SMART EYE FOR BLIND PEOPLE

A PROJECT REPORT

Submitted in partial fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

by

PIYUSH DHIMAN
(Univ. Roll No.: 1900290149071)

Under the Supervision of
Mr. Ankit Verma
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
DR. A.P. J. ABDUL KALAM TECHNICAL UNIVERSITY
LUCKNOW (AUGUST ,2021)

DECLARATION

I hereby declare that the work presented in this report entitled “SMART EYE FOR BLINDPEOPLE", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution.I have used quotation marks to identify verbatim sentences and given credit to the originalauthors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reportedin the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

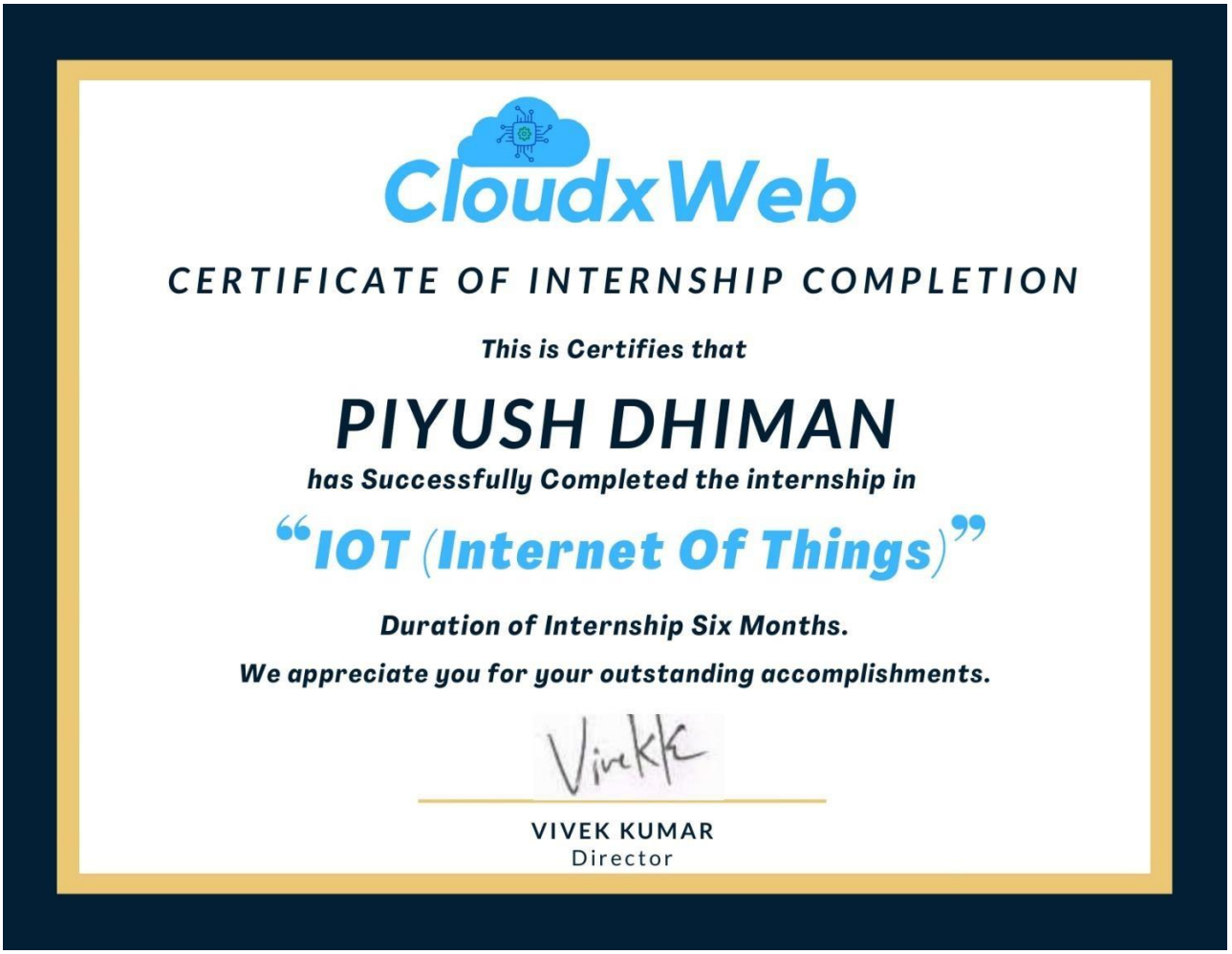
Name: Piyush Dhiman

Roll.No.:1900290149071

Branch: MCA

(Candidate Signature)

TRAINING CERTFICATE



CERTIFICATE

Certified that **Piyush Dhiman** **(**enrollment no 190029014005256) has carried out the project work presented in this thesis entitled **"SMART EYE FOR BLIND PEOPLE"** for the award of **Master of Computer Application** from Dr. APJ Abdul Kalam Technical University, Lucknow under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University.

Mr. Ankit Verma **External Examiner**

(Internal Examiner)

Assistant Professor

Dept. of Computer Applications

KIET Group of Institutions, Ghaziabad

Date:

ACKNOWLEDGEMENT

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Mr. Ankit Verma** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava**, Professorand Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot in many criticalconditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kinds of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Piyush Dhiman

Enrollment no-190029014005256

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO PROJECT

Blind People confront many problems in moving from one place to another. Vision is human's power to notify him of the obstacles in his way. A solution that is easily available is needed to solve the problems of blind people. The application developed canrecognize the objects in the user's surroundings. It can alert the user of the obstacles in his pathway and this way helps the user to navigate from one place to another saving him from tripping anywhere. It will also solve the problem of keeping a walking stick.

Blindness is a state of lacking the visual perception due to physiological or neurological factors. The partial blindness represents the lack of integration in the growthof the optic nerve or visual center of the eye, and total blindness is the full absence of thevisual light perception. In this work, a simple, cheap, user-friendly headgear designed with a pi camera, mounted on the head, to recognize the obstacle and improve the mobility of both blind and visually impaired people in a specific area. The proposed work includes a wearable equipment headgear to help the blind person to navigate alone safely and to avoid any obstacles that may be encountered, whether fixed or mobile, to prevent any possible accident. The main component of this system is the ultrasonic sensor and pi camera which is used to scan obstacles around blind by emitting-reflecting waves. The reflected signals received from the barrier objects are used as inputs to the microcontroller. The microcontroller [5] is then used to determine the direction and distance of the objects around the blind and obstacles are recognized and informed to the user through

earphones. For object recognition, deep learning is used and implemented using Raspberry pi. It recognizes the obstacles which are present in the dataset of the system and informs about it through earplugs. It detects all kinds of obstacles as it controls the peripheral components that alert the direction to the user by ticking on the forehead. The implemented system is fast, and easy to use and an innovative affordable solution to blindand visually impaired people.

Figure 1 Arduino Uno

1.1.1 OBJECTIVE OF PROJECT

Millions of people live in this world with incapacities of understanding the environment due to visual impairment. Although they can develop alternative approachesto deal with daily routines, they also suffer from certain navigation difficulties as well as social awkwardness [12]. For example, it is very difficult for them to find a particular room in an unfamiliar environment. And blind and visually impaired people find it difficult to know whether a person is talking to them or someone else during a conversation. Computer vision technologies, especially the deep convolutional neural network, have been rapidly developed in recent years. It is promising to use the state-of-art computer vision techniques to help people with vision loss. In this project, we want to explore the possibility of using the hearing sense to understand visual objects. The sense of sight andhearing sense share a striking similarity: both visual object and audio sound can be spatially localized. It is not often realized by many people that we are capable of identifying the spatial location of a sound source just by hearing it with two ears. In our project, we build a real-time object detection model.

1.2 PROJECT DESCRIPTION

Blind people are the people who can't identify the smallest detail with healthy eyes. Those who have the visual acuity of 6/60 or the horizontal extent of the visual field with both eyes open less than or equal to 20 degrees, these people are considered blind. Such people are in need of aiding devices for blindness related disabilities. As 10% of blind have no usable eyesight at all to help them move around independently and safely. The electronic aiding devices are designed to solve such issues. To record information about the obstacles presence in a road, active or passive sensors can be used. In case of a passive sensor, the sensor just receives a signal [2]. It detects the reflected, emitted or transmitted electro-magnetic radiation provided by natural energy sources. In case of using an active sensor, the sensor emits a signal and receives a distorted version of the reflected signal. It detects reflected responses from objects irradiated with artificially generated energy sources. These kinds of active sensors are capable of sensing and detecting far and near obstacles. In addition, it determines an accurate measurement of the distance between the blind and the obstacle. Overall, in the obstacle detection and recognition domain, our different types of active sensors may be used: infrared, laser, ultrasonic, radar sensors in addition to pi camera.

