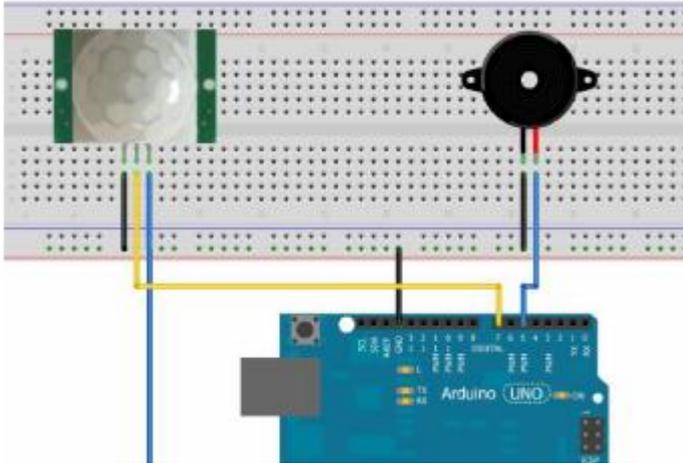


Beschreibung:

Der Bewegungsmelder, auch PIR Sensor genannt, ist sehr einfach konstruiert. Sobald er eine Bewegung detektiert, gibt er auf einem Pin eine Spannung von 5 Volt aus. Diese muss nur ausgelesen und vom Mikrocontroller verarbeitet werden. Die Dauer des Ausgangssignals (linker Regler) und die Sensibilität (rechter Regler) kann über Drehregler eingestellt werden.

Beschaltung:**Bild:****Sketch:**

```
int piezo=5; //Das Wort „piezo“ steht jetzt für den Wert 5.
int bewegung=7; //Das Wort „bewegung“ steht jetzt für den Wert 7.
int bewegungsstatus=0; //Das Wort „bewegungsstatus“ steht jetzt zunächst für den Wert 0. Später wird unter
//dieser Variable gespeichert, ob eine Bewegung erkannt wird oder nicht.
void setup() //Hier beginnt das Setup.
{
  pinMode(piezo, OUTPUT); //Der Pin mit dem Piezo (Pin 5) ist jetzt ein Ausgang.
  pinMode(bewegung, INPUT); //Der Pin mit dem Bewegungsmelder (Pin 7) ist jetzt ein Eingang.
}
void loop() //Der Loop-Teil beginnt
{ //Mit dieser Klammer wird der Loop-Teil geöffnet.
  bewegungsstatus=digitalRead(bewegung); //Hier wird der Pin7 ausgelesen. Das Ergebnis wird unter der
//Variablen „bewegungsstatus“ mit dem Wert „HIGH“ für 5Volt oder „LOW“ für 0Volt gespeichert.
  if (bewegungsstatus == HIGH) // Wenn eine Bewegung detektiert wird (Das Spannungssignal ist hoch)
  { //Programmabschnitt des IF-Befehls öffnen.
    digitalWrite(piezo, HIGH); //dann soll der Piezo piepsen.
    delay(5000); //...und zwar für für 5 Sekunden.
    digitalWrite(piezo, LOW); //...danach soll er leise sein.
  } //Programmabschnitt des IF-Befehls schließen.
  else //ansonsten...
  { //Programmabschnitt des else-Befehls öffnen.
    digitalWrite(piezo, LOW); //...soll der Piezo-Lautsprecher aus sein.
  } //Programmabschnitt des else-Befehls schließen.
} //Mit dieser letzten Klammer wird der Loop-Teil geschlossen.
```