**Project report on**

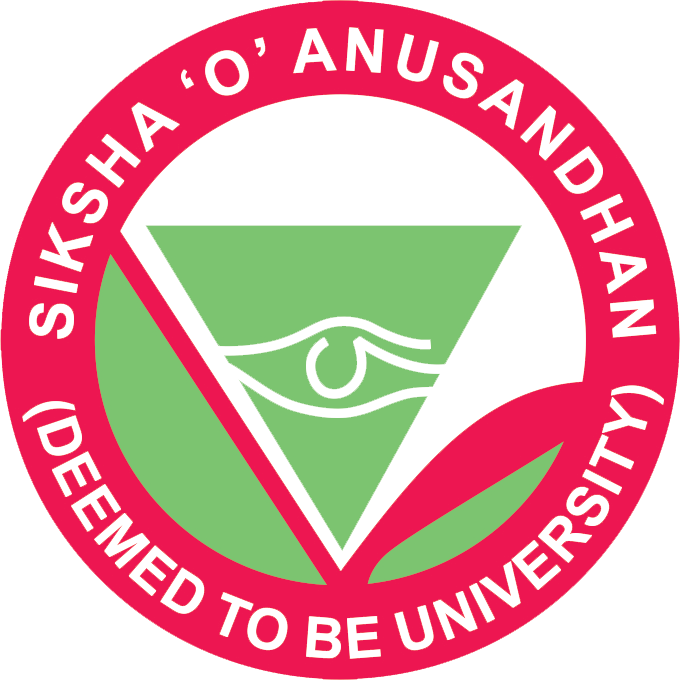
**INNOVATIVE ARDUINO-POWERED AUTOMATIC PILL DISPENSER FOR MEDICATION**

Submitted by

**ANKIT KUMAR (2241011102)**

**SATYAJIT SAHU (2241019222)**

**B. Tech. (CSE(IOT) 5th Semester (Section–2241009)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

Institute of Technical Education and Research

**SIKSHA ‘O’ ANUSANDHAN**

**DEEMED TO BE UNIVERSITY**

Bhubaneswar, Odisha, India.

(Oct, 2024)

# Abstract

# The "Innovative Arduino-Powered Automatic Pill Dispenser for Medication" is designed to assist individuals in managing their medication schedules effectively and efficiently. The system automates the process of dispensing pills, ensuring that users receive the correct dosage at the right time, minimizing human error and enhancing adherence to prescribed treatments.

# The core of the system is built around an Arduino Uno microcontroller, which coordinates the actions of various components. An Infrared (IR) sensor detects the presence of the user or a designated container, triggering the dispensing mechanism. A servo motor, controlled by the Arduino, precisely releases the required number of pills into the container. To alert the user, a buzzer is activated when the pills are ready for collection. The system is programmable, allowing for customization of dispensing times and quantities based on individual needs.

# This project provides a cost-effective, reliable solution for medication management, benefiting elderly individuals, patients with chronic illnesses, and those with busy schedules who require assistance in maintaining their medication regimen. It also offers the potential for further integration with health monitoring systems or mobile applications for remote tracking and notifications.

# Contents

[Abstract i](#_Toc132229941)

[Contents ii](#_Toc132229942)

[Chapter 01: Introduction 1](#_Toc132229943)

[1.1. Introduction 1](#_Toc132229944)

[1.2. Background 1](#_Toc132229945)

[1.3. Project Objectives 1](#_Toc132229946)

[1.4. Scope 1](#_Toc132229947)

[1.5. Project Management 1](#_Toc132229948)

[1.6. Overview and Benefits 2](#_Toc132229949)

[1.7. Organization of the Report 2](#_Toc132229950)

[Chapter 02: Background Review & Survey 3](#_Toc132229951)

[2.1. Related Works 3](#_Toc132229954)

[Chapter 03: Theoretical Aspects 4](#_Toc132229955)

[3.1. Internet of Things (IoT) 4](#_Toc132229958)

[3.2. Features of IoT 4](#_Toc132229959)

[3.3. Advantages of IoT 4](#_Toc132229960)

[3.4. Disadvantages of IoT 4](#_Toc132229961)

[3.5. Application areas of IoT 4](#_Toc132229962)

[3.6. IOT Technologies and Protocols 4](#_Toc132229963)

[3.7. Project Layout 5](#_Toc132229964)

[3.7.1. Brief Description 5](#_Toc132229965)

[Chapter 04: Hardware Requirements 6](#_Toc132229966)

[4.1. Arduino Uno 6](#_Toc132229969)

[4.1.1. Features 6](#_Toc132229970)

