

Automated and data- driven chemistry

LECTURE 8 :
Elements of Robotics
Practical Session

Pascal Miéville
Stefano Di Leone
Edy Mariano
Jean-Charles Cousty

01

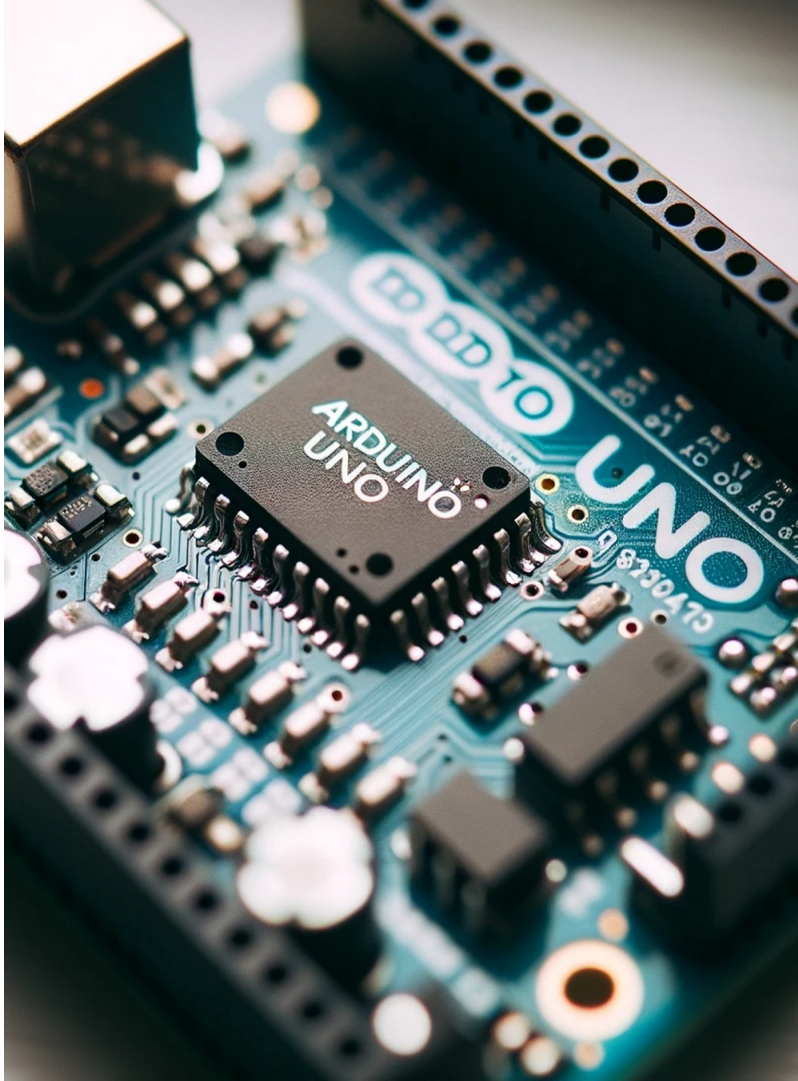
What is Arduino ?

