HEALTH MONITORING USING IOT , MEDICINE REMINDER USING GSM

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

K.Srinivas 3151265121078 P.Yogitha 315126512140

M.Sindhu 315126512103 M.Rohith 315126512085

Under the guidance of

G.Gayatri

M.Tech (Asst.Professor)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)

Sangivalasa, bheemili mandal, visakhapatnam dist.(A.P)

2018-2019

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **G.Gayatri** M.Tech., Department of Electronics and Communication Engineering, ANITS, for her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped usin accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITSfor providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

K.Srinivas (313126512078),
P.Yogitha (313126512140),
M.Sindhu (313126512103),
M.Rohit (313126512085),

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)

Sangivalasa, bheemilimandal, visakhapatnam dist.(A.P)

PRAGNANAM BRAHMA

ANITS

CERTIFICATE

This is to certify that the project report entitled **"HEALTH MONITORING USING IOT, MEDICINE REMINDER USING GSM "** submitted by **K.Srinivas (315126512078), P.Yogitha (315126512140), M.Sindhu (315126512101), M.Rohit (315126512085)** in partial fulfillment of the requirements forthe award of the degree of **Bachelor of Technology**in **Electronics & Communication Engineering** ofAndhra University, Visakhapatnam is a record of bonafide work carried out under my guidanceand supervision.

Project Guide	**Head of the Department**
[illegible]	[illegible]
G.Gayatri	**Dr. V.Rajyalakshmi**
M.Tech	**M.E,Ph.D,MIEEE,MIE,MIETE**
Department of E.C.E	Department of E.C.E
ANITS	ANITS

CONTENTS

ABSTRACT

In the contemporary day life style people have no time to spend with their famil.In such a busy life it’s difficult to keep an isolated day out of their busy schedule for the doctor for consistent medical checkup and taking medicines at time.Their is a necessity for new idea and technology which helps in saving their time.

The proposed model enables users to improve health related risks and reduce healthcare costs by reminding to take medicines at time,collecting,recording and analyzing data in real time efficiently.With the help of this proposal the time of both patients and doctors are saved and doctors can also help in emergency scenario as much as possible.The proposed outcome of the project is to give proper and efficient medical services to patients by remiding them when to take medicines and collecting data information through health status monitors which would include patient’s Heart rate.

LIST OF SYMBOLS

X1,X2 – Pins used to set frequency of device

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

RTC	Real time module
GSM	Global system for mobile
HTTP	Hypertext transfer protocol
IOT	Internet of things
AT	Attention
SCL	Serial clock
SDA	Serial data

CHAPTER-1

INTRODUCTION

1.1 MEDICINE REMINDER

Patient monitoring and management in critical care environments such as the ICU’s , SICU’s and ANCU’s involve estimating the status of the patient and reacting to events that may be life threatening. It is impossible to keep a tab on every patient throughout the day. New solutions are needed in this field to help the doctors and the nursing staff to monitor the patients. A critical element of this is the medicine administration and monitoring. This has been achieved by the patient medicine reminder system. This system consists of Arduino,GSM Module, RTC Module. This system is driven by an program that inputs predefined parameters which is processed based on the input variables entered via a user interface device such as the PC.. The logic for the processing is built into the embedded program to initiate the alert through an audio alarm. Not only does it have an alarm system, but also gives indication when medicine is not taken at the reminder time.

1.2 CARDIAC MONITORING

Heart rhythm disorders, also called cardiac arrhythmias, occur when there is a malfunction in the heart's electrical impulses that coordinate how it beats. As a result, the heart can beat too quickly, too slowly or irregularly. In some cases, abnormal heart rhythms are not serious or life threatening and can be addressed with simple lifestyle changes. In other cases - such as when a patient experiences recurrent fainting, palpitations, unexplained stroke or a trial fibrillation arrhythmias can be serious and potentially life threatening. Importance of Cardiac Monitoring because abnormal heart rhythms and their accompanying cardiac symptoms often come and go in a transient manner, they may be difficult to detect. Tests such as electrocardiograms only allow a physician to look at the heart's activity at one point in time, and a patient may be at risk for future symptoms or events that were not detected at the time the test was administered. To determine the cause of recurrent fainting, palpitations, unexplained stroke or atrial fibrillation, a patient's heart must be monitored over time so his or her physician can diagnose the disorder accurately.

Cardiac monitors are battery-powered devices that record the heart's electrical activity. There are two categories of cardiac monitors: external cardiac monitors, and umplantable or insertable cardiac monitors. External Cardiac Monitors Designed for short-term use, traditional external heart monitors (known as Holter or Event Monitors) are the most common monitors for diagnosing heart rhythm disorders that occur on a relatively frequent basis. They are typically attached with wires to the outside of a patient's body for between 24 hours to 30 days. Other types of extemal monitors include Mobile Cardiac Telemetry (MCT) or Mobile Cardiac Outpatient Telemetry (MCOT), which also can be worn for up to 30 days. These devices monitor, record and store cardiac data, which can be reviewed and sent to physicians for appropriate follow up. Implantable and Insertable Cardiac Monitors Designed for long-term use, implantable and insertable cardiac monitors are devices that help determine the causes of infrequent, unexplained arnythmias. The small devices are placed just under the skin of the chest during outpatient procedures. The devices detect and record abnormal heart rhythms over long periods of time (up to three years) to help determine whether patients have abnormal heart rhythms.

1.2.1 HEART RATE

What does the heart rate signify? It's a window into your muscles and lungs; it reveals how hard they are working. Your heart pounds to pump oxygen-rich blood to your muscles and to carry cell waste products away from your muscles. The more you demand of your muscles the harder your heart has to work to perform these tasks. That means your heart must beat faster to deliver more blood. The heart rate gives a good indication during exercise routines of how effective that routine is improving your health. Once only used by elite athletes, heart rate monitors are now becoming an essential tool for everyone from the casual athlete to the personal trainer. Heart monitors provide an easy and scientific measure of the effort you are putting into your workouts. A heart rate monitor is simply a device that takes a sample of heartbeats and computes the beats per minute so that the information can easily be used to track heart condition.

Current technology consists of optical and electrical monitors. The electrical method provides a bulky strap around one's chest. The optical method does not require the strap and can be used more conveniently than the electrical method. Heartbeat sensor provides a simple way to study the function of the heart which can be measured based on the principle of psycho-physiological signal used as a stimulus for the virtual- reality system. The amount of the blood in the finger changes with respect to time. When it comes to your heart, timing is everything. Without a strong heartbeat, your bloodcannot get to where it needs to go, and to have a strong heartbeat, it must be steady. Even if you're not an athlete, knowledge about your heart rate can help you monitor your fitness level and it might even help you spot developing health problems. Your heart rate, or pulse, is the number of times your heart beats per minute. Normal heart rate varies from person to person. Knowing yours can be an important heart-health gauge.As you age, changes in the rate and regularity of your pulse can change and may signify a heart condition or other condition that needs to be addressed.

CHAPTER-2

HAREWARE COMPONENTS

Chapter 2

2.1 ARDUINO UNO

Fig 2.1 AURDUINO UNO BOARD

2.1.1 DESCRIPTION

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassemble form, or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (/O) pins that may be interfaced to various expansion boards or Breadboards (shields) and other circuits.The boards feature

serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers.
The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the arduino project provides an Integrated Development Environment (IDE) based on the Processing language project.

2.1.2 FEATURES:

1)Atmega328 Microcontroller
2)Input voltage 7-12V
3)14 Digital I/O pins (6 PWM outputs)
4)6 Analog Inputs
5)32K Flash Memory
6)16Mhz clock speed

2.1.3 ADVANTAGES:

1)Inexpensive-Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50.
2)Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows
3)Simple, clear programming environment - The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.

4) Open source and extensible software -The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language.

5)Open source and extensible hardware -The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

2.2 REAL TIME CLOCK MODULE

A real-time clock (RTC) is an IC that keeps an updated track of the current time.

This information can be read by a microprocessor, usually over a serial interface to facilitate the software performing functions that are time dependent. RTCs are designed for ultra-low power consumption as they usually continue running when the main system is powered down. This enables them to maintain current time against an absolute time reference, usually set by the microprocessor directly. Figure 1 depicts the typical internal workings of a simple RTC.

