North South University

Department of ECE

CSE-299 Final Project Report

Name	Id
Imam Hossain Tanvir	172 2361 642
Nazmul Haque	171 2766 642
Md. Fahimuzzaman Shaurav	171 1764 642
Md. Zahidul Islam	171 1548 642

Project Title: Automatic Medicine Reminder Using Arduino.

Course Title: Junior Project Design

Course Code: CSE-299

Section: 05

Project Group: 03

Submission Date: 14-01-2021

INTRODUCTION

When it comes to our loved ones, we always want to stay them healthy and fit. But what will happen if they get ill and forget to take medicine on time. We would be worried, right? At hospitals, there are many patients and it is difficult to remind every patient to take medicine on time. The traditional ways require human efforts to remind them to take medicines on time. The digital era doesn’t follow that and we can use machines to do that. The application of Smart Medicine Reminder is very wide and can be used by patients at home, doctors at hospitals, and at many other places. When it comes to reminding, there can be many ways to remind it.

For example: At home, it is very much easy to forget to remind about the medicine to a patient. But using our ‘Automatic Medicine Reminder’, it will become very hard to forget. This machine will remind the patient about his/her medicine taking time and date correctly. This system can be also used in the hospital when doctors are not in front of patient and patient is unable to take medicine by himself/herself. Doctors will get notification in their mobile phone. And they will know about the timing of the medicine. Thus the timing of taking medicine will not be delayed.

Description of 3 closely related works

1. Burglar Alarm

For Burglar alarm, it is important to use sensor that sense presence or something as input device and it is also important to use an output device as a alarm (buzzer). First the sensor is needed to be chosen. Here Hall effect sensor has been chosen because it's very simple and easy to use, it's working based on the proximity of magnet, so that the magnetic field can be measured based on the output voltage and if the sensor and magnet can be fixed between our door and we can program it to alarm when the magnetic field is low, and it's very low power and smaller footprint.

2. Fire Alert

Fire Detectors play a very important role in Industries, Shops, Malls, Residential complexes, parking areas, etc. They help in detecting fire or smoke at an early stage and can help in saving lives. Commercial Fire detecting systems usually have an alarm signaling, with the help of a buzzer or Siren. This project is designed as an IOT based Fire Alerting System using Temperature and a smoke sensor which would not only signal the presence of fire in a particular premise but will also send related information through IOT. Internet of Things (IoT) is basically the network of ‘things’ by which physical things can exchange data with the help of sensors, electronics, software, and connectivity. These systems do not require any human interaction. In this Adriano fire alarm system using temperature and smoke sensor using the IOT project, we can send LIVE information like Temperature, Smoke Value detected by a particular device to the Fire Department.

3. Arduino Based Vehicle Accident Alert System.

Here in this project, we are going to build a Arduino based vehicle accident alert system using GPS, GSM and accelerometer. Accelerometer detects the sudden change in the axes of vehicle and GSM module sends the alert message on your Mobile Phone with the location of the accident. Location of accident is sent in the form of Google Map link, derived from the latitude and longitude from GPS module. The Message also contains the speed of vehicle in knots. This Vehicle Accident alert project can also be used as a Tracking System and much more, by just making few changes in hardware and software. Main Components in this project are Arduino, GPS receiver, GSM module, Accelerometer.

