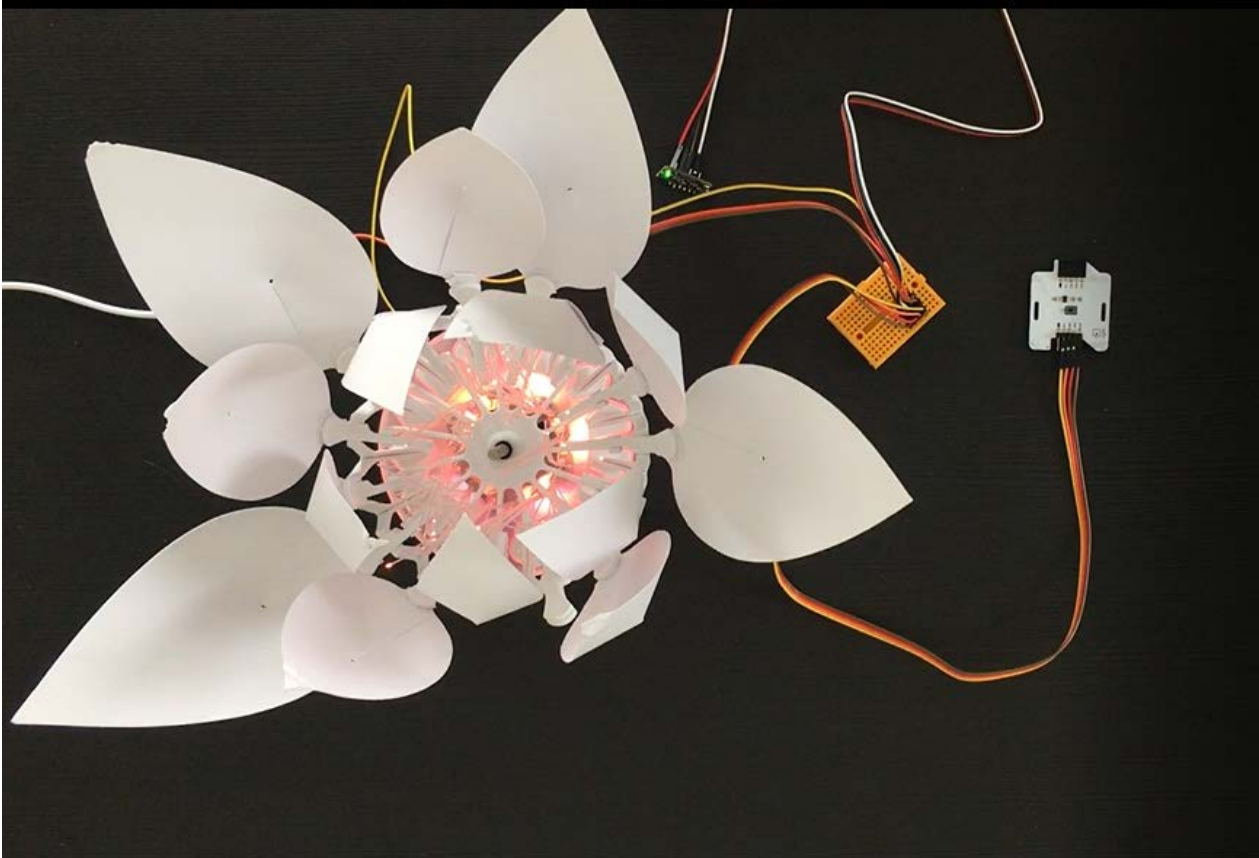


Schritt für Schritt

Der Tages Rhythmus der Blume



Blumen brauchen Licht und eine warme Umgebung (hohe Temperatur) zum Wachsen. In diesem Projekt werden wir den Tagesablauf einer Blume nachbilden. Dabei öffnet sich die Blume morgens und bleibt den ganzen Tag geöffnet. Außerdem hängt der Öffnungsgrad der Blüte von der Temperatur ab. Wenn es Nacht wird, schließt sich die Blume wieder.

