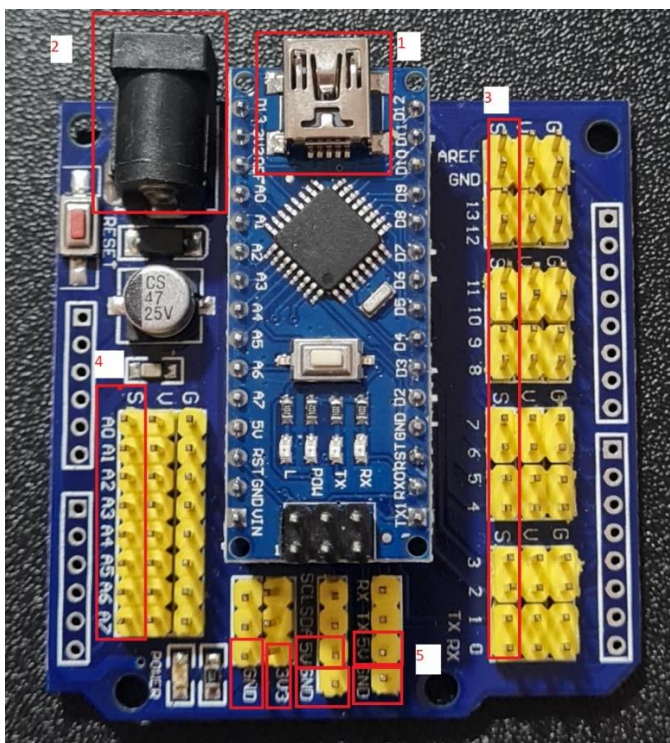


Getting started...

In this workshop we are going to cover the Arduino Basics as well as some basics for C++ so that we can write our first own program for the Arduino. We are going to use the Arduino Nano and the Arduino IDE 2.1.0 which is the latest version. We'll also have a look at some of the different gadgets/sensors that we can connect to it.

Basics/Arduino Nano:



To give you an overview of the Arduino and explain what all the different connectors are, in the picture above the different parts are labelled from 1- 5.

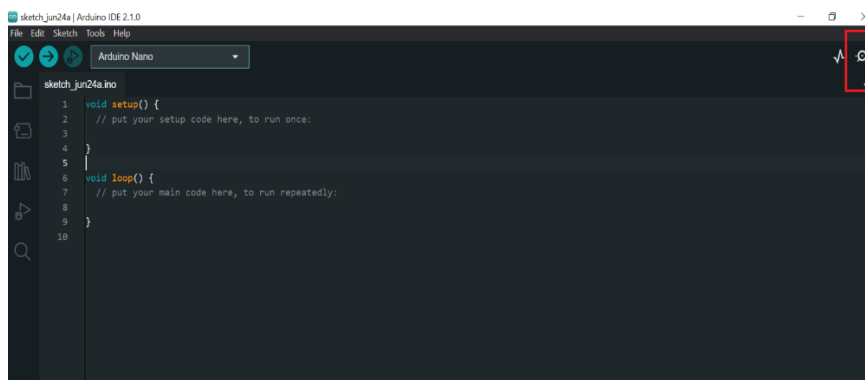
1. This is the **USB port** which is used to connect the Arduino to your computer and download any programs you may have written.
2. The Arduino can be **powered up** by connecting it to a computer via USB, however there is also the option to connect it to a battery packet via this socket. Please note, while the cable used for this is a USB cable, it is not a data transfer cable, so you cannot transfer data onto the Arduino by using this socket. Please use the USB port (1) to transfer data.