[4.1.2. Pin Configuration 6](#_Toc132229971)

[4.2. ESP–01 6](#_Toc132229972)

[4.3. Sensors 6](#_Toc132229973)

[4.3.1. Pressure Sensor 6](#_Toc132229974)

[4.3.2. Temperature Sensor 6](#_Toc132229975)

[4.4. 6](#_Toc132229976)

[4.5. Block diagram of the proposed system 6](#_Toc132229977)

[4.5.1. Working of the system 6](#_Toc132229978)

[4.5.2. Circuit Diagram 6](#_Toc132229979)

[4.5.3. Components Required 6](#_Toc132229980)

[Chapter 05: Software Requirements 8](#_Toc132229981)

[5.1. Arduino IDE (Embedded C / C++) 8](#_Toc132229984)

[5.2. Logic and Flowchart 8](#_Toc132229985)

[Chapter 06: Project development & Testing Aspects 9](#_Toc132229986)

[6.1. 9](#_Toc132229989)

[Chapter 07: Conclusion & Future Scope 10](#_Toc132229990)

[7.1. Result 10](#_Toc132229993)

[7.2. Conclusion 10](#_Toc132229994)

[7.3. Limitations 10](#_Toc132229995)

[7.4. Further Enhancement and Future Scope 10](#_Toc132229996)

[References 11](#_Toc132229997)

[Appendix 01 12](#_Toc132229998)

[A01.1. Code Listing 12](#_Toc132229999)

[A01.2. Main Code 12](#_Toc132230000)

[A01.3. Libraries 12](#_Toc132230001)

[Appendix 02 13](#_Toc132230002)

[A02.1. Project Proposal Form 13](#_Toc132230003)

[A02.2. Project Management 13](#_Toc132230004)

[A02.3. Bill of Material 13](#_Toc132230005)

[Appendix 03 14](#_Toc132230006)

[A03.1. Data Sheets 14](#_Toc132230007)

# Chapter 01: Introduction

## Introduction

Medication adherence is a crucial aspect of effective healthcare, especially for elderly individuals or patients managing chronic conditions. However, missed doses or incorrect administration of medication can lead to serious health complications. The "Innovative Arduino-Powered Automatic Pill Dispenser for Medication" aims to address this challenge by providing an automated solution for dispensing medication at prescribed intervals.

This project leverages the Arduino Uno microcontroller to control a simple yet effective pill dispensing mechanism. An IR sensor detects when the user is present, prompting the system to release the required dosage. A servo motor is used to precisely control the dispensing of pills, ensuring accuracy. Additionally, a buzzer sounds to notify the user when the pills are ready for collection, making it easier for them to adhere to their medication schedule without constant manual supervision.

## Background

Medication non-adherence is a common problem, particularly among elderly patients or those managing multiple prescriptions, leading to serious health complications, hospitalizations, and increased healthcare costs. Traditional methods such as pillboxes and manual reminders are often ineffective, prone to human error, and provide no real-time monitoring or automation. With advancements in technology, there is an increasing demand for smarter healthcare solutions to improve medication adherence and reduce errors. The Innovative Arduino-Powered Automatic Pill Dispenser project addresses this need by developing an automated system that accurately dispenses medication at scheduled times, ensuring timely dosage while minimizing human oversight. Powered by an Arduino microcontroller, this device offers features such as reminders, controlled pill dispensing, aiming to improve patient compliance, safety, and overall health outcomes. By leveraging simple yet powerful technology, this project seeks to offer a reliable and cost-effective solution for medication management.

## Project Objectives

The Innovative Arduino-Powered Automatic Pill Dispenser for Medication is an automated system designed to assist users in managing their medication intake with precision and ease. The core of the system revolves around the Arduino Uno microcontroller, which controls and coordinates the actions of the various components involved in the pill dispensing process.

An IR sensor is employed to detect the presence of the user or a container, triggering the dispenser to release the required dosage of medication. A servo motor, controlled by the Arduino, operates the dispensing mechanism with precision, ensuring that the correct number of pills is dispensed. A buzzer is integrated into the system to alert the user when the pills have been dispensed, making it easier for them to collect and take their medication at the right time.

The project is highly customizable, allowing for programming of dispensing times and quantities based on individual needs. This automated solution minimizes the risk of human error, enhances medication adherence, and offers a reliable, user-friendly tool for patients who require consistent medication management.

