

Πανεπιστήμιο Θεσσαλίας

Τμήμα Ψηφιακών Συστημάτων



Προπτυχιακή εργασία στα ενσωματωμένα συστήματα
με θέμα:

Χειρισμός Video Player εφαρμογής μέσω του Arduino
Uno και δυο αισθητήρων ultrasound.

Φράνκο Καρόλι Α.Μ 3119178

Καθηγητής: Αδάμ Γεώργιος

Περίληψη

Στην εργασία αυτή θα δούμε τον χειρισμό μιας εφαρμογής αναπαραγωγής βίντεο με την βοήθεια 2 αισθητήρων κίνησης ultrasound που αναλόγως με την τοποθεσία των χεριών, θα ενεργοποιούν και μια συγκεκριμένη λειτουργία. Οι αισθητήρες θα είναι τοποθετημένοι πάνω στην οθόνη του laptop και θα είναι συνδεδεμένοι στο Arduino Uno, όπου και θα ανιχνεύουν τις κινήσεις. Η κάθε κίνηση θα στέλνει ένα συγκεκριμένο μήνυμα, το οποίο θα λαμβάνει η Python και θα υλοποιεί τις λειτουργίες που έχουμε προγραμματίσει.

Περιεχόμενα

Περίληψη	2
Εισαγωγή	4
Hardware που θα χρησιμοποιούμε	5
Σχεδίαση της πλακέτας	6
Software που θα χρησιμοποιήσουμε	7
Λειτουργίες του προγράμματος.....	7
Ο κώδικας για το Arduino	8
Ο Κώδικας για την Python.....	11
Φωτογραφίες από το Project.....	12
Βιβλιογραφία	14

Εισαγωγή

Ένα ενσωματωμένο σύστημα είναι ένα υπολογιστικό σύστημα που εμπεριέχει κυκλώματα επεξεργασίας και περιφερειακών, με μειωμένο μέγεθος, κατανάλωση και κόστος, και με εφαρμογές σε συστήματα συνήθως πραγματικού χρόνου. Τα ενσωματωμένα συστήματα είναι εξειδικευμένα συστήματα υπολογιστών, τα οποία είναι αφοσιωμένα στην εκτέλεση μιας συγκεκριμένης λειτουργίας. Στη συνέχεια θα δούμε πως στα ενσωματωμένα συστήματα συνδυάζονται το υλικό και το λογισμικό.

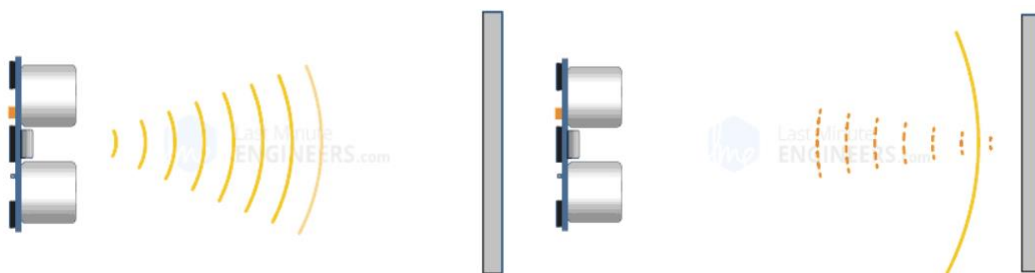
Hardware που θα χρησιμοποιούμε

Για την υλοποίηση του project μας θα χρησιμοποιήσουμε έναν μικροελεγκτή Arduino Uno και δύο αισθητήρες HC-SR04 Ultrasonic Module Distance.

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++).

