Arduino based medicine reminder for Senior Citizens.

Mini Project Report

Submitted in partial fulfillment of the requirements for the
Degree of Bachelor of Technology in Electronics and Communication Engineering
under Maulana Abul Kalam Azad University of Technology

By

RUPESH SINGH
ROLL NUMBER: 14200320021

SOUMYOPRABHA KARMAKAR
ROLL NUMBER: 14200320003

NELADRI SENGUPTA
ROLL NUMBER: 14200320026

ASMITA ADHIKARY
ROLL NUMBER: 14200320015

Under the guidance of
PROF. SUDIPTA GHOSH
Professor

Department of Electronics and Communication Engineering
Meghnad Saha Institute of Technology

Kolkata – 700150

MEGHNAD SAHA INSTITUTE OF TECHNOLOGY

Meghnad Saha Institute of Technology

Nazirabad, P.O. : Utchepota, Via Sonarpur, Kolkata 700 150, Phone : 033-2443-1754, Telefax : 033-2443-1032

CERTIFICATE OF APPROVAL

I hereby recommend that the work in preparing the mini project report entitled "Arduino based medicine reminder for Senior Citizens." carried out by **SOUMYOPRABHA KARMAKAR, RUPESH SINGH, NELADRI SENGUPTA** and **ASMITA ADHIKARY** under my supervision may be accepted in partial fulfillment of the requirements for the degree of Bachelor of Technology in Electronics and Communication Engineering of Maulana Abul Kalam Azad University of Technology (MAKAUT).

……………………………………

Prof. Sudipta Ghosh
Project Guide
Professor, Dept. of ECE, MSIT

……………………………………

Dr. Manash Chanda
HOD
Dept. of ECE, MSIT

ll

ACKNOWLEDGEMENTS

We would like to express our sincere regards to Professor (Prof.) Sudipta Ghosh of the Department of Electronics & Communication Engineering, Meghnad Saha Institute of Technology for his guidance, valuable advice and constructive suggestions for carrying out this mini project work.

We would like to record our indebtedness to Dr. Manas Chanda, HOD, Department of Electronics and Communication Engineering Meghnad Saha Institute of Technology for providing us with all the support that was needed.

We would also like to thank all the faculty members of ECE department, MSIT for their valuable suggestions during the course of our work.

Finally, our sincere thanks go to our parents for their encouragement and support during this work.

……………………………………….

Name: **SOUMYOPRABHA KARMAKAR**
Roll No.: 14200320003

……………………………………….

Name: **NELADRI SENGUPTA**
Roll No.: 14200320026

……………………………………….

Name: **RUPESH SINGH**
Roll No.: 14200320021

……………………………………

Name: **ASMITA ADHIKARY**
Roll No.**:** 14200320015

ll

CONTENT

Description	Page Number
CHAPTER 1: ABSTRACT	5
CHAPTER 2: INTRODUCTION	6
2.1: BACKGROUND/Literature Survey	7
2.2: PROBLEM STATEMENT	8
2.3: OBJECTIVES	9
2.4: SCOPE	10
CHAPTER 3: PRESENT WORK	11
3.1: ML IMPLIMENTATION	12
3.1.1: IMAGE ACQUISITION	13-14
3.1.2: PREPROCESSING	15-16
3.1.3: FEATURE EXTRACTION	17-18
CHAPTER 4: RESULT AND DISCUSSION	19-20
CHAPTER 5: CONCLUSION	21
CHAPTER 6: LIMITATIONS	22-23
CHAPTER 7: FUTURE SCOPE	24-25
CHAPTER 8: REFERENCES	26
CHAPTER 9 CODE	27-39

ll

CHAPTER ONE*: **ABSTRACT***

This is an android based application which reminds the patients by ringing an alarm system and by giving notification so that there is no need to remember the entire medicine doses name and their timings throughout the month. The application is user friendly and has easy to understand functions. The user can also see the medicine history which is consumed by the patient in a few steps. Many medicines related application have been developed but they have some minor problems related to their functionality. But we tried to overcome these problems so it will be convenient to the users in medical adherence**.**

CHAPTER TWO: **INTRODUCTION** to

ARDUINO BASED MEDICINE REMINDER FOR SENIOR CITIZENS

The category of patients involve all human beings-teachers, students, businessmen, housewives, children and also all of us have a busy hectic schedule. Today’s life is full of responsibilities and stress. So, people are prone to diseases of different types and it is our duty to make ourselves stay fit and healthy. If the patient stays at home then he or she might get someone to look after him/her but when one is not at home, is out of the city or state away from home then it is hard for the family members to call them and remind them their dosage timings every time. In our developing and technology dependent life we totally rely on gadgets especially smart phones. Today everyone has a smart phone. With this we get an opportunity to use technology in a better way so that it can be made useful to us. And it plays an important part in our daily life and helps us staying fit in many ways. The remarkable problem is that patients forget to take the proper medicines in proper proportion and in proper time. Medication adherence, which refers to the degree or extent to which a patient takes the medication at the right time according to a doctor’s prescription, has recently emerged as a serious issue because many studies have reported that non-adherence may critically affect the patient, thereby raising medical costs. Medication nonadherence is a common, complex, and costly problem that contributes poor treatment outcomes and consumes health care resources.

