DESIGN AND DEVELOPMENT OF GLUCOSE DRIPS MONITORING AND CONTROL SYSTEM

A Project Report submitted to

Ramaiah Institute of Technology, Bangalore

(Autonomous Institute Affiliated to VTU)

In partial fulfilment of the requirement for the award of degree of

BACHELOR OF ENGINEERING

IN

TELECOMMUNICATION ENGINEERING

For the Academic Year 2017-18

Submitted by

DISHA K (1MS13TE012)

MAHALAXMI KAVADIKERI (1MS13TE022)

PRAJWAL R HEGDE (1MS13TE032)

Under the guidance of

Dr. K NATARAJAN

Dept. of Telecommunication Engineering

MSRIT, Bangalore-560054

M.S. RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute Affiliated to VTU)

Department of Telecommunication Engineering

CERTIFICATE

Certified the project work entitled "DESIGN AND DEVELOPMENT OF GLUCOSE DRIPS MONITORING AND CONTROL SYSTEM "carried out by Ms DISHA K (USN 1MS13TE012), Ms MAHALAXMI KAVADIKERI (USN 1MS13TE022) and Mr PRAJWAL R HEGDE (USN 1MS13TE032) bonafide students of Ramaiah Institute of Technology, Bangalore, in partial fulfilment for the award of Bachelor of Engineering in Telecommunication Engineering of the Visveswaraiah Technological University, Belgaum during the year 2016-2017. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Dr. K NATARAJAN	**Dr. N. V. R NAIDU**
HOD, Dept. of TCE Engg,	Principal,
RIT	RIT

External Viva

Name of the Examiners	**Signature with date**
1.	
2.	

Department of Telecommunication Engineering

DECLARATION

We hereby declare that the project entitled "DEVELOPMENT AND DESIGN OF GLUCOSE DRIPS MONITORING AND CONTROL SYSTEM "has been carried out independently by us, under the guidance of **Dr. K NATARAJAN**, HOD, Telecommunication Engineering, Ramaiah Institute of Technology, Bangalore. This report has been submitted in part or full for the award of any degree of this or any other University.

Disha K,

Mahalaxmi Kavadikeri,

Prajwal R Hegde,

TE, RIT,

Bangalore- 560 054

Place: Bangalore

Date:

ACKNOWLEDGEMENT

We take this opportunity to convey our gratitude to all those who have been kind enough to offer their advice and provide assistance when needed which has led to the successful completion of the Project work.

We would like to express our immense gratitude to **Dr. N V R Naidu** , Principal, **Ramaiah Institute Of Technology**, Bangalore and **Dr. K Natarajan**, Professor and head, Dept. of Telecommunication Engg. **Ramaiah Institute Of Technology**, Bangalore ,for their constant support and motivation that has encouraged us to come up with this project and also for providing the right ambience for carrying out the work and the facilities provided to us .

We express our warm thanks to our project guide **Dr. K Natarajan**, Professor and head, Dept. of Telecommunication Engineering , **Ramaiah Institute Of Technology** for his skillful guidance ,constant supervision, timely suggestion and constructive criticism in successful completion of our project in time.

We would like to express our sincere gratitude to **Prof. Umesha Raddy** and **Prof. Venu K N** for their continuous support,feedback,encouragement and for sparing their valuable time.

We are thankful to all the staff members of the telecommunication department who have rendered their whole hearted support at all times for successful completion of this project.
Finally we are also thankful to our family and friends for their continuous support and encouragement.

DISHA K,

MAHALAXMI KAVADIKERI,

PRAJWAL R HEGDE

ABSTRACT

For good patient care in hospitals, assessment and management of patient's fluid and electrolyte need is the most fundamental thing required. All most in all hospitals, an assist/nurse is responsible for monitoring the IV fluid level continuously. But unfortunately during most of the time, the observer may forget to change the saline bottle at correct time due to their busy schedule. This may lead to several problems to the patients such as backflow of blood, blood loss etc.

To overcome this critical situation, a low cost Wifi based automatic alerting, monitoring and controlling device is proposed .

Here a load cell is used to monitor the weight of the IV fluid continuously and a certain threshold is set .Once the load value goes below this threshold, a buzzer rings and an

alert message with the is sent using GUI.

A laser and LDR (light dependent resistor) setup is used to design a drip sensor. When the laser beam is broken due to a droplet, the intensity will vary and this intensity variations changes the output voltage level and this is used to detect the droplets . A threshold value is set for the volume of the leftover IV fluid where in a notification with an alert message is sent to the nurse/assist.

A DC motor with a coupler is used to automatically control the flow rate of the drips.

TABLE OF CONTENTS

CONTENTS **PAGE NO**

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

Medical health has been a primary concern in the society for a long time. It is one of the most important aspects to consider when talking about human resource development. The world has been facing innumerous diseases and consequent deaths. These diseases are a huge threat as they result in loss of a major part of the ecosystem i.e. biotic beings. Projects on biomedicine find a lot of applications. Each year, there are more than 1.7 billion cases of diarrhea, which lead to the death of many children. These deaths are the result of excess fluid loss and dehydration. Loss of fluids from the body can be treated with intravenous (IV) therapy. In this paper, we discuss primarily about the infusion of blood and glucose into the patient. Due to low or high infusion rates or even no infusion for a long time leads to several deaths. Thus a device that monitors the rate of infusion accurately as and when needed by the patient is required. This device will overcome the challenges of manual monitoring like checking the flow from time to time, as well as those of the basic infusion system like probable leakage of fluid.

1.1 Need for monitoring and controlling of glucose drips

For good patient care in hospitals, assessment and management of patient's fluid ,an electrolyte need is the most fundamental thing required. All most in all hospitals, an assist/nurse is responsible for monitoring the IV fluid level continuously.

But unfortunately during most of the time, the observer may forget to change the saline bottle at correct time due to their busy schedule. This may leads to several problems to the patients such as backflow of blood, blood loss etc.

Death rates from drug overdose and other forms of accidental poisoning among young adults have been raised from 3.8 deaths per 100,000 to 12.8 deaths per 100,000.To overcome this critical situation, a low cost Wifi based automatic alerting, monitoring and controlling device has been proposed.

1.2 Motivation behind the project

The National Institute for Health and Care Excellence (NICE) has issued guidance on when and how IV fluids should be prescribed. It states that there is evidence that as many as one in five patients on IV fluids and electrolytes (salts and minerals) suffer complications because of inappropriate use.If a patient receives too little fluid, it increases the risk of dehydration and in severe cases can lead to kidney failure.

The risks of receiving too much fluid often depend on individual circumstances, but they can include:

- excess fluid collecting inside the lungs, which can cause breathing difficulties and increased risk of pneumonia
- swelling of the ankles
- an imbalance of electrolytes in the blood, which can disrupt organs

heart failure in serious cases.

Hence to assure the safety of the patient during IV period there is a need to develop an efficient health monitoring system. Thus we decided to develop and design a prototype for monitoring the fluid continuously and automatically control the drips through an android application.

1.3 Introduction to glucose drips monitoring and controlling

Generally, as the population growth increases, the need for health care also increases. Hence it is a mandatory thing for everyone in this world to take care of their health properly.

In this scenario, maintaining patient's safety is the top most priority to be given in all. Now days, many automatic health monitoring devices are developed to ensure patients safety and to reduce the stress of the doctors.

The invention of such devices introduces a drastic change in medical field for monitoring the parameters like heart beat rate, detection of heart attack symptoms and much more automatically with interdisciplinary nature. Even though many advanced automatic devices are used, ensuring the safety of the patients during IV period is still a challenging issue. Intravenous (within vein) therapy is the infusion of liquid substances directly into the vein. Therapies administered intravenously are often called specialty pharmaceuticals or drips.

Even though monitoring the IV fluid level of patient is a small thing for a nurse but it will affect the patient's health severely during illness if the assist does not monitor it regularly. This may leads to blood loss or backflow of blood to IV tube from their vein. This results in the reduction of hemoglobin level of patients and it may also make the person anemic.

The task of assessing and managing the patients with sufficient skill needs to be a fundamental thing for a good patient care. Hence to assure the safety of the patient during IV period there is a need to develop an efficient health monitoring system. This can be achieved with the proposed idea of a load cell , laser , ldr set up for continuously monitoring the drips and a DC motor with a coupler for automatically controlling the flow rate of the drips.

1.4 Advantages

- ➢ Quantitative control: The flow rate of the drips can be automatically controlled as required by each patients.

- Automatic control of valves: The flow rate is controlled through an android application instead of handling it manually

- Available with different flow rate

- Less power consumption: It runs with a low power and hence it is available at a low cost

- Portable

- User friendly: This prototype can be used by anyone through an android phone.

- High accuracy: It has an accuracy of more than 85%.

- Good precision.

1.5 Limitations

- It requires an external power supply to run the DC motor.

- Since we create our own Wi Fi system, it has a limited range.

- If the intensity of sunlight varies , the IV drops will not be detected precisely.

1.6 Challenges

- Alignment of the Laser module and LDR should be in a straight line because the sensitivity of LDR varies with light..

- The DC motor should have enough torque to pinch the tube sufficiently so that the speed of drips can be controlled.

CHAPTER 2

2. BACKGROUND THEORY

2.1 Exploration of background working of the system

The administration of intravenous (IV) fluids is an integral part of patient care in hospital allowing for the delivery of parenteral fluids and additives. Ensuring the accuracy of the infusion rate is an important goal in all settings, but there are some rate critical settings where the risks of both under-and over-infusion must be avoided. These would include resuscitation, the delivery of toxic therapies and in the young and elderly. The control of the infusion rate is usually achieved with a roller-clamp or by a DC motor. Our aim is to make IV flow automatic with more accuracy. Hence we combine wireless and medical sensors to achieve this.

Wireless Sensor Networks

A wireless sensor network is a collection of nodes organized into a cooperative network. Each node consists of processing capability 9one or more microcontrollers, CPUs or DSP chips), may contain multiple types of memory (program, data and flash memories), have a RF transceiver (usually with a single omni-directional antenna), have a power source (e.g., batteries and solar cells), and accommodate various sensors and actuators. The nodes communicate wirelessly and often self-organize after being deployed in an ad hoc fashion. Systems of 1000s or even 10,000 nodes are anticipated. Such systems can revolutionize the way we live and work. Currently, wireless sensor networks are being deployed at an accelerated place. It is not unreasonable to expect than in 10-15 years the world will be covered with wireless sensor networks with access to them via Internet. This can be considered as the Internet becoming physical network. This new technology is exciting with unlimited potential for numerous application areas including environmental, medical, military, transportation, entertainment, crisis management, homeland defences, and smart spaces.

2.2 Characteristics of Wireless Sensor Networks

In ad hoc networks, wireless nodes self-organize into an infrastructure less network with a dynamic topology. Sensor networks have several distinguishing features. The number of nodes in a typical sensor network is much higher than in a typical ad hoc network, and dense deployments are often desired to ensure coverage and connectivity; for these reasons, sensor network hardware must be cheap. Nodes typically have stringent energy limitations, which make them more failure-prone. They are generally assumed to be stationary, but their relatively frequent breakdowns and the volatile nature of the wireless channel nonetheless result in a variable network topology. Ideally, sensor network hardware should be power-efficient, small, inexpensive, and reliable in order to maximize network life-time, add flexibility, facilitate data collection and minimize the need for maintenance.

- **Life time**

Life time is extremely critical for most applications, and its primary limiting factor is the energy consumption of the nodes, which need to be self-powering. Although it is often assumed that the transmit power associated with packet transmission accounts for the lion's share of power consumption, sensing, signal processing and even hardware operation in standby mode consume a consistent amount of power as well. In some applications, extra power is needed for macro-scale actuation.

Energy-efficient routing should avoid the loss of anode due to battery depletion. May proposed protocols tend to minimize energy consumption on forwarding paths, but if some nodes happen to be located on most forwarding paths (e.g., close to the base station), their lifetime will be reduced.

- **Flexibility**

Sensor networks should be scalable, and they should be able to dynamically adapt to changes in node density and topology, like in the case of the self-healing minefields. In surveillance applications, most nodes may remain quiescent as long as nothing interesting happens. However, they must be able to respond to special events that the network intends to study with some degree of granularity. In a self-healing minefield, a number of sensing mines may sleep as long as none of their peers explodes, but need to quickly become operational in the case of an enemy attack. Response time is also very critical in control applications in which the network is to provide a delay-guaranteed service.

- **Maintenance**

The only desired form of maintenance in a sensor network is the complete or partial update of the program code in the sensor nodes over the wireless channel. All the sensor nodes should be updated, and the restrictions on the size of the new code should be the same as in the case of wired programming. The portion of code always running in the node to guarantee reprogramming support should have a small footprint, and updating procedures should only cause a brief interruption of the normal operation of the node.

The nodes should be able to assess the quality of the network deployment and indicate any problems that may arise, as well as adjust to changing environmental conditions by automatic reconfiguration. Location awareness is important for self-configuration and has definite advantages in terms of routing and security. Time synchronization advantageous in promoting cooperation among nodes, such as data fusion, channel access, coordination of sleep mode, or security-related interaction.

CHAPTER 3

3. LITERATURE REVIEW

R Vasuki and et al., proposes “A portable monitoring device of measuring drips rate by using an intravenous (IV) set”**.** In this method the IV set is attached to the drips chamber. The flow sensor is used to detect each drops of IV set. For each drop, the beam of light is broken at each time and that is transmitted and received by IR sensor. This provides a change in sensor output and comparator gives a pulse output for each drop. The drip rate is indicated using the LCD with which the observer can identify the volume of fluid in IV set. If the device is not sensed for 45 seconds it will give an alarm.

C C Gavimath and et al., proposes a method of “Design and development of versatile saline flow rate measuring system and GSM based remote monitoring device”. In this device an indigenously developed sensor is attached to the neck of the drips bottle. For every drop of the saline, the signal conducting circuit produces one pulse. The signal conditioning circuit consists of a multivibrator, comparator and phototransistor. The 8051 microcontroller is used to count the pulse in unit time. This will resemble the flow rate. Through GSM technology the information about the flow rate is send to the observer’s mobile. The cost of this device is high.

