MEDICINE REMINDER AND MONITORING SYSTEM

Main Project Report

Submitted by

NAINA VINCENT

Reg No:FIT20MCA-2079

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications

Of

A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

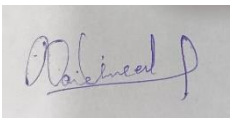
ANGAMALY-683577, ERNAKULAM(DIST)

JULY 2022

DECLARATION

I, **Naina vincent**, hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.



Date :

Place: Angamaly **Naina Vincent**

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

(An ISO 9001:2015 Certified, NAAC ('A' Grade) Accredited Institution with NBA Accredited Programmes
(Approved by AICTE – Affiliated to APJ Abdul Kalam Technological University, Kerala)
Owned and Managed by Federal Bank Officers' Association Educational Society
HORMIS NAGAR, MOOKKANNOOR P.O., ANGAMALY - 683 577, ERNAKULAM DT., KERALA, S. INDIA.
Tel: (O) 0484-2725272 Fax: 0484 – 2725250 E-mail: mail@fisat.ac.in Website: www.fisat.ac.in

11TH July 2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr./Ms. NAINA VINCENT (Reg.No.FIT20MCA-2079) has successfully completed his/her Main Project with the title " IOT Based Medicine Reminder and Monitoring System ", in the Department of Computer Applications, FISAT, during the period from 30th March 2022 to 11th July 2022.

Dr DEEPA MARY MATHEWS

HEAD OF THE DEPARTMENT

FEDERAL INSTITUTE OF SCIENCE & TECHNOLOGY (FISAT) ANGAMALY
DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY, ERNAKULAM-683577

Focus on Excellence

CERTIFICATE

This is to certify that the project report titled **"MEDICINE REMINDER AND MONITORING SYSTEM USING FACE DETECTION AND IOT"** submitted by **Naina Vincent, (Reg No: FIT20MCA-2079**) towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by her during the year 2022.

8/8/22

Project Guide

Dr. Sujesh P. Lal

FEDERAL INSTITUTE OF SCIENCE & TECHNOLOGY (FISAT) ANGAMALY
DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

Head of the Department

Dr. Deepa Mary Mathews

Submitted for the viva-voice held on ..... at .....

Examiner :

ACKNOWLEDGEMENT

I am extremely glad to present my main project which i did as a part of our curriculum. I take this opportunity to express my sincere thanks to those who helped me in bringing out the report of my project.

We are deeply grateful to Dr. Manoj George, Principal, FISAT, Angamaly and Dr. C. Sheela, Vice Principal, FISAT, Angamaly. Our sincere thanks to Dr. Deepa Mary Mathews, Head of the Department of Computer Applications, FISAT, who had been a source of inspiration. During the period of our project work, we have received generous help from Dr. Sujesh P. Lal, my project guide and Ms. Manju Joy ,my scrum master, which i like to put on record here with deep gratitude and great pleasure.

I express my heartfelt thanks to all the faculty members in our department for their constant encouragement and never ending support throughout the project.

Finally i am grateful to all our friends who gave us a lot of suggestions for the successful completion of this project.

Contents

1 INTRODUCTION **3**

2 PROOF OF CONCEPT **5**

3 IMPLEMENTATION **6**

3.1 Architecture 7

3.2 HARDWARE REQUIREMENTS 8

3.2.1 Arduino Uno 8

3.2.2 Servo Motors 9

3.2.3 LCD Display 9

3.2.4 Arduino buzzer 10

3.2.5 Jumper Wires 11

3.2.6 Voltage Regulator 11

3.2.7 Camera 12

3.2.8 NodeMCU 12

3.3 SOFTWARE REQUIREMENTS 13

3.3.1 Arduino IDE 13

3.3.2 Features of Arduino IDE 14

3.3.3 Pycharm 14

3.3.4 Android 15

3.3.5 FireBase 15

3.3.6 C (Programming language) 16

3.3.7 Features of C 17

3.3.8 Python 18
3.3.9 Computer Vision 19
3.4 Modules 21
3.4.1 IoT based application 21
3.4.2 Medicine dispatching 21
3.4.3 User authentication 22

4 RESULT ANALYSIS 23
4.1 Connection Diagram 23
4.2 Connection Diagram of System 24
4.3 Testing 24
4.3.1 Test Cases 24
4.3.2 Functionality Testing 25

5 SCOPE AND FUTURE ENHANCEMENT 26

6 APPENDIX 27
6.1 Source Code 27
6.1.1 Arduino.ino 27
6.1.2 faceRecognition.py 42
6.2 SCREENSHOTS 45
6.2.1 UI Design 45
6.2.2 Device Images 47

7 References 50

ABSTRACT

There are several challenges that old people face, and of them is taking their medicines on time. Old people usually forget to take their medication on time and also have a hard time recollecting whether they had their medication, which sometimes could lead to overdose and severe medical complications. There are several expensive medicine dispensers available in the market now. However, most of the elderly people around the world don't even know of such products and still resort to storing the medicines in a box. Several types of medicine dispensers are available commercially worldwide. However, they have several drawbacks that requires to be resolved. These drawbacks can be resolved using a Automatic Medicine Dispenser that is reliable,affordable and can carry up to 2-3 weeks of medicines, in such a way that, old people wont need to depend on someone else. The product is designed to make sure that the quantity and timing of the pills to be dispensed can be controlled using an app and monitoring using database, which makes things easier for everyone, including for children who work abroad. Also, it offers clear contact between the consumer and parental figures as it will immediately notify the guardian in case the patient has missed medicine intake.

Chapter 1

INTRODUCTION

1.1 Background

Studies show that many people irrespective of age become forgetful when it comes to consuming medicines. This, seen mostly in the elderly, leads to many other life risks. This is where medicine dispensers come into play. There are different types of dispensers all of which are just implemented using solid medicines. Most of them are jut simple with no reminders etc. This paper proposes a dispenser that can dispense both solid and liquid medicine, has an app integrated with it and helps to set alarms.

One of the biggest problems in the health care industry is medication adherence. Often elderly people fail to take their medication on time and for elders with more than one medication, the chances of overdosing are considerably high. This could easily lead to catastrophic events such as permanent disability or even death. Hence, it is evident that it is a widespread problem and needs a solution. It was revealed that, in a study conducted at the University of Washington, 30.7 percent of participants disliked and 18.3 percent liked at least one drug.

1.1.1 Motivation

In modern society, most of the time people remain busy in their daily life schedule. It is true that they give more preference to their work than taking care of their health. Several diseases like diabetes, blood pressure is nowadays very common. Maintaining daily medication become very difficult for old people. Sometimes younger is faced with the same problem. There are many people in our family who need constant help may it be our elderly people, younger or others. But it is not always possible for us to remind them of their medicine's dosages every time. For this purpose, there needs to be some facility for us which monitoring patient and take care.

