College of Computer Engineering and Science (CCES)

Spring 2018/2019

Senior Design Project Report

Smart Glasses for Blind people

In partial fulfillment of the requirements for the
Degree of Bachelor of Science in Computer Engineering

Team Members

Name:	ID:
Hawra Al Said	201401207
Lina Alkhatib	201403245
Aqeela Aloraidh	201401820
Shoaa Alhaidar	201303866

Advisors:
Dr. Loay AlZubaid
Dr. Abul Bashar

Table of Contents

Acknowledgment

Team members would like to express our deep appreciation and gratitude to Prince Mohammad Bin Fahd University (PMU) and all those who support and provide us to complete this project. Special gratitude to our supervisors Dr. Loay Al Zubaidi and Dr. Abul Bashar for their contribution and help to coordinate our project especially in writing this report. Team members would also like to appreciate the crucial role of Dr. Nazeeruddin Mohammad, Dr. Majed Khan, Ms. Zeenat Al Kassim and Mr. Awadallah who helped us to assemble the project parts and gave us great suggestions about the project. In addition, the team members would like to extend thanks to Prince Mohammad bin Fahd University, College of Computer Engineering and Science for supporting us in this project by providing the components, appropriate labs, and classes. Finally, no words can describe our huge gratitude and appreciation to our parents and family for their tremendous encouragement and support throughout this challenging and great journey to complete this project.

Abstract:

These “Smart Glasses” are designed to help the blind people to read and translate the typed text which is written in the English language. These kinds of inventions consider a solution to motivate blind students to complete their education despite all their difficulties. Its main objective is to develop a new way of reading texts for blind people and facilitate their communication. The first task of the glasses is to scan any text image and convert it into audio text, so the person will listen to the audio through a headphone that’s connected to the glasses. The second task is to translate the whole text or some words of it by pressing a button that is connected to the glasses as well. The glasses used many technologies to perform its tasks which are OCR, (gTTS) and Google translation. Detecting the text in the image was done using the OpenCV and Optical Character Recognition technology (OCR) with Tesseract and Efficient and Accurate Scene Text Detector (EAST). In order to convert the text into speech, it used Text to Speech technology (gTTS). For translating the text, the glasses used Google translation API. The glasses are provided by Ultrasonic sensor which is used to measure the required distance between the user and the object that has an image to be able to take a clear picture. The picture will be taken when the user presses the button. Moreover, the motion sensor was used to introduce the user to the university’s halls, classes and labs locations using Radio-frequency identification (RFID) reader. All the computing and processing operations were done using the Raspberry Pi 3 B+ and Raspberry pi 3 B. For the result, the combination of using OCR with EAST detector provide really high accuracy which showed the ability of the glasses to recognize almost 99% of the text. However, the glasses have some drawbacks such as: supporting only the English language and the maximum distance of capturing the images is between 40-150 cm. As a future plan, it is possible to support many languages and enhance the design to make it smaller and more comfortable to wear.

1. Introduction:

In our lives, there are many people who are suffering from different diseases or handicap. According to NCBI (1986), 1.5% of the population in Saudi Arabia is blind and another 7.8% have vision difficulties. These people need some help to make their life easier and better. The main goal of “Smart Glasses” is to help blind people and people who have vision difficulties by introducing a new technology that makes them able to read the typed text. These glasses are provided with technology to scan any written text and convert it into audio text. Also, it can translate words from English to Arabic using Google API. The goal of “Smart Glasses” is helping those people in different life aspects. For example, these glasses effectively helpful in the education field. Blind people and people with vision difficulties can be able to read, study and learn everything from any printed text images. “Smart Glasses” encourage blind people or people with vision difficulties to learn and succeed in many different fields.

1.1. Project Background:

There are special schools and universities for people with special needs. There are different levels of needs and not all levels require special places and special schools. For instance, people with vision difficulties can study with normal students if they have an appropriate chance. Most blind people and people with vision difficulties did not study and that is because special schools for people with special needs not everywhere and most of them are private and expensive or they study at home acquiring basic knowledge from their parents.

Most blind people are smart people and can study if they have the chance to be able to study in normal schools because they are government school everywhere. Most people thought blind people and people with vision difficulties cannot live alone and they need help all the times. In fact, they do not need help all the times, they can depend on them self in most of the times and they have the chance to live like a normal person in this life. The main reason for implement “Smart Glasses” for blind people was to prove for all people that blind people and people with vision difficulties have the chance to live a normal life with normal people and study in any school or university without the need for help all the times. By “Smart Glasses”, the percentage of educated people will increase.

Most Schools will be able to accept people with vision difficulties instead of open special schools. Next year Prince Mohammad bin Fahd University (PMU) will accept blind people to study. The team would like to start using “Smart Glasses” for the first time in this chance and help students to improve their education level without help and be able to study as a normal person.

1.2. Literature Review:

Text detection and recognition have been a challenging issue in different computer vision fields. There are many research papers that have discussed different methods and algorithms for extracting the text from the images. the main purpose of this literature review is to view some of these methods and their effectiveness regarding their accuracy rates. In end-to-end text recognition with the power of neural network combined with the new unsupervised feature, learning growth took advantage of the known framework for the train to achieve high accuracy of the text and character detection and recognition modules. These two models have been combined using simple methods to build end to end text recognition system. The datasets that been used are ICDAR 2003 and SVT. The method of 62-way character classifier obtained 83.9% of accuracy for a cropped character from the first dataset [1]. In novel scene text recognition, which is an algorithm that mainly depended on machine learning methods. Two types of classifiers have been designed to achieve more accuracy, the first one was developed to generate candidates, but the second one was for filtering of candidates that are not text. a novel technique has been developed to take advantage of multi-channel information. two datasets have been used in this study, ICDAR 2005, ICDAR 2011. This method has achieved significant results in different evaluation protocols [2].

In photoOCR which is a system designed to detect and extract any text from any image using machine learning techniques, it also used different distributed language modeling. The goal of this system was to recognize any text from any challenging image such as poor quality or blurred images. This system has been used in different application such as Google Translate. The datasets that are been used for this system are ICDAR and SVT. the results showed that the processing time for text detection and recognition is around 600ms for one image [3]. For text recognition in natural scene images method, this method has proposed an accurate and robust method for detecting texts in natural scene images. This method has used an (MSER) algorithm to detect almost all characters

from any image. the datasets used for this system are ICDAR 2011 and Multilingual datasets. The results showed that the MSER has achieved 88.52% in character level recall [4].

End-to-end real-time text recognition and localization system have used ER (External Regions) detector that covered about 94.8% of the characters, and the processing time of an image with 800×600 resolution was 0.3s on a regular personal computer. The system used two datasets ICDAR 2011 and SVT. The average run time of the method on an 800×600 image was 0.3s on a standard PC. On the ICDAR 2011 dataset, the method achieved 64.7% of image-recall. for SVT, it achieved 32.9% of image-recall [5]. Text detection and localization using Oriented Stroke Detection is a method that took advantage of two important methods that are connected to a component with a sliding window. The character or the letter has been recognized as a region in the image that has some strokes in a particular direction and particular position. The dataset that has been used is ICDAR 2011, the experiment results showed 66% recall better than the previous methods [6].

EAST is an abbreviation of an Efficient and Accurate Scene Text Detector. This method is a simple and powerful pipeline that allows detecting a text in natural scenes, and it achieves high accuracy and efficiency. Three datasets have been used in this study, ICDAR 2015, COCO-Text and MSRA-TD500. The experiment has shown that this method has better results than previous methods regarding accuracy and efficiency [7].

2. Previous Work:

The idea of “Smart Glasses” is to create wearable computer glasses for a different purpose. These uses determine the type of glasses that are needing to be created. In the early stage, the “Smart Glasses” were simple and provide basic tasks which serve as a front-end display for the remote system. Now “Smart Glasses” become more efficient and provide several features and the below table compare some examples:

Product Name	Headset (The eSight 3)	Oton Glass	Aira
Created by	CNET's "Tech Enabled"	Japanese company (Keisuke Shimakage)	Suman Kanuganti
The Idea	1. This device has a camera with high-resolution to help people with low vision. 2. There is also a video image.	1. The smart glasses are designed to help dyslexic to read. 2. The camera will capture pictures of words that the user wants to read and reads the words for the user via the earpiece.	1. Aira uses smart glasses to scan the environment.
Benefits	1. No surgery needs. 2. Help some low vision conditions.	1. The glasses going to convert symbols into sounds 2. It looks like normal glasses. 3. The glasses currently feature a conversion of English and Japanese text into voice audio and translate multilingual text into Japanese and English voice audio.	1. Aira agents help users to interpret their surroundings by smart glasses.
Drawbacks	Do not improve visions.	Oton Glasses will only help people who have difficulty reading. Blind people will not get benefit from it.	The blind person should wait to be connected to the Aira agents in order to be able to discover things around.
Need to improve	The company wants to improve waterproofing versions.		

Table 1.1: Previous Work

Product name	Eyesynth-Smart glasses for the Blind	Google Glasses	Smart Glasses
Created by	Eyesynth (Spanish company), Marcelo Alegre (Designer)	Google Company	Shoaa, Hawra, Lina and Aqeela (CE students)
The Idea	1. Eyesynth is consist of special glasses with 3D cameras 2. These glasses make a 3D volumetric analysis of the scene and process the information, turning it into abstract sound, which provides nuances of position, size, and shape. 3. This audio signal is non-verbal, so it is a universal product in terms of language, so it can be used in any country. 4. The method that is used to transmit the sound is through cochlear headphones which improve the safety of the user.	• Google Glasses show information without using hands • Users can communicate with the Internet via normal language voice commands.	Help people who have vision difficulties especially blind people.
Benefits	1. allow blind people and those with limited sight to 'feel the space' through sounds 2. These glasses use an algorithm to convert spatial and visual information into audio.	1. Take a picture 2. Record a video 3. Get directions 4. Send messages 5. Phone calls 6. Google 7. Real- time Translation using word lens app	1. Help blind people. 2. Contains a sensor to warn the users if there is something in front of them or behind them. 3. Increase the education level for blind people. 4. Help to search for more information about any word in the image that the camera scans it.

Drawbacks	1. It is not affordable for everyone because it is expensive (575.78$) 2. It only recognizes the objects and directions.	1. It is not affordable for everyone because it is expensive (1,349.99$) 2. The glasses are not very helpful for blind people.	1. It supports only the English language. 2. Cannot use these Google Glasses while driving, as the picture, video or data will be in front of the eyes of the user, which can distract them 3. Raspberry Pi 3 is separated from the glasses which might be not comfortable for the users 4. It captures the object with a specific distance.
Need to improve			However, these limitations can be improved in the future, for example, the glasses can support different languages, it also can be smaller and easy to wear.

Table 1.2: Previous Work

3. Product Architecture and Components:

The idea of the project is to make glasses for blind people that help them in education life at university. So, based on the goal of the project and after making some searches team members decided to work with the following components:

3.1. Sensor:

3.1.1. Ultrasonic Sensors:

Figure 1: Ultrasonic Sensor

Description:

The purpose of ultrasonic sensors is to measure the distance using ultrasonic waves. Ultrasonic sensors emit the ultrasonic waves and receive back the reflected. So, by this time the ultrasonic sensor will measure the distance to the object. It can sense from 2-400 cm.

Function:

In “Smart Glasses”, the Ultrasonic sensor is used to measure the distance between the camera and an object to detect the text from the text image. The distance should be from 40 cm to 150 cm and that is because this is the required range to capture a clear image.

Distance L = 1/2 × T × C

L: The distance

T: Time between the emission and reception

C: Sonic speed

*The value is multiplied by 1/2 because T is the time for the go-and-return distance.

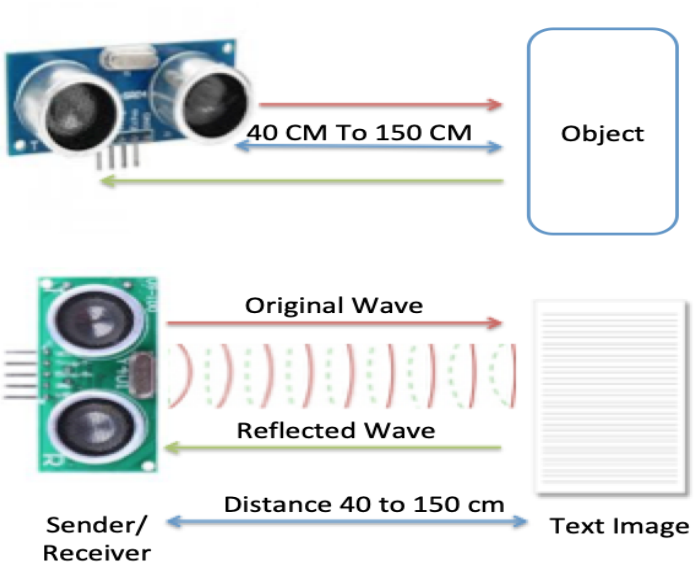


Figure 2: Measuring the destination of Ultrasonic Sensor

3.1.2. RFID Sensor

Figure 3: RFID Sensor

Description:

A radio frequency identification sensor consists of two main devices, which are RFID reader and the RFID tag. The RFID tag has digital data, integrated circuits, and a tiny antenna to send information to the RFID reader. Signals frequencies are usually between 125 to 134 kHz and 140 to 148.5 kHz for low frequencies and 850 to 950 MHz and 2.4 to 2.5 GHz for high frequencies.

Function:

The RFID reader is mainly used to collect information from the RFID tag with the help of electromagnetic fields. The process of transformation data from the tag to the reader is done by the radio waves. However, in order to achieve this process successfully the RFID tag and the RFID reader should be within a range at 3 cm. Any object can be identified quickly when it is scanned, and the RFID can recognize it. RFID has many applications such as passport, smart cards, and home applications. The RFID sensor is used in the project to attach the RFID reader in the hall and many classes so the blind person can recognize them.

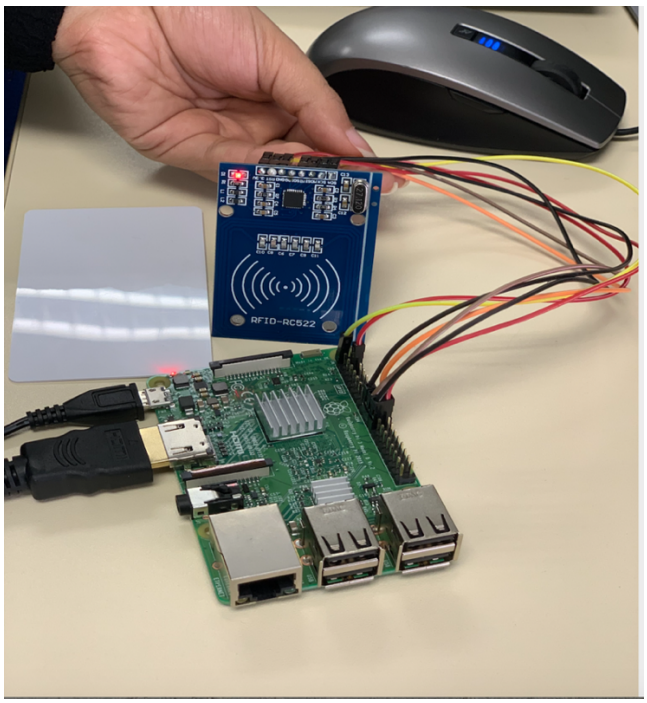


Figure 4: Connections of RFID Reader

3.2. Communication:

3.2.1. Webcam:

Figure 5: Webcam

Description:

The webcam has a view angle of 60°with a fixed focus. It can capture images with maximum resolutions of 1289 x 720 pixels. It is compatible with most operating systems platforms such as Linux, Windows, and MacOS. It has a USB port and a built-in mono mic.

Function:

In the project, the Webcam will be used the eyes of the person who wears the “Smart Glasses.” The camera is going to capture a picture when the button is pressed, in order to detect and recognize the text from the image.

3.2.2. Headphones:

Figure 6: Headphones

Description:

Wired headphone will be used in the project since Raspberry Pi 3 Model B & B+ come with Audio jack, it is better to take advantage of this feature rather than occupying one of the four USB ports that can be useful for other peripherals in the project.

Function:

The headphones will be used to help the user listen to the text that is been converted to audio after it is been captured by the camera or to listen to the translation of the text. The headphones are going to be small, light and connected to the glasses, so the user will not worry about losing the headphones or bothered by wearing them.

3.3. Controller:

3.3.1. Raspberry Pi Model B+ and B:

Figure 7: Raspberry Pi 3 Model B+

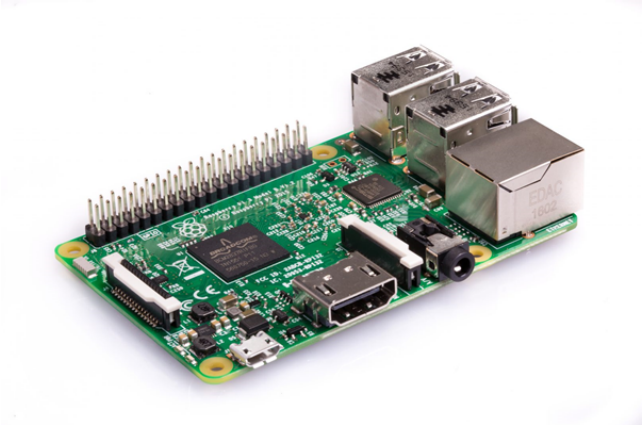


Figure 8: Raspberry Pi 3 Model B

Description:

Raspberry Pi is a credit card-sized computer. It needs to be connected with a keyboard, mouse, display, power supply, SD card and installed operating system. Raspberry Pi is a low-cost embedded system that can do a lot of significant tasks. It can be run as no-frills PC, a pocketable coding computer, a hub for homemade hardware and more. It includes GPOI (general purpose input/output) pins to control electronics components. It is also a great machine to attract children to learn more about how computers work and motivate them to improve their programming skills which help to create the next generation of developers.