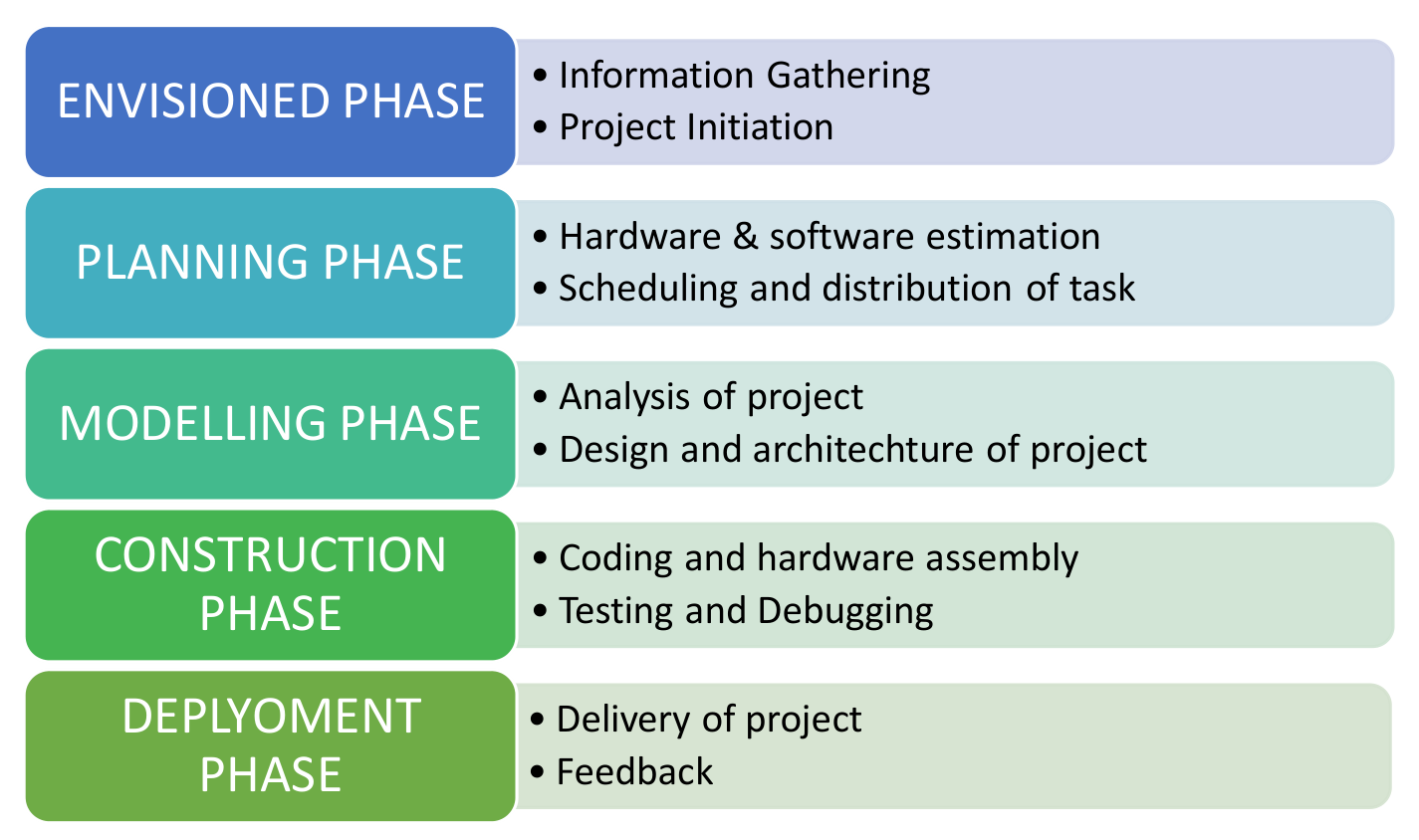
## Scope

Develop the physical structure of the pill dispenser, including compartments for different medications, a dispensing mechanism, and integration of sensors for pill counting and detection

A user-friendly interface can be created for setting medication schedules, which can be through a simple LCD screen, buttons, or even a smartphone application for remote setup and monitoring.

Incorporate safeguards, such as password protection for schedule changes, lockable compartments, and backup power options to ensure functionality even during power failures.

## Project Management



**Figure 1. Model of phases in project management.**

## Overview and Benefits

The "Innovative Arduino-Powered Automatic Pill Dispenser for Medication" is an automated device aimed at simplifying and enhancing the process of medication management. It uses an Arduino Uno microcontroller to automate the dispensing of pills at scheduled times. Key components include an IR sensor, which detects user presence, a servo motor that operates the dispensing mechanism, and a buzzer that alerts the user when the medication is ready.

