Sudan University of Science and Technology

College of Engineering

School of Electrical and Nuclear Engineering

Design and Implementation of obstacles avoiding blind stick

تصميم وتنفيذ عصا متجنّبة للعوائق لفاقدي البصر

A Project Submitted In Partial Fulfillment for the Requirements of the Degree of B.Sc. (Honor) In Electrical Engineering

Prepared By:

1. Lina Omer Mahjoub Ali
2. Montaha Gibril Fadlemula FadlElseed
3. Nafisa Yasin Abdalla Yousif
4. Rayan Moustafa Eissa Uthman

Supervised By:

Dr. Awadalla Taifour Ali

November 2020

الآية

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

﴿أَفَلَمْ يَسِيرُواْ فِى الْأَرْضِ فَتَكُونَ لَهُمْ قُلُوبٌ يعْقِلُونَ بِهَا أَوْ آذَانٌ يَسْمَعُونَ بِهَا فَإِنَّهَا لَا تَعْمَى الْأَبْصَارُ

وَلَكِن تَعْمَى الْقُلُوبُ الَّتِي فِي الصُّدُورِ﴾

[سورة الحج: الآية 46]

DEDICATION

To those who face our laborious life…

Deprived of sight…

We wish you all a beautiful life…

With less struggle...

And overjoyed happiness…

ACKNOLEDGMENT

Our sincere gratitude to our Supervisor Professor Awadallah Taifour Ali for his elegant manners, dedicated guidance and his wealthy knowledge which brought this project to the light.

To doctor Omer Abdelrazag Sharif for his support and constructive criticize throughout this project.

To our sincere brother first then our polite colleague Mohammed Hafiz.

Abstract

Navigation might be a simple task for ordinary people yet it's a challenging procedure for those with vision impairment, this project presents a smart assistive cane based on ultrasonic sensors measures which are processed by a Raspberry Pi, and with aid of dc gear motors an avoidance algorithm is implemented. Experimental tests have showed promising results in both obstacles detection and avoidance, upstairs detection, downstairs and holes detection.

مستخلص

يتعرض المعاق بصريا أثناء تنقله إلى الخطر لعدم معرفته التامة بالبيئة المحيطة، الأمر الذي يستدعي استعمال وسيلة تساعده على التعرف على العقبات في طريقه، ثم تجاوزها والعودة لمساره الأصلي.

يهدف هذا المشروع إلى تطبيق خوارزمية تعتمد على قياسات محساسات الموجات فوق الصوتية لإكتشاف العقبات، ثم معالجة هذه القياسات بواسطة وحدة التحكم المسؤولة عن اختيار القرار المناسب و تنفيذه عبر محركات التيار المستمر.

أُختبر النظام وتم التحصل على نتائج واعدة، حيث أُكتشفت العقبات الموجودة في مسار الحركة وتجاوزها بنجاح، وتعرف النظام على وجود سلالم صاعدة، وسلالم هابطة ، و الحفر.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OFABBREVIATIONS

RAM	Random Access Memory
ROM	Read Only Memory
CD	Compact Disk
LCD	Liquid Crystal Display
OS	Operating System
CPU	Central Processing Unit
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
ARM	Advanced Risc Machine
USB	Universal Serial Bus
GPS	Global Position System
Laser	Light Amplification by Stimulated Emission of Radiation
Radar	Radio Detection and Ranging
IR	Infrared
AC	Alternating Current
DC	Direct Current
EMF	Electromotive Force
RC	Radio Control
CNC	Computer Numerical Control
COM	Common
NO	Normally Open
NC	Normally Close
ENVS	Electro-Neural Vision System
SMS	Short Message Service
YOLO	You Only Look Once
SOC	System On Chip
GPU	Graphics Processing Unit
SD	Secure Digital
HDMI	High Definition Multimedia Interface
GPIO	General Purpose Input/Output
BCM	Body Control Module
TTL	Transistor-Transistor-Logic

VCC	Voltage Common Collector
IN1, IN2	Input 1, Input 2
GND	Ground
PPR	Poly Propylene
NOOBS	New Out Of Box System
PCB	Printed Circuit Board
VNC	Virtual Network Computing
SSH	Secure Shell
M4A	MPEG-4
WAV	Waveform Audio File Format

LIST OF SYMBOLS

λ	Wavelength, m
T	Period, second
f	Frequency, Hz
C	Ultrasound velocity, m/sec
L	Distance measured by ultrasonic sensor, cm
P	Power of motor, hp
N	Angular speed of motor, RPM
D	Distance range of ultrasonic, m
θ	Ultrasonic radiation angle, degree
o	Upper sensor position, cm
x	Bottom sensor position, cm
s	Stair sensor position, cm
h	Stick length, cm
F	Angle of sensor's inclination from vertical axis, degree
Y	Set distance value, cm
Vin	Voltage input to echo
Vout	Voltage output from raspberry pi
R1	Resistor 1KΩ
R2	Resistor 2KΩ

CHAPTER ONE

INTRODUCTION

1.1 General Concept

Losing one sense or more makes the person an impaired. Hearing, vision and physical disabilities have noticeable impact on the daily life of the impaired people as they deprive them from natural interactions with the community let alone the psychological, economical and above all social influence.

The eye is the most important organ in human body as 83% of information which human being gets from the environment is via sight [1]. Visually impaired has lack of these information therefore they may face the danger of collision or falling in holes. Their vision needs to be substituted by either tactile sense or auditory.

Both people who can perceive apart of visual light and those who suffer from complete absence of visual light are introduced to the world as visually impaired, there are 123.7 million people in the world suffering from moderate,severe impaired or complete blindness according to the International Classification of Diseases [2].

1.2 The Problem Statement

The main problem which faces visually impaired people when they interact with their surroundings is to avoid obstacles based on their memorized locations or relying on external guidance, the things that limits their ability to navigate conveniently. Assistive aids for orientation and mobility have a long history of developing that extend to the postwar period .More recent efforts like specialized eye glasses, smart canes and other techniques focused on detecting and indicating obstacles presenes , without leading users to the appropriate path.

1.3 Objective

The main goals of this project is to develop an affordable, light weight and unconductive blind stick able to detect and avoid obstacles, stairs and holes with guidance.

The objectives of this project are to :

- Design of avoiding obstacles system.
- Implementation of the system.

1.4 Methdology

- Choosing the appropriate sensor for detecting.
- Choosing the appropriate motor and wheels for avoiding.
- Choosing the appropriate controller and its programming language.
- Writing the programing code.
- Designing and testing the simulation circuit.
- Designing the stick body by determining the location of the hardware components on it.
- Testing the system and analyzing the results.
- Calculating the system's efficiency.

1.5 Layout

The work of this project is divided into five chapters. Chapter two illustrates the theoretical background of control systems, embedded systems and basics of microcontrollers, sensors, motors and literature review. Chapter three discusses in details the system's hardware and software. Chapter four concern with calculations, project design, implementation, testing results and disscusion. Chapter five show the conclusion and recommendations.

CHAPTER TWO

THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1 Control Systems Preview

Engineering is concerned with studying, understanding and analyzing problems in environment and controlling the materials, resources and forces of nature to create technical solutions and provide safe and reliable economic products. Control engineering is a branch of engineering science which relies on a rigid mathematical base, it plays a vital role in the advancement of engineering, science and industry. Automatic control provides the means to achieve optimum performance in dynamic systems, improving productivity and reducing the hardship and risks of routine manual operations, even more than that, automatic control has become an important part of modern industrialization such as robotic systems, space-vehicle systems, and missile-guidance systems. A system is a combination of components that act together and perform a certain objective [3]. The system's concept is not limited to physical systems only, but can be applied to social, economic and biologic phenomena. Control process aims to monitor these systems in order to achieve the desired response. There are two common classes of control systems:

2.1.1 Open -loop control systems

In the open-loop system shown in Figure 2.1, the controller operates independently of the output value, where the output depends on the input and the system's transfer function. The controller output is not measured or fed back for comparison with the reference input, thus in the case of internal or external disturbances the open loop control system will not give the desired output accurately. Control of washers, fans, and traffic lights, are examples of open loop control systems.

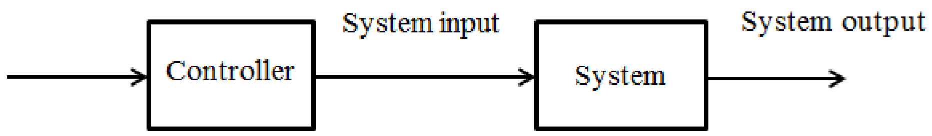

Figure 2.1: Open-loop control system

The major advantages of open-loop control systems are as follow:

- Simple construction and ease of maintenance.
- Less expensive than a corresponding closed-loop systems.
- No stability problems.
- Convenient when the output is hard to measure or measuring the output precisely is economically not feasible. The major disadvantages of the open-loop control systems are:
- Disturbances may cause errors, and the output might be different from the desired.
- To maintain the required output's quality, recalibration is necessary from time to time.

2.1.2 Close-loop control systems

Close loop control system is a system that maintains a prescribed relationship between the output and the reference input by comparing them and using the difference (error signal) as a mean of control. The output value is continuously measured by a sensor which converts this measurement into an electric signal and passes it back to the controller as shown in Figure 2.2. In the presence of disturbances, the controller tends to reduce the difference between the system's output and the reference input, it does so basis on the error's signal value. Here, only unpredictable disturbances are specified, since predictable or known disturbances can be compensated within the system.

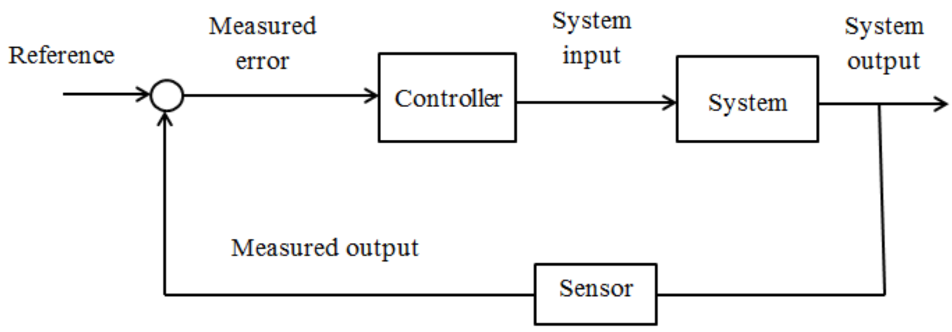

Figure 2.2: Close-loop control system

The major advantage of close-loop control systems is :

- The use of the feedback makes the system's response relatively insensitive to external disturbances and internal variations in system's parameters.

The major disadvantages of close-loop control systems are as follow:

- Closed-loop control systems tend to overcorrect errors which may cause oscillations and create a major stability problem.
- The number of components used in a closed-loop control system is more than that for a corresponding open-loop control system. Thus, the closed-loop control system is generally higher in cost and power [3].

2.2 Embedded Systems

Embedded systems are electronic systems which are combinations of software and hardware components designed for a specific function or applications. An embedded system is a reliable, real time control system and autonomous, human or network interactive, operating on diverse physical variables in diverse environments.

A computer is a system that has the following or more components:

- A microprocessor.
- A large memory of the following:
    a. Primary memory or semiconductor memories: Random Access Memory (RAM), Read Only Memory (ROM), and fast accessible caches.
    b. Secondary memory: is a magnetic memory located in the hard disk, optical memory in the Compact Disk (CD) or memory sticks in mobile computers using with different user programs can be loaded into primary memory and run.
- Input units such as keyboard, mouse, scanners.
- Output units such as Liquid Crystal Display (LCD) screen, video monitor, printers.
- Network units such as Ethernet card, bus drivers.
- An Operating System (OS) has a fixed program for general purposes and extra software applications in secondary memory.

The embedded system differs from computer in three main points:

- Its software usually stored in the ROM or flash memory and does not need a secondary hard disk or CD memory as in a computer.
- Its software may perform concurrently a series of tasks or processes.
- It embeds a real-time operating system that supervise the application software running on hardware.

2.2.1 Embedded system microcontroller

The controller is the main component of the embedded systems which is a device, or set of devices, that manages, commands, directs or regulates the system's behavior. Microcontroller is single low cost computer chip capable of storing and running a program and connecting with other electronic devices. It contains:

- **Central processing unit**

Central Processing Unit (CPU) is the controller's brain that administers all activities in the system and performs all operations on data. It continuously performs two operations: fetching and executing instructions. It understands and executes instructions based on a set of binary code from the memory.

- **Memory**

It's used to store data or program, microcontroller has many types of memories, all types or just few of them might be found. The main types of memories are:

- **Random access memory**

It's a general-purpose memory which usually stores user's data that used in a program. This memory loses stored data after power removal.

- **Read only memory**

It's a type of memory which usually holds program or fixed user data. It is programmed during the manufacturing process and their contents cannot be changed by the user.

- **Analog to digital converter**

Analog to Digital Converter (ADC) is used to convert the analog signal (sensor's signal) to digital signal.

- **Digital to analog converter**

Digital to Analog Converter (DAC) performs inversely to ADC, it converts the digital signal to analog.

- **Timers and counters**

Timers and counters perform all the function of timing and counting in the microcontroller.

2.2.2 Comparison between raspberry pi and ARDUINO

Both Raspberry pi and ARDUINO are used in embedded systems, there are many differences between them illustrated in Table 2.1

Table 2.1: ARDUINO and raspberry pi comparison

Property	ARDUINO	Raspberry pi
Processor	ATmega processors	ARM processors
RAM memory	Ranges from (2-32) MB and up to 96MB in some versions	512MB and up to 4GB in recent versions
ROM	Ranges from (32-512) KB.	SD card (2-16GB)
Architecture	8 bits	32 bits
Speed	The speed of different versions ranges from 8-48MHZ	700MHZ and it can be accelerated to speed of 1GH
Input voltage	7-12 volt	5 volt
Min power	42 mA, 0.3w	700 mA, 3.5w
Operating system	None	Linux distributions
Integrated development environment	ARDUINO IDE	Scratch, idle, anything with Linux support
Programming language	ARDUINO c, scratch	Python, c, and any language supported by Linux
Ethernet port, Wi-Fi, camera, monitor, and audio	It require external hardware	Supported and build on board
USB ports	Mostly one port for powering	Two or four ports

2.2.3 Raspberry pi

The beginning of the twenty–first century witnessed a huge expansion in microcontrollers manufacturing and the development. In 2006 a group of Cambridge university scientists adopted the idea of producing a small, cheap, and educational computer for students and amateurs called raspberry pi. It aimed to provide the necessary skills for computer science's students [4].