2.2.1 FEATURES

- Consumes less than 500nA in battery backup mode with oscillator running
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized
- Real-time clock (RTC) counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 56-byte non-volatile RAM for data storage
- Two-wire interface (I2C)

2.2.2 PIN DIAGRAM

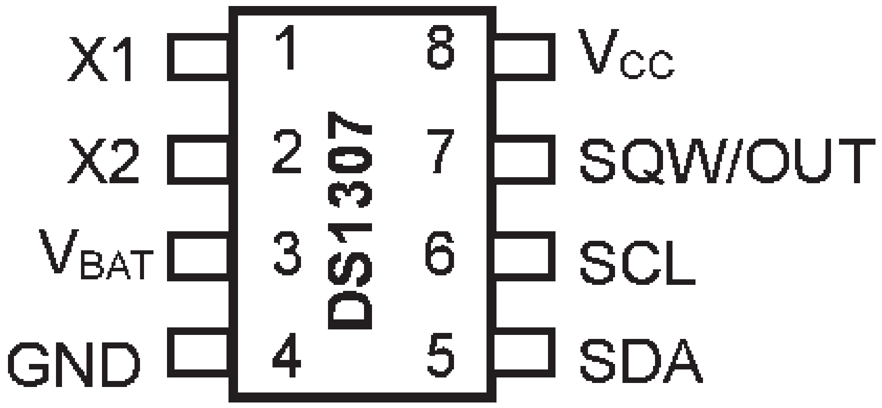


Fig: 2.2 Pin Description of DS1307

Pin 1, 2: Connections for standard 32.768 kHz quartz crystal. The internal oscillator circuitry is intended for operation with a crystal having a specified load capacitance of 12.5pF. X1 is the input to the oscillator and can alternatively be connected to an external 32.768 kHz oscillator. The output of the internal oscillator, X2 is drifted if an external oscillator is connected to X1.

Pin 3: Battery input for any standard 3V lithium cell or other energy source. Battery voltage should be between 2V and 3.5V for suitable operation. The nominal write protect trip point voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x VBAT nominal. A lithium battery with 48mAhr or greater will backup the DS1307 for more than 10 years in the absence of power at 25ºC. UL recognized to ensure against reverse charging current when utilized as a part of conjunction with a lithium battery.

Pin 4: Ground.

Pin 5: Serial data input/output. The input/output for the I2C serial interface is the SDA, which is open drain and requires a pull up resistor, allowing a pull up voltage upto 5.5V. Regardless of the voltage on VCC.

Pin 6: Serial clock input. It is the I2C interface clock input and is used in data synchronization.

Pin 7: Square wave/output driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4 kHz, 8 kHz, and 32 kHz). This is also open drain and requires an external pull-up resistor. It requires application of either Vcc or Vb at to operate SQW/OUT, with an allowable pull up voltage of 5.5V and can be left floating, if not used.

Pin 8: Primary power supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and VCC is below VTP, read and writes are inhibited. However at low voltages, the timekeeping function still functions.

Using the DS1307 is primarily written to and read the registers of this chip. The memory contains all 64 DS1307 8-bit registers are addressed from 0 to 63 (from 00H to 3FH the hexadecimal system). The first eight registers are used for the clock register the remaining 56 vacant can be used as RAM contains temporary variable if desired. The first seven registers contain information about the time of the clock including: seconds, minutes, hours, secondary, date, month and year. The DS1307 include several components such as power circuits, oscillator circuits, logic controller and I2C interface circuit and the address pointer register (or RAM). Let's see the working of DS1307.

2.2.3 WORKING OF RTC MODULE

In the simple circuit the two inputs X1 and X2 are connected to a 32.768 kHz crystal oscillator as the source for the chip. VBAT is connected to positive culture of a 3V battery chip. Vcc power to the I2C interface is 5V and can be given using microcontrollers. If the power supply Vcc is not granted read and writes are inhibited.

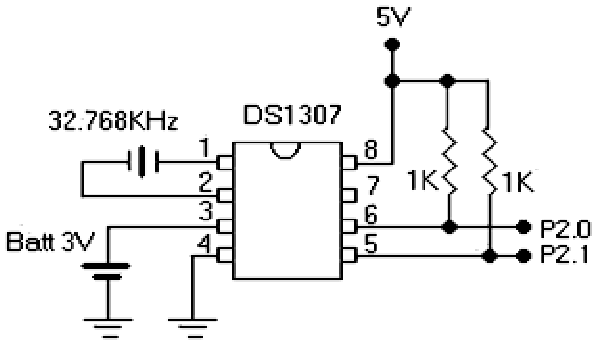


Fig 2.3 Connections of rtc module

START and STOP conditions are required when a device wants to establish communication with a device in the I2C network.

- By providing a device identification code and a register address, we can implement the START condition to access the device.
- The registers can be accessed in serial order until a STOP condition is implemented

The START condition and STOP condition when the DS1307 I2C communication with the microcontroller is shown in the figure below.The device is configured mentioned in the figure below. The DS1307 has the 2-wire bus connected to two I/O port pins of the DS5000: SCL – P1.0, SDA – P1.1. The VDD voltage is 5V, RP = 5KΩ and the DS5000 is by means of a 12-MHz crystal. The other secondary device could be any other device that recognizes the 2-wire protocol, such as the DS1621 Digital Thermometer and Thermostat. The interface with the D5000 was skilled using the DS5000T Kit hardware and software. These development kits allow the PC to be used as a dumb terminal using the DS5000's serial ports to substitute a few words with the keyboard and monitor. Typical 2-wire bus arrangement, the following bus protocol has been defined during data exchange information; the data line must remain stable whenever the clock line is high. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Start data transfer: A change in the state of the data line from high to low, while the clock line is high, defines a START condition.

Stop data transfer: A change in the state of the data line from low to high, while the clock line is high, defines the STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the high period of the clock signal. The data on the line must be changed during the low period of the clock signal. There is one clock pulse per bit of data.
Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and the STOP

conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit.

2.3 MAGNETIC REED SWITCH

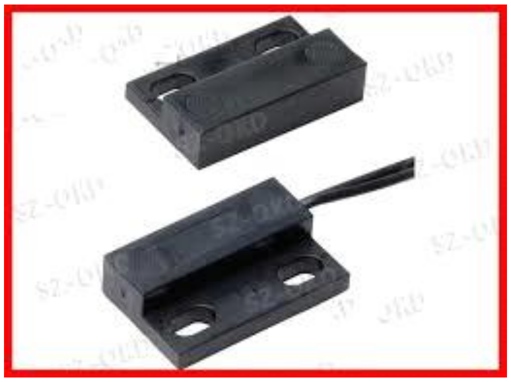


Fig 2.4 Magnetic Reed Switch

- The **Reed switch** is an electrical switch operated by an applied magnetic field.
- The contacts may be normally open, closing when a magnetic field is present, or normally closed and opening when a magnetic field is applied.
- The switch may be actuated by a coil, making a reed relay, or by bringing a magnet near the switch.
- Once the magnet is pulled away from the switch, the reed switch will go back to its original position.

2.3.1 OPERATION

Reed switches come in two main varieties called normally open (normally switched off) and normally closed (normally switched on). The key to understanding how they work is to realize that they don't just work as an electrical bridge but as a *magnetic* one as well: magnetism flows through them as well as electricity.

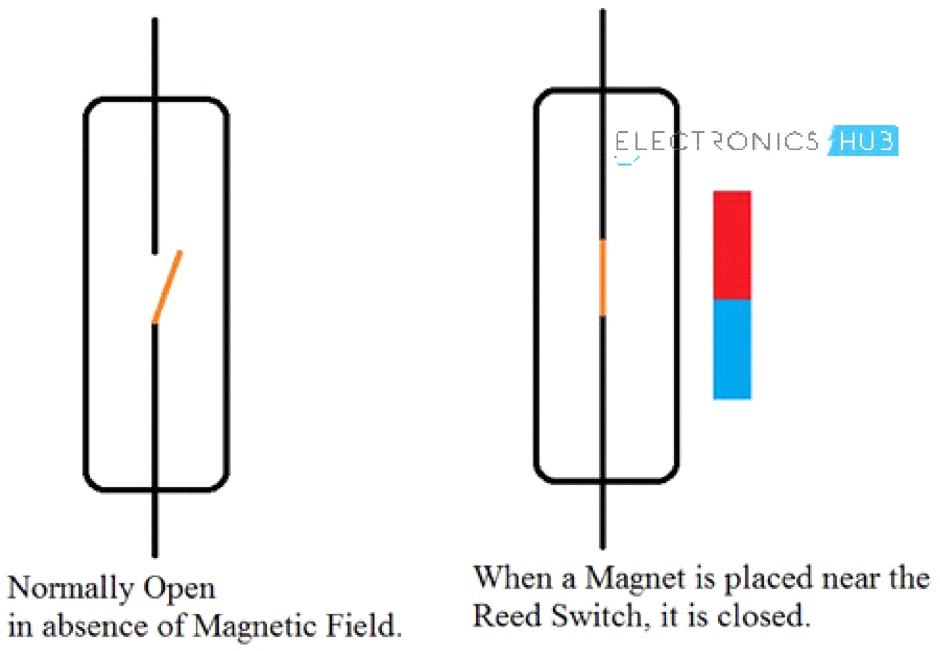

ELECTRONICS HUB
Normally Open
in absence of Magnetic Field.
When a Magnet is placed near the
Reed Switch, it is closed.