Visually impaired people find difficulties detecting and recognizing obstacles in front of them, during walking in the street, which makes it dangerous. The smart eye comes as a proposed solution to enable them to get notified when obstacles are around them. In this project we propose a solution, represented in a cap with ultrasonic sensor to detect obstacles and a servo motor attached with a needle through which direction of any obstacle can be notified to the user by ticking of needle on forehead, within a range of four meters. In this, a pi camera module is also mounted on the cap for the recognition of obstacles which later on inform the person through earplugs by converting text to sound. It will recognize the objects presented in its dataset and inform the user through earplugs.

1.2.1 Advantages of Project

- The cap which is mounted by a sensor produces the beep sound when any obstacle isdetected.
- Objects are recognized and make the person alert.
- The user of these devices will be able to walk on the street with more comfort.

1.2.2 Disadvantages of Project

- Recognition of obstacles can be a little delayed.
- It does not detect pit holes.
- It only recognizes objects which are present in the dataset.

1.3 PROJECT SCOPE

Nowadays, technology and human life cannot be separated as it has become the phenomenon of the world. But, how can technology help people that are visually impaired? [1] Blind people usually can estimate the obstacle in front of them without knowingthe actual distance of the obstacle from them. Mobility for the blind people can be defined as mobility to move with safety and ease through the environment without relying on others. Most commonly mobility aid used by the blind are cane and guide dogs to facilitate their movement. But there are problems with this navigation support. The cane provides limited preview for the user and as a result, the user has to be more careful to walk and mobile very slowly. As for the guide dogs, the training and coordinating the dogs with blind people is a difficult task and the results are minimal. In order to overcome this problem, research on the assistant devices for the blind has been done by many people tohelp reduce the limited ability of the blind people. The assistive headgear for the blind, is a device that can help visually impaired to facilitate obstacle detection as well as recognition without relying too much on others. To achieve the objectives, the scope of this project is determined. For the hardware ultrasonic sensors will be used as a sensor to detect obstacles at the front and it will send signals to Arduino UNO which is a microcontroller. The microcontroller will then process data and send the signal to

Servo Motor which will guide through its vibrating feedback [3]. Pi camera module which is mounted on the top of cap, will recognize the obstacle and inform it to the person by producing a sound beep from the buzzer. For the software, the design of the circuit is done using Arduino IDE and Visual Studio. Thiswill also produce:

- Independent Mobility
- Easy to detect obstacle
- Better way to explore surrounding
- Easy to recognize obstacle
- System is flexible and secure to use
- Saves times and reduce inter-dependency

1.4 HARDWARE / SOFTWARE

1.4.1 HARDWARE

1.4.1.1 Arduino Uno

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. It is an open- source platform, means the boards and software are readily available and anyone can modify and optimize the boards for better functionality [10].

The software used for Arduino devices is called IDE (Integrated Development Environment) which is free to use and requires some basic skills to learn it. It can be programmed using C and C++ language.

The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and is programmable with the Arduino IDE(Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is alsosimilar to the Arduino Nano and Leonardo [8]. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The ATmega328 on the

Arduino Uno comes preprogrammed with a bootloader that allows uploading new code to it without theuse of an external hardware programmer.

Fig-1.1 Arduino UNO

- **General Pin functions**

1. **LED:** There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
2. **VIN:** The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
3. **5V:** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
4. **3V3:** A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
5. **GND:** Ground pins.
6. **IOREF:** This pin on the Arduino/Genuino board provides the voltage reference

with which the microcontroller operates. A properly configured shield can read theIOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

7. **Reset:** Typically used to add reset button to shields which block one on the board.

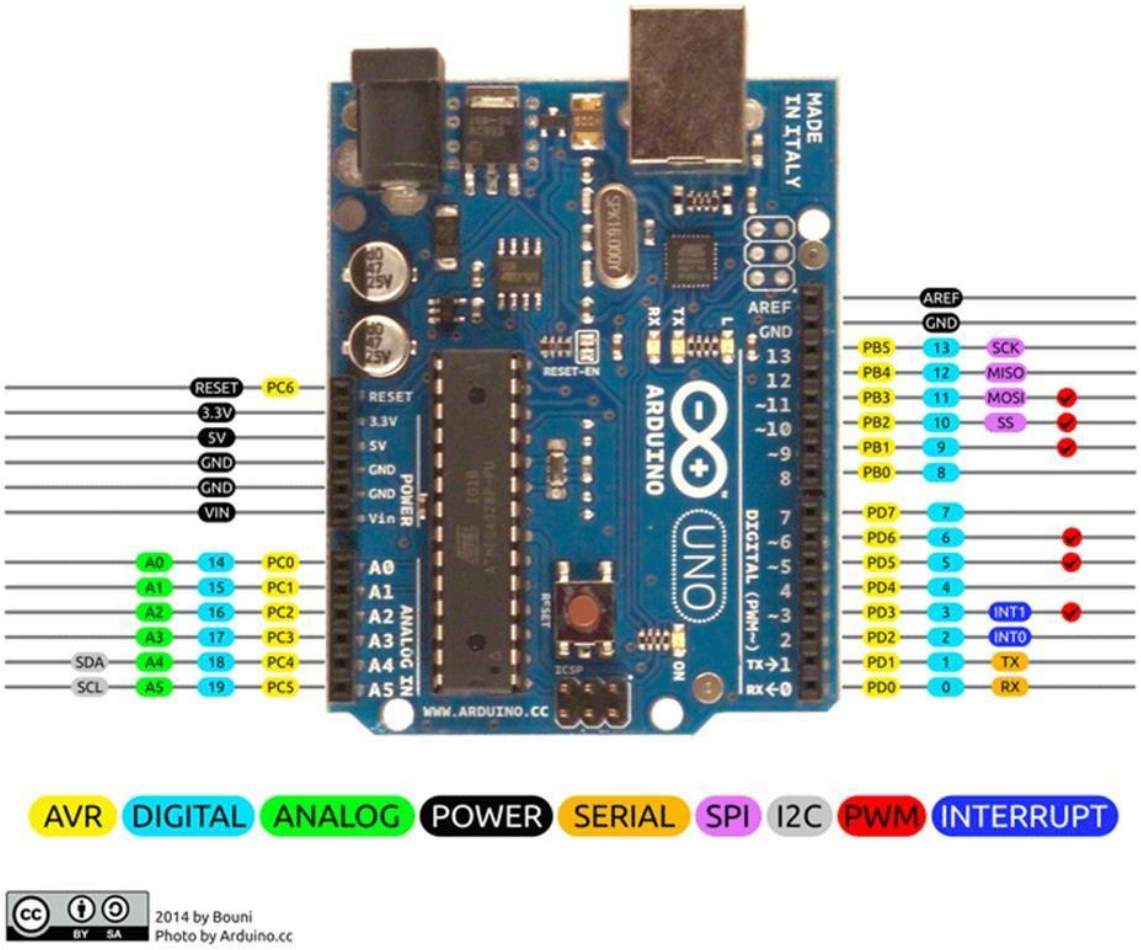


Figure 2 Arduino Pin Diagram

1.4.1.2 Raspberry pi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python[10]. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, tomaking spreadsheets, word-processing, and playing games.

What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting bird houses with infra-red cameras. Wewant to see the Raspberry Pi being used by kids all over the world to learn to program andunderstand how computers work.

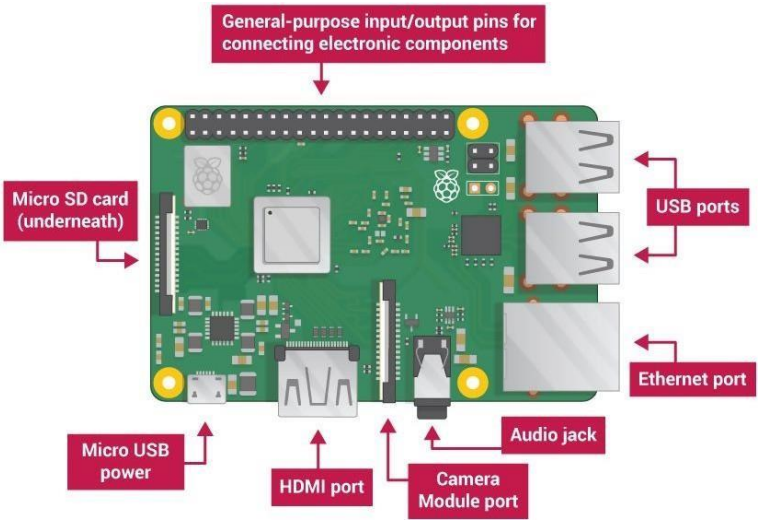


Fig. : 2 Raspberry pi

Advantages of Raspberry Pi

- Low cost.
- Huge processing power in a compact board
- Many interfaces (HDMI, multiple USB, Ethernet, onboard Wi-Fi and Bluetooth, many GPIOs, USB powered, etc.)

Disadvantages of Raspberry Pi

- The memory of the Raspberry Pi is more limited than you're probably used to, with just 512MB or 256MB available. You can't expand that with extra memory in the way you can a desktop PC.
- The graphics capabilities lag behind today's market

1.4.1.3 Ultrasonic Sonar

As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves. The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. It comes complete with ultrasonic transmitter and receiver modules.

