**(ΕΝΤΕΛΩΣ ΠΡΩΤΟΤΥΠΗ ΕΡΓΑΣΙΑ-ΟΧΙ COPY PASTE )**

**GYMNASIO-BRYSWN2-open robotics/ΓΥΜΝΑΣΙΟ ΒΡΥΣΩΝ ΟΜΑΔΑ2 ΝΟΜΟΣ ΧΑΝΙΩΝ**

ΜΑΚΕΤΑ ΕΞΥΠΝΗΣ ΠΟΛΗΣ-ΦΑΝΑΡΙΑ ΚΥΚΛΟΦΟΡΙΑΣ ΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΣΤΑ ΑΣΘΕΝΟΦΟΡΑ. Παρατηρώντας σε μια σχολική εκδρομή με λεωφορείο , σε κεντρικό δρόμο της πόλης μας ασθενοφόρο να ακινητοποιείται μπροστά μας σε μια μεγάλη ουρά αυτοκινήτων, με φανάρια κυκλοφορίας μπροστά και στο κάλεσμα της σειρήνας του ασθενοφόρου. τα αυτοκίνητα να μην μπορούν να ανοίξουν δρόμο παρά μόνο εάν τα φανάρια κυκλοφορίας μείνουν για αρκετό χρόνο πράσινα, γεννήθηκε η ιδέα μας . ΠΑΡΑΤΗΡΗΣΑΜΕ ΚΑΙ ΑΛΛΟΥ ΑΝΑΛΟΓΑ ΠΡΟΒΛΗΜΑΤΑ :<https://www.youtube.com/watch?v=l3hOBeQvc3M> ΚΑΙ <https://www.youtube.com/watch?v=0MQ7U7T0X8c> ΕΞΕΡΕΥΝΟΥΜΕ ΤΟΝ ΤΡΟΠΟ ΝΑ ΤΟΥΣ ΔΩΣΟΥΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΑ Η ΑΡΧΙΚΗ ΙΔΕΑ ΗΤΑΝ ΑΠΟ ΤΟ ΜΟΤΙΒΟ ΗΧΟΥ ΤΩΝ ΑΣΘΕΝΟΦΟΡΩΝ ΝΑ ΕΝΕΡΓΟΠΟΙΗΤΑΙ ΠΡΑΣΙΝΟ ΦAΝΑΡΙ ΓΙΑ ΑΥΤΑ ΚΟΚΚΙΝΟ ΓΙΑ ΤΑ ΑΛΛΑ. Η ΔΕΥΤΕΡΗ ΚΑΙ ΤΕΛΙΚΗ ΙΔΕΑ ΕΙΝΑΙ ΜΕ ΥΠΕΡΗΧΗΤΙΚΟ ΣΗΜΑ(**πατέντα η οποία σε αντίθεση με τους συνηθισμένους ασύρματους ηλεκτρομαγνητικούς διακόπτες χρησιμοποιεί το υπερηχητικό φάσμα :**[**https://github.com/nektarios25ma/Alternative-to-electromagnetic-signal-ULTRASOUND-TRIGGER-**](https://github.com/nektarios25ma/Alternative-to-electromagnetic-signal-ULTRASOUND-TRIGGER-) **,η απουσία ενός σύνηθους module π.χ. Bluetooth , με την πίεση των παιδιών να βρούμε τρόπο με έκανε να τροποποιήσω το HCSR04 ΚΑΙ Ως HARDWARE ΚΑΙ Ως SOFTWARE και έτσι να το μετατρέψω σε ασύρματο διακόπτη**) ΑΠΟ ΤΟ ΑΣΘΕΝΟΦΟΡΟ ΝΑ ΕΝΕΡΓΟΠΟΙΗΤΑΙ- ΑΣΥΡΜΑΤΑ -ΠΡΑΣΙΝΟ ΓΙΑ ΑΥΤΑ ΦΑΝΑΡΙ ΚΑΙ ΚΟΚΚΙΝΟ ΓΙΑ ΤΑ ΑΛΛΑ. Κάνοντας πειράματα με τον ultrasonic sensor HC sr04 και διαβάζοντας το datasheet παρατηρήσαμε ότι στέλνει ένα μοτίβο 8 υπερηχητικών παλμών 40khz και αναγνωρίζει ως επιστροφή μόνο το ίδιο μοτίβο άρα ο τυχόν περιβάλλον θόρυβος δεν μας απασχολεί. Επιμερισμός εργασιών: α' ομάδα μαθητών : σταδιακός φωτισμός πόλης-μακέτα (όσο μειώνεται ο περιβάλλον φωτισμός -ldr αισθητήρας- τόσο αυξάνει ο φωτισμός από τους στύλους φωτός - κώδικας και κατασκευή, β' ομάδα μαθητών : μπάρα-πύλη νοσοκομείου που ανοίγει όταν πλησιάζει το ασθενοφόρο και οθόνη που δείχνει την προσέλευση του ασθενοφόρου -κώδικας και κατασκευή, γ' ομάδα μαθητών : φανάρια και πεζοφάναρο **σε δύο δρόμους που τέμνονται κάθετα και δουλεύουν αρμονικά μεταξύ τους** και ταυτόχρονα δέκτης υπέρηχου, κώδικας και κατασκευή, δ' ομάδα μαθητών: αισθητική μακέτας - επίβλεψη , ε' ομάδα(Καθηγητής) : πατέντα υπερηχητικής σκανδάλης-πομπού και δέκτη και επίβλεψη όλου του έργου. ΥΛΙΚΑ ΠΟΥ ΘΑ ΧΡΕΙΑΣΤΟΥΝ α) 3 ARDUINO (1ο ως πομπό-ασθενοφόρο ,2ο δέκτης και ελεγκτής φαναριών, και 3ο για έξυπνο σταδιακό φωτισμό πόλης και μπάρα νοσοκομείου που ανοίγει με την προσέλευση του ασθενοφόρου β) 3 ultrasonic sensor γ) model-maquette city/ΜΑΚΕΤΑ ΠΟΛΗΣ δ)jumper wires/ΚΑΛΩΔΙΑ ΣΥΝΔΕΣΗΣ ε)led /led διαφόρων χρωμάτων στ) servo (SG90) η) 2 breadboard θ) LCD 16X2

