Mobile phone controlled Smart Glasses

"VISION"

By

Areeba Waheed (Leader)

Muhammad Abdullah Fakhar

Muhammad Talha Akram

Wahaj Khalid Sultan

Submitted to the Faculty of Department of Electrical (Telecommunications) Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of B.E Degree in Electrical (Telecommunications) Engineering

JUNE 2021

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

"Mobile phone controlled Smart Glasses"

is carried out by

Areeba Waheed, Muhammad Abdullah Fakhar, Muhammad Talha Akram,

Wahaj Khalid Sultan

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical Engineering from National University of Sciences and Technology (NUST), Islamabad.

Approved By:

Signature:

Supervisor: **Asst. Prof. Mir Yasir**

MCS, Rawalpindi

DECLARATION OF ORIGINALITY

We hereby declare that no content of work presented in this report has been submitted in support of another award of qualification or degree either in this institution or anywhere else.

PLAGIARISM CERTIFICATE (TURNITIN REPORT)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Areeba Waheed
214201

Muhammad Talha Akram
215976

Muhammad Abdullah Fakhar
214885

Wahaj Khalid Sultan
138031

_____

Signature of Supervisor

“What we know is a drop, what we don't know is an ocean”

-

Isaac Newton

ACKNOWLEDGEMENTS

Allah Subhan'Wa'Tala is the sole guidance in all domains.

Our parents, colleagues and most of all supervisor, Dr. Mir Yasir, this could not have been possible without your input.

The group members, who through all adversities worked steadfastly.

ABSTRACT

Mobile phone controlled Smart Glasses

Our project is based on giving our users the opportunity to use a platform to increase their multitasking abilities. Using mobile phone for basic functions makes our hands tied and multitasking becomes difficult. The project is aimed with designing a solution that can combine our mobile phones and our real lives together and give user an augmented reality experience and introduce a smart solution in that regard.

A prototype glasses frame is designed for people in academics and business environment with a camera for photo capture and OCR feature. This captured data is fed into a Raspberry Pi microcontroller that uses Image Processing and processes these images. After processing, microcontroller communicates with android application on the phone via Bluetooth where the processed text and data is stored.

This project aims at designing a scalable and economical hands-free solution for productivity with the ability to add additional features as per user requirement.

Table of Contents

List of Figures

CHAPTER 1

Introduction

INTRODUCTION

Our team has designed smart glasses for people with business life or in academics which has three main features:

- Answering calls and messages
- Display notifications
- Optical Character Recognition (to make easily editable notes)

This project can be divided into four main parts. The first unit contains a Raspberry Pi microcontroller which will be used as the main processing unit, taking raw values from the components through a wired medium and processing them. The Pi has connected to it; a camera, a display, a Bluetooth module, a transducer and a battery to power it. The second unit consists of the camera which would be used for capturing images and Optical Character Recognition (OCR). The third unit is the special near eye display unit which displays the incoming notifications from the mobile phone via Bluetooth. The fourth platform is an android application which communicates with the Raspberry Pi via Bluetooth. While in use, the user has the option to press a button to trigger OCR. An image is captured then gets processed by the microcontroller and data in the form of text is saved in the mobile phone via Bluetooth. Smart glasses provide additional features like Photo capture and voice call. In addition, several features can be integrated inside the glasses to increase the productivity of our users like voice and sound assistance, video calling etc. Keeping in mind, that the price of the glasses increases relatively if we add more features, we developed the prototype with just the basic phone functions and Optical Character Recognition (OCR).

1.1 Problem Statement

Everyone wants to be more productive. Taking care of emails, calls and text messages at priority basis is what people seek since the innovation in the mobile technology. We've all tried to implement ways to make ourselves or our teams more productive at some point in time. Optimizing productivity is becoming more technical [21]. For centuries, the primary purpose of eyeglasses has been to improve our vision. However in the current times, eyeglass makers and internet pioneers are joining forces to make our one-trick-pony glasses smarter as well [20].

1.2 Solution

The goal of the project is to introduce a smart glasses wearable technology device with Bluetooth and a camera for connectivity with one's. These glasses will aid the users in their daily routines which will make them blend their work with their mobile phones without the need of reaching for it every time it rings. Unavailability of a similar product in Pakistani market and wearables being the next big innovation in technology is also a motivation as time management and multitasking is a major issue world-wide including most of the population in Pakistan. Also keeping in mind there are many types of disabilities. Advanced technology has been used in helping people overcome such impairments. However, developing such an advanced technology is expensive, thus increasing the retail price as more features are introduced.

1.3 Scope and limitations

This project uses modern day tech to design a wearable computer with the following scope:

- Using Raspberry-Pi and a Camera unit to improve a traditional pair of glasses.
- The Smart Glasses are more than just fashion equipment with introduction of smart features.

- Using advanced Python based back-end programs to aid the user in their daily tasks' management.
- User friendly android application.
- Wireless (via Bluetooth) integration of android application and raspberry-pi improves interfacing.

1.4 Objectives

We have the following objectives for the design of our project:

- Our focus is on developing a hands-free solution for academic and commercial purposes.
- Features that are added in the android application are:
    - Captured image storage.
    - Detected text storage.
    - Pushing notifications to the smart glass device.
    - Calls transfer to device

Also, this project aims to provide '*cost-effective*' smart glasses for students and corporate employees keeping in mind the financial constraints of a common resident of Pakistan. This will not only introduce a pair of smart glasses accessible to the public but is also equipped with an array of functions to aid the user. It would introduce smart features like Bluetooth audio connectivity for voice calls as well as a mobile application for a customized user experience.

1.5 Organization

The first section of the thesis contains the introduction and literature review. The chapter on introduction includes the problem statement and the proposed solution. Literary review gives readers a survey on the research conducted and the hardware equipment that is used for developing

the project. The second section contains the third and fourth chapters. The third chapter focuses on the communication between the connected modules and Raspberry Pi and “Image Detection Mechanisms”. The fourth chapter encompasses the mobile phone side of the project. The third and the last section includes our results, conclusions and future works.

CHAPTER 2

Literature Review

LITERATURE REVIEW

This section provides an overview of the research carried out by various scholars, companies and engineering students which helped us in getting a sense of direction as well as aided us in formulating a strategy to achieve our desired goals.

2.1 Scholarly Articles

Before choosing the equipment to be used, we consulted available literature related to our project. These research papers gave us a rough idea of the design approach that is most viable in terms of cost and efficiency. The base design model we opted is based on the following diagram:

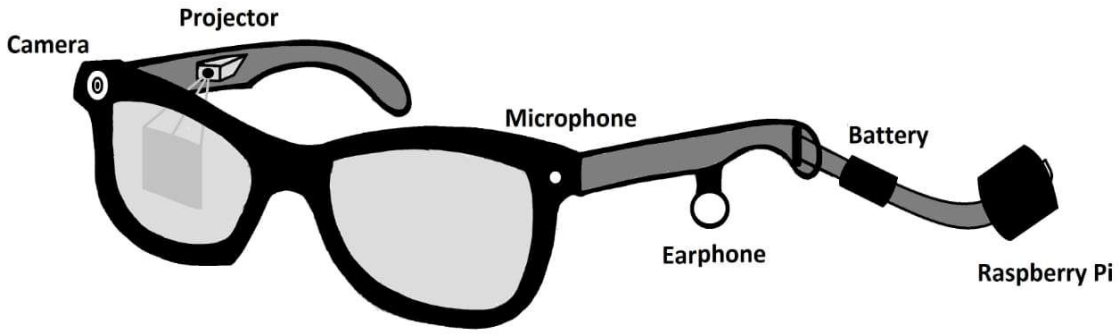

Figure 1 Project architecture

In the light of this model, information from literature is explained in the following subsections.