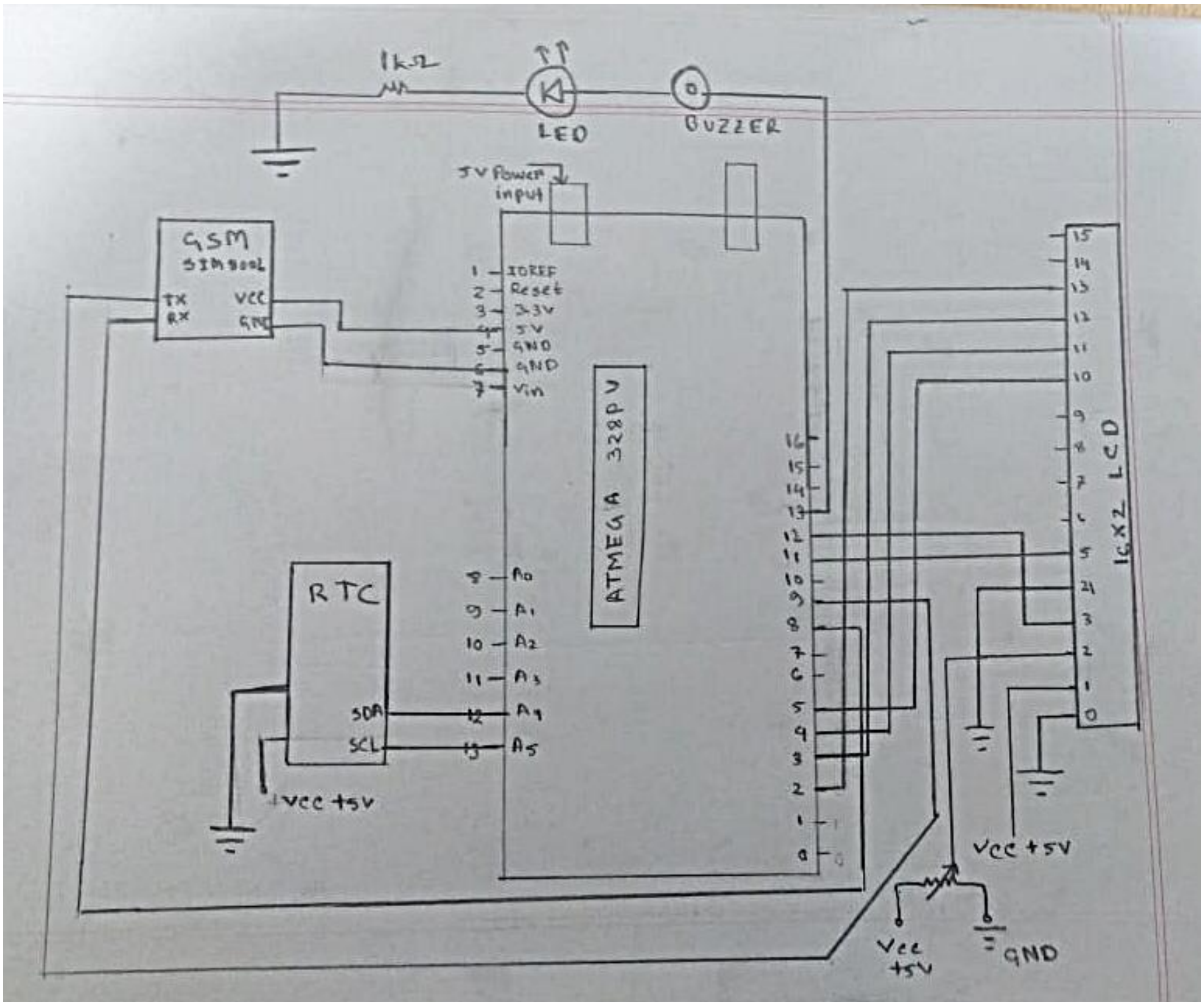
In this project, Arduino is used for controlling whole the process with a GPS Receiver and GSM module. GPS Receiver is used for detecting coordinates of the vehicle, GSM module is used for sending the alert SMS with the coordinates and the link to Google Map. Accelerometer namely ADXL335 is used for detecting accident or sudden change in any axis. And an optional 16x2 LCD is also used for displaying status messages or coordinates.

When we are ready with our hardware after programming, we can install it in our vehicle and power it up. Now whenever there is an accident, the car gets tilt and accelerometer changes his axis values. These values read by Arduino and checks if any change occurs in any axis. If any change occurs then Arduino reads coordinates by extracting $GPGGA String from GPS module data and send SMS to the predefined number to the police or ambulance or family member with the location coordinates of accident place. The message also contains a Google Map link to the accident location, so that location can be easily tracked. When we receive the message then we only need to click the link and we will redirect to the Google map and then we can see the exact location of the vehicle. Speed of Vehicle, in knots (1.852 KPH), is also sent in the SMS and displayed on the LCD panel.