Fig 2.5 Operation of Magnetic Reed Switch

Normally open

As you bring a magnet up to the reed switch, the entire switch effectively becomes a part of a "magnetic circuit" that includes the magnet (the dotted line in the artwork shows part of the magnetic field). The two contacts of the reed switch become opposite magnetic poles, which is why they attract and snap together. It doesn't matter which end of the magnet approaches first: the contacts still polarize in opposite ways and attract one another. A reed switch like this is **normally open (NO)** (normally off), unless a magnet is positioned right next to it, when it switches on, allowing a current to flow through it.

Take the magnet away and the contacts—made from fairly stiff and springy metal—push apart again and return back to their original positions.

2.4 GSM MODULE

- GSM is a mobile communication modem.It is widely used mobile communication system in the world. GSM is an open and digital cellular technology used for transmitting mobile voice and data services.
- A GSM modem is a device which can be either a mobile phone or a modem device which can be used to make a computer or any other processor communicate over a network

A GSM modem requires a SIM card to be operated and operates over a network range subscribed by the network operator.

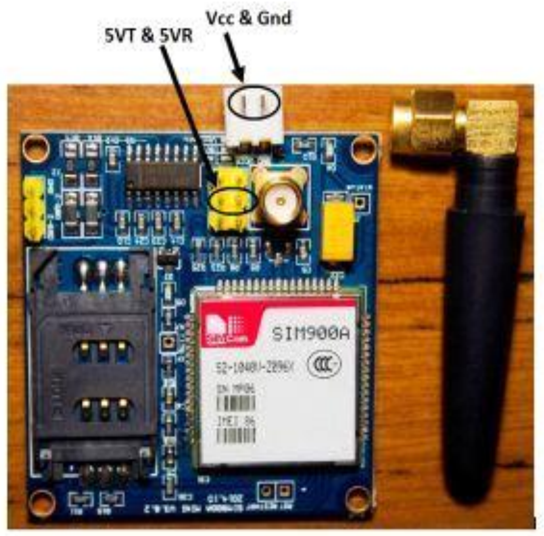


Fig 2.6 GSM Module

2.4.1 FEATURES

- Dual-Band 900/ 1800 MHz
- GPRS multi-slot class 10/8GPRS mobile station class B
- Compliant to GSM phase 2/2+Class 4 (2 W @850/ 900 MHz**)**
- Class 1 (1 W @ 1800/1900MHz)
- Control via AT commands (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands)

- Low power consumption: 1.5mA(sleep mode)’
- Operation temperature: -40°C to +85 °C
- Status indicator (D5): It will flash continuously whenever the call arrives otherwise it is left ON.
- Network LED (D6): This led will blink every second which indicates that the GSM module is not connected to the mobile network. Once the connection is established successfully, the LED will blink continuously every 3 seconds.

2.4.2 BOOTING THE GSM MODULE

- Insert the SIM card to GSM module and lock it.
- Connect the adapter to GSM module and turn it ON!
- Now wait for some time (say 1 minute) and see the blinking rate of ‘status LED’ or ‘network LED’ (GSM module will take some time to establish connection with mobile network)
- Once the connection is established successfully, the status/network LED will blink continuously every 3 seconds. You may try making a call to the mobile number of the sim card inside GSM module. If you hear a ring back, the gsm module has successfully established network connection.

2.5 NODE MCU

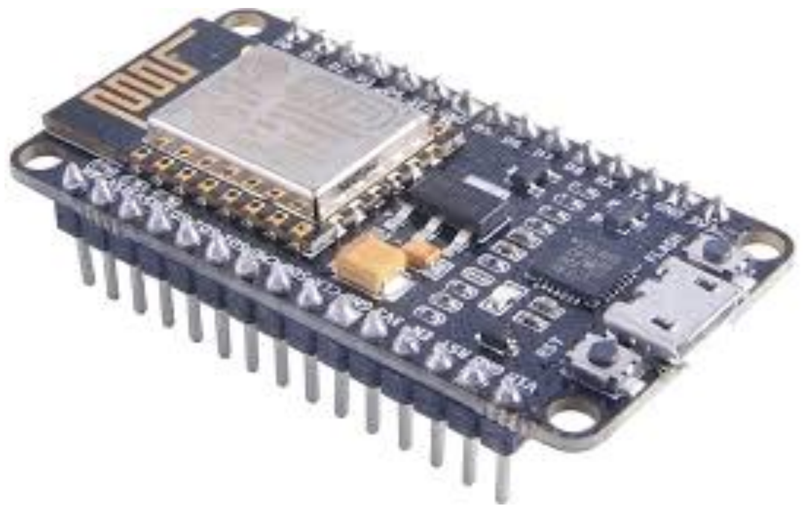


Fig 2.7 NODE MCU

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.To communicate with the ESP8266 module, microcontroller needs to use set of AT commands. Microcontroller communicates with ESP8266-01 module using UART having specified Baud rate.

2.5.1 PIN DIAGRAM

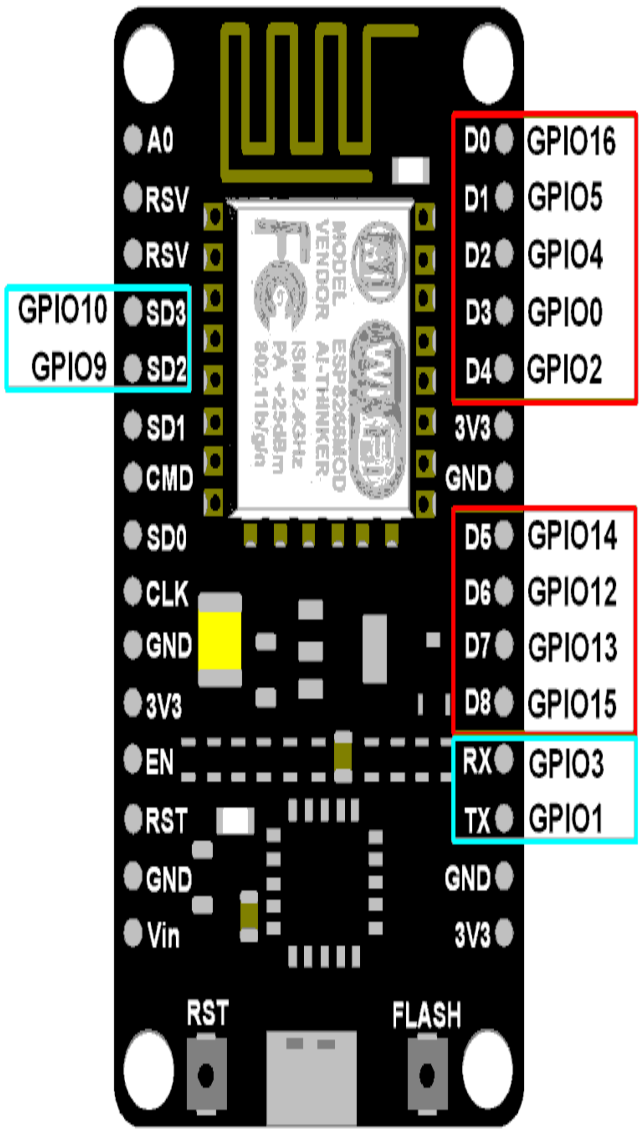


Fig 2.8 Pin diagram of NODE MCU

TABLE 2.1 NODEMCU PINS VS ESP8266 PINS

Pin names on NODEMCU Development kit	**ESP8266 Internal GPIO Pin number**
D0	GPIO16
D1	GPIO5
D2	GPIO4
D3	GPIO0
D4	GPIO2
D5	GPIO14
D6	GPIO12
D7	GPIO13
D8	GPIO15
D9/RX	GPIO3
D10/TX	GPIO1
D11/SD2	GPIO9
D12/SD3	GPIO10

2.6 PROBLEMS WITH AURDINO

In the connected world, we need to connect our devices to the Internet. Let's take a simple example. You need to monitor your home temperature from anywhere in the world. We did a simple temperature monitor project earlier using LM35 and Arduino. Now we need to connect this to the Internet. We can use your Home WiFi network for this. Or else make the connection using Ethernet cable.