Function:

Raspberry Pi 3 used for many purposes such as education, coding, and building hardware projects. It is the main component of the project. It used as a low-cost embedded system to control and connect all of the components together. It uses the Raspbian or NOOBs as the operating system which can accomplish many important tasks However, for the project the team decided to work on Raspbian as the operating system.

3.4. Battery:

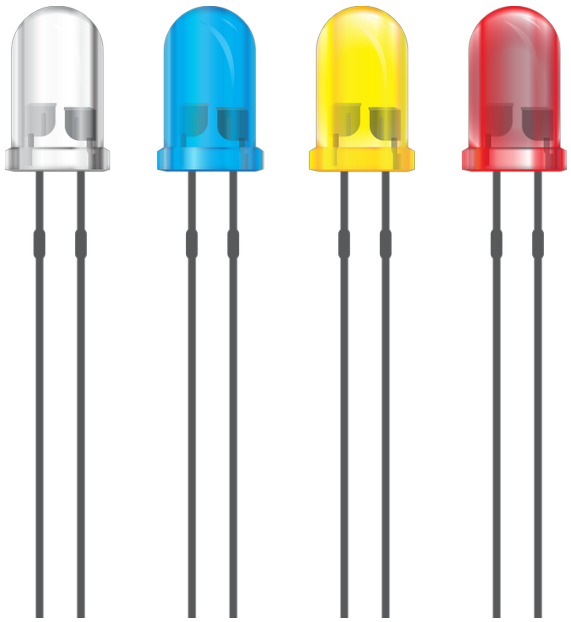
Figure 9: Power Bank Charger

In the beginning, “Smart Glasses” used Normal battery that provided with the raspberry pi 3 which is 5V and 2. A. Then the team thought that the blind students will use “Smart Glasses” all the time during university time and the way for the battery provided with raspberry not helpful because the student needs to move from one class to another. With the updated “Smart Glasses”, it uses Power Bank 5V and 2.5 A, and the student able to use the power everywhere in the university or school.

3.5. Other Components:

These are other component used to complete the project provided by the university or some owned by team members.

- LED



- SD Cards 32GB and 64GB

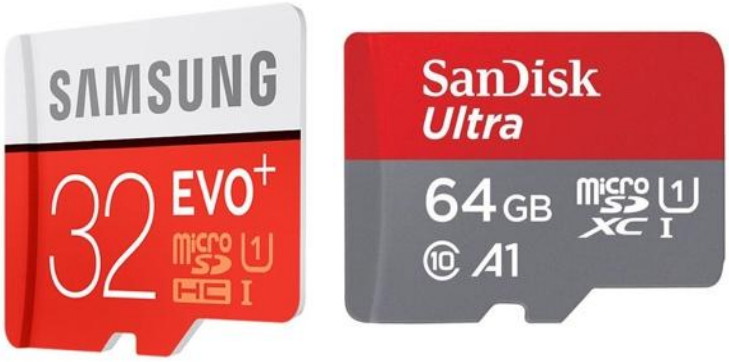


Figure 10: SD Cards

- 2 Buttons

Figure 11: Push Button

- Jumpers

Figure 12: Jumpers Wires

- Resistors

Figure 13: Resistor 1k Ω

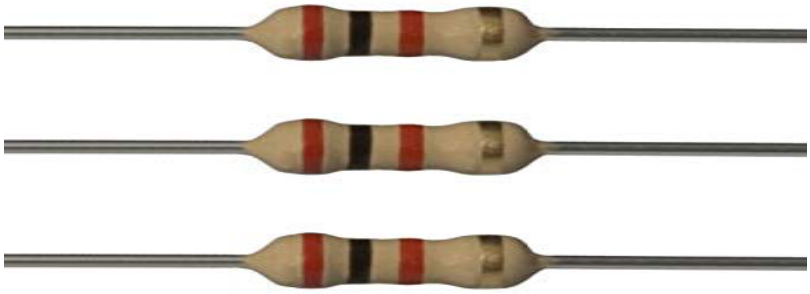


Figure 14: Resistor 2kΩ

- Breadboard

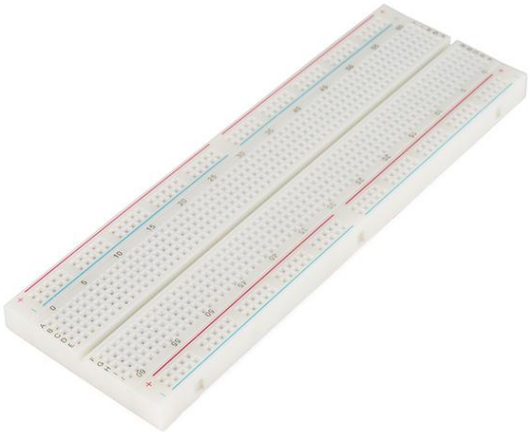


Figure 15: Breadboard

- Glasses

Figure 16: Glasses frame

4. Tasks Breaking Down and Plan Description:

4.1. Task Breaking Down:

All team members will work together, and each member of the team will be assigned to finish several tasks in order to accomplish this project as expected:

Team Member	Tasks	Start Date	End Date
Ms. Hawra Alsaid	• Select and order components	21/11/2018	1/4/2019
	• Coding using python	20/1/2019	28/4/2019
	• System testing and analysis	1/4/2019	14/4/2019
	• Collecting statistical data	12/2/2019	14/2/2019
	• Assembling hardware	16/1/2019	10/4/2019
	• Soldering hardware	6/4/2019	6/4/2019
	• Final Report	28/11/2018	10/4/2019
Ms. Shoaa Alhaider	• Select and order components	21/11/2018	1/4/2019
	• Coding using python	20/1/2019	28/4/2019
	• System design	7/4/2019	9/4/2019
	• System testing and analysis	12/2/2019	14/2/2019

	• Embedded system code	16/1/2019	10/4/2019
	• Assembling hardware	16/1/2019	10/4/2019
	• Final Report	28/11/2018	10/4/2019
Ms. Aqeela Aloraidh	• Select and order components	21/11/2018	1/4/2019
	• Coding using python	20/1/2019	21/1/2019
	• System design + Schematic design	8/4/2019	9/4/2019
	• Roll-up poster design	5/4/2019	7/4/2019
	• Prototype Design	12/2/2019	14/2/2019
	• Assembling hardware	16/1/2019	10/4/2019
	• Final Report	28/11/2018	10/4/2019
Ms. Lina Alkhatib	• Select and order components	21/11/2018	1/4/2019
	• Coding using python	20/1/2019	21/1/2019
	• Literature review	2/2/2019	5/2/2019
	• Assembling hardware	12/2/2019	14/2/2019
	• System testing and analysis	16/1/2019	10/4/2019
	• Installing OpenCV	18/2/2019	19/2/2019
	• Final Report	28/11/2018	10/4/2019

Table 2: Task Breaking Down

4.2. Plan Description:

Weeks	Milestone
Week 1	Meeting for searching for a project
Week 2	Requirement and specification

Week 3	Learning about Python
Week 4	Search about Raspberry and Arduino and learn how to use them
Week 5	Initial design
Week 6	Deciding what component are needed
Week 7	Ordering hardware components from the internet
Week 8	Start to work on the Documentation
Week 9	Search and learn about methodology
Week 10	Documentation
Week 11	Documentation
Week 12	Documentation
Week 13	Documentation

Table 3: Plan Description for first semester

Weeks	**Milestone**
Week 1	Collecting equipment
Week 2	Set up raspberry and operating system
Week 3	Installing python
Week 4	Installing GPIO
Week 5	Installing OpenCV and OCR
Week 6	Setup the Webcam and test OCR
Week 7	Implementing Ultrasonic sensors
Week 8	Setup the buttons
Week 9	Installing gTTS
Week 10	Google translate API
Week 11	Installing Python, GPIO, gTTS for the 2nd raspberry

Week 12	Installing RFID
Week 13	Combine all codes together

Table 3: Plan Description for second semester

5. Requirements Analysis:

5.1. Requirements Specification:

5.1.1. Project Objective:

Glasses are designed to be the eye for the blind person and people who suffer from vision difficulties to make their life easier and be able to continue living their life as a normal human to follow up and achieve their goals and dreams.

- Convert printed text to audio.
- Inform the user by the location of the classes in the green zone.
- It makes their life easier and they will be able to live a normal life.
- Increase education level because “Smart Glasses” will help all people with vision difficulties to study by these glasses with normal people in any school and University.
- Help to translate any English word to Arabic using Google Translate.

5.1.2. Project Specification:

“Smart Glasses” can read any English text images by converting the text in the image into an audible one. It also can translate that text into Arabic and heard by the user using headphones. Moreover, there is an RFID sensor that scans the class’s ID and sends the classes number by a voice message.

5.1.3. Software Specification:

5.1.3.1. OCR Mechanism Steps:

OCR is an abbreviation of optical character recognition method, it is used to convert typed, printed or handwritten text into machine-encoded text. There are some OCR software engines which try to recognize any text in images such as Tesseract and ABBYY FineReader. In this project Tesseract version 4 is used

because it is the best open source OCR engines. OCR process consists of multiple stages:

- **First: preprocessing**

  The main goal of this step is to reduce the noise that resulted from scanning the document where the characters might be broken or smeared and causes poor rates of recognition. preprocessing is done by smoothing the digitized characters through filling and thinning. Another aim of this step is to normalize the data to get characters of uniform size, rotation, and slant.

  Moreover, compression in the amount of information to be kept through thresholding and thinning techniques.

- **Second: Segmentation**

  In this process, the characters or words will be isolated. The words will be segmented into isolated characters that are recognized separately. Most of OCR algorithms segment words into isolated characters which are recognized individually. Usually, segmentation is done by isolating every connected component.

- **Third: Feature Extraction**

  This process will capture the significant features of symbols and it has two types of algorithms which are pattern recognition and feature extraction/detection.

- **Fourth: Classification**

  OCR systems use the techniques of pattern recognition that assigns an unknown sample into a predefined class. One of the ways to help in character classifying is using English dictionary.

- **Fifth: post processing**

  This process includes grouping and error detection & correction. In grouping, the symbols relate to strings. The result of plain symbol recognition in the text is a group of individual symbols.

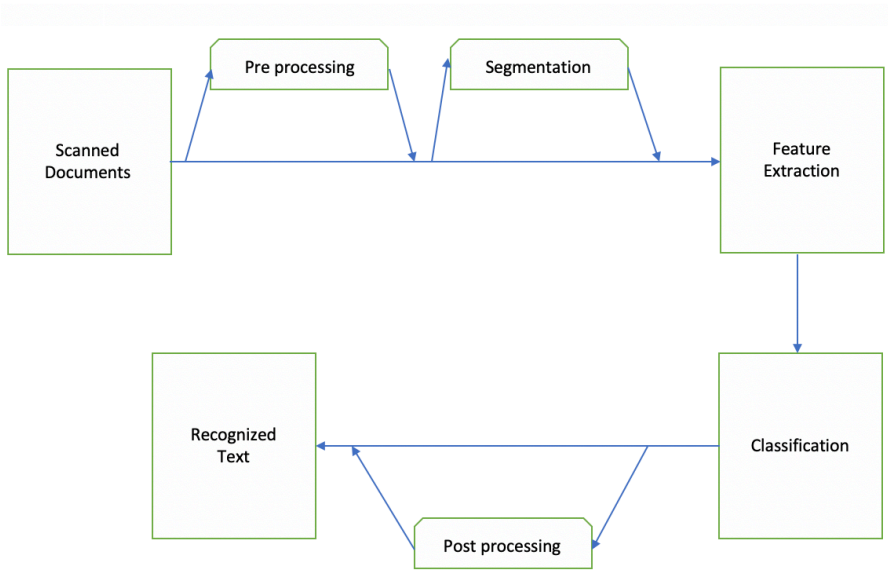


Figure 17: OCR Mechanism Steps

5.1.4. Functional and Non-Functional:

Functional specification	Non-functional specification
Text identification: when the capture button is pressed, the glasses will capture an image and convert the text in the image into audio text.	**Performance:** the glasses must be able to perform properly and extract the text from any image.
Text Translation: when the translation button is pressed, the same text will be translated into Arabic audio text.	**Reliability:** the glasses must achieve high accuracy in identifying the text in the image. **Flexibility:** The glasses can be used by all people and in different places such as college, school, hospital and even in the streets.
Classrooms identification: RFID reader will identify the tags on each classroom's door.	**Scalability:** The glasses are designed to take a clear picture of the distance between 40 and 150 cm. **Usability:** The glasses are light, safe and easily wearable.

Table 4: Functional and Non-Functional

5.1.5. Engineering Standards:

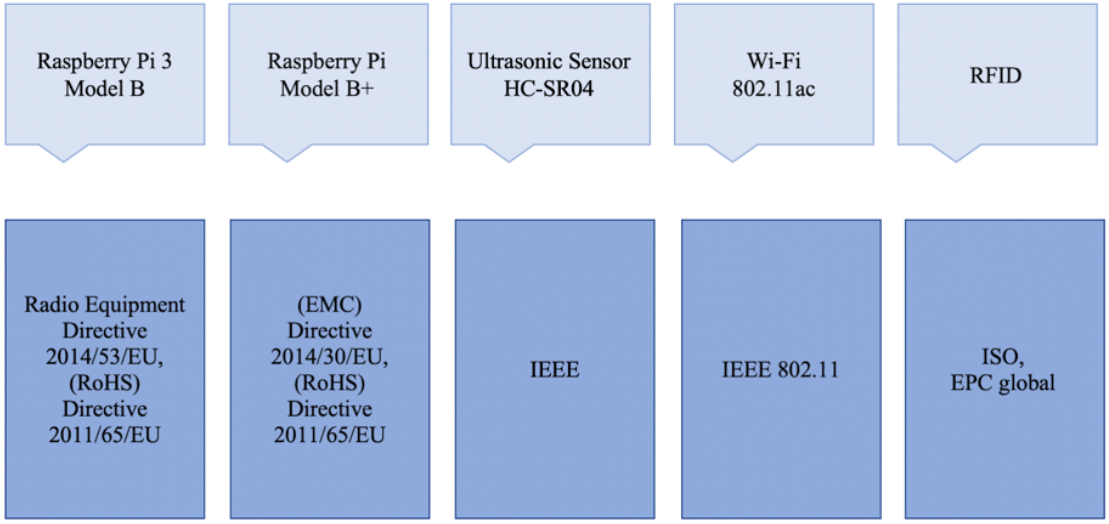


Figure 18: Engineering Standards

5.1.5.1. Raspberry Pi Model B:

Raspberry Pi has declared that Raspberry Pi 3 Model B is in conformity with essential requirements and another relevant requirement of Radio Equipment Directive 2014/53/EU and it also conformity with Restriction of Hazardous Substances (RoHS) Directive 2011/65 / EU.

5.1.5.2. Raspberry Pi Model B+:

All Raspberry Pi product has undergone extensive compliance testing, for Europe, USA and other countries around the world. Raspberry Pi has declared that Raspberry Pi Model B+ is in conformity with the following applicable community harmonized legislation: Electromagnetic Compatibility Directive (EMC) 2014/30/EU, and Restriction of Hazardous Substances (RoHS). Directive 2011/65/EU. The following harmonized standards have been used to demonstrate conformity to these standards: 55032:2012 Class B, EN55024:2010, EN61000-3-2:2014, EN61000-3-3:2013(Raspberry Pi,2017).

5.1.5.3. Ultrasonic Sensor:

Ultrasonic HC – SR04 provides 2 cm - 400 cm non-contact measurement function, the modules include ultrasonic transmitters, receiver and control circuit.

- The basic principle of work:
    1. Using IO trigger for at least 10us high-level signal,
    2. The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
    3. IF the signal back, through high level, time of high output IO duration is the time from sending ultrasonic to return (Elecfreaks, n.d.).

5.1.5.4. WiFi 802.11ac:

The 802.11ac is part of the initial standard 802.11 which is the first WLAN standard created by IEEE. It uses dual-band wireless technology and supports concurrent connections on 2.4 GHz and on 5 GHz Wi-Fi bands. Moreover, 802.11ac is compatible with older versions of the initial standard which are 802.11b/g/n with bandwidth reached up to 1300 Mbps on the 5 GHz band and up to 450 Mbps on 2.4 GHz.

5.1.5.5. RFID

Radio Frequency Identification (RFID) sensors are governed by two main international RFID standards organizations; ISO (International Standards Organization) and EPCglobal (Electronics Product Code Global Incorporated). There are also many standards used in different RFID fields. Although these two organizations provide the main RFID standards organizations, there is also a plethora of other standards that apply to niche areas of RFID.

5.1.6. Constraints:

5.1.6.1. Raspberry Model A Vs. Raspberry Model B:

Both have 700 MHz Low Power ARM1176JZ-F Applications Processor and Dual Core VideoCore IV Multimedia Co-Processor. On the other hand, model A has 56MB SDRAM memory and Single USB Connector while model B has 512MB SDRAM memory and Dual USB Connector. Also, because we are using optical character recognition (OCR), we need the one with better memory which

is model B and our choice will be model 3 B+ because this model came with some important features that the original model B does not have. For example, the **PoE** (Power over Ethernet) is supported via a stand-alone 4-pin connector. And the connector also supports the use of a HAT (Hardware Attached on Top). Moreover, **Gigabit Ethernet** is supported by the new Ethernet connector, so the Speeds on the Pi's slim hardware will reach to average around 300Mb/second.it also has a **Dual-band 802.11ac Wi-Fi** is a standard via a wireless chip and Bluetooth connection which we need to connect the wireless headphones.