Φωτογραφίες των μαθητών επί το έργον



ΤΟ ΛΙΝΚ ΤΗΣ «ΠΑΤΕΝΤΑΣ»

<https://github.com/nektarios25ma/Alternative-to-electromagnetic-signal-ULTRASOUND-TRIGGER->

ΤΑ ΒΙΝΤΕΟ:

1η ΠΡΟΒΑ

<https://www.youtube.com/watch?v=jfiOYmjc7f0&t=83s>

ΑΠΟ ΤΑ ΠΑΙΔΙΑ: <https://www.youtube.com/watch?v=UQ0LskqQZeM&t=1s>

Η ΦΩΝΗΤΙΚΗ ΠΡΟΣΘΗΚΗ-ΣΕ ΣΥΝΔΕΣΗ ΜΕ ΘΕΡΜΟΚΡΑΣΙΑ ΥΓΡΑΣΙΑ: <https://www.youtube.com/watch?v=sbNdtB8n2A0&t=1s>

ΚΑΙ ΤΩΡΑ ΟΙ ΚΩΔΙΚΕΣ , ΠΡΩΤΟΣ ΤΑ ΕΞΥΠΝΑ ΦΑΝΑΡΙΑ ΚΑΙ ΤΑΥΤΟΧΡΟΝΑ ΔΕΚΤΗΣ ΥΠΕΡΗΧΗΤΙΚΟΥ ΣΗΜΑΤΟΣ,ΔΕΥΤΕΡΟΣ ΡΥΘΜΙΣΗ ΜΠΑΡΑΣ ΟΘΟΝΗΣ-ΕΝΔΕΙΞΗ ΘΕΡΜΟΚΡΑΣΙΑΣ ΥΓΡΑΣΙΑΣ-ΦΩΝΗΤΙΚΗΣ ΠΡΟΕΙΔΟΠΟΙΗΣΗΣ ΣΤΗΝ ΥΨΗΛΗ Ή ΧΑΜΗΛΗ ΘΕΡΜΟΚΡΑΣΙΑ- ΤΗΣ ΣΤΑΔΙΑΚΗΣ ΦΩΤΕΙΝΟΤΗΤΑΣ ΤΗΣ ΠΟΛΗΣ(ΟΣΟ ΠΕΡΙΣΣΟΤΕΡΟ ΣΚΟΤΑΔΙ ΤΟΣΟ ΠΙΟ ΠΟΛΥ ΦΩΤΙΖΟΥΝ) ΚΑΙ ΤΑΥΤΟΧΡΟΝΑ ΔΕΚΤΗΣ ΥΠΕΡΗΧΗΤΙΚΟΥ ΣΗΜΑΤΟΣ ΚΑΙ ΤΡIΤΟΣ ΚΑΙ ΤΕΛΕΥΤΑΙΟΣ Ο ΠΟΜΠΟΣ ΤΟΥ ΥΠΕΡΗΧΗΤΙΚΟΥ ΜΟΤΙΒΟΥ

Α)

unsigned long start\_time; // Δημιούργησε μια μεταβλητή τύπου unsigned long integer με όνομα start\_time

#define echo 12 //Ο ΔΕΚΤΗΣ ΤΟΥ ΥΠΕΡΗΧΟΥ

#define trigger 13 // Ο ΠΟΜΠΟΣ ΠΟΥ ΨΕΥΔΟΣΤΕΛΝΕΙ ΕΔΩ

// ΟΡΙΣΜΟΣ ΦΑΝΑΡΙΩΝ Α ΚΥΡΙΟΥ ΔΡΟΜΟΥ ΠΡΟΣ ΝΟΣΟΚΟΜΕΙΟ

int ledRed = 11;

int ledOrange = 10;

int ledGreen = 9;

int pedRed = 5;

int pedGreen = 4;

