE_FLOW_METER_2” (or to disable, send command “DISABLE_FLOW_METER_2”).

DUAL FLOW METER CALIBRATION:
1. Set your pulses per litre for the flow meter by sending the **command** “FM1_HZ” followed by a space and then your value. For example, “FM1_HZ 10”
   (The manufacturer of your flow meter should list the pulses per litre. This value should be used as a starting point to begin calibration)

2. Set your espresso shot ml target at a reasonably high value, such as 100ml by sending the **command** “SHOT_ML” followed by a space and then your value. For example, “SHOT_ML 100”.

3. Prime the system with water by running the pump until water flows out of the group head

4. Place a receptacle under the group head to capture the water AND capture any water from the return line (the return line is normally connected to at least one OPV or pressure regulator valve that bleeds off high-pressure water and returns it to the reservoir)

5. Make sure "Espresso mode" is on

6. Use the coffee machine and run an espresso shot (no coffee in the basket needed). At the end of the shot and before resetting the brew lever, note the displayed value of water dispensed. For this example, let's assume the displayed water value is 102.8ml

7. Next, combine the water captured from the group head **AND** any water from the return line. Measure the total dispensed water on a set of scales (in grams) and note down the value. For this example, the weighted water is 89 grams.

8. To calculate the new corrected pulses per ml value. Using a calculator, first enter the displayed value, then, divide that number by the actual amount of water dispensed in grams, then, multiply by your current pulses per litre value. In our example, the equation would look like this: 102.8 / 89 = 1.155 1.155 x 6 = 6.93 (Round this number to the nearest decimal place, in this example 6.9 is our new pulses per ml value for the flow meter)

9. Update your pulses per millilitre for the flow meter by sending the **command** “FM1_HZ” followed by a space and then your value. For example, “FM1_HZ 6.9”

10. Now we can move on to calibrating flow meter 2 which measures water returned to the reservoir. For calibrating flow meter 2, it’s a good idea to use the portafilter with ground coffee to simulate the conditions and water flow behavior of brewing coffee.

11. Send **command** “ENABLE_FLOW_METER_2_CAL” to enable flow meter 2 calibration mode (this sets the SHOT_ML target at 40ml and the machine displays the amount of water from flow meter 2 only).

12. Install the portafilter with coffee in the group head. Place a receptacle to capture water from the return line. With espresso mode ON, run the machine to make a shot until the pump turns off.

13. At the end of the shot and before resetting the brew lever, note the displayed value of water dispensed. For this example, let's assume the displayed water value is 40.8ml

14. Next, measure the dispensed water on a set of scales (in grams) and note down the value. For this example, the weighted water is 53.7 grams.

15. To calculate the new corrected pulses per ml value for flow meter 2. Using a calculator, first enter the displayed value, then, divide that number by the actual amount of water dispensed in grams, then, multiply by your current pulses per litre value.

    In our example, the equation would look like this:
    40.8 (displayed output of water) / **53.7**(actual amount of water dispensed) = **0.7597**
    0.7597 x 20 (current FM2_HZ value) **= 15.194**
    Round this number to the nearest decimal place, in this example 15.2 is our new pulses per ml value for the flow meter 2.

16. Update your pulses per litre for the flow meter by sending the **command** “FM2_HZ” followed by a space and then your new value. For example, “FM2_HZ 15.2”

17. Lastly disable flow meter 2 calibration mode by sending **command** “DISABLE_FLOW_METER_2_CAL”. Set your desired shot volume by sending the **command** “SHOT_ML” followed by a space and then your value. For example, “SHOT_ML 60”.

Pre-infusion:

INFO: When Pre-infusion is enabled and espresso mode is on, a pre-infusion cycle will run for each espresso shot.

Pre-infusion stages explained:
1. Prime the system with water → 2. Cycle the pump on/off for a set amount of time → 3. Run the pump until the programmed espresso target has been achieved.