Today in medical services frameworks, the usage of cell phones is turning into an expanding number of values . IoT may be helpful to monitor real-time condition and IoT can be a powerful and effective paradigm to store data collected by sensors devices to the cloud. In our project, the IoT enabled device will control the overall monitoring system. And developed an android application which help patients by reminding medicine in take time and so on.

Patients may often fail to comply with their medication whether it was from forgetting to take the medicine, from taking medicine at the wrong time or even from taking too much medicine. Therefore, there are many systems such as reminder, alarm, and so on to remind patient. We have focus on those patients who having difficulty to take medication on time, we tried to design and to aid patients with managing their medical prescriptions, through a reminder app they will use to look at and manage their medications. The Pill Reminder will facilitate users to require the right medication on time. This system provides a real time monitoring system that allow related people to monitor the patient's activity remotely.

Chapter 2

PROOF OF CONCEPT

Objectives

There are many people around us who are the victims of chronic disease. Most of them suffering from dementia. Some people overlook to take care of health. Because of the lack of an expert system, people are forced to submit in frequent health related problems. By analyzing the data, an internet of things (IoT) based reminder system has been developed. It is designed to assist the patient who forgets to take medicine. The proposed system consists of an IoT enabled device and an android application. It mainly focuses on dementia patient. But it is beneficial for all. Patients will no longer have to worry about daily medication. The application will send a notification when it's time to take medicine. The mobile application is used for keeping the record in medicine details and reminding the schedule of medicine. We have used the IoT enabled Arduino device for monitoring the whole system.

Chapter 3

IMPLEMENTATION

Here we introduce a smart medicine reminder system based on IoT. The proposed scheme was particularly created for the Android platform. For our system, we implement a reminder system which provides an alarm when it is time for taking medicine. Along with that, there is an android application where the user can set their medicine time. In the application, there will some feature that help the user to know more details about their medicine. It keeps track for the medicine which means how much medicine they have to take they can be fixed in the application. The device setup consists of an IoT enabled pill box having compartments, each having a lid to open, and an servo motors are attached to it. Whenever the medication is loaded into the pillbox it'll be updated the medicine data and saved in database. The Arduino device fetching real time data and send it to the application.

- Deciding system specifications
- Selection of system components
- Design of circuit diagram
- Manufacturing of PCB layout
- Component mounting soldering
- Design of enclosure or structure

3.1 Architecture

The pill dispenser loaded by the patient himself or by somebody helps the patient. The Entered details of medicine are kept within the cloud database and reminders are set. The system can read the details which saved in the database and generates the reminder to the user to require medication . The system provides alerts once it's time to take medication. These details are regularly updated automatically from the cloud.

1. The lid opens only when the system recognise the face and takes the pills, the servo motors connected to the lid. Lids are open and therefore will send the output to Arduino which is able to stop the reminder. This can be recorded that the patient has taken his medication with success.

2. In case the patient fail while recognizing the face refuses required drugs, the lid won't open and therefore the reminder can automatically stop after a pre-set time and can be placed on snooze. If someone once more misses the drugs, the output can be sent to the database and send a message to the patient reminding him that he has missed a dose on the screen.

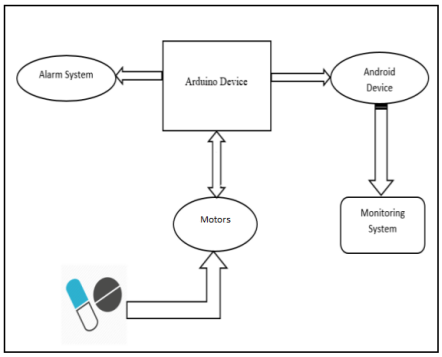


Figure 3.1: Block Diagram of Proposed System

3.2 HARDWARE REQUIREMENTS

3.2.1 Arduino Uno

- **Arduino Uno-ESP32:** ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica's 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth. The good thing about ESP32, like ESP8266 is its integrated RF components like Power Amplifier, Low-Noise Receive Amplifier, Antenna Switch, Filters and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components.

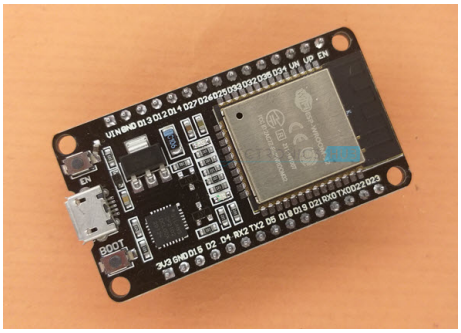


Figure 3.2: Arduino Uno

Microcontroller :ESP32

Operating Voltage : 5 Volts

Input Voltage :2.2 to 3.6Volts

Digital I/O Pins : 14 (of which 6 can provide PWM output)

UART: 3, I2C: 2, SPPI : 4

Analog Input Pins : 15

DC Current per I/O Pin : 3 mA or 15 mA

DC Current for 3.3V Pin : 50 mA

Flash Memory : 520 KB of which 0.5 KB used by bootloader

SRAM: 2 KB, EEPROM : 1 KB

3.2.2 Servo Motors

Servo Motor: An electrical rotary actuator which has the ability to precisely push an object by controlling the angular position and velocity defines a servo motor. It pairs the functions of a regular motor with a senor for position feedback. The motor runs through servo mechanism. It is made up three parts namely an output sensor, a controlled device and a feedback system. The feedback system is a closed loop that can control the shaft's movement and final position using a positive feedback. PWM signals that are generated from the control wires control the servo motor. It has a maximum and minimum pulse and its length determines how much the motor will turn. From its initial position, it can rotate 90 degrees towards any of the two directions. The angle of rotation of the servo depends on the length of the pulse of the Control PIN which is a significance of pulse width modulation principle . In short, it can deduced that it is made up of a DC motor which is controlled by several gears and a potentiometer.

Figure 3.3: Servo Motors

3.2.3 LCD Display

The JHD162A dot-matrix liquid crystal display controller and driver LSI displays alphanumeric, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver The JHD162A character generator ROM is extended to generate 208 5×8 dot character fonts and 325×10 dot char-

acter fonts for a total of 240 different character fonts. The low power supply (2.7V to 5.5V) of the JHD162A is suitable for any portable battery-driven product requiring low power dissipation.

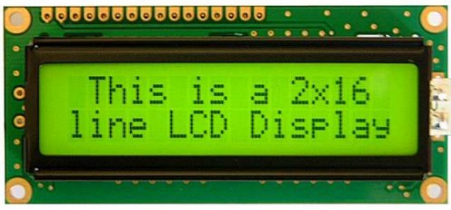


Figure 3.4: LCD Display

3.2.4 Arduino buzzer

An arduino buzzer is also called a piezo buzzer. It is basically a tiny speaker that you can connect directly to an Arduino. You can make it sound a tone at a frequency you set. The buzzer produces sound based on reverse of the piezoelectric effect.The buzzer produces the same noisy sound irrespective of the voltage variation applied to it. It consists of piezo crystals between two conductors. When a potential is applied across these crystals, they push on one conductor and pull on the other. This, push and pull action, results in a sound wave. Most buzzers produce sound in the range of 2 to 4 kHz.