The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

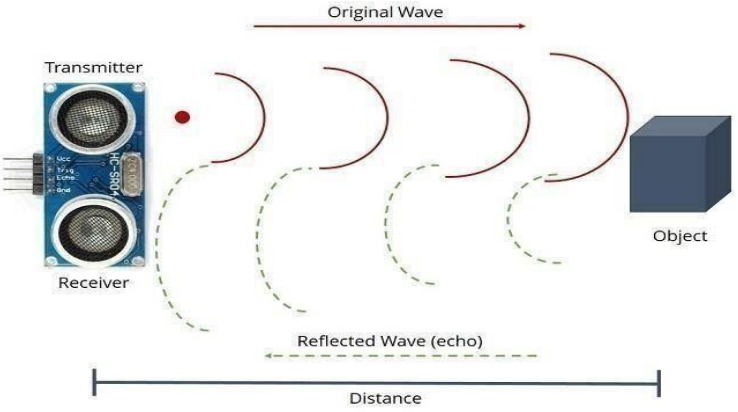


Fig.:3 Ultrasonic Sensor

- **HC-SR04 Ultrasonic Sensor Pinout**

VCC: +5VDC

Trigger: Trigger (INPUT)Echo: Echo(OUTPUT)

GND: GND

Fig-1.4 Ultrasonic sensor with pins & circuit

- **Formula**

  Distance= duration*0.034/2

➔ ➔**Advantages of Ultrasonic Sensors**

- Not affected by color or transparency of objects can be used in darkenvironments
- Low-cost option
- Not highly affected by dust, dirt, or high-moisture environments

➔➔ **Limitations of Ultrasonic Sensors**

- Cannot work in a vacuum
- Not designed for underwater use
- Sensing accuracy affected by soft materials
- Sensing accuracy affected by changes in temperature of 5-10 degrees or more

1.4.1.4 Pi Camera Module

The Pi camera module is a portable lightweight camera that supports Raspberry Pi. It communicates with Pi using the MIPI camera serial interface protocol. It is normally used in image processing, machine learning or in surveillance projects.

The steps for installation of Raspberry Pi Camera are:

1. Open up your Raspberry Pi Camera module.
2. Install the Raspberry Pi Camera module by inserting the cable into the RaspberryPi.
3. Boot up your Raspberry Pi.
4. From the prompt, run "sudo raspi-config".
5. Run "sudo raspi-config" again - you should now see the "camera" option.
6. Navigate to the "camera" option, and enable it.

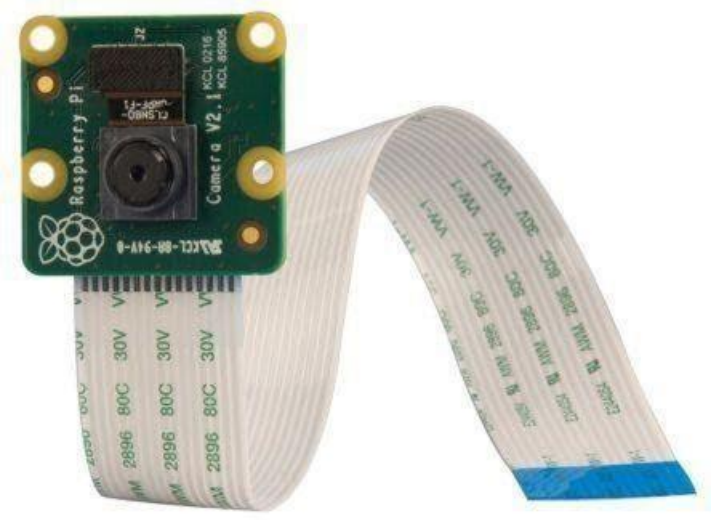


Fig.: 5 Pi Camera Module

1.4.1.5 Jumper Wires

A jump wire (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the

components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

Fig.: 6 Jumper Wires

1.4.2 SOFTWARE

1.4.2.1 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, withthe help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires twobasic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

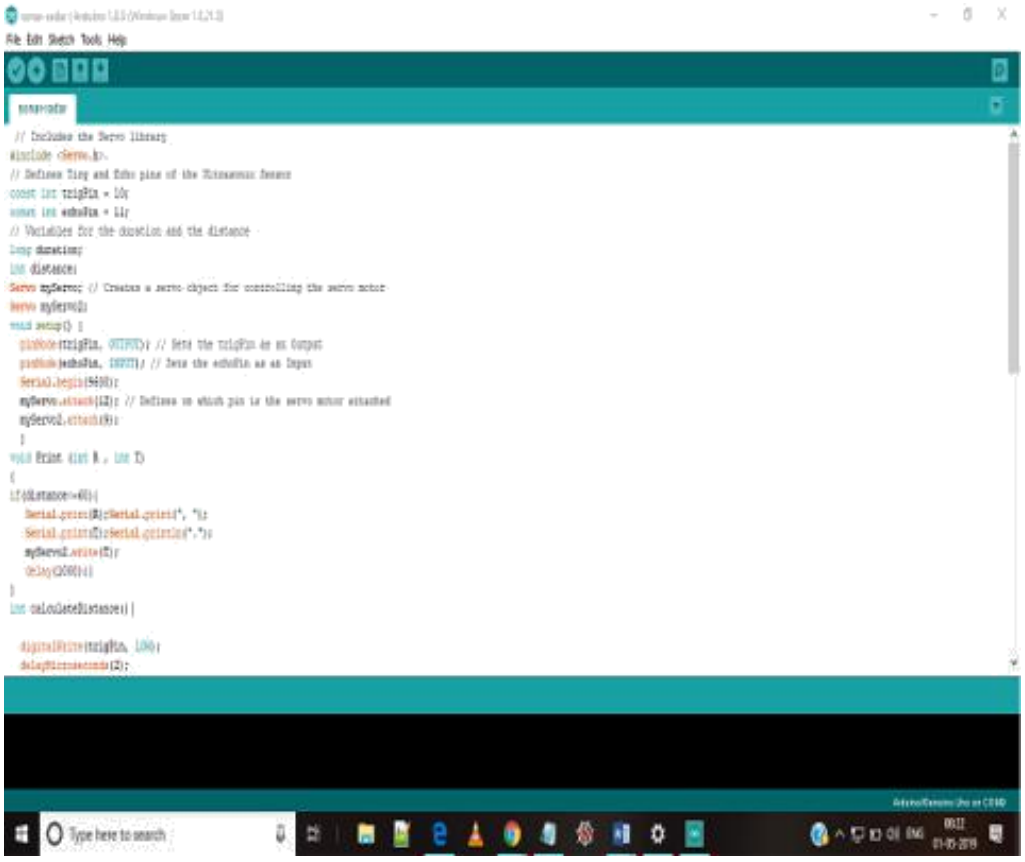


Fig.: 7 Arduino IDE

1.4.2.2 Visual Studio Code

Visual Studio Code is a source-code editor developed by Microsoft for Windows,Linux and macOS. It includes embedded Git and support for debugging, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. The source code is free and open- source, released under the permissive MIT License. The compiled binaries are freeware for any use.

Visual Studio Code is a source code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js and C++. It is based onthe Electron framework, which is used to develop Node.js web apps that run on the Blinklayout engine. Although it uses the Electron framework, the software does not use

Atom and instead employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse.

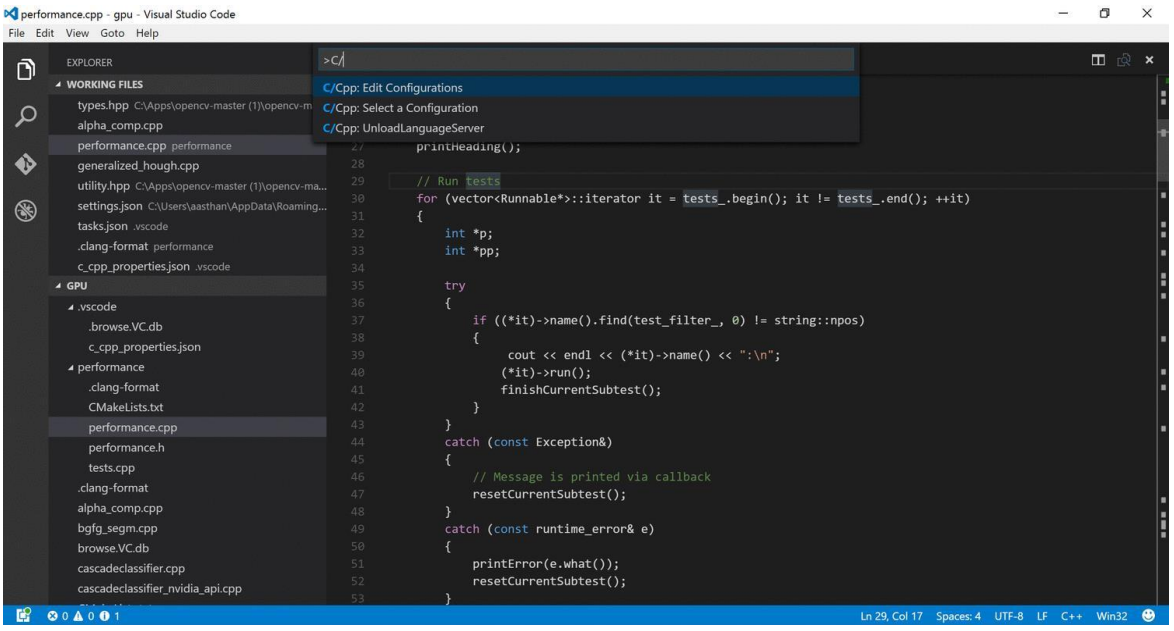


Fig-1.9 Visual Studio Code

CHAPTER 2

LITERATURE REVIEW

2.1 OBJECT DETECTION AND RECOGNITION

Object detection can be applied into many computer vision areas, such as video surveillance, robotics, and human interaction. However, due to the factors of complex background, illumination variation, scale variation, occlusion, and object deformation, object detection is very challenging and difficult. In the past few decades, researchers have done a lot of work to push the progress of object detection[1].