To **enable** pre-infusion, send **command** "ENABLE_ PRE_INFUSION" (or to disable send command "DISABLE_ PRE_INFUSION")

Prime Pump: At the beginning of pre-infusion, it's often necessary to run the pump for a brief amount of time to flush the group head with hot water.
To change the prime pump timer, send the **command** "PRIME_PUMP" followed by a space and then your value in milliseconds. For example: "PRIME_PUMP 3000" for a 3 second timer (NOTE: the timer is measured in milliseconds. 1 second = 1000ms)
Default value: 2000

Infusion Hz: Infusion Hz programs the speed the pump cycles on/off during pre-infusion per second. For example, an infusion_hz value of 5 will cycle the pump 5x per second.
To change the infusion_hz value, send the **command** "INFUSION_HZ" followed by a space and then your value. For example, "INFUSION_HZ 6"
Default value: 5

Infusion Time: You can program the duration of the pre-infusion stage. To change the value, send **command** "INFUSION_TIME" followed by a space, then your value in milliseconds. For example, "INFUSION_TIMER 10000" for a 10 second pre-infusion duration (NOTE: the timer is measured in milliseconds. 1 second = 1000ms).
Default value: 5000

Uploading the firmware:

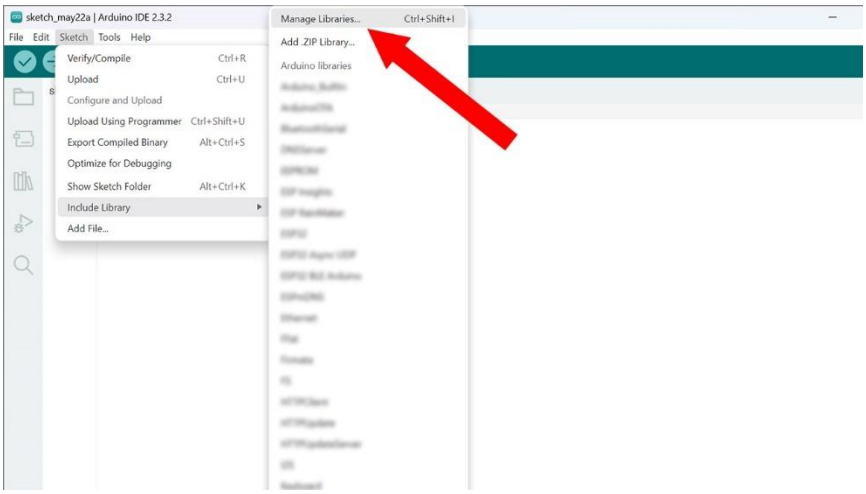
Installing Arduino IDE libraries:

To upload the firmware, its necessary to install the required libraries to compile the firmware.

