

# Automatic Cat Feeder

Happy cat Happy Life





🐾 **Introduction**

🐾 **A look inside**

🐾 **Coding**

🐾 **Demo**



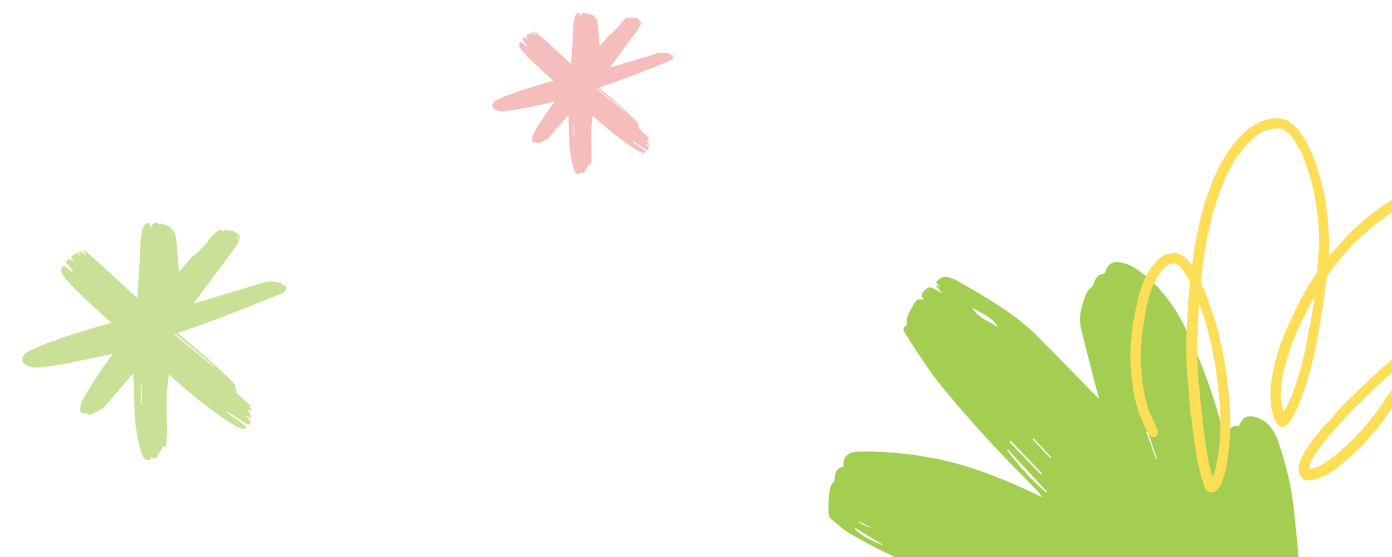


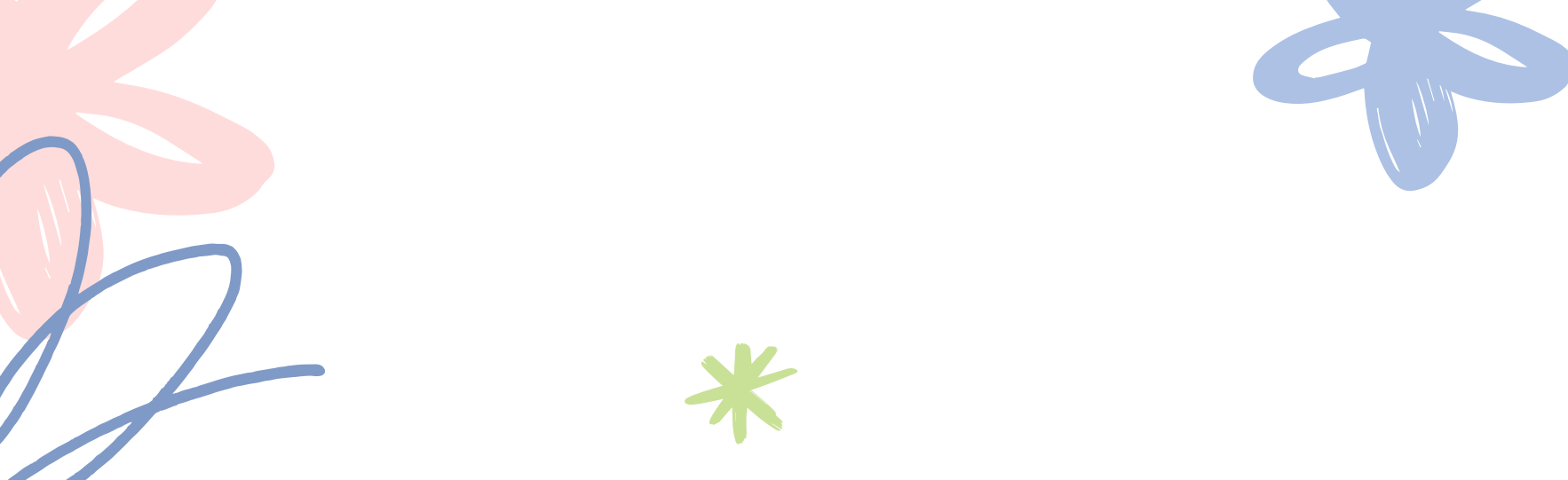
# 1. Introduction





Make mealtimes easier and more fun for you and your cat  
with one of the great cat feeders.





\*  
it detects the pet near the food bowl and allows desired quantity of food to them. Automatic cat feeders that take the work out of feeding your cat whenever it is hungry.

You can go out, sleep, shower, and farther more  
**without worrying about your cat being hungry**

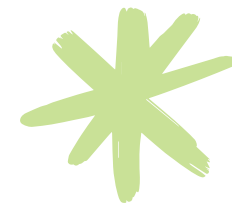




## 2. A look inside



# Components Used:



## 1- Arduino Uno

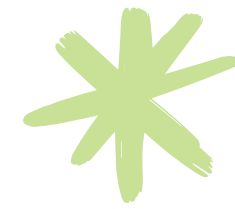
The Arduino Uno is a microcontroller board based on the ATmega328P. It contains 14 digital input/output pins, 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power connector, an ICSP header, and a reset button.

## 2-Ultrasonic Distance Sensor HC-SR04

The HC-SR04 ultrasonic sensor measures the distance to an object using sonar. This sensor has a range of 2cm to 400cm (0.8inch to 157inch) with an accuracy of 0.3cm (0.1inch), making it suitable for most hobbyist projects.



# Components Used:



## 3-Servo motor SG90

Servo motors are great devices for turning to a certain position. Usually, they have a servo arm that can turn 180 degrees. We can use the Arduino to direct a servo to a specific location, and it will do so.

