UDC 378.147.111

COMPUTER GESTURE CONTROL

***Borovynska Y.**, student, MST Department, KhNURE*
***Pinzari M.**, student, Department, UAB ("1 Decembrie 1918" University of Alba Iulia)*
***Chebotarova I.**, Senior Lecturer, MST Department KhNURE*

***Abstract.** This article deals with the interaction of the high-level programming language Python in combination with an Arduino to create a video gesture control. The program was created with the required hardware and software. This gesture control is capable of performing various functions in the video properties, namely: video rewinding, playback and pause, volume change.*

***Keywords**: ARDUINO, PYTHON, GESTURE CONTROL, VIDEO, HANDS, CODE.*

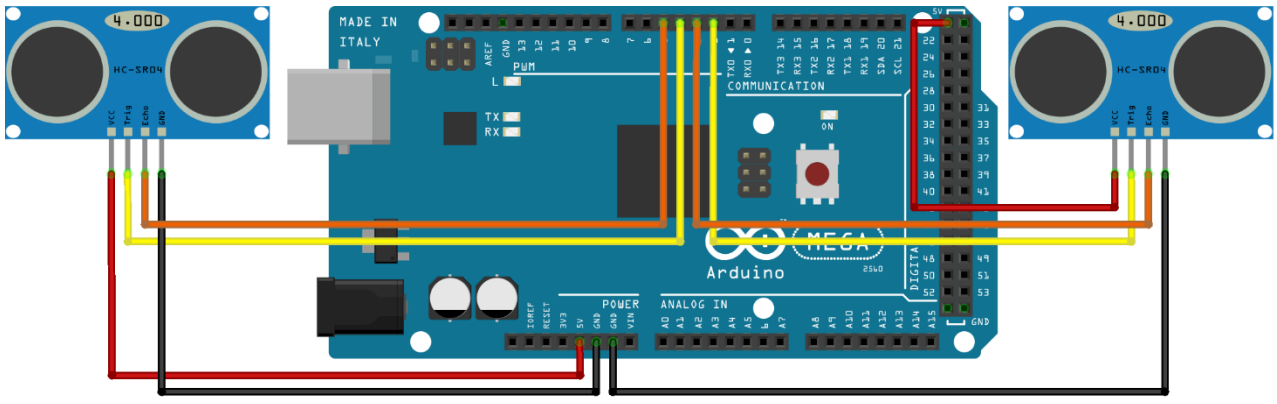
Recently mouse-controlled Laptops or computers are a common thing. However, the world is evolving and gesture control is becoming more and more popular. Nowadays we can create controlled systems using high-level general-purpose programming language Python coupled with Arduino. This code is used to control the video, namely, to rewind, pause, increase and decrease the volume.

The required equipment: two ultrasonic sensors (US), 8 wires, Arduino Mega 2560 Rev3. To create this controller you also need to install Python, Arduino, PySerial library, and Pyautogui [1].

The concept behind the project:

Two ultrasonic sensors (US) are placed on top of the monitor. The distance between the monitor and our hand is calculated with the Arduino. The commands from the Arduino are sent to the computer via the serial port (USB). This data is then read by the Python running on the computer and an action is performed based on the read data.

To control the computer with hand gestures, you need to connect two ultrasonic sensors to the Arduino as shown in the diagram below. The Arduino must be connected to a PC/laptop for power (pic. 1).



Picture 1 – Circuit Diagram

This program can perform 3 actions.

Action 1: When both hands are in front of the sensor at a certain distance, the video should play/pause.

Action 2: When the right hand is in front of the sensor at a certain short distance, the video should fast forward when moving towards the sensor, and rewind when moving away.

Action 3: With the left hand is in front of the sensor at a certain short distance, and then moving toward the sensor, the video volume should increase, and moving away from the sensor, the video volume should decrease.

Precisely, the calculate_distance() function returns the distance between the sensor and the hand. Knowing the distance between both sensors, the distance is compared to predefined values and certain actions are taken. For example, if both hands are 40 μm apart, we can play/pause the video. The words here are "Play/Pause."

We also control the video track by using the right sensor. If we move the right hand toward the right sensor, the video moves forward, and if we move it away from the sensor, the video rewinds. The words here are "Rewind" or "Forward". We use the same method for the left sensor to control the volume but with the left hand. The words here are "Vup" or "Vdown". The code for Arduino (pic. 2).

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

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

//Pause Modes - Hold
if ((distL >10 && distR>10) && (distL <20 && distR<20))//Detect both hands
{Serial.println("Play/Pause"); delay (500);}

calculate_distance(trigger1,echo1);
distL =dist;

calculate_distance(trigger2,echo2);
distR =dist;

//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 ("Vup"); delay (300);}
      if (distL>20) //Hand pulled out
      {Serial.println ("Vdown"); 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 (100);}
      if (distR>20) //Right hand pulled out
      {Serial.println ("Forward"); delay (100);}
  }
}
delay(200);
}
```

Picture 2 – The code for Arduino

We use the Arduino code to set the condition of a function, while we use the Python code to execute the functions on the video. At the same time, we need to install PySerial to communicate between the two codes. PySerial is a Python API module used to read and write serial data to the Arduino [2].

The second step with Python should be to install the Pyautogui module. You need open Windows Command prompt and change the directory to the folder where you have installed Python: cd C:\Python27. After that you need inside your Python directory use the command C:\Python27>python –m pip install --upgrade pip. And the last step is to use the command C:\Python27>python –m pip install pyautogui.

The final step in writing a program for the gesture control is to work with the Python code directly. It is required to connect Python and Arduino via COM port and specify the actions for each condition. The code for Python (pic. 3).

```
import serial
import time
import pyautogui

ArduinoSerial = serial.Serial('com5',9600)
time.sleep(2)

while 1:
    incoming = str (ArduinoSerial.readline())
    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 = "";
```

Picture 3 – The code for Python

References.

1. Arduino Web Editor 2022. https://www.arduino.cc/en/software.

2. Aswinth Raj. PySerial. Using Python with Arduino. https://circuitdigest.com/microcontroller-projects/arduino-python-tutorial.

3. Project Hub. https://create.arduino.cc/projecthub.

4. Biziuk, A., Biziuk, V., & Shakurova, T. (2021). Analysis of Teaching Elements on Technical and Mathematical Disciplines in Modern Distance Education. Technology Audit and Production Reserves, 4(2), 60.

5. Deineko, Z., Zeleniy, O., Lyashenko, V., & Tabakova, I. (2021). Color space image as a factor in the choice of its processing technology. Abstracts of I International scientific-practical conference «Problems of modern science and practice» (September 21-24, 2021). Boston, USA, pp. 389-394.