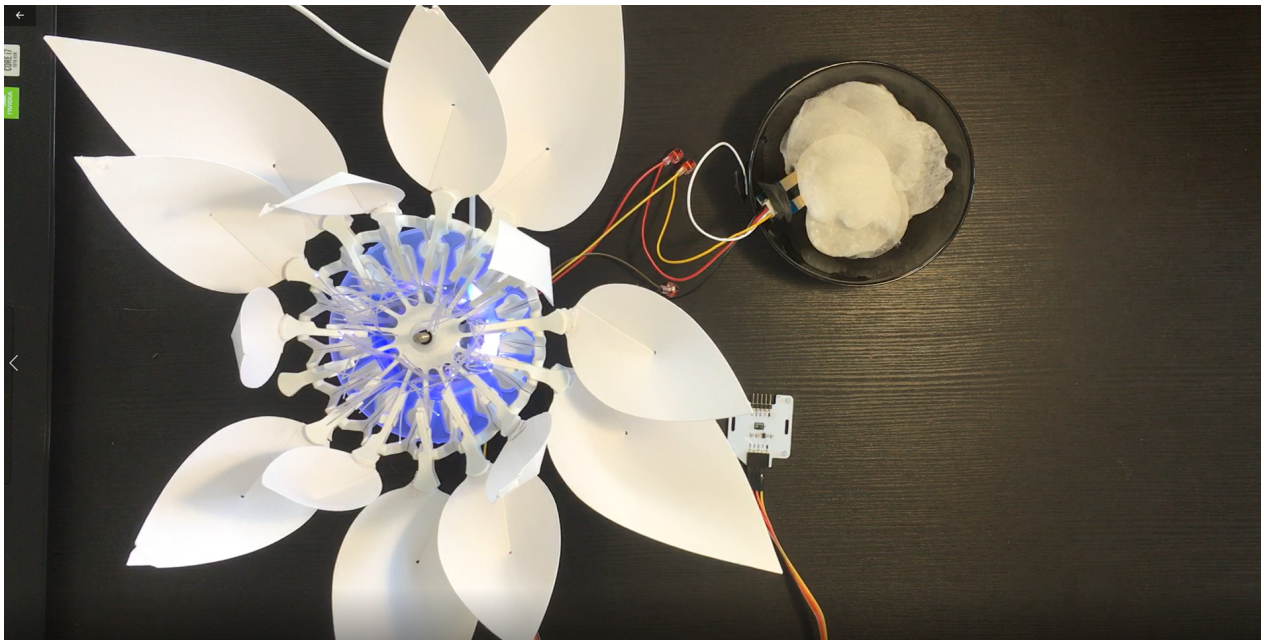
STEP-BY-STEP

Nutrition of the flower



Flowers need light and essential nutrients to thrive. Amongst others nutrients can be found in covering the soil. In this project we will show a continuous growth of the flower depending of the nutrients that will be imitated by the amount of water we add to the soil (or in this experiment cotton).