Depending on whether deep learning is used or not, object detection methods can be divided into two main classes: the handcrafted feature-based methods and the deep learning-based methods. In the first decade of the twenty-first century, the traditional handcrafted feature-based methods are mainstream. During this period, many famous and successful image feature descriptors are proposed. Based on these feature descriptors and the classical classifiers (e.g., SVM and AdaBoost), these methods achieve great success at that time. However, when it comes to 2010, the performance of objection detection tends to be stable. Though many methods are still proposed, the performance improvement is relatively limited. Meanwhile, deep learning begins to show superior performance on some computer vision areas (e.g., image classification).In 2012, with the big image data (i.e., ImageNet), deep CNN network (called AlexNet) achieved the best detection performance on ImageNet ILSVRC2012, which output forms the second best method by 10.9% on ILSVRC-2012 competition.

With the great success of deep learning on image classification, researchers start to explore how to improve object detection performance with deep learning.

2.2 THE ARDUINO AND RASPBERRY PI

2.2.1 Microcontroller: Arduino

The microcontrollers we select for this book come from the Arduino family, because they are supported by an easy-to-use programming environment that allows us to quickly start developing software [11]. There are limitations, but it is a wonderful system with which to start learning about microcontrollers. Here we need to point out that by "Arduino" we mainly mean the development ecosystem, rather than the processor itself, because that has evolved and comprises a number of different hardware incarnations. In this chapter we will discuss a few of them.

2.2.1.1 Hardware

The original Arduinos are based on Atmel microcontrollers, and we mostly discuss the Arduino UNO[9]. Support for a second family of controllers, based on the ESP8266 microcontroller, was recently integrated into the Arduino development environment. These controllers can be programmed in much the same way as UNOs, but have native wireless support built in. But let's start with the UNOs.

2.2.2 Arduino UNO

An Arduino UNO is shown in Figure 2.4, where the main component is the ATmega328p microcontroller from Atmel (now Microchip TM), which is the large chip with 28 legs in the image. It is a controller with 8-bit-wide registers, and operates at a clock frequency of 16 MHz It has 32 kB RAM memory and 1 kB non-volatile EEPROM memory, which can be used to store persistent data that need to survive turning off and on the supply voltage[4]. There are three timers on board, which are basically counters that count clock cycles and are programmable to perform some action, once a counter reaches some value. The UNO interacts with its environment through 13 digital input– output (IO) pins, of which most can be configured to be either input or output, and have software-configurable pull-up resistors. Several of the pins are configurable to support I2C, SPI, and RS-232 communication. Moreover, there are six analog input pins. They measure voltages of up to the supply voltage of 5 V. An

alternative internal reference voltage source provides a 1.1 V reference. All digital and analog pins are routed to pin headers that are visible on both sides of the Arduino printed circuit board (PCB) in Figure. Furthermore, the built-in hardware RS-232 port is connected to an RS-232-to-USB converter that allows communication and programming from a host computer. There is no Wi-Fi, Bluetooth, or Ethernet support on the UNO board, but extension boards, so-called shields, are available for mounting directly onto the pin headers.

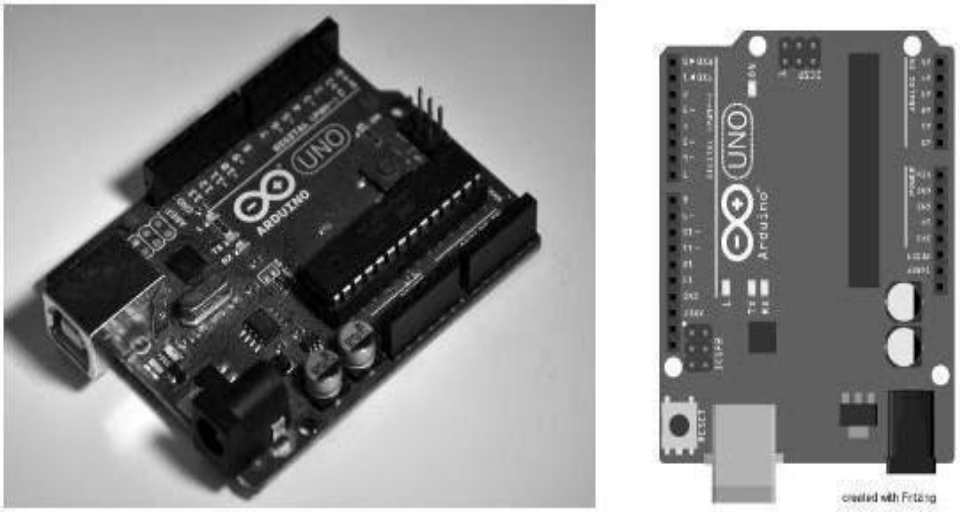


Fig.: 9 Arduino Uno Circuit

2.2.3 Raspberry Pi

The Raspberry Pi is a small single-board computer that first appeared in 2012, with the intention of providing an inexpensive platform to introduce students and other interested parties to computers in general and to programming in particular.

2.2.3.1 Hardware

Since its first appearance, the Raspi went through several hardware revisions, and the present incarnation, model 3, appeared in February 2016. This version, shown in Figure 5.1, features a quad core ARM central-processing unit (CPU) operating at 1.2 GHz and has 1 GB RAM memory on board. This hardware base is sufficiently powerful to run a full-fledged Linux system. An external and replaceable micro-SDHC card (preferably speed class 10) serves as a hard disk to hold the operating system and user files.

But beyond the CPU, the Raspberry Pi sports a video processor that can display videos at full- HD resolution (1920×1080) via the built-in HDMI-connector[6]. Audio output is available either via the HDMI connector or via a 3.5-mm headphone connector. Moreover, there are four USB-2 ports on board to connect peripheral components, such as USB sticks, keyboard, mice, or web cameras. Communication with the outside world is feasible via a built-in wired Ethernet port, and since version 3, the Raspi has had built-in Bluetooth (V4.1) and Wi-Fi (802.11n).

The Raspis are very attractive due to their built-in low-level peripherals[7]. There are17 general- purpose input–output (GPIO) pins exposed on the board, some of which support I2C, SPI, and UART (RS-232-like) communication. Moreover, a specific audio bus (I2S) is available, as well as a high- speed CSI interface to connect the tailor-made Raspberry Pi camera, and a DSI interface to connect LCD panels. Some of these featurescan be used to implement the same functionality as on the Arduino, but we will not use that here, using the Raspi as a standardized host computer system instead. [6]

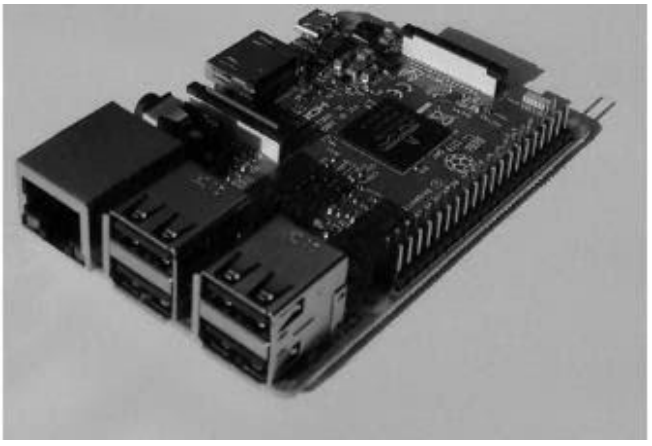


Fig-10 Raspberry pi Board

2. 3 PROGRAMMING LANGUAGE: C/C++ and Microcontroller

2.3.1 What Is Arduino Programming?

The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL. **Sketch** − The first new terminology is the Arduino program called "**sketch**".

