Co-funded by the European Union

Gerda Stetter Stiftung
Technik *macht* Spaß!

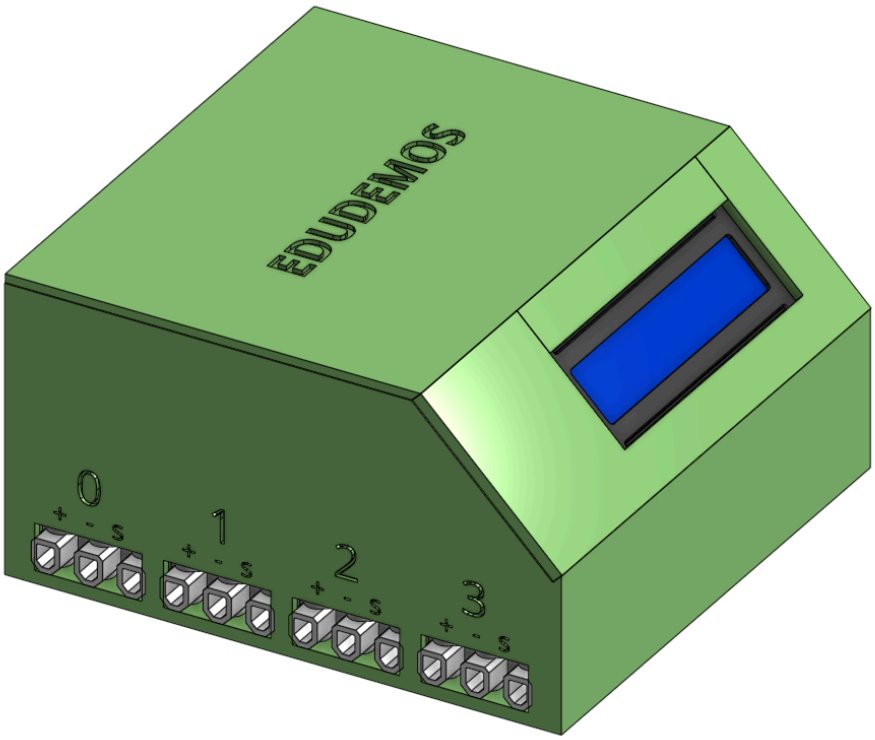
EDUDEMOS

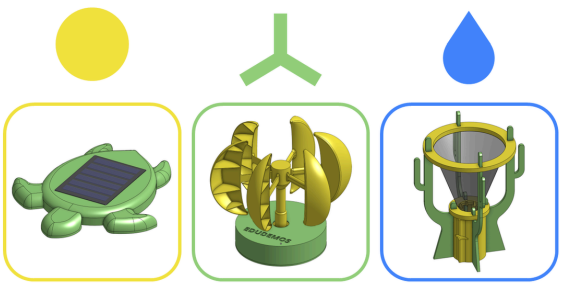
EDUcating through Sustainable **DEMO**nstrators

Assembly Guide

Modular Weather Station

Controlbox Module





Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Gerda Stetter Stiftung Technik macht Spaß! · Finnova · gbs sg.ch · FUNDACIÓN SERGIO ALONSO · I.E.S. El Rincón

Content

Materials

Electrical Components

- 1x Microcontroller (ESP8266 NodeMCU Board Amica)
- 1x LCD-Display (I2C 16x2)
- 1x PCF8591 (Analog to digital converter)
- 1x ACS712 (5A) (Current Sensor)
- 1x Breadboard (Midi)
- 8x 1kΩ Resistor
- Jumper Wires
  - 16x 20cm F-M
  - 16x 10cm M-M
  - 12x 20cm M-M
- 13x Luster Terminals
- USB-C Port (Female)

3D-Printer

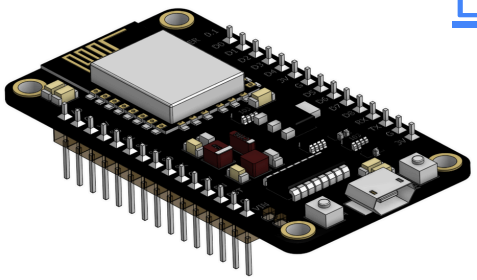
- 3D-Printer filament (ca. 160g)

Not included

- USB-A to USB-C cable for power supply
- USB-C to Micro-USB adapter or USB-A to Micro-USB cable
- Flathead Screwdriver
- Tweezers (recommended for assembly)

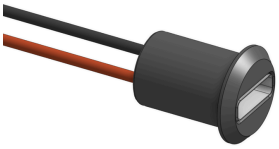
Modular Controlbox

Electronics

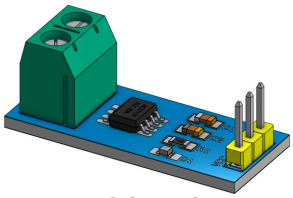


ESP8266 NodeMCU Board
x1

Resistor 1kOhm
x8

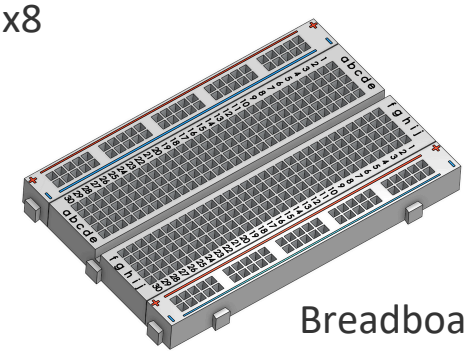


USB-C Port Female
x1

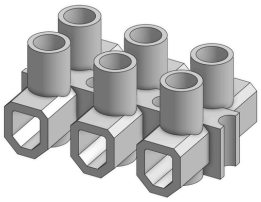


ACS712
x1

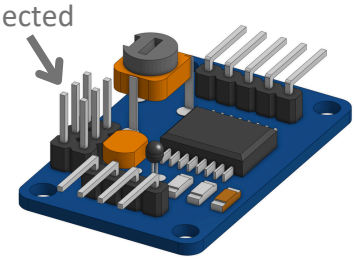
These 6 pins should not be connected



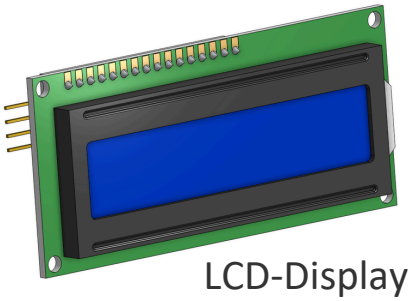
Breadboard
x1



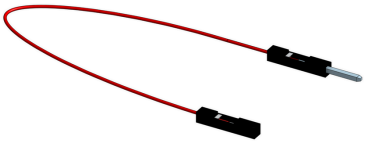
Luster Terminals
x14



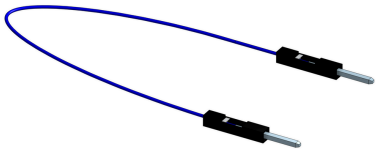
PCF 8591*
x1



LCD-Display
x1



Jumper Wire female-to-male
(F-M)**
x15



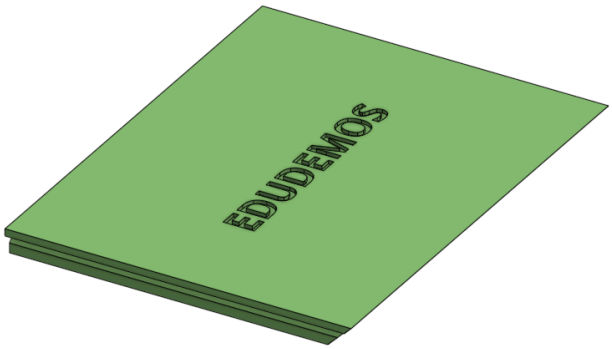
Jumper Wire male-to-male
(M-M)**
x28

*Make sure to remove all jumpers from the PCF8591
**The colours of the jumper wires are irrelevant.

