
Arduino Tutorial

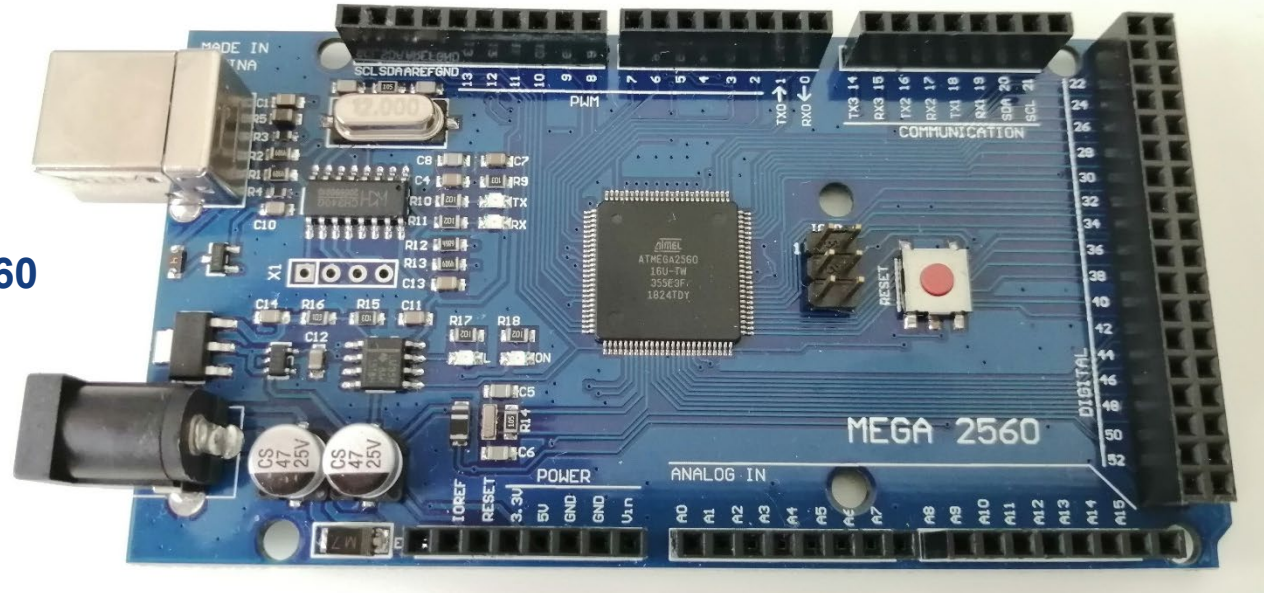
AUTOPISTA

Arduino MEGA 2560



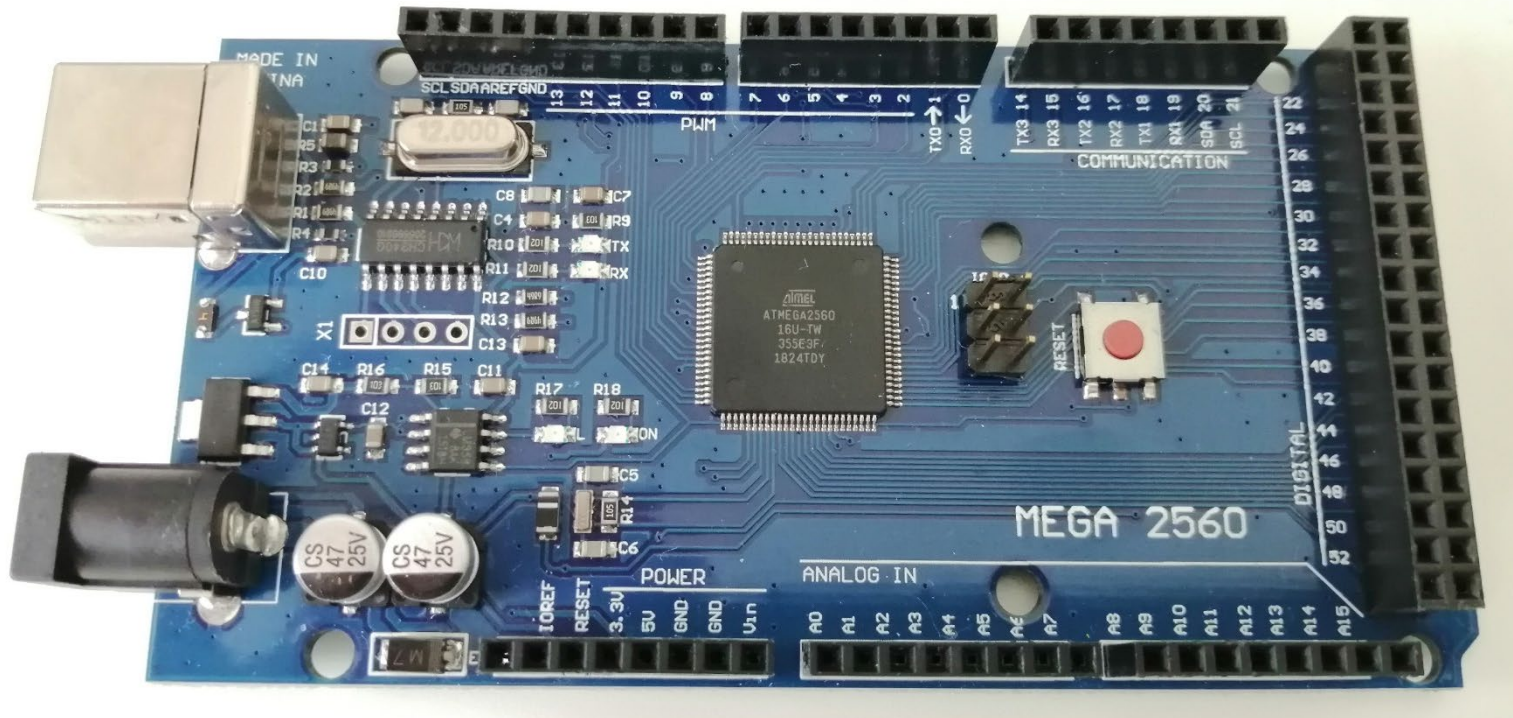
Characteristics

- Processor Atmega2560
- 8-bit CPU
- 16 MHz
- 8 kB SRAM
- 256 kB
- 54 digital pins
- 16 analogic pins
- Size: 101,52 mm x 53,3 mm



Arduino MEGA 2560

The MEGA 2560 controller is a micro-controller based on ATmega2560. The Mega 2560 is compatible with many Arduino systems.