Raspberry pi took years to work on before it became usable. Many companies contributed in raspberry pi development such as Advanced Risc Machine (ARM), raspberry pi processor relied on ARM architecture and it used ARM cortex core processors. With the development of processors technologies, it became possible to obtain a small computer with multimedia function at low cost. In 2011, the first models were being produced and tested, two versions of the raspberry pi were manufactured, namely model-A and model-B, the difference between them is in the RAM space and the number of the Universal Serial Bus (USB) ports, where model-A has a RAM with 256MB and two USB ports while model-B has 512MB RAM and four USB ports.

2.2.4 Embedded system's applications

Embedded systems are commonly found in consumer, industrial, automotive, home appliances, medical, commercial and military applications.

- Consumer applications such as mobile phones, MP3 players, video games consoles, digital cameras, Global Position System (GPS) receivers, printers, telephone and banking.
- Home automation applications in controlling lighting, air conditioning, heating and voice control. It's also used in microwave ovens, washing machines and dishwashers.

- Medical applications such as vital signs monitors, medical electronic stethoscopes and various medical imaging.
- Transport applications such as steering systems and automated monitoring.
- Telecommunications systems, satellite systems and military defense systems.
- Computer networking applications such as displays and monitors, image processing, network cards and systems.
- Industrial applications like controlling the factories production lines and robotics manufacturing.

2.3 Sensor

Sensor can be defined as a device whose purpose is to detect and measure the physical quantities in the environment. Depending on the type of the measured quantity there are several types of sensors such as proximity sensors, thermal sensors, pressure sensors, and so forth. Distance sensors and proximity sensors perform the same task that vision and tactile sensing do in living beings. They are used to measure the relative distance between the sensor and objects in the environment [5]. For that, passive or active sensors can be used. In the case of using passive sensors, the sensor just receives the reflected or emitted electro-magnetic radiation provided by natural energy sources, therefore it only works when the natural energy is available. Some types of infrared sensors and cameras are passive.

Active sensor emits its own energy such as microwave, sound, and light, and receives a distorted version of the reflected signal which provides information about the surrounding. The most common active sensors are Light Amplification by Stimulated Emission of Radiation (laser), Radio Detection and Ranging (radar), infrared (IR) sensor, and ultrasonic. All of them are developed to find the range of obstacle within certain meters hence, the selection process of the appropriate sensor depends on several factors such as, cost, atmospheric condition, kind of obstacle to be detected, detection range, miniature, the desired precision of measurements and safety. To choose the appropriate sensor the following is a brief explanation of the operating principles, some of the advantages and disadvantages of the four sensors.

2.3.1Laser

Laser sensors transmit light wave and receive the reflected light to measure distance. The distance to the obstacle is calculated using the speed of the light and the recorded time between the transmitted and received waves.

Laser has a narrow beam, high frequency data, large range of detection, and provides high resolution. Nevertheless laser might cause eye's damage if it's directly aimed to. The device would be used by blinds and therefore can pose great danger to others surrounding the user, Laser is also extremely expensive.

2.3.2 Radar

Radar sensors transmit electromagnetic energy which is emitted from an antenna and receive the reflected energy to measure distance. The time required for the energy to travel to the obstacle and return back to the sensor determines the distance to the obstacle [5].

Radar has a large range of detection and its beam width depends on the size of the antenna. In addition, Radar technology offers good accuracy in different environmental conditions (e.g. Rain, snow, poor visibility) without any strong limitations. But it does not detect small obstacles which do not have a high reflectance, beside its high cost.

2.3.3 Infrared

In infrared sensors (IR), a pulse of infrared light is emitted and then reflected back by some obstacle or not reflected at all [6]. To determine the distance to the obstacle, infrared sensors use the triangulation's method, when the light is reflected, it is detected with a different angle which depends on the distance to the obstacle, the detected angle gives enough information to calculate the distance.

Infrared sensor is cheap, small, provides fast response, has specific spectrum and has low power consumption, but infrared sensor might give different responses for obstacles at the same distance, because light does not reflect back in the same angle and intensity for different surfaces, colors, and shades. It also gives inaccurate measures as a result of the distortion that occurs in reflected light due to its exposure to direct or indirect sunlight. Another fact about infrared sensor is its ability to penetrate the glass, therefore glass obstacles will not be detected, due to IR limitations, it is inappropriate to meet the requirements of safety and accuracy.

2.3.4 Ultrasonic

Ultrasonic sensors determine the distance to an obstacle by transmitting ultrasound waves which are reflected back to the receiver. Since the speed of the sound is constant in air (340.29 m/s), the time between the transmitted signal and the received signal can be measured, therefore the distance to the obstacle can be determined [6].

The only disadvantage of ultrasonic sensors is that sound wave is slightly affected by some factors such as humidity, dust, temperature, pressure, and interference. But it is not affected by obstacle materials, and colors. It also does not penetrate objects, and its ability to perform under low visibility conditions makes it ideal for all day use, thus it has the ability to detect all types of obstacles, unlike infrared sensors. Ultrasonic sensors are cheaper than laser and radar sensors, have appropriate size, wide beam width and fast response.

2.4 Electrical Motors

Electric motors are devices that convert electrical energy to mechanical energy using interacting magnetic fields.

Almost, all the electromechanical movements used for a wide variety of residential, commercial and industrial operations, linear or rotational are caused either by an Alternating Current (AC) or Direct Current (DC) motor.

2.4.1 Alternating current motors

An AC motor as shown in Figure 2.3, consists essentially of two main parts: stator and rotor, the 3-phase stator windings are set 120 electrical degree apart, each one is connected to one phase of the power supply then a rotating magnetic flux of a constant magnitude is generated according to Faraday's law.

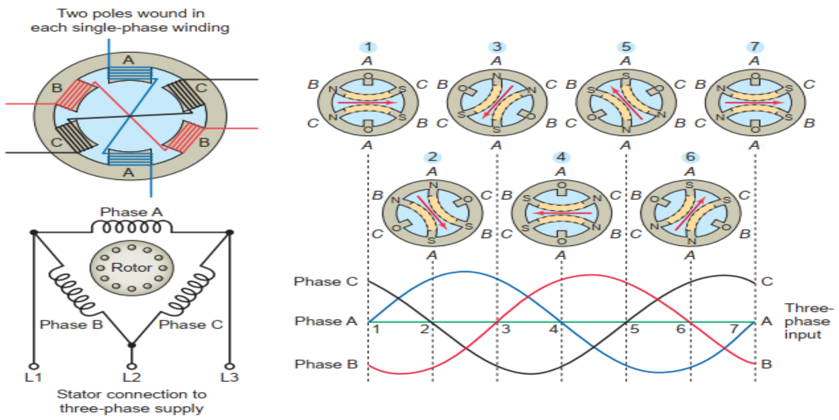

Figure 2.3: Rotating magnetic field

The rotor doesn't receive electrical power by conduction, but induction. when the stator's magnetic flux passes through the air gap to the rotor's winding an Electromotive Force (EMF) is induced in the rotor, in the case of closed circuit induction current is produced in the rotor causing magnetic field which follows the rotation pattern by being attracted and repelled by the stator's field, since the rotor is free to run, it follows the rotating magnetic field in the stator.

AC motors are rarely used in mobile robots, because most of the robots are powered with DC coming from batteries. Since electronic components use DC, it is more convenient to have the same type of power supply for the actuators as well. AC motors are mainly used in industrial environments where very high torque is required, or where the motors are connected to the mains / wall outlet.

2.4.2 Direct current motors

DC motors consist of stationary magnet poles and a rotor of conductors. When a direct current passes through the rotor's conductor a magnetic field is generated according to Faraday's law and when it is placed in the magnetic field of the stator the two fields interacts, similar poles repel and dissimilar poles attract causing a torque and force tending to rotate the armature as shown in Figure 2.4. The force magnitude is proportional directly to the applied current.

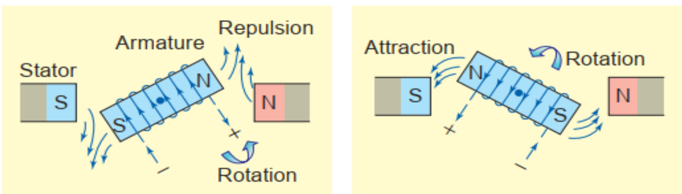

Figure 2.4: Field's interactions

When the dissimilar poles line up the armature stops, in order to continue rotating the commutator reverses the direction of the current in the conductors.

The direction of rotation of this motor is given by Fleming's left hand rule shown in Figure 2.5 which states that if the index finger, middle finger and thumb of a left hand are extended mutually perpendicular to each other and if the index finger represents the direction of magnetic field, middle finger indicates the direction of current, then the thumb represents the direction in which force is experienced by the shaft of the DC motor.

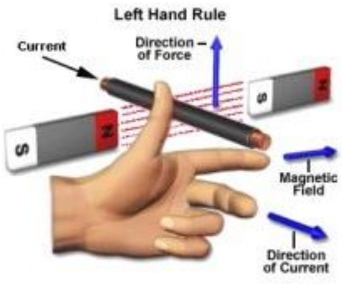

Figure 2.5: Fleming's left hand rule

2.4.3 Stepper motors

As electrical motors, stepper motor has a rotor that rotates through a fixed angular step in response to each input current pulse received by computers, microprocessors, and programmable controllers. Rotation occurs because of the magnetic interaction between rotor's poles and the sequentially energized stator windings' poles. The rotor has no electrical windings, commutator, or brushes but has salient and/or magnetized poles, this feature makes the motor quite, robust and reliable.

The shaft of the stepper motor rotates in a series of discrete angular intervals/steps, one step is taken each time a command pulse is received, every revolution is divided into number of steps, when a definite numbers of pulses are supplied the shaft turns through a definite known angle, the speed of rotation is relative to the frequency of those pulses. When it stopped, a stepper motor inherently holds its position. Therefore it is ideally suitable for open-loop embedded systems where precise positioning / speed or both are required.

2.4.4 Servo motors

Servo motors have the basic operation principle same as the other electromagnetic motors. However, their construction, design and mode of operation are different.

As shown in Figure 2.6, all servo motors operate in closed-loop systems. Servomotor's wires are not connected to the conductors but to the controller which sends position or speed command signals to the amplifier which drive the servo motor. An encoder for position or tachometer for speed are incorporated within the servo to provide feedback information to the controller which compares it with the programmed motion and modify the servo position or speed.

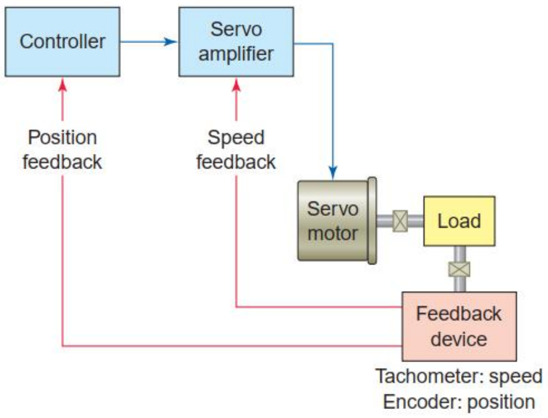

Figure 2.6: Servo motor's closed-loop system

Servo motors have high torque capabilities, they can operate at low speeds including zero speed beside their ability to accelerate and decelerate very quickly, they can also hold still at static positions. There are three basic types of servo motors which are used in modern servo systems: AC servo motors, based on induction motor designs DC servo motors, based on DC motor designs; and DC or AC brush-less servo motors.

Table 2.2: Motor's applications, advantages and limitations

Motor Type	Applications	Advantages	Limitations
DC motor	• Toys • RC Servos • Gear motors	• Cheap • Lightweight • Reasonably efficient • Good low-speed torque	In addition to the audible whine from the commutator brushes, these motors create a lot of electrical noise which can find its way back into other circuitry and cause problems.
Servo Motors	• Robotics • Animations • Radio Control Cars/Boats/Planes	• Low cost - (RC Servos) Smaller sized servos can be purchased for just a few dollars. • Variety - There is a wide range of sizes and torque ratings. • Simple to control.	Most RC servos are limited to 180 degrees of motion and positioning accuracy and repeatability of +/- 1 degree is typical.
Stepper Motors	• 3D Printers • CNC Machines • Camera rigs Robotics • Precision Gear motors	• Precise repeatable positioning • Precise speed control • Excellent low-speed torque • Excellent 'holding torque' to maintain position	• Low efficiency • May need encoder or limit switch to establish a reference position • Subject to missed steps if overloaded

2.5 Relay

Relays are automatic protective and switching devices in most of the control processes or equipment. Regardless of their type, whether electronic or electromechanical, they are one of the most wildly used devices, therefor their classification or types varies according to the function they are used for. Depending on their operating principle and their structure features there are electromagnetic relays, electronic relays, time relays and thermal relays, with a varied power supply AC or DC, rating voltage or current and sizes. Electromagnetic relays shown in Figure 2.7 consist of an electromagnet coil and armature a spring and a series of electrical contacts. These contacts are usually Common (COM), Normally Open (NO) so the circuit attached to gets open when the coil is de-energized, and Normally Closed (NC) which circuit is closed by default. When the coil is energized the moveable part (COM) changes its primary position causing the NO circuit to be closed and the NC circuit gets opened, when the coil is de-energized the spring turns back the (COM) to its original place. This feature enables the relay to control at least two separate circuits by one signal.

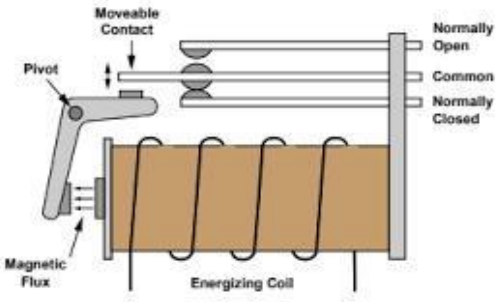

Figure 2.7: Relay

2.6 Power Supply

Energy management plays a major role in almost every electronic system as it controls, regulates, and distributes energy throughout the system. This affects the reliability, performance, and electronic equipment's cost.

The power unit can be defined as an electrical device used to supply electrical power to electrical loads. Power units receive their energy from various sources such as electrical power transmission systems, electromechanical systems such as generators, solar energy inverters, as well as energy storage devices like batteries, fuel cells and others. Sometimes these devices convert the electrical current from a source to the accurate voltage, frequency and current needed by the supplied load. Some types of power supplies are separated of load, whereas others are fabricated into the appliances they control.