2.1.1 Circuit

The concept of multitasking and providing a hands-free solution is the fundamental part of our project. Choosing the appropriate sensors and modules to provide such features and a microcontroller that can compute the raw values efficiently was considered in this section. The components adopted in the circuit are as follows:

- **Camera –** This camera board module for Raspberry Pi is the best option for every Raspberry Pi enthusiast. It has 5 MP native resolution along with sensor capability of 2592 x 1944 pixels for static images, making it the best solution for a more accurate Optical Character Recognition feature [15].

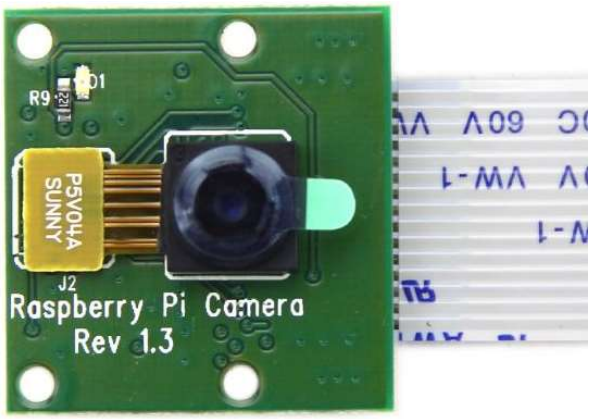

Figure 2 Camera Module V1

- **OLED Display –** The graphic OLED display module made of 128x64 individual white OLED pixels diagonally is only 0.96 inch in size. The controller IC SSD1306, communicates via I2C serial interface only. This device has a built-in voltage generation and only a single 3.3V power supply is needed. Because the display makes its own light, no backlight is required. [14]

Figure 3 OLED display

This reduces the power required to run the OLED and is why the display has such high contrast, extremely wide viewing angle and an extreme operating temperature between -40°C and 85°C. It's easily controlled by a Micro Controller Unit such as the one used in our project, the Raspberry Pi 4.

- **Raspberry Pi 4 Microcontroller –** The Pi 4 is a full computer packed onto a single board which is popularly known as a single board computer. The introduction of Pi 4 allowed one to embed an entire computer onto a tiny integrated circuit. We used the latest model of the "Pi" product line, the Raspberry Pi 4, which provides features like dual display output of up to 4K resolution. At the heart of the Pi 4 is a 1.5GHz BCM-2711 Quad-Core Processor, with the option of choosing between 2GB, 4GB or 8GB LPDDR4-3200 SDRAM. [13]

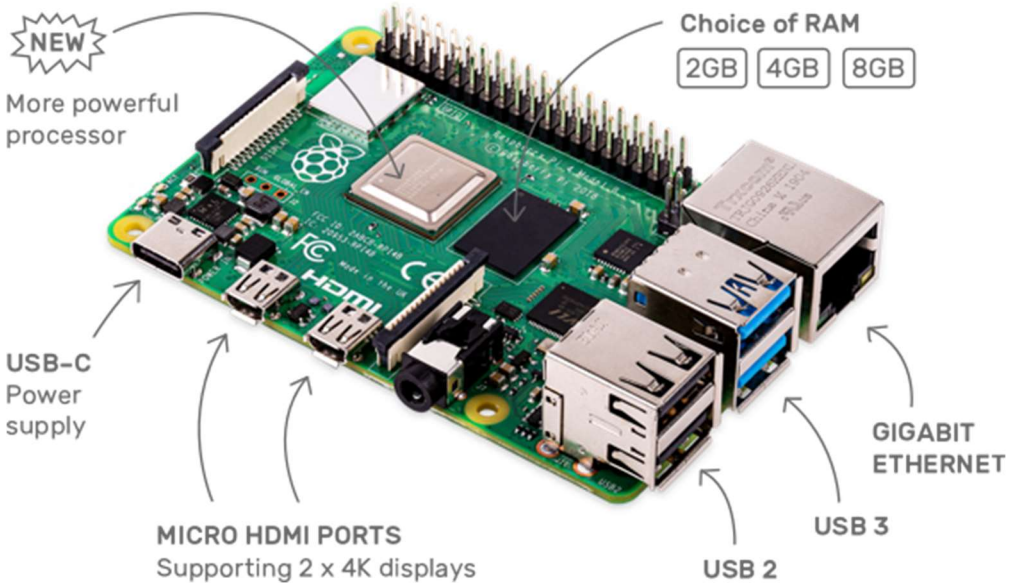

Figure 4 Raspberry pi 4 architecture

- **Lithium-Polymer (Li-Po) Battery –** This is a battery pouch used to power the smart glasses device. It is a standard piece of hardware with a voltage of 5.1V and a capacity of 105mAh.

Figure 5 Li-Po battery

- **3D Designed and Printed Frame –** A 3D-printed frame which houses the mechanism for near eye display is shown in the following figure. [18]

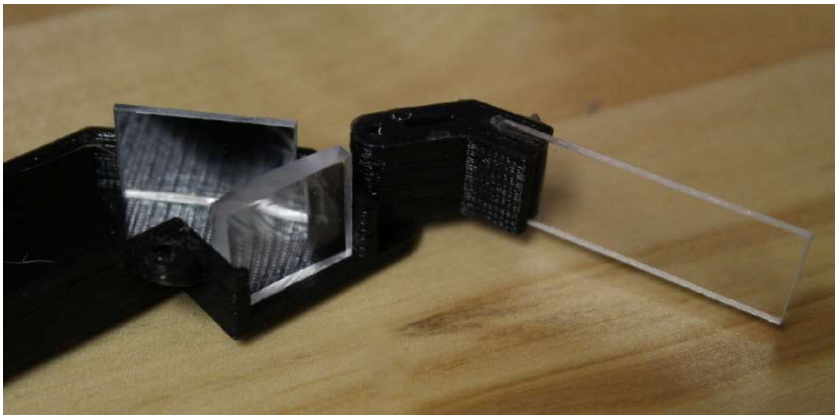

Figure 6 3D-Printed design

- **Bone conduction transducer –** A special transducer which converts sound waves to vibrations is used to be mounted on the glasses frame.

Figure 7 Bone conduction transducer

2.1.2 Interface

In this section we thoroughly researched about the approaches to communicate between modules, microcontroller, and android application.

- **Bluetooth Module –** This could be used in case of an independent microcontroller with no means of communicating with the android application, so we can keep the costs as low as possible. We opted to go with a microcontroller with the inbuilt Bluetooth capability as programming an analog microcontroller and coding Bluetooth protocol at our level was not too complex to handle.

2.1.3 Software and Programing

This section covers the software part of the project and the platforms used to code the project. The detailed information is as follows:

- **Python3** – Keeping in view all the aspects of our project, novelty in the project was using Optical Character Recognition (OCR). For this purpose, we are required to feed a code in raspberry pi OS. All the code is written in python and the IDE used is ‘Thonny’.

- **Terminal –** To stream Bluetooth audio from mobile to Raspberry Pi and vice versa, notification mirroring from mobile phone and installing required packages for other features; Shell Scripting (shebang) is required. The commands are passed in the Raspbian terminal to obtain desired output. Shell scripts are executed by interpreters of Linux kernel.

- **Android studio For App** – Android Studio is used to develop the application for the glasses. The software contains all the functions to completely design the layout and code an application. The language in which the applications are programmed is JAVA.

- **Raspbian –** Raspbian is the Linux based Operating System required to use a Raspberry pi computer.