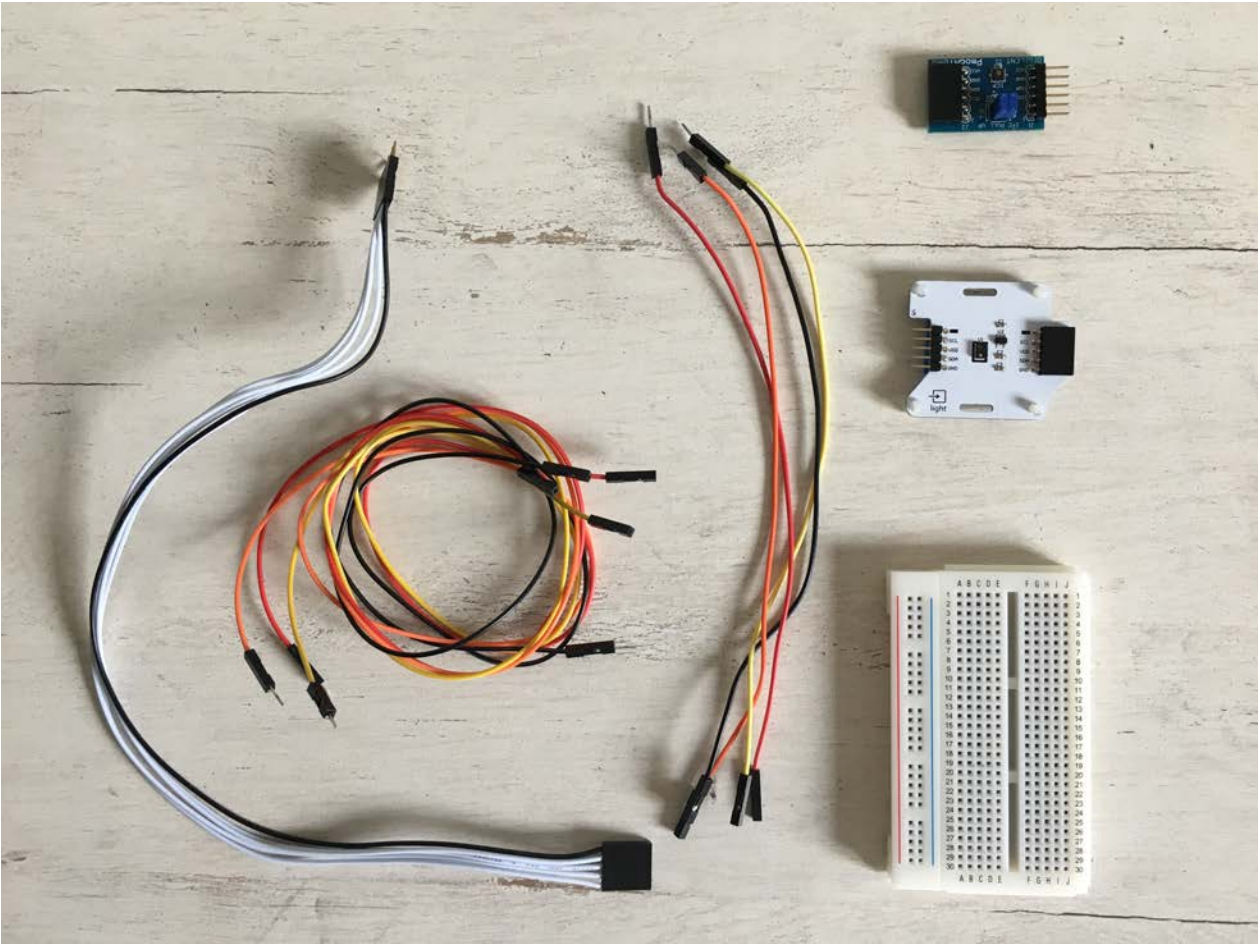
Die Farbe der Blume hängt von der Tageszeit ab Morgen (Umgebungs-helligkeit), Tag (mit dem Licht deines Smartphones beleuchten) oder Nacht (die Hand auf den Sensor legen). Die Blume braucht Licht, um sich zu öffnen (sie ist offen, wenn es Tag oder Morgen ist und geschlossen, wenn es Nacht ist). Und eine hohe Temperatur (Temperatursensor in Ihren Händen) verbessert das Wachstum der Blume, so dass sich die Blume mehr öffnet.