Total 62 elements

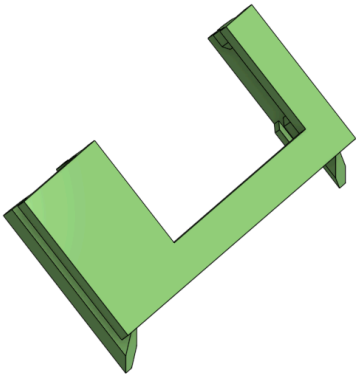
Modular Controlbox

3D Printed Parts

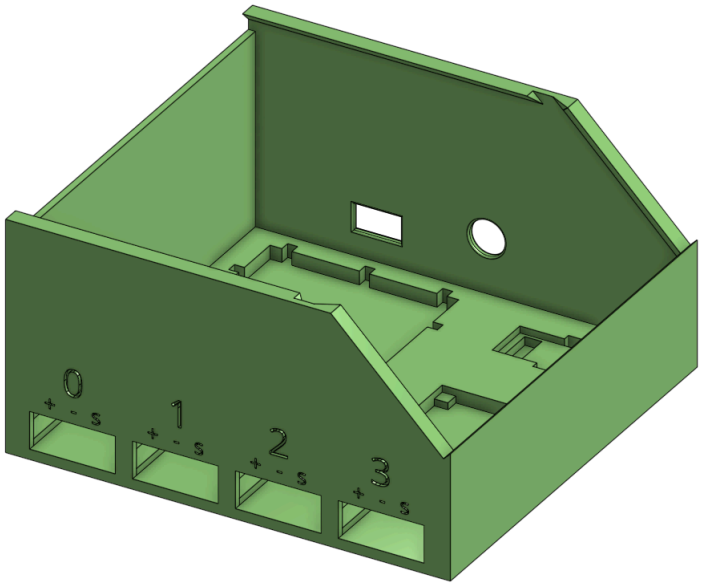


Controlbox Lid
x1

Display Fastener
x1

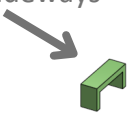


Display Holder
x1



Controlbox Base
x1

This part should be printed sideways



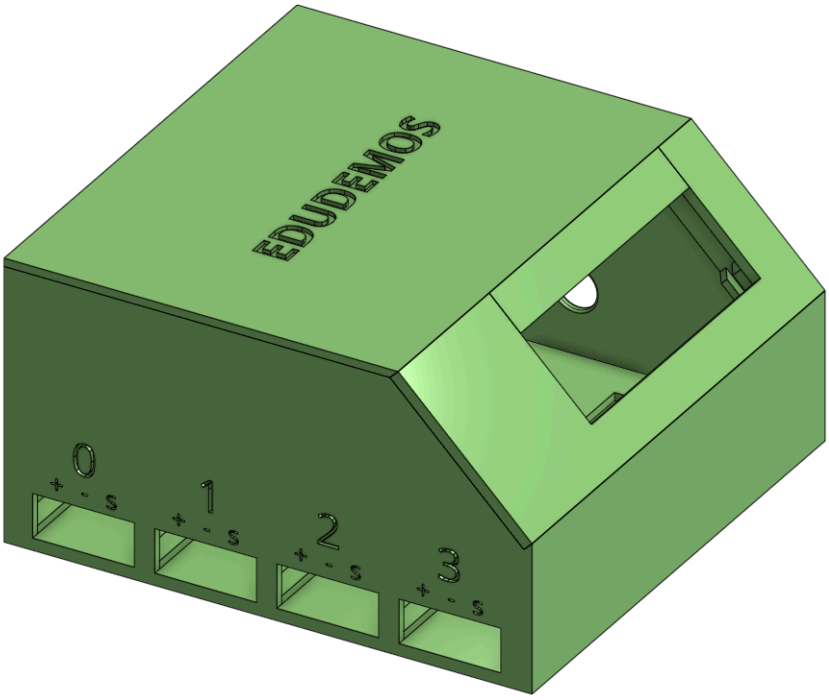
ACS712 Fastener
x1

Total 5 elements

Step By step Instructions

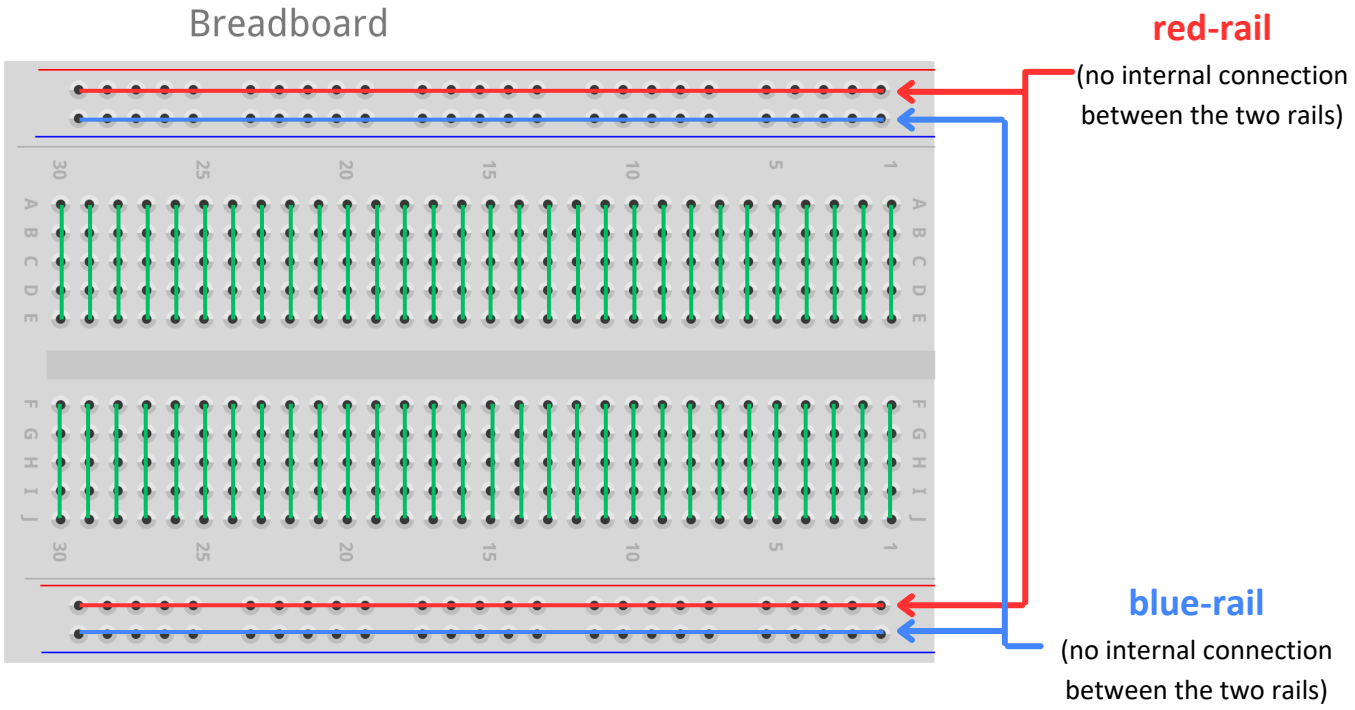
---

a. Print the parts of the Modular Controlbox on any 3D-Printer



The STL-printfiles are available on the EduDemoS webpage.

B. Wiring and assembling Instructions

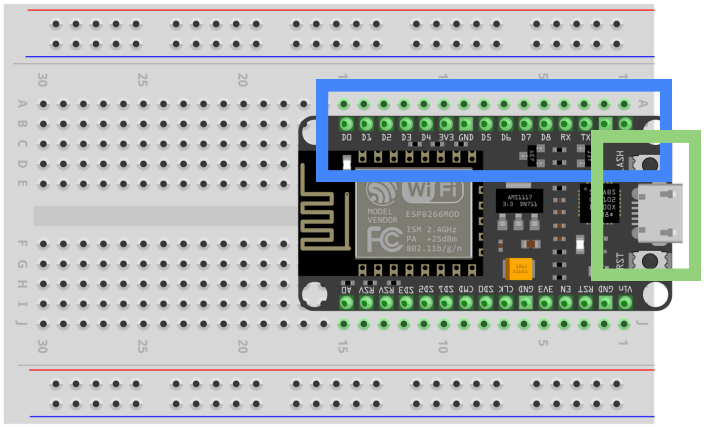


1. Take the breadboard and rotate it like in the picture.

ⓘ In our case we use the blue rail for Ground connection and the red rail for power supply.

ⓘ Keep in mind that the breadboard connects each line and rail in the way the lines are drawn. For example a pin or Jumper plugged in to A5 is connected with all Jumper and pins plugged into B5, C5, D5 and E5.

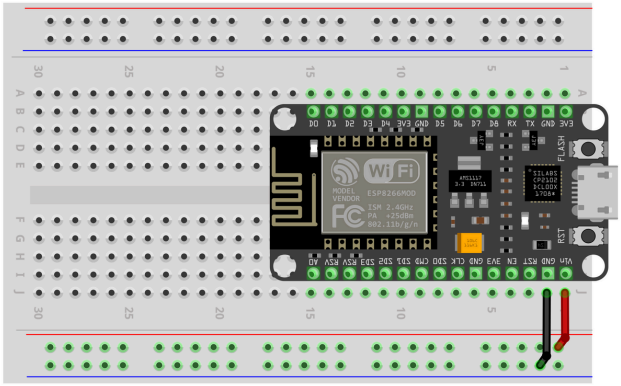
Breadboard & ESP8266



2. Place the ESP8266 on the Breadboard
   - Make sure that the **USB - Port** is facing outward
   - The **first pin** of the ESP8266 must be **in line with the first row** of the breadboard

ⓘ You can find more information on the ESP8266 pinout on page 29.

Breadboard & ESP8266

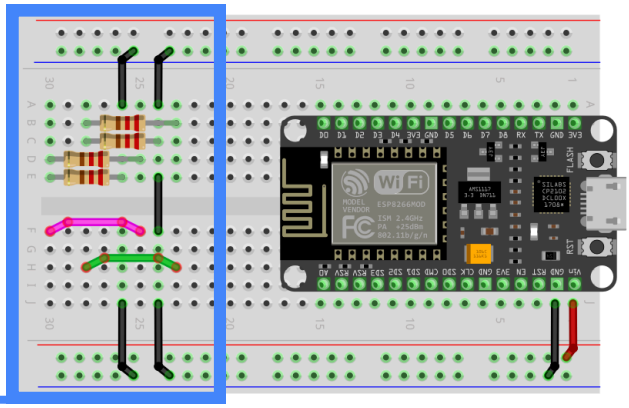


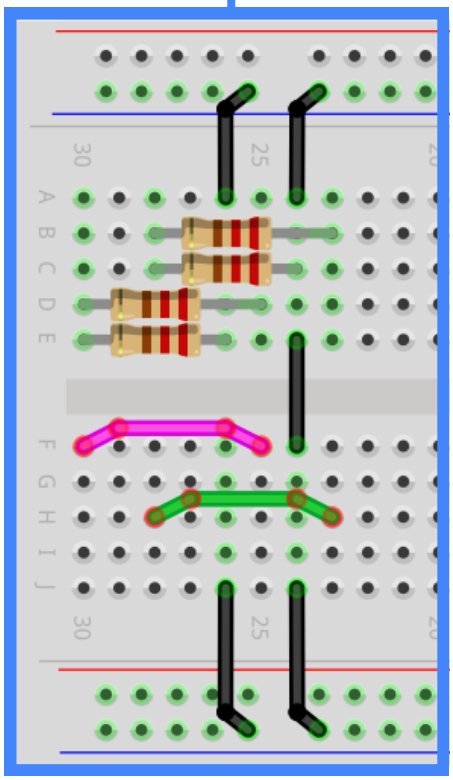
3. Connect the **VIN pin** and **GND pin** of the ESP8266 to the **red-rail** and the **blue-rail** on the breadboard using short male-to-male (M-M) jumper wires.

ⓘ You can trimm the resistor legs for a better fit, allowing them to lay flat on the breaboard.

4. Insert the four **1kΩ resistors** into the breadboard as shown on the right. These resistors will form the two voltage dividers. Also insert **two short jumper wires** as placeholders which can be used for your own demonstrators.

Breadboard & ESP8266

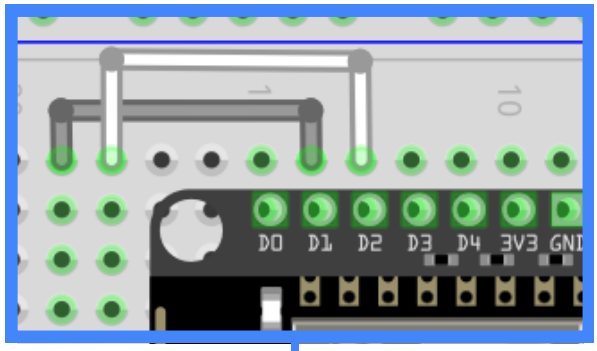




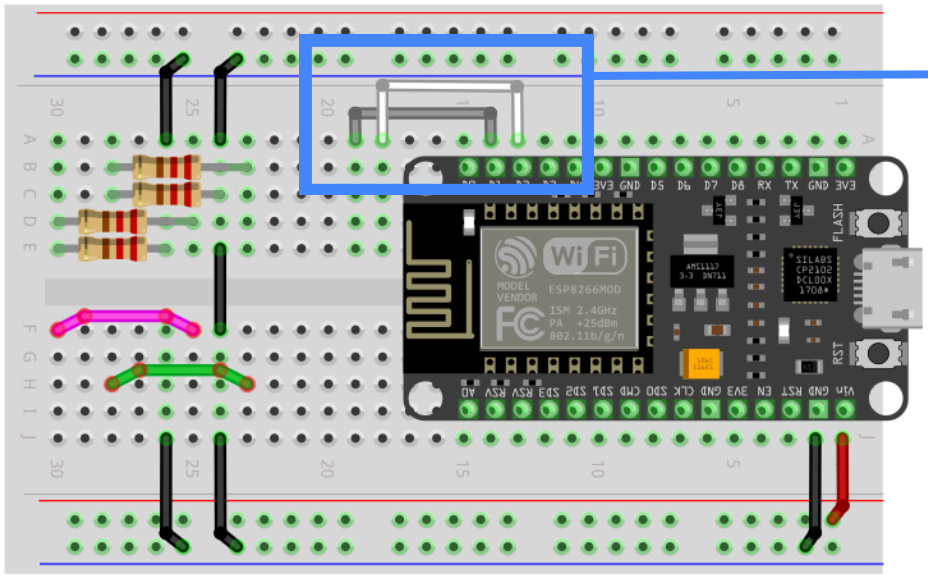
5. Connect the voltage dividers to the **blue-rail** of the breadboard.
   - Use short M-M jumper wires

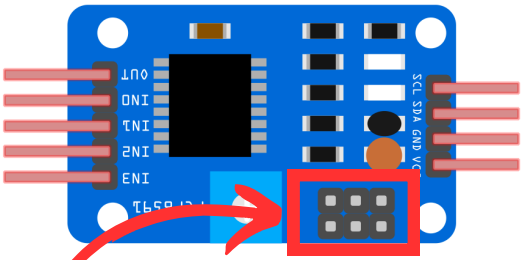
ⓘ A voltage divider is a simple electrical circuit that reduces a higher voltage to a lower voltage by a factor depending on the used resistors.
In this case we use two equal resistors so the voltage is divided by the factor two.

6. Connect the D1 and D2 pins from the ESP8266 to rows 18 and 19 on the breadboard.



Breadboard & ESP8266



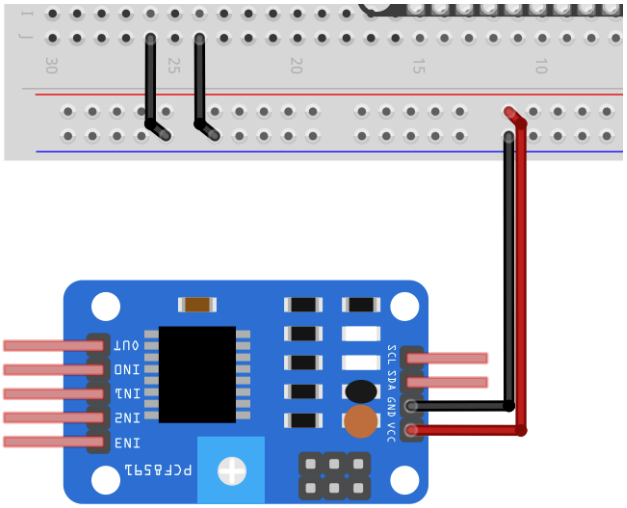


7. Now you need the PCF8591 Board. Try to position it as shown in the picture so it will fit into the Controlbox later.

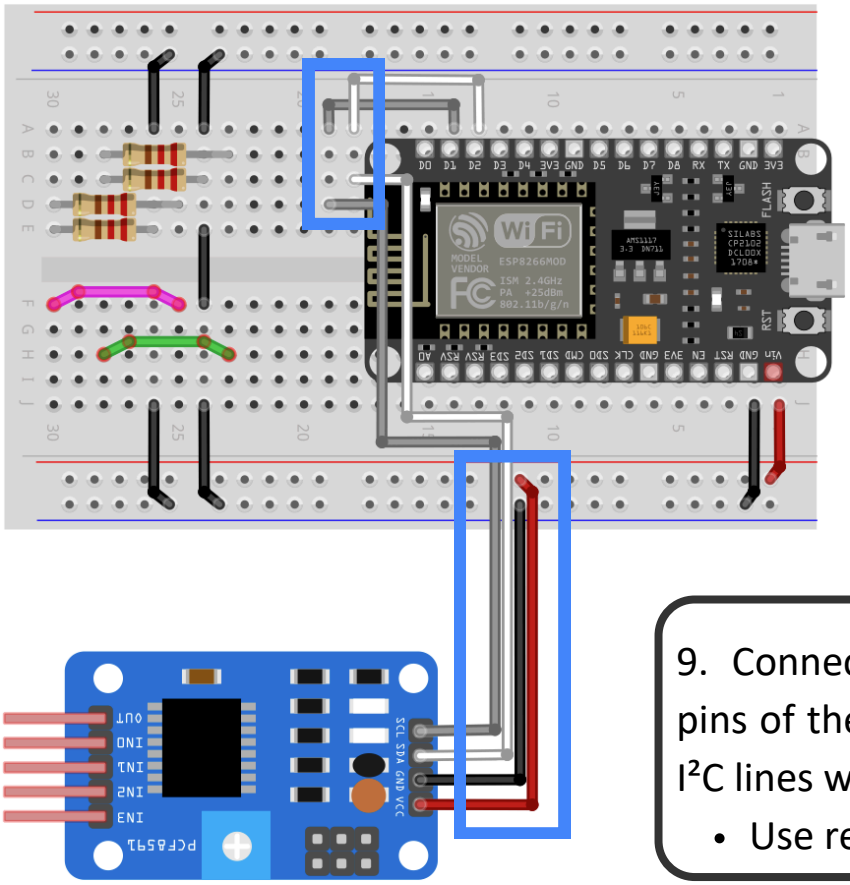
On the PCF8591 you might find **3 small jumpers** which connect 3x2 pins. Make sure to take the jumpers off!