// ΟΡΙΣΜΟΣ ΦΑΝΑΡΙΩΝ Β ΚΑΘΕΤΟΥ ΔΡΟΜΟΥ

int ledRedB = 8;

int ledOrangeB = 7;

int ledGreenB = 6;

int pedRedB = 3;

int pedGreenB = 2;

// ΣΤΗΝ ΟΥΣΙΑ Η ΠΑΡΑΚΑΤΩ ΛΕΙΤΟΥΡΓΙΑ ΕΛΕΓΧΕΙ ΣΥΝΕΧΩΣ ΕΑΝ ΥΠΑΡΧΕΙ //ΥΠΕΡΗΧΟΣ

long TP\_init()

{

digitalWrite(trigger,LOW);

delayMicroseconds(2);

digitalWrite(trigger,HIGH);

delayMicroseconds(2);

digitalWrite(trigger,LOW);

delayMicroseconds(2);

long microseconds=pulseIn(echo,HIGH,100000);

return microseconds ;

}

void setup() {

pinMode(echo,INPUT);

pinMode(trigger,OUTPUT);

Serial.begin(9600);

pinMode(ledRed, OUTPUT);

pinMode(ledOrange, OUTPUT);

pinMode(ledGreen, OUTPUT);

pinMode(pedRed, OUTPUT);

pinMode(pedGreen, OUTPUT);

pinMode(ledRedB, OUTPUT);

pinMode(ledOrangeB, OUTPUT);

pinMode(ledGreenB, OUTPUT);

pinMode(pedRedB, OUTPUT);

pinMode(pedGreenB, OUTPUT);

}

void loop() {

//?Serial.println(TP\_init());

while (TP\_init()>= 5){ // Αν ανιχνευτει υπερηχος κάνε:

digitalWrite(ledRed, LOW);

digitalWrite(ledOrange, LOW);

digitalWrite(ledGreen, HIGH);

digitalWrite(ledRedB, HIGH);

digitalWrite(ledOrangeB, LOW);

digitalWrite(ledGreenB, LOW);

digitalWrite(pedRed, HIGH);

digitalWrite(pedGreen, LOW);

digitalWrite(pedRedB, LOW);

digitalWrite(pedGreenB, HIGH);

delay(100000); //καθυστερησε 100 sec

}

//? Serial.println(TP\_init());

if ( TP\_init()<= 5 || TP\_init() >=29930 ) { // Αν ο δεν ανιχνευτει υπερηχος

// κόκκινο για 3 δευτερόλεπτα, πράσινο στους πεζούς

start\_time=millis(); // Βάλε στην start\_time την τιμή της millis()

while (millis() - start\_time < 3000) { // Εφόσον δεν πέρασαν 3 δευτ κάνε:

digitalWrite(ledRed, HIGH);

digitalWrite(ledOrange, LOW);

digitalWrite(ledGreen, LOW);

digitalWrite(pedRed, LOW);

digitalWrite(pedGreen, HIGH);

digitalWrite(ledRedB, LOW);

digitalWrite(ledOrangeB, LOW);

digitalWrite(ledGreenB, HIGH);

digitalWrite(pedRedB, HIGH);

digitalWrite(pedGreenB, LOW);}}

else {

start\_time=millis(); // Βάλε στην start\_time την τιμή της millis()

while (millis() - start\_time < 100000) { // Εφόσον δεν πέρασαν 100 δευτ κάνε:

// πράσινο κυκλοφορίας για το ασθενοφορο

digitalWrite(ledRed, LOW);

digitalWrite(ledOrange, LOW);

digitalWrite(ledGreen, HIGH);

digitalWrite(ledRedB, HIGH);

digitalWrite(ledOrangeB, LOW);

digitalWrite(ledGreenB, LOW);

digitalWrite(pedRed, HIGH);

digitalWrite(pedGreen, LOW);

digitalWrite(pedRedB, LOW);

digitalWrite(pedGreenB, HIGH);

}}

if ( TP\_init()<= 5 || TP\_init() >=29930 ) { // Αν ο δεν ανιχνευτει υπερηχος

// κόκκινο στους πεζούς, περιμένω για 1 δευτερόλεπτο πριν δώσω πράσινο στα αμάξια

start\_time=millis();

while (millis() - start\_time < 1000) { // Εφόσον δεν πέρασαν 1 δευτ κάνε:

digitalWrite(pedRed, HIGH);

digitalWrite(pedGreen, LOW);

digitalWrite(ledRed, HIGH);

digitalWrite(ledOrange, LOW);

digitalWrite(ledGreen, LOW);

digitalWrite(ledRedB, LOW);

digitalWrite(ledOrangeB, HIGH);

digitalWrite(ledGreenB, LOW);

digitalWrite(pedRedB, HIGH);

digitalWrite(pedGreenB, LOW); }}

else {

start\_time=millis(); // Βάλε στην start\_time την τιμή της millis()

while (millis() - start\_time < 100000) { // Εφόσον δεν πέρασαν 100 δευτ κάνε:

// πράσινο κυκλοφορίας για το ασθενοφορο

digitalWrite(ledRed, LOW);

digitalWrite(ledOrange, LOW);

digitalWrite(ledGreen, HIGH);

digitalWrite(ledRedB, HIGH);

digitalWrite(ledOrangeB, LOW);

digitalWrite(ledGreenB, LOW);

digitalWrite(pedRed, HIGH);

digitalWrite(pedGreen, LOW);

digitalWrite(pedRedB, LOW);

digitalWrite(pedGreenB, HIGH);

}}

if ( TP\_init()<= 5 || TP\_init() >=29930 ) {

// πράσινο κυκλοφορίας για 5 δευτερόλεπτα

start\_time=millis();

while (millis() - start\_time < 5000) { // Εφόσον δεν πέρασαν 5 δευτ κάνε:

digitalWrite(ledRed, LOW);

digitalWrite(ledOrange, LOW);

digitalWrite(ledGreen, HIGH);

digitalWrite(ledRedB, HIGH);

digitalWrite(ledOrangeB, LOW);

digitalWrite(ledGreenB, LOW);

digitalWrite(pedRed, HIGH);

digitalWrite(pedGreen, LOW);

digitalWrite(pedRedB, LOW);

digitalWrite(pedGreenB, HIGH);}}

else{

// πράσινο κυκλοφορίας για το ασθενοφορο

start\_time=millis(); // Βάλε στην start\_time την τιμή της millis()

while (millis() - start\_time < 100000) { // Εφόσον δεν πέρασαν 100 δευτ κάνε:

digitalWrite(ledRed, LOW);

digitalWrite(ledOrange, LOW);

digitalWrite(ledGreen, HIGH);

digitalWrite(ledRedB, HIGH);

digitalWrite(ledOrangeB, LOW);

digitalWrite(ledGreenB, LOW);

digitalWrite(pedRed, HIGH);

digitalWrite(pedGreen, LOW);

digitalWrite(pedRedB, LOW);

digitalWrite(pedGreenB, HIGH);

}}

if ( TP\_init()<= 5 || TP\_init() >=29930 ) {

// πορτοκαλί για 1 δευτερόλεπτο

start\_time=millis();

while (millis() - start\_time < 1000) { // Εφόσον δεν πέρασαν 1 δευτ κάνε:

digitalWrite(pedRed, HIGH);

digitalWrite(pedGreen, LOW);

digitalWrite(ledRed, LOW);

digitalWrite(ledOrange, HIGH);

digitalWrite(ledGreen, LOW);

digitalWrite(pedRedB, HIGH);

digitalWrite(pedGreenB, LOW);

digitalWrite(ledRedB, HIGH);

digitalWrite(ledOrangeB, LOW);

digitalWrite(ledGreenB, LOW);}}

else {

// πράσινο κυκλοφορίας για το ασθενοφορο

start\_time=millis(); // Βάλε στην start\_time την τιμή της millis()

while (millis() - start\_time < 100000) { // Εφόσον δεν πέρασαν 100 δευτ κάνε:

digitalWrite(ledRed, LOW);

digitalWrite(ledOrange, LOW);

digitalWrite(ledGreen, HIGH);

digitalWrite(ledRedB, HIGH);

digitalWrite(ledOrangeB, LOW);

digitalWrite(ledGreenB, LOW);

digitalWrite(pedRed, HIGH);

digitalWrite(pedGreen, LOW);

digitalWrite(pedRedB, LOW);

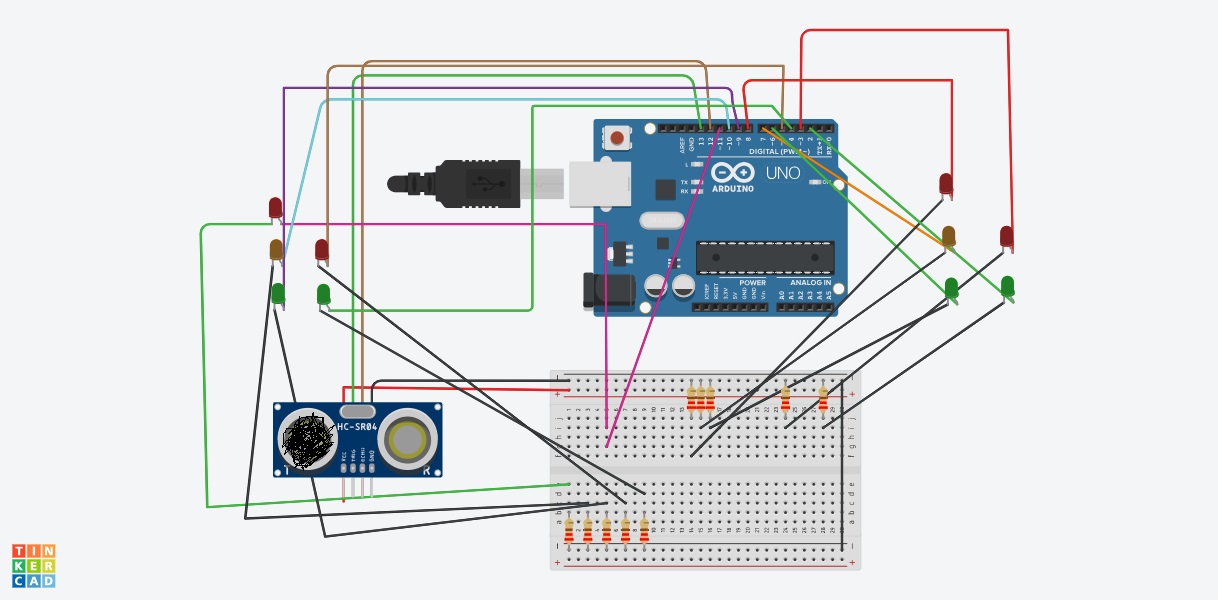
digitalWrite(pedGreenB, HIGH);

