UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

(Universidad del Perú, DECANA DE AMÉRICA)

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE

CURSO

SISTEMAS DIGITALES

Segundo avance de proyecto

ALUMNOS

Calderon Flores, Ricardo Aldair

Choy Rios, Alejandro Mateo

Cueto Salazar, Sebastian Antonio

Mattos Hilario, Yayir Flabio

DOCENTE

Fermín Perez, Félix Armando

LIMA – PERÚ

2023

Índice

Introducción

La tecnología está cada vez más presente en nuestras vidas, y los microprocesadores y los sistemas digitales juegan un papel fundamental en ella. Estos dispositivos se utilizan en una amplia gama de aplicaciones, desde los teléfonos inteligentes y las computadoras hasta los automóviles y los electrodomésticos.

En este proyecto, utilizaremos un módulo ESP32, sensores, cables, protoboard y un buzzer para crear un espantapájaros de búho. El ESP32 es un microprocesador de doble núcleo que cuenta con una amplia gama de características, incluyendo conectividad Wi-Fi y Bluetooth. Los sensores que utilizaremos nos permitirán detectar la presencia de pájaros, y el buzzer nos permitirá emitir un sonido para ahuyentarlos.

Este proyecto puede ser útil para proteger cultivos y jardines de los pájaros. Los espantapájaros tradicionales son a menudo ineficaces, ya que los pájaros se acostumbran a ellos con el tiempo. Este proyecto, por otro lado, utiliza sensores para detectar la presencia de pájaros, lo que lo hace más efectivo.

El proyecto se encuentra en una fase inicial de desarrollo. En la actualidad, se ha completado la construcción del hardware y se ha desarrollado el código base. El código base permite al ESP32 detectar la presencia de pájaros utilizando un sensor PIR. Cuando se detecta un pájaro, el ESP32 emite un sonido a través del buzzer.

Objetivos

Objetivos principales:

Crear un espantapájaros de búho que sea capaz de detectar la presencia de pájaros.

Emitir un sonido para ahuyentar a los pájaros cuando se detecten.

Objetivos secundarios:

Utilizar sensores adicionales para mejorar la detección de pájaros.

Desarrollar un algoritmo más sofisticado para evitar que los pájaros se acostumbren al espantapájaros.

Añadir un sistema de iluminación para aumentar la eficacia del espantapájaros durante la noche.

Marco Teórico

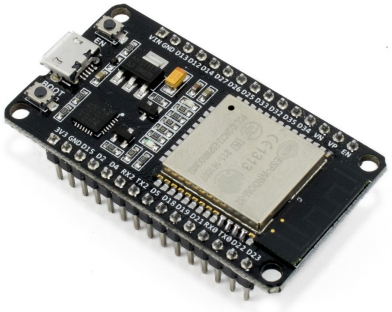
Sensores de movimiento

Los sensores de movimiento desempeñarán un papel crítico en la detección de la presencia de aves, permitiendo la activación o desactivación de nuestro sistema de disuasión. Aquí se presentan detalles relevantes:

Para el siguiente proyecto se hizo uso de un sensor HC-S04, es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición.

ESP32

Es una placa de desarrollo altamente versátil que se utiliza para crear proyectos electrónicos que requieren conectividad WiFi y Bluetooth. Su alta compatibilidad con Arduino IDE lo hace ideal para desarrolladores que buscan crear proyectos más potentes y sofisticados a un costo accesible y sin complicaciones.



