

CONTROLAREA UNUI COMPUTER PRIN GESTURI

Șerban Alexandru-Gabriel

Facultatea: Ingineria și Managementul Sistemelor Tehnologice, Specializarea: Inginerie industrială în limba engleză
(industrial engineering), Anul de studii: I, e-mail: serbanalexandru02@yahoo.com

Conducător științific: Prof. univ. dr. ing. **Cătălin Gheorghe AMZA**

REZUMAT: Acest proiect utilizează tehnica de mișcare în salt, definită ca fiind utilizarea unui număr de senzori pentru a cartografia un spațiu de lucru sau cu alte cuvinte – crearea unui spațiu de lucru tridimensional. Conceptul care sta la baza proiectului este foarte simplu. Doi senzori ultrasonici (SU) sunt amplasați deasupra monitorului, iar aceștia o să citească distanța dintre mana noastră și monitor. Prin programarea unei plăcuțe de dezvoltare cu microprocesor Arduino, în funcție de valoarea distanței citite prin program, se vor efectua anumite acțiuni. Pentru efectuarea acțiunilor pe computer se folosește biblioteca Python pyautogui. Comenzile de la Arduino sunt trimise prin portul serial (USB). Aceste date o să fie citite ulterior de către python care rulează pe computer, iar în baza datelor citite o acțiune specifică va fi efectuată. Programul python este utilizat împreună cu programul arduino deoarece python preia informația brută de la placa de dezvoltare și senzori și o traduce în comenzi simple de tastatură.

CUVINTE CHEIE: arduino, python, senzori

1. Introducere

Prin folosirea unor materiale ieftine și a cunoștințelor de programare, cât și a înțelegerii interacțiunilor dintre materiale diferite și programe diverse avem capacitatea de a modifica și îmbunătăți un obiect deja existent precum un computer spre a îi mări posibilitățile de utilizare și a face ceva complet nou precum un artist ce desenează pe o pânză. Înaintea explicării pașilor producție este necesară definirea următorilor termeni și anume:

- placa de dezvoltare cu microprocesor (arduino) = o placă de circuit imprimat ce conține un procesor, cu puțin sau chiar fără hardware dedicat unei interfațe de utilizator având posibilitatea de a accepta și rula un program furnizat de utilizator;
- senzor ultrasonic = un senzor ultrasonic este un dispozitiv care poate măsura distanța către un obiect utilizând unde ultrasonore. Acesta măsoară locația prin trimiterea unei unde ultrasonore la o frecvență specifică și ascultând întoarcerea respectivei unde de sunet. Prin înregistrarea timpului trecut dintre unda de sunet generată și unda de sunet ce se întoarce este posibilă calcularea distanței dintre senzorul sonar și obiect;
- fire de conexiune = un cablu capabil să transporte energie sau semnale dintr-un loc în altul;
- software Arduino = software-ul Arduino este o platformă de tip ‘open-source’ specializată ce se bazează pe software și hardware ușor de utilizat. Plăcuțele Arduino sunt capabile de a citi semnale de intrare precum lumină sau sunet (de la diversi senzori) și de a le transforma în semnale de ieșire prin utilizarea limbajului de programare Arduino;
- Python software = Python este un limbaj de programare de nivel înalt cu scop general. Suportă paradigme de programare multiple incluzând de asemenea și o bibliotecă standard.

Materialele utilizate au fost:

1. Placa de dezvoltare arduino



Figura 1.Placa de dezvoltare arduino Leonard

2.Senzorii ultrasonici



Figura 1.1.Senzori ultrasonici

3.Fire de conexiune de tip mamă-tată



Figura 1.2.Fire de conexiune de tip mamă-tată

Pentru această lucrare prin program se codeaza cinci acțiuni de interactivitate cu un calculator și anume:

- acțiunea 1: Atunci când ambele mâini se află în fața sensorului la o anumită distanță un videoclip din VLC player o să se oprească/pornească
- acțiunea 2:Atunci când mâna dreaptă se află în fața sensorului la o distanță anume videoclipul o să meargă pe repede-înainte un pas (10 secunde)
- acțiunea 3: Când mâna stângă se află în fața sensorului la o distanță anume videoclipul o să se deruleze un pas (10 secunde)

- acțiunea 4: Când mâna dreaptă se află în fața senzorului la o anumită distanță, iar după se mișcă în fața senzorului videoclipul o să se pună pe repede înainte, iar dacă mișcă înapoi videoclipul se va derula.
- acțiunea 5: Când mâna stângă este pusă în fața senzorului la o distanță anume, iar apoi dusă înspre senzor volumul videoclipului o să crească și dacă este mișcată înapoi volumul o să scadă.