STRUCTURE OF ARDUINO PROGRAM

Arduino programs can be divided in three main parts: **Structure, Values** (variables and constants), and **Functions**. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the **Structure**. Software structure consist of two main functions −

- Setup () function
- Loop () function

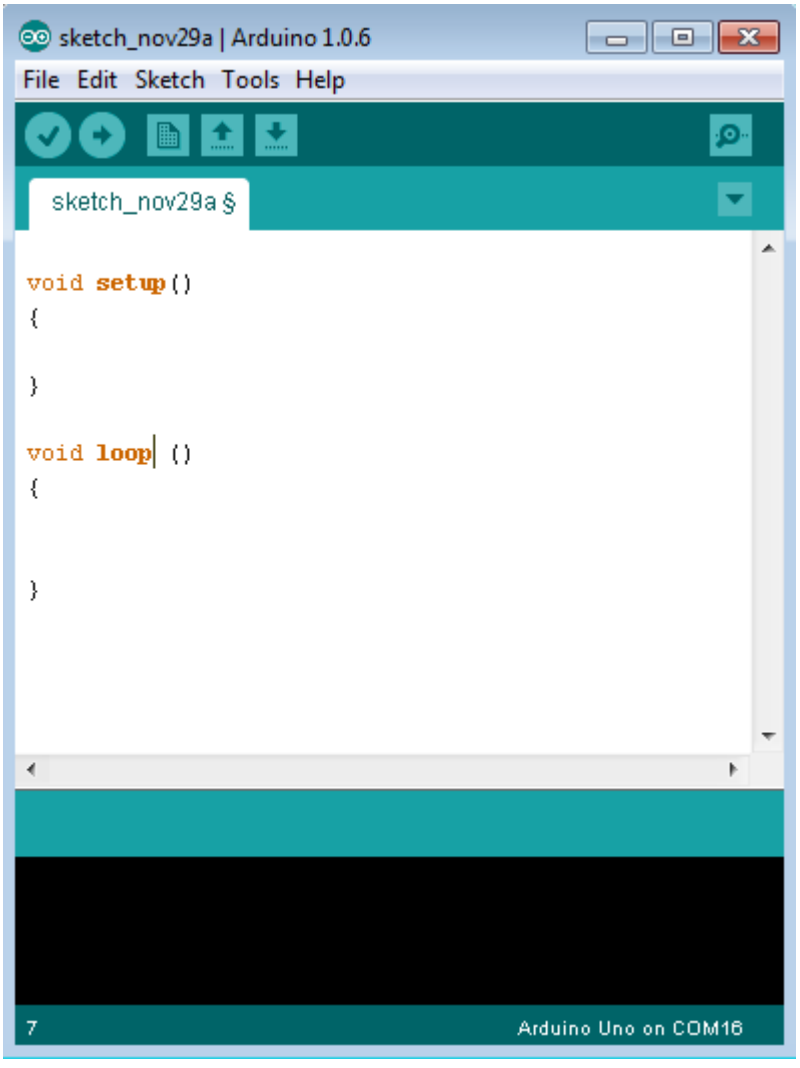


Fig. 11.: Program Structure

2.4 Data Structures

The real power of Programming lies in the liberal usage of its data structure. The common criticism of Arduino is that it is slow compared to C/C++. This is especially true if multiple for-loops are used in programming.. Hence for-loops are not the most efficient way for programming. The alternative is to use data structures such as lists, tuples, dictionary and sets. We describe each of these data structures in this section.

Lists

Lists are similar to arrays in C/C++. But, unlike arrays in C/C++, lists in Python can hold objects of any type such as int, float, string and including another list. Lists can also have objects of different type. Lists are mutable, as their size can be changed by adding or removing elements. The following examples will help show the power and flexibility of lists.

List functions

Let us consider the list that we created in the previous section. We can sort the list using the sort function as shown in line 2. The sort function does not return a list; instead, it modifies the current list. Hence the existing list will contain the elements in a sorted order. If a list contains both numbers and strings, It sorts the numerical values first and then sorts the strings in alphabetical order.

List comprehensions

A list comprehension allows building a list from another list. Let us consider the case where we need to generate a list of squares of numbers from 0 to 9. We will begin by generating a list of numbers from 0 to 9. Then we will determine the square of each element. For a new programmer, the list comprehension might seem daunting. The best way to understand and read a list comprehension is by imagining that you will first operate on the for-loop and then begin reading the left part of the list comprehension. In addition to applying for-loop using list comprehension, you can also apply logical

Tuples

Tuples are similar to lists except that they are not mutable, i.e. the length and the content of the tuple cannot be changed at runtime. Syntactically, the list uses [ ] while tuples use ( ). Similar to lists, tuples may contain any data type including other tuples.

Sets

A set is an unordered collection of objects. A set can contain objects of any datatype. To create a set, we need to use the function set. We create a set from a list containing four values. Then we create a set containing a tuple. The elements of a set need to be unique. Hence when the content of s2 is printed, we notice that the duplicates have been eliminated. Sets can be operated using many of the common mathematical operations on sets such as union, intersection, set difference, symmetric difference etc.

Since sets do not store repeating values and since we can convert lists and tuples to sets easily, they can be used to perform useful operations faster which otherwise would involve multiple loops and conditional statements. For example, a list containing only unique values can be obtained by converting the list to a set and back to a list.

File handling

This book is on obstacle; however, it is important to understand and be able to include in your code, reading and writing of text files so that the results of computation or the input parameters can be read from external sources. Programming in arduino providesthe ability to read and write files. It also has functions and methods for reading specialized formats such as csv, Microsoft Excel (xls) format etc. We will look into each method in this section.

User defined functions

A function allows reuse of code fragments.

```
import math
void circleproperties(r):
area = math.pi*r*r;
circumference = 2*math.pi*r;return (area,circumference)
```

(a,c) = circleproperties(5) # Radius of the circle is 5 print “Area and Circumference of the circle are”a,c

The function circle properties takes in one input argument, the radius (r). The return statement at the end of the function definition passes the computed values (in this case area and circumference) to the calling function, circle properties. To invoke the function, use the name of the function and provide the radius value as argument enclosed in parenthesis. Finally, the area and circumference of the circle are displayed using the print command. The variables area and circumference have local scope. Hence the variables cannot be invoked outside the body of the function. It is possible to pass on variables to a function that have global scope using the global statement.

2.5 ARDUINO BASED EMBEDDED SYSTEM

Arduino is a user-friendly open-source platform. Arduino has an onboard microcontroller and an integrated development environment (IDE) is used to program it. Arduino board can be programmed directly from the PC using FTDI which is easy compared to other similar platforms. The advantages are as follows:

- **Low cost:** Arduino boards are of relatively low cost as compared to other microcontroller platforms.
- **Cross-platform:** The Arduino software (IDE) is compatible with theWindows, Macintosh OSX, and Linux operating systems.
- **User friendly:** The Arduino software (IDE) is user friendly and easy to use forbeginners and very flexible for skilled programmers.
- **Open source:** The Arduino is an open-source software and can be programmed with C, C++, or AVR-C languages. So, a variety of modules can be designed by the users.

Arduino platform comprises a microcontroller. It can be connected to a PC through a USB cable. It is freely accessible and can be easily downloaded. It can also be modified by a programmer[14]. Different versions of Arduino boards are available in the market depending on the user requirement.

2.5.1 Arduino Uno

The Arduino/Genuino Uno has an onboard ATmega328 microcontroller. It has onboard six analog input ports (A0–A5). Each pin can operate at 0–5 V. It has 14 digital input/output (I/O) pins out of which 6 are PWM output, 6 analog inputs, 2 KB SRAM, 1KB EEPROM, and operates at 16 MHz of frequency [11].

2.6 INTERNET OF THINGS

There have been several international organizations and research centers involvedin the creation of common standards for the IoT. One of the first steps in this process hasbeen to find a common definition [9]. The first definitions of the IoT were tightly coupled to the radio-frequency identification (RFID)–related context of the Auto- ID Labs at MIT, where the term first emerged. As the concept became universal, the definition started to evolve to more general terms. For Kevin Ashton, the meaning of the IoT and the consequences of its implementation in our environments are the following(Ashton, 2009):

"If we had computers that knew everything there was to know about things—usingdata they gathered without any help from us—we would be able to track and count everything, and greatly reduce waste, loss and cost. We would know when things neededreplacing, repairing or recalling, and whether they were fresh or past their best. We need to empower computers with their own means of gathering information, so they can see, hear and smell the world for themselves."

Another definition of the IoT is the following (Atzori et al., 2010):

"The basic idea of this concept is the pervasive presence around us of a variety ofthings or objects such as Radio-Frequency Identification (RFID) tags, sensors, actuators, mobile phones, etc. – which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals."

The definition published on the IERC website states that the IoT is [9]

"A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual 'things' have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network. "

CHAPTER 3

FEASIBILITY STUDY

3.1 TECHNICAL FEASIBILITY

A large part of determining resources has to do with assessing technical feasibility. The technical requirements considered in the proposed project are as this project is basically built for blind people so the basic requirement is to generate an environment for the person through the help of the project so that he can detect and recognize the obstacleson his way. The technical work which is done in project Object Recognition for Visually Impaired People is considered technically feasible because its internal technical capability is sufficient to support the project requirements.