The system is programmed to dispense the correct amount of medication at the right time, reducing the need for constant manual oversight. The user-friendly design and straightforward functionality make it suitable for individuals who struggle with keeping track of their medication schedules, such as the elderly or patients managing multiple medications.

Benefits :

* Improved Medication Adherence.
* Reduced Risk of Human Error.
* This is customizable and scalable too.
* This project has the potential for further enhancements.

## Organization of the Report

The report is organised into the following chapters. Each chapter is unique on its own and is described with necessary theory to comprehend it.

Chapter 2 deals with background survey and review, Chapter 3 has the description of the theoretical aspects that has been acquired to commence the project work.

# Chapter 02: Background Review & Survey



## Related Works

Medication adherence is a significant challenge in healthcare, particularly for the elderly and patients with chronic conditions who must manage multiple medications. Studies show that non-adherence to prescribed treatments leads to deteriorating health outcomes, hospitalizations, and increased healthcare costs. Traditional methods of managing medication, such as pill boxes or manual reminders, are prone to human error, forgetfulness, and confusion, especially for those with cognitive impairments or busy lifestyles.

In response to these challenges, various automated pill dispensers have been developed, aiming to enhance medication management through technology. Early versions of these devices were bulky, expensive, and often lacked the flexibility to accommodate personalized schedules or multiple medications. Over time, advancements in microcontroller technology, such as Arduino, have enabled the creation of more affordable, customizable, and user-friendly solutions.

**Arduino-Based Solutions:** Recent developments in open-source hardware like Arduino have enabled hobbyists and researchers to create cost-effective, automated pill dispensers. The Arduino Uno microcontroller is widely used due to its ease of programming, affordability, and ability to control various sensors and actuators. Projects using Arduino combine essential features like automated dispensing, user alerts, and sensor-based operations, providing a balance between functionality and simplicity.

**Conclusion:** This project—leveraging an Arduino Uno, IR sensor, buzzer, and servo motor—presents a solution that is simple, affordable, and efficient. It automates the dispensing process, alerts the user via a buzzer, and offers potential customization options. The system not only improves medication adherence but also reduces the risk of human error.

# Chapter 03: Theoretical Aspects



## Internet of Things (IoT)

The **Internet of Things (IoT)** refers to the network of physical objects or "things" embedded with sensors, software, and other technologies that enable them to connect and exchange data over the internet. These objects can range from everyday household items like smart thermostats and wearable fitness devices to complex industrial machinery. IoT allows devices to communicate with each other, collect and share data, and be remotely monitored or controlled, leading to smarter, more automated systems in homes, industries, healthcare, and many other fields.

## Features of IoT

1. **Intelligence :** IoT devices use data processing and decision-making capabilities, often driven by AI, to automate tasks and respond intelligently to changing conditions.
2. **Connectivity :** IoT relies on robust networking technologies to connect devices to the internet, enabling communication between them and with cloud-based services.
3. **Dynamic Nature :** IoT systems continuously adapt to changes in their environment or in user preferences, thanks to real-time data collection and processing
4. **Enormous Scale :** IoT involves a vast number of interconnected devices, requiring scalable infrastructure to manage data, communication, and operations.
5. **Sensing :** IoT devices gather data from the physical world using various sensors (e.g., temperature, motion, humidity) to provide context-aware services.
6. **Heterogeneity :** IoT systems integrate diverse types of devices, platforms, and communication protocols, ensuring compatibility and seamless interaction.
7. **Security :** IoT devices handle sensitive data, so strong encryption, authentication, and privacy measures are essential to protect against cyber threats.

## Advantages of IoT

1. **Communication :** IoT facilitates seamless communication between devices, allowing data to be shared and acted upon in real time.
2. **Automation and Control :** IoT enables devices to automatically perform tasks based on predefined rules or conditions, reducing human intervention.
3. **Information :** IoT provides valuable insights by collecting and analyzing vast amounts of data from sensors, which can be used for decision-making.
4. **Monitoring :** IoT allows for continuous monitoring of systems, environments, and devices, enabling proactive maintenance and real-time alerts.
5. **Efficiency :** IoT optimizes resource usage by automating tasks and providing real-time data, leading to cost savings and improved productivity.

## Disadvantages of IoT

1. **Compatibility** : With many different devices and protocols in use, ensuring compatibility across IoT systems can be challenging.
2. **Complexity :** The integration and management of numerous interconnected devices can introduce technical and operational complexity.
3. **Privacy/Security :** IoT devices often handle sensitive data, making them vulnerable to cyber-attacks, data breaches, and privacy concerns.
4. **Safety :** Misconfigurations, security breaches, or malfunctioning IoT devices can potentially cause physical harm or safety hazards.