R Aravind and et al., Proposed a paper, “Design of family health monitoring system using wireless communication”.This is an ARM based embedded system through which the data of the patient is transmitted and received via Zigbee or RF transmitter and receiver. Then the information is stored as database and send to GSM. The database consists of all the details about the patient health conditions such as temperature, blood pressure and heartbeat by using visual basics. This makes the residential people to check their health by themselves but computer should need an IE. But It is not suitable for illiterate people for whom it is very difficult to operate and understand.

V Ramya and et al., proposes an “Embedded patient monitoring system”. In this system the status of the patient is monitored continuously by using an embedded system. Here the PIC microcontroller and sensor are used to sense temperature and drip status.

This status is given to the PC. If the temperature is greater than the set value it will send an alert to mobile phone and produces an alarm until the doctor response to that message.

CHAPTER 4

4. PROBLEM STATEMENTS

4.1 Problem statement objectives

- ➢ Interfacing of load sensor with the microcontroller and measuring the weight of fluid in milliliter.
- ➢ Interfacing of laser module and ldr system to the microcontroller to detect the drops .
- ➢ Interfacing of the DC motor to the microcontroller to control the flow rate,.
- ➢ Verifying flow rate for different cases .

4.2 Use Case

Use Case Name:

Glucose drip monitoring and controlling using android application

Use Case Description:

The application developed keeps monitoring the flow rate of the drips . The user will be provided with buttons to vary and control the speed. The user will also receive an alert message once the level becomes less than a certain value.

Triggers:

The user receives an alert message and indicates that he/she wants to control the flow rate of the drips.

Normal Flow:

1. The user receives an alert message and indicates that he/she wants to control the flow rate.

2. The user uses the buttons provided by the system to do so.

3. Thereby the system or the application provides the current flow rate and the patients' body temperature.

4. The user sets up the speed as per the condition of the patient using the button provided.

5. The flow rate is thus adjusted.

6. User exits the system.

CHAPTER 5

5. PROPOSED METHOD

An Intravenous therapy is a medical procedure in which the saline water directly enters into the veins through an IV tube and needle is inserted into the patients' vein. A drip chamber is usually used so that the IV fluid passes as droplets. An IV drip is usually used for long term treatments. But it can also be used for a short term to treat the patients who are dehydrated to revitalize them. This is an easy and an efficient method to supply the prescribed medicines quickly into the entire body.

Due to lack of caring, many problems will arise such as blood loss, backflow of blood through an IV tube. To overcome this situation an effective idea is proposed to develop an effective health monitoring system which alerts the doctor or nurse when the fluid level of saline bottle is beyond the threshold limit.

The idea is to use a load cell which continuously monitors the weight of the IV fluid in the drips bottle. A minimum threshold value is set according to the volume of the drips.Once the volume goes below this threshold a notification appears on the android app which is developed using GUI.

A drip sensor is designed to detect the drops and also count the number of drops .Initial volume of the drips is noted which is in millilitres say V1 .Each drop is converted into millilitre say V2 and the difference is calculated in each trial i.e. V1-V2. Two thresholds are set up and when the difference goes below the first threshold limit an alert message will be sent to the user .When the difference goes below the second threshold an emergency message is sent to the assist/nurse to indicate an immediate assistance to that particular patient.

The controlling part is developed using dc motor which is used to control the flow rate by monitoring steps of the motor.

CHAPTER 6

6. DESIGN APPROACH

A Glucose Drip Monitoring System has been developed using minimal number of circuit components, using the techniques of microcontroller based embedded system design.

It can be used for monitoring the drip rate of Glucose at hospital level .It is designed to operate with varying supply voltage from the battery without needing a regulator in order to conserve power.

Here the sensor assembly is clipped around the drip chamber whic senses the fluid drops in it. The load sensor which is assembled weighs the glucose level in the drips bottle continuously. Once the value of the load level goes below a certain threshold an alert message will be immediately sent wirelessly to the monitoring/caring Centre using a Wi-Fi module.

Meanwhile the hall flow sensor monitor the flow rate of the drips. This flow rate can be controlled using a solenoid valve.

An android application is developed at the later stage such that the flow rate can be monitored and controlled wirelessly.

CHAPTER 7

7. HARDWARE REQUIREMENTS

- ➢ CC3200 module
- ➢ Load cell
- ➢ Laser module
- ➢ Light dependent resistor(ldr)
- ➢ DC motor
- ➢ L293d (DC motor driver)
- ➢ Coupler

7.1 CC3200 module specifications

- 802.11 b/g/n Station and Access Point with fully integrated radio, baseband, and MAC.
- Single-chip Wi-Fi MCU: Wi-Fi network processor + Arm® Cortex®-M4 MCU integrated into one chip including RF reference design.
- Embedded TCP/IP stack and Wi-Fi security transparent to the application MCU user
- Internet protocols including mDNS, DNS, SSL/TLS in ROM + web server (HTTP) in Flash. Example codes include instant messaging (XMPP) and email (SMTP)
- Embedded hardware crypto engine for TLS/SSL internet security
- Flexible connection methods (provisioning) including access point mode, WPS, SmartConfig™ technology,and WAC

- Low-power radio and advanced low-power modes
- Peripheral set includes parallel camera, I2S audio, SDMMC, ADC, SPI, UART, I2C, and PWMs
- Small 9mm x 9mm QFN package

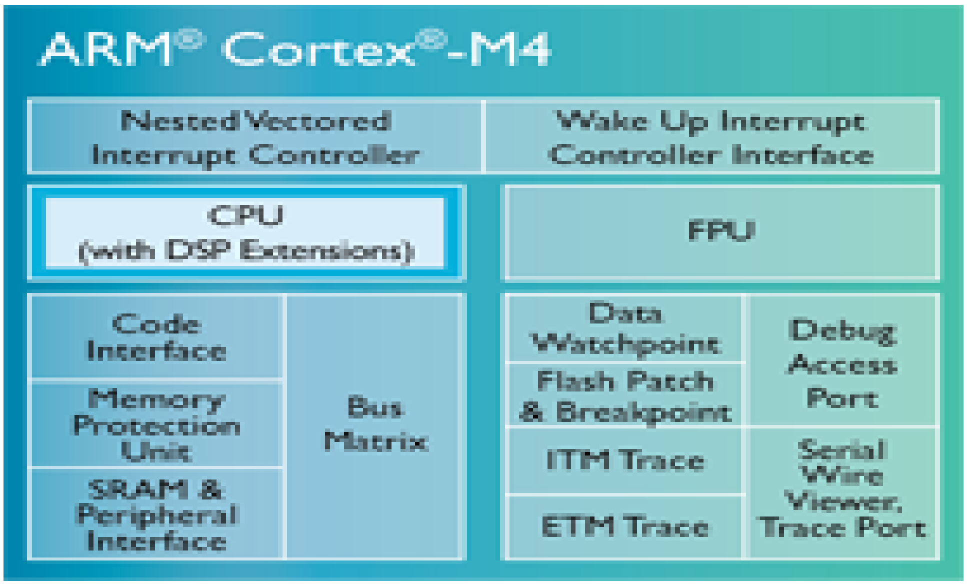


Figure 1

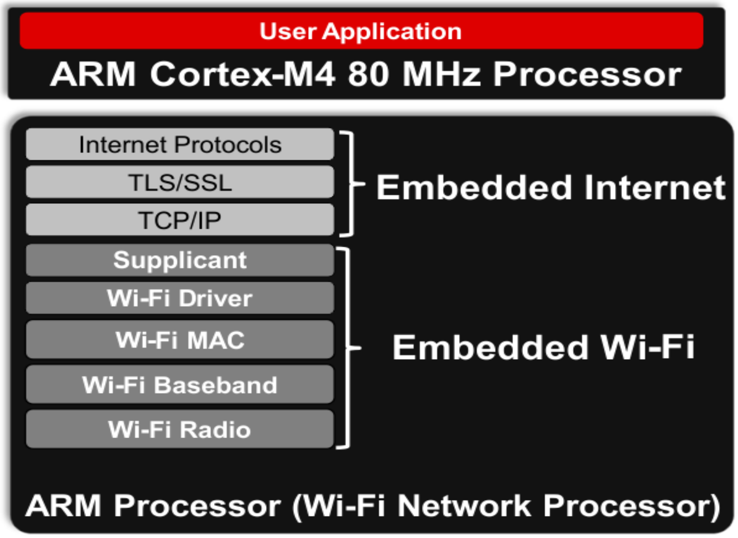


Figure 2

7.2 Load Cell specifications

Rated capacity:	0.2 kg, 0.6 kg, 1 kg, 2 kg, 3 kg
Non-linearity:	<0.04 % RO
Hysteresis:	<0.02 % RO
Creep, 30 min:	<0.05 % RO
Non-repeatability:	<0.01 % RO
Rated output (RO), nominal:	2.0 mV/V
Zero balance:	±0.3 mV/V (optional ±10.02 mV/V)
Temperature effect on zero balance:	<0.03 % RO/K (optional <0.004 % RO/K)
Supply voltage:	5 … 20 VDC or VAC
Bridge configuration	Full Bridge, 4 wire closed
Bridge resistance:	3000 Ω ±2 %
Insulation resistance (at 50 VDC):	>1000 MΩ
Static overload, safe:	150 % Full Scale Capacity

Static overload, ultimate:	300 % Full Scale Capacity
Long term stability:	<0.1 % R.O./year

Figure 3

Wiring:

Red + supply voltage

Blue - supply voltage

Yellow + output

7.3 Laser specifications

Parameter	Value
Supply Voltage	5 Vdc
Current	30 mA
Wavelength	650 nm
Color	Red

KY-008 Pin Outs

You need only supply ground and power to this device as shown in the picture below:

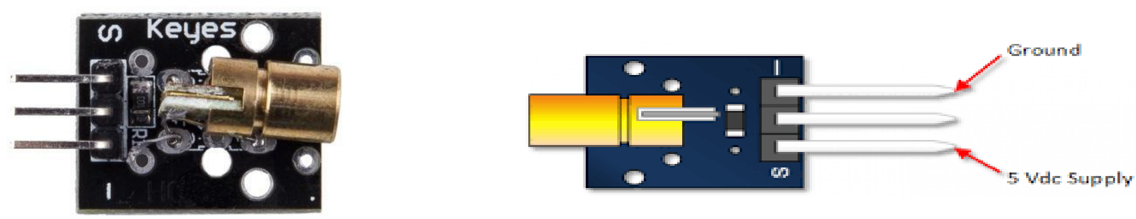


Figure 4

7.4 LDR Specifications

Specification	Light resistance (10Lux) (KΩ)	Dark resistance (MΩ)	γ_{10}^{100}	Response time (ms)		Illuminance resistance Fig. No.
				Increase	Decrease	
Φ5 series	4-7	0.5	0.5	30	30	2
	5-10	0.8	0.6	30	30	2
	10-20	2	0.6	20	30	3
	20-30	3	0.7	20	30	4
	30-50	4	0.8	20	30	4
	50-100	8	0.9	20	30	5
	100-200	15	0.95	20	30	6

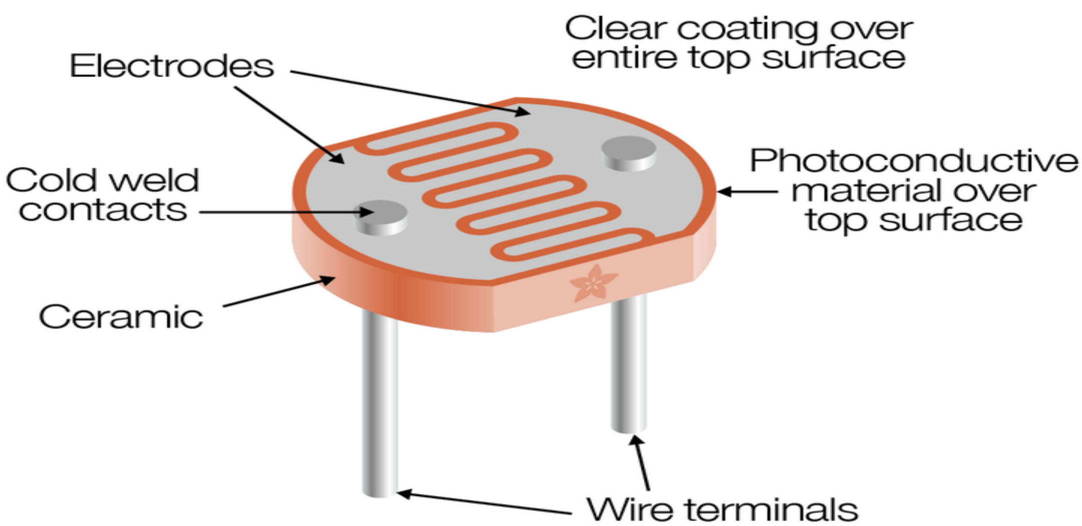


Figure 5

7.5 DC motor specifications

Horse Power Cont.	0.8 W (60 mA no load)	**Length of Motor (excluding spindle)**	54 mm
Gear Ratio **Toque**	1: 20 3.3 Lb-Inch	**Diameter of Motor**	25 mm
Voltage & Current	12 V DC	**Length of Spindle**	8 mm
RPM	200	**Diameter of Spindle (with a flat)**	4 mm
Reversibility	Reversible	**Weight per motor**	3 oz

Figure 6

7.6 L293D specifications

ITEMS	MIN	TYPICAL	MAX	UNIT
Control Voltage	4.5	5	5.5	v
Driver Voltage	6	9	15	v
Output Current	/	/	1.2	A
Dimension				CM
Weight				g

Figure 7

7.7 Coupler specifications

THICKNESS : Flexible coupler of 6 to 8 mm

MATERIAL USED : Stainless steel

Figure 8

CHAPTER 8

8. SOFTWARE REQUIREMENTS

- Energia
- MIT App Inventor 2

8.1 Energia

Energia is an open-source electronics prototyping platform started by Robert Wessels in January of 2012 with the goal to bring the Wiring and Arduino framework to the Texas Instruments MSP430 based LaunchPad. The Energia IDE is cross platform and supported on Mac OS, Windows, and Linux.

Energia uses the msp gcc compiler by Peter Bigot and is based on the Wiring and Arduino framework. Energia includes an integrated development environment (IDE) that is based on Processing. Energia is also a portable framework/abstraction layer that can be used in other popular IDEs. Utilize a web browser based environment with CCS Cloud at dev.ti.com. Community maintained Energia plug-ins and integrations are available for Xcode, Visual Studio, and Code Composer Studio.