Here we are using an ultrasonic sensor with an Arduino microcontroller and pi camera with Raspberry pi with some set of instructions. These instructions are all about obstacle detection as well as recognition and providing info to the user through earplugs of the user and direction by ticking on the forehead. This project will use 4 major hardware components i.e. Arduino UNO, Pi camera, Raspberry pi and ultrasonic sonar.

3.2 OPERATIONAL FEASIBILITY

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented.

Operational feasibility is a measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Operational feasibility reviews the willingness of the organization to support the

proposed system. This is probably the most difficult of the feasibility to gauge. In order to determine this feasibility, it is important to understand the management commitment tothe proposed project.

When we enable this device on the head, it will detect any hurdle or obstacle whichcomes in its path, and informs the direction to the user by ticking on the forehead and informs what type of obstacle is present through earplugs, so that they can take further action to avoid them.

3.3 BEHAVIORAL FEASIBILITY

Behavioral Feasibility includes how strong the reaction of users will be towards the development of a new system that involves an embedded system to be used in their daily life. So resistant to change is identified. It is a device which is fixed on the head, Visual challenge is a global problem among people and a large population is suffering due to this problem. The sense of dependency on someone else hurts a lot. Here we are buildinga technical solution to ease this pain.

3.4 ECONOMICAL FEASIBILITY

Smart headgear is economically very feasible, and accurate results because we are trying to make it at minimal cost so that the needy can afford it. It will use a minimal amount of power for its working which will be provided by a 9v battery or users can use other sources of power conveniently.

Arduino Uno (1)	Rs. 250
Raspberry Pi (1)	Rs. 2500
Ultrasonic sensor (1)	Rs. 90
Pi Camera Module & 9V Battery (1 & 1)	Rs. 560 + Rs. 100

Rs.3500

CHAPTER 4

FORM DESIGN

4.1 PROJECT VIEW (SCREENSHOT)

Fig-4.1 Project View

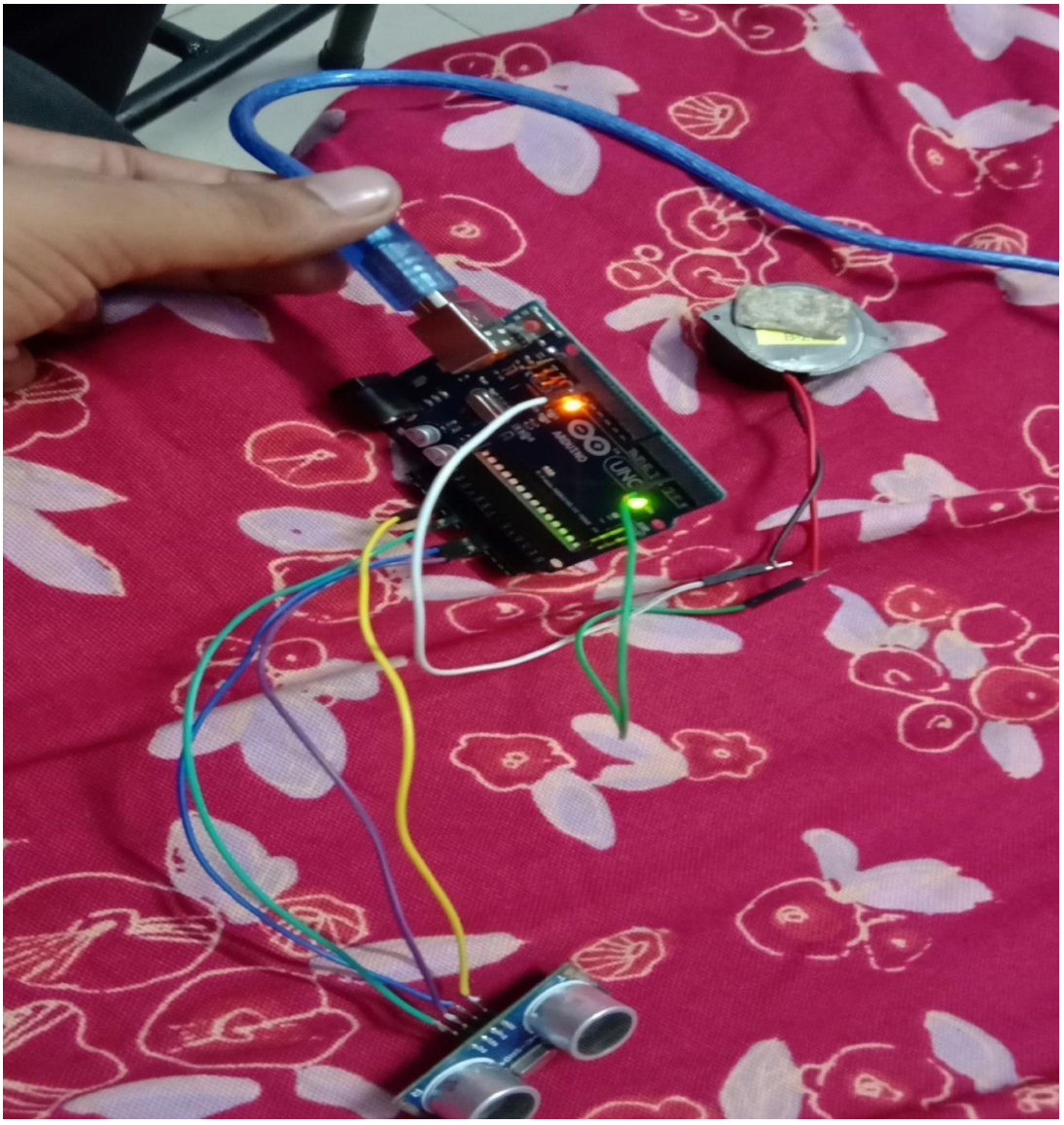


Fig-4.2 Arduino and Sensor View

CHAPTER 5

CODING

```
#define trigPin1 A0
#define echoPin1 A1
#define trigPin2 A2
#define echoPin2 A3
#define trigPin3 A4
#define echoPin3 A5
#define motor1 5
#define motor2 6
#define buzzer 3
void setup()
{
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(trigPin3, OUTPUT);
pinMode(echoPin3, INPUT);
pinMode(motor1, OUTPUT);
```

```
pinMode(motor2, OUTPUT);
pinMode(buzzer,OUTPUT);
}
void loop()
{
long duration, distance;
digitalWrite(trigPin1, LOW);
delayMicroseconds(2);
digitalWrite(trigPin1, HIGH);
delayMicroseconds(1);
digitalWrite(trigPin1, LOW);
duration = pulseIn(echoPin1, HIGH);
distance = (duration/2) / 29.1;
if (distance < 45)
{
digitalWrite(buzzer,HIGH);
```

```
} else

{

digitalWrite(buzzer,LOW);

} delay(50);

digitalWrite(trigPin2, LOW);

delayMicroseconds(2);

digitalWrite(trigPin2, HIGH);

delayMicroseconds(1);

digitalWrite(trigPin2, LOW);

duration = pulseIn(echoPin2, HIGH);

distance = (duration/2) / 29.1;

if (distance < 40)

{

digitalWrite(motor1,HIGH);

} else

{

digitalWrite(motor1,LOW);
```

```
} delay(100);

digitalWrite(trigPin3, LOW);

delayMicroseconds(2);

digitalWrite(trigPin3, HIGH);

delayMicroseconds(2);

digitalWrite(trigPin3, LOW);

duration = pulseIn(echoPin3, HIGH);

distance = (duration/2) / 29.1;

if (distance < 30)

{

digitalWrite(motor2,HIGH);

} else

{

digitalWrite(motor2,LOW);

} delay(100);

}
```

CHAPTER 6

TESTING

6.1 Software Testing

Software testing is defined as an activity to check whether the actual results matchthe expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

6.2 Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

6.3 Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

6.4 Types of testing

There are many types of testing like-

6.4.1 Functional Testing

Functional Testing verifies that each function of the software application operatesin conformance with the requirement specification. Functional testing is a quality assurance (QA) process and mainly involves Black Box Testing. It is concerned about theresults of processing and not about the source code of the application.

Definition: Functional Testing is a type of Software Testing whereby the system is tested against the functional requirements/specifications.

Functions (or features) are tested by providing appropriate input and examining the output. The actual results are then compared with expected results. Functional testing ensures that the requirements are properly satisfied by the application. The testing can bedone either manually or using automation.

Functional Testing is a type of software testing whereby the system is tested against the functional requirements/specifications.

Functions (or features) are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how processing occurs, but rather, with the results of processing. It simulates actual system usage but does not make any system structure assumptions.

During functional testing, Black Box Testing technique is used in which the internal logic of the system being tested is not known to the tester. Functional testing is normally performed during the levels of System Testing and Acceptance Testing. Typically, functional testing involves the following steps:

- Identify functions that the software is expected to perform.
- Create input data based on the function's specifications.
- Determine the output based on the function's specifications.
- Execute the test case.
- Compare the actual and expected outputs.

