Title: Data Acquisition with Ultrasonic sensor and PIR (Passive Infrared Rays)

Description:

US HC-SR04 is the US used in the project. **Ultrasonic sensor** is a sensor that works on **principle** similar to radar or **sonar**. It generates high frequency sound and calculate the time interval between the sending of signal and the receiving of echo. Therefore, ultrasonic sensor can be used to measure distance.

# PIR used in the project is HC-SR501.PIR sensor detects a human being moving around within approximately 10m from the sensor. This is an average value, as the actual detection range is between 5m and 12m.PIR are fundamentally made of a pyro electric sensor, which can detect levels of infrared radiation.

The US and PIR are connected with Arduino UNO, and made to find the value of what each sensor sensed. In the Serial Plotter, both the sensor’s graph are plotted. Same with the serial monitor, the values are displayed

Hardware:

1.HC-SR04

2.HC-SR501

3.Arduino UNO

4.Wires

5.USB Port

Codes:

int inputPin = 2;               // choose the input pin (for PIR sensor)  
int pirState = LOW;             // we start, assuming no motion detected  
int val = 0;                    // variable for reading the pin status  
int clk=0;  
int i;  
int trigPin=A0; //Sensor Trip pin connected to Arduino pin A0  
int echoPin=A1;  //Sensor Echo pin connected to Arduino pin A1  
int myCounter=0;  //declare your variable myCounter and set to 0  
int servoControlPin=6; //Servo control line is connected to pin 6  
float pingTime;  //time for ping to travel from sensor to target and return  
float targetDistance; //Distance to Target in inches  
float speedOfSound=776.5; //

void setup() {  
   
pinMode(trigPin, OUTPUT);  
pinMode(echoPin, INPUT);  
pinMode(inputPin, INPUT);     // declare sensor as input  
pinMode(LED\_BUILTIN, OUTPUT);   
Serial.begin(9600);

  // put your setup code here, to run once:

}

void loop() {  
  val = digitalRead(inputPin);  // read input value  
   
  if (val == HIGH) // check if the input is HIGH  
    
  {         
     i++;  
    digitalWrite(LED\_BUILTIN, HIGH);  // turn LED ON  
    if (pirState == LOW)  
    {  
        
       
      
      // we have just turned on  
      Serial.println("Motion detected!");

      // We only want to print on the output change, not state  
      pirState = HIGH;  
    
clk=millis()/1000;  
    }  
      
Serial.print(i);  
Serial.print("\t");  
      
  } else {  
    digitalWrite(LED\_BUILTIN, LOW); // turn LED OFF  
    if (pirState == HIGH){  
      // we have just turned off

      Serial.println("Motion ended!");  
      // We only want to print on the output change, not state  
      pirState = LOW;  
//     delay(5000);  
    
    }  
  }

digitalWrite(trigPin, LOW); //Set trigger pin low  
  delayMicroseconds(2000); //Let signal settle  
  digitalWrite(trigPin, HIGH); //Set trigPin high  
  delayMicroseconds(15); //Delay in high state  
  digitalWrite(trigPin, LOW); //ping has now been sent  
  delayMicroseconds(10); //Delay in high state  
    
  pingTime = pulseIn(echoPin, HIGH);  //pingTime is presented in microceconds  
  pingTime=pingTime/1000000; //convert pingTime to seconds by dividing by 1000000 (microseconds in a second)  
  pingTime=pingTime/3600; //convert pingtime to hourse by dividing by 3600 (seconds in an hour)  
  targetDistance= speedOfSound \* pingTime;  //This will be in miles, since speed of sound was miles per hour  
  targetDistance=targetDistance/2; //Remember ping travels to target and back from target, so you must divide by 2 for actual target distance.  
  targetDistance= targetDistance\*63360;    //Convert miles to inches by multipling by 63360 (inches per mile)  
    
   
  delay(250); //pause to let things settle  
  Serial.println(targetDistance);

   // put your main code here, to run repeatedly:

}

Output:

