



Design and Development of an SMS Controlled Faculty's Status Display System.

Developed by:

Amit Roy

ID: 2012-2-55-014

Md.Kudrat-E-Khuda

ID: 2012-2-55-056

Department of Electronics & Communications Engineering

East West University

Project Supervisor:

Dr. Md. Habibur Rahman

Professor

Department of Electrical & Electronics Engineering

University of Dhaka

DECLARATION

We hereby declare that we completed our project report on the topic entitled “**Design and Development of an SMS Controlled Faculty’s Status Display System**” as well as Submitted report to the Department of Electronics and Communications Engineering of East West University in partial fulfillment of the requirement for the degree of B.Sc.in Electronics and Telecommunications Engineering, under the supervision **Dr. Md. Habibur Rahman, Professor of department Electrical and Electronics Engineering, Dhaka University.**

Signature:

.....
Amit Roy

ID: 2012-2-55-014

.....
Md.Kudrat-E-Khuda

ID: 2012-2-55-056

.....
Supervisor

Dr. Md. Habibur Rahman

Professor

Department of Electrical & Electronics Engineering

University of Dhaka

APPROVAL

This is to certify that the project titled as “**Design and Development of an SMS Controlled Faculty’s Status Display System**” submitted to the respected member of the faculty of Engineering for partial fulfillment of requirement for the degree of Bachelor of Electronics and Telecommunication Engineering by the following student and has been accepted as satisfactory.

Submitted By:

Amit Roy

ID: 2012-2-55-014

Md. Kudrat-E-Khuda

ID: 2012-2-55-056

.....
Dr. Md. Habibur Rahman
Professor,
Dept. of Electrical & Electronics Engineering
University of Dhaka

.....
Dr. M. Mofazzal Hossain
Chairperson & associated Professor,
Dept. of Electronics and Telecommunication Engineering
East West University

ACKNOWLEDGEMENT

We should like to thank our department for giving us this chance to do this project. We want to thank and deep appreciation of our supervisor **Dr. Md. Habibur Rahman**, Professor, Department of Electrical & Electronics Engineering, University of Dhaka for his constant guidance, supervision, constructive suggestions and constant support during this project work.

We have special acknowledgement for our group member understanding each other and working from beginning up to end.

At last we want to thank all of our friends for their moral support and valuable helpful discussion during this project work.

ABSTRACT

In this project, the idea of wireless Digital Faculty's Status Display System Controlled by SMS Using GSM Technology has been presented. Our main aim is to develop a system where any faculty can easily provide any important information for student and the information will be displayed in the LCD Display which is located on the door of the faculty's room. As a result, students will get the information easily when they visit the faculty's room. Here, we are trying to implement our system in such a way that it can display message when the faculty sends SMS from mobile phone to GSM module which is located on the faculty room. So in short, the GSM module which is located in the faculty's room receives the message from faculty teacher and displays on the LCD display. For the sake of security, the system has been made password protected. If anyone sends message without password it won't show anything on the display. The faculty teacher sends the message from anywhere and this message is displayed on LCD display.

Table of Contents

Chapter	page
1 INTRODUCTION	9
1.1 Project overview	10
1.2 Information transfer	11
1.2.1 Broadcast	
1.3 Project Overview	10
1.3.1 LCD display	11
1.3.2 GSM Module	11
1.3.3 GSM features	12
1.4 Potentiometer	12
1.4.1 Power Supply of Arduino	12
1.4.2 Switched-mode power supply	13
1.5 System operation	13
2 Arduino with LCD Display	14
2.1 Introduction	15
2.2 What is Arduino	15
2.3 History of Arduino	16
2.4 Programming Environment	16-17
2.5 Hardware	18
2.6 Programming Language	19
2.7 IDE Setup	19

2.8 Advantage of arduino	19
2.9 16×2 LCD Module Pin Out Diagram	20-21
2.10 Basics of LCD Displays	21
2.11 Basic structure of an LCD	21
2.12 Types of LCD/Liquid Crystal Displays	22
2.13 Color Liquid Crystal Display	22
2.14 FEATURES OF 16×2 ARDUINO with LCD Display	23
2.15 APPLICATIONS OF 16×2 ARDUINO LCD Display	23
3 Design of the system	24
3.1 Connection diagram Arduino with LCD display and switch	25
3.2 Block diagram – Arduino to 16×2 LCD Display	25-26
3.3 Building the Circuit	26-27
4 Hardware Profile	28
4.1 GSM Module	29
4.2 Use of GSM Module	29
4.3 GSM Feature	30
4.4 Block Diagram connection of GSM, Arduino with LCD	30
4.5 Which type of GSM Module uses	31
4.5.1 GSM Module SIM 800H features	31
4.6 System modeling and design	32
4.6.1 Circuit Description	32-33
4.7 Advantage of GSM Module	33
4.8 APPLICATIONS	33

4.9 LIMITATIONS	33
5 Software Requirement	34
5.1 SOFTWARE REQUIREMENTS	35
5.2 Arduino Software (IDE)	35
5.3 Writing Sketches	35
5.4 Embedded C	36
6 Programming of Arduino and GSM	37
6.1 Programming in Arduino with 16*2 LCD display	38-39
6.1.1 Adding Text to the Display	39
6.1.2 WARNING	40
6.2 GSM SIM 808H library for receive SMS	40
6.2.1 Data sheet	40
6.2.2 Programming	40-44
6.2.3 Programming Language Description of GSM	44-47
6.3 Combine Programming of GSM and Arduino with LCD Display	48-51
6.3.1 GSM programming of Password Protection	51-52
6.3.2 Programming language description of password Protection	50
7 Result, Conclusion and References	52-55

Chapter-1

Introduction

1.1 Project Overview:

This paper explains the design and Development of an SMS control faculty's status display system based on GSM (Global System for Mobile communication) for controlling by SMS (Short Message Service) in LCD display. It presents an SMS based on controlled system of faculty members and incorporating the widely used GSM to facilitate the communication of displaying message on LCD display by user's mobile phone. Its operation is based on C programming assembly language.