Zielsetzung

- Du kannst eine LED schalten.
- Du kannst einen Lichtsensor integrieren.
- Du kannst einen Temperatur integrieren.
- Du kannst einen Schrittmotor ansteuern.
- Du verstehst Bedingungen.
- Du kannst mit globalen und lokalen Variablen umgehen.
- Du verstehst logische Operatoren.
- Du verstehst das Schreiben und Aufrufen von Funktionen.

Material

- 1 Bionic Flower
- 1 Lichtsensor
- 1 Temperatursensor
- Jumper Kabel
- *DAY_OF_THE_FLOWER_Code_Challenge.ino* (download on github)



Aufgabe 1: Steuer die LEDs an

Ändere die Farbe der LED's und lass sie aufblinken Die Bionic Flower besteht aus 5 eingebauten LEDs. Die Farbe jeder LED ist durch einen RGB-Code gegeben. Die LEDs sind an GPIO 16 angeschlossen.

Verdrahtungsplan:

LEDs	ESP32
LEDs	GPIO 16

Code:

1. Öffne die Datei: *DAY_OF_THE_FLOWER_Code_Challenge.ino*.
2. *Bibliothek*

Füg die Bibliothek zur Steuerung der LEDs hinzu.

3. *Globale Variablen*

- ** Definiere den GPIO der LEDs und gib ihm den Variablennamen "LED_PIN".
- ** Erstelle das Objekt für die LEDs.
- ** Erstelle 3 Funktionen, um die LEDs für den Morgen, den Tag und für die Nacht zu beleuchten. Denk daran, eine Funktion zum Ausschalten der LEDs (schwarze Farbe) hinzuzufügen.

Hier findest du Hilfe zu den Farben: RGB-Code Website-Link : https://www.w3schools.com/colors/colors_picker.asp

4. *setup()*

- ** Initialisiere die LEDs.
- ** Schalte die LEDs aus.

5. *loop()* (Schleife)

Erstellen Sie eine Farbabfolge :

- ** Schalte die LEDs in der Farbe ein, die für die Nacht verwendet wird.
- ** Warte 500 ms.
- ** Schalte die LEDs in der Farbe ein, die für den Morgen verwendet wird.
- ** Warte 500 ms.
- ** Schalte die LEDs in der Farbe ein, die für den Tag verwendet wird.
- ** Warte 500 ms.

Aufgabe 2: Integriere den Lichtsensor

Der Lichtsensor ermöglicht die Messung der Umgebungshelligkeit. Dieser Sensor kommuniziert über das I2C-Protokoll, er nutzt also die Pins SCL und SDA. Seine I2C-Adresse ist 0x38.

In dieser Aufgabe liest du den Helligkeitswert und änderst die Farbe der Blume in Abhängigkeit davon.



Verdrahtungsplan:

Lichtsensor	ESP32
SCL	GPIO 5
SDA	GPIO 4
(+)	5 V
(-)	GND

Code:

1. Bibliothek

Füg die Bibliothek für die I2C-Kommunikation und für den Lichtsensor hinzu.

2. Globale Variablen

** Definiere die Pins der I2C-Kommunikation und die I2C Adresse des Sensors

** Füge die Variable für den Lichtsensor hinzu.

** Definiere die globale Variable für einen Helligkeitsschwellenwert (Grenze zwischen Morgen und Tag) und 1 globale Variable für einen Helligkeitsschwellenwert für die Grenze zwischen Tag und Nacht.