Buzzer

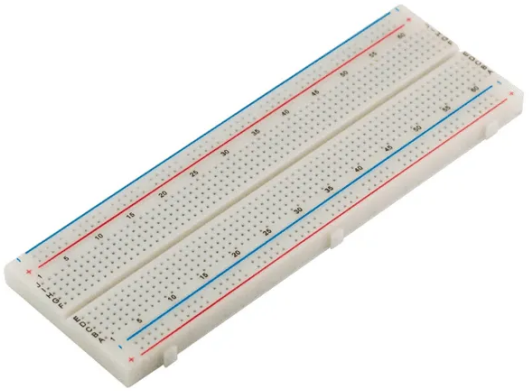
Un buzzer o zumbador es un pequeño transductor electroacústico que produce un sonido o zumbido continuo o intermitente de un mismo tono (generalmente agudo) . Sirve como mecanismo de señalización o aviso y se utiliza en múltiples sistemas, como en automóviles, electrodomésticos, incluidos los despertadores, entre otros

Cables Jumper

Cables que se utilizan para conectar diferentes componentes electrónicos entre sí. Estos cables son muy populares en proyectos de electrónica y robótica, ya que permiten conectar fácilmente diferentes componentes sin necesidad de soldarlos

Protoboard

Es un instrumento utilizado en electrónica que permite realizar conexiones eléctricas sin soldadura, muy útil para pruebas y prototipos de circuitos. Es una herramienta básica y muy importante en la electrónica, ya que permite revisar el funcionamiento de un circuito de forma rápida y sin consumir recursos.



Spiffs

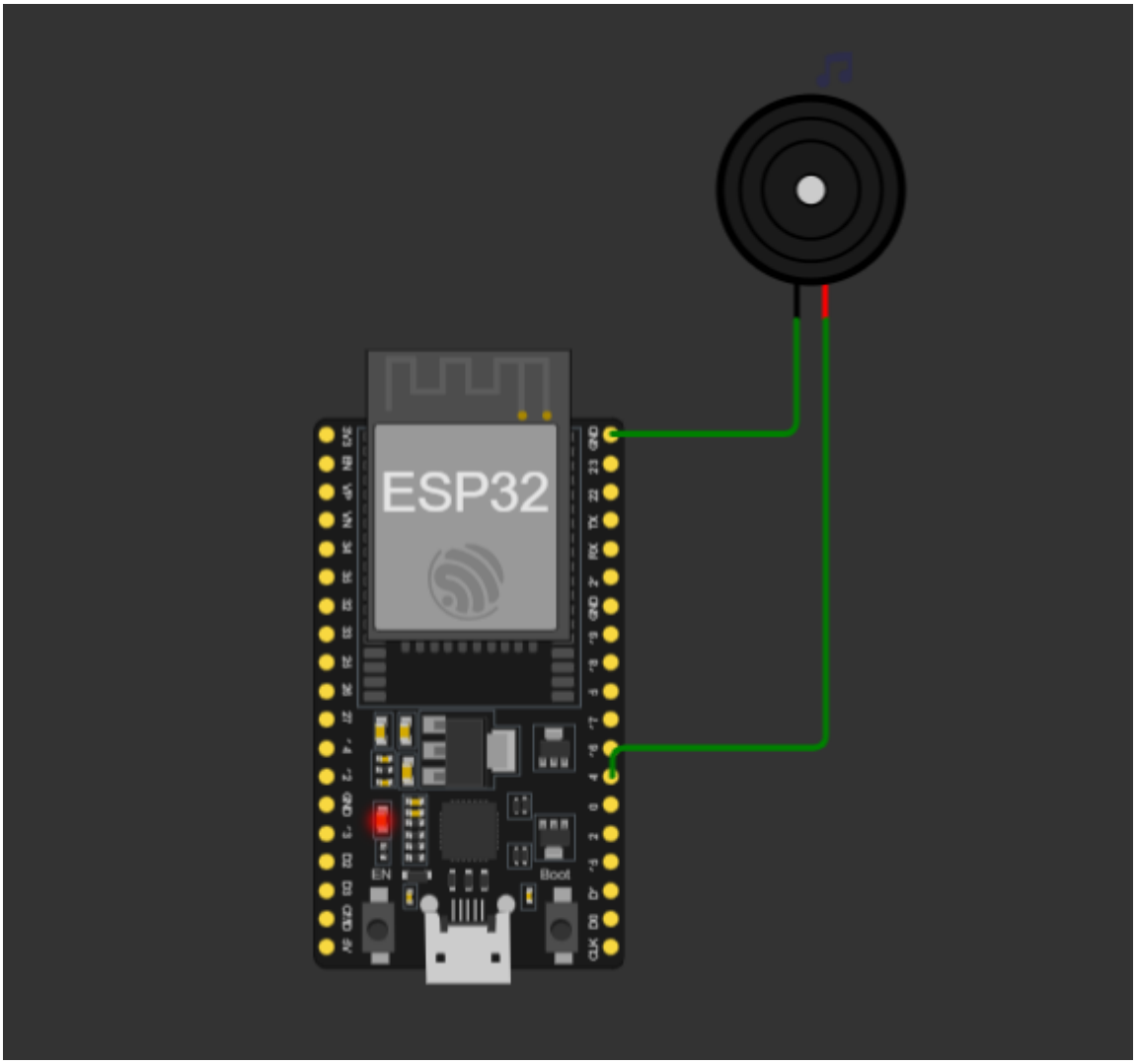
Los SPIFFS es un sistema de archivos que funcionan en las memorias SPI Flash NOR en objetivos integrados, justo este tipo de memorias es la que usa la placa ESP 32. Este sistema admite nivelación de desgaste, comprobaciones de coherencia del sistema de archivos y muchas cosas más.