## 4-Connecting Wires

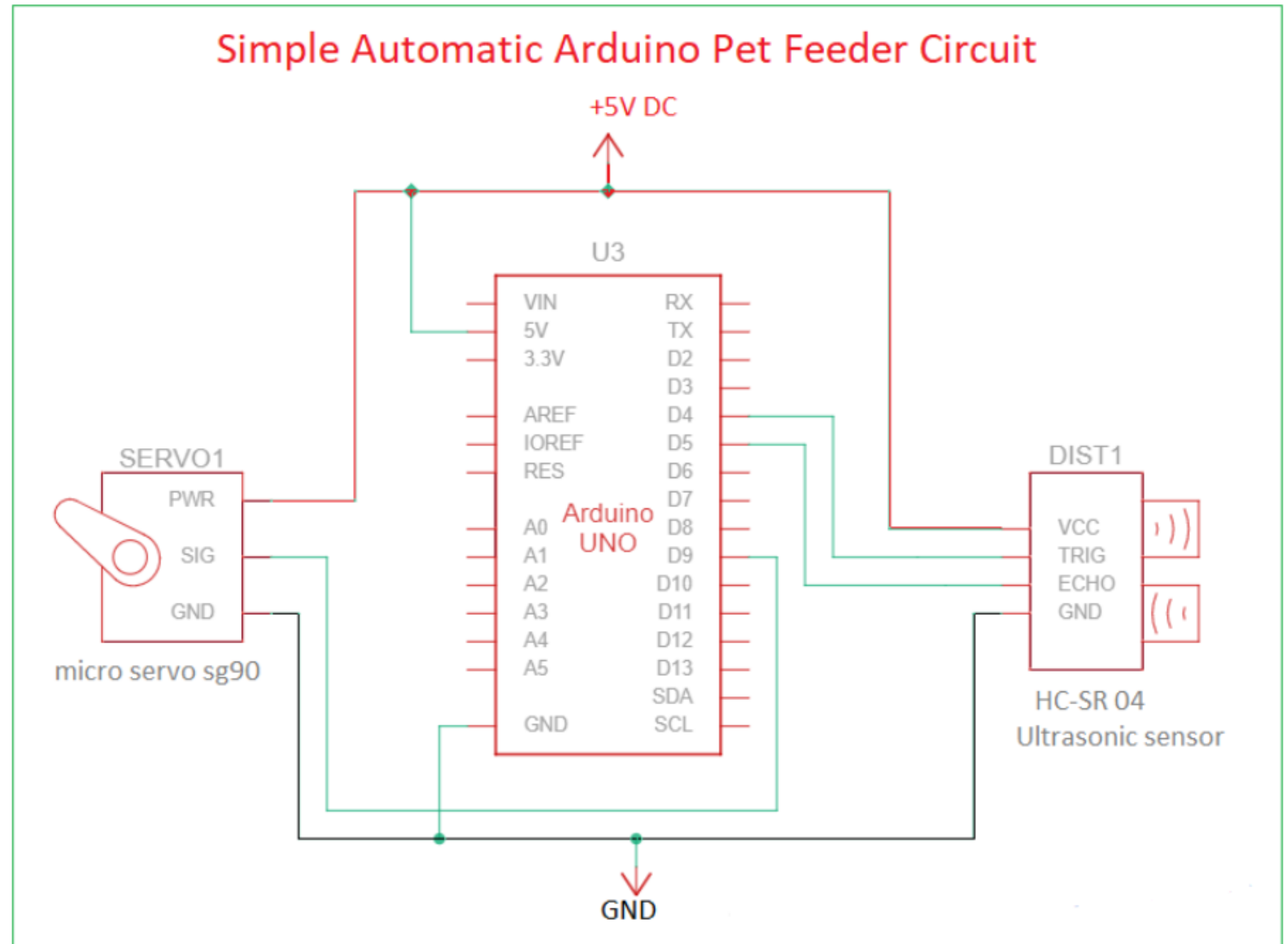
wires are used for making connections between items and your Arduino's header pins.

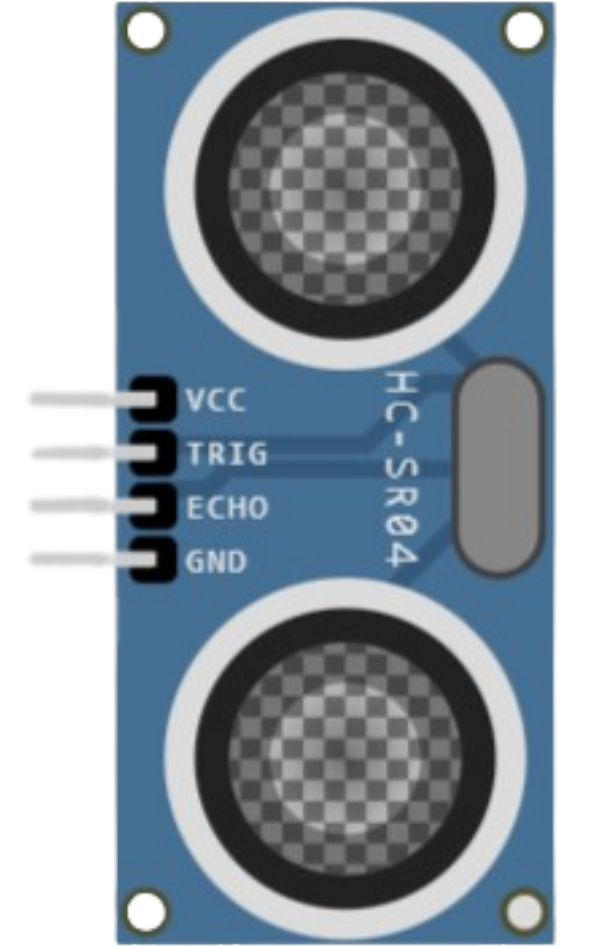
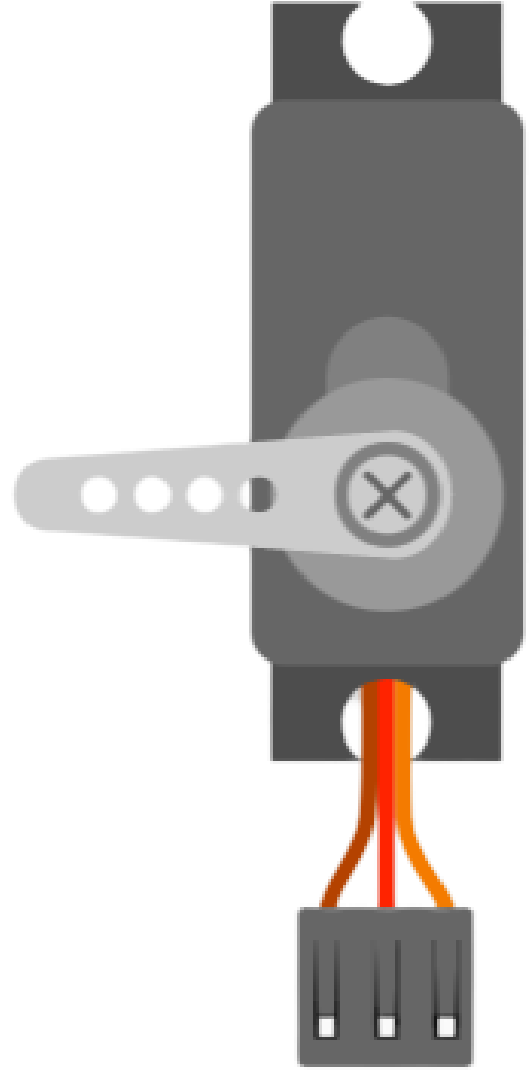
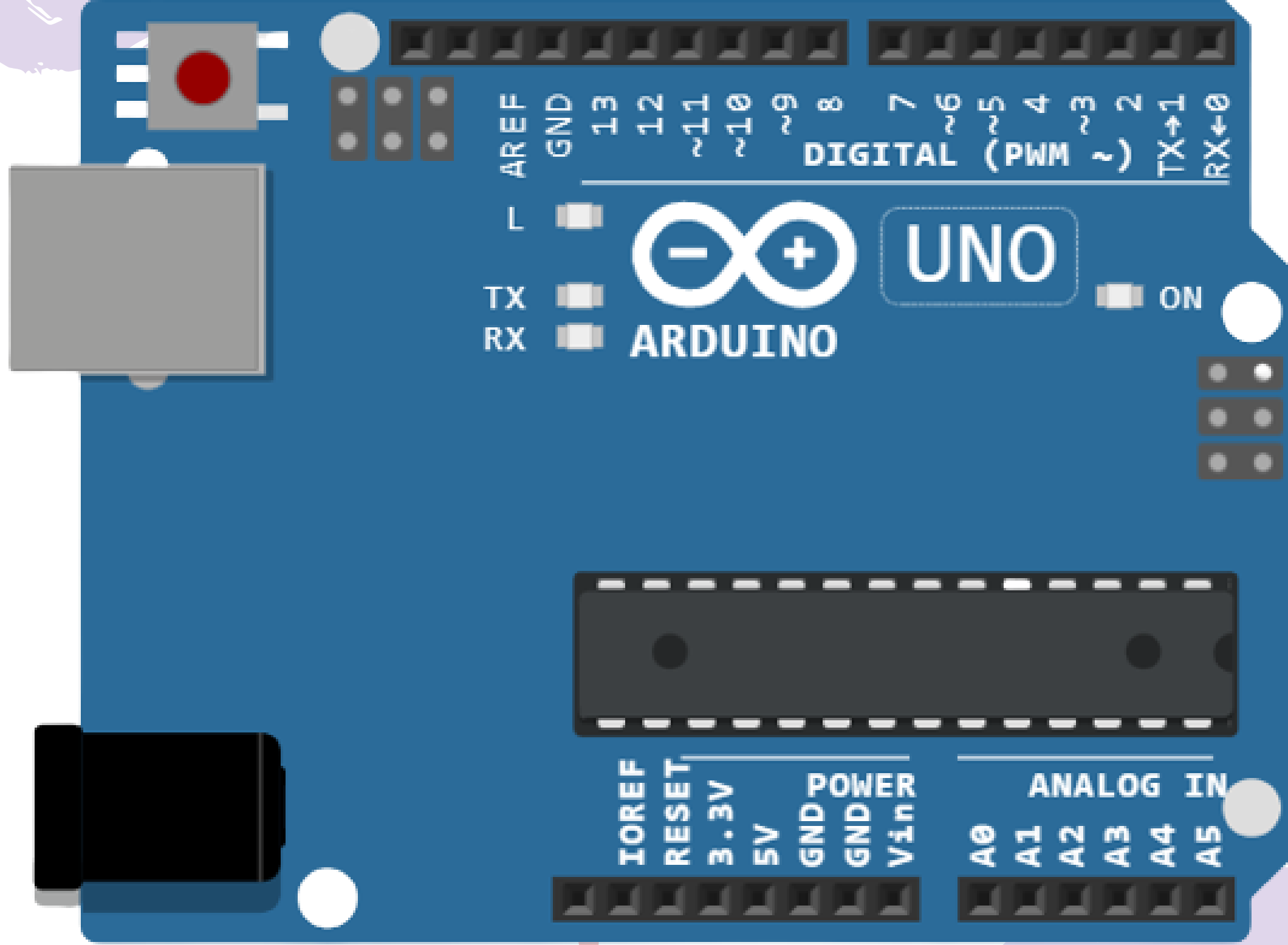
## 5-Battery as your need