The embedded systems are fed with direct current, either by batteries or by converting alternating current into direct current, this is done in the steps shown in Figure 2.8:

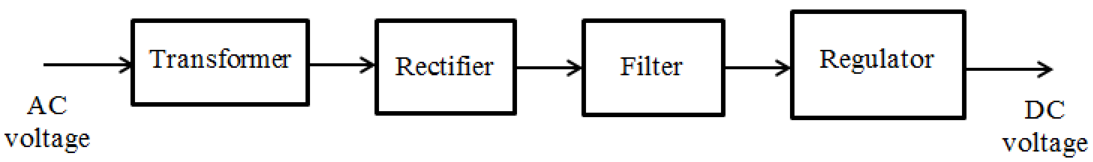

Figure 2.8: Power supply block diagram

- The job of the transformer is to step down the AC voltage to suit the requirements of the solid-state electronic devices and circuits.
- Rectifier is a circuit which employs one or more diodes to convert AC voltage into pulsating DC voltage.

- The function of the filters is to remove the ripples present in the output voltage supplied by the rectifier.
- The main function of the regulator is to keep the terminal voltage of the DC supply constant even when the AC input voltage to the transformer varies or the load varies.

2.7 Literature Review

In the past, the visually impaired faced difficulties in navigation and movement. Some of them used to have a guide dog to help walking around and avoid collisions, others used to ask for someone else's help. After postwar period a lot of techniques were developed in order to make the visually impaired feel more independent. In this section some of these techniques are discussed:

In 1997 Johann Borenstein and Iwan Ulrich [7], designed a cane supported by wheels. This cane consisted of ten ultrasonic sensors, servomotor, a mini joystick for controlling and a built-in computer. When an object is detected by ultrasonic sensors the servomotor can steer the wheels right or left depending on the microcontroller’s command. The guide cane managed to guide the user physically around obstacles and toward a safe path.

In 2004 m. Bouzit Et Al [8], proposed a device which is portable, effective, and allowed visually impaired individuals to navigate through familiar and non-familiar environments without relying on the assistance of a guide. The "tactile handle" consisted of an array of vibro-tactile actuators positioned to match the finger phalanxes, proximity sensors, and an embedded micro-controller. The handle processes sensor signals and outputs information to the user through variable and synchronized vibrations, which will enhance the sense of orientation and distance for the user. The prototype has an ergonomic design, lightweight, compact, and adjustable to different hand sizes. Users were able to correctly perceive the intensity and distinguish between each of the tactile excitations from the total of sixteen actuators incorporated inside the handle. The results showed that the perception accuracy increased while the perception delay decreased as the user became more familiar with the device outputs.

In 2004 Sylvain Cardin et al. [9], developed an obstacle detection system for visually impaired people using multi-sonar system and sending appropriate vibro-tactile feedback to the user. The equipped user was able to walk through the corridor in a reasonable time after a bit of training. The collisions are perfectly avoided and the user was able to distinguish an obstacle from its left and right and to localize himself in his environment. It provided reliable feedback about the surrounding environment by vibro-tactile mapping. The combined motion of the user and a moving obstacle provided an accurate perception of its trajectory. One aspect which reduced the performance of the system was the difference of the ultrasonic reflectance of the different materials that constitute our environment. This difference alters the reliability of the sonar sensor and can lead to some anomaly in the vibro-tactile feedback.

In 2005 s. Meers and koren ward [10], presented a novel approach for enabling the blind to achieve obstacle avoidance and navigation in outdoor environments with the aid of visual sensors, GPS and electro-tactile stimulation. The Electro-Neural Vision System (ENVS) works by extracting a depth map from stereo cameras by measuring the disparity between the stereo images. This range data is delivered to the fingers via electro-neural stimulation to indicate the user with the range of objects viewed by the cameras. To perceive the location of obstacles and the 3d structure of the environment the user imagines that the hands are held in the direction viewed by the cameras, with fingers extended, and the amount of stimulation felt by each finger indicates the range of objects in the direction pointed at by each finger. The relative location of significant landmarks is also determined using GPS and stored GPS coordinates and delivered to the fingers via encoded pulses, when the landmarks are in the field of view of the stereo cameras. Perceiving the 3D structure of the environment and the location of landmarks in real time effectively enables the user to navigate the environment without using eyes or other blind aids. Experimental results are provided demonstrating the potential that this form of 3D environment perception has enabled the user to achieve localization, obstacle avoidance and navigation without using the eyes. The main problem with the existing substitute vision systems for the blind is that the delivered information to the user is in a form that is either hard for him to understand or difficult for the brain to derive a 3D model of the environment from. ENVS is able to provide a blindfolded user with the ability to avoid obstacles, navigate the environment and locate his or her position in the environment via familiar landmarks by interpreting sensory data via the electro-tactile data gloves

In 2005 Kiyohide Ito. Al [11], developed an electronic aid device which called ‘Cyarm’ for objects detection, space, and tracking object movement using ultrasonic sensor. The preliminary experiments conducted on the device showed its usability to provide safe and accurate movement of the blind.

In 2006 Lise a et al [12], described a compact, wearable device that converted visual information into a tactile signal. The device, constructed entirely from commercially available parts, which enabled the user to perceive distant objects via a different sensory modality.

Preliminary data suggested that the device is useful for object avoidance in simple environments. In the preliminary implementation of the device the visual data is acquired by inexpensive webcams and the tactile signal is provided by an array of vibrating motors.

In 2007 D.Yuan and R. Manduchi. [13], Developed a prototype composed of a laser-based range sensor and of an onboard processor. As the user swung the hand-held system around, he/she received local range information by means of a tactile interface. In addition, the time profile of the range was analyzed by the onboard processor to detect environment features that are critical for mobility, such as curbs, steps and drop-offs. The range was collected by a short-baseline triangulation system formed by a point laser and a miniaturized camera, producing readings at frame rate. An Extended Kalman filter was used to trank the range data and detect environmental features of interest. The prototype sensor produced reliable local range measurements using active triangulation.

In 2007 Mounir Bousbia -Salah at. Al [14], proposed an obstacle detection system using ultrasounds and vibrators, detects the nearest obstacle via stereoscopic sonar and send back vibro-tactile feedback to inform the blind its localization through speech synthesizer. Despite the ability of the proposed system to detect the nearby obstacle, it was not able to solve the blind's problems ultimately, due to the limitation of the characteristics of ultrasound reflections in detecting very small obstacles.

In 2012 Ziad o. Abu-faraj et al [15], designed a prototype shoes and spectacles, each shoe was mounted with three pairs of ultrasonic transducers placed on medial, central, and lateral aspects of the toe cap to detect ground-level obstacles of different heights as well as ground pits and holes. The corresponding tactile outputs are provided by three miniature-sized vibrating motors embedded within the collar of the shoe. The spectacles are instrumented with a pair of ultrasonic transducers mounted centrally above the bridge, and with a buzzer at one of the temples. They are used to detect obstacles at head level. The developed shoes and spectacles are controlled via a battery-operated, microcontroller-based belt pack unit. The standards of obstacle detection, ergonomic design, and ease of use have been fully implemented, but there was some limitation in how obstacles are warned, as well as its operational flexibility.

In 2013 Sonda Ammar Bouhamed et al [16], proposed a system with ultrasonic sensors and one monocular camera to make the user aware of the presence and nature of potential encountered obstacles and stair-case detection. The recognition result is estimated to 82.7% for detecting stair presence and 89.8% for detecting their type either ascending or descending.

In 2014 Rohit Sheth et al [17], designed smart white cane used ultrasonic sensors to detect pits, potholes, downfalls, a staircase (up and down), low lying and knee level obstacles and even those above the waist. The user was notified about the presence of obstacles by the pre-recorded sound messages and a haptic feedback in form of vibrations. The smart white cane was a feasible and a convenient product.

In 2015 Ayat Nada et al [18], designed a system with infrared sensor to detect stair-case and ultrasonic sensors to detect any other obstacles in front of the user within a range of four meters. Moreover, another sensor was placed at the bottom of the stick for the sake of avoiding puddles. Speech warning message and vibration motor were activated when any obstacle was detected. Good results were given in detecting obstacle, stair, and water pits within four meters.

In 2015 Ankit Agarwal et al [19], presented a stick with ultrasonic sensor to scan three different directions, a camera as alternative tool in the places that surrounds with a low signal coverage and GPS module was used to provide information according to the current location. A buzzer and a vibrating motor were activated when any obstacle was detected. Short Message Service (SMS) was sent to the numbers saved in the microcontroller in case of emergency.

In 2018 S.D.Asha Mahesh et al [20], designed a device that consisted of smart shoes and smart cane that alerts visually-impaired people over obstacles coming in their ways through voice alert and per-recorded messages. The shoes is integrated with self-power generation unit. The designed device was able to detect obstacles and alert the blind person to them.

In 2018 Vanita Bobanne et al [21], proposed framework that used ultrasonic sensors and ARDUINO to detect any obstacle, water, or dark areas and alarm the blind through beeping sounds or vibrations. It promoted great outcomes in recognizing the obstacles within a range of three meters.

In 2019 Pruthvi.s et al [22], designed a device consisted of raspberry pi, hi-res camera which captured images and processed them with an object detection algorithm (You Only Look Once (YOLO)) and alert the user if he came across any obstacle using text to speech unit. The designed device gave a clear result that the users could greatly benefit in terms of knowing what was around them. The device, constructed entirely from commercially available parts, which enabled the user to perceive distant objects via a different sensory modality. Preliminary data suggested that the device is useful for object avoidance in simple environments. In the preliminary implementation of the device the visual data is acquired by inexpensive webcams and the tactile signal is provided by an array of vibrating motors.

CHAPTER THREE

SYSTEM HARDWARE AND SOFTWARE CONSIDERATIONS

3.1 System Description

Figure 3.1 shows the structure of the system. All hardware and software components are illustrated in the block diagram.

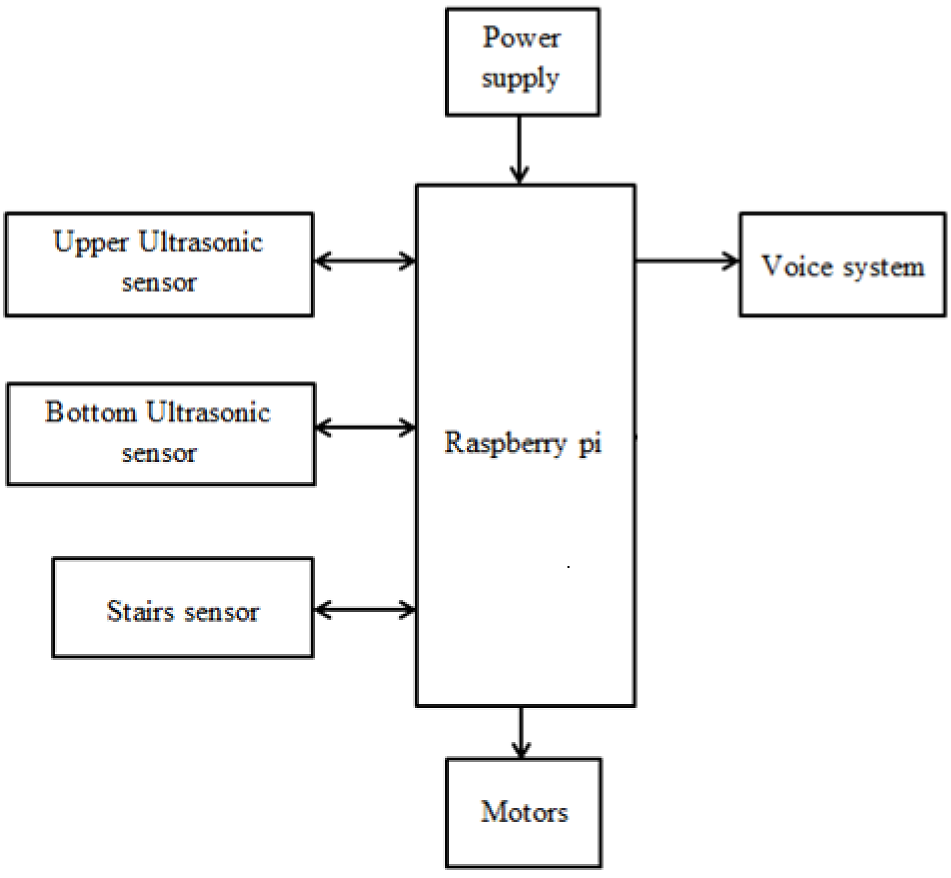

Figure 3.1: Block diagram

3.2 System Hardware

The system consists of many hardware components which are controlled by raspberry pi in order to perform obstacles detecting and avoiding:

3.2.1 Raspberry pi

A raspberry Pi computer is a System on Chip (SOC), designed for education, where a single board carries all the essential circuits. Like any other computer, its operating system acts as backbone for operation. Raspberry Pi facilitates open source operating systems based on Linux. Its processing unit contains a central processing unit which is responsible for carrying out the instructions of the computer through logical and mathematical operations, the Graphics Processing Unit "GPU" which is responsible for processing graphics, images and video, and RAM.

Raspberry Pi requires additional hardware like Keyboard, Mouse, Power Supply, display monitor and Secure Digital "SD" Card with OS Installed (Acting like Hard Disk) for operation. Raspberry Pi also facilitates USB ports, High Definition Multimedia Interface (HDMI) port and Ethernet for Internet connection. In recent versions Bluetooth and Wi-Fi are included as shown in Figure 3.2.

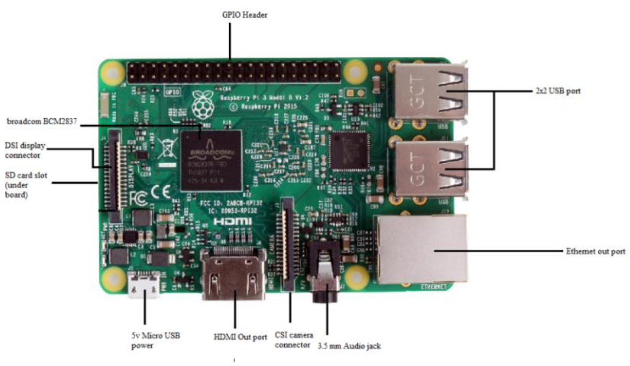

Figure 3.2: Raspberry pi 3 model B

Raspberry pi is used for embedded systems projects for its small size, low cost, high performance and the availability of General Purpose Input /Output (GPIO) pins which provide ability to connect electronic devices directly as input or output based on programming commands, or for communication protocols between different components. All GPIO pins are digital but it is possible to add external module for analog connections. Since the pins are not shown in raspberry pi, it can be illustrated in Figure 3.3.

