

ABSTRACT

- The way our device functions is very simple. The text to be converted is given as an input in the form of a string of characters. The string is then transferred to a database which is then converted into Braille. The database sends signals to the mechanism which is a combination of servo motors which raise the respective tiles in order to represent the individual letters of the string.
- In order to achieve this task, we are going to use multiple servo motors which will raise tiles in order to represent letters in braille letters. They will be co-ordinated using Raspberry pi and Arduino module. The input will be given using a keypad system (electronic devices such as computer etc.) and will then be transferred to a database.

TABLE OF CONTENTS

- 1) Introduction
- 2) Background
- 3) Problem Definition
- 4) Result and Discussion
- 5) Methodology
- 6) Conclusion and Future scope
- 7) References
- 8) Appendix

INTRODUCTION

- Our project aims at helping the people who do not possess sight and hearing. In our world we believe in equality and hence everyone must be given the same platform. People who are deaf and dumb have to struggle with every day to day task. Our device strives to make these mundane tasks easier for them.
- The biggest asset of these people is their sense of touch. Hence, they use braille as a language for reading. They are raised dots which represent words of the English alphabet. Hence they are able to read the whole of the English language. Our device aims to convert any text which is given into Braille instantaneously. This makes reading very convenient and in turn day to day life much easier for them.

BACKGROUND / PROBLEM DEFINITION

The major problem faced by the blind people is to read and write. Education for these students is relatively tough and even the books in braille are bit costly. If a blind child belongs to a poor family then it become really tough for that child to survive with the current pace and competition in education.

Our product is very cost effective and comes with a graphical user interface so that it become easy for teacher to use. Moreover this product is quite handy which is definitely a plus point as students can be take this product with them anywhere.

RESULTS AND DISCUSSIONS

Keeping the comfortability of teacher as well as students, we have designed our project in best possible way without much technical complexity. A simple GUI is made where teacher can pass the string or characters bit by bit or maybe group of character, servo act according to the written code and raise the dots. A flat surface is made with servo motors attached behind it.

The project is assumed to give best quality beside it's low cost and could be easily implemented in special schools for blind students.

Methodology:

We wanted to create such a device which could be used easily by anybody. In order to do so we came to the conclusion that the input will be received using keyboard of a computer and the output is reflected in the device.

The input is given in a GUI which is created using Python. First the user is asked to input their login credentials. The system checks the input credentials with the ones already stored in the database. If the credentials entered are incorrect then the system gives a pop up saying “try-again”. If the credentials are accepted then another GUI pops up in which the user can enter the string (sentence) which is to be converted in braille. Once the sentence is entered the user can click the submit button to proceed with the given input.

The input then gets stored in a database. The database has been created using Firebase. As there are process that need to be conducted with the database and the GUI which we have created and Raspberry Pi, we need to use such a programming language which is compatible with Firebase. Hence Python is used as a programming language.

Once the database gets updated with a string, it gets immediately extracted by Raspberry Pi. The database is linked to both the GUI and Raspberry Pi using Python. Once the string is extracted by the Raspberry Pi it gets transferred to the Arduino board. This is done as servo motors require analog input but Raspberry Pi is only able to send digital outputs. The Arduino board is hence used to convert the digital signal to analog signals.

The analog signals are sent to a combination of six servo motors arranged in a rectangular 3 by 2 matrix fashion. In braille each character is represented as a dot with a maximum of six dots which are arranged in a 3 by 2 matrix fashion. In our device a dot

is represented by an activated servo motor. The flap of the servo motor in an activated form acts as a protrusion from the normal plane surface hence it can be identified as a dot by a blind person. Every letter of the English alphabet has a given method of representation in braille and we have accordingly programmed the servo motors to get activated in the needed combination. The device represents one letter at a time for a period of 6 seconds after which it begins to represent the next letter. A blank space represented as a delay where all the servo motors get deactivated for a period of 10 seconds. This process is repeated till the entire string gets converted.