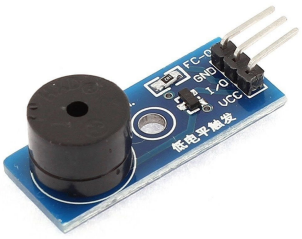


Figure 3.5: Arduino buzzer

3.2.5 Jumper Wires

A jump wire (also known as jumper, jumper wire, DuPont wire) 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. Jumper Wires are accustomed transmit electricity between 2 points in an exceedingly circuit in the main jumper wires are used to analyze defects among the circuit or accustomed change the circuits.

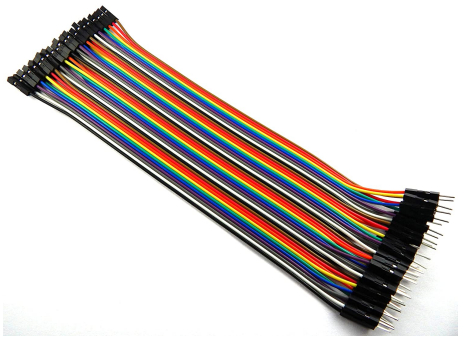


Figure 3.6: Arduino buzzer

3.2.6 Voltage Regulator

The Arduino Uno has a 5-volt linear voltage regulator on its circuit board, allowing you to use its coaxial power connector to connect a power supply of 8 to 20 volts DC. The regulator reduces this to the 5 volt DC level that the Arduino uses.

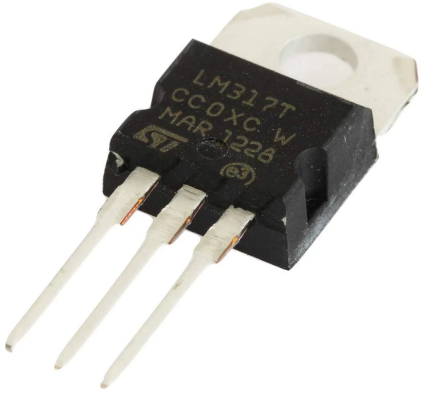


Figure 3.7: Voltage Regulator

3.2.7 Camera

facial recognition system uses biometrics to map facial features from a photograph or video. It compares the information with a database of known faces to find a match. Facial recognition can help verify a person's identity, but it also raises privacy issues.

Figure 3.8: Camera

3.2.8 NodeMCU

NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

The NodeMCU ESP8266 development board comes with the ESP-12E module containing the ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

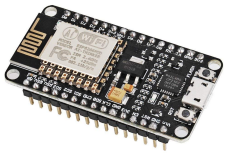


Figure 3.9: NodeMCU

3.3 SOFTWARE REQUIREMENTS

3.3.1 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, with the help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2.

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 two basic 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 tool chain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text.

The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port.

3.3.2 Features of Arduino IDE

- Multi-Platform Application Arduino IDE works on the three most popular operating systems: Windows, Mac OS, and Linux. Aside from that, the application is also accessible from the cloud. These options provide programmers with the choice of creating and saving their sketches on the cloud or building their programs locally and upload it directly to the board.

- Board Management Arduino IDE comes with a board management module, where users can select the board they want to work with at the moment. If they wish to change it, they can do so easily from the drop down menu. Modifying their selection also automatically updates the PORT infos with the data they need in relation to the new board.

- Straightforward Sketching With Arduino IDE, users can create programs called sketches that are built with a text editor. The process is a straightforward one though it has several bells

- The tool is armed with a board management module, wherein users can choose which board they want to use. If another board is needed, they can seamlessly select another option from the dropdown menu. PORT data is updated automatically whenever modifications are made on the board or if a new board is chosen.

3.3.3 Pycharm

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. o start developing in Python with PyCharm you need to download and install Python from python.org depending on your platform.

PyCharm supports the following versions of Python:

- Python 2: version 2.7

- Python 3: from the version 3.6 up to the version 3.11

Besides, in the Professional edition, one can develop Django , Flask, and Pyramid applications. Also, it fully supports HTML (including HTML5), CSS, JavaScript, and XML: these languages are bundled in the IDE via plugins and are switched on for you by default. Support for the other languages and frameworks can also be added via plugins (go to Settings — Plugins or PyCharm — Preferences — Plugins for macOS users, to find out more or set them up during the first IDE launch).

3.3.4 Android

The Android operating system is the most popular mobile operating system in the world. Any phone that runs Android is often referred to as an Android phone.

Android phones are classified as smartphones, mainly because they can do a lot more than just make calls or send and receive text messages. The Android operating system allows you to install a wide variety of apps, and essentially use your device as a mini-computer. droid OS is a Linux-based mobile operating system that primarily runs on smartphones and tablets.

The Android platform includes an operating system based upon the Linux kernel, a GUI, a web browser and end-user applications that can be downloaded. Although the initial demonstrations of Android featured a generic QWERTY smartphone and large VGA screen, the operating system was written to run on relatively inexpensive handsets with conventional numeric keypads.

3.3.5 FireBase

It is nothing but a set of tools that are used to create, develop and evolve your application. It takes care of a large portion of those processes and services that programmers would often create on their own, but wouldn't want to, because they often like to focus on the application. The cloud hosts the processes and services like databases, authentication etc. The client SDKs provided by Firebase communicate directly with these backend

services, without the need for establishing any middleware between the service and your application

You will only need to write code to query the information from your client's application database while using one of the Firebase database option. This is a stark change from the usually tried and tested app development methods which usually includes writing both the backend and frontend software. The frontend code invites API endpoints which the backend presents, and the job is done by the backend codes. However, with Firebase, the usual backend goes through, putting the work into the client. Firebase Authentication deals with getting your clients signed in and recognized. This product is fundamental to getting configuration done properly, particularly in the event that you have to limit access to per- client data (which is what each application would want to do). The special thing about Firebase Authentication is that it makes it simple to perform secure logins, which is amazingly hard to execute effectively all alone. Firebase Realtime Database and Cloud Firestore gives database services. Firebase brings in SDKs to use in your application to make direct data easily accessable, expelling the requirement for that annoying middleware segment.

What's extremely unique about these databases is that they give you "realtime" updates to information as it changes in the database. You utilize the customer SDK to set up a "listener" at the location of the information your application needs to utilize, and the "listener: gets summoned with that information more than once, every time a change is noticed. This lets you keep your application's presentation new and updated, without surveying the information of intrigue.

3.3.6 C (Programming language)

C was originally developed at Bell Labs by Dennis Ritchie, between 1972 and 1973. It was created to make utilities running on Unix. Later, it was applied to re-implementing the kernel of the Unix operating system.During the 1980s, C gradually gained popularity. Nowadays, it is one of the most widely used programming languages.

