AN IoT BASED MEDICINE DISPENSER TO TACKLE PATIENT MEDICATION non-ADHERENCE

A MAJOR PROJECT REPORT

submitted by

SHUBHAM KUMAR(251901004)

to
University Institute of Engineering and Technology
Kurukshetra University, Kurukshetra

in partial fulfilment of the requirements for the award of the Degree
of
Bachelor of Technology
in
Electronics and Communication Engineering

Department of Electronics and Communication Engineering
University Institute of Engineering and Technology
Kurukshetra University, Kurukshetra
May 2023

DECLARATION

We hereby declare that the project report entitled "**AN IoT BASED MEDICINE DISPENSER TO TACKLE PATIENT MEDICATION non-ADHERENCE**" submitted by us to the University Institute of Engineering and Technology, Kurukshetra University, during the academic year 2022-23 in partial fulfilment of the requirements for the award of Degree of Bachelor of Technology in Information Technology is a record of bonafide project work carried out by us under the guidance and supervision of **Mr. Abhishek Choudhary** . We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other university.

SHUBHAM KUMAR (251901004)

Place: Kurukshetra, Haryana
Date: May 29, 2023

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY
KURUKSHETRA UNIVERSITY

CERTIFICATE

This is to certify that the report entitled "**AN IoT BASED MEDICINE DISPENSER TO TACKLE PATIENT MEDICATION non-ADHERENCE**" submitted by **SHUBHAM KUMAR (251901004)** , to the University Institute of Engineering and Technology, Kurukshetra University, in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Information Technology is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other universities or institutes for any purpose.

Guide
Mr. Abhishek Choudhary
Assistant Professor
Dept. of ECE

HOD
Dr. Priyanka Jangra
Assistant Professor
Dept. of ECE

CONTENTS

ACKNOWLEDGEMENT

Many noble hearts contributed immense inspiration and support for the successful completion of the project. We are unable to express our gratitude in words to such individuals.

We would like to express our deep regard to **Prof. Sunil Dhingra**, Director, University Institute of Engineering and Technology, Kurukshetra University, for providing facilities throughout the works of our project.

We take this opportunity to express our profound gratitude to **Dr. Priyanka Jangra**, Head of the Department, Department of Electronics and Communication Engineering,University Institute of Engineering and Technology, Kurukshetra University, for providing permission and availing all required facilities for undertaking the project in a systematic way. We are extremely grateful to **Mr. Abhishek Choudhary** who guided us with their kind, ordinal and valuable suggestions.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this project.

Last, but not the least, We take pleasant privilege in expressing our heartful thanks to our friends who were of precious help in completing this project.

ABSTRACT

Internet of Things (IoT) technology refers to a connection of physical objects embedded with sensors, actuators, software, etc., also known as "things" on the platform of internet so that they can communicate with each other to enhance human-machine interaction. Using IoT technology, this project targets to tackle the issue of Patient Medication non-Adherence by tracking their medication adherence rates.

The project aims to solve the aforementioned issue by prototyping an IoT device that automates timely medicine dispensing along with notifying/alerting the user about it. It also tracks the pattern of medication adherence of the patient. The project is developed on ESP8266 microcontroller development board, which is a low-cost and high-performance device that supports Wi-Fi connectivity and IoT applications. The device is designed to be user-friendly, portable, and customisable according to the patient's needs and preferences.

The project will evaluate the feasibility and usability of the IoT device in a pilot study with a small sample of patients with chronic diseases who take oral medications. The study assess the technical performance and reliability of the device. The project will contribute to the advancement of IoT technology in healthcare and provide a potential solution for improving medication adherence and health outcomes.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

There has been advancement of various technologies in the field of medicine. However, still, there is a lack of regular monitoring of the patient and improper dosage of medication being taken by the patients. [1]

Patient medication non-Adherence is a critical issue with significant risks associated with the patient's health . Patients usually struggle to maintain a regular medication schedule due to forgetfulness, complex dosage, and lack of monitoring. This leads to below par treatment outcomes, increased healthcare costs, and even life-threatening situations.

The existing solutions for this purpose, such as traditional pillboxes and reminders, are limited as far as their effectiveness is concerned and they fail to provide real-time monitoring and feedback. Also, the absence of a reliable system to track and ensure medication adherence hinders healthcare providers from providing timely interventions and personalized care to patients.

Hence, there is a serious need to innovate a solution that takes advantage of the power of the Internet of Things (IoT) to develop a smart medicine dispenser that can tackle the issue of patient medication non-adherence.

1.2 Problem Statement

To develop an IoT-based medicinal pill dispenser to tackle the issue of Patient Medication non-Adherence along with aiding the patients to follow medication routine correctly, remind them to take medicine, and track their medication adherence rates.

1.3 Project Objective

The fundamental objective of our project is to improve the frequency of patient's medication adherence along with following allied objectives:

1.3.1 Enhanced Medication Adherence

Medication non-adherence is a prevalent issue that has a significant impact on the health outcomes of patients. The primary objective of this project is to address this challenge by creating an IoT-based pill dispenser. The aim is to substantially improve medication adherence rates through the implementation of various features. The system will offer timely reminders, real-time adherence tracking, and support patients in adhering to their prescribed medication regimens more effectively

1.3.2 Enhanced Patient Safety

The safety of patients is of utmost importance, and administering incorrect or missed medication doses can have serious implications. Through the utilization of an IoT-based pill dispenser, this project aims to safeguard patient safety by ensuring the accurate delivery of medications at the appropriate times. By reducing the occurrence of medication errors and adverse drug reactions, this technology will enhance patient safety and mitigate the potential harm associated with medication non-adherence.

1.3.3 Financial Savings

Non-adherence to medication regimens leads to significant expenses within the healthcare system, such as emergency room visits, hospitalizations, and extended treatments. This project aims to mitigate these costs by enhancing medication adherence through the use of an IoT-based pill dispenser. By reducing non-adherence, this technology has the potential to lower healthcare expenses associated with non-compliance. Additionally, it promotes overall efficiency within

the healthcare system by minimizing unnecessary resource utilization.

1.3.4 Highlighting the Potential of IoT Technology in Healthcare

The project serves as an evidence to the wide capabilities of IoT technology within the healthcare sector. Through the integration of connectivity, sensors, and data analytics, the IoT-based pill dispenser shows the significant impact these technologies can have on enhancing medication adherence and improving patient care.

