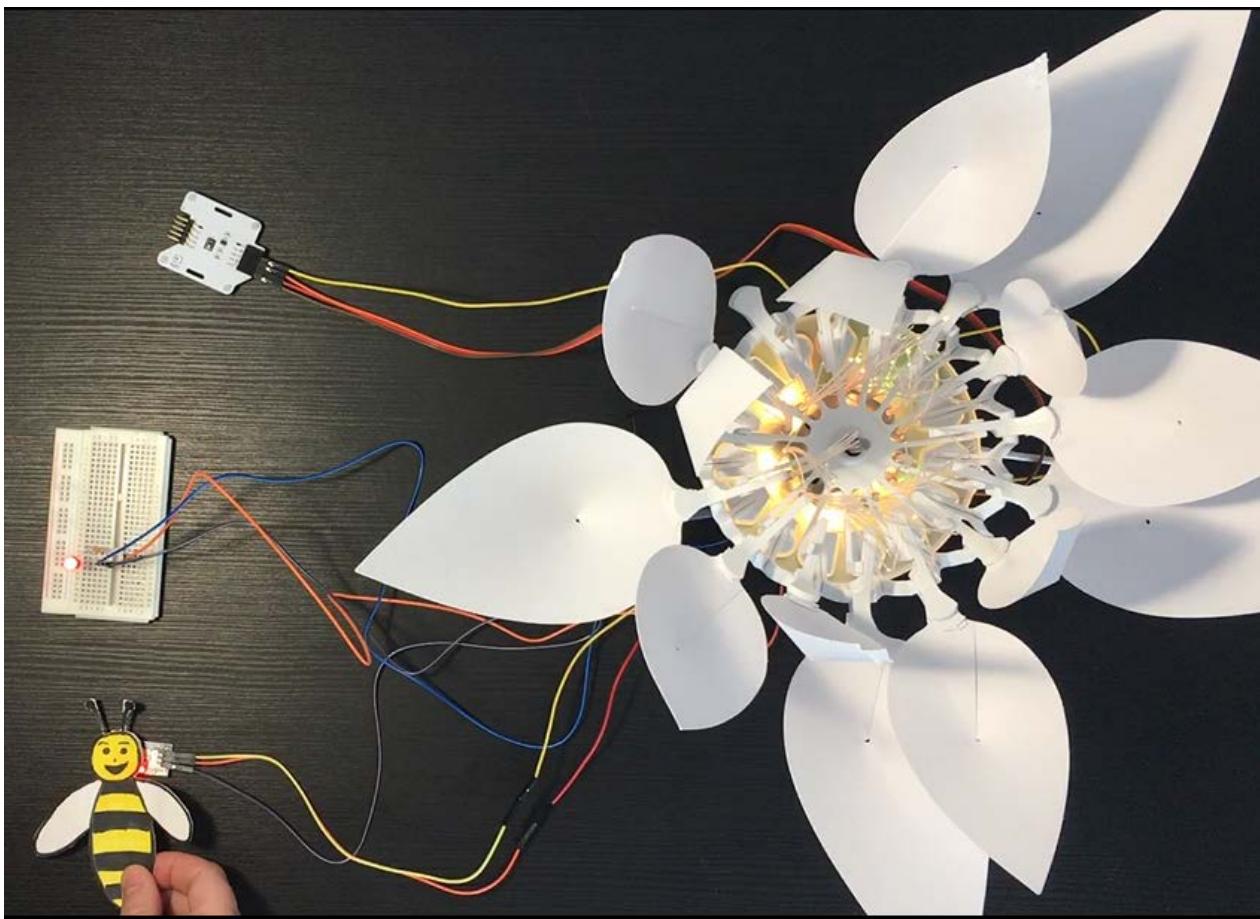


STEP-BY-STEP

Bee



Bees are important in the life of flowers. Indeed, the main role of bees is the pollination of flowers that allows the reproduction of plants. In this project you will recreate the interaction between the bees and the flowers. And more specifically, reproduce the sound of bees (buzz) when they are close to flowers and underline the fact that most bees are diurnal animals (active only during the day).