With C compilers from various vendors available for the majority of existing computer architectures and operating systems. C has been standardized by the American National Standards Institute (ANSI) since 1989 (see ANSI C) and subsequently by the International Organization for Standardization (ISO).

C is an imperative procedural language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run time support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming.

A standards compliant C program that is written with portability in mind can be compiled for a wide variety of computer platforms and operating systems with few changes to its source code; the language has become available on various platforms, from embedded micro-controllers to supercomputers.

3.3.7 Features of C

- It is a robust language with rich set of built-in functions and operators that can be used to write any complex program.
- The C compiler combines the capabilities of an assembly language with features of a high-level language.
- It is many time faster than BASIC.
- C is highly portable this means that programs once written can be run on another machines with little or no modification.
- Another important feature of C program, is its ability to extend itself.
- A C program is basically a collection of functions that are supported by C library. We can also create our own function and add it to C library.

- C language is the most widely used language in operating systems and embedded system development today.
- Unlike assembly language, c programs can be executed on different machines with some machine specific changes. Therefore, C is a machine independent language.

3.3.8 Python

BACK END :

- Python

  Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

  Python is Interpreted - Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

  Python is Interactive - You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

  Python is Object-Oriented - Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

  Python is a Beginners Language - Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games

3.3.9 Computer Vision

Computer vision is the field of computer science that focuses on creating digital systems that can process, analyze, and make sense of visual data (images or videos) in the same way that humans do. The concept of computer vision is based on teaching computers to process an image at a pixel level and understand it. Technically, machines attempt to retrieve visual information, handle it, and interpret results through special software algorithms.

Here are a few common tasks that computer vision systems can be used for:

- Object classification: The system parses visual content and classifies the object on a photo/video to the defined category. For example, the system can find a dog among all objects in the image.

- Object identification. The system parses visual content and identifies a particular object on a photo/video. For example, the system can find a specific dog among the dogs in the image.

- Object tracking. The system processes video finds the object (or objects) that match search criteria and track its movement.

  Computer vision algorithms that we use today are based on pattern recognition. We train computers on a massive amount of visual data—computers process images, label objects on them, and find patterns in those objects. For example, if we send a million images of flowers, the computer will analyze them, identify patterns that are similar to all flowers and, at the end of this process, will create a model "flower." As a result, the computer will be able to accurately detect whether a particular image is a flower every time we send them pictures.

Golan Levin, in his article Image Processing and Computer Vision, provides technical details about the process that machines follow in interpreting images. In short, machines interpret images as a series of pixels, each with their own set of color values. For example, below is a picture of Abraham Lincoln. Each pixel's brightness in this image is represented by a single 8-bit number, ranging from 0 (black) to 255 (white). These numbers are what software sees when you input an image. This data is provided as an input to the computer vision algorithm that will be responsible for further analysis and decision making.

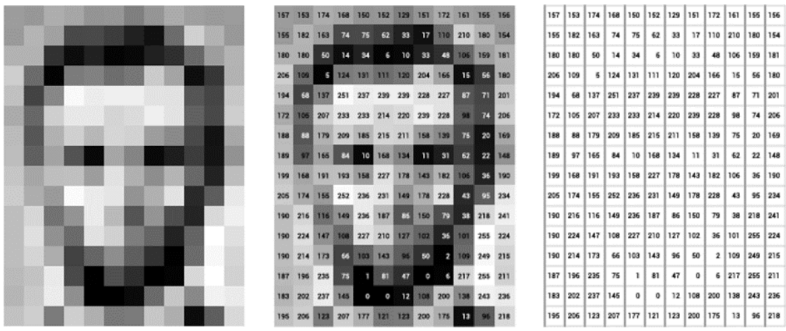


Figure 3.10: Computer Vision Demo

3.4 Modules

In this project,it divided into Three modules.

- Medicine Dispatching
- IoT based Pill Reminder and Monitoting app
- User authentication

3.4.1 IoT based application

By the help of MIT.App create an UI for the application. Reminder and Monitoring mobile application is work while connect with Nodemcu Wifi module.Admin can login, and update the medicine dosage via this application.The medicine will provide in three time schedules,morning, noon and night. We can set up to three medicine per time. The updated data will load to database as a encrypted single string.
NodeMCU

3.4.2 Medicine dispatching

User verification can be done by, verifying the face of the patient or by the bystander. After the user verification, user can able to take prescribed amount of medicine which is Displayed on the screen.The details will display in the LCD screen,When to take the medicine and how much amount of medicine should be taken. They can able to take medicine from the medicine container. The servo motor of the container will open automatically to the users. If the user isn't recognised by the system,the container lid won't open.

Microcontroller : ESP328 (Arduino uno)

Camera

LCD Display

Servo motor

3.4.3 User authentication

In this module verify the user by the face, after the identification the system will start Working .Medicine lid will open and can take the medicine from the container.

Microcontroller : ESP32 (Arduino Uno)

Verification : Camera

Servo motor

Buzzer

LCD Display

Chapter 4

RESULT ANALYSIS

The medicine dispense loaded by the patient himeself or by somebody helps the patient. The Entered details of medicine are kept within the cloud database and reminders are set. The system can read the details which saved in the database and generates the reminder to the user to require medication and it transmits a notice message to LCD screen. The system provides alerts once it's time to take medication. These details are regularly updated automatically from the cloud. If the system recognise the face,opens the lid by the servo motors and the patient can take the medicine, that the lid is opened and therefore will send the output to Arduino which is device stop the reminder. This can be recorded that the patient has taken his medication with success.

In case the patient fails to require the drugs or refuses to, the lid won't open and therefore the reminder can automatically stop after a pre-set time and can be placed on snooze. If someone once more misses the drugs, the output can be sent to the database.

4.1 Connection Diagram

The Circuit Diagram is the Graphical representation of the electrical Circuit. A pictorial circuit diagram uses simple images of components, while a schematic diagram shows the components and interconnections of the circuit using standardized symbolic representations. The presentation of the interconnections between circuit components in the

schematic diagram does not necessarily correspond to the physical arrangements in the finished device. Unlike a block diagram or layout diagram, a circuit diagram shows the actual electrical connections. A drawing meant to depict the physical arrangement of the wires and the components they connect is called artwork or layout, physical design, or wiring diagram. Circuit diagrams are used for the design (circuit design), construction (such as PCB layout), and maintenance of electrical and electronic equipment.

4.2 Connection Diagram of System

In this proposed design of IOT Based Medicine Reminder and Monitoring System, Buzzer, LCD display is connected to ESP32.User can update the medicine details on the mobile application and it will loaded to cloud. These details are regularly updated automatically from the cloud. Based on the updated time to take medicine the buzzer will start beeping. If the user's is recognised the lid of the box will open. The lids are connected to a servo motors which will help to open the lid. If face is recognised it will lid will open otherwise it won't open and therefore the reminder can automatically stop after a pre-set time and can be placed on snooze. If someone once more misses the drugs, the output can be sent to the database.