5.1.6.2. Raspberry Pi zero W Vs. Raspberry Pi 3 Model B+:

Features	Raspberry Pi 3	Raspberry pi zero w
Size & price	It has a height of 2.22 inches and width of 3.37 inches 35$	It has a height of 1.18 inches and width of 2.55 inches 10$
USB	It has four USB 2.0 ports	It only has one Micro USB
Wi-Fi & Bluetooth	Both has wireless capabilities, 2.4GHz 802.11n wireless LAN and Bluetooth Classic 4.1	
Ethernet	10/100 Mbit/s	No Ethernet connectivity
Analog Video	Available access	
LCD	Available	Not Available
Camera	Available connections	
GPIO (General Purpose Input/Output)	Both has 40 GPIO pins Both allow the usage of +3.3V, +5V, ground, I2C, and SPI	
Speed	It is 20% faster than the Raspberry pi zero w	It is slower than Raspberry pi 3
Core	Quad core	Single core
GPU	Both use Broadcom VideoCore IV	
Memory (SDRAM)	It has a 1GB memory	It has a 512 MB memory

Table 5: Comparison between Raspberry zero W and Raspberry Pi 3 Model B+

5.1.6.3. Operating System:

In the beginning, team members installed NOOBS as an operating system for the Raspberry Pi 3 since it supports Raspbian, but later on, the team found that Raspbian stretch is better for OpenCV version 4 that used for image processing, so the team decided to replace the NOOBS with Raspbian Stretch.

5.1.6.4. Technical:

To meet the project goal and developed the “Smart Glasses” as it should be read and translate the text properly. The team members faced many new technical aspects and learn a new concept that has not been taught before in college courses. The team learned a new programing language which is python in order to code each component to do its functionality correctly. Also, the team worked on new components such as Ultrasonic sensors, RFID sensors, and webcam. However, these components have some limitations due to manufacturing technology. RFID sensors can only detect very small distance which is 3 cm so the user should be very close to it. Also, the image quality needs to be considered so the team chose to capture the images by webcam at 40-150 cm distance. As a result of that the team program the Ultrasonic sensors to sense the distance within the same range that can provide high-quality images.

5.1.6.5. Physical:

The team faced some physical constraint during project development. For example, the image that captured by webcam should be clear in order to enable the OCR with OpenCV to detect the text inside it and read it correctly. The size of the raspberry is too large to wear so the team decides to separate it from the frame. Also, the webcam can capture the images only when the Ultrasonic sensor detects the distance between 40-150 cm. Moreover, the quality and speed of the RFID sensors are very sensitive because the user can pass by the class or hole so it should send the signal quickly.

5.1.6.6. Time:

“Smart Glasses” use image processing and it takes time to be accurate. First, need to take pictures of text and detect the text especially if the image or the font not clear from the image then convert the text to audio using (gTTS). All these processing need time to be done which is one of the constraints. In addition, “Smart Glasses” take some time to control the distance to take a picture from 40-150 cm to be accurate.

5.1.6.7. Economic:

Components	Supplier	Price
CanaKit Raspberry Pi 3 B+ (B Plus) Ultimate Starter Kit (32 GB Edition, Clear Case)	Amazoon.com	89.99$
Samsung microSD Card EVO Plus 32GB	Souq.com	16.00$
WebCam C310	Jarir Bookstore	34.40$
Glasses	Amazon.com	45.99$
Female to Female Jumpers	PMU	7.50$
RFID sensor	Hobby District	8.00$
Power Bank battery 20100mAh	Amazon.com	59.99$
Earbuds Headphones	Amazon.com	12.98$
Raspberry P Model B	PMU	38.50$
Sandisk 64 GB Memory Card	Souq.com	17.15$
		Total: 330.50$

Table 6: Economics

The budget that the team members to complete this project is 400$. The reason by putting a reasonable price is to make it affordable when it released to markets. Putting an attractive price will attract people especially students and who need these glasses and immediately buy it without any hesitation.

5.1.6.8. Social:

Many schools and universities in Saudi Arabia do not provide many teachers or advanced ways to educate blind people. “Smart Glasses” are going to improve the rate of schools and universities that educate blind people. This invention will help social because it is faster and more efficient than other old techniques used

such as “Braille”, which is a writing system used by people who are visually impaired.

5.1.6.9. Environmental:

Any new project must focus on the importance of environmental health that affects human health. “Smart Glasses” project is environmentally friendly because it is one of the products that do not cause any harms to the environment and help preserve the environment.

In the beginning, it has been thinking of using Micro: Bit connected to batteries to give a sound to the person who wears the glasses the name of each hall when entering. However, the idea of using Micro Bit has been changed to RFID sensors to make the project friendly to the environment. Batteries are dangerous to the environment because they are made of different kinds of materials and chemical such as cadmium, lead, zinc, manganese, nickel, silver, mercury, and lithium, and acids. Batteries have a detrimental effect on nature because they end up in garbage and landfills. After that, it will cause emissions of greenhouse gases and give a result of air pollution and global warming. Also, batteries can find a path and mix up into the local water supply and kill plants, animals, and humans who drink or eat seafood from polluted water. It has been mentioned up, that battery made of chemicals that can be absorbed by the soil. When the chemical infiltrates into any area, it will cause harm to the whole area this will give a result of soil pollution. So, it has been decided to use rechargeable batteries in this project to not throw them and affect the environment.

5.1.6.10. Manufacturability:

“Smart Glasses” are a very helpful device for people with vision difficulties. They can use this product everywhere all the time and improve their lives. “Smart Glasses” can be available in different fields like education, entertainment, medical and personal use. There are many people could not be able to continue their study because they have vision difficulties. So, one of the most important fields is education, by “Smart Glasses”, the percentage of educated people will increase. While with “Smart Glasses” they could study as a normal person in any school

and university. School and University, they will be able to accept people with vision difficulties because they can know what was written by the voice instead of reading the text by their eyes.

Prince Mohammad bin Fahd University (PMU) will accept blind people next semester. It considers an achievement and will improve the education level in Saudi Arabia. It considers a helpful chance for blind people to study at a good university that has many different majors. “Smart Glasses” are the appropriate product to be used in this situation. The blind student will be able to use normal PMU books instead of using special books for blind people. Moreover, the student is able to translate difficulties words that they hear but and do not know the meaning of the words. In “Smart Glasses”, there is a sensor at the top to sense for objects in front of the person who wears it and scans the text if there is. Also, the glasses control the appropriate distance between the student and the object that has text by a sensor. In addition, “Smart Glasses” help students to know their classes and labs numbers and names by sending a voice message to the student since he or she enters the hall using the RFID sensor. This message protects any accident or miss classes.

5.1.6.11. Safety:

Raspberry Pi 3 is the main component to operate the “Smart Glasses”. However, the raspberry pi 3 got warm when it is connected to the breadboard through wires while it’s on. As a part of safety conditions, the raspberry pi 3 was shut down until all the setup connection with breadboard were done, then the Raspberry Pi 3 is turned on and connected to the breadboard with wires. Team members notice this issue with the raspberry pi 3 when the Ultrasonic sensor was connected to the Raspberry Pi 3 in order to perform the distance detection operation.

6. System Design:

6.1. Conceptual Design:

At the early stage, the initial idea of the project was to create “Smart Glasses” that capture any text images and convert it to voice, then, if the user wants to translate the text, he/she can press another button to translate. Initially, the team chose NOOBs as an operating system to be installed on the Raspberry Pi B+ and implement all the glass functions. Also, the camera was built in the camera inside the glass and the Raspberry Pi was put directly on the glass. The main target for the “Smart Glasses” was blind people to allow them conceptually to use it anytime anywhere such as:

- Schools
- Malls
- Hospitals
- Museums
- Roads

In order to accomplish the project, at the conceptual design the team proposed the initial hardware below:

- Smart glass with a built-in camera
- Raspberry Pi 3 Model B+
- Ultrasonic sensor
- Headphone
- Buttons

Team members decided to draw an initial design to make it easy for them to set up the components to be done to complete the project.

The Initial Design:

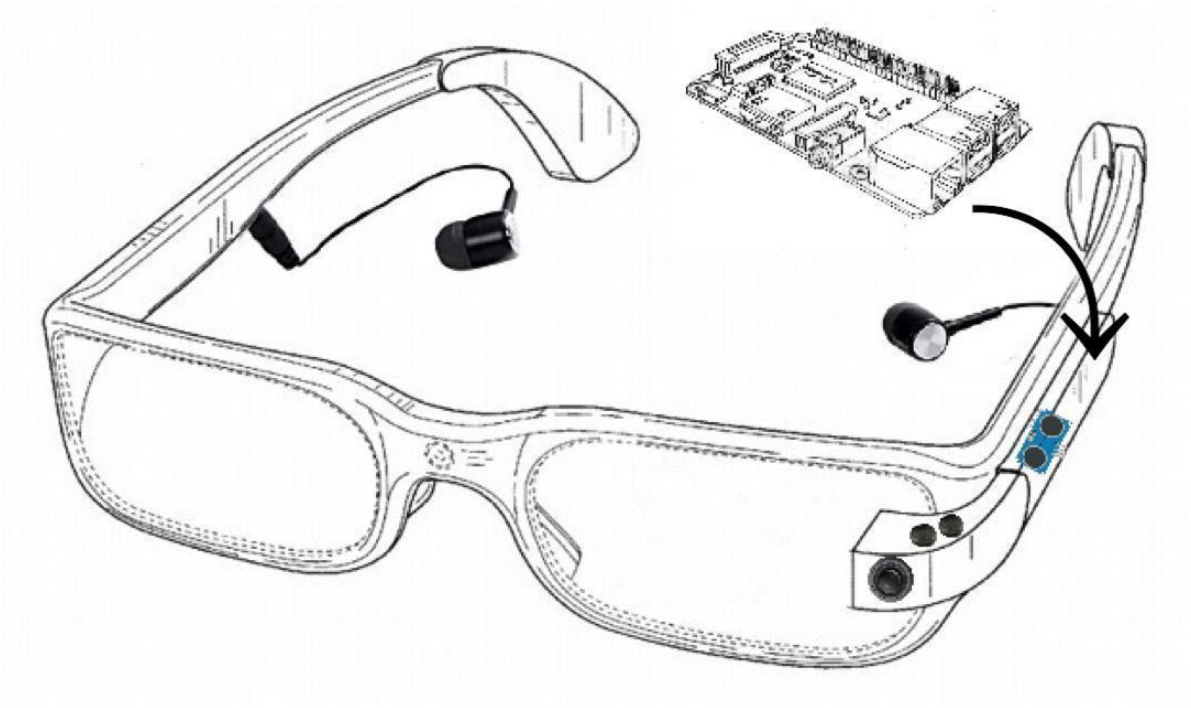


Figure 19: Initial design of “Smart Glasses”

However, during the work on the hardware part, the design was improved and modified by the team members due to the inconveniences they faced. changing the camera to Webcam that takes pictures within 40 cm to 150 cm, rather than using a normal camera, team members used Webcam that supports Raspberry Pi. Also, the Raspberry Pi has been changed from putting it on the glasses to put it outside because it is large, and it is not comfortable for the user to wear it on the user’s head. So, the user will wear the Raspberry Pi 3 Model B+ on his/her upper arm. Also, team members added a new feature to the project which introduced the blind to the class locations using the RFID sensor. As a result of that, the design was drastically changed.

The Improved Design:

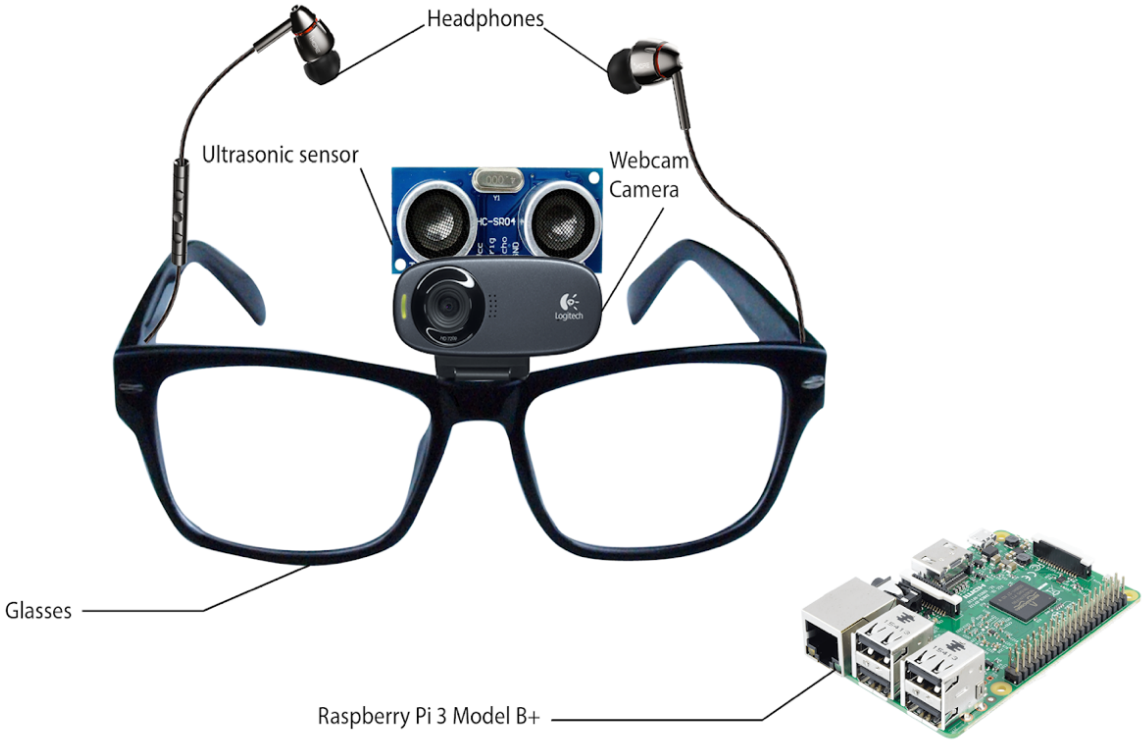


Figure 20: Improved design of "Smart Glasses"

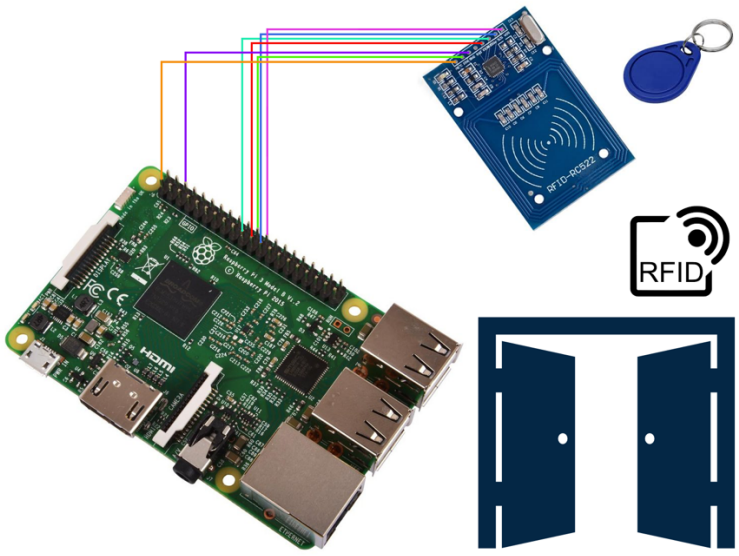


Figure 21: Design of RFID Sensor

6.2. Hardware Design:

The below figure represents the structure of the “Smart Glasses” components’ using a schematic diagram. All the components connections are highlighted including Raspberry pi 3 B+, Ultrasonic sensor, Webcam, buttons, resistors, and breadboard. The purpose of the schematic diagram is to provide a deeper view of the interconnecting between all the components that have been used to develop the project.