In this case you need some additional hardware. To connect Arduino to Ethernet, you need Arduino compatible Ethernet Shield. If WiFi is the option, then you need Arduino compatible WiFi shield.

But let’s forget about all these Shields. Also forget about Arduino as well. Now we have NodeMCU which is an Internet of Things Development board for a cheap price but with more capabilities. NodeMCU is an open source IoT platform. Basically the NodeMCU uses Lua scripting language to program. But don't worry. Your familiar Arduino IDE also can be used to Program NodeMCU. NodeMCU runs an ESP8266 WiFi SoC from Espressif systems. NodeMCU has a built in WiFi module. That means you can easily connect it to WiFi with few lines of codes.

.

CHAPTER-3

SOFTWARE TOOLS

Chapter 3

3.1 IOT PLATFORM

There are ample IoT platform definitions which all point to differently named but similar capabilities and reasons why IoT platforms are important. The variety of types of IoT platforms, as well as their backgrounds/origins and how they work together in ecosystems matter so let’s first add a few more first.

"**ThingSpeak** is an open source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates"

ThingSpeak is an IoT platform, that allows you to connect and save sensor data in the cloud and develop IoT applications. Also, the platform provides apps that let you analyze and visualize data. MATLAB support helps you act on data. Sensor data can be easily integrated and sent from Arduino or Raspberry Pi or any other IoT gateway

3.1.1ThingSpeak Key Features

ThingSpeak allows you to aggregate, visualize and analyze live data streams in the cloud. Some of the key capabilities of ThingSpeak include the ability to:

- Easily configure devices to send data to ThingSpeak using popular IoT protocols.
- Visualize your sensor data in real-time.

.

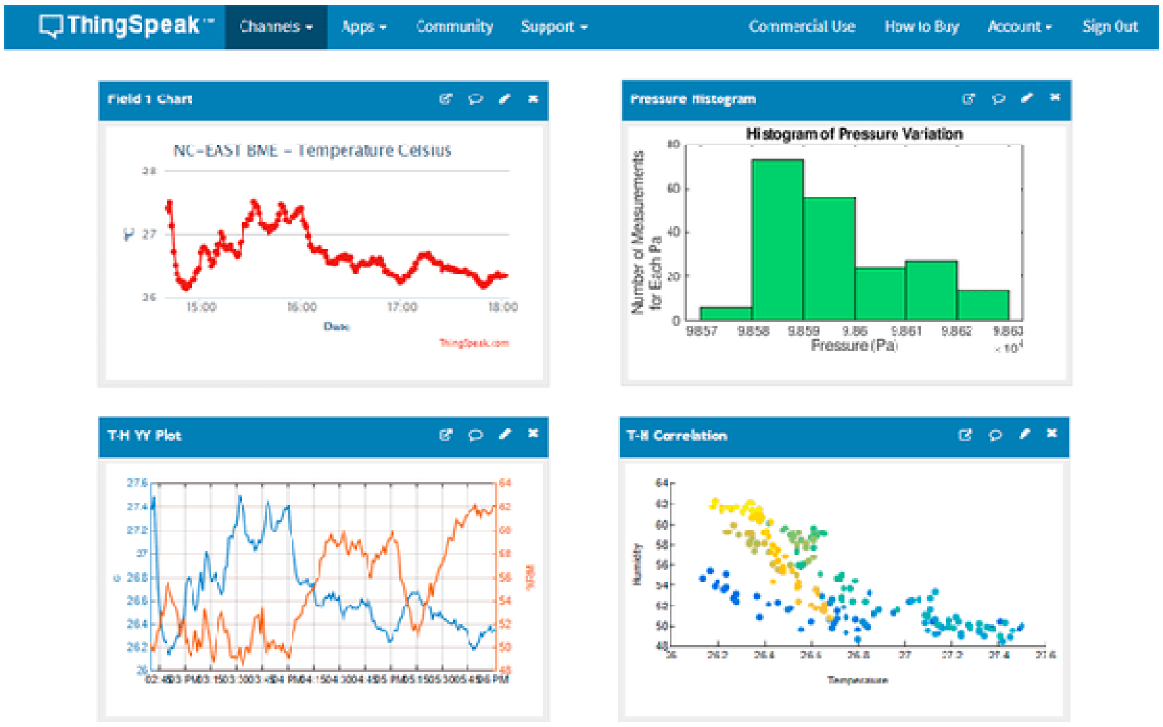


Fig 3.1 Results observation in ThinkSpeak

- Aggregate data on-demand from third-party sources.
- Use the power of MATLAB to make sense of your IoT data.
- Run your IoT analytics automatically based on schedules or events.
- Prototype and build IoT systems without setting up servers or developing web software.
- Automatically act on your data and communicate using third-party services like Twilio® or Twitter®.

3.2 ARDUINO-UNO IDE SOFTWARE :

We need the Arduino IDE to create ,open and change sketches (Arduino calls "sketches". We will use the two words interchangeably in this book). Sketches define what the board will do. You can either use the buttons along the top of the IDE or the menu items.

The parts of ide are :

Compile: Before you program "code" can be sent to the board, it needs to bo converted into instructions that the board understands. This process is called compiling.

Stop: This stops the compilation process

Create new sketch: This opens a new window to create a ncw skctch.

Open existing sketch: This loads a sketch from a file on your computer

Save sketch :This saves the changes to the sketch you are working on.

Upload to board: This compiles and then transmits over the USB cable to board.

Tab button :This lets you create multiple files in tour sketch. This is for more advanced programming then we will do in this class

Sketch editor: This is where you write or edit sketches.

Text console :This shows you what the IDE is curreatdly doing and is also where error messages display if you make a mistake in typing the code (syntax error)

Line number : This shows you what line number your cursor is on. It is useful since the compiler gives error messages with a line number

3.2.1 INTRODUCTION

Processing is an open-source computer programming language and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching non-programmers the fundamentals of computer programming in a visual context. The Processing language builds on the Java language, but uses a simplified syntax and a graphics user interface.

3.2.2 FEATURES

Processing includes a sketchbook, a minimal alternative to an integrated development environment (IDE) for organizing projects. Every Processing sketch is actually a subclass of the Java class (formerly a subclass of Java's built-in Applet) which implements most of the Processing language's feature. When programng in Proccssing, all additional classes defined will be treated as inner classes when the code is translated into pure Java before compiling. This means that the use of static variables and methods in classes is prohibited unless Processing is explicitly told to code in pure Java mode processing also allows for users to create their own classes within the PApplet sketch.This allows for complex data types that can include any number of arguments and avoids the limitations of solely using standard data types such as int (integer) ,char (character), float (real number), and color (RGB, RGBA, hex)

3.2.3 ENVIRONMENT OF PROCESSING

The Processing Development Environment(PDE)

Programs are written in the Text Editor makes it easy to write Processing programs.

Programs are written in the Text Editor and started by pressing the Run button. In processing as computer program is called a sketch. Sketches are stored in the Sketchbook.Which is a folder on your computer.Sketches can draw two- and three-dimensional graphics. The default renderer is for drawing two-dimensional graphics. The P3D renderer makes it possible to draw three-dimensional graphics, which includes

controlling the camera, lighting, and materials. The P2D renderer is a fast, but less accurate renderer for drawing two-dimensional graphics. Both the P2D and
P3D renderers are accelerated if your computer has an OpcnGL compatible graphics card. The capabilities of Processing are extended with Libraries and Tools. Libraries make it possible for sketches to do things beyond the core Processing code. There are hundreds of libraries contributed by the Processing community that can be added to your sketches to enable new things like playing sounds, doing computer vision, and working with advanced 3D geometry. Tools extend the PDE to help make creating sketches easier by providing interfaces for tasks like selecting colors. Processing has different programming modes to make it possible to deploy sketches on different platforms and program in different ways. The Java mode is the default. Other programming modes may be downloaded by selecting "Add Mode”. from the menu in the upper-right corner of the PDE.

3.2.4 PROCESSING DEVOLPMENT ENVIORNMENT (PDE)

The Processing Development Environment (PDE) consists of a simple text editor for writing a code, a message area, a text console, tabs for managing files, a toolbar with buttons for common actions, and a series of menus. The menus options change from mode to mode. The default Java mode is documented here
programs written using Processing are called sketches. These sketches are written in the text editor. It has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by Processing sketches including complete error messages and text output from sketches with the printO and println() functions. (Note that the console works well for occasional messages, but is not intended for higlh-speed, real-time output.) The buttons on the toolbar can run and stop programs.