The foundation of Energia and Arduino is the Wiring framework that is developed by Hernando Barragan. The framework is thoughtfully created with designers and artists in mind to encourage a community where both beginners and experts from around the world share ideas, knowledge and their collective experience. The Energia team adopts the philosophy of learning by doing and strives to make it easy to work directly with the hardware. Professional engineers, entrepreneurs, makers, and students can all benefit from the ease of use Energia brings to the microcontroller.

Energia started out to bring the Wiring and Arduino framework to the Texas Instruments MSP430 LaunchPad. Texas Instruments offers a MSP430, MSP432, TM4C, C2000, and CC3200 LaunchPad. The LaunchPad is a low-cost microcontroller board that is made by Texas Instruments. The latest release of Energia supports the majority of the LaunchPad product offerings. Additional community kits from RedBearLab are also supported.

Together with Energia, LaunchPad can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. LaunchPad projects can be stand-alone (only run on the Target Board, i.e. your LaunchPad), or they can communicate with software running on your computer (Host PC). You can also add wireless modules to enable communication over various types of RF including Wi-Fi, NFC, Bluetooth, Zigbee, cellular, and more.

8.2 MIT App Inventor

MIT App Inventor is an innovative beginner's introduction to programming and app creation that transforms the complex language of text-based coding into visual, drag-and-drop building blocks. The simple graphical interface grants even an inexperienced novice the ability to create a basic, fully functional app within an hour or less.

App Inventor for Android is an open-source web application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT).

It allows newcomers to computer programming to create software applications for the Android operating system (OS). It uses a graphical interface, very similar to Scratch and the StarLogo TNG user interface, which allows users to drag-and-drop visual objects to create an application that can run on Android devices. In creating App Inventor, Google drew upon significant prior research in educational computing, as well as work done within Google on online development environments.

App Inventor and the projects on which it is based are informed by constructionist learning theories, which emphasizes that programming can be a vehicle for engaging powerful ideas through active learning. As such, it is part of an ongoing movement in computers and education that began with the work of Seymour Papert and the MIT Logo Group in the 1960s and has also manifested itself with Mitchel Resnick's work on Lego Mindstorms and StarLogo

The application was made available through request on July 12, 2010, and released publicly on December 15, 2010. The App Inventor team was led by Hal Abelson[1] and Mark Friedman.In the second half of 2011, Google released the source code, terminated its server, and provided funding for the creation of The MIT Center for Mobile Learning, led by App Inventor creator Hal Abelson and fellow MIT professors Eric Klopfer and Mitchel Resnick. The MIT version was launched in March 2012.

On December 6, 2013 (the start of the Hour of Code), MIT released App Inventor 2, renaming the original version "App Inventor Classic" Major differences are:

The blocks editor in the original version ran in a separate Java process, using the Open Blocks Java library for creating visual blocks programming languages and programming

Open Blocks is distributed by the Massachusetts Institute of Technology's Scheller Teacher Education Program (STEP) and is derived from master's thesis research by Ricarose Roque. Professor Eric Klopfer and Daniel Wendel of the Scheller Program supported the distribution of Open Blocks under an MIT License. Open Blocks visual programming is closely related to StarLogo TNG, a project of STEP, and Scratch, a project of the MIT Media Lab's Lifelong Kindergarten Group. App Inventor 2 replaced Open Blocks with Blockly, a blocks editor that runs within the browser.

The MIT AI2 Companion app enables real-time debugging on connected devices via Wi-Fi, not just USB.

CHAPTER 9

9. INTRODUCTION TO CC3200 LAUNCH PAD

The high performance CC3200 is the industry's first single-chip Microcontroller (MCU) with built-in Wi-Fi connectivity for the LaunchPad™ ecosystem. Created for the Internet of Things (IoT), the SimpleLinkWiFi CC3200 device is a wireless MCU that integrates a high-performance ARM® Cortex®-M4 MCU allowing customers to develop an entire application with a single IC. With on-chip Wi-Fi, internet and robust security protocols, no prior Wi-Fi experience is needed for faster development. The CC3200 LaunchPad is a low-cost evaluation platform for ARM® Cortex™-M4F-based microcontrollers. The LaunchPad design highlights the CC3200 Internet-on-a-chip™ solution and WiFi capabilities.

The CC3200 LaunchPad also features programmable user buttons, RGB LED for custom applications and onboard emulation for debugging. The stackable headers of the CC3200 LaunchPad XL interface demonstrate how easy it is to expand the functionality of the LaunchPad when interfacing with other peripherals on many existing BoosterPack add-on boards such as graphical displays, audio codec, antenna selection, environmental sensing, and much more. Figure 1 shows a photo of the CC3200 LaunchPad.

9.1 Key Features

- CC3200, SimpleLink Wi-Fi, internet-on-a-chip™ solution with integrated MCU
- 40-pin LaunchPad standard that leverages the BoosterPack ecosystem
- FTDI based JTAG emulation with serial port for Flash programming
- Two buttons and three LEDs for user interaction
- Backchannel universal asynchronous receiver/transmitter (UART) through USB to PC

➢ On-board chip antenna with U.FL for conducted testing

9.2 Hardware Description

CC3200 LAUNCHPAD OVERVIEW

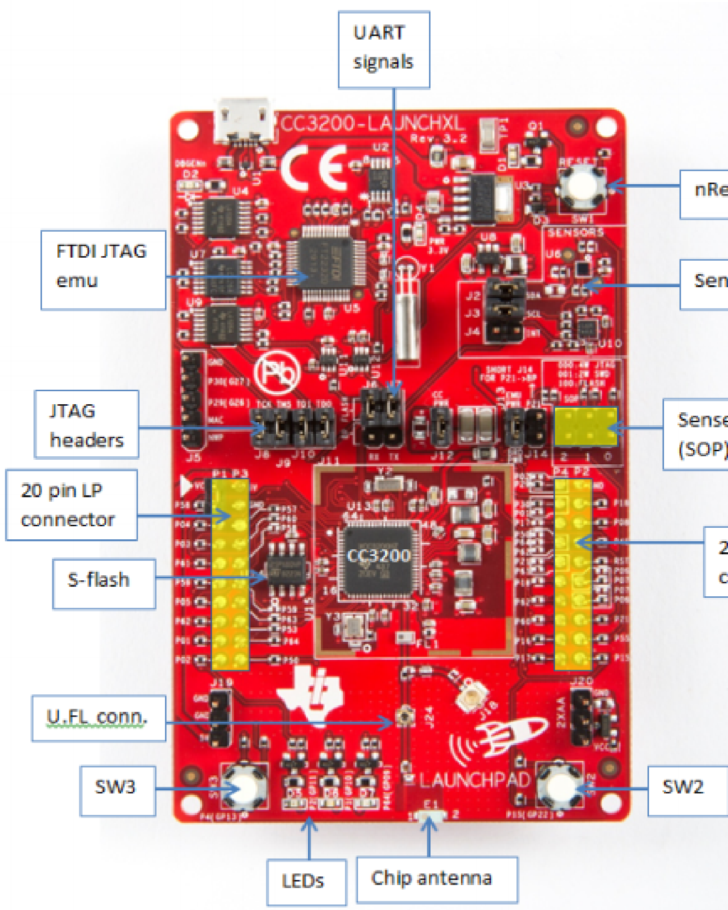


Figure 9

9.3 Block diagram of CC3200 Launchpad

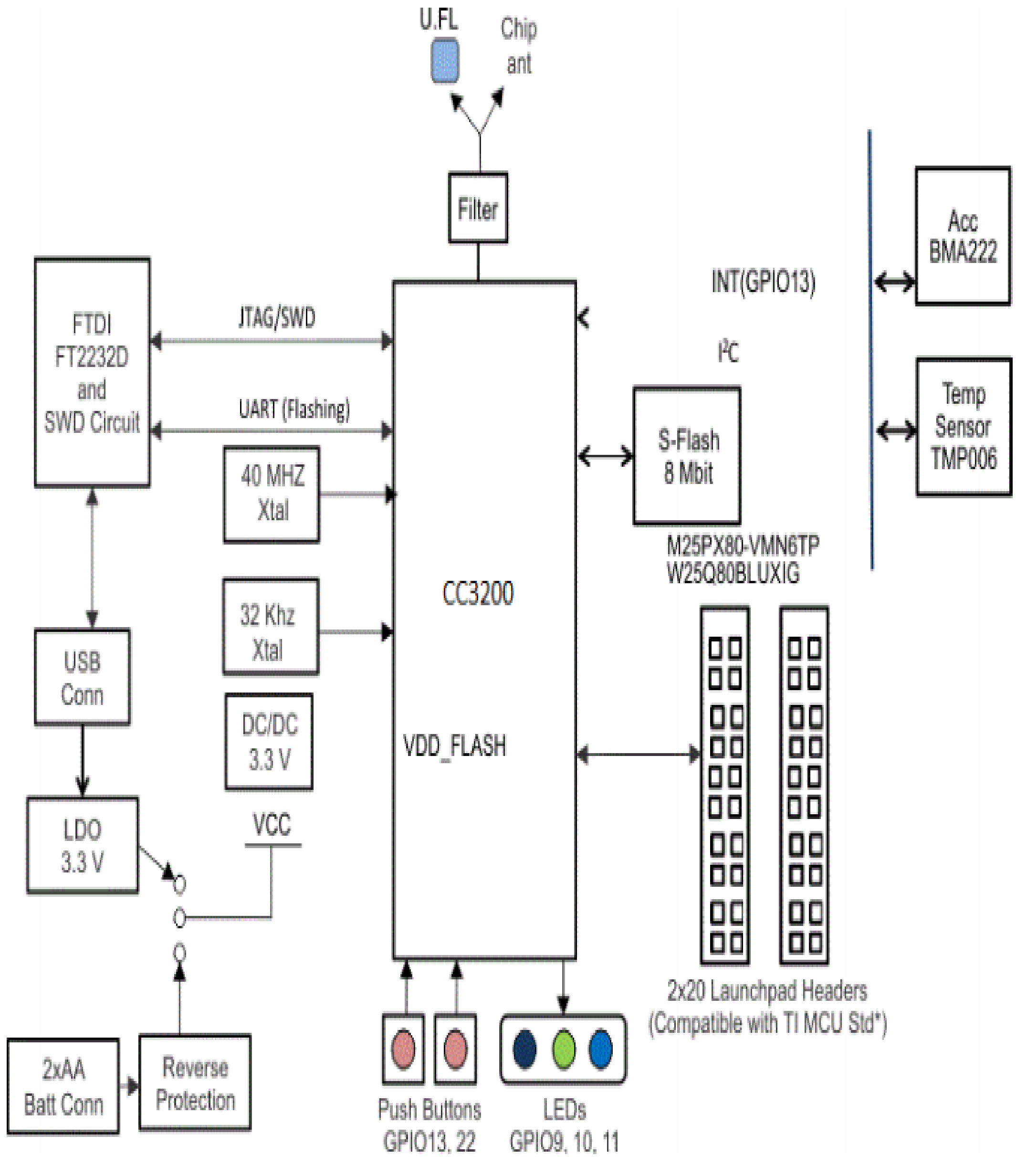


Figure 10

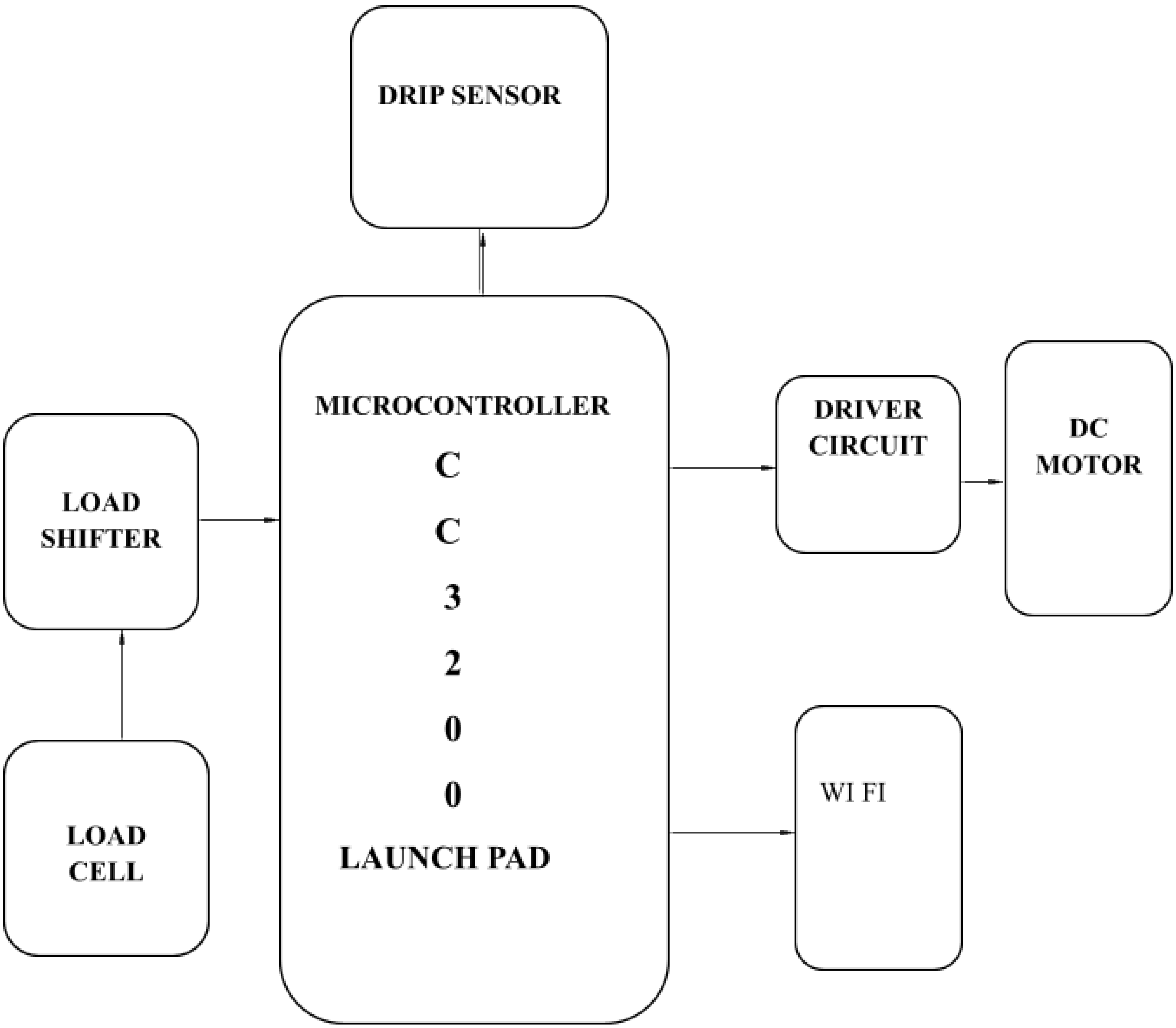
9.4 Hardware Features

- CC3200, SimpleLink Wi-Fi, internet-on-a-chip solution with integrated MCU40-pin LaunchPad standard that leverages the BoosterPack ecosystem
- FTDI-based JTAG emulation with serial port for Flash programming
- Supports both 4-wire JTAG and 2-wire SWD
- Two buttons and three LEDs for user interaction
- Virtual COM port UART through USB on PC
- On-board chip antenna with U.FL for conducted testing
- On-board accelerometer and temperature sensor for out-of-box demo with option to isolate them from the inter-integrated circuit (I2C) bus
- Micro USB connector for power and debug connections
- Headers for current measurement and external JTAG connection
- Bus-powered device with no external power required for Wi-Fi
- Long range transmission with highly optimized antenna (200m typical in open air with a 6dBi antenna AP)
- Can be powered externally, with 2xAA or 2xAAA alkaline batteries working down to 2.3V typical

