B.C.A Project Report

"Automated Pill Dispenser"

Submitted by

Gowthami K Shetty (20231BCI0022)
Lavani Sameera Banu (20231BCI0036)
Mangalampeta Harshitha (20231BCI0037)

Under the Guidance of

AKSHAY DILIP PALVE
Assistant Professor
School of Information Science

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

BACHELOR OF COMPUTER APPLICATIONS

SCHOOL OF INFORMATION SCIENCE

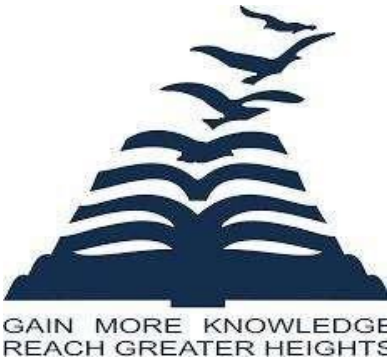
PRESIDENCY UNIVERSITY

BENGALURU

NOV 2025

SCHOOL OF INFORMATION SCIENCE

PRESIDENCY UNIVERSITY



CERTIFICATE

This is to certified that the Project report "Automate Pill Dispenser" being submitted by ***Gowthami K Shetty, Lavani Sameera Banu, Mangalampeta Harshitha*** Fifth semester Presidency University. Bengaluru, as the Internet of Things project in partial fulfilment for the award of BCA Degree course conducted by the Presidency University. The Project Report presented here is the bonafide work of the student.

AKSHAY DILIP PALVE
Assistant Professor,
School of information science
Presidency University

Dr. VETRIMANI ELANGOVAN
Head of the Department,
School of information science
Presidency University

Group Members:

Gowthami K Shetty	20231BCI0022
Lavani Sameera Banu	20231BCI0036
Mangalampeta Harshitha	20231BCI0037

ABSTRACT

This project presents a contactless automatic pill dispenser designed to simplify and automate the process of taking medication. The system is built using an Arduino microcontroller, Infrared (IR) sensor, servo motor, Real-Time Clock (RTC) module, LCD display, and a buzzer.

When a user places their hand near the dispenser, the IR sensor detects the motion and signals the Arduino to activate the servo motor, which rotates and releases one dose of medication. The RTC module displays the current date and time on the LCD screen, while the buzzer provides an audible alert to indicate successful dispensing.

This contactless and hygienic mechanism minimizes the risk of contamination, improves user convenience, and is especially beneficial for elderly or physically challenged individuals. The proposed prototype demonstrates a low-cost and efficient solution for smart healthcare automation, with potential future integration of IoT features such as reminder alerts, dose tracking, and remote monitoring.

Keywords

Automatic pill dispenser, Arduino, IR sensor, servo motor, Real-Time Clock (RTC), LCD display, buzzer, contactless system, smart healthcare, automation

ACKNOWLEDGEMENT

While performing our project, we had to take the help and guidelines of some respected persons who deserve our greatest gratitude. The completion of this project gave us immense pleasure. We would like to express our gratitude to our parents for their kind co-operation and encouragement. While performing our project, we had to take the help and guidelines of some respected persons who deserve our greatest gratitude. The completion of this project gave us immense pleasure. We are highly indebted to HOD and Prof. Akshay Dilip Palve sir, for their guidance, constant supervision and for their support in completing our project.

We would like to express our gratitude to our parents for their kind co-operation and encouragement.