Codul este scris să facă aceste acțiuni în programul Arduino după cum urmează:

- definim pini I/O, cei doi senzori ultrasonici sunt conectați la pini digitali 2,3,4 și 5 și sunt alimentați de un curent egal cu cinci volți al pinului de alimentare. Pini Trigger sunt pini de ieșire, iar pini Echo sunt pini de intrare;
- comunicarea dintre Arduino și Python se face la o rată baud (unitatea de măsură a vitezei de comunicare într-un canal de date) de 9600 (1/9600 de secunde).
- programul în Python necesar este foarte simplu. Trebuie creată o comunicare în serie cu Arduino la rata baud corectă și apoi folosite funcțiile if și while pentru a produce acțiunile.

Pași sunt următorii: se deschide Windows command prompt și se schimbă directiva folderului python. Comanda implicită este: cd C:\Python27. În interiorul ferestrei de comanda python se folosește comanda >python -m pip install --upgrade pip pentru a îmbunătății Pip. Pip este o unealtă folosită de python pentru a instala mai ușor module.

Se folosește comanda 'python -m pip install pyautogui' pentru a instala modulul specific folosit.

```

C:\Python27>python -m pip --version
pip 1.5.6 from C:\Python27\lib\site-packages (python 2.7)

C:\Python27>python -m pip install --upgrade pip
Downloading/unpacking pip from https://pypi.python.org/packages/b6/ac/7015eb97dc749283ffdec1c3a88ddb8ae03b8fad0f0e611408f196358da3/pip-9.0.1-py2.py3-none-any.whl#md5=297dbd16ef53bcef0447d245815f5144
Installing collected packages: pip
  Found existing installation: pip 1.5.6
    Uninstalling pip:
      Successfully uninstalled pip
Successfully installed pip
Cleaning up...

C:\Python27>python -m pip install pyautogui
Collecting pyautogui
  Downloading PyAutoGUI-0.9.36.tar.gz (46kB)
    100% |#####| 51kB 156kB/s
Collecting pymsgbox (from pyautogui)
  Downloading PyMsgBox-1.0.6.zip
Collecting PyTweening>=1.0.1 (from pyautogui)
  Downloading PyTweening-1.0.3.zip
Collecting Pillow (from pyautogui)
  Downloading Pillow-4.3.0-cp27-cp27m-win32.whl (1.3MB)
    100% |#####| 1.3MB 177kB/s
Collecting pycreeze (from pyautogui)
  Downloading PyScreeze-0.1.13.tar.gz
Collecting olefile (from Pillow->pyautogui)
  Downloading olefile-0.44.zip (74kB)
    100% |#####| 81kB 227kB/s
Installing collected packages: pymsgbox, PyTweening, olefile, Pillow, pycreeze,
pyautogui
  Running setup.py install for pymsgbox ... done
  Running setup.py install for PyTweening ... done
  Running setup.py install for olefile ... done
  Running setup.py install for pycreeze ... done
  Running setup.py install for pyautogui ... done
Successfully installed Pillow-4.3.0 PyTweening-1.0.3 olefile-0.44 pyautogui-0.9.
36 pymsgbox-1.0.6 pycreeze-0.1.13

C:\Python27>

```

Figura 1.3. Instalarea cu succes a modulului pyautogui

Arduino și Python comunică prin portul serial 3 (COM3), acesta poate fi schimbat în funcție de preferința utilizatorului.

```

python code for arduino leonard.py - C:\Python27\python code for arduino leonard.py (2.7.9)
File Edit Format Run Options Windows Help
import serial #Serial imported for Serial communication
import time #Required to use delay functions
import pyautogui

ArduinoSerial = serial.Serial('com3',9600) #Create Serial port object called arduinoSerialData
time.sleep(2) #wait for 2 seconds for the communication to get established

while 1:
    incoming = str (ArduinoSerial.readline()) #read the serial data and print it as line
    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 'Vup' in incoming:
        pyautogui.hotkey('ctrl', 'down')

    if 'Vdown' in incoming:
        pyautogui.hotkey('ctrl', 'up')

    incoming = ""