02

Arduino IDE

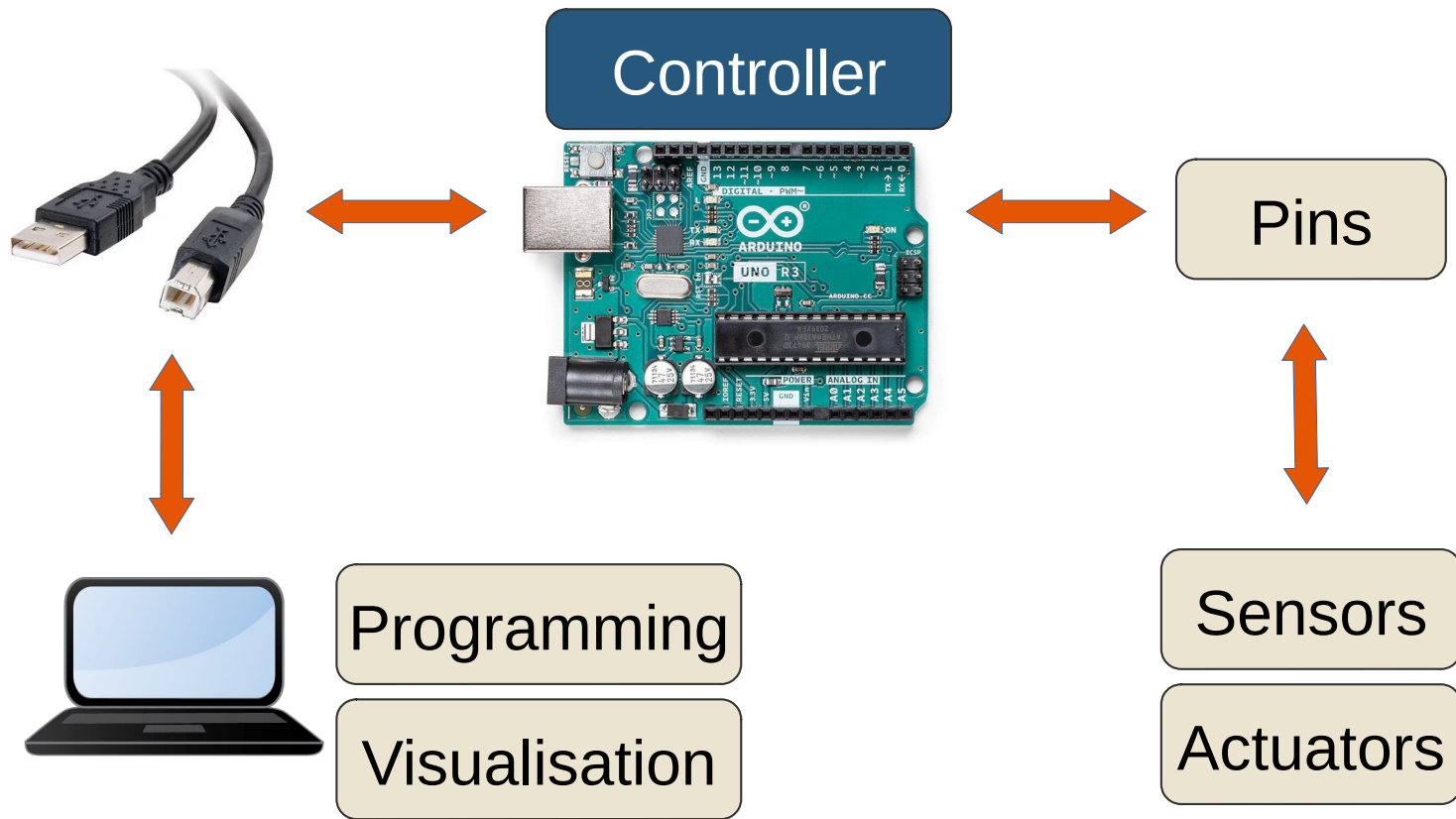
03

Practice

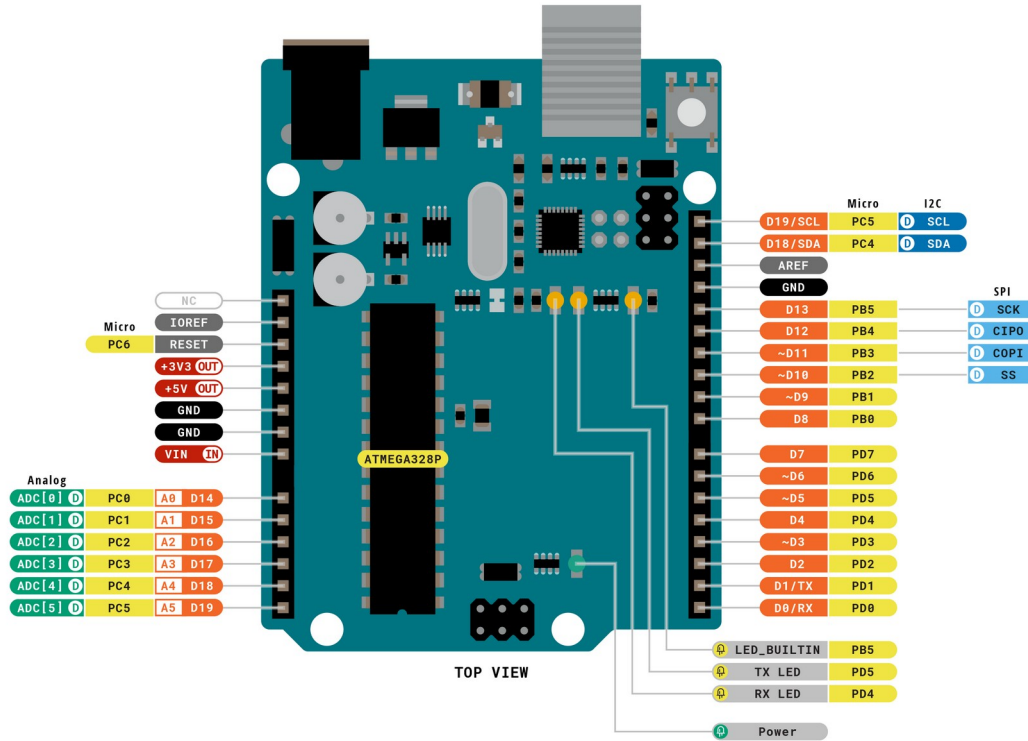


1. What is Arduino ?


EPFL What is Arduino ?



EPFL What is Arduino ?



Legend:	■ Digital	■ I2C
■ Power	■ Analog	■ SPI
■ Ground	■ Main Part	■ Analog



ARDUINO
 ARDUINO UNO REV3
 SKU code: A000066
 Pinout
 Last update: 6 Oct, 2022

EPFL What is Arduino ?