CHAPTER 2

LITERATURE REVIEW

2.1 Internet of Things (IoT)

The Internet of Things (IoT) describes the network of physical objects called "things", that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools. With around 15.14 billion connected IoT devices today, the numbers are expected to grow to 22 billion by 2025. Over the past few years, IoT has become one of the most important technologies of the 21st century. Now that we can connect everyday objects—kitchen appliances, cars, thermostats, baby monitors—to the internet via embedded devices, seamless communication is possible between people, processes, and things. By means of low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyperconnected world, digital systems can record, monitor, and adjust each interaction between connected things. The physical world meets the digital world—and they cooperate. [2]

2.1.1 IoT Functional Blocks

- IoT systems consists of building blocks, known as things, such as sensors/actuators, connectivity, security, services, etc.
- The functional blocks are responsible for sensing, verification, actuation, management, and communication.
- These functional blocks are made up of devices that handle interactions between a web server and the client, enable controls and monitoring functions, manage data transfer, secure the IoT system through authentication

and various purposes, and offer an interface for monitoring and managing various concepts. [3]

Sensor/Actuator Block

The sensor/actuator block collects data and drive physical processes in an IoT system. *Sensors collect data from their surroundings, whereas actuators drive physical processes.* Sensors gather data on temperature, humidity, light, motion, and other variables, whereas actuators turn on lights, open doors, and control machines. These gadgets work together to collect data and operate in the physical world.

Connectivity Block

After data collection from the sensor/actuator block , it has to be sent to the upcoming blocks. Connectivity Blocks completes the gap between that data and other upcoming blocks by creating and managing communication channels between the IoT system devices using technologies such as Wi-Fi, Bluetooth, ZigBee, and cellular networks.

Data Processing Block

The obtained data from previous block is examined and processed in the data processing block. This block is responsible for data wrangling , data munging,recognizing patterns and detecting anomalies in the data. In short, it gathers information and insights from the data to enable fast data-driven decisions.

Application Block

The application block provides value to the end user. This block is responsible for utilization of the processed data to provide a specified function or service.

Security Block

The purpose of security block is to assure the IoT system's security. This block handles authentication and authorization, as well as data encryption during transmission and storage. [3]

2.1.2 IoT Design Methodology

There exist a generic IoT design methodology which is independent of specific product, service or programming language. IoT systems designed with this method-

ology reduces design time, testing time, maintenance time, complexity and better interoperability. [4]

The steps involved in the designing of an IoT system are described in figure 2.1.

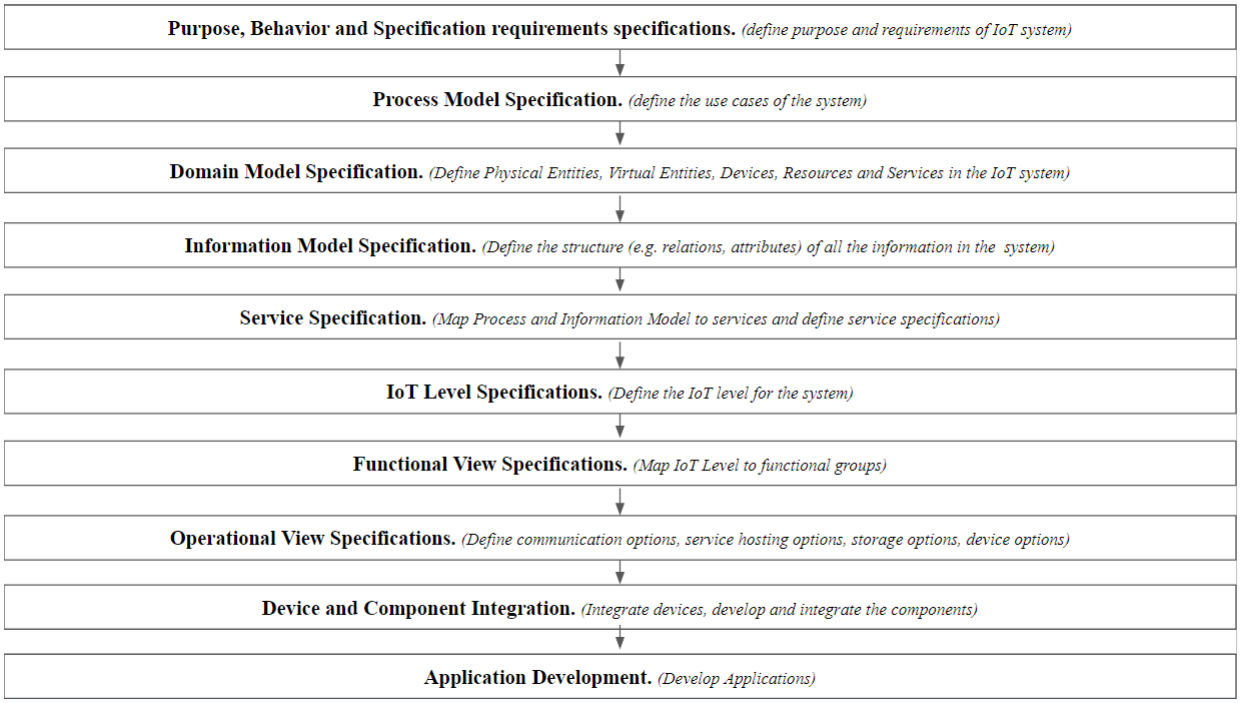

Figure 2.1: IoT Design Methodology

2.2 IoT Development on Embedded Systems Platform

2.2.1 Node-MCU ESP8266 Microcontroller Board

ESP8266 is the a micro controller designed by Espressif Systems. The ESP8266 is a self-contained WiFi networking solution offering as a bridge from existing micro controller to WiFi and is also capable of running self-contained applications. This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, we can connect NodeMCU development kit to our laptop and flash it just like Arduino. It is also breadboard friendly. [5]

Specifications

Voltage:3.3V; Wi-Fi Direct (P2P); Current consumption: 10uA 170mA; Flash memory attachable: 16MB max (512K normal); Integrated TCP/IP protocol

stack; Processor: Tensilica L106 32-bit; Processor speed: 80-160MHz; RAM: 32K + 80K; GPIOs: 17 (multiplexed with other functions); Analog to Digital: 1 input with 1024 step resolution; +19.5dBm output power in 802.11b mode; 802.11 support: b/g/n; Maximum concurrent TCP connections: 5; [5]