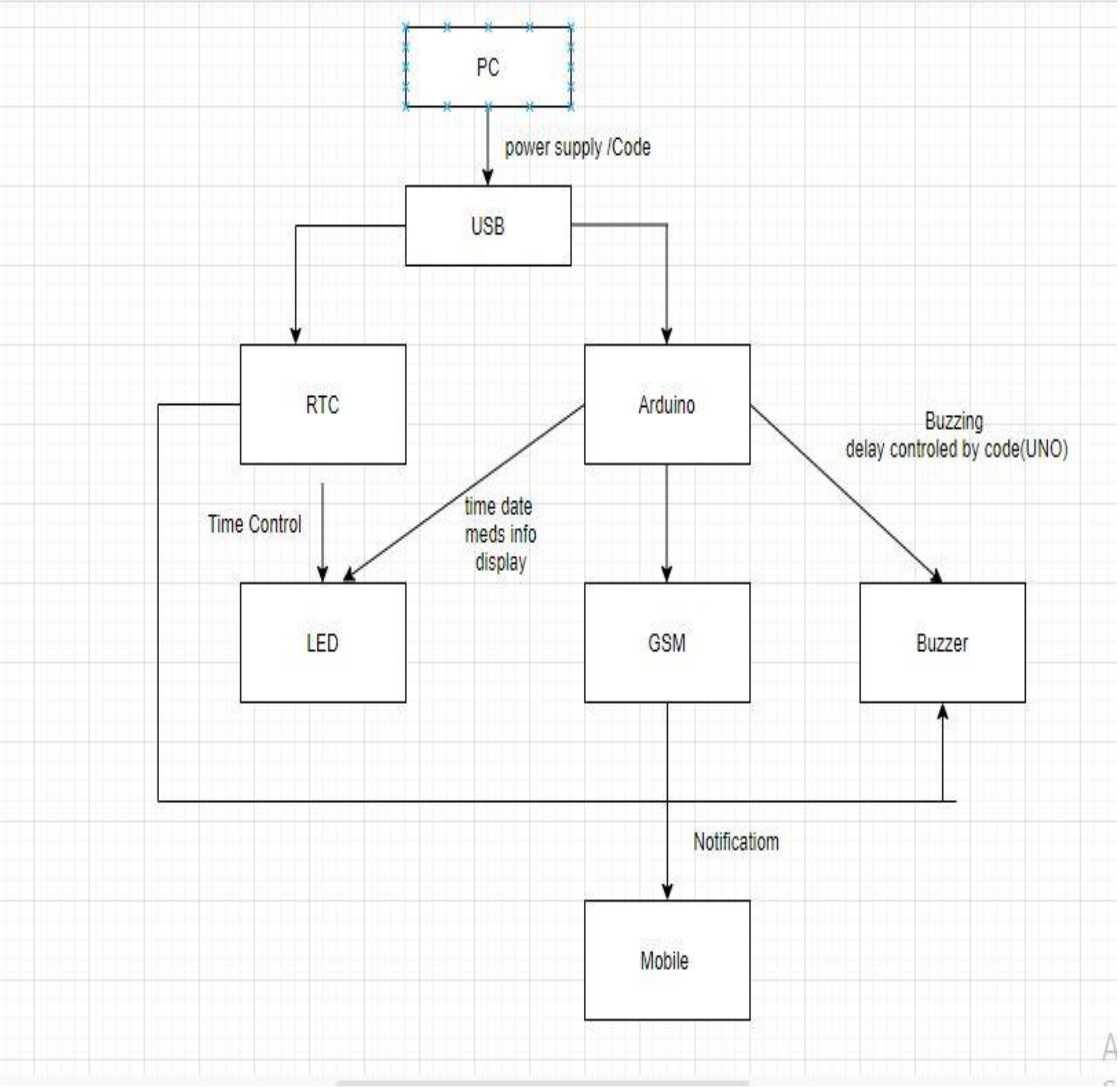
Solution Adopted

Explanation of the functioning of the complete system, and all subsystems

In this project we are using an Arduino uno r3 board which is operated by ATMEGA328 microcontroller. The analog pin 4 is connected with the SDA pin of RTC module and analog pin 5 is connected with the SCL pin of RTC module. The digital pin 8 and 9 are connected with the TX and RX pin of GSM. The data bus D4-D7 is connected with digital pin 2-5 of Arduino. Digital Pin no 11 is enable and pin 12 is reset. Digital pin 13 is connected with buzzer and led. The VCC and GND is coming from Arduino. The 15 and 16 no pin of LCD display is common VCC and GND which is used to light up the backlight of LCD panel. We are using a potentiometer to control the opacity of the LCD panel.