Documentation

<https://www.arduino.cc/reference/en/>

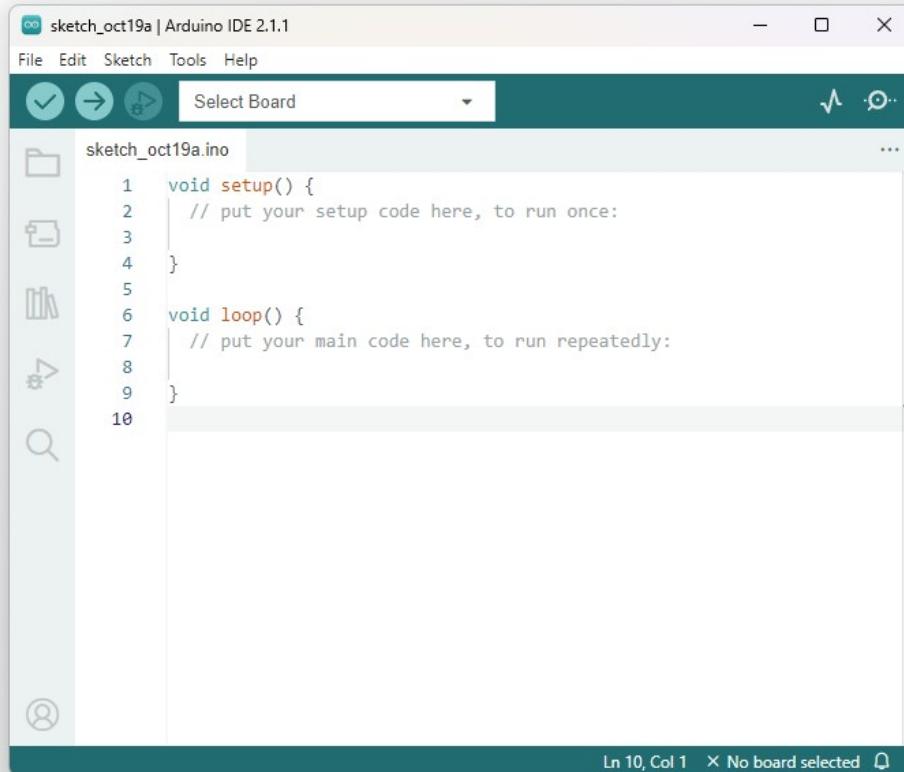
Download

<https://www.arduino.cc/en/software>



1. Arduino IDE

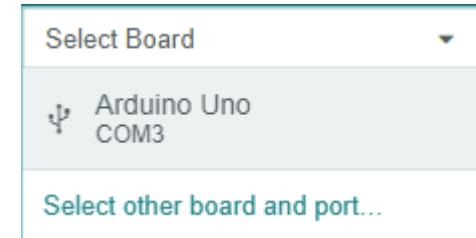
Software to program Arduino boards



Verify your program syntax and check if there are errors



Upload your program into the Arduino board through the USB cable



Select Board type depending on the Arduino model that is connected to the PC

Code Editor

```
sketch_oct19a.ino  
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 }  
10
```

It is always composed of a minimum of 2 functions.

Setup

It runs once. Typically contains initialisations, communication definitions, ...

Loop

It runs repeatedly. Typically contains the core logic of the program, handle input and output

Functions

Performs a specific action or calculation
Always in the following format

functionName(parameter 1, parameter 2, ...)

```
1 void setup() {  
2   pinMode(1, OUTPUT); // sets the digital pin 1 as output  
3 }  
4  
5 void loop() {  
6   digitalWrite(1, HIGH); // sets the digital pin 1 on  
7 }
```

Variables

Used to store a specific value (number, text, ...)
Always in the following format

VariableType variableName

```
1 int pinNumber = 1; // Store the value 1 in the variable pinNumber to use it after
2
3 void setup() {
4   pinMode(pinNumber, OUTPUT);
5 }
6
7 void loop() {
8   digitalWrite(pinNumber, HIGH);
9 }
```

Structures

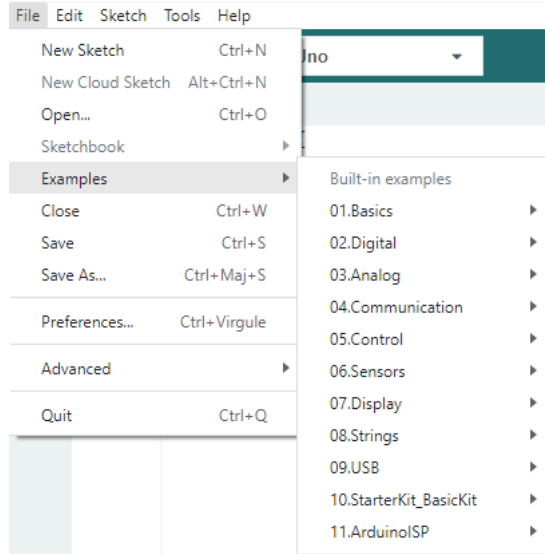
Control Structures are used to create the logic of the program, for example conditions

if (condition) {statement(s)}

```
1 int pinNumber = 1;
2 bool isReady = false;
3
4 void setup() {
5   pinMode(pinNumber, OUTPUT);
6 }
7
8 void loop() {
9   if(isReady == true){
10    digitalWrite(pinNumber, HIGH);
11  }
12 }
```