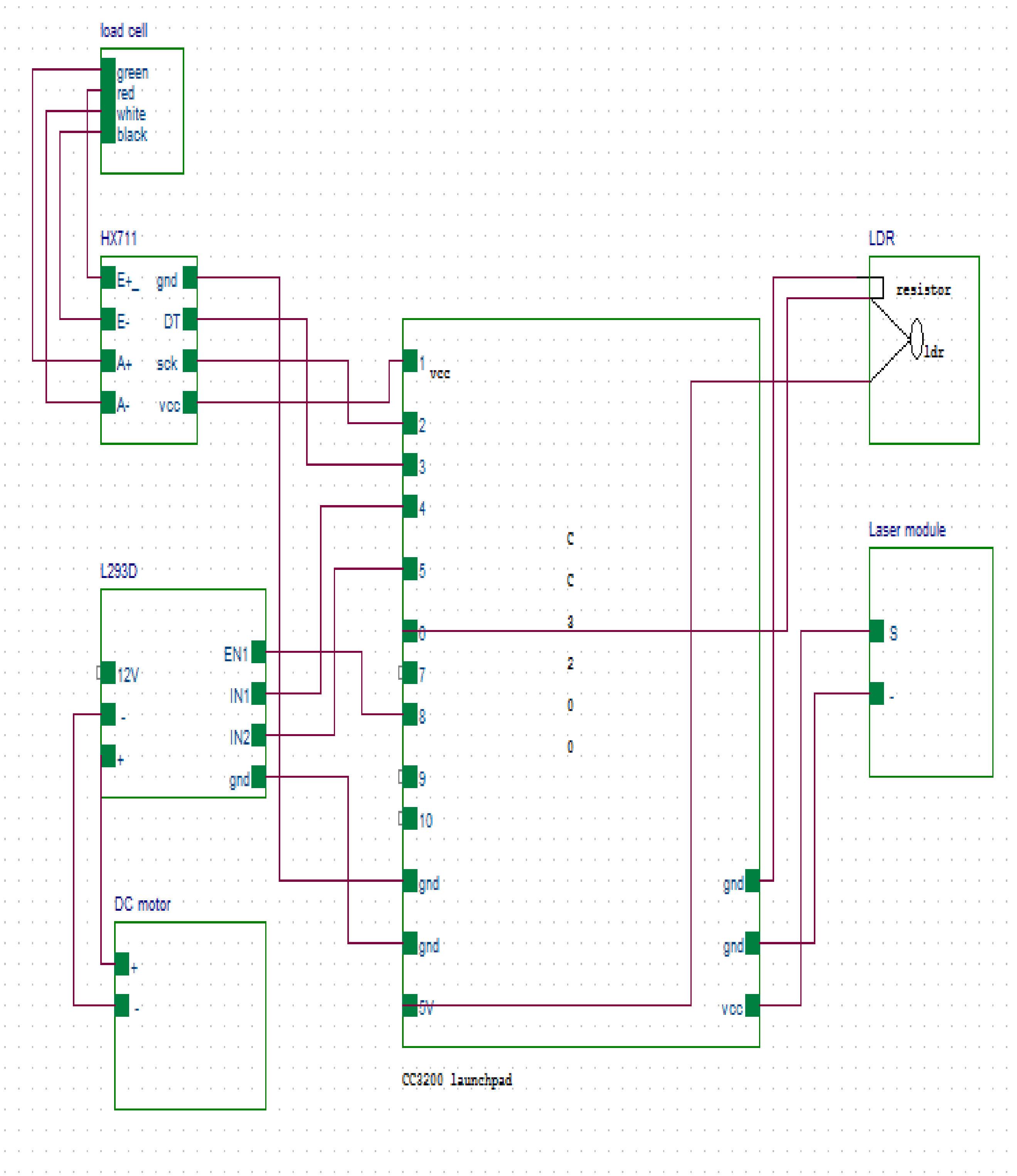
CHAPTER 10

10. SYSTEM ARCHITECTURE

10.1 Block Diagram



10.2 Schematic



CHAPTER 11

11. HARDWARE IMPLEMENTATION

The idea is to use a load cell which continuously monitors the weight of the IV fluid in the drips bottle. A minimum threshold value is set according to the volume of the drips.

Once the volume goes below this threshold a notification appears on the android app which is developed using GUI.

A drip sensor is designed to detect the drops and also count the number of drops .Initial volume of the drips is noted which is in milliliters say V1 .Each drop is converted into milliliter say V2 and the difference is calculated in each trial i.e.

V1-V2. Two thresholds are set up and when the difference goes below the first threshold limit an alert message will be sent to the user .When the difference goes below the second threshold an emergency message is sent to the assist/nurse to indicate an immediate assistance to that particular patient.

The controlling part is developed using DC motor.

11.1 Monitoring

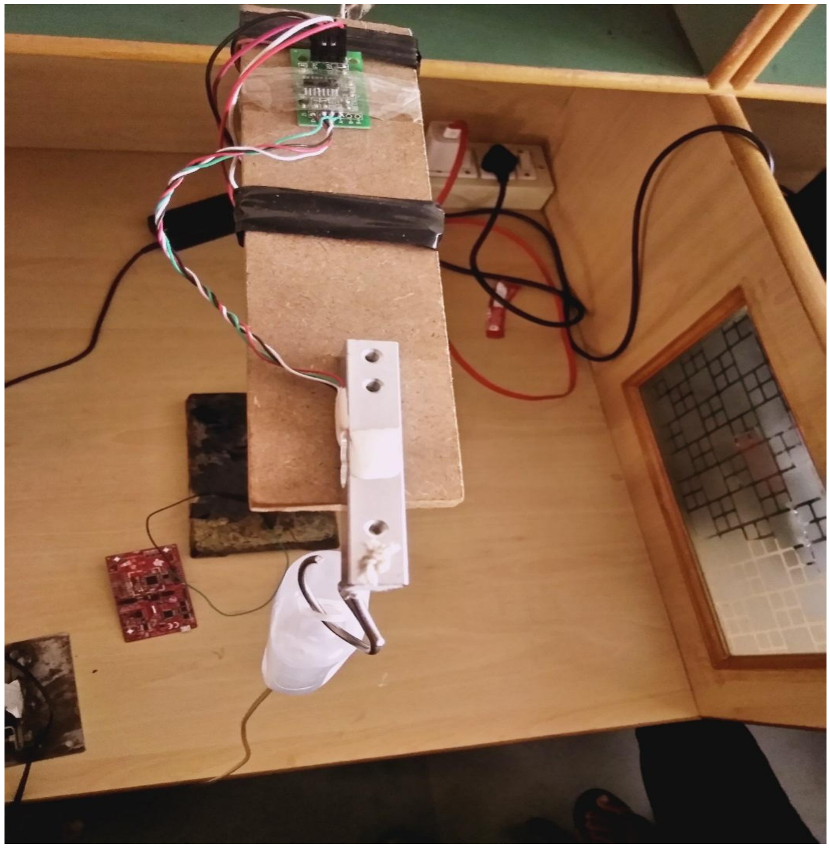


Figure 11

Monitoring part is designed using the load cell – hx711 (1 kg) as shown in the above figure. It is interfaced with CC3200 to monitor the weight continuously. Two threshold level , one at 20% and other at 10% is kept. When the weight of the fluid goes below this threshold a buzzer will be activated to indicate an immediate assistance to the patient.

The figure below shows a prototype of our implementation to monitor the fluid .

Figure 12

11.2 Detection

Detection part is designed using a laser module and LDR set up as shown in the below figure

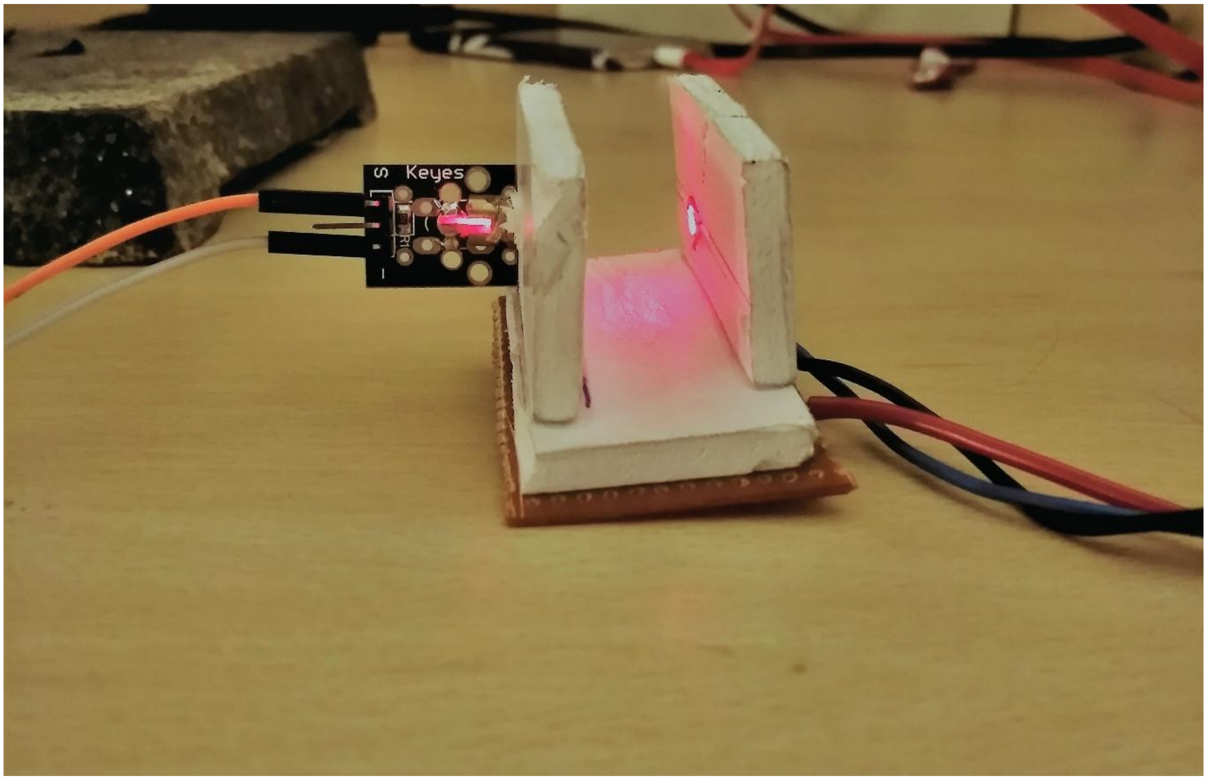
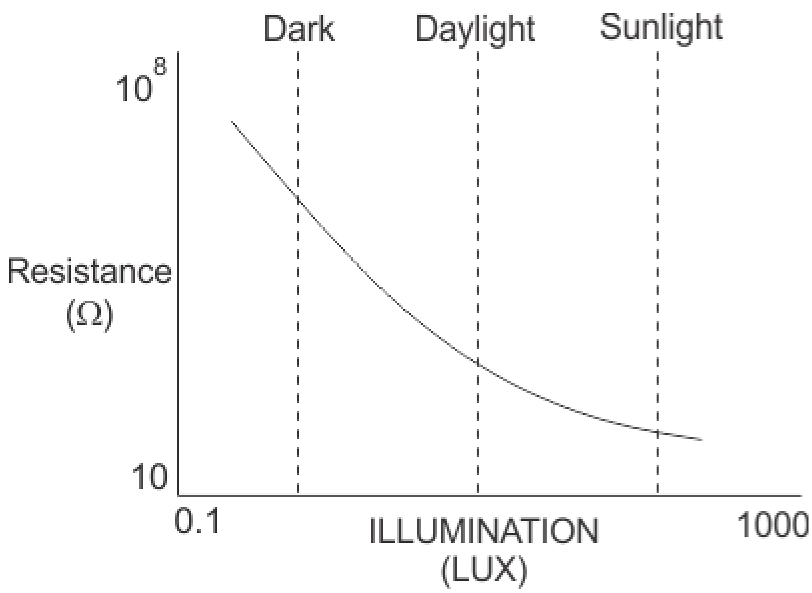


Figure 13

A light dependent resistor is a device whose resisitivity is a function of the incident electromagnetic radiation.Hence, they are light sensitive devices.They are also called asphototconductors, photo conductive cells or simply phototcells. They are made up of semiconductor materials having high resistance.

LDR's are light dependent devices whose resistance is decreased when light falls on them and that is increased in the dark. When a light dependent resistor is kept in dark, its resistance is very high. This resistance is called as dark resistance. It can be as high as 1012Ω and if the device is allowed to absorb light its resistance will be decreased drastically. If a constant voltage is applied to it and intensity of light is increased the current starts increasing.

Figure below shows resistance vs. illumination curve for a particular **LDR**.