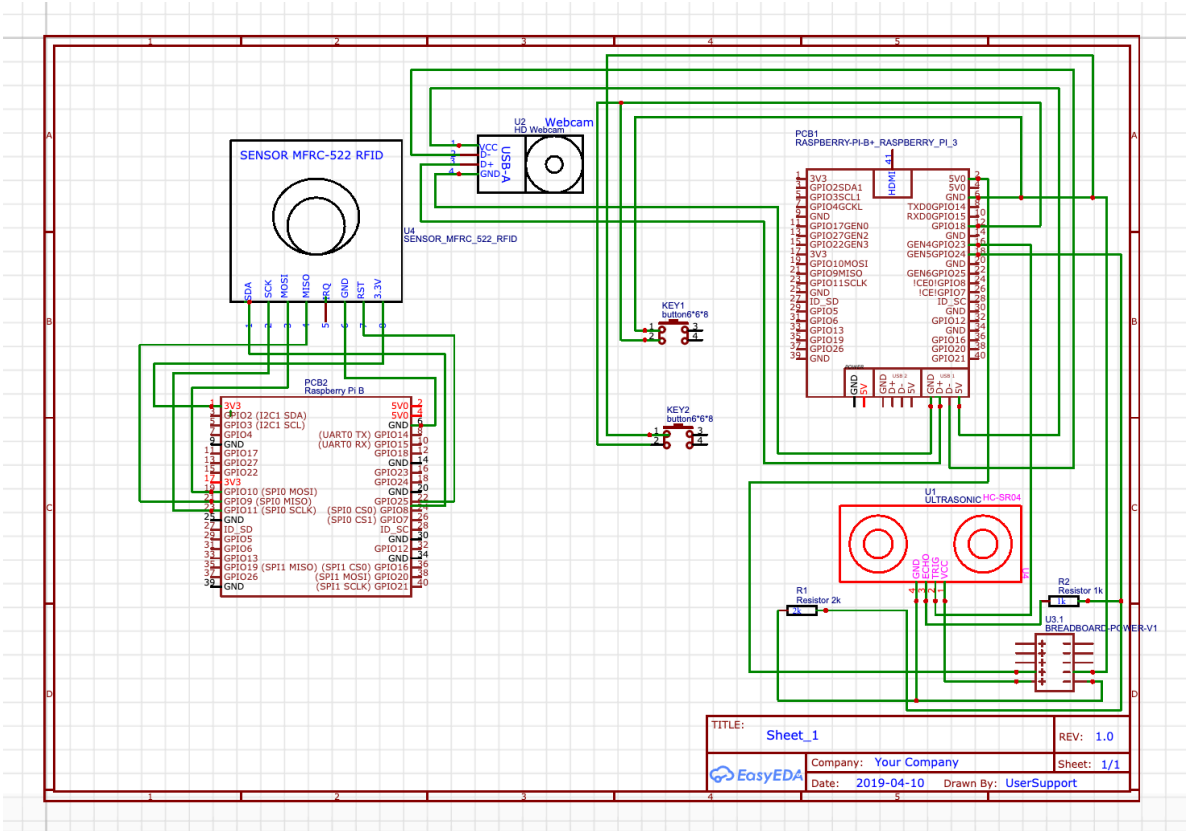


Figure 22: Schematic Diagram of hardware design

6.3. Design Methodology:

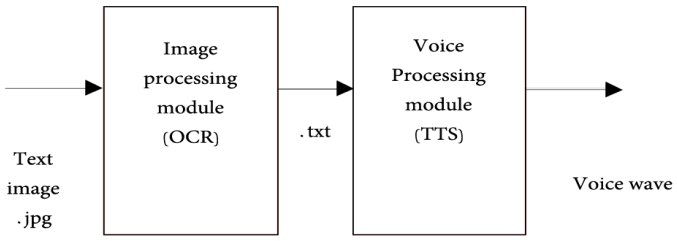


Figure 23.1: Design Methodology

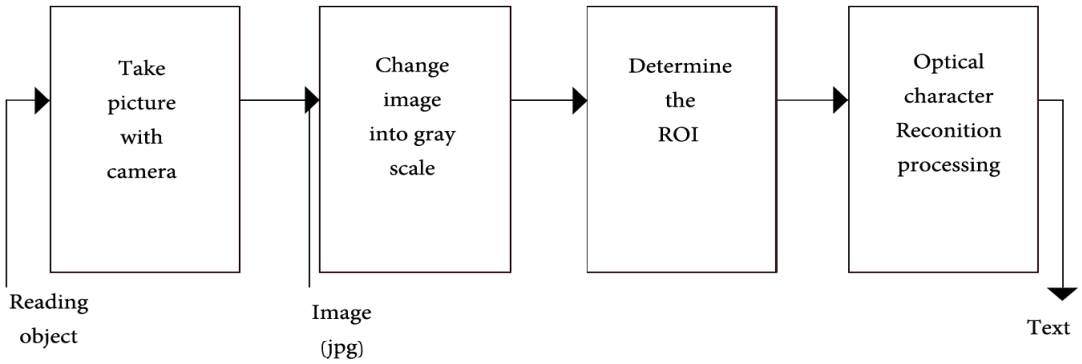


Figure 23.2: Design Methodology

7. Implementation:

7.1. Installing Raspbian Stretch Operating System:

Before downloading the operating system on the SD card, the SD card should be formatted, it was used SD Card Formatter for this purpose, Figure 1.9. Then the Raspbian stretch files were moved to the SD card. After that, the SD card was inserted into the Raspberry Pi. Finally, the keyboard, the mouse, HDMI cable and lastly the power were inserted into the correct raspberry ports.

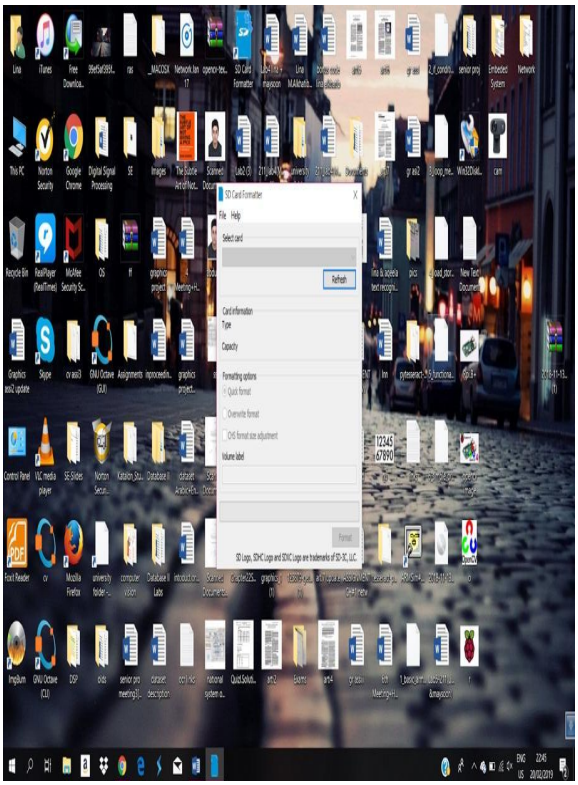


Figure 24: Formatting SD Card

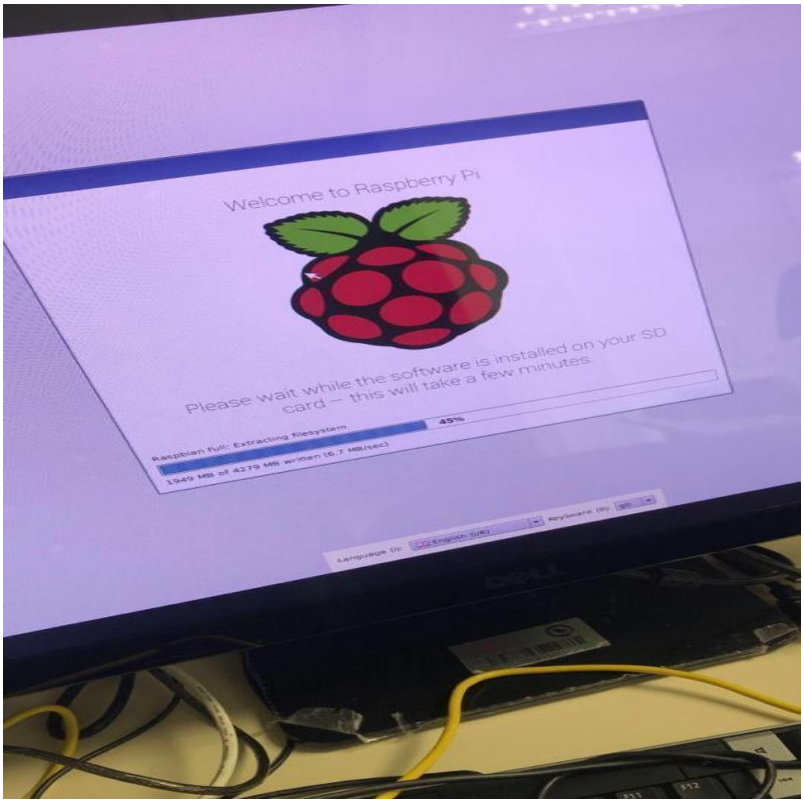


Figure 25: Installing Raspbian Stretch

7.2. Installing OpenCV 4 Libraries:

OpenCV is a library of programming functions for real-time computer vision, the library is cross-platform and free for use under the open-source BSD license, Figure 1.10. For the installation of the OpenCV 4 libraries, the recommended operating system for the raspberry pi B+ which is Raspbian Stretch was installed. Win32 Disk Imager was used to flash the SD card.

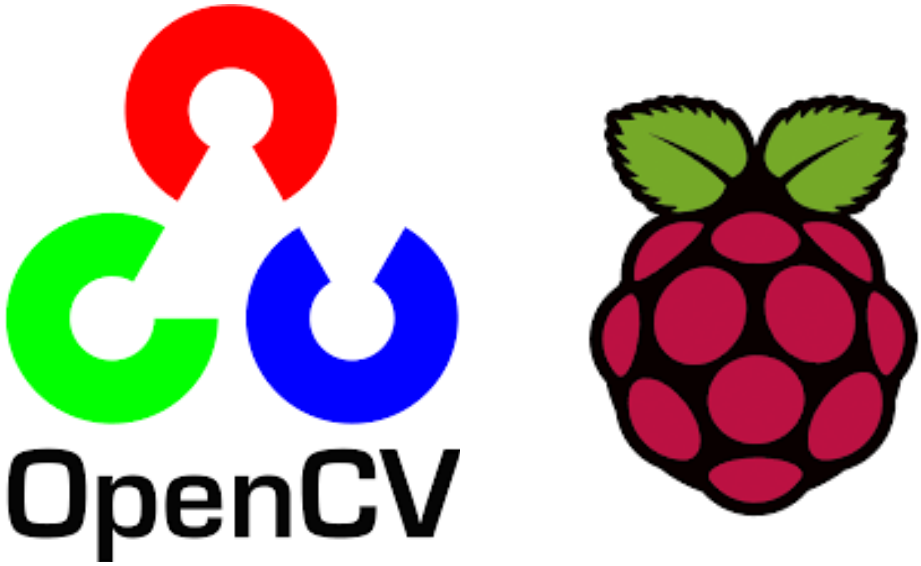


Figure 26: OpenCV

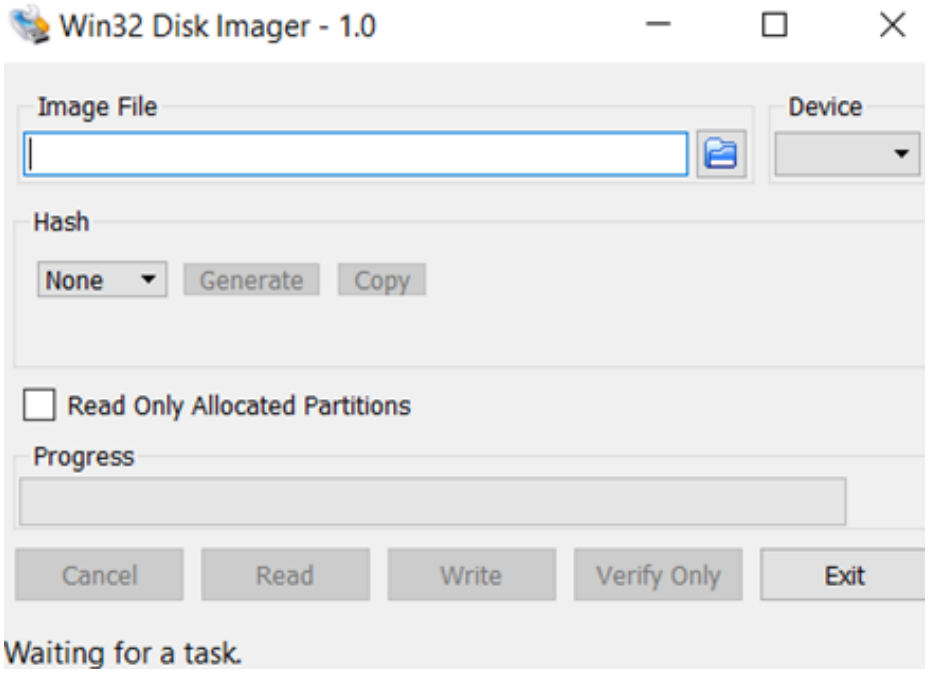


Figure 27: Win32 Disk Imager -1.0

Here are the installation steps for OpenCV 4:

Step (1): installing the OpenCV 4 dependencies:

Update the system:

- $ sudo apt-get update && sudo apt-get upgrade

Install CMake:

- $ sudo apt-get install build-essential cmake unzip pkg-config

Install image and video libraries:

- $ sudo apt-get install libjpeg-dev libpng-dev libtiff-dev
- $ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
- $ sudo apt-get install libxvidcore-dev libx264-dev

Install GTK, our GUI backend:

- $ sudo apt-get install libgtk-3-dev

Install a package to reduce GTK warnings:

- $ sudo apt-get install libcanberra-gtk*

Install numerical optimizations packages:

- $ sudo apt-get install libatlas-base-dev gfortran

Install the Python 3 development headers:

- $ sudo apt-get install python3-dev

Step (2): Download OpenCV 4 for Raspberry Pi:

Download Opencv & Opencv_contrib:

- $ cd ~
- $ wget -O opencv.zip https://github.com/opencv/opencv/archive/4.0.0.zip
- $ wget -O opencv_contrib.zip
  https://github.com/opencv/opencv_contrib/archive/4.0.0.zip

Unzip the archives:

- $ unzip opencv.zip
- $ unzip opencv_contrib.zip

Rename the directories:

- $ mv opencv-4.0.0 opencv
- $ mv opencv_contrib-4.0.0 opencv_contrib

Step (3): Configure your Python 3 virtual environment for OpenCV 4:

Install pip, a Python Package Manager:

- $ wget https://bootstrap.pypa.io/get-pip.py
- $ sudo python3 get-pip.py

Install Virtualenv & Virtualenvwrapper:

- $ sudo pip install virtualenv virtualenvwrapper
- $ sudo rm -rf ~/get-pip.py ~/.cache/pip

Update ~/.profile:

- $ echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile
- $ echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.profile
- $ echo "export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3" >> ~/.profile
- $ echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile
- $ source ~/.profile

Create OpenCV 4 Python 3 virtual environment:

- $ mkvirtualenv cv -p python3

Verify the cv environment:

- $ work on cv

Installing NumPy:

- $ pip install numpy

Step (4): CMake and compile OpenCV4:

- $ cd ~/opencv
- $ mkdir build
- $ cd build

Compile OpenCV 4:

- $ make

Install OpenCV 4 with two additional commands:

- $ sudo make install
- $ sudo ldconfig

Step (6): Link OpenCV 4 into your Python 3 virtual environment:

- $ cd ~/.virtualenvs/cv/lib/python3.5/site-packages/
- $ ln -s /usr/local/python/cv2/python-3.5/cv2.cpython-35m-arm-linux-gnueabihf.so cv2.so
- $ cd ~

7.3. How to set Wi-Fi on Raspberry Pi Model B+:

We can configure WIFI from the command line

1. Log in to the Pi.
2. If you do not know the password can write on command to generate a list of wireless networks available in your area.
3. Add WIFI name and password.
4. Enter code
5. Press CTRL-X, then Y to save and exit.
6. If not work restarts the Pi.
7. If you have monitor and keyboard you can access the WIFI from the desktop.

7.4. GPIO (General Purpose Input/Output):

Write on command window:

sudo pip install pySerial

sudo pip install nose

sudo pip install cmd2

then we import the GPIO:

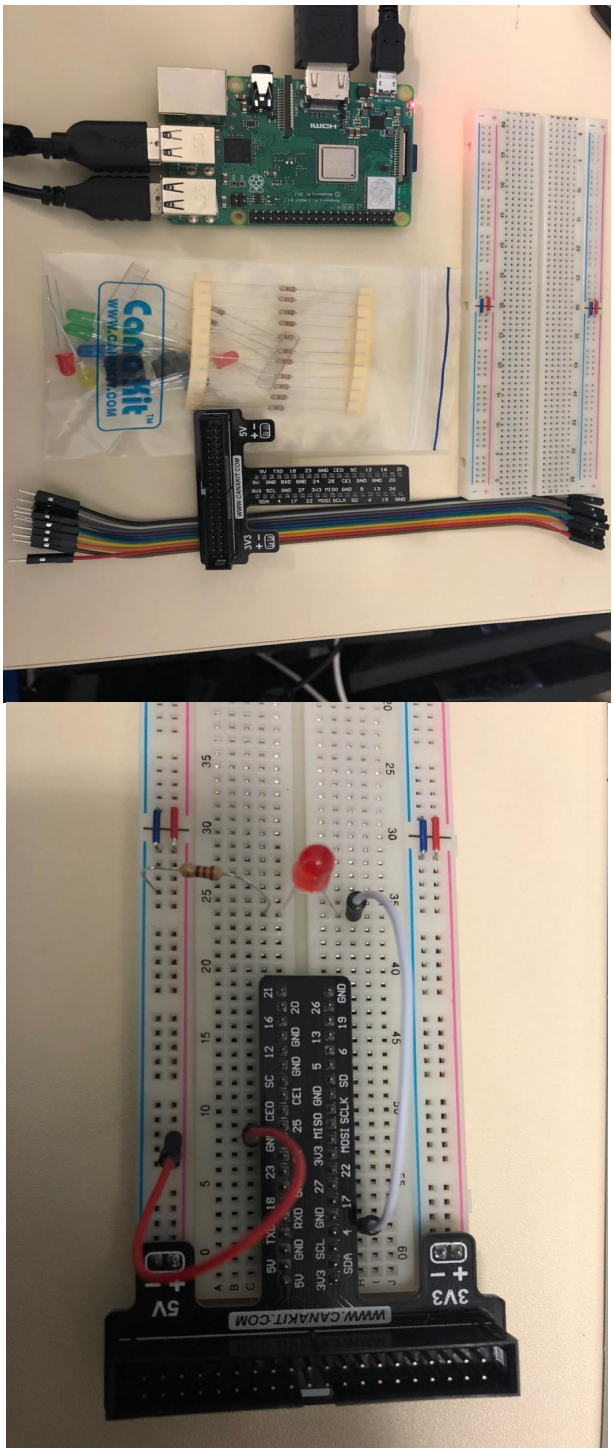


Figure 28: Importing GPIO Library

7.5. Importing Ultrasonic Sensor:

"Smart Glasses" used the Ultrasonic Sensor to detect the text image. To connect the sensor the team used 1k Ω Resistor, 2k Ω Resistor, and Jumper Wires to connect Vcc, TRIG, ECHO, and GND. The team used python code to control the Ultrasonic Sensor detection then be able to measure the destination.