Pin Diagram

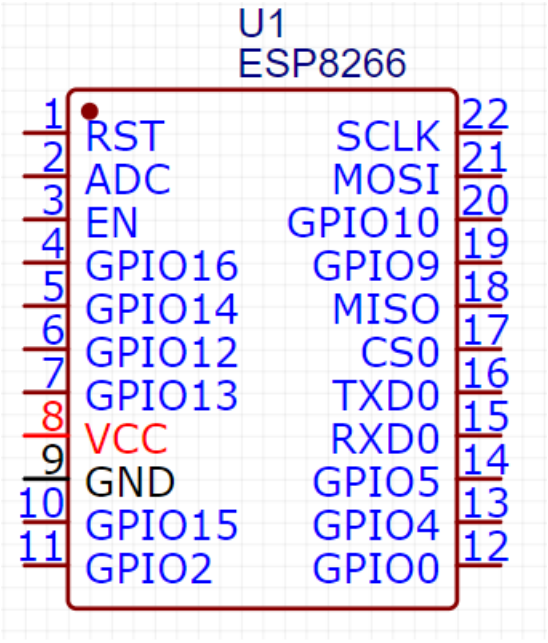

Figure 2.2: ESP8266 Pin Diagram

Setup

We use Arduino ESP8266 project, which is a modified version of the Arduino IDE that we need to install on the computer. This makes it very convenient to use the ESP8266 chip over the well-known Arduino IDE. Within Arduino IDE, we install the ESP8266-Arduino project package through additional board manager.

2.2.2 DS1307 I2C Real Time Clock Module

Real Time Clock (RTC) is used to track the current time and date. In many embedded system, we need to put time stamp while logging data i.e. sensor values, GPS coordinates etc. For getting timestamp, we need to use RTC (Real Time Clock). DS1307 RTC supports the I2C protocol. It uses a Lithium cell battery (CR1225). The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year.

The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. [6]

Specifications

5V DC supply; Programmable Square-Wave output signal; Automatic Power-Fail detect and switch circuitry ; Consumes less than 500nA in Battery-Backup Mode with Oscillator Running; 56-Byte, Battery-Backed, Non-volatile (NV)RAM for data storage; [6]

Pin Diagram

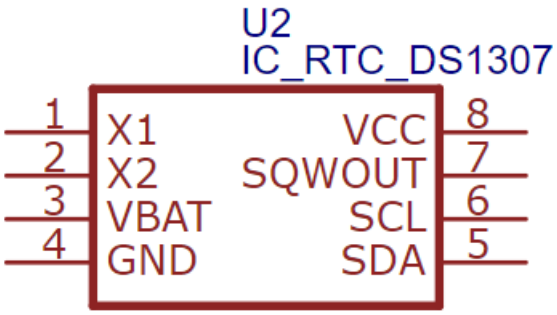

Figure 2.3: DS1307 I2C RTC Module Pin Diagram

Interfacing with ESP8266

The interfacing of DS1307 I2C RTC Module with ESP8266 is shown in figure 2.4.

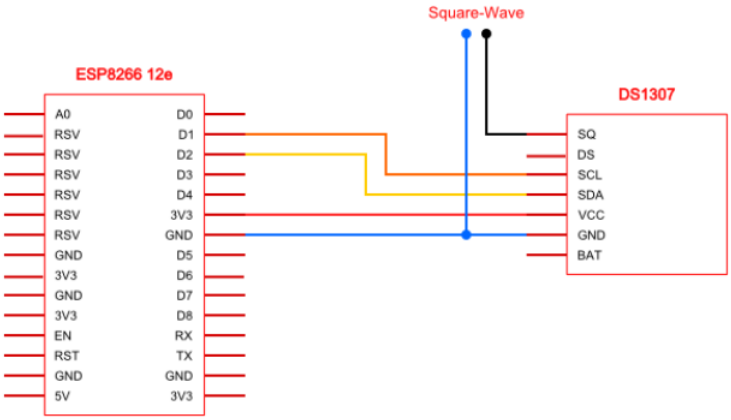

Figure 2.4: Interfacing DS1307 I2C RTC with ESP8266

2.2.3 TP4056 Linear Li-Ion Battery Charger

The TP4056 is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries. Due to its compact packaging and low external component count, the TP4056 suits portable applications. [7]

Specifications

Charge Controller: TP4056 / TC4056; Protection IC: DW01A; Charge/Discharge Control: MOSFET FS8205A; Charge Method: Constant-Current/Constant-Voltage (CC/CV); Input Supply Voltage: 4.5 6.0 V; Constant Charge Current: 1A(Configurable with Rprog resistor R3); Charge Complete (Float) Voltage: 4.2 V ±1.5[7]

Interfacing with ESP8266

The interfacing of DS1307 I2C RTC Module with ESP8266 is shown in figure 2.5

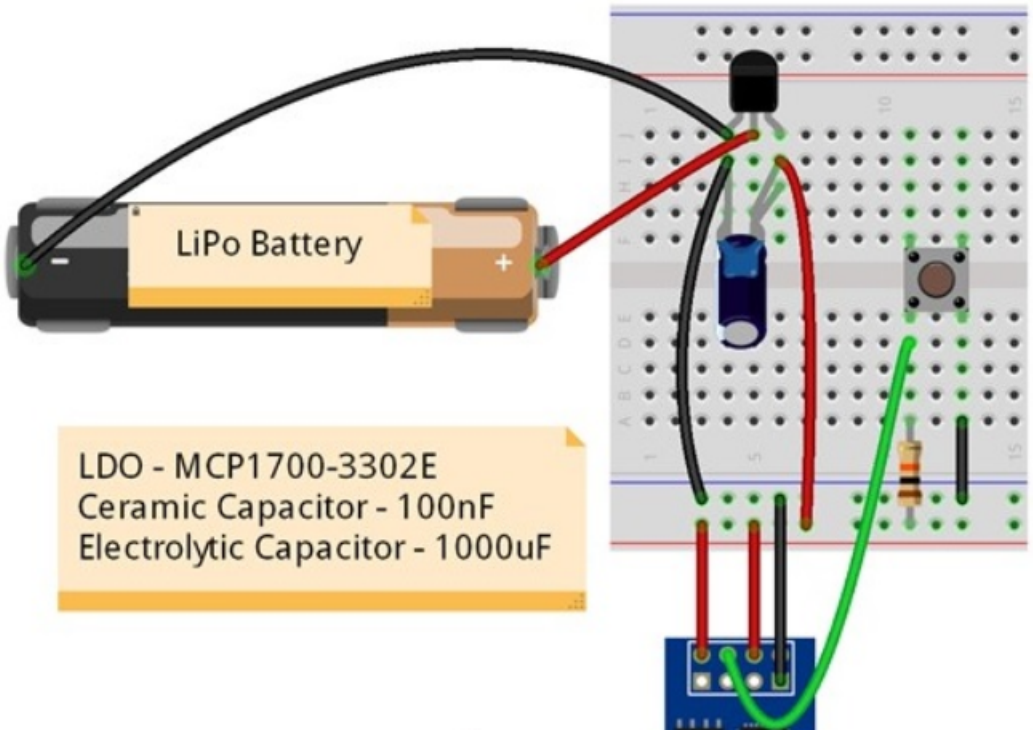

Figure 2.5: Interfcaing TP4056 Battery Charger Module with ESP8266 [8]