Run

Runs the sketch. In Java mode, it compiles the code and opens a new display window

Stop

Terminates a running sketch. Addtional commands are found within the six menus: File, Edit, Sketch, Debug. Tools, Help The menus are context sensitive which means only those items relevant to the work currently being carried out are available

File

1)New

a new sketch in a new window, named as the current date is the format Creates "sketch YYMMDD"

2)Open

Open a sketch in a new window

3)Open Recent

Select a sketch to open from the list of recently closed sketches

4)Sketchbook

Open a new window to show the list of sketches in the skctchbook

5)Examples

Open a new window to show the list of the examples.

6)Close

Close the sketch in the frontmost window. If this is the last sketch that’s open, you will be prompted whether you would like to quit. To avoid the prompt, use Quit instead of Close when you want to exit the application.

7)Save

Saves the open sketch in it's current state.

8)Save as

Saves the currently open sketch, with the option of giving it a different name. Doewqs not replace the previous version of the sketch

9)Export

Exports a Java application as an executable file and opens the folder containing the exported files.

10)Page Setup

Define page settings for printing

11)Print (Ctrl+P)

Prints the code inside the text editor.

12)Preferences

change some of the ways Processing works. (This item is located in the Processing menu on Mac OS X.)

13)Quit

Exits the Processing Environment and closes all Processing windows. (This item is located in the Processing menu on Mac OS X.)

Edit

1)Undo

Reverses the last command or the last entry typed. Cancel the Undo command by choosing Edit »Redo.

2)Redo

Reverses the action of the last Undo command. This option is only available if there has already been an Undo action.

3) Cut

Removes and copies selected text to the clipboard (an off-screen text buffer).

4)Copy

Copies selected text to the clipboard.

5)Copy as HTML

Formats code as HTML in the same way it appears in the Processing environment and copies it to the clipboard so it can be pasted somewhere else.

6)Paste

Inserts the contents of the clipboard at the location of the cursor, and replaces any sclected text.

7)Select All

Selects all of the text in the file which is currently open in the text editor.

8)Auto Format

Attempts to format the code into a more human-readable layout. Auto Format was previously called Beautify.

9)Comment/Uncomment

Comments the selected text. If the selected text is already commented, it uncomments it.

10)Increase Indent

Indents the selected text two spaces.

11)Decrease Indent (Ctrl+[)

If the text is indented, removes two spaces from the indent.

12)Find

finds an occurence of a text string within the file open in the text editor and gives the option to replace it with a different text.

13)Find Next

Finds the next occurence of a text string within the file open in the text cditor.

14)Find Previous

Finds the previous occurence of a text string within the file open in the text editor.

15)Use Sclection for Find

Sets the currently selected text as thc itern to find with Find Next and Find Previous.

Sketch

1)Run

Runs the code (compiles the code, opens the display window, and runs the sketoclh inside)

2)Present

Runs the code in the center of the screen with a solid-color background. Click the "stop" button in the lower left to exit the presentation or press the Escape key. Change the background color in the Preferences.

3)Tweak

Runs the code in a way where some color and variable values can be changed while the code is running. The sketch needs to be saved before it can be run as a sketch to Tweak

4)Stop

If the code is running, stops the execution. Programs written without using the draw() function are stopped automatically after they draw.

5) Import Library

Adds the necessary import statements to the top of the current sketch. For example Selecting sketch » Import Library »pdf adds the statement "import processing pdf.:"

to of the file. These import statements are necessary for using Libraries. Select Libraries... to open the Library Manager to browse and install new libraries.

6)Show Sketch Folder

open the folder for the current sketch.

7)Add File

opens a file navigator window. Select an image, font, or other media fles to add it to the sketch's "data" folder.

Debug

1)Enable debugger

Activates the debugger. Note that the Run button will change to Debug. New Continue and Step buttons will appear, along with a separate window for viewing variable values

2)Continue

Advances the code until the next breakpoint

3)Step

Advances the debugger into the interior of a function call. one line at a time. (Note that oncc the code reaches the end of the current function call, the debugger will revert to "continue.")

4)Step Into

Advances the debugger into the interior of a function call. This only works for user-defined functions in the sketch

5)Step Out

Advances the debugger outside of a function to the calling area. This only works for user-defined functions in the sketch

6)Toggle Breakpoint

Add or remove a breakpoint. When a breakpoint is added, the line number is replaced with the symbol:<>.

Tools

1)Create Font..

Converts fonts into the Proccssing font format (VLW) and adds to the current sketch. Upens a dialog box that gives options for setting the font, its size, if it is anti-aliased (smooth), and which characters to be generated. The amount of memory required for the font is determined by the size selected and the number of characters selected through the "Characters..."menu; Processing fonts are textures, so larger fonts require more image data. Fonts can also be created in the code with the createFont() function.

2)Color Selector

Interface for selecting colors. For each color, the HSB, RBG, and Hex values are shown. The Hex value can be copied into the clipboard with the Copy button.

3)Archive Sketch

Archives a copy of the current sketch in zip format. The archive is placed in the same folder as the sketch.

4)Install "processing-java"

Installs the processing-java program to make it possible to build and run Java mode sketches from the command line.

5) Movie Maker

Creates a QuickTime movie from a sequence of images. Options include setting the **size, frame rate, and compression, as well as an audio file.**

6)Add Tool..

Opens the Tool Manager to browse and install new Tools.

Help

1) Environment

Opens the reference for the Processing Development Environment (this page) in the default web browser.

2) Reference

Opens the reference in the default web browser. Includes references for the language, programming environment, and core libraries.

3)Find in Reference

Select an element of the Processing language in the text editor and select Find in Reference to open that page in the default web browser.

4)Libraries Reference

Select from the list to open the reference for compatible Libraries.

5)Tools Reference

Select from the list to open the reference for compatible Tools.

6)Getting Started

Opens the online Getting Started tutorial in the default browser.

7)Troubleshooting

Opens the online Troubleshooting wiki page in the default browser.

8)Frequently Asked Questions

Opens the online FAQ wiki page in the detault browser.

9)The Processing Foundation

Opens the Foundation website in the default browser.

10)Visit Processing.org

Opens Processing website in the default browser.

Chapter 3

Preferences

The processing development Environment (PDE) is highly configurable. The most common

Preferences can be modified in the Preferences window, located in the File menu on Windows and Linux and in the Processingmenu on Mac Os X. The full list of preferences are stored in “preferences.txt" file. This file can be opened and edited directly only when processing is not running.

1)Sketchbook location

Any folder can be used as the Sketchbook. Input a new location or select"Browse' to set the folder you want to use

2)Language

Select the language to use for the menus. Processing needs to be restarted after making a new selection.

3)Editor and Console font

Select a different font t o use for text in the Editor and Console. Note: the selected font should match the language used in the Text Editor. Sce the "Enable complex text input" preference below.

4)Editor font size

Sets the font size of the code in the text editor

5)Console font size

Sets the font size of the text in the console.

6)Background colour when Presenting

Defined the background colour used when a sketch is run with Present.

7)Use smooth text in editor window

By default, the text in the editor is aliased. When checked, the editor switches to an anti-aliased (smoothed) font. Restart Processing after making this change.

8)Enable complex text input

Enables the Text Editor to display non-Latin fonts such as Japanese. Processing needs to be restarted after making this selection.

9)Continuously check for errors and Show warnings

Turn on and off the features that continuously check for and report potential code errors.

10)Code completion with Ctrl-space

Turn on and off code completion Press Ctrl-space to activate code completion while typing.

11)Suggest import statements

When checked, Proccssing will try to suggest libraries to import when code from that library is detected.

12) Increase maximum available memory

Allocates more RAM to Processing sketches when they rm. Sketches that use media files (images, audio, etc.) sometimes require more RAM. Increase the amount of RAM if a sketch is throwing Out of Memory Errors.

13)Delete previous folder on export

When checked (default behavior), Processing deletes the complete export folder before re-creating it and adding the new media.

14)Check for updates on startup

When checked (default behavior), youll be informed of new Processing software releases as they become available through a small dialog box that opens as Processing starts.

15)Run sketches on display

If more than one monitor is attached, seleet the monitor on which to display the sketch.

3.2.5 SKETCHES AND SKETCHBOOK

All Processing projects are called sketches. Each sketch bas its own folder. The main file for each sketch has the same name as the folder and is found inside. For example, if the sketch is named "Sketch 123", the folder for the sketch will be called Sketch 123" and the main file will be called "Sketch_123.pde". The PDE file extension is an acronym for the Processing Development Environment.

Processing sketches can be stored anywhere on your computer, but by default they are stored in the sketchbook, which will be in different places on your computer or network,depending if you usc PC, Mac, or Linux and how the preferences are set. To locate this folder, select the"Preferences" option from the File menu (or from the "Processing" menu on the Mac) and look for the "Sketchbook location."