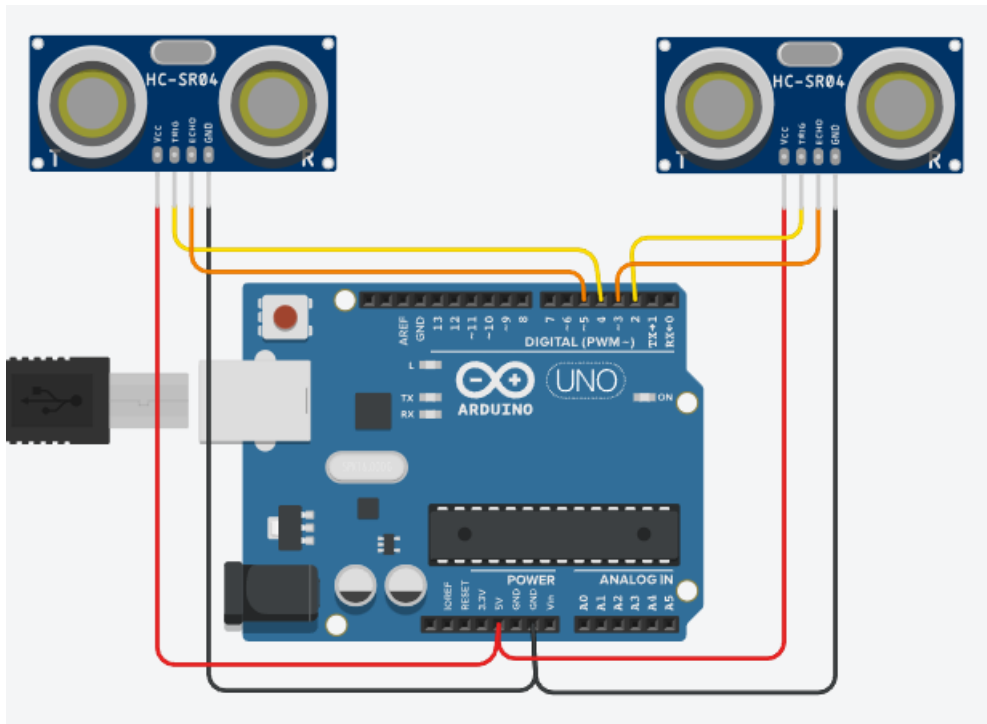


Ο αισθητήρας HC-SR04 Ultrasonic Module Distance είναι ένας μικροελεγκτής που στέλνει 8 ultrasound παλμούς σε συχνότητα 40 KHz. Αν ο παλμός αυτός δεν επιστρέψει σημαίνει πως δεν έχει συναντήσει κάποιο εμπόδιο. Διαφορετικά, μέσω συνάρτησης υπολογίζεται ο χρόνος που πέρασε για την αντανάκλαση του παλμού πίσω στον αισθητήρα και έτσι έχουμε πληροφορίες για το πόσο μακριά είναι το εμπόδιο.

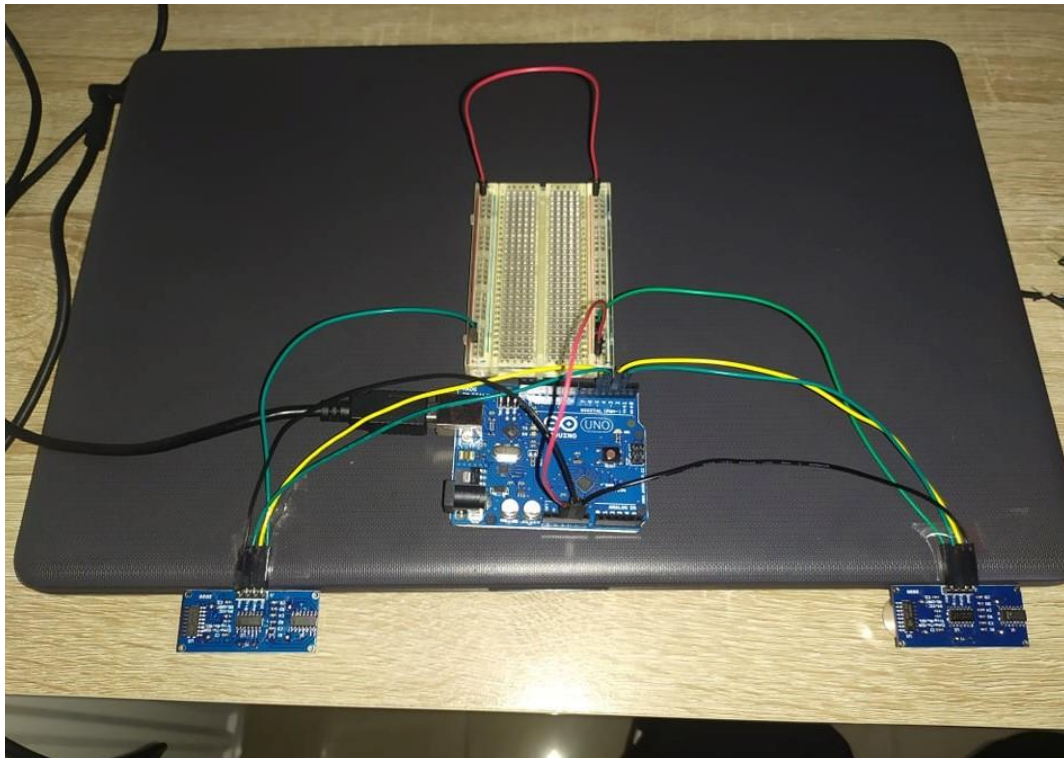


Σχεδίαση της πλακέτας

- Σε προσομοιωτή (Tinkercad)



- Τοποθετημένη πάνω στο Laptop



Software που θα χρησιμοποιήσουμε



Πέρα από το υλικό κομμάτι του project, χρειαζόμαστε και το software. Το κομμάτι αυτό θα το υλοποιήσουμε με Python. Το πρώτο πράγμα λοιπόν είναι να έχουμε το python IDLE. Αμέσως μετά θα πρέπει να κατεβάσουμε την PySerial library την οποία χρησιμοποιούμε για να διαβάσουμε και να γράψουμε δεδομένα στο Arduino η σε οποιοδήποτε άλλο μικροελεγκτή. Τέλος , θα χρειαστούμε την PyAutogui βιβλιοθήκη η οποία μας επιτρέπει να χειριστούμε τις λειτουργίες του πληκτρολόγιου και του ποντικιού με ευκολία.