4.3 Testing

Hardware testing is the process of evaluating and verifying that a hardware product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

4.3.1 Test Cases

Testing is based on test cases. It describes which feature or service test attempts to cover. In test cases specify what you are testing and which particular feature it tests.

- Test the normal use of system

- Test the abnormal, but reasonable use of system
- Test the abnormal and reasonable use of system
- Test the boundary Conditions

4.3.2 Functionality Testing

Can be performed on hardware or software products to verify that your product functions exactly as designed. The general purpose of hardware and software functionality testing is to verify if the product performs as expected and documented, typically in technical or functional specifications. Developers creating a new product start from a functional specification, which describes the product's capabilities and limitations.

Chapter 5

SCOPE AND FUTURE ENHANCEMENT

We have demonstrated a mobile application that generates alarm signals to remind a patient to take medication. We focus on helps patients and improving the monitoring system. The application Medicare is easily accessible. Combination of a sensing system with android application helps us to measure how well a patient can take their daily medication in real-time. It allows real-time monitoring. Better compliance in terms of the taking of medicine can be acquired with the use of our proposed framework. This framework assures the security of the patient, prevent wrong dosages, support medication adherence. As a future work, we are wanting to improve our drug update framework by presenting extra highlights utilizing portable application and incorporate other medical services. A data-sharing feature between patient and health care professionals would also be developed. Voice-alert notification is being considered as part of the future works; a system that will not only send notification however also read the content of the notification alert to the listening of the patient.

