

# BEGINNING WITH PROGRAMMING IN ARDUINO

**27 October 2024**

ARCHIVED BY  
**b33tr00t-444**

## Contents

<b>1</b>	<b>The Basics</b>	<b>1</b>
<b>2</b>	<b>Writing my first program</b>	<b>4</b>
<b>3</b>	<b>Programs</b>	<b>6</b>

## List of Figures

1	Listing the connected boards . . . . .	1
2	Searching my board . . . . .	2
3	Installing the required platform . . . . .	2
4	Listing all the boards . . . . .	3

## Listings

1	Arduino code format . . . . .	1
2	Viewing the list of connected boards . . . . .	1
3	Search for my board . . . . .	1
4	Installing the platform . . . . .	2
5	Listing all the boards . . . . .	2
6	First program . . . . .	4
7	Compiling an arduino program . . . . .	4
8	Uploading the sketch . . . . .	4
9	Blinking LED . . . . .	5

# 1 The Basics

- ◇ Format of code in for Arduino:

Listing 1: Arduino code format

```
void setup() {  
}  
  
void loop() {  
}
```

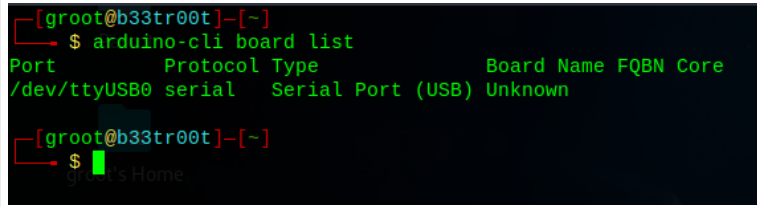
Commands put inside of `setup()` are executed once. This function is used to 'setup' the arduino board. Commands that we want to run over and over again are put inside of the `loop()` function.

- ◇ Now, I am using [Arduino CLI](#) here. To see the list of connected boards use the following command:

Listing 2: Viewing the list of connected boards

```
$ arduino-cli board list
```

This will display a list of connected boards with their name, FQBN(Fully Qualified Board Name), port, etc. In my case, however, it's showing the following:



```
[groot@b33tr00t]~  
$ arduino-cli board list  
Port          Protocol Type      Board Name FQBN Core  
/dev/ttyUSB0  serial            Serial Port (USB) Unknown  
  
[groot@b33tr00t]~  
$
```

Figure 1: Listing the connected boards

Here, it doesn't show my board's name, instead, it shows 'Unknown'. I know that my board is an Arduino Uno, so I will do a `core search` for my board by using the following command:

Listing 3: Search for my board

```
$ arduino-cli core search uno
```

This will generate the following output:

```
[groot@b33tr00t]-[~]  
$ arduino-cli core search uno  
Downloading index: package_index.tar.bz2 downloaded  
ID          Version Name  
arduino:avr 1.8.6  Arduino AVR Boards  
arduino:megaavr 1.8.8  Arduino megaAVR Boards  
arduino:renesas_uno 1.2.2  Arduino UNO R4 Boards  
[groot@b33tr00t]-[~]  
$
```

Figure 2: Searching my board

Those names mentioned below the ID column are the names of the platform needed for the board, the correct platform needs to be installed to be able to use the board. Based on what I have found (by googling), I need to install the `arduino:avr` platform for my UNO.

- ◇ To install the required platform, just use the following command:

Listing 4: Installing the platform

```
$ arduino-cli core install arduino:avr
```

```
[groot@b33tr00t]-[~]  
$ arduino-cli core install arduino:avr  
Platform arduino:avr@1.8.6 already installed  
[groot@b33tr00t]-[~]  
$
```

Figure 3: Installing the required platform

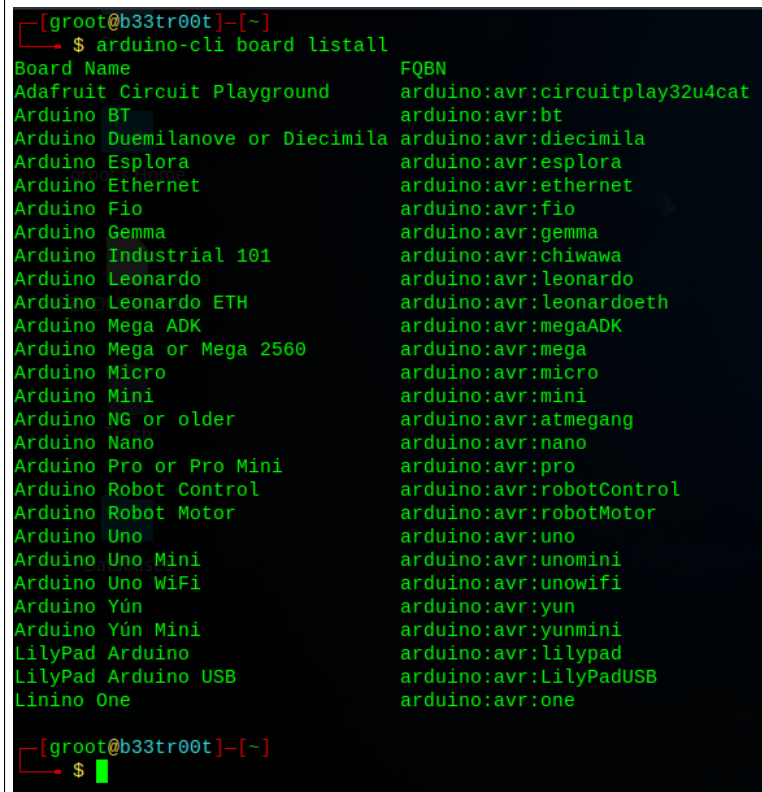
In my case, it has already been installed for my board (because I had already installed it earlier and this was done for illustration purposes).