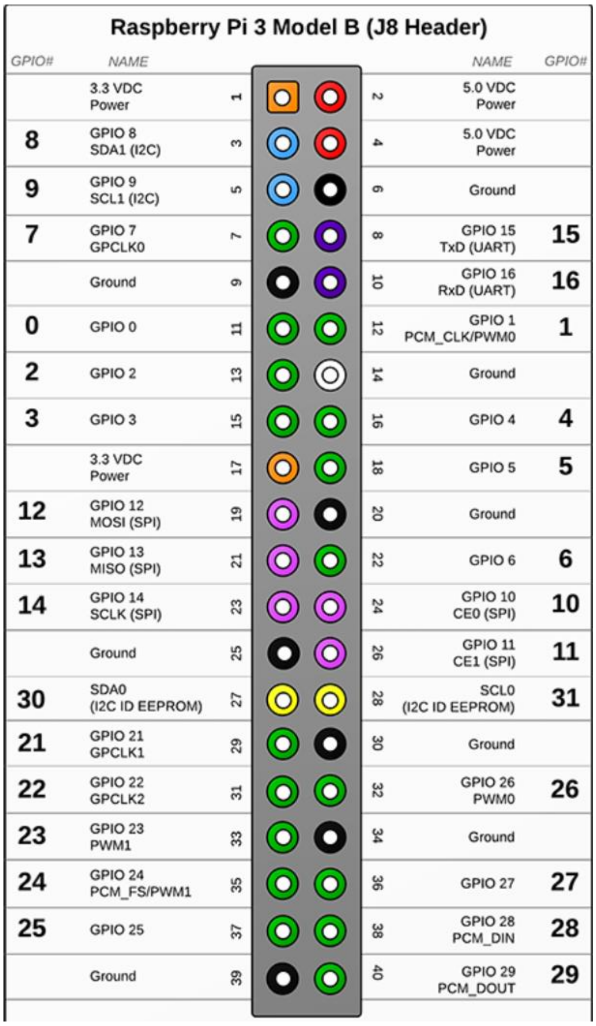

Figure 3.3: Raspberry pi GPIO pin map

Raspberry pi 3 model B contains 40 pins, these pins are numbered according to two methods, either BOARD or Body Control Module

(BCM). It is important to specify the numbering method in the code. Some pins are used as power outlet such as:

- Pins (1, 17): provide out power with 3.3V and 50 mA.
- Pins (2, 4): provide in or out power with 5V, it is connected to MicroUSB and can provide current according to a charger connected to MicroUSB.
- Pins (6, 9, 14, 20, 25, 30, 34, 39): Ground.

Some pins are used for communication, as raspberry pi supports 4 protocols UART, 1Wire, i2C, and SPI.

3.2.2 Ultrasonic

Ultrasonic sensor is one of the most available sensors commercially, it is commonly used in the fields of medicine, science, engineering, industry, and such other fields.

Sound waves occurs when mechanical energy is transmitted through a medium causing vibrations in the medium around the sensor. This affects in the density and elasticity of the medium and results in a change in the position and kinetic energy in the medium, thereby the sound is propagated. Sound waves can propagate through any kind of media including solids, liquids and gases except vacuum, with a constant speed when pressure and temperature are constants.

Ultrasonic sensor uses ultrasound wave which is a sound generated above the human hearing range (typically 20 KHz) [23]. Ultrasound has outstanding capability to probe and reflect of surfaces and materials without altering or destroying them.

- **Piezoelectric Disk**

The piezoelectric ceramic disk is used in ultrasonic transducer to generate ultrasound and record the reflected waves, it has an element with two electrodes, one electrode is fixed but the other is movable as shown in Figure 3.4. When high frequency voltage is applied to the two electrodes, a physical deformation of the disk surface occurs causing vibration. The vibration generates high frequency sound waves, when these waves strike an obstacle, echoes cause a mechanical displacement in the receiver, which leads to change the electrode's voltage.

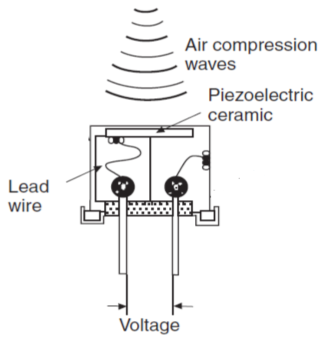

Figure 3.4: Piezoelectric disk

- **Basic parameters of waves**

Ultrasonic vibrations travel in form of waves which have basic parameters as wavelength (λ), period (T) of complete cycle as shown in Figure 3.5.

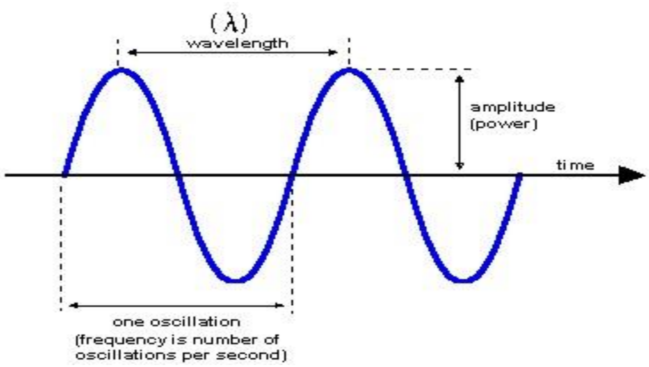

Figure 3.5: Basic parameters of wave

The period (T) is the time required to complete a full cycle, it is measured in second. From Equation 3.1, it is inversely proportional to frequency (f) which is the number of completed cycles in one second, it is measured in Hertz (Hz).

$$f = 1/T \qquad (3.1)$$

The wavelength (ʎ) is the distance between two consecutive peaks or two successive bottoms. It has relation with period, frequency, and velocity given by Equations 3.2 and 3.3.

$$ʎ = C/f \qquad (3.2)$$

$$ʎ = C.T \qquad (3.3)$$

From Equation 3.2 wavelength is proportional inversely with frequency, therefore, the ultrasound has short wavelength, which means it can be reflected off very small surfaces.

- **Wave Propagation**

In respect of ultrasound waves propagation there are two common types of waves as shown in Figure 3.6:

I. Longitudinal wave in which the movement direction is same to propagation direction.
II. Shear wave in which the movement direction is perpendicular to propagation direction.

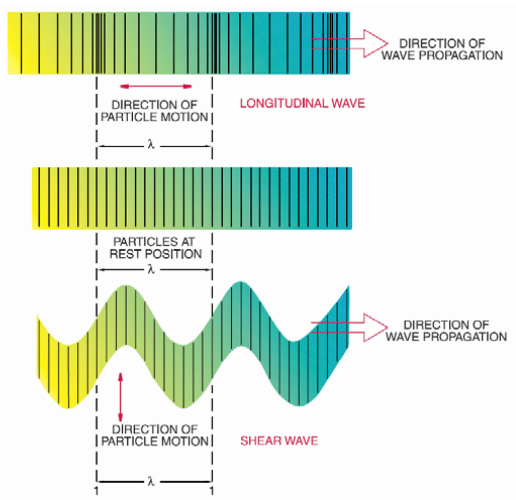

Figure 3.6: Longitudinal wave and shear wave

- **Applying ultrasonic**

As shown in Figure 3.7, ultrasonic sensor can detect obstacles by sending sound waves and receiving reflected echoes depending on two factors:

I. Time of flight: amount of delay between the emission of sound and receiving an echo depending on the distance of an obstacle.

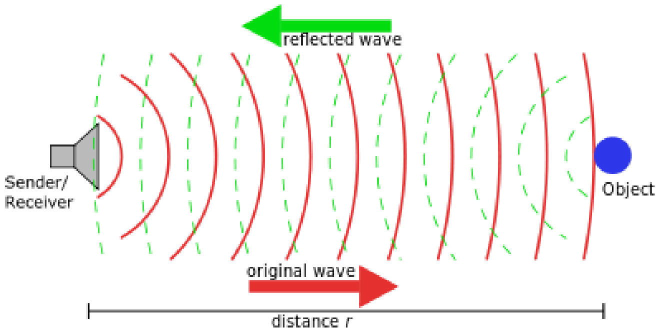

Figure 3.7: Object detecting illustration

II. Beam size: amount of reflected waves, depends on dimensions of an obstacle compared to beam size, all sound will be reflected to the receiver if an obstacle is larger than the beam size, but only a part of waves will be reflected from obstacles with smaller dimensions than the beam size, and the rest will be losing as shown in Figure 3.8

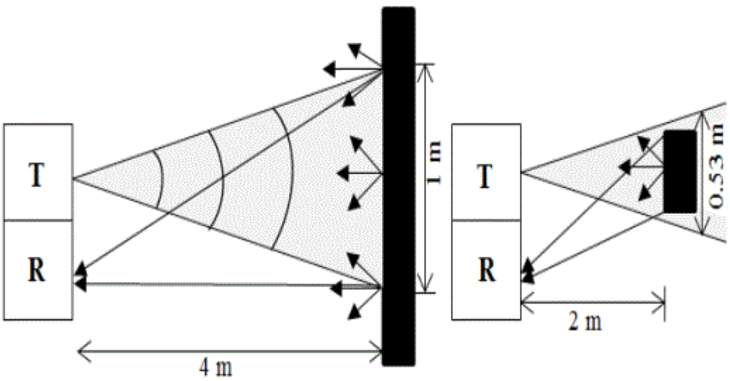

Figure 3.8: Reflections from different sizes of obstacle

- **Distance Measuring**

The radiation pattern of ultrasonic sensor has dimensional cone shape as shown in Figure 3.9, it consists of a main cone and several neighboring cones. The approximate angle of the main cone depends on sensor's manufacturing.

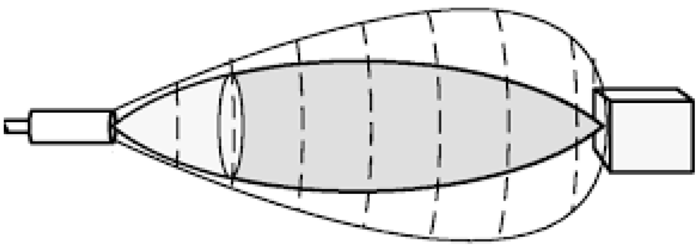

Figure 3.9: Conical shape of ultrasonic waves

The distance between the obstacle and the sensor is directly proportional to the time between transmitting the signal and receiving the echo to detect the obstacle within the sensing range. The distance is given by Equation 3.4.

$$L = C.T/2 \tag{3.4}$$

As shown in Figure 3.10, the detection does not start at the sensor's surface, but there is a distance in front of the sensor called blind zone. Obstacles which exist in the blind zone are not detectable. Depending on the sensor, the blind zone is from 6 to 80 cm [1]. Because it's already being detected in case of motion, the blind zone does not affect the detection.

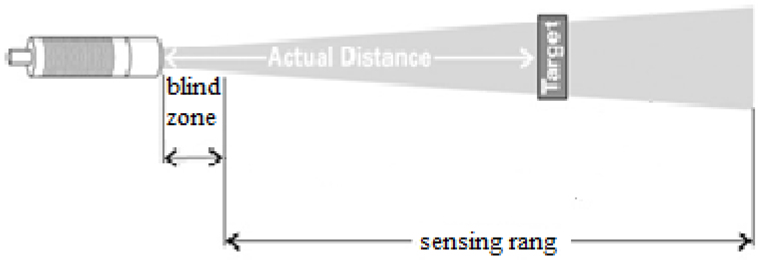

Figure 3.10: Blind zone

- **Application of Ultrasonic**

Sonic technique is used in multiple fields, medical imaging, food observation, Oil-water de-emulsification and industrial field as shown in Table 3.1.

Table 3.1: Ultrasonic techniques applications

Technique	**Parameters**	**Application**
Ultrasonic reflection technique	Phase Frequency Amplitude	• Distance, level, position, speed of sound. • Object structure and presence of objects. • Density, viscosity, concentration. • Motion, velocity (Doppler effect)
Ultrasonic transmission techniques	Phase Amplitude	• Concentrations in multi-component systems. • Particle size distribution in suspensions, emulsions. • Volume and mass flow, velocity. • Density, viscosity. • Temperature.
Ultrasonic emission techniques	Amplitude Frequency	• Process condition monitoring.
Ultrasonic resonance techniques	Frequency Amplitude	• Mass. • Viscosity, viscoelasticity, density. • Specific chemical or biological species. • Multi-component analysis.

Table 3.2: HC-SR04 technical specifications

Voltage	5v (DC)
Current	Less than 2mA
Frequency	40 kHz
Detection distance	2cm-450cm
Sensor angle	Not more than 15 degrees
High precision	Up to 0.3cm
Input trigger	10us TTL impulse
Echo pulse	Output TTL PWL signal
Size	45×20×16mm

3.2.3 Gear motor

Is an electrical AC or DC motor with an integrated series of gear or gear box attached to, this gear box consists of a set of various sizes of gears. The driver gear is connected to the input shaft, the driven rotates the output shaft .The principle of driving the large radius gear via small radius gear in order to reduce the rotor speed is known as Reduction Gear. The comparison of angular input and output gear is known as Gear Ratio, it is calculated with Equation 3.5.

$$\text{Gear ratio} = \frac{\text{speed of drive}}{\text{speed of driven}} = \frac{\text{teeth of driven}}{\text{teeth of drive}} \quad (3.5)$$

The geared DC motors work over a various range of voltages, the applied voltage is directly proportional to the shaft's speed, also the speed is inversely proportional to the torque according to Equation 3.6.

$$\text{Torque} = \frac{\text{p} \times 60}{2\pi \times \text{N}} \quad (3.6)$$

Reduction gearing provides high torque that allows small electrical motor to move significantly large weight at reasonable speed, hence it is suitable for many applications such as: navigating robotics, medical tables and garage door openers.

3.2.4 Relay

Relay Module is an electrically operated switch that helps microcontroller to control the motor. It acts as a mediator between a microcontroller, a power supply or batteries device, and motors. Although the microcontroller decides the speed and direction of motors, it cannot drive them directly because of its very limited power (current and voltage) output. The relay module, on the other hand, can provide the current at the required voltage but cannot decide how motors should run. Thus, the microcontroller and the relay module have to work together in order to make motors move appropriately.

An electronic circuit of transistors to amplify the signal coming from the controller and diodes for complete isolation between the controller and the load circuit are added to electromagnetic relay to present the internal componants of the relay module. It consists of multiple channels supplied with (5v,12v,24v..etc).

A relay module of two channels as shown in Figure 3.11, has two sets of pins:

- Output Terminal Blocks: each channel have three terminals labeled for thier function: COM, NO, NC.
- Control Pins:

  VCC: supply the relay module.

  IN1,IN2: control the relay module.

  GND: common ground connection.