TABLE OF CONTENTS

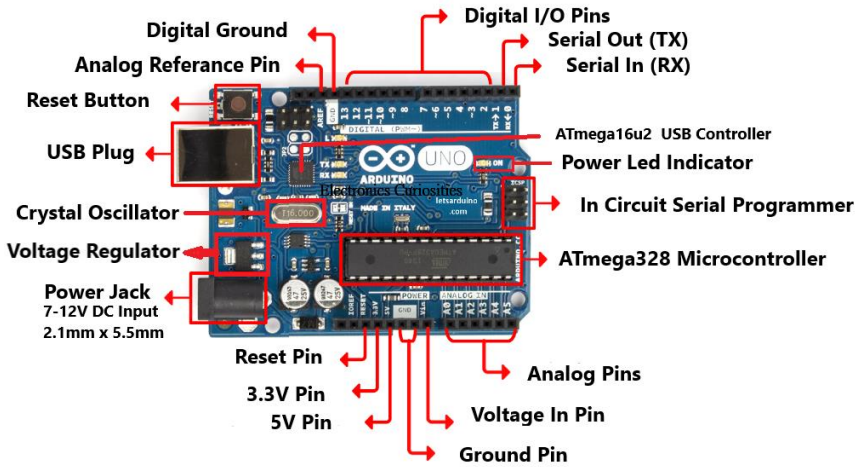
COMPONENTS USED

- Arduino uno
- Power Cable
- Bread Board
- SG90 Servo Motor
- RM065 Trimming Potentiometer
- Liquid Crystal Display (LCD)
- DS3231 Real Time Clock (RTC)
- IR Sensor
- Active Buzzer
- 10kΩ Resistors
- Jumper Wires

FEATURES OF THE COMPONENTS USED:

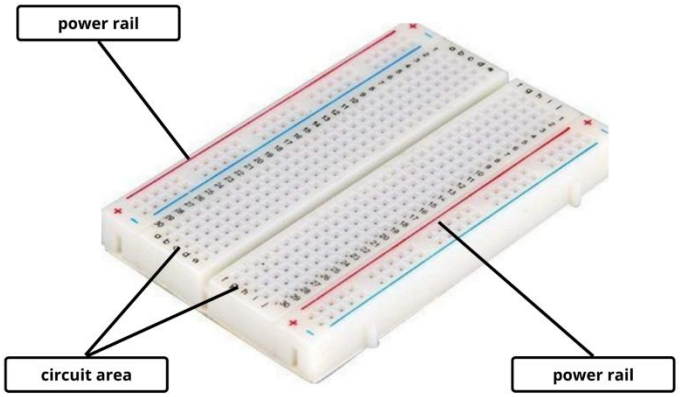
1) Arduino uno

The **Arduino Uno** is a popular, open-source **microcontroller board** based on the ATmega328P. It serves as the "brain" for your electronic projects, executing the code (sketch) you upload to it. Its key features include **14 digital input/output (I/O) pins** (six of which can provide PWM output) and **six analog input pins**, allowing it to interact with a wide range of sensors and actuators. It operates at a clock speed of **16 MHz** and is easily powered and programmed via a USB connection. The Uno is valued for its **simplicity, extensive community support**, and robust design, making it the ideal platform for beginners and rapid prototyping.



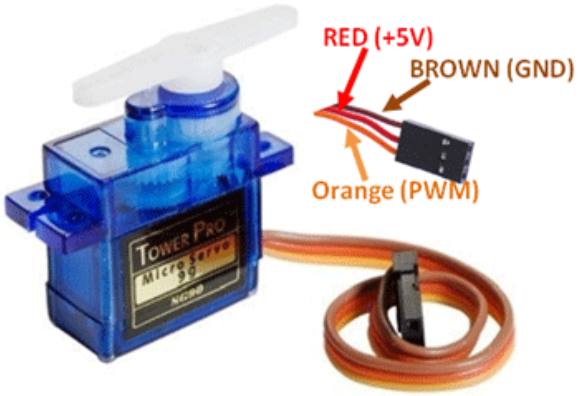
2) Breadboard

A **Breadboard** is a fundamental tool for **solderless circuit prototyping**. It allows you to quickly and temporarily assemble electronic circuits by plugging in components and connecting them with jumper wires. Inside the breadboard, metal strips connect rows of holes, typically in groups of five, while long strips along the sides (called power rails) are used for distributing power and ground. This design enables **easy modification and reuse** of components, making it perfect for experimenting and verifying a circuit design before committing to a permanent solution like a PCB.