Λειτουργίες του προγράμματος

1. Pause /Play

Όταν τοποθετούμε και τα δύο χέρια κοντά στους αισθητήρες τότε το πρόγραμμα θα εκτυπώνει το μήνυμα 'Pause/Play' το οποίο λαμβάνει η python και κάνει το Pause ή Play ανάλογα.(Βλέπε σελ.12)

2. Volume +/-

Όταν τοποθετούμε το ένα χέρι στον αριστερό αισθητήρα τότε μπαίνει σε Lock Mode όπου έπειτα αν μετακινήσουμε το χέρι προς τον αισθητήρα η ένταση αυξάνεται ενώ αν το μετακινήσουμε προς τα πίσω η ένταση θα μειωθεί.(Βλέπε σελ.12)

3. Rewind/Fast Forward

Όταν τοποθετούμε το δεξί χέρι στον αισθητήρα αντίστοιχα, τότε μπαίνει σε Lock Mode όπου έπειτα αν μετακινήσουμε το χέρι προς τον αισθητήρα το βίντεο προχωρά(Fast Forward) ενώ αν το μετακινήσουμε προς τα πίσω το βίντεο πάει γρήγορα προς τα πίσω(Rewind).(Βλέπε σελ.13)

Ο κώδικας για το Arduino

```
const int trigger1 = 2; //To Trigger pin του 1ου αισθητήρα
const int echo1 = 3; //To Echo pin του 1ου αισθητήρα
const int trigger2 = 4; //To Trigger pin του 2ου αισθητήρα
const int echo2 = 5; //To Echo pin του 2ου αισθητήρα

long time_taken;

int dist,distL,distR;

void setup() {
    Serial.begin(9600);
    pinMode(trigger1, OUTPUT);
    pinMode(echo1, INPUT);
    pinMode(trigger2, OUTPUT);
    pinMode(echo2, INPUT);
}

/*###Συνάρτηση με την οποία υπολογίζουμε την απόσταση###*/
void calculate_distance(int trigger, int echo)
{
    digitalWrite(trigger, LOW);
    delayMicroseconds(2);
    digitalWrite(trigger, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigger, LOW);
    time_taken = pulseIn(echo, HIGH);

    /*Ταχύτητα του ήχου 340 m/s οπότε την διαιρούμε κατά 2 γιατί μας ενδιαφέρει
πόσο μακριά έφτασε και όχι πόσο έκανε για να επιστρέψει ο παλμός*/

    dist= time_taken*0.034/2;

    if (dist>60)
        dist = 60;
```



```

}

void loop() { //Η λούπα που επαναλαμβάνεται συνεχώς

  calculate_distance(trigger1,echo1);

  distL =dist; //παίρνω την απόσταση από τον αριστερό αισθητήρα

  calculate_distance(trigger2,echo2);

  distR =dist; // παίρνω την απόσταση από τον δεξι αισθητήρα

  //To Pause και το Play

  if ((distL >40 && distR>40) && (distL <60 && distR<60)) //Αν εντοπίζει και τα δύο χέρια

  {Serial.println("Play/Pause"); delay (500);}

  calculate_distance(trigger1,echo1);

  distL =dist;

  calculate_distance(trigger2,echo2);

  distR =dist;

//Control Modes

/*Lock Mode για το αριστερό χέρι ώστε να κλειδώσει την απόσταση και αναλόγως με την
κατεύθυνση του χεριού θα αλλάζει και η ένταση*/

  if (distL>=13 && distL<=25)

  {

    delay(100); //Χρόνος για να κλειδώσει

    calculate_distance(trigger1,echo1);

    distL =dist;

    if (distL>=13 && distL<=25)

    {

      Serial.println("Left Locked");

      while(distL<=40)

      {

        calculate_distance(trigger1,echo1);

        distL =dist;

        if (distL<20) //Όταν το χέρι κατευθύνεται προς τον αριστερό αισθητήρα

        {Serial.println ("Volume Increased"); delay (300);}

        if (distL>20) // Όταν το χέρι απομακρύνεται από τον αριστερό αισθητήρα