2.2.4 OLED Panel

The OLED display module is used to display text, images and patterns.

Specifications

Display Mode: Passive Matrix; Drive Duty: 1/64 Duty Number of Pixels: 128 × 64 ; Panel Size: 26.70 × 19.26 × 1.45 (mm) ; Active Area: 21.744 × 10.864 ; Pixel Pitch: 0.17 × 0.17 (mm) ; Pixel Size: 0.154 × 0.154 (mm); Weight: 1.54 (g) [9]

Interfacing OLED Panel with ESP8266

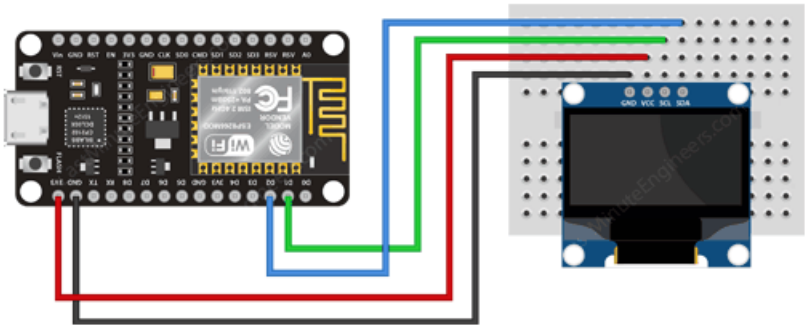

Figure 2.6: Interfacing OLED Panel with ESP8266
[10]

2.3 Patient Medical non-Adherence

According to The American Medical Association,

> "A patient is considered adherent if they take 80 percent of their prescribed medicine(s). If patients take less than 80 percent of their prescribed medication(s), they are considered non adherent." [11]

There are several barriers to achieving medication adherence. The two major barriers are as follows:

Forgetfulness

Including the act of taking medication into the daily routine can be a challenge initially. One usually adopts the methods of reminders, accountability, and drive for consistency. Even for doctors, a change in schedule can throw off their routine and lead to missed doses. It's human nature to forget things, even when things have become routine. [11]

Complex Medication Schedules

Complex medication schedules requires medication organization as well as setting reminders to take doses in a day at the right time intervals. For some patients like children, others may have to be involved to support the patient in medicating, like their parent. [11]

CHAPTER 3

METHODOLOGY

3.1 Purpose and Behavior Specifications

Purpose

A medicine pill dispenser that addresses the challenges of Patient Medication non-Adherence. The system should enable patients to follow their medication schedule accurately, provide reminders, and track adherence in real-time.

Behavior

The system supports two modes:

- **Auto**: If, within around a couple of minutes from medication time, if the user does not takes the medication, then the device automatically sends the dose missing notification through mail to the user.
- **Manual**: At the specified time of medication, user himself, presses button the device and takes the medication.

3.2 Process Model Specifications

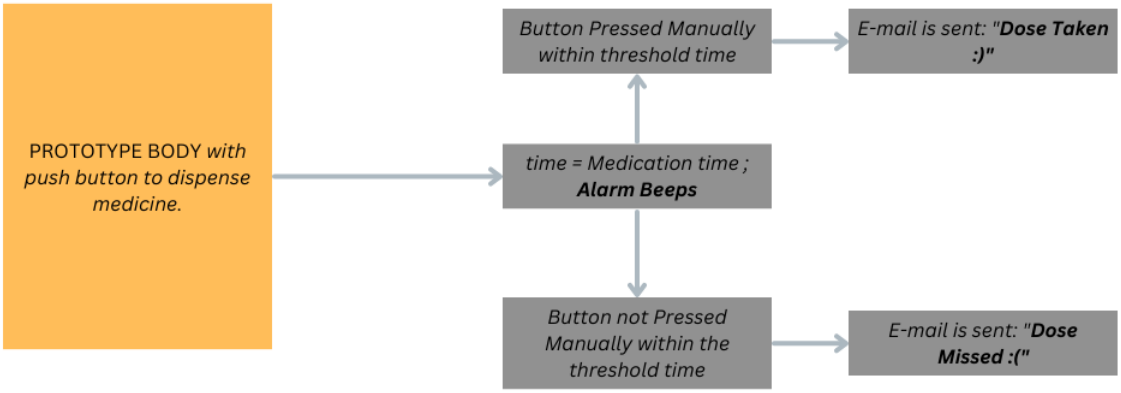

Figure 3.1: Process Model Specification

3.3 Device and Component Integration

Prototype Body

- For the prototype's body, we made a box using cardboard of relevant dimensions. We also made gaps and cuts to fit our circuitry into it.

Prototype Circuitry

- For the circuitry, we took ESP8266 micro controller board and connect the RTC Module, Lithium-Ion Battery, OLED Panel, TP4056 Battery Charger, Servo Motors and the other miscellaneous components (capacitor, resistors, etc.) as per the following circuit diagram.

