**ABSTRACT**

Nowadays health sector has developed based on wireless sensing node technology. Paying attention for each patient on time is very paramount in any hospital. Therefore, we intend to aid that problem by bringing in an automatic control system which will provide medication on time for every patient, updating his/her medical background and whenever the doctors prescribe medicines or scans for them that will automatically inform relevant medical staff. This device uses IR and ToF sensors for tracking both pills and syrup levels. Other than that condition of the container measured via temperature and humidity sensors and indicate when there is an unnecessary environment for pills and syrups also. LCD display shows patients details and the number of pills and syrups he/she consume for a given time. That will help doctors to track the medical condition of the patient without any doubt whether they took medicines on time or not. Along with IoT we intend to implement this device with low power consumption. For that we intend to use some low power communication protocols that are not listening to servers in every time. Such as MQTT application layer protocol, IEEE 802.11ah -Wifi protocol. This device collects sensor data and send them to server via ESP 8266 wifi module using Mosquitto MQTT broker with the help of Publish Subscribe architecture.

**TABLE OF CONTENT**

Page No.

Abstract …………………………………………………………………………

1. Chapter 01 – Introduction
   1. Overview…………………………………………………………….
   2. Objectives……………………………………………………………
   3. Methodology…………………………………………………………
2. Chapter 02 – Hardware implementation
   1. Introduction………………………………………………………….
   2. Hardware components………………………………………………
   3. Power………………………………………………………………..
   4. Schematics and PCB layouts………………………………………..
3. Chapter 03 – Mechanical design
   1. Introduction…………………………………………………………
   2. Enclosure design……………………………………………………
4. Chapter 04 – Software implementation
   1. Introduction…………………………………………………………
   2. Software flow……………………………………………………….
   3. Protocols…………………………………………………………….
   4. User interfaces………………………………………………………
5. Chapter 05 – Conclusion
6. Chapter 06 – Risks and Challenges
7. Chapter 07 – References

**C****HAP****TER 01 – INTRODUCTION**

**1.1 Overview**

Due to the increasing number of patients in the hospitals, it is difficult for the nursing staff of the hospitals to maintain it properly and efficiently. For this reason, the chance of a single attention on a patient is minimized. Also, the disease of the patients in the ward and the prescriptions prescribed by the doctors are different from each other. As a result, nursing staff may be unable to provide medication on time and paying attention to their medical records. This inefficiency can be caused to make the patient's condition worse. During such a pandemic, it is difficult for the nursing staff to visit and treat the patient and check whether the corona patient received the medication on time frequently. As a solution for that problem, we decided to implement the following solution using IoT.

**1.2 Objectives**

We proposed to create a medicine container with automatic control capability which provide timely medication with alarm, patient’s updated details by doctors and nurses, medication prescribed by the doctor and details of diagnosis reports. This eliminates the need of nursing staff to give different medications from patient to patient on time, as well as to memorize patient medical records and diagnostic data. Also, it has the ability to deliver not only tablets but also syrups in the required quantity at the right time automatically. Once the doctor prescribe medication the system will update current details of the patient. According to that, the nurses fill out the medication and include the time periods for which it should be given to the patients. An alarm will sound when the correct time is reached, and the available medicine in the device will be released.

**1.3 Methodology**

We have created an IoT based system which can solve all of the above problems.

![Diagram

Description automatically generated]()

**CHAPTER 02 – HARDWARE IMPLEMENTATION**

**2.1 Introduction**

When designing hardware, size of the device, user safety, user-friendly environment, and device cost are the main factors we need to consider. We intend to implement a device that can be mounted near to the patient. Then patient can easily take his medicine without any difficulty. Therefore, the device must ensure the patient’s safety. For that it must have a good electrical system and mechanical system. ///////////////////Isuru

**2.2 Hardware components**

Components we used:

* Microcontroller Unit – Arduino Mega 2560 Board
* A NodeMCU development kit with ESP8266 Wi-Fi SoC is used to implement this system.
* VL53L0X Time of Flight sensors used to measure the liquid level.
* DHT22 Humidity and temperature sensor used to measure the environmental condition of the container.
* FUXU NE555 IR emitter and transmitter used to count pills.
* LCD 20x4 display with I2C module, used to display all the necessary information of the patients.
* Buzzer is being used to inform the patients when the medication time occurred.
* SG 5010 servo motors used to achieve the accurate angular or linear position and for specific velocity and

acceleration.

We intend to use Arduino Mega 2560 board with ESP8266 Wi-Fi module to control and process all commands, and the reliability of the product can be enhanced by using a Raspberry Pi board. The PWM and Digital pins are used for servos and sensor connections respectively.

**Sensor Panel & Actuators**

* A picture containing text, electronics, circuit

  Description automatically generatedIR receiver & Transmitter

FUXU NE555 IR receiver and transmitter is used to determine the number of pills using an open loop controlling technique. It is important to measure it accurately. This is achieved by placing the IR transmitter and receiver pointing at each other. If a pill falls through the IR beam, it will block the transmitted signal which means that a pill has fallen through it.

* A picture containing electronics

  Description automatically generatedHumidity and temperature sensor

The quality of some medicines depends on the humidity and temperature of the environment. Therefore, it is essential to keep the humidity and temperature in the container at a favorable level. DHT22 module is used to measure both humidity and temperature using a capacitive humidity sensor and a thermistor with an accuracy of ±2% and ±0.5ºC respectively.