SIM 808 GSM module with a SIM card is interfaced to the ports of the arduino with the help of C programming language. A user sends a SMS via a registered number from his mobile phone at the receiver's end.

The GSM based Campus Display System (CDS) is aimed at the colleges and universities for displaying day-to-day information continuously or at regular intervals during the working hours. Being GSM-based system, it offers flexibility to display flash news or announcements faster than the programmable system. GSM-based campus display system can also be used at other public places like schools, colleges etc. without affecting the surrounding environment.

The CDS mainly consists of a GSM receiver and a display toolkit which can be programmed from an authorized mobile phone. It receives the SMS, validates the sending Mobile Identification Number (MIN) and displays the desired information after necessary code conversion. It can serve as an LCD display the important notices instantaneously thus avoiding the latency. Being wireless, the GSM based CDS is easy to expand and allows the user to add more display units at any time

The developed system in this paper is ideally based on GSM with interfacing Arduino and LCD display. It is simple to be installed and used by faculty members in different reasons. Besides, most people use their cell phones to communicate and send messages. In this project we make a hardware system which shows a message in LCD display when a user send a sms from any distance. It is also password protected system only that's person send sms which know the password. Without password it does not show any message in display. For this system

students can know the status of faculty member when faculty member send a sms and the status show in the display.

1.2. Information Transfer:

A coordinated sequence of user and telecommunication system action that causes information present at a source user to become present at a destination user. An information-transfer transaction usually consists of three consecutive phases called the access phase, the information-transfer phase, and the disengagement phase.

1.2.1. Broadcast

A term to describe communication, when a piece of information is sent or transmitted from one point to another points. There is just one sender, but the information is simultaneously sent to all connected receivers. In networking, a distinction is made between broadcasting and multicasting. Broadcasting sends a message to everyone on the network whereas multicasting sends a message to a select list of recipients

1.3. Components Overview

1.3.1 LCD display

LCD display is used as the physical notice board that displays the contents received through SMS, which is directly interfaced to microcontroller.

Project used 16x2 LCD display- that means 2 rows, each of 16 characters.

Example: Hitachi HD44780 module.

1.3.2. GSM Module

GSM (Global System for Mobile) / GPRS (General Packet Radio Service) TTL - Modules SIM800 Quad-band GSM / GPRS device, works on frequencies 850 MHZ, 900 MHZ, 1800MHZ and 1900 MHZ. It is very compact in size and easy to use as plug in GSM Modem. The Modem is designed with 5V DC TTL interfacing circuitry, which allows User to directly interface with 5v micro-controller (Arduino).

1.3.3 GSM features

A GSM module is a wireless modem that works with a GSM wireless network. It is a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator. It is suitable for SMS, as well as DATA transfer application in mobile phone to mobile phone interface. The modem can be interfaced with a Microcontroller using USART (Universal Synchronous Asynchronous Receiver and Transmitter) feature (serial communication).

1.4. Potentiometer:

A potentiometer is an instrument for measuring the potential (voltage) in a circuit. It works on the principle that the potential dropped between two points in a wire of uniform cross section is directly proportional to the distance between the points. The constant resistance potentiometer is a variation of the basic idea in which a variable current is fed through a fixed resistor. These are used primarily for measurements in the millivolt and microvolt range. The constant resistance potentiometer is a variation of the basic idea in which a variable current is fed through a fixed resistor. These are used primarily for measurements in the millivolt and microvolt range.

1.4.1 Power Supply of Arduino:

- must be a DC adapter (i.e. it has to put out DC, not AC);
- should be between 9V and 12V DC (see note below);
- must be rated for a minimum of 250mA current output, although you will likely want something more like 500mA or 1A output, as it gives you the current necessary to power a servo or twenty LEDs if you want to.
- must have a 2.1mm power plug on the Arduino end, and the plug must be "center positive", that is, the middle pin of the plug has to be the + connection.

1.4.2 Switched-mode power supply:

A switched-mode power supply is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently .

Like other power supplies, an SMPS transfers power from a source. while converting voltage and current characteristics. Unlike a linear power supply, the pass transistor of a switching-mode supply continually switches between low-dissipation, full-on and full-off states, and spends very little time in the high dissipation transitions, which minimizes wasted energy. Ideally, a switched-mode power supply dissipates no power. Voltage regulation is achieved by varying the ratio of on-to-off time. In contrast, a linear power supply regulates the output voltage by continually dissipating power in the pass transistor .

1.5 System operation:

The operation of the system is very simple. A user can send a message from any of the distant using GSM mobile. For sending the text message from LCD display we need to interface the mobile phone with GSM Modem. For developing some of GSM based applications we need to have some commons peripherals including GSM MODEM, SIM, Arduino , LCD (Liquid crystal display), power supply and also some connecting wires. Moreover GSM based applications could be easily developed and enhanced due to easily accessibility of components in local markets at very pocket friendly prices .

Chapter 2

Arduino with LCD Display

2.1 Introduction: A Liquid Crystal Display commonly abbreviated as LCD is basically a display unit built using Liquid Crystal technology. When we build real life/real world electronics based projects, we need a medium/device to display output values and messages. The most basic form of electronic display available is 7 Segment display – which has its own limitations. The next best available option is Liquid Crystal Displays which comes in different size specifications. Out of all available LCD modules in market, the most commonly used one is 16×2 LCD Module which can display 32 ASCII characters in 2 lines (16 characters in 1 line). Other commonly used LCD displays are 20×4 Character LCD, Nokia 5110 LCD module, 128×64 Graphical LCD Display and 2.4 inch TFT Touch screen LCD display.