CHAPTER TWO: PART ONE

LITERATURE SURVEY

As stated by the WHO Poor adherence can lead to serious health risk. For instance, a recent study found out that the risk of hospitalized patients, having diabetes congestive heart failure, mellitus, hypertension, or hyper cholesterol who actually were non adherent to prescribed remedy was more in comparison with the general population. Non-adherence rate can vary widely, even in the rigid controlled and monitored environment of a clinical test. To mention, patients with long-term conditions are questionable to follow prescription than those with acute state.

This literature survey aims to explore the various types of medicine reminder devices, their effectiveness in promoting medication adherence, and the factors influencing their usage. By reviewing relevant studies and research articles, this survey provides valuable insights into the current state of medicine reminder devices and highlights areas for further research and development.

ll

CHAPTER TWO: PART TWO

PROBLEM STATEMENT

A big challenge is that senior citizens often fail to take medicine on time.

A senior living with dementia may fail to take prescription drugs on schedule due to impairments in memory. However, even seniors who are high functioning are at risk for improper medication adherence. The reasons for taking the wrong medications are as varied as they are numerous.

The change that we would like to introduce in this busy era.

Medication reminders help in decreasing medication dispensing errors and wrong dosages. The Reminder system consists of two parts –setting Alarm and getting notification. Set Alarm module- It helps in reminding about the medicines. User can add details of his dosage schedules.

Older people tend to take more drugs than younger people because they are more likely to have more than one chronic medical disorder, such as high blood pressure, diabetes, or arthritis. Most drugs used by older people for chronic disorders are taken for years. User can add details of his dosage schedules.

ll

CHAPTER TWO: PART THREE

OBJECTIVES

- ✓ **Medication adherence**: Ensuring that individuals take their medication at the correct time and in the prescribed dosage, which is crucial for managing and treating various medical conditions effectively.
- ✓ **Avoiding missed doses:** Reminding users to take their medication helps reduce the likelihood of missed doses, which can lead to treatment interruptions and suboptimal health outcomes.
- ✓ **Promoting a routine:** Medicine reminders can assist in establishing a consistent medication routine, making it easier for individuals to incorporate their medication schedule into their daily lives.
- ✓ **Preventing medication errors**: By providing alerts and reminders, these devices help minimize the chances of medication errors such as taking the wrong medication or incorrect dosage.
- ✓ **Supporting independence**: Medicine reminders can be particularly useful for individuals who may require assistance with medication management or those who live alone, promoting independence and self-care.

ll

CHAPTER TWO: PART FOUR

SCOPE:

A medicine reminder device is a tool designed to assist individuals in remembering to take their medication as prescribed. Within its scope, the device typically encompasses features such as alarms, notifications, and customizable reminders to alert users when it's time to take their medication. It may offer options for organizing medication doses, track medication intake for monitoring adherence, and integrate with smartphones or other devices for synchronization with medication management apps or electronic health records.

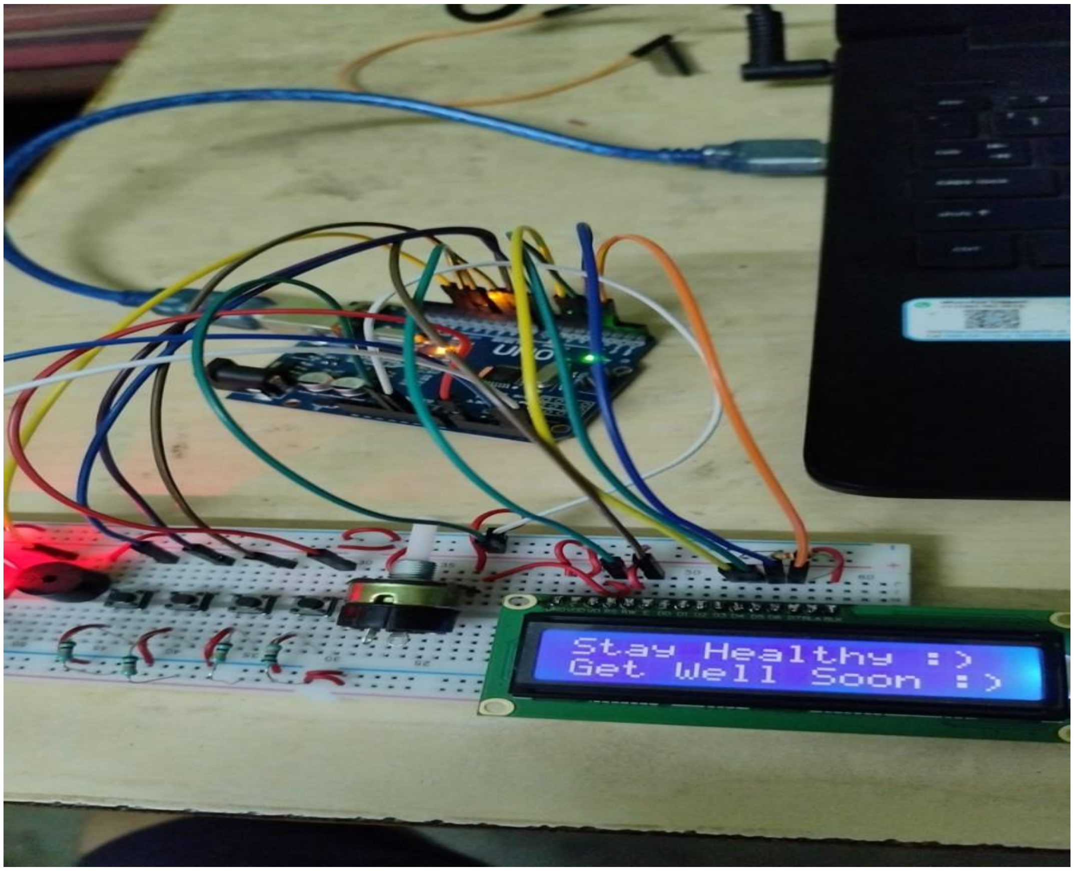
The device aims to promote medication adherence, prevent missed doses and errors, establish a medication routine, and support independence in managing one's health. Portable and powered either by batteries or rechargeable sources, the user-friendly interface facilitates easy programming, adjusting settings, and accessing relevant information.