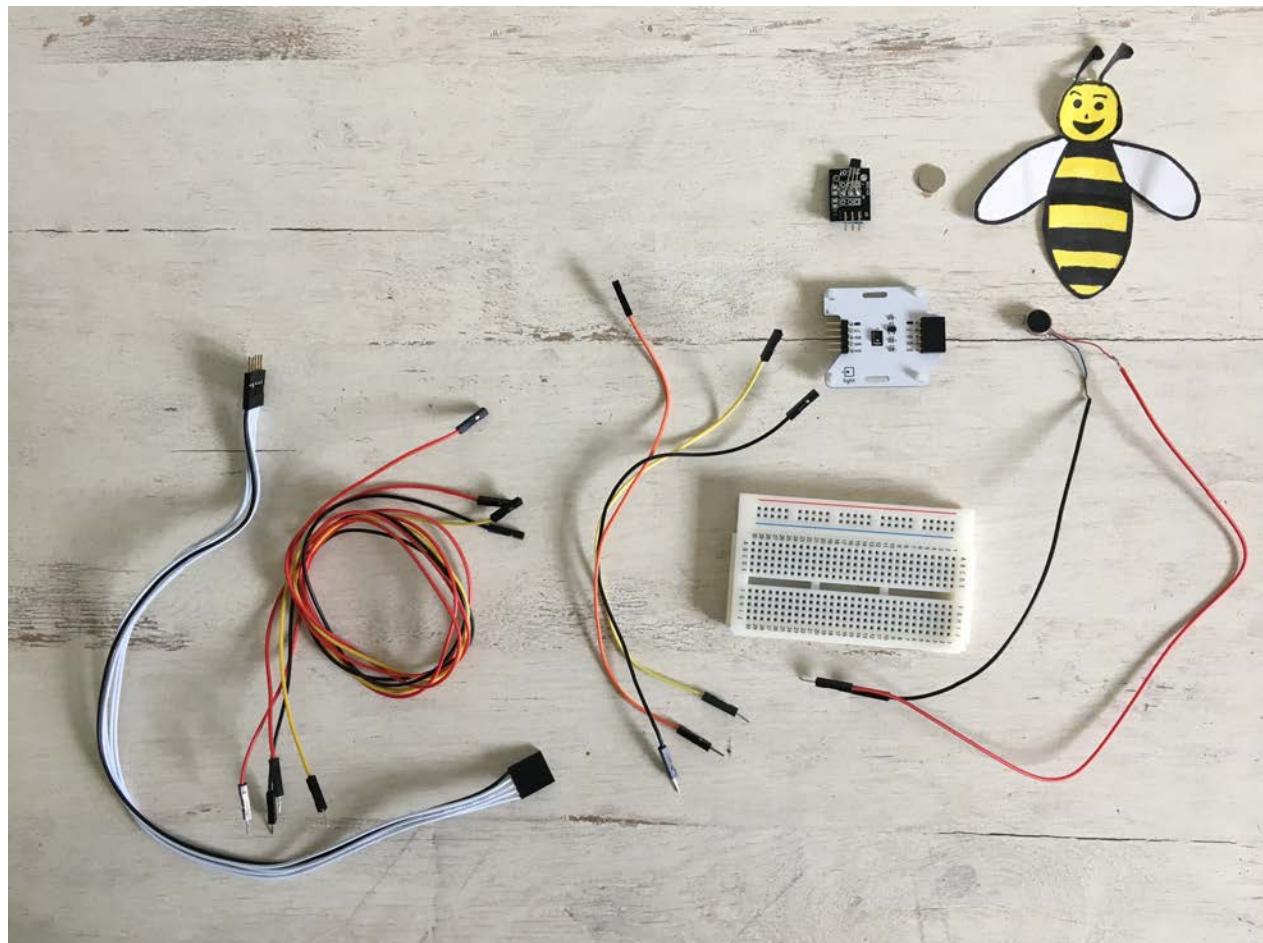
You will write a small program to interact with the flower. If it's day (bright environment) the flower opens. Otherwise, it's night and the flower closes. In addition, you will draw a bee and place a magnet underneath. The hall effect sensor will then detect the presence of the bee (i.e. the magnet), which will activate the vibrating motor. This can only be done if it's during the day.

Objective

- You can toggle an LED.
- You can integrate a hall sensor.
- You can integrate a light sensor.
- You can control a vibrating motor.
- 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.

