Arduino Based Hand Gesture Control of Your Computer

Submitted in partial fulfilment for the award of the degree of

Bachelor of Engineering

IN

Electronics and Communication Engineering

Submitted by

Saksham Saini -UE195099, Saurabh Sabharwal- UE195102,

Shashikant Sharma- UE195107, Sonali Joshi- UE195115

UNDER THE GUIDANCE OF

CHARU MADHU

(Asst. Professor, ECE, UIET)

UNIVERSITY INSTITUTE OF ENGINEERING & TECHNOLOGY

PANJAB UNIVERSITY: CHANDIGARH 160014

JUNE, 2022

DECLARATION

We hereby certify that the work which is being presented in this project entitled "Arduino based Hand Gesture Control of Your Computer", in partial fulfillment of the requirement for the award of degree of "Bachelor of Engineering in Electronics and Communication" submitted in University Institute of Engineering and Technology, Panjab University, Chandigarh, is an authentic record of our own work carried out under the supervision of Dr. Charu Madhu, UIET.

The matter embodied in this synopsis/ Project/ Thesis has not been submitted for the award of any other degree of this or any other university.

Name of the Student: Saksham Saini
Branch/Stream: BE, ECE
Roll No: UE195099

Name of the Student: Saurabh Sabharwal
Branch/Stream: BE, ECE
Roll No: UE195102

Name of the Student: Shashikant Sharma
Branch/Stream: BE, ECE
Roll No: UE195107

Name of the Student: Sonali Joshi
Branch/Stream: BE, ECE
Roll No: UE195115

This is to certify that the statements made above by the candidate are correct and true to the best of our knowledge.

TABLE OF CONTENTS

MOTIVATION

You might have seen Hand Gesture Controlled Robots, where the motion of a robot is controlled by the gestures of the hand. Another interesting project based on a similar principle is an Arduino based Hand Gesture Control of your computer or laptop.

Human Machine Interface or HMI is a system consisting of hardware and software that helps in communication and exchange of information between the user (human operator) and the machine.

We normally use LED Indicators, Switches, Touch Screens and LCD Displays as a part of HMI devices. Another way to communicate with machines like Robots or Computers is with the help of Hand Gestures

Instead of using a keyboard, mouse or joystick, we can use our hand gestures to control certain functions of a computer like play/pause a video, move left/right in a photo slideshow, scroll up/down in a web page and many more.

In this project, we have implemented a simple Arduino based hand gesture control where you can control few functions of your web browser like switching between tabs, scrolling up and down in web pages, shift between tasks (applications), play or pause a video and increase or decrease the volume (in VLC Player) with the help of hand gestures.

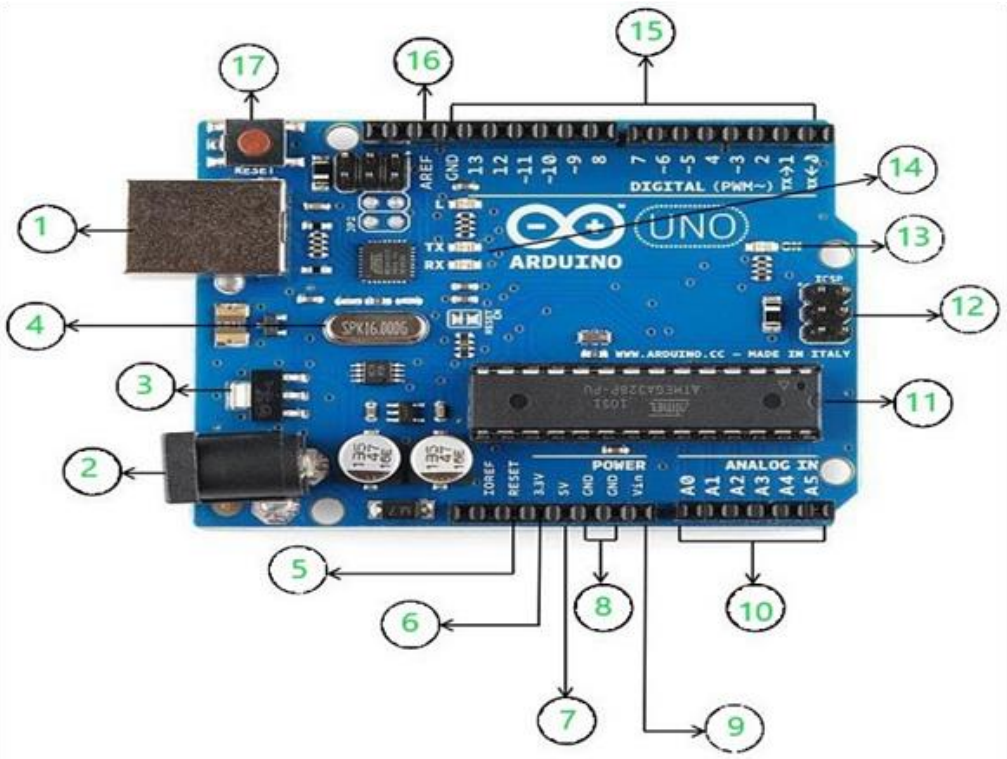
Chapter I. INTRODUTION TO ARDUINO

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, or the internet.