* A close-up of a microchip

  Description automatically generated with low confidenceToF sensor

The doctor prescribes not only tablets but also syrups to the patient. We have designed this device that can give the patient even two types of syrups. We mounted a VL53L0X ToF sensor on the top surface of the syrup container to accurately measure the amount of syrup required to deliver. It precisely measures the volume to be delivered using the liquid stem height calculation technique. Also, this sensor has 1 mm resolution up to 8cm and it is best suited for IoT platform devices due to its energy saving capabilities.

* Servo Motors

A picture containing adapter, light

Description automatically generatedRotational actuators have to be used to select the type of tablet to be given by rotating the tablet container, keeping the aperture open until the syrup volume to be released, and delivering the drug in a timely manner. Also, different mechanical mechanisms should be used for the above separate functions, and for this, the servo motors should be rotated to the corresponding angles with different rotational acceleration and velocity. The prototype we propose to implement uses an SG5010 servo motor which has a rotational angle of 180 degrees with fast control response (operating speed of 0.16sec/º60) and stall torque of 6.5kg/cm at 6.6V.

**2.3 Power**

We intend to use an AC to DC 12V - 2A power adapter to operate our medical device. Then the 12V is efficiently reduced to 5V and 6V using Pololu D24V22F5 and D24V22F6 voltage regulators to obtain the required 5V for the sensors and 6V for the actuators. These buck converters can deliver typical continuous output currents between 1.8 A and 2.5 A, depending on the input voltage. Especially in the module there is an enable input that can be used to put the regulator in a low-power state with a current draw of 5 µA to 10 µA per volt. In addition, a 12v Li-Ion battery pack is used to prevent the device from malfunctioning in the case of an emergency power breakdown.

The device is operated supporting two modes to reduce the power consumption which is an important feature of an IoT device.

1. Low power sleeping mode – All the sensors will turn off (sleeping mode) and only leave the microcontroller unit into low power mode. In this case no data is received from the sensors and no data is sent to the actuators.

2. Operating mode – In this mode, the microcontroller usually listens to MQTT requests for controlling the actuators and receives sensor readings from selected sensors.

**CHAPTER 03 – MECHANICAL DESIGN**

**3.1 Introduction**

**CHAPTER 04 – SOFTWARE IMPLEMENTATION**

**4.1 Introduction /////shaggy need to be edited**

We intend to implement this system using Arduino software,

**4.2 Software flow**

* In addition to Arduino in-built libraries, we hope to use ESP8266 WiFi libraries and PubSubClient mqtt client library. And also, ESPAsyncTCP and ESPAsyncWebServer libraries have to be used. ESPAsyncTCP is an asynchronous TCP library which allows multi connection network environment for ESP8266 based systems. ESPAsyncWebServer is a library based on ESPAsyncTCP and it can create an asynchronous server instance running on ESP8266. In this application scenario, the system needs to handle multiple concurrent connections (ex: dealing with user inputs and outputs), therefore it is important to use an asynchronous approach which can easily integrate with the user interface. On ESP8266, an asynchronous server listening to http traffic is created instead of a normal WiFi server.
* Using PubSubClient library, an mqtt client instance is created on ESP82266 which is connected to a mqtt broker. (“test.mosquitto.org”).
* ESP8266 is connected to the WiFi LAN in station mode. Mqtt client connection drives on top of this.
* Any user input filled by the user under any field will be collected and and published to the mqtt broker under relevant topics.
* Real time information that are stored in cloud will be published to MQTT server and they are subscribed by the nodeMCU. On the other hand, data coming from sensors are processed by the microcontroller and publish those into relevant topics to MQTT via nodeMCU. And they are subscribed by the backend server.
* The relevant data in the cloud are published to following topics (these same topics are subscribed by nodeMCU via MQTT). Those data is published and subscribed on topics unique to the patient through the MQTT broker.
* outgoingTopic4a : patients details (Name,Age,Gender)
* outgoingTopic4b : patients’ description given by the doctor in brief.
* outgoingTopic4c : relevant medicines prescribed by the doctor.

//////////////////////////////////////////////////////need to be edited

* Topics to which ESP8266 mqtt client has subscribed to and corresponding data published under that topic by the node-red application,
* incommingCurrency : Inserted currency amount converted to t
* incommingCovid : Covid-19 updates (real time data)
* incommingWeather : Weather updates of the location provided
* incommingAirQuality : Air quality of the location provided
* Servo.h, DHT.h and Adafruit\_VL53L0X libraries are used to drive servo motors and corresponding sensors.
* Reliability and QoS – Since IoT is dealing in constrained environment there are so many things we need to consider. Such as data transmission, handling data losses etc. Higher QoS level means higher reliability. Here we need to have reliable data transfer as we are dealing with human lives. Such as MQTT with level 2 QoS. Since MQTT uses TCP protocol in transport layer it also ensures the reliability of communication.

**4.3 Protocols///////shaggy need to be edited on HTTP section**

* HTTP : Used to host the user interface webpage on the internet (code on ESP8266) Used to communicate between node-red application and the Open API data bases to obtain necessary data
* MQTT : The entire system is implemented in a publish-subscribe architecture via an MQTT broker

Other than these application layer protocols, in lower layers normal internet protocol suite is used (TCP/IP). Here we use an asynchronous TCP library to create an asynchronous web server (on ESP8266) that can handle up to multiple connections at a time. Publishers and subscribers don’t need to know each other. Therefore, events can handle asynchronous way allowing greater scalability and flexibility. Both publishers and subscribers rely on MQTT broker then infrastructure can be easily scaled up.

* IEEE 802.11ah(Wifi HALOW) : Some of the key features of this protocol will be help to implement our system in an efficient manner. Such as low throughput, high node density, low power consumption and long range. Therefore, we intend to use this protocol for our sensor networks without using conventional WiFi.