2.2 What is arduino:

The Arduino is a single-board microcontroller, intended to make the application of interactive objects or environments more accessible .The Arduino is designed to make the process of using electronics multidisciplinary projects more accessible. Its surroundings by controlling the lights, motors, LCD display and other actuators.



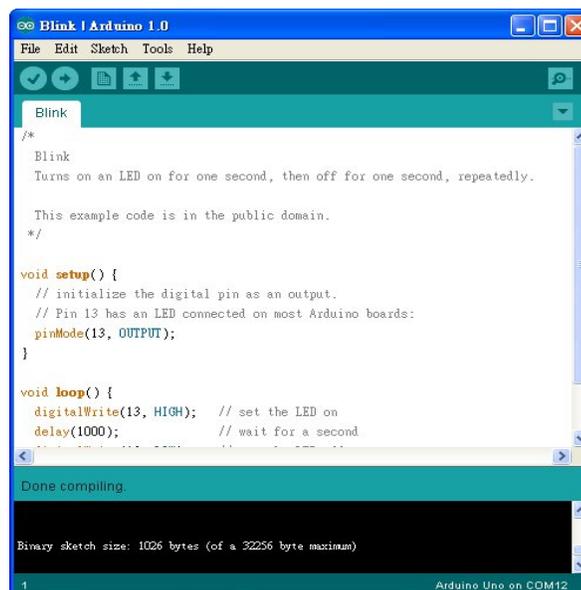
Fig.2.2

2.3 History of Arduino:

- The project began in Ivrea, Italy in 2005 to make a device for controlling student-built interaction design projects less expensively than other prototyping systems available at the time.
- As of October 2008 more than 50,000 arduino boards had been shipped. This has risen to more than 120,000 shipped boards as of February 2010.

2.4 Programming Environment

- The Arduino can be programmed with the Arduino software IDE (integrated development environment).
- Its Includes a code editor with brace matching, automatic indentation, API lookup, capable of compiling and uploading programs to the board with one click
- This software can runs on Windows, Mac OS X and Linux.
- Written in Java and based on processing programming language, avr-gcc and other open source software.



```
Arduino IDE - Blink | Arduino 1.0
File Edit Sketch Tools Help
Blink
/*
 * Blinks an LED on pin 13.
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
}
```

Done compiling.

Binary sketch size: 1026 bytes (of a 32256 byte maximum)

1 Arduino Uno on COM12

Fig 2.4

2.5 Hardware:

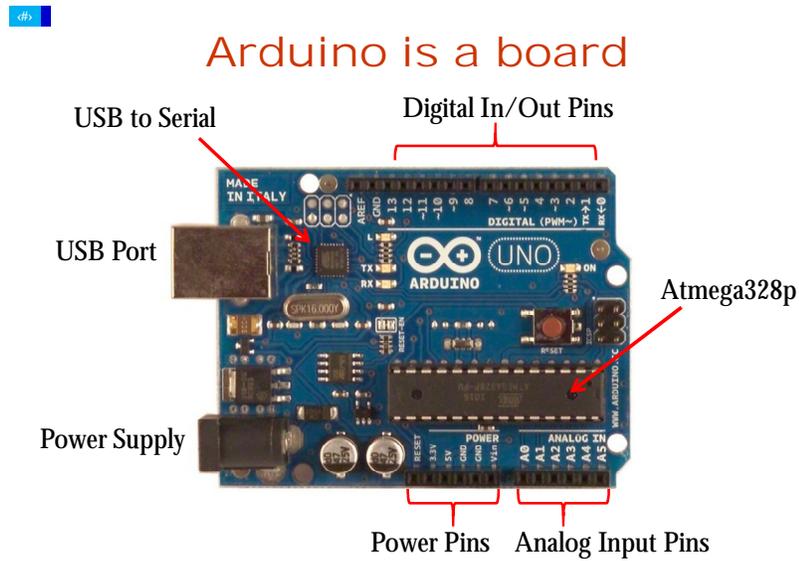


Fig 2.5

- Arduino board
- USB (standard A-B) cable
- Resistors, potentiometers, LEDs, buttons, etc. Wires, breadboards.

2.6. Programming Language:

- Code written in Arduino is translated into C language.
- Passed on to the avr-gcc compiler that makes the translation into the language understood by the microcontroller.
- Uploaded to the board.

2.7 IDE Setup:

- Download and install Arduino (<http://www.arduino.cc/en/Main/Software>) and Processing (<http://processing.org>).
- Connect the board to the USB port on computer.
- Run Arduino IDE.
- Set the correct port your device is using
- Tools -> Serial Port -> select port beginning with “/dev/cu.usbserial-“
- Set the Arduino board type your are using.
Tools -> Board -> select correct board type (printed on the chip)

2.8. Advantage of ardiuo:

- Open source
- Simplified and user-friendly programming language
- No additional programmer/burner hardware required for programming board
- Portable
- Low power consumption

2.9. 16×2 LCD Module Pin Out Diagram :

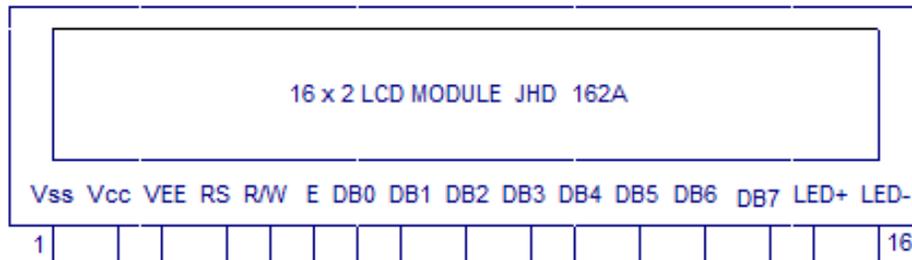


Fig 2.9

Pin1(Vss): Ground pin of the LCD module.