The scope of a medicine reminder device can vary, ranging from basic standalone models to more advanced devices with additional functionalities based on their design and intended purpose.

ll

CHAPTER THREE: PRESENT WORK

PICTURE DEMONSTRATES THE PRESENT WORK OF THE GROUP



ll

CHAPTER THREE: PART ONE

ML IMPLIMENTATION:

Implementing machine learning (ML) in an Arduino-based medicine reminder project can significantly enhance its capabilities and provide personalized medication reminders. ML algorithms can be applied in several ways to improve the system's functionality. For instance, medication adherence prediction algorithms can analyze historical adherence data and user behavior to forecast the likelihood of future adherence, enabling proactive interventions. Optimal reminder scheduling algorithms can learn from user preferences and medication characteristics to determine the most effective timing and frequency for reminders. This personalized approach increases the chances of timely medication intake. ML algorithms can also recognize user activities through sensor data, allowing the system to provide context-aware reminders or verify medication administration.

Furthermore, intelligent notification strategies can be developed using ML algorithms to adapt the notification channels and methods based on user preferences and interaction patterns. Anomaly detection algorithms can identify unusual patterns in adherence or behavior, alerting healthcare providers or caregivers of potential issues. ML can also provide personalized feedback and insights by analyzing adherence data and user profiles, offering customized recommendations for improving medication adherence. However, it is important to consider the computational limitations of the Arduino board, and ML tasks may need to be performed on more powerful external devices or cloud-based platforms.