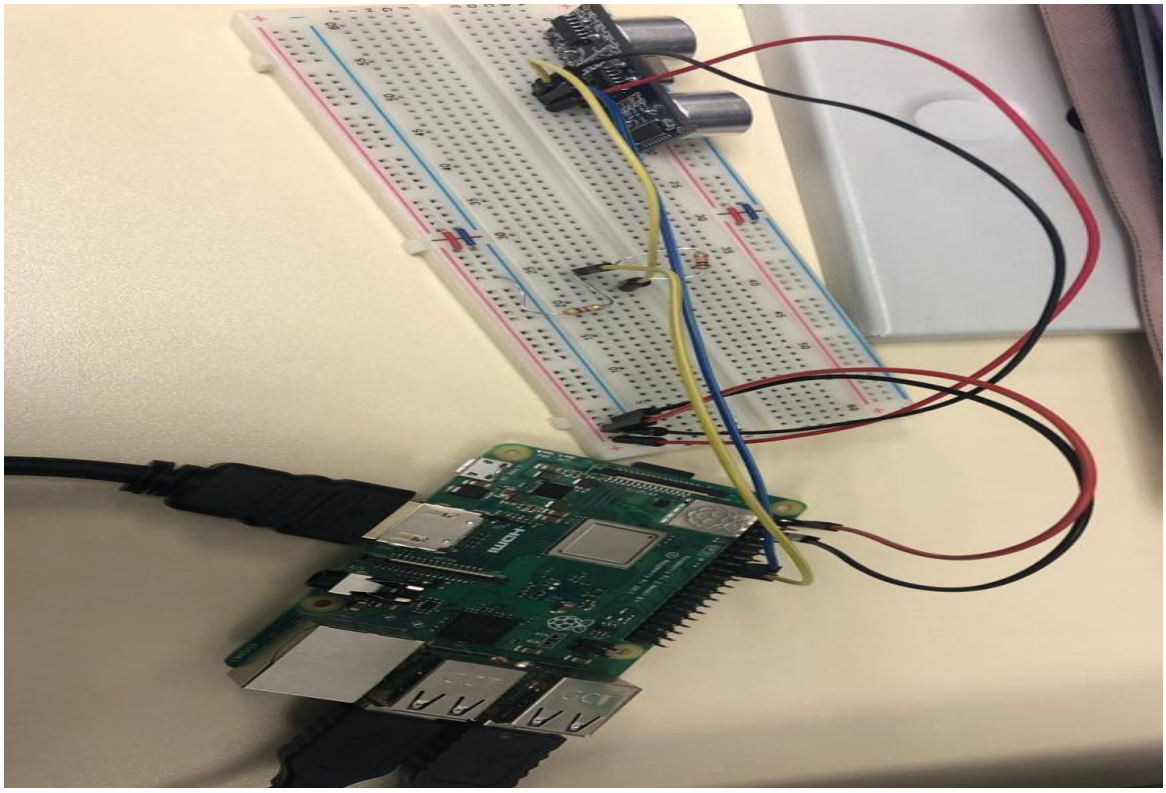


Figure 29: Connections of Ultrasonic Sensor

7.6. Setting Up the Button:

"Smart Glasses" used two buttons one for capture the image and other for translating from English to Arabic using Google API. In order to, set the button the team runs small python code to connect the button. To check the button the team, add condition statement to print "Button Pressed" when the button was pressed. The connection was easy just two wires one for the ground and the other one connects the pin.

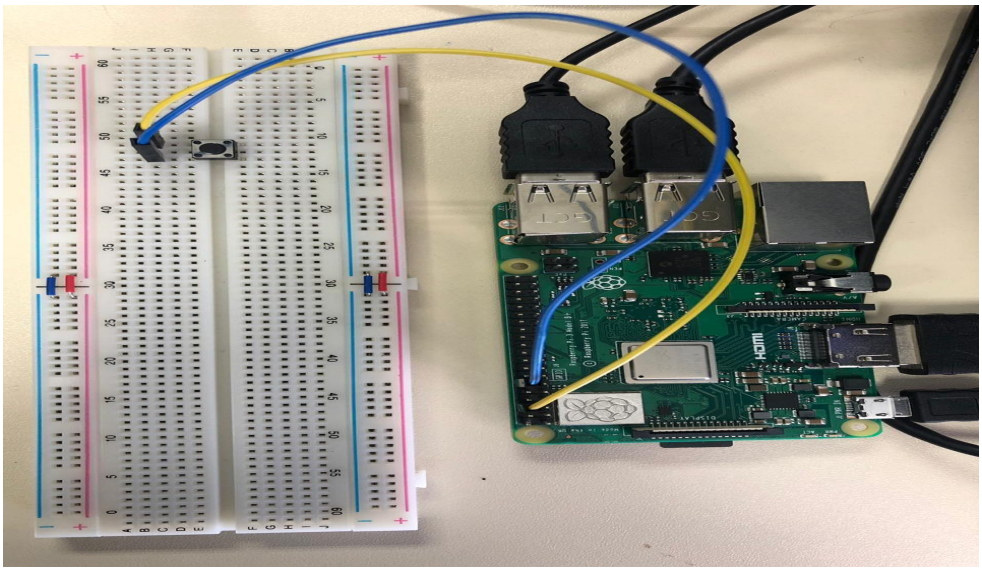


Figure 30: Circuit for button

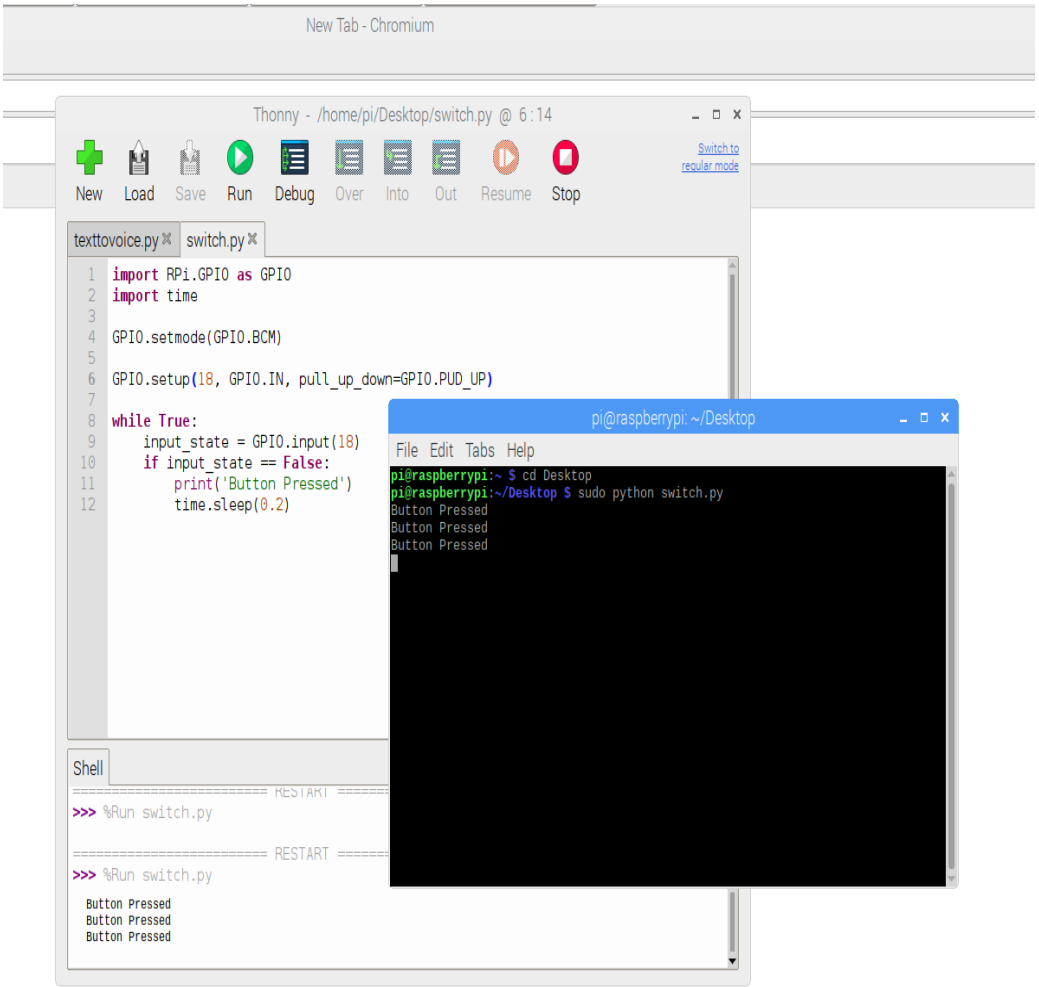


Figure 31: Code of the button

7.7. Taking Picture Setting Up:

Install the fswebcam package:

sudo apt-get install fswebcam

Add user to video group:

sudo usermod -a -G video <username>

Taking picture with no banner:

fswebcam -r 1280x720 --no-banner image3.jpg
fswebcam -r --no-banner image2.jpg

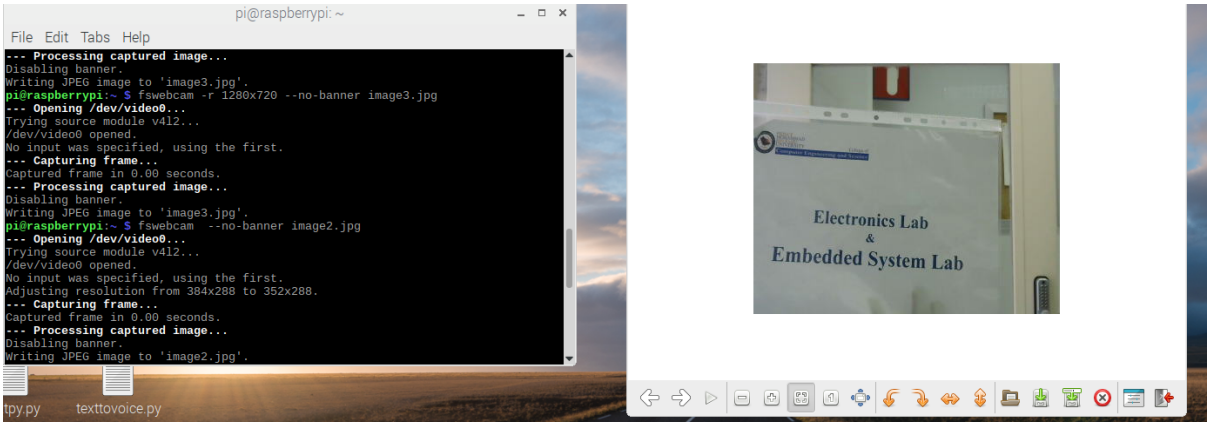


Figure 32: Taking picture

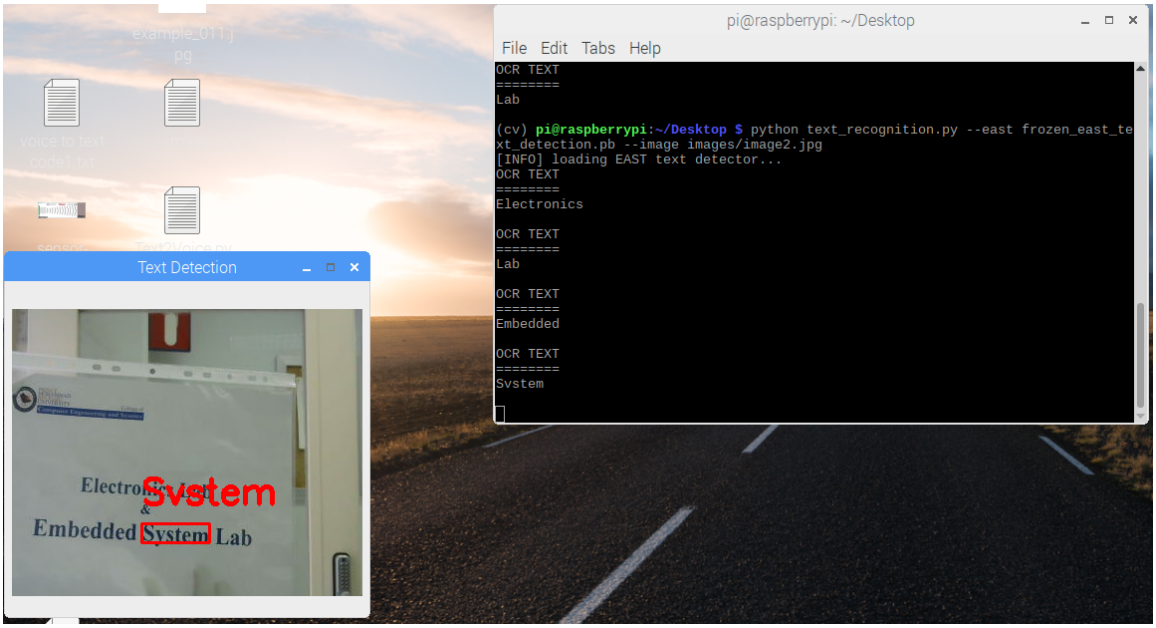


Figure 33: OCR recognition

7.8. Setup a Raspberry Pi RFID RC522:

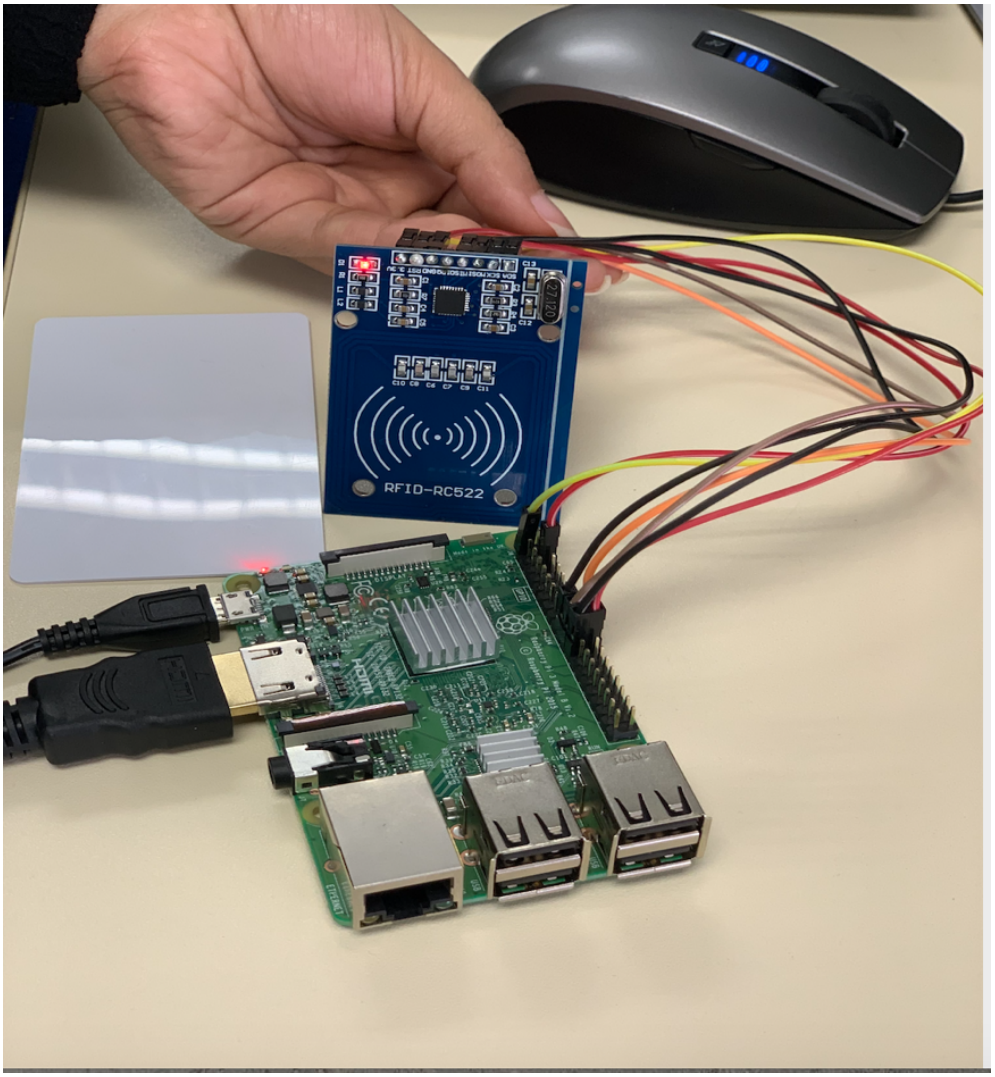


Figure 34: Setup the RFID Sensor

The connections:

- Team members connected **SDA** to **Pin 24**.
- Team members connected **SCK** to **Pin 23**.
- Team members connected **MOSI** to **Pin 19**.
- Team members connected **MISO** to **Pin 21**.
- Team members connected **GND** to **Pin 6**.
- Team members connected **RST** to **Pin 22**.
- Team members connected **3.3v** to **Pin 1**.

Open raspi-config tool:

```
sudo raspi-config
```

Then select:

"**5 Interfacing Options**", and press **Enter.**
"**P4 SPI** ", and press **Enter.**

It will ask if you want to enable the SPI Interface, select **Yes,** and press **Enter.**
It will show a screen of "**The SPI interface is enabled** ", press **Enter** and then **ESC**.

To restart your Raspberry Pi:

```
sudo reboot
```

When your Raspberry Pi has finished rebooting, we can now check to make sure that it has in fact been enabled:

```
lsmod | grep spi
```

If you see spi_bcm2835, means it is unbales

Update and upgrade the Raspberry Pi:

```
sudo apt-get update
sudo apt-get upgrade
```

Install python3-dev, python-pip and git:

```
sudo apt-get install python3-dev python3-pip
```

Install the Python Library spidev:

```
sudo pip3 install spidev
```

Install the MFRC522 library:

```
sudo pip3 install mfrc522
```

Create a folder:

```
mkdir ~/pi-rfid
```

Change the directory, and to write on Write.py script:

```
cd ~/pi-rfid
sudo nano Write.py
```

Write this code in Write.py:

```
#!/usr/bin/env python
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()
try:
     text = input('New data:')
     print("Now place your tag to write")
     reader.write(text)
     print("Written")
finally:
     GPIO.cleanup()
```

Press Ctrl + X, Y then Enter to exit.