Examples Programs

Basic examples about typical applications



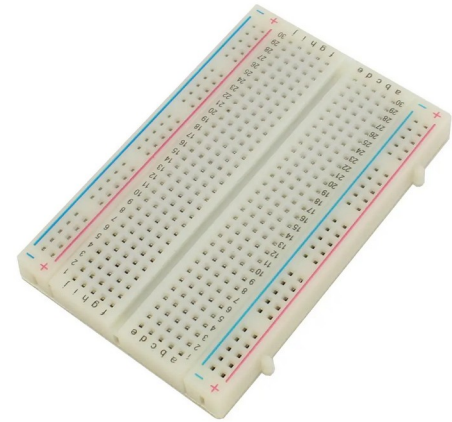
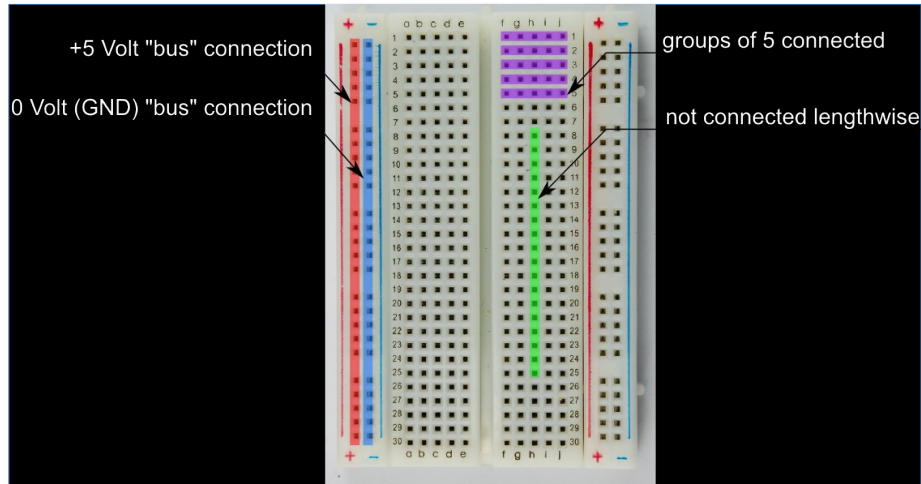
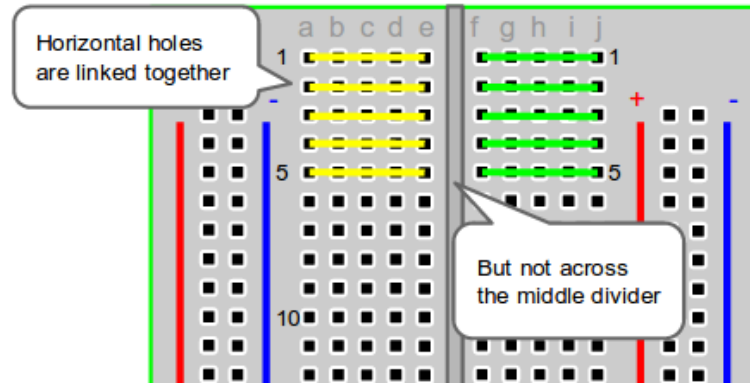
Blink.ino

```

1  /*
2  Blink
3
4  Turns an LED on for one second, then off for one second, repeatedly.
5
6  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
7  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
8  the correct LED pin independent of which board is used.
9  If you want to know what pin the on-board LED is connected to on your Arduino
10 model, check the Technical Specs of your board at:
11 https://www.arduino.cc/en/Main/Products
12
13 modified 8 May 2014
14 by Scott Fitzgerald
15 modified 2 Sep 2016
16 by Arturo Guadalupi
17 modified 8 Sep 2016
18 by Colby Newman
19
20 This example code is in the public domain.
21
22 https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36   delay(1000); // wait for a second
37 }
38

```

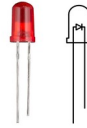
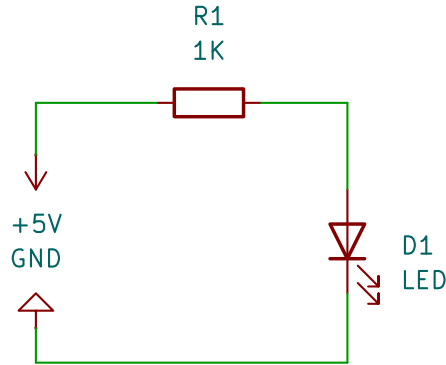
EPFL Material – Breadboard



