Proiect PMP

Sistem de alarma inteligent

Student : Gligor Mihai

Grupa : 302310

Profesor coordonator : Meseșan Bogdan

Scurta descriere :

Obiectivul principal consta in creearea unui sistem de alarma inteligent care are ca modul principal un senzor cu ultrasunete care calculeaza distanta dintre un posibil „intrus” si casa , si afiseaza prin intermediul unui afisor cu 7 sergmente un mesaj corespunzator in functie de distanta , mesaje acompaniate de avertismente luminoase (led-uri rosii) si sonore(in cazul in care distanta este prea mica) . De asemenea sistemul are si un senzor cu infrarosu care ii permite controlul de la distanta (activarea si dezactivarea alarmei) prin intermediul unei telecomande .

Componente folosite :

1 x Breadboard 830p

1 x Arduino Nano

5 x Red LED

1 x Green LED

1 x Passive Buzzer

1 x IR Receiver Module

1 x Four Digits Seven Segments Display

1 x Ultrasonic Sensor Module

7 x 220 ohm resistors

n x fire

Cod :

#include <TimerOne.h>

#include "SR04.h"

#include "pitches.h"

#include "IRremote.h"

#define TRIG\_PIN 12

#define ECHO\_PIN 11

int melody[] = { NOTE\_C5, NOTE\_B5};

int duration = 700; // 500 miliseconds

int RECV\_PIN = 19; // the pin where you connect the output pin of IR sensor

IRrecv irrecv(RECV\_PIN);

decode\_results results;

const int digit1 = 5;

const int digit2 = 4;

const int digit3 = 3;

const int digit4 = 16;

const int A = 6;

const int B = 7;

const int C = 8;

const int D = 9;

const int E = 10;

const int F = 14;

const int G = 15;

const int RedLED = 18;

const int GreenLED = 13;

int segmente[]={A,B,C,D,E,F,G};

int codesForStop[]={109,49,63,115};

int codesFor112[]={6,6,91};

int codesForCALL[]={57,119,56,56};

int codesForALLE[]={119,56,56,121};

int codesForLLEd[]={56,56,121,94};

SR04 sr04 = SR04(ECHO\_PIN,TRIG\_PIN);

long volatile a;

int volatile tel=0;

/////////////////////////////////////////////////////////////////////////////////////////////////

void setup() {

//Timer1.initialize(10000);

//Timer1.attachInterrupt(StopF);

pinMode(A, OUTPUT);

pinMode(B, OUTPUT);

pinMode(C, OUTPUT);

pinMode(D, OUTPUT);

pinMode(E, OUTPUT);

pinMode(F, OUTPUT);

pinMode(G, OUTPUT);

pinMode(RedLED, OUTPUT);

pinMode(GreenLED, OUTPUT);

pinMode(digit1,OUTPUT);

pinMode(digit2,OUTPUT);

pinMode(digit3,OUTPUT);

pinMode(digit4,OUTPUT);

digitalWrite(digit1, HIGH);

digitalWrite(digit2, HIGH);

digitalWrite(digit3, HIGH);

digitalWrite(digit4, HIGH);

tel=0;

Serial.begin(9600);

irrecv.enableIRIn();

}

/////////////////////////////////////////////////////////////////////////////////////////////////

void loop() {

StopF();

if(tel==1)

{

digitalWrite(GreenLED,LOW);

a=sr04.Distance();

Serial.println(a);

if(a>15 && a<40){

digitalWrite(RedLED,HIGH);

StopF();

//noInterrupts();

for(int i=0;i<60;i++){

for(int n=1;n<5;n++){

writeOnDigit(n,codesForStop[n-1]);

}

//interrupts();

StopF(); }

digitalWrite(RedLED,LOW);

}

if(a<15){

StopF();

//noInterrupts();

digitalWrite(RedLED,HIGH);

for (int thisNote = 0; thisNote < 2; thisNote++) {

tone(17, melody[thisNote], duration);

delay(400);

}

digitalWrite(RedLED,LOW);

for(int i=0;i<100;i++){

for(int n=1;n<4;n++){

writeOnDigit(n,codesFor112[n-1]);

}

}

for(int i =0;i<80;i++) {

for(int n=1;n<5;n++){

writeOnDigit(n,codesForCALL[n-1]);

}

}

digitalWrite(RedLED,HIGH);

for(int i =0;i<80;i++){

for(int n=1;n<5;n++){

writeOnDigit(n,codesForALLE[n-1]);

}

}

digitalWrite(RedLED,LOW);

for(int i =0;i<80;i++){

for(int n=1;n<5;n++){

writeOnDigit(n,codesForLLEd[n-1]);

}

}

//interrupts();

StopF();

}

}else

{

digitalWrite(GreenLED,HIGH);

}

}

/////////////////////////////////////////////////////////////////////////////////////////////////

void StopF(){

Serial.println("e in intrerupere");

if (irrecv.decode(&results))// Returns 0 if no data ready, 1 if data ready.

{

int resultsValue = &results;// Results of decoding are stored in result.value

Serial.print("Code: ");

Serial.println(resultsValue); //prints the value a a button press

irrecv.resume(); // Restart the ISR state machine and Receive the next value

if(tel==0)

{

tel=1;

}

else

{

tel=0;

}

}

}

/////////////////////////////////////////////////////////////////////////////////////////////////

void writeOnDigit(int nr,int bt){

chooseDigit(nr);

writeOnSeg(bt);

delay(2);

allHIGH();

}

/////////////////////////////////////////////////////////////////////////////////////////////////

void writeOnSeg(byte b){

for(int i=0;i<7;i++){

if(bitRead(b,0)==1){

digitalWrite(segmente[i], HIGH);

}

else{

digitalWrite(segmente[i], LOW);

}

b=b>>1;

}

}

/////////////////////////////////////////////////////////////////////////////////////////////////

void allHIGH(){

digitalWrite(digit1, HIGH);

digitalWrite(digit2, HIGH);

digitalWrite(digit3, HIGH);

digitalWrite(digit4, HIGH);

}

/////////////////////////////////////////////////////////////////////////////////////////////////

void chooseDigit(int nr){

if(nr==1){

digitalWrite(digit1, LOW);

}

if(nr==2){

digitalWrite(digit2, LOW);

}

if(nr==3){

digitalWrite(digit3, LOW);

}

if(nr==4){

digitalWrite(digit4, LOW);

}

}