3) SG90 Servo Motor

The SG90 Servo Motor is a small, lightweight actuator primarily used for precise angular positioning. Unlike continuous motors, a standard servo can only rotate within a limited range, usually about 180 degrees. Its position is controlled by a Pulse Width Modulation (PWM) signal generated by the Arduino. The motor receives a continuous stream of electrical pulses, and the width of the pulse determines the exact angle it moves to. It's widely used in small robots, camera mounts, and simple mechanical linkages that require controlled, limited movement.

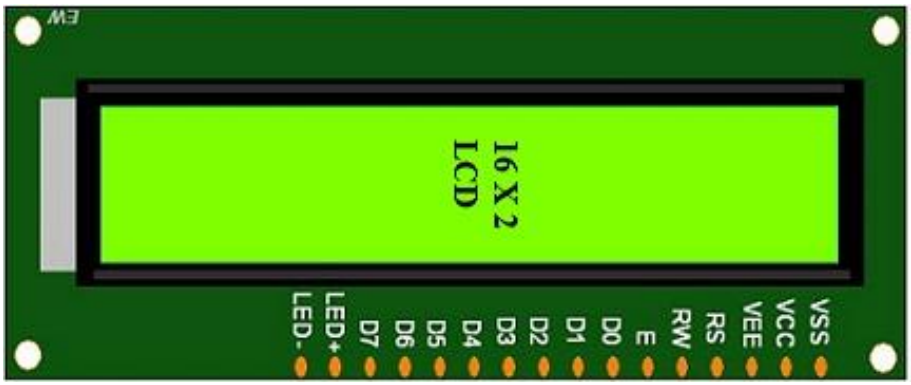


4) RM065 Trimming Potentiometer

The **RM065 Trimming Potentiometer** (often called a trimpot or preset) is a specialized type of **three-terminal variable resistor** designed for **precise, semi-fixed adjustment** of resistance or voltage in an electronic circuit. Unlike the larger, knob-controlled potentiometers used for frequent user input, the RM065 is a miniature component, typically around 6mm, that is usually **mounted directly onto a Printed Circuit Board (PCB)** via its through-hole pins.

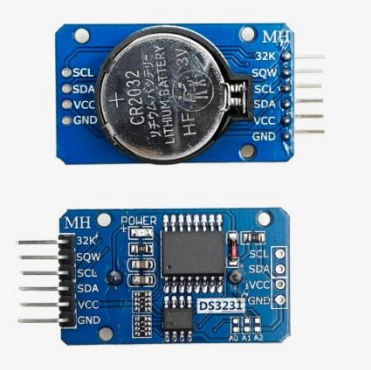
5) Liquid Crystal Display (LCD)

The **Liquid Crystal Display (LCD)** is an **electronic visual display module** commonly used to provide **user feedback** by showing text and, in some cases, simple custom characters. The most common configuration in hobby electronics is the **16x2 LCD**, capable of displaying 16 characters on two separate lines. While it requires several data and control pins for direct connection, it is often paired with an **I2C Serial Adapter**, which simplifies the wiring to just two data pins (SDA and SCL), making it a practical choice for displaying sensor data, time, or system status.



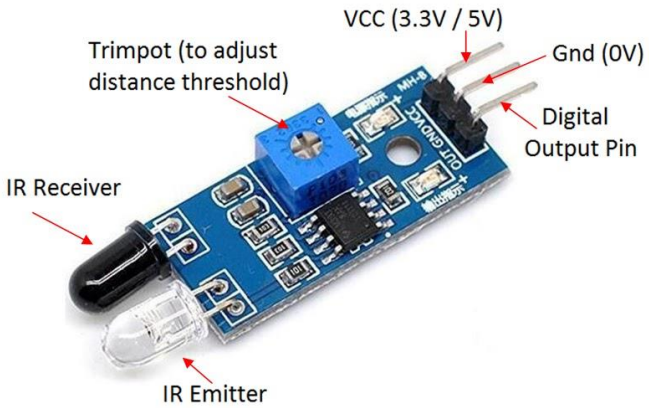
6) DS3231 Real Time Clock (RTC)

The **DS3231** is a highly popular and extremely **accurate Real Time Clock** integrated circuit (IC), commonly found on small modules that interface with microcontrollers like the Arduino via the **I²C communication protocol**. It is generally preferred over older RTC chips like the DS1307 due to its superior precision.



