American University of Science and Technology
Faculty of Engineering
Department of Computer and Communications Engineering
Department of Mechatronics Engineering

CCE 499X: Computer and Communications Engineering Senior Design
MTE 499X: Mechatronics Engineering Senior Design

Internet of Blind (IoB)

By

Raed Diab
Jad Kabakibi
Mostafa Tormos
Rami Youssef

Project Advisor: Mr. Nashaat El Halabi

A Proposal Submitted to the Department of Computer and Communications Engineering
and to the Department of Mechatronics Engineering,
American University of Science & Technology,
in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science

Beirut, Lebanon
June-21-2022

This senior design report has been approved by the following committee:

_____ Nashaat El Halabi, M.S. (Project Director)

_____ Roger Achkar No. 1, Ph.D.

_____ Hassan Diab No. 2, Ph.D.

_____ Rida Nuwayhid No. 3, Ph.D.

_____ Michel Owayjan No. 4, M.E.

Investigators:
Raed Diab, Jad kabakibi, Mostafa Tormos, and Rami Youssef.

Project Director:
Mr. Nashaat El Halabi.

Abstract

Visually impaired people are people who have lost or were born without eyesight. A lot of products are designed in order to help visually impaired people live their life normally. However, some products are designed for a specific objective such as “Obstacle Detection” only. This fact motivated the creation of this product which includes many features that help blind people in various ways.

Blind people tend to face several difficulties in their everyday life. Therefore, an assistive product is created. This project helps blind people to move more freely and helps them do various difficult tasks that they might face.

The project consists of one product that acts as the eyes of the blind and much more. It takes orders and helps them fulfill their desired tasks. These features are dependent on each other to communicate and execute actions that suit the user the best. The evolution of technology such as “AI” and “5G” is used in the creation of this product in order to serve the user in the best way possible.

This product is expected to achieve many tasks that help blind people. It is expected to act as their eyes and help them in many ways. The major features of this project are: Obstacle detection, text scanning, and reading, taking orders from the user and achieving them, reaching the desired destination using Google Maps, recognition of the user's home, and saving its structure to help them freely move inside his home, reading of vital signs such as body temperature, blood pressure, pulse rate.

Moreover, the product has the access to call someone or some organization such as Red Cross if the user’s vital signs show any emergency case

Blind people unfortunately are not able to experience life fully. That’s why the IoB is created to aid them in doing their tasks and connect them to the outer world using one set for all.

Acknowledgments

We are extremely thankful to our advisor Mr. Nashaat El-Halabi, for his valuable suggestions and encouragement.

We also wish to extend our thanks to the Faculty members of the Engineering department for their compassion and unlimited support.

Finally, we wish to extend our sincere thanks to our parents for their compassion and unlimited support.

Table of Contents

List of Figures

List of Tables

List of Symbols / Abbreviations

I2C	Inter-Integrated Circuit
MFCC	Mel Frequency Cepstral Coefficients
LPC	Linear Predictive Coding
LPCC	Linear Predictive Cepstral Coefficient
LSF	Line Spectral Frequencies
DWT	Discrete Wavelet Transform
Seq2seq	Sequence to Sequence

Chapter 1
INTRODUCTION

Lack of vision has an outstanding impact on the lives of those who experience it. The complete loss or the deterioration of the eyesight can be felt as frightening and overwhelming, at the same time. Leaving those affected to wonder about their ability to maintain their life independently.

Blindness can overwhelm a person's quality of life, and mobility and has been connected to his/her falls, injury, and worsened status in domains spanning mental health and social function. knowing that surprising variables are likely to have a role in some of the negative consequences of vision impairment, statistics from visually impaired people show that vision has a major impact on health and social well-being.

In 2007, a study done by Rein and colleagues in the United States of America concluded that the quality of life for people declined dramatically after losing their eyesight. They became unhappy and started to have a medical condition called Depression. Another study made in 2015 in Korea by the Epidemiologic Survey Committee of the Korean Ophthalmologic Society concerning the chronic condition of blind people has shown significant results. They have discovered that on a scale of 1 to 14 - 1 being the lowest and 14 the highest-, blind people sadly scored an outstanding 14 score. This means that the quality of life of blind people is low and discouraging. [1]

A. Statement of the Problem

Safety is the state of being "safe", the condition of being protected from harm or other danger. Safety can also refer to the control of recognized hazards to achieve an acceptable level of risk management. Safety is considered one of the most important factors to be present in any given situation, and must always be taken into consideration.

In society, people with special needs require special help. They often find difficulties trying to achieve normal tasks. Sometimes visually impaired individuals feel a bit left out. Moreover, according to the European Foundation social exclusion (1995), social exclusion is the process by which individuals or groups are totally or partially excluded from full participation in the

society in which they live. But they face troubles due to inaccessible infrastructure and social challenges. One of the biggest challenges for a blind person, especially the one with complete loss of vision, is to navigate around places. Blind people roam easily around their house without any help because they know the position of everything in the house. The actual problem lies when they try to guide themselves outside their homes, where they might face multiple problems [2].

Safety precautions made to help blind people live normal lives and become independent are getting outdated. Inventions made to help blind people are limited. An example of an invention is the white cane. The white cane can help blind people but its function is limited. It can only help them by guiding them where to go and where not to, the small white ball at the end helps them identify if their surrounding is safe or not [3].

With time, technology is evolving. The products of technology are helping people in a great way, so humans refer most of the time to technology to do nearly any task in our life. Technological tools are fundamental for human beings; they contribute to generating autonomous individuals who can thrive in different fields in society. That is why the principle of autonomy goes hand in hand with the role of tools in society, such principle argues that "people are free and therefore can choose for themselves what is best", hence if each individual has the right tools, they will be able to make their own decisions.

The lack of tools for blind people is something that undermines their human rights since they lose the opportunity to integrate into educational institutions and thus be able to enter the professional field and have a good quality of life. Nowadays, educational institutions have implemented different types of tools to improve the academic level of their students, however, those tools have not been designed with a society with disabilities in mind [4].

B. Hypothesis or Key Questions

As seen in today's community, society is getting harder over time. Humans are becoming more initiative. Technology is spreading at a surreal speed and it's all in favor of humans. But sadly, not everyone is benefiting from this evolution. Blind people are members of society who might experience technology's evolution but not to the fullest. Technology is not benefiting them all the way. Blind people, who are considered disabled people, must be the first people to experience the evolution of technology. Due to their unfortunate disability, they are considered

the weakest people in society. Technology must be turned and focused toward them for them to experience life to the fullest.

Their disability is an issue that must be addressed seriously because it leads to many more questions that one can only find more if thought about them.

After tackling the importance of solving this problem, many key questions may be asked; Can people find a way to minimize the number of injuries caused that way? What can be done to provide a simpler and easier life for blind people? How can the evolution of technology help in the safety of the visually impaired? Can man and machine work together to grant a safer environment for blind people to live in? What can someone create do to boost the acceptance of blind people in society?

C. Proposed Solution

To help the visually impaired people, a decision was made to gather today's top technology and create a device that can help them overcome their disability. It is not sustainable that in the 21st century, there are still people who aren't getting the full experience and favor from technology.

That's why a device was created to aid them to perform their activities. It can assist them to overcome their daily obstacles. It can help them move more freely. As it also can boost their confidence.

Achieving this project will create a device that draws a direct connection between the visually impaired and machine. Create a device that boosts the safety of the visually impaired while living their everyday life. This device will make the visually impaired more accepted in society because they will be able to achieve more tasks than they used to, (knowing that some of them are essential).

The product will allow the user to comprehend life around them. They will be able to experience life more dynamically and live it in an upgraded manner. IoB will elevate the user's self-esteem and makes them raise their confidence while doing any task possible. Knowing that they have an advanced device backed by AI and IoT, it will make their life easier to its connection with the evolution of technology. By creating this device, a connection between

the man -in the need-and machine will be achieved so that they experience the true objective of the evolution of technology, which is the assistance of the man by technology.

D. Specific Objectives

The objective of this project is to create an invention that aids blind people in understanding their surroundings.

The first task of the project is to create a headset and attach to it an AI camera that scans the surroundings of the person. This gadget allows the camera to comprehend the location of the person so that it alerts him/her of any possible dangers around them. The second task is to design a sensor that will be attached to the earlobe. That sensor measures important factors in the blood circulatory system, such as blood pressure, number of heartbeats per minute, etc. The results of the measurements are shared with the AI to know if the user is in any critical condition concerning his/her health. The third task is to create an earpiece to communicate between the AI and the person wearing it. And lastly, Connect the blind person to the Red Cross in case of an emergency (an emergency can be recognized by the results of the earlobe sensor and its analysis by the AI).

E. Relevance of this Work

To help the visually impaired people, a decision was made to gather today's top technology and create a device that can help them overcome their disability. It is not sustainable that in the 21st century, there are still people who aren't getting the full experience and favor from technology.

That's why a device was created to aid them to perform their activities. It can assist them to overcome their daily obstacles. It can help them move more freely. As it also can boost their confidence.

Achieving this project will create a device that draws a direct connection between the visually impaired and machine. Create a device that boosts the safety of the visually impaired while living their everyday life. This device will make the visually impaired more accepted in society because they will be able to achieve more tasks than they used to, (knowing that some of them are essential).

The product will allow the user to comprehend life around them. They will be able to experience life more dynamically and live it in an upgraded manner. IoB will elevate the user's self-esteem and makes them raise their confidence while doing any task possible. Knowing that they have an advanced device backed by AI and IoT, it will make their life easier to its connection with the evolution of technology. By creating this device, a connection between the man -in the need-and machine will be achieved so that they experience the true objective of the evolution of technology, which is the assistance of the man by technology.

F. State-of-the-Art

In the field of helping blind people, many engineers and engineering companies tried designing and producing products that will help blind people.

Some products use signal processing only to help blind people avoid obstacles like the RFIWS.

Some other products are done to detect obstacles and inform the user about the obstacles in a more intelligent way such as the ELC, Sili Eye, Smart Cane, and Eye Substitution.

Some other products use more complex technologies such as AI to detect objects and text such as the Aira Smart Glasses, and the Finger reader.

Some other products uses also AI to assist the user with the voice commands such as the Ultra Ass Headset.

IoB uses the combination of "AI", "IoT", and "5G" technologies to help blind people by understanding their needs. it uses the embedded Intelligent Virtual Assistant to understand and inform them, the Object Detections System to help them in avoiding obstacles, and it uses the embedded Safety System to take care of them.

As it uses the advanced AI in its Intelligent Virtual Assistant, the 5G connection to connect the user to a very high-speed network, and the IoT technologies to connect all blind people in one metaverse.

Chapter 2
LITERATURE REVIEW

Several researchers and companies started researching this specific issue. In the following section, there will be stated multiple products which are similar to the selected solution used:

ELC

Figure 1. Existing Solution – ELC (source: https://rb.gy/n67r4f)

The proposed electronic long cane (ELC), done by A.R. Garcia, in Vale of Itajai University, on April 10, 2012, is primarily based totally on haptics technology for the mobility resource for the blind humans. ELC is an improvement of the conventional cane so that you can offer accurate detection of the items which might be across the user. A small grip of the cane proven within side the figure above includes an embedded digital circuit that consists of an ultrasonic sensor for the detection process, a micro-motor actuator because of the feedback interface, and a 9 V battery as an energy supplier. This grip is capable of coming across the barriers above the waistline of the blind man or woman. Tactile feedback through a vibration could be produced as a caution to a near impediment. The frequency of the comments could be extended because the blind man or woman receives towards the obstacle.

The figure beneath suggests how the ELC ought to assist the blind humans in detecting the obstacle above their waistline, that's taken into consideration as one of the motives of severe damage for folks who are visually impaired or blind.

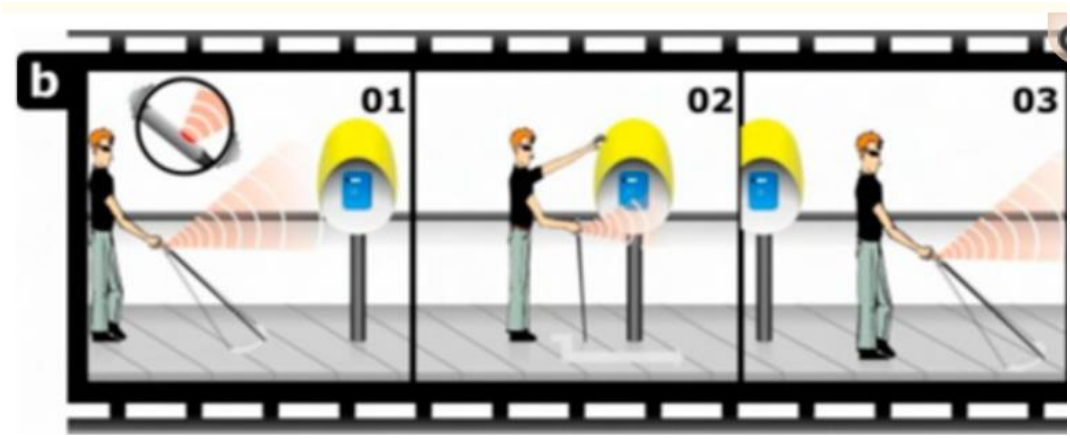


Figure 2. Existing Solution - Obstacles detecting over waistline (source: https://rb.gy/awo7zx)

The ELC has been examined on 8 voluntarily blind people. Physical obstacles, data boundaries, and cultural boundaries are the primary examined classes for the boundaries classification. The outcomes have been categorized primarily based totally on a taken quiz by the blind people who used the tool. The outcomes confirmed the performance of the tool for physical boundaries detection above the waistline of the blind individual. However, the tool facilitates a blind individual simply in detecting boundaries; however, now no longer within side the orientation function. So, the blind individual nevertheless wishes to discover his/her route himself/herself and is based on the subculture cane for the navigation as proven within side the figure above [5].

Sili Eye

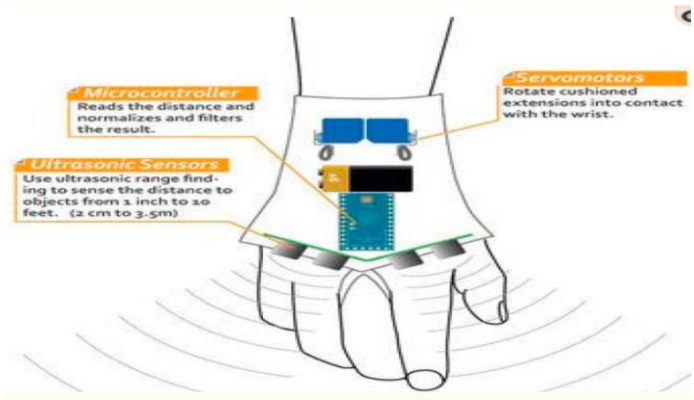


Figure 3. Existing Solution - Sili Eyes (source: https://rb.gy/pm6hp2)

By adapting GSM and GPS coordinator, B.R Prudhvi in 2013 WCCIT, delivered an assistive navigator for blind people. It allows the customers to detect their current location, hence, navigating them using haptic feedback. In addition, the user can get information about the time, date, or even the color of the gadgets in front of him/her in audio format. The proposed tool is hooked up inside a silicon glove to be wearable as proven within side the figure above. The prototype of the proposed tool is primarily based totally on a microcontroller that's 32-bit cortex-M3 to govern the complete system, a 24-bit color sensor to apprehend the colors of the objects, a light/temperature sensor, and SONAR to stumble on the gap among the item and the user. The gadget helps a hint keyboard the use of Braille method to go into any statistics. After the consumer chooses the favored destination, he/she could be directed to the use of a MEMS accelerometer and magnetometer via the road. The commands could be dispatched via headset this is linked to the tool through an MP3 decoder. The consumer could be notified via way of means of SONAR at the detected distance between the user and the closet obstacle. In case of emergency, the current location of the disabled user could be sent through SMS to a person whose phone number is provided by the user with the use of each technology GSM and GPS. The layout of the gadget is pretty comfortable as it's far wearable. Also, the features which can be furnished to the user can supply him/her extra experience with the surrounded environment. However, the gadget wishes a power tracker to maintain the music of the battery. The emergency useful resource isn't always effective because the user wishes to press the button in case of an emergency and she/he has to go into the phone numbers of his/her relatives, which is probably a limiting factor [6].

Ultra Ass Headset

Figure 4. Existing Solution - Ultra Ass headset (source: https://rb.gy/4isjb8)

An assistive headset was proposed by S.Aymaz and T.Cavdar, published by IEEE,2016 39th International Conference of TSP, to navigate visually-impaired human beings based on the ultrasonic distance measurement technology. The figure beneath illustrates the layout of the ultrasonic headset which incorporates 4 ultrasonic sensors; sensors cover every membrane to detect left and right obstacles. DYP-ME007 is the selected form of the ultrasonic sensor for distance measurement. ISD2590 recording storage is used to document the recommended directions. There are six recorded messages, the chosen information is based on the intersection of ultrasonic sensors in case there's an obstacle.

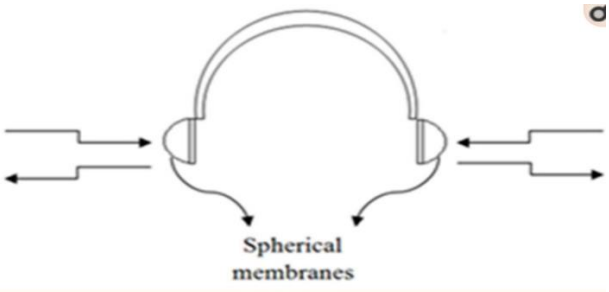


Figure 5. Existing Solution - Layout of the 4 Ultrasonic (source: https://rb.gy/2o3vnb)

The features of this device are as follows: every sensor has an ID that's produced as a binary code. Once the sensor gets a mirrored image of the ultrasonic wave, an output of "1" can be sent to the microcontroller, otherwise "0" can be sent. Using the binary code, the microcontroller can decide which sensor is the receiver. Based on that, the audio remarks can be played back to the user. the Figure above shows the finished layout of the proposed device.

The device is a good energy-saving solution. However, the device is restrained withinside the directions it offers to the consumer. Six directions cannot be enough sufficient to manual the consumer interior and outdoors. Furthermore, the headset obscures the outside noise, which blind people depend upon to make their choice in case the device fails [7].

Smart Cane

Figure 6. Existing Solution - Smart Cane (source: https://rb.gy/qxhqmx)

Mohd Helmy Abd Wahab, Central Michigan University's understudies on October 24, 2011, concentrated on the advancement of the Smart Cane item for recognizing the objects and producing exact navigation instructions. The plan of the Smart Cane is displayed in the figure above. It is a versatile gadget that is furnished with a sensor framework. The framework comprises ultrasonic sensors, microcontroller, vibrator, buzzer, and water identifier to guide visually-impaired people. It utilizes servo engines, ultrasonic sensors, and fluffy regulators to distinguish the hindrances before the client and afterward give guidelines through voice messages or hand vibration. The servo engines are utilized to give an exact position criticism. Ultrasonic sensors are utilized for distinguishing the hindrances. Subsequently, the fluffy regulator can give precise choices dependent on the data got from the servo engines and ultrasonic sensors to navigate the user.

The result of the Smart Cane relies on gathering the above data to deliver sound messages through the speaker to the user. Moreover, hearing impaired individuals have extraordinary vibrator gloves that are furnished with the Smart Cane. There is a particular vibration for each finger, and everyone has particular importance. The Smart Cane has accomplished its objectives in identifying the objects and obstructions and creating the required feedback. As displayed in the figure, the Smart Cane is handily conveyed and effectively twisted [8].