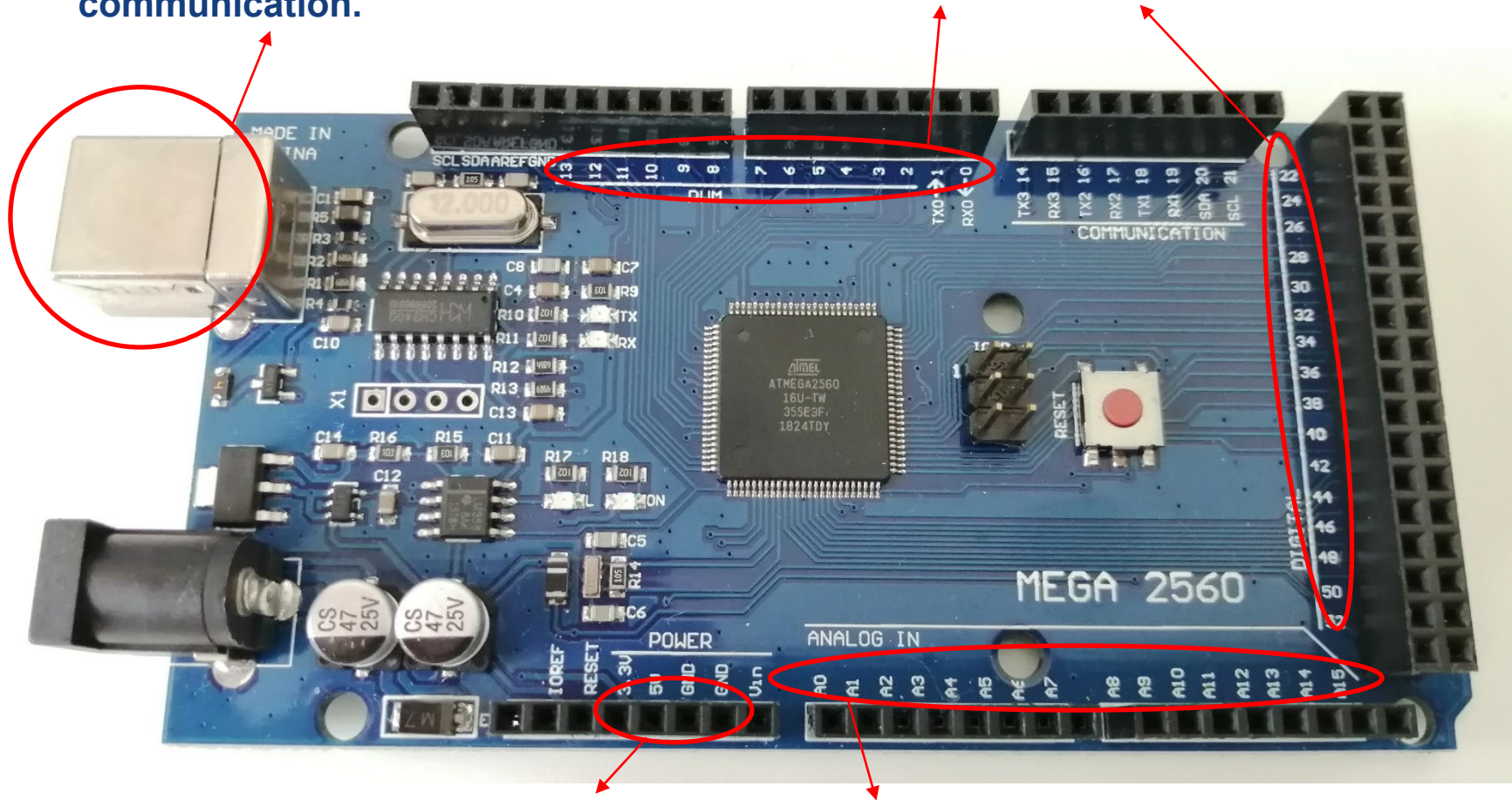
The use of this micro-controller is for large projects where many inputs and outputs are required.

Arduino MEGA 2560



USB connector input:
Voltage (5V) and computer
communication.

Digital Signals

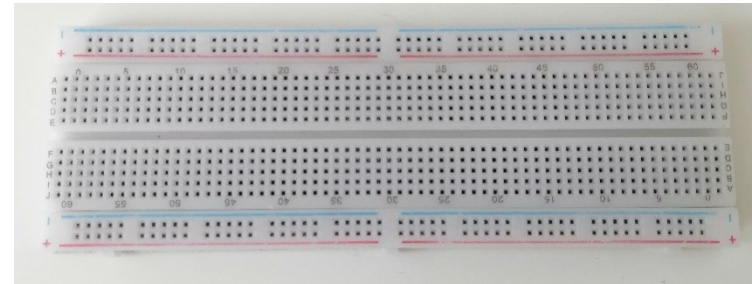


Voltage and Ground (GND)