ⓘ I²C is a way for electrical components to communicate. It uses two wires to send and receive digital messages. You can connect multiple components with the same pins because each device has a special adress assigned to it.

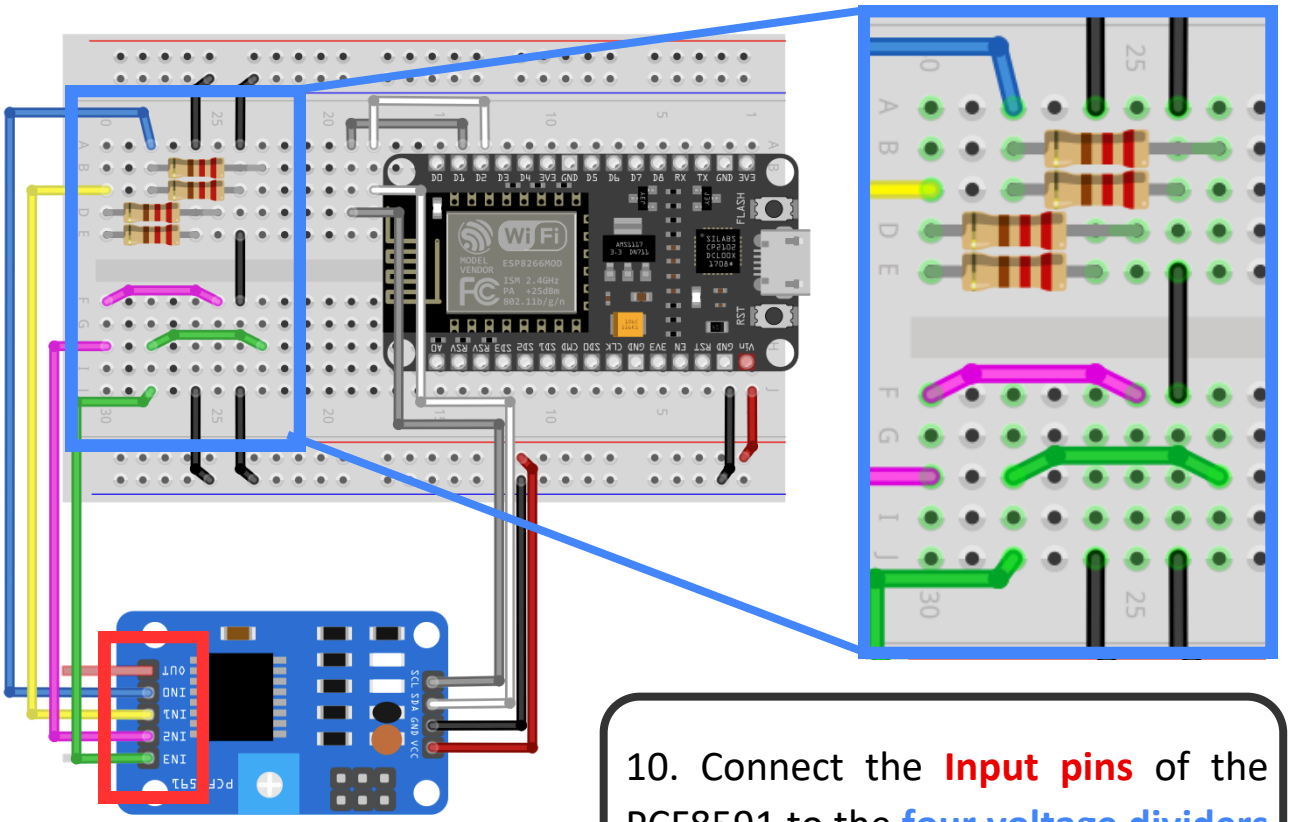
8. Connect the **VCC pin** and **GND pin** of the ESP8266 to the **red-rail** and **blue-rail** of the breadboard using short female-to-male (F-M) jumper wires.



Make sure to check the pin-names of your PCF8591 Board since some manufacturers use a different order.



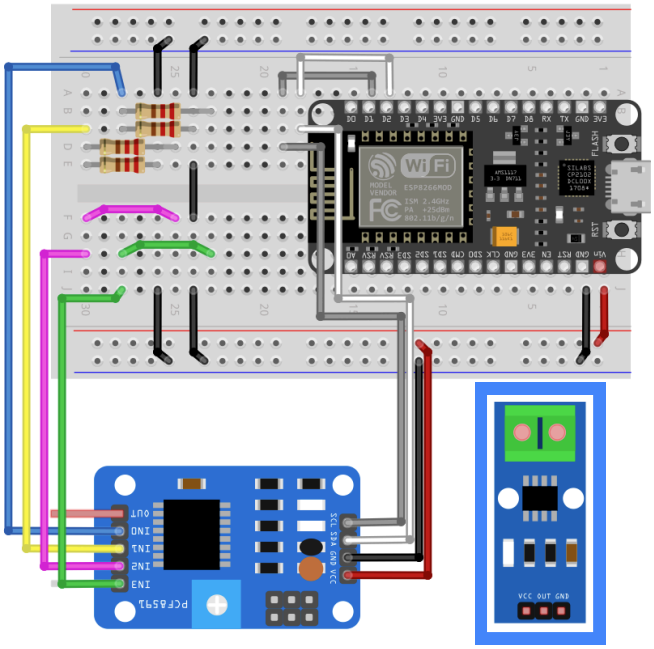
9. Connect the **SCL** and **SDA** pins of the PCF8591 to the two I²C lines we started before.
   - Use regular F-M Jumpers

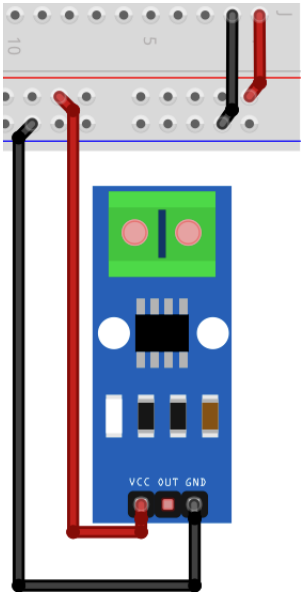


10. Connect the **Input pins** of the PCF8591 to the **four voltage dividers** we made before.
    - Use regular F-M Jumpers

ⓘ The PCF591 takes the voltage input and converts it into a value between 0 and 255. It sends this value to the ESP8266 in a digital signal called I2C.
The ESP8266 can request the assigned value for each input channel of the PCF8591 seperately.

11. For the next steps you need the **ACS712 (Current Sensor)**. Try to position it as shown in the picture to get it to fit into the box later.

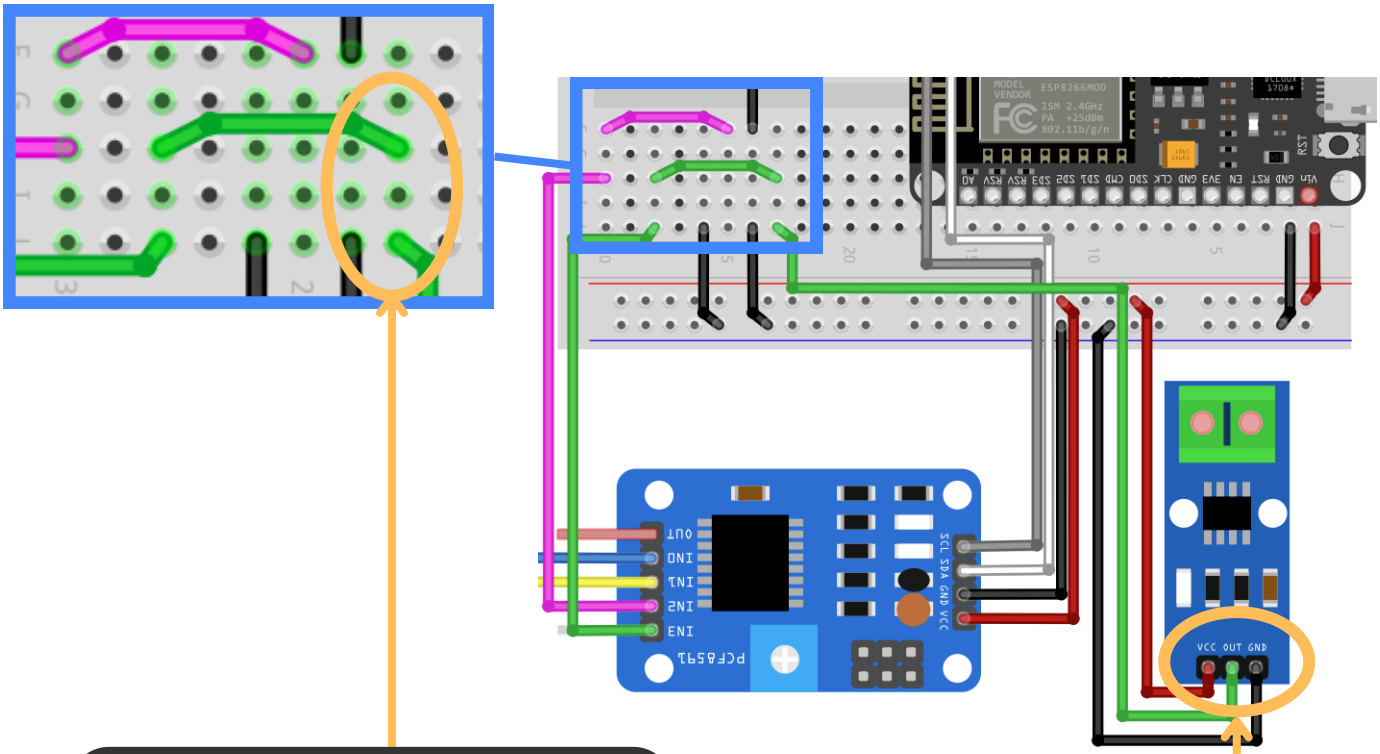




12. Connect the **VCC pin** and **GND pin** of the current sensor to the **red-rail** and **blue-rail** of the breadboard.

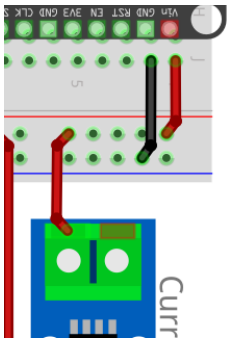
- Use short F-M Jumpers

ⓘ The ACS712 is a current sensor that uses a Hall effect sensor to measure AC or DC current and outputs an analog voltage proportional to the current flow.



13. Connect the **OUT pin** of the Current Sensor to the **input of the fourth voltage** divider.

- Use a regular F-M Jumper



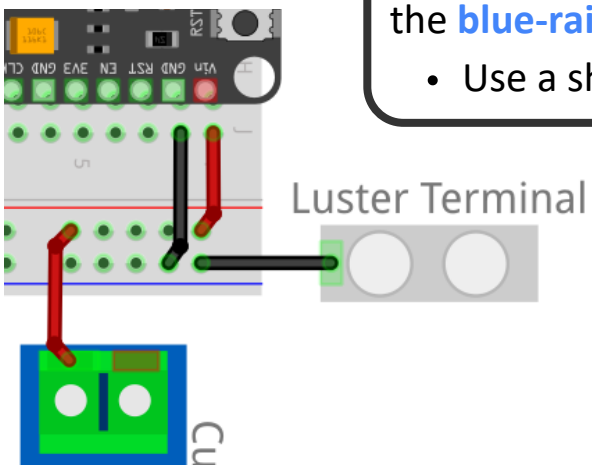
14. Partially unscrew the **left terminal connector** of the Current Sensor, put one side of the jumper in the opening and screw it tight again. Connect the other end of the jumper wire to the **red-rail** of the breadboard.