```

Figura 1.4.Codul intreg utilizat de programul Python

```

arduino_project | Arduino 1.8.5
File Edit Sketch Tools Help

arduino_project

const int trigger1 = 2; //Trigger pin of 1st Sensor
const int echo1 = 3; //Echo pin of 1st Sensor
const int trigger2 = 4; //Trigger pin of 2nd Sensor
const int echo2 = 5; //Echo pin of 2nd Sensor

long time_taken;
int dist,distL,distR;

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

/*Function to calculate distance*/
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);
  dist= time_taken*0.034/2;
  if (dist>50)
  dist = 50;
}

void loop() { //infinite loopy
  calculate_distance(trigger1,echo1);
  distL =dist; //get distance of left sensor

  calculate_distance(trigger2,echo2);
  distR =dist; //get distance of right sensor

  //Uncomment for debugging
  Serial.println("L=");
  Serial.println(distL);
  Serial.println("R=");
  Serial.println(distR);

  //Pause Modes -Bold
  if (distL >= 50 || distR >= 50 || distL <= 50 || distR <= 50) //Pause both hands

```

```

arduino_project | Arduino 1.8.5
File Edit Sketch Tools Help

arduino_project

//Pause Modes - Hold
if ((distL >40 && distR<40) && (distL <50 && distR<50)) //Detect both hands
{Serial.println("Play/Pause"); delay (500);}

calculate_distance(trigger1,echo1);
distL =dist;

calculate_distance(trigger2,echo2);
distR =dist;

//Control Modes
//Lock Left - Control Mode
if (distL==13 && distL==17)
{
  delay(100); //Hand Hold Time
  calculate_distance(trigger1,echo1);
  distL =dist;
  if (distL==13 && distL==17)
  {
    Serial.println("Left Locked");
    while(distL==40)
    {
      calculate_distance(trigger1,echo1);
      distL =dist;
      if (distL==10) //Hand pushed in
      {Serial.println ("Up"); delay (300);}
      if (distL==20) //Hand pulled out
      {Serial.println ("Down"); delay (300);}
    }
  }
}

//Lock Right - Control Mode
if (distR==13 && distR==17)
{
  delay(100); //Hand Hold Time
  calculate_distance(trigger2,echo2);
  distR =dist;
  if (distR==13 && distR==17)
  {
    Serial.println("Right Locked");
    while(distR==40)
    {
      calculate_distance(trigger2,echo2);
      distR =dist;
      if (distR==10) //Right hand pushed in

```

```

arduino_project | Arduino 1.8.5
File Edit Sketch Tools Help

arduino_project

calculate_distance(trigger1,echo1);
distL =dist;

//Control Modes
//Lock Left - Control Mode
if (distL==13 && distL==17)
{
  delay(100); //Hand Hold Time
  calculate_distance(trigger1,echo1);
  distL =dist;
  if (distL==13 && distL==17)
  {
    Serial.println("Left Locked");
    while(distL==40)
    {
      calculate_distance(trigger1,echo1);
      distL =dist;
      if (distL==10) //Hand pushed in
      {Serial.println ("Up"); delay (300);}
      if (distL==20) //Hand pulled out
      {Serial.println ("Down"); delay (300);}
    }
  }
}

//Lock Right - Control Mode
if (distR==13 && distR==17)
{
  delay(100); //Hand Hold Time
  calculate_distance(trigger2,echo2);
  distR =dist;
  if (distR==13 && distR==17)
  {
    Serial.println("Right Locked");
    while(distR==40)
    {
      calculate_distance(trigger2,echo2);
      distR =dist;
      if (distR==10) //Right hand pushed in
      {Serial.println ("Rewind"); delay (300);}
      if (distR==20) //Right hand pulled out
      {Serial.println ("Forward"); delay (300);}
    }
  }
}

delay(200);
}

```

Figurile 1.5,1.6 și 1.6 Codul complet al programului Arduino

Modul de asamblare este dupa cum urmează:

Pașul 1: Schema electrică

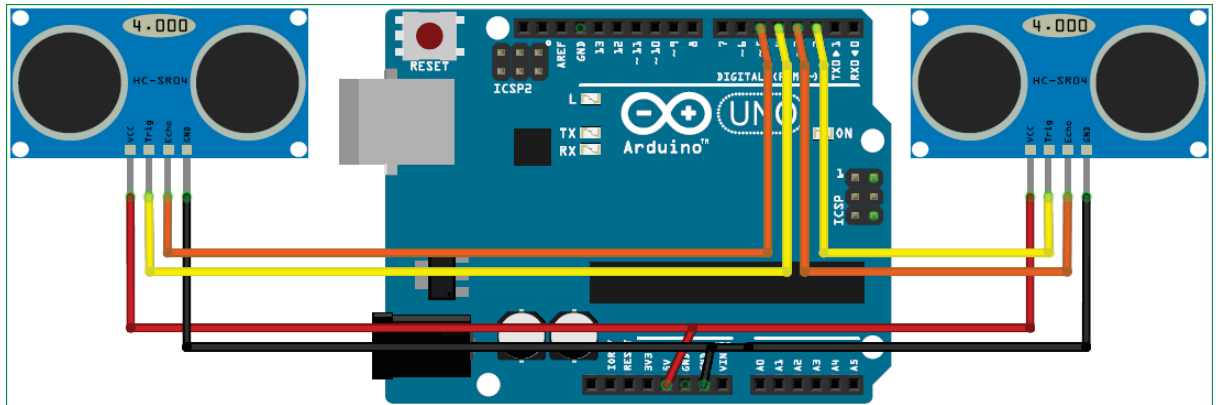


Figura 1.7.Schema electrică de conectare a senzorilor ultrasonici la placuța de dezvoltare

Pasul 2: Asamblarea

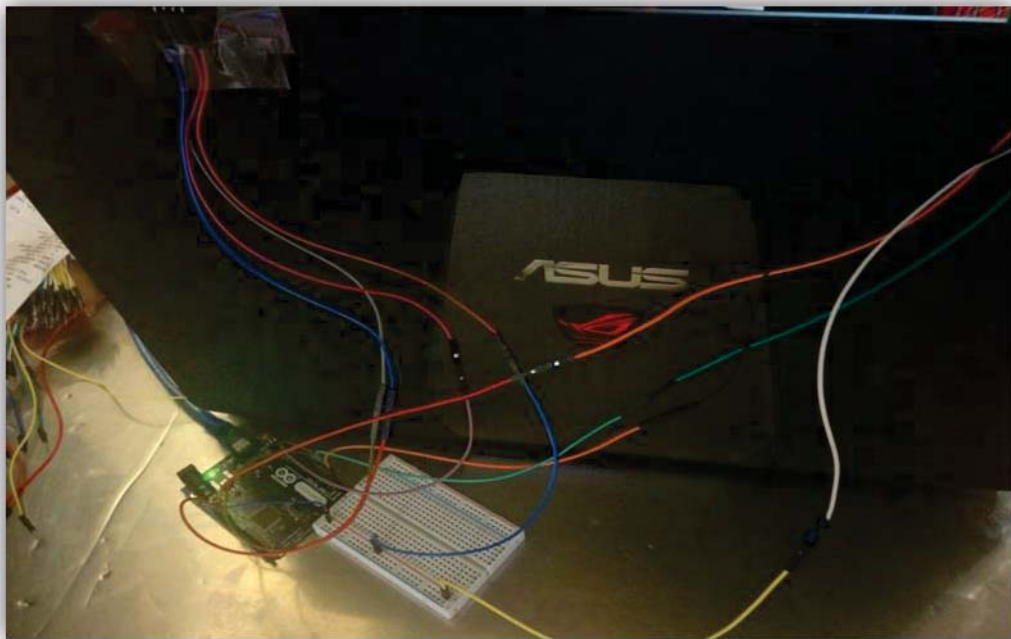


Figura 1.8.Asamblarea proiectului

Pasul 3: Testarea funcționalității



Figura 1.9. Test al asamblajului

2. Concluzii

Prin intermediul unor linii de cod și a materialelor programabile se poate transforma un calculator de tip PC, obiect utilizat în mod normal prin intermediul tastaturii și al mouse-ului într-un aparat wireless ce poate fi utilizat de la o distanță ce ține de alegerea proprie, creandu-se astfel o noua modalitate de a interacționa cu acesta. Acest lucru se poate aplica oricărui dispozitiv, atât timp cât sunt adăugate și programate componente precum senzorii, placile de dezvoltare și/sau modulele. Nu există o limită când vine vorba de ceea ce poate fi făcut prin dedicație și imaginație. O posibilitate este începutul a mii de experimente, iar prin încercare și eșec tehnologia avansează cu mult peste limitele anterioare.