- ◇ Once it's installed, we need to know the FQBN for our board and for that we need to list all the boards but this time it's a different command:

Listing 5: Listing all the boards

```
$ arduino-cli board listall
```

This produces the following:



```
[groot@b33tr00t]~  
$ arduino-cli board listall  
Board Name                                FQBN  
Adafruit Circuit Playground               arduino:avr:circuitplay32u4cat  
Arduino BT                               arduino:avr:bt  
Arduino Duemilanove or Diecimila         arduino:avr:diecimila  
Arduino Esplora                           arduino:avr:esplora  
Arduino Ethernet                         arduino:avr:ethernet  
Arduino Fio                              arduino:avr:fio  
Arduino Gemma                            arduino:avr:gemma  
Arduino Industrial 101                    arduino:avr:chiwawa  
Arduino Leonardo                         arduino:avr:leonardo  
Arduino Leonardo ETH                     arduino:avr:leonardoeth  
Arduino Mega ADK                         arduino:avr:megaADK  
Arduino Mega or Mega 2560                arduino:avr:mega  
Arduino Micro                            arduino:avr:micro  
Arduino Mini                             arduino:avr:mini  
Arduino NG or older                      arduino:avr:atmegang  
Arduino Nano                             arduino:avr:nano  
Arduino Pro or Pro Mini                   arduino:avr:pro  
Arduino Robot Control                     arduino:avr:robotControl  
Arduino Robot Motor                      arduino:avr:robotMotor  
Arduino Uno                              arduino:avr:uno  
Arduino Uno Mini                         arduino:avr:unomini  
Arduino Uno WiFi                        arduino:avr:unowifi  
Arduino Yún                              arduino:avr:yun  
Arduino Yún Mini                         arduino:avr:yunmini  
LilyPad Arduino                          arduino:avr:lilypad  
LilyPad Arduino USB                      arduino:avr:LilyPadUSB  
Linino One                               arduino:avr:one  
  
[groot@b33tr00t]~  
$
```

Figure 4: Listing all the boards

As seen from the image, the FQBN for Arduino UNO is mentioned alongside with its Board Name. Now, I can use this FQBN for compiling and uploading my sketch to my board.

## 2 Writing my first program

- ◇ The pin 13 of the arduino is connected to a LED (labelled as L on the board).
- ◇ My first program in Arduino:

Listing 6: First program

```
void setup() {
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);
}
```

- ◇ To compile this program in arduino CLI, we compile it using the following command:

Listing 7: Compiling an arduino program

```
$ arduino-cli compile -b arduino:avr:uno /home/groot/
arduino/programs/first/first1/first1.ino
```

- ◇ After compiling the sketch, we need to upload it using the following command:

Listing 8: Uploading the sketch

```
$ arduino-cli upload -p /dev/ttyUSB0 -b arduino:avr:uno /
home/groot/arduino/programs/first/first1/first1.ino
```

- ◇ After uploading it to the board, the LED will be turned on.
- ◇ Basically in [Listing 6](#) in the function `pinMode()` I have added two parameters, the first is the *pin number* and the second is whether that pin should be treated as an *input* or an *output*. Since I am going to turn it on/off (it's going to output its state), so it's set to `OUTPUT`. Next, in the function `loop()`, I have put the line `digitalWrite(13, HIGH);` which basically means that it's going turn the LED (13) on (`HIGH`) and this command is going to repeat over and over again because that's what `loop()` does.



- ◇ Here's a modified version of [Listing 6](#):

Listing 9: Blinking LED

```
void setup() {
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}
```

- ◇ I have added a another `digitalWrite()` that turns the LED (13) off (LOW).
- ◇ The `delay(1000)` function pauses the program for a 1000 milliseconds (that 1 second).

### 3 Programs

- ◇ <https://github.com/b33tr00t-444/arduino/tree/arduinoFun/programs/first/first1>
- ◇ <https://github.com/b33tr00t-444/arduino/tree/arduinoFun/programs/first/first2>
- ◇ <https://github.com/b33tr00t-444/arduino/tree/arduinoFun/programs/first/first3>
- ◇ <https://github.com/b33tr00t-444/arduino/tree/arduinoFun/programs/first/assignment>