2/25/2020

Crop Protection System

Final Report

SE/2016/032
CHATHMINI JEYATHILAKA

SE/2016/032
CHATHMINI JEYATHILAKA

Contents

Abstract

The main objective of this project is to develop a crop protection system using an Arduino board with PIR sensor being remotely controlled Automatically. As technology is advancing so farms are also needed to use these technologies. Remote controlled crop protection automation system provides a most modern solution.
Most of the farmers need to protect their crops from the animals so we need to protect the crops using new technologies. We use PIR sensor and detect the motion according to the movement make the sound to Repellency the animals. We will hope this it will be helpful to the farmers in rural area or the farmers who have their crops near to the wild areas.

Acknowledgement

First and foremost, I would like to take this opportunity to thank our lecturer Dilum sir for his guidance and advice on this project. At the same time I also won’t forget my partner participant and also friend to because they quite good with sharing some of her information to complete this year embedded system project successfully. Last but not least, I am very grateful to our University, lectures and friends where they gave us enough of time to complete this project and at the same time I would like to thank classmates who helps me a lot to complete this project.

Thank You.

Introduction

Before the beginning of every farm season, most farmers prefer to plan potential yields. On the other hand, some farmers chose to skip planning. Whether a farmer plans the **potential yield** or not, certain expectations are still present. While hoping for the best, farmers are often presented with various challenges and obstacles that require them to constantly question their **productivity and resulting final success**.

The greatest importance is usually given to **crop protection** from **diseases**, **insect pests**, and **weeds**, as well as to protection from unfavorable weather events such as **frost** or **hail** ,along with other **crop maintenance practices**. The aforementioned challenges are well-known and often discussed. However, farmers also face another **interesting challenge**, often forgotten about or not realized; **wild animal crop protection**.

In the dry season most of the elephants come to the people living area and damage the crops. Most of the pigs and other animals come in the night and eat the crops and damage the farming field, So the farmers face so much lost. This kind of the war between the animals and humans happen from the starting of the history.

Wild animals are a special challenge for farmers throughout the world. Animals such as deer, wild boars, rabbits, moles, elephants, monkeys, and many others may cause serious **damage to crops**. They can damage the plants by **feeding on plant parts** or simply by **running over the field** and trampling over the crops. Therefore, wild animals may easily cause significant **yield losses** and provoke additional **financial problems**. Another aspect to consider is that wild animal crop protection requires a **particularly cautious approach**. In other words, while utilizing his crop production, every farmer should be aware and take into consideration the fact that animals are living beings and need to be protected from any potential suffering.

Requirements

- Protect the crops from the animals
- Don’t harm the animal while protecting the farms
- Reduce the cost
- Low human effort

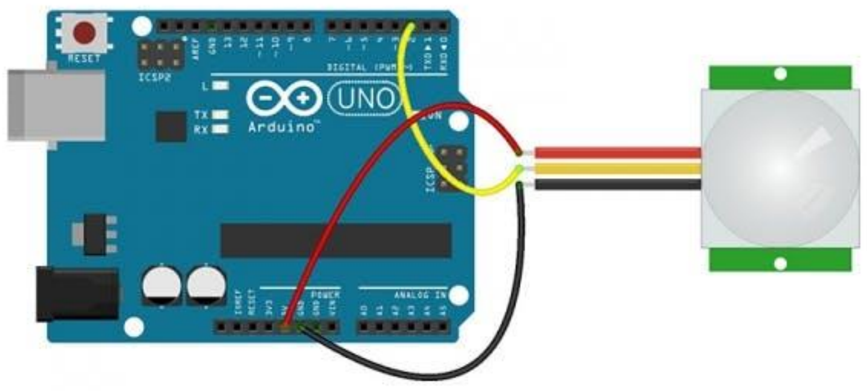
Description of Project

For the protect the crops from the animals we are planed to create an Arduino based crop protection system. It will be help to protect the crops from animals.

This system uses a motion sensor to detect wild animals approaching near the field. In such a case the sensor signals the microcontroller to take action. The microcontroller now sounds an alarm to woo the animals away from the field as well as sends message to the farmer so that he may know about the issue and come to the spot in case the animals don’t turn away by the alarm. This ensures complete safety of crops from animals thus protecting the farmers loss.

