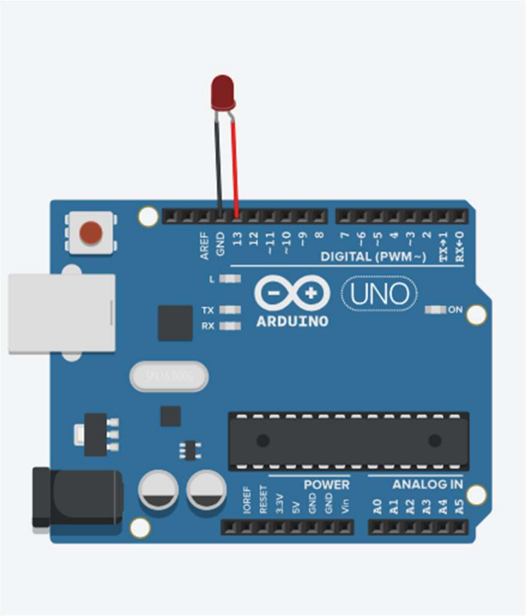
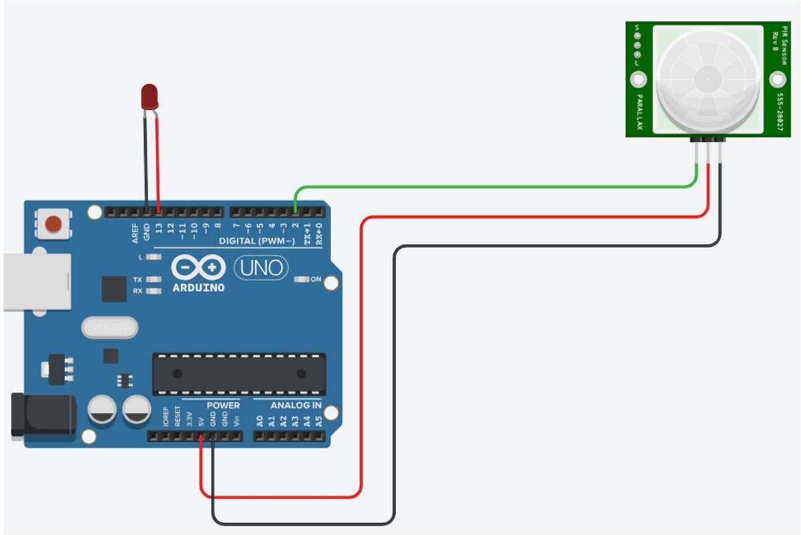
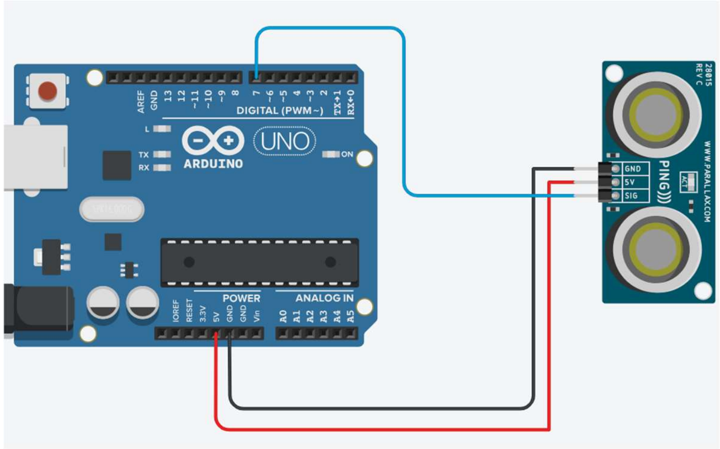


```
void setup() {
  pinMode(13, OUTPUT); // Set pin 13 as an OUTPUT
}

void loop() {
  digitalWrite(13, HIGH); // Turn the LED ON
  delay(1000);            // Wait for 1 second
  digitalWrite(13, LOW);  // Turn the LED OFF
  delay(1000);            // Wait for 1 second
}
```