Eye Substitution

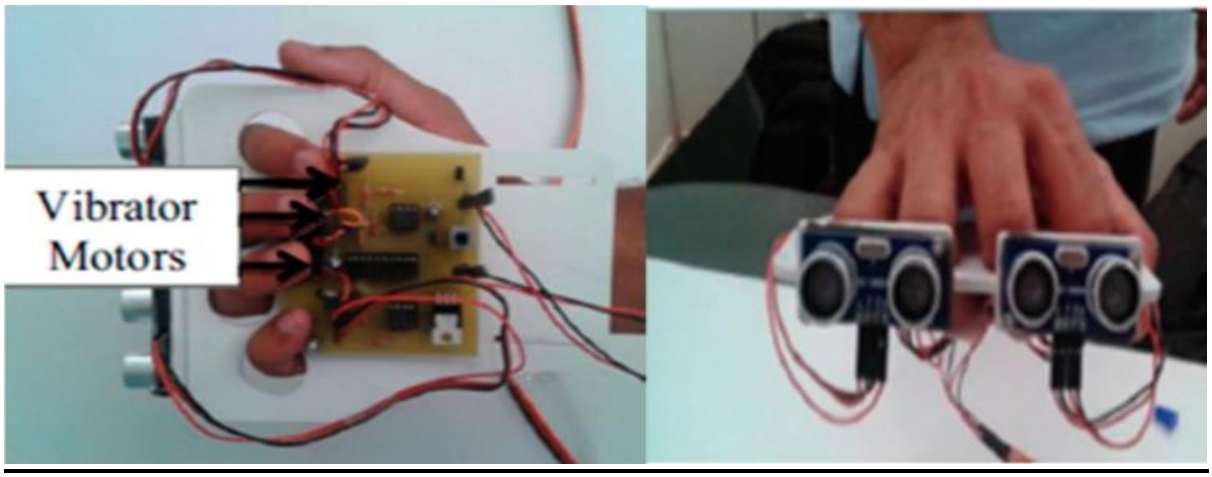


Figure 7. Existing Solution - Eye Substitution Design (source: https://rb.gy/o18syp)

Sachin Bharambe, in 2013, Texas Instruments India Educators' Conference, fostered an installed gadget to go about as an eye replacement for the vision disabled individuals that aids in directions and navigation as displayed in the above figure. Fundamentally, the installed gadget is a TI MSP 430G2553 miniature Regulator (Texas Instruments Incorporated, Dallas, TX, USA). The creators carried out the proposed calculations utilizing an Android application. The job of this application is to utilize GPS, further developed GSM, and GPRS to get the area of the individual and produce better bearings. The implanted gadget comprises two HC-SR04 ultrasonic sensors (Yuyao Zhaohua Electric Appliance Factory, Yuyao, China), and three vibrator motors.

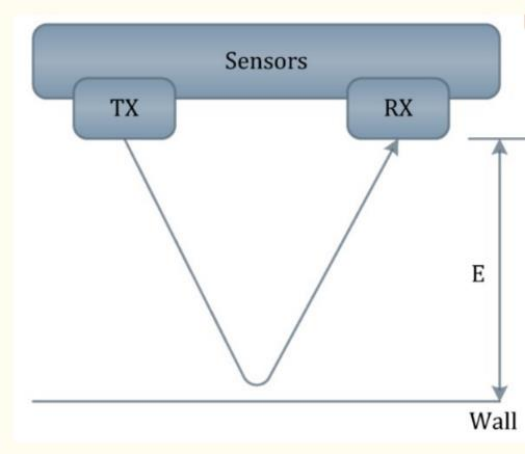


Figure 8. Existing Solution - Eye Substitution Design - Sound reflection back (source: https://rb.gy/regknr)

The ultrasonic sensors send a succession of ultrasonic pulses. If the hindrance is identified, then, at that point, the sound will be reflected in the receiver as displayed in the figure above. The miniature regulator processes the readings of the ultrasonic sensors to actuate the motors by sending pulse width modulation. It likewise gives a low power utilization.

The design of the gadget is light and extremely advantageous. Moreover, the framework utilizes two sensors to conquer the issue of restricted cone points as displayed in the figure beneath. Along these lines, rather than covering two territories, the ultrasonic gadgets cover three territories. This doesn't just assist in distinguishing obstacles, yet additionally in finding them. Nonetheless, the design could be better assuming the creators didn't utilize the wood establishment that will be conveyed by the user more often than not. Moreover, the framework isn't dependable and is restricted to Android devices [9].

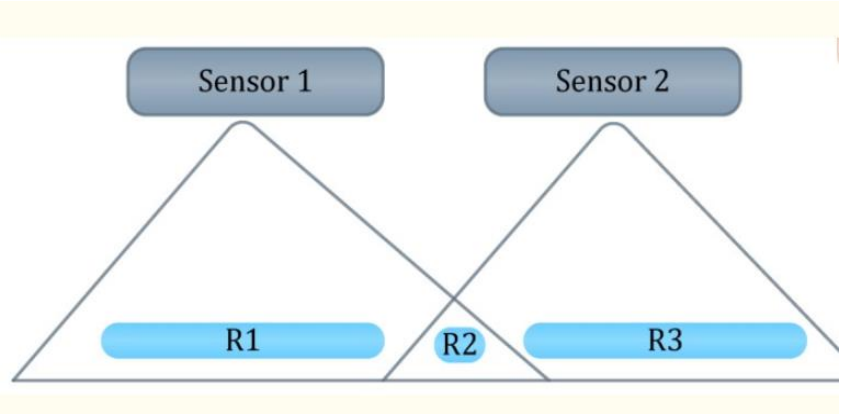


Figure 9. Existing Solution - Eye Substitution Design - restricted cone point issue (source: https://rb.gy/llixix)

Fusion of Artificial Vision and GPS (FAV&GPS)

Figure 10. Existing Solution - FAV and GPS Design (source: https://rb.gy/lxstu3)

An assistive gadget for blind individuals was published in IEEE by Adrien Brilhault, Slim Kammoun, Olivier Gutierrez, Philippe Truillet, and Christophe Jouffrais in the 2011 fourth IFIP International Conference on New Technologies, Mobility and Security to further develop a mapping of the user's location and positioning the surrounding objects using two functions that are: based on a map matching approach and artificial vision. The first function helps in locating the necessary item just as permitting the user to give directions by moving her/his head toward the objective. The subsequent function aids in the automatic detection of visual aims. As displayed in Figure over, this gadget is a wearable gadget that is mounted on the user's head, and it comprises two Bumblebee sound system cameras for video input that are introduced on the cap, GPS beneficiary, earphones, receiver, and Xsens Mti GPS beacon for movement detecting. The framework processes the video transfer utilizing the SpikNet recognition algorithm to locate the visual features that handle the 320 × 240 pixels picture.

Because of the absence of the accessibility of a few data about the consistency of passerby portability by commercial GIS, this framework maps the GPS signal with the adjusting GIS to gauge the user's present situation as displayed in the Figure beneath. The 3D objective's position is determined utilizing lattices of focal points and stereoscopic change. In the wake of identifying the user and target positions, the vision specialist sends the ID of the objective and its 3D directions [10].

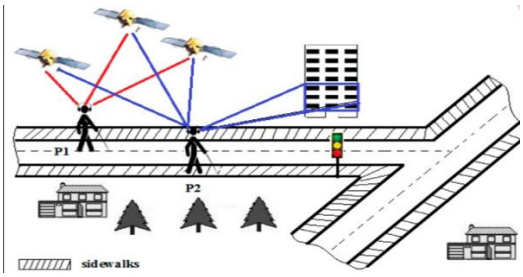


Figure 11. Existing Solution - FAV and GPS Design - GPS mapping of the user (source: https://rb.gy/q4eqdj)

RFIWS

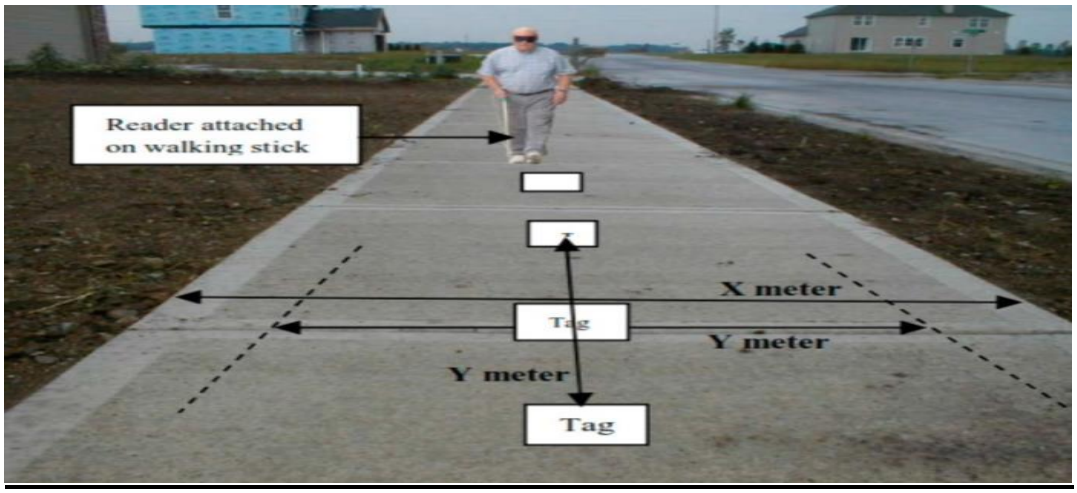


Figure 12. Existing Solution – RFIWS (source: https://rb.gy/lr0aak)

A Radio Frequency Identification Walking Stick (RFIWS) was designed by M.F Saiid in 2009, 5th International Colloquium on Signal Processing & its Applications, to assist blind people in navigating on their walkway. This framework helps in recognizing and computing the surmised distance between the walkway line and the visually impaired individual. A Radio Frequency Identification (RFID) is utilized to move and get data through a radio wave medium. RFID tag, reader, and middle are the principal parts of RFID innovation.

Various RFID labels are put in the walkway with the thought of an equivalent and explicit distance between one another and the RFID reader. The RFID will be associated with the stick to recognize and handle received signals. Sounds and vibrations will be delivered to inform the user of the distance between the line of the walkway and himself/herself. Stronger sounds will be created as the user draws nearer to the line. the Figure above shows the distance of recurrence recognition (Y) and the width of the walkway (X). Each label should be tried independently because of various scopes of identification.

RFID innovation has an ideal reading capacity between the labels and readers that makes the gadget solid in the detection of location [11].

Finger reader and Eye Ring

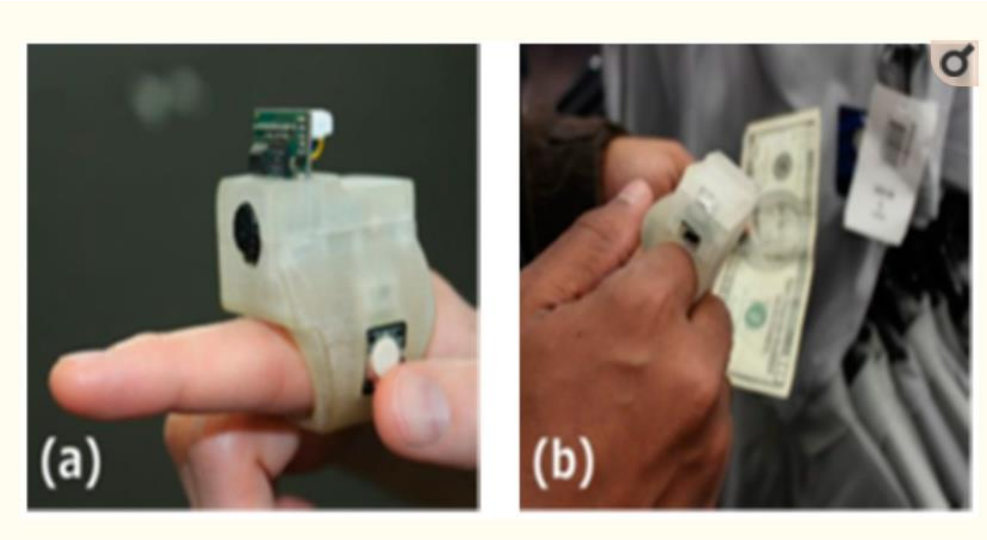


Figure 13. Existing Solution - Finger Reader and Eye ring (source: https://rb.gy/vteap4)

A strong perusing answer for blind individuals called Finger Reader was presented by Roy Shilkrot at,2013,4th Augmented Human International Conference, to help handicapped individuals in reading printed texts with real-time response. This gadget is a wearable gadget on the forefinger for close-up filtering. Thus, the gadget examines the printed text one line at that point, then, at that point, the response comes in tactile feedback and audio format. Finger Reader is continuous work to Eye Ring, which is for identifying a specific object once at a time by pointing and afterward scanning that item involving the camera on the highest point of the ring as displayed in the Figure above.

In this plan, two vibration engines with extra multimodal feedback, the double material case for more solace around the finger, and high goal video transfer are the growing of the Finger Reader gadget as displayed in the Figure underneath. The haptic feedback was given to direct the user to where he/she should move the camera.

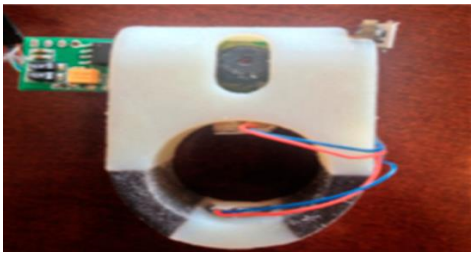


Figure 14. Existing Solution - Finger Reader and Eye ring - Reader design (source: https://rb.gy/aouzsv)

The group utilized Text Extraction Algorithm that is incorporated with Flite Text-To-Speech and "ORC". The proposed calculation removes the printed text however close-up camera. Then, at that point, it coordinates the pruned bends with the lines. The copied words will be ignored by a 2D histogram. From that point onward, the calculation will characterize the words from characters and send them to ORC. Those distinguished words will be saved in a layout as the user keeps on examining. Consequently, those words will be followed by the calculation for any match. The user will get sound and haptic feedback at whatever point he/she derails the current line. Moreover, the user will get signals through the haptic feedback to illuminate her/him about as far as it goes assuming the system didn't observe any more printed text blocks. the figure underneath shows the extraction and recognition interaction of the system.

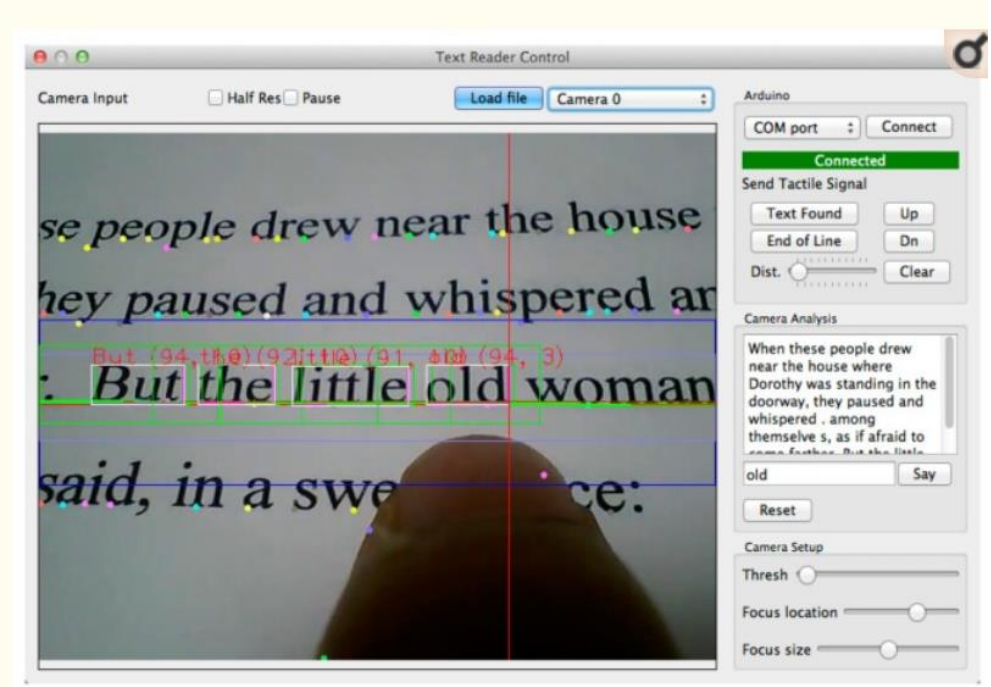


Figure 15. Existing Solution - Finger Reader and Eye ring - Text reader (source: https://rb.gy/h4co1z)

The gadget was tried on four clients after individual preparation which kept going for 1 h. The input of the clients showed that the haptic feedback was more effective than the sound reaction with respect to the direction. Moreover, there was a long stop between each word which confounds the user with respect to what he/she ought to do later. Be that as it may, the possibility of the system is an incredibly strong perusing answer for blind individuals [12].

Aira Smart Glasses

Figure 16. Existing Solution - Aira smart glasses (source: https://rb.gy/y09ign)

The glasses integrate a 120-degree wide-angle camera so guides can gain a fuller picture of a user's surroundings and won't have to instruct them to point their head in a different direction quite as much. It's powered by what the startup calls the Aira Horizon Controller, which is just a repurposed Samsung smartphone that powers the device in terms of computing, battery, and network connection. The controller is appropriately controlled entirely through the physical buttons and also can connect to a user's smartphone if they want to route controls through the Aira mobile app.

Aira has built a service that puts a human assistant into a blind user's ear by beaming live-streaming footage from the glasses camera to the company's agents who can then give audio instructions to the end-users. The guides can present them with directions or describe scenes for them. It's the combination of high-tech hardware and highly attentive assistants.

The company charges based on usage; $89 per month will get users the device and up to 100 minutes of usage. There are various pricing tiers for power users who need a bit more time. [13]

Chapter 3

MATERIALS AND METHODS

A. Alternative Design Approaches

- Hardware:

This device is used to take action with respect to the user's needs by using a virtual assistant, it will also help the user detect objects in front of him/her by using the ultrasonic sensor, and will continuously measure his/her heartrate using the heart rate sensor.

This device could be done in two ways. Firstly, by using Arduino AT-Mega Microcontroller, GSM module or Wi-Fi module, Ultrasonic sensor, Heart rate sensor, Microphone, Speaker, GPS module, Battery, and charger port. Secondly, by using Raspberry pi, Ultrasonic sensor, Heart rate sensor, Microphone, Speaker, GPS module, Battery, and charger port.

In all of the scenarios, Arduino and Raspberry PI are used to process the commands and take actions with respect to the reading of the sensors, the GSM or WIFI modules are used to connect the system to the internet, and the Mic is used to change the user's sound signal into an electronic signal to fed the Arduino, the Speaker is used to change the electronic signal from the Arduino to a sound signal to feed the user, the Heartrate sensor is used to measure user's heart rate, the Ultrasonic sensor is used to measure the distance between the user and the object in front of him/her, the Battery is used to turn on the system without being plugged into an AC source, and the Charger Port is used to charge the battery of the device.

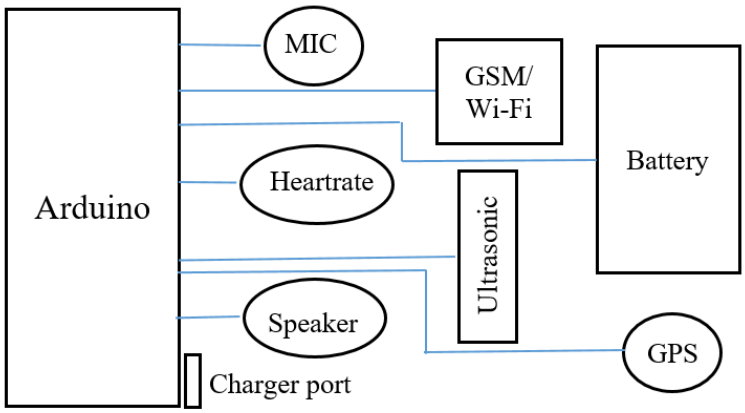


Figure 17. First Alternative Solution

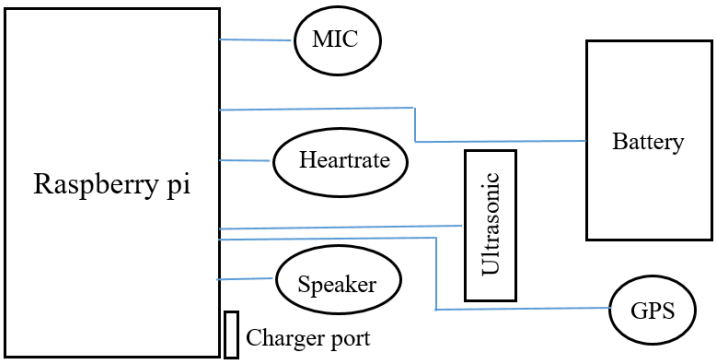


Figure 18. Second Alternative Solution

- Programming:

The programming language that will be used in the first device will be Arduino language, this is because all elements are controlled via Arduino Micro Controller, and when using these types of controllers, the only way is to use the Arduino default language which is the C++ language. Where the variables, inputs, and outputs pins should be defined before the setup function. However, in the setup function, the types of used pins and objects are defined and initiated, which runs only once. Where the loop function is where it runs infinitely and is used for updating or reading inputs.

However, the programming language that will be used in the second device will be Python language, this is because all elements are connected to the raspberry pi, and this type of controller could be programmed by python. Where using GPIO library to define the pins of the input/output and the whole program could be run in any way.