2.2 Market Research

A survey of the market revealed that some companies have been working rigorously on developing smart glasses with specific functions out of which most are just for entertainment purposes. Once the stuff of science fiction, smart glasses are now a very real product bringing built in capabilities like bone conduction earphones and microphone allowing the user to make and receive calls, audio multitasking, cameras, a head up display etc. Google being the most prominent of all, has developed their own smart glasses product, the 'Google Glass', with basic capabilities focused more on entertainment rather than being productivity focused. [1]

2.3 Components

After a month of research which included understanding concepts used by several individuals (students and professionals) in IEEE research papers for developing smart wearable platforms,

online research, and consulting with our professors, we decided to use the following equipment for designing our own smart glasses platform.

- Raspberry Pi 4 Microcontroller
- Camera Module
- Micro Projector
- Audio Transducer

2.3.1 Raspberry Pi 4

The Raspberry Pi is a low cost, credit-card sized computer that can get plugged into any display device and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing and to learn and apply new languages like Python. It is capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. [16]

The Raspberry Pi 4 has the ability to interact with the outside world and has been used in a wide array of digital projects. [2]

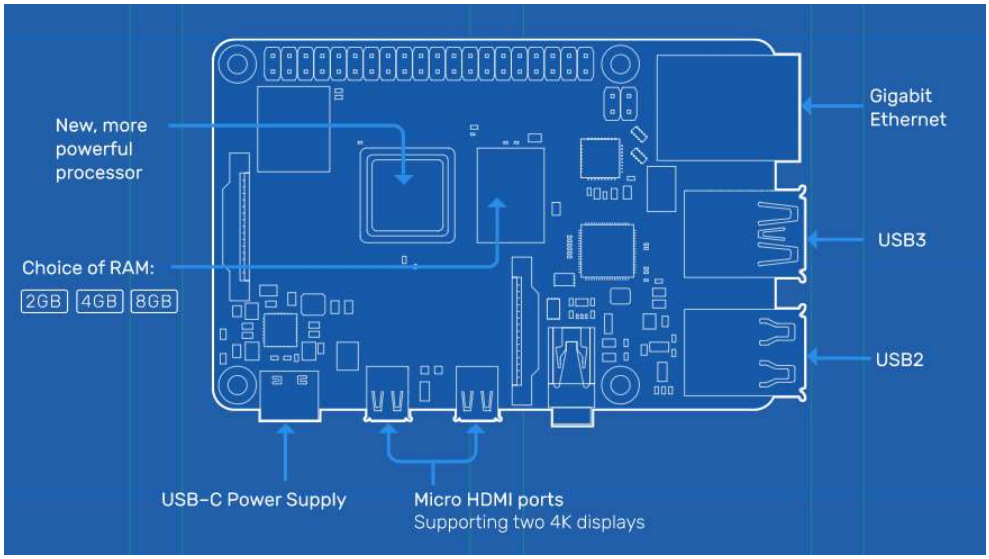

Figure 8 Raspberry Pi 4

The following equipment is necessary to use a Pi.

- Micro SD card (for OS)
- Monitor
- Keyboard
- Mouse
- Power Source

2.3.1.1 Hardware overview of Raspberry Pi 4

Raspberry Pi 4 has the following features:

- Micro – HDMI ports
- USB 2.0 and 3.0 ports
- Gigabit Ethernet port
- General Purpose Input Output Ports

2.3.1.2 Operating System of Raspberry Pi 4

The Raspberry Pi Foundation has developed Raspbian, a "Debian-based" Linux [13] distribution for download, as well as third-party "ubuntu", Microsoft Windows-10 IOT core, RISC Operating System and specialized Media Centered distributions. It promotes Python and Scratch as the primary programming languages alongside the support of many other languages. The default firm-ware is "Closed Source", whereas an unofficial "Open Source" is also available.

2.3.1.3 Pin Configuration

The pin configuration of the Raspberry pi 4 is shown in the following figure [17].

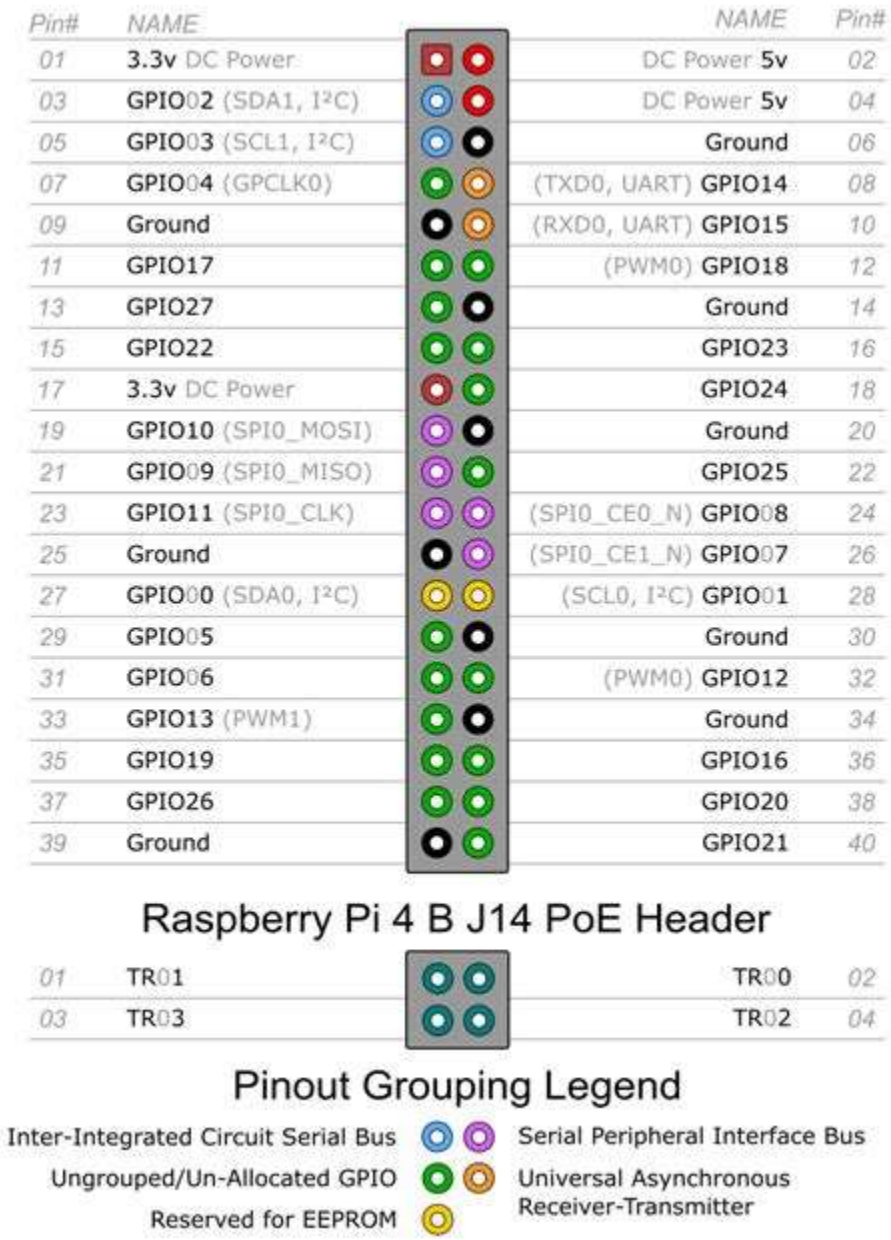

Figure 9 Pin Configuration Raspberry Pi 4

2.3.2 Raspberry Pi Camera Module V1

Below is the Camera module v1.3 interfaced with the raspberry pi via the ribbon connecting cable.