A **laser module** differs from other sources of light in that it emits light coherently. Spatial coherence allows a laser to be focused to a tight spot, enabling applications such as laser cutting and lithography. Spatial coherence also allows a laser beam to stay narrow over great distances (collimation), enabling applications such as laser pointers. Lasers can also have high temporal coherence, which allows them to emit light with a very narrow spectrum, i.e., they can emit a single color of light. Temporal coherence can be used to produce pulses of light as short as a femtosecond.

The criticality in this set up is the alignment of laser and the ldr set up since ldr is sensitive to sunlight changes .

The alignment of laser and ldr is as shown below

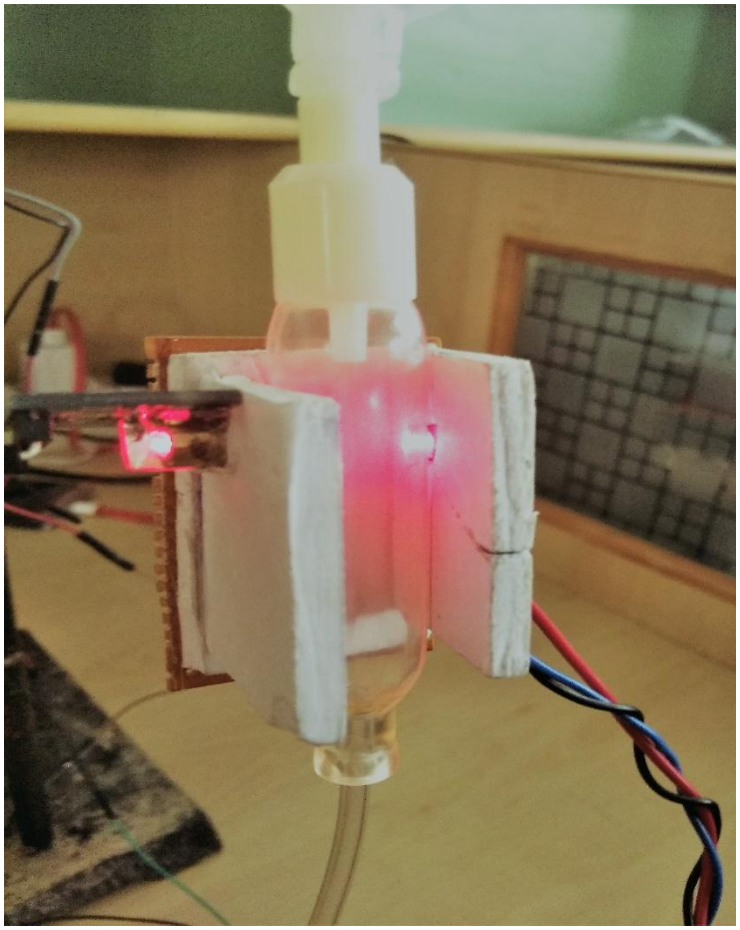


Figure 14

When the laser beam is broken due to a droplet ,the intensity will vary and this intensity variations changes the output voltage level and this is used to detect the droplets.Initial volume of the drips is noted which is in milliliters say V1 .Each drop is converted into milliliter say V2 and the difference is calculated in each trial i.e. V1-V2.

11.3 Controlling

Controlling part is designed using a DC motor which requires an external power supply. It is used to pinch the IV tube such that the speed of the drips is controlled as per the requirement.

The set up is shown below

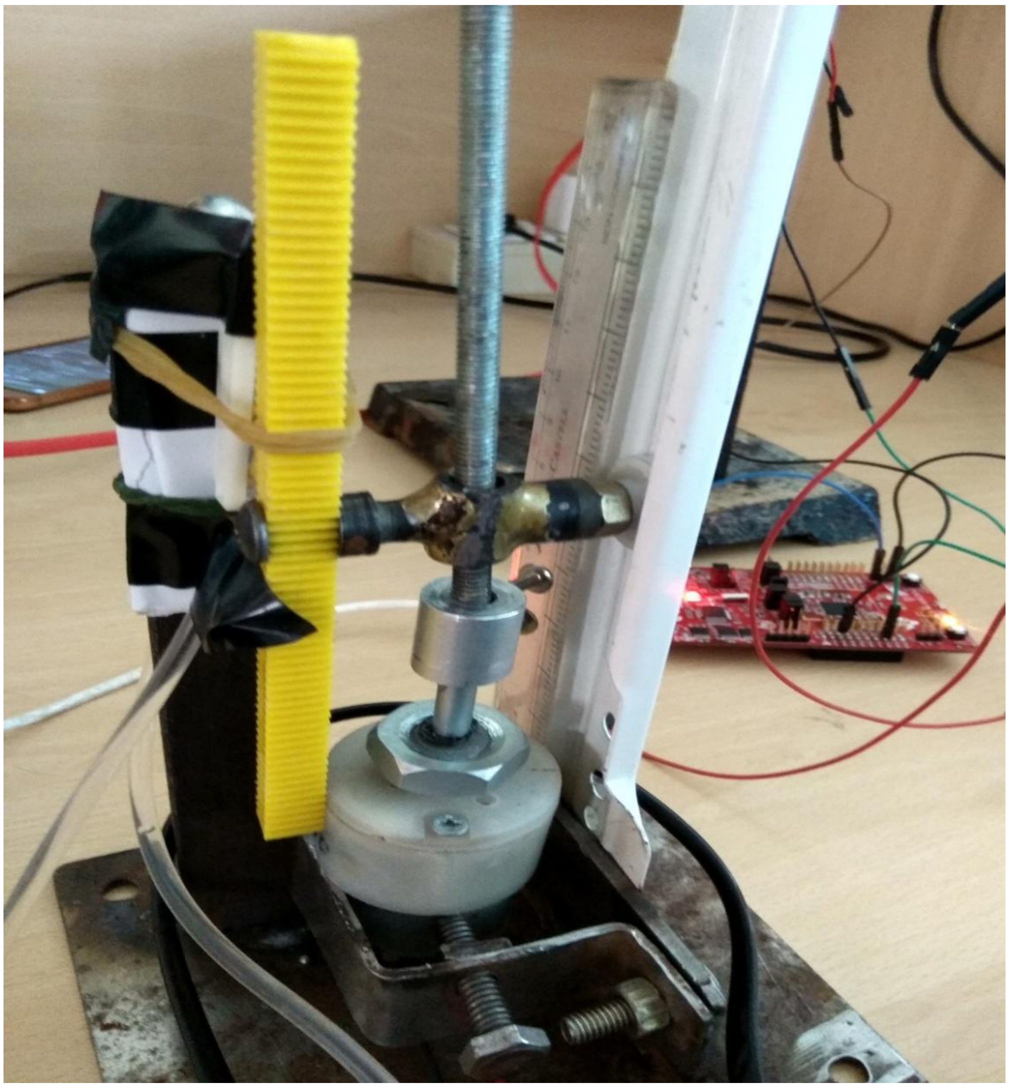


Figure 15

The roller of the IV tube is clamped in such a way that as the DC motor runs, the clamped roller also slides up and down to pinch the tube .This helps in controlling the speed of the IV drips as the requirement of the patient.

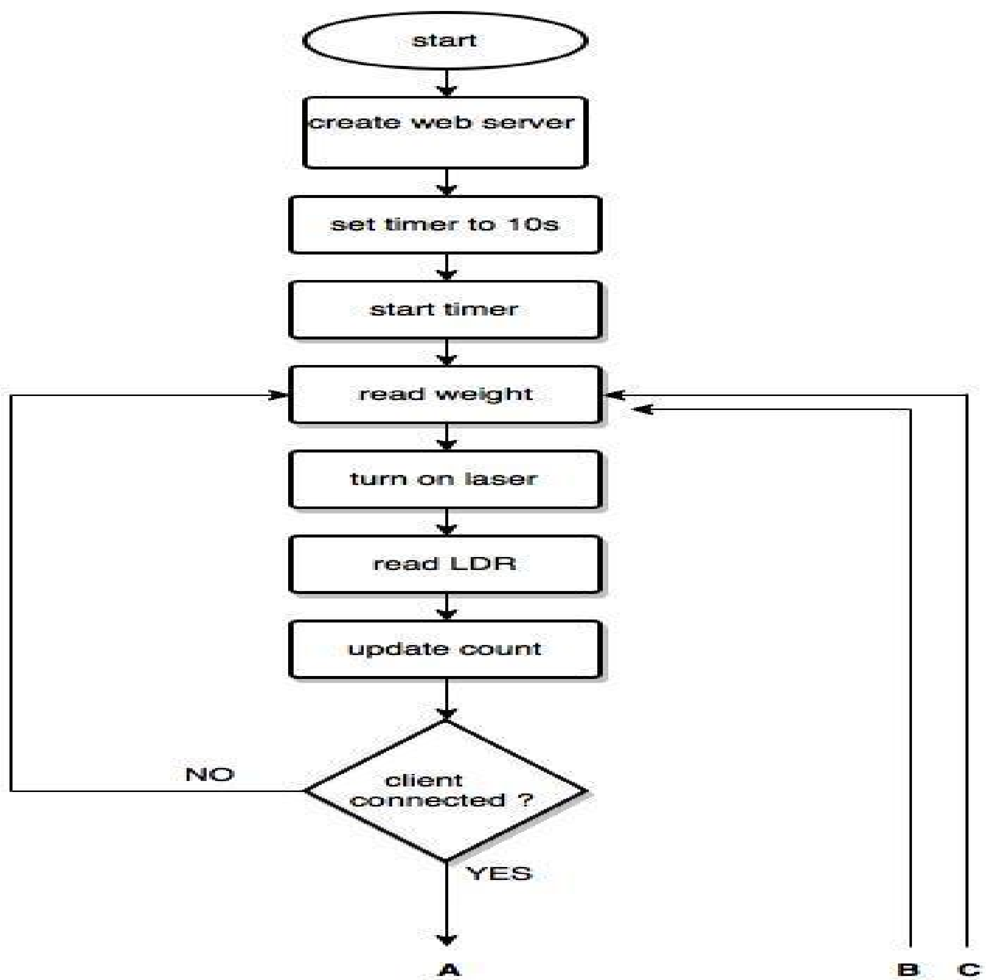
CHAPTER 12

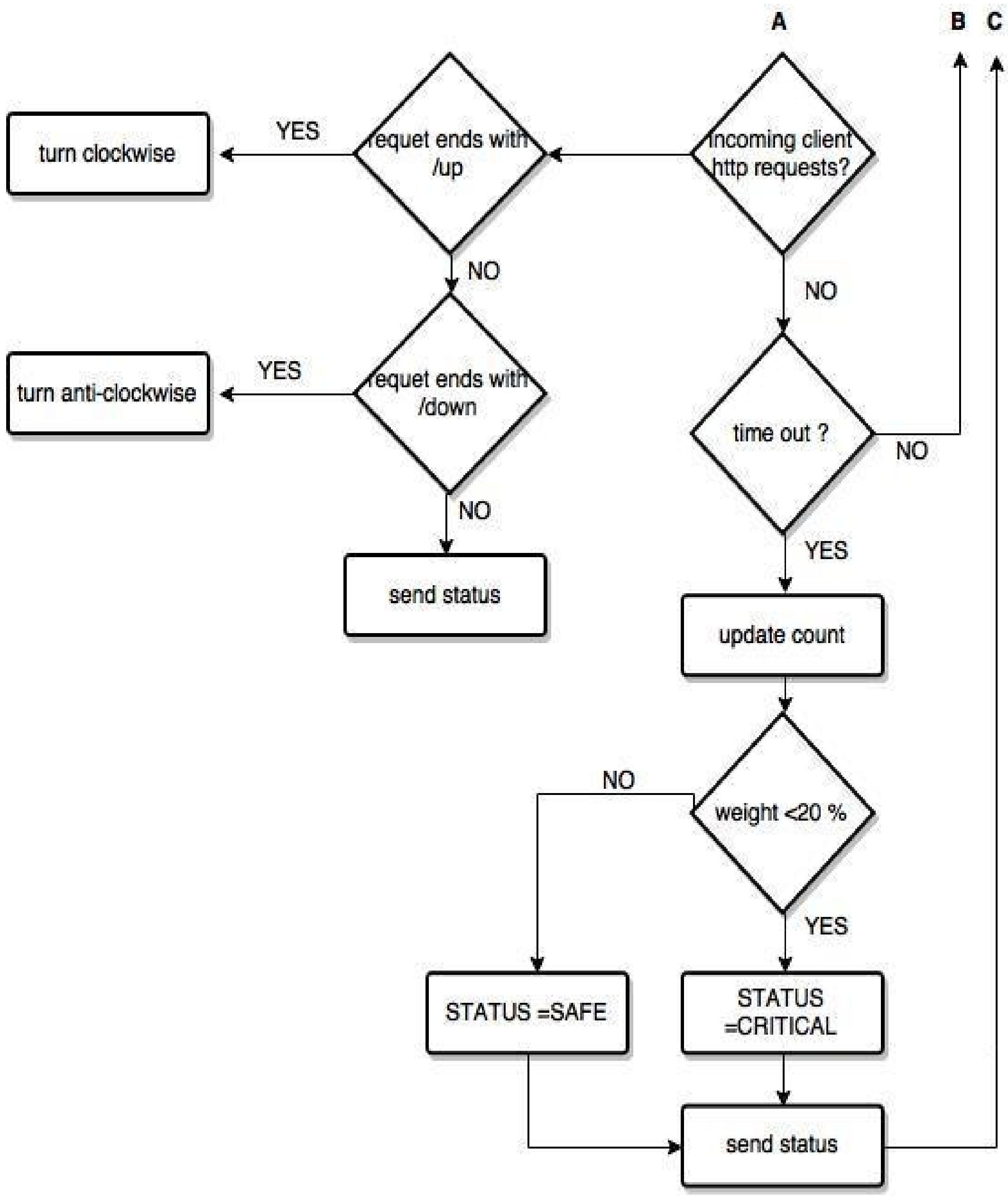
12. SOFTWARE IMPLEMENTATION

We use **energia** to interface the load sensor with CC3200 to monitor the weight of the fluid continuously .

Android application for controlling the flow rate automatically is developed using **MIT APP inventor 2.**

12.1 Flow chart





12.2 Energia

Simplelink WiFi enabled boards can use the WiFi library to connect to the internet. It can serve as either a server accepting incoming connections or a client making outgoing ones.