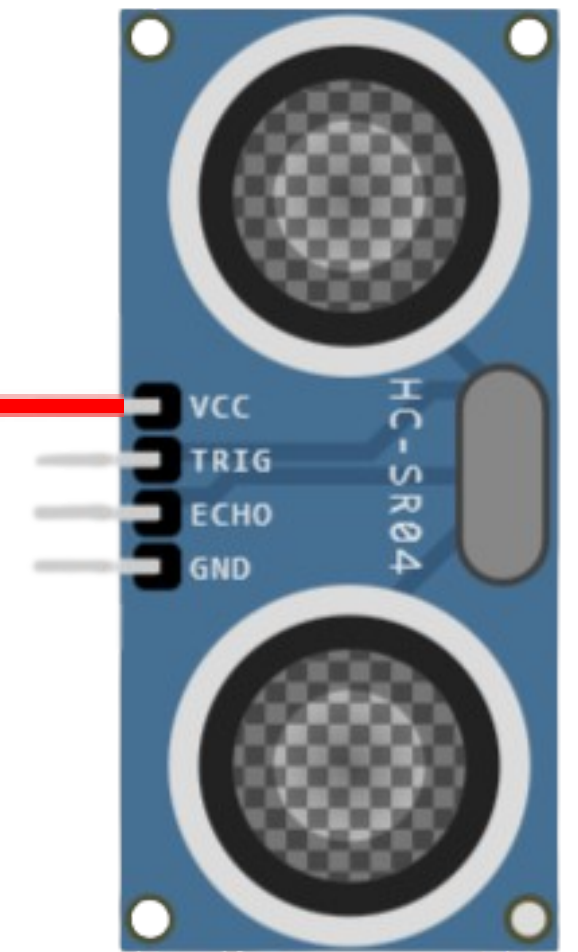
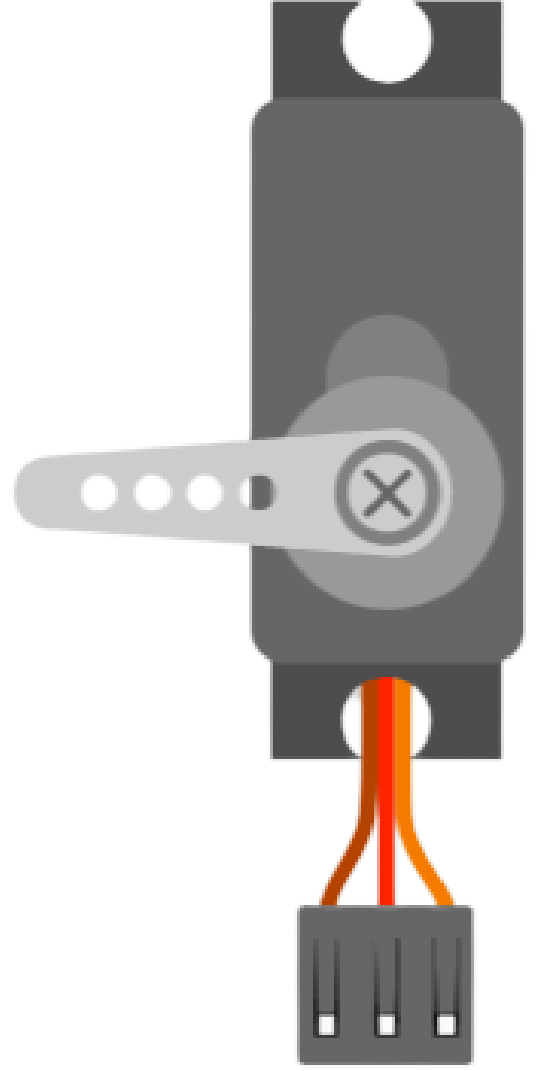
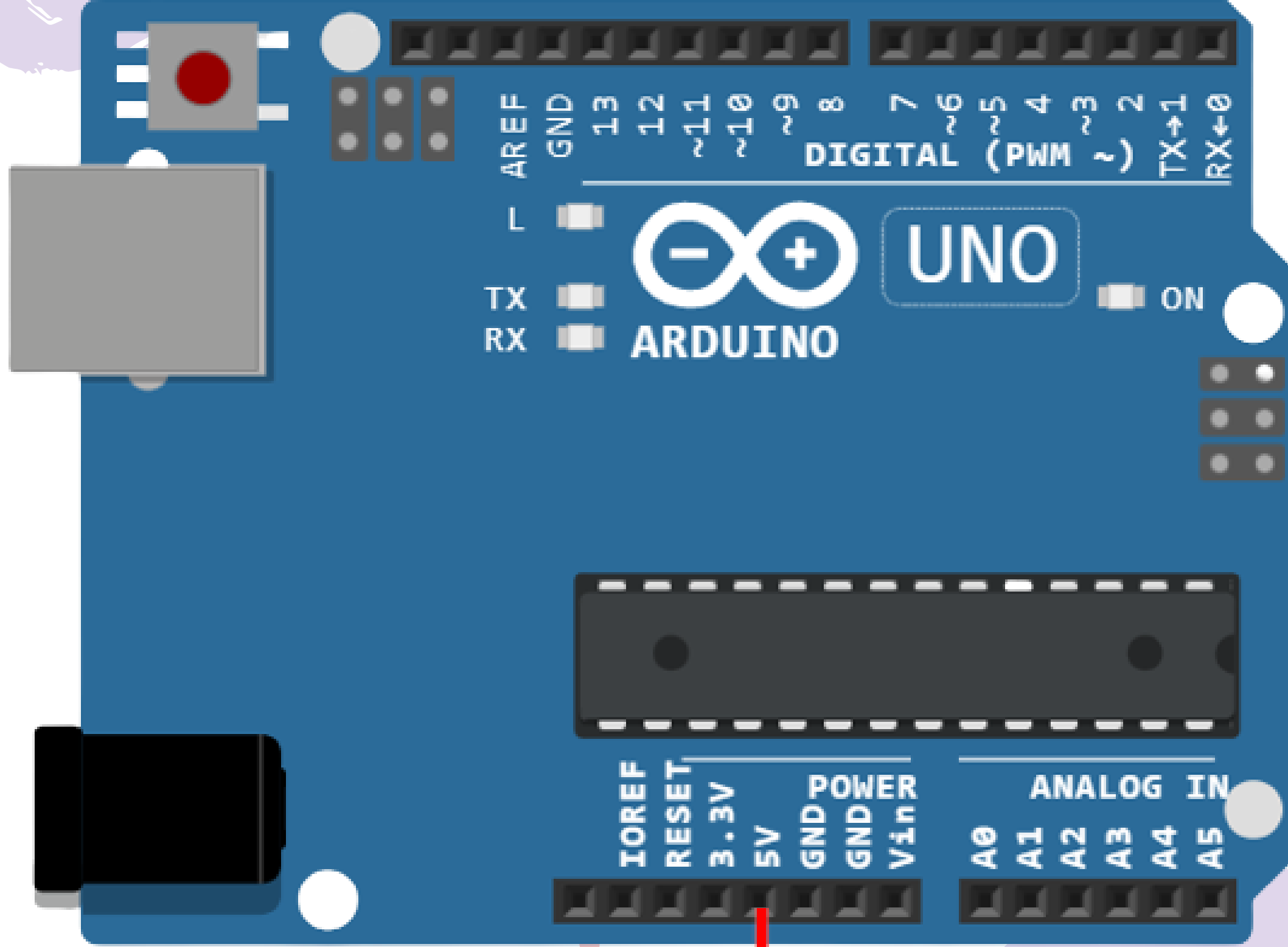