Analogic Signals

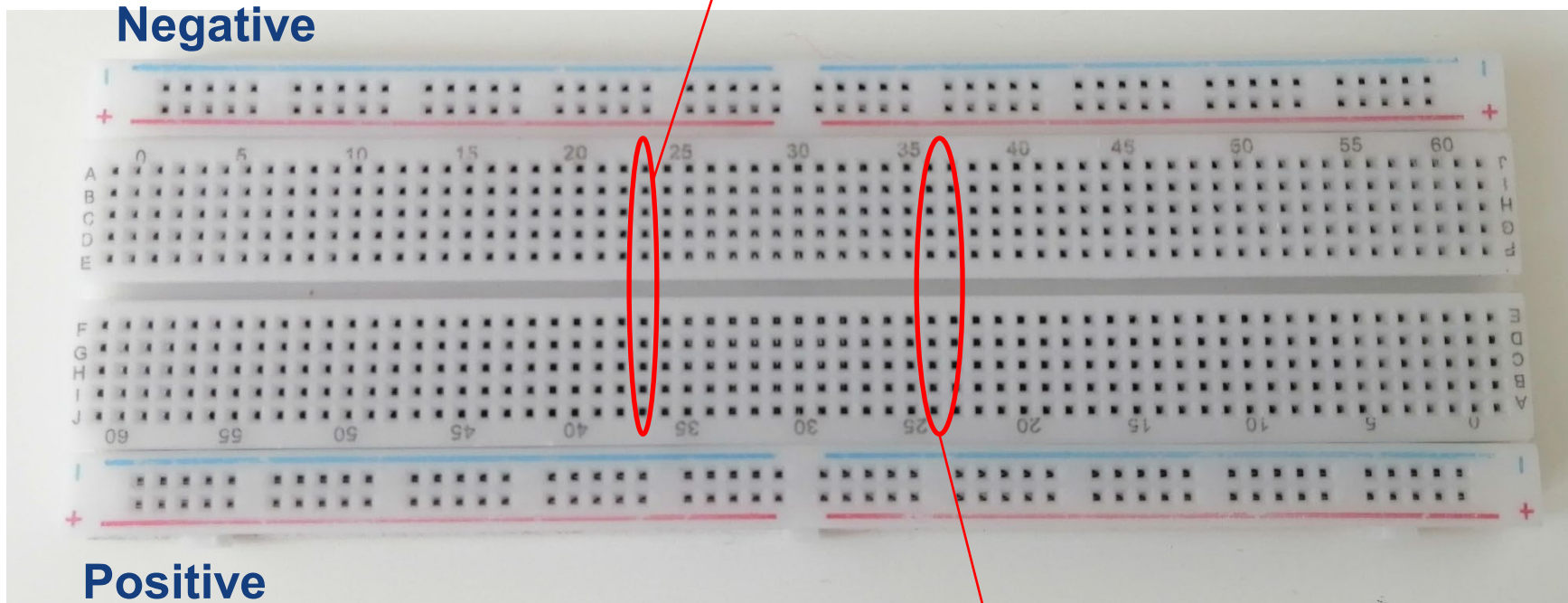
Breadboard

- It is known as Breadboard because before the first electronic projects people used real breadboards. That is, the "tuppers" of their moms in those days.
- To this day the word "Breadboard" is still used even though they are now made of plastic.
- Application: Connect your equipment or system quickly and avoid soldering. Normally a breadboard is always used in the test and development stage.