Hence the device is easier to use and the conversion of the string happens almost instantaneously. The device has been made such that it is portable and very easy to use. This device will be successful in making the lives of people who are deaf and blind much easier.

CONCLUSIONS AND FUTURE SCOPE

Based on the findings and our experience of developing this project, we may draw following conclusions: -

- 1) Our prototype and the idea is fit for commercialisation.
- 2) The project merely acts as a prototype show piece and not for real time practical usage immediately.
- 3) Further research and analysis, along with capital investment, is required to make it practically, economically and commercially applicable.

The future scope of the project includes:-

- 1) Building an android as well as desktop application
- 2) The project aims to be an open ended module, In other words, the very nature of the project is to always be open for development.
- 3) Need to reduce the size of the product and to make it as small as wrist watch.

References:

<https://www.youtube.com/watch?v=Dgo7F-lpyYE>

<http://pyfound.blogspot.com/2019/10/cpython-core-developer-sprint-2019.html?>

<https://pythoninsider.blogspot.com/2019/11/python-359-is-released.html?>

<http://pyfound.blogspot.com/2019/10/chris-angelico-2019-q2-community.html?>

<https://projects.raspberrypi.org/en/pathways/build-a-robot>

<https://projects.raspberrypi.org/en/codeclub/python-module-1>

APPENDIX

Login.py:

```
import tkinter as tk
```

```
from tkinter import ttk
```

```
import mysql.connector
```

```
from subprocess import call
```

```
# Creating connection with the  
database
```

```
db_connection =  
mysql.connector.connect(
```

```
    host="localhost",
```

```
    user="root",
```

```
    passwd="password",
```

```
    database="Bareli")
```

```
# Creating db cursor for sql  
operation
```

```
db_cursor = db_connection.cursor()
```

```
win = tk.Tk()
```

```
win.title("LOGIN")
```

```
win.geometry("300x200+200+100")
```

```
def validate():
```

```
    action.configure(text='Done')
```

```
    usr = name.get()
```

```
    psd = pswd.get()
```

```
cred = (usr, psd)
```

```
    # Validate username & password  
from database
```

```
    data = "SELECT * FROM Barfi  
WHERE username = %s and  
password = %s"
```

```
    db_cursor.execute(data, cred)
```

```
count = 0
```

```
for db in db_cursor :
```

```
    # print(db)
```

```
    count+=1
```

```
if count != 0:
```

```
    print("Congrats!!!")
```

```
    print("You are logged in.")
```

```
    win.destroy()
```

```
    # executing 'Form_Submission'  
file
```

```
    call(["python3",  
"Form_Submission.py"])
```

```
else:
```

```
    status['text']="Incorrect  
Username/Password"
```

```
    print("Incorrect  
Username/Password")
```

```
# Creating User Interface
```

```
ttk.Label(win,  
text="Name").place(x=40, y=25)
```

```
name = tk.StringVar()
```

```
nameEntered = ttk.Entry(win,  
width=15, textvariable=name)
```

```
n = nameEntered.place(x=130,  
y=25)
```

```
ttk.Label(win,  
text="Password").place(x=40, y=65)
```

```
pswd = tk.StringVar()
```

```
pswdEntered = ttk.Entry(win,  
width=15, textvariable=pswd)
```

```
p = pswdEntered.place(x=130,  
y=65)
```

```
status = ttk.Label(win, text="ready  
to go ?")
```

```
status.place(x=50 ,y=110)
```

```
action = ttk.Button(win,  
text="Login", command=validate)
```

```
action.place(x=120, y=150)
```

```
win.mainloop()
```

Upload.py:

```
import tkinter as tk
```

```
from tkinter import ttk
```

```
import firebase_admin
```

```
from firebase_admin import  
credentials
```

```
from firebase_admin import db
```

```

cred =
credentials.Certificate(r"rules.json"
)

firebase_admin.initialize_app(cred,
{
    'databaseURL' : 'https://barfi-
9a5f3.firebaseio.com/'
})

ref = db.reference()

users_ref = ref.child('values')


win = tk.Tk()

win.title("Input Form")

win.geometry("320x180+200+100")


def clickMe():

action.configure(text='Converting...'
)

    users_ref.update({'transfer':
name.get()})

