**Sistem de alarma**

Semigrupa 1.1

BORLEA CATALIN

BOCIORT DINU

BALABAN CALIN

NEAGU MIA

1. Obiectivul temei şi potenţiala utilitate în lumea reală

Obiectivul echipei este de a programa un sistem de alarmă care este utilizat pentru depistarea şi anunţarea unor intruşi care au pătruns în perimetrul securizat.

1. Componentele montajului

Montajul este compus dintr-un sensor de lumină, LED-uri, minidifuzor, sensor vibraţii, shield LCD 16X2 şi un buton.

1. Acţiunile asociate cu programul în C şi schema logică

Limbajul folosit se aseamănă cu limbajul C prin sintaxa identică a instrucţiunilor, existent funcţiilor, operatorii sunt aceeaşi şi îndeplinesc aceleaşi funcţii.

1. Sursa in limbaj C

#include <LiquidCrystal.h>

LiquidCrystal lcd(7,6,5,4,3,2);

int lr=11, lb=12;

const int buttonPin = 10; // pinul de care e atasat butonului

const int lg = 13; // pinul de care e etasat LED-ul

// Variables will change:

int buttonPushCounter = 0; // contor pentru numarul apasarilor butonului

int buttonState = 0; // starea actuala a butonului

int lastButtonState = 0; // ultima stare a butonului

void setup() {

Serial.begin(9600);

lcd.begin(16, 2);

pinMode(lr, OUTPUT);

pinMode(lb, OUTPUT);

// initialize the button pin as a input:

pinMode(buttonPin, INPUT);

// initialize the LED as an output:

pinMode(lg, OUTPUT);

}

void loop() {

// citim pinul butonului

buttonState = digitalRead(buttonPin);

// comparam buttonState cu starea anterioara

if (buttonState != lastButtonState) {

// daca statulul este schimbat incrementam contorul

if (buttonState == HIGH) {

// daca statulul curent este HIGH atunci butonul

// merge incet de la off la on:

buttonPushCounter++;

Serial.println("on");

Serial.print("number of button pushes: ");

Serial.println(buttonPushCounter);

} else {

// daca statutul curent este LOW atunci butonul

// merge incet de la on la off:

Serial.println("off");

}

// amanati putin pentru a evita bouncing

delay(50);

}

// salvati statutul curent ca cel anteriorsave,

//pentru data viitore prin loop

lastButtonState = buttonState;

// porneste LED-ul la fiecare 4 apasari ale butonului

// verificarea modulo contorului butonului

// functia modulo va da restul

// divizarea a doua numere:

if (buttonPushCounter%2==0) {

digitalWrite(lr, HIGH);

digitalWrite(lg, LOW);

}

else {

digitalWrite(lr, LOW);

digitalWrite(lg, HIGH);

}

int valoareIluminare = analogRead(A0);

Serial.println(valoareIluminare, DEC);

delay(10); //verificarea intensita?ii luminii

if(valoareIluminare<15)

{

lcd.setCursor(0, 0);

lcd.print("ARMAT!"); //afisarea mesajului pe timp de noapte

lcd.print(" ");

delay(300);

Serial.println(analogRead(A1)); //verificarea senzorului de vibra?ii

if(analogRead(A1)>0) //declan?area alarma

{

int x=0; //alarma este activa cât timp x este mai mic decât 50

while(x<50) //repetarea de x ori

{

lcd.setCursor(0, 0);

lcd.print("ALARMA!"); //afisarea pe ecranul LCD un mesaj in caz de alarma

lcd.print(" "); //aprinderea led-urilor

digitalWrite(lr, HIGH);

delay(50);

digitalWrite(lb, HIGH);

delay(50);

digitalWrite(lr, LOW);

delay(50);

digitalWrite(lb, LOW);

delay(50); //declan?area ton

tone(8,1200);

delay(50);

tone(8,1200);

delay(50);

x++;

}

}

}

else

{

lcd.setCursor(0, 0);

lcd.print("Dezarmat!"); //afisarea mesajului pe timp de zi

lcd.print(" ");

}

}

1. Fotografii





1. Plan de testare

Am verificat fiecare functie daca merge, in urma executie sa aiba loc actiunea specifica functiei, daca se aprind Led-urile corect sau daca merge butonul, afisarea mesajelor pe Display.

1. Concluzii

Am invatat sa programam in Arduino. A fost greu la inceput pana am inteles cum sa folosim functiile, ce trebuie sa folosim si cum. Am deposit momentul de la inceput lucrand in echipa, toti mobilizandu-ne sa cautam solutia optima. Am folosit Liquid cristal pentru initializarea ecranul LCD, am folosit constant ca HIGH, LOW.