A sketch folder sometimes contains other folders for media files and other code. When a font or image is added to a sketch by selecting "Add File..." from the Sketch menu, a "data" folder is created. Files may also be added to your Processing sketch by dragging them into the text editor. Image and sound files dragged into the application window will automatically be added to the current sketch's "data" folder. All images, fonts, sounds, and other data files loaded in the sketch must be in this folder.

Using the 3-dimension coordinate system of P3D, the z- coordinate is zero at the surface of the image, With the negative z-values moving back in space. When drawing in 3D, the camera is positioned in the center of the screen.

3.2.6. TABS, MULTIPLE FILES AND CLASSES

It can be inconvenient to write a long program within a single file. When Processing sketches to hundreds or thousands of lines, breaking them into modular units helps manage the different parts. Processing manages files with the Sketchbook and each sketch can have multiple files that are managed with tabs.

The arrow button to the right of the tabs in the Processing Development Environment

manage these files. Click this button to reveal options to create a new
tab, and delete the current tab. Tabs are intended for more advanced users, and for this reason, the menu that controls the tabs is intentionally made less prominent.

Debug

The Processing Debugger is a tool for diagnosing problems with a sketch. Enable it to

pause

a sketch while running and advance through the code one line at a time. The debugger is enabled through the File menu (Debug> Enable Debugger) or by clicking the Debugger icon,the butterfly in the upper-right coner of the PDE. When the Debugger is enabled, the program runs as normal, but stops at create a breakpoint, set the cursor at the line you want to pause the sketch Toggle Breakpoint. Thec keyboard shortcut is Command-B. To remove the breakpoint, select Toggle Breakpoint again. When a breakpoint is added, the line number is replaced with the symbol:<> Running the sketch in Debug mode causes the sketch to pause at any breakpoints. When paused, current variable values are visible in a separate panel. You can advance to the nextbreakpoint by selecting "Continue" or advance line by line through the code with Stepping only works within the scope of the current function being run.

3.2.7 PROGRMMING MODES

Processing has different programming modes to make it possible to deploy different platforms and program in different ways. The current default is Java mode. Other programming modes such as Android Mode and Python are added by selecting "Add Mode..." from the meau in the upper-right comer of the PDE.

3.2.8 Adding Libraries, Tools, and Modes

Processing 3.0 includes a set of features to make it easier to install , update , and remove libraries , tools , modes , and examples .Add a contributed library by selecting "Add Library..." from the "Import Library.." submenu within the Sketch menu. Tlhis opens the Library Manager. Next, select alibrary and then elick on Install to download it. Add a contributed tool by selecting "Add Tool." from the Tools menu, then select a Tool to download from the Tool Manager.Add contributed modes by selecting "Add Mode..." from the Mode menu in the upper-right cormer of the PDE, then select a Mode to install. Add contributed Examples by first opening the "Examples.." submenu from the File menu. Cick on the Add Examples button to open the Examples Manager. Next, select an examples package and select Install to download.

CHAPTER-4

IMPLEMENTATION

Chapter 4

4.1 INTERFACING GSM MODULE TO ARDUINO

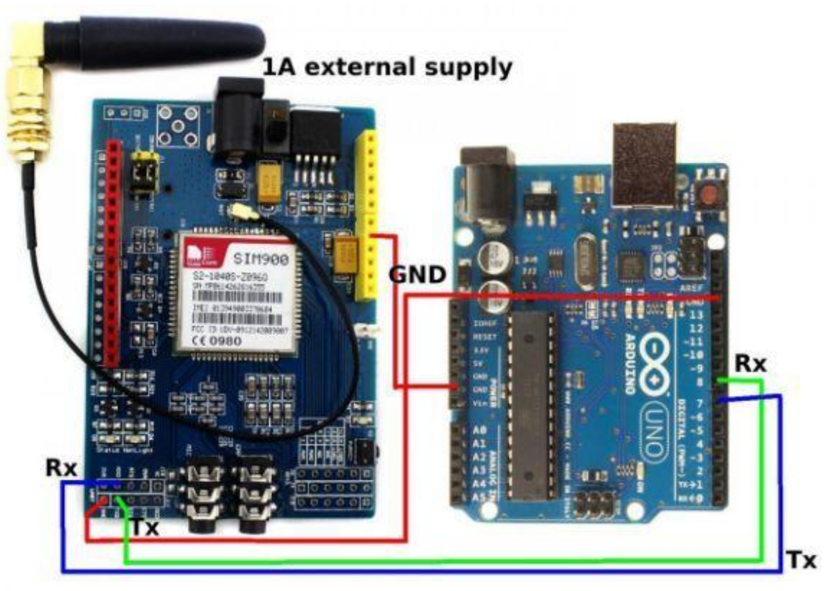


Fig: 4.1 INTERFACING GSM MODULE TO ARDUINO

There are two ways of connecting GSM module to Arduino. In any case, the communication between Arduino and GSM module is serial. So we are supposed to use serial pins of Arduino (Rx and Tx). So if you are going with this method, you may connect the Tx pin of GSM module to Rx pin of Arduino and Rx pin of GSM module to Tx pin of Arduino. You read it right? GSM Tx –> Arduino Rx and GSM Rx –> Arduino Tx. Now connect the ground pin of Arduino to ground pin of gsm module! So that's all! You made 3 connections and the wiring is over! Now you can load different programs to communicate with gsm module and make it work.

4.2 INTERFACING RTC MODULE TO ARDUINO

There are only 5 pins: **5V GND SCL SDA SQW**.

- **5V** is used to power to the RTC chip when you want to query it for the time. If there is no 5V signal, the chip goes to sleep using the coin cell for backup.
- Connect **GND** to common power/data ground
- Connect the **SCL** pin to the I2C clock **SCL** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A5**, on a Mega it is also known as **digital 21** and on a Leonardo/Micro, **digital 3**
- Connect the **SDA** pin to the I2C data **SDA** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A4**, on a Mega it is also known as **digital 20** and on a Leonardo/Micro, **digital 2**

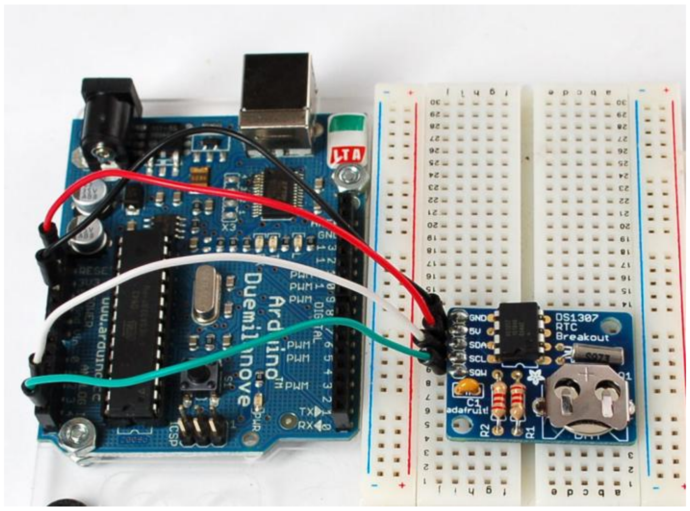


Fig 4.2 Interfacing RTC Module to Aurduino

4.3 INTERFACING LED'S AND BUZZER

In every box we placed an led for indication . We have used three medicine boxes so led in 1st box is connected to 8th pin in arduino, led in 2nd box is connected to 9th pin in arduino and led in 3rd box is connected to 8th pin in arduino. A common buzzer is connected to 7th pin in Arduino.Another led is connected to D0 pin in nodemcu to indicate whether the heart rate value is read or not.

4.4 INTERFACING MAGNETIC REED SWITCHES TO ARDUINO

In every box there is a magnetic reed switch at the lid side.one end is connected to supply and other end to arduino pin's (11,12,13).when the lid is closed it read's 1 and when it is open it read's 0.Based on that information,we can conclude that the lid is open or not.

4.5 INTERFACING HEART RATE MODULE TO NODEMCU

Heart rate module has three pins supply,ground and data pin.data pin is connected to A0 pin of nodemcu ,vcc of module to vcc of nodemcu and ground of module to ground of nodemcu.

4.6 FINAL IMPLEMENTATION

We are going to place leds in each section of tablet box and we set the time of each tablet section .

When the given time and the RTC time matches buzzer along with led will be on .

After opening the correct tablet section it can automatically switch off the buzzer and led using Magnetic Reed Switch.

If the Medicine is not taken at the respective time ,buzzer and led will be turned off after particular time sending notification to their Pre-defined Guardian.

In e-health monitoring system we are connecting different health sensors to the Arduino and read values from those sensors can be sent toIot platform using Esp8266 wifi module.