Design and implementation

Using a PIR Sensor with Arduino



Circuit

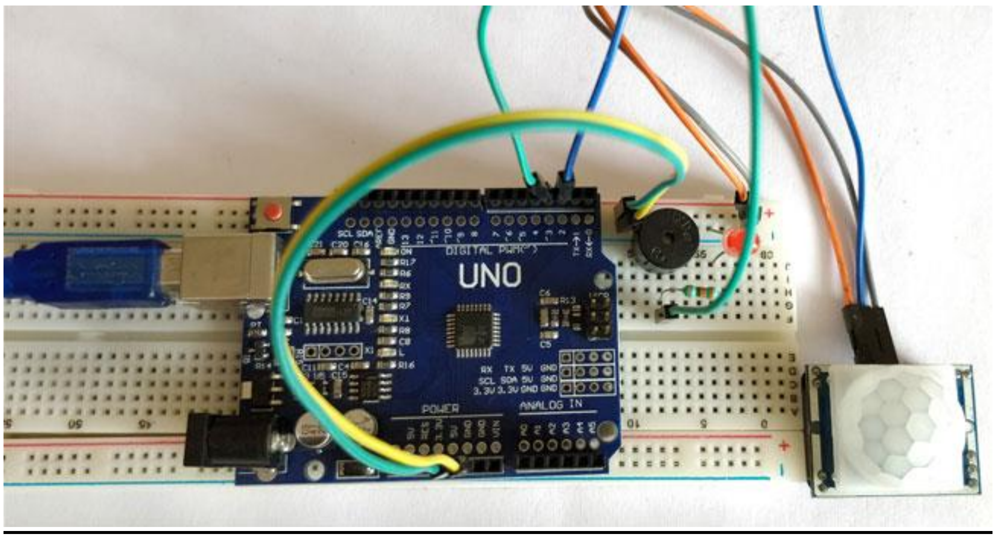
You can connect PIR output to any digital pin.

There is a jumper behind this module. If you move the jumper to L position, the sensor will ‘toggle’ (change state) whenever motion is detected. This is unlikely to be of much use in a practical applications. This mode is called non-triggering or Single Triggering mode.

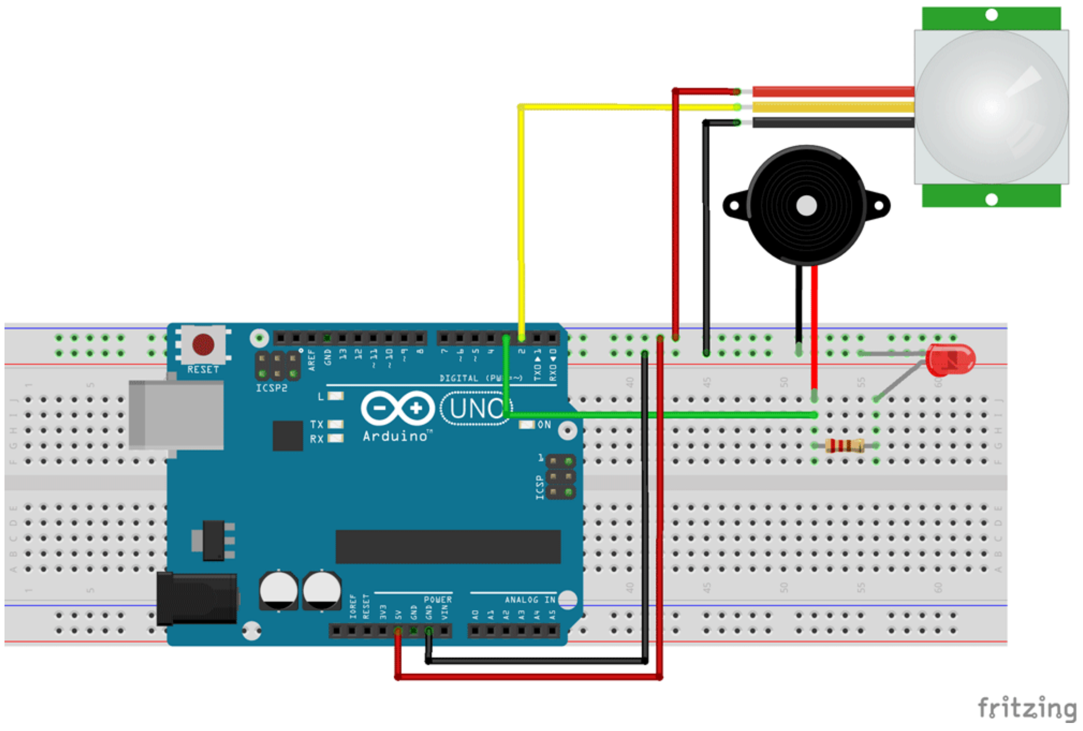
Moving the jumper to the H position will result in the more usual sensor logic. The sensor will turn on when motion is detected and turn off a while after the last motion is detected. This sensor will reset the timer (which would otherwise turn the output off) each time motion is detected; this would be applicable, for example, for room occupancy lighting control where you don’t want the lights to blink off while the unit resets. This is called Retriggering mode. (or repeatable trigger mode).