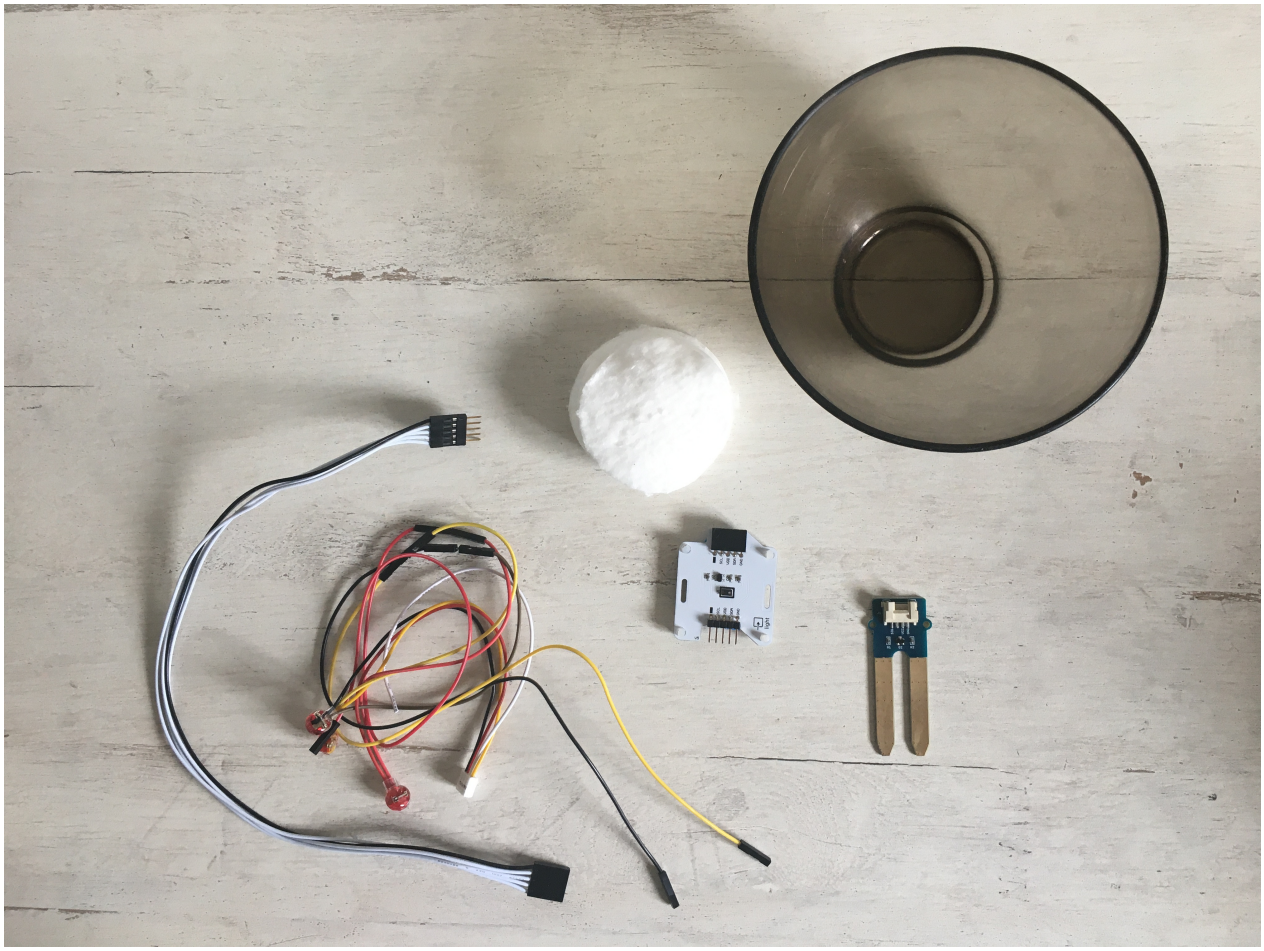
You will write a program to interact with the flower. The flower opens if it's daytime (ambient luminosity) and closes if it's night (put your hand on the light sensor).In addition, the degree of opening of the flower depends on the amount of the water contained in the glass filled with cotton (representing the soil).
The more water there is, the better the flower grows well and thus opens more and more.

Objective

- You can toggle an LED.
- You can integrate a light sensor.
- You can integrate a moisture soil sensor.
- You can control a stepper motor.
- You understand conditionals.
- You can deal with global and local variables.
- You understand logical operators.
- You understand writing and calling functions.

Material

- 1 Bionic Flower
- 1 light sensor
- 1 moisture soil sensor
- Jumper cables
- Cotton and glass
- *NUTRITION_OF_THE_FLOWER_Code_Challenge.ino* (download on github)



Task 1 : Implement the stepper motor

Use the step by step motor to open or close the Bionic Flower.

Code:

1. Open the *NUTRITION_OF_THE_FLOWER_Code_Challenge.ino* file.
2. *library*

   Add the library to control the motor of the Bionic Flower.
3. *global variables*
   ** Create the object for the motor.
   ** Add the *motor_calibration()* function.
4. *setup()*
   ** Initialize the motor.
   ** Calibrate the motor.
5. *loop()*
   ** Open the Bionic Flower all the way
   ** Wait 1s
   ** Close the Bionic Flower

Task 2: Implement the light sensor

The light sensor allows to measure the luminosity. This sensor use the I2C communication, so it uses the SCL and SDA pins.

In this task, you will read the luminosity value and open the flower if it's day and close the flower is if it's night.

Wiring scheme:

Light sensor	ESP32
SCL	GPIO 22
SDA	GPIO 21
(+)	5 V
(-)	GND

Code:

1. *library*

   Add the library for the I2C communication and the library for the light sensor.
2. *global variables*
   ** Define the I2C communication pins.
   ** Add the variable of the light sensor.
   ** Define 1 global variable for a luminosity for the limit between the day and the night.
   ** Create the object for the light sensor.
3. *setup()*
   ** Start the I2C communication.
   ** Initialize the light sensor.
4. *loop()*
   ** Read the value from the light sensor.

```
 //Read the luminosity value
luminosity_sensor= rpr0521rs.get_psalsval(&proximity,&luminosity);
Serial.println("Luminosity value");
Serial.println(luminosity);
```

** Write an while-structure :
° while luminosity is high enough (day), the flower opens.
° otherwise (night), the flower closes.