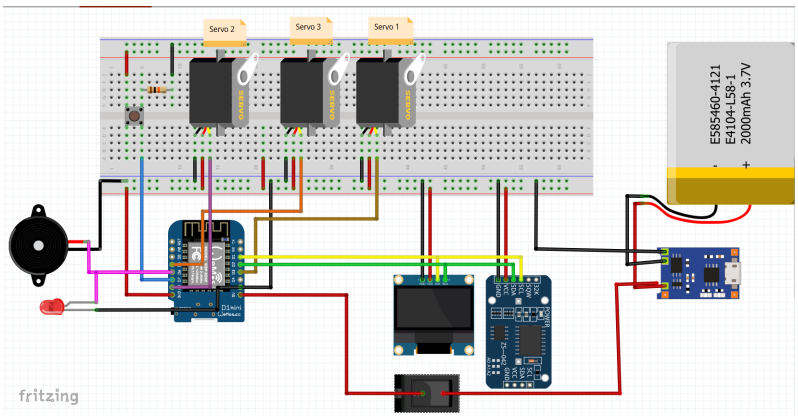

Figure 3.2: Schematic of Prototype's Circuitry

- The entire circuitry was prototyped on a protoboard through soldering process.
- The battery charger is connected to a power source to charge the battery throug a type-C cable.