CHAPTER-5

RESULTS AND DISCUSSIONS

- Fig 5.1 shows the setup of medicine reminder kit.
- Fig 5.2 indicates that you have to take the medicine in the middle box(led glowing one) as the time set for that tablet was occurred.
- Fig 5.3 shows that the notication was sent to the mobile if the tablet was not taken.
- Fig 5.4 shows the Thingspeak iot platform in which heart rate data was uploaded.
- Fig 5.5 and Fig 5.6 shows the nodemcu web browser and the heart rate data uploaded in that browser.

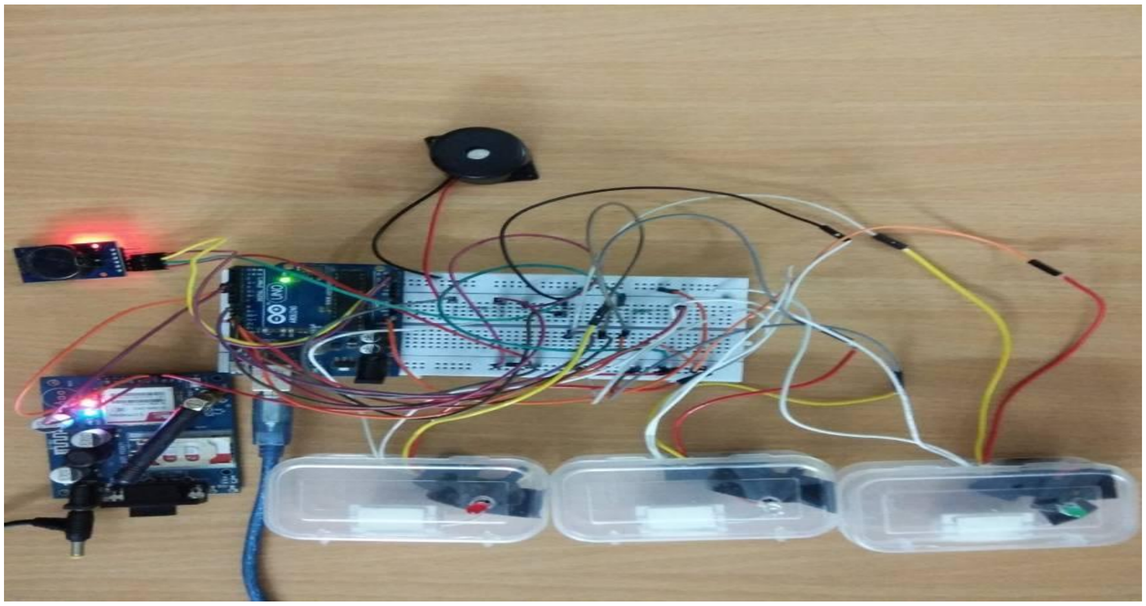


Fig 5.1 : Result of RTC module

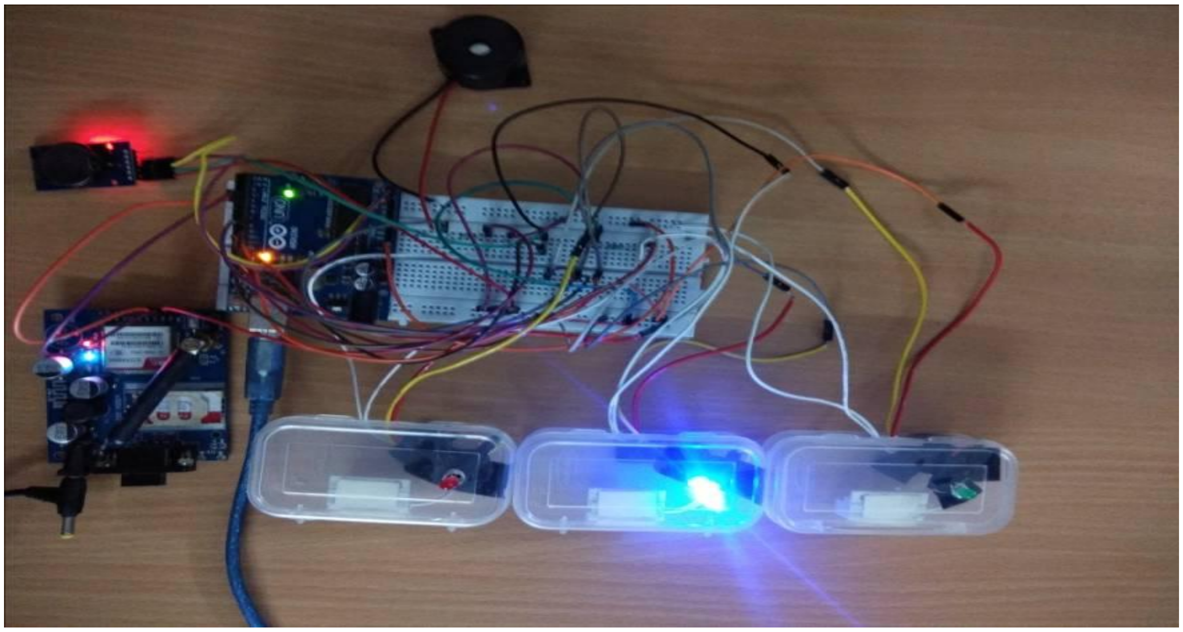


Fig 5.2 : Result of medicine remainder kit

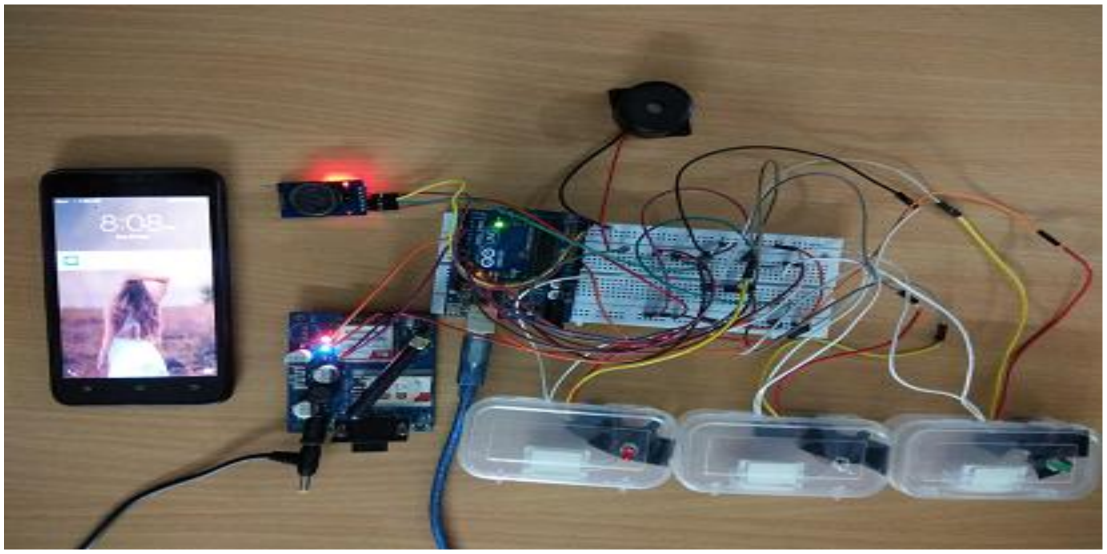
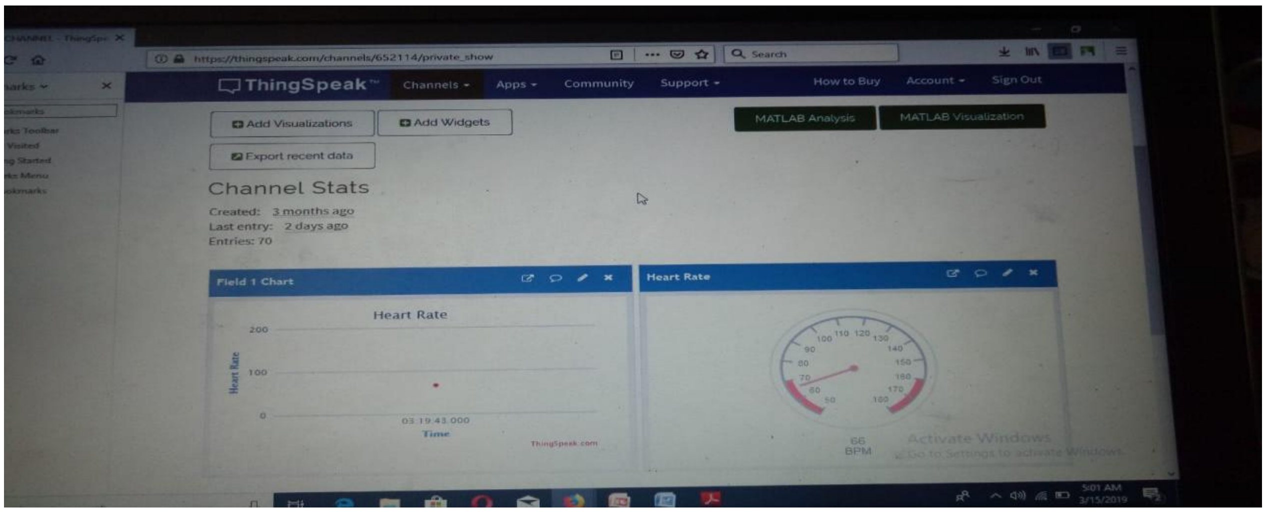
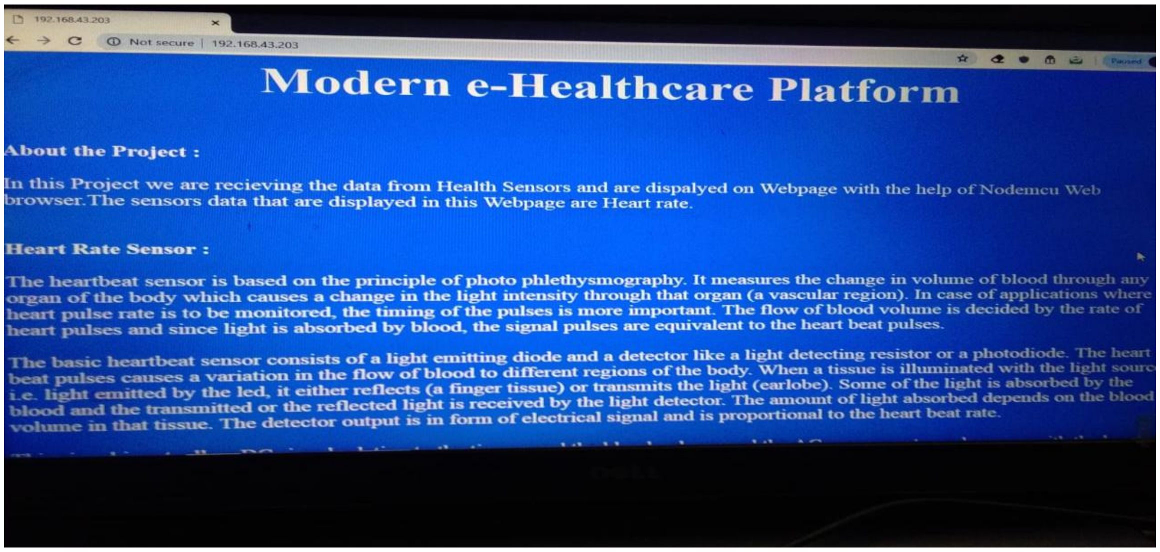