7) IR Sensor

The IR (Infrared) Sensor is a module designed for object detection and distance measurement. It typically operates by sending out a continuous beam of infrared light from an LED (emitter) and measuring the reflected light that returns to a photodiode or phototransistor (receiver). When an object enters the beam's path, more light is reflected, triggering a change in the receiver's output. Most low-cost modules provide a simple digital output (object detected/not detected), making them effective as obstacle detectors in robotics or as basic count/tripwire sensors.



8) Active Buzzer

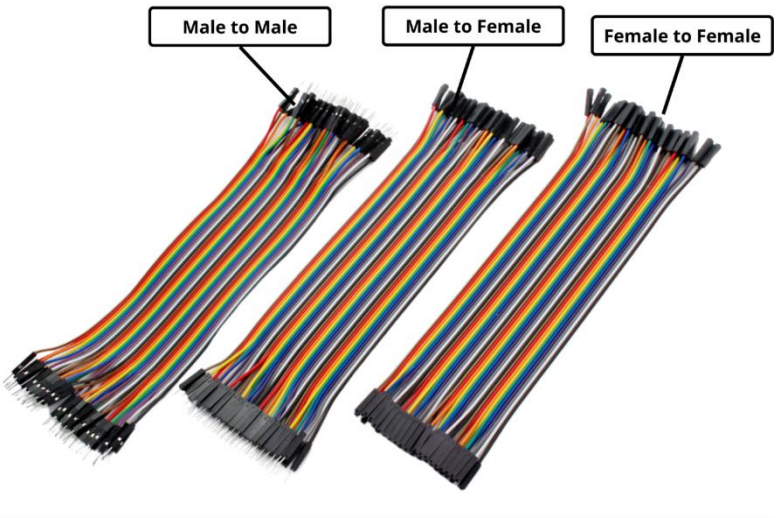
An **Active Buzzer** is a simplified audio signaling device widely used in electronic projects because it contains its **own internal oscillating circuitry**. This key feature means that unlike a passive buzzer, it does not require a complex, frequency-varying signal (like PWM) from the microcontroller to generate sound; it only needs a stable **DC voltage** (typically 3V to 5V) applied across its polarized terminals to turn **ON** and emit a **fixed-frequency, continuous tone**. Because the frequency is set internally, you **cannot change the pitch or create melodies** with a basic active buzzer, but this ease of use makes it perfect for applications requiring simple, immediate **audio feedback** or basic **alarms**, such as indicating a button press, a system error, or a status change.