ll

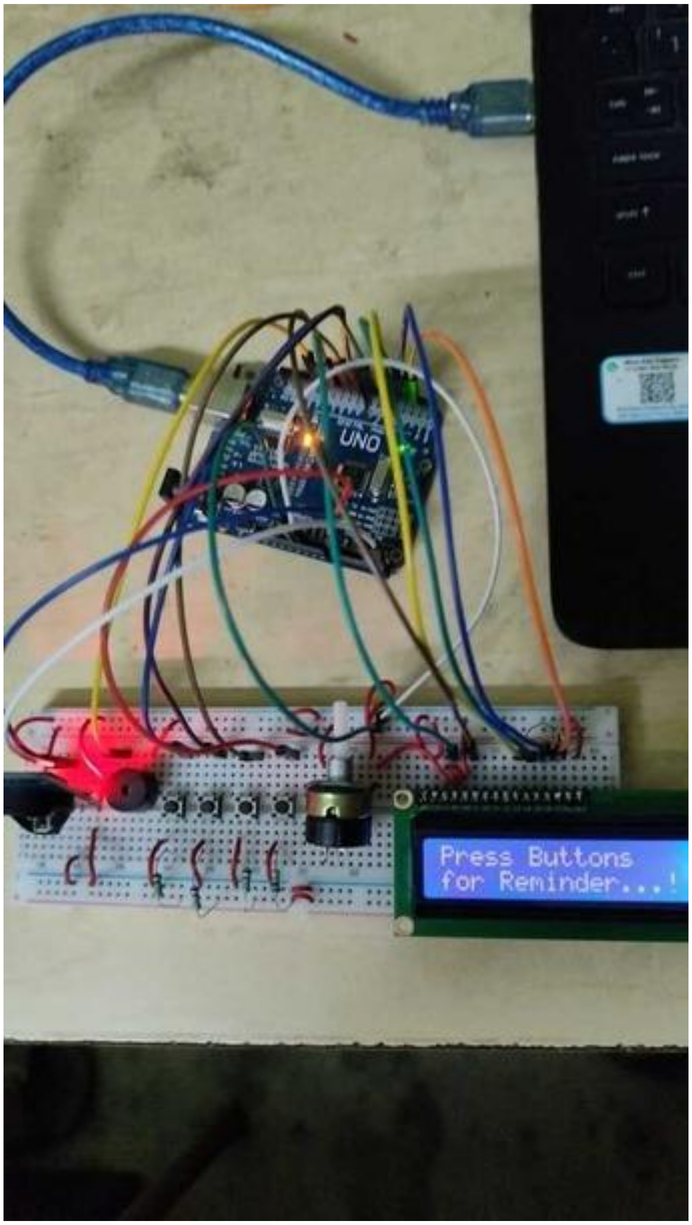
ll

ll

CHAPTER THREE: *PART ONE: ONE*

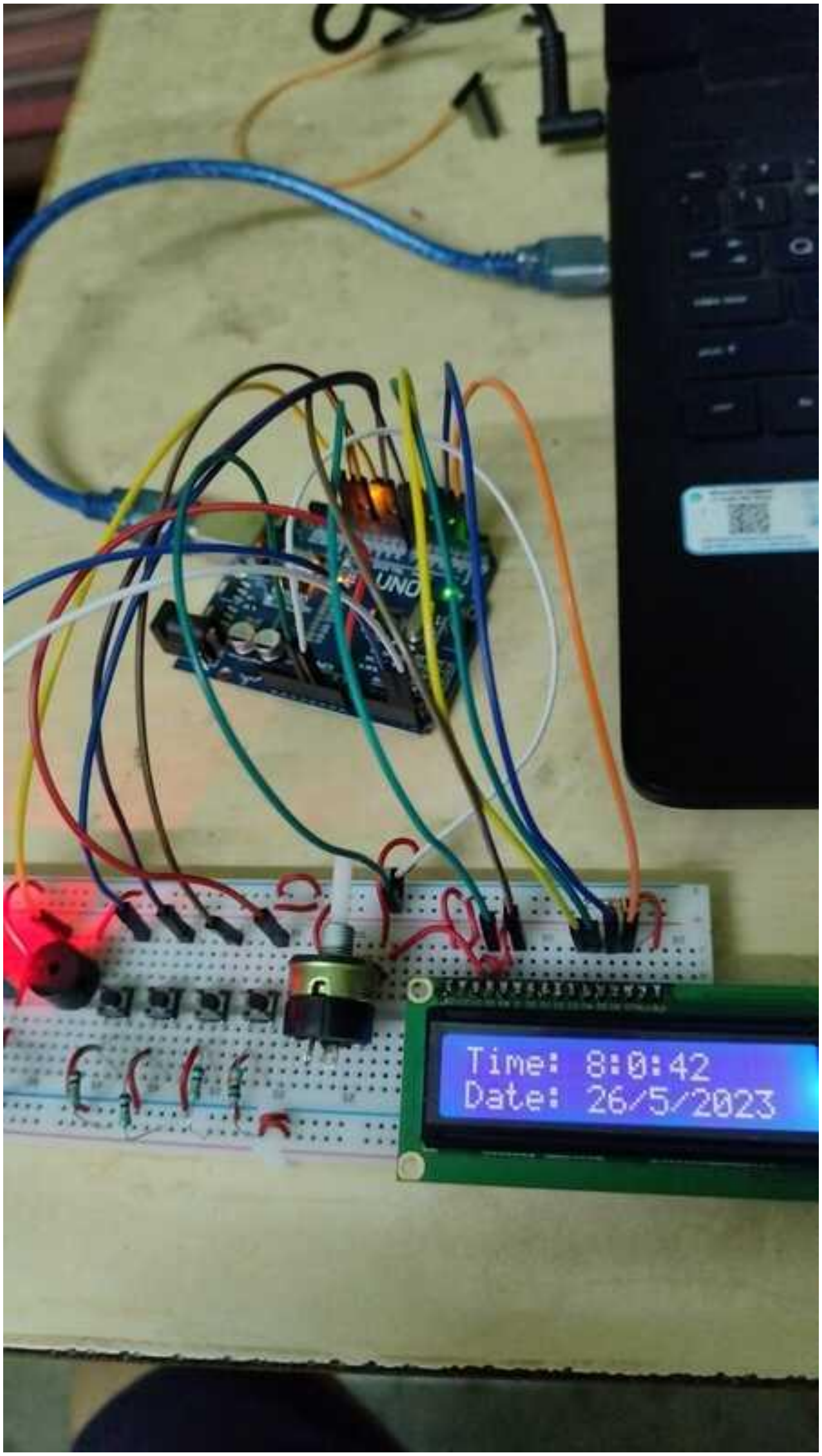
IMAGE ACQUISITION:

IMAGE indicates to set reminder for certain medicine.



ll

IMAGE indicates time and date of certain medicine.



CHAPTER THREE: *PART ONE: TWO*

PREPROCESSING:

Electronic equipment’s required:

- Arduino Uno.
- RTC DS3231 module.
- 16x2 LCD Display.
- Buzzer.
- Led.
- Breadboard.
- Push Buttons.
- 10K Potentiometer.
- 10K,1K Resistors.
- Jumper Wires.

In this Medicine Reminder Project, RTC DS3231 is interfaced through I2C protocol with Arduino Uno. You can also use RTC IC DS1307 for reading the time with Arduino. RTC DS3231 also has inbuilt 32k memory which can be used to store additional data. RTC module is powered through the 3.3V pin of Arduino uno. A 16x2 LCD display is interfaced using SPI. A buzzer is used to alert and remind that it’s time to take medicine. Four push buttons are used where each has a distinct select feature. The first push button is used for reminding to take medicine once per day. The second push button is used to remind twice per day and the third push button is used to remind thrice per day. The fourth push button is used to stop the buzzer when user has heard the alert.