Chapter 6

APPENDIX

6.1 Source Code

6.1.1 Arduino.ino

```
/*********************/
// #include <WiFi.h>
#include <time.h>
#include <LiquidCrystal.h>
#include <Servo.h>
LiquidCrystal lcd(13, 12, 14, 27, 26, 25);
Servo servo_1;
Servo servo_2;
Servo servo_3;
/***************************/
#if defined(ESP32)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>
```

```
#include <addons/TokenHelper.h>
#include <addons/RTDBHelper.h>
/***************************/
#define WIFI_SSID "GNXS-3D42F0"
#define WIFI_PASSWORD "BC62D23D42F0"
#define API_KEY "AIzaSyCcLrEC1ohgOj2TQSeH55cEAVEhp-WLh14"
#define DATABASE_URL "medcine-remainder-default-rtdb.firebaseio.com"
#define USER_EMAIL "projectmedicine21@gmail.com"
#define USER_PASSWORD "promed21"
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
void loop()
{
  Time();
  time_counter();
  /*sprintf(buffer,"Current Date and Time%d:%d:%d
  %d/%d/%d ",h,m,s,d,mo,y);
  Serial.println(buffer);*/
  delay(500);
}
    void Wifi_connect()
    {
      int i=0;
          Serial.println("");
          lcd.begin(16, 2);
          lcd.print("WiFi Connecting");
          lcd.setCursor(0,1);
          lcd.print("");
```

```
WiFi.disconnect();
WiFi.begin(WIFI_SSID,WIFI_PASSWORD);
while ((!(WiFi.status() == WL_CONNECTED)))
{
  i++;
  delay(300);
  Serial.print("*");
  lcd.print("*");
  if(i>15)
  {
    lcd.begin(16,2);
    lcd.print("WiFi Connecting");
    lcd.setCursor(0,1);
    i=0;
  }
}
lcd.begin(16,2);
lcd.print("CONNECTED");
Serial.println("\n\nConnected..");
Serial.println();
Serial.print("Connected with IP: ");
Serial.println(WiFi.localIP());
Serial.println();
Serial.printf("Firebase Client v%s\n\n",
FIREBASE_CLIENT_VERSION);
config.api_key = API_KEY;
auth.user.email = USER_EMAIL;
auth.user.password = USER_PASSWORD;
config.database_url = DATABASE_URL;
```

```
        config.token_status_callback = tokenStatusCallback;
        #if defined(ESP8266)
        fbdo.setBSSLBufferSize(2048 /* Rx buffer size in bytes from
        512 - 16384 /, 2048 / Tx buffer size in bytes from 512 - 16
        #endif
        Firebase.begin(&config, &auth);
        Firebase.reconnectWiFi(true);
        Firebase.setDoubleDigits(5);
        config.timeout.serverResponse = 10 * 1000;
        Serial.println("Getting Time zone info please wait.. \n");
        configTime(timezone, dst, "pool.ntp.org","time.nist.gov");
        delay(100);
            while(!time(nullptr))
            {
                Serial.print("*");delay(100);
            }
       Serial.println("\nTime response....OK\n");
       delay(800);
    }
   void FireBaseGet_Dat()
   {
      M_Time = (Firebase.RTDB.getString(&fbdo, F("/Naina/Time"))
      ? fbdo.to<const char *>() : fbdo.errorReason().c_str());
      M_Med  = (Firebase.RTDB.getString(&fbdo, F("/Naina/MED"))
      ? fbdo.to<const char *>() : fbdo.errorReason().c_str());
          Serial.println(c_n);
int Extract_msg(String msg,char a)
{
  int num=0,i=0;
```

```
  for(i=0;msg[i]!='\0';i++)
  {

    if(msg[i]==a)
    {
      i++;
      while(msg[i] >=48 && msg[i] <= 57)
      {
        num = num*10 +(msg[i]-48);
        i++;
      }
      return num;
    }
  }
}
   int calc_Time(int hh,int mm)
   {
      int h_c=0,m_c=111;

      h_c=hh-h;
      if(h_c ==0)
      {
        m_c = mm-m;
      }
      else if(h_c == 1)
      {
        m_c=(60-m)+mm;
      }
      return m_c;
```

```
    }
void time_counter()
{
  pt=millis()/100;
  if(pt-ot > 15)
  {
    ot=pt;
    dt++;
    if(m_d >=-5 && m_d <= 5 && mrng==0)
    {
        lcd.begin(16, 2);
        lcd.setCursor(0,0);
        lcd.print("Morning Pill");
        lcd.setCursor(0,1);
        lcd.print("Alert");
        m_pill=1;
        check_face();
            if(p==0)
            {
              lcd.setCursor(8,1);
              lcd.print("Waiting");
              delay(500);
            }
            else if(p==1)
            {
              lcd.setCursor(8,1);
              lcd.print("Naina");
              delay(500);
              open_morning();
```

```
            }
            else  if(p==2)
            {
               lcd.setCursor(8,1);
               lcd.print("Unknown");
               delay(500);
            }
            digitalWrite(23,HIGH);
            delay(100);
            digitalWrite(23,LOW);
}
}
else  if(n_d >=-5 && n_d <= 5 && night==0 )
{
     lcd.begin(16, 2);
     lcd.setCursor(0,0);
     lcd.print("night  Pill");
     lcd.setCursor(0,1);
     lcd.print("Alert");
     n_pill=1;
     check_face();
         if(p==0)
            {
               lcd.setCursor(8,1);
               lcd.print("Waiting");
               delay(500);
            }
            else  if(p==1)
            {
```

```
            lcd.setCursor(8,1);
            lcd.print("Naina");
            delay(500);
            open_night();
          }
          else if(p==2)
          {
            lcd.setCursor(8,1);
            lcd.print("Unknown");
            delay(500);
          }
          digitalWrite(23,HIGH);
          delay(100);
          digitalWrite(23,LOW);
}
else if(dt==1)
{
  int y1;
    lcd.begin(16, 2);
    lcd.setCursor(0,0);
    lcd.print(h);
    lcd.print(":");
    lcd.print(m);
    lcd.setCursor(8,0);
    if(d<10)
      lcd.print("0");
    lcd.print(d);
    lcd.print("/");
    if(mo<10)
```

```
        lcd.print("0");
      lcd.print(mo);
      lcd.print("/");
      y1=y%100;
      lcd.print(y1);
      lcd.setCursor(0,1);
      lcd.print("Have a nice day");
}
      if(m_pill ==1)
      {
        Serial.println("Morning SKPED");
        String tag ="";
        char buffer[16];
        tag += "test/MSG/";
        sprintf(buffer,"%d_%d_%d_%d_%d_%d",h,m,s,d,mo,y);
        tag += buffer;
        Firebase.RTDB.setString(&fbdo, tag, "Morning_Skipped");
        delay(500);
        m_pill =0;
      }
if(dt==4)
{
  p=5;
  lcd.begin(16, 2);
  digitalWrite(23,LOW);
  lcd.print("Getting Info...");
  FireBaseGet_Dat();
  dt=0;
}
```

```
  }
}
void check_face()
{
 char x;
  Serial.println("a");
  while(Serial.available())
  {
    x=Serial.read();
    if(x=='p')
    {
      p=Serial.parseInt();
      Serial.print("Persion count >> ");
      Serial.println(p);
    }
  }
}
/********************/
void open_morning()
{

          lcd.begin(16, 2);
           m_pill=0;
          if(a_m >0)
          {
            lcd.setCursor(0,0);
            lcd.print("A : ");
            lcd.print(a_m);
            servo_1.write(OPEN);
```

```
}
else
{
  lcd.setCursor(0,0);
  lcd.print("A : ");
  lcd.print(a_m);
  servo_1.write(CLOSE);
}

if(b_m >0)
{
  lcd.setCursor(8,0);
  lcd.print("B : ");
  lcd.print(b_m);
  servo_2.write(OPEN);
}
else
{
  lcd.setCursor(8,0);
  lcd.print("B : ");
  lcd.print(b_m);
  servo_2.write(CLOSE);
}
if(c_m >0)
{
  lcd.setCursor(0,1);
  lcd.print("C : ");
  lcd.print(c_m);
  servo_3.write(OPEN);
```

```
}
else
{
  lcd.setCursor(0,1);
  lcd.print("C : ");
  lcd.print(c_m);
  servo_3.write(CLOSE);
}
Serial.print("Morning_Pill_OK");
String tag ="";
  char buffer[16];
  tag += "test/MSG/";
  sprintf(buffer,"%d_%d_%d_%d_%d_%d",h,m,s,d,mo,y);
  tag += buffer;
  Firebase.RTDB.setString(&fbdo, tag, "Morning_OK");
  delay(500);
delay(1000);
lcd.setCursor(14,1);
lcd.print("1  ");
delay(1000);
lcd.setCursor(14,1);
lcd.print("2  ");
delay(1000);
lcd.setCursor(14,1);
lcd.print("3  ");
delay(1000);
lcd.setCursor(14,1);
lcd.print("4  ");
delay(1000);
```

```
        lcd.setCursor(14,1);
        lcd.print("5   ");
        delay(1000);
        lcd.setCursor(14,1);
        lcd.print("6   ");

        lcd.begin(16, 2);
        lcd.print("Closing");
        delay(2000);

    servo_1.write(CLOSE);
    servo_2.write(CLOSE);
    servo_3.write(CLOSE);
    mrng=1;
    noon=0;
    night=0;
}
/********************/
void check_face()
{
 char x;
  Serial.println("a");
  while(Serial.available())
  {
    x=Serial.read();
    if(x=='p')
    {
      p=Serial.parseInt();
      Serial.print("Persion count >> ");
```

```
        Serial.println(p);
      }
    }
  }
```

```
          Serial.print("Morning_Pill_OK");
          String tag ="";
            char buffer[16];
            tag += "test/MSG/";
            sprintf(buffer,"%d_%d_%d_%d_%d_%d",h,m,s,d,mo,y);
            tag += buffer;
            Firebase.RTDB.setString(&fbdo, tag, "Morning_OK");
            delay(500);
          delay(1000);
      servo_1.write(CLOSE);
      servo_2.write(CLOSE);
      servo_3.write(CLOSE);
      mrng=1;
      noon=0;
      night=0;
}
```

```
            String tag ="";
            char buffer[16];
            tag += "test/MSG/";
            sprintf(buffer,"%d_%d_%d_%d_%d_%d",h,m,s,d,mo,y);
            tag += buffer;
            Firebase.RTDB.setString(&fbdo, tag, "Noon_OK");
          Serial.print("Noon_Pill_OK");
            delay(1000);
```

```
    servo_1.write(CLOSE);
    servo_2.write(CLOSE);
    servo_3.write(CLOSE);
    mrng=0;
    noon=1;
    night=0;
}
void open_night()
{
 n_pill=0;
      lcd.begin(16, 2);
        if(a_n >0)
        {
          lcd.setCursor(0,0);
          lcd.print("A : ");
          lcd.print(a_n);
          servo_1.write(OPEN);
        }
        else
        {
          lcd.setCursor(0,0);
          lcd.print("A : ");
          lcd.print(a_n);
          servo_1.write(CLOSE);
        }
       Serial.print("Night_Pill_OK");
       String tag ="";
        char buffer[16];
        tag += "test/MSG/";
```

```
            sprintf(buffer,"%d_%d_%d_%d_%d_%d",h,m,s,d,mo,y);
            tag += buffer;
            Firebase.RTDB.setString(&fbdo, tag, "Night_OK");
            delay(1000);
        servo_1.write(CLOSE);
        servo_2.write(CLOSE);
        servo_3.write(CLOSE);
        mrng=0;
        noon=0;
        night=1;
    }
```

6.1.2 faceRecognition.py

```
arduino = serial.Serial(port='COM5', baudrate=9600, timeout=.1)
from datetime import datetime
import csv
msg = "p6"
face_cascade = cv2.CascadeClassifier
('haarcascades/haarcascade_frontalface_default.xml')
path = 'res'
images = []
classNames = []
myList = os.listdir(path)
print(myList)
for cl in myList:
curImg = cv2.imread(f'{path}/{cl}')
images.append(curImg)
classNames.append(os.path.splitext(cl)[0])
```