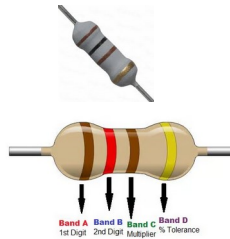


1. Practice

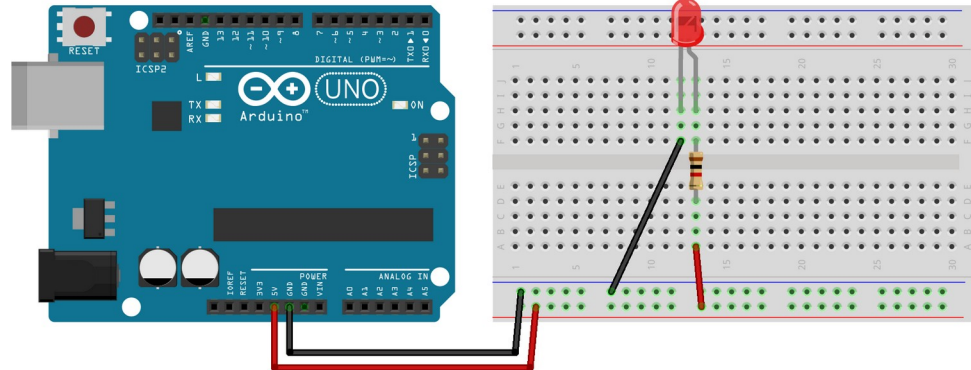
EPFL Exercice 1 - LED



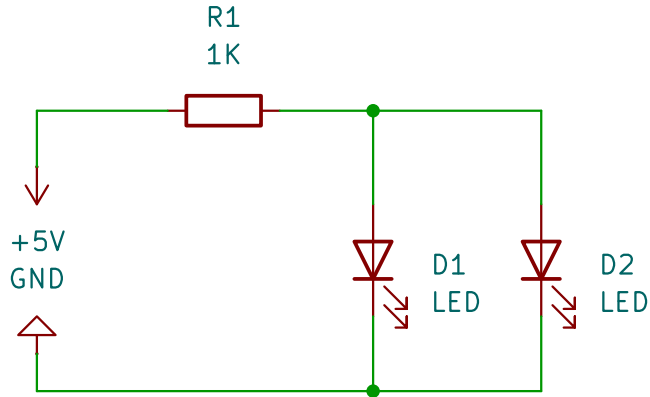
LED Diode that emits light. It blocks the current in one direction. The longer pin is positive (+5V) and the smaller is negative (GND)



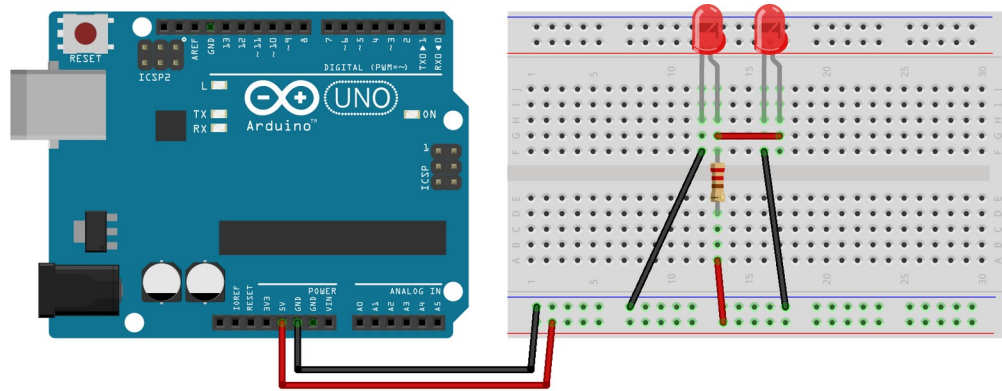
Resistor $U (5V) = R * I$. As the Voltage V is constant, the bigger R is, the less current I will flow through the circuit cables and the LED. LED and Arduino pins can only withstand a certain amount of current I . It is then really important to put a resistor in serial with a LED, otherwise it will overheat.



