

Sensor PIR dan Buzzer

Disusun Guna Memenuhi Tugas Mata Kuliah

Robotika

Dosen Pengampu : Bp. Dr. Agus Efendi, M.Pd.

Oleh :

Nama : Syahrul Bagus H
NIM : K3520072
Prodi : Pendidikan Teknik Informatika dan Komputer

A. Alat dan Bahan

1. Board Arduino UNO
2. Project Board
3. Resistor
4. Kabel jumper
5. Lampu LED
6. Sensor PIR
7. Buzzer

B. Gambar Rangkaian



C. Keterangan

Program untuk menggunakan sensor PIR (Passive Infrared Sensor) untuk mendeteksi gerakan dan memberikan output suara dan cahaya. Berikut adalah penjelasan tiap bagian program: Baris 1-12: Deklarasi variabel dan pin yang akan digunakan. Terdapat variabel `calibrationTime`, `lowIn`, dan `pause` yang akan digunakan pada program. Selain itu, juga

terdapat pin pirPin, ledPin, dan Buzzer yang akan diatur pada program. Baris 14-24: Konfigurasi awal pada program, di mana program akan menunggu selama calibrationTime sebelum memulai program utama. Ketika sensor PIR sudah siap, program akan mencetak "SENSOR ACTIVE" pada serial monitor dan menunggu selama 50 ms sebelum melanjutkan program. Baris 26-56: Program utama yang akan dijalankan secara berulang-ulang pada loop(). Pada bagian ini, program akan membaca nilai input dari pin pirPin. Jika nilai input tersebut HIGH, maka LED akan dinyalakan dan buzzer akan berbunyi dengan frekuensi 1000 Hz. Jika lockLow bernilai true, maka program akan menunggu transisi dari HIGH ke LOW sebelum memberikan output lebih lanjut pada serial monitor. Jika takeLowTime bernilai true, maka program akan menyimpan waktu saat terjadi transisi dari HIGH ke LOW. Jika sensor PIR masih dalam keadaan LOW selama lebih dari pause, maka program akan menganggap tidak ada gerakan lagi dan akan memberikan output pada serial monitor dengan mencetak "motion ended at" dan waktu saat gerakan diakhiri. Baris 58-60: Program akan diulang kembali ke bagian loop() dan membaca input dari pin pirPin. Jika nilai input tersebut LOW, maka LED akan dimatikan dan buzzer akan berhenti berbunyi. Selain itu, program akan memeriksa apakah takeLowTime bernilai true. Jika true, maka program akan menyimpan waktu saat terjadi transisi dari HIGH ke LOW. Jika lockLow bernilai false, maka program akan memeriksa selisih waktu saat ini dengan waktu saat terjadi transisi dari HIGH ke LOW. Jika selisih waktu tersebut lebih besar dari pause, maka program akan menganggap tidak ada gerakan lagi dan akan memberikan output pada serial monitor dengan mencetak "motion ended at" dan waktu saat gerakan diakhiri.

D. Kode Program

```
//the time we give the sensor to calibrate (10-60 secs according to the
    datasheet)
int calibrationTime = 30;

//the time when the sensor outputs a low impulse
long unsigned int lowIn;

//the amount of milliseconds the sensor has to be low
//before we assume all motion has stopped
long unsigned int pause = 5000;

boolean lockLow = true;
boolean takeLowTime;

int pirPin = 7;    //the digital pin connected to the PIR sensor's output
int ledPin = 5;    //the digital pin connected to the LED output
```

```

int Buzzer = 6;    //the digital pin connected to the BUZZER output

////////////////////
//SETUP
void setup(){
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(Buzzer, OUTPUT);
  digitalWrite(pirPin, LOW);

  //give the sensor some time to calibrate
  Serial.print("calibrating sensor ");
  for(int i = 0; i < calibrationTime; i++){
    Serial.print(".");
    delay(1000);
  }
  Serial.println(" done");
  Serial.println("SENSOR ACTIVE");
  delay(50);
}

////////////////////
//LOOP
void loop(){

  if(digitalRead(pirPin) == HIGH){
    digitalWrite(ledPin, HIGH);    //the led visualizes the sensors output
    pin state
    tone(Buzzer,1000);
    if(lockLow){
      //makes sure we wait for a transition to LOW before any further output
      is made:
      lockLow = false;
      Serial.println("---");
      Serial.print("motion detected at ");
      Serial.print(millis()/1000);
      Serial.println(" sec");
      delay(50);
    }
    takeLowTime = true;
  }

  if(digitalRead(pirPin) == LOW){
    digitalWrite(ledPin, LOW);    //the led visualizes the sensors output pin
    state
    noTone(Buzzer);
    if(takeLowTime){
      lowIn = millis();           //save the time of the transition from high
    to LOW
      takeLowTime = false;        //make sure this is only done at the start
    of a LOW phase
    }
    //if the sensor is low for more than the given pause,
    //we assume that no more motion is going to happen
    if(!lockLow && millis() - lowIn > pause){
      //makes sure this block of code is only executed again after

```

```
    //a new motion sequence has been detected
    lockLow = true;
    Serial.print("motion ended at ");          //output
    Serial.print((millis() - pause)/1000);
    Serial.println(" sec");
    delay(50);
  }
}
```