Schematic diagram of Automatic medicine reminder



Flowchart of automatic medicine reminder

Arduino Uno

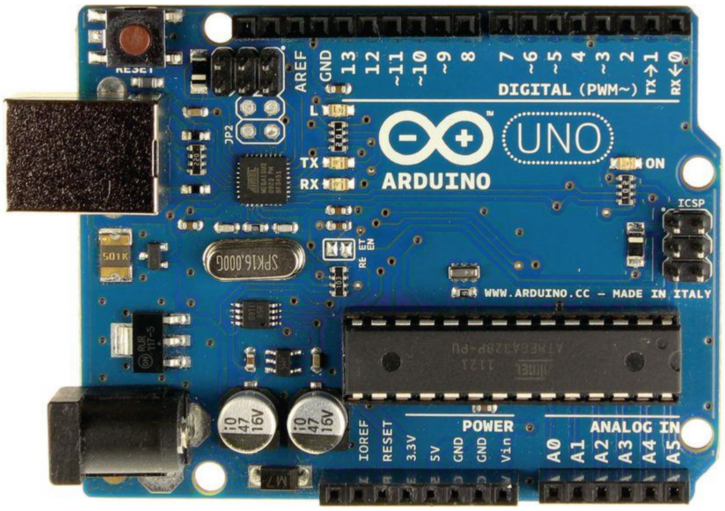
The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

To write program in Arduino we use Arduino IDE

In this project we use Arduino to make the circuit programmable.



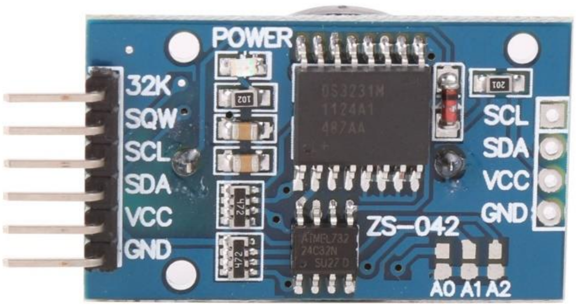
RTC DS3231 module:

The DS3231 is a low-cost, highly accurate Real Time Clock which can maintain hours, minutes and seconds, as well as, day, month and year information. Also, it has automatic compensation for leap-years and for months with fewer than 31 days.

The module can work on either 3.3 or 5 V which makes it suitable for many development platforms or microcontrollers. The battery input is 3V and a typical CR2032 3V battery can power the module and maintain the information for more than a year.

The module uses the I2C Communication Protocol which makes the connection to the Arduino Board very easy.

The first question that comes here is why we actually need a separate RTC for our Arduino Project when the Arduino itself has built-in timekeeper. Well the point is that the RTC module runs on a battery and can keep track of the time even if we reprogram the microcontroller or disconnect the main power.



10K Potentiometer:

A potentiometer is defined as a 3 terminal variable resistor in which the resistance is manually varied to control the flow of electric current. A potentiometer acts as an adjustable voltage divider.

In our project we will use it to maintain the opacity of our LCD panel and the sound of the buzzer.

1k and 10k resistor:

A resistor is an electric component which gives resistance to current flow in a circuit.

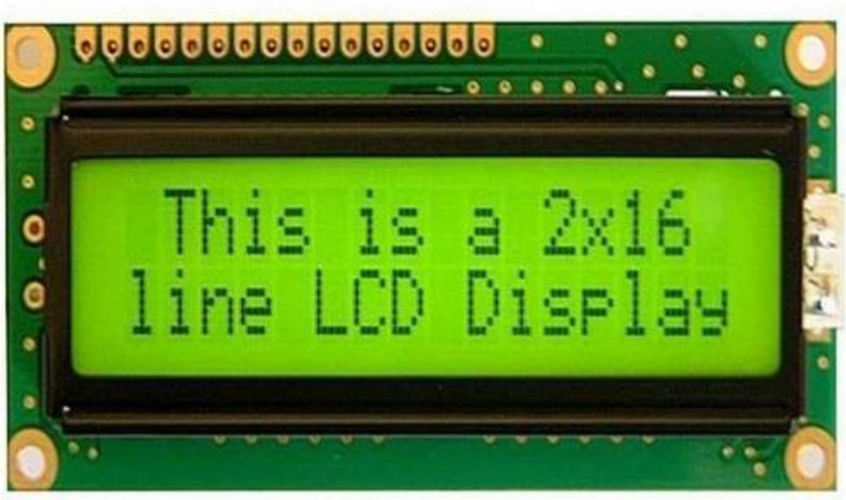
Here we are using 10k resistor with push button for turning off the transient state of push buttons and 1k resistor with LCD because of safety issue so that excessive voltage cannot flow with in the LCD panel.