}

}

}

A---ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ



Β)

//ΤΟ ΠΡΟΓΡΑΜΜΑ ΤΗΣ ΜΠΑΡΑΣ ΤΟΥ ΝΟΣΟΚΟΜΕΙΟΥ , ΤΗΣ ΟΘΟΝΗΣ LCD ΚΑΙ ΤΗΣ ΣΤΑΔΙΑΚΗΣ ΦΩΤΕΙΝΟΤΗΤΑΣ ΤΗΣ ΠΟΛΗΣ(ΟΣΟ ΠΕΡΙΣΣΟΤΕΡΟ ΣΚΟΤΑΔΙ ΤΟΣΟ ΠΙΟ ΠΟΛΥ ΦΩΤΙΖΟΥΝ) KAI!.....

// I2C LCD screen demo

//Compatible with the Arduino IDE 1.0

//Library version:1.1

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,16 ,2);

#define echo 12

#define trigger 13

#include <Servo.h> // Συμπερίλαβε τη βιβλιοθήκη του σέρβο

Servo myservo; // Δημιούργησε ένα αντικείμενο τύπου Servo

int s=1; // Το βήμα (μοίρες) που θα αυξάνουμε τη θέση του σέρβο σε κάθε επανάληψη

//const int LDR=A0; //ΕΔΩ ΣΥΝΔΕΩ ΤΗΝ ΦΩΤΟΑΝΤΙΣΤΑΣΗ LDR ΓΙΑ ΝΑ ΔΙΑΒΑΖΩ ΤΟ ΕΠΙΠΕΔΟ ΦΩΤΕΙΝΟΤΗΤΑΣ

int LDR;

int input=0;

int brightness=0;

float hum; //Stores humidity value,ΤΙΜΗ ΥΓΡΑΣΙΑΣ

float temp; //Stores temperature value,ΤΙΜΗ ΘΕΡΜΟΚΡΑΣΙΑΣ

const int CityLIGHTSA=5;

const int CityLIGHTSB=3;

unsigned long start\_time; // Δημιούργησε μια μεταβλητή τύπου unsigned long integer με όνομα start\_time

#include <DHT.h>; //ΒΙΒΛΙΟΘΗΚΗ ΑΙΣΘΗΤΗΡΑ DHT

//Constants

#define DHTPIN 4 // what pin we're connected to ,ΤΟ ΠΙΝ ΠΟΥ ΣΥΝΔΕΟΥΜΕ ΤΟΝ ΑΙΣΘΗΤΗΡΑ ΘΕΡΜΟΚΡΑΣΙΑΣ - ΥΓΡΑΣΙΑΣ

#define DHTTYPE DHT22 // DHT 22 (AM2302),ΤΥΠΟΣ ΑΙΣΘΗΤΗΡΑ,ΔΙΑΛΕΞΑΜΕ DHT22 ΩΣ ΠΙΟ ΑΞΙΟΠΙΣΤΟ

DHT dht(DHTPIN, DHTTYPE); //// Initialize DHT sensor for normal 16mhz Arduino

//Variables

int red1=2;

int blue1=6;

//ΕΔΩ ΟΙ ΟΡΙΣΜΟΙ ΓΙΑ ΦΩΝΗΤΙΚΗ ΠΡΟΕΙΔΟΠΟΙΗΣΗ -ΠΑΙΖΕΙ ΑΠΟ DFPLAYER MINI MP3 ΚΑΙ SD CARD

#include "SoftwareSerial.h"

SoftwareSerial mySerial(10, 11);