- Technical Experience

Implementing one of the mentioned alternative devices will make the system inaccurate.

For the First device, if Arduino is used, so only defined commands will be responding, in which the blind person should memorize all needed commands, thus this makes the system not interactive and it will be acting like a machine. Also, only Ultrasonic detection is allowed, this is because Arduino does not support an operating system so it won't have the requirement to detect live images.

Using a GSM module is a not good idea this is because the user will be holding 2 SIM cards, one in his/her phone and the other in this device.

Using a wired charger will decrease the efficiency of this product, the blind person can’t see the cable and the port to plug it.

For the second device, it might be done using raspberry pi but with the same features as the Arduino one, so the blind person should memorize all needed commands, thus this makes the system not interactive and it will be acting like a machine. Also, if using only an Ultrasonic sensor for detection, this will make the system accurate and can’t perform many tasks like live image processing for detecting traffic lights, streets, crosswalks, cars, and people.

B. Selected Design Approach

The IoB system consists of 1 wearable device and 1 standalone device, the wearable device consists of 3 sub-systems, the first one is the Virtual Assistant, the Object Detection and Avoidance, and the Safety System. However, the second device consists only of 1 system which is the charging system.

The Virtual Assistant will be used to recognize the voice of the user and take action for the detected voice, this detection process should be done intelligently, so that the user will be able to speak in any form he/she wants and the system should always work properly by running a specific action that the user requests like assisting him/her with the directions to arrive at a specific destination, turning on/off the music, detecting objects in front of him/her, telling him/her news, telling him/her date and time, and telling him/her facts and jokes.

The Object Detection and Avoidance will be used for detecting live frames around the user, so it will guide him/her to avoid obstacles, fires, cars, people, or even traffic lights. Also, this system gave the user the ability not to be fooled by people through the recognizing text feature, so that it will be recognizing the text, the money, or even people's face signs in front of the user.

The Safety system consists of 2 main parts heart rate detection and falling detection. Where the heart rate detection is used to detect if the user’s heart rate is above or below the accepted threshold and the falling detection is used to detect if the user was fallen and in both cases, it will be informing the competent authorities.

The charging system is used to charge the product wirelessly, and this is done to make the user compatible when charging this device.

- Technical Capabilities

This system consists of 1 wearable device that should be worn on the head, in which this product will be connected to the internet by using the User's phone internet.

In addition, this system uses a deep learning algorithm to detect the voice of the user, and since the detected voice could be any command, so Natural language processing will be used to find the weighted words in the detected command, then the system should take action with respect to the user needs.

Moreover, the system uses Deep Learning with Convolutional Neural Networks, to process live frames around the user to be able to guide him/her and assist them to avoid obstacles.

The programming language that will be used, is python this is because it has a lot of libraries dedicated to traditional and advanced Machine Learning and it is an open-source language.

- Cost and Experience

This system will be somehow costly, but it provides guidance and help a blind person, by detecting live frames around the user to avoid obstacles and to inform the user of the place around him. Also, it takes action with respect to the user's needs, so it could recognize text, and money, send emails and chats or even call someone.

Concerning the experience, this device will be built in a smart way which leads to being easily managed by a blind person like using wireless technology in order to make it easy to be used with such types of clients. Also, Using NLP leads to letting the user be comfortable with his/her way of speaking, so they will not be forced to use specific commands.

- Development Time

The prototype requires some time to be manufactured, but developing more from it, will not require that time, this is because the circuit, the design, and the codding will be all ready, which in this, it leads to a quick process of developing such a product because the simply same algorithm, same model, same circuit, same body, same logic, and even same components will be used.

C. Design Specifications

Design specifications are detailed information about the characteristics of your project. They are used to set the criteria you need to meet. They can include hardware physical size, power consumption, computing speed, cost, etc.

The system is a 1 wearable device and 1 standalone device, the wearable device consists of 2 sub-systems, the first one is the Virtual Assistant and the second one is the object detection and avoidance. However, the second device consists only of 1 system which is the charging system.

The whole device consists of a raspberry pi microcontroller, an Arduino mini pro microcontroller, a Heartrate sensor, an accelerometer and gyroscope breakout module, a microphone, a speaker, a GPS module, a camera, an Ultrasonic sensor, a wireless receiver charging, and a Battery. However, the standalone device consists of an Arduino mini microcontroller and a wireless transmitter charger.

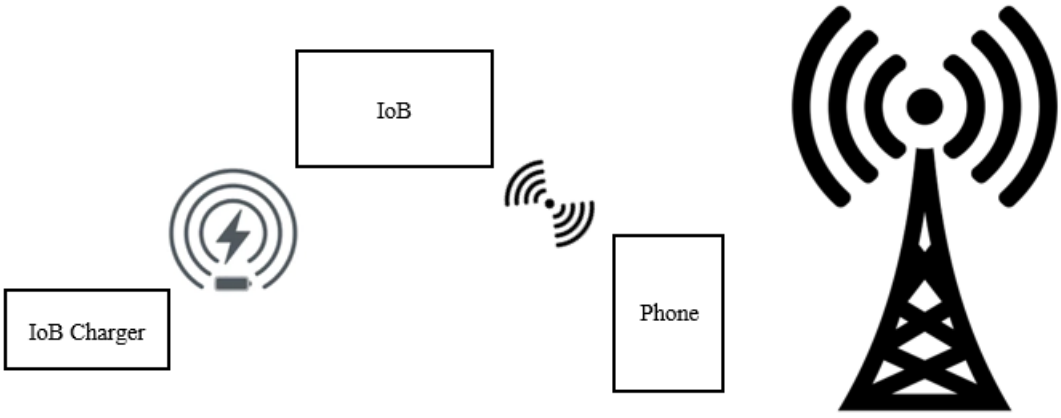


Figure 19. Selective Design Specifications

D. Multiple Realistic Design Constraints

- Economic Constraints

There are some economic constraints when dealing with electronics components in Lebanon, as Lebanon nowadays is suffering from a dollar rate which leads to facing some constraints.

- Manufacturability Constraints

Lebanon is a very poor country in electronics components, and with manufacturers, this leads to facing some manufacturability problems when buying components from outside Lebanon. As Buying components from outside of Lebanon can't be afforded, this is because dealers there couldn't be trusted and the components are so expensive.

- Environmental Constraints

Lebanon is a country that faces a lot of network issues, till now Lebanon is only implemented by not pure 4G, which is the worst scenario, thus using the internet provided here by the phone will be very slow and has a very high latency.

E. Materials

The wearable device consists of a raspberry pi (4 GB Ram), Arduino (mini pro), AI camera (OV5647), Microphone (LD3320), Speaker, GPS (NEO–6M), Heart rate detector (RS 330), accelerometer and gyroscope module (MPU6050), Buzzer, wireless charger receiver (9v, 2A), and Battery (9v, 2A).

The Wireless Charger Device consists of a Transformer(DB-16OV), Diodes, Capacitors (0.01 uf and 470 uf), Regulator (IC 7809), and the Wireless Receiver Module.

- Background Theory:

➢ Raspberry Pi 4 Model B

Raspberry Pi 4 Model B is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation Raspberry Pi 3 Model B+ while retaining backward compatibility and similar power consumption. For the end-user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems.

Figure 20. Selected Solution - Raspberry PI 4B (source: https://rb.gy/asap71)

- Arduino Pro Mini

This board was developed for applications and installations where space is premium and projects are made as permanent setups. Small, available in 3.3 V and 5 V versions, powered by ATmega328P.

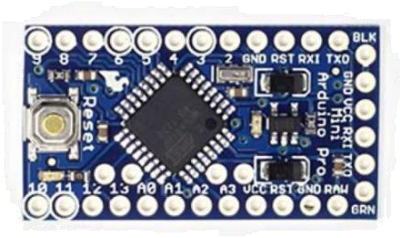


Figure 21. Selected Solution – Arduino Pro Mini (source: https://rb.gy/s2sr4o)

- Camera

The sensor itself has a native resolution of 5 megapixels and has a fixed focus lens onboard. In terms of still images, the camera is capable of 2592 x 1944 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video.

Figure 22. Selected Solution - Camera OV5647 (source: https://rb.gy/xdvyic)

- Microphone

A microphone, colloquially called a mic or mike is a device or a transducer that converts sound into an electrical signal. In which it has 95% accuracy in voice detection.

Figure 23. Selected Solution - Microphone (source: https://rb.gy/omgxaq)

- Speaker

Speakers are made up of a cone, an iron coil, a magnet, and a housing (case). When the speaker receives electrical input from a device, it sends the current through causing it to move back and forth. This motion then vibrates the outer cone, generating sound waves picked up by our ears.

Figure 24. Selected Solution - Speaker (source: https://rb.gy/bgqzvb)

- GPS Module

The NEO-6M GPS module is a GPS receiver that can locate all locations on Earth as it can track approximately 22 satellites. It consists of a high-performance u-blox 6 positioning engine. Measuring 16 x 12.2 x 2.4 mm, its compact architecture along with its low power consumption makes it a good choice for IoT projects. Overall it is a good cost-effective GPS receiver.

Figure 25. Selected Solution - GPS NEO-6M (source: https://rb.gy/dj2j8e)

- Heartrate Sensor

The heartbeat is measured in beats per minute or bpm, which indicates the number of times the heart is contracting or expanding in a minute. the working of this Heartbeat Sensor is a Photo plethysmograph. According to this principle, the changes in the volume of blood in an organ are measured by the changes in the intensity of the light passing through that organ.

Figure 26. Selected Solution - RS 330 (source: https://rb.gy/o5lhmk)

- Accelerometer

MPU6050 is a Micro Electro-mechanical system (MEMS), it consists of a three-axis accelerometer and a three-axis gyroscope. It helps us to measure velocity, orientation, acceleration, displacement, and other motion-like features.

Figure 27. Selected Solution - MPU6050 (source: https://rb.gy/ozqela)

- Buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short).

Figure 28. Selected Solution - Buzzer (source: https://rb.gy/7ye0fg)

- Battery

A lithium-ion battery or Li-ion battery is a type of rechargeable battery in which lithium ions move from the negative electrode through an electrolyte to the positive electrode during discharge and back when charging. Li-ion batteries use an intercalated lithium compound as the material at the positive electrode and typically graphite at the negative electrode.

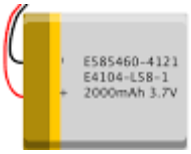


Figure 29. Selected Solution - Battery (source: https://rb.gy/o2qhuk)

- Wireless Transmitter/ Receiver

Wireless charging modules use an electromagnetic field to transfer energy between two objects. This is usually done with a charging station. Energy is sent through an inductive coupling to an electrical device, which can then use that energy to charge batteries or run the device.

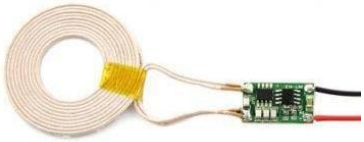


Figure 30. Selected Solution - Wireless Transmitter Reciver charger (source: https://rb.gy/ntz4kc)

- Transformer

A transformer is an electrical device that, by the principles of electromagnetic induction, transfers electrical energy from one electric circuit to another, without changing the frequency.

Figure 31. Selected Solution - Transformer (source: https://rb.gy/jxwpz3)

- Diode

A diode is a two-terminal electronic component that conducts current primarily in one direction (asymmetric conductance); it has low (ideally zero) resistance in one direction, and high (ideally infinite) resistance in the other.

Figure 32. Selected Solution - Diode (source: https://rb.gy/6jcmcd)

- Capacitor

A capacitor consists of two metal plates separated by a dielectric. The dielectric can be made of many insulating materials such as air, glass, paper, plastic, etc. A capacitor is capable of storing electrical charge and energy. The higher the value of capacitance, the more charge the capacitor can store.

Figure 33. Selected Solution - Capacitor (source: https://rb.gy/dkruqr)

- Regulator

7809 Voltage regulator is a type of self-contained fixed linear voltage regulator integrated circuit. The IC belongs to the 78xx voltage regulator family

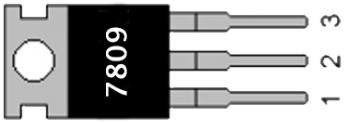


Figure 34. Selected Solution - Regulator (source: https://rb.gy/pdkene)

F. Methods

- Hardware

As discussed earlier the system consists of 2 devices, the wireless charger, and the head wearable devices.

In the Wearable device, the Mic, the Camera, and the Speaker are directly connected to the raspberry pi. However, the Heartrate sensor, the Ultrasonic sensor, the Accelerometer sensor, the Buzzer, the Battery, and the Wireless charger are directly connected to the Arduino Pro-Mini, and the Arduino board is connected to the Raspberry pi with an I2C connection.

In the Wireless Charger device, a transformer is connected directly to a 220V AC source, the 4 Diodes are connected in a bridge connection, some capacitors are connected in parallel at

the output of the bridge, and the regulator IC is connected in series with the capacitors, and the Wireless transmitter is connected after the regulator in parallel with the capacitors.

The Raspberry Pi will act as the master controller, in which it runs the Virtual Assistant and the Object Detection Algorithms, it also will be contacting aid agents with respect to the signal received from the Arduino microcontroller.

The camera will be used to capture live images transform them into digital signals and fed them to the raspberry pi, the microphone will be used to detect the audio of the user by changing the vibration signal into an electric signal and feeding it to the raspberry pi, the speaker will be converting the electric signals from raspberry pi to a vibration signal and fed it to the user, the GPS module will be used to track user's location and fed it to the raspberry pi.

The Arduino board acts as a slave where it will be used to analyze some analog signals and convert them to digital signals to be able to feed them to the raspberry pi throughout the ISP protocol (to connect the pins through the MISO, MOSI, SCL, and SKL). Also, it continuously read the battery level connected to it to inform the user when to charge it.

The Heartrate sensor will continuously read the user's heart rate and fed it to the Arduino board, (then the Arduino should check if the heart rate was above or below the acceptance criteria, and it will send a digital signal to the raspberry pi), the accelerometer will be used to continuously checking user's geometry and fed it back to the Arduino (then the Arduino should check if the user fallen, it will send a digital signal to the raspberry pi), the ultrasonic sensor is used to continue sending and receiving the sonic signal and fed it back to the Arduino (then the Arduino should check if the calculated distance was less than a threshold, this means there is an object in front of him/her so it will turn on the buzzer with respect to the distance).

The wireless charger device consists of an electronic circuit that converts the AC signal to a DC 9V signal. In this circuit the transformer is used to step down the voltage from 220 AC volts to 16 AC volts, then the diodes were connected to a bridge connection to convert these 16 AC volts to 16 DC volts, then some capacitors were used to filter out the signal, the IC 7809 was used as a voltage regulator, so it will regulate the output of the filters from 16 pure DC volts to 9 pure DC volts. Lastly, the wireless transmitter sensor is used to convert the electric 9 DC V signal to a magnetic signal to be transmitted wirelessly.

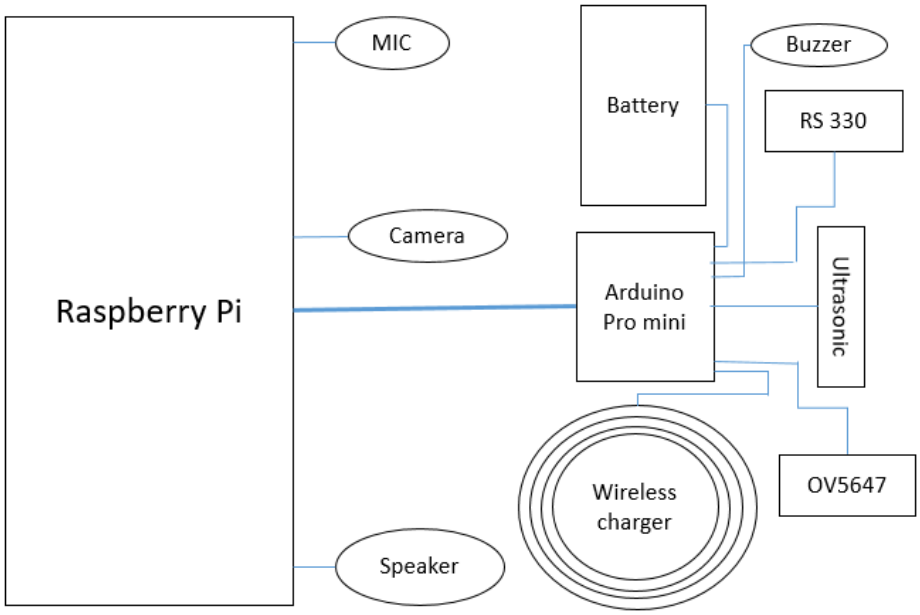


Figure 35. Selected Solution – IoB Circuit

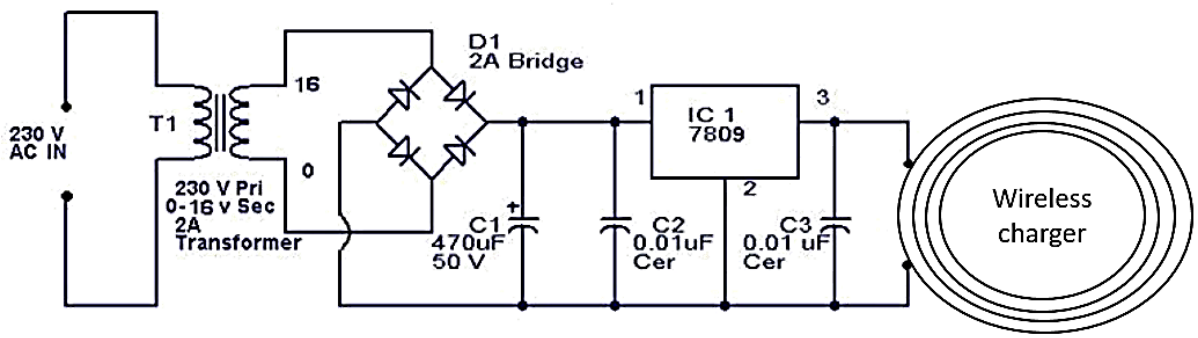


Figure 36. Selected Solution – IoB Charger Circuit

- Software

➢ Virtual Assistant

The Virtual Assistant that was deployed in this project, was divided into 3 main parts the Speech to Text converter, the Text Text, and the Text Speech converter.

The Speech to Text converter was used to transform the user's voice to text to be able to analyze it, then the Text to Text was used in the raspberry pi to extract Intents and Entities from the text and run some actions related to extracting data to feed the user needs, lastly, the

Text to Speech converter was used to transform back the Text returned from Raspberry pi to Speech to feed the user and to inform him/her.

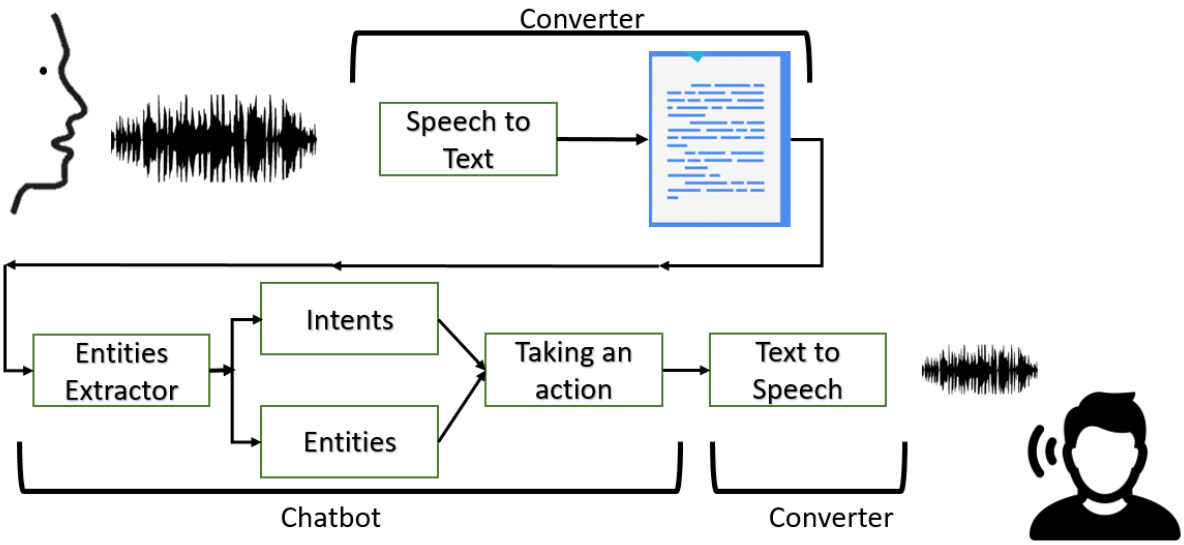


Figure 37. Selected Solution – Virtual Assistant

In the Text to Speech and Speech to Text converters, Google API was used to request the conversion of audio to text and vice-versa from Google's Speech to Text Converter.

