AUDIO CONTROLLED WINDOWS INTERFACE

Project Report

Submitted for the course:

Human Computer Interaction

By

Sukrit Bedi	16BCE0658
Malay Joshi	16BCE0953
Abhaya Shrestha	16BCE2291

Slot: C2+TC2

Name Of faculty: Thenmozhi T

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and I would not forget to thank them.

In preparation of our project, we had to take the help and guidance of some respected persons, who deserve our deepest gratitude. As the completion of this project gave us much pleasure, we would like to show our gratitude to Ms. Thenmozhi, Course Instructor, on VIT University for giving us a good guideline for the project throughout numerous consultations. We respect and thank Ms. Thenmozhi T, for providing us an opportunity to do the project and giving us all support and guidance till the completion of our project work by providing all the necessary information for developing a good system.

We would also like to expand our gratitude to all those who have directly and indirectly guided us in writing this assignment. Many people, especially our classmates have made valuable comment suggestions on our project which gave us an inspiration to improve the quality of the project.

Literature Survey

The graphical user interface (GUI), which is dominated by almost all user interface (UI) designs, is sometimes cumbersome on small display devices. If the device does not have a display, it is almost useless if it is visually impaired or useless. We need to consider the appropriate input and output (I / O) modal and interactive techniques to suit various conditions. Using these devices now is part of everyone's daily life, whether we realize it or not. Since the GUI is not the most suitable in many cases, we should have an alternative, which is an audio-based interface. The goal is to provide information when transferring additional cognitive loads to different modalities.

The audio-based interface designed for the public should emphasize ease of use and provide a lot of help and guidance for first-time callers. In contrast, audio designed for a small number of advanced users, including field service personnel, should focus more on productivity than on more help and guidance. Such applications should streamline the call flow, minimize prompts, eliminate unnecessary iterations, and allow a well-designed "hybrid active conversation," which allows callers to enter multiple pieces of information in a single utterance and in any order or combination. In short, voice applications must be carefully designed for specific business processes that are being automated.

Problem Description

The evolution of user interface has witnessed the development of input from text-based keyboard input too GUI based on mouse. However, this is inconvenient and unnatural. Our objective is to develop a new interface to improve the interaction between human beings and the computer without the use of any mechanical devices such as the mouse. The system works by the use of an accelerometer which is attached to the head of the user and records the coordinates. These readings, by means of the Arduino microcontroller are input into a python application which reflects the movement of the user's hand onto the screen. We have used this method as such gesture-controlled systems usually use image processing techniques but have aimed to implement this because image processing requires very large processing power and there is usually a lot of external dependencies such as illumination conditions are involved. The system we have proposed will make the tedious and redundant process of moving the cursor again and again using the mouse simpler by eliminating the need for the mouse at all.

Working

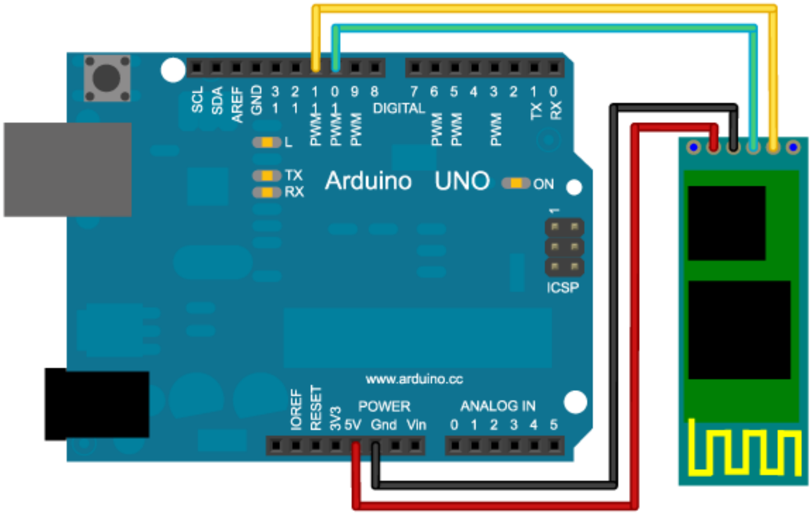
The product we designed is a small size device that when attached to the computer takes a serial comm. port and behaves as a serial device. When the program file is executed, it looks for the device and which port it is on. After locating the product, the code connects to it using serial communication method. It gives a signal to the device that system is ready to take inputs.

The device is also connected to an App in the phone. This app functions as a command relay module. As the command is taken in audio form, it is simultaneously being processed using Google Speech to Text API, which is embedded in the app. The processed text is then sent to the device via Bluetooth.

In the code, we have used a library called as pyautogui which is a module for controlling mouse and keyboard. It is used in real life for creating an automation setup between humans and computers. PyAutoGui can simulate a lot of things, a few of which are listed down:

- Mouse movement
- Mouse drag
- Tween/Easing Functions
- Mouse Clicks (all sorts of possible combinations)
- Mouse up and mouse down functions
- Scrolling
- Typewrite
- Press
- Keydown, keyup
- Hotkeys

......and many more.