```

Servo Motor's Code

// Include the Servo library

#include <Servo.h>

Servo Servo1;

Servo Servo2;

Servo Servo3;

Servo Servo4;

Servo Servo5;

Servo Servo6;

```

# interface

ttk.Label(win, text="Enter
sentence").place(x=20, y=85)

ttk.Label(win,
text="~WELCOME~").place(x=1
22, y=12)

ttk.Label(win, text="Hope we will
serve you the best.").place(x=52,
y=37)

name = tk.StringVar()

nameEntered = ttk.Entry(win,
width=20, textvariable=name)

a = nameEntered.place(x=130,
y=85)

action = ttk.Button(win,
text="Convert",
command=clickMe)

action.place(x=127, y=130)

win.mainloop()

```

void setup() {

**// We need to attach the
servo to the used pin number**

Servo1.attach(7);

Servo2.attach(6);

Servo3.attach(5);

Servo4.attach(12);

Servo5.attach(11);

Servo6.attach(10);

retriever.py

Import database module.

import firebase_admin

**from firebase_admin import
credentials**

from firebase_admin import db

```

cred =
credentials.Certificate(r"rules.json"
)

```

```

firebase_admin.initialize_app(cred,
{'databaseURL' : 'https://barfi-
9a5f3.firebaseio.com/'})

```

ref = db.reference()

**# Read the data at the posts
reference (this is a blocking
operation)**

print(ref.child('values').get())

Serial.begin(9600);

}

char rx_byte = 0;

void loop() {

if (Serial.available() > 0) {

Serial.println("Enter any char");	Servo1.write(90);	{
rx_byte = Serial.read();	Servo3.write(90);	Servo2.write(0);
	}	Servo1.write(0);
	else if(rx_byte == 'd')	Servo3.write(0);
// check if a number was received	{	
if (rx_byte == 'a')	Serial.print("Wooh.We received letter d: ");	delay(5000);
{	Servo1.write(0);	Servo2.write(90);
Serial.print("aman");	Servo3.write(0);	Servo1.write(90);
Servo1.write(0);	Servo4.write(0);	Servo3.write(90);
delay(5000);	delay(5000);	}
Servo1.write(90);	Servo1.write(90);	
}	Servo3.write(90);	
else if(rx_byte == 'b')	Servo4.write(90);	else if(rx_byte == 'g')
{	}	{
Serial.print("Wooh.We received letter b: ");	else if(rx_byte == 'e')	Servo2.write(0);
Servo1.write(0);	{	Servo1.write(0);
Servo2.write(0);		Servo3.write(0);
delay(5000);	Serial.print("Wooh.We received letter d: ");	Servo4.write(0);
Servo1.write(90);	Servo1.write(0);	delay(5000);
Servo2.write(90);	Servo4.write(0);	
}		Servo4.write(90);
else if(rx_byte == 'c')	delay(5000);	Servo2.write(90);
{		Servo1.write(90);
Serial.print("Yooo Weceived letter c: ");	Servo1.write(90);	Servo3.write(90);
Servo1.write(0);	Servo4.write(90);	}
Servo3.write(0);	}	
delay(5000);	else if(rx_byte == 'f')	else if(rx_byte == 'h')