3. You may have noticed different pins (**S**, **V**, **G**) as well as numbers. **S = Signal**, **V = Volt**, **G = Ground**. These are **digital pins**. For now, we are only going to use the column with “S” pins from 13 to 0. These are signal pins to which we can connect the different sensors that we are going to use later. If we connect a sensor to a pin, this basically tells the Arduino were to get the information from the sensor from. For now, you can ignore “AREF” and “G” at the top.
4. There is also the option to use **analog pins**, from A0 to A7. Most sensors can be used with the digital pins, however there are some (for example the joystick), that require analogue pins. As with (3), you can ignore “V” and “G” here as we are only going to use “S” (signal).
5. **This is the most important part**, so please take a moment to look at your Arduino and familiarize yourself with it. If we get it wrong here, we can break the Arduino, so it is important that you pay close attention. If we connect a sensor or a gadget to the Arduino, those need power as well. The gadgets can be powered up through the Arduino by connecting them to a **3V (3 Volt) or 5V (5 Volt) pin**. There are some “GND” pins, which stands for “Ground” and basically prevents our gadget from frying from a **too high voltage**. **Always make sure you have the gadget/sensor connected to “GND” if you are connecting it to**. The labels on the back of the gadget/sensor will tell you which cable needs to be connected to 3V and which one needs to be connected to Ground. It is important to always double check that you don’t accidentally swap 3V and Ground. 3V typically uses a **red cable** and Ground typically has a **black or grey cable**.

Arduino IDE

“IDE” stands for **Integrated Development Environment**, which is what we will be working with to write the code for our programs. You might be already familiar with a different IDE if you have worked with Python before.

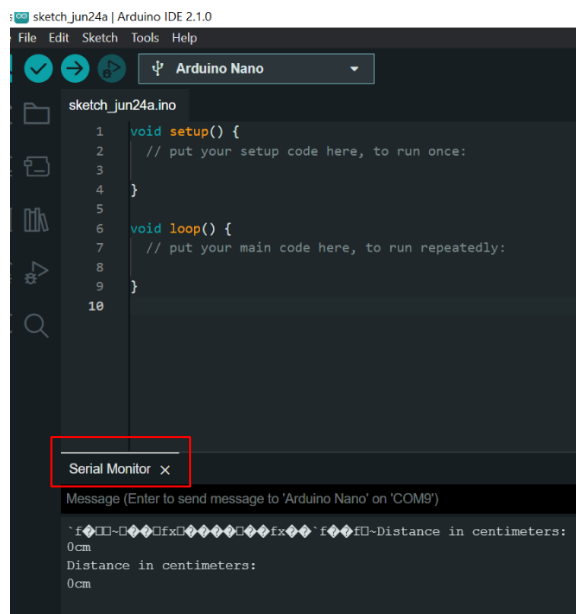
First, open the Arduino IDE on your computer. You should see this layout:



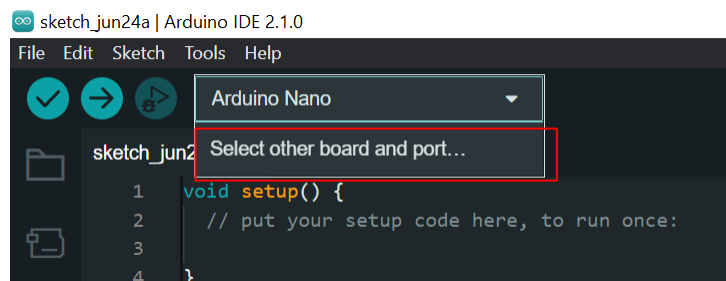
It already has a function for the setup as well as the main function included. In the **set up function** we can place any code that we only want to run once. In the **main function**, the `void loop()`, we place any code that we want to run repeatedly. Every function definition is followed by **curly brackets { }**. Within those brackets, we place the code for the function.

For now, we are just going to leave it as it is.

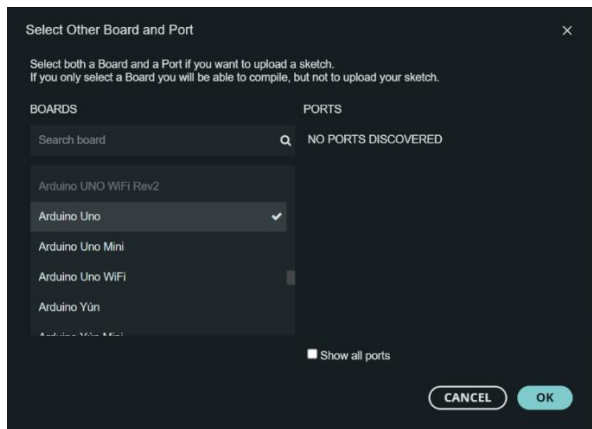
In the top right corner (circled in red) you can see the **Serial Monitor** where you can see any output produced by your program, as well as error codes if there are any errors in your program.