Test the script by:

```
sudo python3 Write.py
```

Write Hello in new data, then place your RFID Tag on top of your RFID RC522 Until it shows you “Written”.

Run Read.py script to write the code:

```
sudo nano Read.py
```

Write rhis code in Read.py:

```
#!/usr/bin/env python
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()
try:
     id, text = reader.read()
     print(id)
     print(text)
finally:
```

```
        GPIO.cleanup()
```

Press Ctrl + X, Y then Enter to exit.

place your RFID Tag on top of your RFID RC522
Until it shows you:

```
pi@raspberrypi:~/pi-rfid $ sudo python3 Read.py
827843705425
Hello
```

7.9. Convert text to voice:

One of the most important functions of the “Smart Glasses” is text to voice conversion. In order to implement this task, team installs gTTS which is abbreviations of Google Text-to-Speech. It is a python library that interfaced with Google Translate API. gTTS has many features such as convert ultimate length of text to voice, provide error pronunciation using customizable text pre-processors and support many languages and retrieve them when needed. Using python code, team members implemented the text to voice conversion task. The gTTS was installed using the following command:

$ pip install gTTS

8. System Testing and Analysis:

8.1. Test Ultrasonic Sensor:

Before using the Ultrasonic sensor on “Smart Glasses”, the team tested the sensor if it was able to sense and detect the object from 2 cm to 400 cm or not. The team run a small python code and connected the sensor to test it. It was easy to connect the sensor using breadboard, wires, and resistors. The team set up time in the code to measure the distance. First, the team tested the sensor by placing the ultrasonic sensor in front of the table and it was sense and wrote the distance on the output. The team repeats the testing at a different distance. The problem was that the “Smart Glasses” used the ultrasonic sensor to control the distance between 40 to 150 cm to capture the image using the Webcam. So, the team was trying to set a condition to control the ultrasonic sensor to sense just between 40 to 150 cm. The team used If condition to sense when the object in front 40 cm to 150 cm so the webcam can capture the image.

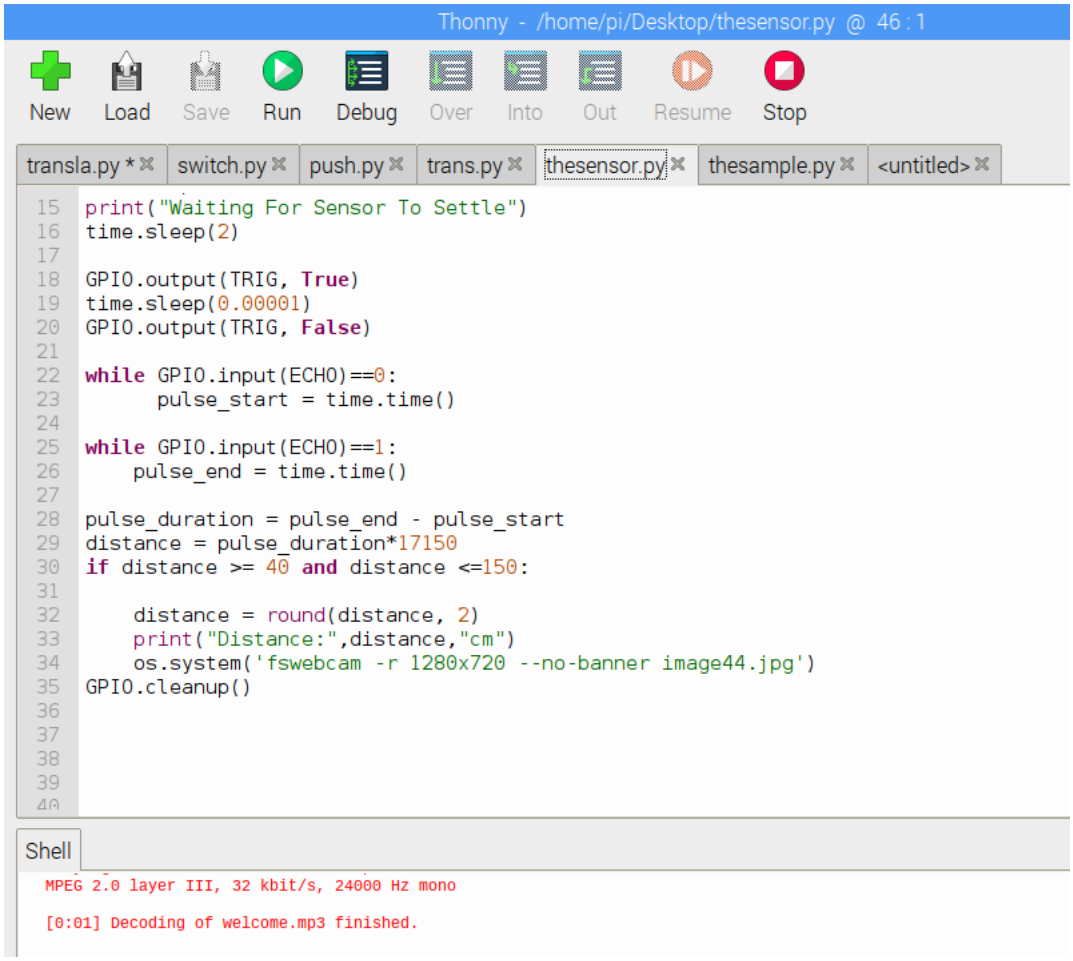


Figure 35: Code for testing Ultrasonic sensor

8.2. Test RFID Sensor:

The team members used python code, jumpers, RFID sensor, and tags and breadboard. Team members faced many difficulties while setting up the RFID sensor and test it. Team members tested the RFID using Arduino to see if the RFID sensor is working or not. Because it was the first time for all team members to use RFID sensors, they did not know that they should do soldering to keep all jumpers and connections working. Then team members tested the RFID and gave us the right outputs. After doing the connection, team members wrote the codes in the terminal and placed the tag on the top of the RFID sensor, and it gave them the right outputs for both read and write. Team member used if condition to let the person who have the tag with the right ID to hear a voice using gTTS of the hall or class name when he/she enter.

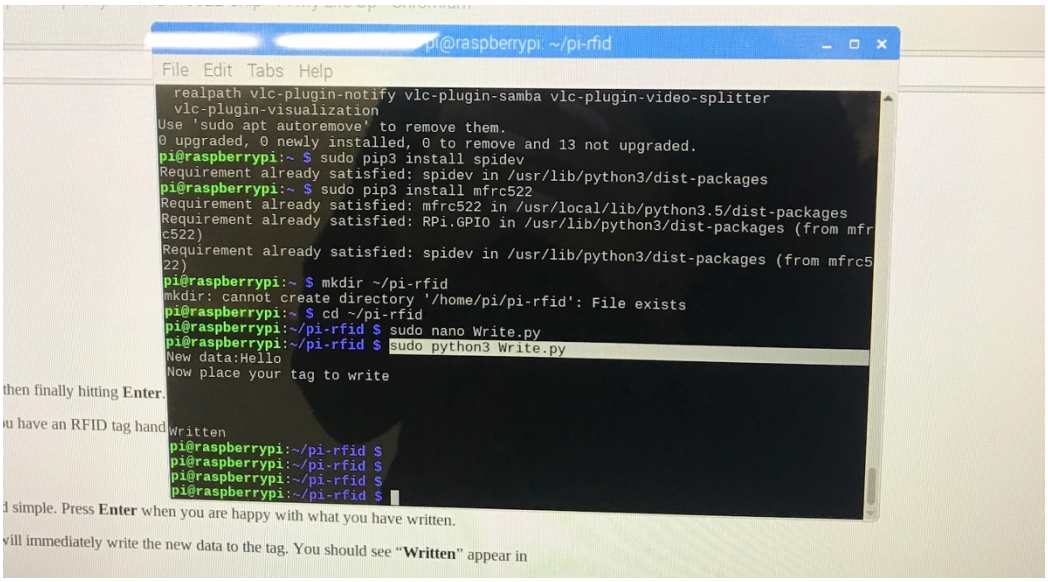


Figure 36: Code for testing RFID Sensor

8.3. System Testing:

The "Smart Glasses" has been tested with all components to make sure that everything is working. The team has done multiple tests for each part before putting them together. The first test was to make sure that the camera is taking a picture after pressing the button and checking the distance if it is in the required range. The second test was to see if the text in the picture has been detected and recognized or not. The third text was to make sure that the detected test has been converted into audio text. The fourth test was to see if the button pressed the detected text has been translated or not. The final test was to check if the RFID can identify the class's ID.

Test 1: Taking picture:

This test aimed to examine taking picture process which starts from pressing the button then checking the range between the camera and the picture using the Ultrasonic sensor. If the picture placed in the required range the picture will be taken. The camera was able to take a clear picture when the button is pressed and if the person is standing stable, but when the person moves after she/he presses the button the picture will be blurred or not clear.

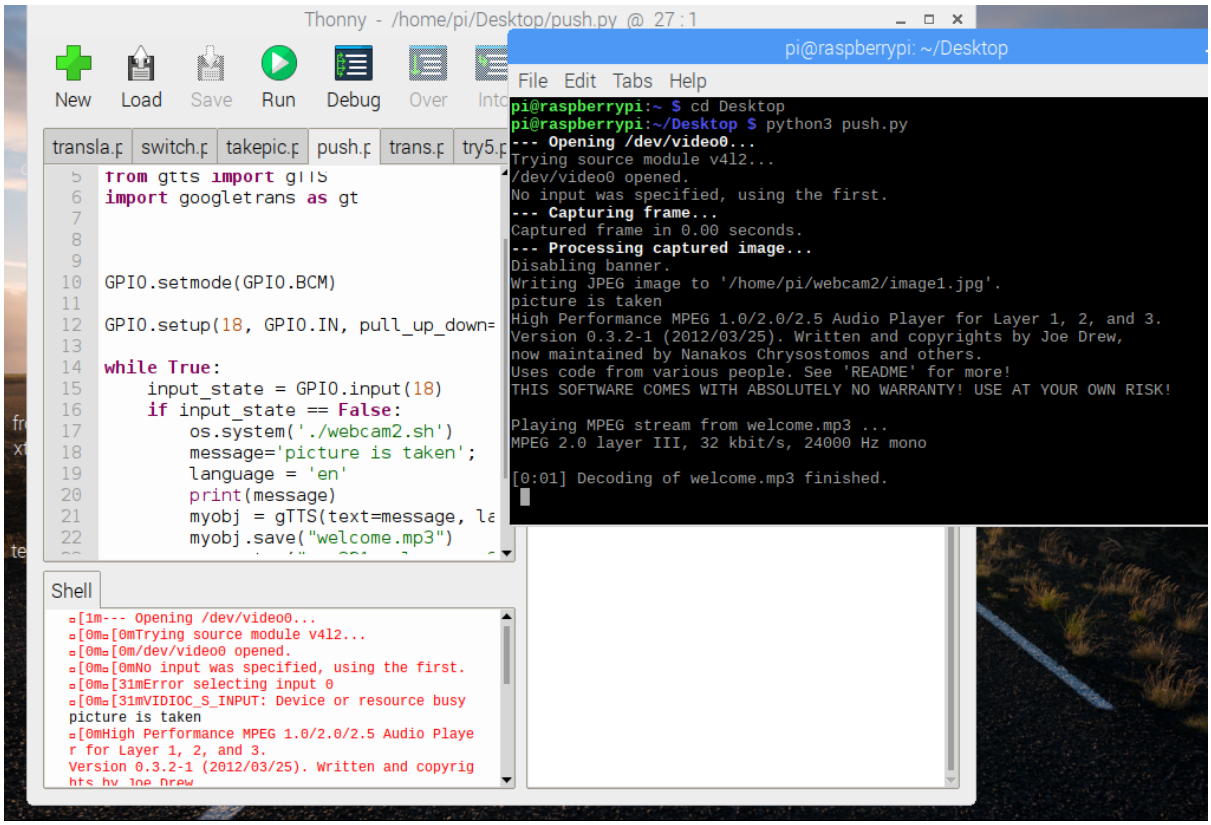


Figure 37: Testing of taking picture

Test 2: Text detection and Recognition:

In this test, the main goal was to check if the text detector that was used in this project which is EAST pre-trained text detector and the text recognizer which is OCR using Tesseract is working well. The test showed some good results on big clear texts and some bad ones on small texts. The team found out that this depends on the clarity of the text in the picture, it also depends on the font theme, size, and spaces between the words.

Figure 38: Testing of detection the texts of images

Test 3: Text to Voice:

The test here was to check if the detected text is converted to audio text. The team used gTTS (Google Text To Speech) libraries after they found that the voice is better and clearer than the other TTS libraries such as Festival TTS, Espeak TTS, and Pico TTS. The voice was clear for the right detected words.

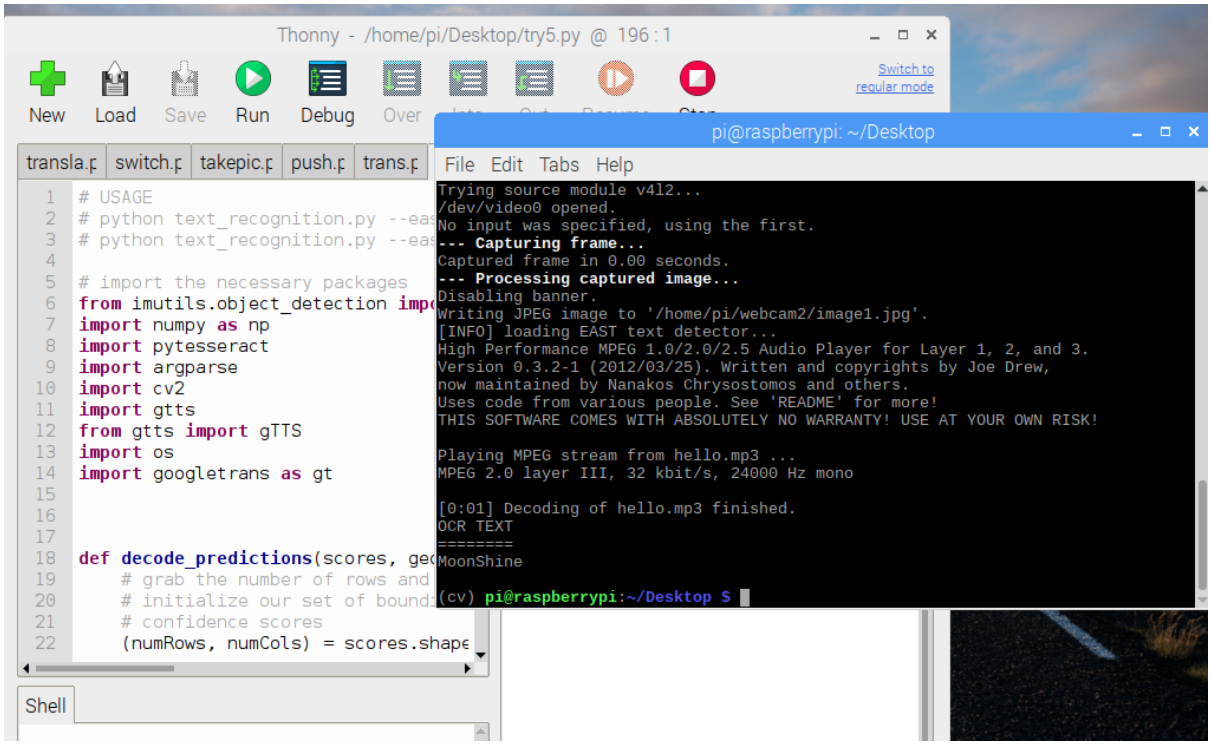


Figure 39: Testing the gTTS

Test 4: Text Translation:

In this test when the person presses the button, the detected text should be translated into Arabic and should be heard clearly. The right detected text was translated well using Google translate and gTTS libraries.

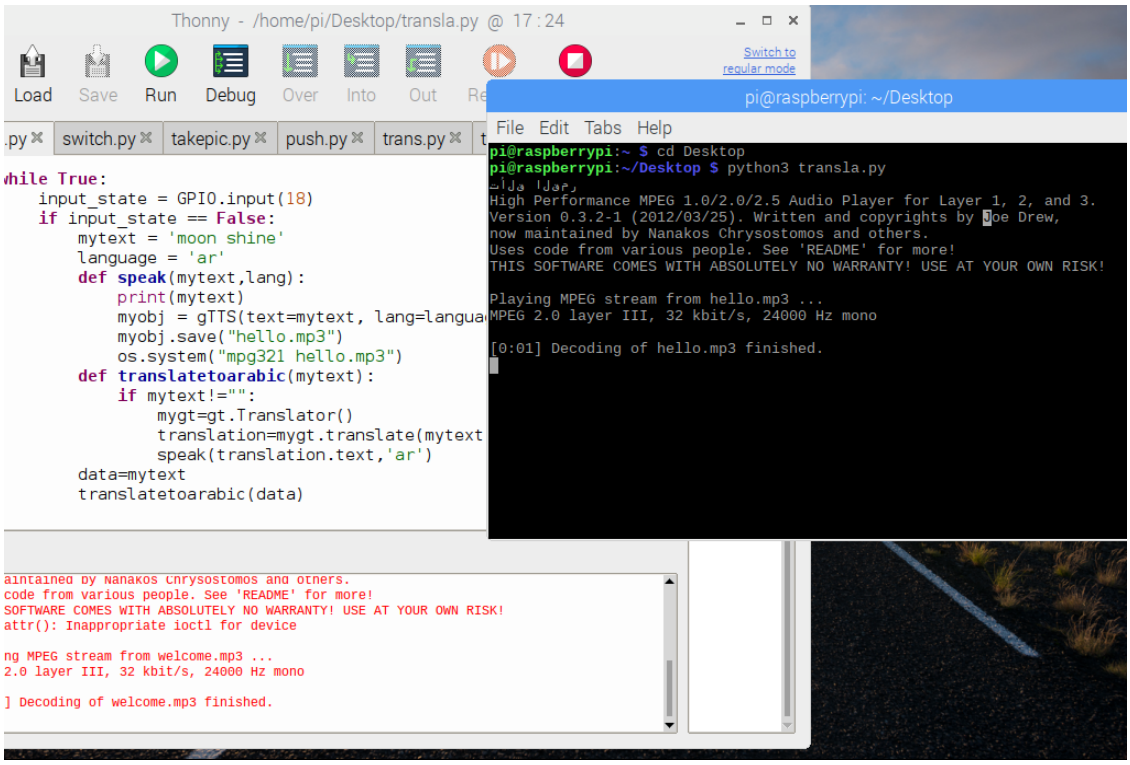


Figure 40: Testing the Google Translate API

Test 5: ID's identification by RFID:

The goal of this test is to check if the RFID reader can read the ID of the tag that is placed on the class or halls. The RFID was able to read the number of the class correctly.