# define Start\_Byte 0x7E

# define Version\_Byte 0xFF

# define Command\_Length 0x06

# define End\_Byte 0xEF

# define Acknowledge 0x00 //Returns info with command 0x41 [0x01: info, 0x00: no info]

# define ACTIVATED LOW

int buttonPause = 3;

long TP\_init() //ΤΟ CONTROL-ΔΕΚΤΗΣ ΠΟΥ ΘΑ ΕΛΕΓΧΕΙ ΤΗΝ ΠΑΡΑΛΑΒΗ Ή ΟΧΙ ΤΟΥ ΣΗΜΑΤΟΣ-ΠΟΙΟ ΚΑΤΩ ΦΑΙΝΕΤΑΙ ΟΤΑΝ ΧΡΗΣΙΜΟΠΟΙΗΤΑΙ

{

digitalWrite(trigger,LOW);

delayMicroseconds(2);

digitalWrite(trigger,HIGH);

delayMicroseconds(2);

digitalWrite(trigger,LOW);

delayMicroseconds(2);

long microseconds=pulseIn(echo,HIGH,100000);

return microseconds ;

}

void setup()

{

pinMode(buttonPause, INPUT);

digitalWrite(buttonPause,HIGH);

mySerial.begin (9600);

delay(1000);

Serial.begin(9600);

pinMode(CityLIGHTSA,OUTPUT); //LIGHTSA AND B ΣΤΑΔΙΑΚΟΣ ΦΩΤΙΣΜΟΣ ΠΟΛΗΣ

pinMode(CityLIGHTSB,OUTPUT);

pinMode(red1, OUTPUT);

pinMode(blue1, OUTPUT);

pinMode(trigger, OUTPUT);

pinMode(echo, INPUT);

lcd.init(); // initialize the lcd

// Print a message to the LCD.

lcd.backlight();

lcd.setCursor(0,0);

myservo.attach(9); // Σχετίζει το σερβο με το πιν 9

LDR=analogRead(A0);

myservo.write(0); // Γύρισε τον άξονα του servo στην τιμή 0

delay(35); // Περίμενε λίγο για να γυρίσει το servo στην τιμή που θέλουμε

dht.begin(); //ΘΕΡΜΟΜΕΤΡΟ DHT,ΞΕΚΙΝΑ.

digitalWrite(CityLIGHTSA,LOW);

digitalWrite(CityLIGHTSB,LOW);

}

void loop()

{

//Read data and store it to variables hum and temp,ΔΙΑΒΑΖΩ ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΤΙΣ ΑΠΟΘΗΚΕΥΩ ΣΤΙΣ ΜΕΤΑΒΛΗΤΕΣ hum and temp

hum = dht.readHumidity();

temp= dht.readTemperature();

lcd.setCursor(0,0); // Η ΟΘΟΝΗ LCD ΘΑ ΔΕΙΞΕΙ ΘΕΡΜΟΚΡΑΣΙΑ ΚΑΙ ΥΓΡΑΣΙΑ % ΣΤΗΝ ΠΡΩΤΗ ΓΡΑΜΜΗ

lcd.print( temp );

lcd.print( "C ");

lcd.setCursor(7,0);

lcd.print( hum );

lcd.print(" % ");

lcd.setCursor(0,1);

lcd.print("GYMNASIO BRYSWN ");

myservo.write(0); // Γύρισε τον άξονα του servo στην τιμή 0

delay(1500); // Περίμενε λίγο για να γυρίσει το servo στην τιμή που θέλουμε

//EAN ANIXNEYTEI YPERHXOS

if ( TP\_init()>=10) {

start\_time=millis(); //ΑΡΧΙΖΩ ΤΟ ΧΡΟΝΟΜΕΤΡΟ

while (millis() - start\_time < 40000) { // Εφόσον δεν πέρασαν 40 δευτ κάνε:

myservo.write(110); // Γύρισε τον άξονα του servo στην τιμή 110-άνοιξε την μπάρα

delay(20);

//Read data and store it to variables hum and temp,ΔΙΑΒΑΖΩ ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΤΙΣ ΑΠΟΘΗΚΕΥΩ ΣΤΙΣ ΜΕΤΑΒΛΗΤΕΣ hum and temp

hum = dht.readHumidity();

temp= dht.readTemperature();

input = analogRead(LDR);

brightness = map(input,520,1023,0,255); //ΣΤΑΔΙΑΚΟΣ ΦΩΤΙΣΜΟΣ ΠΟΛΗΣ ΑΝΤΙΣΤΡΟΦΩΣ ΑΝΑΛΟΓΗ ΤΗΣ ΕΞΩΤΕΡΙΚΗΣ ΦΩΤΕΙΝΟΤΗΤΑΣ

analogWrite(CityLIGHTSA,brightness);

analogWrite(CityLIGHTSB,brightness);

Serial.println(TP\_init());

lcd.setCursor(0,0);

lcd.print("AMBULANCE DETECT");

lcd.setCursor(0,1);

lcd.print(" DOORS OPEN!!! ");