Google's Speech to Text Converter takes the audio, split it over small windows, then extracts the MFCCs, LPCs, LPCCs, LSFs, and the DWTs coefficients from each window, then it combines all extracted coefficients in a 1D array, then using a Neural Network Model which consists of an Encoder and a Decoder (Seq2Seq Model) it returns the highest probable word from its defined vocabulary.

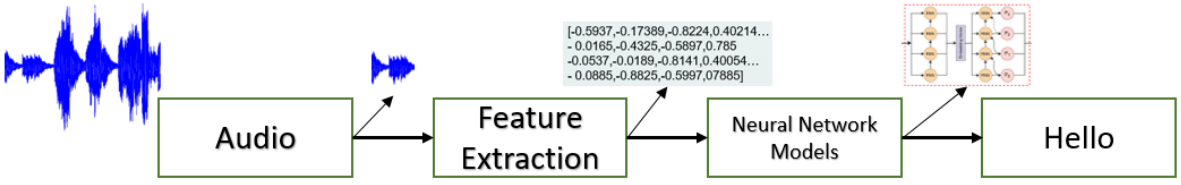


Figure 38. Google 's Speech to Text Overview

In the Text to Text Converter, the RASA framework was used to extract Intents and Entities from the provided text, then run some actions with respect to the extracted data.

RASA framework is an open-source framework that works at level 3coversational AI it consists of 4 main parts Intent, Entities, Responses, and Actions. Where all are defined with respect to the developer's needs. The Intent is what the user is implying (Hello, Good morning, Good Evening... All are considered as a Greeting Intent), and the Entities is the data that could be extracted from the user's message (Take me home → home is considered an extracted Entity), the Actions is to inform RASA how to Reply, and the Responses is what Rasa should reply (How I can help you).

RASA uses a predefined Embedding model, in which words are presented as embedding vectors with respect to the semantic and syntactic meaning of the word. Moreover, this Rasa model has been trained over 100 Billion words and achieved an accuracy greater than 97%.

To the RASA model works properly with respect to the project's needs, it should be trained on how to extract data and how to reply. To train the RASA model 5 files should be modified (NLU, Domain, Rules, Stories, and Actions files).

NLU file is used to label the data that RASA should be trained on, this data should be in sentence form and the labeling criteria label the Intent and the Entities in each sentence used for training.

Domain file is used to define all used Intents, Entities, Responses, and Actions.

The rules file is used to define a link between the intent and the action that could not be changed (always when the user greets the Chatbot, the Chatbot should reply with a greet).

Stories file is used to train the model on predicting the next intent from the user (if the user greets the Chatbot, how probable he/she will ask for a direction).

Actions file is used to run actions when needed; for example, if the user asks for information from google, RASA can't directly reply to the user however it should go to google to retrieve information about user needs and then reply with the found information. Here RASA uses this class to go outside of the scope. However, the RASA Custom class has some limitations because it is running on RASA's server, so it might be able to retrieve information from google but it can't run a direction class on its server; for this problem, RASA custom class in this project was used just to extract Intents and Entities from the text and return them with a response equal to None. Thus, at the backend, if the response was None, this means that the reply from RASA was not a response but it was Intents and Entities instead.

After Modifying the above 5 classes with respect to the project's needs, RASA Chatbot and RASA Custom Class should be running on 2 different servers on the end device. In this project, a decision was made to run the Chatbot on the localhost port 5000 and the Custom class on the localhost port 5500.

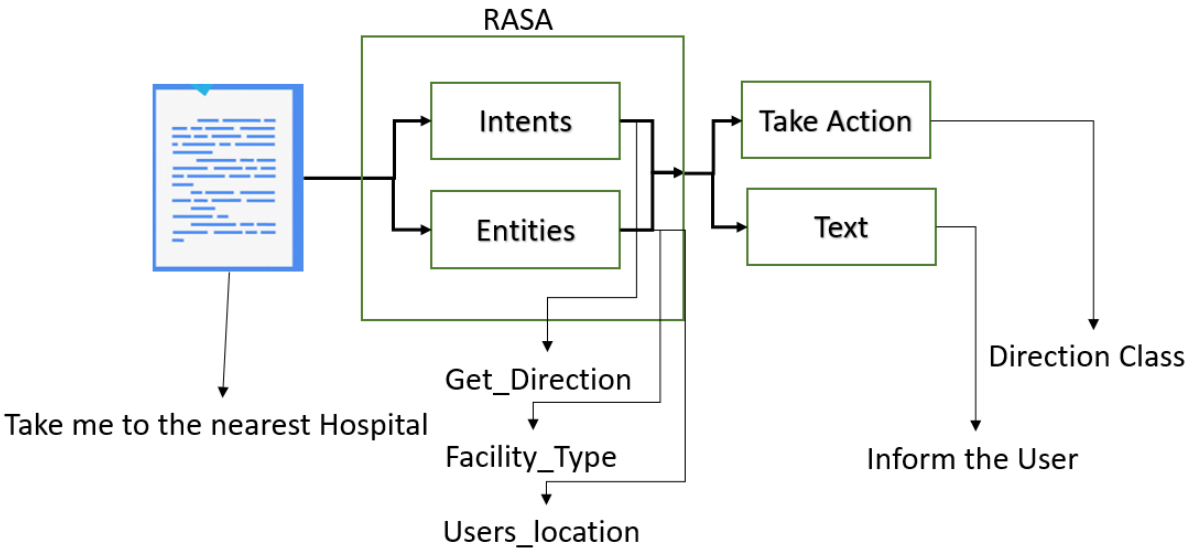


Figure 39. RASA Model performance after training it

After extracting the Intents and the Entities from the provided text, an action should be running with respect to the extracted data. This is why the Backend Classes were defined.

In the Email class, the Smtplib library was used to send and receive emails.

In the Date class, the Datetime library was used to get the actual date and time.

In the Facts class, the RandFacts library was used to return a random fact.

In the Jokes class, the pyjokes library was used to return a random joke.

In the News class, a request should be made over the newsapi.org to get back the top 5 news of the actual date.

In the Weather class, a request should be made over the Weatherapi.com to get back the information and the temperature of the actual date.

In the Location class, a request should be made over googlemaps.com to get back the latitude and the longitude of the user's desired location.

In the Users Location class, the latitude and the longitude of the user's location will be retrieved using the GPS module.

In the Web class, 2 main functions were defined. The first one was used for requesting information from Wikipedia.com (like what is an iPhone) and the second one was used for Google.com (like whom is Elon Musk), in here the defined request will try requesting one of the top 5 websites returned by Google (some of the sites could not allow to give information back using an API).

In the Music class, the YouTube server was used because it doesn't need any credentials. YouTube doesn't provide an API to allow developers to access it and run any video using the request, so the Web Scrapping technique was used; as a first step a chrome drivers.exe file was placed in the same directory of the code, then initially the code will start by opening the chrome file, finding for the X-Path of the search bar, typing YouTube.com, finding the X-Path of the search bar on YouTube, typing the requested song (name, artist, or music only), clicking on the first video, and then clicking on the play button of the video.

In the Direction class, the first step was done is returning the User's location (using the User's location class) and the desired location of the user (using the Location class), then requesting googlemaps.com with the source and the destination latitudes and longitudes, the google maps will be returning the routes needed starting from the user's location to his desired destination, this reply was in the form of a dictionary that consists of the needed routes for each step, the time needed to reach each step and the distance between the actual location and the second step. Then, after getting these results, Google Text speech API was used to inform the user of the required message (go forward and in 20 min return to left). But, since google maps don't provide live tracking, a checking function was also defined in this class; the checking function will be called after half of the time needed to reach the second step has been passed. Moreover, when calling it, it will use the User's location class to get the actual location of the user, then it will request Googlemaps.com to get the information for reaching the same next step; after getting the information from Googlemaps.com, this function will be checking if the distance to reach the same step decrease or increased; if the distance decrease, this means that the user is going right, otherwise the user is going in the wrong direction.

In the Speech_Text class, using Google's Text to Speech API, this project will be informing the user of the output of each class. And, using Text to Speech API, this project will be able to identify what the user is saying.

In the RASA class, a request should be made over the RASA Chatbot localhost server in order to retrieve the extracted Intents and Entities from the provided sentence.

After creating all of the needed classes, one class is still missing which is the IOB class, the IoB class will be used to link all classes together and instantiates an object from each class.

- Object Detection

The object detection system was divided into 2 main systems, the Ultrasonic system, and the Camera system. Both of them have the same target which is detecting objects, and they were used to detect these objects more accurately.

The Ultrasonic detection was running on the Arduino board with a Buzzer, which continuously sends and receives an ultrasonic signal. The Arduino program will then calculate the distance in front of the user from the time needed to receive the signal that was sent by the sensor, then after getting the distance if it was less than the threshold, the Buzzer will be beeping with respect to the distance.

The Image detection was running on the Raspberry Pi, where it continuously captures a frame and processes it using the OpenCV library, then it calculates the estimated distance between the user and the object in front of him/her, then uses the Google's Text to Speech class it will inform the user if he/she should move to the left, the right, the back, or to complete in his way.

- Safety system

The safety system is divided into 2 main parts, Heart rate detection, and Falling Detection. Since both systems were running on Arduino board so both of them were written using the Arduino language.

The heartrate and the falling detection system were running on the same program since the connection between the Arduino and the Raspberry pi was ISP so the wire library was used.

Using the analog read function, the program continuously read the heartrate sensor pin, then checks the reading if it was greater than the 100 or less than 60 as thresholds it will send a 5v digital signal using the I2C protocol to the Raspberry Pi using the wire address 44 else it will send 0v signal. Moreover, using the CurieIMU.h library, the program will also continuously check if the falling was detected and it will also send a 5v or 0v signal to the Raspberry Pi but using another wire address 45.

On the Raspberry Pi side, a python program was written, that will continuously read the wires 44 and 45, and if there is a 5v or 0v signals, both will be added to a list; if 3 times over a range of time the read value was 5v for any of the sensors, the Raspberry Pi will be sending an email to the specialized associations.

G. Cost Analysis

Table 1. Cost description.

Item No.	Description	Value	Quantity	Unit Price	Total Cost
1	Raspberry Pi	4B	1	$ 250	$ 250
2	Arduino	Pro Mini	2	$ 5	$ 10
3	Microphone	LD3320	1	$ 2.5	$ 2.5
4	Speaker	-	1	$ 0.5	$ 0.5
5	Camera	OV5647	1	$ 10	$ 10
6	Ultrasonic	-	1	$ 0.5	$ 0.5
7	Heartrate	RS 330	1	$ 0.5	$ 0.5
8	Accelerometer	MPU6050	1	$ 3.5	$ 3.5
9	Buzzer	-	1	$ 0.5	$ 0.5
10	Battery	9v,2A	1	$ 10	$ 10
11	Wireless charger	9v,2A	2	$ 7	$ 14
12	Capacitor	0.01 uF	2	$ 0.15	$ 0.3
13	Capacitor	470 uF	2	$ 0.15	$ 0.3
14	IC	7809	1	$ 5	$ 5
15	Transformer	220 - 16	1	$ 5	$ 5
16	Diodes	-	4	$ 0.25	$ 1
17	3D printing	-	1	$ 600	$ 600
17	Labor Cost[1]			$	$ 6480
				Total	$ 7393.6

[1] Labor Cost: 4 Engineers * 900$/month * 20% Full-Time * 9 months = 6480$

Chapter 4
SOCIETAL ISSUES

The most important issue regarding scientific inventions is the social aspect. It relates to accessing rights and gaining the trust of users. This part will be discussing all social issues and illustrates their benefits to society.

A. Ethical

In the journey of life, the ultimate goal is to reach the real divine truth, which is to become the embodiment of God. This continuous process requires integrity, honesty, frequent self-reflection, and most of all love.

In the field of scientific research, both innovation and inventions are critical to society, and society implicitly tests scientists and researchers to be ethical and honest.

B. Social

This product is much concerned with those who lost their most precious gift, their sight. Since they are not able to see anymore, using the virtual assistant, this product helps them to reach their destination, it instructs them and gives them the right direction they have to follow. Moreover, it warns them if there's an obstacle in their way.

This product helps blind people to trust themselves, to become independent and free in their movement. In other words, it replaces their eyes and guides them during their movement.

C. Economic

The product consists of electronic components that were shipped from outside the country. Since they are not locally manufactured, and with respect to the dollar and its fall compared to the Lebanese currency and because of the economic crisis that Lebanon is facing, this product may be a bit expensive.

D. Health and Safety

The product takes into consideration the safety of the user and his/her satisfaction with using it. It is safe with no risks that may cause harm for the person who uses it.

Instead, it gives him/her precise and accurate instructions, guides him/her step by step, and tells him/her how to do things independently.

The headsets are safe, they don't cause any harm to the user's eyes, on the contrary, it is hard to break and protect his eyes from dust and other harmful materials. The headsets are made of safe materials so that the person doesn't feel any pain even when they are used for a long time. They are soft and give clear instructions as if the user has a nice fellow to guide him/her on his/her way.

E. Manufacturability

After building the prototype of this system, and the provision of the needed components, it can easily be manufactured. The programming algorithm will be done to be used the same for all products, the outer body design will be done and saved to be able to use it, and the circuits needed will be ready with us. Thus, this will make the system well defined and will facilitate the manufacturing process.

F. Sustainability

The product is of great maintenance. The headsets hold a camera that contains a memory and a battery that can also be charged. In addition, the headsets contain a microphone and a speaker that help the user communicate efficiently with the virtual assistant. All of the parts are made of high-quality components which make their life span longer. So, this product is a lifetime product.

The product can work over a 3G, 4G, and 5G network, which leads it to be a very robust system that can work in the country. (nowadays the worst country has a 3G network)

G. Environmental Impact

Like most of the products in any industry, this product involves substances like plastic and other materials that can be harmful to the environment if it was left in nature. The

manufacturing of this product will release toxic substances that can harm nature and the atmosphere. However, the substances that the product is made of can be recycled instead of being dumped in nature, and this decreases the bad environmental impact.

H. Usability

This system will be built in a way to make it very easy to be used since the aim of this product is to serve blind people.

Using wireless technology in charging makes the system more expensive but improves usability, where the user will not face problems while charging any of the used devices instead of using the wired charger. Also, when the system is placed on the flat wireless transmitter charger it will notify the user by saying “The device is charging” and then it will add the percentage of its battery.

Moreover, this system can simply be placed on the head, and when a motion is detected the device will be turned on automatically which will decrease the power consumption in case it is not used.

I. Lifelong Learning

During the implementation of this device, the team members will learn a lot of experience because of its complexity. As, it consists of a deep learning algorithm for recognizing the speech of the user, a natural language processing for processing the text and its sentiments, a neural network algorithm for detecting places, obstacles, and more features, and uses Arduino as an analog to digital converter and for some processing requirements. Thus, the gained experience will be in hardware, and software implementation.

Chapter 5
RESULTS AND DISCUSSION

A. Testing and Validation

To validate and test this project, many tests were done

Starting with voice recognition, the AI was spoken to using a microphone. The user talked to the AI agent and commends it to do certain tasks. When asking “how are you?” The AI replied with its answer. The user asked the agent multiple questions. For example, how is the weather today? The answer is projected to the user using the earphone attached to the headset.

Another test was the AI camera, the user pointed the camera in a room that includes multiple objects. The camera scanned the objects of the room and recognized them, and when asked by the user what is in front of him/her, the AI responded to his answer with the objects that are placed in front of him/her.

Concerning the rs220 sensor, after its placed on the ear and turned on the device, the sensor tested the blood pressure and when asked, the results are sent to the user through the earpiece.

B. Results

In the virtual assistant; when the system starts, if there is a voice the system will be started by recording the voice of the user, then the system will retrieve the text using the Google’s Speech to Text API, then the system will extract the Intent, the Entities, and the Respond from the text using the RASA Chatbot running on the server.

If the returned response was not None, this means that the RASA has been replied to the user by a message, thus the system will be using Google’s Text to Speech API to inform the user of the response; else, if the return Respond was None, this means that the user is asking for something that can’t be directly returned by RASA (getting information from Google), so the RASA has been extracted the Intent and the Entities from the sentence. Thus, the system will be running classes with respect to the extracted Intents and Entities. And, then using Google’s Text to Speech API, the system will be informing the user of the results.

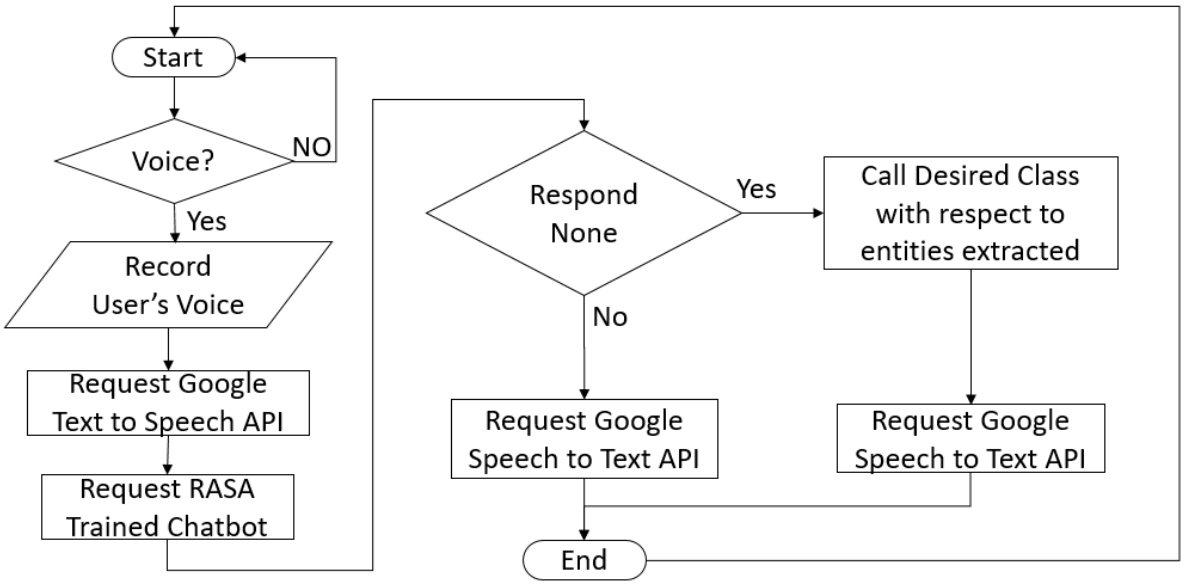


Figure 40. Virtual Assistant Block Diagram

In the object detection; when the system starts, the program will start by reading the ultrasonic sensor and will compute the distance, if the distance was less than 50 cm so the object in front of the user is about 50 cm far, so the system will be turning the buzzer with respect to the distance otherwise it will turn off the buzzer.

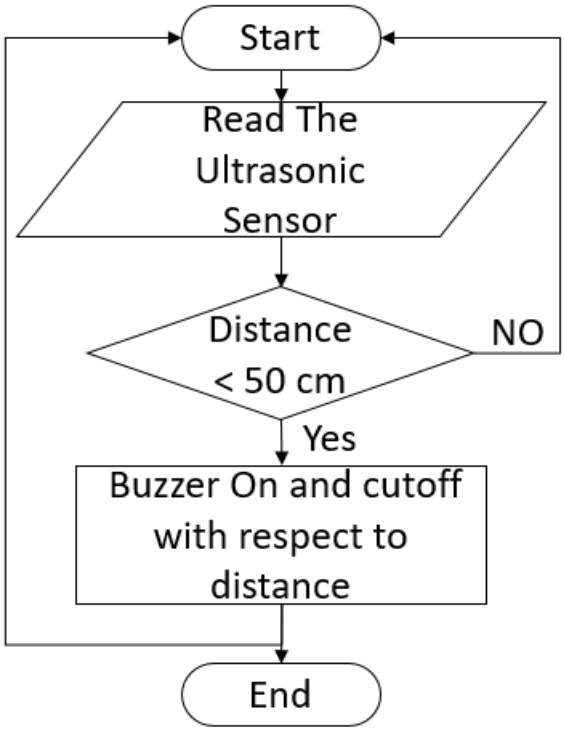


Figure 41. Object Detection Block Diagram

In the heart and the falling detection; when the system starts it will start by reading the heartrate and the accelerometer sensors, then it checks if the heart rate was greater than or less than a threshold, the system will send a 5v signal to the Raspberry PI through the I2C protocol on the 44 address; also, it will check the Accelerometers reading if the falling was detected the system will be sending a 5v signal to the Raspberry PI using the I2C protocol on the 45 address.

