

Gas Sensor

int redLed = 12;

int greenLed = 11;

int buzzer = 10;

int smokeA0 = A5;

// Your threshold value

int sensorThres = 400;

void setup() {

pinMode(redLed, OUTPUT);

pinMode(greenLed, OUTPUT);

pinMode(buzzer, OUTPUT);

pinMode(smokeA0, INPUT);

Serial.begin(9600);

}

void loop() {

int analogSensor = analogRead(smokeA0);

Serial.print("Pin A0: ");

Serial.println(analogSensor);

// Checks if it has reached the threshold value

if (analogSensor > sensorThres)

{

digitalWrite(redLed, HIGH);

digitalWrite(greenLed, LOW);

tone(buzzer, 1000, 200);

}

else

{

digitalWrite(redLed, LOW);

digitalWrite(greenLed, HIGH);

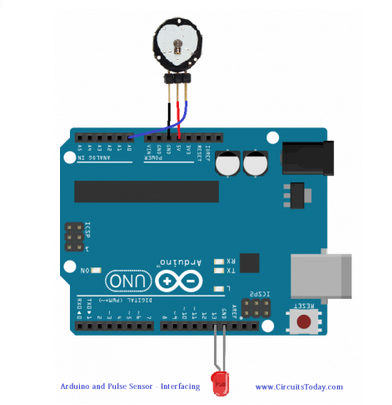
noTone(buzzer);

}

delay(100);

}

Pulse Sensor



int sensor\_pin = 0;

int led\_pin = 13;

volatile int heart\_rate;

volatile int analog\_data;

volatile int time\_between\_beats = 600;

volatile boolean pulse\_signal = false;

volatile int beat[10];         //heartbeat values will be sotred in this array

volatile int peak\_value = 512;

volatile int trough\_value = 512;

volatile int thresh = 525;

volatile int amplitude = 100;

volatile boolean first\_heartpulse = true;

volatile boolean second\_heartpulse = false;

volatile unsigned long samplecounter = 0;   //This counter will tell us the pulse timing

volatile unsigned long lastBeatTime = 0;

void setup()

{

  pinMode(led\_pin,OUTPUT);

  Serial.begin(115200);

  interruptSetup();

}

void loop()

{

      Serial.print("BPM: ");

     Serial.println(heart\_rate);

  delay(200); //  take a break

}

void interruptSetup()

{

  TCCR2A = 0x02;  // This will disable the PWM on pin 3 and 11

  OCR2A = 0X7C;   // This will set the top of count to 124 for the 500Hz sample rate

  TCCR2B = 0x06;  // DON'T FORCE COMPARE, 256 PRESCALER

  TIMSK2 = 0x02;  // This will enable interrupt on match between OCR2A and Timer

  sei();          // This will make sure that the global interrupts are enable

}

ISR(TIMER2\_COMPA\_vect)

{

  cli();

  analog\_data = analogRead(sensor\_pin);

  samplecounter += 2;

  int N = samplecounter - lastBeatTime;

  if(analog\_data < thresh && N > (time\_between\_beats/5)\*3)

    {

      if (analog\_data < trough\_value)

      {

        trough\_value = analog\_data;

      }

    }

  if(analog\_data > thresh && analog\_data > peak\_value)

    {

      peak\_value = analog\_data;

    }

   if (N > 250)

  {

    if ( (analog\_data > thresh) && (pulse\_signal == false) && (N > (time\_between\_beats/5)\*3) )

      {

        pulse\_signal = true;

        digitalWrite(led\_pin,HIGH);

        time\_between\_beats = samplecounter - lastBeatTime;

        lastBeatTime = samplecounter;

       if(second\_heartpulse)

        {

          second\_heartpulse = false;

          for(int i=0; i<=9; i++)

          {

            beat[i] = time\_between\_beats; //Filling the array with the heart beat values

          }

        }

        if(first\_heartpulse)

        {

          first\_heartpulse = false;

          second\_heartpulse = true;

          sei();

          return;

        }

      word runningTotal = 0;

      for(int i=0; i<=8; i++)

        {

          beat[i] = beat[i+1];

          runningTotal += beat[i];

        }

      beat[9] = time\_between\_beats;

      runningTotal += beat[9];

      runningTotal /= 10;

      heart\_rate = 60000/runningTotal;