Pin2(Vcc): Power to LCD module (+5V supply is given to this pin)

Pin3(VEE): Contrast adjustment pin. This is done by connecting the ends of a 10K potentiometer to +5V and ground and then connecting the slider pin to the VEE pin. The voltage at the VEE pin defines the contrast. The normal setting is between 0.4 and 0.9V.

Pin4(RS): Register select pin. The JHD162A has two registers namely command register and data register. Logic HIGH at RS pin selects data register and logic LOW at RS pin selects command register. If we make the RS pin HIGH and feed an input to the data lines (DB0 to DB7), this input will be treated as data to display on LCD screen. If we make the RS pin LOW and feed an input to the data lines, then this will be treated as a command (a command to be written to LCD controller – like positioning cursor or clear screen or scroll).

Pin5(R/W): Read/Write modes. This pin is used for selecting between read and write modes. Logic HIGH at this pin activates read mode and logic LOW at this pin activates write mode.

Pin6(E): This pin is meant for enabling the LCD module. A HIGH to LOW signal at this pin will enable the module.

Pin7(DB0) to Pin14(DB7): These are data pins. The commands and data are fed to the LCD module through these pins.

Pin15(LED+): Anode of the back light LED. When operated on 5V, a 560 ohm resistor should be connected in series to this pin. In arduino based projects the back light LED can be powered from the 3.3V source on the arduino board.

Pin16 (LED-): Cathode of the back light LED.

2.10 Basics of LCD Displays:

The liquid-crystal display has the distinct advantage of having low power consumption than the LED. It is typically of the order of microwatts for the display in comparison to the some order of milliwatts for LEDs. Low power consumption requirement has made it compatible with MOS integrated logic circuit. Its other advantages are its low cost, and good contrast. The main drawbacks of LCDs are additional requirement of light source, a limited temperature range of operation (between 0 and 60° C), low reliability, short operating life, poor visibility in low ambient lighting, slow speed and the need for an ac drive.

2.11 Basic structure of an LCD:

A liquid crystal cell consists of a thin layer (about 10 μ m) of a liquid crystal sandwiched between two glass sheets with transparent electrodes deposited on their inside faces. With both glass sheets transparent, the cell is known as transmissive type cell. When one glass is transparent and the other has a reflective coating, the cell is called reflective type. The LCD does not produce any illumination of its own. It, in fact, depends entirely on illumination falling on it from an external source for its visual effect

2.12 Types of LCD/Liquid Crystal Displays:

Two types of display available are dynamic scattering display and field effect display.

When dynamic scattering display is energized, the molecules of energized area of the display become turbulent and scatter light in all directions. Consequently, the activated areas take on a frosted glass appearance resulting in a silver display. Of course, the unenergized areas remain translucent.

Field effect LCD contains front and back polarizer's at right angles to each other. Without electrical excitation, the light coming through the front polarizer is rotated 90° in the fluid.

Now, let us take a look at the different varieties of liquid crystals that are available for industrial purposes. The most usable liquid crystal among all the others is the nematic phase liquid crystals.

2.13 Color Liquid Crystal Display

Color LCD's are those that can display pictures in colors. For this to be possible there must be three sub-pixels with red, green and blue color filters to create each color pixel. For combining these sub-pixels these LCDs should be connected to a large number of transistors. If any problem occurs to these transistors, it will cause a bad pixel.

One of the main disadvantages of these types of LCDs is the size. Most manufacturers try to reduce the height than gain it. This is because more transistors and greater pixels will be needed to increase the length. This will increase the probability of bad pixels. It is very difficult or also impossible to repair a LCD with bad pixels. This will highly affect the sale of LCDs.

2.14 FEATURES OF 16×2 ARDUINO With LCD Display :

- Operating Voltage: **5V.**
- Backlight Colour: **Green.**
- Horizontal Characters: **16.**
- No of lines: **2.**
- Supports MCU speeds: **up to 2Mhz.**

2.15 APPLICATIONS OF 16×2 ARDUINO LCD DISPLAY

- As a Display Device.
- Displaying data on Backlit Screen.
- DIY projects requiring need for a display Device.