9) 10kΩ Resistors

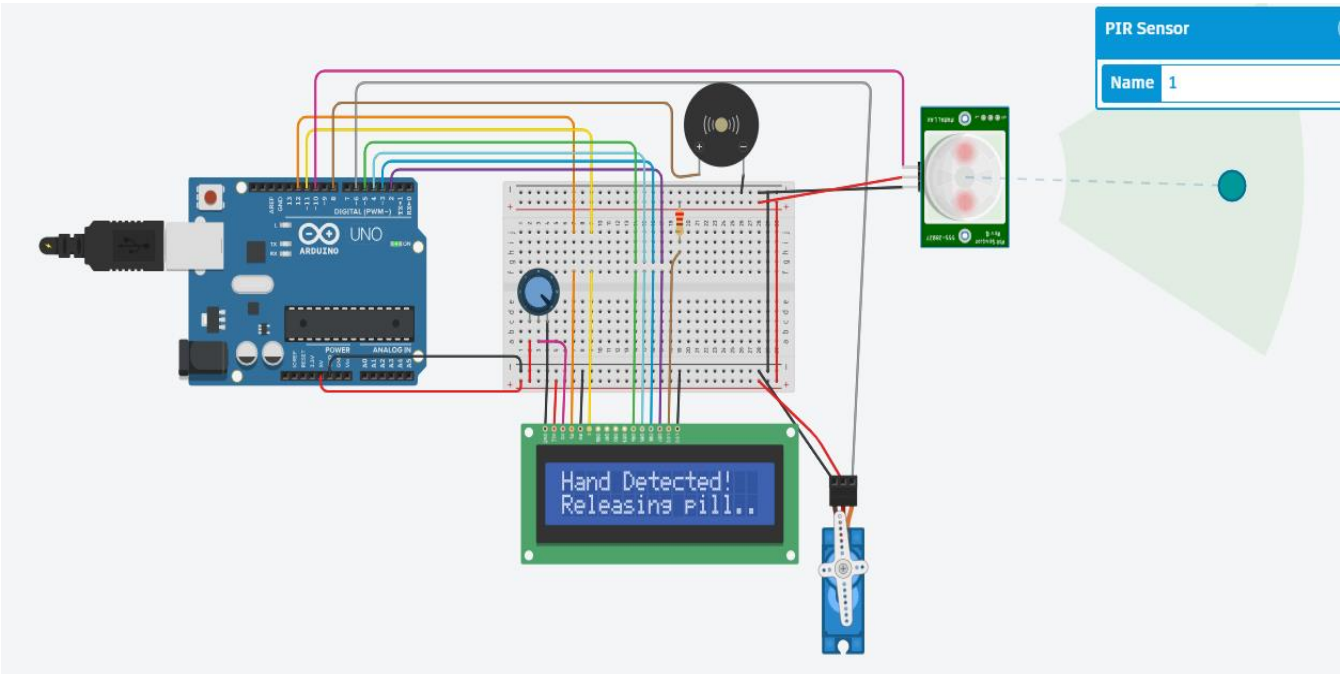
A Resistor is a passive electrical component that opposes the flow of electric current, thereby converting electrical energy into heat. The 10k ohm (10,000 Ω) resistor is a standard value used in many different applications. A key use is for pull-up or pull-down configurations, where they are connected to digital inputs (like switches or buttons) to ensure the pin is held at a definite high (5V) or low (GND) voltage when the button is not pressed, preventing "floating" signals. They are also used to limit current in certain small-signal applications.