//ΕΑΝ Η ΘΕΡΜΟΚΡΑΣΙΑ ΕΙΝΑΙ <14 C ΓΡΑΨΕ " COLD ,PROTECT!!!" ΣΤΗΝ ΟΘΟΝΗ LCD ΚΑΙ ΠΑΙΞΕ ΤΟ ΑΝΤΙΣΤΟΙΧΟ ΚΟΜΜΑΤΙ ΑΠΟ DFPLAYER-SD CARD

if (temp<14.00)

{

lcd.setCursor(0,1);

lcd.print(" COLD ,PROTECT!!!");

playSecond();

digitalWrite(blue1,HIGH);

delay(50);

playFifth();

}

//ΕΑΝ Η ΘΕΡΜΟΚΡΑΣΙΑ ΕΙΝΑΙ >28 C ΓΡΑΨΕ "HOT CLIMATE,PROTECT" ΣΤΗΝ ΟΘΟΝΗ ΚΑΙ ΠΑΙΞΕ ΤΟ ΑΝΤΙΣΤΟΙΧΟ ΚΟΜΜΑΤΙ ΑΠΟ DFPLAYER-SD CARD

if (temp>28.00)

{

lcd.setCursor(0,1);

lcd.print("HOT CLIMATE,PROTECT");

playFourth();

digitalWrite(red1,HIGH);

delay(50);

playThird();

}

}}

//ΕΑΝ ΔΕΝ ΑΝΙΧΝΕΥΤΕΙ ΥΠΕΡΗΧΟΣ

if ( TP\_init()<=10) {

start\_time=millis(); //ΑΡΧΙΖΩ ΤΟ ΧΡΟΝΟΜΕΤΡΟ

while (millis() - start\_time < 2500) { // Εφόσον δεν πέρασαν 2,5 δευτ κάνε:

myservo.write(0); // Γύρισε τον άξονα του servo στην τιμή 0-κλεισε την μπάρα

delay(20);

//Read data and store it to variables hum and temp,ΔΙΑΒΑΖΩ ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΤΙΣ ΑΠΟΘΗΚΕΥΩ ΣΤΙΣ ΜΕΤΑΒΛΗΤΕΣ hum and temp

hum = dht.readHumidity();

temp= dht.readTemperature();

input = analogRead(LDR);

brightness = map(input,400,1023,0,255); //ΣΤΑΔΙΑΚΟΣ ΦΩΤΙΣΜΟΣ ΠΟΛΗΣ ΑΝΤΙΣΤΡΟΦΩΣ ΑΝΑΛΟΓΗ ΤΗΣ ΕΞΩΤΕΡΙΚΗΣ ΦΩΤΕΙΝΟΤΗΤΑΣ

analogWrite(CityLIGHTSA,brightness);

analogWrite(CityLIGHTSB,brightness);

Serial.println(TP\_init());

lcd.setCursor(0,0);

lcd.print(" NO AMBULANCE! ");

lcd.setCursor(0,1);

lcd.print(" DOORS CLOSED ");

if (temp<14.00)

{

lcd.setCursor(0,1);

lcd.print(" COLD ,PROTECT!!!");

playSecond();

digitalWrite(blue1,HIGH);

delay(50);

playFifth();

}

if (temp>28.00)

{

lcd.setCursor(0,1);

lcd.print("HOT CLIMATE,PROTECT");

playFourth();

digitalWrite(red1,HIGH);

delay(50);

playThird();

}

}

}

// τα παρακατω ειναι για τον έλεγχο παραλαβης υπερηχου,ldr,φωτεινοτητας

Serial.println(TP\_init());

Serial.println(input);

Serial.println(brightness);

Serial.print(temp); Serial.print(" C ");

}

//ΓΙΑ ΤΑ MP3

void playFirst()

{

execute\_CMD(0x3F, 0, 0);

delay(2500);

setVolume(30);

delay(500);

}

void playFifth() //κατευθυνθητε σε θερμρνομενους χωρους

{

execute\_CMD(0x03, 0, 4);

delay(5500);

setVolume(30);

delay(50);

}

void playSecond() //ΠΡΟΣΟΧΗ ΧΑΜΗΛΗ ΘΕΡΜΟΚΡΑΣΙΑ

{

execute\_CMD(0x03, 0, 1);

delay(4000);

setVolume(30);

delay(50);

}

void playThird() //ΚΑΤΕΥΘΥΝΘΗΤΕ ΣΕ ΚΛΙΜΑΤΙΖΟΜΕΝΟΥΣ ΧΩΡΟΥΣ

{

execute\_CMD(0x03, 0, 2);

delay(5500);

setVolume(30);

delay(50);

}

void playFourth() //ΠΡΟΣΟΧΗ ΥΨΗΛΗ ΘΕΡΜΟΚΡΑΣΙΑ

{

execute\_CMD(0x03, 0, 3);

delay(4000);

setVolume(30);

delay(50);

}

void pause()

{

execute\_CMD(0x0E,0,0);

delay(500);

}

void play()

{

execute\_CMD(0x0D,0,1);

delay(500);

}

void setVolume(int volume)