    }

  }

  if (analog\_data < thresh && pulse\_signal == true)

    {

      digitalWrite(led\_pin,LOW);

      pulse\_signal = false;

      amplitude = peak\_value - trough\_value;

      thresh = amplitude/2 + trough\_value;

      peak\_value = thresh;

      trough\_value = thresh;

    }

  if (N > 2500)

    {

      thresh = 512;

      peak\_value = 512;

      trough\_value = 512;

      lastBeatTime = samplecounter;

      first\_heartpulse = true;

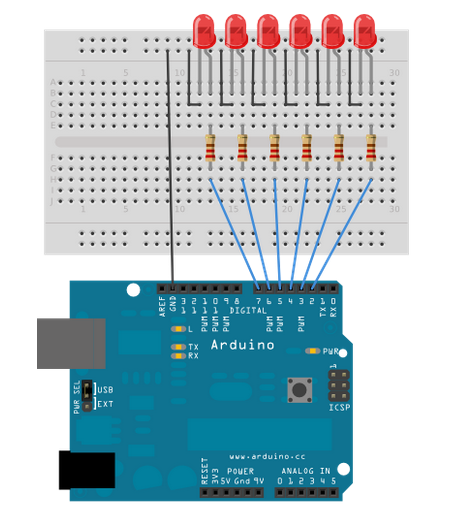
      second\_heartpulse = false;

    }

  sei();

}

Led using array



int timer = 100;           // The higher the number, the slower the timing.  
int ledPins[] = {  
  2, 7, 4, 6, 5, 3  
};       // an array of pin numbers to which LEDs are attached  
int pinCount = 6;           // the number of pins (i.e. the length of the array)  
  
void setup() {  
  // the array elements are numbered from 0 to (pinCount - 1).  
  // use a for loop to initialize each pin as an output:  
  for (int thisPin = 0; thisPin < pinCount; thisPin++) {  
    pinMode(ledPins[thisPin], OUTPUT);  
  }  
}  
  
void loop() {  
  // loop from the lowest pin to the highest:  
  for (int thisPin = 0; thisPin < pinCount; thisPin++) {  
    // turn the pin on:  
    digitalWrite(ledPins[thisPin], HIGH);  
    delay(timer);  
    // turn the pin off:  
    digitalWrite(ledPins[thisPin], LOW);  
  
  }  
  
  // loop from the highest pin to the lowest:  
  for (int thisPin = pinCount - 1; thisPin >= 0; thisPin--) {  
    // turn the pin on:  
    digitalWrite(ledPins[thisPin], HIGH);  
    delay(timer);  
    // turn the pin off:  
    digitalWrite(ledPins[thisPin], LOW);  
  }  
}

Touch Sensor

int in = 2;

int out = 13;

int state = HIGH;

int r;

int p = LOW;

long time = 0;

long debounce = 200;

void setup()

{

pinMode(in, INPUT);

pinMode(out, OUTPUT);

}

void loop()

{

r = digitalRead(in);

if (r == HIGH && p == LOW && millis() - time > debounce) {

if (state == HIGH)

state = LOW;

else

state = HIGH;

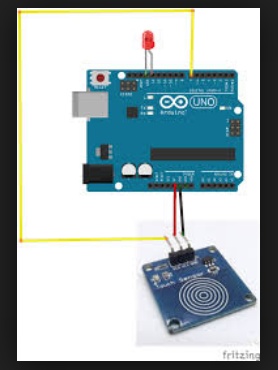
time = millis();