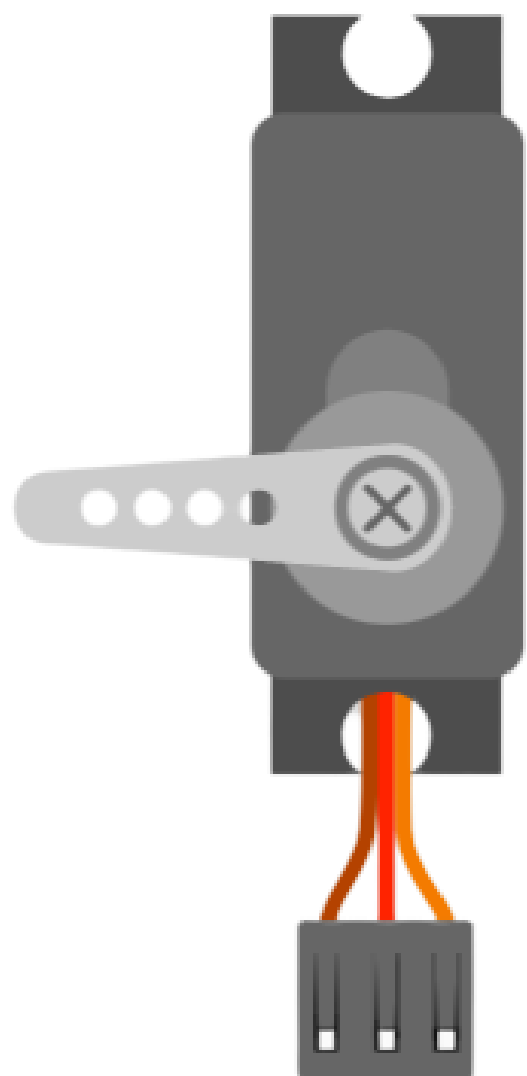
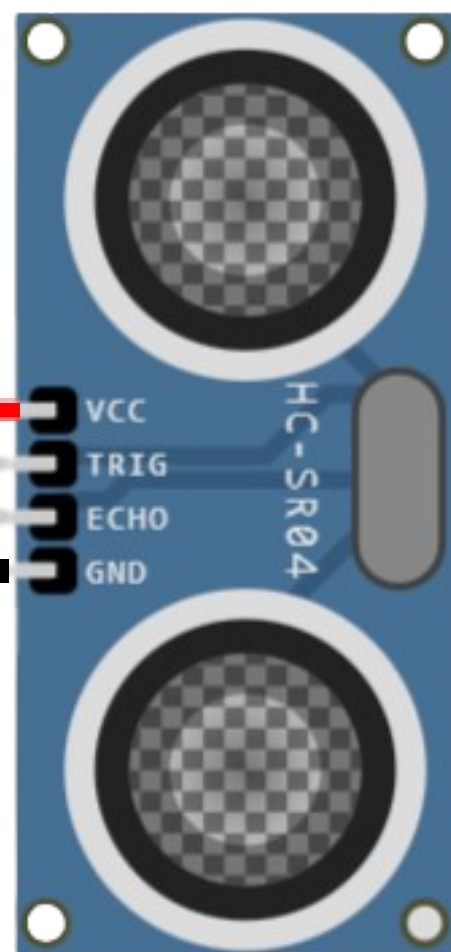
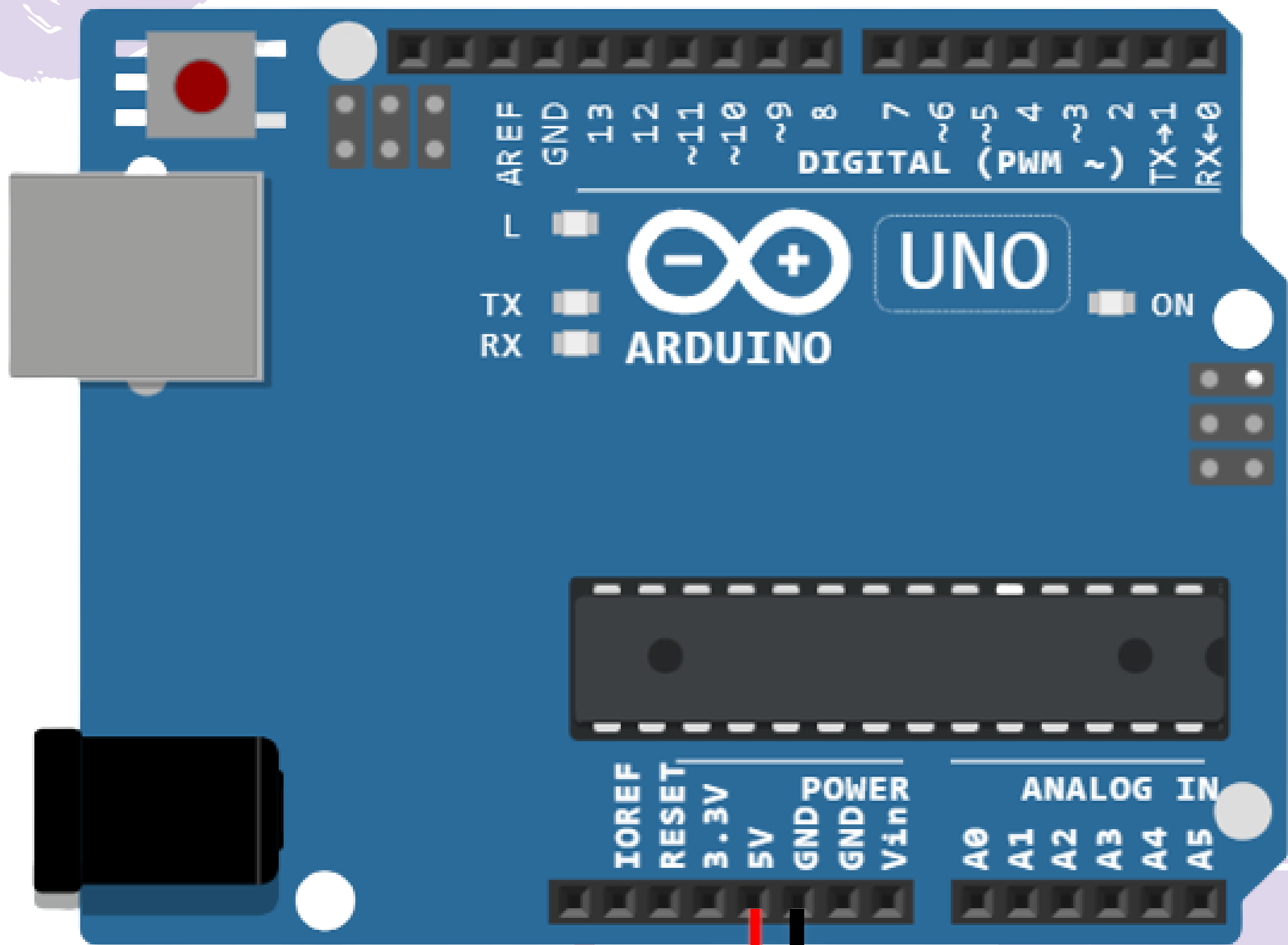