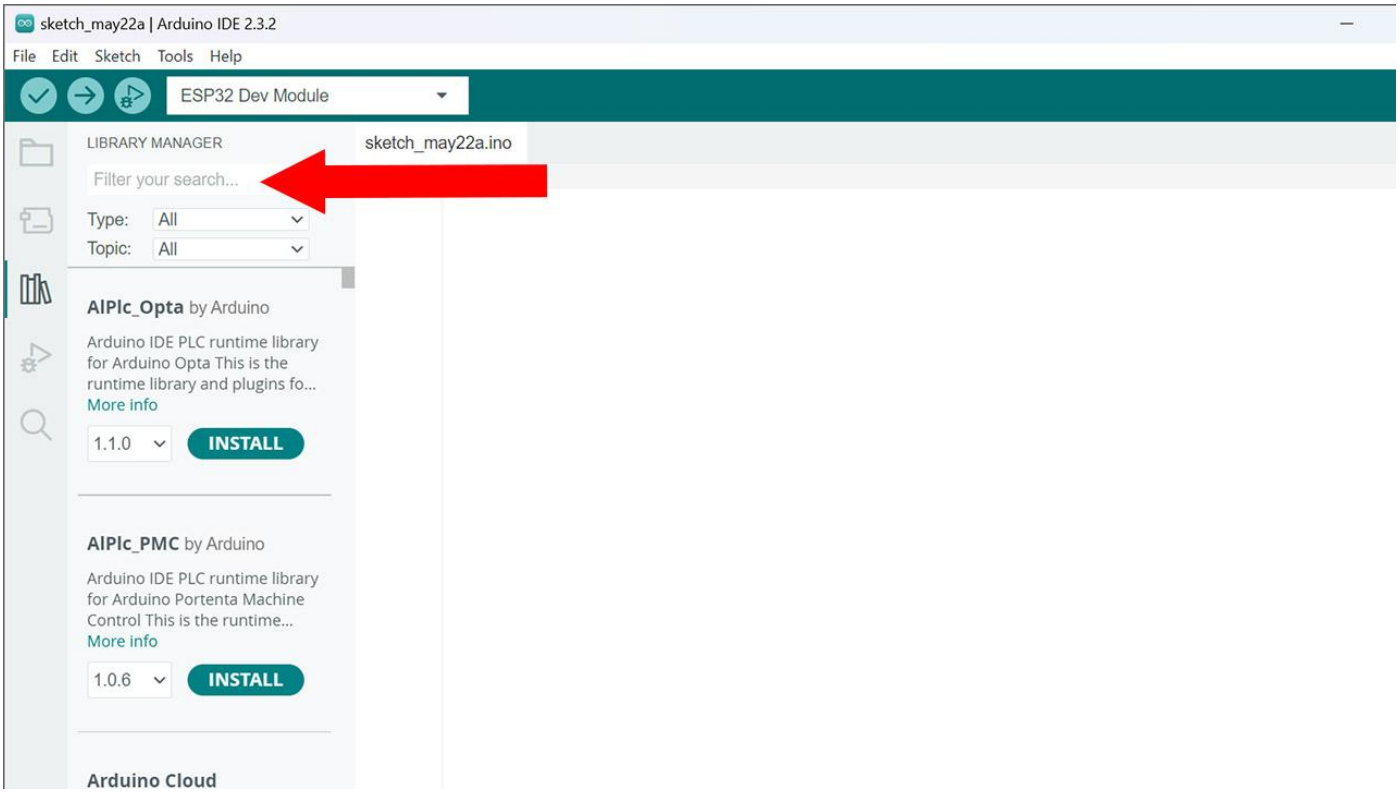
List of libraries to be installed in Arduino IDE:

- HX711 by Rob Tillaart - https://github.com/RobTillaart/HX711
- Adafruit SH110X by Adafruit - https://github.com/adafruit/Adafruit_SH110X
- AutoPID by Ryan Downing - https://r-downing.github.io/AutoPID/
- ArduinoJson by Benoit Blanchon - https://arduinojson.org/?utm_source=meta&utm_medium=library.properties
- LittleFS_esp32 by lorol - https://github.com/lorol/LITTLEFS
- Adafruit MAX31865 by Adafruit - https://github.com/adafruit/Adafruit_MAX31865