Jumper wires:

Jumper wires are simple wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.

16*2 LCD Display:

The part 16×2 means that the LCD has 2 lines, and can display 16 characters per line. Therefore, a 16×2 LCD screen can display up to 32 characters at once. It is possible to display more than 32 characters with scrolling though. *Uses of 16*2 Lcd display: In our project Lcd monitor is using for displaying the name of user when the alarming time started. The details of user(medicine name, quantity) displaying in lcd monitor. When lcd monitor shows user details then buzzer and led light is open for alarming the user.



Buzzer:

A buzzer is basically a tiny speaker that you can connect directly to an Arduino. "Piezoelectricity" is an effect where certain crystals will change shape when you apply electricity to them. By applying an electric signal at the right frequency, the crystal can make sound. *Uses of buzzer: It's simple, tone(buzzer, 1000) sends a 1KHz sound signal to pin 9, delay(1000) pause the program for one second and Not One(buzzer) stops the signal sound. The loop() routine will make this run again and again making a short beeping sound. In my project, buzzer for beeping when the timer started. When the alarming started the buzzer switched open.

Led(Any color):

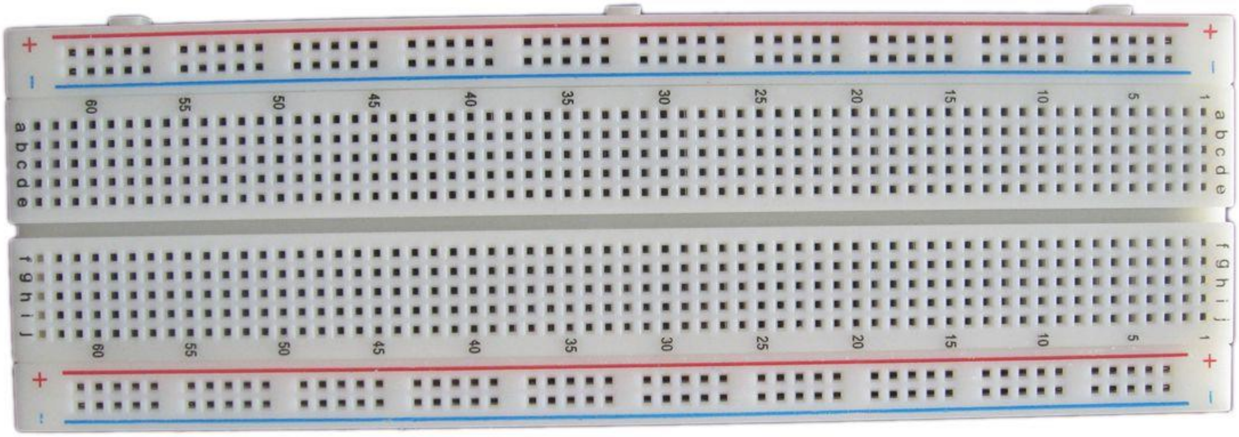
To turn on an LED, the Arduino needs to send a HIGH signal to one of it's pins. To turn off the LED, it needs to send a LOW signal to the pin. It can make the LED flash by changing the length of the HIGH and LOW states. The Arduino has an on-board surface mount LED that's hard wired to digital pin 13.

In our project led is mainly use for alarming time. When the alarm buzzer is started, the led light is on. Led light is on for security and alarming purpose.

Breadboard:

Breadboard is a way of constructing electronics without having to use a soldering iron. Components are pushed into the sockets on the breadboard and then extra 'jumper' wires are used to make connections.