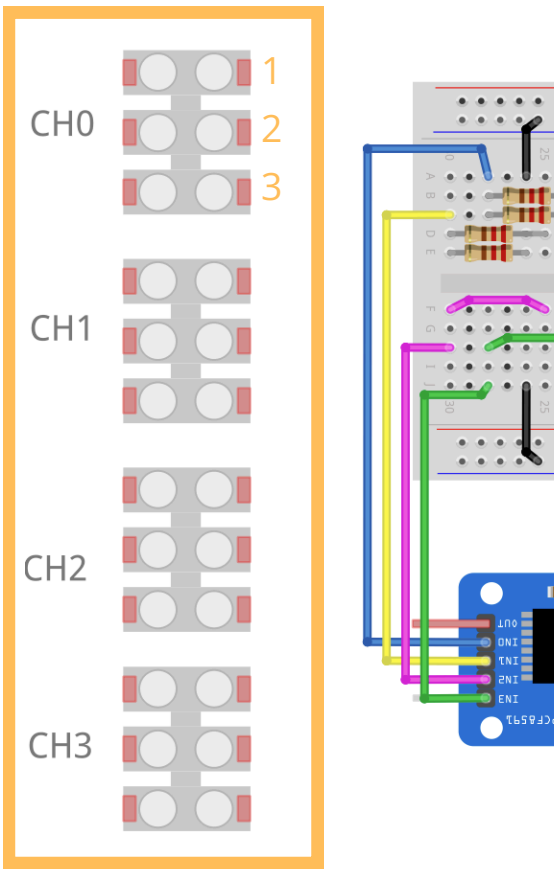
- Use a short M-M Jumper

15. Get a **single Luster Terminal** so you can connect the USB-C connector later.

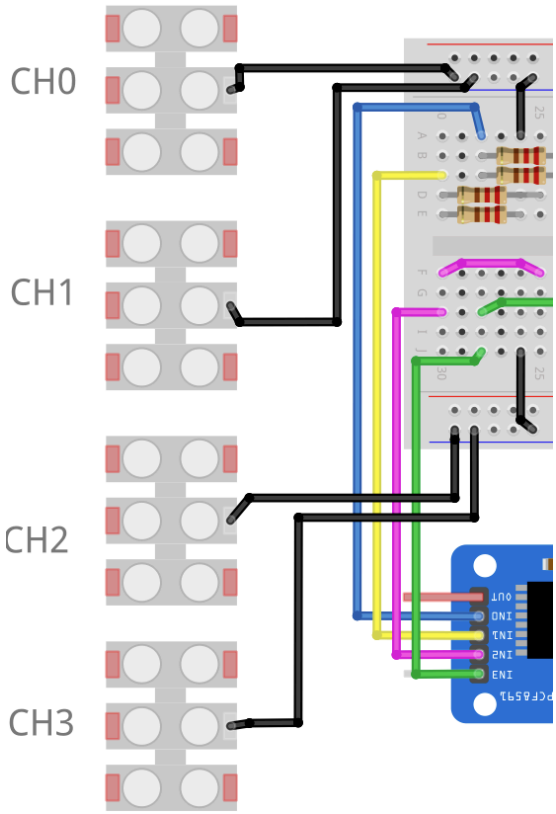
16. Connect one end of a jumper to one pole of the luster terminal and stick the other end of the jumper into the **blue-rail** of the breadboard.

- Use a short M-M jumper wire

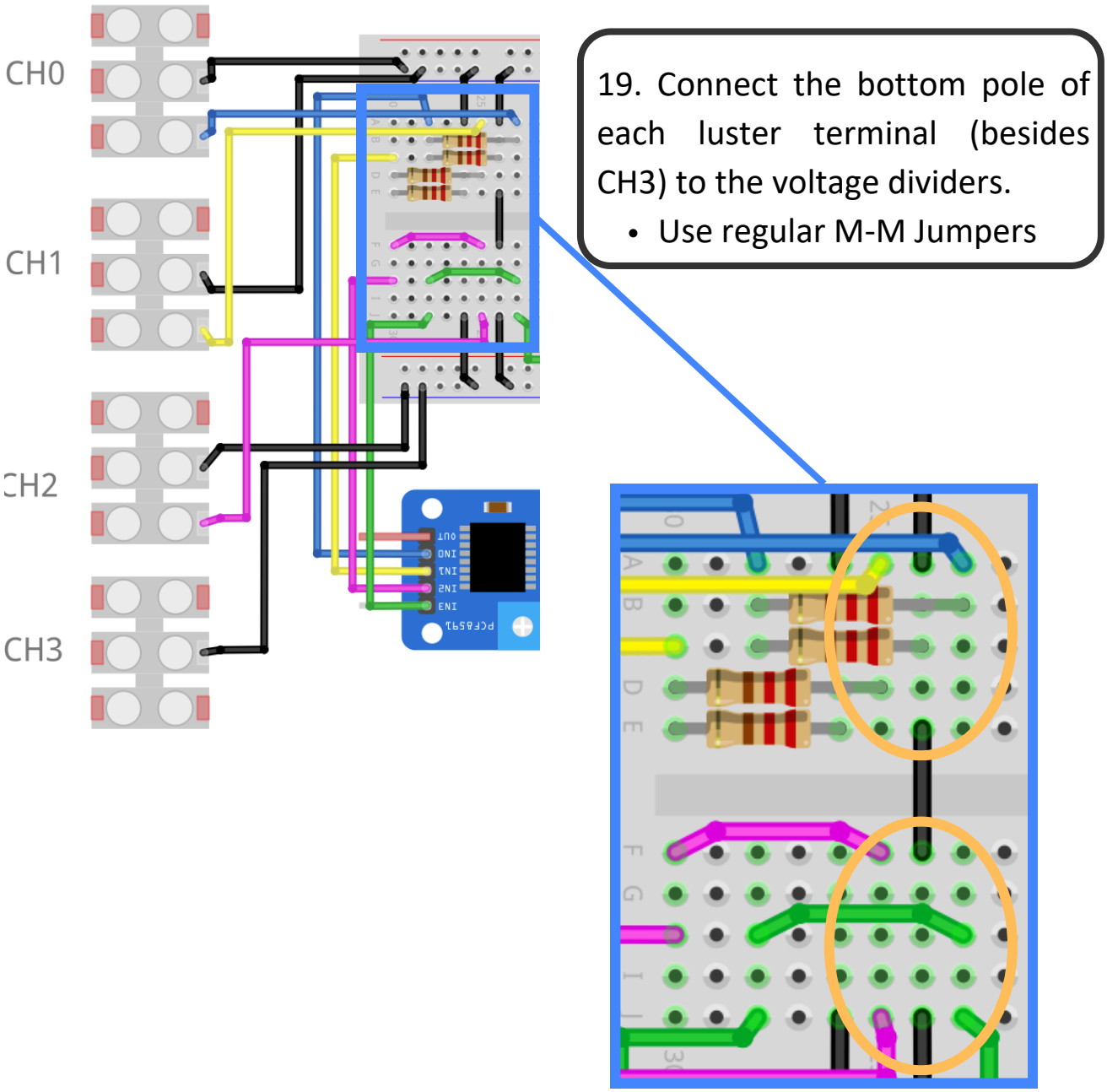




17. For the next steps, you will need **four luster terminals**, each with **3 poles**. These terminals will connect the control box to various modules. Position them as shown in the picture to ensure everything fits into the box later.



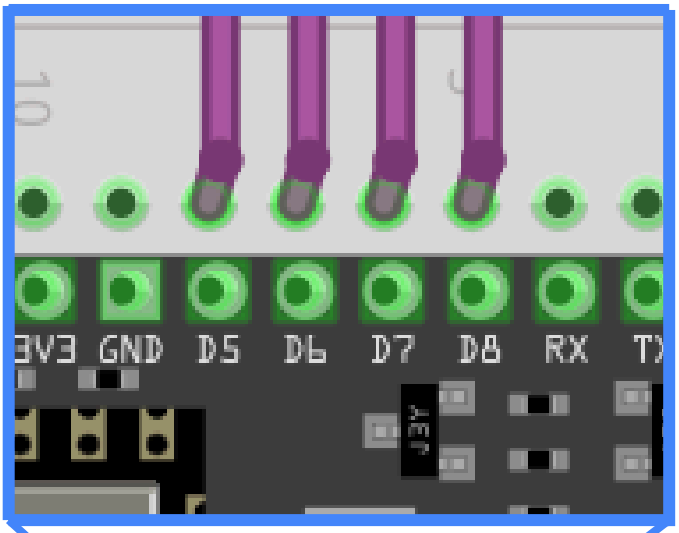
18. Connect the middle poles of all luster terminals to the **blue-rail** of the breadboard.
    - Use regular M-M jumper wires