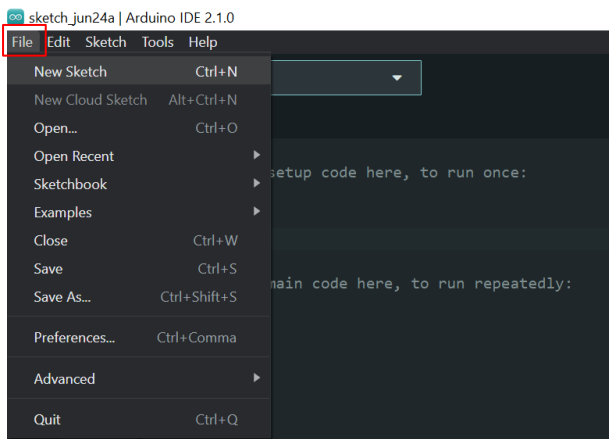
Under “**Serial Monitor**” at the bottom, you can see the output of the program loaded onto the Arduino. In this example, it is related to a sensor measuring a distance in centimeters.



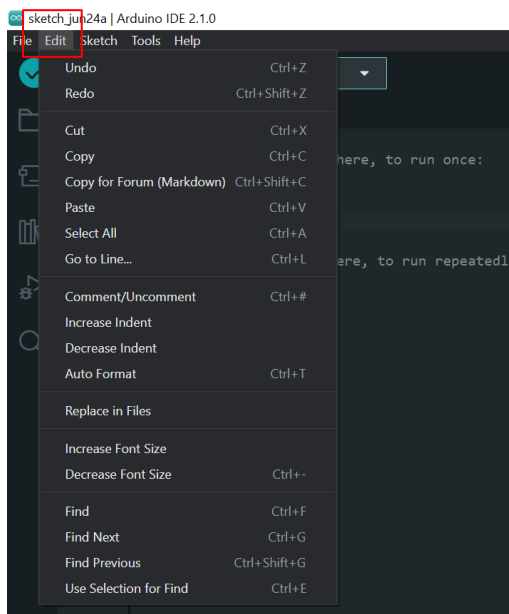
If your IDE does not show a selected Arduino board yet, click on the drop down menu and select “**Select other board and port...**”.



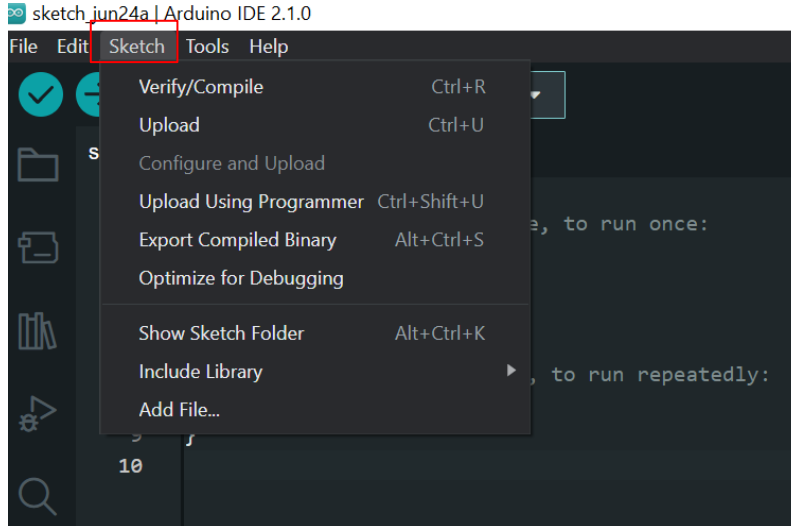
Next, search for **“Arduino Nano”** on the list and then select the USB port your Arduino is connected to on your computer. This should show up automatically as being named **“COM XX”** with XX being a number. Don’t forget to click on **“Ok”** to confirm.