Functional testing is more effective when the test conditions are created directly from user/business requirements. When test conditions are created from the system documentation (system requirements/ design documents), the defects in that documentation will not be detected through testing and this may be the cause of end-users' wrath when they finally use the software.

6.4.2 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing.

Ideally, each test case is independent from the others. Substitutes such as methodstubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

Because some units may have references to other units, testing a single unit can frequently spill over into testing another unit. A common example of this is units that depend on a database: in order to test the unit, the tester often writes code that interacts with the database. This is a mistake, because a unit test should usually not go outside of its own unit boundary, and especially should not cross such process/network boundaries because this can introduce unacceptable performance problems to the unit test-suite. Crossing such unit boundaries turns unit tests into integration tests, and when such test cases fail, it may be unclear which component is causing the failure. Instead, the

software developer should create an abstract interface around the database queries, and then implement that interface with their own mock object. By abstracting this necessary attachment from the code (temporarily reducing the net effective coupling),an independent unit can be more thoroughly tested than may have been previously achieved. This results in a higher-quality unit that is also more maintainable.

Unit testing is commonly automated, but may still be performed manually. The IEEE does not favor one over the other. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by-step instructional document. However, automation is efficient for achieving this, and enables the many benefits listed in this article. Conversely, if not planned carefully, a careless manual unit test case may execute as an integration test case that involves many software components, and thus preclude the achievement of most if not all of the goals established for unit testing.

To fully realize the effect of isolation while using an automated approach, the unit or code body under test is executed within a framework outside of its natural environment. In other words, it is executed outside of the product or calling context for which it is intended. Testing in such an isolated manner reveals unnecessary dependencies between the code being tested and other units or data spaces in the product. These dependencies can then be eliminated.

Using an automation framework, the developer codes criteria, or a test oracle or result that is known to be good, into the test to verify the unit's correctness. During test case execution, the framework logs tests that fail any criterion. Many frameworks will also automatically flag these failed test cases and report them in a summary. Depending upon the severity of a failure, the framework may halt subsequent testing.

As a consequence, unit testing is traditionally a motivator for programmers to create decoupled and cohesive code bodies. This practice promotes healthy habits in software development. Software design patterns, unit testing, and code refactoring often work together so that the best solution may emerge.

Writing and maintaining unit tests can be made faster by using Parameterized Tests (PUTs). These allow the execution of one test multiple times with different input sets, thus reducing test code duplication. Unlike traditional unit tests, which are usually closed methods and test invariant conditions, PUTs take any set of parameters. PUTs have been supported by TestNG, JUnit and its .Net counterpart, XUnit. Suitable parameters forthe unit tests may be supplied manually or in some cases are automatically generated by the test framework. In recent years support was added for writing more powerful (unit) tests, leveraging the concept of Theory. A parameterized test uses the same execution steps with multiple input sets that are pre-defined. A theoryis a test case that executes thesame steps still, but inputs can be provided by a data generating method at run time.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of
code must satisfy. As a result, it affords several benefits.

Unit testing finds problems early in the development cycle. This includes both bugs in the programmer's implementation and flaws or missing parts of the specificationfor the unit. The process of writing a thorough set of tests forces the author to think through inputs, outputs, and error conditions, and thus more crisply define the unit's desired behavior. The cost of finding a bug before coding begins or when the code is first written is considerably lower than the cost of detecting, identifying, and correcting the bug later. Bugs in released code may also cause costly problems for the end-users of the software Code can be impossible or difficult to unit test if poorly written, thus unit testing can force developers to structure functions and objects in better ways.

In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a

bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, potential problems are caught early in the development process.

Unit testing allows the programmer to refactor code or upgrade system libraries ata later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API).

Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development.

When software is developed using a test-driven approach, the combination of writing the unit test to specify the interface plus the refactoring activities performed after the test has passed, may take the place of formal design. Each unit test can be seen as a design element specifying classes, methods, and observable behavior.

Testing will not catch every error in the program, because it cannot evaluate every execution path in any but the most trivial programs. This problem is a superset of the

halting problem, which is undecidable. The same is true for unit testing. Additionally, unit testing by definition only tests the functionality of the units themselves. Therefore, it will not catch integration errors or broader system-level errors (such as functions performed across multiple units, or non-functional test areas such as performance). Unit testing should be done in conjunction with other software testing activities, as they can only show the presence or absence of particular errors; they cannot prove a complete absence of errors. To guarantee correct behavior for every execution path and every possible input, and ensure the absence of errors, other techniques are required, namely the application of formal methods to proving that a software component has no unexpected behavior.

An elaborate hierarchy of unit tests does not equal integration testing. Integration with peripheral units should be included in integration tests, but not in unit tests. Integration testing typically still relies heavily on humans testing manually; high-level or global-scope testingcan be difficult to automate, such that manual testing often appearsfaster and cheaper.

Software testing is a combinatorial problem. For example, every Boolean decision statement requires at least two tests: one with an outcome of "true" and one with an outcome of "false". As a result, for every line of code written, programmers often need 3to 5 lines of test code. This obviously takes time and its investment may not be worth the effort. There are problems that cannot easily be tested at all – for example those that are non-deterministic or involve multiple threads. In addition, code for a unit test is likelyto be at least as buggy as the code it is testing. Fred Brooks in The Mythical Man-Month. Never go to sea with two chronometers; take one or three. Meaning, if two chronometers contradict, how do you know which one is correct?

Another challenge related to writing the unit tests is the difficulty of setting up realistic and useful tests. It is necessary to create relevant initial conditions so the part of the application being tested behaves like part of the complete system. If these initial conditions are not set correctly, the test will not be exercising the code in a realistic context, which diminishes the value and accuracy of unit test results.

To obtain the intended benefits from unit testing, rigorous discipline is needed throughout the software development process. It is essential to keep careful records not only of the tests that have been performed, but also of all changes that have been madeto the source code of this or any other unit in the software. Use of a version controlsystem is essential. If a later version of the unit fails a particular test that it had previously passed, the version-control software can provide a list of the source codechanges (if any) that have been applied to the unit since that time.

It is also essential to implement a sustainable process for ensuring that test case failures are reviewed regularly and addressed immediately. If such a process is not implemented and ingrained into the team's workflow, the application will evolve out of sync with the unit test suite, increasing false positives and reducing the effectiveness of the test suite.

Unit testing embedded system software presents a unique challenge: Because the software is being developed on a different platform than the one it will eventually run on, you cannot readily run a test program in the actual deployment environment, as is possiblewith desktop programs.

Unit tests tend to be easiest when a method has input parameters and some output.It is not as easy to create unit tests when a major function of the method is to interact with something external to the application. For example, a method that willwork with a database might require a mock up of database interactions to be created, which probably won't be as comprehensive as the real database interactions.

6.4.3 Smoke Testing

Smoke Testing is performed after software build to ascertain that the critical functionalities of the program are working fine. It is executed "before" any detailed functional or regression tests are executed on the software build. The purpose is to reject a badly broken application, so that the QA team does not waste time installing and testingthe software application. It is a part of Functional Testing. It can be performed by the developers or testers. Sometimes developers also test the application after the

deployment, before handing it over to QA to test. Or sometimes QA team Smoke Test the application before accepting any new build from the developers to avoid any time wastage.

Smoke testing covers most of the major functions of the software but none of them in depth. The result of this test is used to decide whether to proceed with further testing. If the smoke test passes, go ahead with further testing. If it fails, halt further tests and ask for a new build with the required fixes. If an application is badly broken, detailed testing might be a waste of time and effort.

Smoke test helps in exposing integration and major problems early in the cycle. It can be conducted on both newly created software and enhanced software. Smoke test is performed manually or with the help of automation tools/scripts. If builds are prepared frequently, it is best to automate smoke testing.

As and when an application becomes mature, with addition of more functionalitiesetc, the smoke test needs to be made more expansive. Sometimes, it takesjust one incorrect character in the code to render an entire application useless.

6.4.4 Sanity Testing

The terminologies such as Smoke Test or Build Verification Test or Basic Acceptance Test or Sanity Test are interchangeably used, however, each one of them is used under a slightly different scenario. It is a part of Functional Testing.

Sanity Testing is a software testing technique performed by the test team for somebasic tests. Whenever a new build is received, after minor changes in code or functionality, Sanity testing is performed to ascertain that the bugs have been fixed.

Sanity Testing is done after thorough regression testing is over. It is done to make sure that any defect fixes or changes after regression testing does not break the core functionality of the product. The goal is to determine that the proposed functionality works roughly as expected. It follows a narrow and deep approach and concentrates on

detailed testing of some limited and main features.

Sanity Test determines whether it is reasonable to proceed with further testing.
Sanity tests are mostly non scripted.

6.4.5 Regression Testing

Regression testing is the retesting of a software system to confirm that changes made to few parts of the codes has not any side effects on existing system functionalities. It is to ensure that old codes are still working as they were before introduction of the new change. The ideal process would be to create an extensive test suite and run it after each and every change.