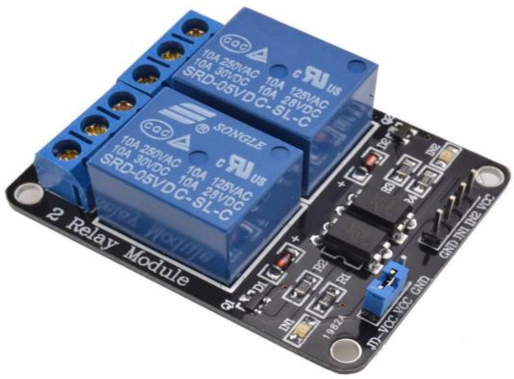

Figure 3.11: Relay module

3.2.5 Power unit

The system is powered with two power sources:

Power bank with 5V,1000mA output and capacity of 6000mAh.

Power source with 3V.

3.2.6 Cane Material

Polypropylene (PPR) which belongs to polyolefin family, was first produced in 1954 by G.Natta following the work of K.Zieglar by polymerization of the propylene monomer, in 1957 the first commercial product was introduced by Montecatini [24].

PPR can be used as a plastic or fiber in several applications, it offers better advantages than most competitive material on the basis of specific modulus such as low cost, flexibility, lightness, process ability, it provide higher stiffness at lower density. All of this encouraged the large usage of PPR in industrial products.

PPR is synthetic, rigid, and crystalline thermoplastic material made from polymerization of propylene monomer, it contains 10000 to 20000 monomer units [24].

- **Polypropylene types**

Polypropylene can be divided mainly into two parts:

I. Polypropylene Homopolymer: it contains only propylene monomer in a semi crystalline solid form.

II. Polypropylene Copolymer: it is produced by polymerizing Propylene and Ethan, it contains two types: Random and Block.

- **Polypropylene manufacturing**

There are two common methods of PPR manufacturing:

I. Zieglar –Natta polymerization.

II. Metallocene Catalysis polymerization

The Figure 3.12 shows the chemical formula of PPR.

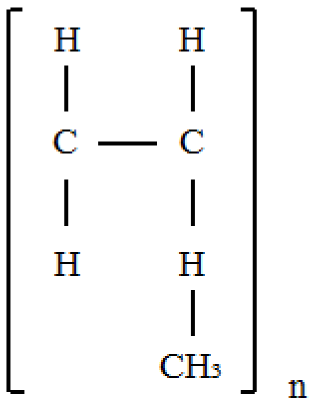

Figure 3.12: Chemical formula

Polypropylene has three basic forms depending on the position of the methyl groups which are attached to every second carbon atom, these forms are known as:

- Isotactic form: methyl groups arranged on one side of carbon chain as shown in Figure 3.13.

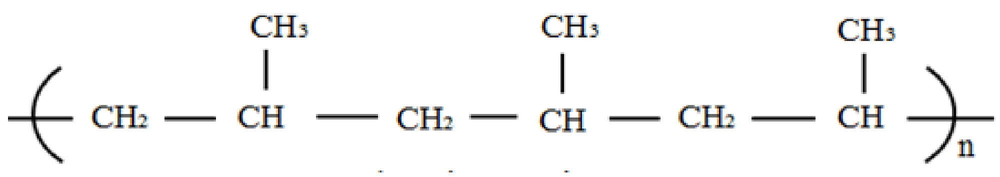

Figure 3.13: Isotactic form

- Syndiotactic form: alternating methyl group's arrangement as shown in Figure 3.14.

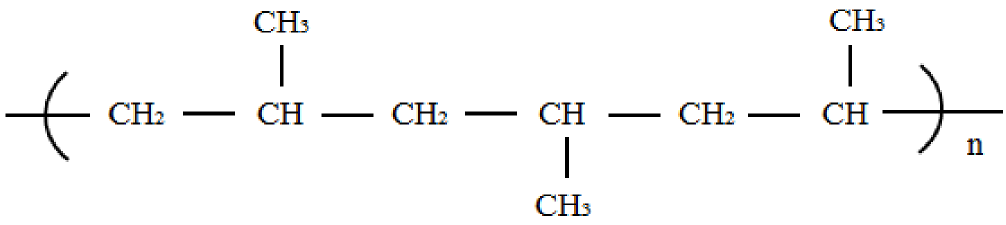

Figure 3.14: Syndiotactic form

- Atactic form: irregular methyl groups arrangement as shown in Figure 3.15.

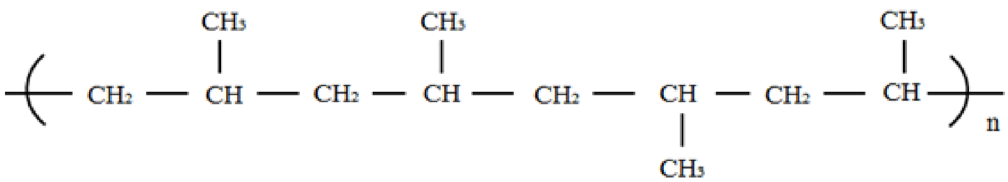

Figure 3.15: Atactic form

- **Polypropylene properties**

I. Good insulation material.
II. Formable.
III. Easy to dismount and assemble.
IV. Environmental stress resistance.
V. **Chemical Resistance**
VI. Good fatigue resistance and good hardness.

- **Polypropylene applications**

I. Automotive industry.
II. Consumer goods.
III. Furniture industry.
IV. Household goods.
V. Industrial applications.

3.3 System Software

The project's software consists of the operating system, simulation and programming code.

3.3.1 Operating System

- **Linux**

Linux which is an alternative to the closed- source Minix created in 1991 by Linus Torvalds [25]. The term "Linux" refers to the Linux kernel, which is responsible of the communication between hardware and a set of programs, tools, and services, to provide the fully functional of the operating system. Similar to other operating systems, Linux has graphical interface and software. The code which is used to create Linux is free for public, any programmer can develop, edit and derive proprietary versions by a process called Linux distributions derivation.

Linux is a safe and reliable system, most of the embedded device technologies depends on Linux, such as routers, switches, servers and smart phones. Raspbian the official operating system of raspberry pi is based on Linux Debian.

- **Raspbian**

Raspbian is the operating system created by Mike Thompson and Peter Green, optimized for raspberry pi hardware, it provides over 35,000 packages pre-compiled software bundled for easy installation on raspberry pi [26], Raspbian packages were optimized for the best performance in raspberry pi, and they are still under development to improve stability. Raspbian is picked from several versions including Raspbian Stretch with desktop and Raspbian Stretch lite, a minimal Debian Stretch-based Raspbian image. Raspberry pi foundation maintains its own recommended version of Raspbian installed using foundation's New Out Of Box System (NOOBS) installer.

- **Python**

All programming languages supported by Linux can be used with raspberry pi, Python is one of these languages, created by Guido van Rossum in 1991[27], it is an interpreted, high level, object-oriented programming language and suitable for rapid application development. Python syntax is simple and easy to learn, which reduces the cost of maintaining programs, there is no compilation step, the edit-test-debug cycle is incredibly fast, it supports modules and packages, which encourages program modularity and code reuse. But Python is slow; since it is interpreted, it takes time to create a module while it is running the program. As well it is sensitive to spaces, adding a space at the start of a line will often break the code. Python has two versions:

I. Python 3.x: which is the current version, it is under development.
II. Python 2.x: has versions of 2.0 to 2.7.

These two versions of python are not compatible with each other, so the written code for Python 2.x may not run with Python 3.x (and vice versa).

- **Python 3**

Python 3 was designed to overcome the fundamental design flaws of the previous versions and therefore could not be fully compatible with the 2.x series. Thus, it was necessary to update a new major version number. Python 3 focused on removing redundant constructs and modules to update a straightforward way to do a specific task while maintaining the feature of multi programming paradigms.

3.3.2 Algorithm

Step 1: Start.

Step 2: Activate GPIO pins, play audio 1 “welcome”.

Step 3: Measure the distance of the three Ultrasonic sensors (bottom, upper, stairs and holes) three times and determine the average distance.

Step 4: If the determined value of the third ultrasonic (stairs and holes sensor) is more than 177.55cm play audio 2 “down stair” and move to step 12, else if the determined value of the third ultrasonic is in the range between 157.55-171.55 cm play audio 3 “up stair” and move to step 12, else move to step 5.

Step 5: If the determined values of the upper ultrasonic and the bottom ultrasonic are more than the desired distance 120 cm move to step 6, else move to step 7.

Step 6: Save as Elementary path, Give motor’s pins 11 and 37 high, and go forward for 5 seconds, move to step 4.

Step 7: Give motor’s pins low for one second, turn left 90 degree to position “W”.

Step 8: Determine and save ultrasonic average distances.

Step 9: Turn right 30 degree and move to step 8 then repeat this step till position “E”.

Step 10: Choose the best option (the saved value must be more than 240 cm, give priority to NW1 , then NE1 ,then NW2 ,then NE2 ,then N, then W) and move to step 11. If there is no option, play audio 4 “there is no option to take, better wait right here and ask for help” and move to step 12.

Step 11: Rotate to the best option’s direction and move forward for 5 seconds, then rotate back to the elementary path, then go to step 3.

Step 12: Stop.

The Figure 3.16 shows the imaginary directions.

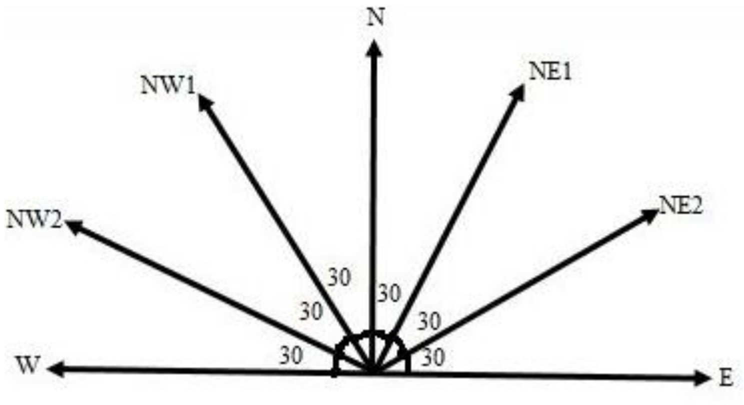

Figure 3.16: Imaginary directions

- **Flow Chart**

The Figure 3.17 shows system's flow chart.

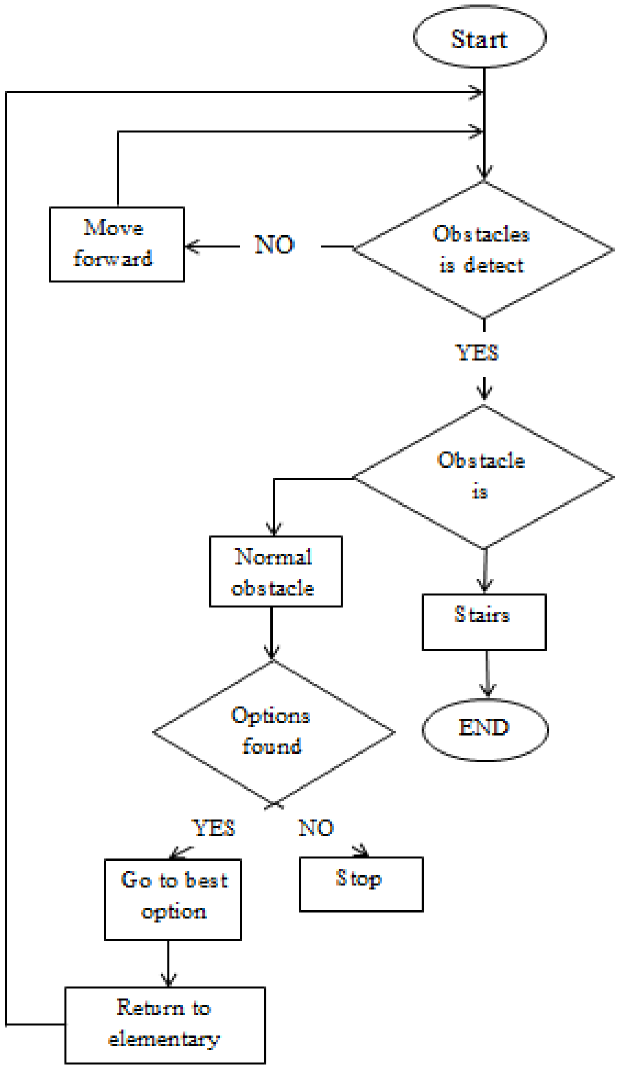

Figure 3.17: Flow chart

3.3.3 Code

Python programming language has been used to implement the project algorithm. We developed a python script that runs on the raspberry pi to get data from ultrasonic sensors, depending on the data received it controls the operation of the motors. The code utilizes the distance measures to avoid obstacles that take place in the pathway of the visually impaired person and redirect him to eventually be on his original destination.

Python modules used in main script are

- Sensor predefinitions

A module that consists of two functions: “distance ()” which expects two arguments, the trigger pin and the echo pin of a particular ultrasonic sensor. The return of this function is the measured distance. The second function called “detected ()”, it returns True if there is an obstacle presence in the desired distance, otherwise it returns False.

- motor predefinitions

A module that drives the motors, and controls the direction of the movement. It includes a function called “ROTATE ()” which expect one argument called direction, according to it, motors are directed to certain location.

The second function called “GO ()” which expect two arguments, direction and Period. It runs motors in the specific direction for the specific period of time.

- sound

A module that contains a function called “SAY ()” which expect file name of stored voice and play it.

3.3.4 System Simulation

Building the system directly exposes the complete system to the damage risk with unnoticed problems, which reveals the importance of simulation in observing system’s behavior. PROTEUS program Simulink is an electronic design automation tool includes schematic capture, simulation and Printed Circuit Board (PCB) layout module. It's widely used in both simulation and design.

PROTEUS 8.9 which supports Raspberry Pi simulation was used, it has some limitations since the Raspberry Pi runs a full copy of Linux and an arbitrary code can be executed. Full simulation model is not possible if unsupported libraries were pulled in the program code. Specifically, the following libraries are supported: Smbus, Pygame, Wiringpi, RPi.GPIO, and Spidev. The Figure 3.18 shows the simulated circuit.