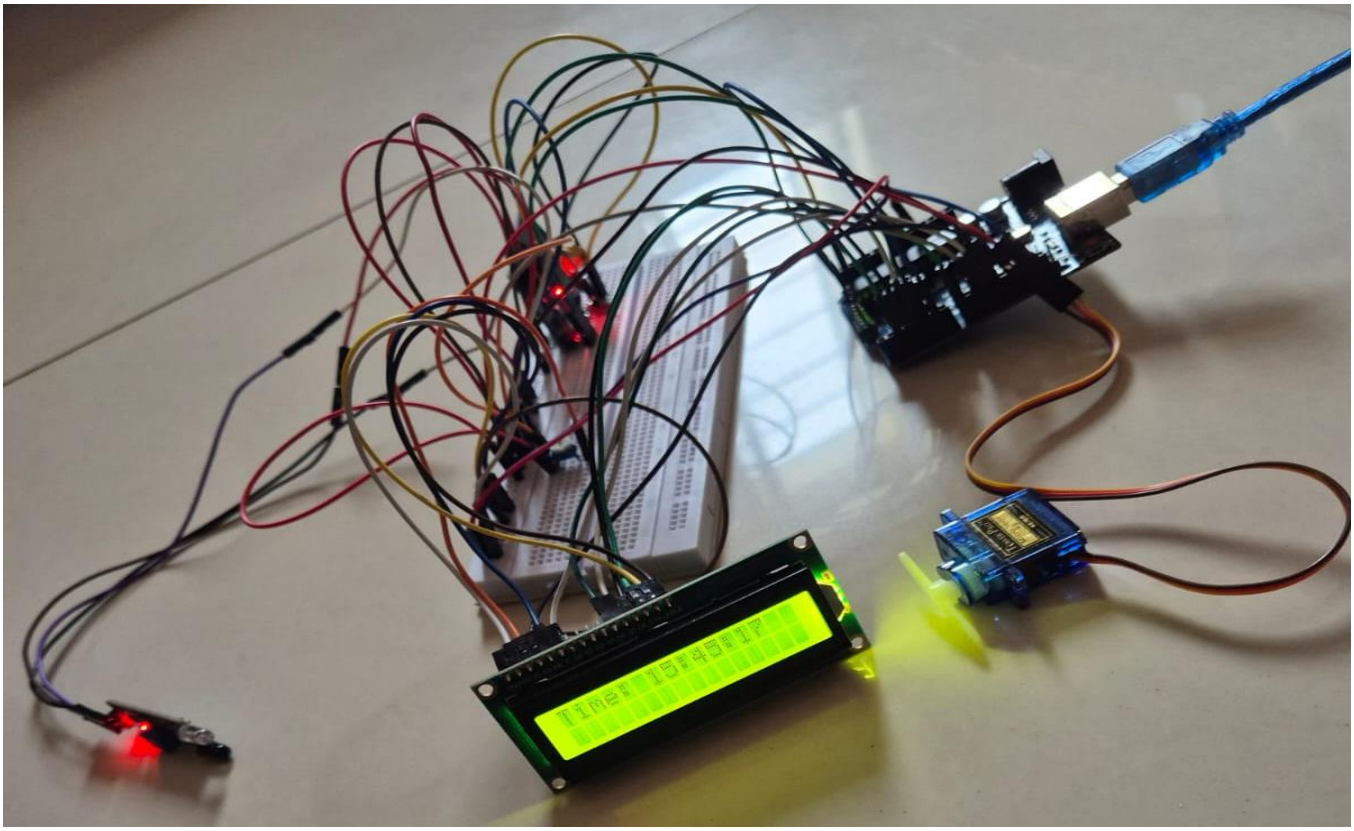
10) Jumper Wires

Jumper Wires are simple but indispensable tools for making temporary electrical connections between components, particularly on a breadboard or between a component and the Arduino. They consist of a flexible wire with a connector terminal at each end. They come in various configurations—Male-to-Male (to connect points on a breadboard), Male-to-Female (to connect a header pin to a component), and Female-to-Female—allowing for versatile, rapid, and non-permanent assembly of circuits during the prototyping phase.



CIRCUIT





MANUAL CONNECTION

CODE

```
#include <Wire.h>
#include <RTClib.h>
#include <LiquidCrystal.h>
#include <Servo.h>

RTC_DS3231 rtc;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
Servo servo;

// --- Pins ---
const int SERVO_PIN = 6;
const int BUZZER_PIN = 8;
const int IR_PIN = 10;

// --- Time slot structure ---
struct TimeSlot {
  int hour;
  int minute;
  int second;
  bool firstAlertDone;
  bool secondAlertDone;
  bool dispensed;
  bool missed;
  unsigned long firstAlertMillis;
  unsigned long secondAlertMillis;
};

TimeSlot slots[3] = {
  {15, 32, 0, false, false, false, false, 0, 0},
  {15, 35, 0, false, false, false, false, 0, 0},
  {15, 37, 0, false, false, false, false, 0, 0}
};
int currentAngle = 0;

// For testing use 20 seconds = 20000 ms
```