ISTQB Definition: Regression testing is a type of software testing that intends to ensure that changes (enhancements or defect fixes) to the software have not adversely affected it.

As software is updated or changed, or reused on a modified target, emergence of new faults and/or re-emergence of old faults is quite common. Sometimes re-emergence occurs because a fix gets lost through poor revision control practices (or simple human error in revision control). Often, a fix for a problem will be "fragile" in that it fixes the problem in the narrow case where it was first observed but not in more general cases which may arise over the lifetime of the software. Frequently, a fix for a problem in one area inadvertently causes a software bug in another area. Finally, it may happen that, when some feature is redesigned, some of the same mistakes that were made in the original implementation of the feature are made in the redesign.

Therefore, in most software development situations, it is considered good coding practice, when a bug is located and fixed, to record a test that exposes the bug andre-run that test regularly after subsequent changes to the program. Although this may be done through manual testing procedures using programming techniques, it is often done using automated testing tools. Such a test suite contains software tools that allow the testing environment to execute all the regression test cases automatically; some projects even

set up automated systems to re-run all regression tests at specified intervals and report any failures (which could imply a regression or an out-of-date test). Common strategies are torun such a system after every successful compile (for small projects), every night, or oncea week. Those strategies can be automated by an external tool.

Regression testing is an integral part of the extreme programming software development method. In this method, design documents are replaced by extensive, repeatable, and automated testing of the entire software package throughout each stage ofthe software development process. Regression testing is done after functional testing hasconcluded, to verify that the other functionalities are working.

In the corporate world, regression testing has traditionally been performed by a software quality assurance team after the development team has completed work. However, defects found at this stage are the most costly to fix. This problem is being addressed by the rise of unit testing. Although developers have always written test cases as part of the development cycle, these test cases have generally been either functional tests or unit tests that verify only intended outcomes. Developer testing compels a developer to focus on unit testing and to include both positive and negative test cases.

6.4.5.1 When to perform Regression Testing

We do software regression testing whenever the production code is modified. Usually, we do execute regression tests in the following cases: Some regression testing examples are here.

- When new functionalities are added to the application.

Example: A website has a login functionality which allows users to do login only with Email. Now the new features look like "providing a new feature to do login using Facebook".

- When there is a Change Requirement (In organizations, we call it as CR).

Example: Remember password should be removed from the login page which is available earlier.

- When there is a Defect Fix.

Example: Imagine Login button is not working in a login page and a tester reports a bug stating that the login button is broken. Once the bug is fixed by the developers, testers test it to make sure whether the Login button is working as per the expected result. Simultaneously testers test other functionalities which are related to login button

- When there is a Performance Issue Fix.

Example: Loading the home page takes 5 seconds. Reducing the load time to 2seconds.

- When there is an Environment change.

Example: Updating the Database from MySQL to Oracle)

So far we have seen what regression is and when do we do regression Now let'ssee how we do regression testing.

6.4.5.2 How to select Test for Regression Test Suite

- Include the test cases which have frequent or recent defects.

- Include the test cases which verify core features of the product.

- Include the test cases for Functionalities which have undergone more and recentchanges.

- Include set Integration test cases which touch the entire major component.

- Include all Complex Test Cases.
- Include P1 & P2 test cases for regression testing.

6.4.5.3 Steps to take in Regression Testing

- Identify the need and components- Anytime a modification is

- made to code within a project, regression testing should be implemented. Modified pieces of code canbe checked using existing tests to determine if the changes broke anything that was previously functioning properly, and by writing new tests when needed. It's important to test fixed bugs and to watch for side effects of such fixes. Be aware,while a bug is being fixed, it's possible for another one to be introduced.

- Set requirements for testing Collaborations- between business stakeholders, developers, and software testing engineers should identify the appropriate test/use cases for the project or organization at hand. Agreeing upon and creating this list of cases provides a framework for future cost (and time) savings for all parties involved. By doing so, the test cases put in place can be leveraged down the road and used many times.

- Develop entry criteria for regression testing-To effectively determine when to start testing, entry criteria need to be put into place relating to the project. Entry criteria are the minimum eligibility or minimum set of conditions that should be met before starting testing work. Before regression testing, the engineer should: Confirm that the defect is repeatable and has been properly documented. Open a defect tracking record to identify and track the regression testing efforts. Createa regression test that's specific to the defect, and get the test reviewed and acceptedby project managers

- Develop exit criteria for regression testing-Similar to entry criteria, exit criteria should be developed to set the minimum eligibility or minimum conditions that need to be met before the testing phase is closed. These elements are agreed uponduring the test-planning phase and signed off prior to product release. During this process, software engineers should: Make sure all tests have been run. Ensure code coverage requirement levels have been met. Leave no severebugs untested or unfixed. Ensure all "high-risk" areas have been fully tested Follow a schedule-When the above steps have been completed, and the components that need regression testing have been identified, it's time to create a testing schedule. Be sure to identify when initial unit tests for the targeted components should becompleted (within one hour, one day, two days, 5 days, etc.). Regression tests can also bescheduled to automatically run at a specific time for more controlled, thorough testing ofthe

applications. While this is just a short list of steps that should be followed when implementing a regression testing strategy, we feel they are critical in maintaining maximum code coverage for frequently changed components. VectorCAST/Manage is a valuable tool for obtaining the regression testing results of previously developed unit and integration test environments, including code coverage data aggregation, helping organizations achieve a valuable continuous integration process. Failing to test and analyze software after changes have been made can result in defects and put lives at risk,especially in safety-critical applications.

6.5 TEST CASES

Test Case 1

In the first test case the range of the ultrasonic sensor was fixed to 2cm, so it was able to detect obstacles in the range of 2cm, but it was not a good range because obstacles can fallat any distance more than 2cm.

Test Case 2

In this test case, the range of the ultrasonic sensor was fixed to 20cm, it was able to detectobstacles in the range of 20cm, it can even detect small and very near obstacles also but 20cm was not an ideal distance for any obstacle detecting system.

Test Case 3

In the third test case, the range of the ultrasonic sensor was fixed to 1m, it was able to detect obstacles in the range of 1m, but 1m was not a good range because of the same reason we previously faced.

Test Case 4

In this test case, the range of the ultrasonic sensor was fixed to 2m, it was able to detect obstacles in the range of 2m, it can even detect small and very near obstacles also.

Test Case 5

In this test case, the range of the ultrasonic sensor was fixed to 3m, it was able to detect obstacles in the range of 3m, it can even detect small and very near obstacles also but

3m is not an ideal distance for anyone to take action for anything that comes in their way.

Test Case 6

In this test case, the range of the ultrasonic sensor was fixed to 5m, it was able to detect obstacles in the range of 5m, it can detect all obstacles in a range of 5m and provide accurate results but again the same issue is encountered as a previous test case.

Test Case 7

In this test case, the range of the ultrasonic sensor was fixed to 7m which is the maximumrange of Ultrasonic Sensor, it was able to detect obstacles in the range of 7m, it was not able to detect far and small obstacles properly.

REFERENCES

[1] Abdenour Hadid, Eric Granger, Xiaoyi Feng, Xiaoyue Jiang, Yanwei Pang, Deep *Learning inObject Detection*

[2] Volker Ziemann, *A Hands-On Course in Sensors Using the Arduino and Raspberry Pi*

[3] Anita Gehlot, Bhupendra Singh, Rajesh Singh, Sushabhan Choudhury, *Arduino-Based Embedded Systems*

[4] Atta ur Rehman Khan, Qusay F. Hassan, Sajjad A. Madani I*nternet of Things Challenges, Advances, and Applications*

[5] Samridhi Sajwan, Shabana Urooj, Manoj Kumar Singh, *Design and Implementation of Unauthorized Object and Entity Detector with Arduino Uno.*

[6] Xingyu Fang, Han Cao, Qindog Sun, *An Multi Feature Object Detection System IoT Devices.*

[7] Jin Diao, Deng Zhao, Jine Tang, Zehui Cheng, Zhangbing Zhou, *Continuous Object Detection System in IoT sensing networks.*

[8] Tanisha Dey Roy, Jaiteg Singh, *A note on wired and wireless sensor communication usingArduino Board.*

[9] Tianhong Pan, Tian Zhu, *Using sensors with the arduino..*

[10] Jeff Cicolani, *Raspberry Pi and Arduino*

[11] Raghvendra Kumar, Brojo Kishore Mishra, Prasant Kumar Pattnaik.

[12] Pavol Bistak, Mikulas Huba, Zdenek Slanina, Arduino Support for P*ersonalized Learning Of Control Theory Basics.*

[13] Shashi Rekha, Lingala Thirupathi, Sreekanth Renikunta, Rekha Gangula, *Study of security issues and solutions in IoT.*

[14] Yang LuSavvas Papagiannidis, Eleftherios Alamanos, *The Spill-Over Effect of the Emotional Reaction to the Use of Internet on the Intention to Use Internet of Things (IoT) Services.*

[15] Syed Tariq Anwar, *Internet of Things (IoTs) and Marketing: Conceptual Issues, Applications, and a Survey*