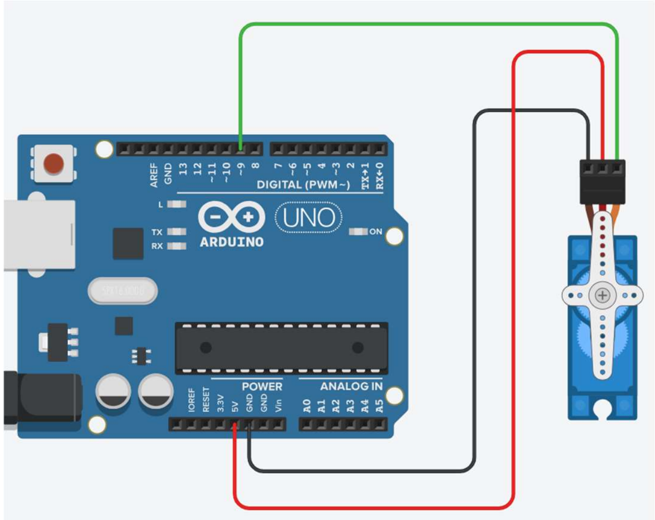
```
}
```



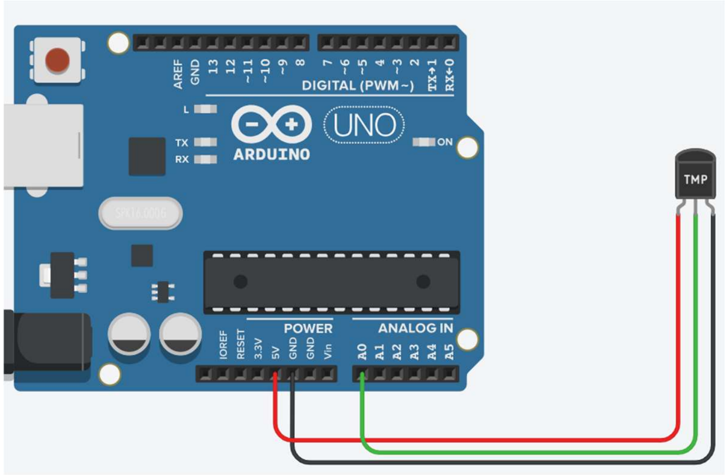
```
// Define pins
int pirPin = 2;    // PIR sensor is connected to digital pin 2
int ledPin = 13;   // LED is connected to digital pin 13

void setup() {
  pinMode(pirPin, INPUT);  // Set PIR sensor pin as input
  pinMode(ledPin, OUTPUT); // Set LED pin as output
  Serial.begin(9600);      // Initialize serial monitor for debugging
}
void loop() {
  int motionDetected = digitalRead(pirPin); // Read PIR sensor status

  if (motionDetected == HIGH) {  // If motion is detected
    digitalWrite(ledPin, HIGH); // Turn on LED
    Serial.println("Motion Detected!"); // Print to Serial Monitor
  } else {
    digitalWrite(ledPin, LOW);  // Turn off LED
  }

  delay(100); // Small delay for stability
}
```



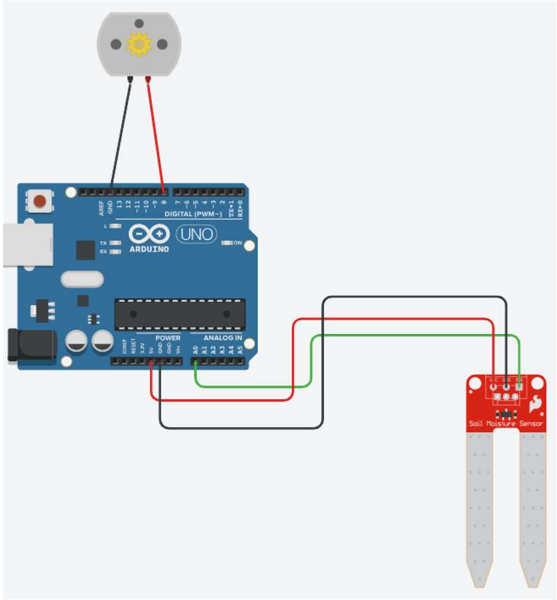
```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  pinMode(7,OUTPUT);
  long duration;
  long distance;
  digitalWrite(7,LOW);
  delayMicroseconds(2);
  digitalWrite(7,HIGH);
  delayMicroseconds(10);
  digitalWrite(7,LOW);
  //set sensorPin as input to read
  pinMode(7,INPUT);
  duration = pulseIn(7,HIGH);//reading echo
  distance = duration*0.034/2;
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(1000);
}
```



```
#include <Servo.h>  // Include the Servo library

Servo myServo;  // Create a Servo object

void setup() {

  myServo.attach(9);  // Attach the servo to pin 9

}

void loop() {

  // Turn 180° clockwise

  myServo.write(180);  // Move to 180 degrees

  delay(3000);        // Wait for 1 second



  // Turn 180° counterclockwise

  myServo.write(0);    // Move back to 0 degrees

  delay(3000);        // Wait for 1 second

}
```



```
float temp;        // Variable to store the analog sensor value
int tempPin = A0;   // TMP36 sensor connected to analog pin A0
float voltage;      // Variable to store the voltage
float temperatureC; // Temperature in Celsius
void setup() {
  Serial.begin(9600); // Initialize Serial Communication
}
void loop() {
  // Read the analog voltage from the sensor (0-1023)
  temp = analogRead(tempPin);
  // Convert the analog reading to voltage (assuming a 5V reference)
  voltage = temp * (5.0 / 1024.0);
  // Calculate temperature in Celsius
  temperatureC = (voltage - 0.5) * 100.0;

  // Display the results
  Serial.print("The temperature is: ");
  Serial.print(temperatureC);
  Serial.println(" *C");

  delay(1000); // Wait 1 second before updating
}
```



```
// Define pins
const int moisturePin = A0;  // Soil moisture sensor analog pin
const int Motor = 8;    // SPDT relay control pin
int moistureValue = 0;       // Variable to store soil moisture value
void setup() {
  pinMode(Motor, OUTPUT);  // Set relay pin as output
  digitalWrite(Motor, LOW); // Turn off relay initially (motor off)
  Serial.begin(9600);          // Start serial communication
}
void loop() {
  moistureValue = analogRead(moisturePin);  // Read soil moisture sensor value
  Serial.print("Moisture Value: ");
  Serial.println(moistureValue);

  // Adjust the threshold value based on soil moisture calibration
  if (moistureValue < 500) {
    // Soil is dry: Turn on the motor
    digitalWrite(Motor, HIGH);  // Activate relay (motor ON)
    Serial.println("Motor ON - Soil is Dry");
  } else {
    // Soil is wet: Turn off the motor
    digitalWrite(Motor, LOW); // Deactivate relay (motor OFF)
    Serial.println("Motor OFF - Soil is Wet");
  }

  delay(1000);  // Wait for 1 second
}
```