There are also two potentiometers behind this module. By changing the SENSITIVITY potentiometer, you can reduce or increase the sensitivity of the sensor (clockwise increase), and also by changing TIME potentiometer the output delay after movement detection will be changed.

Attach buzzer and the led bulbs

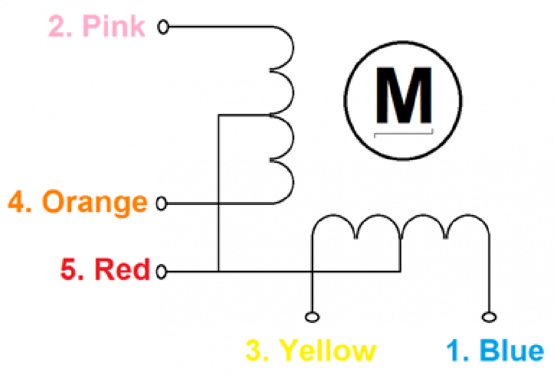


Detecting motions or movements has always been important in most projects. With the help of the PIR Sensor it has become very easy to detect human/animal movements. In this project we will learn how we can **interface a PIR Sensor with a microcontroller like Arduino**. We will **interface an Arduino with PIR module** and blink a LED and beep a Buzzer whenever a movement is detected.



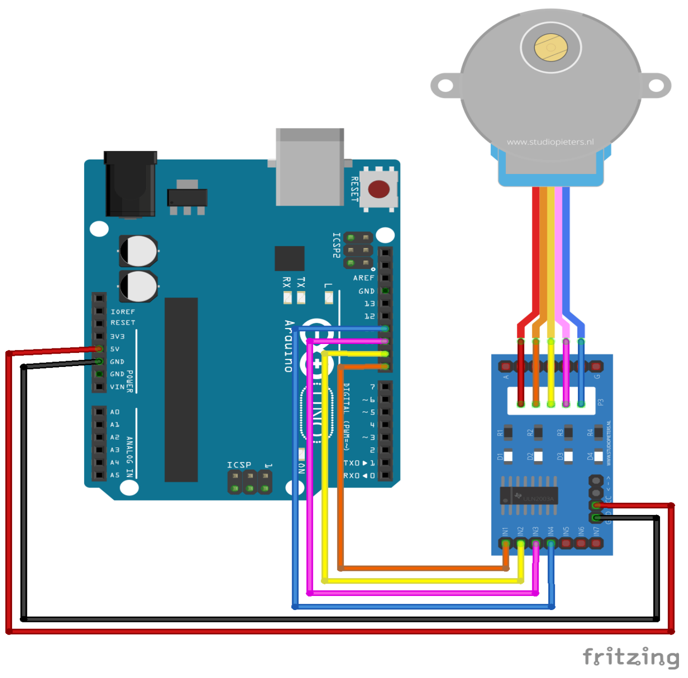
Attach Stepper Motor

Stepper motors are increasingly taking its position in the world of the electronics. Starting from a normal Surveillance camera to a complicated CNC machines/Robot these Stepper Motors are used everywhere as actuators since they provide accurate controlling. In this tutorial we will learn about the most commonly/cheaply available stepper motor **28-BYJ48** and how to interface it with **Arduino** using **ULN2003 stepper module**.



As you can see the motor has Unipolar 5-lead coil arrangement. There are four coils which have to be energized in a particular sequence. The Red wires will be supplied with +5V and the remaining four wires will be pulled to ground for triggering the respective coil. We use a microcontroller like Arduino energize these coils in a particular sequence and make the motor perform the required number of steps.

So now, why is this motor called the **28-BYJ48**? Seriously!!! I don’t know. There is no technical reason for this motor for being named so; maybe we should dive much deeper into it. Let us look at some of the important technical data obtained from the datasheet of this motor in the picture below.