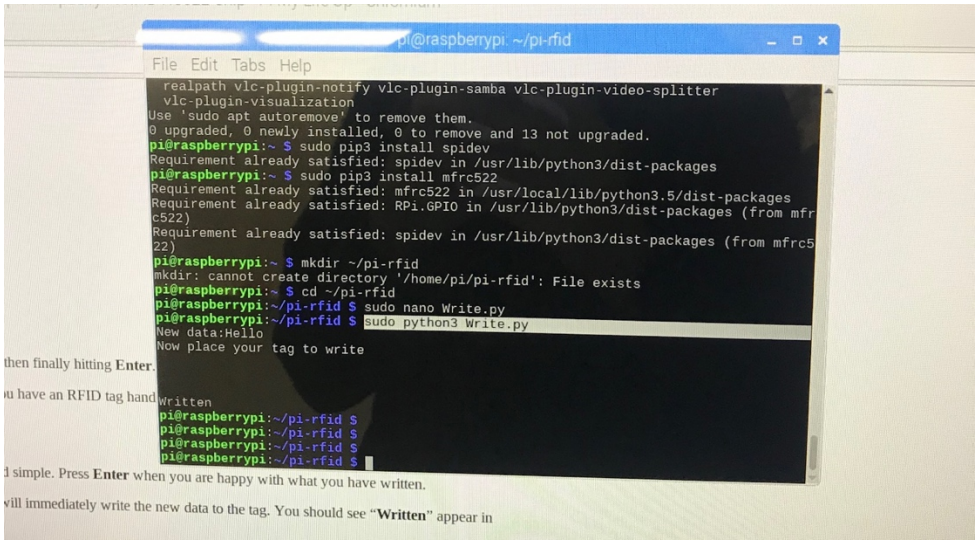


Figure 41: Testing the RFID Reader

9. Project Plan:

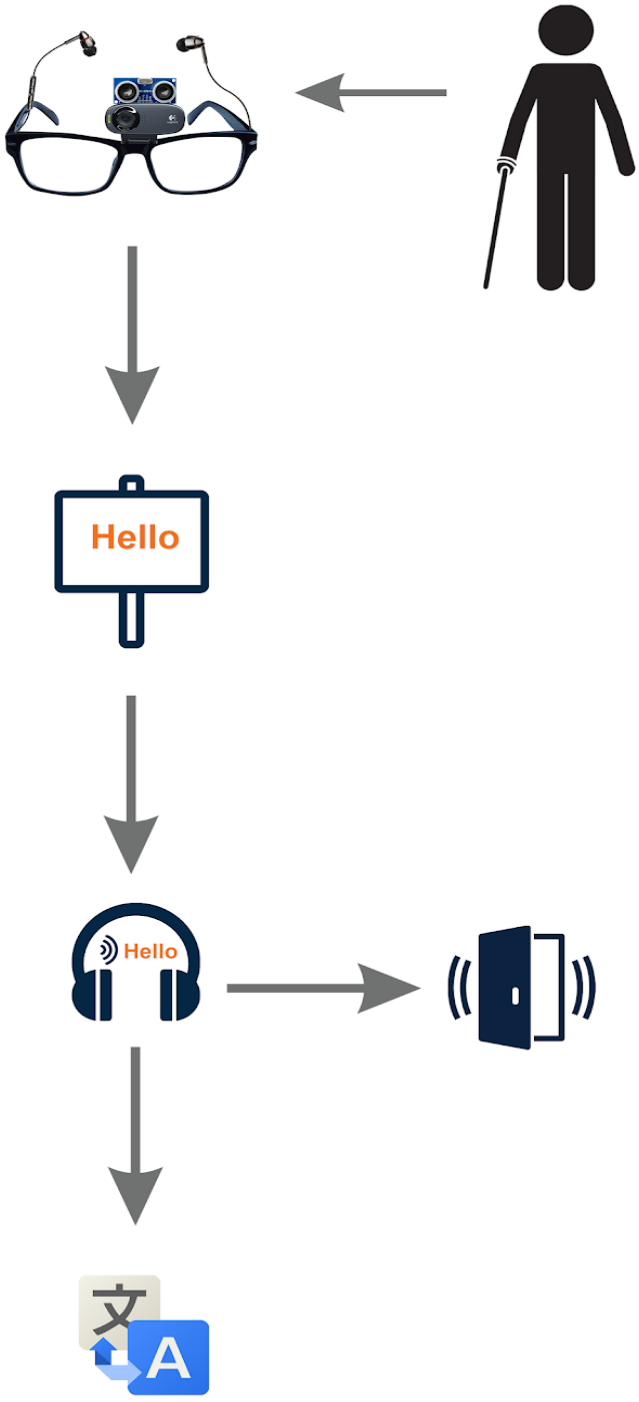


Figure 42: Project flow chart

The First Plan:

For the first plan, team member used:

- Raspberry Pi 3 model B+
- Ultrasonic sensor
- Headphones
- 2 buttons
- Webcam
- SD card 64GB

Team member Installed:

- Python 3
- Open CV
- OCR using OpenCV
- gTTS

The user will wear glasses and team members called is “Smart Glasses” connected to Raspberry Pi 3 Model B+. When the wearer wants to take a picture of text images, the Ultrasonic will measure the distance between the ultrasonic and the image. If the image is between 40-150 cm, the glasses will take a picture of the image Using a Webcam. Then, using OCR the text will be identified. Finally, using gTTS the text will be converted to a voice that the user is going to hear through a headphone. The user will wear the Raspberry Pi Model B+ on his/her upper arm. The user can do the translation of texts by pressing the second button.

The Second Plan:

For the second plan, team member used:

- Raspberry Pi 3 model B
- RFID sensor + Tags
- Headphones
- SD card 32 GB

Team member Installed:

- Python 3
- gTTS

Team members used the RFID sensor to let the user know the name of classes and halls in the university. When the wearer places the tag near to the RFID sensor, the user will hear the voice using gTTS through headphones. This plan completed with another Raspberry Pi 3 Model B that the user will wear it on his/her wrist.

10. Challenging and Decision Making:

Challenges, difficulties, and mistakes are inevitable things in every journey to create a great project. For this project, the challenges start when the team wanted to decide which microcontroller is the best for the project, after some researches it turned out that Raspberry Pi is the one. after that the team started to search for the best model of the Raspberry, in the beginning, Raspberry Pi zero w was the choice because it is smaller and lighter than the others, so it can be set up easily on the glasses rather than hold it by hand, but then the team found out that the Raspberry Pi 3 has stronger processing power since it has quad core and it is faster than Raspberry Pi zero w. It also has a larger memory and extra connectivity. Since (OCR) technology will be used with OpenCV and Efficient and Accurate Scene Text Detector (EAST), the Raspberry Pi should be able to process multitasks properly, which is the case in Raspberry Pi 3. Another challenge is the glasses with build in camera, it turned out that this build in camera is not compatible with Raspbian OS, and it is only compatible with Windows and MacOS, so the team tried to install Windows IOT (operating systems from Microsoft designed for use in embedded systems.) on the Raspberry but it didn't work. To solve this problem, it's been decided to use a webcam that works perfectly with the Raspberry Pi.

11. Embedded System Code:

Two Raspberry Pi is used as a platform in the project to achieve a more convenient result for the users to reach the project goal. In order to control the whole embedded system code, the team decided to choose python 3 as the main language for both raspberry and use it to implement all the project functions. In the first Raspberry, python is used to implement three main functions which are text detection and recognition, text to voice conversion and text

translation. All the needed library is installed and used to perform the project tasks. For text detection and recognition, OpenCV OCR (Optical Character Recognition) is used with Tesseract. Tesseract is a very popular OCR engine and in order to make the python script able to communicate with Tesseract, Tesseract + Python bindings are installed. The process starts when the user presses the first button to check if the image is in the range of 40-150 from the camera using the Ultrasonic sensor. When the image is captured by the Webcam, apply OCR on the image and process it using OpenCV. To detect the text with high accuracy, OpenCV's EAST text detector is also used to recognize the text. After that, Tesseract is used to extract the text from the image. The second main function is text to voice conversion. Google Text-to-Speech (gTTS) interface is used to convert text to voice for python 3. The third main function is text translation. In order to perform this function, the following line is written in the python code (import google trans as gt) so the text can be translated from English to Arabic when the user pressed the button. Moreover, the required libraries for the button and Ultrasonic sensor such as (import RPI. GPIO, import time and import os) are all imported in the buttons code and Ultrasonic sensor code. For the second Raspberry, the main function is to recognize the hall and classes location using RFID sensor. In order to perform the task, there are many libraries are imported such as (import RPI. GPIO, from mfrc522 import Simple MFRC522) so whenever the user passes by the RFID tag, he can place the RFID reader and get the id that will determine the name of class or hall and then send it to the headphone that is attached to the raspberry using the gTTS interface in the RFID code.

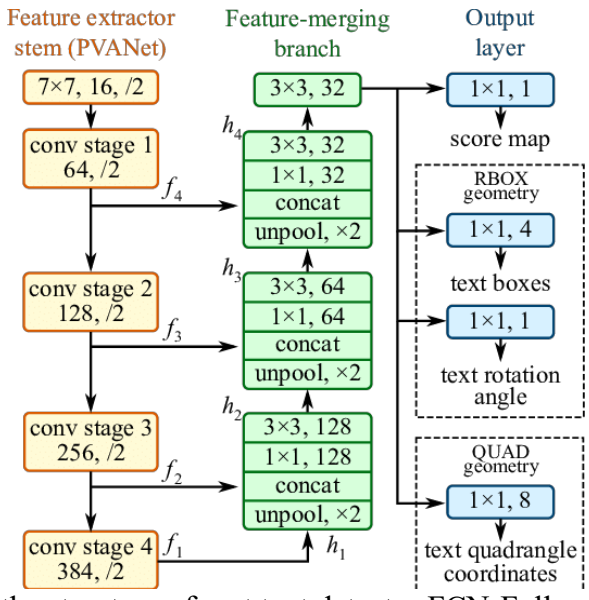


Figure 43: the structure of east text detector FCN-Fully convolutional Network

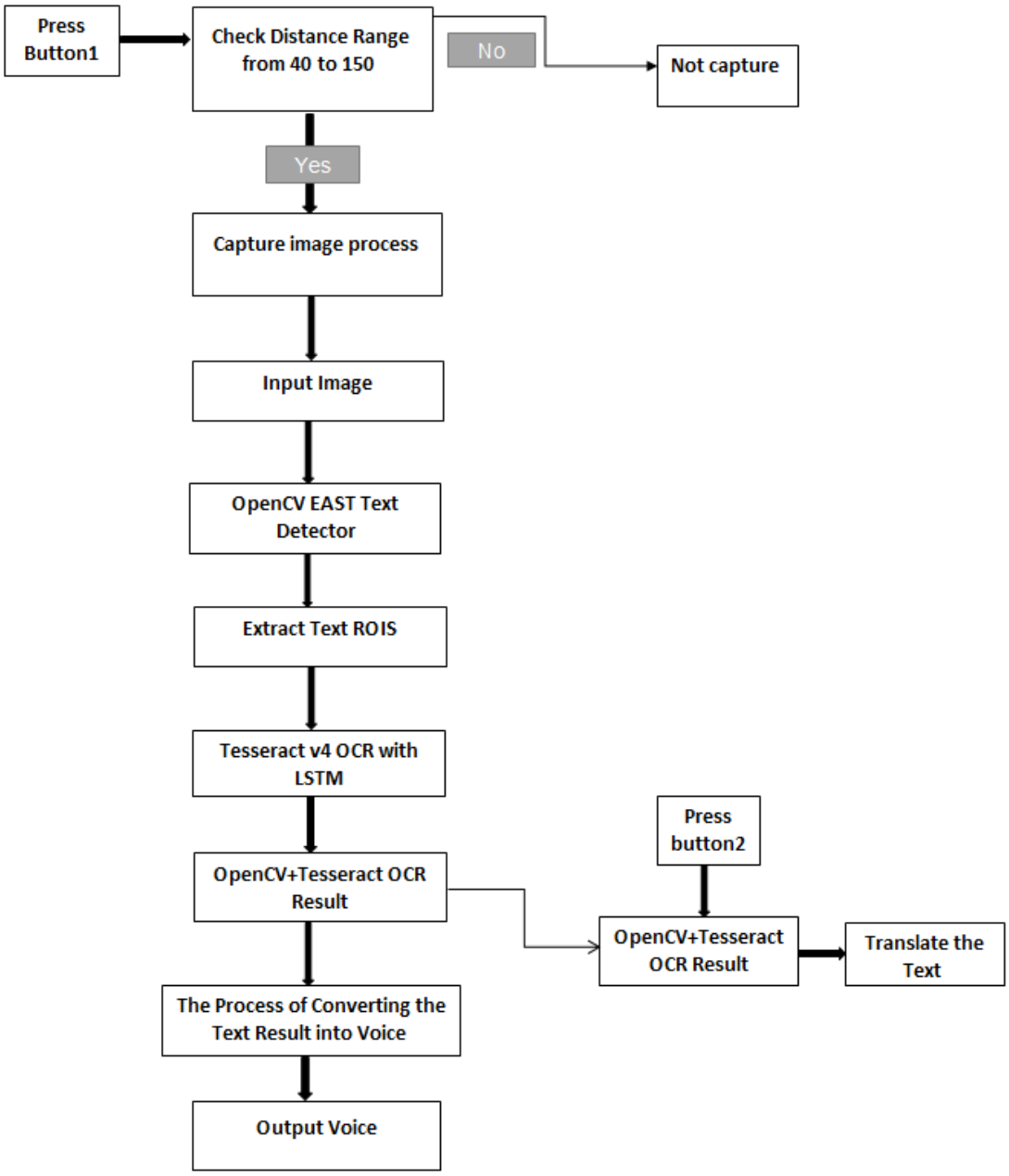


Figure 44: Flowchart demonstrates the control code

12. Prototype Development:

The team started working on the glasses frame that was provided from the university. when the team faced constraints then the team used Webcam and separate the Raspberry from the frame. The team decided to wear the Raspberry on the upper arm.

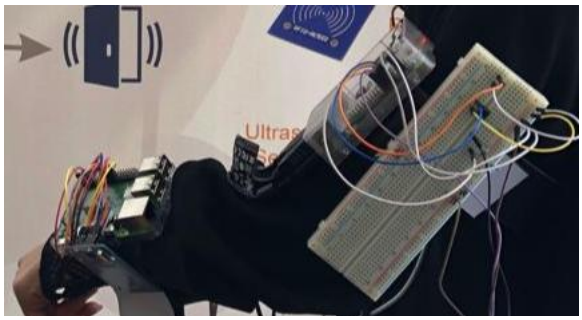


Figure 45.1: Final Prototype

Figure 45.2: Final Prototype

The team used other Raspberry to RFID sensor, so the user needs to pass hand to the door of the class or lab to scan the ID then receive a voice message to inform the user by the class number. To be more professional and easier, the team decided to wear the Raspberry on hand when the user wants to scan the ID. The user will wear the Raspberry Pi Model B on his/her wrist.

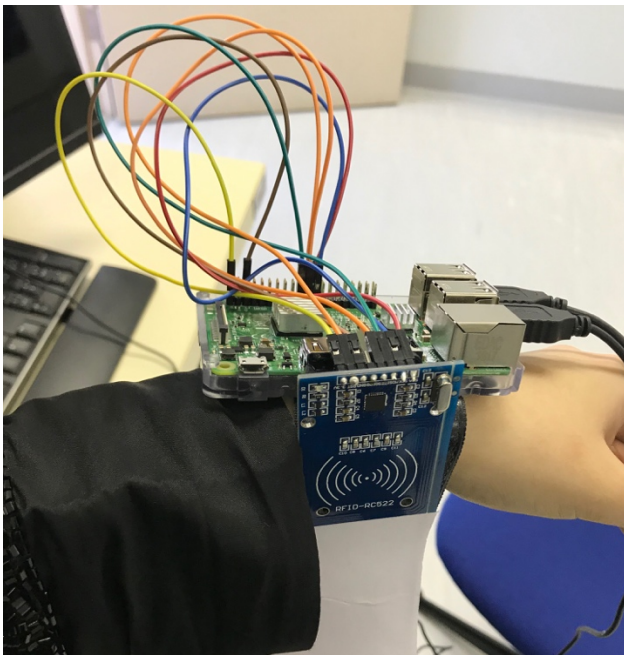


Figure 45.3: Final Prototype

13. Conclusion and Future Recommendations:

13.1. Conclusion:

Technology played a very important role in our life. We use it almost everywhere and every time. The distinct and quick development that we discover each day proof for us that there is no point to give up and struggle with our obstacle in life. Technology offers us a lot of significant solutions to our problems and disapplies. Our role is to use it properly to reach the success level that benefits individual, society and whole country as well.