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users

Using the image as a reference, the labelled components of the board respectively are -



(Figure 1) *Adruino UNO Board with markings done (explained below)*

- USB: can be used for both power and communication with the IDE
- Barrel Jack: used for power supply
- Voltage Regulator: regulates and stabilizes the input and output voltages
- Crystal Oscillator: keeps track of time and regulates processor frequency
- Reset Pin: can be used to reset the Arduino Uno
- 3.3V pin: can be used as a 3.3V output
- 5V pin: can be used as a 5V output
- GND pin: can be used to ground the circuit
- Vin pin: can be used to supply power to the board
- Analog pins(A0-A5): can be used to read analog signals to the board
- Microcontroller(ATMega328): the processing and logical unit of the board
- ICSP pin: a programming header on the board also called SPI
- Power indicator LED: indicates the power status of the board
- RX and TX LEDs: receive(RX) and transmit(TX) LEDs, blink when sending or receiving serial data respectively
- Digital I/O pins: 14 pins capable of reading and outputting digital signals; 6 of these pins are also capable of PWM
- AREF pins: can be used to set an external reference voltage as the upper limit for the analog pins
- Reset button: can be used to reset the board

Chapter II. GETTING STARTED WITH ARDUINO

(Figure – 2) *Arduino IDE Code Window to write down the code*

As you open the IDE, you'll be greeted by a window similar to the one shown in the above image. The text editor is where you'll be writing your code; you'll use the verify button to compile and debug the written program, the save button to save the program and the upload button to upload the program to the board. Before you click on the upload button, it is necessary to select your board, Uno in this case, from the tools menu in the Menu Bar. After you choose your appropriate board, make sure you specify the correct port on your PC or Mac that you've connected your Uno to, in the IDE.

For example, we'll be using the inbuilt 'Blink' program. To open this program, go to the Files menu in the Menu Bar; click on Examples; click on 01.Basics; select Blink. Now that you've opened the

example program, it's time to upload the program, to do this, click on the upload button and wait for the process to complete. If your Output Pane header turns amber and shows an error which reads “Serial Port COM ‘x’ not found”, you’ve not connected your board correctly or that you’ve not specified the correct port that your board is connected to in the IDE. When you advance and start writing your own programs, you might run into errors while compiling and uploading; this can be because of a syntax error in the program. After you’ve corrected the errors and uploaded the program, you’ll see that the inbuilt LED blinks, alternating between the ON and OFF state every second.

Chapter III. HAND GESTURE AS APPLICATION

Gesture recognition refers to the mathematical interpretation of human motions using a computing device. It is a component of perceptual user interface (PUI). Other popular PUI components are voice recognition, facial recognition, and eye tracking. At present, gesture recognition is mainly centered on hand-gesture recognition and facial emotion recognition.

Recognizing gestures as input can be very helpful for physically impaired persons. In addition, gesture recognition triggers better, more natural interaction for a 3D virtual world or a gaming environment.

In this project, we have implemented Arduino based Hand Gesture Control of Your Computer, where a few hand gestures made in front of the computer will perform certain tasks on the computer without using a mouse or keyboard

Such Gesture based Control of Computers is already present and a company called Leap Motion has been implementing such technology in computers.

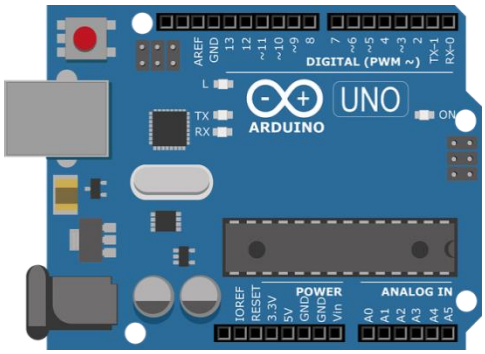
This type of hand gesture control of computers can be used for VR (Virtual Reality), AR (Augmented Reality), 3D Design, Reading Sign Language, etc.