The library supports WEP and WPA2 Personal encryption, but not WPA2 Enterprise.

Energia communicates with the WiFi network processor using the SPI bus.

The WiFi library is very similar to the Ethernet library, and many of the function calls are the same.

The WiFi class initializes the ethernet library and network settings.

- begin()
- disconnect()
- config()
- setDNS()
- SSID()
- BSSID()
- RSSI()
- encryptionType()
- scanNetworks()
- getSocket()
- macAddress()

IPAddress class

The IPAddress class provides information about the network configuration.

- localIP()
- subnetMask()
- gatewayIP()

Server class

The Server class creates servers which can send data to and receive data from connected clients (programs running on other computers or devices).

- Server
- WiFiServer()
- begin()
- available()
- write()
- print()
- println()

Client class

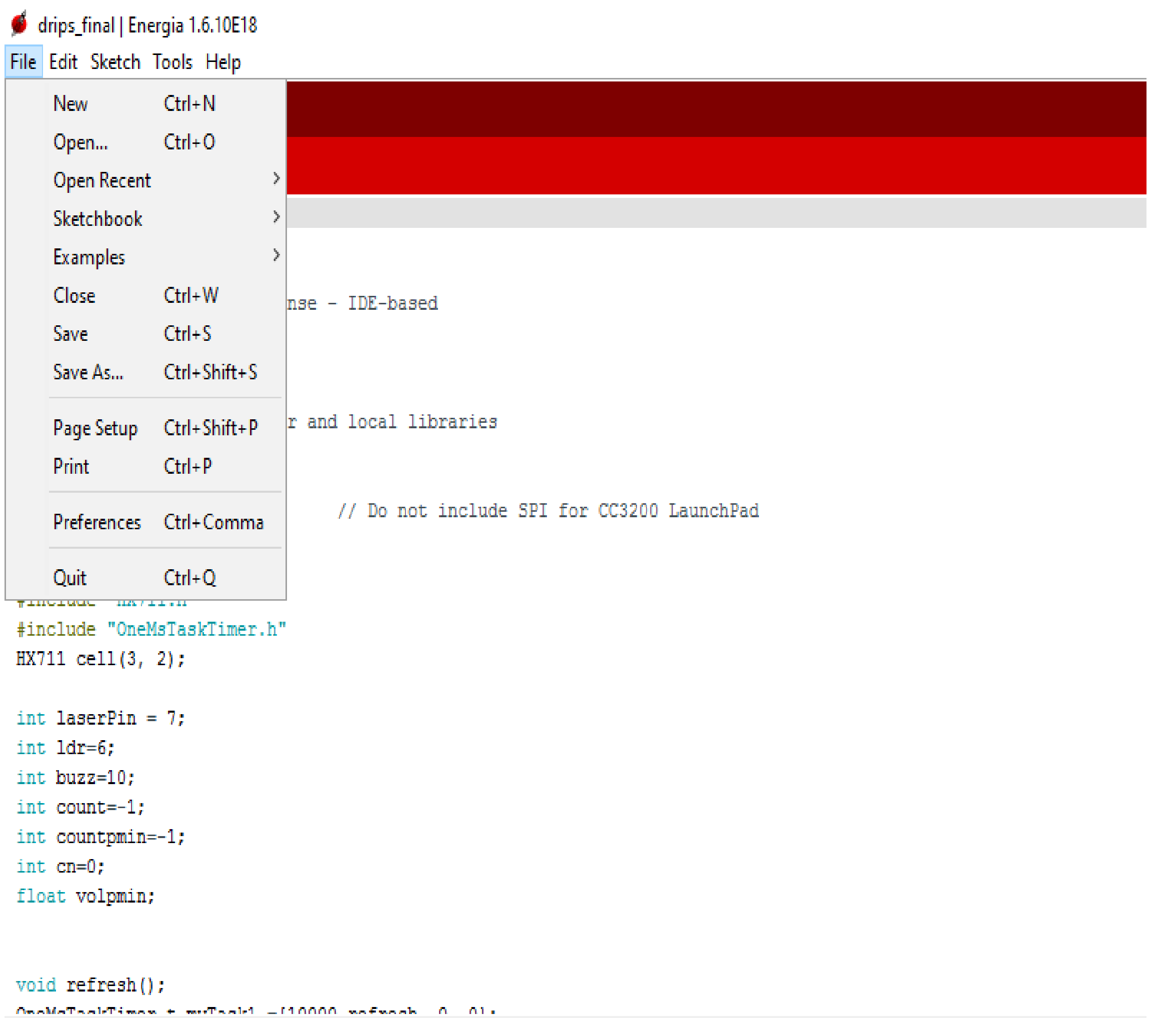
The client class creates clients that can connect to servers and send and receive data.

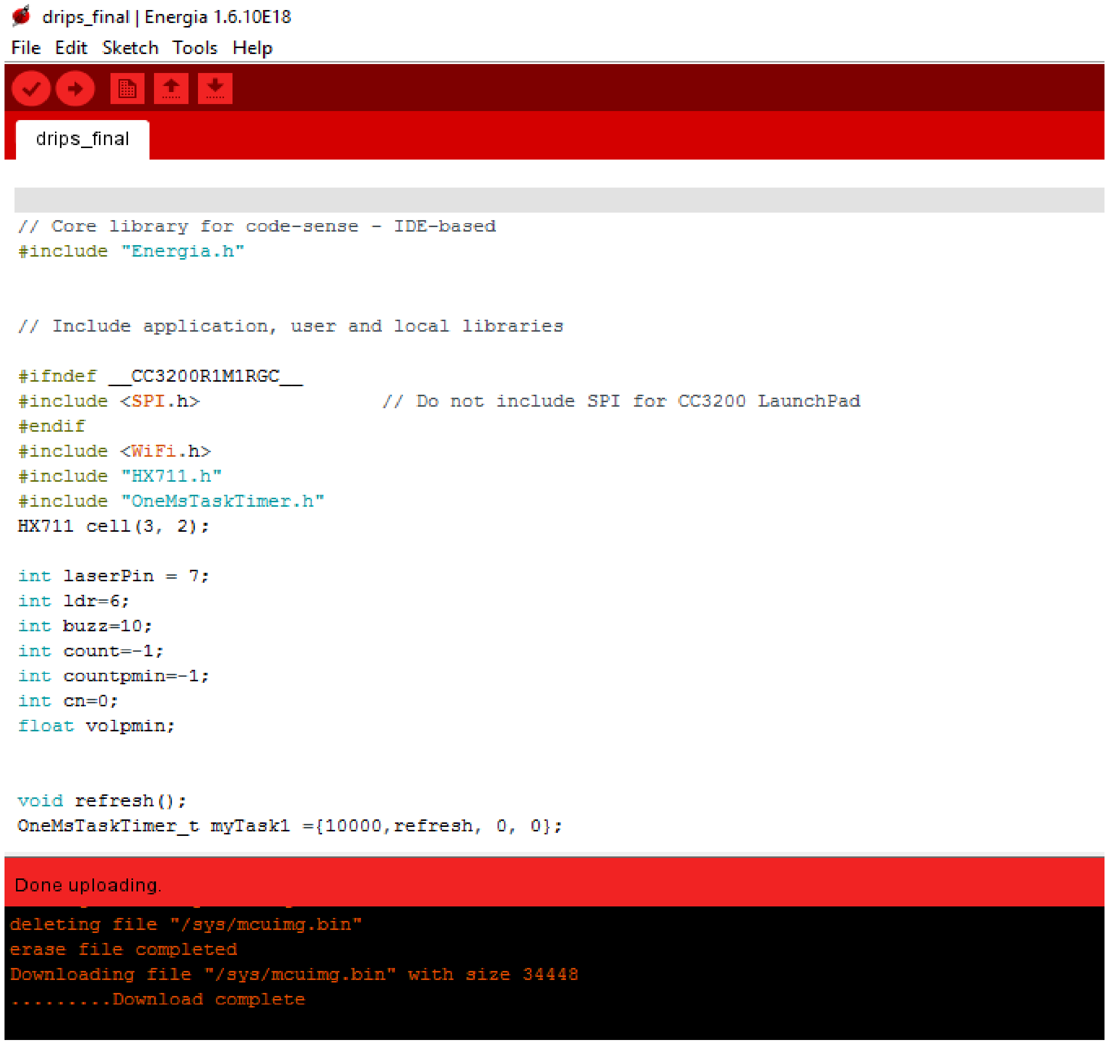
- Client
- WiFiClient()
- connected()
- connect()
- write()
- print()
- println()
- available()
- read()
- flush()

- stop()

These are some of the wifi class which will be used in energia.

The figure below shows an example of how we have used energia in our project



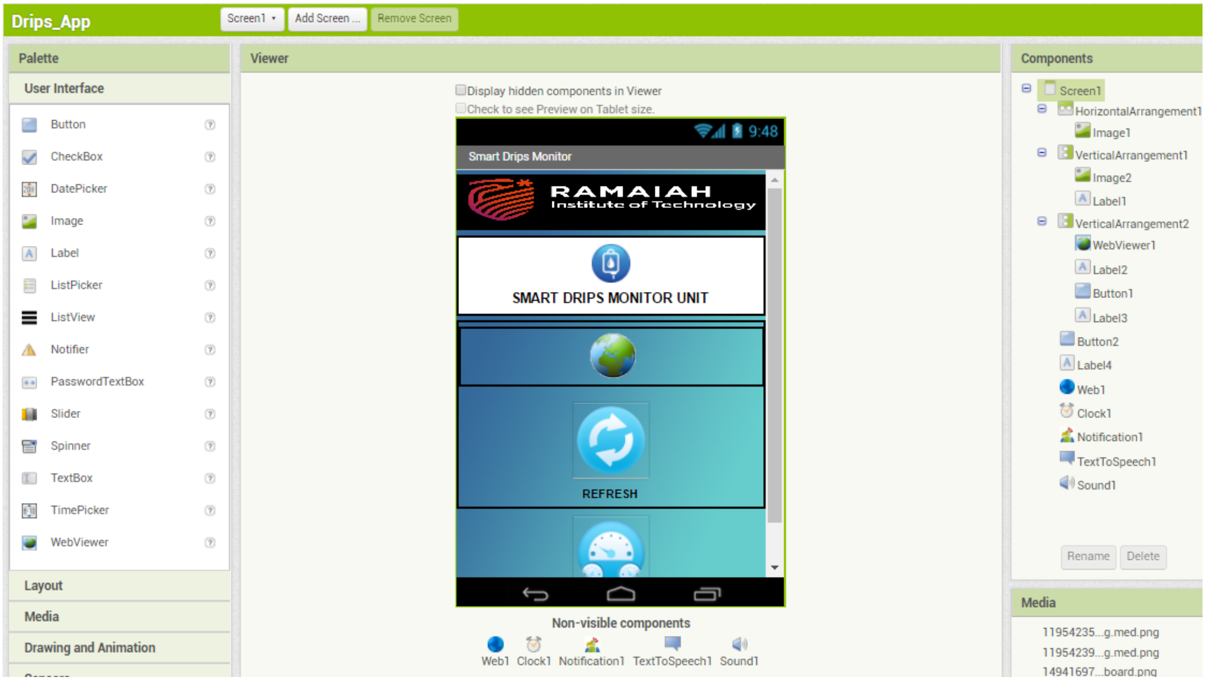



```
// Core library for code-sense - IDE-based
#include "Energia.h"


// Include application, user and local libraries

#ifndef __CC3200R1M1RGC__
#include <SPI.h>                // Do not include SPI for CC3200 LaunchPad
#endif
#include <WiFi.h>
#include "HX711.h"
#include "OneMsTaskTimer.h"
HX711 cell(3, 2);

int laserPin = 7;
int ldr=6;
int buzz=10;
int count=-1;
int countpmin=-1;
int cn=0;
float volpmin;


void refresh();
OneMsTaskTimer_t myTask1 ={10000,refresh, 0, 0};
```

```
deleting file "/sys/mcuimg.bin"
erase file completed
Downloading file "/sys/mcuimg.bin" with size 34448
.........Download complete
```