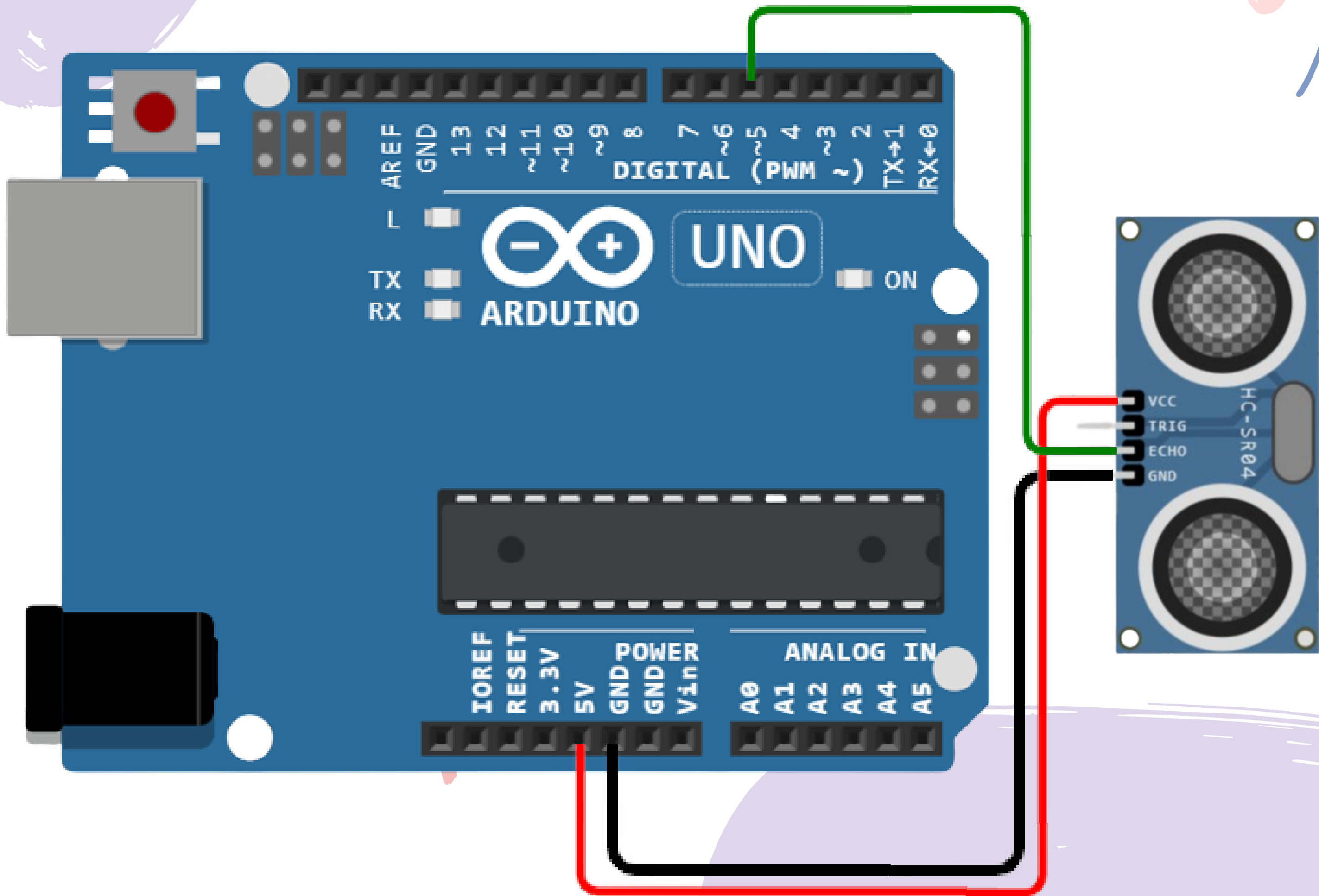
# 🐾 flowchart description

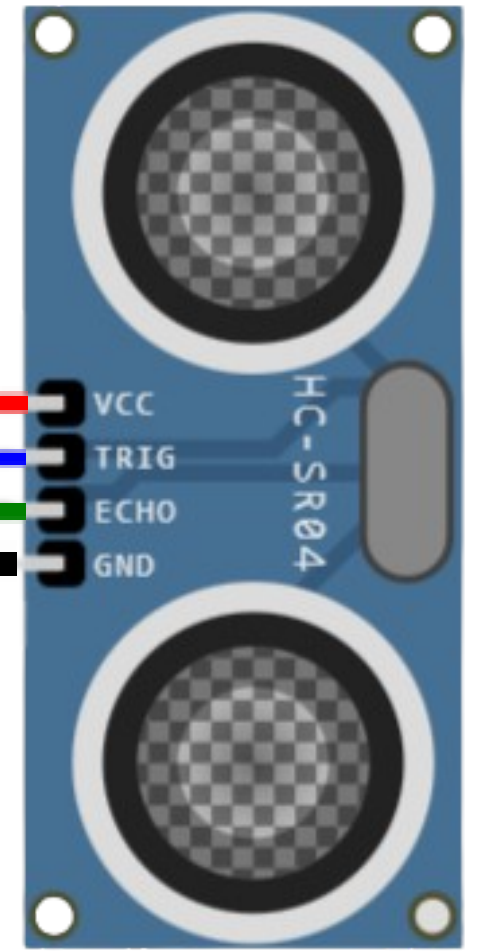
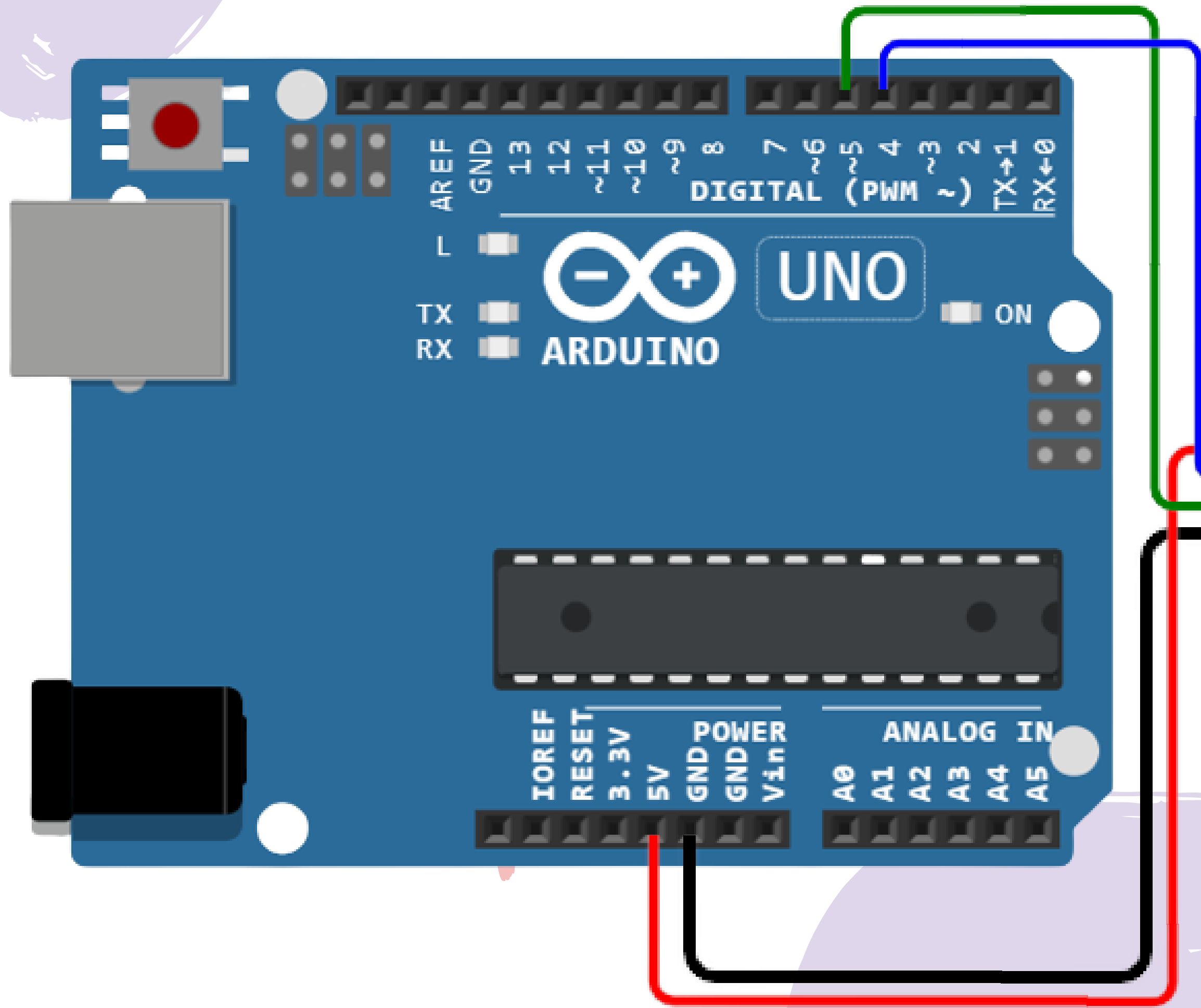
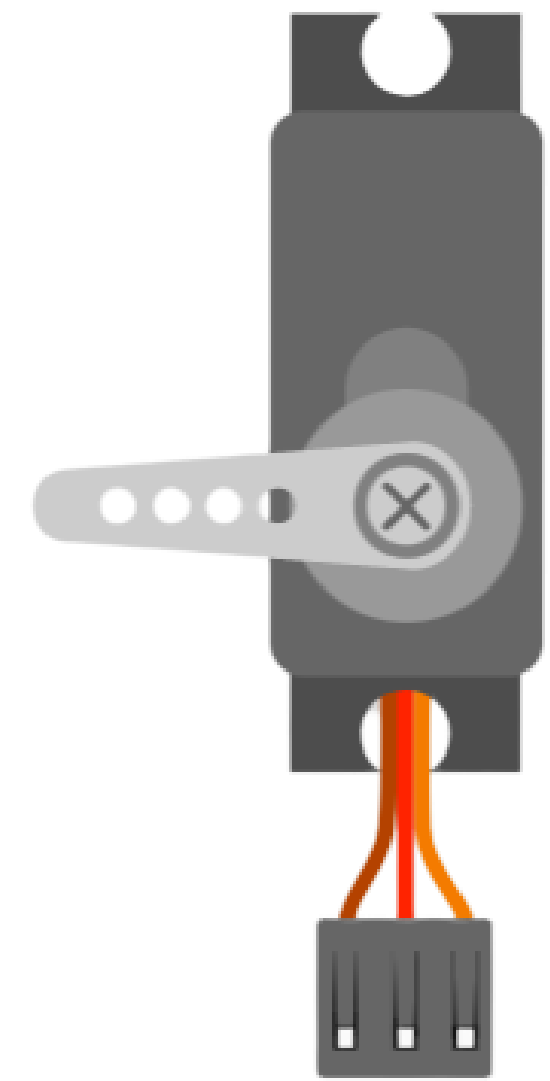