Made with Fritzing.org

Fig 1. Simulation of circuit

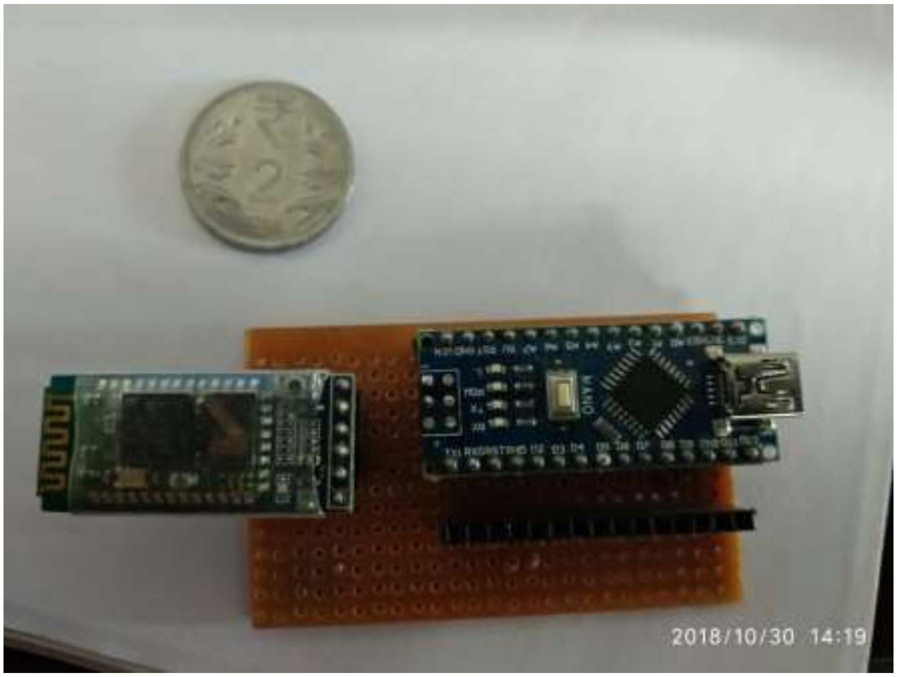


Fig 2. Device in scale with Rs2 coin

Coding Involved

Test.py

```
from pyautogui import *
import serial
from threading import Thread
import time
import serial
import os
import csv

def program1():
ser3=serial.Serial("com3",9600)
finish=False
print("JJ")

while True:
t=ser3.readline().strip()#.decode('utf-8')
t=t.replace("*","")
t=t.replace("#","")
print(t)
words=t.split();
print(words)
try:
if t=="left click":
click()
if t=="double click":
doubleClick()
if t=="right click":
rightClick()
if words[0]=="open":
tw1=words[1]#.split("#")
print(tw1)
try:
os.startfile(tw1)
except:
if tw1.lower() == "excel":
target = "C:\Program Files (x86)\Microsoft Office\Office12\excel"
tw1 = "excel"
if tw1.lower() == "powerpoint":
    target = "C:\Program Files (x86)\Microsoft Office\Office12\powerpnt"
    tw1 = "powerpnt"
    if tw1.lower() == "word":
    target = "C:\Program Files (x86)\Microsoft Office\Office12\winword"
    tw1 = "winword"
    if tw1.lower() == "arduino":
    target = "C:\Program Files (x86)\Microsoft Office\Office12\arduino"
    tw1 = "arduino"
    os.startfile(target)


    if t=="start typing":
    while not finish:
    t=ser3.readline().strip().decode('utf-8')
    t=t.replace("*",")
    t=t.replace("#",")
    print(t)
    if(t=="stop typing"):
    finish=True
    else:
    tem1=t
    tem1=tem1.replace("full stop",".")
    tem1=tem1.replace("comma",",")
    tem1=tem1.replace("question mark","?")
    tem1=tem1.replace("tab","   ")
    typewrite(tem1+"\n",interval=0.001)
    if t=="close":
    os.system("TASKKILL /F /IM "+tw1+".exe")
    if t=="go to sleep":
    quit()
    finish=False

    except:
    print("Command not found")
program1()
```

ard.ino

```
    #include<SoftwareSerial.h>

    SoftwareSerial bluetooth(10, 11);
    void setup() {
    bluetooth.begin(38400);
    Serial.begin(9600); // Default communication rate of the Bluetooth module
    }
    void loop() {
    //int state = Serial.read(); // Reads the data from the serial
    if(bluetooth.available()){
    String state = bluetooth.readStringUntil('\n'); // Reads the data from the serial
    Serial.println(state);
    }
    //Serial.write(state);
}
```

Future Endeavours

In future, we plan to make a batch scripted model as a UI for the program. Also, there would an option to add custom commands, so the user can customize the product according to its own needs and style. Also, we are planning to switch to another microcontroller platform as in Arduino, there is a lot of wastage of pins. Due to this, we have a steep rise in cost than using another cheaper alternative, which would be having lesser number of wastages in terms of computational power, IO pins and size. Also, a cleaner PCB and a custom printed shell would help to improve the appearance of the product.

References

a. S. (2016, July 25). Controlling Mouse Pointer using Python Script and Accelerometer. Retrieved February 27, 2017, from https://www.engineersgarage.com/contribution/controlling-mouse-pointer-using-python-script-andaccelerometer

b. How do I get an 3-Axis Accelerometer to control processing game? - Please Help. (n.d.).
Retrieved February 27, 2017, from http://forum.arduino.cc/index.php?topic=301108.0

c. Berlia, R., & Santosh, P. (2014). Mouse Brace: A convenient computer mouse using accelerometer, flex sensors and microcontroller. *2014 International Conference on Contemporary Computing and Informatics (IC3I)*.

d. Huang, W., Hou, H., & Fang, W. (2013). A remote control solution for mouse cursor of computer by using accelerometer. *2013 IEEE International Symposium on Consumer Electronics (ISCE)*