Figure 10 Raspberry Pi Camera Module V1

2.3.2.1 Raspberry Pi Camera Module V1 features

- 5MP sensor
- Wider image (capable of 2592x1944 stills, 1080p30fps)
- 1080p video supported.
- Camera Serial Interface (CSI)

- Compact Size
- Lightweight

2.3.2.2 Raspberry Pi Camera Module V1 working

This is a simple plug and play device which does not require an extensive setup to use. Just connect it to the Camera Serial Interface (CSI) port on the Raspberry Pi via ribbon cable and then boot the latest version of Raspbian. [15]

2.3.2.3 Pin Configuration

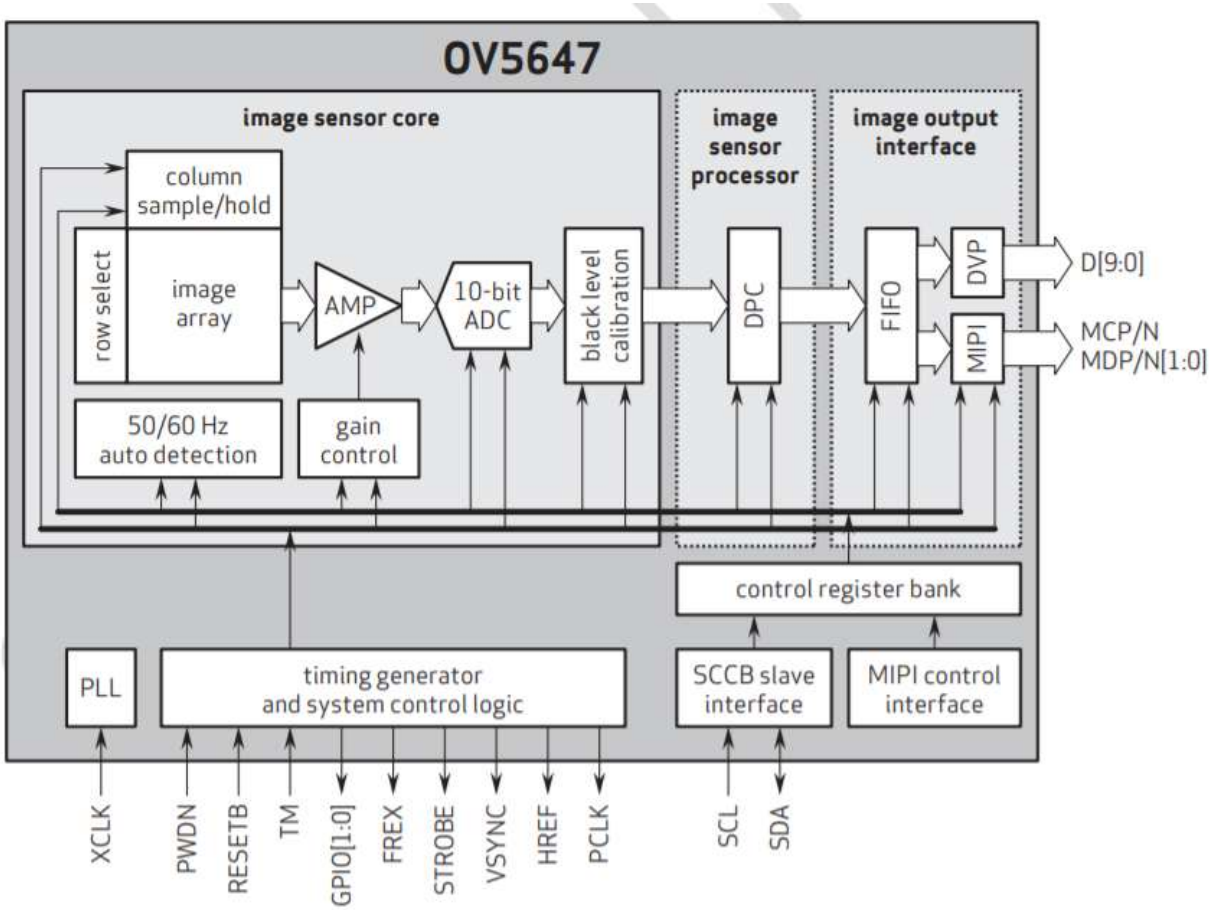

Figure 11 Camera Pin configuration

CHAPTER 3

Interfacing and Working

INTERFACING AND WORKING

This section describes how the Raspberry Pi Camera Module V1.3, and OLED Display modules are communicating with the Raspberry Pi 4 microcontroller.

3.1 Communication between Raspberry Pi Camera Module V1.3 with Raspberry Pi

The camera module of v1.3 compatible to our specific Raspberry pi is used for our two major features and it needs to be handled with care. There is a little CMOS camera present over module and the flat ribbon cable attached to the camera module interfaces with the Raspberry pi. A zip 15 CSI connector in the raspberry pi is used to connect the camera module. The connector will be pushed down hence we grab on to two sides of this connector gently, stretch it upwards; which will create a cavity from where we insert the camera module to Pi; then press the connector to fit it inside the Pi. The power supply is connected after the camera module has been successfully interfaced because it can damage the module.

After successfully installing an operating system which in our case is Raspbian; the camera interface from the raspberry pi's configuration options must be enabled. The system requires a reboot after enabling the interface to save the changes.

3.2 Optical Character Recognition (OCR)

OCR is coded in python language, and Tesseract library is used to perform the operation [4]. The Optical Character Recognition feature's functionality is explained in a few steps.

- When an image is captured; the code first does Adaptive Thresholding which changes the image into a binary image thus outputting a black and white image [10].

- The image is fed into the next function that is the connected component analysis. It performs the base line detection i.e., detecting the start and end of a line.
- Then in a line with the help of spacing, it checks and detects the word, copy the outline of the letters of the word and compare it with its dictionary.
- Thus, it detects each line one by one and move on to the next thus completely analyzing all the words in the photo captured by the camera.

3.3 Special Near-eye display

The near point of a human eye i.e. the shortest object distance that a human eye can accommodate to form a clear image on retina is 25cm. Thus a simple transparent screen for displaying the contents is not an option.

In all types of smart glasses, special near to the eye displays are designed which form a virtual image in user's field of vision, typically an arm's length away giving an augmented reality experience. This is the key requirement for any type of head-mounted display or Augmented reality wearable computers.

- The design adopted for Vision uses a small OLED display, a mirror, a Plano-convex lens and a reflector enclosed in a specially designed 3D-printed frame.
- An image formed on the OLED is reflected from the mirror towards the Plano-convex lens having a focal length of 100mm. Thus the OLED distance between the OLED and lens is almost 7mm.
- The virtual image formed is then reflected towards the eye because of the transparent reflector and appears an arm's length away from the eye [18].