** Erstellen Sie das Objekt für den Lichtsensor.

3. setup()

** Starte die I2C-Kommunikation.

** Initialisiere den Lichtsensor.

4. loop() (Schleife)

** Lies den Wert des Lichtsensors aus.

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

** Schreibe eine if-Schleife :

** wenn es Morgens ist, leuchten die LEDs in der "Morgenfarbe".

** else - wenn es tagsüber ist, leuchten die LEDs in der Farbe "Tag".

** else wenn es nachts ist, leuchtet die LEDs in der "Nachtfarbe".

Aufgabe 3: Integriere den Schrittmotor

Benutze den Schrittmotor zum Öffnen und Schließen der Bionic Flower.

Code:

1. Bibliothek

Füg die Bibliothek zur Steuerung des Motors der Bionic Flower hinzu.

2. Globale Variablen

** Definier den GPIO und die Variable für den Motor.

** Erstelle das Objekt für den Motor.

** Füge die Funktion *motor_calibration()* hinzu.

3. setup()

** Initialisiere den Motor.

** Kalibriere den Motor.

4. loop() (Schleife)

** Schreiben Sie eine while-Schleife :

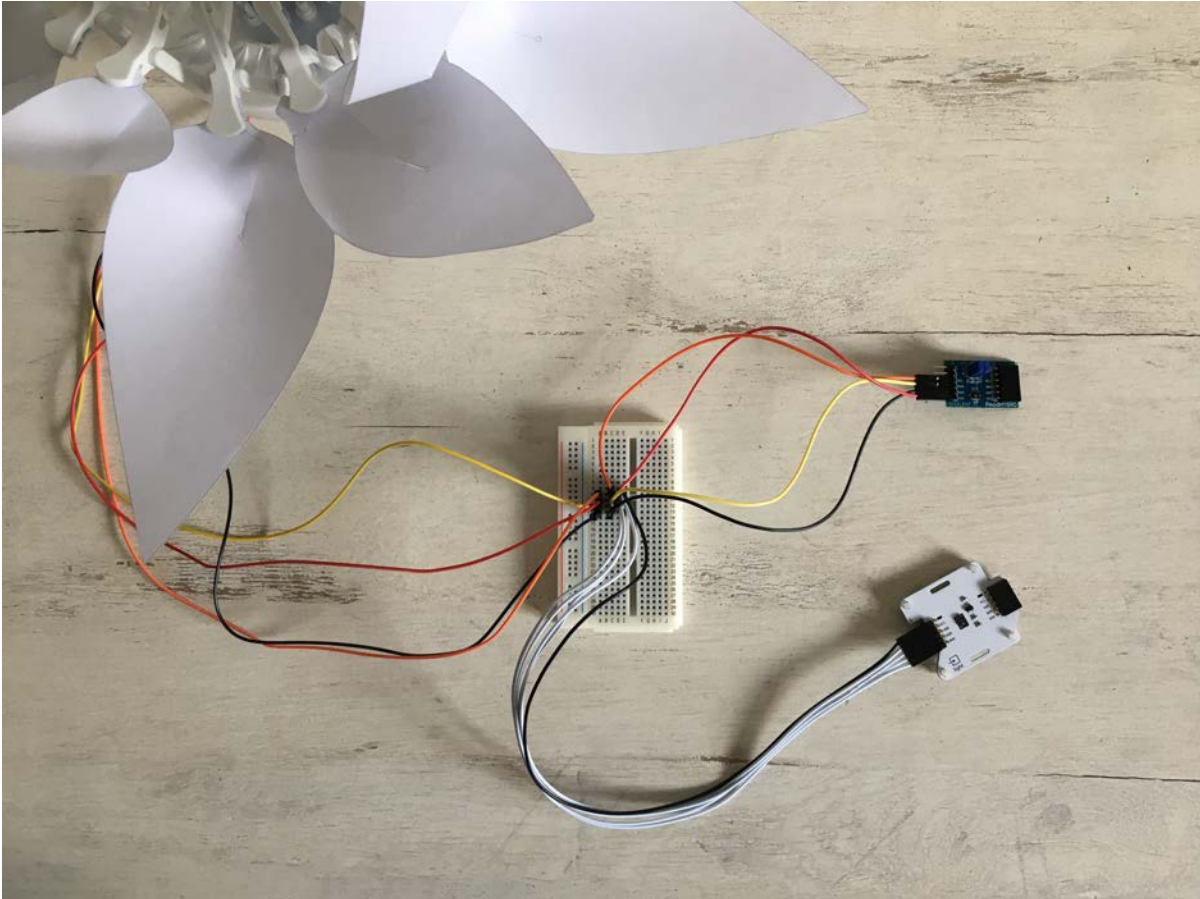
** solange es hell genug ist (Morgen- oder Tageshelligkeit), ist die Blume geöffnet. Die Farbe der LEDs ist abhängig vom Helligkeitwert.