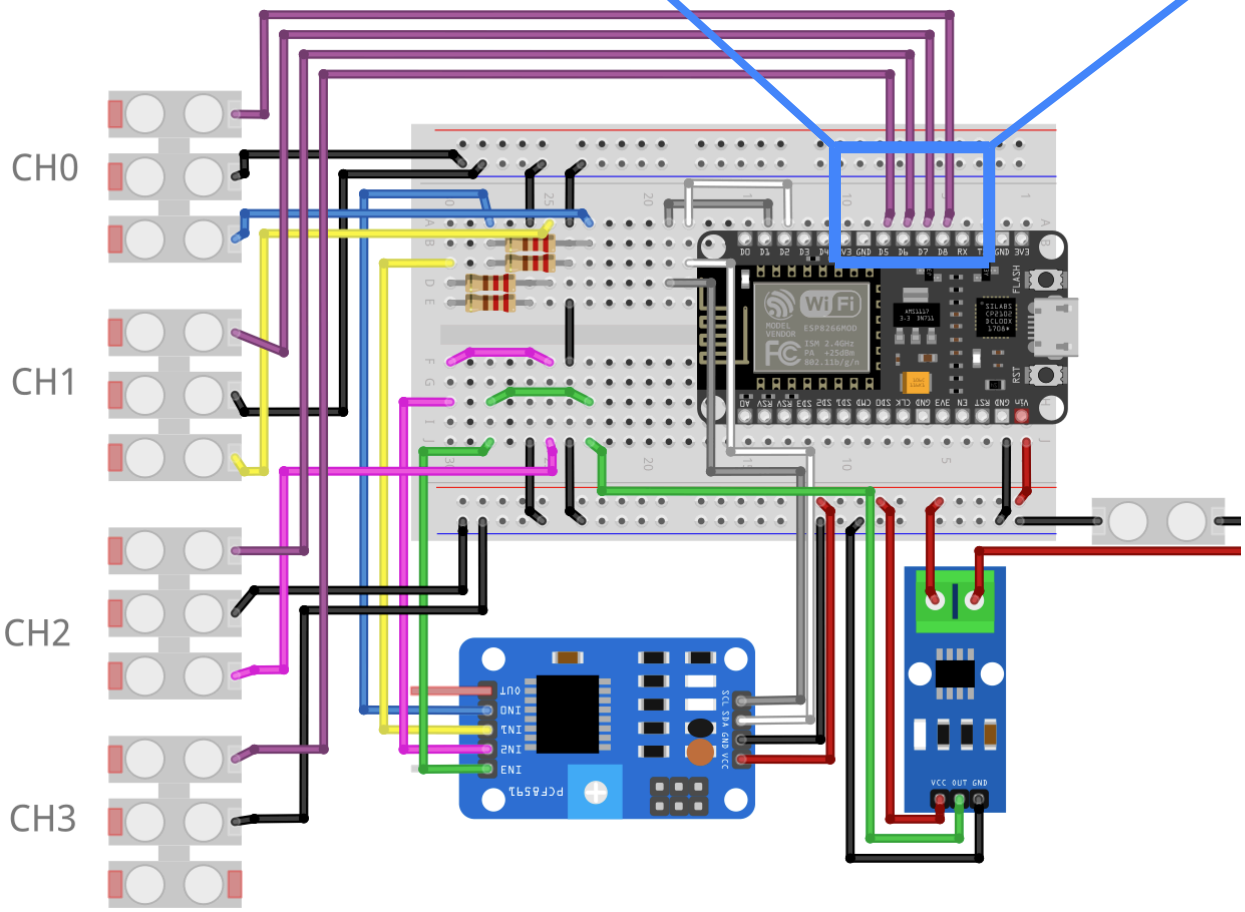


19. Connect the bottom pole of each luster terminal (besides CH3) to the voltage dividers.
    - Use regular M-M Jumpers

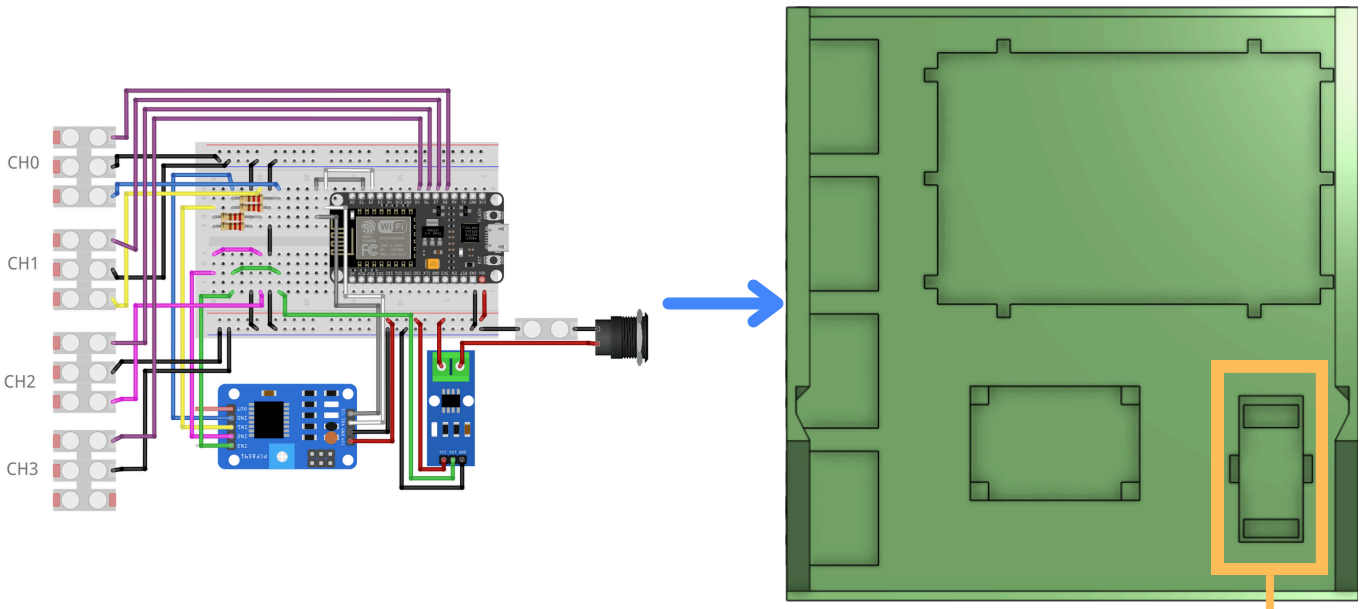
20. Connect the top poles of each luster terminal to the ESP8266.

- **CH0->D8**
- **CH1->D7**
- **CH2->D6**
- **CH3->D5**

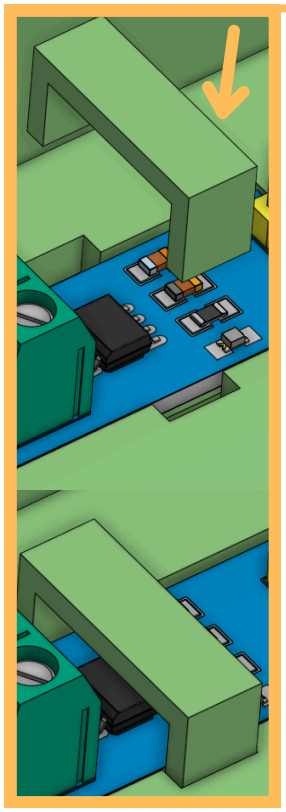




GPIO pins can provide power to low-power modules and devices, enabling them to operate while also facilitating signal input for control and communication.
In this case they are especially important for the water level sensor because it should not be powered the whole time but only every few seconds to prevent it from rusting.



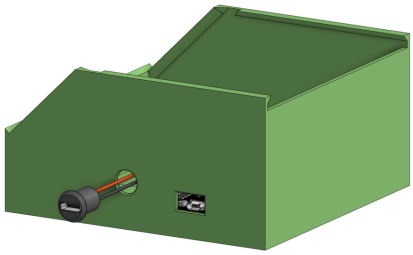
21. Put everything in the 3D-printed **Controlbox** base, the parts should fit in the holes.
    - It might be helpful to glue the luster terminals in place. If you decide to do so, keep in mind that you have to reach all screws with a screwdriver to connect the modules later.

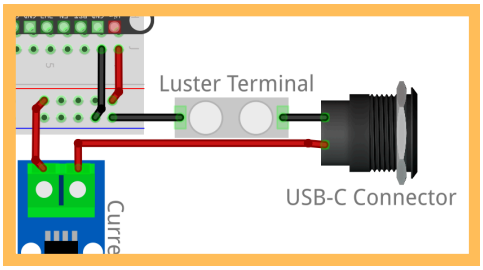


22. Fasten the Current Sensor with the small bridge to keep it in place.

ⓘ It is important to keep the Current Sensor in place because of the way it measures the current. If the sensor is near other wires, the current in the wires could interfere with the magnetic field of the current that is measured.

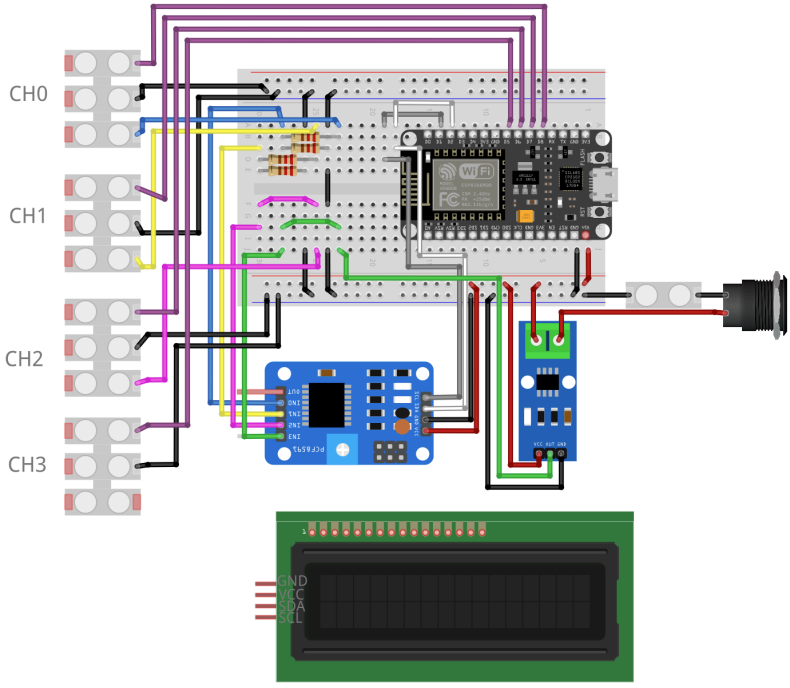
23. Put the **USB-C Port** through the hole in the right wall of the controlbox.



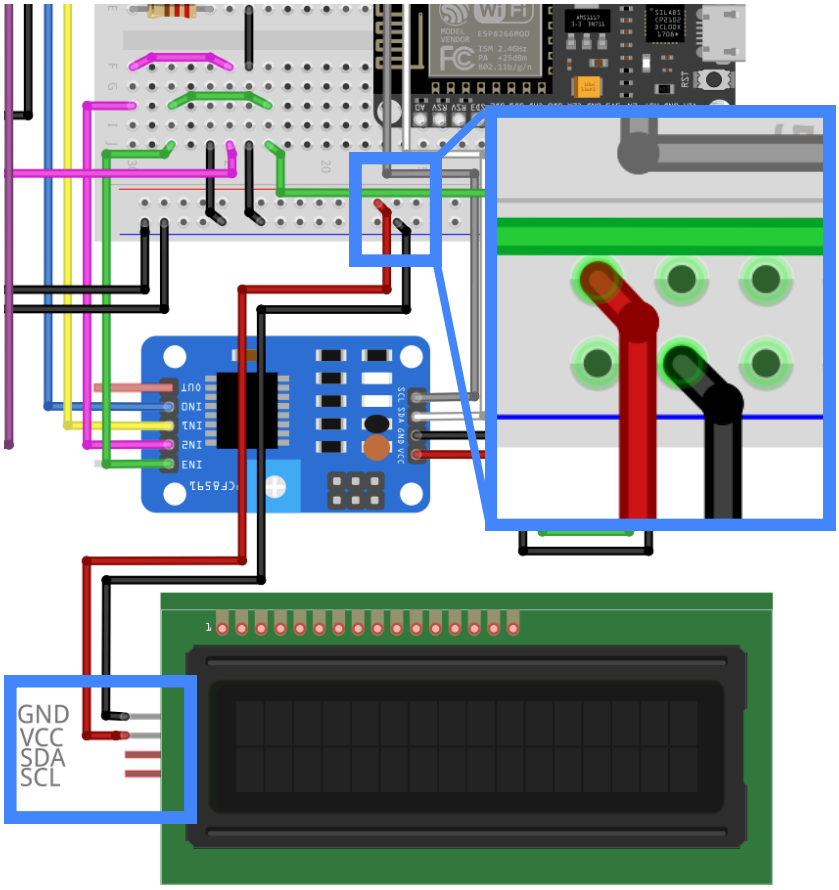


24. After placing the **USB-C Port** in the wall of the Controlbox, you can now connect the cable to the **luster terminal**, which is connected to the breadboard and Current Sensor. Make sure to connect the **black wire** from the USB-C connector to the terminal which is connected with the **blue-rail** and the **red wire** from the USB-C connector goes into the **ACS current sensor**.

25. For the next steps we need the **LCD Display**. Try to position it in the same direction as shown in the picture to get it to fit into the box later.
Do not put the LCD display in the box yet.



An LCD display, or Liquid Crystal Display can control which colors you see by adjusting the crystals in different ways. The model we use only shows one color.