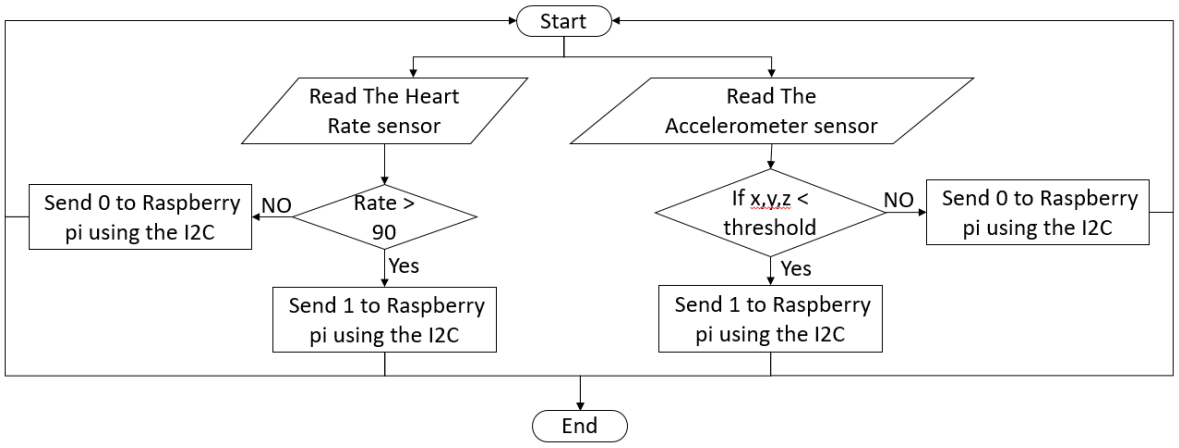


Figure 41. Heart and Falling Detection Block Diagram

C. Discussion

After the completion and the testing of the project, we established that the project is in a great working condition. The AI camera has worked properly and as it should, the AI agent replied to the user and answered exactly what he/she desires, and in addition, the earlobe blood pressure measuring sensor calculated the blood pressure statistics correctly and projected it correctly to the user.

Knowing that blind people face many daily difficulties and sometimes there isn't anyone that can help them overcome them. It is surely believed that the features presented by this project will help them overcome their problems. For example, the AI camera can be used as an extra set of eyes that can scan for them their surroundings, and thanks to the AI agent, the result of the scan will be copied to the user. These tasks can mostly fill in the place of their disability.

As an assumption from the AI simulation result, the outcome of the replies from the AI has demonstrated to us multiple factors. The most important one was the success of communication between the user and the AI agent. The AI is an easy agent to work with in

addition to the very wide dictionary used which allowed the user to use the basic word and he/she can be understood.

During the making of the project, many skills were learned. Programing the artificial intelligence agent was one of them. Multiple data sheets were created in order to be taught to the AI. This data sheet included the questions and the answers that the agent must reply to the user when asked.

Another skill that had to be learned was the implementation and wiring of the components. Knowing that the IOB system includes multiple devices such as a microphone, a camera, a Raspberry PI, and a pulse reading sensor, their connections needed average skills. Sadly, due to the failure of the project, this particular skill couldn’t be learned. But in general, the wiring and the connectivity of the elements could’ve been completed. The components used were easy to be connected and no unfamiliar ones were used.

Chapter 6
CONCLUSIONS AND FUTURE WORK

A. Conclusions

In conclusion, there are a lot of people in the world that are blind, whether they were born blind initially, or due to a certain incident that happened to them, which lead to them losing their vision. In the end, they are still humans and they ought to live their life normally but with some help. This help may be viewed as products and systems that were created and designed by engineers.

After a lot of researching and viewing a lot of products, many ideas and products help blind people a lot and make their lives easier. It is remarkable that every system supported unique characteristic(s) over the alternative and can have extra capabilities than the alternative, however, none of them supported all of the evaluated features. That means they can't be considered as an excellent device or system that the blind man or woman can depend upon and sense assured about using. Devices that have all of the essential features will provide effective performance. The perfect tool has to not only include a brand new feature but additionally fulfill the principle and primary desires of the person. The person needs to sense the experience of the encircling surroundings always and everywhere. The system can't be restricted for any particular case, otherwise, the design will be considered incomplete.

This project aims to design a device and a system that has many essential features bonded together and work in harmony as mentioned in chapter 3. A wearable device was designed that offers a wide variety of features. Some of these features are: Help the blind person navigate while detecting obstacles and make sure that the user doesn’t hit them. Help the visually impaired people to go to their desired place they want to reach, by using google maps. In addition, the design would include some medical sensors which can read many parameters like heart rate and blood pressure and know if the user is in a fine health condition or not.

AI and IoT will be used to have the most efficient device possible. Knowing that it is smart, self-driven, and has many useful features. In addition to making the design easy to use and feel comfortable to be worn, not making them feel uncomfortable while in use. The design of the product will be as basic as possible so that the wearer won’t be looked at in a weird/ unfamiliar way while wearing it.

B. Future Work

Due to the time, available materials restrictions, and financial problems, many features couldn't be added to the project. However, in the future, some attributes might be added that would make the device and system better and even more competent.

Being visually impaired might create an obstacle for the person. It can make them feel left out of the evolution of technology. That's why This IoB system can have more advanced features, like creating a virtual online multiverse. This multiverse can offer many activities to the person. It allows disabled people to meet with other kinds of disabled people. This platform features to help blind people communicate with deaf people for example. If the other person is blind, the user would activate a feature, where it listens to what the user says and convert it to text/subtitles, and so it would appear on a corresponding screen (phone screen, smartwatch screen…) and reveal it to the deaf person. The deaf person would read and understand what the blind man wants to say, hence the deaf person may write on the screen what he wants to say and the system will pronounce what was written. Moreover, it can offer the users multiple games to play with their friends made on the Multiverse.

A feature can also be added later on in this project. This feature reads the medical parameters of the corresponding sensors of the person. If any unusual results occur, the system will call an emergency number that the user would've entered previously or even 911. This feature would help in making the product more useful and efficient in terms of safety and help the user.

To ensure an additional safety layer for the user, the health data recorded by the blood pressure sensor on the ear can be installed on a specific part of the circuit. This act will allow a relative/significant other to extract the data of the user when the headsets are not in use and submit them to a doctor to look more into the health state of the user.

Another safety feature can be added. This feature focuses on the AI processing part of the project. The AI agent can be more advanced and scan more advanced parts. For example, it can scan the colors of the street lights to notify the user that it is safe to cross the road, even more, it can scan the sidewalks and make sure that they aren't high in a way that can harm the user.

References

[1] Welp A, Woodbury RB, McCoy MA from the book: “Making Eye Health a Population Health Imperative: Vision for Tomorrow” Source: https://www.ncbi.nlm.nih.gov/books/NBK402367/

[2] M. Campus Puebla. “Are blind people excluded from technology? - Tec”.2016 Source:https:// shorturl.at/elyOX

[3] M. Campus Puebla. “Are blind people excluded from technology? - Tec”. 2016 Source:https:// shorturl.at/tMRV7

[4] Envision and its team. “All you need to know about the White Cane”. November 26, 2018 Source: https:// shorturl.at/oENWX

[5] Ramirez, A.R.G. & Fonseca, R. & Durán Carrillo de Albornoz, Alejandro. ‘Electronic long cane for locomotion improving on visual impaired people.’, Universidade do Vale do Itajaí (Univali),2011 Source:https:// shorturl.at/lGJP4

[6] B. R. Prudhvi and R. Bagani, "Silicon eyes: GPS-GSM based navigation assistant for visually impaired using capacitive touch braille keypad and smart SMS facility," 2013 World Congress on Computer and Information Technology (WCCIT), 2013 Source: https://ieeexplore.ieee.org/abstract/document/6618775

[7] Ş. Aymaz and T. Çavdar, "Ultrasonic Assistive Headset for visually impaired people," 2016 39th International Conference on Telecommunications and Signal Processing (TSP),2016 Source: https://ieeexplore.ieee.org/abstract/document/7760903

[8] Mohd Helmy Abd Wahab,”Smart Cane”,Cornell University,2011 Source: https://arxiv.org/abs/1110.5156

[9] S. Bharambe, R. Thakker, H. Patil and K. M. Bhurchandi, "Substitute Eyes for Blind with Navigator Using Android," 2013 Texas Instruments India Educators' Conference, 2013 Source: https://ieeexplore.ieee.org/abstract/document/6757112

[10] A. Brilhault, S. Kammoun, O. Gutierrez, P. Truillet and C. Jouffrais, "Fusion of Artificial Vision and GPS to Improve Blind Pedestrian Positioning," 2011 4th IFIP International Conference on New Technologies, Mobility and Security, 2011 Source: https://ieeexplore.ieee.org/abstract/document/5721061

[11] M. F. Saaid, I. Ismail and Mohd Zikrul Hakim Noor, "Radio Frequency Identification Walking Stick (RFIWS): A device for the blind," 2009 5th International Colloquium on Signal Processing & Its Applications, 2009 Source: https://ieeexplore.ieee.org/abstract/document/5069227

[12] Roy Shilkrot," FingerReader: a wearable device to support text reading on the go "4th Augmented Human International Conference,2013

[13] Lucas Matney, "Aira's new smart glasses give blind users a guide through the visual world". TechCrunch March 27, 2018 Source:https:// shorturl.at/ituB0

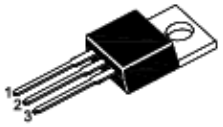
Datasheets

7809

3-terminal 1 A positive voltage regulator

Features

- Output Current up to 1 A
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection



1.Input 2.Common 3.Output
TO-220 Plastic Package

Absolute Maximum Ratings (T_a = 25 °C)

Parameter	Symbol	Value	Unit
Input Voltage	V_I	35	V
Thermal Resistance Junction-Cases	$R_{\theta JC}$	5	°C/W
Thermal Resistance Junction-Air	$R_{\theta JA}$	65	°C/W
Operating Temperature Range	T_{OPR}	0 to + 125	°C
Storage Temperature Range	T_S	- 65 to + 150	°C

Electrical Characteristics

(0 °C < T_J < 125 °C, I_O = 500 mA, V_I = 15 V, C_I = 0.33 µF, C_O = 0.1 µF, unless otherwise specified)

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Output Voltage	V_O	T_J = 25 °C		8.65	9	9.35	V
		5 mA ≤ I_O ≤ 1 A, P_O ≤ 15 W V_I = 11.5 V to 24 V		8.6	9	9.4	
Line Regulation [1]	Regline	T_J = 25 °C	V_I = 11.5 V to 25 V	-	-	180	mV
			V_I = 12 V to 17 V	-	-	90	
Load Regulation [1]	Regload	T_J = 25 °C	I_O = 5 mA to 1.5 A	-	-	180	mV
			I_O = 250 mA to 750 mA	-	-	90	
Quiescent Current	I_Q	T_J = 25 °C		-	-	8	mA
Quiescent Current Change	ΔI_Q	I_O = 5 mA to 1 A		-	-	0.5	mA
		V_I = 12 V to 26 V		-	-	1.3	
Output Voltage Drift	$\Delta V_O/\Delta T$	I_O = 5 mA		-	-1	-	mV/°C
Output Noise Voltage	V_N	f = 10 Hz to 100 KHz, T_A = 25 °C		-	58	-	µV
Ripple Rejection	RR	f = 120 Hz, V_I = 13 V to 23 V		56	-	-	dB
Dropout Voltage	V_{Drop}	I_O = 1 A, T_J = 25 °C		-	2	-	V
Output Resistance	R_O	f = 1 KHz		-	15	-	mΩ
Short Circuit Current	I_{SC}	V_I = 35 V, T_A = 25 °C		-	250	-	mA
Peak Current	I_{PK}	T_J = 25 °C		-	2.2	-	A

[1] Load and line regulation are specified at constant junction temperature, Changes in Vo due to heating effects must be taken into account separately, Pulse testing with low duty is used.

SEMTECH ELECTRONICS LTD.

Dated : 17/09/2016 Rev: 01

Figure 42. 7809 Voltage Regulator Datasheet

Aluminum Electrolytic Capacitors
Axial High Temperature, High Ripple Current

Fig. 1

FEATURES

- Extra long useful life: up to 8000 h at 125 °C
- Extended temperature range: usable up to 150 °C
- Low ESR levels provide very high ripple current capability
- Miniaturized, high CV-product per unit volume
- Polarized aluminum electrolytic capacitors, non-solid electrolyte
- Axial leads, cylindrical aluminum case, insulated with a blue sleeve
- Taped versions up to case Ø 15 mm x 30 mm available for automatic insertion
- Lead diameter Ø d = 1.0 mm, available on request
- AEC-Q200 qualified
- Material categorization: for definitions of compliance please see www.vishay.com/doc?99912

AUTOMOTIVE GRADE
Pb-free
RoHS COMPLIANT

APPLICATIONS

- Automotive, industrial, and telecommunication
- Smoothing, filtering, buffering
- Low mounting height applications, vibration and shock resistant
- SMPS and standard power supplies

MARKING

The capacitors are marked (where possible) with the following information:

- Rated capacitance (in µF)
- Tolerance on rated capacitance, code letter in accordance with IEC 60062 (M for ± 20 %)
- Rated voltage (in V)
- Upper category temperature (125 °C)
- Date code in accordance with IEC 60062
- Code for factory of origin
- Name of manufacturer
- Negative terminal identification
- Series number (120)

QUICK REFERENCE DATA

DESCRIPTION	VALUE
Nominal case sizes (Ø D x L in mm)	10 x 30 to 21 x 38
Rated capacitance range, C_R	47 µF to 6800 µF
Tolerance on C_R	± 20 %
Rated voltage range, U_R	16 V to 100 V
Category temperature range	-40 °C to +125 °C
Endurance test at 150 °C	1000 h
Endurance test at 125 °C	4000 h
Useful life at 125 °C	8000 h
Useful life at 85 °C, 1.4 x I_R applied	40 000 h
Shelf life at 0 V, 125 °C	1000 h (100 V: 500 h)
Shelf life at 0 V, 150 °C	≤ 63 V: 500 h
Based on sectional specification	IEC 60384-4 / EN 130300
Climatic category IEC 60068	40 / 125 / 56

SELECTION CHART FOR C_R, U_R, AND RELEVANT NOMINAL CASE SIZE (Ø D x L in mm)

C_R (µF)	U_R (V)				
	16	25	40	63	100
47	-	-	-	-	10 x 30
68	-	-	-	-	12.5 x 30
100	-	-	-	10 x 30	12.5 x 30
150	-	-	-	12.5 x 30	15 x 30
220	-	-	10 x 30	12.5 x 30	18 x 30
330	-	-	12.5 x 30	15 x 30	18 x 38
470	-	10 x 30	12.5 x 30	18 x 30	21 x 38
680	10 x 30	12.5 x 30	15 x 30	18 x 38	-
1000	12.5 x 30	12.5 x 30	18 x 30	21 x 38	-
1500	12.5 x 30	15 x 30	18 x 38	-	-
2200	15 x 30	18 x 30	21 x 38	-	-
3300	18 x 30	18 x 38	-	-	-
4700	18 x 38	21 x 38	-	-	-
6800	21 x 38	-	-	-	-

Figure 43. 470 uf Capacitor Datasheet

Ceramic Singlelayer DC Disc Capacitors (Straight Leads) Gap-Kap, 1 kV_{DC} to 3 kV_{DC}

FEATURES

- High reliability
- Straight leads
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912

Pb e3 RoHS COMPLIANT

APPLICATIONS

- Monitors
- Color TV

DESIGN

The capacitors consist of a ceramic disc both sides of which are silver-plated. Connection leads are made of tinned copper having a diameter of 0.8 mm.

The capacitors are supplied with straight leads and lead spacings from 5.0 mm to 10.0 mm. Encapsulation is phenolic resin coated, flammable resistant in accordance with "UL 94 V-0".

QUICK REFERENCE DATA			
DESCRIPTION	VALUE		
Ceramic Class	2		
Ceramic Dielectric	Z5P, Z5U		
Voltage (V_{AC})	1000	1500	3000
Min. Capacitance (pF)	0.75		
Max. Capacitance (pF)	22 000		
Mounting	Radial		

INTRODUCTION

Vishay BCcomponents Gap-Kap capacitors provide a safe reliable discharge path for stray transient overvoltages and static voltage build-up. Combination of capacitor-spark-gap construction allows the circuit designer to specify lower voltage components and consequently lower cost, with assurance that overvoltage conditions will be prevented.

The Gap-Kap capacitor is ideally suited for many industrial commercial equipment applications. A typical application in color TV monitors utilizes a minimum capacitance Gap-Kap which is inserted between the grid lead and chassis ground. This protects the components of control circuitry by providing a low impedance path to ground for transient voltages of 1500 V and above.

MARKING

Marking indicates capacitance value and tolerance in accordance with "EIA 198" and voltage marks.

OPERATING TEMPERATURE RANGE

- 30 °C to + 85 °C

TEMPERATURE COEFFICIENTS

EIA code Z5P or Z5U

SECTIONAL SPECIFICATIONS

Class 2, IEC 60384-9, EIA 198

CAPACITANCE RANGE

At 1 kHz, 1 V_{RMS} ± 0.2 V_{RMS}: 0.75 pF to 22 000 pF

RATED DC VOLTAGE

1 kV; 1.5 kV; 3 kV

INSULATION RESISTANCE AT 500 V_{DC}

≥ 10 000 MΩ min.

TOLERANCE ON CAPACITANCE

± 10 %; ± 20 %

DISSIPATION FACTOR

At 1 kHz, 1 V_{RMS} ± 0.2 V_{RMS}: 2.5 % max.

Note

- The capacitors meet the essential requirements of IEC 60384-9 and EIA 198.
  Unless stated otherwise all electrical values apply at an ambient temperature of 25 °C ± 3 °C, at normal atmospheric conditions.