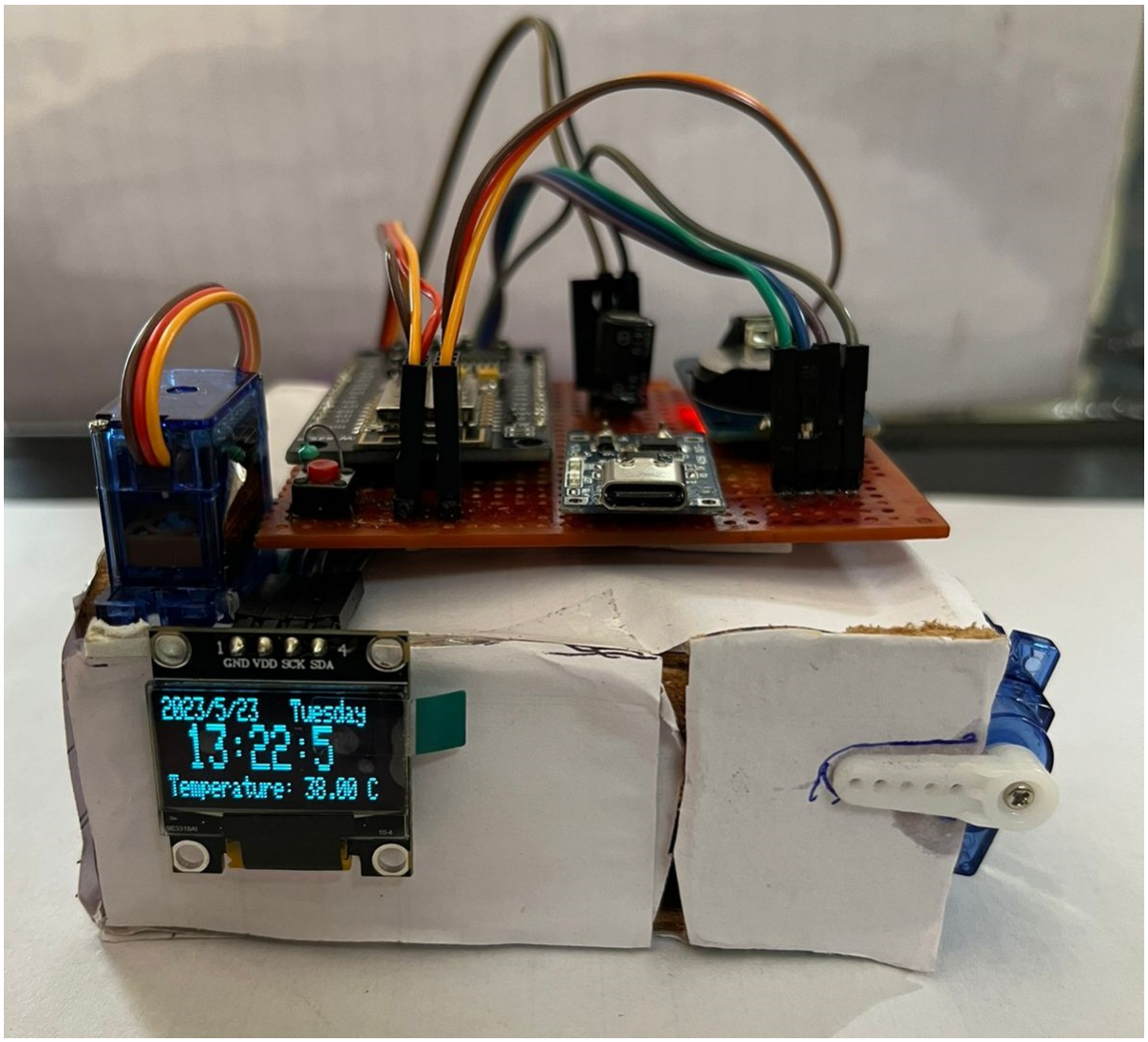

Figure 3.3: Front view of the Prototype

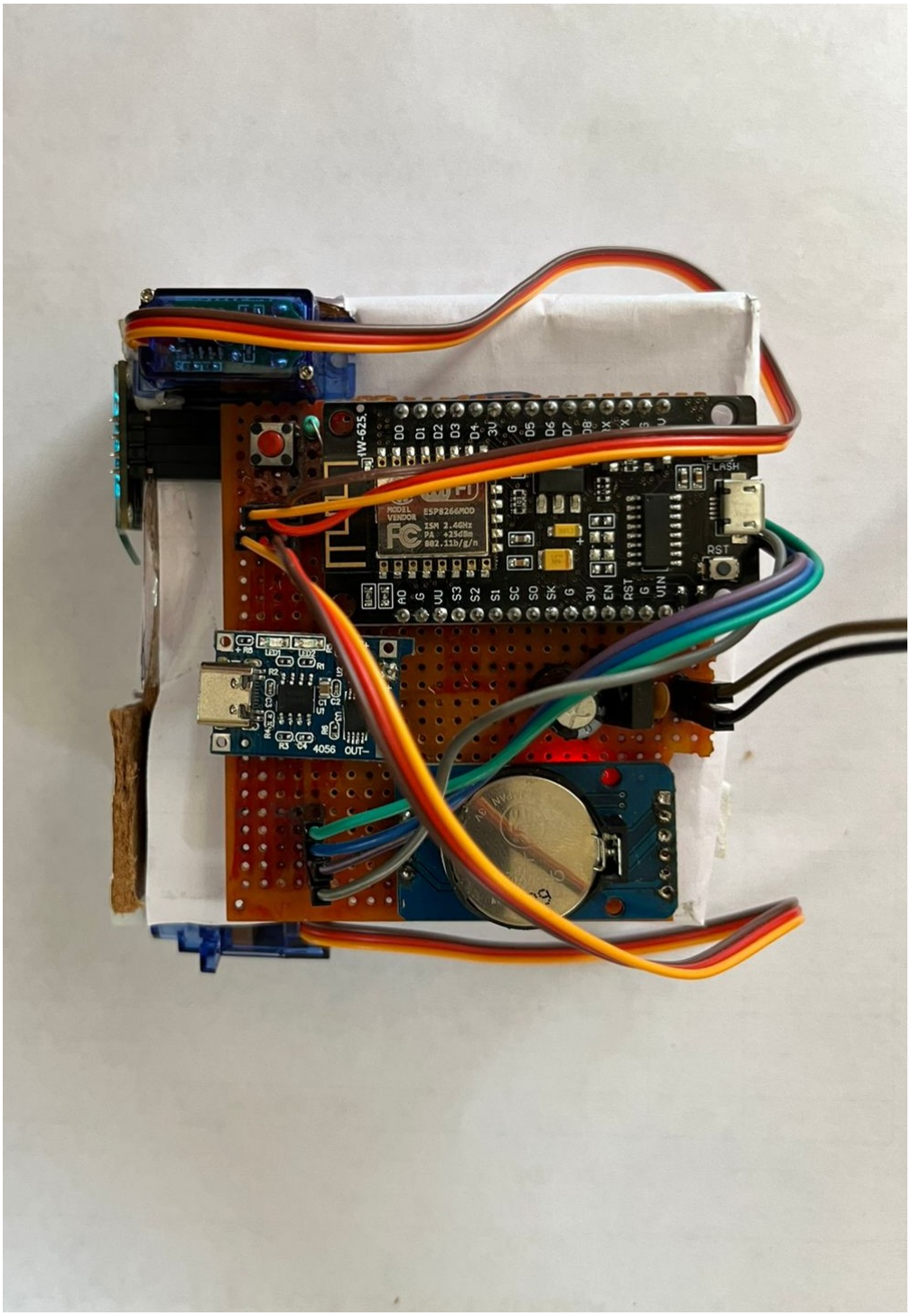

Figure 3.4: Top view of the Prototype

Prototype Code

The the micro controller board was then connected to the Arduino IDE on the laptop and the code to control the device was uploaded on the micro controller board. Following is the code [12]:

```
#include "RTClib.h"
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Servo.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32

#define OLED_RESET     -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
↪  OLED_RESET);

#define LOGO_HEIGHT   32
#define LOGO_WIDTH    70

RTC_DS3231 rtc;
Servo s1;
Servo s2;

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday",
↪  "Wednesday", "Thursday", "Friday", "Saturday"};

int hour_1 = 9;
int min_1 = 10;

const int buzzer = D6;
const int button = D7;

unsigned long previous;
unsigned long current;

int state;
int alarm = 0;
```

```
const char* ssid = "realme7";
const char* password = "";
const char* address1 =
↪  "http://maker.ifttt.com/trigger/button/with/key/bEuQ_p07_1yZ23wn5K480X";
↪  //Replace EVENT_NAME with applet's event and KEY with your
↪  key
const char* address2 =
↪  "http://maker.ifttt.com/trigger/not_pressed/with/key/bEuQ_p07_1yZ23wn5K480X
const uint16_t port = 17;

void ifttt(int);

void drawbitmap();

static const unsigned char PROGMEM logo_bmp[] = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
↪  0x00, 0x00, 0x00, 0x00, 0x00,
  0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00,
↪  0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
↪  0xc0, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xa0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
↪  0x00, 0x00, 0xa0, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xa0, 0x00, 0x00, 0xff, 0xff,
↪  0xff, 0xff, 0xff, 0xff, 0x20,
  0x00, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x0f,
↪  0x80, 0x00, 0x01, 0xff, 0xff,
  0xff, 0x10, 0x00, 0x1c, 0x00, 0x00, 0x01, 0xff, 0xff, 0xfe, 0x1e,
↪  0x00, 0x38, 0x00, 0x00, 0x01,
  0xff, 0xff, 0xc0, 0x00, 0xf0, 0x30, 0x00, 0x00, 0x01, 0xff, 0xff,
↪  0xe0, 0x00, 0xf0, 0x60, 0x00,
  0x00, 0x01, 0xff, 0xff, 0xfc, 0x0f, 0x00, 0x60, 0x00, 0x00, 0x01,
↪  0xff, 0xff, 0xff, 0x1e, 0x00,
  0xe0, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0x1e, 0x00, 0xc0, 0x00,
↪  0x00, 0x01, 0xff, 0xff, 0xff,
  0x3e, 0x00, 0xc0, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xbf, 0x00,
↪  0xc0, 0x00, 0x00, 0x01, 0xff,
  0xff, 0xff, 0xbf, 0x00, 0xc0, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff,
↪  0xbf, 0x00, 0xc0, 0x00, 0x00,
```

```
  0x01, 0xff, 0xff, 0xff, 0xfe, 0x00, 0xe0, 0x00, 0x00, 0x01, 0xff,
↪  0xff, 0xff, 0xfe, 0x00, 0x60,
  0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x70, 0x00, 0x00,
↪  0x01, 0xff, 0xff, 0xff, 0xfc,
  0x00, 0x30, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x38,
↪  0x00, 0x00, 0x01, 0xff, 0xff,
  0xff, 0xf8, 0x00, 0x1e, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xf0,
↪  0x00, 0x0f, 0x80, 0x00, 0x01,
  0xff, 0xff, 0xff, 0xe0, 0x00, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff,
↪  0xff, 0xc0, 0x00, 0x00, 0x7f,
  0xff, 0xff, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
↪  0x00, 0x00, 0x00, 0x00, 0x00
};

void setup () {

  pinMode(buzzer, OUTPUT);
  pinMode(button,INPUT);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);   //connect to sssid

  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print("!!");
  }
//  WiFiClient wificlient;
//  WiFi.mode(WIFI_OFF);  //Otherwise lot of power is consumed if
↪  esp is used in ap mode

  #ifndef ESP8266
  while (!Serial);
  #endif

  Serial.begin(9600);

  delay(3000);

  if (! rtc.begin()) {
```

```
  Serial.println("Couldn't find RTC");
  while (1);
}

if (rtc.lostPower()) {
  Serial.println("RTC lost power, lets set the time!");   //if
↪  ds3231 lost power then set rtc clock to compilation time
  rtc.adjust(DateTime(F(_DATE), F(__TIME_)));
}

if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
  Serial.println(F("SSD1306 allocation failed"));
  for(;;);
}

drawbitmap();

display.clearDisplay();

display.setTextSize(1);
display.setTextColor(SSD1306_BLACK, SSD1306_WHITE);
display.setCursor(0,0);
display.println(F("      MEDICINE       "));

display.setTextSize(1);
display.setTextColor(SSD1306_BLACK, SSD1306_WHITE);
display.println(F("      DISPENSER      "));

display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.println(F("Made By:"));

display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.println(F("Akshat and Shubham "));

display.display();

delay(4000);
```

```
}

void loop () {
    DateTime now = rtc.now();

    Serial.print(now.year(), DEC);
    Serial.print('/');
    Serial.print(now.month(), DEC);
    Serial.print('/');
    Serial.print(now.day(), DEC);
    Serial.print(" (");
    Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