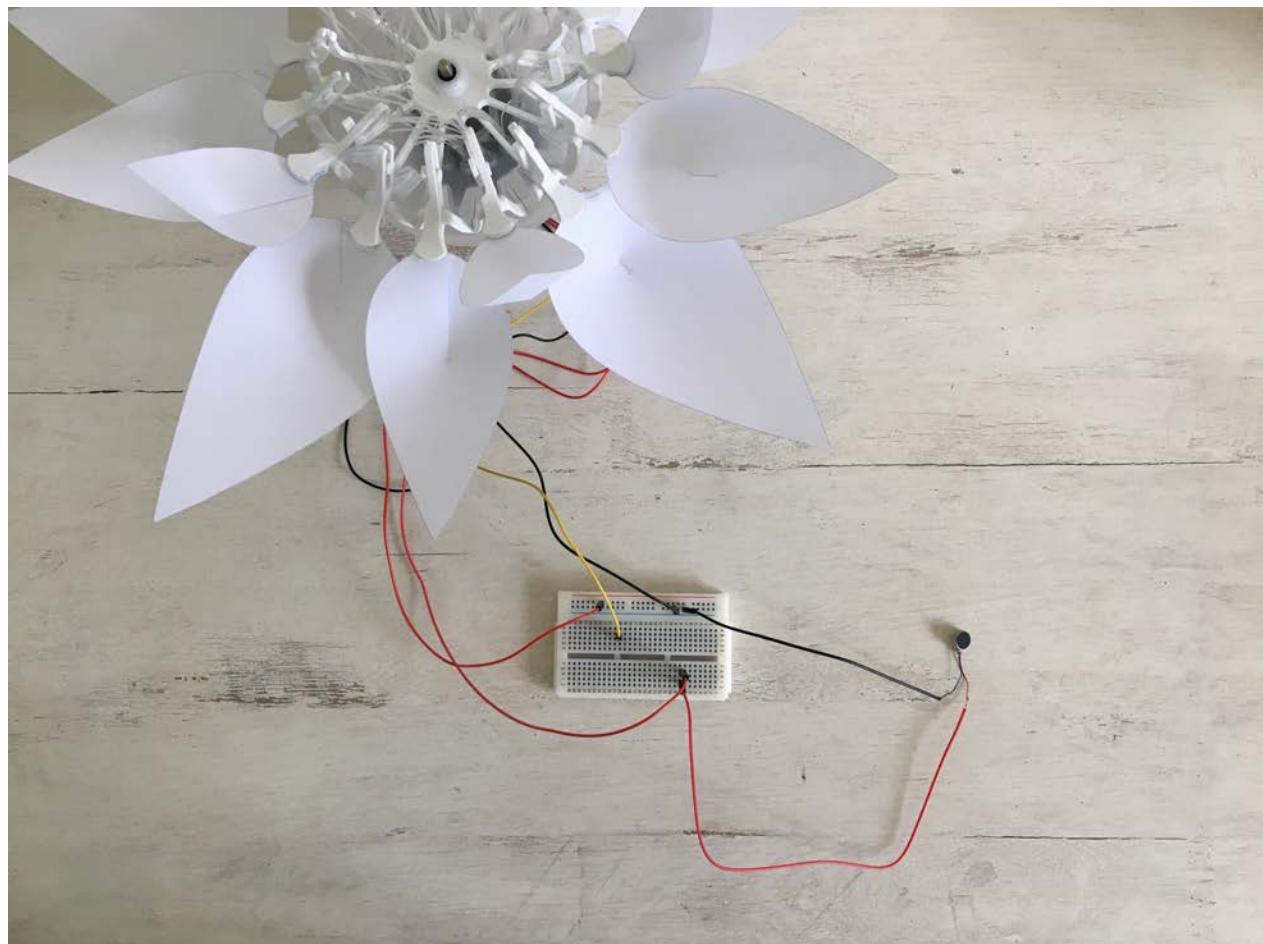
Material

- 1 Bionic Flower
- 1 light sensor
- 1 hall sensor
- 1 vibrating motor
- Jumper cables
- *BEE_Code_Challenge.ino* (download on github)



Task 1: Active the vibrating motor

Activate the vibrating motor every 1s. The vibrating motor is defined on GPIO 22.



Wiring scheme:

Vibrating motor	ESP32
Red	GPIO 22
Black	GND

Code:

1. Open the *BEE_Code_Challenge.ino* file.

2. *global variables*

Define the GPIO of the vibrating motor and give it the variable name "MOTOR_PIN".