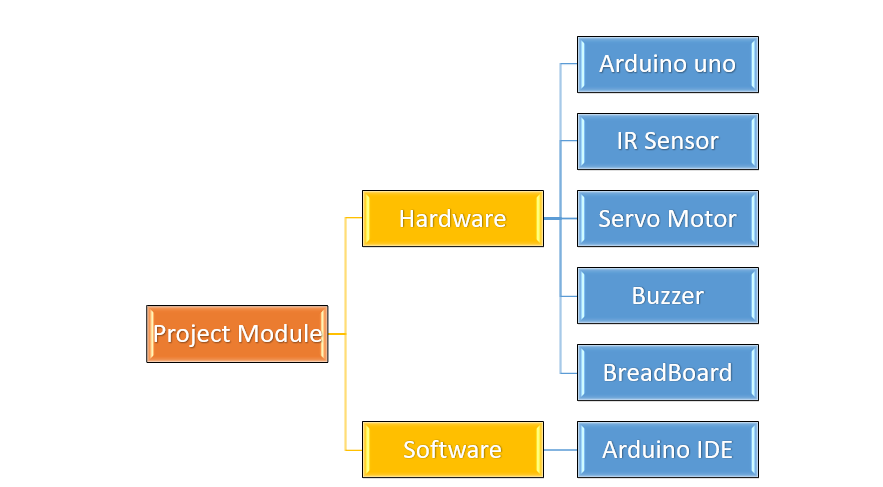
## Application areas of IoT

The Internet of Things (IoT) has vast applications across numerous sectors, enhancing efficiency, automation, and real-time decision-making. In **smart homes**, IoT allows for remote control and automation of household devices, improving convenience and energy efficiency. In **healthcare**, IoT enables remote monitoring of patients through wearable devices, improving patient care and reducing hospital visits. **Industrial automation** benefits from IoT by optimizing production processes and enabling predictive maintenance. In **agriculture**, IoT is used for precision farming, monitoring crop conditions, and automating irrigation, leading to better yields and resource management. **Smart cities** leverage IoT to manage infrastructure like traffic, waste, and energy systems, creating more sustainable urban environments.

## IOT Technologies and Protocols

1. **Bluetooth :** A short-range wireless communication technology used in IoT devices for data transfer over short distances with low power consumption.
2. **Zigbee :** A low-power, low-data-rate wireless mesh network protocol ideal for IoT applications like home automation and sensor networks.
3. **Z-Wave** : Another low-power wireless communication protocol used primarily for smart home devices, offering longer range than Bluetooth.
4. **Wi-Fi :** A widely used wireless technology providing high-speed internet connectivity to IoT devices, typically in local networks.
5. **Cellular :** Mobile networks (e.g., 4G, 5G) used to connect IoT devices to the internet over long distances, especially in smart city or vehicular applications.
6. **NFC :** A short-range wireless technology enabling IoT devices to communicate when in close proximity, commonly used in contactless payments.
7. **LoRaWAN :** A low-power, long-range wireless communication protocol designed for IoT applications requiring wide-area coverage, like smart cities and agriculture**.**

## Project Layout



**Figure 2. Layout of project module**

### Brief Description

# The "Innovative Arduino-Powered Automatic Pill Dispenser for Medication" is a smart healthcare solution designed to assist patients in managing their medication schedules effectively. This device, powered by an Arduino microcontroller, automates the dispensing of pills at pre-set times, ensuring accurate dosages and preventing missed doses. The system features a user-friendly interface for setting medication schedules, one can put his hands near dispenser and the medicine will automatically drop from the machine. This project aims to improve patient compliance, enhance safety, and reduce the risk of medication errors, especially for individuals with chronic conditions or memory impairments.

# Chapter 04: Hardware Requirements



## Arduino Uno

The **Arduino Uno** is a popular open-source microcontroller board based on the ATmega328P microcontroller. It is designed for beginners and professionals to build electronic projects. The board features 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection for programming, a power jack, and a reset button. The Arduino Uno can be programmed using the Arduino IDE (Integrated Development Environment) and is widely used for prototyping in IoT, robotics, and various electronics projects due to its simplicity and versatility.

### Features

The Arduino Uno is a versatile microcontroller board known for its simplicity and ease of use. It features 14 digital input/output pins, 6 of which can be used for PWM (pulse-width modulation) output, and 6 analog input pins for reading sensors. Powered by the ATmega328P microcontroller, it operates at a clock speed of 16 MHz. The board includes a USB port for programming, a power jack, and an ICSP (In-Circuit Serial Programming) header for direct firmware uploads. It supports a variety of communication protocols like UART, SPI, and I2C, making it suitable for a wide range of projects. Additionally, the board can be powered via USB or an external power supply, providing flexibility for different applications. Its open-source design and large community support make it a popular choice for beginners and professionals in electronics, IoT, and robotics projects.