Figure 44. 0.01 uf Capacitor Datasheet

RASA Files

```
-version: "3.1"
nlu:
- intent: greet
  examples: |
    - hey
    - hello
    - hi
    - good morning
    - good evening
    - hey there
    - hello there
    - let's go
    - hey dude
    - good afternoon
    - how is it going
    - what's up
    - nice to meet you
    - pleased to meet you
    - good to meet you
    - how are you doing
    - nice to see you
    - it's great to see you
    - good to see you
    - yo
    - what's up
    - sup
    - lovely to meet you
    - lovely to see you
    - welcome back
    - hey there
    - hi there
    - hello there
- intent: ask_about_me
  examples: |
   - how about you?
   - what about you?
   - i am fine, what about you?
   - how are you?
   - how have you been?
   - how do you do?
- intent: goodbye
  examples: |
    - see you
    - good bye
    - see you later
    - good night
    - bye
    - byebye
    - goodbye
    - have a nice day
    - see you around
    - bye bye
    - see you later
    - farewell
    - bye for now
    - talk to you later
    - peace out
```

```
    - bye, have a nice day
    - catch you later
    - later
    - i am off
    - i have to go
    - i gotta go
    - looking forward for our next meeting
    - i am out
    - peace
    - peace out
    - keep in touch
    - see you soon
    - see you next time
    - have a good day
- intent: affirm
  examples: |
    - yes
    - indeed
    - totally agreed
    - totally right
    - true
    - why not
    - confirm
    - approve
    - accept
    - sure
    - certify
    - of course
    - that sounds good
    - correct
    - affirmative
    - affirm
    - positive
    - yeah
    - yas
    - why not
    - yes please
    - please do
    - please
    - please yes
    - fuck yes
    - always
    - okay
    - ok
    - deal
    - noted
    - yea
    - all right
    - certainly
    - definitely
    - exactly
    - gladly
    - good enough
    - sure thing
    - surely
    - undoubtedly
    - yep
```

```
    - yup
- intent: deny
  examples: |
    - not at all
    - no
    - disagree
    - do not agree
    - false
    - refuse
    - turn down
    - reject
    - decline
    - dismiss
    - never
    - I don't think so
    - don't like that
    - no way
    - not really
    - impossible
    - negative
    - no indeed
    - of course not
    - never
    - nah
    - nope
    - nae
    - nay
    - naw
    - forget
    - by no means
    - in no wise
    - hardly
    - none
    - nowhere
    - no way
    - no wise
    - denial
    - rejection
- intent: thank
  examples: |
    - thanks
    - thank you
    - thanks for your help
    - thanks for being here
    - so pleased to have you
    - appreciated
    - grateful
    - thanks a lot
    - i owe you
    - much obliged
    - bless you
    - thank you for your help
    - much appreciated
```

```
- glad doing business with you
    - thank you for helping me
    - thank you so much
    - thank you very much
    - i appreciate your help
    - i appreciate your efforts
    - i appreciate your time
    - i appreciate your guidance
    - i appreciate your assistance
    - i appreciate your consideration
    - please accept my deepest thanks
    - thank you for your assistance
    - i wanted to thank you as soon as possible
    - thank you for everything
    - you have my gratitude
    - i am very thankful for your time.
    - i appreciate the information and advice you have shared.
    - i sincerely appreciate the assistance.
    - many thanks for your assistance.
    - many thanks for your time.
    - thank you for taking the trouble to help me.
    - thank you for all the help
    - your help is greatly appreciated.
    - i am grateful for your help.
    - i very much appreciate your support.
    - thank you for your support
- intent: joke
  examples: |
    - tell me a joke
    - can you tell me a joke
    - please give me a joke
    - i am sad tell me a joke
    - give me another joke
    - give me something funny
    - joke
    - joke now
    - mess with me
    - amuse me
    - entertain me
    - tell me a good joke
    - tell me silly jokes
    - tell me dark jokes
    - tell me a funny story
    - please tell me a joke
- intent: User_needs #
  examples: |
    - what is this
    - what is the object in front of me
    - what's in front of me
    - what is in front of me
    - what is the closest object to me
    - what is the [text](text_recognition) in front of me
    - read this [sentence](text_recognition)
```

```
- read this [text](text_recognition)
    - read the [following](text_recognition)
    - read the following [topic](text_recognition)
- intent: Quantity #
  examples: |
    - how many quantity
    - how much quantity is there
    - how many quantity do i have in my
    - how much quantity is there
    - what is the number of [chairs](facility_type)
    - what is the number of [doors](facility_type)
    - how much quantity did he give me
- intent: mood_great
  examples: |
    - perfect
    - great
    - amazing
    - feeling like a king
    - wonderful
    - i am feeling very good
    - i am great
    - i am amazing
    - i am going to save the world
    - super stoked
    - extremely good
    - so so perfect
    - so good
    - so perfect
    - good
    - feeling well
    - feeling good
    - feeling great
    - feeling amazing
    - positive
    - feeling positive
    - feeling happy
    - feeling relaxed
    - feeling excited
    - excited
    - fantastic
    - glad
    - i'm glad
    - pleased
    - i am pleased
    - wonderful
    - satisfied
    - i'm over the moon
    - delighted
    - thrilled
- intent: mood_unhappy
  examples: |
    - my day was horrible
    - i am sad
```

```
    - i don't feel very well
    - i am disappointed
    - super sad
    - i'm so sad
    - sad
    - very sad
    - unhappy
    - not good
    - not very good
    - extremely sad
    - i don't feel good
    - devastated
    - wrecked
    - horrible
    - bad
    - unfortunate
    - feeling wrecked
    - feeling sad
    - feeling unhappy
    - not feeling good
    - not feeling happy
    - not feeling well
    - unlucky
    - awful
    - shit
    - frustrated
    - heartbroken
    - pain
    - lonely
    - feeling bad
    - it was a bad day
    - it is the worst day
    - i hate everything
- intent: information
  examples: |
    - [saida](location)
    - [hasbaya](location)
    - [dekwany](location)
    - [beirut](location)
    - [tyre](location)
    - [byblos](location)
    - [zalka](location)
    - [dbayeh](location)
    - [sour](location)
    - [sawfar](location)
    - [Baalbek](location)
    - [dora](location)
    - [achrafieh](location)
    - [sassine square](location)
    - [rawshe](location)
    - [batroun](location)
    - [nasre](location)
    - [hospital](facility_type)
```

```
- intent: get_info
  examples: |
        - tell me about [Lebanon](location)
        - tell me about the [weather](weather) in [beirut](location)
        - how is the [weather](weather) [today](date)
        - how is the [weather](weather) [tomorrow](date) in [beirut](location)
        - how is the [weather](weather) [today](date) in [saida](location)
        - how is the [weather](weather) [today](date) in [beirut](location)
        - how is the [weather](weather) [today](date) in [hasbaya](location)
        - how is the [weather](weather) [today](date) in [nabatieh](location)
        - tell me about the [weather](weather) in lebanon this [week](date)
        - what is the [temperature](weather) now [around me](my_location)
        - is it [sunny](weather) [today](date)
        - is it [sunny](weather) [today](date) [here](my_location)
        - is it [rainy](weather) [here](my_location) [today](date)
        - is it [windy](weather) [here](my_location) [today](date)
        - is it [cloudy](weather) [here](my_location) [today](date)
        - what [time](clock) is it now
        - what [time](clock) is it [britain](location) now
        - what is the [time](clock) now
        - what is the [time](clock)
        - can you tell me the [time](clock)
        - could you tell me the [time](clock) please
        - do you have the [time](clock)
        - have you got the [time](clock)
        - what do you recommend to read
        - what do you recommend to eat
        - what do you suggest to read
        - what do you commend to read
        - what is a [phone](product)
        - what is a [car](product)
        - give me some details about [water](product)
        - latest version of [huawei p30](product)
        - what are the [skills](requiremnts) needed for a job
        - tell me what are the [requirements](requiremnts) to enter a
university
        - what are the seasons of the year
        - what is the latest [iphone](product) on the market
        - what [phone](product) is better
        - best [phones](product) in the market
        - tell me about [lebanon](location)
        - tell me about the [lebanese](location) history
        - what is the area of [lebanon](location)
        - best places to visit in [lebanon](location)
        - best countries to visit in summer
        - who is [Elon Musk](person)?
        - give me the [news](news)
        - what are [todays](date) [news](news)
        - tell me about the [news](news)
        - what are the [news](news)
- intent: search_location_provider
  examples: |
    - i need a [hospital](facility_type)
```

```
    - find me a [nearby](my_location) [hospital](facility_type)
    - show me [home health agencies](facility_type)
    - find me a nearby [hospital](facility_type) in [beirut](location)
    - i need a [home health agency](facility_type)
    - i need a [Bank](facility_type)
    - i need a [restaurant](facility_type)
    - find me a [restaurant](facility_type)
    - show me [restaurant](facility_type)
    - find me a nearby [restaurant](facility_type) in [saida](location)
    - what [restaurants](facility_type) are [around me](my_location)
    - what is the nearest [restaurant](facility_type) from
[here](my_location)
    - what is the best rated [restaurant](facility_type) near
[me](my_location)
    - what is the nearest [hotel](facility_type) to [me](my_location)
    - where do i live(home_location)
    - where is [home](home_location)
    - where can i find a [pet shop](facility_type)
    - where is the [nearest](my_location) pharmacy
    - what is the nearest [restaurant](facility_type) to [me](my_location)
    - how much [time](clock) needed from [here](my_location) to the nearest
[restaurant](facility_type)
    - how much [time](clock) needed from [here](my_location) to
[beirut](location) [restaurant](facility_type)
    - how much [time](clock) i need from [here](my_location) to the nearest
[library](facility_type)
    - how much [time](clock) needed to get from [here](my_location) to
[saida](location)
    - how much [time](clock) needed from [here](my_location) to the
[airport](facility_type)
    - how much [time](clock) needed from [home](home_location) to the
[airport](facility_type)
    - top [restaurants](facility_type) in [beirut](location)
    - top [restaurants](facility_type) in [lebanon](location)
    - top 10 [universities](facility_type) in [lebanon](location)
- intent: fact
  examples: |
    - tell me a fact
    - tell me some facts
    - facts
    - i want some facts
- intent: text_recognition
  examples: | #facility type only works
    - read for me [Harry Potter](Books)
    - what is written in front of me
    - i want to read [layla wa al thyb](Books)
    - read the [message](facility_type) sent by [raed diab](old_contact)
    - please read the [email](facility_type) sent by my
[manager](old_contact)
- intent: stop_direction
  examples: |
    - stop the directions
    - close the directions
```

```
    - close the navigation
    - switch off the navigation
    - turn off the GPS
    - stop telling me the directions
- intent: get_direction
  examples: |
    - i want to go from [here](my_location) to [work](work_location)
    - i want to go from [my home](home_location) to [work](work_location)
    - i want to go to [my home](home_location)
    - how to go to [my home](home_location)
    - take me [home](home_location)
    - give me the directions to go [home](home_location)
    - please i want my [home](home_location)
    - take me back to [home](home_location)
    - to [home](home_location)
    - take me to [Fransa Bank](facility_type)
    - take me from [here](my_location) to the nearest [bank](location)
    - take me to [Audi Bank](facility_type)
    - take me from [here](my_location) to [Audi Bank](facility_type)
    - take me to [beirut](location)
    - take me to [saida](location)
    - take me to [kfc restaurant](facility_type)
    - take me to [kfc](facility_type) [beirut](location)
    - assist me to arrive to [mcdonald's](facility_type)
    - take me to [al bohsasa restaurant](facility_type)
    - take me to the [hospital](facility_type)
    - take me to the [nearest](my_location) [hospital](facility_type)
    - give me the directions to go to the [hospital](facility_type)
    - assist me to arrive the [nearest](my_location)
[hospital](facility_type)
    - take me to the [pharmacy](facility_type)
    - i want to go to a [pharmacy](facility_type) [around me](my_location)
    - please give me the direction to go to the [nearest](my_location)
[pharmacy](facility_type)
    - help me to reach the [nearest](my_location) [pharmacy](facility_type)
    - help me to reach a [pharmacy](facility_type)
    - take me to a [pharmacy](facility_type)
    - take me to the nearest [library](facility_type)
    - help me to go from [here](my_location) to the [library](facility_type)
- intent: play_music
  examples: |
    - play for me music
    - play some musics
    - play music
    - please play music
    - play me some music
    - play for me some music
    - i feel like listening to music
    - music now
    - music
    - now music
    - any music
    - i want to dance
```

```
    - i feel like dancing
    - give me something to dance to
    - play me something to dance to
    - i want to listen to music
    - i want a playlist for [Eminem](artist)
    - play for me [top 10](song) songs this year
    - play for me my [GYM](song) playlist
    - please play for me [wael kfory](artist) musics
    - please play [wael kfoury](artist) playlist
    - i want to listen to music
    - play for me [nancy ajram](artist)
    - play for me [beed lshar](song)
    - i want a [sad](song) song
    - i want to listen to [rap](song)
    - play some [rap](song) songs
    - play for me [rap](song) music
- intent: stop_music
  examples: |
    - stop the music
    - stop that music
    - stop the play
    - stop the playing of that music
    - pause the music
    - pause this music
    - stop playing the music
    - exit
    - stop playing music
- intent: next_music
  examples: |
    - next song
    - give me the next one
    - run for me the next
    - play for me the next
    - i don't love this, please run me the next one
    - i hate it, play another one
- intent: prev_music
  examples: |
    - give me the previous song
    - give me the previous one
    - run for me the previous
    - play for me the previous
    - i don't love this, please run me the previous one
    - i hate it, play the one before it
- intent: phone_call
  examples: |
    - i want to make a [reservation](reserve) at [Al bohsasa
restaurant](facility_type)
    - call [raed diab](old_contacts)
    - please call [my mother](old_contacts)
    - could you call [my brother](old_contacts)
    - call my [manager](old_contacts)
    - call an [ambulance](facility_type)
- intent: call_taxi
```

```
  examples: |
    - call me a taxi to [beirut](location)
    - could you organize a taxi for me [this evening](date) please to
[kfc](facility_type) [beirut](location)
    - could you hail a taxi for me please to [saida](location)
    - call me an uber to take me to the [nearest](my_location)
[hospital](facility_type)
    - call me a bolt to take me to [hasbaya](location)
- intent: call_delivery
  examples: |
    - i want to order [food](request_for) from [kfc](facility_type)
    - i want [cold drink](request_for) from [starbucks](facility_type)
    - i need some [coffe[(request_for) from [B Hieve](facility_type)
- intent: send
  examples:  |
    - please send an [email](facility_type) to [mostafa](old_contacts)
    - please read the [email](facility_type) sent by my
[manager](old_contacts)
    - send a [message](facility_type) to [raed diab](old_contacts)
    - send a [message](facility_type) to [mostafa](old_contacts)
    - please make[raed] (old_contacts) and [jad](old_contacts)  as
[CC](facility_type)
    - could you mark the last [email](facility_type) from [joe](old_contacts)
as unread
    - please reply to [rami](old_contacts)
- intent: calculate
  examples:  |
    - calculate for me 3 plus 4
    - calculate for me 8 plus 2
    - calculate 8 times 3
```

Figure 45. RASA - NLU File

```yaml
version: "3.1"

intents:
  - greet:
      triggers: utter_greet
  - goodbye:
      triggers: utter_goodbye
  - affirm
  - deny
  - mood_great
  - mood_unhappy
  - bot_challenge
  - thank:
      triggers: utter_thank
  - get_info:
      triggers: action_extract
  - get_direction:
      triggers: action_extract

  - text_recognition
  - play_music
  - joke
  - User_needs
  - ask_about_me:
       triggers: utter_ask_about_me

  - phone_call
  - calculate
  - stop_music
  - next_music
  - prev_music
  - Quantity
  - pause_music
  - information
  - search_location_provider
  - fact
  - stop_direction
  - call_taxi
  - call_delivery
  - send

entities:
  - location
  - my location
  - home_location
  - facility_type
  - weather
  - date
  - clock
  - product
  - artist
  - song
  - Books
  - old_contacts
  - reserve
```

```
slot:
    - my_location:
        type: text
    - facility_type:
        type: text
    - address:
        type: text

actions:
  - utter_greet
  - utter_cheer_up
  - utter_did_that_help
  - utter_happy
  - utter_goodbye
  - utter_iamabot
  - utter_assist_him
  - utter_search_provider
  - utter_direction
  - utter_ask_about_me
  - utter_ask_about_him
  - utter_thank
  - utter_play_music
  - action_extract


responses:
  utter_greet:
  - text: "Hey!"
  - text: "Hello there!"
  - text: "Hi"
  - text: "Hello!"

  utter_ask_about_me:
  - text: "Am fine! Thanks for your concern"
  - text: "Am Great! Thanks for asking!"
  - text: "I am good! Thank you"
  - text: "I am more than Great! Thanks for your question"

  utter_cheer_up:
  - text: "Here is something to cheer you up:"

  utter_did_that_help:
  - text: "Did that help you?"

  utter_happy:
  - text: "Great, carry on!"
  - text: "That's Great!"

  utter_goodbye:
  - text: "Bye, if you need any help iam here for you"
  - text: "Byebye, when you need me, you know where am I"
  - text: "GoodBye, whenever you want me feel free to call me"

  utter_iamabot:
  - text: "I am your virtual assistant, powered by IOB Engineering Team at
the American University of Science and Technology. I am here to assist you as
possible as I can"
```

```
  utter_assist_him:
  - text: "How can I help you?"
  - text: "what I can do for you?"
  - text: "what you need me to help you?"

  utter_ask_about_him:
  - text: "What about you?"
  - text: "How are you?"
  - text: "Are you fine?"
  - text: "Are you good?"

  utter_search_provider:
  - text: "Working on it!"

  utter_direction:
  - text: "Working to get the direction!"

  utter_thank:
  - text: "no need, I am here for you!"
  - text: "no need to mention it!"
  - text: "you're most welcome!"
  - text: "Welcome dear!"

  utter_play_music:
  - text: "playing music"

session_config:
  session_expiration_time: 60
  carry_over_slots_to_new_session: true
```

Figure 46. RASA - Domain File

```yaml
version: "3.1"

rules:
- rule: Say goodbye anytime the user says goodbye
  steps:
  - intent: goodbye
  - action: utter_goodbye

- rule: thanking
  steps:
  - intent: thank
  - action: utter_thank

- rule: When user asks for a joke
  steps:
  - intent: joke
  - action: action_extract
  - action: utter_did_that_help

- rule: when user asks for a fact
  steps:
  - intent: fact
  - action: action_extract
  - action: utter_did_that_help

- rule: search for him any time he provide a search example
  steps:
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help

- rule: askig for facilities in a location
  steps:
    - intent: search_location_provider
    - action: action_extract
    - action: utter_did_that_help

- rule: assist him any time he asks for assisting to go to direction
  steps:
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help

- rule: stop directing him any time he asks for assisting to go to direction
  steps:
  - intent: stop_direction
  - action: action_extract

- rule: Read for him his desired Book
  steps:
  - intent: text_recognition
  - action: action_extract
  - action: utter_did_that_help
```

```yaml
- rule: play for him music anytime he ask for that
  steps:
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help

- rule: stop for him music anytime he ask for that
  steps:
  - intent: stop_music
  - action: action_extract

- rule: play next music anytime he ask for that
  steps:
  - intent: next_music
  - action: action_extract
  - action: utter_did_that_help

- rule: play prev music anytime he ask for that
  steps:
  - intent: prev_music
  - action: action_extract
  - action: utter_did_that_help

- rule: Call for him someone
  steps:
  - intent: phone_call
  - action: action_extract
  - action: utter_did_that_help

- rule: Call Taxi when he ask for that
  steps:
  - intent: call_taxi
  - action: action_extract

- rule: Call Delivery when he ask for that
  steps:
  - intent: call_delivery
  - action: action_extract

- rule: Send Somthing when he ask for that
  steps:
  - intent: send
  - action: action_extract

- rule: Calculate for him when he ask for that
  steps:
  - intent: calculate
  - action: action_extract

- rule: Quantity infront of him when he ask for that
  - intent: Quantity
  - action: action_extract
  - action: utter_did_that_help
```

```
- rule: assist him when he ask for User_needs
  steps:
  - intent: User_needs
  - action: action_extract
  - action: utter_did_that_help

- rule: just giving 1 name so as information
  steps:
  - intent: information
  - action: action_extract
  - action: utter_did_that_help

- rule: Say 'I am a bot' anytime the user challenges
  steps:
  - intent: bot_challenge
  - action: utter_iamabot
```

Figure 47. RASA - Rule File

```
version: "3.1"

stories:

- story: greet and ask about bot and bot then ask about him and he is not
fine
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: deny
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and ask about bot and bot then ask about him and he is not
fine 2
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: deny
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: deny
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye

- story: greet and ask about bot and bot then ask about him and he ask for
search
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and ask about bot and bot then ask about him and he ask for
search 2
  steps:
  - intent: greet
  - action: utter_greet
```

```
  - action: utter_assist_him
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and ask about bot and bot then ask about him and he ask for
search 3
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye

- story: greet and search
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and search 2
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: get_info
  - action: action_extract
```

```
- action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and search 3
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye

- story: greet and ask about me then him then direction
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and ask about me then him then direction 2
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_direction
```

```
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and ask about me then him then direction 3
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye


- story: greet and direction
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and direction 2
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and direction 3
```

```
steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye

- story: greet and ask about me then him then ask about music
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and ask about me then him then ask about music 2
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and ask about me then him then ask about music 3
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
```

```
  - intent: ask_about_me
  - action: utter_ask_about_me
  - action: utter_ask_about_him
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye

- story: greet and music
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and music 2
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: greet and music 3
  steps:
  - intent: greet
  - action: utter_greet
  - action: utter_assist_him
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: play_music
  - action: action_extract
```

```
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye

- story: happy path
  steps:
  - intent: mood_great
  - action: utter_happy

- story: sad then happy path 1
  steps:
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: sad then happy path 2
  steps:
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: thank
  - action: utter_thank

- story: sad then happy path 3
  steps:
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: deny
  - action: utter_cheer_up
  - intent: affirm
  - action: utter_happy

- story: sad then happy path 4
  steps:
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: deny
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: thank
  - action: utter_thank

- story: sad path
  steps:
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: deny
  - action: utter_cheer_up
```

```
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye

- story: sad path 2
  steps:
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: thank
  - action: utter_thank

- story: sad path 3
  steps:
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: deny
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: thank
  - action: utter_thank

- story: search_provider path
  steps:
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: search_provider path 2
  steps:
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: search_provider path 3
  steps:
  - intent: get_info
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_info
  - action: action_extract
```

```
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye

- story: direction path
  steps:
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: direction path path 2
  steps:
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: direction path 3
  steps:
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: get_direction
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye

- story: music path
  steps:
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: music path path 2
  steps:
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
```

```
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy

- story: music path 3
  steps:
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_assist_him
  - intent: play_music
  - action: action_extract
  - action: utter_did_that_help
  - intent: deny
  - action: utter_goodbye
```