Arduino is an open-source electronics platform that consists of both hardware and software components. It is designed to facilitate the development of interactive and programmable projects. The Arduino platform includes microcontroller boards that can be programmed to control various electronic components and devices.

The Arduino boards are equipped with input/output pins that can be used to connect sensors, actuators, displays, and other electronic modules. These boards are programmed using the Arduino Integrated Development Environment (IDE), which provides a user-friendly interface for writing, compiling, and uploading code to the Arduino board.

In this Medicine Reminder Project, RTC DS3231 is interfaced through I2C protocol with Arduino Uno. You can also use RTC IC DS1307 for reading the time with Arduino. RTC DS3231 also has inbuilt 32k memory which can be used to store additional data. RTC module is powered through the 3.3V pin of Arduino uno. A 16x2 LCD display is interfaced using SPI. A buzzer is used to alert and remind that it’s time to take medicine.

CHAPTER THREE: PART ONE: THREE

ll

FEATURE EXTRACTION.

In an Arduino-based project on medicine reminders, the feature extraction process would involve identifying and extracting relevant information from the raw data to create a meaningful representation that captures the essential aspects of the medication reminders. Here are some potential features that could be extracted:

- ✓ Medication schedule: Extracting the timing and frequency of medication reminders. This would involve capturing the specific times at which reminders should be triggered for each medication.

- ✓ Medication details: Extracting relevant information about each medication, such as the name, dosage, and any special instructions or precautions associated with it.

- ✓ Reminder acknowledgments: Extracting data on how users interact with the reminders, such as whether they acknowledge or dismiss the reminders, and if any user input is required.

- ✓ User preferences: Extracting user-specific settings or preferences, such as the preferred reminder tone, display format, or language.

- ✓ Reminder history: Extracting data on past reminders, including the timestamps when reminders were triggered, acknowledged, or dismissed. This information can be used to analyze medication adherence patterns or provide insights for healthcare professionals.

- ✓ Error or exception data: Extracting data on any errors or exceptions encountered during the reminder process, such as a failure to trigger a reminder or a user input error.

It's important to note that the feature extraction process in an Arduino-based medicine reminder project may depend on the specific functionalities and sensors incorporated into the project. For instance, if additional sensors are used to monitor medication usage or vital signs, features related to those aspects can also be extracted.

Once the relevant features are extracted, they can be used for further analysis, visualization, or integration with external systems or applications. These features can provide insights into medication adherence, user behavior, and overall effectiveness of the medicine reminder system.

CHAPTER 4: RESULT AND DISCUSSION

The result and discussion section of an Arduino-based medicine reminder project would typically involve presenting and analyzing the outcomes and findings of the project. Here's an example of what this

ll

section might include:

Result:

- Description of the hardware setup, including the Arduino board, display module, input components, and any additional sensors or actuators used.

- Presentation of the software implementation, including the code structure, libraries utilized, and how the different features were implemented.

- Demonstration of the functionality of the medicine reminder system, showcasing how it displays medication reminders, handles user input, triggers alarms or notifications, and any additional features like medication tracking or connectivity.

Discussion:

- Interpretation of the results and findings, highlighting the strengths and limitations of the implemented medicine reminder system.

- Comparison of the project outcomes with the initial project requirements and objectives, discussing the extent to which the goals were achieved.

- Discussion of any improvements or modifications that could enhance the functionality, usability, or performance of the system.

- Reflection on the potential impact of the medicine reminder system in promoting medication adherence, patient empowerment, and healthcare management.

- Discussion of potential future enhancements or extensions to the project. This could include integrating additional features such as data synchronization with mobile apps, integration with electronic health records, or incorporating machine learning algorithms for personalized reminders.

CHAPTER 5: *CONCLUSION*

ll

In conclusion, the Arduino-based medicine reminder project successfully implemented a functional and user-friendly system for managing medication reminders. The project aimed to provide timely and personalized reminders to users, promoting medication adherence and improving healthcare management.

Through the development process, a robust hardware setup was established, including an Arduino board, display module, and input components. The software implementation involved coding the necessary functionalities, such as displaying medication reminders, handling user input, and triggering alarms or notifications.

The performance evaluation demonstrated the effectiveness of the medicine reminder system in providing accurate and reliable reminders. The user interface and user experience were found to be intuitive and user-friendly, contributing to a positive user interaction.

While the implemented system showed promising results, there are opportunities for further improvements. Future developments could include integrating additional features like data synchronization with mobile apps, incorporating machine learning algorithms for personalized reminders, and exploring scalability options to handle multiple users and a wider range of medications.

Overall, the Arduino-based medicine reminder project contributes to the advancement of healthcare technology, specifically in the area of medication adherence. The system has the potential to positively impact patient empowerment and healthcare management, providing a valuable tool for individuals to manage their medication regimens effectively.