Chapter IV. TECHNOLOGIES USED

Software/Programming Languages: Python, C++, Arduino IDE

Hardware Components:

Arduino UNO x 1



(Figure – 3) *Arduino UNO Board*

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board

Ultrasonic Sensors x 2

(Figure – 4) *Ultrasonic sensor and its components*

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal.
Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear).

Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emissions of the sound by the transmitter to its contact with the receiver.

The formula for this calculation is D = ½ T x C (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second)

USB Cable (for Arduino)

This cable is used to interface any of the Arduino board with your computer; you can also connect your USB printer, scanner, and more to your computer. These cables Transmits data at high speeds with the error-free, high-performance transmission

(Figure – 5) *An Arduino USB cable*

Few Connecting Wires

Connecting wires allows an electrical current to travel from one point on a circuit to another because electricity needs a medium through which it can move.

Most of the connecting wires are made up of copper or aluminum

(Figure – 6) *Bunch of connecting wires*

Chapter V. PYTHON LIBRARIES USED

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

We have integrated python into our project to make use of the varieties of the libraries available. To implement the gestures, the arduino code is run by the 'serial' library in the python module which is responsible for collecting data by the serial monitor, which collects information from the actions performed in front of the ultrasonic sensors.

Next, the gestures are run with the help of 'pyautogui' library in the python modules which can control keyboard and mouse actions. It is done by specifying what is the task to be performed.

In our project, we have specified conditions in terms of distance of the hand in front of the sensors. This distance changes with time, and according to the gesture received, the arduino code prints the action on the serial monitor. This information is picked up by the 'serial' module and then using conditional statements, the 'pyautogui' makes a decision as per the specified command.

For example, for 'play/pause' action on a media player, we need to click 'space-bar' on the keyboard. This can be achieved by the python module by specifying it to take enable action similar to one press of the space-bar key. This is done in our project by stating if the distance received by the sensors is between 25 and 50, print 'Play/Pause' on the serial monitor. This data is read by the python script and the conditional statement returns a 'TRUE' value and hence the action is taken corresponding to the data received on the serial monitor.

(Figure – 7) *Python Language*

V.I. Serial Module in Python (Background)

This module encapsulates the access for the serial port. It provides backends for Python running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named “serial” automatically selects the appropriate backend.

For our project, we create a variable ‘serial’ class to specify the serial port that will be used for communication. A COM port is simply an I/O interface that enables the connection of a serial device to a computer. In our variable python class of type ‘serial’, the COM port and the baud rate are specified.

It basically connects the serial port data of Arduino code to be used by the python script in further steps of this project. The serial port data is then stored into a string variable, which is then processed by the ‘pyautogui’.

V.II. Pyautogui Module in Python (Background)

PyAutoGUI lets your Python scripts control the mouse and keyboard to automate interactions with other applications. The API is designed to be simple.

PyAutoGUI has several features: Moving the mouse and clicking in the windows of other applications, sending keystrokes to applications (for example, to fill out forms), take screenshots, and given an image (for example, of a button or checkbox), and find it on the screen, locate an application’s window, and move, resize, maximize, minimize, or close it (Windows-only, currently), display alert and message boxes.

It’s hard to use the mouse to close a program if the mouse cursor is moving around on its own. As a safety feature, a fail-safe feature is enabled by default. When a PyAutoGUI function is called, if the mouse is in any of the four corners of the primary monitor, they will raise a pyautogui.FailSafeException. There is a one-tenth second delay after calling every PyAutoGUI function to give the user time to slam the mouse into a corner to trigger the fail safe.

Using the functions given in the module, we can implement keyboard and mouse control and hence, for example, using ‘PgUp’ and ‘PgDn’ keyboard buttons, we have implemented the scroll up and down feature as a gesture.

Chapter VII. GESTURES IMPLEMENTED

Media Controls (4):

- Play/Pause
- Reverse
- Forward
- Next

Other Controls (2):

- Page Scroll Up
- Page Scroll Down

(Figure – 9) *VLC Media Player controls added in the project*

(Figure – 10) *Person demonstrating a game being played using gestures*

Chapter VIII. SCOPE OF IMPROVEMENT

The project can be extended by enabling more interactive features. This project sets up a basis for augmented reality. We can have more applications (or tasks) which can be controlled via this same model. Our project is focused on taking the next step towards future automation.

This project opens up options for applications and devices which can offer gesture control as a feature. It can be viewed as making tasks easier and faster

Improvements on accuracy, precision and testing can be explored.

- More controls can be added to the portfolio using the ‘pyautogui’ module
- Complete mouse control using gestures - basis of augmented reality

Chapter IX. REFERENCES

- https://www.tinkercad.com/dashboard
- https://pypi.org/project/PyAutoGUI/
- https://pyserial.readthedocs.io/en/latest/pyserial.html
- https://docs.python.org/3