It is important to know how to calculate the steps per Revolution for your stepper motor because only then you can program it effectively.

In Arduino we will be operating the motor in 4-step sequence so the stride angle will be 11.25° since it is 5.625°(given in datasheet) for 8 step sequence it will be 11.25° (5.625*2=11.25).

Steps per revolution = 360/step angle

Here, 360/11.25 = **32 steps per revolution**

The circuit Diagram for the **arduino stepper motor control project** is shown above. We have used the 28BYJ-48 Stepper motor and the ULN2003 Driver module. To energise the four coils of the stepper motor we are using the digital pins 8,9,10 and 11. The driver module is powered by the 5V pin of the Arduino Board.

But, power the driver with External Power supply when you are connecting some load to the steppe motor. Since I am just using the motor for demonstration purpose I have used the +5V rail of the Arduino Board. Also remember to connect the Ground of the Arduino with the ground of the Diver module.

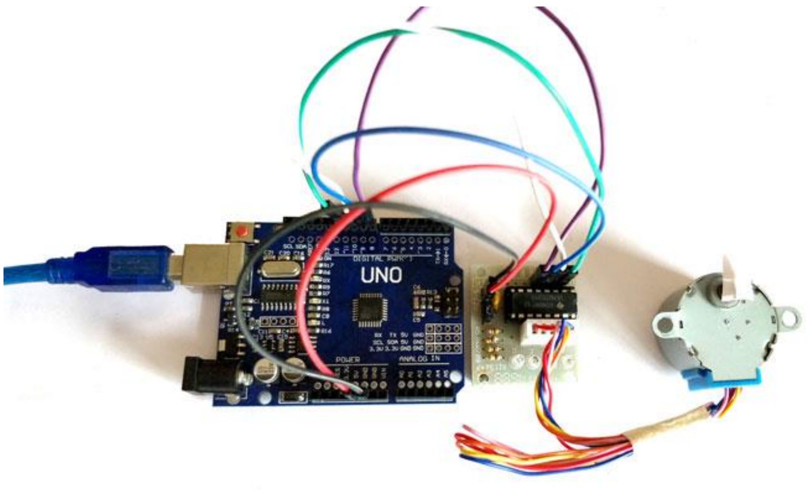
The number of steps per revolution for our stepper motor was calculated to be 32; hence we enter that as shown in the line below

```
#define STEPS 32
```

Next you have to create instances in which we specify the pins to which we have connected the Stepper motor.

```
Stepper stepper (STEPS, 8, 10, 9, 11);
```

Note: The pins number are disordered as 8,10,9,11 on purpose. You have to follow the same pattern even if you change the pins to which your motor is connected.



Now, upload the program in your Arduino UNO and open the serial monitor. As discussed earlier we will have to make 2048 steps to make one complete rotation, so when we enter 2048 the motor will make one complete rotation in clockwise direction by making 2048 steps. To rotate in anti-clockwise just enter the number with "–"negative sign. So, entering -1024 will make the motor to rotate half the way in anti-clock wise direction. You can enter any desired values, like entering 1will make the motor to take only one step.

Materials use

Arduino Uno Microcontroller AT Mega328P

ATmega328 is a single chip microcontroller created by Atmel in the mega AVR family. The Atmel 8-bit AVR RISC-based microcontroller combines 32 kB ISP flash memory with read-while-write capabilities, 1 kB EEPROM, 2 kB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The device achieves throughput approaching 1 MIPS per MHz.

PASSIVE INFRARED SENSOR (PIR)

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. PIRs are basically made of a pyroelectric sensor, which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

Features

Output: Digital pulse high (3V) when triggered (motion detected) digital low when idle (no motion detected). Pulse lengths are determined by resistors and capacitors on the PCB and differ from sensor to sensor.

Power supply: 5V-12V input voltage for most modules (they have a 3.3V regulator), but 5V is ideal in case the regulator has different specs

PIEZO BUZZER

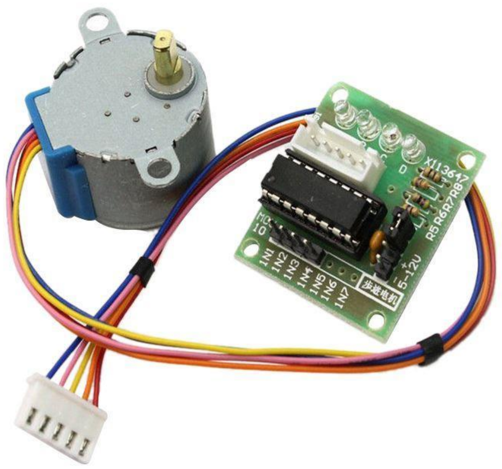
A "piezo buzzer" is basically a tiny speaker that you can connect directly to an Arduino.