3. *setup()*

** Setup MOTOR_PIN as OUTPUT.

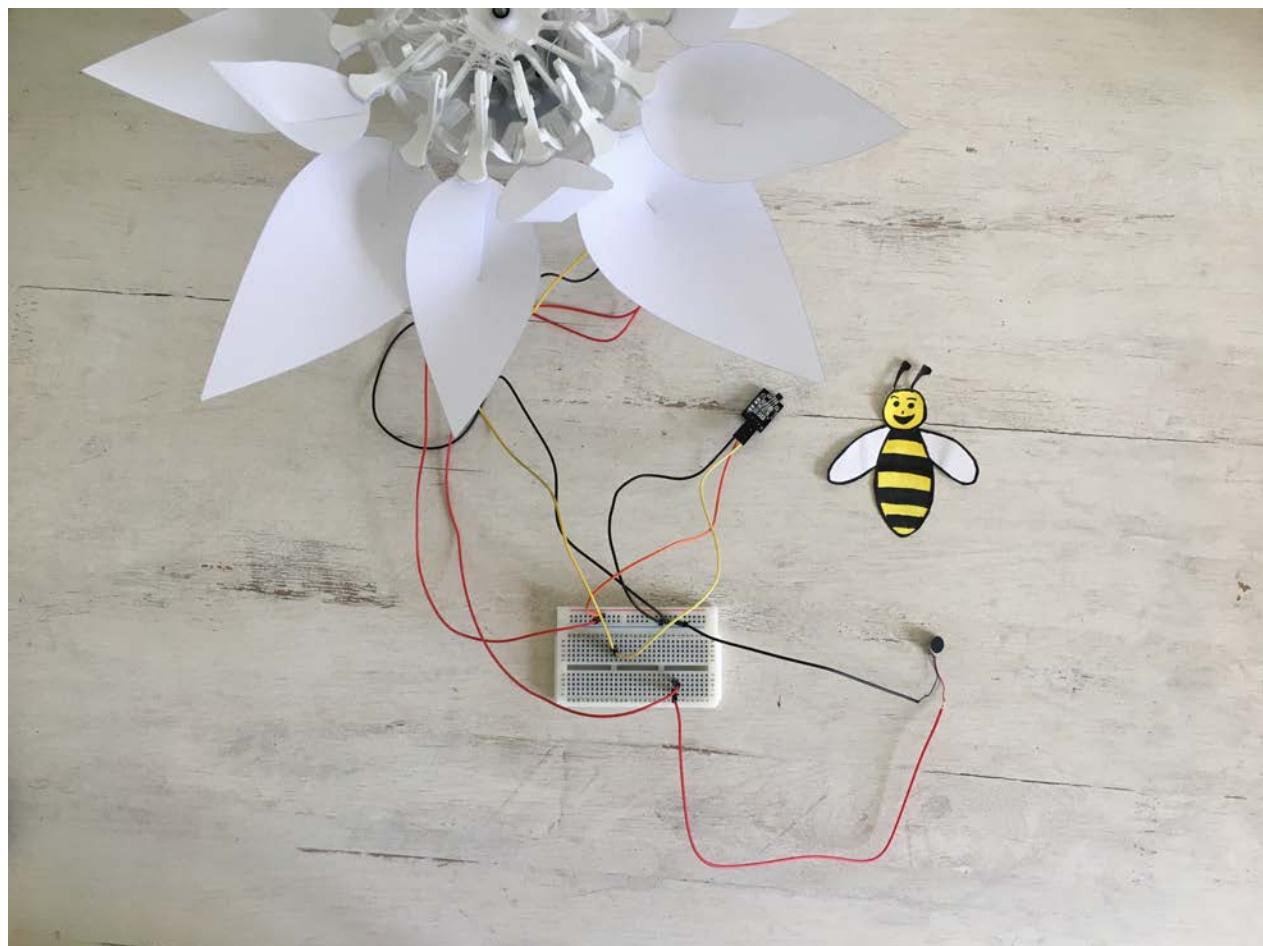
** Power off the vibrating motor. Use the function *digitalWrite(variable, value)*; to control the power of the vibrating motor via LOW value.

4. *loop()*

Control MOTOR_PIN via power on/power off. Use the function *digitalWrite(variable, value)*; to control the power of the vibrating motor via HIGH and LOW values. Include a *delay()*; of 1000 ms between each HIGH and LOW value.

Task 2: Implement the hall sensor

Now we want to work with the hall sensor to detect a magnet. The results, if a magnet is close or not will be viewed on the serial monitor in the Arduino IDE. If a magnet is approaching the hall sensor it detects the change of the magnetic field. The hall sensor has 3 pins : ground (-), power supply and signal pin (S). This sensor is a digital sensor: the hall sensor value is 0 if a magnet is detected, otherwise is 1.



Wiring scheme:

Hall sensor	ESP32
S	GPIO 21
(+)	5V
(-)	GND

Code:

1. global variables

** Define the GPIO of the hall sensor and give it the variable name "HALL_PIN".

** Define a integer variable to store the hall sensor value and give it the variable name "magnet".

2. *setup()*

Setup HALL_PIN as INPUT.

