

Entrega Parcial 1

(Código morse)

Grupo 6

Integrantes

Andreachi Santino – Brandon Ian

Fernández Cristian – Incutti Mateo

Panzera Lucas

Material a necesitar:

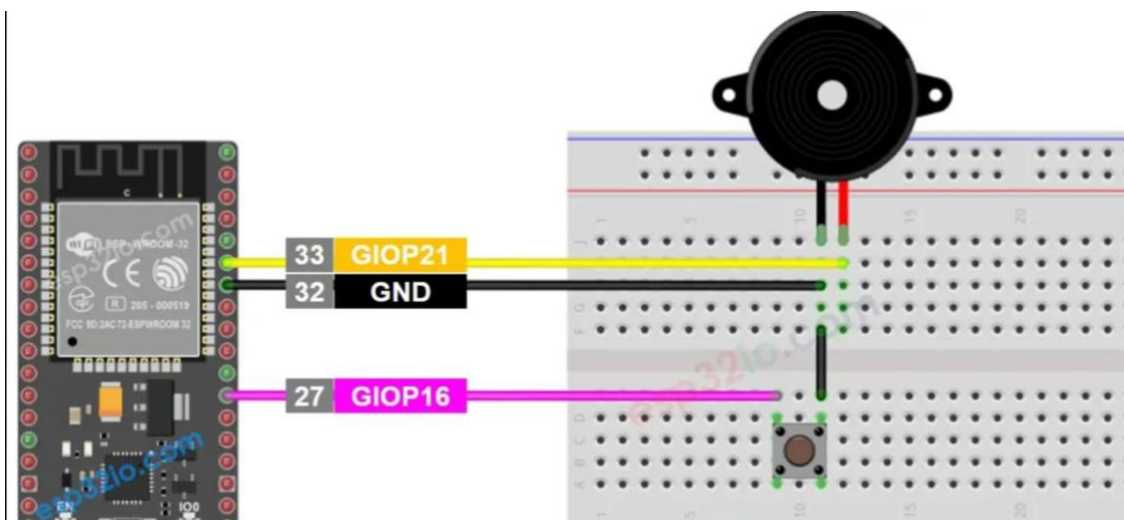
- ESP32
- Altavoz/Buzzer
- Cables
- Switch

Descripción:

Vamos a cargar un código en el ESP32 de tal manera que haga sonar determinado código morse en el Buzzer de la sala al presionar un botón.

También habrá una tabla que indica cómo es el alfabeto en morse para que los jugadores se puedan ayudar con eso.

El Buzzer va a sonar las veces y el tiempo que corresponda para formar la clave que se necesita para acceder a la computadora con el “preguntados”.



Grupo 6

```
#define buzzerPin 13
#define switchPin 8

void setup(){
    pinMode (buzzerPin, OUTPUT);
    pinMode (switchPin, INPUT);
}

//Estructuras

void punto(){ digitalWrite(buzzerPin, HIGH); delay(1000);
               digitalWrite(buzzerPin, LOW);  delay(1000);}

void barra(){ digitalWrite(buzzerPin, HIGH); delay(2000);
               digitalWrite(buzzerPin, LOW);  delay(1000);}

//Símbolos

void letraA(){ punto(); barra(); delay(2000); }
void letraB(){ barra(); punto(); punto(); punto(); delay(2000); }
void letraC(){ barra(); punto(); barra(); punto(); delay(2000); }
void letraD(){ barra(); punto(); punto(); delay(2000); }
void letraE(){ punto(); delay(2000); }
void letraF(){ punto(); punto(); barra(); punto(); delay(2000); }
void letraG(){ barra(); barra(); punto(); delay(2000); }
void letraH(){ punto(); punto(); punto(); punto(); delay(2000); }
void letraI(){ punto(); punto(); delay(2000); }
void letraJ(){ punto(); barra(); barra(); barra(); delay(2000); }
void letraK(){ barra(); punto(); barra(); delay(2000); }
void letraL(){ punto(); barra(); punto(); punto(); delay(2000); }
void letraM(){ barra(); barra(); delay(2000); }
void letraN(){ barra(); punto(); delay(2000); }
void letraO(){ barra(); barra(); barra(); delay(2000); }
void letraP(){ punto(); barra(); barra(); punto(); delay(2000); }
void letraQ(){ barra(); barra(); punto(); barra(); delay(2000); }
void letraR(){ punto(); barra(); punto(); delay(2000); }
void letraS(){ punto(); punto(); punto(); delay(2000); }
void letraT(){ barra(); delay(2000); }
void letraU(){ punto(); punto(); barra(); delay(2000); }
void letraV(){ punto(); punto(); punto(); barra(); delay(2000); }
void letraW(){ punto(); barra(); barra(); delay(2000); }
void letraX(){ barra(); punto(); punto(); barra(); delay(2000); }
void letraY(){ barra(); punto(); barra(); barra(); delay(2000); }
void letraZ(){ barra(); barra(); punto(); punto(); delay(2000); }
void cero() { barra(); barra(); barra(); barra(); barra();
delay(2000); }
void uno() { punto(); barra(); barra(); barra(); barra();
delay(2000); }
void dos() { punto(); punto(); barra(); barra(); barra();
delay(2000); }
void tres() { punto(); punto(); punto(); barra(); barra();
delay(2000); }
void cuatro(){ punto(); punto(); punto(); punto(); barra();
delay(2000); }
void cinco() { punto(); punto(); punto(); punto(); punto();
delay(2000); }
```

Grupo 6

```
void seis() { barra(); punto(); punto(); punto(); punto();  
delay(2000); }  
void siete() { barra(); barra(); punto(); punto(); punto();  
delay(2000); }  
void ocho() { barra(); barra(); barra(); punto(); punto();  
delay(2000); }  
void nueve() { barra(); barra(); barra(); barra(); punto();  
delay(2000); }
```

//Contraseñas

```
void contraseniaI() { letraP(); letraO(); dos(); cuatro(); }  
void contraseniaIII() {}  
void contraseniaIV() {}  
void contraseniaV() {}  
void contraseniaVI() {}  
void contraseniaVII() {}  
void contraseniaVIII() {}  
void contraseniaIX() {}  
void contraseniaX() {}  
  
void loop() {  
    int estado=digitalRead(switchPin);  
    if(estado==HIGH) {  
        contraseniaI();  
    }  
}
```