"Piezoelectricity" is an effect where certain crystals will change shape when you apply electricity to them. By applying an electric signal at the right frequency, the crystal can make sound.

If your buzzer has a sticker on top of it, pull the sticker off. Connect one pin (it doesn't matter which one) to the Arduino's ground (Gnd) and the other end to digital pin 8.

From the Arduino, you can make sounds with a buzzer by using **tone**. You have to tell it which pin the buzzer is on, what frequency (in Hertz, Hz) you want, and how long (in milliseconds) you want it to keep making the tone.

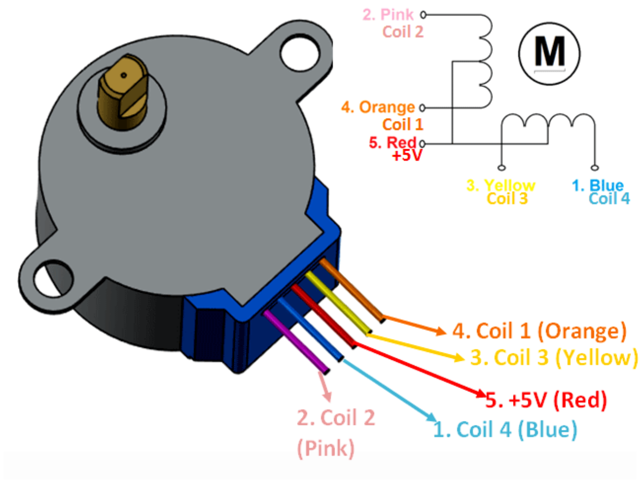
STEPPER MOTOR



Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called "phases". By energizing each phase in sequence, the motor will rotate, one step at a time.

With a computer controlled stepping you can achieve very precise positioning and/or speed control. For this reason, stepper motors are the motor of choice for many precision motion control applications.

Stepper motors come in many different sizes and styles and electrical characteristics. This guide details what you need to know to pick the right motor for the job.

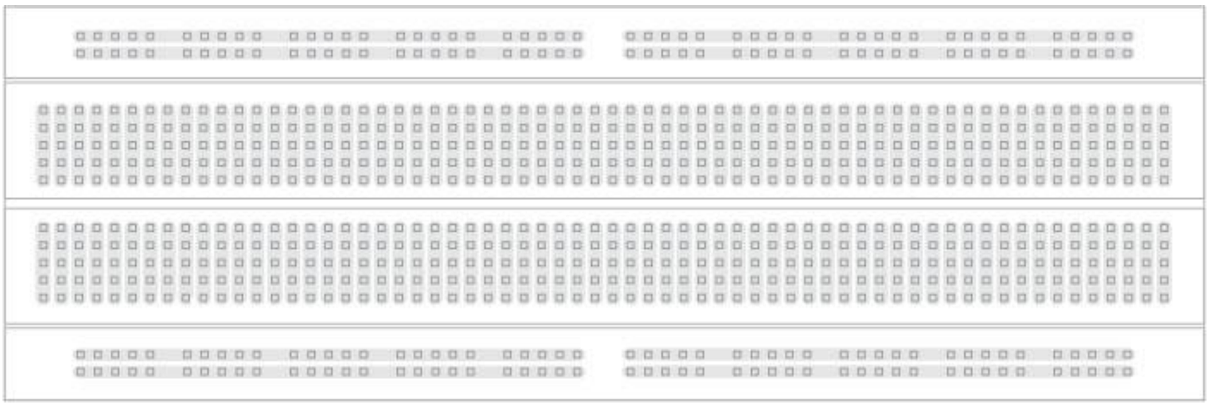


Every revolution of the stepper motor is divided into a discrete number of steps, in many cases 200 steps, and the motor must be sent a separate pulse for each step. The stepper motor can only take one step at a time and each step is the same size.