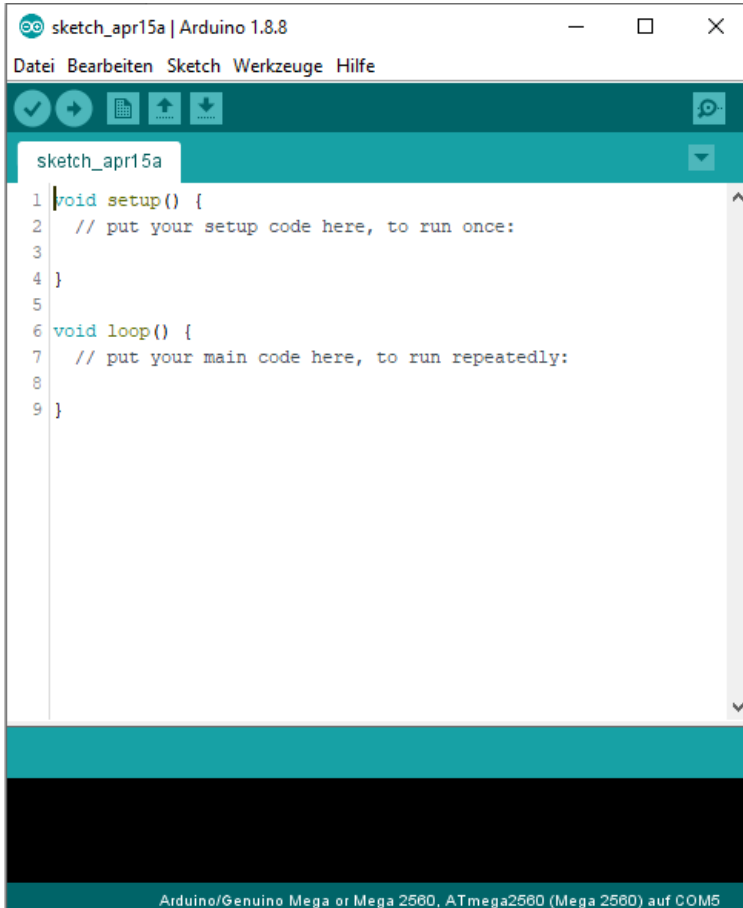


Breadboard

Same line (23), i.e.
everything is in
contact (series)



36 y 37 are not
connected



```
sketch_apr15a | Arduino 1.8.8
Datei Bearbeiten Sketch Werkzeuge Hilfe

sketch_apr15a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) auf COM5
```

The basic structure of Arduino programming is divided into two parts:

1. Setup
2. Loop

The code you write is called "sketches" and is written in C++.

Each sketch needs two void functions: setup() and loop().

Basic Structure – setup()



```
sketch_apr15a | Arduino 1.8.8
Datei Bearbeiten Sketch Werkzeuge Hilfe
sketch_apr15a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) auf COM5
```

The setup() runs only once.

It is triggered when the Arduino is turned on.

The setup() is the part where the initialization steps of your program are indicated.

Basic Structure – loop()



```
sketch_apr15a | Arduino 1.8.8
Datei Bearbeiten Sketch Werkzeuge Hilfe

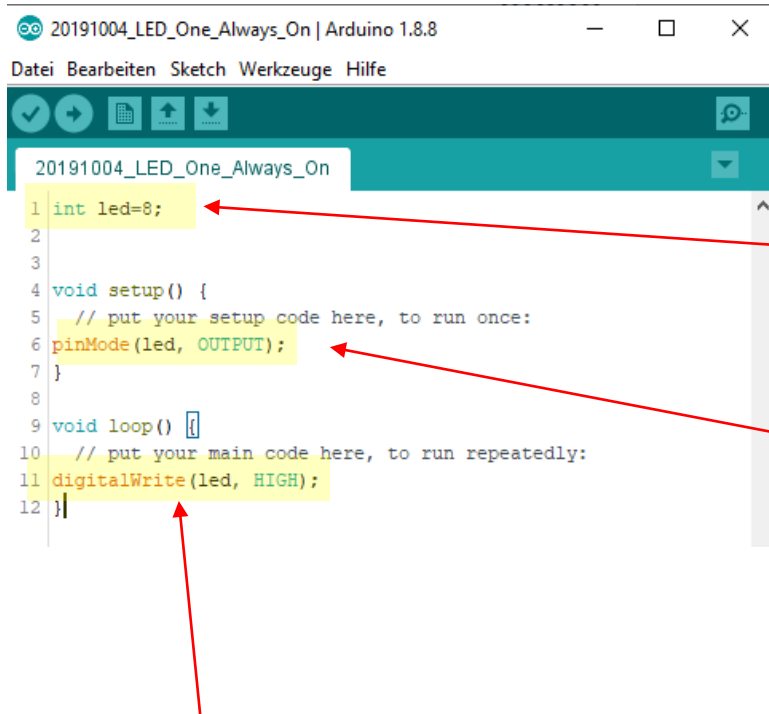
sketch_apr15a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) auf COM5

The loop() runs over and over, continuously.

The loop() includes the code you want to run over and over again.

Example – Turning on a LED