Figure 48. RASA - Stories File

```
from rasa_sdk import Tracker
from rasa_sdk.executor import CollectingDispatcher
from typing import Dict, Text, Any, List
from rasa_sdk import Action

class Play_music(Action):

    def name(self) -> Text:
        self.artist_type = ''
        self.song_type = ''

        return "action_extract"

    def run(self,
            dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict]:
        all_found = {}
        try:
                all_found['intent_category'] =
tracker.latest_message['intent'].get('name')
           except:
                   all_found['intent_category'] = None
           try:

            entities_category = tracker.latest_message['entities']
            all_found['content'] = {}
            for i in entities_category:
                all_found['content'][i['entity']] = i['value']
        except:
                   all_found['content'] = None

        dispatcher.utter_message(json_message = all_found)
```

Figure 49. RASA – Custom Action File

Backend Classes

```
import pyttsx3
import speech_recognition as sr
class AI():
    __name = ""
    __skills = []
    def __init__(self,name = None):
        self.engine = pyttsx3.init()
        self.r = sr.Recognizer()
        self.m = sr.Microphone()
        if name is not None:
            self.__name = name
        with self.m as source:
            self.r.adjust_for_ambient_noise(source)
    @property
    def name(self):
        return self.__name
    @name.setter
    def name(self, value):
        self.__name = value
        sentence = "Hello, my name is"+self.__name
        self.engine.say(sentence)
        self.engine.runAndWait()
    def setvol(self, vol):
        self.engine.setProperty('volume',vol)
    def say(self, sentence):
        self.engine.say(sentence)
        self.engine.runAndWait()
    def listen(self):
        with self.m as source:
            audio = self.r.listen(source)
        try:
      phrase = self.r.recognize_google(audio,show_all=False,language='en-GB')
            sentence = "Got it, you said"+phrase + ", is that Right?"
            self.engine.say(sentence)
            self.engine.runAndWait()
        except:
            print("sorry, didn't catch that")
            phrase = "sorry, didn't catch that"
            self.engine.say("sorry, didn't catch that")
            self.engine.runAndWait()
        return phrase
    def just_listen_for_confirm(self):
        with self.m as source:
        audio = self.r.listen(source)
        print("got it!")
        try:
      phrase = self.r.recognize_google(audio,show_all=False,language='en-GB')
            if 'yes' in phrase or 'right' in phrase:return True
            else: return False

        except:
            phrase = "sorry, didn't catch that"
            self.engine.say("sorry, didn't catch that")
            self.engine.runAndWait()
```

Figure 50. Virtual Assistant Backend – Speech_Text Class

```
import re

import requests
class Custom_Action:
    def __init__(self):
        print("initializing the bot that will give the respond")

    def get_entities(self,message,URL):
        self.message = message
        myobj = {
            "message": message,
            "sender": "Raed",
        }
        x = requests.post(URL, json=myobj)
        print(x.text)
        responces = []
        intity_for_each_respond = []
        entities_values_extracted_for_each_respond = []
        responces_recived = re.findall(r'\{.*?\}', x.text)
        #print(responces_recived)
        for respond in responces_recived:
            if len(respond.split('custom')) > 1:
                all = respond.split('custom')[-
1].split('"intent_category":"')[-1].split('","')
                intity = all[0]
                entities = [all[i] for i in range(len(all)) if i != 0]
                entities_values_extracted_string = []
                entities_values_extracted = {}
                for index, entity in enumerate(entities):
                    if index == 0:
entities_values_extracted_string.append(entity.split('":{"')[-1])
                    elif index == len(entities) - 1:
entities_values_extracted_string.append(entity.split('"}}}]')[0])
                    else:
entities_values_extracted_string.append(entities[index])
                for i in entities_values_extracted_string:
                    try:
                        value, key = i.split('":"')
                        if len(entities_values_extracted_string) > 1:
                            entities_values_extracted[value] = key
                        else:
                            entities_values_extracted[value] =
key.split('"}}}]')[0]
                    except:
                        entities_values_extracted = {}
                responde = "None"
            else:
                intity = 'None'
                entities_values_extracted = {}
                responde = respond.split('"text":"')[-1].split('"}')[0]
                print(responde)
            responces.append(responde)
            intity_for_each_respond.append(intity)
entities_values_extracted_for_each_respond.append(entities_values_extracted)
        return
responces,intity_for_each_respond,entities_values_extracted_for_each_respond
```

Figure 51. Virtual Assistant Backend – RASA Request Class

```
import datetime
class Date:
    def __init__(self):
        print("Initiating the date")
    def get_date_time(self):
        self.today = datetime.datetime.now()
        date = self.date()
        time = self.time()
        return date,time
    def date(self):
        return
self.today.strftime('%d'),self.today.strftime('%B'),self.today.strftime('%Y')
    def time(self):
        return self.today.strftime("%H:%M:%S")
```

Figure 52. Virtual Assistant– Date Class

```
import requests
class News:
    def __init__(self):
        print("Initiating the News Class")
        api_file = open("api-key-news.txt", "r")
        self.api_key = api_file.readline()
        api_file.close()
    def get_news(self):
        api_address = "http://newsapi.org/v2/top-
headlines?country=us&apiKey=" + self.api_key
        json_data = requests.get(api_address).json()
        ar = []
        for i in range(3):
            ar.append("Number "+str(i+1)+" news:
"+json_data['articles'][i]["title"]+".")
        return ar
```

Figure 53. Virtual Assistant Backend – News Class

```
import randfacts
class Fact:
    def __init__(self):
        print("I n initiating the FACT Class")
    def funfact(sef):
        return randfacts.get_fact()
```

Figure 54. Virtual Assistant Backend – Facts Class

```
import pyjokes
class Joke:
    def __init__(self):
        print("Ininitiating the JOKE Class")
    def get_joke(self):
        return pyjokes.get_joke()
```

Figure 55. Virtual Assistant Backend – Jokes Class

```
import smtplib
class Email():
    def __init__(self):
        print("Initializing the email class!")
        credentials_file = open("credentials_file", "r")
        credential_details = credentials_file.readlines()
        self.my_email = credential_details[0].split('\n')[0]
        self.password = credential_details[1]
        credentials_file.close()
        self.session = smtplib.SMTP("smtp.gmail.com", 587)
    def Send_Email(self,destination,subject,content):
        recipient = destination
        subject = subject
        message = content
        email = "Subject: {}\n\n{}".format(subject, message)
        self.session.starttls()
        self.session.login(self.my_email, self.password)
        self.session.sendmail(self.my_email, recipient, email)
        self.session.quit()
        return "Done Sending the Email"
```

Figure 56. Virtual Assistant Backend – Email Class

```
import requests
class Weather:
    def __init__(self):
        print("Initialing the weather class")
        api_file = open("api-weather.txt", "r")
        self.api_key = api_file.readline()
        api_file.close()
    def get_wether_info(self,loc):
        self.api_address =
f'http://api.weatherapi.com/v1/current.json?key={str(self.api_key)}&q={loc}&a
qi=no'
        self.json_data = requests.get(self.api_address).json()
        #print(self.json data)
        temp,description = self.temp(),self.description()
        return temp,description
    def temp(self):
        return (self.json_data['current']['temp_c'])
    def description(self):
        return self.json_data['current']['condition']['text']
```

Figure 57. Virtual Assistant Backend – Weather Class

```
import time
from selenium import webdriver
import win32clipboard
class Location:
    def __init__(self):
        self.driver = webdriver.Chrome("chromedriver.exe")
        print("Initiating the location class")
    def get_location(self,location):
        loc_facility_type_len = len(location.split(' '))
        self.driver.get(url='https://www.google.com/maps')
        search =
self.driver.find_element_by_xpath('//*[@id="searchboxinput"]')
        search.click()
        search.send_keys(location)
        enter = self.driver.find_element_by_xpath('//*[@id="searchbox-
searchbutton"]')
        enter.click()time.sleep(10)
        if loc_facility_type_len <= 1:
            go =
self.driver.find_element_by_xpath('//*[@id="QA0Szd"]/div/div/div[1]/div[2]/di
v/div[1]/div/div/div[2]/div[1]/div[3]/div')
            go.click()time.sleep(20)
            share_button =
self.driver.find_element_by_xpath('//*[@id="QA0Szd"]/div/div/div[1]/div[3]/di
v/div[1]/div/div/div[2]/div[4]/div[5]')
            share_button.click()time.sleep(20)
        else:share_button =
self.driver.find_element_by_xpath('/html/body/div[3]/div[9]/div[9]/div/div/di
v[1]/div[2]/div/div[1]/div/div/div[4]/div[5]')
            share_button.click()time.sleep(20)
        share_info =
self.driver.find_element_by_xpath('/html/body/div[3]/div[1]/div/div[2]/div/di
v[3]/div/div/div/div[2]/button[2]')
        share_info.click()time.sleep(10)
        URL_Gotten =
self.driver.find_element_by_xpath('/html/body/div[3]/div[1]/div/div[2]/div/di
v[3]/div/div/div/div[3]/div[1]/button[2]')
        URL_Gotten.click()
        win32clipboard.OpenClipboard()
        data = win32clipboard.GetClipboardData()
        win32clipboard.CloseClipboard()
lat,long =data.split('2d')[-1].split('!')[0],data.split('3d')[-
1].split('!')[0]
        lat_long = [lat,long]
        location = ','.join(lat_long)
        return location
```

Figure 58. Virtual Assistant Backend – Location Class

```
import geocoder
class My_Location:
    def __init__(self):
        print("Initializing My Location Class")
    def get_my_location(self):
        g = geocoder.ip('me')
        return g.latlng
    def lat(self):
        return self.get_my_location()[0]
    def lng(self):
        return self.get_my_location()[1]
```

Figure 59. Virtual Assistant Backend – User's_Location Class

```
from selenium import webdriver
import time
import threading
class music(object):
    def __init__(self):
        self.driver = webdriver.Chrome("chromedriver.exe")
    def Threading_to_play(self,query,type = "youtube"):
        thread = threading.Thread(target=self.run, args=[query,type])
        thread.daemon = True  # Daemonize thread
        thread.start()
    def run(self,query,type = 'youtube'):
        self.query = query
        if type == 'youtube':
            self.driver.get(url =
"https://www.youtube.com/results?search_query="+query)
            video = self.driver.find_element_by_xpath('//*[@id="title-
wrapper"]')
            video.click()
   def play_next(self):
        next =
self.driver.find_element_by_xpath('//*[@id="movie_player"]/div[29]/div[2]/div
[1]/a[2]')
        next.click()
    def play_prev(self):
        print("will work on it")
    def pause(self):
        pause = self.driver.find_element_by_xpath('//*[@id="movie_player"]')
        pause.click()
    def stop(self):
        self.driver.close()
```

Figure 60. Virtual Assistant Backend – Music Class

```
import requests
from bs4 import BeautifulSoup
from urllib.request import urlopen
import re
from googlesearch import search
class inflow():
    def __init__(self):
        api_file = open("api-key-maps.txt", "r")
        api_key = api_file.readline()
        api_file.close()
        self.api_key = api_key
    def get_info_places(self,entity,intent,location = '33.89541726421575,
35.47814649701072'):
        api_key = self.api_key
        url =
f'https://maps.googleapis.com/maps/api/place/nearbysearch/json?location={loca
tion}&radius=10&type={entity}&keyword={intent}&key={api_key}'
        r = requests.get(url)
        results = r.text.split('"business_status" : "')
        results.pop(0)
        searching_for_a_place = {}
```

```
for index, place in enumerate(results):
            place_name = re.compile('{}(.*){}'.format(re.escape('name" : "'),
re.escape('"'))).findall(place)
            place_id = re.compile('{}(.*){}'.format(re.escape('"place_id" :
"'), re.escape('"'))).findall(place)
            place_rating = re.compile('{}(.*){}'.format(re.escape('"rating" :
'), re.escape(','))).findall(place)
            place_lat = place.split('"location" : {')[-1].split('"viewport" :
{')[0].split('"lat" : ')[-1].split('"lng" : ')[0].split(',')[0]
            place_lng = place.split('"location" : {')[-1].split('"viewport" :
{')[0].split('"lat" : ')[-1].split('"lng" : ')[-1].split('\n')[0]
            searching_for_a_place[f'place {index+1}'] = [place_name[0],
place_rating[0], place_id[0], place_lat,place_lng]
        return searching_for_a_place
def get_destination_distance_time_needed(self,source,destination):
        url =
"https://maps.googleapis.com/maps/api/distancematrix/json?units=imperial&"
        api_key = self.api_key
        r = requests.get(url + "origins=" + source + "&destinations=" +
destination + "&key=" + api_key)
        distance = r.json()["rows"][0]["elements"][0]["distance"]["text"]
        time = r.json()["rows"][0]["elements"][0]["duration"]["text"]

        return distance,time
    def get_data_wikipidia(self,search):
        r = requests.get('https://www.wikipedia.org/wiki/{}'.format(search))
        return r.text
    def get_text_wikipidea(self,seach_for):
        htmldata = self.get_data_wikipidia(seach_for)
        soup = BeautifulSoup(htmldata, 'html.parser')
        data_txt = []
        for data in soup.find_all("p"):
            if data.get_text() != '\n':
                data_txt.append(data.get_text().split('.'))
        return data_txt[0][0]
    def get_text_google(self,query):
        for URL in search(query, tld="co.in", num=10, stop=10, pause=2):
            try: URL_Name = URL.split('//')[1].split('/')[0].split('www.')[1]
            except:
                try: URL_Name =
URL.split('//')[1].split('/')[0].split('.')[0]
                except: URL_Name = URL.split('//')[1].split('/')[0]
            try:
                page = urlopen(URL)
                html bytes = page.read()
                html = html_bytes.decode("utf-8")
                content = html.split('</title>')
                content_ = content[-
1].split('name="description"')[1].split('content="')[1].split('" />')[0]
                return URL_Name, content_[0:100]
            except:
                continue
```

Figure 61. Virtual Assistant Backend – Web Class

```
import time
from My_Location import My_Location
from ai import AI
import requests
import threading
class Direction(object):
    def __init__(self):
        self.Assistant = AI("Map Assistant")
        self.Assistant.setvol(0.5)
        self.forced_stopped = False
        self.my_loc = My_Location()
        self.url = "https://maps.googleapis.com/maps/api/directions/json?"
        api_file = open("api-key-maps.txt", "r")
        api_key = api_file.readline()
        api_file.close()
        self.api_key = api_key
    def Threading_to_get_direction(self,destination):
        thread = threading.Thread(target=self.run, args=[destination])
        thread.daemon = True  # Daemonize thread
        thread.start()
    def run(self, destination):
        r = requests.get(self.url + "origin=" + self.my_loc.get_my_location()
+ "&destination=" + destination + "&mode=walking" + "&key=" + self.api_key)
        steps = r.json()['routes'][0]['legs'][0]['steps']
        for step in steps:
            time_for_this_step = step['duration']['text']
            go_to = step['html_instructions']
            distance_to_this_step = step["distance"]["text"]
            if 'Turn' in go_to: sentence = f'in {time_for_this_step} {go_to}'
            else: sentence = f'{go_to} for {time_for_this_step}'
            Somthing_wrong = False
            self.Assistant.say(sentence)
            while (self.my_loc.lat != step['end_location']['lat'] and
self.my_loc.lng != step['end_location']['lng']):
                new_distance =
self.check_location(self.my_loc.get_my_location(),
f"{step['end_location']['lat']},step['end_location']['lng']")
                if new_distance <= distance_to_this_step:
                    remaining_distance = distance_to_this_step - new_distance
                    self.Assistant.say(
                        f"You are in the right path, you still have like
{remaining_distance} meters to go to the next step")
                else:
                    Somthing_wrong = True
                    break
                if self.forced_stopped: #if the user stiops the directions
                    break
                time.sleep(30)
            if Somthing_wrong:
 self.Assistant.say("waww! Something Wrong with your direction, you are going
in the wrong way!! try again please")
                return None
            if self.forced_stopped:
                self.Assistant.say("OK! will stop the direction for you")
                return None
```

```
def check_location(self, actual_location, destination_checking_point):
        req = requests.get(
            self.url + "origin=" + actual_location + "&destination=" +
destination_checking_point + "&mode=walking" + "&key=" + self.api_key)
        step = req.json()['routes'][0]['legs'][0]['steps'][0]
        distance = step["distance"]["value"]
        return distance
    def stop(self):
        self.forced_stopped = True
```