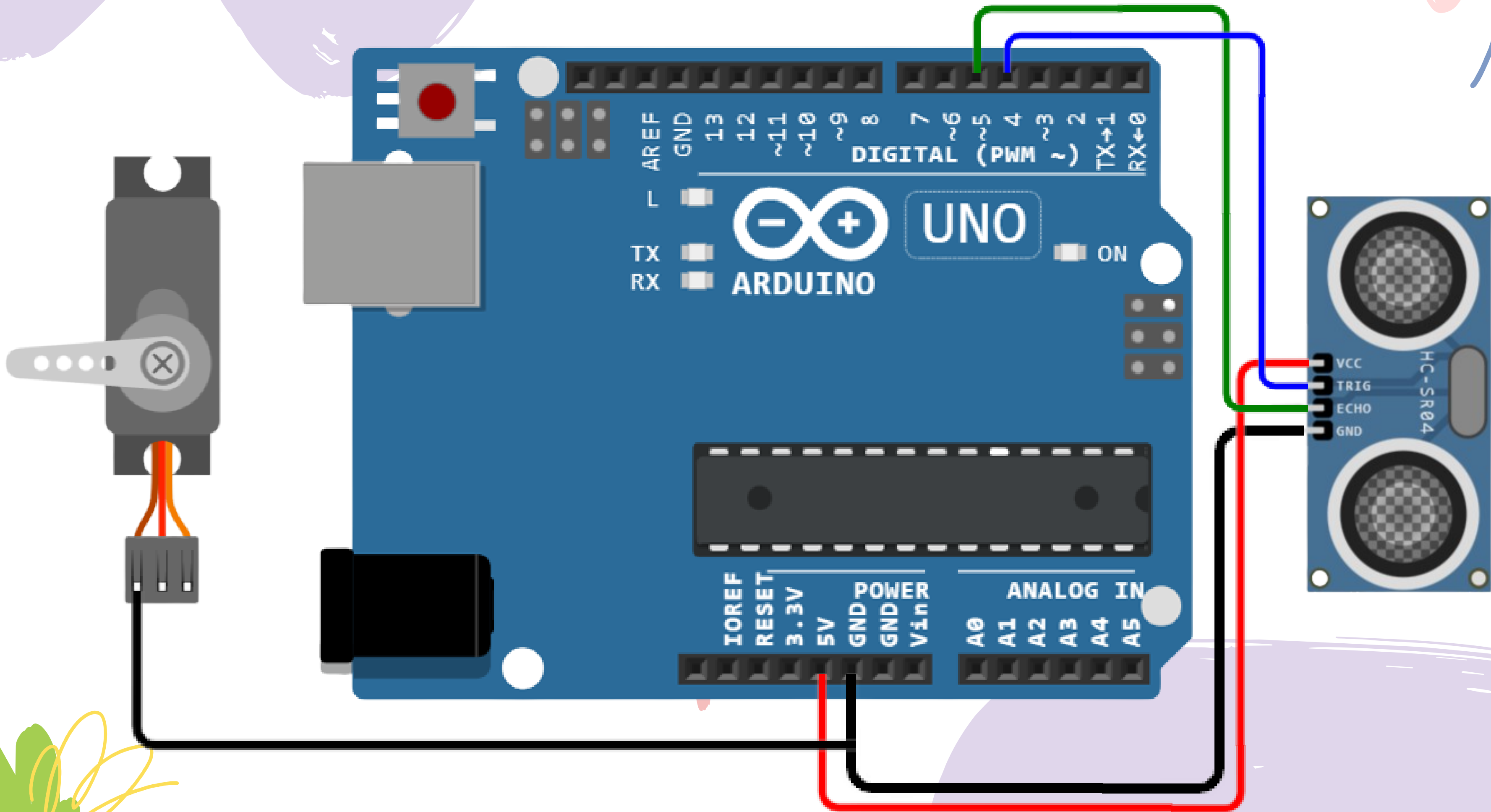




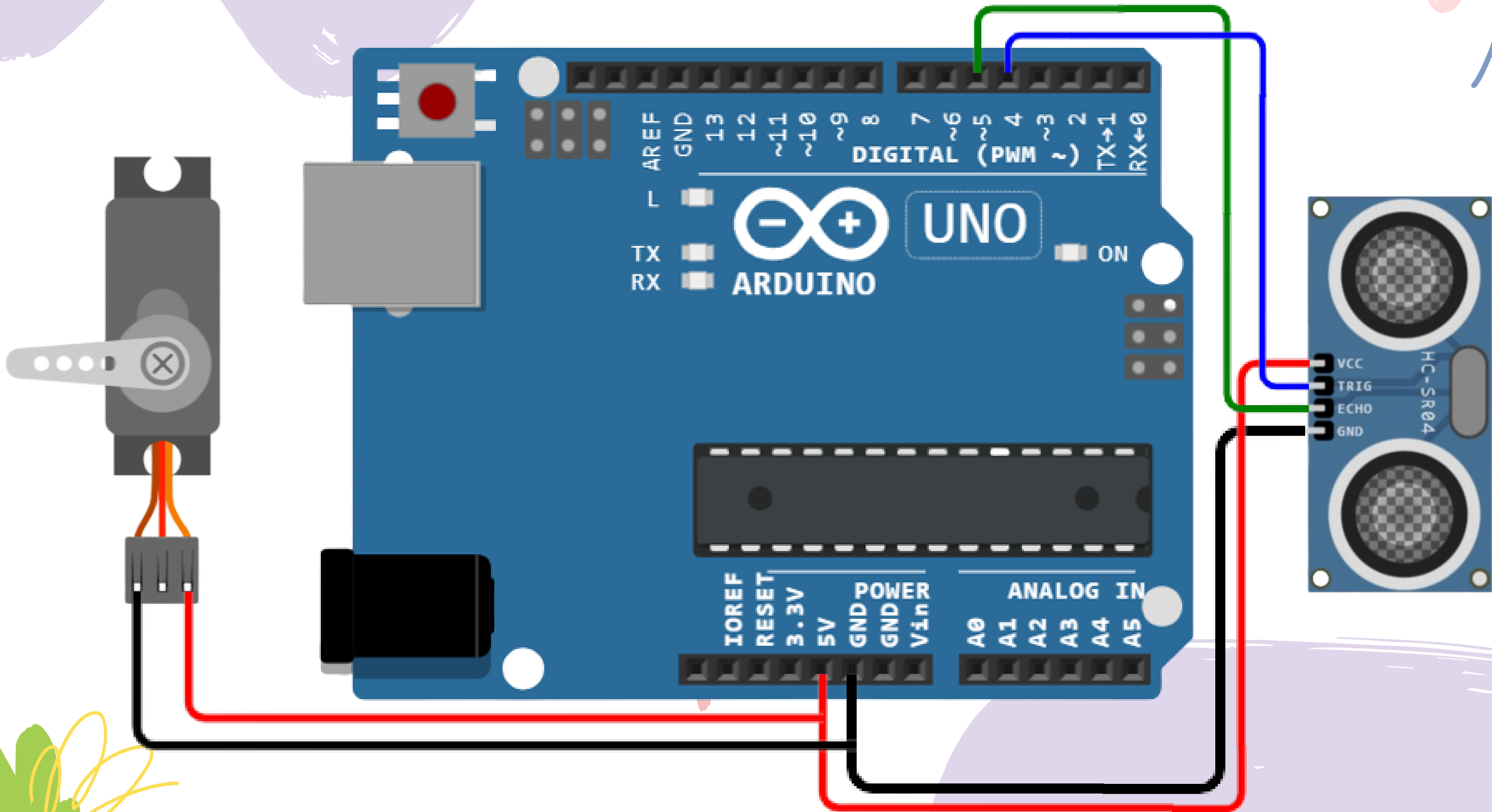




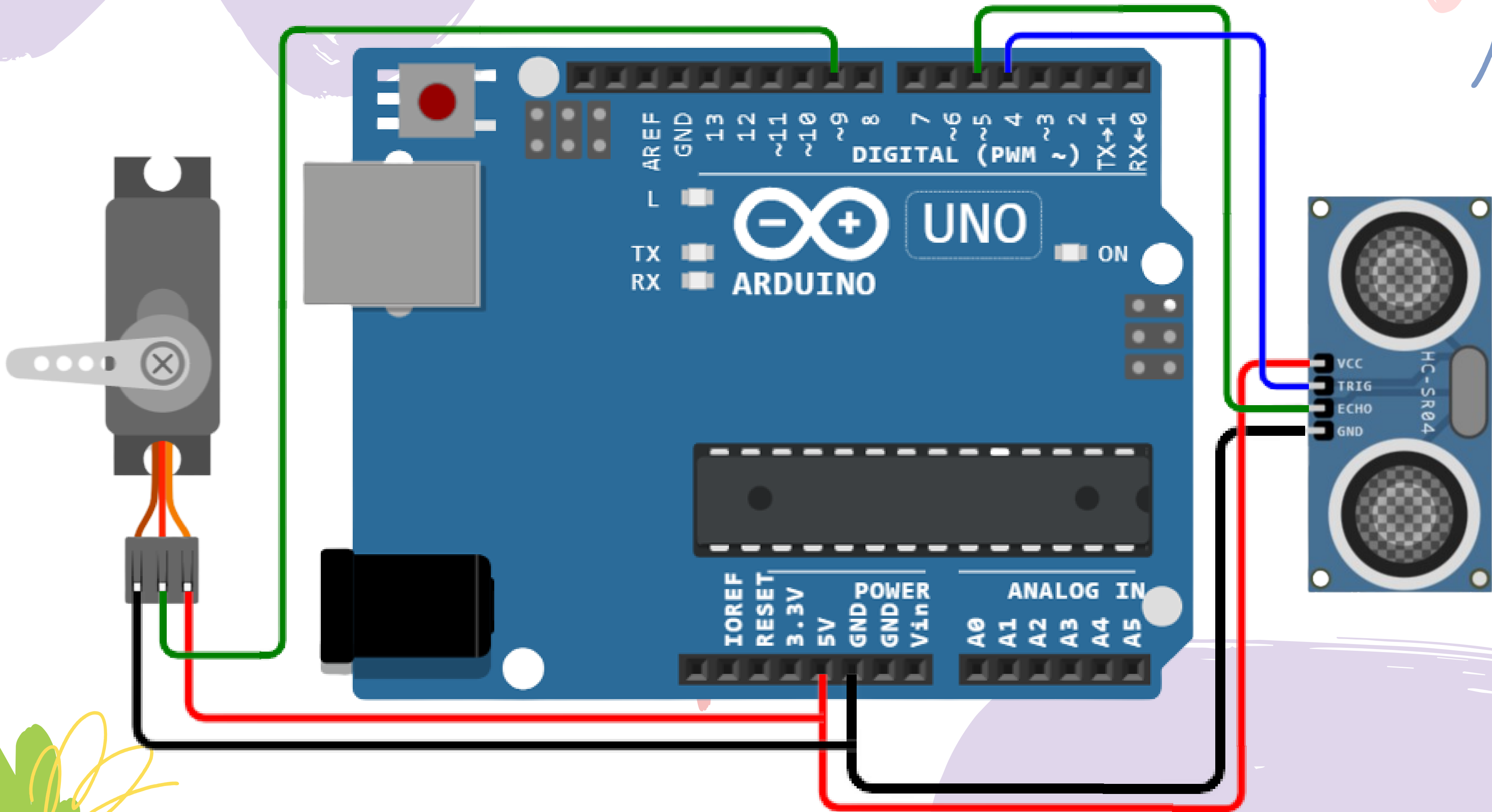








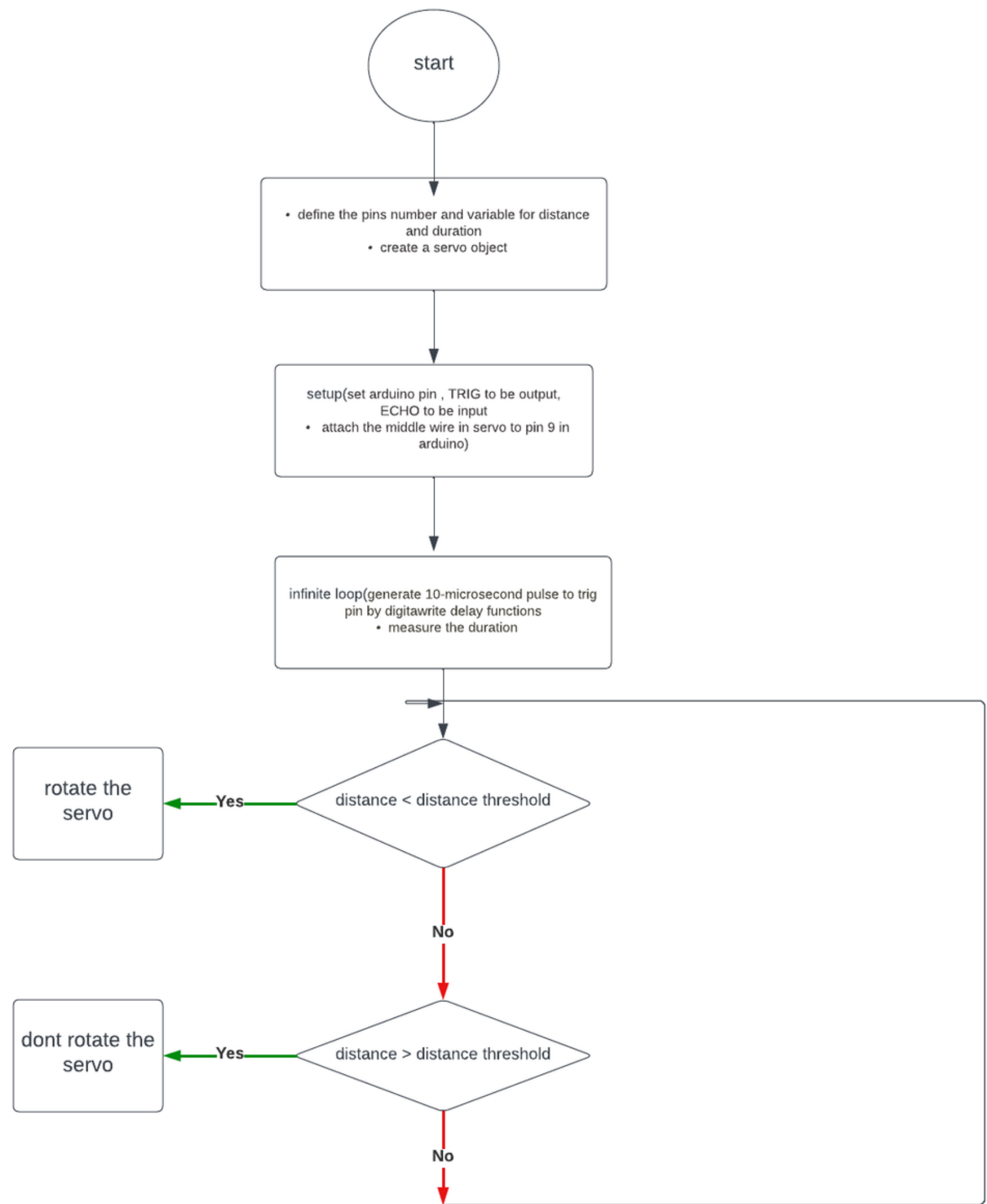





The background is white with various colorful, hand-drawn style elements. In the top left, there's a pink flower and blue scribbles. In the top right, a blue flower is on an orange brushstroke. In the middle left, a green asterisk is present. In the bottom left, a large orange brushstroke is visible. In the bottom center, a small pink asterisk is located. In the bottom right, there's a green plant-like shape with yellow loops. The word 'Code' is written in a bold, black, rounded font, with a black paw print replacing the letter 'o'.