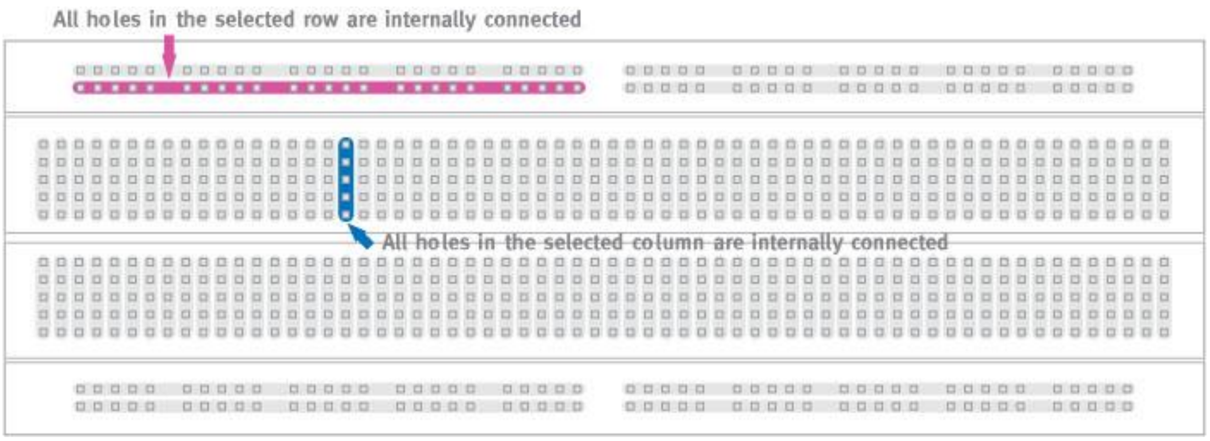
Since each pulse causes the motor to rotate a precise angle, typically 1.8°, the motor's position can be controlled without any feedback mechanism. As the digital pulses increase in frequency, the step movement changes into continuous rotation, with the speed of rotation directly proportional to the frequency of the pulses.

BREAD Board

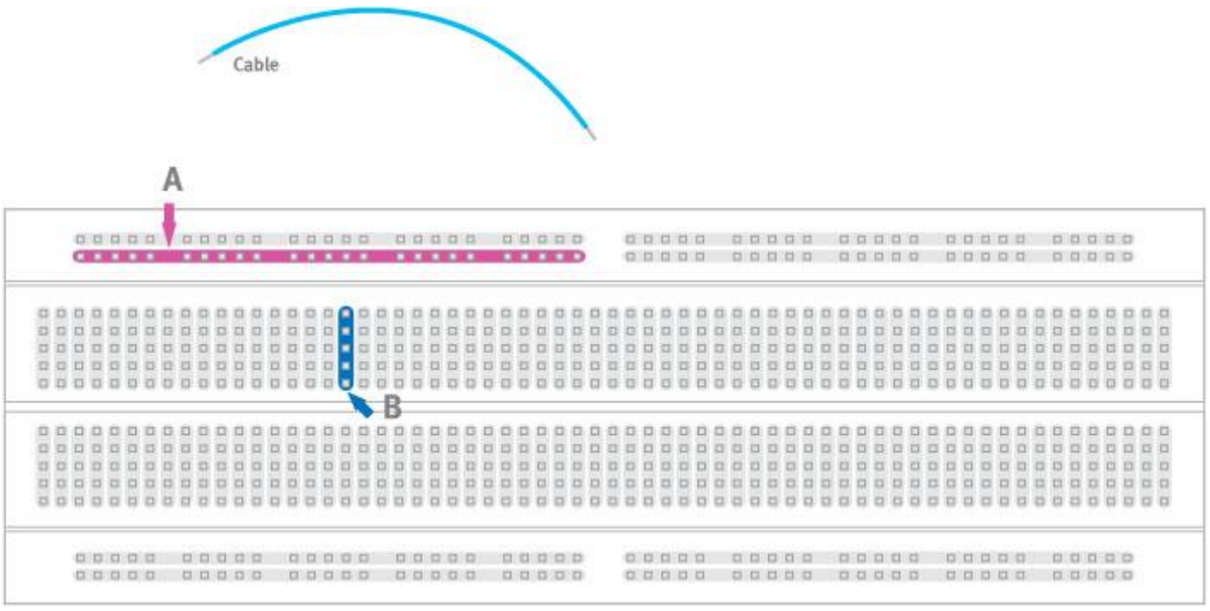
A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.