### Pin Configuration :

The provided Arduino code controls a servo motor and a buzzer in response to an input signal, simulating a pill dispenser mechanism. The program begins by setting up the input and output pins: one pin is used to read an external signal (such as a button press or sensor input), another pin controls the buzzer, and a third pin operates the servo motor. In the setup function, the servo is initially set to 0 degrees (closed position) for one second and then detached to save power. In the main loop, the code continuously checks the state of the input signal. If the signal is detected (button press or low input), the buzzer is activated for one second to provide an audible alert. After the buzzer stops, the servo is attached, rotates to 70 degrees to release a pill, pauses for 3 seconds, and then returns to 0 degrees to close the dispenser. The servo is then detached again. This process repeats each time the signal is detected, providing both an audible warning and a mechanical action to dispense a pill.

## Sensor

**IR proximity sensor** is a device that uses infrared light to detect objects or measure the distance between the sensor and an object without physical contact. It works by emitting infrared light from an LED, which reflects off nearby objects, and a photodetector receives the reflected light. The sensor determines the presence or proximity of an object based on the amount of reflected light detected.

### 

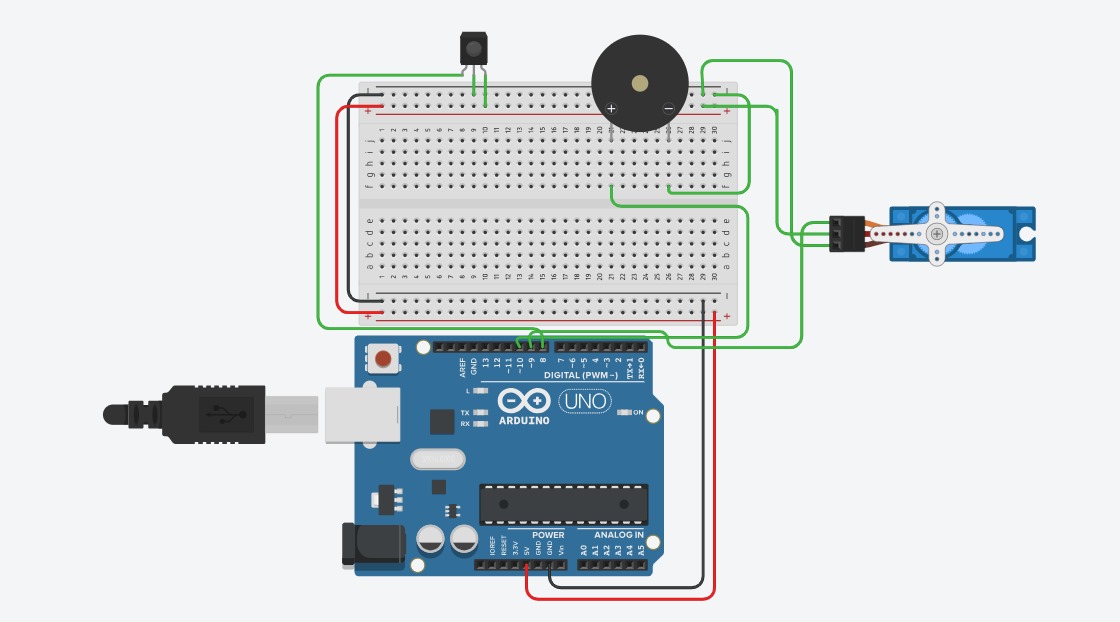
## Block diagram of the proposed system



### Working of the system

The working of the system is very simple i.e. user have to bring his hand in the gap below the arrow, the IR Sensor will detect his hand and the Servo motor will move to 70 degree and the mechanism attached to the Servo will throw out a pill.