    Serial.print(") ");
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    Serial.print(now.minute(), DEC);
    Serial.print(':');
    Serial.print(now.second(), DEC);
    Serial.println();

    Serial.print("Temperature: ");
    Serial.print(rtc.getTemperature());
    Serial.println(" C");

    display.clearDisplay();

    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0,0);
    display.print(now.year(), DEC);
    display.print('/');
    display.print(now.month(), DEC);
    display.print('/');
    display.print(now.day(), DEC);
    display.print("   ");
    display.print(daysOfTheWeek[now.dayOfTheWeek()]);

    display.println();
```

```
display.setTextSize(2);
display.setTextColor(SSD1306_WHITE);
display.print(" ");
display.print(now.hour(), DEC);
display.print(':');
display.print(now.minute(), DEC);
display.print(':');
display.print(now.second(), DEC);

display.println();

display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.print("Temperature: ");
display.print(rtc.getTemperature());
display.println(" C");

display.display();

state = digitalRead(button);

if(state == HIGH)   //button is pressed for sometime
{
  display.clearDisplay();

  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0,1);
  display.println("  Dispensing.....");

  display.display();

  s1.attach(D3);

  s1.write(0);
  delay(1000);
  delay(1000);
```

```
    s2.attach(D5);

    s2.write(0);
    delay(1000);
    s2.write(180);
    delay(1000);
    s2.write(0);

    s2.detach();

    s1.write(180);

    s1.detach();

    display.clearDisplay();

    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0,1);
    display.println("   Pill dispensed!!");

    display.display();

    ifttt(1);
    Serial.println("Pill taken!");

    digitalWrite(buzzer,HIGH);
    delay(500);
    digitalWrite(buzzer,LOW);
  }
  if(hour_1 == now.hour() && min_1 == now.minute() &&
↪ (now.second() == 0 || now.second() == 1))
  {
    alarm = 1;
    previous =  millis();
    current = previous;

    while(current - previous < 60000)
    {
```

```
  current = millis();
  state = digitalRead(button);
  digitalWrite(buzzer,HIGH);

  display.clearDisplay();

  display.setTextSize(2);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0,1);
  display.println("Take Pill!");

  display.display();

  if(state == 1)    //button pressed
  {
    digitalWrite(buzzer,LOW);   //turn off the alarm

    display.clearDisplay();

    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0,1);
    display.println("  Dispensing.....");

    display.display();



s1.attach(D3);

s1.write(80);
delay(1000);
s1.write(0);
delay(1000);
s1.write(80);

s1.detach();

s2.attach(D5);
```

```
s2.write(10);
delay(1000);
s2.write(90);
delay(1000);
s2.write(10);

s2.detach();




    display.clearDisplay();

    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0,1);
    display.println("   Pill dispensed!!");

    display.display();

    ifttt(1);   //sent email indicating pill was dispensed
    Serial.println("Pill taken!");
    alarm = 0;

    digitalWrite(buzzer,HIGH);
    delay(500);
    digitalWrite(buzzer,LOW);

    break;
  }
  else
  {
    Serial.println("Time to take your pill!");
  }
}

if(alarm == 1)  //if 60s passed and button wasn't pressed
{
```

```
          ifttt(2);      //

          digitalWrite(buzzer,LOW);

          display.clearDisplay();

          display.setTextSize(1);
          display.setTextColor(SSD1306_WHITE);
          display.setCursor(0,1);
          display.println("Pill wasn't dispensed!");

          display.display();

          delay(1000);

          display.clearDisplay();

          display.setTextSize(1);
          display.setTextColor(SSD1306_WHITE);
          display.setCursor(0,1);
          display.println("Notification Sent!");

          display.display();

        }
      }

      Serial.println();

      delay(1000);
}


void ifttt(int opt)
{

  if(WiFi.status()== WL_CONNECTED){
      HTTPClient http;
      WiFiClient wificlient;
```

```
      if(opt == 1)
      {
        http.begin(wificlient,address1);
      }
      else if(opt == 2)
      {
        http.begin(wificlient,address2);
      }

      http.addHeader("Content-Type",
↪  "application/x-www-form-urlencoded");
      String httpRequestData = "value1=" + String(20) + "&value2="
↪  + String(20)+ "&value3=" + String(20); // build the string
      int httpResponseCode = http.POST(httpRequestData);

      http.end();
    }

//  WiFi.disconnect(WIFI_OFF);   //again turn esp off otherwise a
↪  lot of power is wasted

}

void drawbitmap(void) {
  display.clearDisplay();

  display.drawBitmap(
    (display.width()  - LOGO_WIDTH ) / 2,
    (display.height() - LOGO_HEIGHT) / 2,
    logo_bmp, LOGO_WIDTH, LOGO_HEIGHT, 1);
  display.display();
  delay(2000);
}
```

3.4 Application Development

The user will be able to receive notifications through emails using IFTTT's Webhooks service.