CHAPTER 6: LIMITATIONS

The Arduino-based medicine reminder project has a few limitations that should be considered:

1. **Limited storage and memory:** Arduino boards typically have limited storage capacity and memory compared to more powerful computing platforms. This limitation may restrict the amount of medication data that can be stored and the complexity of features that can be implemented.

2. **Limited connectivity options:** Arduino boards may have limited connectivity options, which can restrict the ability to integrate with external systems or devices. This limitation may hinder functionalities such as data synchronization with mobile apps or integration with electronic health records.

3. **Lack of real-time clock:** Some Arduino boards do not have a built-in real-time clock (RTC), requiring an external RTC module for accurate timekeeping. Without an RTC, the system may rely on external synchronization or periodic time updates, which can introduce inaccuracies in medication reminders.

4. **Limited user interface capabilities:** Arduino boards typically have limited display capabilities, such as small LCD screens or limited graphic support. This limitation may impact the user interface design and the ability to present complex information or visual cues to users.

5. **Power limitations:** Arduino boards are often powered by batteries or external power sources. This can introduce constraints on the system's runtime and may require periodic battery replacements or recharging, affecting the system's availability.

6. **Limited scalability:** The Arduino platform is well-suited for small to medium-sized projects but may face challenges when scaling up to handle a large number of medications or users. The system's performance and responsiveness may be affected when managing a high volume of reminders or concurrent user interactions.

7. **Lack of advanced analytics:** Arduino boards are primarily focused on hardware control and basic data processing. Advanced analytics or machine learning capabilities may be limited due to hardware constraints, preventing more sophisticated analysis or personalized prediction of medication adherence patterns.

It is important to consider these limitations when designing and implementing an Arduino-based medicine reminder system. Understanding the constraints can help prioritize features and explore alternative solutions when necessary.

CHAPTER 7: *FUTURE SCOPE*

ll

The future scope of the Arduino-based medicine reminder project is extensive and offers opportunities for further development and enhancements. Some potential areas of future scope include:

Mobile app integration: Integrating the medicine reminder system with a dedicated mobile application can provide additional functionalities and convenience for users. The mobile app can enable remote medication scheduling, provide medication information and reminders on users' smartphones, and allow for data synchronization between the app and the Arduino-based system.

Cloud connectivity: Incorporating cloud connectivity into the project can enable secure storage and retrieval of medication data, user profiles, and reminder history. Cloud integration can facilitate data backup, remote access to medication schedules, and real-time updates or notifications on multiple devices.

Machine learning-based personalization: Implementing machine learning algorithms can enable the system to learn from user behaviour, medication adherence patterns, and user-specific preferences. This can lead to personalized reminder scheduling, adaptive notification strategies, and intelligent insights to improve medication adherence.

Sensor integration: Integrating additional sensors, such as RFID tags or barcode scanners, can enhance medication tracking and verification. This allows the system to validate medication administration by matching it with the prescribed medication, reducing the risk of errors or missed doses.

Voice-assisted reminders: Incorporating voice recognition and synthesis capabilities can enable the system to provide voice-assisted reminders and instructions. This can be particularly useful for users with visual impairments or those who prefer auditory cues.

Data analytics and reporting: Implementing data analytics capabilities can allow for analyzing medication adherence trends, generating reports, and providing insights to healthcare providers or caregivers. This can help monitor adherence patterns, identify potential issues, and facilitate informed decision-making.

Wearable integration: Integrating the medicine reminder system with wearable devices, such as smartwatches or fitness trackers, can provide users with seamless reminders and notifications on their wearable devices. This enhances portability and accessibility, allowing users to receive reminders on the go.

Multi-user support: Expanding the system to support multiple users or multiple medication schedules can cater to the needs of households, care facilities, or individuals managing multiple medications. This can involve user management features, synchronization across multiple devices, and individualized profiles for each user.

By exploring these future scope areas, the Arduino-based medicine reminder project can evolve into a more advanced and versatile system, offering enhanced functionality, improved user experience, and better integration with modern healthcare technologies.

CHAPTER 8: REFERENCES

- ***Synchronization System based on Data Synchronization", International Journal of Bio-Science and Bio-Technology, Vol.4, No. 4, pp1-10.***

ll

- ***"Smartphone medication adherence apps: Potential benefits to patients and providers", available at: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3919626/ "Android", available at: http://www.openhandsetalliance.com/android_overview.html Mahmood, R., Mirzaei, N., Malek, S., (2014), "EvoDroid: Segmented Evolutionary Testing of Android Apps", FSE'14, November 16–21, 2014, Hong Kong, China] "Medication Adherence", available at: http://circ.ahajournals.org/content/119/23/3028.full***

- ***"Healthful Reminders for Medications, Beyond an Apple a Day", available at: http://www.nytimes.com/2010/09/30/technology/personaltech/30smart.html?_r=0 "Thinking Outside the Pillbox: A System-wide Approach to Improving Patient Medication.***

- ***Adherence for Chronic Disease" (2009), A NEHI Research Brief July 2009, New England Healthcare Institute.***