```
1 int led=8;
2
3
4 void setup() {
5 // put your setup code here, to run once:
6 pinMode(led, OUTPUT);
7 }
8
9 void loop() {
10 // put your main code here, to run repeatedly:
11 digitalWrite(led, HIGH);
12 }
```

This small program keeps an LED on.

Before `setup()` we define our pins. We say that pin 8 is called "led".

In the `setup()` we define what "led" is. We tell it that it is a "pinMode" where you have either an input or an output. We will see the "pinMode" later. And we tell it that it is an output with "OUTPUT".

In the `loop()` we say that we want a "HIGH" or "LOW" output. This indication is made by means of "digitalWrite". The "digitalWrite" as well as "pinMode" are commands for inputs and outputs. These commands will be defined later.

Input and Output Commands



```
sketch_apr15a | Arduino 1.8.8
Datei Bearbeiten Sketch Werkzeuge Hilfe
sketch_apr15a $
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode
4
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   digitalWrite
9   digitalWrite
10  analogRead
11  analogWrite
12 }
```

Arduino has some **commands** to define inputs and outputs.

The `pinMode` is a command used to define a pin as input or output within `setup()`.

On the Arduino, digital pins are always understood as input by `setup`.

If you use an input pin, you do not need to redefine it. However, you can do it. Here is an example:

```
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode(1, INPUT); // El pin1 es entrada
4   pinMode(2, OUTPUT); // El pin2 es salida
5 }
```

Input and Output Commands



```
sketch_apr15a | Arduino 1.8.8
Datei Bearbeiten Sketch Werkzeuge Hilfe
sketch_apr15a $
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode
4
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   digitalWrite
9   digitalWrite
10  analogRead
11  analogWrite
12 }
```

Arduino has some **commands** to define inputs and outputs.

The `digitalRead` is a function used to read a digital value from a pin. The result can only be HIGH or LOW.

Example:

```
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode(1, INPUT); // El pin1 es entrada
4   pinMode(2, OUTPUT); // El pin2 es salida
5 }
6
7 void loop() {
8   // put your main code here, to run repeatedly:
9   int valor = digitalWrite(1); // La variable valor
10                                     // es igual a la entrada
11                                     // del pin 1
12 }
```

Input and Output Commands



```
sketch_apr15a | Arduino 1.8.8
Datei Bearbeiten Sketch Werkzeuge Hilfe
sketch_apr15a $
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode
4
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   digitalWrite
9   digitalWrite
10  analogRead
11  analogWrite
12 }
```

Arduino has some **commands** to define inputs and outputs.

The `digitalWrite` sets the pin as ON or OFF by writing the variable HIGH or LOW to its output.

Example:

```
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode(1, INPUT); // El pin1 es entrada
4   pinMode(2, OUTPUT); // El pin2 es salida
5 }
6
7 void loop() {
8   // put your main code here, to run repeatedly:
9   digitalWrite(2, HIGH); // El valor del pin 2 se establece
10  | // como HIGH (encendido)
11 }
```


Input and Output Commands



```
sketch_apr15a | Arduino 1.8.8
Datei Bearbeiten Sketch Werkzeuge Hilfe
sketch_apr15a $
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode
4   |
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   digitalWrite
9   digitalWrite
10  analogRead
11  analogWrite
12 }
```

Arduino has some **commands** to define inputs and outputs.

The `analogRead` is a function that can read a value from an analog pin. The result ranges from 0 to 1023. This function only works with analog pins 0 to 5.

Example:

```
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode(3, INPUT); // El pin3 es entrada
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   analogRead(3, HIGH); // El valor del pin análogo 3 es leído
9                       // La respuesta será entre 0 y 1023
10 }
```

Input and Output Commands



```
sketch_apr15a | Arduino 1.8.8
Datei Bearbeiten Sketch Werkzeuge Hilfe
sketch_apr15a $
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode
4
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   digitalWrite
9   digitalWrite
10  analogRead
11  analogWrite
12 }
```

Arduino has some **commands** to define inputs and outputs.

The `analogWrite` is a function to give an analog output. Check your Arduino manual as only some analog pins support `analogWrite`. For example, on Arduino Uno it is pins 3,5,6,9,10 and 11. The value that can be given is between 0 and 255.

Example:

```
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode(6, OUTPUT); // El pin6 es salida
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   analogWrite(6, x); // El pin6 adquiere el valor de x que
9                       // uno haya decidido entre 0 y 255.
10 }
```