13.2. Future Recommendation:

While the team members were working on the implementation, they thought of many ideas and improvements for the “Smart Glasses”. However, they wished they have more time and knowledge to do them. “Smart Glasses” can be improved in the future for blind people and people who have vision difficulties by adding new techniques. For instance, direction and warning messages to prevent expected accidents, messages to tell the user about the battery level, video detection to provide a full healthy life for people with vision difficulties, develop mobile application to control “Smart Glasses”, use 270 camera to have more wider view angle., provide the glasses with GPS notification and develop the glasses’ design to have little, small and light components so the user can wear it easily.

References:

[1] Wang, T., Wu, D. J., Coates, A., & Ng, A. Y. (2012, November). End-to-end text recognition with convolutional neural networks. In *Pattern Recognition (ICPR), 2012 21st International Conference on* (pp. 3304-3308). IEEE.

[2] Koo, H. I., & Kim, D. H. (2013). Scene text detection via connected component clustering and nontext filtering. *IEEE transactions on image processing*, *22*(6), 2296-2305.

[3] Bissacco, A., Cummins, M., Netzer, Y., & Neven, H. (2013). Photoocr: Reading text in uncontrolled conditions. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 785-792).

[4] Yin, X. C., Yin, X., Huang, K., & Hao, H. W. (2014). Robust text detection in natural scene images. *IEEE transactions on pattern analysis and machine intelligence*, *36*(5), 970-983.

[5] Neumann, L., & Matas, J. (2012, June). Real-time scene text localization and recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (pp. 3538-3545). IEEE.

[6] Neumann, L., & Matas, J. (2013). Scene text localization and recognition with oriented stroke detection. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 97-104).

[7] Zhou, X., Ylao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). EAST: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 5551-5560).

Appendix A

Foundation, R. P. (n.d.). Teach, Learn, and Make with Raspberry Pi. Retrieved from https://www.raspberrypi.org/

Mitchell, B. (2019, March 15). 802.11 WiFi Standards Explained. Retrieved from https://www.lifewire.com/wireless-standards-802-11a-802-11b-g-n-and-802-11ac-816553

Gudino, M. (2018, March 6). Raspberry Pi 3 vs. Raspberry Pi Zero W. Retrieved from https://www.arrow.com/en/research-and-events/articles/raspberry-pi-3-vs-raspberry-pi-zero-w

Savvides, L. (2017, February 15). This 'Star Trek'-like headset helps the legally blind see again. Retrieved from https://www.cnet.com/news/esight-video-glasses-restores-sight-legally-blind-star-trek-visor-headset/

News and blogs by author. (2019, April 30). Retrieved from https://www.abilitynet.org.uk/news-blogs/three-cool-smart-glasses-help-people-who-are-blind-or-have-sight-

Ray, K., & Ray, K. (2017, July 24). Why I'm Excited about the Future of Aira. Retrieved from https://medium.com/aira-io/why-im-excited-about-the-future-of-aira-79be882fc34e

Google. (2019, April 15). Retrieved from https://en.wikipedia.org/wiki/Google_

Eyesynth. (2019). Inicio - Eyesynth. [online] Available at: https://eyesynth.com/ [Accessed 99Jan. 2019].

Detection based on "ultrasonic"What is an Ultrasonic Sensor? (n.d.). Retrieved from https://www.keyence.com/ss/products/sensor/sensorbasics/ultrasonic/info/

Martin, T. (2016, May 20). How to setup Bluetooth on a Raspberry Pi 3. Retrieved from https://www.cnet.com/how-to/how-to-setup-bluetooth-on-a-raspberry-pi-3/

Rosebrock, A. (2018). Install OpenCV 4 on your Raspberry Pi - PyImageSearch. [online] PyImageSearch. Available at: https://www.pyimagesearch.com/2018/09/26/install-opencv-4-on-your-raspberry-pi/ [Accessed 2 Feb. 2019].

OpenCV. (2019, March 12). Retrieved from https://en.wikipedia.org/wiki/OpenCV

Tabbara, K. F., & Ross-Degnan, D. (1986, June 27). Blindness in Saudi Arabia. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/3712697

Raspberry Pi Rfid Reader- How To Integrate Rfid With Raspberry Pi

https://www.deviceplus.com/connect/integrate-rfid-module-raspberry-pi/

Hc-sr04 Ultrasonic Range Sensor on the Raspberry Pi

ModMyPi LTD - https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi

Opencv Ocr and Text Recognition with Tesseract

https://www.pyimagesearch.com/2018/09/17/opencv-ocr-and-text-recognition-with-tesseract/

Chaudhuri, A., Mandaviya, K., Badelia, P., & Ghosh, S. K. (2017). *Optical character recognition systems for different languages with soft computing*. Springer.

What Is a Radio Frequency Identification Reader (rfid Reader)? - Definition from Techopedia

https://www.techopedia.com/definition/26992/radio-frequency-identification-reader-rfid-reader

Static.raspberrypi.org. (n.d.). Retrieved from https://static.raspberrypi.org/

What Is Braille?
https://www.afb.org/blindness-and-low-vision/braille/what-braille

An Easier and Powerfulonline Pcb Design Tool
https://easyeda.com/

Appendix B

This section will show all the codes that have been used in the project:

Taking Picture Code:

```
import RPi.GPIO as GPIO
import time
import os
from PIL import Image
from gtts import gTTS
import googletrans as gt
import RPi.GPIO as GPIO
import time



GPIO.setmode(GPIO.BCM)

GPIO.setup(18, GPIO.IN, pull_up_down=GPIO.PUD_UP)

while True:
    input_state = GPIO.input(18)
    if input_state == False:
        os.system('./webcam2.sh')
        message='picture is taken';
        language = 'en'
        print(message)
        myobj = gTTS(text=message, lang=language,
slow=False)
        myobj.save("welcome.mp3")
        os.system("mpg321 welcome.mp3")

time.sleep(0.2)

```

Ultrasonic sensor Code:

```
import RPi.GPIO as GPIO
import time
import os
GPIO.setmode(GPIO.BCM)

TRIG = 23
ECHO = 24

print("Distance Measurement In Progress")

GPIO.setup(TRIG,GPIO.OUT)
```

```
GPIO.setup(ECHO,GPIO.IN)

GPIO.output(TRIG, False)
print("Waiting For Sensor To Settle")
time.sleep(2)

GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)

while GPIO.input(ECHO)==0:
      pulse_start = time.time()

while GPIO.input(ECHO)==1:
    pulse_end = time.time()

pulse_duration = pulse_end - pulse_start
distance = pulse_duration*17150
if distance >= 40 and distance <=150:

    distance = round(distance, 2)
    print("Distance:",distance,"cm")
    os.system('fswebcam -r 1280x720 --no-banner
image44.jpg')
GPIO.cleanup()
```

Text Detection and Recognition Code:

```
# USAGE
# python text_recognition.py --east
frozen_east_text_detection.pb --image
images/example_01.jpg
# python text_recognition.py --east
frozen_east_text_detection.pb --image
images/example_04.jpg --padding 0.05

# import the necessary packages
from imutils.object_detection import
non_max_suppression
import numpy as np
import pytesseract
import argparse
import cv2
import gtts
from gtts import gTTS
import os
import googletrans as gt


```

```
def decode_predictions(scores, geometry):
    # grab the number of rows and columns from the scores volume, then
    # initialize our set of bounding box rectangles and corresponding
    # confidence scores
    (numRows, numCols) = scores.shape[2:4]
    rects = []
    confidences = []

    # loop over the number of rows
    for y in range(0, numRows):
        # extract the scores (probabilities), followed by the
        # geometrical data used to derive potential bounding box
        # coordinates that surround text
        scoresData = scores[0, 0, y]
        xData0 = geometry[0, 0, y]
        xData1 = geometry[0, 1, y]
        xData2 = geometry[0, 2, y]
        xData3 = geometry[0, 3, y]
        anglesData = geometry[0, 4, y]

        # loop over the number of columns
        for x in range(0, numCols):
            # if our score does not have sufficient probability,
            # ignore it
            if scoresData[x] < args["min_confidence"]:
                continue

            # compute the offset factor as our resulting feature
            # maps will be 4x smaller than the input image
            (offsetX, offsetY) = (x * 4.0, y * 4.0)

            # extract the rotation angle for the prediction and
            # then compute the sin and cosine
            angle = anglesData[x]
            cos = np.cos(angle)
            sin = np.sin(angle)

            # use the geometry volume to derive the width and height
            # of the bounding box
            h = xData0[x] + xData2[x]
            w = xData1[x] + xData3[x]

```

```
            # compute both the starting and ending (x, y)-coordinates
            # for the text prediction bounding box
            endX = int(offsetX + (cos * xData1[x]) + (sin * xData2[x]))
            endY = int(offsetY - (sin * xData1[x]) + (cos * xData2[x]))
            startX = int(endX - w)
            startY = int(endY - h)

            # add the bounding box coordinates and probability score
            # to our respective lists
            rects.append((startX, startY, endX, endY))
            confidences.append(scoresData[x])

    # return a tuple of the bounding boxes and associated confidences
    return (rects, confidences)

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", type=str,
    help="path to input image")
ap.add_argument("-east", "--east", type=str,
    help="path to input EAST text detector")
ap.add_argument("-c", "--min-confidence", type=float, default=0.5,
    help="minimum probability required to inspect a region")
ap.add_argument("-w", "--width", type=int, default=320,
    help="nearest multiple of 32 for resized width")
ap.add_argument("-e", "--height", type=int, default=320,
    help="nearest multiple of 32 for resized height")
ap.add_argument("-p", "--padding", type=float, default=0.0,
    help="amount of padding to add to each border of ROI")
args = vars(ap.parse_args())

# load the input image and grab the image dimensions
image = cv2.imread(args["image"])
orig = image.copy()
(origH, origW) = image.shape[:2]

# set the new width and height and then determine the ratio in change
# for both the width and height
(newW, newH) = (args["width"], args["height"])
rW = origW / float(newW)
```

```
rH = origH / float(newH)

# resize the image and grab the new image dimensions
image = cv2.resize(image, (newW, newH))
(H, W) = image.shape[:2]

# define the two output layer names for the EAST detector model that
# we are interested -- the first is the output probabilities and the
# second can be used to derive the bounding box coordinates of text
layerNames = [
    "feature_fusion/Conv_7/Sigmoid",
    "feature_fusion/concat_3"]

# load the pre-trained EAST text detector
print("[INFO] loading EAST text detector...")
net = cv2.dnn.readNet(args["east"])

# construct a blob from the image and then perform a forward pass of
# the model to obtain the two output layer sets
blob = cv2.dnn.blobFromImage(image, 1.0, (W, H),
    (123.68, 116.78, 103.94), swapRB=True, crop=False)
net.setInput(blob)
(scores, geometry) = net.forward(layerNames)

# decode the predictions, then  apply non-maxima suppression to
# suppress weak, overlapping bounding boxes
(rects, confidences) = decode_predictions(scores, geometry)
boxes = non_max_suppression(np.array(rects), probs=confidences)

# initialize the list of results
results = []

# loop over the bounding boxes
for (startX, startY, endX, endY) in boxes:
    # scale the bounding box coordinates based on the respective
    # ratios
    startX = int(startX * rW)
    startY = int(startY * rH)
    endX = int(endX * rW)
    endY = int(endY * rH)

    # in order to obtain a better OCR of the text we can potentially
```

```
    # apply a bit of padding surrounding the bounding box -- here we
    # are computing the deltas in both the x and y directions
    dX = int((endX - startX) * args["padding"])
    dY = int((endY - startY) * args["padding"])

    # apply padding to each side of the bounding box, respectively
    startX = max(0, startX - dX)
    startY = max(0, startY - dY)
    endX = min(origW, endX + (dX * 2))
    endY = min(origH, endY + (dY * 2))

    # extract the actual padded ROI
    roi = orig[startY:endY, startX:endX]

    # in order to apply Tesseract v4 to OCR text we must supply
    # (1) a language, (2) an OEM flag of 4, indicating that the we
    # wish to use the LSTM neural net model for OCR, and finally
    # (3) an OEM value, in this case, 7 which implies that we are
    # treating the ROI as a single line of text
    config = ("-l eng --oem 1 --psm 7")
    text1 = pytesseract.image_to_string(roi, config=config)
    mytext = text1
    language = 'en'
    myobj = gTTS(text=mytext, lang=language, slow=False)
    myobj.save("hello.mp3")
    os.system("mpg321 hello.mp3")

    # add the bounding box coordinates and OCR'd text to the list
    # of results
    results.append(((startX, startY, endX, endY), text1))

# sort the results bounding box coordinates from top to bottom
results = sorted(results, key=lambda r:r[0][1])

# loop over the results
for ((startX, startY, endX, endY), text1) in results:
    # display the text OCR'd by Tesseract
    print("OCR TEXT")
    print("========")
```

```
    print("{}\n".format(text1))

    # strip out non-ASCII text so we can draw the text on the image
    # using OpenCV, then draw the text and a bounding box surrounding
    # the text region of the input image
    text1 = "".join([c if ord(c) < 128 else "" for c in text1]).strip()
    output = orig.copy()
    cv2.rectangle(output, (startX, startY), (endX, endY),
        (0, 0, 255), 2)
    cv2.putText(output, text1, (startX, startY - 20),
        cv2.FONT_HERSHEY_SIMPLEX, 1.2, (0, 0, 255), 3)

    # show the output image
    cv2.imshow("Text Detection", output)
    cv2.waitKey(0)
```

Text to Voice Code:

```
# Import the required module for text
# to speech conversion
from gtts import gTTS
import os
import googletrans as gt
import RPi.GPIO as GPIO
import time
from PIL import Image

GPIO.setmode(GPIO.BCM)

GPIO.setup(18, GPIO.IN, pull_up_down=GPIO.PUD_UP)

while True:
    input_state = GPIO.input(18)
    if input_state == False:
        mytext = 'moon shine'
        language = 'ar'
        print(mytext)
        myobj = gTTS(text=mytext, lang=language, slow=False)
        myobj.save("hello.mp3")
        os.system("mpg321 hello.mp3")
```

Translation Code:

```
# Import the required module for text
# to speech conversion
from gtts import gTTS
import os
import googletrans as gt
import RPi.GPIO as GPIO
import time
from PIL import Image

GPIO.setmode(GPIO.BCM)

GPIO.setup(18, GPIO.IN, pull_up_down=GPIO.PUD_UP)

while True:
    input_state = GPIO.input(18)
    if input_state == False:
        mytext = 'moon shine'
        language = 'ar'
        def speak(mytext,lang):
            print(mytext)
            myobj = gTTS(text=mytext, lang=language,
slow=False)
            myobj.save("hello.mp3")
            os.system("mpg321 hello.mp3")
        def translatetoarabic(mytext):
            if mytext!="":
                mygt=gt.Translator()
                translation=mygt.translate(mytext,dest='a
r')
                speak(translation.text,'ar')
        data=mytext
        translatetoarabic(data)

```

RFID Reader Code:

```
#!/usr/bin/env python

import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522

reader = SimpleMFRC522()

try:
        text = input('New data:')
        print("Now place your tag to write")
        reader.write(text)
        print("Written")
finally:
```

```
        GPIO.cleanup()
#!/usr/bin/env python

import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522

reader = SimpleMFRC522()

try:
        id, text = reader.read()
        print(id)
        print(text)
finally:
        GPIO.cleanup()
```

Appendix C

Below are the data sheets for all of the hardware component used in the tour-guide

A. HC-SR04 Ultrasonic Sensor

Module	HC-SR04
Operating Voltage	5V DC
Operating Current	2mA
Effectual Angle	<15 degree
Ranging Distance	2 cm - 400 cm
Resolution	1 cm

B. Raspberry Pi 3 B+

Processor	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory	1GB LPDDR2 SDRAM
Connectivity	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) 4 × USB 2.0 ports
Access	Extended 40-pin GPIO header
Video & sound	1 × full size HDMI MIPI DSI display port MIPI CSI camera port 4 pole stereo output and composite video port

Multimedia	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support	Micro SD format for loading operating system and data storage
Input power	5V/2.5A DC via micro USB connector 5V DC via GPIO header Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment	Operating temperature, 0–50°C
Compliance	For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry - pi-3-model-b+
Production lifetime	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.
weight	42 g
Dimentions	3.35 × 2.2 × 0.8

C. RFID Reader

Module	MFRC522
Supply Voltage	3.3V
Current	13-26mA

Operating frequency	13.56MHz
Read Range	3 cm (Approximately)
Dimensions	60mm × 39mm
Weight	15g

D. Power Bank 10000

Model	A1263
Capacity	10000mAh/36Wh
Voltage	5 V
Current	2 A
weight	6.4 ounces.
Dimensions	3.6x2.4x0.9

F. Raspberry Pi 3 B

Processor	Broadcom BCM2387 chipset, 1.2GHz Quad-Core ARM Cortex-A53 64-bit
Memory	1GB LPDDR2
Connectivity	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip, Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
GPIO Connector	Extended 40-pin GPIO header
Video & sound	HDMI (rev 1.3 & 1.4), Composite RCA (PAL and NTSC) Audio Output 3.5mm jack, HDMI, USB 4 x USB 2.0 Connector
GPU	Dual Core Video Core IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated Open VG, and 1080p30 H.264 high-profile decode, Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
SD card support	Micro SD format for loading operating system and data storage
Input power	5V/2.5A DC via micro USB connector 5V DC via GPIO header Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Power	Micro USB socket 5V1, 2.5A
Dimensions	85 x 56 x 17mm

Weight	15g