Chapter 3

Design of the system

3.1 Connection diagram arduino uno with LCD display with GSM and switch:

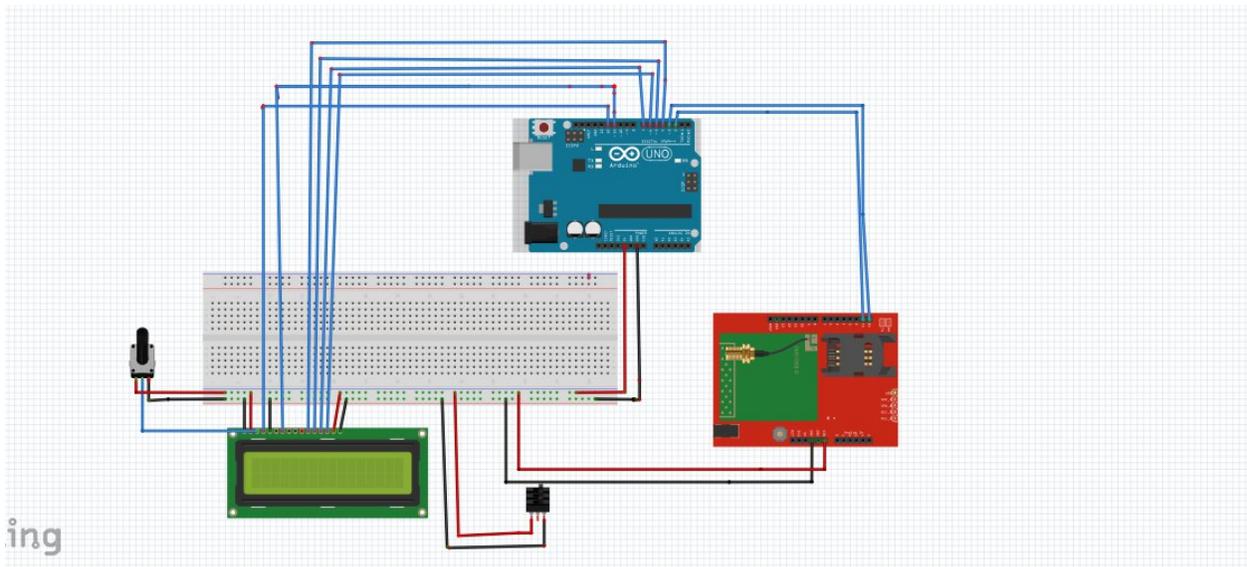


Fig: 3.1

3.2 Block diagram – Arduino to 16×2 LCD Display:

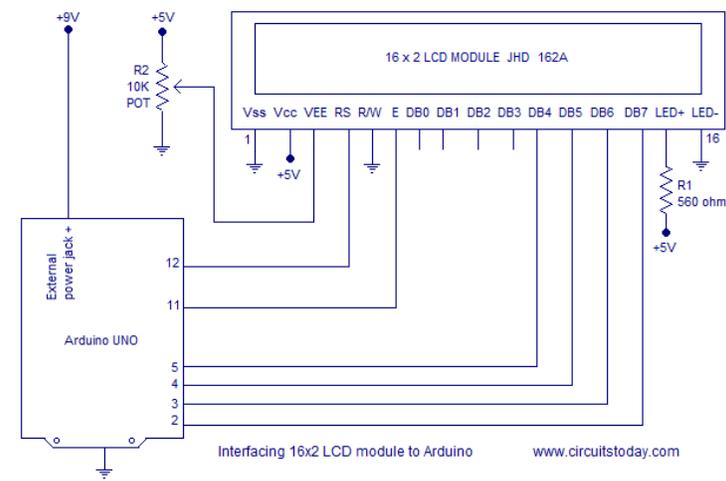


Fig: 3.2

The circuit diagram shows that the following connections are made:

Arduino → LCD

Pin 2 → DB7

Pin 3 → DB6

Pin 4 → DB5

Pin 5 → DB4

Pin 11 → E (Enable)

Pin 12 → RS

➤ The following are all connected to the Arduino GND:

- The LCD R/W pin
- One of the outer pins of the potentiometer
- The LCD VSS pin
- The LCD backlight cathode pin (BLC)

➤ The following are connected to the Arduino 5V pin:

- The LCD VDD pin
- One of the potentiometer outer pins

The condition of switch case:

S3	S2	S1	STATUS
0	0	0	Mobile mode
0	0	1	In the room
0	1	0	In the class
0	1	1	In the meeting
1	0	0	In the campus
1	0	1	Out the campus
1	1	0	I am busy
1	1	1	You may come

3.3 Building the Circuit:

After preparing your LCD display by soldering a pin header to it, get ready to plug it into the breadboard. Also be sure to check the datasheet for your LCD to see what the pin names are so that you can connect it correctly. Be very careful not to connect the LCD power pins the wrong way around as this will most likely destroy the LCD.

Insert the LCD into the top holes of the top vertical breadboard connecting strips. Do not insert it into either of the top horizontal rails. Get four wire links ready for connecting to the LCD data lines.

Connect DB7 of the LCD to pin 6 of the Arduino, DB6 to pin 7 of the Arduino, DB5 to pin 4 of the Arduino and DB4 to pin 5 of the Arduino

Connect the LCD E pin to pin 11 of the Arduino and the LCD RS pin to pin 12 of the Arduino.

Connect the R/W pin of the LCD to the bottom rail of the breadboard. Connect the Arduino GND to the bottom rail of the breadboard.

Connect the 10k potentiometer so that the centre pin connects to the LCD V0 pin. If using a trimpot, make sure that it is placed across the centre breadboard channel.

Connect the VDD pin of the LCD to one of the potentiometer outer legs. Connect this same leg to the Arduino 5V pin

Connect the other outer leg of the potentiometer to the bottom GND rail of the breadboard. Connect the LCD VSS pin to GND. Connect the LCD backlight cathode BLC pin to GND if present on your LCD.

Chapter 4

Hardware Profile

4.1 GSM Module:

- A GSM module is a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone.
- From the mobile operator perspective, a GSM module looks just like a mobile phone.
- When a GSM module is connected to a computer, this allows the computer to use the GSM modem to communicate over the mobile network.
- While these GSM module are most frequently used to provide mobile internet connectivity, many of them can also be used for sending and receiving SMS.

4.2 Use of GSM Module:

- SIM 800 GSM Module – Fixed Cellular Terminal for data applications.
- Compact & portable, easy to set up.
- A wired connection at one end with the help of RS232C serial port and wireless at the other.
- Features like SMS receiving, call receiving.
- LED Function including operating Status
- External antenna (via SMA)
Serial and control link
- Power Supply

4.3 GSM Feature:

- Multiple receiver MODEMS at the different positions in a geographical are carrying duplicate (cloned) SIM Cards.
- Multilingual display.
- Graphical Display.
- Use of MMS technology.
- Power supply
- Power supply switch
- SIM card holder
- GSM module
- JP5 & JP6 connectors
- GSM module I/O connectors
- RS232