Note how all holes in the selected row are connected together, so the holes in the selected column. The set of connected holes can be called a node:


All holes in the selected row are internally connected
All holes in the selected column are internally connected

To interconnect the selected row (node A) and column (node B) a cable going from any hole in the row to any hole in the column is needed:



Code for the System

```
#include <Stepper.h>

// Define connection pins:
#define buzzerPin 5
#define pirPin 2
#define ledPin 13

//Define the Motor inputpins
int in1Pin = 8;
int in2Pin = 9;
int in3Pin = 10;
int in4Pin = 11;
const int stepsPerRevolution = 2048;// Update the number of steps per revolution r
equired for your motor
// Create a stepper object
// Note: We are using 28BYJ-
48 5VDC Stepper Motor, for this motor, we need to set wiring sequence to (1-3-2-
4) instead of (1-2-3-4)
Stepper motor(stepsPerRevolution, in1Pin, in3Pin, in2Pin, in4Pin);

// Create variables:
int val = 0;
bool motionState = false; // We start with no motion detected.

void setup() {
  // Configure the pins as input or output:
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(pirPin, INPUT);

  int user_speed=10;    // Read entered integer value and store it in a variable
  motor.setSpeed(user_speed);   // Set stepper motor speed to user defined speed

  // Begin serial communication at a baud rate of 9600:
  Serial.begin(9600);
}

void loop() {
  for(int AngleDivision=0; AngleDivision<4; AngleDivision++){
      Serial.println("clockwise"); // print text on serial monitor
      motor.step(stepsPerRevolution/4); // rotate motor in clockwise direction for
 one revolution
      delay(500);
       for(int checkTime=0; checkTime<20;checkTime++){
          check();
       }
  }
  for(int AngleDivision=0; AngleDivision<4; AngleDivision++){
      Serial.println("clockwise"); // print text on serial monitor
      motor.step(-
stepsPerRevolution/4); // rotate motor in clockwise direction for one revolution
      delay(500);
       for(int checkTime=0; checkTime<20;checkTime++){
          check();
       }
  }
}


//function to check the animal is there or not
void check(){
   // Read out the pirPin and store as val:
  val = digitalRead(pirPin);

  // If motion is detected (pirPin = HIGH), do the following:
  if (val == HIGH) {
    digitalWrite(ledPin, HIGH); // Turn on the on-board LED.
    alarm(500, 2000);  // Call the alarm(duration, frequency) function.
    delay(150);

    // Change the motion state to true (motion detected):
    if (motionState == false) {
      Serial.println("Motion detected!");
      motionState = true;
    }
  }

  // If no motion is detected (pirPin = LOW), do the following:
  else {
    digitalWrite(ledPin, LOW); // Turn off the on-board LED.
    noTone(buzzerPin); // Make sure no tone is played when no motion is detected.
    delay(150);

    // Change the motion state to false (no motion):
    if (motionState == true) {
      Serial.println("Motion ended!");
      motionState = false;
    }
  }
}

// Function to create a tone with parameters duration and frequency:
void alarm(long duration, int freq) {
  tone(buzzerPin, freq);
  delay(duration);
  noTone(buzzerPin);
}
```

Budget of the System

Arduino uno board	–	1000RS
Stepper Motor	-	400RS
PIR sensor	-	700RS
BUZZER	-	50RS
Connection wires	-	50RS
BREAD Board	-	100RS
Total	**-**	**2300RS**

Problem We face

In this project we try to connect the server motor without the server motor controller it didn't work After that we search it in websites most of the searches tell us to use the server motor controller. After that we attach the server motor controller to this project. Then it is work properly.

Future Modification

We plan to add the GSM module to find the particular position of the system and it can able to send the notifications to the user if the animal didn't go. So user can able to find the place and protect his crops.

It can get energy from using the solar cell. So It can use the green energy also.

Conclusion

The crop protection system has been experimentally proven to work satisfactorily by connecting sample appliances to it and the appliances were successfully work.

We learned many skills such as soldering, wiring the circuit and other tools that we use for this project and was able to work together as a team during this project.

References

https://www.microchip.lk/ - product price Analysis

https://circuitdigest.com/microcontroller-projects/arduino-motion-detector-using-pir-sensor - get idea how to implement

https://create.arduino.cc/projecthub/electropeak/pir-motion-sensor-how-to-use-pirs-w-arduino-raspberry-pi-18d7fa - get idea how to implement

https://circuitdigest.com/microcontroller-projects/arduino-stepper-motor-control-tutorial/ - get stepper motor idea