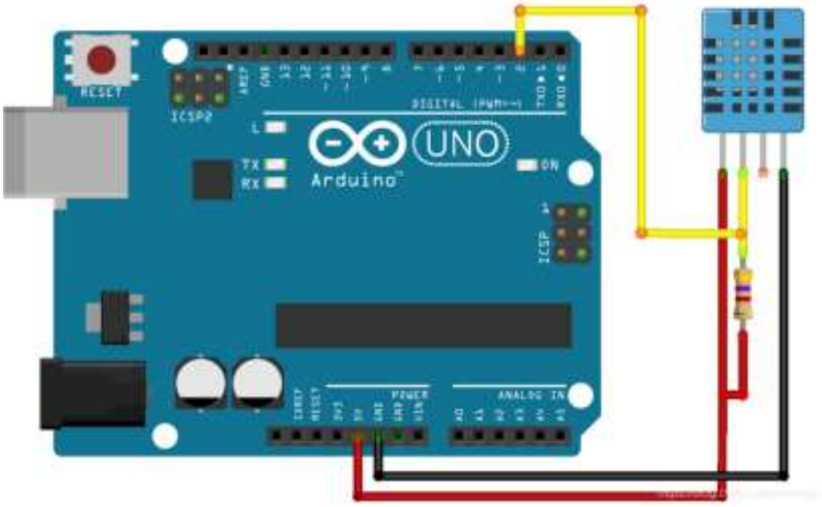
1. Microcontroller Experiments using Arduino or MSP430:
   Sensor based Experiments
   a. Temperature and humidity Sensor.
   b. HEART BEAT SENSOR.

a. Tempaerature and humidity Sensor

Components Required:

Arduino UNO – Read Best Arduino Starter Kits
DHT11 Temperature and Humidity Sensor
Breadboard
10k Ohm Resistor
Jumper Wires

Pin Diagram:



Program:

```
#include "DHT.h"
#define DHTPIN 2       // Pin connected to DATA
#define DHTTYPE DHT11  // or DHT22
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature(); // Celsius
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" *C");
  delay(2000);
}
```

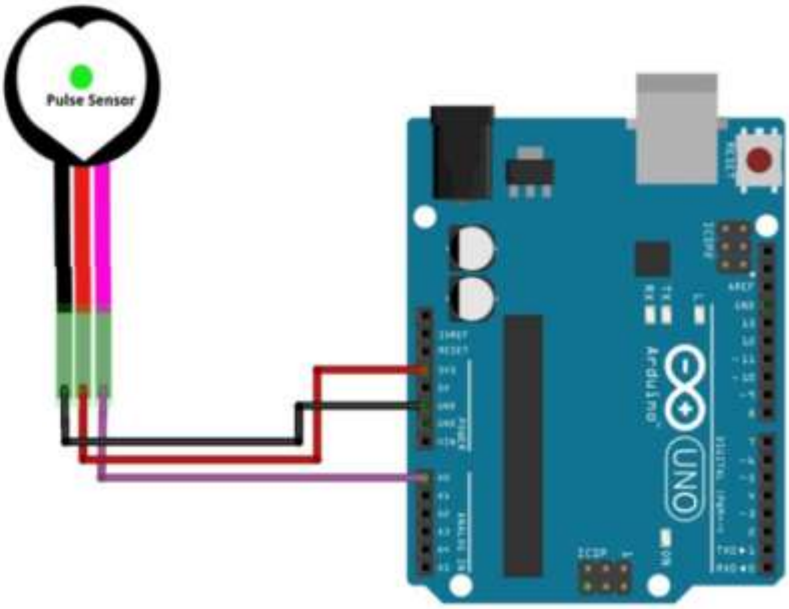
Output:

b. Heart Beat Sensor

Components Required:

Heart Beat Sensor
Arduino UNO – Read Best Arduino Starter Kits
Breadboard
Jumper Wires

Pin diagram:



Program

```
#define USE_ARDUINO_INTERRUPTS true
// Include necessary libraries
#include <PulseSensorPlayground.h>

// Constants
const int PULSE_SENSOR_PIN = 0;  // Analog PIN where the PulseSensor is connected
const int LED_PIN = 13;          // On-board LED PIN
const int THRESHOLD = 550;       // Threshold for detecting a heartbeat

// Create PulseSensorPlayground object
PulseSensorPlayground pulseSensor;

void setup()
{
  // Initialize Serial Monitor
  Serial.begin(9600);
  // Configure PulseSensor
  pulseSensor.analogInput(PULSE_SENSOR_PIN);
  pulseSensor.blinkOnPulse(LED_PIN);
  pulseSensor.setThreshold(THRESHOLD);

  // Check if PulseSensor is initialized
  if (pulseSensor.begin())
  {
    Serial.println("PulseSensor object created successfully!");
  }
}

void loop()
{
  // Get the current Beats Per Minute (BPM)
  int currentBPM = pulseSensor.getBeatsPerMinute();

  // Check if a heartbeat is detected
  if (pulseSensor.sawStartOfBeat())
  {
    Serial.println("A HeartBeat Happened!");
    Serial.print("BPM: ");
    Serial.println(currentBPM);
  }

  // Add a small delay to reduce CPU usage
  delay(20);
}
```