Fig 5.3 : Result – message to guardians phone if the medicine is not taken



Channel Stats
Created: 3 months ago
Last entry: 2 days ago
Entries: 70

Fig 5.4 : Values from health monitoring sensors



192.168.43.203

Not secure | 192.168.43.203

Modern e-Healthcare Platform

About the Project :

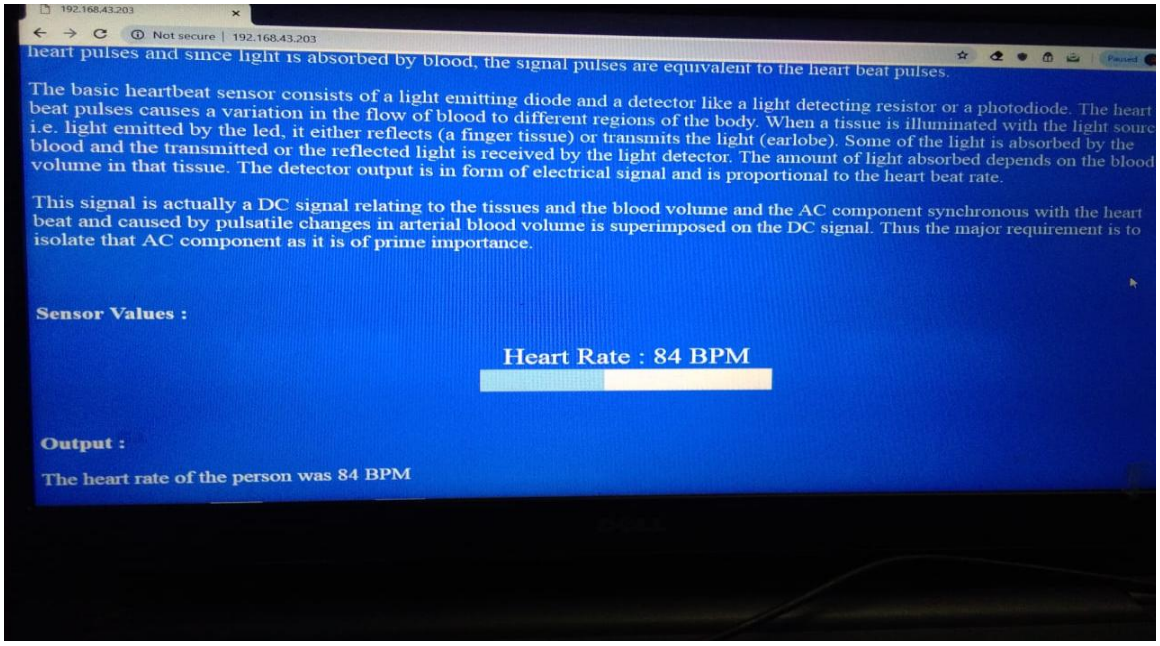
In this Project we are recieving the data from Health Sensors and are dispalyed on Webpage with the help of Nodemcu Web browser.The sensors data that are displayed in this Webpage are Heart rate.

Heart Rate Sensor :

The heartbeat sensor is based on the principle of photo phlethysmography. It measures the change in volume of blood through any organ of the body which causes a change in the light intensity through that organ (a vascular region). In case of applications where heart pulse rate is to be monitored, the timing of the pulses is more important. The flow of blood volume is decided by the rate of heart pulses and since light is absorbed by blood, the signal pulses are equivalent to the heart beat pulses.

The basic heartbeat sensor consists of a light emitting diode and a detector like a light detecting resistor or a photodiode. The heart beat pulses causes a variation in the flow of blood to different regions of the body. When a tissue is illuminated with the light sourc i.e. light emitted by the led, it either reflects (a finger tissue) or transmits the light (earlobe). Some of the light is absorbed by the blood and the transmitted or the reflected light is received by the light detector. The amount of light absorbed depends on the blood volume in that tissue. The detector output is in form of electrical signal and is proportional to the heart beat rate.

Fig 5.5 : Nodemcu webserver



192.168.43.203

Not secure | 192.168.43.203

heart pulses and since light is absorbed by blood, the signal pulses are equivalent to the heart beat pulses.

The basic heartbeat sensor consists of a light emitting diode and a detector like a light detecting resistor or a photodiode. The heart beat pulses causes a variation in the flow of blood to different regions of the body. When a tissue is illuminated with the light sourc i.e. light emitted by the led, it either reflects (a finger tissue) or transmits the light (earlobe). Some of the light is absorbed by the blood and the transmitted or the reflected light is received by the light detector. The amount of light absorbed depends on the blood volume in that tissue. The detector output is in form of electrical signal and is proportional to the heart beat rate.

This signal is actually a DC signal relating to the tissues and the blood volume and the AC component synchronous with the heart beat and caused by pulsatile changes in arterial blood volume is superimposed on the DC signal. Thus the major requirement is to isolate that AC component as it is of prime importance.

Sensor Values :

Heart Rate : 84 BPM

Output :

The heart rate of the person was 84 BPM

Fig 5.6 : Data Uploaded to nodemcu server

CHAPTER-6

CONCLUSIONS

Chapter 6

Hence an attempt was made to build a medicine reminder kit which gives indication at proper time and also an alert to pre-gaurdian about tablet consumption status.In addition to it we also have heart rate monitor which calculates our heart rate and upload it to online website.

REFERENCES

>“Wireless health monitoring system for patients”by Salman Ahmed ,Sabrin Milat ,Aymanur Rehman on 2015 IEEE international WIE Conference on Electrical and Computer Enginnering .

>”Intelligent Medicine Box For Medication Management Using IOT” by M.Srinivas, P.Durga Prasad, V. Naga Prudvi Raj on 2018 IEEE second international coference on inventive systems and control