4.4 Block Diagram connection of GSM, Ardiuno with LCD:

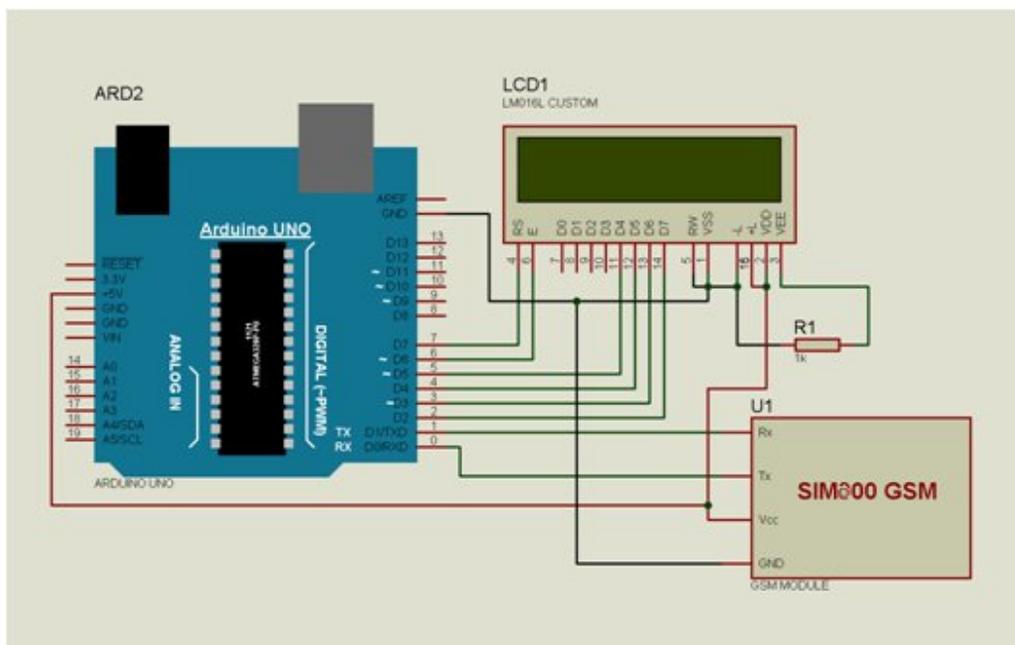


Fig 4.4

4.5 Which type of GSM Module uses:

In this project we use SIM 808H Module. SIM800H is a complete Quad-band GSM/GPRS solution in a LGA type which supports Quad band 808/850/900/1800/1900MHz. It can transmit Voice, SMS and can be embedded in the customer applications. SIM 800H data information is low power consumption. With tiny size of 15.8*17.8*2.4 mm, it can fit into slim and compact demands of customer design. Featuring of GSM 800H Module is Bluetooth, FM and Embedded AT, it allows total cost savings and fast time-to-market for customer applications.

4.5.1 GSM Module SIM 800H features:

Feature	Implementation
Power supply	3.4V ~4.4V
Power saving	Typical power consumption in sleep mode is 1.2mA
Ground	Ground Level of Power Supply
TXD (Transmit)	<ul style="list-style-type: none">• Outputs data bytes at voltage Level same as the V Interface• Pin Usually connected to the Rx pin of the microcontroller
RXD (Receive)	<ul style="list-style-type: none">• Receives data bytes at voltage Level same as the V Interface Pin.• Usually connected to the TX pin of the microcontroller

4.6 System modeling and design:

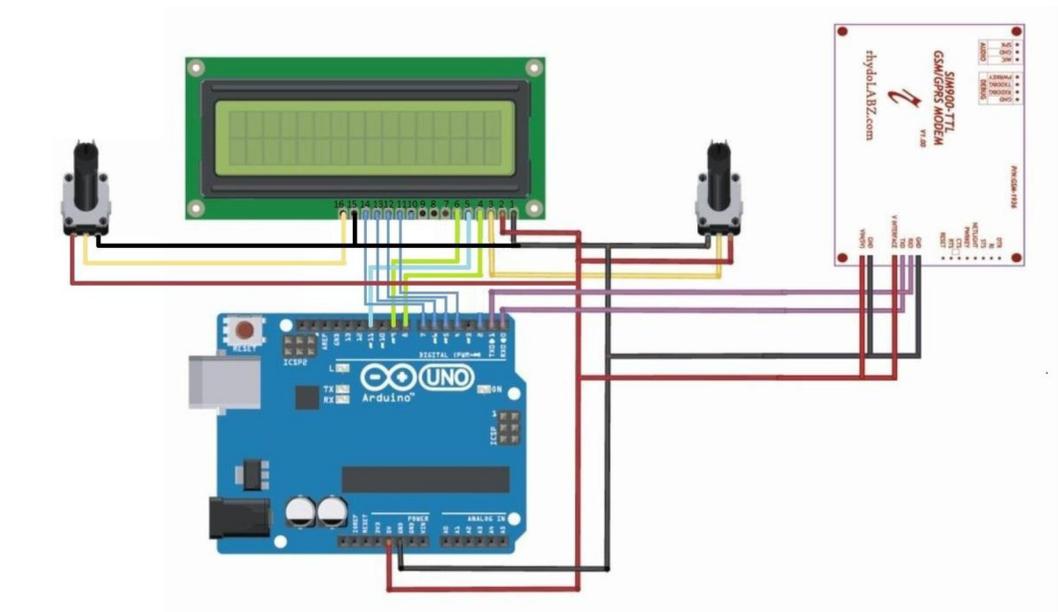


Fig 4.6

4.6.1 Circuit Description:

- The circuit mainly consists of Microcontroller, LCD display, GSM module, necessary resistors, capacitors, pots and an External DC Power supply.
- The microcontroller 5v dc power supply is given to the μ c with necessary clock pulse of 16 MHz given using crystal oscillator.
- Reset button can be used to clear registers and to execute program from beginning.
- 16 \times 2 LCD display is used for displaying data received through a valid SMS.
- 5V DC power supply is given for both LCD and backlight. Two 10k pots are used, one two adjust the contrast and another to control the brightness Of LED backlight.
- Here LCD is working in 4-bit mode that requires seven I/O pins from the μ c 4 data lines, RS,R/W,ENABLE. For reading or writing data to the display, logic1 is set to RS pin and 1,0 respectively to the R/W pin.
- GSM GPRS TTL UART MODEM-SIM900 is powered with 5v Dc. 5v dc is given also to the Interface pin for proper interfacing with μ c.