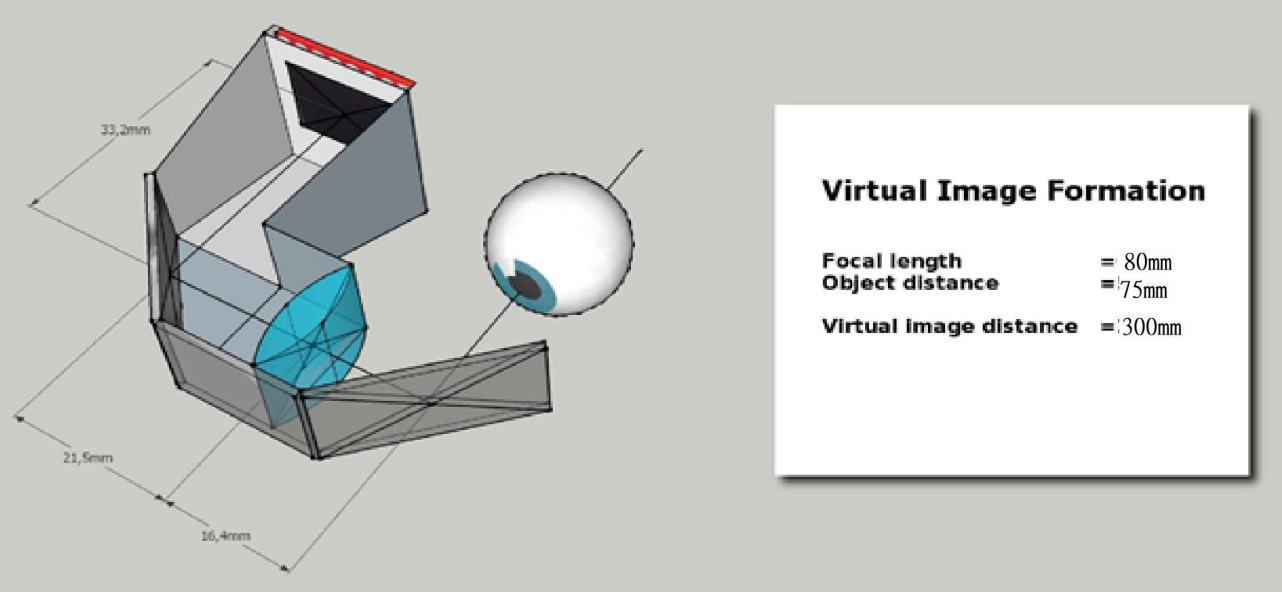

Figure 12 Near-eye display

3.4 Interfacing and overall working

Camera, Near-eye display, bone conduction audio transducer, microphone and battery are connected to the Raspberry pi directly mounted on the 3D-printed glasses frame. The whole setup forms Vision and gets connected to the phone having the specific app wirelessly via Bluetooth. The app contains the options for connectivity with the glasses, opening captured data from Vision and is the main source behind the notification mirroring feature of the glasses. The app sends all the incoming notifications to Vision which then displays them in user's field of vision.

3.5 Android Application:

Android applications are developed in the Java language using the Android Software Development Kit (Android studio in our case). Once they are developed, Android applications can be uploaded on Google Play Store or can be transferred through APKS.

Android powers millions of smart phones around the globe. It is the largest installed operating system on any smart phone, and it is growing exponentially.

The following figure shows the **Android Architecture** [12]**.**

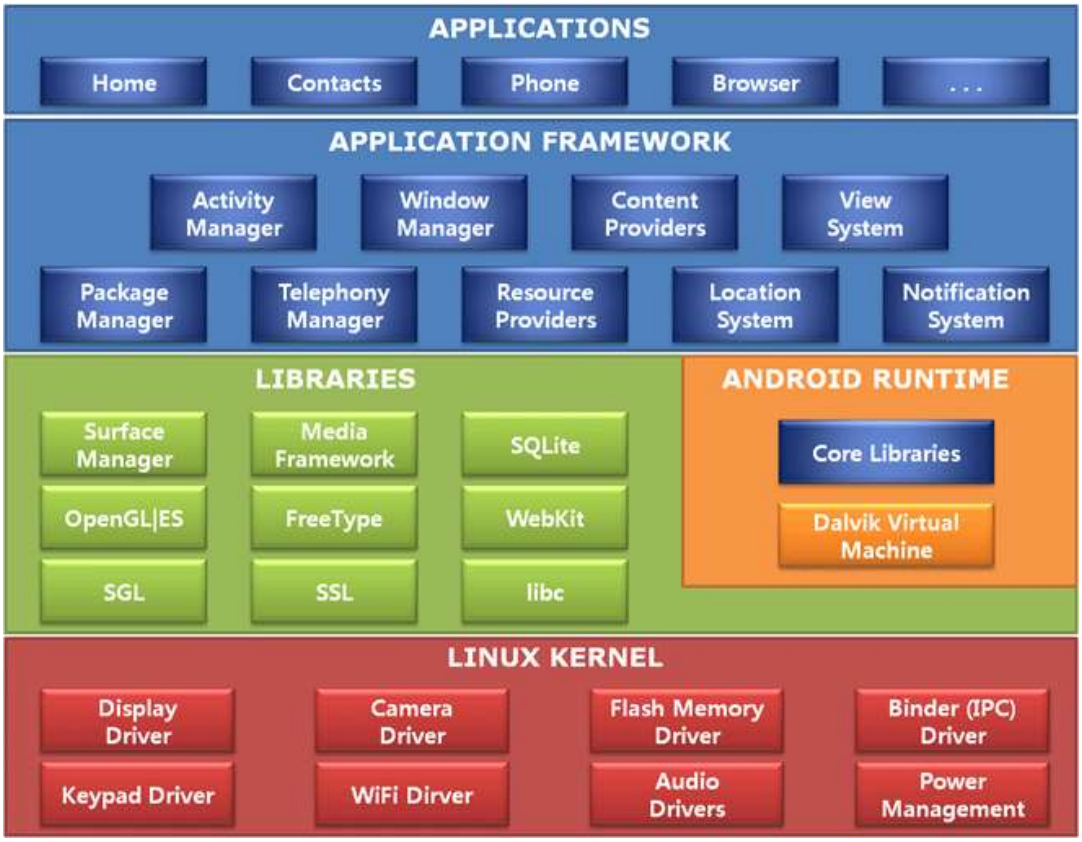

Figure 13 Android Architecture

3.5.1 Linux Kernel:

The first or the bottom layer is Linux kernel. This provides a level of abstraction between the device hardware and software further controlling all the essential hardware drivers. The kernel handles all the things such as networking and controlling the drivers.

3.5.2 Libraries:

On top of Linux kernel there is a combination of open-source libraries that help automate the OS's tasks and processes. These include libraries like SQLite, SSL, audio and video, networking libraries etc.

3.5.3 Android Core Libraries:

These are java-based libraries offered to the developer to use them and design custom applications. These include but are not limited to:

- **android.App** : It Provides accessibility to the application model
- **android.webkit** : It is Related to web browsing in applications
- **android.Content** : It Provides content access.
- **android.database** : SQLite management classes.
- **android.OS** : It Provides accessibility to standard OS services
- **android.text** : Render and manipulate display.
- **android.view** : Application UI.
- **android.widget** : Buttons, list views, labels, layout managers etc.
- **android.opengl** : Graphics rendering API.

3.5.4 Android Runtime:

This is present on the second layer of the architecture. This layer contains a major framework called **Dalvik Virtual Machine.** This is a JVM. It is specifically designed for Android. It uses Linux core features like multi-threading and memory-management. Every Android application runs in its own instance and this is provided by the Dalvik VM.

3.5.5 Application-Framework:

The Application-Framework is the third layer and provides many higher level services to apps. Developers make use of these services in their apps.

The Android framework provides the following features:

- **Activity Manager**: It Controls the activity stack and apps lifecycle.
- **View System:** Create application UI (user interface).
- **Notifications Manager:** Display notifications and alerts to the user.
- **Resource Manager:** Provides access to resources such as color, strings, UI layouts and settings.

3.5.6 Applications:

All the Android applications are present at the fourth layer. They use all the available resources from layers beneath and customizes them in a way to work together. Examples of such applications are Calculator, Browser, Games, Contacts etc.

3.6 VISION APP

The application made for the project which acts as an interface between the Mobile phone and vision device is Vision which is responsible for connecting with the phone, notifications and calls transfer to the phone and receiving and storing the OCR data and captured images from vision device.

3.6.1 Permissions/Manifest File

The core libraries needed for the functioning of the VISION application are included in the manifest file. The XML file contains the necessary permissions required by the application for its proper functioning. This file must be presented to the operating system prior to running the application.

The main entries in our manifest file are shown below

< android.permission.ACCESS_INTERNAL_STORAGE/>

< android.permission.ACCESS_MICROPHONE/>

< android.permission.ACCESS_AUDIO/>

< android.permission.ACCESS_NOTIFICATIONS/>

< android.permission.INTERNET />

These five entries show that the VISION app can and will perform functions related to Call Audio, SMS, Internet, Storage and Notifications.