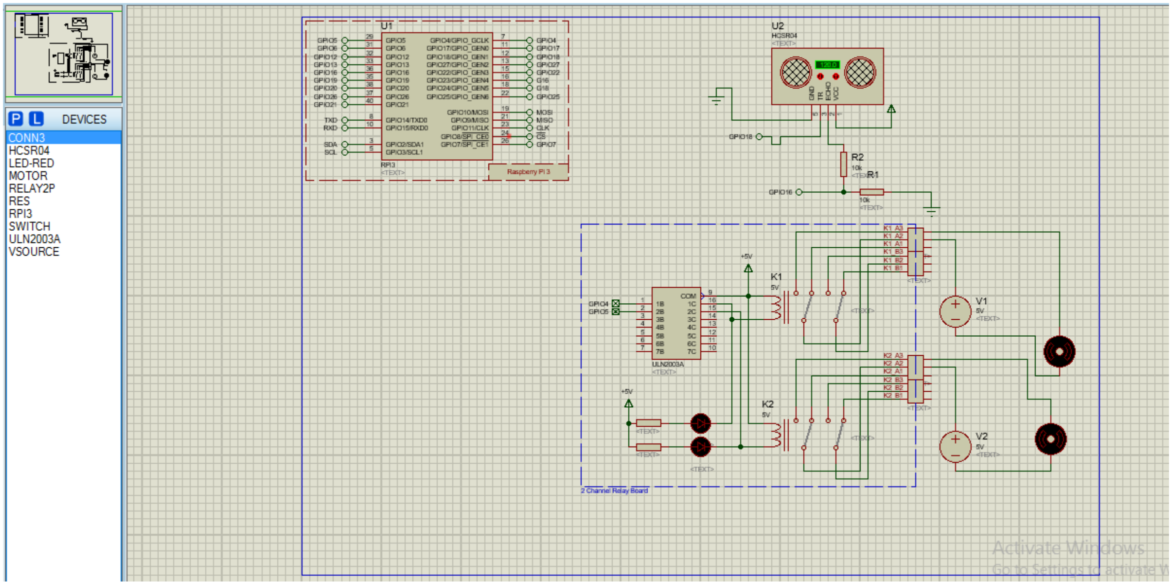

Figure 3.18: System simulation

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND RESULTS

4.1 System Software Implementation

4.1.1 Getting started with raspberry pi

Raspbian image has been downloaded from the official webpage as a zip file, in order to use this image a 16GB-class10 SD card was inserted into a laptop through suitable reader, then it was formatted and the Raspbian image has been successfully burned using the Win32 disk, finally SD card was inserted into the raspberry pi.

Raspberry pi has been powered with 5V and 2A, interfaced via HDMI connection, keyboard and a mouse to adjust a Hostname and a Password, activated the Virtual Network Computing (VNC) and the Secure Shell (SSH), to control the raspberry pi remotely.

4.1.2 Voice alarm

The system has voice indications when different events occur such as initiating the system, stairs and stop. This feature has been implemented by installing python Pygame library.

Voice messages were recorded using a smart phone sound recorder, which records sounds in MPEG-4 (M4A) format, these files were converted to Waveform Audio File Format (WAV) which is supported by Pygame library then stored in raspberry pi. The appropriate voice message was played via a headphone which was plugged into raspberry pi audio port.

4.2 Calculations

4.2.1 Calculations of ultrasonic sensors positions on the stick

The Figure 4.1 shows radiation angle of the ultrasonic sensor. D is a point in the operating range of the sensor in which the controller executes a command when an obstacle is detected, x can be calculated from Equation 4.1.

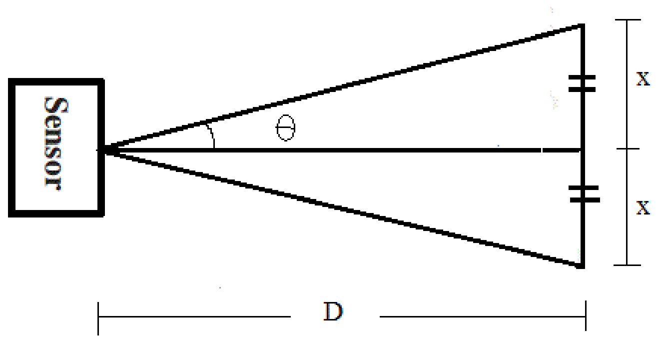

Figure 4.1: Radiation angle of the sensor

$$x = D.\tan\theta \tag{4.1}$$

$$x = 120 \times \tan 15$$

$$x = 32.15 \text{ cm}$$

To detect obstacles on the ground at 120 cm an ultrasonic sensor is placed at 32.15 cm, and to provide protection for the user from the upper obstacles a second ultrasonic is added at 65 cm to avoid sensors overlapping as shown in Figure 4.2.

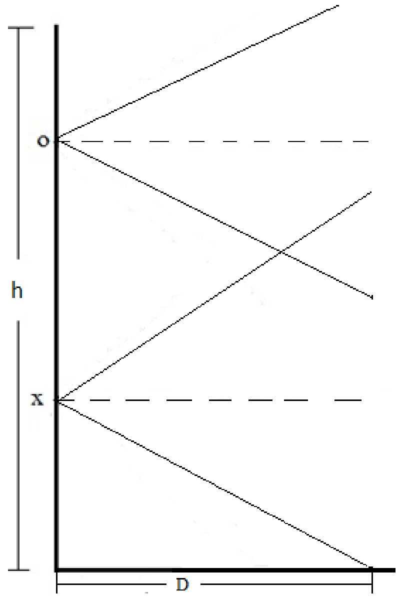

Figure 4.2: Ultrasonic sensors position

4.2.2 Stairs detection

In case of stairs and potholes detection, a third ultrasonic is used, and fixed on the stick at a known angle 57.65 calculated from Equation 4.2. Sets a specific value for distance Y in the program code. As illustrate in Figure 4.3, if the ultrasonic return a value greater than the set value with a certain increase, it gives a command for down stair or pothole, but in case of up stair the ultrasonic return a value less than the set value . Y is calculated by Equations 4.3.

$$\mathrm{F} = \tan^{-1}\frac{\mathrm{D}}{\mathrm{S}} \qquad (4.2)$$

$$\mathrm{F} = \tan^{-1}\frac{150}{95} = 57.65$$

$$\mathrm{Y} = \sqrt{\mathrm{S}^2 + \mathrm{D}^2} \qquad (4.3)$$

$$\mathrm{Y} = \sqrt{95^2 + 150^2}$$

$$\mathrm{Y} = 177.55 \text{ cm}$$

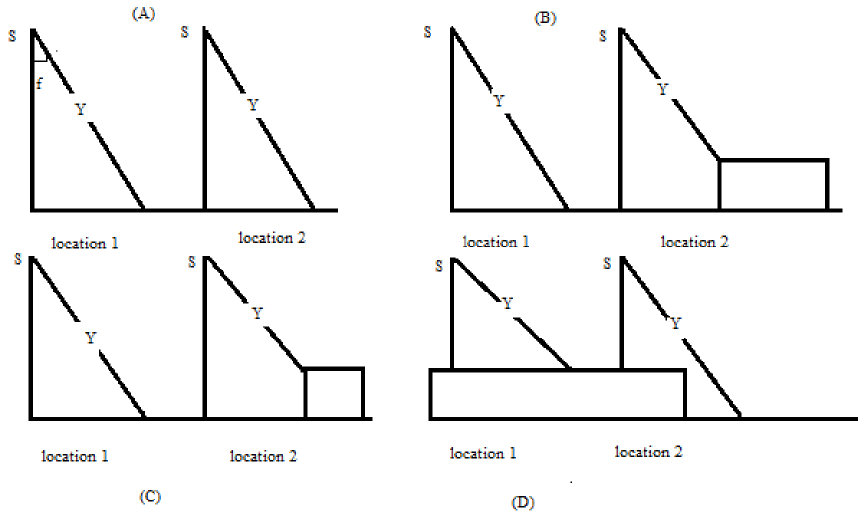

Figure 4.3: Stair detection

Unit rise ranges between (15-17) cm while unit run ranges between (27-30) cm [28], based on these values the decrease range from flat floor measured value has been calculated.

4.2.3 Stick design

- As shown in Figure 4.4, a PPR stick of 110 cm length and 1 inch thickness was chosen. It was hollow inside in order to pass the wires for complete isolation, then three holes were punched in the dimensions (24, 56, and 85) cm for fixing the ultrasonic sensors. A control box was fixed at the opposite direction from the ultrasonic sensors at 30 cm from the ground.

Figure 4.4: Design of stick and sensor's positions

- The wheels of movement and balance with 6cm, 3cm diameter respectively were fixed on the chassis (22*14) cm where the stick can be removed and installed on it as shown in Figure 4.5.

Figure 4.5: Plan, front, and side elevation of chassis

4.3 System Hardware Implementation

In order to obtain the complete system's electrical circuit the connection of each component was performed separately as illustrated in the following steps:

- The raspberry pi pins 16, 19 and 22 were connected respectively to trigger pins of upper, bottom and stair ultrasonic sensors. Also echo pins were connected to the pins 18, 21, and 26 in a way that divide the voltage between two resistors with values of 1 kΩ and 2 KΩ to reduce the voltage from 5V to 3.3V, Equation 4.4 illustrates the voltage division:

$$\text{Vin} = \frac{\text{R1+R2}}{\text{R2}} * \text{Vout} \tag{4.4}$$

VCC pins which power the sensors were connected together and plugged in pin2, finally the sensors GND pins were connected together and inserted to pin 6. The Figures 4.6 and 4.7 show one of the sensors connection.

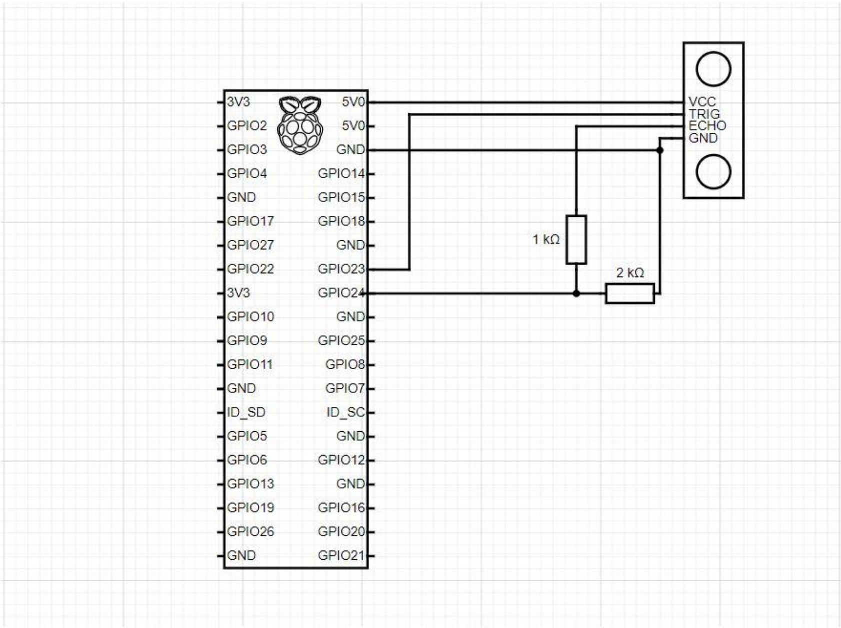

Figure 4.6: Connection of ultrasonic sensor in circuit diagram

Figure 4.7: Real connection of ultrasonic sensor

- DC gear motors were connected to Raspberry pi using relay module which was powered with 3.3V from pin 1 in raspberry pi. GND was connected to pin 6. IN1, IN2 were connected respectively to pins 11, 37. COM in relay was connected to the positive of the source, one of the motor terminals was connected to NO in the relay module and the other terminal was connected to the GND of source. The Figures 4.8 and 4.9 show the connection of motors and relay.

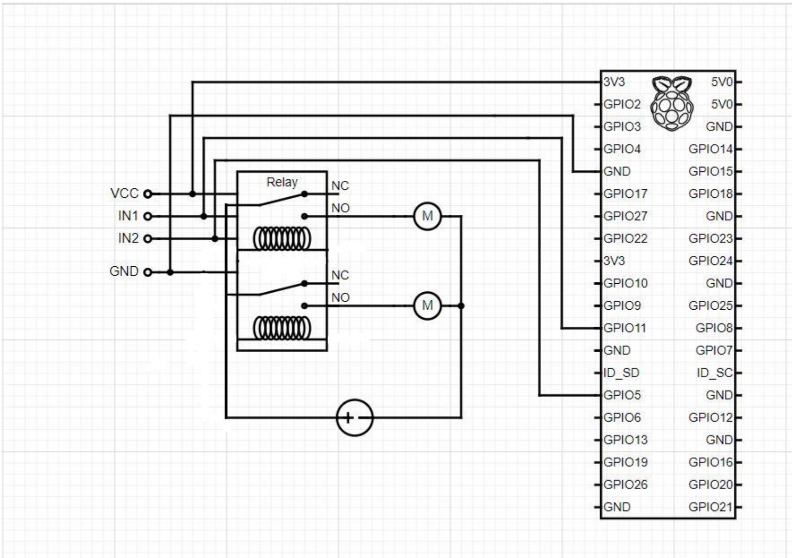

Figure 4.8: connection of motors and relay in circuit diagram

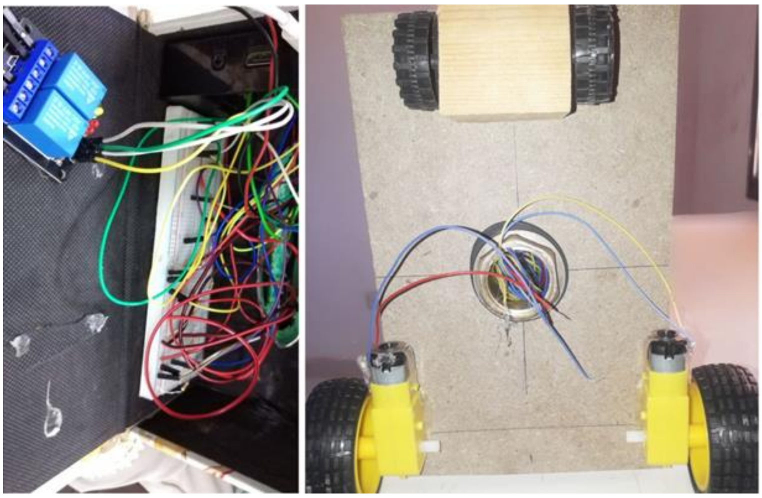

Figure 4.9: Real motors and relay connection

The system’s circuit consists of pair of DC gear motors powered by source with 3V, and three ultrasonic sensors. The complete circuit connections are shown in Figures 4.10 and 4.11.