### Circuit Diagram



### Components Required

**Table 1. Component listing.**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Component and Specification** | **Quantity** |
|  | Arduino Uno | 1 |
|  | Servo Motor | 1 |
|  | I.R. Sensor | 1 |
|  | Bread Board | 1 |
|  | Buzzer | 1 |
|  | Jumper wires |  |

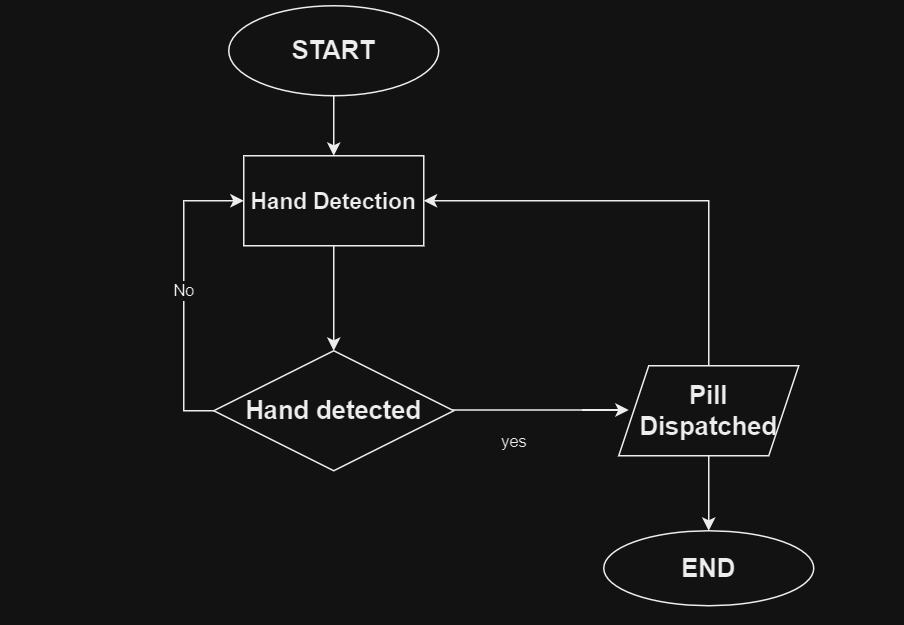
# Chapter 05: Software Requirements



## Arduino IDE (Embedded C / C++)

The Arduino Integrated Development Environment (IDE) is a user-friendly platform designed for programming Arduino microcontrollers using a simplified version of C/C++. It provides a code editor, compiler, and upload functionality, allowing users to write and test code for various hardware projects. The IDE supports a rich library of pre-written code and examples, making it accessible for beginners while also offering advanced features for experienced developers. With built-in functions tailored to hardware interactions, such as digital and analog input/output, the Arduino IDE streamlines the process of creating embedded systems, enabling users to rapidly prototype and develop innovative electronic applications.

## Logic and Flowchart



# Chapter 06: Project development & Testing Aspects



## The development and testing of the \*\*Innovative Arduino-Powered Automatic Pill Dispenser for Medication\*\* project involves several key aspects. The primary component, an Arduino Uno, serves as the microcontroller that orchestrates the entire system. An infrared (IR) sensor is integrated to detect when a user is present, ensuring the dispenser operates only when needed. The servo motor is crucial for dispensing the pills accurately; it is programmed to rotate to 70 degree of angle to release a predetermined quantity of medication. During the development phase, coding in the Arduino IDE focuses on integrating the IR sensor input and controlling the servo motor's actions based on that input. Testing involves evaluating the responsiveness of the IR sensor and ensuring the servo motor dispenses the correct number of pills(1), as well as validating the reliability and accuracy of the entire system. It is also essential to test under various scenarios, such as different lighting conditions and distances from the sensor, to ensure robust performance in real-world applications. Additionally, user feedback can be collected to improve usability and ensure that the device meets the needs of individuals requiring medication management.

# Chapter 07: Conclusion & Future Scope



## Result

The **Innovative Arduino-Powered Automatic Pill Dispenser for Medication** successfully demonstrates the effective integration of an Arduino Uno, an IR sensor, and a servo motor. The system accurately detects user presence through the IR sensor and subsequently activates the servo motor to dispense the specified number of pills. Testing results indicate that the dispenser operates reliably under varying conditions, providing a consistent and user-friendly experience. Feedback from initial users suggests that the device meets its primary goal of enhancing medication management, particularly for individuals with strict medication schedules.

## Conclusion

In conclusion, the project showcases a practical application of Arduino technology in addressing a critical need in healthcare: medication adherence. The automatic pill dispenser not only streamlines the process of medication intake but also reduces the risk of missed doses. By utilizing accessible components like the Arduino Uno, IR sensor, and servo motor, the project illustrates how innovative solutions can be developed to improve everyday health management. Overall, this device represents a significant step toward automated healthcare solutions.

## Limitations

Despite its success, the project has certain limitations. The system's dependency on the IR sensor means it may struggle in environments with fluctuating lighting conditions or obstructions that could hinder sensor performance. Additionally, the current design does not account for varying pill sizes, which may affect the dispensing mechanism's accuracy. The device's manual loading of medication also poses a limitation, as it requires human intervention for refilling. Moreover, the lack of a user interface limits the ability to customize dosage settings, which could enhance usability for different individuals.