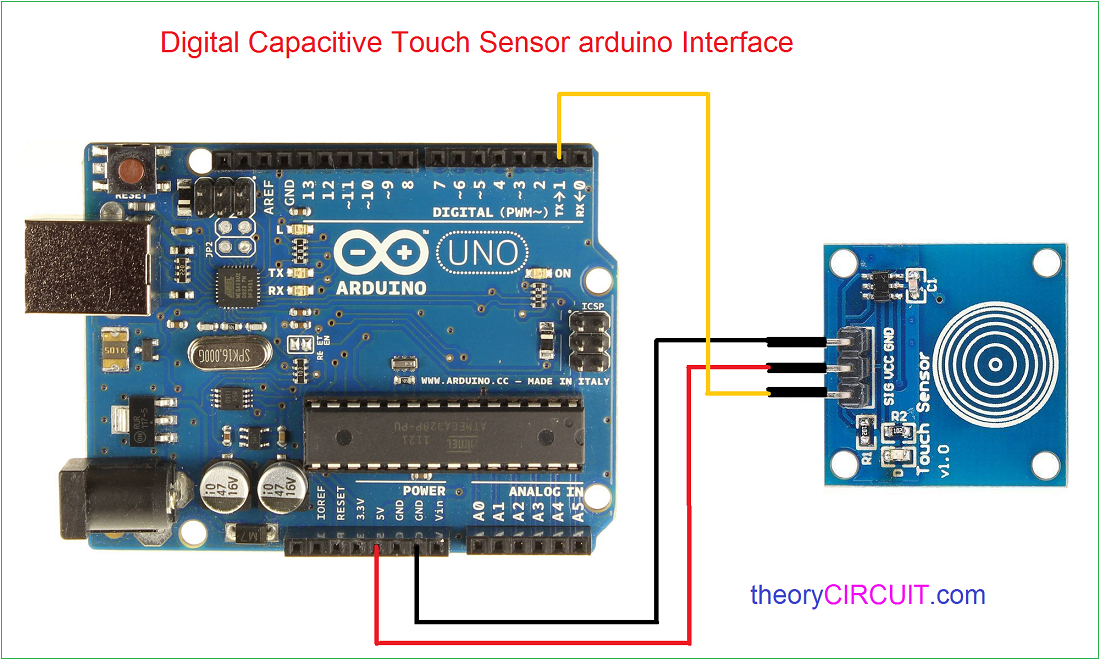
}

digitalWrite(out, state);

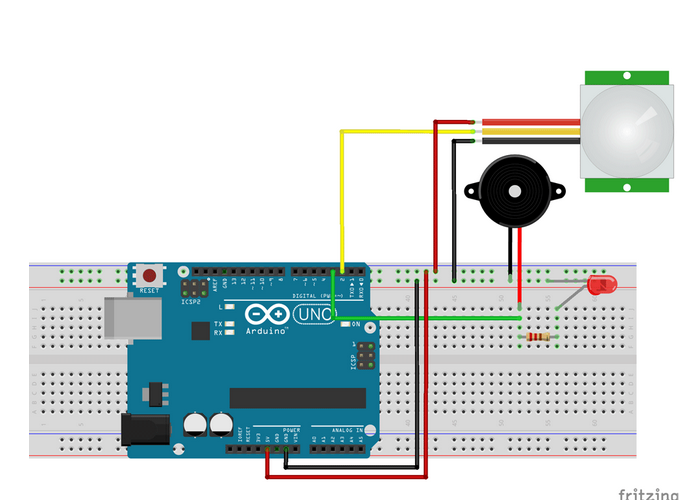
p = r;

}





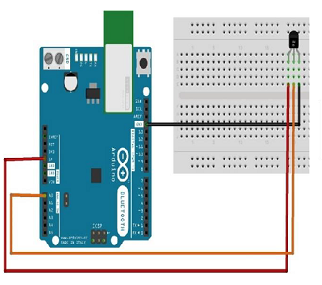
PIR sensor



void setup() {  
  pinMode(2, INPUT); //Pin 2 as INPUT  
  pinMode(3, OUTPUT); //PIN 3 as OUTPUT  
}

void loop() {  
  if (digitalRead(2) == HIGH)  
  {  
  digitalWrite(3, HIGH);   // turn the LED/Buzz ON  
  delay(100);                       // wait for 100 msecond   
  digitalWrite(3, LOW);   // turn the LED/Buzz OFF  
  delay(100);                       // wait for 100 msecond   
  }  
}

Temperature sensor LM35



float temp;

int tempPin = 0;

void setup() {

Serial.begin(9600);

}

void loop() {

temp = analogRead(tempPin);

// read analog volt from sensor and save to variable temp

temp = temp \* 0.48828125;

// convert the analog volt to its temperature equivalent

Serial.print("TEMPERATURE = ");

Serial.print(temp); // display temperature value

Serial.print("\*C");

Serial.println();

delay(1000); // update sensor reading each one second

}