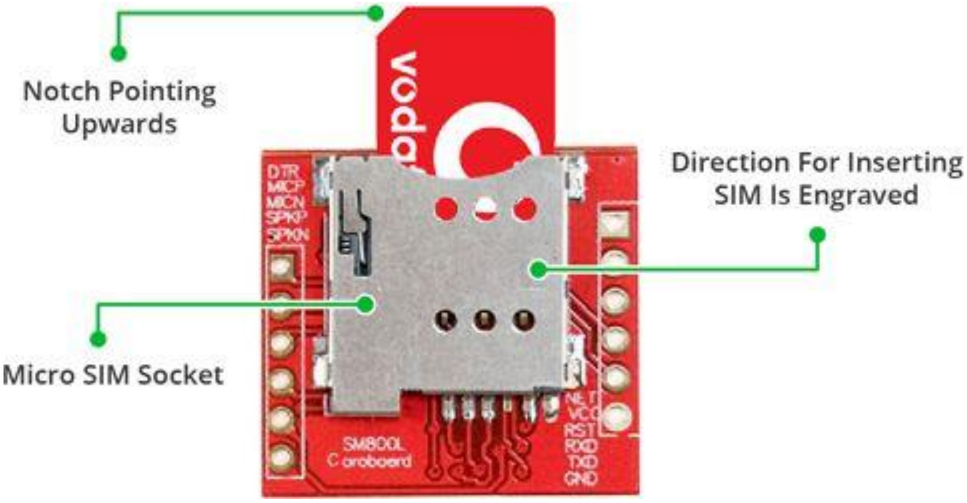
Uses of bread board: The breadboard has strips of metal underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically. All equipments are connected in the breadboard. Buzzer, push buttons, potentiometer, registers, led display, led light all are connected in breadboard.

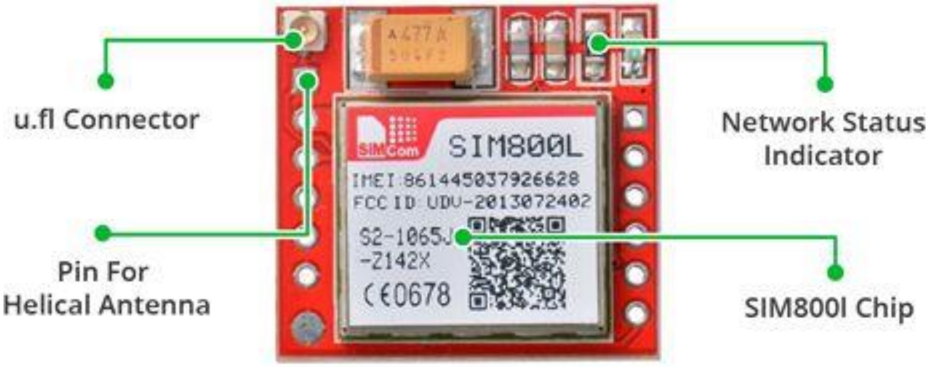


GSM Module (Sim 800L):

Whether you want to listen to what happens in your house that’s miles away from you or activate sprinkler system in your garden just with a silent call; Then SIM800L GSM/GPRS module serves as a solid launching point for you to get you started with IoT!

SIM800L GSM/GPRS module is a miniature GSM modem, which can be integrated into a great number of IoT projects. You can use this module to accomplish almost anything a normal cell phone can; SMS text messages, Make or receive phone calls, connecting to internet through GPRS, TCP/IP, and more! To top it off, the module supports quad-band GSM/GPRS network, meaning it works pretty much anywhere in the world.





Challenges and Difficulties

The main challenge for this project was to configure the LCD and GSM module. We are using a sim800l model GSM, but for catching the 4g network the capacitor has to be very much strong so instead of using the integrated capacitor of sim800l you should use 1k micro farad capacitor. You can easily remove the integrated capacitor with a heating tool and replace the capacitor with a 1k micro farad capacitor. Otherwise it would be very difficult to catch the signal.

Suggestions for improvements

Our Automatic Medicine Reminder is capable of doing some valuable things. But like other machines, we can improve its quality so that people can get benefits from it. After doing some research we have come to an end that we can bring THREE improvements of this project in future. They are:

1. Mobile Apps: Our project highly depends on its developers only. It means only the developers of this projects can change the medicine reminding time and date. If we can make an app, then it will be too easy for the users to upgrade the time and date according to their need.