- The serial communication is accomplished between Modem and μc through RX, TX pins. TXD pin of modem is connected to the Rx (2) pin of μc and RXD of modem to the Tx(3) of μc .
- GSM module is configured and processed through a set of C programming Language. The μc is programmed in such a manner that it gives necessary C programming language to the GSM module for initialization the reading the incoming sms, sending reply to the users etc.

4.7 Advantage of Gsm Module:

- The status of various devices can be monitored and controlled from anywhere.
- The operation of the system is very simple and can be used by anyone with a basic knowledge of operating mobile phones.
- Easy to upgrade as per the user requirement.

4.8 APPLICATIONS:

- Home Automation
- Office Automation
- Industrial Automation.
- The security system can be employed in any organization.

4.9 LIMITATIONS:

- If the main module or the handset is not in the cellular range ,we can't control the devices.
- The sensors use RF transmitter and receiver which has limited range, hence sensors have to be in close vicinity of the main module

Chapter 5

Software Requirement

5.1 SOFTWARE REQUIREMENTS:

- Keil an ARM Company makes C compilers, macro assemblers, real-time kernels, debuggers, simulators, integrated environments, evaluation boards, and emulators for ARM7/ARM9/Cortex-M3, XC16x/C16x/ST10, 251, and 8051 MCU families.
- Compilers are programs used to convert a High Level Language to object code. Desktop compilers produce an output object code for the underlying microprocessor, but not for other microprocessors

5.2 Arduino Software (IDE):

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

5.3 Writing Sketches:

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension.ino. The editor 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 the Arduino Software (IDE), including complete error messages and other information. The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

5.4 Embedded C

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations. The C source code is very high level language, meaning that it is far from being at the base level of the machine language that can be executed by a processor. This machine language is basically just zero's and one's and is written in Hexadecimal format, that why they are called HEX files. There are several types of HEX files; we are going to produce machine code in the INTEL HEX-80 format, since this is the output of the KEIL IDE that we are going to use. *Figure* shows that to convert a C program to machine language, it takes several steps depending on the tool you are using, however, the main idea is to produce a HEX file at the end. This HEX file will be then used by the burner to write every byte of data at the appropriate place in the EEPROM of the 89S52.

Chapter 6

Programming of Arduino and GSM Module

6.1 Programming in arduino with 16*2 LCD display:

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 7, 6);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Proh.Habib");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

6.1.1 Adding Text to the Display:

We add some text and an incrementing number to the display. This exercise demonstrates how to initialize the display, how to write text, and how to move the cursor. First, include the Liquid Crystal library.

```
#include <LiquidCrystal.h>
```

Initialize an LCD object, as follows:

```
LiquidCrystal lcd(12, 11, 5, 4, 7, 6);
```

The arguments for the LCD initialization represent the Arduino pins connected to RS, EN, D4, D5, D6, and D7, in that order. In the setup, you call the library's `begin()` function to set up the LCD display with the character size.

```
lcd.begin(16, 2);
```

After doing that, you can call the library's `print()` and `setCursor()` commands to print text to a given location on the display. For example, if you want to print your name on the second line, you issue these commands:

```
lcd.setCursor(0, 1);
```

```
lcd.print("prof.Habib");
```

The positions on the screen are indexed starting with (0,0) in the top-left position. The first argument of `setCursor()` specifies which column number, and the second specifies which row number. By default, the starting location is (0,0). So, if you call `print()` without first changing the cursor location, the text starts in the top-left corner.

6.1.2 WARNING:

The library does not check for strings that are too long. So, if you try to print a string starting at position 0 that is longer than the number of characters in the row

you are addressing, you might notice strange behavior. Make sure to check that whatever you are printing will fit on the display!!!

6.2 GSM Sim 800H libray for recive sms:

This sketch, for the Arduino GSM module, waits for a SMS message and displays it through the Serial port.

6.2.1 Data sheet:

- GSM module attached to and Arduino.
- TX of GSM is connected to RX of arduino.
- TX of GSM is connected to TX of arduino.
- SIM card that can receive SMS messages.

6.2.2 Programming:

```
// include the library code:
```

```
#include <LiquidCrystal.h>
```

```
// initialize the library with the numbers of the interface pins
```

```
LiquidCrystal lcd(12, 11, 5, 4, 7, 6);
```

```
// include the GSM library
```

```
#include <GSM.h>
```

```
// PIN Number for the SIM
```

```
#define PINNUMBER ""
```

```
// initialize the library instances
```

```
GSM gsmAccess;
```

```
GSM_SMS sms;
```

```

// Array to hold the number a SMS is retrieved from
char senderNumber[20];

void setup()
{
  // initialize serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  Serial.println("SMS Messages Receiver");

  // connection state
  boolean notConnected = true;

  // Start GSM connection
  while(notConnected)
  {
    if(gsmAccess.begin(PINNUMBER)==GSM_READY)
      notConnected = false;
    else
    {
      Serial.println("Not connected");
      delay(1000);
    }
  }
}

```

```

    }
}

Serial.println("GSM initialized");
Serial.println("Waiting for messages");
lcd.begin(16, 2);
// Print a message to the LCD.
//lcd.setCursor(0,1);
lcd.print("Prof. Habib");
}

void loop()
{
  char c, msg[20]="Null";
  int i =0;

  // If there are any SMSs available()
  if (sms.available())
  {
    Serial.println("Message received from:");

    // Get remote number
    sms.remoteNumber(senderNumber, 20);
    Serial.println(senderNumber);

    // An example of message disposal

```

```

// Any messages starting with # should be discarded
if(sms.peek()=='#')
{
  Serial.println("Discarded SMS");
  sms.flush();
}

// Read message bytes and print them
while(c=sms.read())
{
  Serial.print(c);
  msg[i]=c;
  i++;
}
msg[i]='\0';

Serial.println("\nEND OF MESSAGE");

// Delete message from modem memory
sms.flush();

Serial.println("MESSAGE DELETED");
if((msg[0]=='1') && (msg[1]=='2') && (msg[2]=='3') && (msg[3]=='4'))
{
  lcd.setCursor(0, 2);

  // print the number of seconds since reset:for
  lcd.print(msg);
}