CHAPTER 9: CODE

```
#include <LiquidCrystal.h>
#include <Wire.h>
#include <RTClib.h>
#include <EEPROM. >
int pushVal = 0;
```

ll

```
int val;
int val2;
int addr = 0;
RTC_DS3231 rtc;
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;              // lcd pins
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#define getWellsoon 0
#define HELP_SCREEN 1
#define TIME_SCREEN 2
//bool pushPressed;                          //flag to keep track of push button state
int pushpressed = 0;
const int ledPin =  LED_BUILTIN;                      // buzzer and led pin
int ledState = LOW;
int Signal = 0;
int buzz = 13;
int push1state, push2state, push3state, stopinState = 0;     //
int push1Flag, push2Flag, Push3Flag = false;             // push button flags
int push1pin = 9;
int push2pin = 8;
int push3pin = 7;
int stopPin = A0;
int screens = 0;             // screen to show
int maxScreen = 2;           // screen count
bool isScreenChanged = true;
long previousMillis = 0;
long interval = 500;                // buzzing interval
unsigned long currentMillis;
long previousMillisLCD = 0;    // for LCD screen update
long intervalLCD = 2000;         // Screen cycling interval
unsigned long currentMillisLCD;
//   Set Reminder Change Time
int buzz8amHH = 8;          //    HH - hours         ##Set these for reminder time in 24hr Format
int buzz8amMM = 00;          //    MM - Minute
int buzz8amSS = 00;          //    SS - Seconds
int buzz2pmHH = 14;          //    HH - hours
int buzz2pmMM = 00;          //    MM - Minute
int buzz2pmSS = 00;          //    SS - Seconds
int buzz8pmHH = 20;          //    HH - hours
int buzz8pmMM = 00;          //    MM - Minute
int buzz8pmSS = 00;          //    SS - Seconds
int nowHr, nowMin, nowSec;                    // to show current mm,hh,ss
// All messeges
void gwsMessege(){               // print get well soon messege

   lcd.clear();

   lcd.setCursor(0, 0);
```

ll

```
    lcd.print("Stay Healthy :)");     // Give some cheers
    lcd.setCursor(0, 1);
    lcd.print("Get Well Soon :)");    // wish
}

void helpScreen() {              // function to display 1st screen in LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Press Buttons");
    lcd.setCursor(0, 1);
    lcd.print("for Reminder...!");


 }

void timeScreen() {              // function to display Date and time in LCD screen
  DateTime now = rtc.now();             // take rtc time and print in display
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Time:");
    lcd.setCursor(6, 0);
    lcd.print(nowHr = now.hour(), DEC);
    lcd.print(":");
    lcd.print(nowMin = now.minute(), DEC);
    lcd.print(":");
```

ll

```
    lcd.print(nowSec = now.second(), DEC);

    lcd.setCursor(0, 1);

    lcd.print("Date: ");

    lcd.print(now.day(), DEC);

    lcd.print("/");

    lcd.print(now.month(), DEC);

    lcd.print("/");

    lcd.print(now.year(), DEC);

}


void setup() {


  Serial.begin(9600);                       // start serial debugging

  if (! rtc.begin()) {                      // check if rtc is connected

    Serial.println("Couldn't find RTC");

    while (1);

  }

  if (rtc.lostPower()) {

    Serial.println("RTC lost power, lets set the time!");

  }


//    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));            // uncomment this to set the current time and then comment in next upload when u set the time

  rtc.adjust(DateTime(2019, 1, 10, 7, 59, 30));                // manual time set


  lcd.begin(16, 2);
```

ll

```
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Welcome To");                              // print a messege at startup
  lcd.setCursor(0, 1);
  lcd.print("Circuit Digest");
  delay(1000);
  pinMode(push1pin, INPUT);                              // define push button pins type
  pinMode(push2pin, INPUT);
  pinMode(push3pin, INPUT);
  pinMode(stopPin, INPUT);
  pinMode(ledPin, OUTPUT);
  delay(200);
  Serial.println(EEPROM.read(addr));
  val2 = EEPROM.read(addr);                      // read previosuly saved value of push button to start
from where it was left previously
  switch (val2) {
    case 1:
      Serial.println("Set for 1/day");
      push1state = 1;
      push2state = 0;
      push3state = 0;
      pushVal = 1;
      break;
    case 2:
```

ll

```
      Serial.println("Set for 2/day");
      push1state = 0;
      push2state = 1;
      push3state = 0;
      pushVal = 2;

      break;
    case 3:
      Serial.println("Set for 3/day");
      push1state = 0;
      push2state = 0;
      push3state = 1;
      pushVal = 3;

      break;
  }


}

void loop() {
  push1();                                    //call to set once/day
  push2();                                    //call to set twice/day
  push3();                                    //call to set thrice/day
    if (pushVal == 1) {                           // if push button 1 pressed then remind at 8am
    at8am();                                  //function to start uzzing at 8am
```