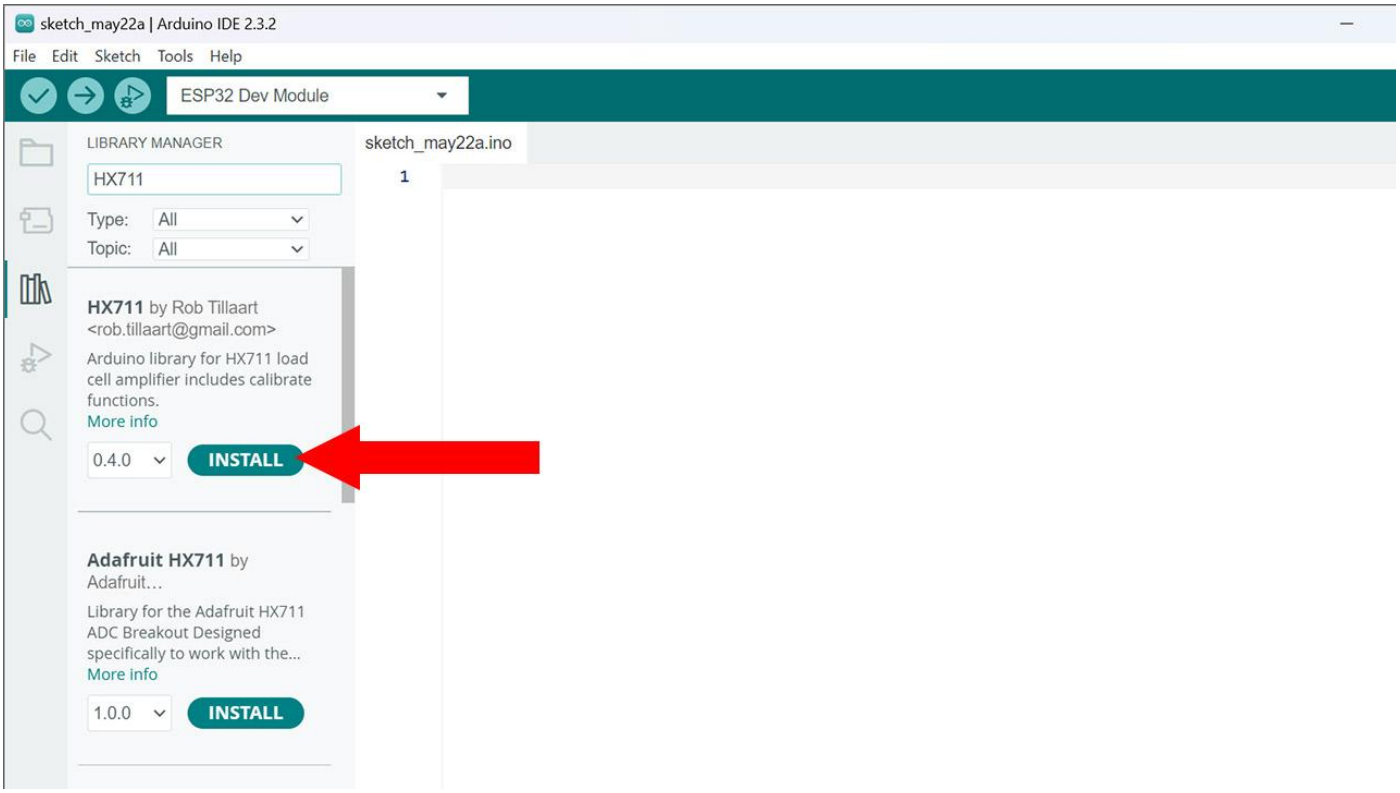
1. To install a library, first start Arduino IDE. From the tool bar select, Sketch → Include Library → Manage Libraries.