{	delay(5000);	else if(rx_byte == 'm')
Servo2.write(0);		{
Servo1.write(0);	Servo4.write(90);	Servo5.write(0);
Servo4.write(0);	Servo2.write(90);	Servo1.write(0);
	Servo3.write(90);	Servo3.write(0);
delay(5000);	}	
	else if(rx_byte == 'k')	
Servo4.write(90);	{	delay(5000);
Servo2.write(90);	Servo5.write(0);	
Servo1.write(90);	Servo1.write(0);	Servo5.write(90);
		Servo1.write(90);
}	delay(5000);	Servo3.write(90);
else if(rx_byte == 'i')		}
{	Servo1.write(90);	else if(rx_byte == 'n')
Servo2.write(0);	Servo5.write(90);	{
Servo3.write(0);	}	Servo5.write(0);
	else if(rx_byte == 'l')	Servo1.write(0);
delay(5000);	{	Servo3.write(0);
	Servo2.write(0);	Servo4.write(0);
Servo2.write(90);	Servo1.write(0);	
Servo3.write(90);	Servo5.write(0);	delay(5000);
}		
else if(rx_byte == 'j')	delay(5000);	Servo5.write(90);
{		Servo4.write(90);
Servo2.write(0);	Servo5.write(90);	Servo1.write(90);
Servo3.write(0);	Servo2.write(90);	Servo3.write(90);
Servo4.write(0);	Servo1.write(90);	}
	}	else if(rx_byte == 'o')

{	Servo5.write(0);	else if(rx_byte == 's')
Servo1.write(0);	Servo2.write(0);	{
Servo5.write(0);	Servo1.write(0);	Servo2.write(0);
Servo4.write(0);	Servo3.write(0);	Servo5.write(0);
	Servo4.write(0);	Servo3.write(0);
delay(5000);		
	delay(5000);	delay(5000);
Servo4.write(90);		
Servo5.write(90);	Servo5.write(90);	Servo5.write(90);
Servo1.write(90);	Servo4.write(90);	Servo2.write(90);
	Servo2.write(90);	Servo3.write(90);
}	Servo1.write(90);	}
else if(rx_byte == 'p')	Servo3.write(90);	else if(rx_byte == 't')
{	}	{
Servo2.write(0);	else if(rx_byte == 'r')	Servo2.write(0);
Servo1.write(0);	{	Servo5.write(0);
Servo3.write(0);	Servo2.write(0);	Servo3.write(0);
Servo5.write(0);	Servo1.write(0);	Servo4.write(0);
	Servo5.write(0);	
delay(5000);	Servo4.write(0);	delay(5000);
Servo5.write(90);	delay(5000);	Servo4.write(90);
Servo2.write(90);		Servo2.write(90);
Servo1.write(90);	Servo4.write(90);	Servo5.write(90);
Servo3.write(90);	Servo2.write(90);	Servo3.write(90);
}	Servo1.write(90);	}
else if(rx_byte == 'q')	Servo5.write(90);	else if(rx_byte == 'u')
{	}	{

```

Servo6.write(0);
Servo1.write(0);
Servo5.write(0);

delay(5000);

Servo5.write(90);
Servo6.write(90);
Servo1.write(90);
}
else if( rx_byte == 'v')
{
Servo2.write(0);
Servo1.write(0);
Servo5.write(0);
Servo6.write(0);

delay(5000);

Servo5.write(90);
Servo2.write(90);
Servo1.write(90);
Servo6.write(90);
}
else if( rx_byte == 'w')
{
Servo2.write(0);
Servo6.write(0);

```

```

Servo3.write(0);
Servo4.write(0);

delay(5000);

Servo4.write(90);
Servo2.write(90);
Servo6.write(90);
Servo3.write(90);
}
else if( rx_byte == 'x')
{
Servo5.write(0);
Servo1.write(0);
Servo3.write(0);
Servo6.write(0);

delay(5000);

Servo5.write(90);
Servo2.write(90);
Servo1.write(90);
Servo6.write(90);
}
else if( rx_byte == 'y')
{
Servo6.write(0);

```

```

Servo5.write(0);
Servo1.write(0);
Servo3.write(0);
Servo4.write(0);

delay(5000);

Servo6.write(90);
Servo4.write(90);
Servo5.write(90);
Servo1.write(90);
Servo3.write(90);
}
else if( rx_byte == 'z')
{
Servo5.write(0);
Servo1.write(0);
Servo6.write(0);
Servo4.write(0);

delay(5000);

Servo5.write(90);
Servo4.write(90);
Servo1.write(90);
Servo6.write(90);
}}

```