Figure 16

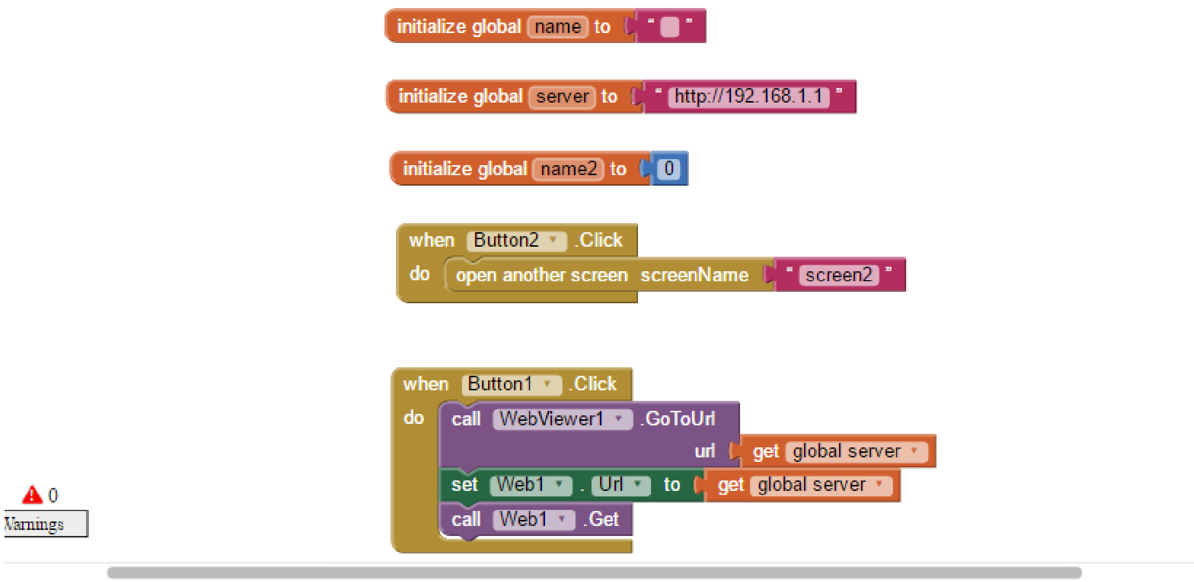
12.3 MIT App Inventor

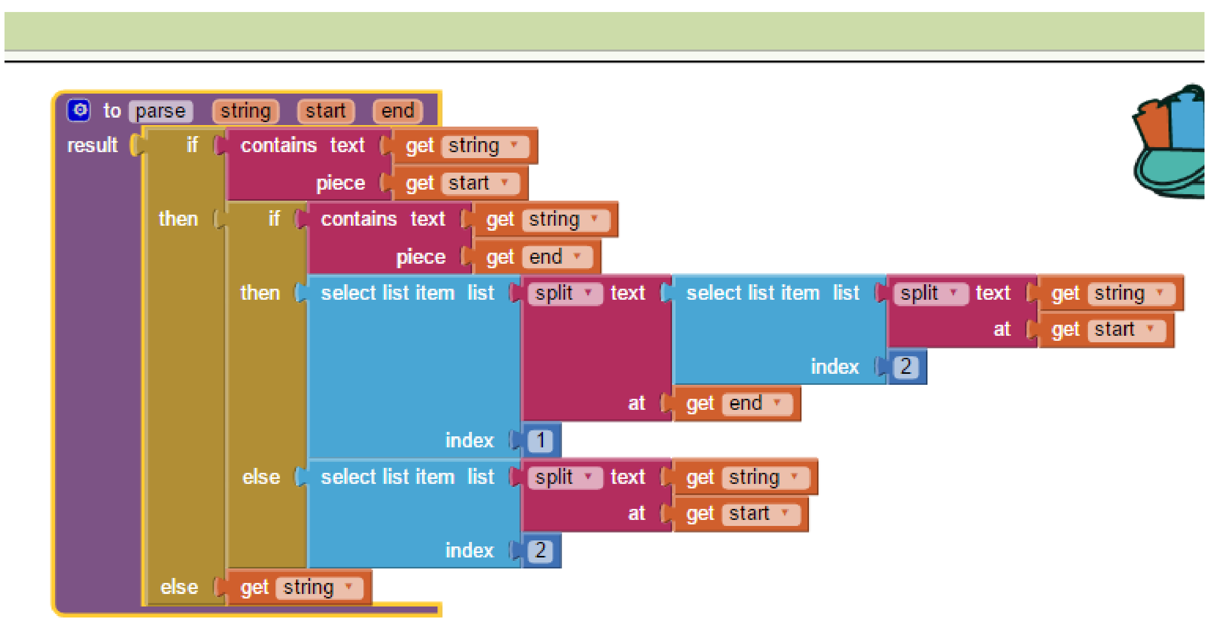


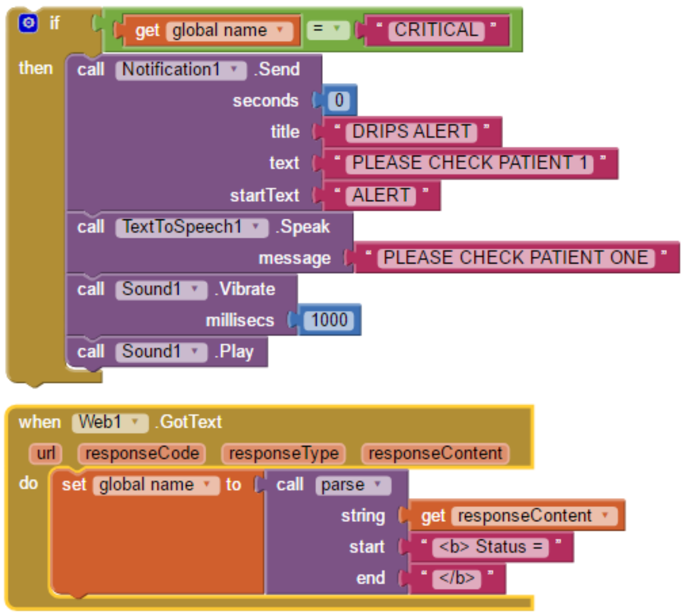
The figure above shows the front end of the Android application which is used to monitor the weight of the fluid as well as the flow rate of the drips.

It has a logo and a web browser component which will be redirected to the server to monitor the contents of the web page hosted by the sever.

It has two image buttons which is **refresh** and **update .** Once the refresh button is pressed, the web browser requests the server to resend the web page. Update button opens another screen .


initialize global name to " "
initialize global server to " http://192.168.1.1 "
initialize global name2 to 0
when Button2 .Click
do open another screen screenName " screen2 "
when Button1 .Click
do call WebViewer1 .GoToUrl
url get global server
set Web1 . Url to get global server
call Web1 .Get
0
Warnings


to parse string start end
result if contains text get string
piece get start
then if contains text get string
piece get end
then select list item list split text select list item list split text get string
at get start
index 2
at get end
index 1
else select list item list split text get string
at get start
index 2
else get string

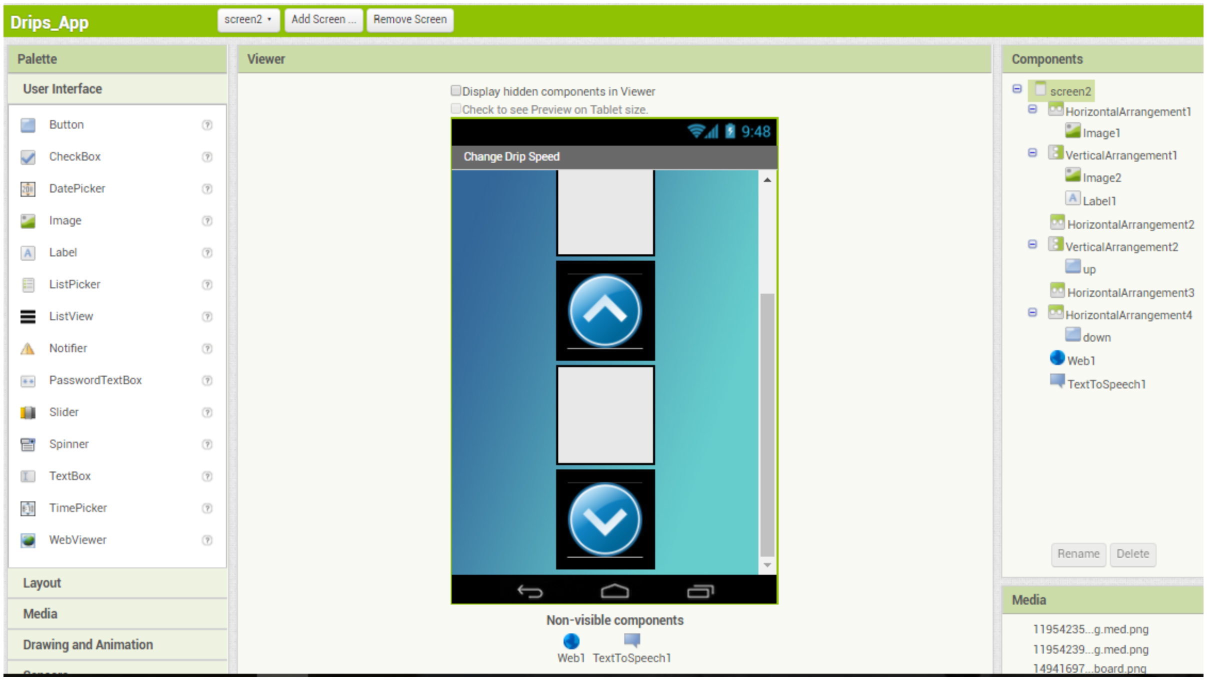


]

The figure above shows the controlling aspects using the block .

A global name is initialised for the server URL. When the button 1 is clicked , server URL is set to the web browser and it sends a HTTP GET request with the URL to get the webpage required from the server. When the web browser gets the web page , **parse** procedural block is called which is used to convert the HTML format to text format. Start and End tags are passed to get a specific content to check the status .If the status **is SAFE** ,it will simply display the web page. If the status is **CRITICAL**, it pops up a voice notification with vibration.

When the button 2 is pressed, it opens another screen by closing the current screen.


Drips_App
screen2
Add Screen ...
Remove Screen
Palette
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Viewer
Display hidden components in Viewer
Check to see Preview on Tablet size.
9:48
Change Drip Speed
Non-visible components
Web1
TextToSpeech1
Components
screen2
HorizontalArrangement1
Image1
VerticalArrangement1
Image2
Label1
HorizontalArrangement2
VerticalArrangement2
up
HorizontalArrangement3
HorizontalArrangement4
down
Web1
TextToSpeech1
Rename
Delete
Media
11954235...g.med.png
11954239...g.med.png
14941697...board.png

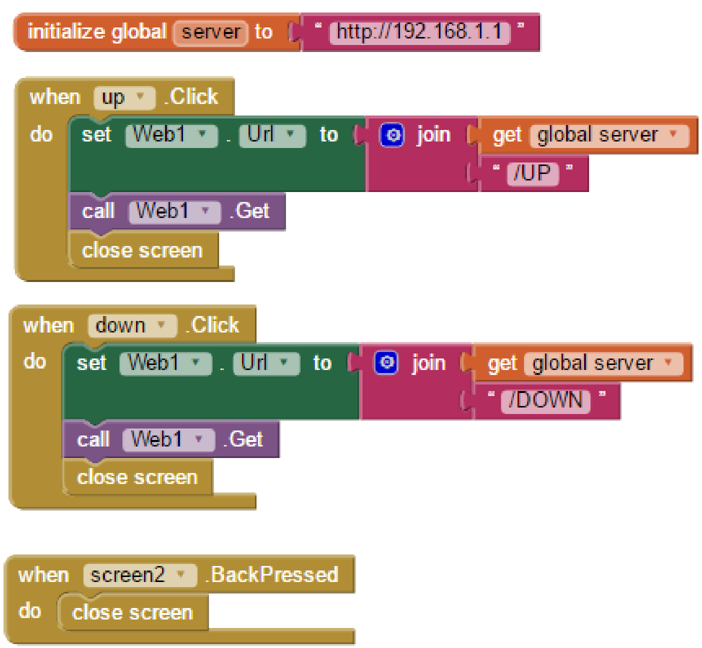

initialize global server to " http://192.168.1.1 "
when up .Click
do set Web1 . Url to join get global server
" /UP "
call Web1 .Get
close screen
when down .Click
do set Web1 . Url to join get global server
" /DOWN "
call Web1 .Get
close screen
when screen2 .BackPressed
do close screen

Figure 17

The figure above shows second screen of the android application which is used to control the flow rate of the drips.

It has two buttons where the first button is used to increase the flow rate by one unit and the second button is used to decrease the flow rate by one unit.

When the first button is pressed , the web component will be set with the server URL and **up tag** . It then sends a HTTP GET request to the server with the same URL having the **up tag** . The current screen will be closed returning to the first screen.

Similarly when the second button is pressed, the web component will be set with the server URL and **down tag** . It then sends a HTTP GET request to the server with the same URL having the **down tag** . The current screen will be closed returning to the first screen.

CHAPTER 13

13. CASE STUDY

Death rates from drug overdose and other forms of accidental poisoning among young adults more than tripled between 1999 and 2014, rising from 3.8 deaths per 100,000 to 12.8 deaths per 100,000. Almost 3,000 young adults died as a result of accidental poisoning in 2014, accounting for 15 percent of all young adult deaths.

In 2014, 96 percent of all accidental poisoning deaths among young adults were due to drug overdoses.New guidelines from the health watchdog the National Institute for Health and Care Excellence (NICE) on the use of intravenous (IV) drips have prompted a flurry of headlines, with The Guardian reporting that, "Thousands of patients [are] dying from incorrect use of IV drips", and The Daily Telegraph claiming that, "Tens of thousands harmed in hospital by IV drip blunders".

What are intravenous (IV) drips used for?

Intravenous (IV) drips are used for a variety of reasons, including:

giving daily fluid to people who cannot drink water, for example during surgery or because they are vomiting to replace lost fluid, such as from bleeding or severe diarrhoea to correct chemical or metabolic imbalance, such as having too much potassium inside the body to manage hydration if fluid is not being distributed around the body normally.

The amount and composition of fluid given depends on a person's weight, physical condition, medical problems and medication.

What has prompted the warning?

The National Institute for Health and Care Excellence (NICE) has issued guidance on when and how IV fluids should be prescribed.

It states that there is evidence that as many as one in five patients on IV fluids and electrolytes (salts and minerals) suffer complications because of inappropriate use.

What are the risks of incorrect use?

If a patient receives too little fluid, it increases the risk of dehydration and in severe cases can lead to kidney failure.

The risks of receiving too much fluid often depend on individual circumstances, but they can include:

- excess fluid collecting inside the lungs, which can cause breathing difficulties and increased risk of pneumonia
- swelling of the ankles
- an imbalance of electrolytes in the blood, which can disrupt organs
- heart failure in serious cases.

CHAPTER 14

14 . RESULTS AND DISCUSSIONS

The following results were obtained and the pictures of the android application developed using MIT APP INVENTOR 2 is as shown below

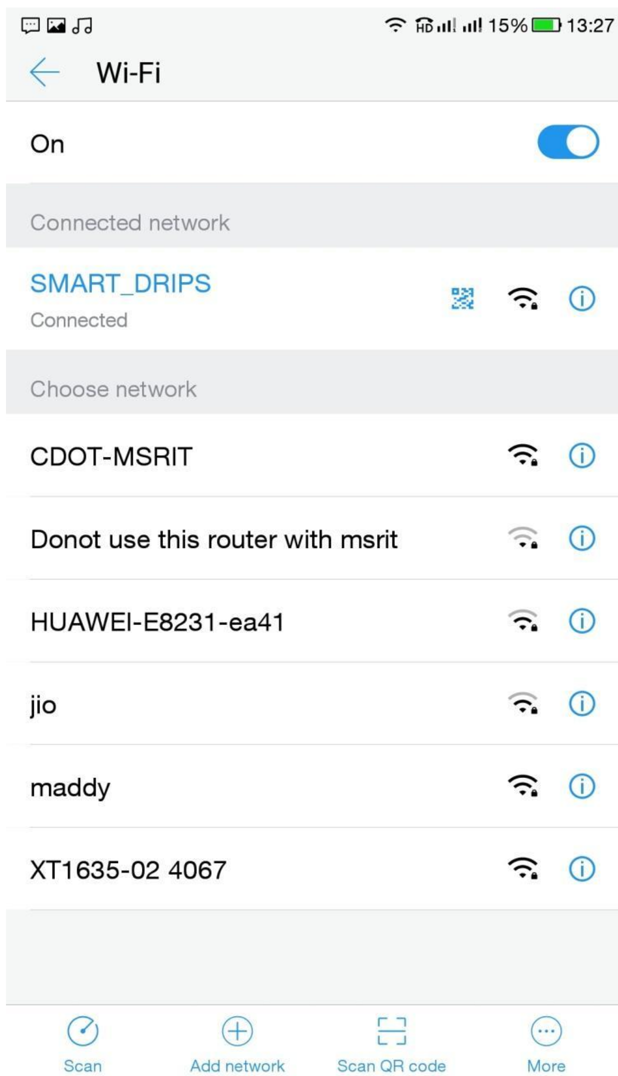


Figure 18

The wifi named “SMART_DRIPS”which was created using CC3200 is marked in the above figure.