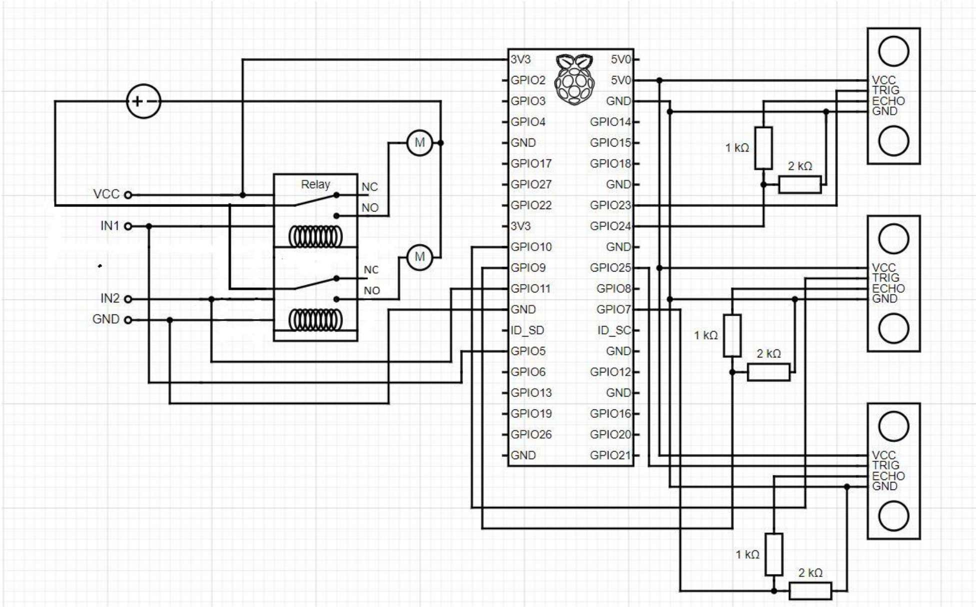

Figure 4.10: Complete circuit

Figure 4.11: Complete connections of the stick

4.4 Testing and Discussion

The first test targeted the detection of eight obstacles with different dimensions, one of them at 397cm of the stick (out of the desired range), the rest of the obstacles were placed at 120 cm of the stick.

The procedure was performed three times to check the ultrasonic reproducibility. From Table 4.1 it was noticed that the efficiency of detection was 100% as all objects were detected by at least one of the ultrasonic sensors. The detection accuracy of each ultrasonic was calculated from Equation 4.5, 97.35% was the upper ultrasonic efficiency, 95.61% was the bottom’s.

$$\text{efficiency} = \frac{\text{real distance} - \text{average error}}{\text{real distance}} * 100 \qquad (4.5)$$

Table 4.1: Upper and bottom sensors measures

NO	Obstacles size	Ultrasonic sensor	Average value	Error value	Detection
1	110*56 cm	Upper	117.56	2.44	True
		Bottom	114.51	5.49	
2	23*8 cm	upper	127.48	7.48	True
		bottom	114.97	5.03	
3	80*80 cm	upper	118.47	1.53	True
		bottom	109.74	10.26	
4	Mirror 104*54 cm	upper	120.43	0.43	True
		bottom	117.08	2.92	
5	8*23 cm	upper	128.61	8.61	True
		bottom	119.1	0.9	
6	47*33 cm	upper	118.78	1.22	True
		bottom	109.08	10.92	
7	Human	upper	120.21	0.21	True
		bottom	118.68	1.32	
8	concrete wall	upper	384.83	12.17	False
		bottom	391.27	5.73	

A graph shows upper and bottom sensor measurements shown in Figure 4.12.

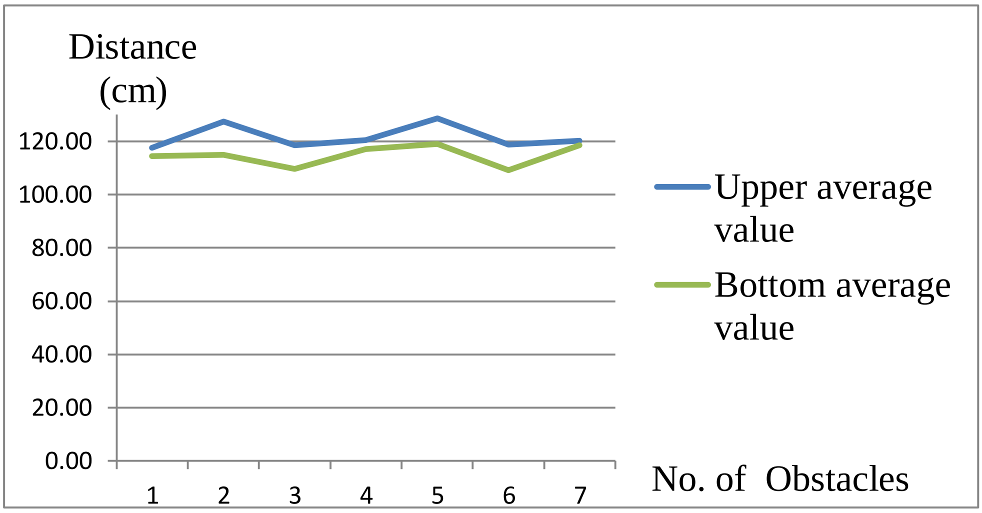

Figure 4.12: A graph shows upper and bottom sensor measurements

The second test was executed in two steps for each case:

- Case A: flat floor

    a. Primary step

It was performed to check the detection of the flat floor by the stair's ultrasonic. Based on the Equation 4.3 the ultrasonic measures should be 177.55cm, Table 4.2 shows the obtained result. From Equation 4.5 the accuracy of the detection was 99.22%.

Table 4.2: Stair sensor measurements in flat floor case

Ultrasonic measures	Error value
176.51	1.04
175.78	1.77
176.22	1.33

    b. Secondary step

The Table 4.3 shows each ultrasonic sensors measures while there's no obstacles in the range of 120 cm, in this case the microcontroller won't change its situation.

Table 4.3: Stair, upper, and bottom sensors measures in flat floor

Upper sensor	Bottom sensor	Flat floor
165.27	157.99	175.74
164.28	156.66	175.98
164.12	158.44	175.82
164.46	156.96	175.57
164.29	156.56	176.09
164.36	156.87	176.73
164.23	157.01	176.1
164.14	157.42	175.97

- Case B : upstairs

   a. Primary step

It was performed to check the detection of the upstairs by the stair's ultrasonic. Based on the Equation 4.3 the ultrasonic measures should range between (157.55 – 171.55) cm. From Table 4.4 it was noticed that the efficiency of detecting upstairs was 100% as all measures were in the range.

Table 4.4: stair sensor measures in up stair case

Up stair case	Test 1	Test 2	Test3
	167.58	167.76	167.86

   b. Secondary step

When the test was performed at the range of 120 cm, it was noticed that the upper and bottom sensors detected the upstairs as a regular obstacle, since the system should react differently in this case an increase in the detection range to 150 cm was necessary to prevent the upper and the bottom sensors from detecting upstairs, also to prevent the stair's ultrasonic from detecting other obstacles as stairs, a decrease range was determined (157.55-171.55).

The controller wouldn’t react to any obstacle as upstairs unless the obstacle had the same stair’s height. This problem is solved by taking another measure in a distance that equals the unit run’s distance (30 cm). If a less value was measured, the controller will deal with the case as upstairs, else, it will ignore the up stair’s ultrasonic signal and wait for the upper’s and bottom’s ultrasonic signals.

Table 4.5 shows that the microcontroller reacts according to the stair’s sensor only, since all the measures were out of the range of the upper and the bottom sensors.

Table 4.5: Stair, upper, and bottom sensors measures in up stair case

Upper sensor	Bottom sensor	Up stair sensor
154.46	146.51	168.44
154.86	146.17	169.36
154.68	146.52	167.92
153.93	149.01	167.56
154.6	147.38	167.29
154.83	146.06	168.74
154.78	147.58	168.93
155.12	146.92	168.69

- Case C: down stairs
  a. Primary step

It was performed to check the detection of the downstairs by the stair’s ultrasonic. The measures detect both downstairs and holes. Since the holes have irregular depth, the set increase from the flat flour’s value cannot be determined. The Table 4.6 shows the measures.

Table 4.6: Stair sensor measures in down stair case

Down stairs sensor	Test 1	Test 2	Test 3
	517.21	517.59	517.81

From Table 4.6 the measured distances in this case were more than 177.55 cm, therefore, the controller would be able to detect holes and down stairs. But these measures were not accurate, that's because of the ultrasonic wide beam which resulted in an incomplete reflection to the echo (unreal measures).

b. Secondary step

Table 4.7 shows that the microcontroller reacts according to the stair's sensor only, since all the measures were out of the range of the upper and the bottom sensors

Table 4.7: Stair, upper, and bottom sensors measures in down stair case

Upper sensor	Bottom sensor	Down stairs sensor
1201	447.95	515.29
1201.07	448.01	514.48
1201.16	448.35	514.07
1201.26	448.06	513.98
1201.17	447.55	514.58
1201.77	448.65	513.83
1201.71	447.74	515.62
1201.79	447.78	514.01

To check the algorithm implementation for the full integrated system a final test was run in four different circumstances: when the system was placed in a clear path it moved forward to continue in the desired path. An obstacle was placed in the system path within detection range, the system detected the obstacle successfully and activated the avoidance algorithm, the system took the clearest path as its best option and avoided the obstacle successfully, then the system scanned the primary path and returned to it as it was available, the system stopped when a similar test was performed as there was no possible path to avoid the obstacle. An indicator of upstairs was given when the system was within the range of 150 cm of acceding stairs.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The project develops an assistive cane for visually impaired to help them in the navigation procedure, thus increasing their self-sufficient ability. Ultrasonic sensors has been mounted in the stick to implement the detection function, and to obtain distance measures that are required in decision making process. Raspberry pi controller represented the suitable control option, with aid of ultrasonic sensors and dc geared motors, an avoidance algorithm has been implemented. PROTEUS environment was used to simulate the system circuit. The results of the integrated system showed that obstacles, upstairs, downstairs and holes were successfully detected and avoided.

5.2 Recommendations

At the end of this project many points will be taken as a suggested future works:

- Servo motors can be used to rotate the ultrasonic sensors instead of rotating the entire system so the user can follow the wheels only when the best path is chosen.
- Addition of water, mud detector and a camera to provide information about the nature of the obstacles in order to enhance the control action.
- Use of narrow beam ultrasonic for more accurate measures of downstairs and holes.
- Use of Radio Frequency technique for calling my stick facility.
- Addition of stairs climber facility.

- Global Positioning System module can be added to locate the user position and send it to stored number in case of emergency, it can also be used for navigation.

REFERENCES

[1] Mohammed Salah El-din, Hussein Kamal El-din Hussein, Abdulla Mustafa Abdul Raheem, Zahid Zain-Alabdin, “Smart Stick for Blind People”, Electrical and Nuclear Engineering, August 2014.

[2] Official website of the World Health Organization (WHO),www.who.int, 4 Nov 2020.

[3] Katsuuhinko Ogata, “Modern Control Engineering”, Aeezih, Fourth Edition 2002.

[4] Andrew K. Dennis, “Raspberry pi Home Automation with ARDUINO”, Packt Publishing Ltd, Birmingham-Mumbai, Second Edition Feb 2015

[5] Anca Discant, Alexandrina Rogozan, and Corneliu Rusul) and Abdelaziz Bensrhair, “Sensors for Obstacle Detection-A Survey”, National Institute of Applied Sciences, Rouen, 2007.

[6] Kajatheepan Kanagaratnam, “Smart Mobility Cane: Design of obstacle Detection”, Electrical and Biomedical Engineering Faculty, Hamilton, April 2007.

[7] Johann borenstein and iwan ulrich, “The GuideCane — A Computerized Travel Aid for the Active Guidance of Blind Pedestrians ", proceedings of the IEEE International Conference on Robotics and Automation IEEE, Apr 1997.

[8] M.Bouzit, A.Chailbi, K.J.De Laurentis, C. Mavroidis, ”Tactile feedback Navigation Handle for The Visually Impaired”, International Mechanical Engineering Congress, Anaheim, Nov 2004.

[9] Sylvin Cardin, Daniel Thalmann, FedericVexo, “Wearable system for Mobility Improvement of Visually Impaired People”, Switzerland, 2004.

[10] S.Meers, Koren Ward, “A substitute Vision System for Providing 3D Perception and GPS Navigation via Electro-tactle Stimulation, University of Wollongong, Australia, 2005.

[11] Kiyohide Ito, Makoto Okamoto, Junichi Akita, Tetsuo Ono, Ikuko Gyobu, Tomohito Takagi,Takahiro Hoshi and Yu Mishima, "CyARM: an Alternative Aid Device for Blind Persons", Conference on Human Factors in Computing Systems, Portland, Apr.2005.

[12] Lise A. Johanson, Charles M. Higgins, “A Navigation Aid for Blind Using Tactile-Visual Sensory Substitution”, 28^{th} IEEE International Annual Conference, New York, Aug 2006.

[13] D. Yuan, “Environmental Exploration via Computer Vision Techniques, ProQuest Information and Learning Company, 2007.

[14] Mounir Bousbia-Salah, Abdelghani Redjati, Mohamed Fezari, Maamar Bettayeb, "An Ultrsonic Navigation System for Blind People", IEEE International Conference on Signal Processing and Communications, Dubai, Nov.2007.

[15] Ziad O. Abu-Faraj, Elie Jabbour, Paul Ibrahim, Anthony Ghaoui, "Design and Development of a Prototype Rehabilitative Shoes and Spectacles for the Blind", 5th International Conference on Biomedical Engineering and Informatics IEEE, Beirut, 2012.

[16] Sonda Ammar Bouhamed,Imene Khanfir Kallel,Dorra Sellami,"New electronic white cane for stair case detection and recognition using ultrasonic sensor", International Journal of Advanced Computer Science and Applications, Sfax, June 2013.

[17] Rohit Sheth, Surabhi Rajandekar, Shalaka Laddha, Rahul Chaudhari,"Smart White Cane – An Elegant and Economic Walking Aid", American Journal of Engineering Research (AJER), 2014.

[18] Ayat Nada, Samia Mashelly, Mahmoud A. Fakhr, and Ahmed F. Seddik, "Effective Fast Response Smart Stick for Blind People", 2th International conference on advances in Bio-informatics and Environmental Engineering, 2015.

[19] Ankit Agarwal, Deepak Kumar, Abhishek Bhardwaj, "Ultrasonic Stick for Blind", International Journal of Engineering and Computer Science, PP. 11375-11378, Vol: 4, issue 4, Apr.2015.

[20] S.D. Asha Mahesh, K.Raj Supriya2, M.V.S.S.N.K.Pushpa Latha3, P.Gowri4, T.Sonia5, B. Nani, "Smart Assistive Shoes and Cane: Solemates for the Blind People", International Journal of Engineering Science and Computing, New Delhi, PP.16665-16672, Volume 8 Issue.4, Andhra Pradesh, Apr.2018.

[21] Vanita Babanne,Simranjeet Kaur2, Tejal Mehta3, Divya Mulay4, Rachana Nagarkar5,"Smart Walking System based on Artificial Intelligence", International Journal of Research in Engineering, Science and Management,PP.170-180,Volume-1, Issue-12, Dec-2018.