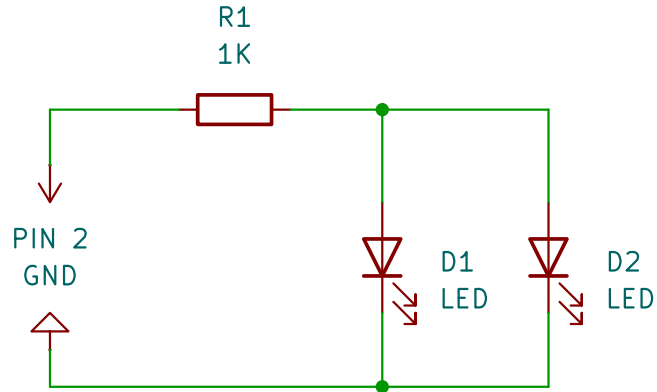
EPFL Exercice 2 – LED in Parallel



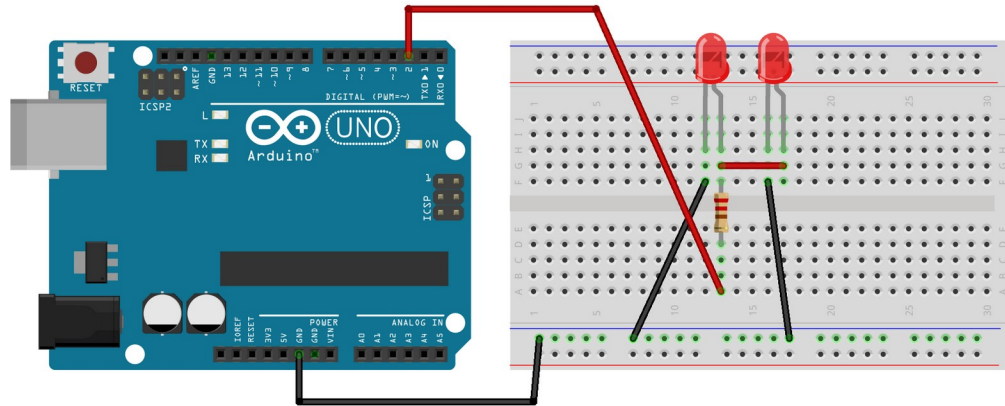
Resistors can be put in parallel. The current will be divided by approximately 2, so the light will be less effective the more LED you put.



EPFL Exercice 3 – LED blink



```
1 void setup() {  
2   pinMode(2, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(2, HIGH);  
7   delay(1000);  
8   digitalWrite(2, LOW);  
9   delay(1000);  
10 }
```



EPFL Exercice 4 – LED blink one after the others

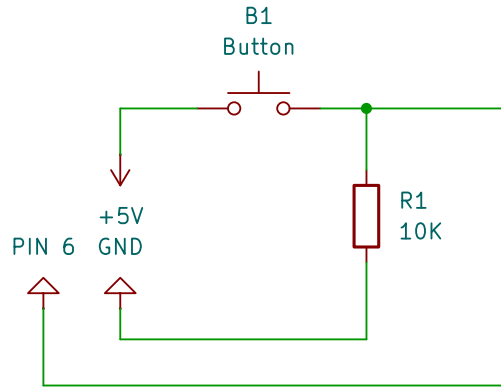
Create a circuit with four LEDs of different colors that blink one after another in sequence.

Each LED should blink for a certain duration, which can be easily modified using a variable.

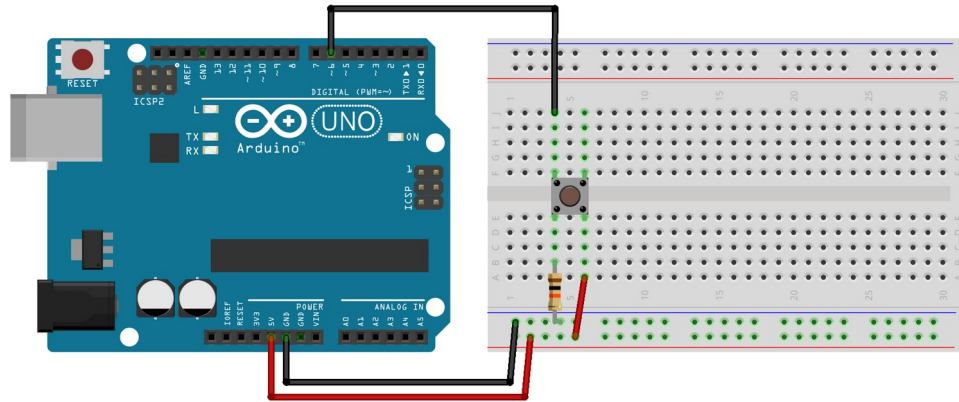
The order of the LEDs in the blinking sequence should also be easily changeable, again through the use of variables.

