

**KASTAMONU ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**GÖMÜLÜ SİSTEMLER**

**Akıllı Ev**

**Raporu Yazan Öğrenci:** Ramazan ŞAHİN

**Öğrenci No:** 134410008

**KODLAR**

//Tus Takimi

#include <Password.h>

#include <Keypad.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(38,39,36,37,34,35);//

const byte satirSayisi= 4;

const byte sutunSayisi= 4;

char tuslar[satirSayisi][sutunSayisi]=

{

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'\*', '0', '#', 'D'}

};

byte satirPinleri[satirSayisi] = {29,28,27,26};

byte sutunPinleri[sutunSayisi]= {25,24,23,22};

Keypad tusTakimi= Keypad(makeKeymap(tuslar), satirPinleri, sutunPinleri, satirSayisi, sutunSayisi);

Password password = Password( "1453" );

//Display

int digit1 = 11; //PWM Display pin 1

int digit2 = 10; //PWM Display pin 2

int digit3 = 9; //PWM Display pin 6

int digit4 = 6; //PWM Display pin 8

int segA = A1; //Display pin 14

int segB = 3; //Display pin 16

int segC = 4; //Display pin 13

int segD = 5; //Display pin 3

int segE = A0; //Display pin 5

int segF = 7; //Display pin 11

int segG = 8; //Display pin 15

int sicaklik;

#include "DHT.h"

#define DHT11PIN 49 // DHT11PIN olarak Dijital 2'yi belirliyoruz.

#define DHTPIN 49

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

int ses;

int smoke\_level;//AlevSensoru

int val;//HareketSensoru

int pirState = LOW;

void setup() {

//Tus Takimi

lcd.begin(16,2);

lcd.setCursor(0,0);

lcd.print("Sifre : ");

tusTakimi.addEventListener(keypadEvent);

//Alev Sensörü

Serial.begin(9600); //sets the baud rate for data transfer in bits/second

pinMode(A2, INPUT);//the smoke sensor will be an input to the arduino

pinMode(52, OUTPUT);//the buzzer serves an output in the circuit

//Hareket Sensörü

pinMode(51,OUTPUT);

//LDR Sensörü

pinMode(48,OUTPUT);

//Su Sensörü

pinMode(46,OUTPUT);

//Display

pinMode(segA, OUTPUT);

pinMode(segB, OUTPUT);

pinMode(segC, OUTPUT);

pinMode(segD, OUTPUT);

pinMode(segE, OUTPUT);

pinMode(segF, OUTPUT);

pinMode(segG, OUTPUT);

pinMode(digit1, OUTPUT);

pinMode(digit2, OUTPUT);

pinMode(digit3, OUTPUT);

pinMode(digit4, OUTPUT);

dht.begin();

}

void loop()

{

sicaklikOlc();// 'C cinsinden.

displayNumber(sicaklik);

alevSensoru();

hareketSensoru();

ldrSensoru();

suSensoru();

tusTakimi.getKey();

}

void keypadEvent(KeypadEvent eKey){

switch (tusTakimi.getState()){

case PRESSED:

if(eKey!='\*' && eKey!='#'){

if(eKey!='A'&& eKey!='B'){

if(eKey!='D' && eKey!='C'){

Serial.print("Sifre :");

lcd.print(eKey);

Serial.println(eKey);

tone(31,200);

delay(100);

noTone(31);

}

}

}

if(eKey=='#' || eKey=='C'){

tone(31,200);

delay(100);

noTone(31);

}

delay(10);

switch (eKey){

case '#': Serial.println("Sifre Deneniyor...");lcd.clear();lcd.setCursor(0,1);lcd.print("Sifre Deneniyor..."); delay(2000); checkPassword(); delay(1); password.reset();lcd.clear();lcd.setCursor(0,0);lcd.print("Sifre : "); break;

case 'A': break;

case 'B': break;

case 'D': break;

case '\*': break;

case 'C': password.reset(); Serial.println("Temizleniyor..."); lcd.clear();lcd.setCursor(0,1);lcd.print("Temizleniyor...");delay(1000);lcd.clear();lcd.setCursor(0,0);lcd.print("Sifre : "); delay(1); break;

default: password.append(eKey); delay(1);