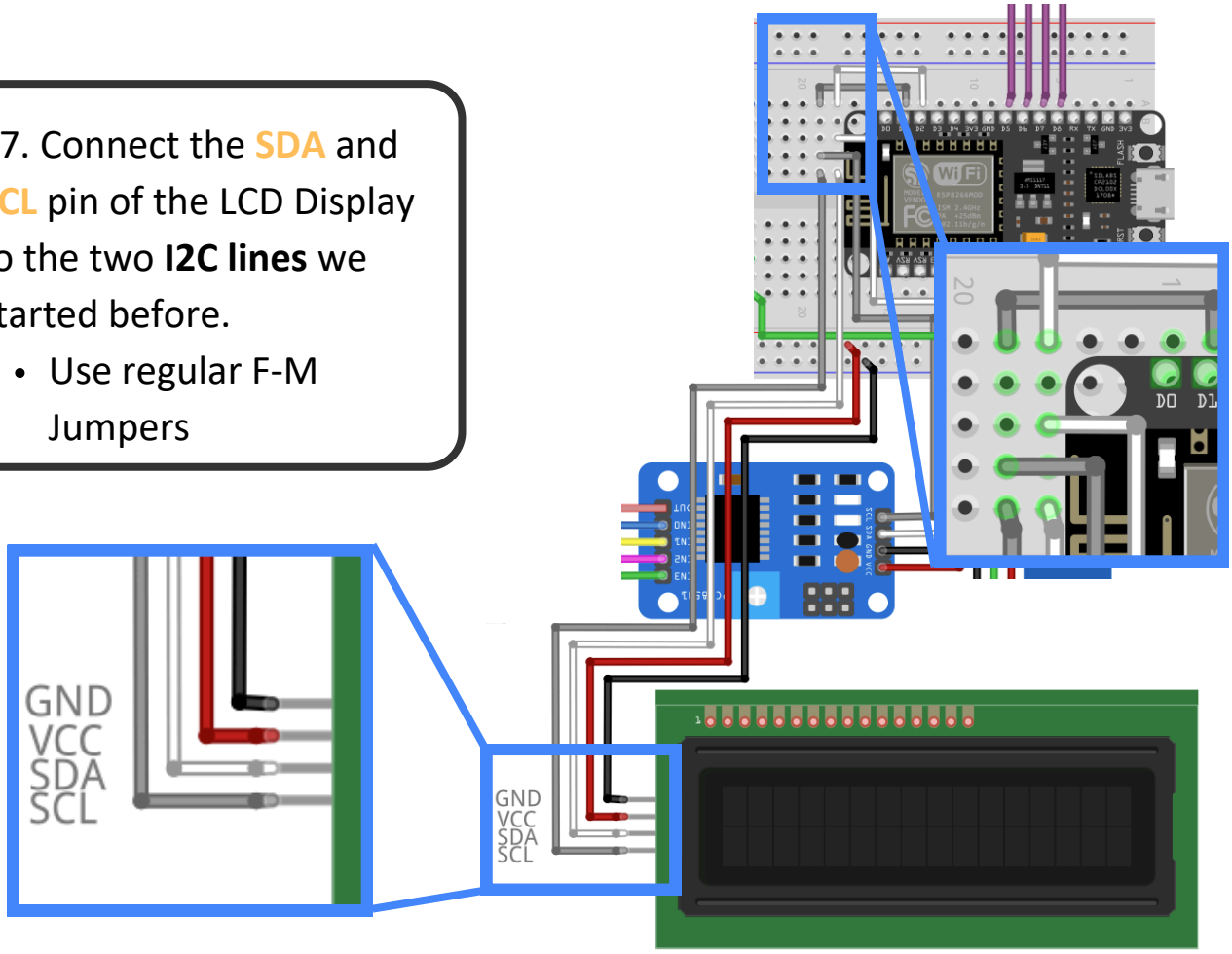


26. Connect the **GND** and **VCC** pins of the LCD Display to the **blue-rail** and **red-rail** of the breadboard.

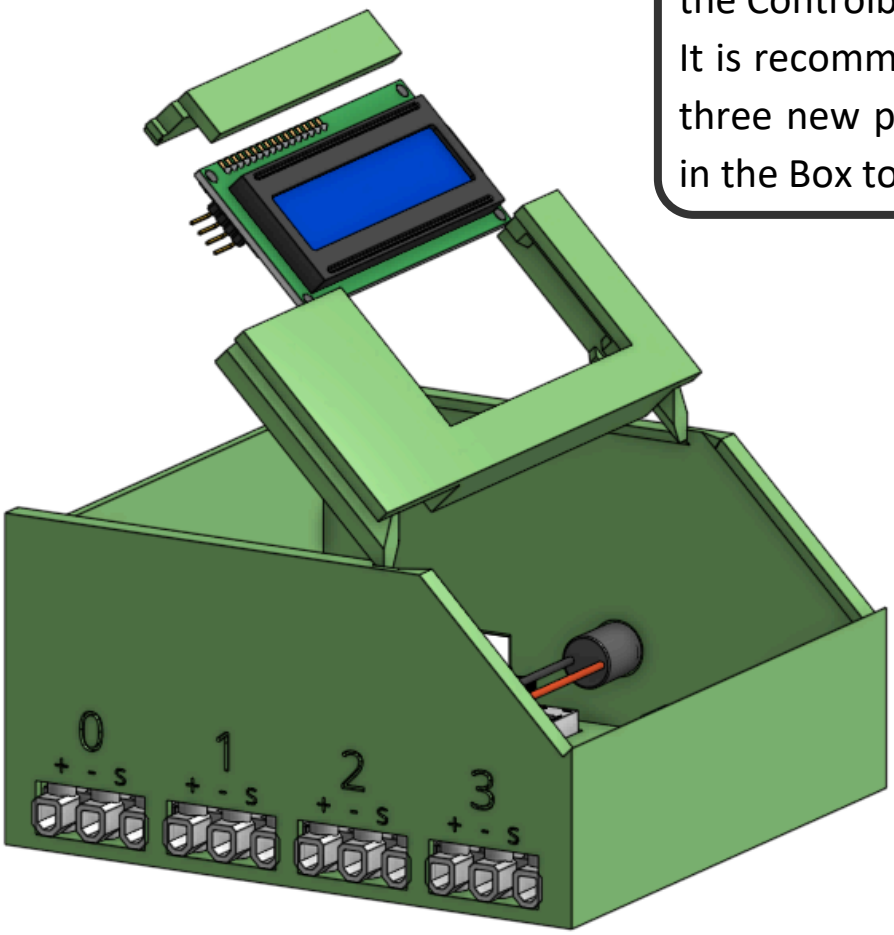
- Use regular F-M Jumpers

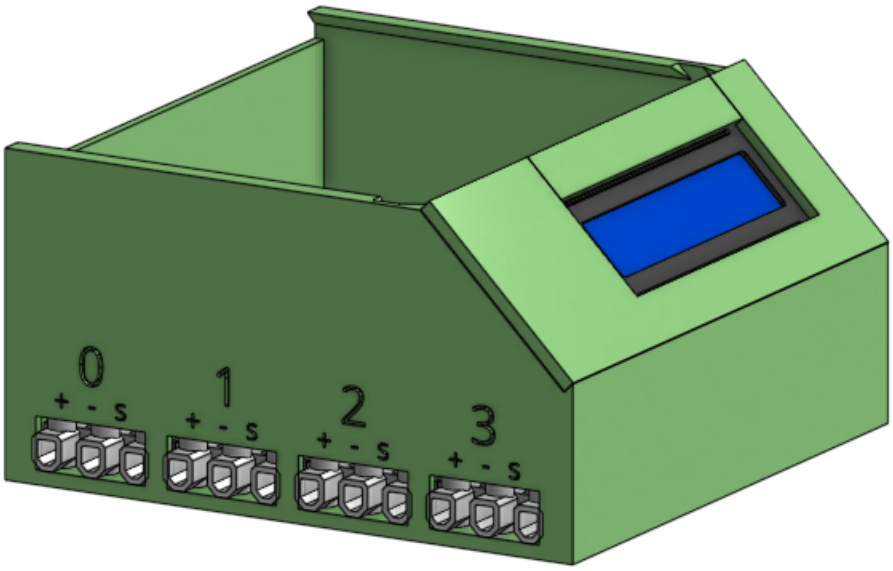
27. Connect the **SDA** and **SCL** pin of the LCD Display to the two **I2C lines** we started before.

- Use regular F-M Jumpers

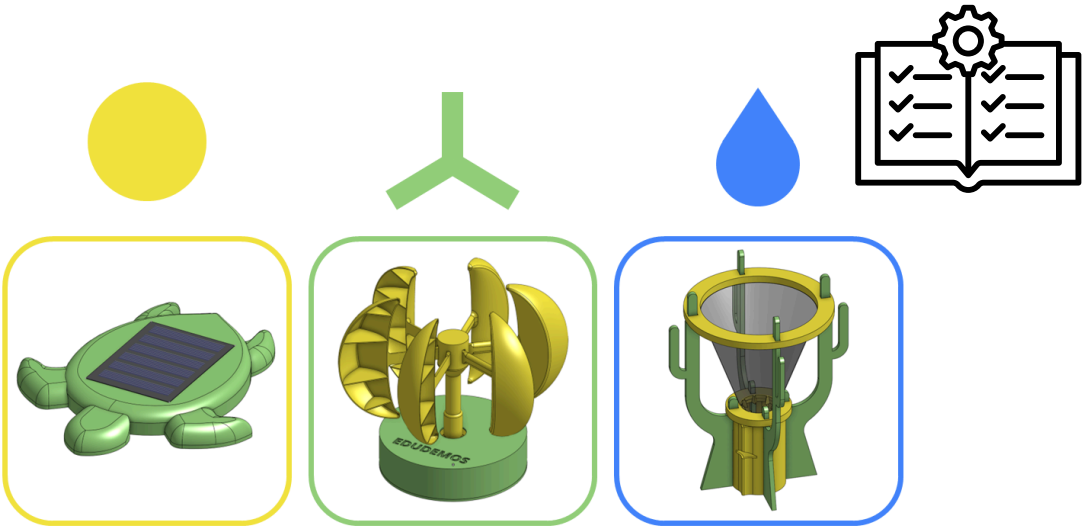


28. After connecting all cables to the LCD Display you can slide it into the Controlbox.
It is recommended to assemble the three new parts first and put them in the Box together.





29. Before closing the control box you need to build and connect the modules.
The individual guides can be found on the website https://edudemos.eu/en/demonstrators/. The instuctions also explain how to connect the modules to the Control Box. Below you can see the modules we have created for you.



Once you have finished one or all the modules you can come back to this guide and see how to install and programm this demonstrator using the Arduino IDE.

Installation Guide

What is the Arduino IDE?

The Arduino IDE (Integrated Development Enviroment) is the software used to write, compile and upload code to your board. (In this case the ESP8266.)

Downloading and installing the Arduino IDE:

Step 1: Go to the Arduino Website

Open your web broswer and visit the official Arduino website: https://www.arduino.cc/

Step 2: Navigate to the Download Page

From the homepage, click on "**Software**" in the top menu, then select "**Downloads**"

Step 3: Choose Your Operating System

On the Downloads page, select the appropriate version of the Arduino IDE for your operating system (Windows, macOS, or Linux).

Step 4: Download the Installer

Click on the download link, and if prompted, you can choose to contribute or simply download without contributing.

Step 5: Install the Arduino IDE

- For **Windows**:
  - Run the downloaded .exe file.
  - Follow the installation wizard to install the IDE.
  - Check the option to install the USB driver when prompted.
- For **macOS**:
  - Open the downloaded .dmg file.
  - Drag the Arduino IDE to the Applications folder.

- For **Linux**:
  - Extract the downloaded .tar.xz file.
  - Run the install.sh script in the terminal.

Step 6: Verify Installation

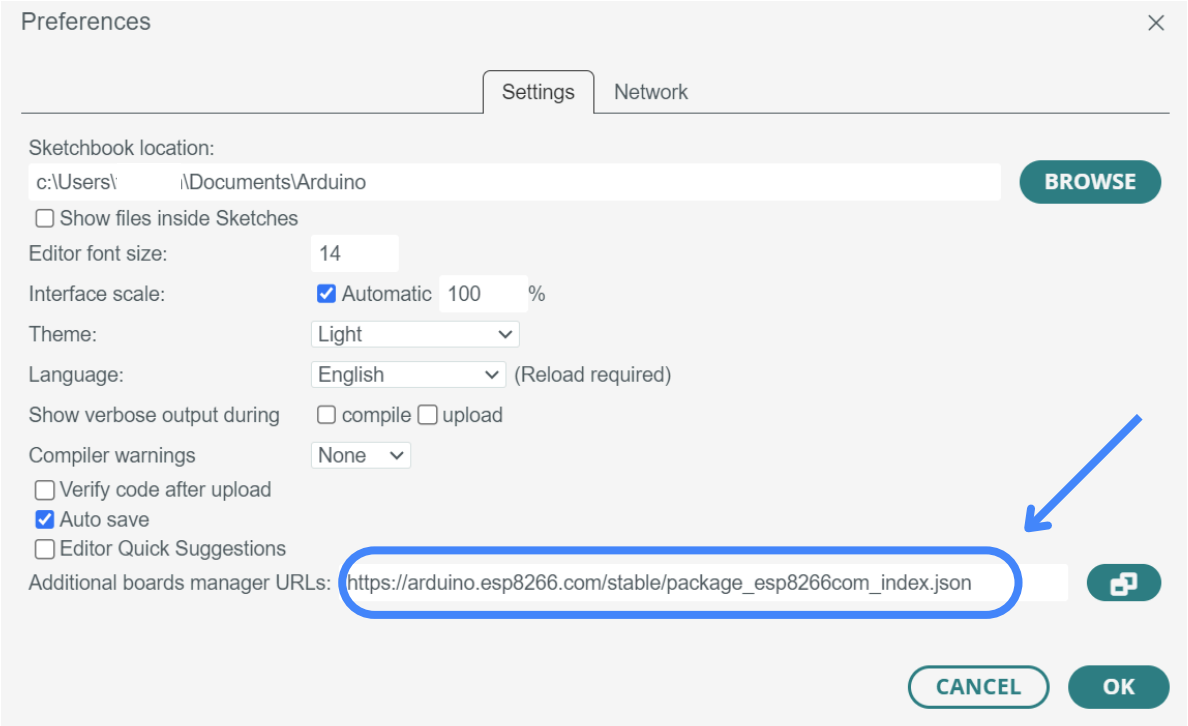
Open the Arduino IDE to ensure it launches correctly.