3.2.1 Activities

Unlike other programming languages, the android system invokes functions by calling out activities rather than calling them through the main() method. This is a very useful feature and allows activities to be arranged in a specific order to be executed.

The VISION application comprises of 3 main activities namely:

[1] Connect to Vision

[2] Text Storage

[3] Photos

CHAPTER 4

Results

RESULTS

This sections explains the results obtained after working on the specified features of the Vision smart glasses.

4.1 Optical Character Recognition:

The OCR feature is triggered by a button mounted on the glasses. On enabling the feature, camera takes a picture, processes it, extracts the text data, saved the data in form of an editable text document and transfers it to the mobile phone via Bluetooth. The results obtained from the Optical character recognition are as follows:

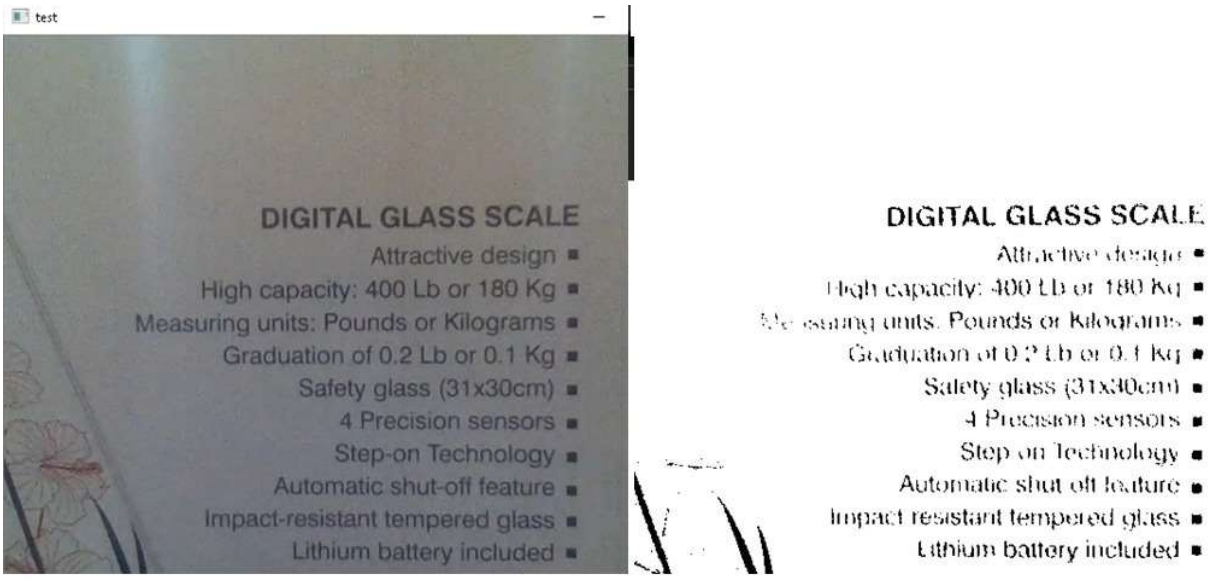

Figure 14 Thresholding picture to extract text

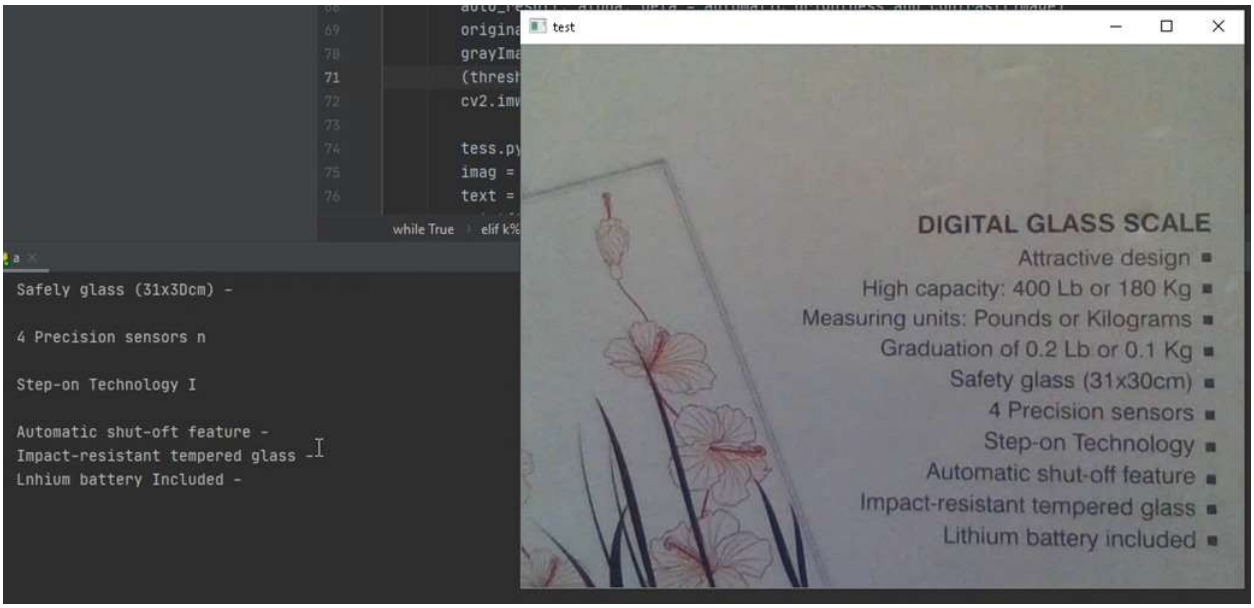

Figure 15 Extracted text output (a)

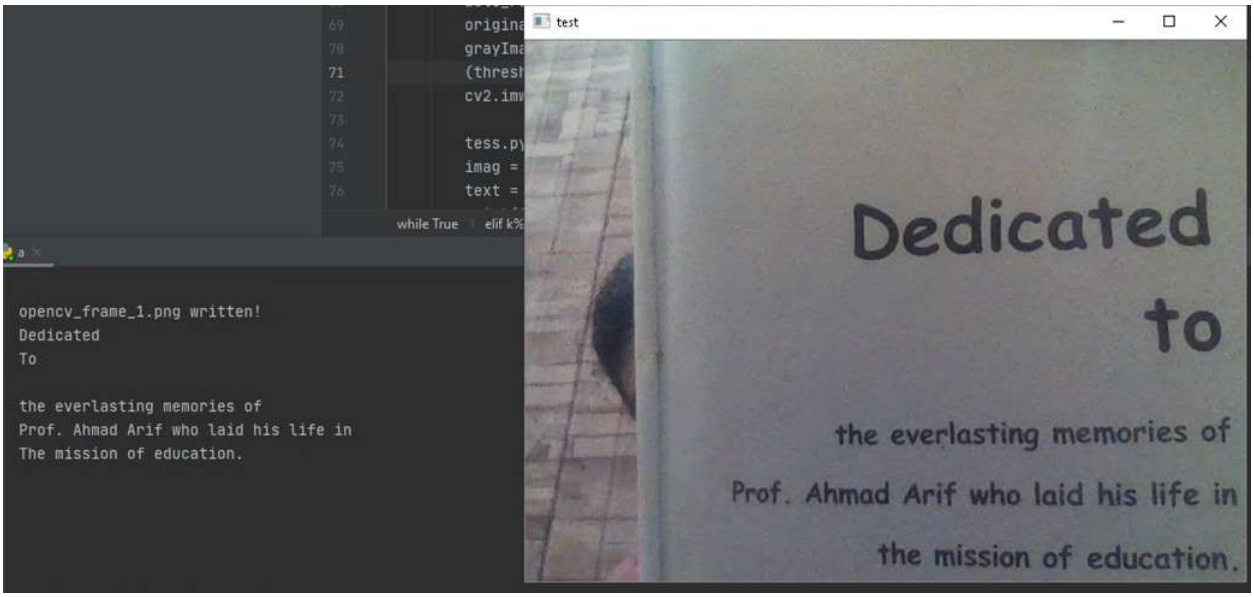

Figure 16 Extracted text output (b)

4.2 Near eye display:

The special near eye displays is used for displaying incoming calls and notifications in user's field of vision. The Vision app captures these notifications and transfer them to the raspberry pi via a web socket. The pi captures those notifications and displays them on the OLED display via I2C interface. The displayed text floats at an arm's length away from user's eye and can only be seen by the wearer of the glasses.