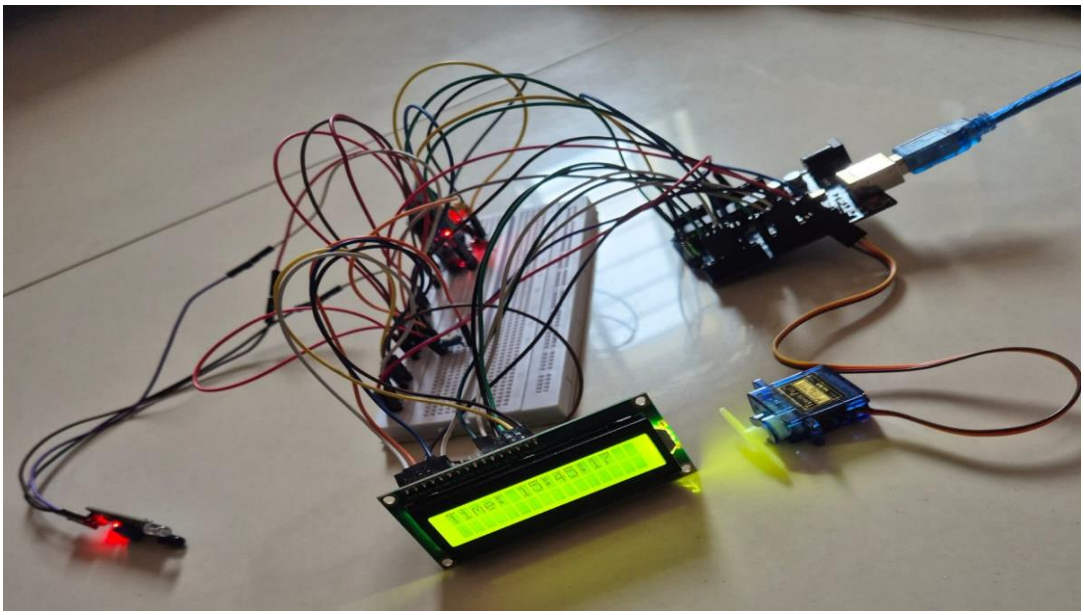
```
unsigned long waitDuration = 6000;   // Change to 1200000 for real 20 minutes

void setup() {
  Serial.begin(9600);
  Wire.begin();
  rtc.begin();

  servo.attach(SERVO_PIN);
  servo.write(0);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(IR_PIN, INPUT);

  lcd.begin(16, 2);
  lcd.clear();
  lcd.print("Smart Pill Box");
  delay(2000);
  lcd.clear();
}
void loop() {
  DateTime now = rtc.now();
  lcd.setCursor(0, 0);
  lcd.print("Time: ");
  printTime(now);

  for (int i = 0; i < 3; i++) {
    handleSlot(slots[i], now, i);
  }
  delay(200);
}
// SLOT LOGIC
void handleSlot(TimeSlot &slot, DateTime now, int slotIndex) {

  // --- FIRST ALERT at scheduled time ---
  if (!slot.firstAlertDone &&
      now.hour() == slot.hour &&
      now.minute() == slot.minute &&
      now.second() >= slot.second) {
```

```
  lcd.clear();
  if (slotIndex == 0) lcd.print("Morning Dose");
  if (slotIndex == 1) lcd.print("Afternoon Dose");
  if (slotIndex == 2) lcd.print("Night Dose");

  Serial.println("FIRST ALERT!");
  alertBuzzer(5);

  lcd.setCursor(0, 1);
  lcd.print("Place your hand");

  slot.firstAlertDone = true;
  slot.firstAlertMillis = millis();
}
// --- WAIT FOR HAND (20 minutes simulated) ---
if (slot.firstAlertDone && !slot.secondAlertDone && !slot.dispensed) {

  // Hand detected
  if (digitalRead(IR_PIN) == LOW) {
    lcd.clear();
    lcd.print("Hand detected");
    dispenseMedicine(slot);
    return;
  }
  // TIMEOUT -> second alert
  if (millis() - slot.firstAlertMillis >= waitDuration) {
    lcd.clear();
    lcd.print("2nd Alert...");
    Serial.println("SECOND ALERT");
    alertBuzzer(5);

    lcd.setCursor(0, 1);
    lcd.print("Place your hand");

    slot.secondAlertDone = true;
    slot.secondAlertMillis = millis();
  }
}
```

```
  // --- WAIT AFTER SECOND ALERT ---
  if (slot.secondAlertDone && !slot.dispensed && !slot.missed) {

    // Hand detected
    if (digitalRead(IR_PIN) == LOW) {
      lcd.clear();
      lcd.print("Hand detected");
      dispenseMedicine(slot);
      return;
    }
    // Timeout -> Missed Dose + Auto Dispense
    if (millis() - slot.secondAlertMillis >= waitDuration) {
      lcd.clear();
      lcd.print("Missed Dose");

      lcd.setCursor(0, 1);
      printTime(now);

      Serial.println("MISSED DOSE - AUTO DISPENSING");
      dispenseMedicine(slot);
      slot.missed = true;
    }
  }
}
// DISPENSE 45°
void dispenseMedicine(TimeSlot &slot) {
  currentAngle += 45;
  if (currentAngle > 180) currentAngle = 180;

  servo.write(currentAngle);
  successBuzzer();

  lcd.clear();
  lcd.print("Medicine Given!");

  Serial.print("Servo moved to: ");
  Serial.println(currentAngle);
```

```
  slot.dispensed = true;
  delay(2000);
  lcd.clear();
}
// PRINT TIME ON LCD
void printTime(DateTime t) {
  lcd.setCursor(6, 0);
  if (t.hour() < 10) lcd.print('0');
  lcd.print(t.hour());
  lcd.print(":");
  if (t.minute() < 10) lcd.print('0');
  lcd.print(t.minute());
  lcd.print(":");
  if (t.second() < 10) lcd.print('0');
  lcd.print(t.second());
}
// BUZZERS
void alertBuzzer(int times) {
  for (int i = 0; i < times; i++) {
    tone(BUZZER_PIN, 1000);
    delay(400);
    noTone(BUZZER_PIN);
    delay(200);
  }
}
void successBuzzer() {
  tone(BUZZER_PIN, 1500);
  delay(300);
  noTone(BUZZER_PIN);
}
```