Un punto importante a tener en cuenta sobre los SPIFFS es que no tienen una organización en directorios, es decir, carpetas como uno trabaja normalmente, sino que usa una estructura plana, por ejemplos si nosotros tenemos la siguiente dirección: /spiffs/tmp/hola.txt, cualquier persona con un poco de conocimiento sobre la organización de directorias pensaría que dentro de la carpeta spiffs, hay una carpeta llamada tmp, que a su vez contiene el archivo hola.txt. Sin embargo, así no es como funciona el sistema de archivos SPIFFS, en este caso lo

que nosotros tendríamos es un archivo llamado /tmp/hola.txt en SPIFFS, lo cual marca una sutil, pero importante diferencia.

Desarrollo

Diagramas (buzzer solo)



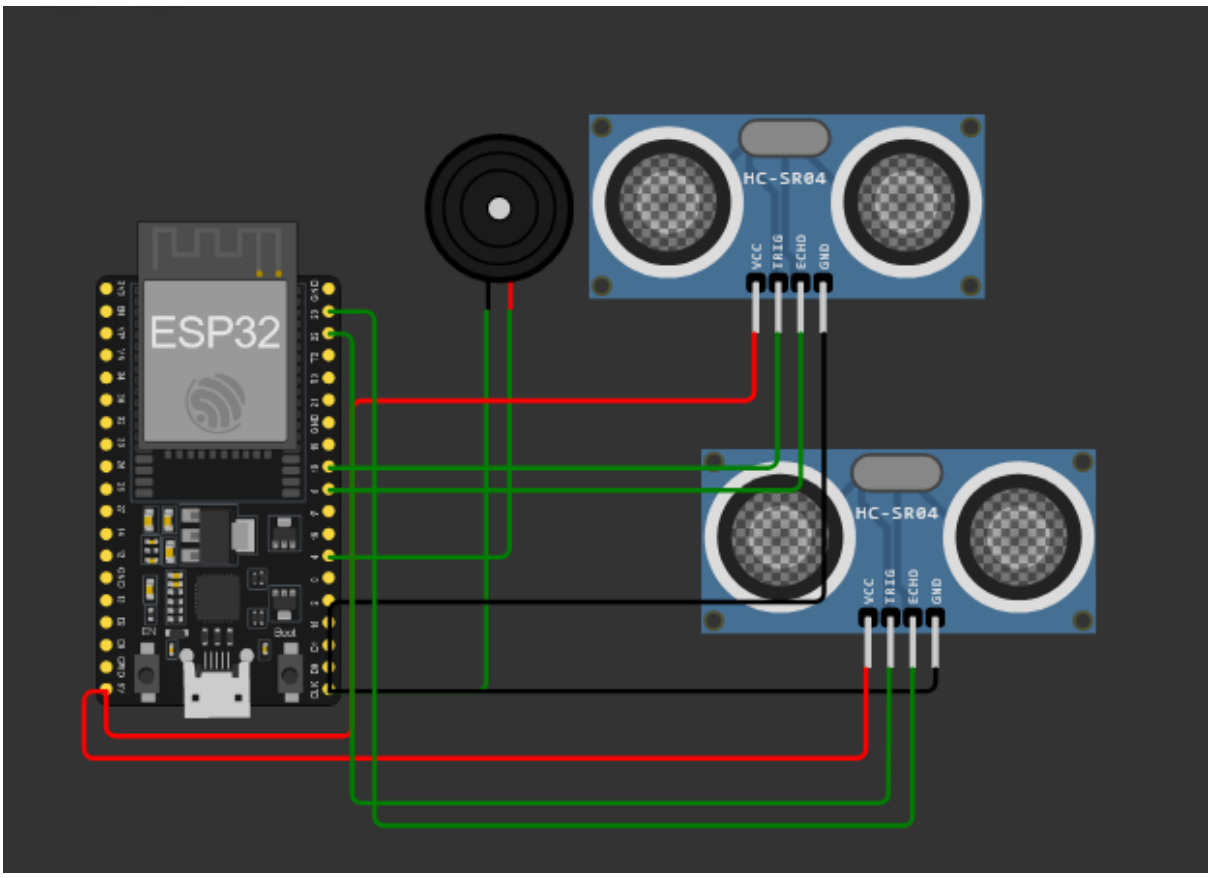
```cpp
#include <Arduino.h>

#define BUZZER_PIN 4 // D4 en el ESP32

void setup() {
  pinMode(BUZZER_PIN, OUTPUT);
}

void loop() {
  // Hacer sonar el buzzer durante 1 segundo
  digitalWrite(BUZZER_PIN, HIGH);
  delay(1000);
  digitalWrite(BUZZER_PIN, LOW);
  delay(1000); // Espera 1 segundo antes de volver a sonar el buzzer
}
```

Diagrama (buzzer con sensores)



```ino
#include <Arduino.h>
int trig1=18, trig2=22;
int echo1=15, echo2=23;
int pinBuzzer=4;

float duracion1 =0, distancia1 =0;
float duracion2 =0, distancia2 =0;

int frecuencia =2000, canal  =0 , resolucion =8;

int sonido =200;


void setup() {
  Serial.begin(115200);
  pinMode(trig1, OUTPUT);