}

}

}

void checkPassword(){

if (password.evaluate()){

Serial.println("KAPI ACILDI");

lcd.clear();

lcd.setCursor(0,0);

lcd.print("KAPI ACILDI");

for(int i=0;i<5;i++){

tone(31,500);

delay(100);

noTone(31);

delay(100);

}

delay(1000); //5 sn bekle

}

else{

Serial.println("Sifre HATALI");

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Sifre HATALI");

for(int i=10;i>=0;i--){

tone(31,500);

delay(100);

noTone(31);

}

delay(1000); //5sn Bekle

}

}

void suSensoru()

{

int sensorValue = analogRead(A4);

Serial.println(sensorValue);

if(sensorValue>350){

for(int i=0;i<5;i++){

tone(47,500);

digitalWrite(46,HIGH);

delay(100);

noTone(47);

digitalWrite(46,LOW);

delay(100);

}

}

else{

noTone(47);

digitalWrite(46,LOW);

}

}

void sicaklikOlc(){

sicaklik = (int)(dht.readTemperature());// 'C cinsinden.

}

void ldrSensoru()

{

int LDRReading = analogRead(A3);

if(LDRReading<40)

digitalWrite(48,HIGH);

else

digitalWrite(48,LOW);

}

void hareketSensoru()

{

val = digitalRead(50); //read state of the PIR

if (val == HIGH) {

digitalWrite(51,HIGH);

if(pirState==LOW){

pirState=HIGH;

}

}

else {

digitalWrite(51,LOW);

if(pirState==HIGH){

pirState=LOW;

}

}

}

void alevSensoru()

{

smoke\_level= analogRead(A2); //arduino reads the value from the smoke sensor

if(smoke\_level < 800)

{ //if smoke level is greater than 500, the buzzer will go off

for(int i=0;i<5;i++){

tone(53,500);

digitalWrite(52,HIGH);

delay(100);

noTone(53);

digitalWrite(52,LOW);

delay(100);

}

}

else

{

noTone(53);

digitalWrite(52,LOW);

}

}

////Display

void displayNumber(int toDisplay) {

#define DISPLAY\_BRIGHTNESS 500

#define DIGIT\_ON HIGH

#define DIGIT\_OFF LOW

long beginTime = millis();

for(int digit = 4 ; digit > 0 ; digit--) {

//Turn on a digit for a short amount of time

switch(digit) {

case 1:

digitalWrite(digit1, DIGIT\_ON);

break;

case 2:

digitalWrite(digit2, DIGIT\_ON);

break;

case 3:

digitalWrite(digit3, DIGIT\_ON);

break;

case 4:

digitalWrite(digit4, DIGIT\_ON);

break;

}

//Turn on the right segments for this digit

lightNumber(toDisplay % 10);

toDisplay /= 10;

delayMicroseconds(DISPLAY\_BRIGHTNESS);

//Display digit for fraction of a second (1us to 5000us, 500 is pretty good)

//Turn off all segments

lightNumber(10);

//Turn off all digits

digitalWrite(digit1, DIGIT\_OFF);

digitalWrite(digit2, DIGIT\_OFF);

digitalWrite(digit3, DIGIT\_OFF);

digitalWrite(digit4, DIGIT\_OFF);

}

while( (millis() - beginTime) < 10) ;

//Wait for 20ms to pass before we paint the display again

}

//Given a number, turns on those segments

//If number == 10, then turn off number

void lightNumber(int numberToDisplay) {

#define SEGMENT\_ON LOW

#define SEGMENT\_OFF HIGH

switch (numberToDisplay){

case 0:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_OFF);

break;

case 1:

digitalWrite(segA, SEGMENT\_OFF);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_OFF);

break;

case 2:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_OFF);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_ON);

break;

case 3:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_ON);

break;

case 4:

digitalWrite(segA, SEGMENT\_OFF);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 5:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_OFF);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 6:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_OFF);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 7:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_OFF);

break;

case 8:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 9:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 10:

digitalWrite(segA, SEGMENT\_OFF);

digitalWrite(segB, SEGMENT\_OFF);

digitalWrite(segC, SEGMENT\_OFF);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_OFF);

break;

}

}