

ΦΥΛΛΟ ΠΡΑΞΗΣ

ΕΠΑΛ ΣΥΚΕΩΝ

ΕΚΠ. ΔΟΝΟΥΛΗΣ Α. ΠΕ83

Μαθητής:.....

Τάξη – Τμήμα: Γ' Ηλεκτρονικών

Εργαστήριο Συστημάτων Ασφαλείας

Ελέγχου

Ημερομηνία: / /

Αριθμός άσκησης: 2^η

Θέμα άσκησης: Συναγερμός με τέσσερις εισόδους, flashing led και μελωδική σειράνα στο Arduino

A. Θεωρητικό μέρος

Η δραστηριότητα αφορά στη δημιουργία συστήματος συναγερμού με είσοδο: Αισθητήρες Ανίχνευσης Κίνησης PIR - HC-SR501 (2 τεμ), μπουτόν πανικού και προσομοίωση μαγνητικών παγίδων (dip switch).

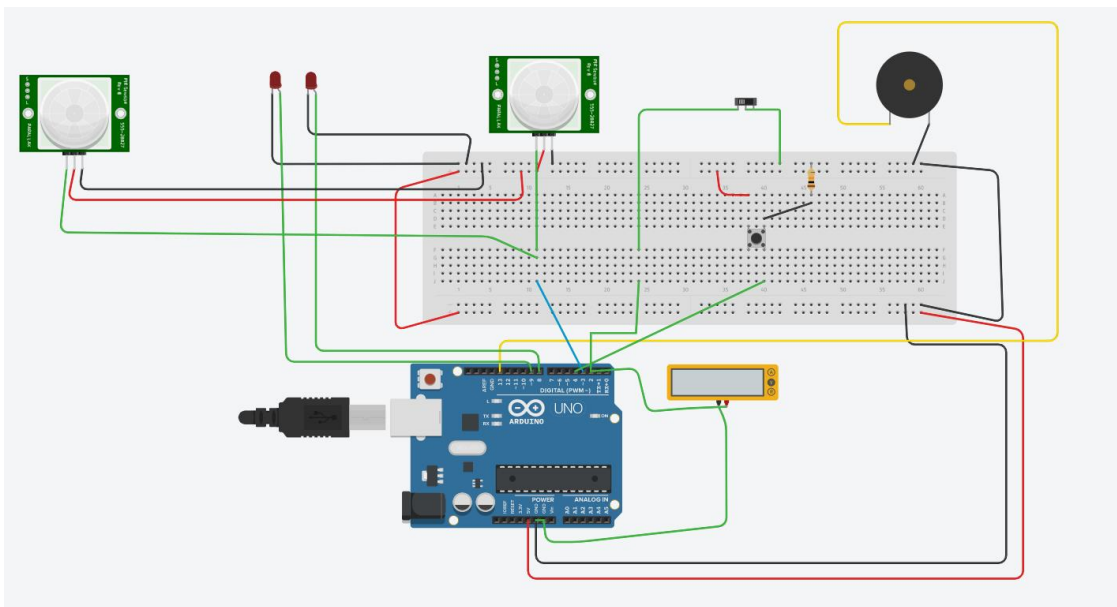
Ως έξοδος χρησιμοποιείται πιεζοστατικό buzzer. Οι μαθητές ενθαρρύνονται να δημιουργήσουν την δική τους μελωδία και την διάρκεια που επιθυμούν

B. Απαιτούμενα εργαλεία και υλικά

- 2x PIR Motion Sensor (HC-SR501)
- Arduino UNO
- 2x LED, αντίσταση 220Ω
- Buzzer
- Slide switch
- Jumper Καλώδια
- Tinkercad για προσομοίωση ή Arduino IDE για υλοποίηση

Γ. Πορεία Εργασίας

Φτιάχνω το παρακάτω κύκλωμα



Στην συνέχεια τρέχω το πρόγραμμα στον προσομοιωτή η στο IDE:

// ΣΥΝΑΓΕΡΜΟΣ 4 ΕΙΣΟΔΩΝ ΜΕ ΜΕΛΩΔΙΚΗ ΕΞΟΔΟ

// προσδιορίζουμε την συχνότητα κάθε νότας, εδώ δίνονται όλες

// οι νότες για περίπου 8 οκτάβες για να δημιουργήσουμε την

// μελωδία που θέλουμε

#define NOTE_B0 31

#define NOTE_C1 33

#define NOTE_CS1 35

#define NOTE_D1 37

#define NOTE_DS1 39

#define NOTE_E1 41

#define NOTE_F1 44

#define NOTE_FS1 46

#define NOTE_G1 49

#define NOTE_GS1 52

#define NOTE_A1 55

#define NOTE_AS1 58

#define NOTE_B1 62

#define NOTE_C2 65

#define NOTE_CS2 69

#define NOTE_D2 73

#define NOTE_DS2 78

#define NOTE_E2 82

#define NOTE_F2 87

#define NOTE_FS2 93

#define NOTE_G2 98

#define NOTE_GS2 104

#define NOTE_A2 110

#define NOTE_AS2 117

#define NOTE_B2 123

#define NOTE_C3 131

#define NOTE_CS3 139

#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988

```
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978
```

```
// παράδειγμα μελωδίας
```

```
int melody[] = {
    NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4, NOTE_F4, NOTE_E4,
    NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4, NOTE_G4, NOTE_F4,
    NOTE_C4, NOTE_C4, NOTE_C5, NOTE_A4, NOTE_F4, NOTE_E4, NOTE_D4,
```

```

NOTE_AS4,NOTE_AS4,NOTE_A4,NOTE_F4,NOTE_G4,NOTE_F4,
NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4, NOTE_F4, NOTE_E4,
NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4,NOTE_G4,NOTE_F4,
NOTE_C4, NOTE_C4,NOTE_C5,NOTE_A4,NOTE_F4, NOTE_E4,NOTE_D4,
NOTE_AS4,NOTE_AS4,NOTE_A4,NOTE_F4,NOTE_G4,NOTE_F4
};

```

// διάρκεια νότας 4 = τέταρτο , 8 = όγδοο ,κλπ

```

int noteDurations[] = {
4, 4,2,2,2,2,
4, 4,2,2,2,2,
4, 4,2,2,2,2,2,
4, 4,2,2,2,2,
4, 4,2,2,2,2,
4, 4,2,2,2,2,
4, 4,2,2,2,2,2,
4, 4,2,2,2,2
};

```

```
const byte BUZZ = 13; // PIN για την συνδεση του BUZZER
```

```
const byte interruptPin = 2;// οι dip διακόπτες
```

```
const byte interruptPin2 = 3;// οι δύο PIR
```

```
const int buttonPin = 4;
```

```
int buttonState = 0;
```

```
volatile byte state = LOW;
```

```
void setup() {
```

```
pinMode(BUZZ, OUTPUT);
```

```
pinMode(8, OUTPUT);
```

```
pinMode(9, OUTPUT);
```

```
pinMode(buttonPin, INPUT);
```

```
pinMode(interruptPin, INPUT_PULLUP);
```

```
// Οι dip διακόπτες δηλώνονται σε συνδεσμολογία INPUT_PULLUP
```

```
pinMode(interruptPin2, INPUT);
```

```
attachInterrupt(0, blink, RISING);/* ΠΡΟΣΟΧΗ στην μεθοδο
με διακοπές χρησιμοποιώ τα pin 2,3 τα οποία όμως δηλώνω
μεσα στην συνάρτηση attachInterrupt ως 0 και 1 αντίστοιχα */
attachInterrupt(1, blink2, CHANGE);

}
```

```
void loop() {
    digitalWrite(8,HIGH);

    digitalWrite(9,LOW);
    delay(1000);
    digitalWrite(8,LOW);

    digitalWrite(9,HIGH);
    delay(1000);
    buttonState = digitalRead(buttonPin);

    // ελέγχω το pushbutton
    if (buttonState == HIGH) {
        blink() ;
    }
    if (state != LOW) {

    for (int thisNote = 0; thisNote < 12; thisNote++) {

        // μπορώ να επέμβω στον counter για πιο σύντομη μελωδία
        //π.χ. το τέταρτο είναι= 1000 / 4, το όγδοο είναι = 1000/8, κτλ.
        int noteDuration = 1000 / noteDurations[thisNote];
        tone(13, melody[thisNote], noteDuration);
```

```
int pauseBetweenNotes = noteDuration +50;
delay(pauseBetweenNotes);
// σταματάει η μελωδία
noTone(13);
state=LOW;
}
}
}
//
void blink() {
    state = !state;
}
void blink2() {
    state = !state;
}
// μπορούμε να παρατηρήσουμε την διαφορετική διαχείριση
// των διακοπών με το pushbutton
```