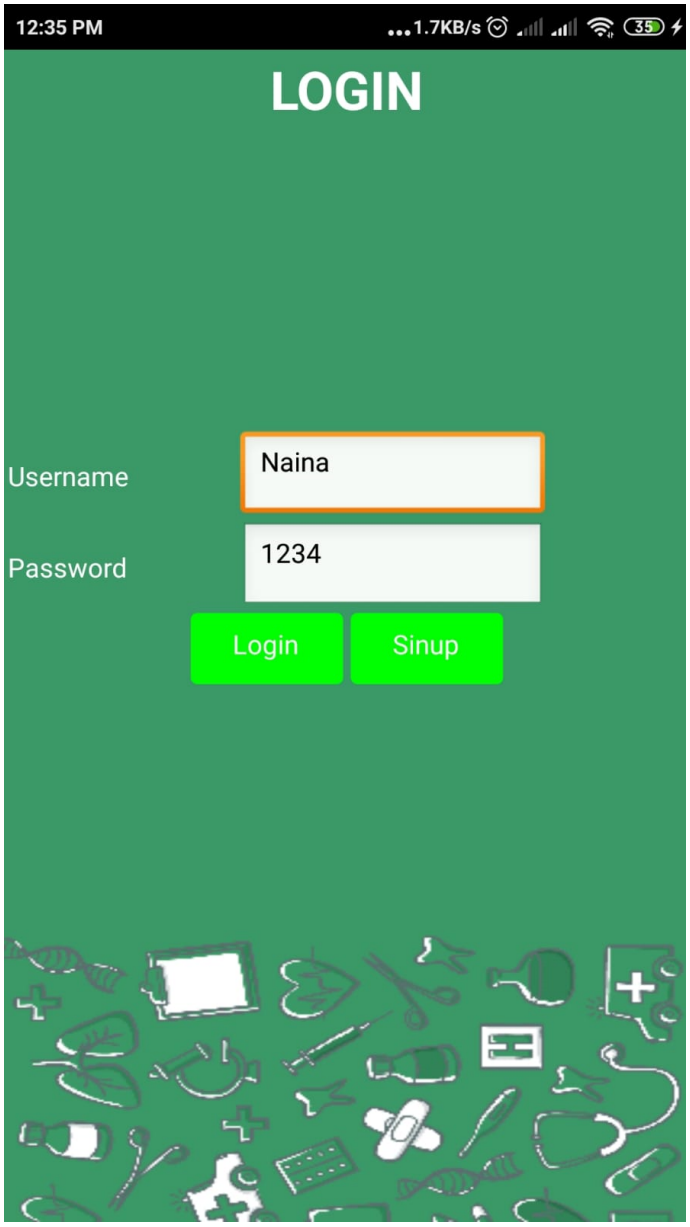
```
print(classNames)
def findEncodings(images):
encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
encode = face_recognition.face_encodings(img)[0]
encodeList.append(encode)
return encodeList
encodeListKnown = findEncodings(images)
print('Encoding complete')
cap = cv2.VideoCapture(0)
while True:
success, img = cap.read()
imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)
msg = "p0"
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  # Convert into grayscal
faces = face_cascade.detectMultiScale(gray, 1.2, 8)  # Detect faces
facesCurFrame = face_recognition.face_locations(imgS)
encodesCurFrame = face_recognition.face_encodings(imgS, facesCurFrame
for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):
matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
#print(faceDis)
matchIndex = np.argmin(faceDis)
if matches[matchIndex]:
name = classNames[matchIndex].upper()
#print(name)
y1, x2, y2, x1 = faceLoc
```

```
y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
cv2.putText(img, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX,
.5,(255, 255, 255), 1)
msg = "p1"
else:
for (x, y, w, h) in faces:  # Draw rectangle around the faces
cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 3)
cv2.putText(img, "Unknown", (x + 6, y - 6), cv2.FONT_HERSHEY_COMPLEX,
.5, (0, 0, 255), 1)
    msg = "p2"

    data = arduino.readline()
    if data:
        data = data.decode()
        print(data[0])
        #print(data[1])
        print(type(data))
        if data[0] == "a":
            arduino.write(bytes(msg, 'utf-8'))
            time.sleep(0.05)
            print(msg)
    cv2.imshow('Webcam', img)
    cv2.waitKey(1)
```

6.2 SCREENSHOTS

6.2.1 UI Design



Login page

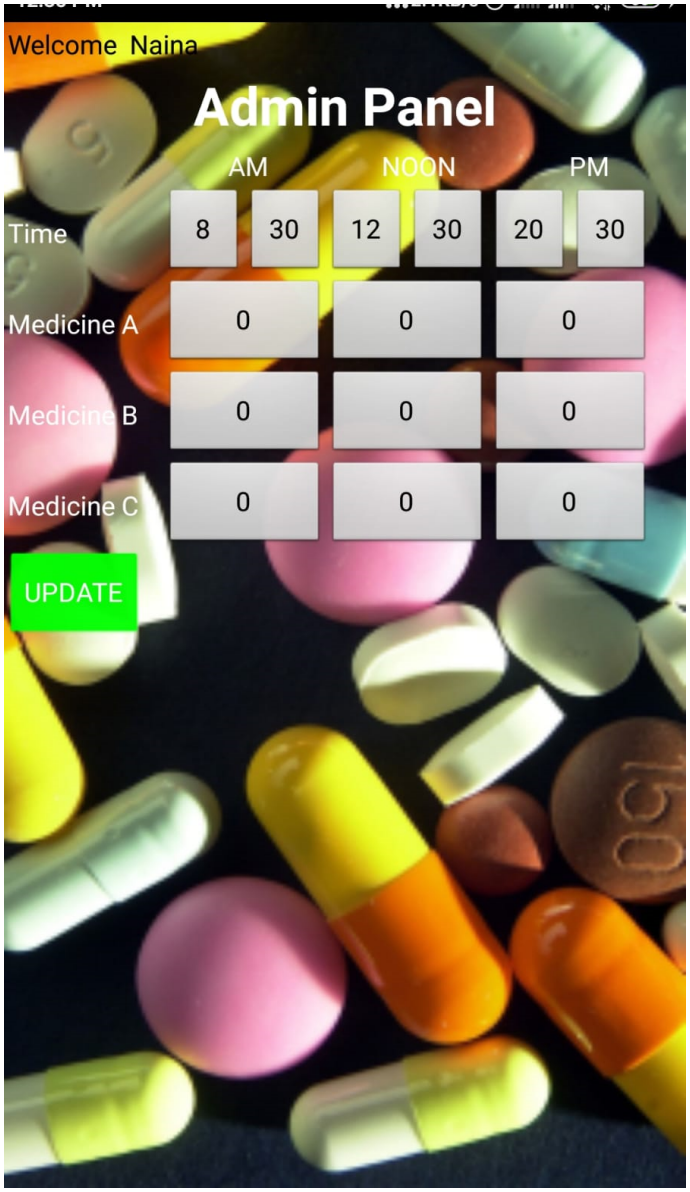
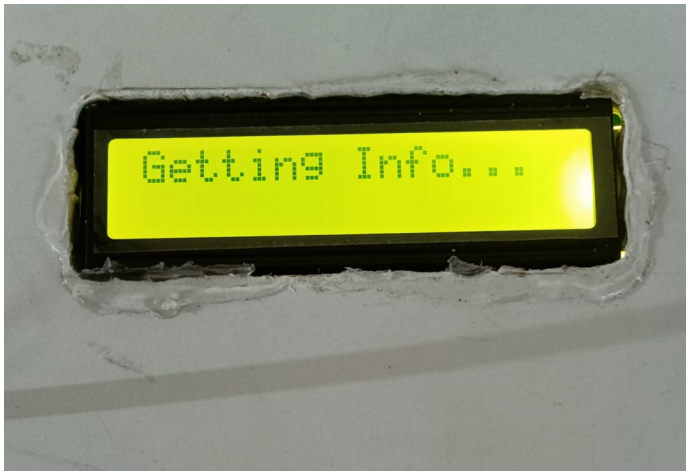