## Further Enhancement and Future Scope

Future enhancements for the **Automatic Pill Dispenser** could include the integration of a more sophisticated user interface, such as an LCD screen, to display medication schedules and notifications. Implementing a wireless connectivity feature, such as Bluetooth or Wi-Fi, could allow users to track their medication intake via a smartphone app. Additionally, incorporating a load cell could enable the dispenser to weigh pills and ensure accurate dosing, accommodating different pill sizes. Exploring the use of rechargeable batteries and a more compact design could also improve the device's portability and convenience. Ultimately, the project has significant potential for further development, with the goal of creating a comprehensive medication management system that enhances patient compliance and promotes better health outcomes.

# References

https://www.youtube.com/watch?v=Z5GADxTa1sA&list=PLUWZpHJMEu-Dv5Va5AmPeZH-E8-v9038u&index=65

https://www.viralsciencecreativity.com/post/arduino-automatic-candy-dispenser-machine

# Appendix 01

## A01.1. Main Code

#include <Servo.h>

Servo servo;

int pos;

int Signal = 8;

int servopin = 9;

int buzzer = 10; // Define the buzzer pin

void setup() {

pinMode(Signal, INPUT);

pinMode(buzzer, OUTPUT); // Set buzzer pin as output

servo.attach(servopin);

servo.write(0); // Close cap on power on

delay(1000);

servo.detach();

}

void loop() {

int buttonState = digitalRead(Signal);

delay(1);

if (buttonState == 0)

{

// Activate buzzer before dispensing pill

digitalWrite(buzzer, HIGH); // Turn on the buzzer

delay(1000); // Buzz for 1 second

digitalWrite(buzzer, LOW); // Turn off the buzzer

// Start servo operation

servo.attach(servopin);

delay(1);

servo.write(70); // Rotate servo to 70 degrees

delay(3000); // Wait 3 seconds

servo.write(0); // Rotate servo back to 0 degrees

delay(1000);

servo.detach();

}

}

## A01.2. Libraries

The required Libraries installation are : Servo.

# Appendix 02

## A02.1. Project Proposal Form

The project proposal form was prepared and duly signed from our Faculty-in-Charge Dr. Biswaranjan Swain. The same is attached at the last of this report.

## A02.2. Project Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Component** | **Individual Contributions in %** | | **Total** |
| **ANKIT KUMAR** | **SATYAJIT SAHU** |  |
|  | Planning | 50% | 50% | 100% |
|  | Background Research and Analysis | 50% | 50% | 100% |
|  | Hardware design | 50% | 50% | 100% |
|  | Software design | 50% | 50% | 100% |
|  | Testing | 50% | 50% | 100% |
|  | Final Assembling | 50% | 50% | 100% |
|  | Project report writing | 50% | 50% | 100% |
|  | Presentation | 50% | 50% | 100% |
|  | Logistics | 50% | 50% | 100% |

## A02.3. Bill of Material

**Table 1. Component listing.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Component** | **Specification** | **Unit Cost** | **Quantity** | **Total** |
|  | Arduino | Uno ATmega328P | 500 | 1 | 500 |
|  | IR Sensor | 3-5V DC supply | 50 | 1 | 50 |
|  | Servo Motor | SG-90 | 130 | 1 | 130 |
|  | Buzzer | 3.3-5v passive | 15 | 1 | 15 |
|  | Jumper wires | M-M | 80 | 1 | 80 |
|  | Jumper wires | F-M | 80 | 1 | 80 |
|  | Bread Board | Full | 90 | 1 | 90 |
| **Grand Total** | | | | | 945 Rs |

# Appendix 03

## A03.1. Data Sheets

Arduino Uno :

**Dimensions** 68.6 mm x 53.4 mm **Weight** 25g

**Power Supply** Type-B USB, Barrel jack

**Features**  **Operating Voltage**: 5V

**Input Voltage**: 7-12V

**Pins**: 14 Digital pins(6 PWM pins), 6 Analog pins

**Clock Speed**: 16 MHz

**Pin Configuration** **Digital Pins**: D0 to D13, **Analog Pins**: A0 to A5

**Power Pins**: 5V, 3.3V, GND, Vin.

**Communication** **Serial Communication**: UART (TTL)

**SPI Communication**: Pins 10-13

**I2C Communication**: Pins A4 (SDA), A5 (SCL)

**Programming** **IDE**: Arduino IDE (supports C/C++)

**Website** https://www.arduino.cc/