Cofiguring the Arduino IDE:

Step 1: Add the ESP8266 Board URL

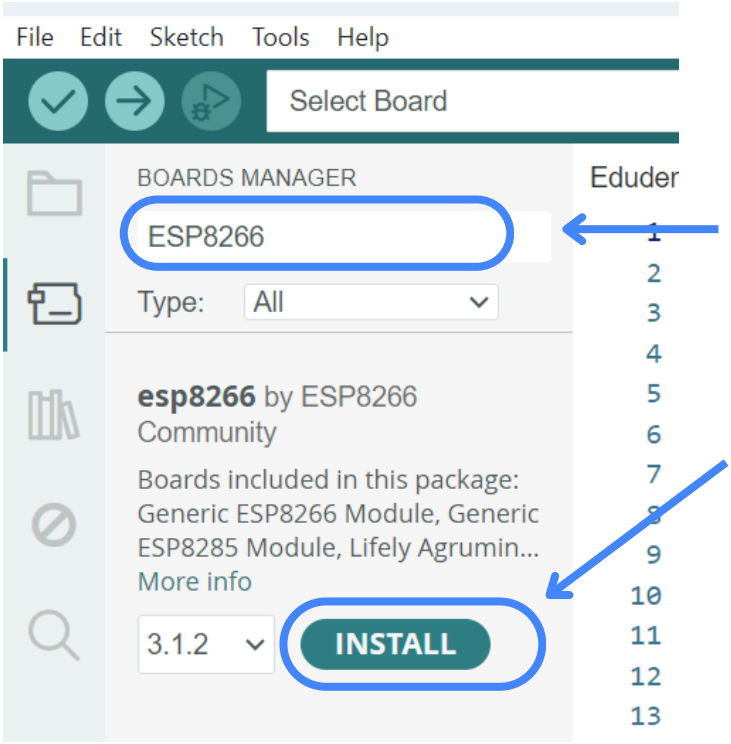
To use the ESP8266 with the Arduino IDE, you need to include the appropriate **URL** in the IDE's preferences:

1. Open the Arduino IDE.
2. Go to **File > Preferences** (on macOS, this might be under **Arduino > Preferences**).
3. In the **Additional Board Manager URLs** field, paste the following URL:
   http://arduino.esp8266.com/stable/package_esp8266com_index.json
4. If there are already other URLs listed, separate them with a comma.



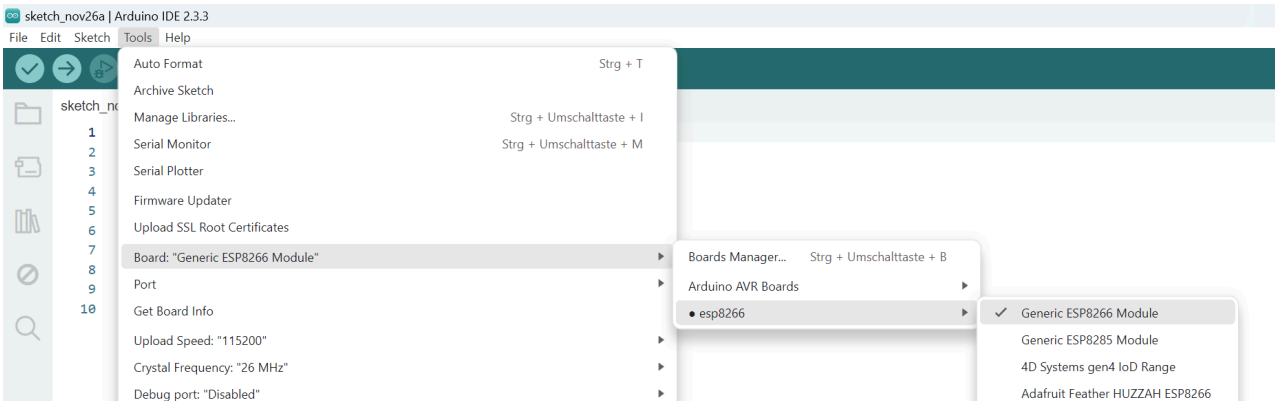
Step 2: Install the ESP8266 Board Package

1. Go to **Tools > Board > Boards Manager**.
2. In the search bar, type **ESP8266**.
3. Select **esp8266 by ESP8266 Community** and click **Install**.
4. Wait for the installation to complete.



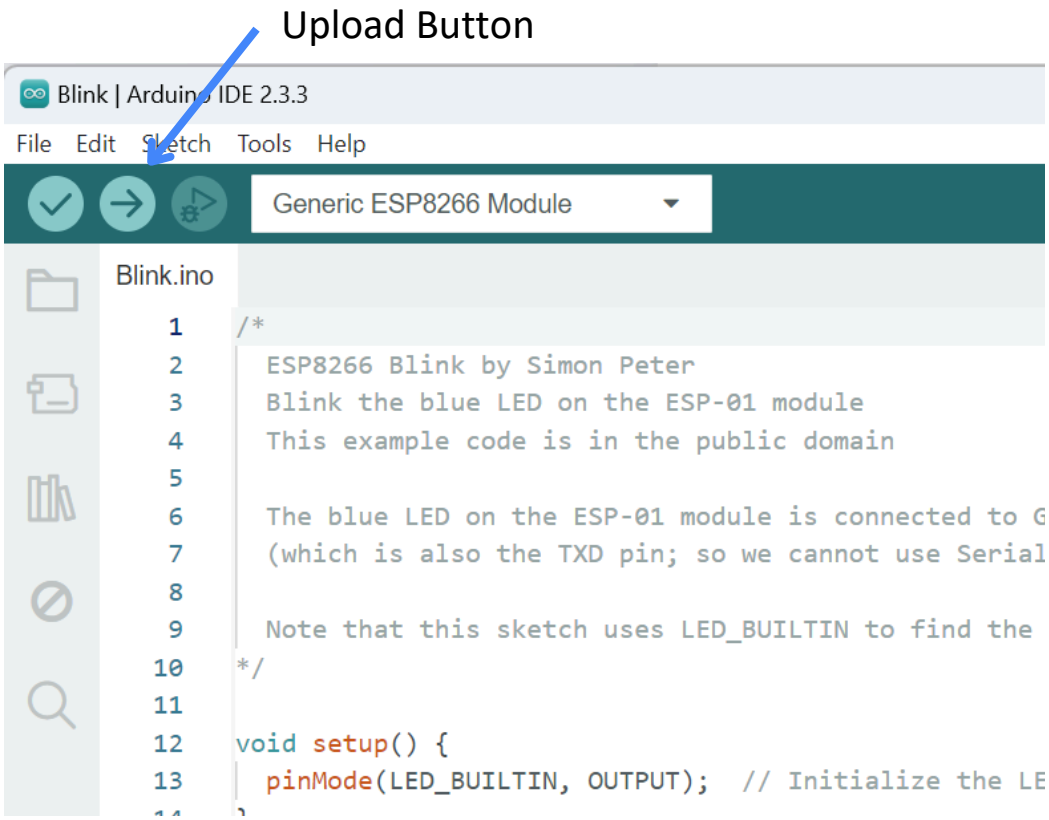
Step 3: Select the ESP8266 Board

1. Connect your **ESP8266** board to your computer using a USB cable.
2. Go to **Tools > Board** and select **Generic ESP8266 Module**.
3. Go to **Tools > Port** and select the correct COM port for your board.



Step 4: Test the Setup

1. Open the **Examples** menu in the Arduino IDE:
2. **File > Examples > ESP8266 > Blink**.
3. Verify and upload the sketch to your ESP8266.
4. The onboard LED on the ESP8266 should blink if everything is set up correctly.



Code

Download the Missing Library to Run the Code

Step 1: Download the Library:

Go to the EduDemoS website https://edudemos.eu/en/demonstrators/ and click on the **Programming** button. Dowload the ZIP folder called "**libraries.zip**".

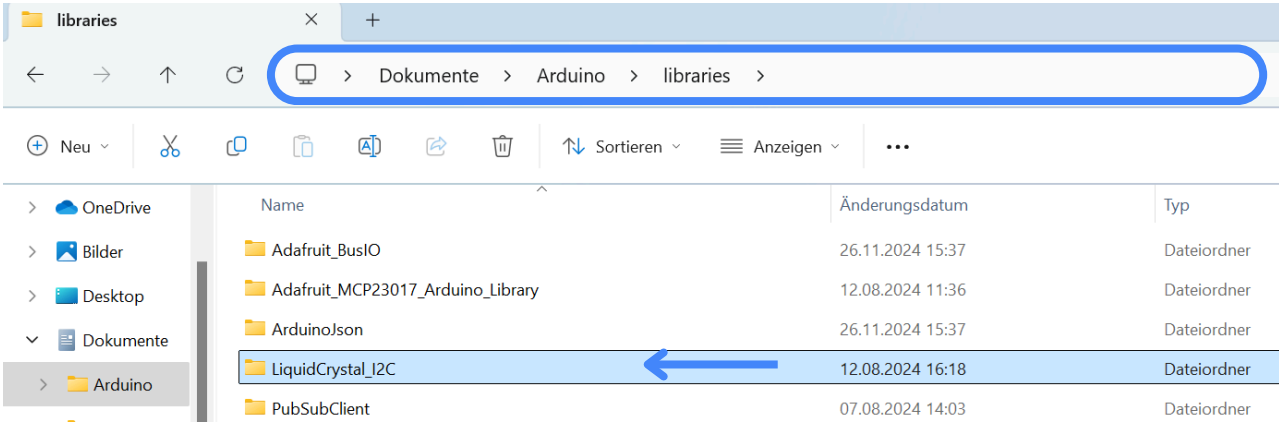
Step 2: Extract the ZIP File

Extract the downloaded ZIP file on your computer using a ZIP extractor such as WinRAR, 7-Zip, or the default extractor of your operating system.

Step 3: Move the Library to the Correct Folder:

- Navigate to the folder where your Arduino sketches are stored (e.g., *Documents/Arduino/libraries* on Windows or *~/Documents/Arduino/libraries* on macOS/Linux).
- Create a new folder in the *libraries* directory with the same name as the library (e.g., *LiquidCrystal_I2C*).
- Move the contents of the extracted archive into that folder. The files inside should have a path like

"*Documents/Arduino/libraries/LiquidCrystal_I2C.h*"



Once the library is in the correct folder, it will be available for use in your Arduino IDE.

Where to Find the Code

The code for the Turtle is available on the EduDemoS website under the **Programming** button.

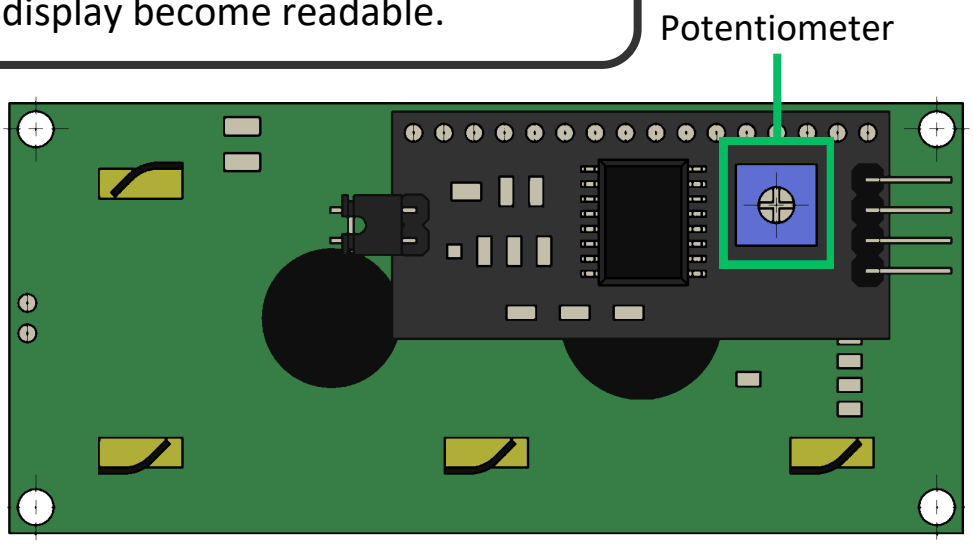
https://edudemos.eu/en/demonstrators/

Running the Code

Open the **Arduino IDE** and load the code into a new sketch. Finally you can upload the code to your ESP8266.

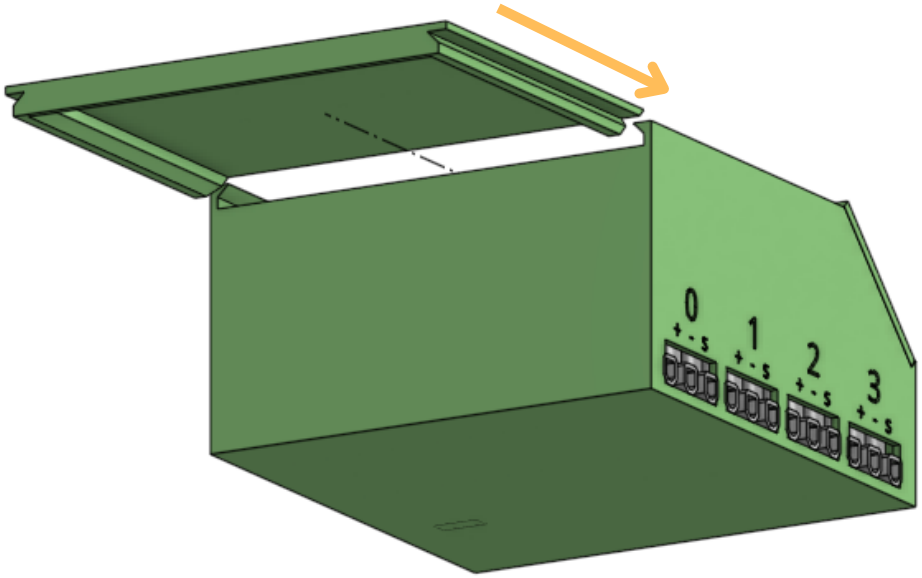
You should now see on the LCD Display how much current voltage is being generated by the solar panel.