Figure 17 Notification display captured by a normal camera

4.3 Calls transfer:

All the incoming calls on the connected mobile phone are routed to the raspberry pi for audio input and output. A microphone mounted is used for the input and the bone conduction audio transducer is used for output. The figure 18 shows that the audio of the phone is being routed to the Raspberry pi during a call. No input/output is thus taken from the mobile phone as long as the phone is connected with the vision device. A button is used to pick up any incoming call to the phone

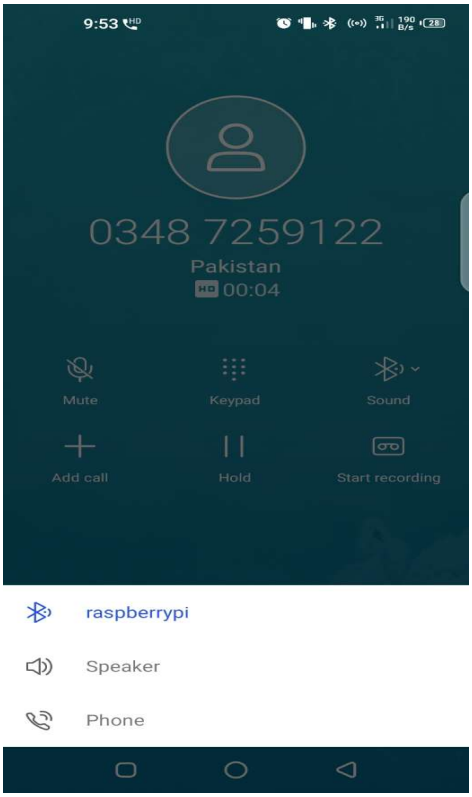

Figure 18 Calls transfer

CHAPTER 5

Conclusion

CONCLUSION

5.1 Overview & Objectives Achieved

The project provides the capability to display text and call notifications by modifying a 3D-printed glasses frame, captures photos and converts text in printed form to a user editable format. This is done by using the camera module in conjunction with the near eye display and a microcontroller which intimates the android application via Bluetooth. It was achieved with design parameters of being as follows:

a. Economical

b. Efficient

c. Scalability

The android application stores images and the detected text in the phone storage via the Bluetooth connection.

5.2 Applications

VISION provides a hands-free glasses solution not only applicable to academic environments specifically but also implementable across a variety of users. The scalability of this project has diverse prospects for the future [20]. This makes a project applicable to a multitude of clientele.

5.2.1 General Public

- A cost-effective solution is the fundamental design parameter that is followed throughout the design of this project.

- Over 4 billion adults in the world wear glasses. This statistic is derived from the Vision Council of America's claim that about 75 percent of the adult population worldwide uses vision correction products, and 64 percent of them wear glasses [22]. Therefore, this project comprises of minimal hardware components and is based on the ideology of improving software to provide a better user experience.
- To further facilitate such people, a smart glasses solution can be introduced through retail store platforms and bulk importers and suppliers. This would make it more affordable and a viable smart solution for the masses.
- Much care was taken about making mass deployment possible by keeping the android application as simple as possible requiring no training as only user's Google account details need to be added.

5.2.2 Corporate Clientele

- There are many corporate offices and businesses that constantly keep people 'on the go', which means they may be driving, or they are in an office environment giving a presentation where they may not be able to check their phones. This product not only aids them but also provides a multitasking platform as not a similar cost-effective solution is available in the market.
- The project falls within United Nation's World Intellectual Property Organization (WIPO) – *Public Health & Well Being.* This aligns our project with global corporate and institutional goals of human wellbeing that makes it an attractive solution to implement. This enables introducing our product to a wide spectrum of corporate clientele that have responsible ethical values.

- The interesting aspect of this project is that it is not a piece of hardware only specific to indoor activities but a platform of smart solution that can be introduced in the fast-paced developing electronics industry. The devices can be installed on any glasses frame design. Therefore, we can sell our algorithm and methodology via patents to wearable smart products' industries, so they can develop a product specific to their needs.

5.2.3 Educational Institutions and Departments

- The teachers and students can be provided with this product. It not only ensures the capability of conversion of text from the notes on their notebooks but also reduce their time, efforts, and disturbances in such environments [8].
- The existing smart glasses solutions available in the market have a starting price of at least USD 1000 which have specific functions which range from a basic calling feature to the detection of brainwaves. Our solution can be modified with added features as per their demand and can be a domestically designed solution that is substantially cheaper.

5.3 Future Work

Certain aspects of the project can be polished to make it more user friendly as well as improve it overall as we have so far worked on a prototype. If time is given and work is done on it, a marketable product can be brought to consumers and clients.

5.3.1 Communication

- Bluetooth Module is currently being used for communication between Raspberry Pi 4 and android application. This does not give room for adding a lot of features on top of the storage and calling functions.

- Therefore, Bluetooth Low Energy (BLE) enabled socket interface, or a Wi-Fi hotspot can be introduced. This provides room to implement features like personal assistant used in automatically sending texts/calls on command or navigation.
- Benefit of this is that it can be fully implemented on software and requires little to no hardware addition. Therefore, costs do not increase per unit; rather one-time development cost to fully implement this communication mode.
- This would enable more energy-efficient and beneficial to the end user in terms that it would enable the developer to add more features via software development.
- Battery conservation can be made possible by using software optimization and addition of battery saving modes which may limit functionality to provide only the basic necessary functions. This improves usage time in between charges as well as the over-all user experience.

5.3.2 Additional Features

- Features like video recording, E-sim, Artificial Intelligence (AI), Personal Assistant (such as Siri, Google Voice, Bixby, Cortana etc.) and various sensors for further enhancing productivity can be added.
- Therefore, the motivation to improve the glasses by the addition of such features would be to cater for specific clientele like people who can afford an expensive, feature intensive and government/corporate clients that would want specific features to be introduced in their variant of smart glasses.
- This will enable our product to target a broader clientele not only limited to the base model but manufacturing variants specific to needs of the consumer.

5.3.3 Product Development

- So far, we have worked on a prototype. This uses raspberry pi which is an expensive microcontroller and whose processing power is not fully utilized. Therefore, it is neither feasible nor smart to use it on a large scale.
- The prototype also costs a lot because market products like battery pack, glasses frame, audio transducer and lens were used. If this project is to be sold as a marketable product, product design tailored to a custom glasses frame with room for circuitry and other equipment needs to be bought in bulk. This will reduce product development costs by approximately 50%.
- To achieve that cost reduction of 50%, a custom microcontroller needs to be used with the sole purpose of interfacing and communication with android application. As raspberry pi costs PKR 15,000, this custom microcontroller will drastically reduce manufacturing costs. But ample research needs to be put in to make it possible.
- Similarly, as a part of market research, it was revealed that glasses manufacturing in Pakistan is not viable as product quality is not up to international safety standards at all. It was also revealed that only viable option is to directly contact Chinese glasses frame manufacturers to design a frame according to our design specifications and assemble it with the required devices. This removes the need for having an independent Pakistan based plant as it is not financially viable against a Chinese alternative. Outsourcing is the future!