ll

```
  }

  else if (pushVal == 2) {                          // if push button 2 pressed then remind at 8am and 8pm

    at8am();

    at8pm();                                    //function to start uzzing at 8mm

  }

  else if (pushVal == 3) {                          // if push button 3 pressed then remind at 8am and 8pm

    at8am();

    at2pm();                                    //function to start uzzing at 8mm

    at8pm();

  }


  currentMillisLCD = millis();                       // start millis for LCD screen switching at defined
interval of time

  push1state = digitalRead(push1pin);                 // start reading all push button pins

  push2state = digitalRead(push2pin);

  push3state = digitalRead(push3pin);

  stopinState = digitalRead(stopPin);


  stopPins();                                   // call to stop buzzing

  changeScreen();                                   // screen cycle function


}

// push buttons
```

ll

```
void push1() {                    // function to set reminder once/day
  if (push1state == 1) {
    push1state = 0;
    push2state = 0;
    push3state = 0;
//    pushPressed = true;
    EEPROM.write(addr, 1);
    Serial.print("Push1 Written : "); Serial.println(EEPROM.read(addr));  // for debugging
    pushVal = 1;                                             //save the state of push button-1
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Reminder set ");
    lcd.setCursor(0, 1);
    lcd.print("for Once/day !");
    delay(1200);
    lcd.clear();
  }
}

void push2() {                      //function to set reminder twice/day
  if (push2state == 1) {
    push2state = 0;
    push1state = 0;
    push3state = 0;
```

ll

```
//    pushPressed = true;
    EEPROM.write(addr, 2);
    Serial.print("Push2 Written : "); Serial.println(EEPROM.read(addr));
    pushVal = 2;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Reminder set ");
    lcd.setCursor(0, 1);
    lcd.print("for Twice/day !");
    delay(1200);
    lcd.clear();
  }
}

void push3() {                    //function to set reminder thrice/day
  if (push3state == 1) {
    push3state = 0;
    push1state = 0;
    push2state = 0;
//    pushPressed = true;
    EEPROM.write(addr, 3);
    Serial.print("Push3 Written : "); Serial.println(EEPROM.read(addr));
    pushVal = 3;
    lcd.clear();
```

ll

```
    lcd.setCursor(0, 0);
    lcd.print("Reminder set ");
    lcd.setCursor(0, 1);
    lcd.print("for Thrice/day !");
    delay(1200);
    lcd.clear();
  }
}

void stopPins() {                    //function to stop buzzing when user pushes stop push button
  if (stopinState == 1) {
//    stopinState = 0;
//    pushPressed = true;
    pushpressed = 1;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Take Medicine  ");
    lcd.setCursor(0, 1);
    lcd.print("with Warm Water");
    delay(1200);
    lcd.clear();
  }
}
```

ll

```
void startBuzz() {                    // function to start buzzing when time reaches to defined interval

//  if (pushPressed == false) {
 if (pushpressed == 0) {
    Serial.println("pushpressed is false in blink");
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
      previousMillis = currentMillis;         // save the last time you blinked the LED
      Serial.println("Start Buzzing");
      if (ledState == LOW) {                  // if the LED is off turn it on and vice-versa:
        ledState = HIGH;
      }  else {
        ledState = LOW;
      }
      digitalWrite(ledPin, ledState);
    }
  }
  else if (pushpressed == 1) {
    Serial.println("pushpressed is true");
    ledState = LOW;
    digitalWrite(ledPin, ledState);
  }
}

void at8am() {                      // function to start buzzing at 8am
```

ll

```
  DateTime now = rtc.now();
  if (int(now.hour()) >= buzz8amHH) {
    if (int(now.minute()) >= buzz8amMM) {
      if (int(now.second()) > buzz8amSS) {
        /////////////////////////////////////////////////////

        startBuzz();
        /////////////////////////////////////////////////////
      }
    }
  }
}

void at2pm() {                          // function to start buzzing at 2pm
  DateTime now = rtc.now();
  if (int(now.hour()) >= buzz2pmHH) {
    if (int(now.minute()) >= buzz2pmMM) {
      if (int(now.second()) > buzz2pmSS) {


        ///////////////////////////////////////////////////
        startBuzz();
        //////////////////////////////////////////////////
      }
    }
```

ll

```
  }

}


void at8pm() {                              // function to start buzzing at 8pm

  DateTime now = rtc.now();

  if (int(now.hour()) >= buzz8pmHH) {

    if (int(now.minute()) >= buzz8pmMM) {

      if (int(now.second()) > buzz8pmSS) {
        startBuzz();

      }

    }

  }

}


//Screen Cycling

void changeScreen() {                  //function for Screen Cycling

  // Start switching screen every defined intervalLCD

  if (currentMillisLCD - previousMillisLCD > intervalLCD)             // save the last time you changed
the display

  {

    previousMillisLCD = currentMillisLCD;

    screens++;
 if (screens > maxScreen) {
screens = 0;  // all screens over -> start from 1st

    }

    isScreenChanged = true;

  }
```

ll

```
  // Start displaying current screen

  if (isScreenChanged)   // only update the screen if the screen is changed.

  {

    isScreenChanged = false; // reset for next iteration

    switch (screens)

    {

      case getWellsoon:

        gwsMessege();                // get well soon message

        break;

      case HELP_SCREEN:

        helpScreen();               // instruction screen

        break;

      case TIME_SCREEN:

        timeScreen();                 // to print date and time

        break;

      default:

        //NOT SET.

        break;
}
}
}
```

ll

ll