2. Calling System: In our project, users get their medicine reminding time and date in their email/phone through messages. We can also remind users by calling them through phone. When we will make an app, there we can provide users two options (1) Remind Through Message (2) Remind Through Call. By clicking option number 2, users can get their medicine reminding time and date through call. This call will go to the user as a notification

Appendix

CODE:

```
#include <RTClib.h>


#include <Wire.h>
```

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
//#include "RTClib.h"

SoftwareSerial mySerial(9, 8); //SIM800L Tx & Rx is connected to Arduino #9 & #8

DateTime now;
char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};

RTC_DS3231 rtc;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // (rs, e, d4, d5, d6, d7)

void showDate(void);
void showTime(void);
void showDay(void);

int hr=11;
int mnt=49;
int sec=0;
int buzzer=13;
int state=0;

void setup ()
{
  Serial.begin(9600);
   mySerial.begin(9600);
    Serial.println("Initializing...");
  delay(1000);



  lcd.begin(16,2);

 Serial.println("Hello !!");
lcd.print("Hello !!");
delay(3000);
```

```
lcd.clear();
delay(500);
 Serial.println("Name of the medicine : ' Napa '");
lcd.print("Name of the ");
lcd.setCursor(0,1);
lcd.print("medicine : NAPA ");
delay(5000);
lcd.clear();
 Serial.print (" take medicine at : ");
 Serial.print( hr );
 Serial.print( ":" );
 Serial.print( mnt );
// Serial.print( ":" );
 //Serial.print( sec);
lcd.setCursor(0,0);
lcd.print("take medicine at:");
lcd.setCursor(0,1);

 lcd.setCursor( 4, 1);
 lcd.print( hr );
 lcd.print( ":" );
 lcd.print( mnt );
 //lcd.print( ":");
// lcd.print( sec );

delay(8000);
lcd.clear();

pinMode(buzzer, OUTPUT);

  if (! rtc.begin())
  {
    Serial.println("Couldn't find RTC Module");
    while (1);
  }

  if (rtc.lostPower())
  {
    Serial.println("RTC lost power, lets set the time!");
    rtc.adjust(DateTime(F(_DATE_), F(_TIME_)));
  }
  rtc.adjust(DateTime(F(_DATE_), F(_TIME_)));
```

```
}


void loop ()
{
  now = rtc.now();
  showDate();
  showDay();
  showTime();
  if( hr == now.hour() && mnt == now.minute() && now.second() <= 1
)
 {
  digitalWrite( buzzer , HIGH );
   Serial.println(" ");
  Serial.println("Take medicine : NAPA ");
  lcd.clear();
  lcd.print(" Take medicine:");
  lcd.setCursor(6,1);
  lcd.print(" NAPA ");


  mySerial.println("AT"); //Once the handshake test is successful, it will
back to OK
```

```
  updateSerial();
 mySerial.println("AT+CMGF=1"); // Configuring TEXT mode
  updateSerial();
 mySerial.println("AT+CMGS=\"+8801671495822\"");//change ZZ with country code and xxxxxxxxxxx with phone number to sms)
  updateSerial();
  mySerial.print(" Take medicine :  NAPA  "); //text content
  updateSerial();
  mySerial.write(26);



  delay (7000);
  lcd.clear();
 }
 else
 digitalWrite(buzzer , LOW );
}



 void showDate()
 {
```

```
 lcd.setCursor(0,0);
 lcd.print(now.day());
 lcd.print('/');
 lcd.print(now.month());
 lcd.print('/');
 lcd.print(now.year());
}
void showDay()
{
 lcd.setCursor(11,0);
 lcd.print(daysOfTheWeek[now.dayOfTheWeek()]);
}
void showTime()
{
 lcd.setCursor(0,1);
 lcd.print("Time:");
 lcd.print(now.hour());
 lcd.print(':');
 lcd.print(now.minute());
 lcd.print(':');
 lcd.print(now.second());
```

```
  lcd.print("    ");
 }



 void updateSerial()
{
  delay(500);
  while (Serial.available())
  {
    mySerial.write(Serial.read());//Forward what Serial received to Software Serial Port
  }
  while(mySerial.available())
  {
    Serial.write(mySerial.read());//Forward what Software Serial received to Serial Port
  }
}
```

References

https://www.pololu.com/product/2191

https://www.elprocus.com/what-is-arduino-uno-r3-pin-diagram-specification-and-applications/

https://stackoverflow.com/questions/62950561/sim800l-gsm-module-working-with-http-and-shows-error-with-https-why

https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/

https://arduino.stackexchange.com/questions/47068/sim800l-not-registering-to-network

https://lastminuteengineers.com/ds3231-rtc-arduino-tutorial/

https://exploreembedded.com/wiki/Setting_up_GPRS_with_SIM800L

https://nettigo.eu/products/sim800l-gsm-grps-module#:~:text=SIM800L%20is%20a%20miniature%20cellular,that%20require%20long%20range%20connectivity.