Figure 6.1: Admin Panel

6.2.2 Device Images

Auto Medicine Dispenser



Getting information from Database

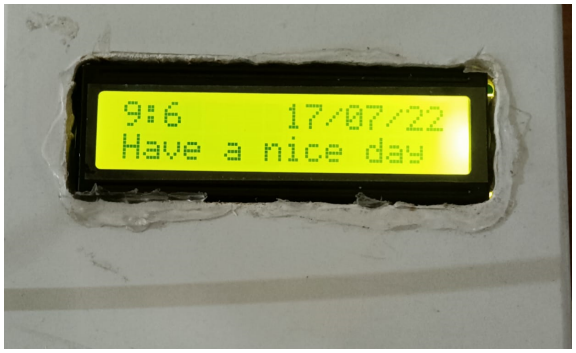


Figure 6.2: Screen

Figure 6.3: Time Scheduled For Medicine

Figure 6.4: Time Scheduled For Medicine

Figure 6.5: Pill Alert

Figure 6.6: Medicine Count

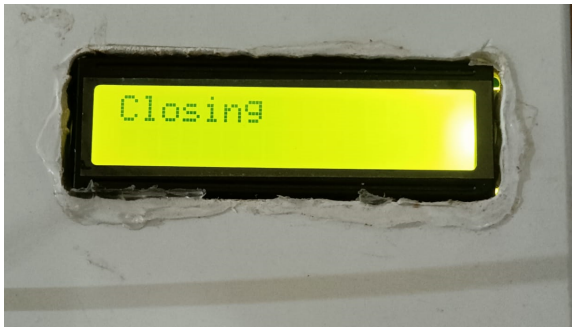


Figure 6.7: Closing

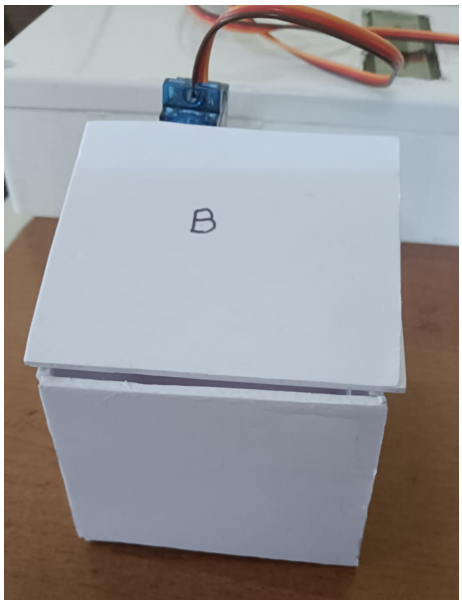


Figure 6.8: Auto Medicine Dispenser

Chapter 7

References

[1] Evgeny Stankevich, Ilya Paramonov, Ivan Timofeev, 'Mobile Phone Sensors in Health Applications'.

[2] Slagle, J.M., Gordon, J.S., Harris, C.E., Davison, C.L., Culpepper, D.K.,Scott P. and Johnson, K.B., (2011) "MyMediHealth– Designing a next generation system for child-centered medication Management", Journal of Biomedical Informatics, Vol. 43, No. 5.

[3] Zao, J.K., Wang, M.Y., Peihsuan, T. and Liu, J.W.S., (2010) "Smart Phone Based Medicine In-take Scheduler, Reminder and Monitor",IEEE e-Health Networking Applications and Services (Healthcom).

[4] Prasad, B., (2013) "Social media, health care, and social networking",Gastroin test Endosc. Vol. 77.

[5] S.T.-B. Hamida, E. Ben Hamida, B. Ahmed, and A. AbuDayya. Towards efficient and secure in-home wearable insomnia monitoring and diagnosis system. 13th IEEE Int. Conf. Bioinforma. Bioeng.pp. 1–6; 2013.

[6] P. Ray.Home Health Hub Internet of Things (H 3 IoT): An architectural framework for monitoring health of elderly people. Sci. Eng. Manag. Res. , pp. 3–5, 2014.

[7] S. S. Al-majeed. Home Telehealth by Internet of Things (IoT). pp.609–613; 201.

[8] S. Huang, H. Chang, Y. Jhu, and G. Chen. The Intelligent Pill Box -Design and Implementation. pp. 235–236; 2014.

[9] C. List, O. F. Authors, D. Moga, N. Stroia, D. Petreus, R. Moga, and R. A. Munteanu. Work Embedded Platform for Web-based Monitoring and Control of a Smart Home no. 53, pp. 1–3; 2015.

[10] J. M. Parra, W. Valdez, A. Guevara, P. Cedillo and J. OrtízSegarra, ”Intelligent pill-box: Automatic and programmable Assistive Technology device,” 2017 13th IASTED International Conference on Biomedical Engineering (BioMed), Innsbruck, Austria, 2017, pp. 74-81, doi: 10.2316/P.2017.852-051.

[11] G H.-W. Kuo, “Research and Implementation of Intelligent MedicalBox,” M.S.thesis, Department of Electrical Engineering, I-ShouUniversity, Kaohsiung, TW, 2009.

[12] A. Sawand, S. Djahel, Z. Zhang, and F. Na. Multidisciplinary Approaches to Achieving Efficient and Trustworthy eHealth MonitoringSystems. Commun. China (ICCC), 2014 IEEE/CIC Int. Conf., pp.187–192; 2014.

[13] Ahmad S., Afzal M.M. (2020), “Deployment of Fog and Edge Computingin IoT for Cyber-Physical Infrastructures in the 5G Era”. In: Karru-pusamy P., Chen J., Shi Y. (eds)Sustainable Communication Networksand Application. ICSCN 2019. Lecture Notes on Data Engineering andCommunications Technologies, vol 39. Springer, Cham

[14] E. E. Abdallah and E. Fayyoumi, ”Assistive Technology for Deaf People Based on Android Platform”, Procedia - Procedia Comput. Sci., vol. 94, pp. 295-301, 2016.