** Ansonsten leuchten die LEDs in der "Nachtfarbe" und die Blume ist geschlossen.

Aufgabe 4: Integriere den Temperatursensor

Der Temperatursensor ermöglicht die Messung der Temperatur. Dieser Sensor kommuniziert über das I2C-Protokoll, er nutzt also die Pins SCL und SDA. Seine I2C-Adresse ist 0x40.

In dieser Aufgabe misst du die Temperatur und vergleichst sie mit der Starttemperatur. Damit wirst du den Öffnungsgrad der Blume ermitteln.



Verdrahtungsplan:

Temperatursensor	ESP32
SCL	GPIO 5
SDA	GPIO 4
(+)	5 V
(-)	GND

Code:

1. Bibliothek

Füg die Bibliothek zur Integration des Temperatursensors hinzu.

```
// temperature's sensor
#include "ClosedCube_HDC1080.h"
```

2. Globale Variablen

- ** Definiere die I2C-Adresse des Temperatursensors.
- ** Definiere eine globale Variable für den Temperaturwert.
- ** Definiere eine globale Variable zum Speichern der Starttemperatur (entspricht einer Referenztemperatur).
- ** Definiere eine globale Variable für den Öffnungsgrad.
- ** Erstelle das Objekt für den Temperatursensor.

```
//Temperature sensor
ClosedCube_HDC1080 hdc1080;
```

3. setup()

- ** Initialisiere den Temperatursensor.

```
//Temperature sensor initialisation
hdc1080.begin(ADRESS_TEMP);
```

- ** Miss die Umgebungstemperatur.

```
//Read the ambient temperature
Wire.beginTransmission(ADRESS_TEMP);
start_temperature=hdc1080.readTemperature();
Wire.endTransmission();
```

4. loop() (Schleife)

- ** Lies den Wert des Lichtsensors aus.

```
//Read the temperature value
Wire.beginTransmission(ADRESS_TEMP);
temperature=hdc1080.readTemperature();
Serial.print("Temperature = ");
Serial.print(temperature);
Serial.println(" °C");
Wire.endTransmission();
```

- ** Schreiben Sie eine if-Schleife
- ** wenn die Temperatur über dem Starttemperaturwert +10 liegt, dann ist der maximale Öffnungsgrad erreicht, sprich120.
- ** ansonsten wird der Öffnungsgrad durch die nächste Operation berechnet:

```
degree_opening =((temperature-start_temperature)*10)+20;
```

Du kannst den Öffnungsgrad anzeigen lassen, um das Ergebnis zu sehen.

Aufgabe 5: Das ganze Szenario

Erstelle nun den endgültigen Code, um das Szenario nachzubilden :

- ** Die Farbe der Blume hängt von der Helligkeit ab, es gibt 3 Lichtstufen: Morgen, Tag und Nacht.
- ** Die Blume öffnet sich, wenn es Morgen oder der Tag ist. Ansonsten schließt sich die Blüte.
- ** Der Grad der Öffnung hängt von der Temperatur ab. Wenn der Öffnungsgrad der Blume nicht der jeweiligen Raumtemperatur entspricht, schließt oder öffnet sich die Blume, bis sie den entsprechenden Öffnungsgrad erreicht hat.