Output:

2. Microcontroller Experiments using Arduino or MSP430:
    a. Fire Sensor
    b. Big sound/small sound sensor

a. Fire Sensor

Components Required:

Arduino Uno
Breadboard
USB Cable
Sound Sensor Module
Buzzer (or small speaker)
Jumper Wires

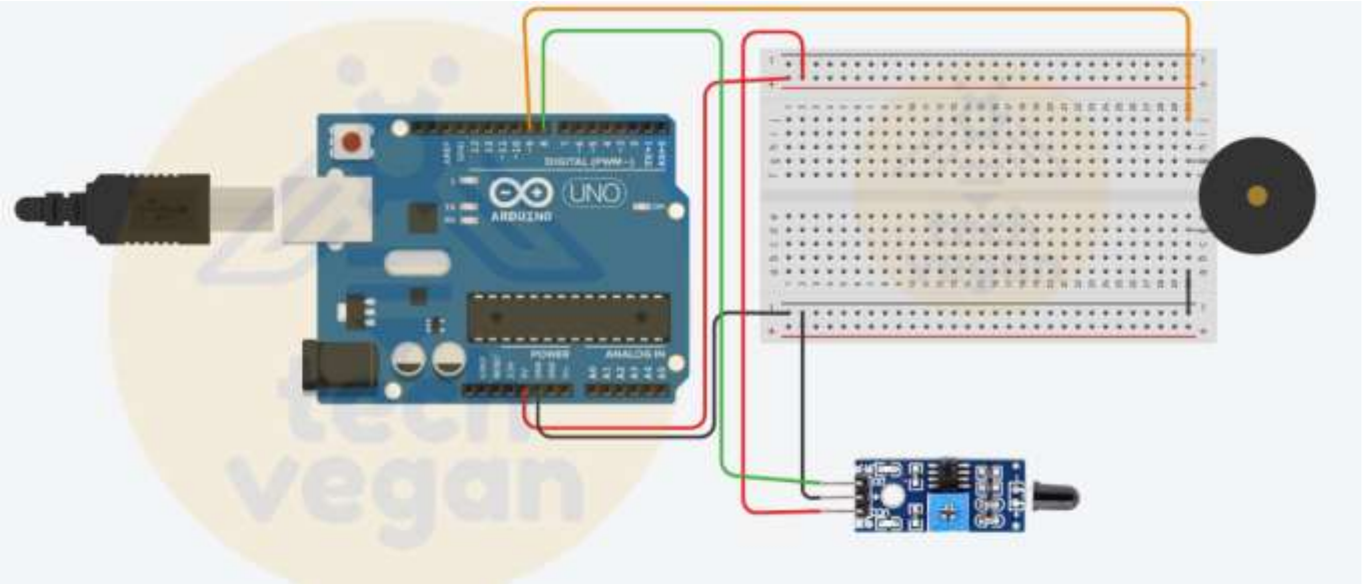
Program:

```
const int fireSensorPin = 8;

void setup()
{
  Serial.begin(9600);
  pinMode(fireSensorPin, INPUT);
}

void loop()
{
  int fireValue = digitalRead(fireSensorPin);
  Serial.println(fireValue);
  delay(500);
}
```

Pin Diagram



b. Big sound/small sound sensor

```
const int soundSensorPin = A0; // Analog pin connected to the Big Sound Sensor Module
const int ledPin = 2; // Digital pin connected to the LED (optional)

void setup() {
  pinMode(soundSensorPin, INPUT); // Set the Sound Sensor pin as INPUT
  pinMode(ledPin, OUTPUT); // Set the LED pin as OUTPUT (optional)
  Serial.begin(9600); // Initialize serial communication for debugging (optional)
}

void loop() {
  int soundValue = analogRead(soundSensorPin); // Read the analog value from the Sound Sensor

  // Display the sound sensor value on the Serial Monitor
  Serial.print("Sound Level: ");
  Serial.println(soundValue);

  // Adjust the threshold value according to your environment
  int threshold = 400;

  if (soundValue > threshold) {
    // Sound detected! Add your desired action here.
    // For example, turn on the LED as a visual indication of sound detection.
    digitalWrite(ledPin, HIGH);
  } else {
    // No sound detected, turn off the LED (optional)
    digitalWrite(ledPin, LOW);
  }

  delay(100); // Add a small delay to avoid rapid repeated detections
}
```