{

execute\_CMD(0x06, 0, volume); // Set the volume (0x00~0x30)

delay(1000);

}

void execute\_CMD(byte CMD, byte Par1, byte Par2)

// Excecute the command and parameters

{

// Calculate the checksum (2 bytes)

word checksum = -(Version\_Byte + Command\_Length + CMD + Acknowledge + Par1 + Par2);

// Build the command line

byte Command\_line[10] = { Start\_Byte, Version\_Byte, Command\_Length, CMD, Acknowledge,

Par1, Par2, highByte(checksum), lowByte(checksum), End\_Byte};

//Send the command line to the module

for (byte k=0; k<10; k++)

{

mySerial.write( Command\_line[k]);

}

}

Γ)

Ο πομπός-το πρόγραμμα:

#define TRIGGERPIN 8

#define ECHOPIN 9

void setup() {

pinMode(TRIGGERPIN,OUTPUT);

pinMode(ECHOPIN,INPUT);

}

void loop() {

digitalWrite(TRIGGERPIN,LOW);

delayMicroseconds(1);

digitalWrite(TRIGGERPIN,HIGH);

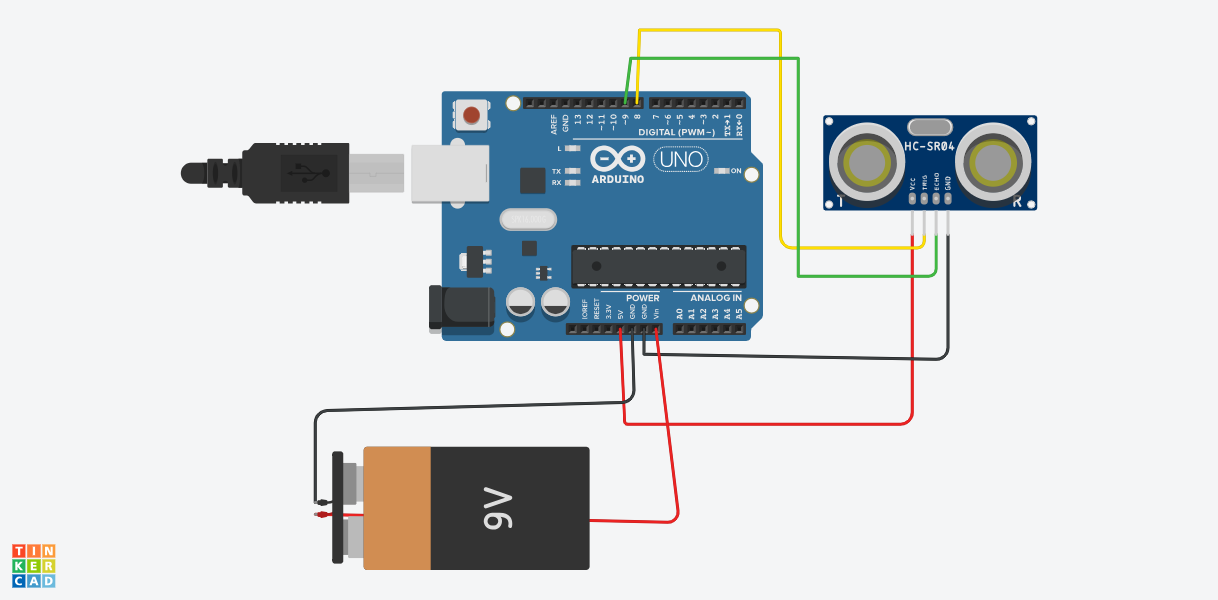
delayMicroseconds(10);

digitalWrite(TRIGGERPIN,LOW);

delayMicroseconds(1);

}

ΤΟ ΑΝΤΙΣΤΟΙΧΟ ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΟΥ ΠΟΜΠΟΥ:



ΤΑ ΟΝΟΜΑΤΑ ΤΩΝ ΜΑΘΗΤΩΝ ΤΗΣ ΟΜΑΔΑΣ:

|  |  |
| --- | --- |
| Α ΑΝΔΡΕΑΔΑΚΗ | ΒΑΣΙΛΙΚΗ |
| ΜΑΡΙΔΑΚΗ | ΖΑΜΠΙΑ |
| ΤΣΑΠΑΚΗ | ΙΖΑΜΠΕΛΛΑ |
| Κ ΚΟΤΣΙΦΑΚΗ | ΑΝΝΑ |
| ΜΠΟΥΝΤΟΥΡΑΚΗ | ΑΘΗΝΑ |
| ΠΑΙΔΑΡΑΚΗ | ΧΡΥΣΑΝΘΗ |
| ΠΑΠΑΔΑΚΗΣ | ΜΙΧΑΗΛ-ΑΝΤΩΝΙΟΣ |
| ΣΥΜΒΟΥΛΑΚΗ | ΖΩΗ |
| ΤΣΙΛΕΔΑΚΗ | ΑΓΓΕΛΙΚΗ |