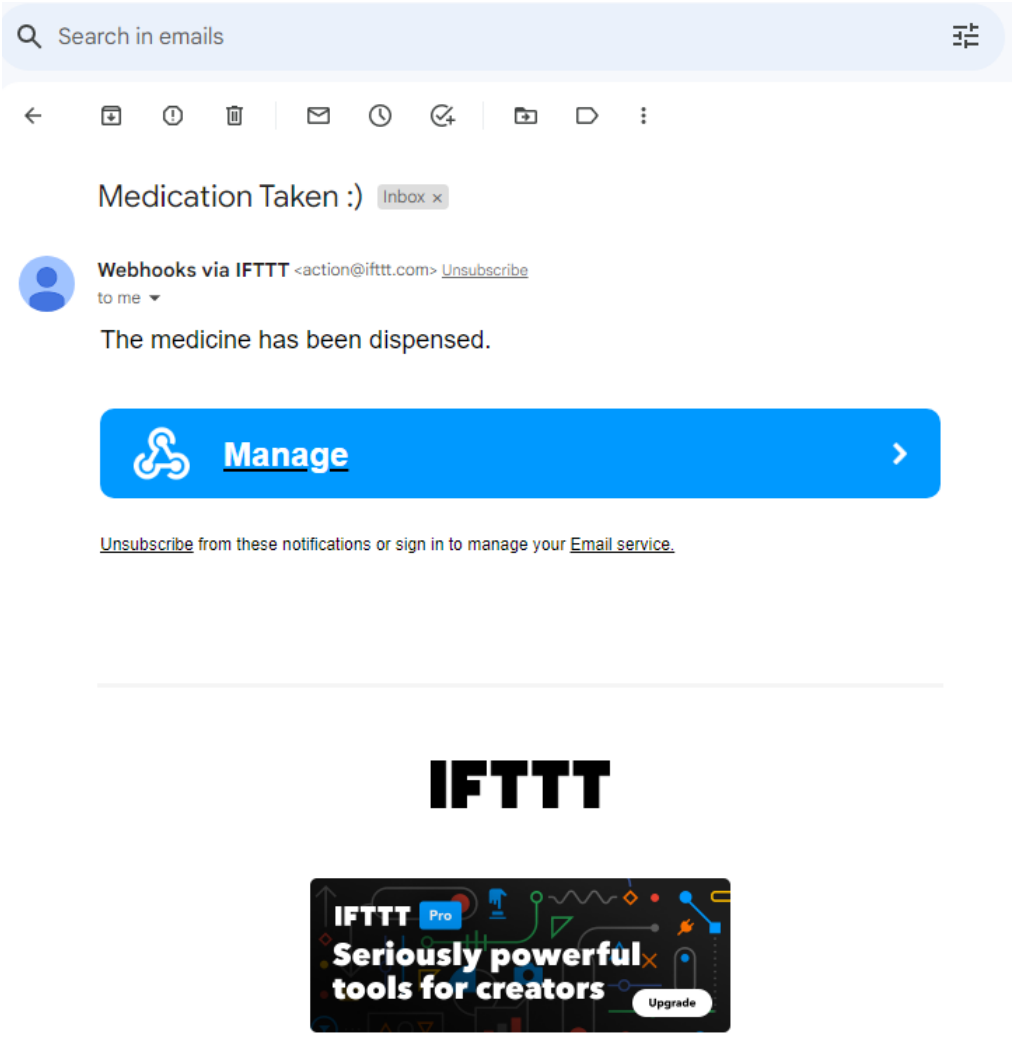

Figure 3.5: User receiving Mail regarding Medication

CHAPTER 4

CONCLUSIONS

4.1 Target Audience for the Project

Though the prototype can be used by any patient on pill based medication suffering from medication non-adherence, in general, the project targets a very niche audience that includes patients suffering from conditions like Alzheimer's or patients in their old age who easily forget their medication. Family members of the patient administering them can also use the prototype.

4.2 Advantages of the Project

- Use of low cost electronic components in the prototype makes it is financially accessible
- Patient will be alerted to take medication when it is the time(using an alarm beep), so that he or she doesn't forget to adhere the medication.
- In case the patient forget to take medicine, an email will be sent to him or her about the same.

4.3 Limitations of the Project

- For time being, the product is a prototype, and is not a finished product as far as aesthetics and convenience is concerned. The feature of sending alerts when the compartment becomes empty is not available for now. This feature is planned as a future enhancement by using relevant sensors.
- The number of medicine has been limited to one, for time being, as each

different medicine will require a compartment of its own and hence a motor and related hardware of its own. The facility for more number of medicines is also planned as future enhancement.

- For now, only email notifications are available.Phone call/Whatsapp notifications are planned as future enhancements.

- The current prototype is useful only for pill based medications, but not for liquid/syrup based or drop based(like eye-drops) medications.

These limitations will be removed with more availability of time further enhancements.

4.4 Project Cost Considerations

The project costed us around Rs.1135. In next iterations of enhancement, we aim to reduce the cost of prototype by selecting cheaper yet efficient components.

- Node MCU ESP8266: Rs.240

- RTC Module: 210

- TP4056 Module: Rs.125

- GME12864-53 OLED Panel: Rs.250

- Li-Ion Battery: Rs.80

- SG9T Servo moters: Rs.100 x 2 = Rs.200

- Miscellaneous(Capacitors, Voltage regulators, etc.) = Rs.30

REFERENCES

[1] Panda N.K, Bhardwaj S, Bharadwaj H, and Singhvi R. Iot based advanced medicine dispenser integrated with an interactive web application, 2018.

[2] What is iot? https://www.oracle.com/in/internet-of-things/what-is-iot/#:~:text=What%20is%20IoT%3F,and%20systems%20over%20the%20internet.

[3] Iot functional blocks. https://www.tutorialspoint.com/iot-functional-blocks#:~:text=IoT%20systems%20are%20composed%20of,actuation%2C%20management%2C%20and%20communication.

[4] Iot design methodology. https://www.startertutorials.com/blog/iot-design-methodology.html.

[5] Esp8266 nodemcu wifi development board. https://handsontec.com/dataspecs/module/esp8266-V13.pdf.

[6] Ds1307 i2c real time clock module. https://5.imimg.com/data5/UM/UC/MY-1833510/ds1307-i2c-real-time-clock-module.pdf.

[7] Tp4056 / tc4056a lithium battery charger and protection module. https://www.addicore.com/TP4056-Charger-and-Protection-Module-p/ad310.htm.

[8] Esp8266 voltage regulator (lipo and li-ion batteries). https://randomnerdtutorials.com/esp8266-voltage-regulator-lipo-and-li-ion-batteries/.

[9] 128x64 oled module spi 0.96.

[10] Interface oled graphic display module with esp8266 nodemcu. https://lastminuteengineers.com/oled-display-esp8266-tutorial/.

[11] What is medication adherence? https://blog.cureatr.com/what-is-medication-adherence.

[12] Anoojstate.ino. https://github.com/anooj-shah/Smart-Pill-Dispenser.