2. In the search box, entre the name of the library you want to install.



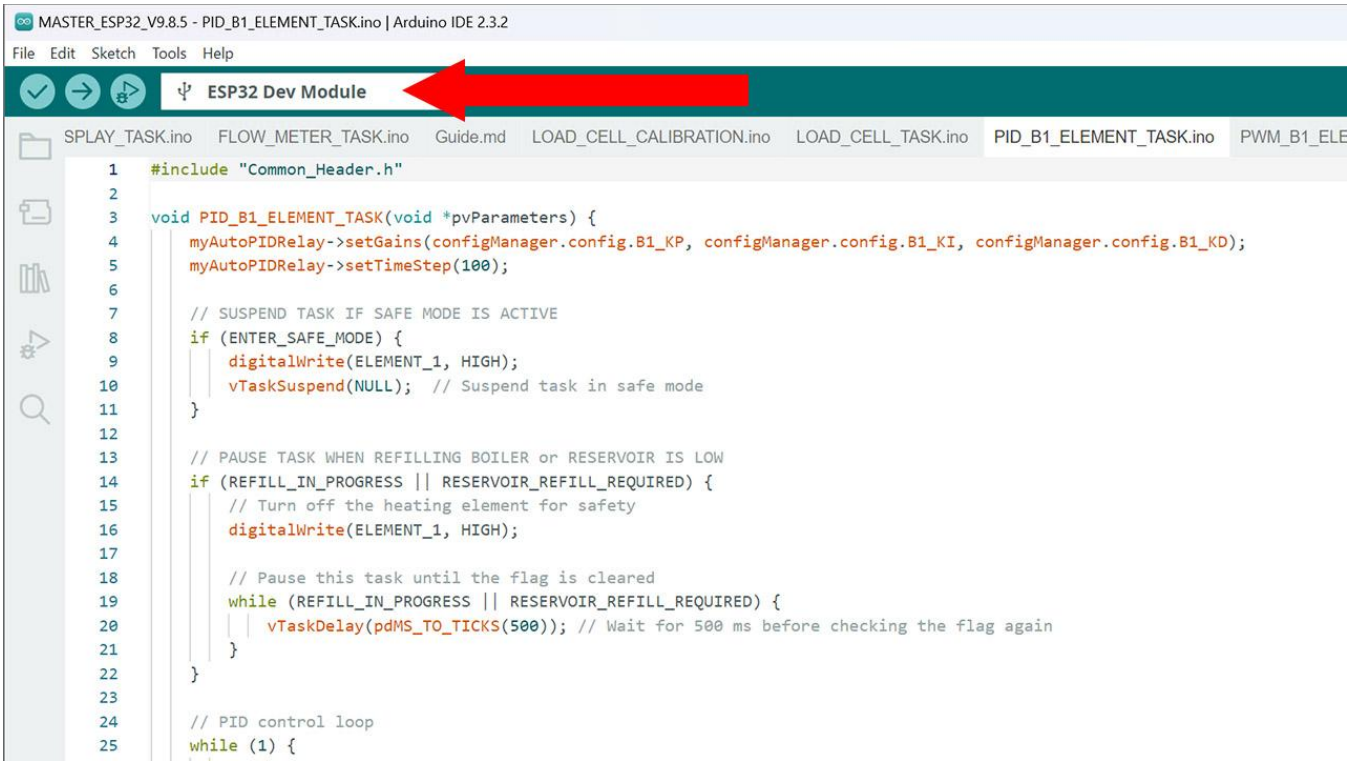
3. Install the library



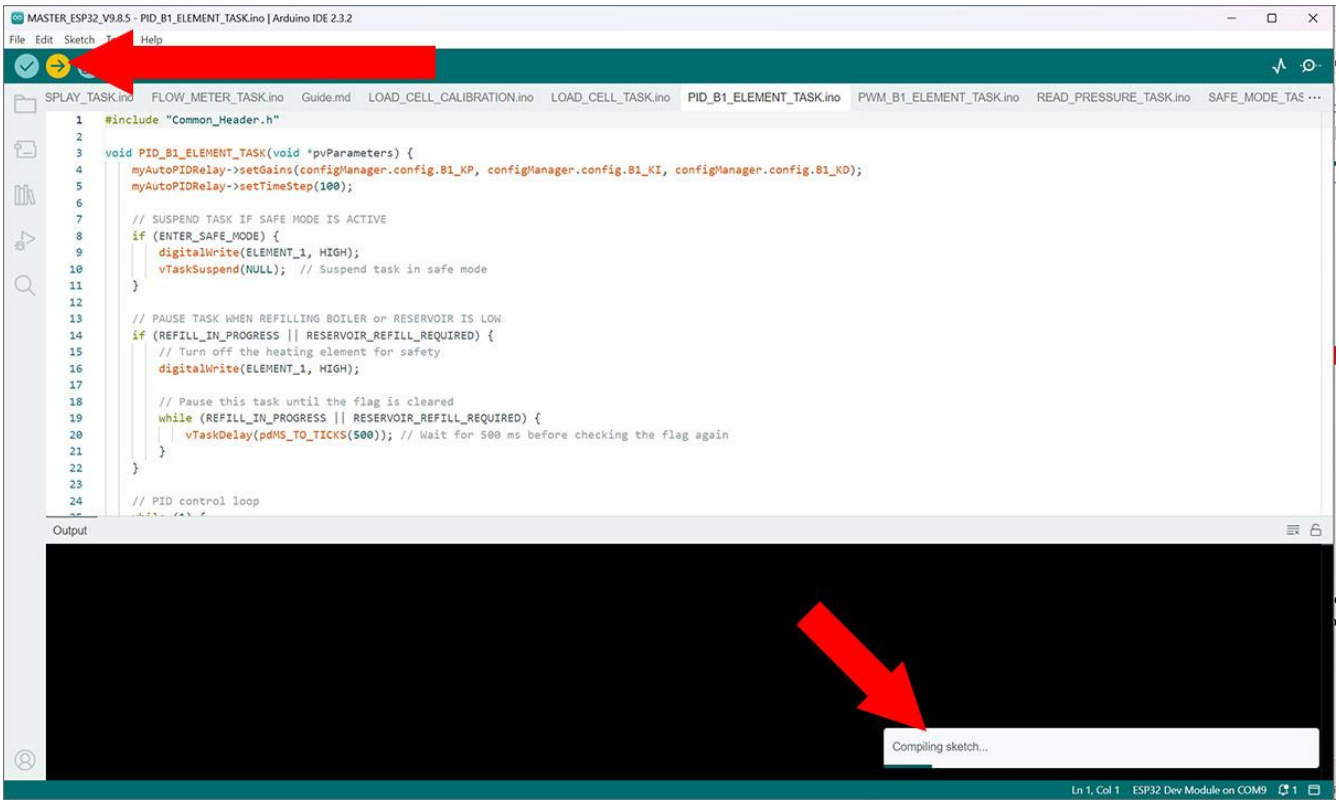
Uploading Firmware:

For Smart Coffee boards that feature 2x ESP32 modules, there will sepirate dedicated programs for each ESP32. The programs are not interchangeable and MUST be written to the correct ESP32. The PCB will have “Master” and “Slave” labels printed next to the ESP32s. Upload the code labeld “Master” to the Master-ESP32. And upload the code labeld “Slave” to the Slave-ESP32.