  pinMode(trig2, OUTPUT);
  pinMode(echo1, INPUT);
  pinMode(echo2, INPUT);
  pinMode(2, OUTPUT);
  ledcSetup(canal,frecuencia,resolucion);
  ledcAttachPin(pinBuzzer, 0);
}

```

```ino

  bool det1 = false;
  void sensor1(){
    digitalWrite(trig1, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trig1, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig1, LOW);
    // Reads the echoPin, returns the sound wave travel time in microseconds
    duracion1 = pulseIn(echo1, HIGH);
    // Calculating the distance
    //distancia= duracion/58.2;//0.034/2;
    distancia1 = duracion1/120;
    // Prints the distance on the Serial Monitor
    Serial.print("Distance: ");
    Serial.println(distancia1);
  }

```

```
void sensor2(){
  digitalWrite(trig2, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trig2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig2, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duracion2 = pulseIn(echo2, HIGH);
  // Calculating the distance
  distancia2= duracion2/120;//0.034/2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(duracion2);
 }
```

```
void loop() {
  sensor1();
  sensor2();
 /*   if(duracion >=235 || duracion <)
  //sensor2();
 */
  if (distancia1 >=1 && distancia1 <=20 || (distancia2 >=1 && distancia2 <=20)){
    digitalWrite(2, HIGH);
    ledcWriteTone(0,sonido);
  }
  else{
    digitalWrite(2, LOW);
    ledcWriteTone(0,0);
  }
  delay(100);
}
```