3. *loop()*

** Read the values from the hall sensor with digitalRead() and display them on the serial monitor.

```
//Read the hall sensor
magnet = digitalRead( HALL_PIN);
Serial.println("Magnet");
Serial.println(magnet);
```

** Write an if-structure:

- ° If a magnet is detected, turn on the vibrating motor.
- ° If no magnet is detected, turn off the vibrating motor.

Task 3: Control the LEDs

Change the LED's color. The Bionic Flower is composed of 5 integrated 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".

** Create the object for the LEDs.

** Create 2 functions to light up the LEDs for the day and for the night. 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 color sequence :

** Turn on the LEDs in the color used for the day

** Wait 500 ms

** Turn on the LEDs in the color used for the night

** Wait 500 ms

Task 4: Implement the stepper motor

Use the stepper motor to open or close the Bionic Flower.

Code:

1. library

Add the library to control the motor of the Bionic Flower.

2. global variables

** Define the GPIO and variable for the motor.

** Create the object for the motor.

** Add the *motor_calibration()* function.

3. setup()

** Initialize the motor.

** Calibrate the motor.

4. loop()

** Light up the LEDs in the color you chose for the day

** Open the flower all the way

** Wait 1 s

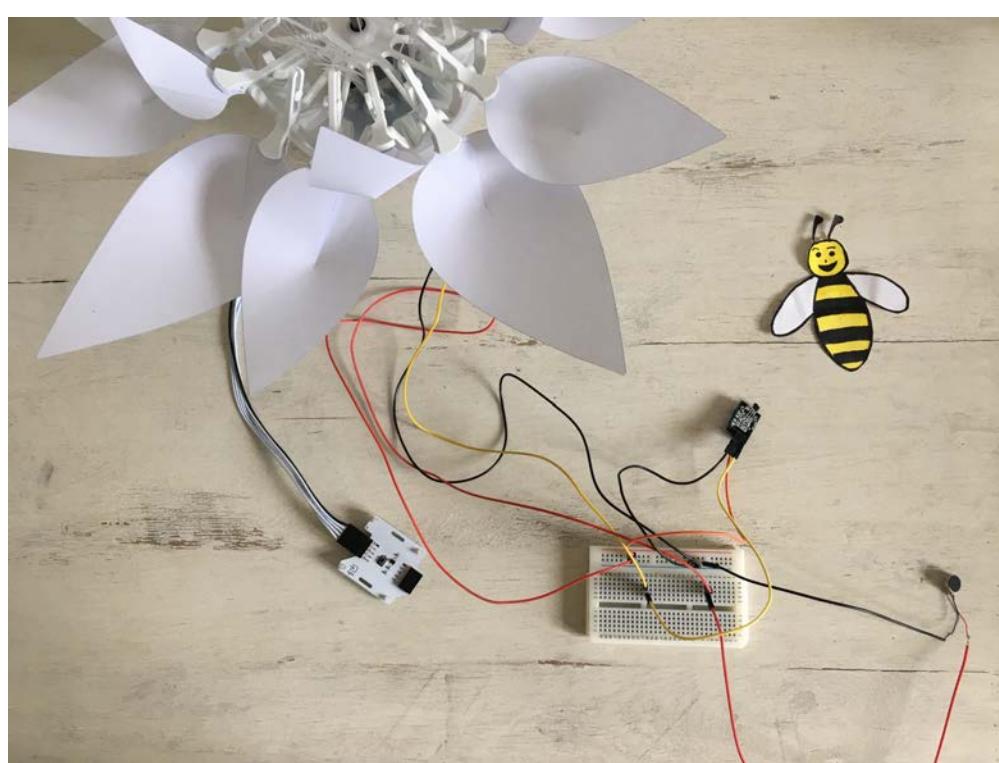
** Light up the LEDs in the color you chose for the night

** Close the flower

** Wait 1 s

Task 5: 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.



Wiring scheme:

Light sensor	ESP32
SCL	GPIO 5
SDA	GPIO 4
(+)	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 global variables for a luminosity threshold value (limit between day and 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 a while-structure :

° if it's bright enough to be defined as daytime the LEDs lights in color you chose for the day and the flower is opening.

° otherwise, the LEDs lights up in the color you chose for the night and the flower is closing.

Task 6: Scenario

Now, create the final code to recreate the scenario :

** Day : LEDs light up in day color, the flower opens. In addition, if a magnet is close to the hall sensor the vibrating motor is activated .

** Night : LEDs light up in night color and the flower closes.