Figure 62. Virtual Assistant Backend – Directions Class

```
from URL_Request_Respond import Custom_Action
from ai import AI
from Date import Date
from facts import Fact
from jokes import Joke
from News import News
from Videos import music
from weather import Weather
from web import inflow
from Email import Email
from Directions import Direction
from My_Location import My_Location
from Location import Location
#from Image_Processing.Avoid_Obstacles import Avoid_obstacles
IOB = AI("IOB")
get_respond = Custom_Action()
global musics, google,
email,weather,news,date,fact,joke,my_loc,direction,location,avoid_obstacles,m
musics = music()
google = inflow()
email = Email()
weather = Weather()
news = News()
date = Date()
fact = Fact()
joke = Joke()
my_loc = My_Location() #in this class we should implemnt the GPS module
direction = Direction()
location = Location()
#avoid_obstacles = Avoid_obstacles()

def get_joke():
    IOB.say("do you know that:")
    funny = joke.get_joke()
    print(funny)
    IOB.say(funny)
def get_date():
    IOB.say("sure! The date today is:")
    print(date.get_date_time()[0])
    IOB.say(date.get_date_time()[0])
def get_time():
    IOB.say("sure! the time now is:")
    print(date.get_date_time()[1])
    IOB.say(date.get_date_time()[1])
def get_fact():
    IOB.say("sure! do you know that:")
    print(fact.funfact())
    IOB.say(fact.funfact())
def get_News():
    IOB.say(f"sure the news for today are as follows:")
    print(news.get_news())
    IOB.say(news.get_news())
def play_musics(query, type = "youtube"):
    IOB.say(f"sure will play for you {query}")
    musics.Threading_to_play(query, type)
```

```
def play_next_music():
    IOB.say(f"sure will play the next one")
    musics.play_next()
def play_prev_music():
    IOB.say(f"sure will play for you the previous one")
    musics.play_prev()
def stop_music():
    IOB.say(f"sure will stop it!")
    musics.stop()
def pause_music():
    IOB.say(f"sure will pause it!")
    musics.pause()
def get_Weather_info(loaction_name,location,date):
    IOB.say(f"The temperature in {loaction_name} is:")
    print(f"{weather.get_wether_info(location)[0]} degree celsius")
    IOB.say(f"{weather.get_wether_info(location)[0]} degree celsius")
    IOB.say(f"and the weather details in {loaction_name}  is:")
    print(weather.get_wether_info(location)[1])
    IOB.say(weather.get_wether_info(location)[1])
def get_data_Wikipedia(data):
    IOB.say(f"Sure! with respect to wikipedia:")
    print(data.split('"}')[0])
    print(google.get_text_wikipidea(data.split('"}')[0]))
    IOB.say(google.get_text_wikipidea(data.split('"}')[0]))
def get_data_from_google(data):
    name,content = google.get_text_google(data)
    IOB.say(f"Sure! with respect to {name} website:")
    print(name)
    IOB.say(content)
    print(content)
def get_nearest_place(facility_type,product,location): #location = 'lat,long'
    IOB.say(f"Sure! the nearest 3 {facility_type} with are as follows:")
    places = google.get_info_places(product,f'{product} shop',location =
location)
    #print(product,places)
    for index,key in enumerate(places.keys()):
        print(f'{key}, which its name is : {places[key][0]} and its rate is:
{places[key][1]}')
        IOB.say(f'{key}, which its name is : {places[key][0]} and its rate
is: {places[key][1]}')
        if index == 2:
            break
def
get_destination_distance_time_needed(source,destination,source_,destination_)
:#source_,destination_ = 'lat,long'
    IOB.say(f"Sure! working on it")
    time,distance =
google.get_destination_distance_time_needed(source_,destination_)
    IOB.say(f"the required distance from {source} to {destination} is
{distance}")
    print(f"the required distance from {source} to {destination} is
{distance}")
    IOB.say(f"the required time from {source} to {destination} is {time}")
    print(f"the required time from {source} to {destination} is {time}")
def get_location(facility_type,location):
   #takes facilty type or 'any' and location as name and returns the lat,lang
   if facility_type == 'any':
```

```
        return location.get_location(location)
    else:
        return location.get_location(f'{facility_type} {location}')
#return them in (lat,lang)
def get_my_location_Lat_Long():
    return my_loc.get_my_location()
def get_home_location_Lat_Long():
    home_location_file = open("home_location.txt", "r")
    home_location = home_location_file.readline()
    home_location_file.close()
    return home_location
def get_work_location_Lat_Long():
    work_location_file = open("work_location.txt", "r")
    work_location = work_location_file.readline()
    work_location_file.close()
    return work_location
def send_emial(destination,subject,content):
    IOB.say(f"Sure! I will send an Email for {destination} with {subject}
subject and {content} content")
    process = email.Send_Email(destination,subject,content)
    IOB.say(process+" sir")
def
extract_data_for_email(entities_values_extracted):##########################
    email, subject, content = entities_values_extracted
    return  email, subject, content
def get_directions_to_location(distination):
    direction.Threading_to_get_direction(distination)
def stop_direction():
    direction.stop()
def Call_Taxi():
    IOB.say(f"Calling a Taxi is still under Constructions, developers are
still working on it")
def Call_Delivery():
    IOB.say(f"The Delivery is still under Construction, and developers are
still working on this feature")

RUNS THE IMAGE PROCESSING FILE
#avoid obstacles.Threading to get help()MAIN FUNCTION
while True:
    message = IOB.listen()
    confirm = IOB.just_listen_for_confirm()
    if confirm:
        get_my_location = get_my_location_Lat_Long()
        get_home_location = get_home_location_Lat_Long()
        get_work_location = get_work_location_Lat_Long()
        responses,intity_all,entities_values_extracted_all =
get_respond.get_entities(message,URL=
"http://localhost:5000/webhooks/rest/webhook/")
        for index,responde in enumerate(responses):
            intity, entities values extracted =
intity_all[index],entities_values_extracted_all[index]
            print(responde,intity,entities_values_extracted)
            if responde == 'None':
IF USER ASKS FOR PLAYING MUSICS
                if intity == 'play_music' and len(entities_values_extracted)
== 0: play_musics("musics")
```

```
elif intity == 'play_music':
                    try:
                        play_musics(entities_values_extracted['artist'])
                    except:
                        play_musics(entities_values_extracted['song'])
IF USER ASKS FOR PLAYING NEXT MUSIC
                elif intity == 'next_music':
                    try:
                        play_prev_music()
                    except:
                        IOB.say("You haven't started a new music, you should
tell me to start for you a music as first step.")
IF USER ASKS FOR PLAYING NEXT MUSIC
                elif intity == 'prev_music':
                    try:
                        play_next_music()
                    except:
                        IOB.say("You haven't started a new music, you should
tell me to start for you a music as first step.")
IF USER ASKS FOR PAUSING MUSIC
                elif intity == 'pause_music':
                    try:
                        pause_music()
                    except:
                        IOB.say("You haven't started a new music, you should
tell me to start for you a music as first step.")
IF USER ASKS FOR CLOSSING MUSIC
                elif intity == 'stop_music':
                    try:
                        pause_music()
                    except:
                        IOB.say("You haven't started a new music, you should
tell me to start for you a music as first step.")
IF USER ASKS FOR Joke
                elif intity == 'joke':
                    get_joke()
IF USER ASKS FOR Facts
                elif intity == 'facts':
                    get_fact()
IF USER ASKS FOR GOOGLE SEARCH ABOUT PLACES
                elif intity == 'search_location_provider': #places or
facility type
                        #Time or places needed for a facility type
                        if 'facility_type' in
entities values extracted.keys():
                            if 'clock' in entities_values_extracted.keys():
#if time to a distance needed #######should add saida to kfc beirut
                    # from his location to a facilty type in a location
                                if 'my_location' in
entities_values_extracted.keys() and 'location' in
entities_values_extracted.keys():#get_destination_distance_time_needed(source
location name, destination facility type in location or without location, and
lat and long of the source and distination
                                    get_destination_distance_time_needed('your
location',f'{entities_values_extracted["facility_type"]} in
{entities_values_extracted["location"]}',get_my_location,get_location(entitie
```

```
s_values_extracted['facility_type'],entities_values_extracted['location']))
            # from his home_location to a facilty type in a location
                                elif 'home_location' in
entities_values_extracted.keys() and 'location' in
entities_values_extracted.keys():

get_destination_distance_time_needed('your home
location',f'{entities_values_extracted["facility_type"]} in
{entities_values_extracted["location"]}',get_home_location,
get_location(entities_values_extracted['facility_type'],entities_values_extra
cted['location']))
                    # from his work location to a facilty type in a location
                                elif 'work_location' in
entities_values_extracted.keys() and 'location' in
entities_values_extracted.keys():get_destination_distance_time_needed('your
work location',f'{entities_values_extracted["facility_type"]} in
{entities_values_extracted["location"]}',get_work_location,
get_location(entities_values_extracted['facility_type'],entities_values_extra
cted['location'])) # from his location to a facilty type
                                elif 'my_location' in
entities_values_extracted.keys():get_destination_distance_time_needed('your
location',entities_values_extracted['facility_type'],get_my_location,
get_location(entities_values_extracted['facility_type'],get_my_location))
                                # from his home location to a facilty type
                                elif 'home_location' in
entities_values_extracted.keys():
get_destination_distance_time_needed('your home
location',entities_values_extracted['facility_type'],get_home_location,
get_location(entities_values_extracted['facility_type'],get_home_location))
                                # from his work location to a facilty type
                                elif 'work_location' in
entities_values_extracted.keys():get_destination_distance_time_needed('your
work location',entities_values_extracted['facility_type'],get_work_location,
get_location(entities_values_extracted['facility_type'],get_work_location))
                                # to a facilty type
                                else:
get_destination_distance_time_needed('your
location',entities_values_extracted['facility_type'],get_my_location,
get_location(entities_values_extracted['facility_type'], get_my_location))
                            # if the user needs facility types in a location
                            elif 'location' in
entities_values_extracted.keys():get_nearest_place(entities_values_extracted[
'facility_type'],entities_values_extracted['facility_type'],
get_location(entities_values_extracted['facility_type'],entities_values_extra
cted['location']))
             # if the user needs the  a facility types in his home location
                            elif 'home_location' in
entities_values_extracted.keys():

get_nearest_place(entities_values_extracted['facility_type'],

entities_values_extracted['facility_type'], get_home_location)
            # if the user needs the  a facility types in his work location
                            elif 'work_location' in
entities_values_extracted.keys():#if the user mentions that he need
faciliteies beside his
homeget_nearest_place(entities_values_extracted['facility_type'],
```

```
entities_values_extracted['facility_type'], get_work_location)
         # if my_location is mention or not same we take the user's location
                            else:
get_nearest_place(entities_values_extracted['facility_type'],
entities_values_extracted['facility_type'],get_my_location)
                        #Time needed from a location to a location
                        elif 'clock' in entities_values_extracted.keys():
                            # from his location to a location
                            if 'my_location' in
entities_values_extracted.keys() and 'location' in
entities_values_extracted.keys():get_destination_distance_time_needed('your
location',f'{entities_values_extracted["location"]}',
get_my_location, get_location('any',entities_values_extracted['location'])) #
from his home_location to a location
                            elif 'home_location' in
entities_values_extracted.keys() and 'location' in
entities_values_extracted.keys():get_destination_distance_time_needed('your
home location',f'{entities_values_extracted["location"]}',
get_home_location, get_location('any',entities_values_extracted['location']))
                            # from his work location to a location
                            elif 'work_location' in
entities_values_extracted.keys() and 'location' in
entities_values_extracted.keys():get_destination_distance_time_needed('your
work location',f'{entities_values_extracted["location"]}', get_work_location,
get_location('any',entities_values_extracted['location']))
                            # to a location
                            else:
                                get_destination_distance_time_needed('your
location',f'{entities_values_extracted["location"]}',

get_my_location,get_location('any',entities_values_extracted['location']))
IF USER ASKS FOR GOOGLE SEARCH ABOUT INFORMATION
                elif intity == 'get_info':#needs some modification on googles
side
                    #waether
                    if 'weather' in entities_values_extracted.keys(): #needed
to customize the time only so the date should be added
                        # waether - date -
location/home_location/work_location
                        if 'date' in entities_values_extracted.keys():
                            if 'location' in
entities_values_extracted.keys():get_Weather_info(entities_values_extracted['
location'], get_location('any',
entities_values_extracted['location']),entities_values_extracted['date'])
                            elif 'home_location' in
entities_values_extracted.keys():
                                get_Weather_info("your home
location",get_home_location, entities_values_extracted['date'])
                            elif 'work_location' in
entities_values_extracted.keys():
                                get_Weather_info("your work
location",get_work_location, entities_values_extracted['date'])
                            else:
                                get_Weather_info("your location",
_my_location, entities_values_extracted['date'])
                    # waether - location/home_location/work_location - date
```

```
                        elif 'location' in entities_values_extracted.keys():
                            if 'date' in
entities_values_extracted.keys():get_Weather_info(entities_values_extracted['
location'], get_location('any',
entities_values_extracted['location']),entities_values_extracted['date'])
                            else:
get_Weather_info(entities_values_extracted['location'], get_location('any',
entities_values_extracted['location']),"now")
                        elif 'home_location' in
entities_values_extracted.keys():
                            if 'date' in entities_values_extracted.keys():
                                get_Weather_info("your home",
get_home_location,entities_values_extracted['date'])
                            else:
                                get_Weather_info("your home",
                                                 get_home_location, "now")
                        elif 'work_location' in
entities_values_extracted.keys():
                            if 'date' in entities_values_extracted.keys():
                                get_Weather_info("your
work",get_work_location, entities_values_extracted['date'])
                            else:
                       get_Weather_info("your work",get_work_location, "now")
                        # waether - now - my_location
                        else:
                            get_Weather_info("your location",get_my_location,
"now")
                    # News
                    elif 'news' in entities_values_extracted.keys():
                        get_News()
                    # Clock
                    elif 'clock' in entities_values_extracted.keys():
                        get_time()
                    # Date
                    elif 'date' in entities_values_extracted.keys():
                        get_date()
                    #asking for a product
                    elif 'product' in
entities_values_extracted.keys():get_data_Wikipedia(entities_values_extracted
['product'])
                    #asking for person or any qurerry that couldnt be
performed by wikipidea
                    else: get_data_from_google(message)
IF USER ASKS FOR getting a direction
                elif intity == 'get_direction':
                    if 'facility_type' in entities_values_extracted.keys():
                        #rastuarant in a location
                        if 'location' in entities_values_extracted.keys():
get_directions_to_location(get_location(entities_values_extracted['facility_t
ype'], entities_values_extracted['location']))
                        # rastuarant beside my work
                        elif 'work_location' in
entities_values_extracted.keys():get_directions_to_location(get_location(enti
ties_values_extracted['facility_type'],

get_work_location))
                        # rastuarant beside my home
```

```
                        elif 'home_location' in
entities_values_extracted.keys():get_directions_to_location(get_location(enti
ties_values_extracted['facility_type'], get_home_location))
                        # Rastaurant beside me
                        else:
get_directions_to_location(get_location(entities_values_extracted['facility_t
ype'],get_my_location))
                    #going to a location
                    elif 'location' in entities_values_extracted.keys():
                        get_directions_to_location(get_location('any',
entities_values_extracted['location']))
                    # going to my work
                    elif 'work_location' in entities_values_extracted.keys():
                        get_directions_to_location(get_work_location)
                    elif 'home_location' in entities_values_extracted.keys():
                        get_directions_to_location(get_home_location)
                #if the user asks to stop the direction
                elif intity == 'stop_direction': stop_direction()
IF USER ASKS FOR Sending an Email
                elif intity == 'text_recognition' and 'facility_type' in
entities_values_extracted.keys():
                    if entities_values_extracted['facility_type'] == 'email':
                        email,subject,content =
extract_data_for_email(entities_values_extracted)
                        send_emial(email,subject,content)
IF USER ASKS FOR a Taxi
                elif intity == 'call_taxi': Call_Taxi()
USER ASKS FOR a Delivery
                elif intity == 'call_delivery': Call_Delivery()
IF USER ASKED FOR SOMTHING NOT TRAINED ON
                else:
IOB.say("Please rephrase your sentence, I didn't Understand what you meant!")
            else:
                IOB.say(responde)
    else:
        IOB.say("Ok, so please rephrase your sentence")
```

Figure 63. Virtual Assistant Backend – Main Class

```
import cv2
import numpy as np
import os
import threading
from Avoid_obstacles_Asiistant import AI
class Avoid_obstacles:
    def __init__(self):
        print("Initializing the Obstacles Avoidance")
        self.testmode = 2
        self.key = ''
        self.stop_helpping = False
        self.assistant = AI()
    def forward(self):  # ... add onto the left
        print("No obstacles detected")
        self.assistant.say("No obstacles detected")
    def backward(self):
        print("This is a closed way, go back please!")
        self.assistant.say("This is a closed way, go back please!")
    def right(self):
        print("go slightly to the right please!")
        self.assistant.say("go slightly to the right please!")
    def left(self):
        print("go slightly to the left please!")
        self.assistant.say("go slightly to the left please!")
    def stop(self):
        print("Stop Helping you")
        self.assistant.say("Stop Helping you")
    def calc_dist(self,p1, p2):
        x1 = p1[0]
        y1 = p1[1]
        x2 = p2[0]
        y2 = p2[1]
        dist = np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
        print(dist)
        return dist
    def getChunks(self,l, n):
        a = []
        for i in range(0, len(l), n):
            a.append(l[i:i + n])
        return a
    def Avoid(self, camp_port):
        cap = cv2.VideoCapture(camp_port)
        try: if not os.path.exists('data'): os.makedirs('data')
        except OSError: print('Error: Creating directory of data')
        StepSize = 5
        currentFrame = 0
        if self.testmode == 1:
            self.F = open("./data/imagedetails.txt", 'a')
            self.F.write("\n\nNew Test \n")
        while (not self.stop_helpping):
            print(self.stop_helpping)
            _, frame = cap.read()
            name = './data/frame' + str(currentFrame) + '.jpg'
            img = frame.copy()
            blur = cv2.bilateralFilter(img, 9, 40, 40)
```

```
            edges = cv2.Canny(blur, 50, 100)
            img_h = img.shape[0] - 1
            img_w = img.shape[1] - 1
            EdgeArray = []
            for j in range(0, img_w, StepSize):
                pixel = (j, 0)
                for i in range(img_h - 5, 0, -1):
                    if edges.item(i, j) == 255: pixel = (j, i) break
                EdgeArray.append(pixel)
            for x in range(len(EdgeArray) - 1):
                cv2.line(img, EdgeArray[x], EdgeArray[x + 1], (0, 255, 0), 1)
            for x in range(len(EdgeArray)):
                cv2.line(img, (x * StepSize, img_h), EdgeArray[x], (0, 255,
0), 1)
            chunks = self.getChunks(EdgeArray, int(len(EdgeArray) / 3))  # 5
            c = []
            for i in range(len(chunks) - 1):
                x_vals = []y_vals = []
                for (x, y) in chunks[i]:
                    x_vals.append(x)
                    y_vals.append(y)
                avg_x = int(np.average(x_vals))
                avg_y = int(np.average(y_vals))
                c.append([avg_y, avg_x])
                cv2.line(frame, (320, 480), (avg_x, avg_y), (255, 0, 0), 2)
            forwardEdge = c[1]
            print(forwardEdge)
            cv2.line(frame, (320, 480), (forwardEdge[1], forwardEdge[0]), (0,
255, 0), 3)
            cv2.imwrite(name, frame)
            y = (min(c))
            print("y = ",y)
            if forwardEdge[0] > 250:  # 200 # >230 works better
                if y[1] < 310: self.left()direction = "left "
                else: self.right()direction = "right "
            else:
                self.forward()direction = "forward "
            if self.testmode == 1:
                self.F.write("frame" + str(currentFrame) + ".jpg" + " | " +
str(c[0]) + " | " + str(c[1]) + " | " + str(c[2]) + " | " + direction + "\n")
                currentFrame += 1
            if self.testmode == 2:
                cv2.imshow("frame", frame)
            k = cv2.waitKey(5) & 0xFF  ##change to 5
            if k == 27: break
        if self.stop_helpping:
            self.stop()
            cv2.destroyAllWindows
            cap.release()
    def stop_help(self):
        self.stop_helpping = True
a = Avoid_obstacles()
a.Avoid(0)
```

Figure 64. Object Detection – Camera Class

```
int const PULSE_SENSOR_PIN = 0;
int Signal; int Threshold = 550; int trigPin = 11; echoPin = 12;int buzzer = 13; long duration, cm, inches;
void setup() {
 pinMode(buzzer,OUTPUT); Serial.begin (9600); pinMode(LED_BUILTIN,OUTPUT);
 pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT); Threshold = 95;}
void loop() {digitalWrite(trigPin, LOW); delayMicroseconds(5); digitalWrite(trigPin, HIGH);
 delayMicroseconds(10); digitalWrite(trigPin, LOW);pinMode(echoPin, INPUT);
 duration = pulseIn(echoPin, HIGH); cm = (duration/2) / 29.1; // Divide by 29.1 or multiply by 0.0343
 if cm <= 50{digitalWrite(buzzer, HIGH); delay(cm + 100); digitalWrite(buzzer, LOW); delay(cm + 100); }
 Signal = analogRead(PULSE_SENSOR_PIN);
 if(Signal > Threshold){
   digitalWrite(LED_BUILTIN,HIGH); } else {digitalWrite(LED_BUILTIN,LOW); }} delay(250); }
```

Figure 65. Object Detection – Arduino Class

```
#include <Wire.h>
#include <"CurieIMU.h">
int const PULSE_SENSOR_PIN = 0; int Signal; int Threshold = 550;
void setup() { Serial.begin(9600);
 CurieIMU.begin();CurieIMU.setDetectionThreshold(CURIE_IMU_SHOCK, 1500); // 1.5g = 1500 mg
 CurieIMU.setDetectionDuration(CURIE_IMU_SHOCK, 50);   // 50ms
 CurieIMU.interrupts(CURIE_IMU_SHOCK);pinMode(LED_BUILTIN,OUTPUT); Wire.begin();Threshold
= 95; }
void loop() { Signal = analogRead(PULSE_SENSOR_PIN);
 if(Signal > Threshold){ Serial.print("HIGH Rate Detected");Wire.beginTransmission(44);
digitalWrite(LED_BUILTIN,HIGH); Wire.write(1); Wire.endTransmission();
  } else {Serial.print("LOW Rate Detected");digitalWrite(LED_BUILTIN,LOW);
 digitalWrite(LED_BUILTIN,HIGH); Wire.beginTransmission(44); Wire.write(0);
Wire.endTransmission();}
 if (CurieIMU.getInterruptStatus(CURIE_IMU_SHOCK)){ Serial.print("Fall
detected");Wire.beginTransmission(45); Wire.write(1); Wire.endTransmission();}
 else{Serial.print("Fall not detected");Wire.beginTransmission(45); Wire.write(0);
Wire.endTransmission(); } delay(10); }}
```

Figure 66. Heartrate and Falling Detection – Arduino Program

```
import smbus
from datetime import datetime
import time
from Asistant.Email import Email
bus = smbus.SMBus(0)
address = 0x60
emergency_email = Email()
def bearing3599():
    bear1 = bus.read_byte_data(address, 2)
    bear2 = bus.read_byte_data(address, 3)
    bear = (bear1 << 8) + bear2
    bear = bear / 10.0
    return bear
heart_call = {'0':[],
              '1':[]
              }
while True:
        now = datetime.now()
        time_now = now.strftime("%H:%M:%S")
        heart_rate_Call = bearing3599()      #this returns the value as a
byte between 0 and 255.
        if heart_rate_Call == 1:
            print("Heart high rate detected!")
        else:
            print("Heart is good")
        heart_call[heart_rate_Call].append(time_now)

        if len(heart_call[1]) >= 3:
            print("Heart high rate detected! 3 times")
            process = emergency_email.Send_Email("redcross@gmail.com",
"Emergency Message", "Help me I have a very high rate for 3 times for 3 mins
continuously")
            heart_call[0] = []
            heart_call[1] = []
        elif len(heart_call[0]) >= 3:
            print("Heart low rate detected! 3 times")
            heart_call[0] = []
            heart_call[1] = []
        time.sleep(60)
```

Figure 67. Heartrate and Falling Detection – Raspberry PI Program