30. If the display is unreadable because it is too bright or too dark, flip the display to its back and turn the screw in the potentiometer until the symbols on the display become readable.

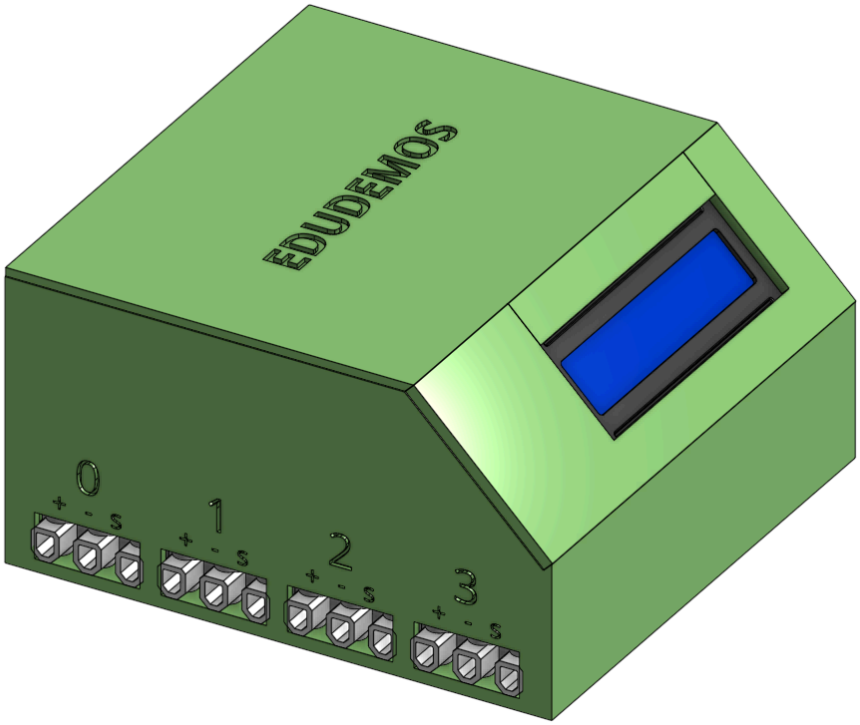


Backside of the LCD- Display

31. If you want to, you can remove the USB Cable connected to the ESP8266. The Controlbox just needs the USB-C cable plugged in to work.



32. Slide in the Lid to complete your Controlbox.



33. The finished control box with all its modules should look somewhat like this when finished.

What else can you imagine?

Personalize your own demonstrator,
and share your ideas and feedback
with us at
edudemos@technikmachtspass.org

Troubleshooting

General Issues

- Restart the Arduino IDE if changes don’t take effect.
- Open the **Serial Monitor** (**Tools > Serial Monitor**) to check for error messages.
- Double-check your wiring, especially the I2C connections and power.
- If nothing works, try disconnecting and reconnecting the ESP8266, then press the **RST** (**Reset**) button to restart the board.

Installation Issues

Arduino IDE Won't Install:

- Ensure your computer meets the minimum requirements for the Arduino IDE.
- Download the correct version for your operating system from https://www.arduino.cc/en/software.
- On Windows, run the installer as an administrator.

Code Upload Errors

- Use a good-quality USB cable (some cables are for power only, not data).

Common Upload Error Messages:

- *Failed to connect to ESP8266*:
  - Hold the **BOOT/FLASH button** on the ESP8266 while uploading the code.
  - After the upload is complete, press the RST (Reset) button to reboot the board and run the code.
- *Serial port not found:*
  - Check the USB connection and ensure the right port is selected in **Tools > Port**.

It could also be a Driver Issue (read next section).

USB Driver Issues

If your ESP8266 or its port is not recognized by your computer, you may need to install the appropriate USB-to-Serial driver. ESP8266 boards use the CP210x chip:

Download the Driver

- Go to the official Silicon Labs website:
  - https://www.silabs.com/developer-tools/usb-to-uart-bridge-vcp-drivers?tab=downloads
- Download the appropriate version for your operating system (Windows Universal Driver for Windows 10 an 11)

Still Not Detected?

Try a different USB cable (some cables are power-only and don't support data).

Ensure no other devices are interfering with the USB ports.

Restart your computer after installing the driver.

I2C Display Not Working

No Display Output:

- Verify **SDA** and **SCL** pins are correctly connected:
  - Default ESP8266 pins: **SDA = GPIO4** (**D2**), **SCL = GPIO5** (**D1**).
- Ensure the I2C address in the code matches the display's address. Use an I2C scanner sketch to confirm the address.
- If you can't see anything or you have trouble reading the Display, make sure that you put everything properly together. If everything is correctly assembled, try adjusting the potentiometer on the back of the Display.

Display Flickers or Shows Garbage Data:

- Use pull-up resistors (4.7kΩ) on SDA and SCL lines if not already present.
- Ensure the power supply is stable (ESP8266 and display both get sufficient power).

Appendix

Components with Example Link: Control Box

Quantity	Name	Specification	Size	Example Link	Last Update
1	ESP8266	NodeMCU Board Amica V2	48x26x 13mm	https://amzn.eu/d/gnZ5Kst	28.11.2024
1	LCD Display	I2C 16x2 (pre-soldered)		https://amzn.eu/d/0KgODaI	28.11.2024
1	PCF8591	AD/DA Konverter, I2C, 8Bit		https://www.roboter-bausatz.de/p/pcf8591-ad-da-konverter-analog-zu-digital-wandler	28.11.2024
1	ACS712	5A		https://www.reichelt.de/de/de/entwicklerboards-stromsensor-bis-5-a-acs712elc-05b-debo2-sen-strom-p282585.html?nbc=1&&r=1	28.11.2024
1	Bread-board	Mini	400 Pin	https://amzn.eu/d/gEZwM3o	28.11.2024
16	Jumper Wire	F-M	20cm	https://amzn.eu/d/cPTe2WY	28.11.2024
12	Jumper Wire	M-M	20cm	https://amzn.eu/d/0WQP3MI	28.11.2024
16	Jumper Wire	M-M	10cm	https://amzn.eu/d/fAQLBr9	29.11.2024
13	Luster Terminal			https://amzn.eu/d/jf2sQKD	28.11.2024
1	USB-C Port	Female, 5V		https://amzn.eu/d/bjhGCws	28.11.2024
4	Resistor	1kOhm		https://amzn.eu/d/39sEK1f	29.11.2024

Components with Example Link: Sun Module

Quantity	Name	Specification	Size	Example Link	Last Update
1	Bearing	608zz	8x22x7 mm	https://amzn.eu/d/2ETHdLh	29.11.2024
1	DC Motor	1.5V - 6V		https://amzn.eu/d/8LkNNqp	29.11.2024
2	Wire	At least one male end	>20cm	https://amzn.eu/d/3GGCBPu	29.11.2024

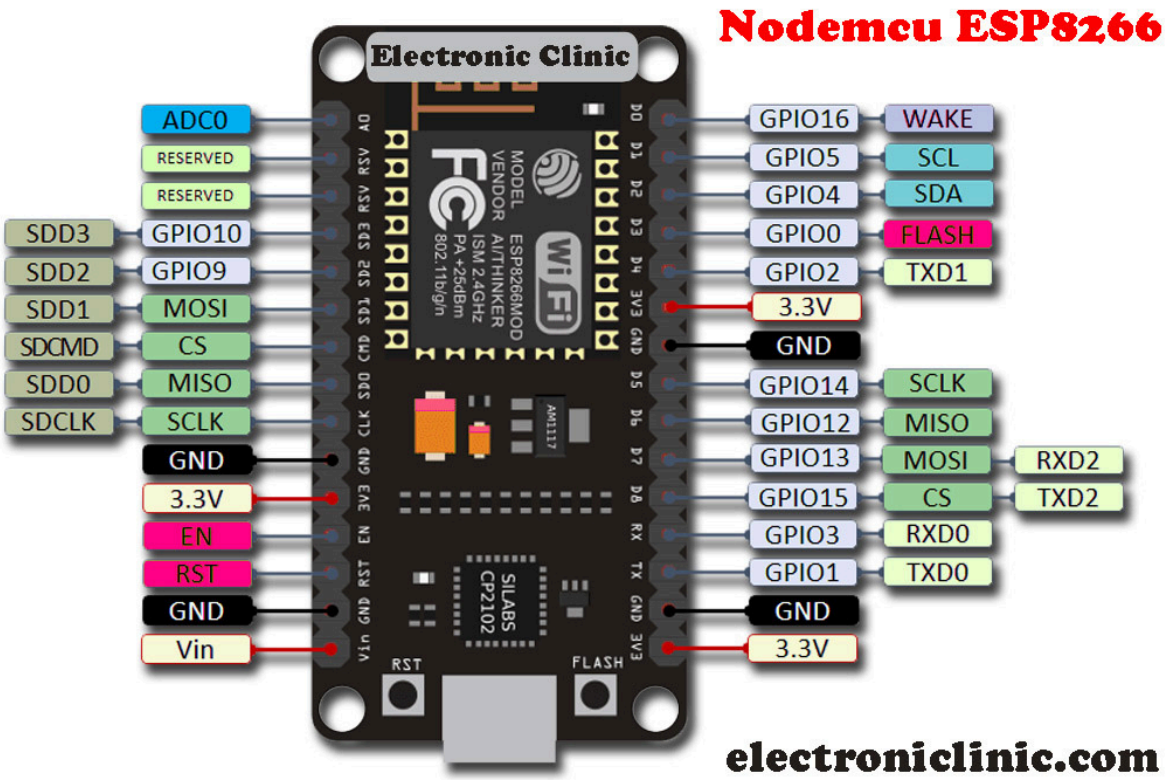
Components with Example Link: Water Module

Quantity	Name	Specification	Size	Example Link	Last Update
1	Water Level Sensor	5V	50x16 mm	https://amzn.eu/d/1Ewnpv9	29.11.2024
3	Jumper Wire		40cm	https://amzn.eu/d/3GGCBPu	29.11.2024

Components with Example Link: Sun Module

Quantity	Name	Specification	Size	Example Link	Last Update
1	Solar Panel	3V, 0.3W	65x48 mm	https://amzn.eu/d/hDaVRu9	29.11.2024

Pinout ESP8266

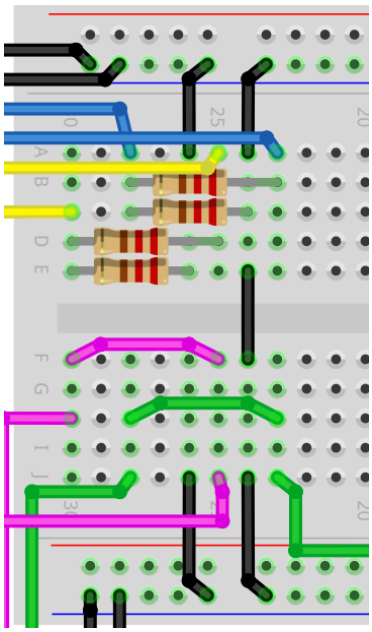


Nodemcu ESP8266 Pinout, Features, and specifications (electroniclinic.com)

Changing Modules

Because of the modular design, you can change the modules or connect your own.

Wiring Changes



- Keep in mind, that the PCF8591 has a range from 0-5V, so everything above that voltage will damage the board.
- To achieve a broader possible voltage range, you can use a voltage divider, similar to how it is done for channels 0 and 1. For doubling the range you need two of the same resistors, we suggest 1kOhm.
- If you stay within the 0-5V range you can connect the two lines with a jumper, as used in channel 2 and 3.
- If you want to change anything on channel 3 you have to disconnect the **ACS712** and connect its **S** luster terminal to **J23**.

Code Changes

- To implement the changes in the code you will need to adjust the Channel settings in the lines 63 to 86 of the code.
- If you just swapped modules you can copy all the standard settings from one Channel to another (except the GPIO_POWER_PIN).
- If you use your own module you might need to calculate your own Slope and Offset. It might be helpful to adjust the update rate as well (standard is 2).
- The formula used to calculate the output value is [output value]= [input voltage]*[slope]+[offset].

Licensing

All Circuits were created with Fritzing.

Some images are based on modules from GrabCAD Library.

Co-funded by the European Union

Gerda Stetter Stiftung
Technik *macht* Spaß!

EDUDEMOS

EDUcating through Sustainable **DEMO**nstrators

Get In Touch With Us

edudemos@technikmachtspass.org

Gerda Stetter Stiftung Technik macht Spaß! · Finnova · gbs sg.ch · FUNDACIÓN SERGIO ALONSO · I.E.S. El Rincón