BIBLIOGRAPHY

[1] Rinna Joy. “Google Glass A Wearable Computer Model (NSRCL-2015)”
[Accessed October 29,2020]

[2] Nikhil Paonikar, Samuel Kumar, Manoj Bramhe. “Raspberry Pi Augmentation: A cost effective solution to Google Glass”
[Accessed October 25].

[3] Alejandro Riquelme Cencerrado. “Smart Glasses, a new way to interact within the world”
Available: https://ddd.uab.cat/record/203356
[Accessed September 5, 2020]

[4] Assistant Professor Gnana Prakash & Anusha. “Text Extraction from Image using Python (Volume-1).”
[Accessed October 8, 2020].

[5] Chowdhury Md Mizan, Tridib Chakraborty and Suparna Karmakar. “Text Recognition using Image Processing (Volume-8, 2017)”
[Accessed September 12, 2020].

[6] Akhil s Nair. “An Overview of Tesseract OCR Engine (Dec 2016)”
Available: https://www.researchgate.net/publication/315834331
[Accessed October 12, 2020]

[7] Caroline El Fiorenza, Sandeep Kumar Barik, Ankit Prajapati, Sagar Mahesh, “Hand Gesture Recognition using Convexity Defect (2019)”
[Accessed September 15, 2020]

[8] Ales Berger, Filip Malay, “Smart Google Glasses Used As Education Support Tool” [Accessed October 15, 2020].

[9] Max,streaming Bluetooth audio from mobile phone to raspberry pi using ALSA (2018),

Available: https://scribles.net/streaming-bluetooth-audio-from-phone-to-raspberry-pi-using-alsa/

[Accessed October 25, 2020]

[10] October 30, 2019, “Learn how you can use Tesseract and OpenCV to extract text from images on PDFs and more with the Raspberry Pi Camera” (2019), [image].

Available: https://maker.pro/raspberry-pi/tutorial/optical-character-recognizer-using-raspberry-pi-with-opencv-and-tesseract

[Accessed September 16, 2020]

[11] Ruben Sanchez, “Send Audio From Mobile Phone to Raspberry Pi”, (2019).

Available http://rubensm.com/send-audio-from-mobile-phone-android-ios-to-raspberry/

[Accessed July 23, 2020].

[12] Android Architecture, (2019), [image].

Available: https://www.dev2qa.com/android-architecture-components-introduction/

[13] Raspberry Pi 4 Model B, (2019), [image].

Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

[14] 0.96 OLED display, [image].

Available: https://digipak.org/product/precision-c-temperature-sensor-lm35dz__sku-314

[15] Camera module V.13, [image].

Available: https://proto-pic.co.uk/product/raspberry-pi-camera-module-v1-3/

[16] Shanto Shaji, “Raspberry pi 4 specifications and features”, (2019).
Available: https://www.tomsonelectronics.com/blogs/news/raspberry-pi-4-specification-and-features

[17] Raspberry pi 4 Model B GPIO pinout diagram, (2019), [image].
Available: https://www.element14.com/community/docs/DOC-92640/l/raspberry-pi-4-model-b-default-gpio-pinout-with-poe-header

[18] Alain Mauer, “Virtual Image formation”, (2016), [image].
Available:https://hackaday.io/project/12211/gallery#48bee2d9479f853429a87a7a40a31bc2

[19] Linda Malecaj, “Best ways that AR smart glasses are guaranteeing business’ success in 2020”, (2020)
Available: https://www.vsight.io/best-ways-that-ar-smart-glasses-are-guaranteeing-business-success-in-2020/

[20] Jay MacDonald, “Smart glasses: How they work and what’s next”, (2020).
Available: https://www.allaboutvision.com/eyeglasses/smart-glasses/

[21] Andre Bourque, “Smart glasses are making workers more productive”, (2017).
Available: https://www.cio.com/article/3196294/smart-glasses-are-making-workers-more-productive.html

[22] Staff Writer, “How many people in the world wear glasses”, (2020).
Available: https://www.reference.com/world-view/many-people-world-wear-glasses-e1268cfa00bdbd41

APPENDIX A

- **Node.js code for notification display**

```
const WebSocketClient = require('websocket').client;
const i2c = require('i2c-bus');
const font = require('oled-font-3x5');
const font2 = require('oled-font-5x7');

const token = 'o.742sjKwZjxrf3uOjQ7orrzfAtEbdjcnx' // Replace with your token
const url = 'wss://stream.pushbullet.com/websocket/' + token
const client = new WebSocketClient();
const i2cBus = i2c.openSync(1);
const Oled = require('oled-i2c-bus');
const oled = new Oled(i2cBus, {
    width: 128, // if you have a lower resolution display, replace these
    height: 64,
    address: 0x3c
});

var isIdle = true;

clearScreen();

client.on('connectFailed', function(error) {
    console.log('Error connecting: ' + error.toString());
});
```

```
client.on('connect', function(connection) {
    console.log('Connected!');

    connection.on('error', function(error) {
        console.log('Connection error: ' + error.toString());
    });

    connection.on('close', function() {
        console.log('Connection closed');
    });

    connection.on('message', function(raw) {
        var message = JSON.parse(raw.utf8Data)
        if (message.type === 'push') {
            console.log('Update:');
            var push = message.push;
            if (push.type === 'mirror') {
                console.log('Title: ' + push.title);
                console.log('Body: ' + push.body);
                isIdle = false;
                oled.clearDisplay();
                var toDisplay = '';
                if (!(push.title === undefined)) {
                    toDisplay = toDisplay + push.title;
```

```
            oled.setCursor(1, 40);
            oled.writeString(font, 2, push.title, 1, true);
          }
          if (!(push.body === undefined)) {
            toDisplay = toDisplay + push.body;
            oled.writeString(font, 2, '\n', 1, true);
            oled.writeString(font, 1, push.body, 1, true);
          }
          oled.startScroll('left', 0, 15);
          var readtime = ((toDisplay.length) / 17) * 2000
          setTimeout(clearScreen, readtime);
        } else if (push.type == 'dismissal') {
          console.log('Notification ' + push.notification_id + ' dismissed.');
        } else {
          console.log('Unhandled type: ' + push.type);
        }
      } else if (message.type === 'nop') {
        console.log('listening...');
      }
    });
});

function clearScreen() {
    isIdle = true;
    oled.stopScroll();
```

```
    oled.clearDisplay();
}

client.connect(url);

function showTime() {
    if (isIdle) {
        var now = new Date();
        oled.setCursor(1, 50);
        var hours = now.getHours(),
            minutes = now.getMinutes(),
            ind = 'AM';
        if (hours > 12) {
            hours = hours - 12;
            ind = 'PM';
        }
        if (minutes < 10) {
            minutes = '0' + minutes;
        }
        oled.writeString(font2, 1, hours + ':' + minutes + ' ' + ind, 1, false);
    }
    setTimeout(showTime, 500);
}
showTime();
```

APPENDIX B

OCR Program Script (TESSERACT)

```
import cv2

import pytesseract

from picamera.array import PiRGBArray

from picamera import PiCamera


camera = PiCamera()

camera.resolution = (640, 480)

camera.framerate = 30


rawCapture = PiRGBArray(camera, size=(640, 480))


for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
        image = frame.array
        cv2.imshow("Frame", image)
        key = cv2.waitKey(1) & 0xFF


        rawCapture.truncate(0)




        if key == ord("s"):
                text = pytesseract.image_to_string(image)
                print(text)
```

```
                cv2.imshow("Frame", image)
                cv2.waitKey(0)
                break

cv2.destroyAllWindows()
```