We connect to this wifi network and the android application is opened using this wifi network.

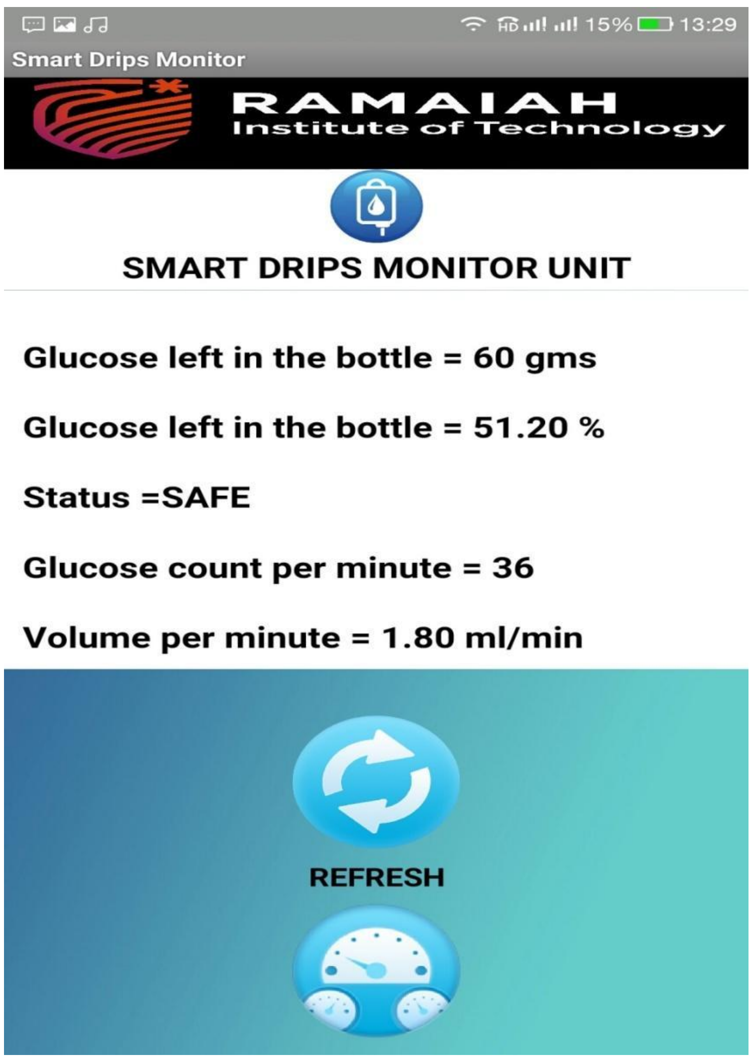


Figure 19

The picture of the android application is shown above where the status of the patient is "SAFE" because the glucose level is not below 20% or 10%. The weight of the fluid left and the glucose count monitored continuously is also shown in the above figure.

Figure 20

The above figure shows the status of the patient as "CRITICAL" because the threshold level is below 20%.

CHAPTER 15

15. CONCLUSION AND FUTURE SCOPE

CONCLUSION

Thus an android application is developed which is user friendly and is easily accessible where a voice notification is popped up with vibration which will alert the nurse .

Thus nurse/assist need not check up on the patients constantly . The application also provides the provision of controlling the flow rate from any place within the wifi range so that the speed of the drips will be controlled as per the patient's requirement.A cost effective and low power dissipation device (embedded system) that controls infusion rate of IV fluid and performs remote health monitoring forming a closed loop system for a real-time measurement of health parameters has been developed. An integrated health information system, in order to create a health care program that would be more cost efficient or provide new treatment options has been designed. We thus evaluate the solution for patient's health monitoring from different perspectives. We look into the advantages and disadvantages for the patient, as well as look at the quality of the software solution with regards to usability, functionality and future extensibility.

FUTURE SCOPE

Miniaturized model can be developed .The system can also be used to control more than one drips bottle simultaneously. More accurate load cells can be used such that it can be tested for more than 100 cases.

The above system may be implemented in hospitals at a reasonable price.

CHAPTER 16

16. REFERENCES

[1] Takalkar Atul S,Lenin Babu M C "**Characterization of Valveless Micropump for Drug Delivery by Using Piezoelectric Effect**" IEEE International Conference on Advances in Computing, Communications and Informatics 2016.

[2]. Fan Yang, Yu Wang "**Research of the New Electronic Valve System with Quantitative Control**" IEEE Third Global Congress on Intelligent Systems, 2012.

[3]. Jingguo Wen,Zisheng Lian "**Electro-Hydraulic Control System for Hydraulic Supports About the Study on Solenoid Valve Driver**" IEEE International Conference on Computing, Measurement, Control and Sensor Network ,2012

[4]. Lu Quan, Sen Bao, Hong Li Jun "**Research on Embedded Electro-hydraulic Proportional Valve Controller**" IEEE Third International Symposium on Intelligent Information Technology Application,2009.

[5].Shuxi ng Guo,Jian Wang, Qinxue Pan and Jian Guo "**Solenoid Actuator based Novel Type of Micropump**" IEEE International Conference on Robotics and Biomimetics,2006

APPENDIX A

```
// Core library for code-sense - IDE-based

#include "Energia.h"



// Include application, user and local libraries


#ifndef __CC3200R1M1RGC__

#include <SPI.h>                // Do not include SPI for CC3200 LaunchPad

#endif

#include <WiFi.h>

#include "HX711.h"

#include "OneMsTaskTimer.h"

HX711 cell(3, 2);


int laserPin = 7;

int ldr=6;

int buzz=10;

int count=-1;
```

```
int countpmin=-1;
int cn=0;
float volpmin;


void refresh();
OneMsTaskTimer_t myTask1 ={10000,refresh, 0, 0};

// Define variables and constants
char wifi_name[] = "SMART_DRIPS";
char wifi_password[] = "launchpad";

WiFiServer myServer(80);
uint8_t oldCountClients = 0;
uint8_t countClients = 0;

int calib;
long caliberation=0;
long initialWeight=0;

// Add setup code
```

```
void setup()
{
  Serial.begin(115200);
  delay(500);

  Serial.println("*** LaunchPad CC3200 WiFi Web-Server in AP Mode");

  // Start WiFi and create a network with wifi_name as the network name
  // with wifi_password as the password.
  Serial.print("Starting AP... ");
  WiFi.beginNetwork(wifi_name, wifi_password);
  while (WiFi.localIP() == INADDR_NONE)
  {
    // print dots while we wait for the AP config to complete
    Serial.print('.');
    delay(300);
  }
  Serial.println("DONE");

  Serial.print("LAN name = ");
  Serial.println(wifi_name);
```

```
  Serial.print("WPA password = ");
  Serial.println(wifi_password);




   pinMode(8,OUTPUT);
 pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  digitalWrite(8,LOW);
  pinMode (laserPin, OUTPUT);
  pinMode(ldr, INPUT);
   pinMode(buzz,OUTPUT);
   digitalWrite(buzz,LOW);
    OneMsTaskTimer::add(&myTask1);

  IPAddress ip = WiFi.localIP();
  Serial.print("Webserver IP address = ");
  Serial.println(ip);

  Serial.print("Web-server port = ");
  myServer.begin();                           // start the web server on port 80
```

```
  Serial.println("80");
  Serial.println("Dont keep any weights . please wait for initialization . . . . .");
for(int i=0;i<25;i++){

calib=cell.read();
// Serial.println("wait for intialization . . . . .");
// Serial.println((calib-8151200)/8345.0f*31);
caliberation=((calib-8178386)/7930.0f*35);
}
Serial.println("Initialization successful. ");
Serial.println("Keep glucose bottle and wait for some time ........");
delay(1000);
 for(int i=0;i<25;i++)
 {
  delay(500);
   calib=cell.read();
  initialWeight=(((calib-8178386)/7930.0f*35)-caliberation);
 // Serial.println(initialWeight);
 }

 if(initialWeight==0) //to get ridoff divide by 0 error
```

```
   initialWeight=1;


   Serial.print("Initial weight = ");
  Serial.println(initialWeight);
  OneMsTaskTimer::start();
    Serial.println();
}


long val = 0;
long reading=0;
float percentage=0;
String Status;
// Add loop code
void loop()
{
    int sen=0;
  //analogReadResolution(10);
  digitalWrite (laserPin, HIGH);
  val = cell.read();
   int sensorValue = analogRead(ldr);
    if(sensorValue>4000)
```

```
sen=1;
 else
sen=0;
 if(sen==1)
 {
  count++;
  countpmin++;
  delay(50);
 }
 Serial.println(sen);
  // delay(100);
// Serial.println(count);
   delay(100);
  countClients = WiFi.getTotalDevices();


  // Did a client connect/disconnect since the last time we checked?
  if (countClients != oldCountClients)
  {
      if (countClients > oldCountClients)
      {
        //          digitalWrite(RED_LED, !digitalRead(RED_LED));
```

```
      Serial.println("Client connected to AP");
      for (uint8_t k = 0; k < countClients; k++)
      {
        Serial.print("Client #");
        Serial.print(k);
        Serial.print(" at IP address = ");
        Serial.print(WiFi.deviceIpAddress(k));
        Serial.print(", MAC = ");
        Serial.println(WiFi.deviceMacAddress(k));
        Serial.println("CC3200 in AP mode only accepts one client.");
      }
    }
    else
    {  // Client disconnect
      //          digitalWrite(RED_LED, !digitalRead(RED_LED));
      Serial.println("Client disconnected from AP.");
      Serial.println();
    }
    oldCountClients = countClients;
  }
```

```
WiFiClient myClient = myServer.available();


if (myClient)

{                          // if you get a client,

   Serial.println(". Client connected to server");        // print a message out the serial port

   char buffer[150] = {0};              // make a buffer to hold incoming data

   int8_t i = 0;

   while (myClient.connected())

   {          // loop while the client's connected

      if (myClient.available())

      {           // if there's bytes to read from the client,

         char c = myClient.read();           // read a byte, then

    //   Serial.write(c);                 // print it out the serial monitor

         if (c == '\n') {                 // if the byte is a newline character


            // if the current line is blank, you got two newline characters in a row.

            // that's the end of the client HTTP request, so send a response:

            if (strlen(buffer) == 0)

            {

                myClient.println("HTTP/1.1 200 OK");

               myClient.println("Content-type:text/html");
```

```
myClient.println("refresh : 30;"); //refresh after every 15 second

myClient.println();

  myClient.print("<br/><b> Glucose left in the bottle = ");

 myClient.println(reading);

 myClient.println(" gms </b></br>");

 myClient.println("");

  myClient.println("<br/><b> Glucose left in the bottle =");

 myClient.println(percentage);

 myClient.println(" % </b></br>");

  myClient.print("<br/><b> Status =");

 myClient.print(Status);

 myClient.print("</b></br>");

  myClient.println("<br/><b> Glucose count per minute =");

 myClient.println(cn);

 myClient.println("  </b></br>");

  myClient.println("<br/><b> Volume per minute =");

 myClient.println(volpmin);

 myClient.println(" ml/min </b></br>");



  myClient.println();
```

```
        // the content of the HTTP response follows the header:



        // The HTTP response ends with another blank line:
        myClient.println();
        // break out of the while loop:
        break;
      }
      else
      {     // if you got a newline, then clear the buffer:
        memset(buffer, 0, 150);
        i = 0;
      }
    }
    else if (c != '\r')
    {   // if you got anything else but a carriage return character,
      buffer[i++] = c;      // add it to the end of the currentLine
    }
```

```
            Serial.println();
            String text = buffer;
            // Check to see if the client request was "GET /H" or "GET /L":
            if (text.endsWith("GET /UP"))
            {
              digitalWrite(8,HIGH); //1-->+ 2--> -ve
digitalWrite(4,HIGH); //4-> 1n1
digitalWrite(5,LOW);//5->in2
delay(1000); //anticlockwise 1 sec
  digitalWrite(8,LOW);
            }
            if (text.endsWith("GET /DOWN"))
            {
                digitalWrite(8,HIGH); //1-->+ 2--> -ve
digitalWrite(4,LOW); //4-> 1n1
digitalWrite(5,HIGH);//5->in2
delay(1000); //anticlockwise 1 sec
  digitalWrite(8,LOW);
            }


        }
```

```
        }


        // close the connection:

        myClient.stop();

        Serial.println(". Client disconnected from server");

        Serial.println();

    }

}

void refresh()

{

  reading=(((val-8178386)/7930.0f*35)-caliberation); //caliberation

percentage=((float)reading/initialWeight)*100;    //calculate %

delay(10);

//percentage=10;

Serial.print("weight :  ");

  Serial.print( reading);

  Serial.println(" gms");

  Serial.print("percentage left :  ");

  Serial.print( percentage);

  Serial.println(" %");

  if(percentage<20.0)
```

```
  {
  Status = "CRITICAL";
 // buzzer();
  } else
  {
    Status="SAFE";
  }
  Serial.print("Count per minute : ");
  Serial.println(countpmin*6);
  cn=countpmin*6;
   Serial.print("volume per minute : ");
  volpmin=countpmin*6*0.05;
  Serial.println(volpmin);


  countpmin=0;
}
void buzzer()
{
  for(int m=0;m<50;m++)
  {
    digitalWrite(10,HIGH);
```

```
  delay(50);
   digitalWrite(10,LOW);
  delay(50);
  }
}
```