Code

# 🐾 Flowchart of code





# C de Explanation

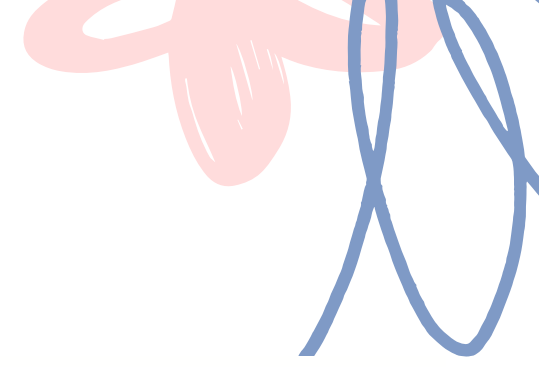
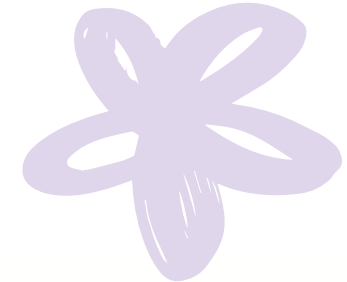

- 
- first include servo libraire
  - define the pins number as constant and threshold of distance
  - create a servo object
  - define a variable for distance and duration

## **in Setup function**

- set arduino pin , TRIG to be output, ECHO to be input
- attach the middle wire in servo to pin 9 in arduino

## **in Loop function**

- generate 10-microsecond pulse to trig pin by digitalWrite delay functions
  - measure the duration of pulse from ECHO pin , then calculate the distance
  - write a condition if the calclated distance less than threshold of distance rotate the servo to 90 degree to open , else do nothing
- 
- 



```
#include <Servo.h>
```

```
// constants won't change
```

```
const int TRIG_PIN = 4; // Arduino pin connected to Ultrasonic Sensor's TRIG pin
```

```
const int ECHO_PIN = 5; // Arduino pin connected to Ultrasonic Sensor's ECHO pin
```

```
const int SERVO_PIN = 9; // Arduino pin connected to Servo Motor's pin
```




```
const int DISTANCE_THRESHOLD = 50; // centimeters
```

```
Servo servo; // create servo object to control a servo
```




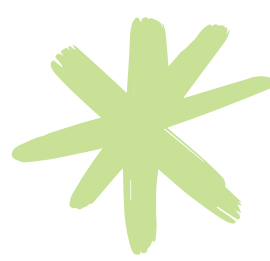

```
// variables will change:
```

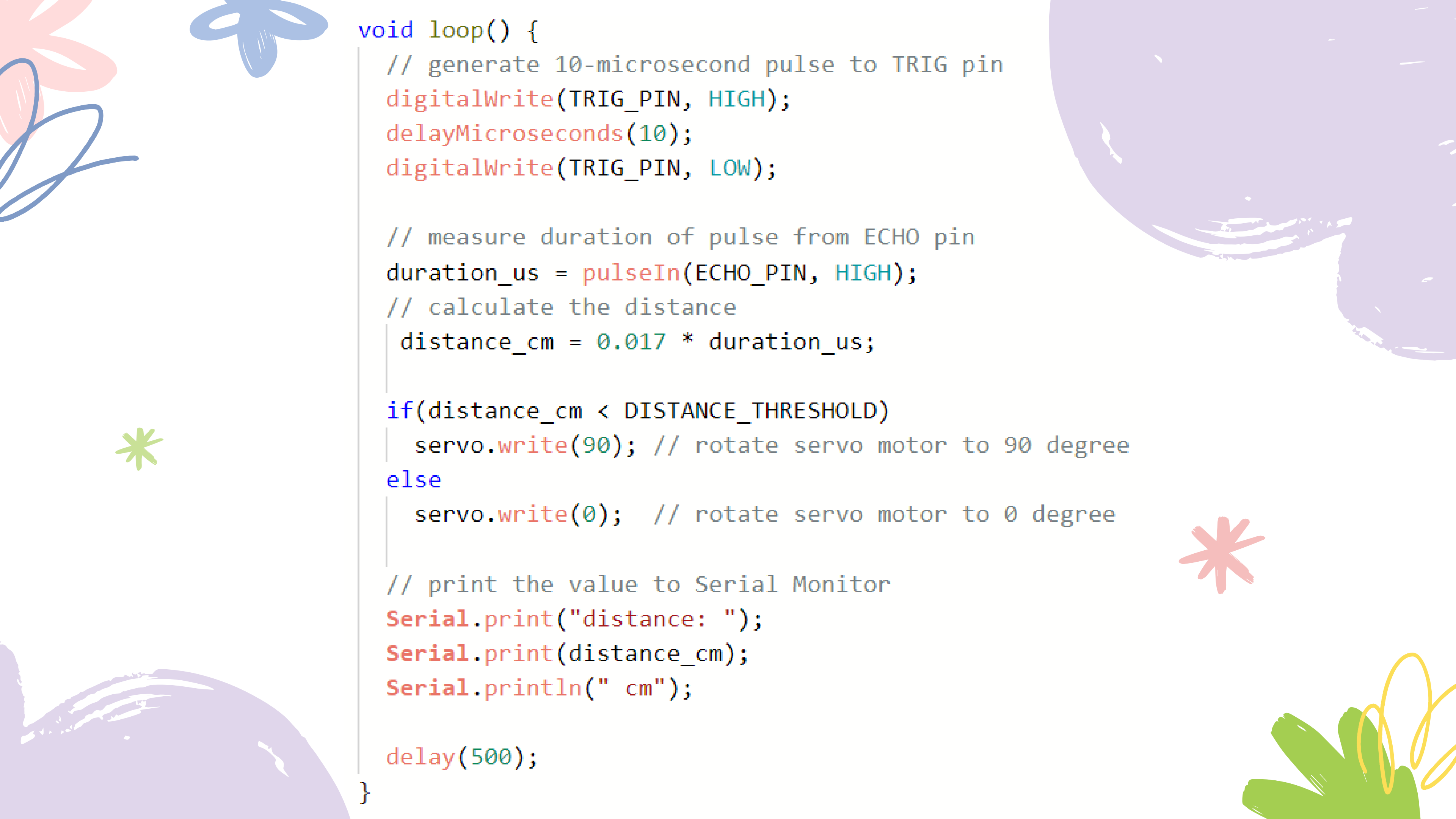
```
float duration_us, distance_cm;
```





```
void setup() {  
  Serial.begin (9600);           // initialize serial port  
  pinMode(TRIG_PIN, OUTPUT);     // set arduino pin to output mode  
  pinMode(ECHO_PIN, INPUT);      // set arduino pin to input mode  
  servo.attach(SERVO_PIN);       // attaches the servo on pin 9 to the servo object  
  servo.write(0);  
}
```





```
void loop() {  
  // generate 10-microsecond pulse to TRIG pin  
  digitalWrite(TRIG_PIN, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(TRIG_PIN, LOW);  
  
  // measure duration of pulse from ECHO pin  
  duration_us = pulseIn(ECHO_PIN, HIGH);  
  // calculate the distance  
  distance_cm = 0.017 * duration_us;  
  
  if(distance_cm < DISTANCE_THRESHOLD)  
    servo.write(90); // rotate servo motor to 90 degree  
  else  
    servo.write(0); // rotate servo motor to 0 degree  
  
  // print the value to Serial Monitor  
  Serial.print("distance: ");  
  Serial.print(distance_cm);  
  Serial.println(" cm");  
  
  delay(500);  
}
```



Dem🐾







# Team Members




**Bedoor Almareni**  
**2005961**

**Nouf Almontashry**  
**1915171**







**Raneem Alomari**  
**2006352**

**Taqia Rashwan**  
**1913088**



***Instructors:***  
***Eng. Maram Alqathmi***  
***DR. Suad***



# reference

- <https://arduinogetstarted.com/tutorials/arduino-ultrasonic-sensor>
- <https://docs.arduino.cc/learn/electronics/servo-motors>

Thank you