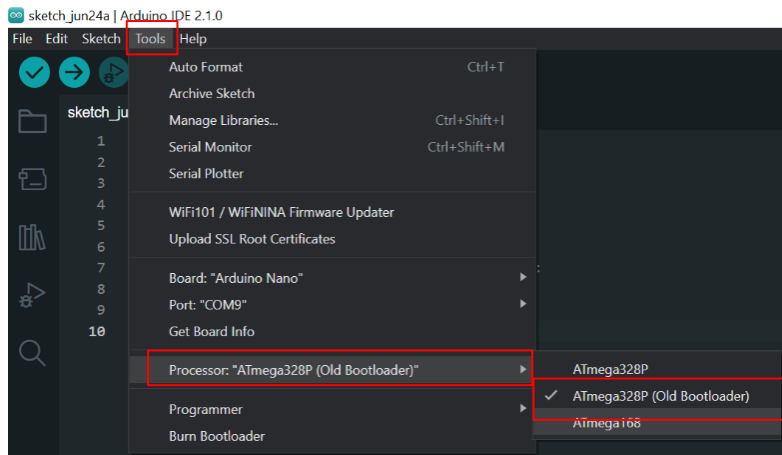
If you click on **“File”** and **“New Sketch”** you can create a new file. You can save it under **“File”** and **“Save As”** but make sure you give it a name so that you can find your file again later. Once you save it, it automatically creates a folder with the name that contains the file.



Under **“Edit”** you can make a number of different changes, such as **“Undo”** if you want to undo any changes previously made. Here you can also copy and paste or change indentation.



If you click on “**Sketch**”, you can upload your program to the Arduino. It is also where you can add a library (We will get to this later in the workshop)



Under “**Tools**” you have the option to select a different Arduino or change the USB port it is connected to. Make sure that the **Processor** is set to “**ATmega328P (Old Bootloader)**”.

Write a “Hello World” Program

Now we are ready to write our own first program and upload it to the Arduino.

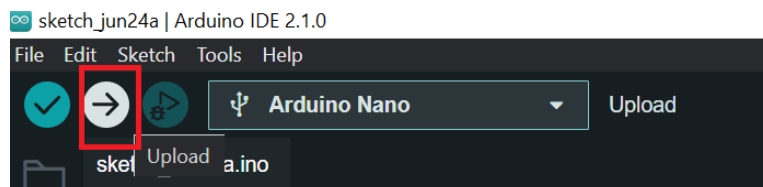
In our setup function, we always have to add a starting line **Serial.begin(9600);** before adding any other code, which sets the speed parameter to 9600, which means that the Arduino exchanges messages with the Serial Monitor at a data rate of 9600 bits per second. Without setting this up, the Serial Monitor wouldn't be able to show anything.

To print a text to the Serial Monitor, we write **Serial.println(“Hello,World”)** where Serial.println tells the program to print something. In brackets we place whatever text we want the program to print. Make sure that you place the text in between “ ” and in brackets with a semicolon at the end.

```

sketch_jun24a | Arduino IDE 2.1.0
File Edit Sketch Tools Help
[Checkmark] [Arrow] [Upload] Arduino Nano
sketch_jun24a.ino
1 void setup() {
2   // put your setup code here, to run once:
3   Serial.begin(9600);
4   Serial.println("Hello World!");
5 }
6
7 void loop() {
8   // put your main code here, to run repeatedly:
9 }
10
  
```

Then click “**Upload**” and check the output on the Serial Monitor. What do you see?



What happens when you change the text from “Hello World” to “Hello (INSERT YOUR NAME)”?

What happens when you remove the semicolon at the end of the line?