```

```

    {Serial.println ("Volume Decreased"); delay (300);}
}
}
}

/*Lock Mode για το δεξί χέρι ώστε να κλειδώσει την απόσταση και αναλόγως με την
κατεύθυνση του χεριού θα αλλάζει σε Rewind ή Forward*/
if (distR>=13 && distR<=25)
{
    delay(100); // Χρόνος για να κλειδώσει
    calculate_distance(trigger2,echo2);
    distR =dist;
    if (distR>=13 && distR<=25)
    {
        Serial.println("Right Locked");
        while(distR<=40)
        {
            calculate_distance(trigger2,echo2);
            distR =dist;
            if (distR<20) //Όταν το χέρι πλησιάζει τον δεξί αισθητήρα
            {Serial.println ("Forward"); delay (300);}
            if (distR>20) //Όταν το χέρι απομακρύνεται από τον δεξί αισθητήρα
            {Serial.println ("Rewind"); delay (300);}
        }
    }
}
}
}
delay(200);
}

```

Ο Κώδικας για την Python

```
import serial # Η βιβλιοθήκη Serial για την σειριακή επικοινωνία
import time #Χρειάζεται για να υλοποιήσουμε για την αναμονή (Delays)
import pyautogui #Χρειάζεται για τον χειρισμό των εφαρμογών

ArduinoSerial = serial.Serial('com4',9600) #Δημιουργία port με όνομα arduinoSerialData
time.sleep(2) #Περίμενε 2 δευτερόλεπτα ώστε να γίνει η επικοινωνία

while 1:

    incoming = str (ArduinoSerial.readline()) #Διάβασε τα δεδομένα και εκτύπωσέ τα
    print (incoming)

    if 'Play/Pause' in incoming:
        pyautogui.typewrite(['space'], 0.2)
    if 'Rewind' in incoming:
        pyautogui.hotkey('ctrl', 'left')
    if 'Forward' in incoming:
        pyautogui.hotkey('ctrl', 'right')
    if 'Volume Increased' in incoming:
        pyautogui.hotkey('ctrl', 'up')
    if 'Volume Decreased' in incoming:
        pyautogui.hotkey('ctrl', 'down')

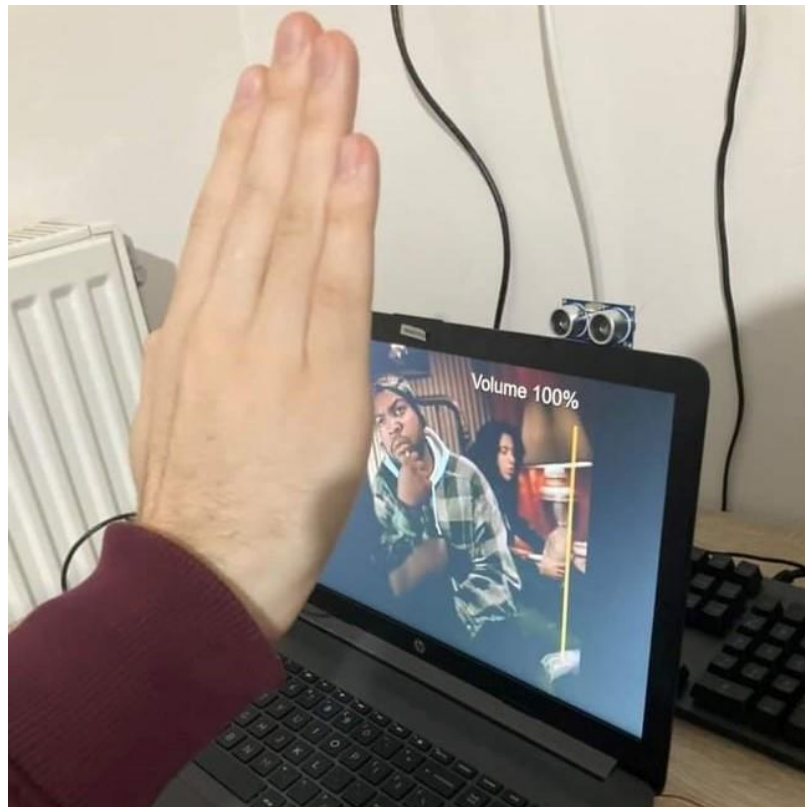
    incoming = "";
```

Φωτογραφίες από το Project

- Pause/Play



- Volume +/-



- Rewind/Fast Forward



Βιβλιογραφία

- https://el.wikipedia.org/wiki/%CE%95%CE%BD%CF%83%CF%89%CE%BC%CE%B1%CF%84%CF%89%CE%BC%CE%AD%CE%BD%CE%BF_%CF%83%CF%8D%CF%83%CF%84%CE%B7%CE%BC%CE%B1
- <https://el.wikipedia.org/wiki/Arduino>
- <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/>
- https://www.youtube.com/watch?v=6R5MRM7trOM&ab_channel=SmartTech
- <https://www.tinkercad.com/>