Task 3 : Control the LEDs

Change the color of the LED's. The Bionic Flower is equipped of 5 LEDs. The color of each LED is given by an RGB code. The LEDs are connected on GPIO 16.

Wiring scheme:

LEDs	ESP32
LEDs	GPIO 16

Code:

1. *library*

   Add the library to control the LEDs.
2. *global variables*
   ** Define the GPIO of the LEDs and give it the variable name "LED_PIN".
   ** Define a variable for the number of the LEDs.
   ** Create the object for the LEDs.
   ** Create a function to light up the LEDs in blue (represents the water). Think about adding a function to turn off the LEDs (black color).
   RGB code website link : https://www.w3schools.com/colors/colors_picker.asp
3. *setup()*
   ** Initialize the LEDs.
   ** Turn off the LEDs.
4. *loop()*

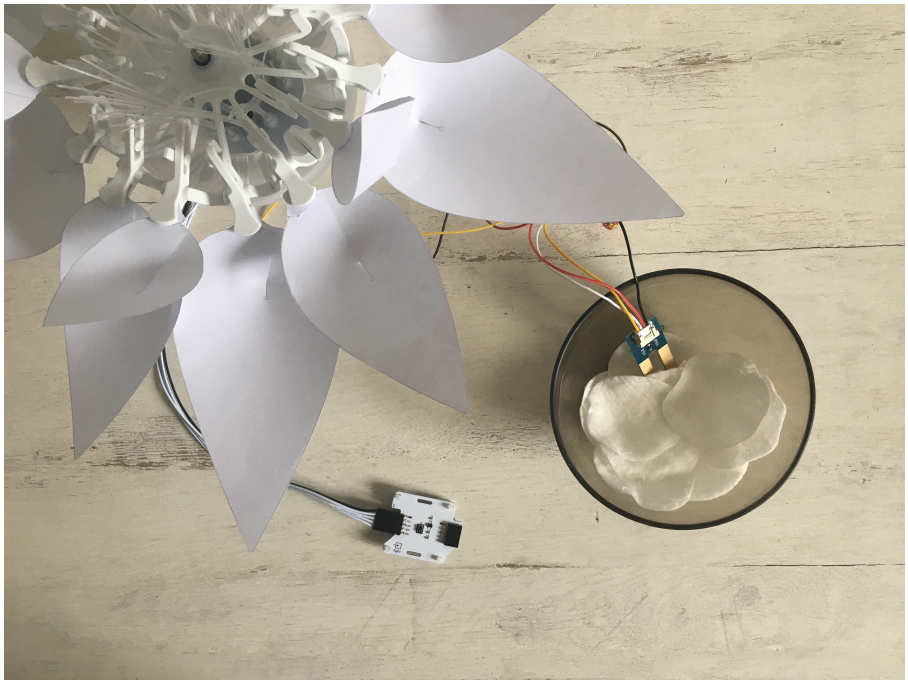
   Make a load movement with the LEDs using a for-structure. For each LED :
   ** turn on the LED in blue
   ** wait 1s
   ** turn off the LED

Try to change the delay values, and see the impact on the speed of the load movement.

Task 4: Implement the moisture soil sensor

The moisture soil sensor measures the amount of the water. This sensor sends a analog value proportional to the amount of water. The sensor is connected on GPIO 4.

In this task, you will read the moisture value and calculate a delay which determinates the speed of the load movement. In addition, you will calculate the degree of opening of the flower depending on the amount of the water.



Wiring scheme:

Moisture soil sensor	ESP32
Signal	GPIO 4
(+)	5 V
(-)	GND

Code:

1. *global variables*
   ** Define the GPIO of the moisture soil sensor.
   ** Define a global variable for the humidity value.
   ** Define a global variable to store the delay (speed of the load movement).
   ** Define a global variable to store the degree of opening of the flower.
2. *setup()*
   Setup the sensor as INPUT.
3. *loop()*
   ** Read the value from the humidity sensor.

```
//Read the humidity value
  humidity_value = analogRead(HUMIDITY_PIN);
  Serial.println(humidity_value);
```

** Write a while-structure :
° if the humidity value is higher than 3000, the degree of the opening is maximum (so equals to 120) and the flower lights up in blue.
° else :

* calculate the delay value with the next operation:
`dl = (100-(humidity_value/30))*10;`

* load movement

* calculate the degree of the opening with the next operation:
`limit_open=humidity_value*FLOWER_CLOSE_TO_OPEN/3000;`

Task 5: Scenario

Now, create the final program to recreate the scenario:

- The flower opens if it's day. Otherwise (it's night) the flower closes.
- The degree of the opening of the flower and the speed of the movement of the LEDs depend on the humidity. The more water there is, the more the flower opens and the faster is the movement of the LEDs.