Solutions will be released on moodle at 9h.

EPFL Exercise 5 – Button



```
1 void setup() {  
2   pinMode(6, INPUT); // Button connected to pin 6  
3   pinMode(13, OUTPUT); // Onboard LED  
4 }  
5  
6 void loop() {  
7   if (digitalRead(6) == HIGH) {  
8     digitalWrite(13, HIGH); // Turn LED on when button is pressed  
9   } else {  
10    digitalWrite(13, LOW); // Turn LED off when button is released  
11  }  
12 }
```



fritzing

EPFL Exercice 6 – Button and LED

Create a circuit with one button and one LED.

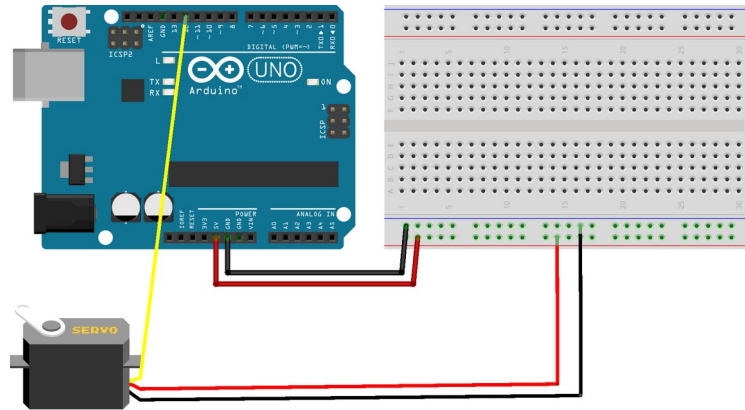
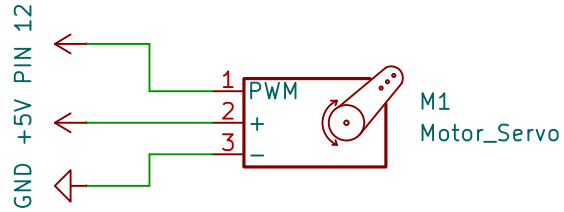
When the button is pressed, the LED should turn on.

When the button is released, the LED should turn off.

The pin numbers for both the button and the LED should be defined as variables, so they can be easily changed if needed.

Solutions will be released on moodle at 9h20.

EPFL Exercice 7 – Servo



fritzing

```
1 #include <Servo.h>
2
3 Servo myServo;
4 int servoPin = 12;
5 int LEFT = 180;
6 int STOP = 90;
7 int RIGHT = 0;
8 int duration = 1000;
9
10 void setup() {
11     myServo.attach(servoPin);
12     myServo.write(STOP);
13 }
14
15 void loop() {
16     myServo.write(LEFT);
17     delay(duration);
18
19     myServo.write(STOP);
20     delay(duration);
21
22     myServo.write(RIGHT);
23     delay(duration);
24
25     myServo.write(STOP);
26     delay(duration);
27 }
```

EPFL Exercice 8 – Servo and Button (advanced)

Create a circuit with two buttons and one servo motor.


When the first button is pressed, the servo should rotate clockwise for a certain amount of time.


When the second button is pressed, the servo should rotate counterclockwise for a certain amount of time.

The duration of rotation and the pin numbers for the components should be defined using variables, so they can be easily modified later.

Thank you for your attention.

Thanks to

 Swiss CAT+ team & partners

 Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

 ETH-RAT

EPFL

 Innosuisse

Gimp, Ubuntu Foundation