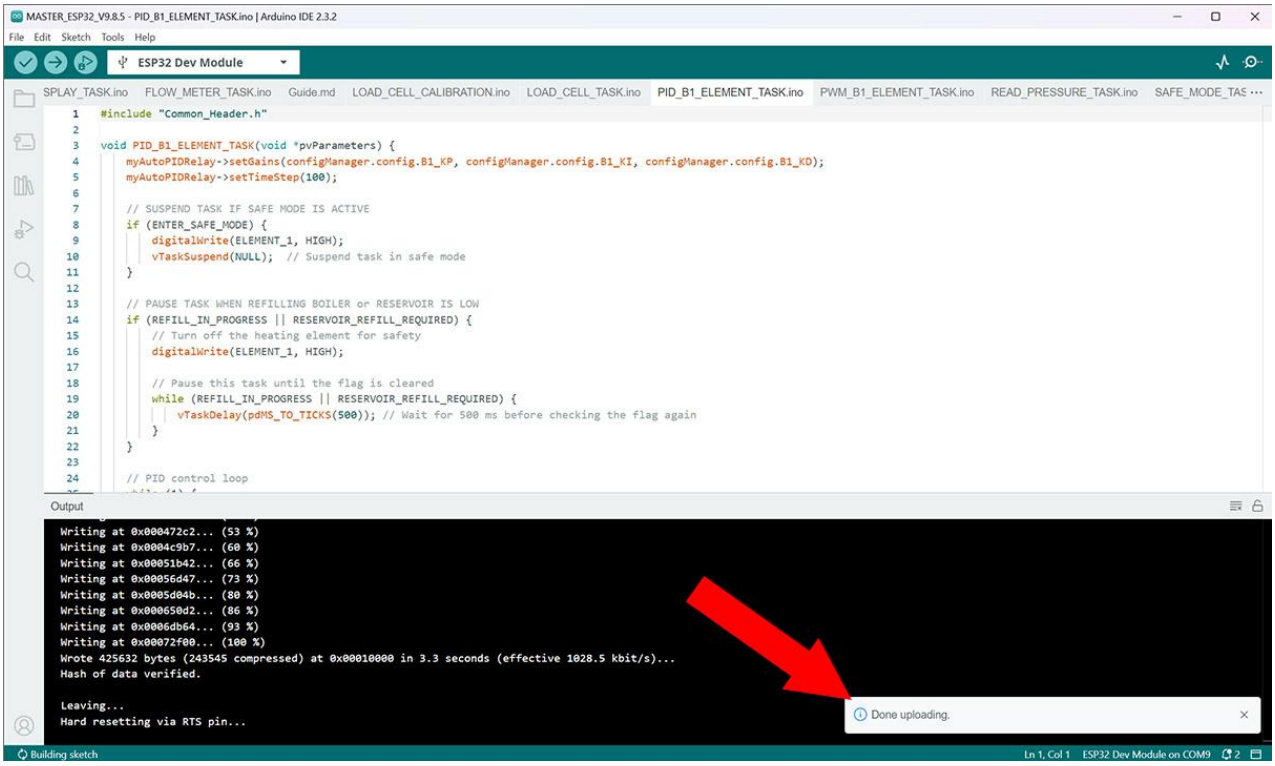
1. Select the COM port the ESP is connected to. To learn more click here



2. Click the top left upload button to begin compiling the code. Once compiling has finished, the code will be written to the ESP32



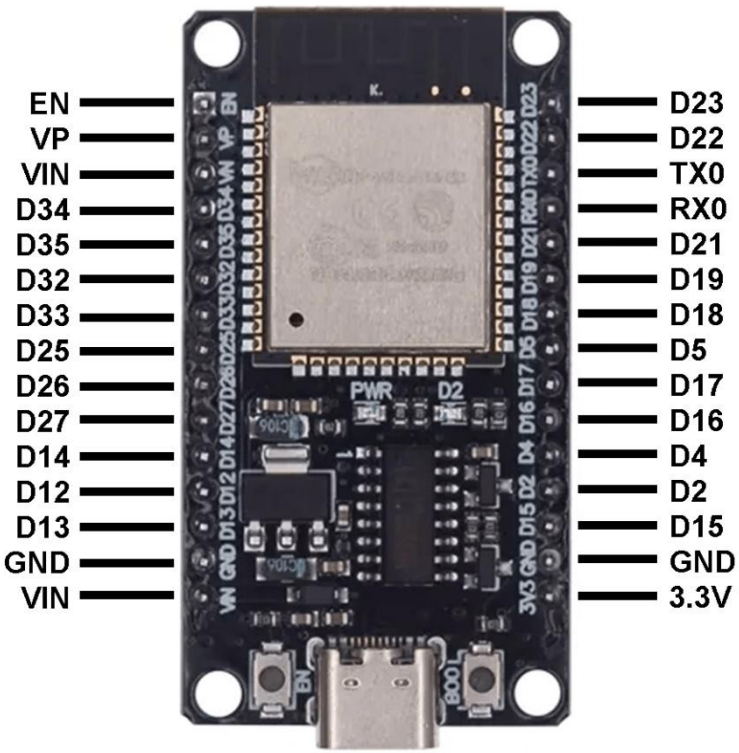
3. After successfully uploading the code, a confirmation message will say “Done uploading.”.



ESP32 pinout compatibility

The Smart – Coffee PCB has 2 sockets that are compatible with standard ESP32, 30pin development boards.

ESP32 - 30pin Development board



Please note: There are other variations of ESP32 modules that have different pin configurations that are not compatible with Smart – Coffee PCB