FINAL ASSEMBLY



FUTURE SCOPE

The core future enhancement involves integrating Wi-Fi connectivity, using a module like the ESP32 or ESP8266, to transform the device into a Smart Healthcare system. This connection enables Remote Scheduling, allowing caregivers to update medication times via a mobile app or web portal, improving flexibility and ensuring the patient is on the correct regimen. Furthermore, this connectivity is vital for implementing Caregiver Alerts, sending immediate notifications (SMS/email) if a dose is missed or if the pill count is low, which significantly enhances patient safety and timely intervention.

The expansion also focuses on leveraging data for better accountability and information. By integrating a Medication Database, the system can display drug information and dosage instructions directly on the LCD at the time of dispensing. While the ESP8266 can handle basic alerts, the ESP32 is the superior platform. Its dual-core processing and greater stability are crucial for managing concurrent tasks securing data transmission to the cloud and ensuring the physical dispensing mechanism remains fast and reliable making it the ideal choice for a safety-critical application.

CONCLUSION

The automatic pill dispenser project successfully achieved its goal of automating medication management and ensuring hygienic dispensing through the integration of several core components on the Arduino Uno platform. The system's design is focused on improving medication adherence and convenience, particularly for vulnerable populations. Key to the system's operation is the IR sensor, which enables contactless activation for a hygienic user experience, minimizing contamination risk. Medication scheduling is precisely managed by the DS3231 Real-Time Clock (RTC), while the SG90 Servo Motor ensures the accurate and controlled release of a single dose. Furthermore, clear communication is maintained via the LCD display and an Active Buzzer, providing both visual and audible alerts to the user about pending doses and system status.

The practical implementation of the prototype demonstrates several key achievements that solidify its value as a smart healthcare solution. The scheduled dispensing functionality addresses the critical challenge of remembering complex medication times. The combined visual and audible user feedback and alerts ensure that the system is accessible and reliable. Crucially, by providing a low-cost, efficient solution, this automated dispenser holds high relevance to healthcare, offering significant support to elderly or physically challenged individuals who often struggle with manual pill management, thus promoting greater independence and safety in their daily lives.