[22] Pruthvi S, Pushyap Suraj Nihal, Ravin R Menon, S Samith Kumar, Shalini Tiwari,"Smart Blind Stick using Artificial Intelligence",International Journal of Engineering and Advanced Technology (IJEAT), PP.19-22,Volume-8, Issue-5S, May, 2019.

[23] Jens E.Wilhjelm, Andreas Illum, Martin Kristensson, OleTrier Andersen," Medical Diagnostic Ultrasound-Physical Principle and imaging", Technical University of Denmark", ver.3.1, Dec-2016.

[24] Devesh Tripathi, "Practical Guide to Polypropylene", Rapra

Technology limited, 2002.

[25] James Sandars, “Linux, Open Source, and Software’s Future”, IEEE software, Sep 1998.

[26] Anand Nayyar, VikramPuri, “Raspberry Pi-A Small, Cost Effective, and Efficient Form Factor Computer: A Review”, International Journal of Advanced Research in Computer Science and software Engineering, PP.720-737, Volume 5, issue 12, Dec 2015.

[27] Charles Severance, “Guido Van Rossum: The Early Years of Python”, IEEE Computer Society, Feb 2015

[28] Donald Waston, Michael J.Crosbie, John Hancock Callender, “Time Saver Standards for Architectural Design Data”, McGrow-Hill Companies, 1991.

[29] Simon Monk, “Raspberry pi cookbook”, published by O’Reilly Media, second Edition, 2014.

[30] Tim Cox, “Raspberry pi Cookbook for python Programmers”, published by Packt publishing Ltd, first Edition, April 2014.

[31] Koichi Mizutani, NaotoWakatsuki, Tadashi Ebihara, “Introduction of Measurement Techniques in ultrasonic electronics: Basic Principles and recent trends”, Japanese Journal for Applied physics, June 2016.

[32] B. Mustapha, A. Zayegh, R.K.Begga, “Multiple sensors based obstacle detection system, 4th International Conference of Intelligent and advanced system”, IEEE, 2012.

[33] Mohd Helmy Abd Wahab, Amirul A.Talib, Herdawatie A.Kadir, Ayob Johari, A. Noraziah, Rosalina M. Sidek.Ariffin A. Mutalib, “Smart Cane: Assistive Cane for Visually Impaired People”, International Journal of Computer Science, PP: 21-27, Volume.8,

issue:4. July 2011.

[34]Muzaffar Khan, Shruti Chauhan, Sultan Ahmed, Pooja Thakre, Mayuri Raut," Calciamicus: Smart Shoes for Visually Impaired People", National Conference on Advanced in Engineering and Applied Science, PP: 148-153, Volume: 3, Issue: 2, Feb 2017.

[35] Mohammad Hazzaz Mahmud, Rana saha, Saymemul Islam, "Smart Blind Stick- an Electronic Approach to Assist Visually Disable Persons", International Journal of Scientific and Engineering Research, PP: 111-114, volume: 4, Issue: 10, Oct 2013.

[36] Mohammad Hassan, MD. Atiqur Rahman, Shakeb Alam, "Design of Arduino Based Shoe for Blind with Wireless Charging", International Journal of Electrical, Electronics and Data Communication, PP:29-34, volume:5, Issue:8, Aug 2017.

[37] Eklas Hossain, Raisuddin Khan, Ahad Ali, "Design and data analysis for a belt for visually impaired people", Int .J. Advanced Mechatronic Systems, 2011.

[38] M. Jouaneh, "Fundamentals of Mechatronics", Global Engineering, 2013.

[39] Austin Hughes. "Electric Motors and drives: Fundamentals, Types, and applications", Elsevier Ltd, Third Edition 2006.

APPENDIX A

Sensor Predefinitions.py

```
import RPi.GPIO as GPIO
import time

# Variables and pins assigning

bottom_sensor_trigger = 13
bottom_sensor_echo = 15
upper_sensor_trigger = 16
upper_sensor_echo = 18
stair_sensor_trigger = 22
stair_sensor_echo = 36


desired_distance = 120 # cm

GPIO.setup(bottom_sensor_trigger, GPIO.OUT)
GPIO.setup(bottom_sensor_echo, GPIO.IN)
GPIO.setup(upper_sensor_trigger, GPIO.OUT)
GPIO.setup(upper_sensor_echo, GPIO.IN)
GPIO.setup(stair_sensor_trigger, GPIO.OUT)
GPIO.setup(stair_sensor_echo, GPIO.IN)

def distance(trigger_pin, echo_pin):
```

```
    GPIO.output(trigger_pin, True)

    time.sleep(.000001) # changes
    GPIO.output(trigger_pin, False)

    start_time = time.time()
    stop_time = time.time()

    while GPIO.input(echo_pin) == False :
        start_time = time.time()

    while GPIO.input(echo_pin) == True :
        stop_time = time.time()

    time_differance = stop_time - start_time
    distance = (time_differance * 34300 )/ 2

    return distance



def detected():
        upper_distance = distance(upper_sensor_trigger, upper_sensor_echo)

        # for debugging
        #print("Upper sensor Distance is", upper_distance)

        #time.sleep(1)
```

```
	bottom_distance = distance(bottom_sensor_trigger,
bottom_sensor_echo)

	# for debugging
	#print("Bottom sensor Distance is", bottom_distance)

	if (upper_distance <= desired_distance or bottom_distance <=
desired_distance):
		return True
	return False

def upstairs():
	stairs_distance = distance(stair_sensor_trigger,stair_sensor_echo)
	if (distance >= 152 and distance <= 172):
		return True
	return False

def downstairs_or_holes():
	stairs_distance = distance(stair_sensor_trigger,stair_sensor_echo)
	if distance > 172:  # OR 180
		return True
	return False
```

APPENDIX B

Motor Predefinitions.py

```
import RPi.GPIO as GPIO
import time
motor_in1 = 11
motor_in2 = 37
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)

GPIO.setup(motor_in1, GPIO.OUT)
GPIO.setup(motor_in2, GPIO.OUT)

def STOP():
    GPIO.output(motor_in1,True)
    GPIO.output(motor_in2,True)
    #print("Motors STOP")

def FORWARD():
    GPIO.output(motor_in1,False)
    GPIO.output(motor_in2,False)
    #print("Motors FORWARD")

def RIGHT():
    GPIO.output(motor_in1,True)
    GPIO.output(motor_in2,False)
    #print("Motors RIGHT")

def LEFT():
```

```
    GPIO.output(motor_in1,False)
    GPIO.output(motor_in2,True)
    #print("Motors LEFT")

def ROTATE(direction):
    if direction == "E":
        RIGHT()

         # TESTING IS REQUIRED
        time.sleep(3) # here i assumed it takes 1 second to turn
        STOP()
    elif direction == "W":
        LEFT()

        # TESTING IS REQUIRED
        time.sleep(3)
        STOP()
    elif direction == "NE1":
        RIGHT()

    # TESTING IS REQUIRED
        time.sleep(3/3)
        STOP()
    elif direction == "NE2":
        RIGHT()

    # TESTING IS REQUIRED
        time.sleep(3*(2/3))
```

```
    STOP()
  elif direction == "NW1":
    LEFT()

  # TESTING IS REQUIRED
    time.sleep(3/3)
    STOP()
  elif direction == "NW2":
    LEFT()

  # TESTING IS REQUIRED
    time.sleep(3*(2/3))
    STOP()

def GO(direction, period): # Direction is one of [N, NE1, NE2, E, NW1,
NW2, W]
    if direction == "N":
        #print("We are on track my friend ;)") # for debugging
        FORWARD()
    elif direction == "E":
        ROTATE("E")
        #print("We are 90 degrees right to the N")
        FORWARD()
        time.sleep(period)
        STOP()
    elif direction == "W":
        ROTATE("W")
        #print("We are 90 degrees left to the N")
        FORWARD()
```

```
        time.sleep(period)
        STOP()
    elif direction == "NE1":
        ROTATE("NE1")
        #print("We are 30 degrees right to the N")
        FORWARD()
        time.sleep(period)
        STOP()
    elif direction == "NE2":
        ROTATE("NE2")
        #print("We are 60 degrees right to the N")
        FORWARD()
        time.sleep(period)
        STOP()
    elif direction == "NW1":
        ROTATE("NW1")
        #print("We are 30 degrees left to the N")
        FORWARD()
        time.sleep(period)
        STOP()
    elif direction == "NW2":
        ROTATE("NW2")
        #print("We are 60 degrees left to the N")
        FORWARD()
        time.sleep(period)
        STOP()
```

APPENDIX C

Relay.py

```
import RPi.GPIO as GPIO
import time

channel = 21

# GPIO setup
GPIO.setmode(GPIO.BCM)
GPIO.setup(channel, GPIO.OUT)

def motor_on(pin):
    GPIO.output(pin, GPIO.HIGH)  # Turn motor on

def motor_off(pin):
    GPIO.output(pin, GPIO.LOW)  # Turn motor off

if __name__ == '__main__':
    try:
        motor_on(channel)
        time.sleep(1)
        motor_off(channel)
        time.sleep(1)
        GPIO.cleanup()
    except KeyboardInterrupt:
        GPIO.cleanup()
```

APPENDIX D

Sound.py

```
import pygame
from os import getcwd

voices_path = getcwd()+"/Voices"

def say(soundfile_name):
        pygame.mixer.init()
        pygame.mixer.music.load(voices_path+"/"+soundfile_name)
        pygame.mixer.music.play()
        while pygame.mixer.music.get_busy() == True:
            continue
```

APPENDIX E

Alogorithm.py

```
from motor_predefinitions import *

from sensor_predefinitions import *

from sound import *

import sys


GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)


say("welcome.wav")


angle2direction_dict = {"30":"NE1", "60":"NE2", "90":"E", "-30":"NW1", "-
60":"NW2", "-90":"W"}

direction2angle_dict = {"NE1":"30", "NE2":"60", "E":"90", "NW1":"-30",
"NW2":"-60", "W":"-90"}

measurement_list = ["W", "NW2", "NW1", "ignored_N","NE1", "NE2", "E"]



def check_right_left():


    '''

    DESCRIPTION:

    - it rotates the sensor right and take 3 measurements (NE1, NE2, E)

    then rotates the sensor left and take more 3 measurements (NW1, NW2,
W)

    and after all six distances, the motors must go back to

    the current direction.

    - based on the highest yet appropriate distance,
```

```
it returns one of [N, NE1, NE2, E, NW1, NW2, W]


'''

angle2direction_dict = {"30":"NE1", "60":"NE2", "90":"E", "-30":"NW1", "-60":"NW2", "-90":"W"}

direction2angle_dict = {"NE1":"30", "NE2":"60", "E":"90", "NW1":"-30", "NW2":"-60", "W":"-90"}

measurement_list = ["W", "NW2", "NW1", "ignored_N","NE1", "NE2", "E"]

measurement_dict = {}

ROTATE(angle2direction_dict["-90"])

time.sleep(1)

upper_distance = distance(upper_sensor_trigger, upper_sensor_echo)

bottom_distance = distance(bottom_sensor_trigger, bottom_sensor_echo)

measurement_dict["W"] = [upper_distance, bottom_distance]


for i in range(6):

    if i == 2:

        ROTATE(angle2direction_dict ["30"])

        continue

    ROTATE(angle2direction_dict["30"])

    time.sleep(1)

    upper_distance = distance(upper_sensor_trigger, upper_sensor_echo)

    bottom_distance = distance(bottom_sensor_trigger, bottom_sensor_echo)

    measurement_dict[measurement_list[i+1]] = [upper_distance, bottom_distance]
```

```
        # for debugging
        print(measurement_dict) # which our algorithm decisions based on

        ROTATE(angle2direction_dict["-90"])

        our_options = []
        for direction in measurement_dict:
                if measurement_dict[direction][0] >= 100 and
measurement_dict[direction][1] >= 100: #240 is the practical choice
                        our_options.append(direction)

        if "NE1" in our_options:
                return "NE1"
        elif "NW1" in our_options:
                return "NW1"
        elif "NE2" in our_options:
                return "NE2"
        elif "NW2" in our_options:
                return "NW2"
        elif "E" in our_options:
                return "E"
        elif "W" in our_options:
                return "W"
        else:
                return None # our list is empty

obstacle_state = False

while True:
```

```
        time.sleep(1)
        if upstairs():
            say("upstairs.wav")
            STOP()
            say("stop")
            print("there is no option to take. better wait right here and ask for help")
            sys.exit()
        elif downstairs_or_holes():
            say("downstairs.wav")
            STOP()
            say("stop")
            print("there is no option to take. better wait right here and ask for help")
            sys.exit()

        if not detected() and obstacle_state == False:
            FORWARD()
            current_direction = "N" # desired direction
            shifted_angle = 0
            continue

        obstacle_state = True
        on_the_path = False

        if obstacle_state == True and current_direction == "N": # starting to to avoid the obstacle
            STOP()
            best_option = check_right_left() # out of six options [E, W, NE1, NE2, NW1, NW2, N]
```

```
            if best_option == None:
                STOP()
                say("stop")
                print("there is no option to take. better wait right here and
ask for help")
                sys.exit() # i don't know if this the right option


            current_direction = best_option # for example "NE1"
            shifted_angle = direction2angle_dict[current_direction] # convert
"NE1" to 30 degree
            GO(best_option, 5)
            ROTATE(angle2direction_dict["-"+shifted_angle])
            time.sleep(.5)
            upper_distance = distance(upper_sensor_trigger,
upper_sensor_echo)
            bottom_distance = distance(bottom_sensor_trigger,
bottom_sensor_echo)
            if upper_distance >= 100 and bottom_distance >= 100: #240
                on_the_path = True
            else:
                on_the_path = False
        if on_the_path == False and obstacle_state == True:
            ROTATE(shifted_angle)
            best_option = check_right_left()
            if best_option == None:
                STOP()
                print("there is no option to take. better wait right here and
ask for help")
                sys.exit() # i don't know if this the right option
```

```
            current_direction = best_option
            shifted_angle = direction2angle_dict[current_direction]
            GO(best_option, 5)
            ROTATE(angle2direction_dict[shifted_angle])
            time.sleep(.5)
            upper_distance = distance(upper_sensor_trigger, upper_sensor_echo)
            bottom_distance = distance(bottom_sensor_trigger, bottom_sensor_echo)
            if upper_distance >= 100 and bottom_distance >= 100:
                on_the_path = True
            else:
                on_the_path = False


        elif on_the_path == True and obstacle_state ==True: # assuming here obstacle has been avoided
            obstacle_state = False
```