```

```
for(i=0;i<16;i++)
msg[i]=msg[i+4];
msg[16]='\0';
}
delay(1000);
}

delay(1000);
}
```

6.2.3 Programming Language Description of GSM:

GSM constructor:

- Description: GSM is the base class for all GSM based functions
- Syntax
 - GSM GSMAccess
 - GSM GSMAccess(debug)

Void()

The void keyword is used only in function declarations. It indicates that the function is expected to return no information to the function from which it was called.

if (conditional)

if, which is used in conjunction with a comparison operator, tests whether a certain condition has been reached, such as an input being above a certain number.

if / else

- if/else allows greater control over the flow of code than the basic if statement, by allowing multiple tests to be grouped together. For example, an analog input could be tested and one action taken if the input was less than 500, and another action taken if the input was 500 or greater.
- else can proceed another if test, so that multiple, mutually exclusive tests can be run at the same time.
- Each test will proceed to the next one until a true test is encountered. When a true test is found, its associated block of code is run, and the program then skips to the line following the entire if/else construction. If no test proves to be true, the default else block is executed, if one is present, and sets the default behavior.

begin()

Description: Checks the modem status, and restarts it. Call this before GSMModem.getIMEI().

while loops:

Description: **while** loops will loop continuously, and infinitely, until the expression inside the parenthesis, () becomes false. Something must change the tested variable, or the **while** loop will never exit. This could be in your code, such as an incremented variable, or an external condition, such as testing a sensor.

- **Parameters:** expression - a (boolean) C statement that evaluates to true or false

Serial print():

Description: Prints data to the serial port as human-readable ASCII text. This command can take many forms. Numbers are printed using an ASCII character for each digit. Floats are similarly printed as ASCII digits,

defaulting to two decimal places. Bytes are sent as a single character. Characters and strings are sent *as is*.

delay() :

- **Description:** Pauses the program for the amount of time (in milliseconds) specified as parameter. (There are 1000 milliseconds in a second.)
- **Syntax:** delay (ms)
- **Parameters:** the number of milliseconds to pause

flush() :

- **Description:** flush () clears the modem memory of any sent messages once all outgoing characters have been sent. flush() inherits from the Stream utility class.
- **Syntax:** SMS.flush()

Char

Description: A data type that takes up 1 byte of memory that stores a character value. Character literals are written in single quotes, like this: 'A' (for multiple characters - strings - use double quotes: "ABC").

Characters are stored as numbers however. You can see the specific encoding in the ASCII chart. This means that it is possible to do arithmetic on characters, in which the ASCII value of the character is used (e.g. 'A' + 1 has the value 66, since the ASCII value of the capital letter A is 65). See Serial.println reference for more on how characters are translated to numbers.

The char data type is a signed type, meaning that it encodes numbers from -128 to 127. For an unsigned, one-byte (8 bit) data type, use the *byte* data type.

int()

- Description: Converts a value to the int data type.
- Syntax: int(x)

6.3 Combine Programming of GSM, Arduino and switch with LCD display:

SMS receiver

This example is in the public domain.

<http://arduino.cc/en/Tutorial/GSMExamplesReceiveSMS>

```
*/  
  
// include the library code:  
#include <LiquidCrystal.h>  
  
// initialize the library with the numbers of the interface pins  
LiquidCrystal lcd(12, 11, 5, 4, 7, 6);  
  
// include the GSM library  
#include <GSM.h>  
  
// PIN Number for the SIM  
#define PINNUMBER ""  
  
// initialize the library instances
```

```

GSM gsmAccess;

GSM_SMS sms;

// Array to hold the number a SMS is retrieved from
char senderNumber[20];

void setup()
{
  // initialize serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  Serial.println("SMS Messages Receiver");

  // connection state
  boolean notConnected = true;

  // Start GSM connection
  while(notConnected)
  {
    if(gsmAccess.begin(PINNUMBER)==GSM_READY)
      notConnected = false;
    else
    {

```

```

Serial.println("Not connected");
delay(1000);
}
}

Serial.println("GSM initialized");
Serial.println("Waiting for messages");
lcd.begin(16, 2);
// Print a message to the LCD.
//lcd.setCursor(0,1);
lcd.print("Prof. Habib");
}

// If there are any SMSs available()
if (sms.available())
{
Serial.println("Message received from:");

// Get remote number
sms.remoteNumber(senderNumber, 20);
Serial.println(senderNumber);

// An example of message disposal
// Read message bytes and print them
while(c=sms.read())
{

```

```

Serial.print(c);

msg[i]=c;

i++;

}

msg[i]='\0';

Serial.println("\nEND OF MESSAGE");

// Delete message from modem memory

delay(1000);

}

delay(1000);

}

```

6.3.1 GSM programming of Password Protection:

```

sms.flush();

Serial.println("MESSAGE DELETED");

if((msg[0]=='1') && (msg[1]=='2') && (msg[2]=='3') && (msg[3]=='4'))

{

lcd.setCursor(0, 2);

lcd.print("      ");

lcd.setCursor(0,2);

// print the number of seconds since reset:for

for(i=0;i<16;i++)

```

```
msg[i]=msg[i+4];  
msg[i]='\0';  
lcd.print(msg);  
}
```

6.3.2 Programming language descriptin of password protection:

`if((msg[0]=='1')`: The first character will show 1

`(msg[1]=='2')`: The second character will show 2

`(msg[2]=='3') && (msg[3]=='4')`: the third and fourth character show 3 and 4

`msg[i]='\0'`: '\0' is the null value. It hide the password from display

`for(i=0;i<16;i++)`: The display character will less then 16

Chapter 7

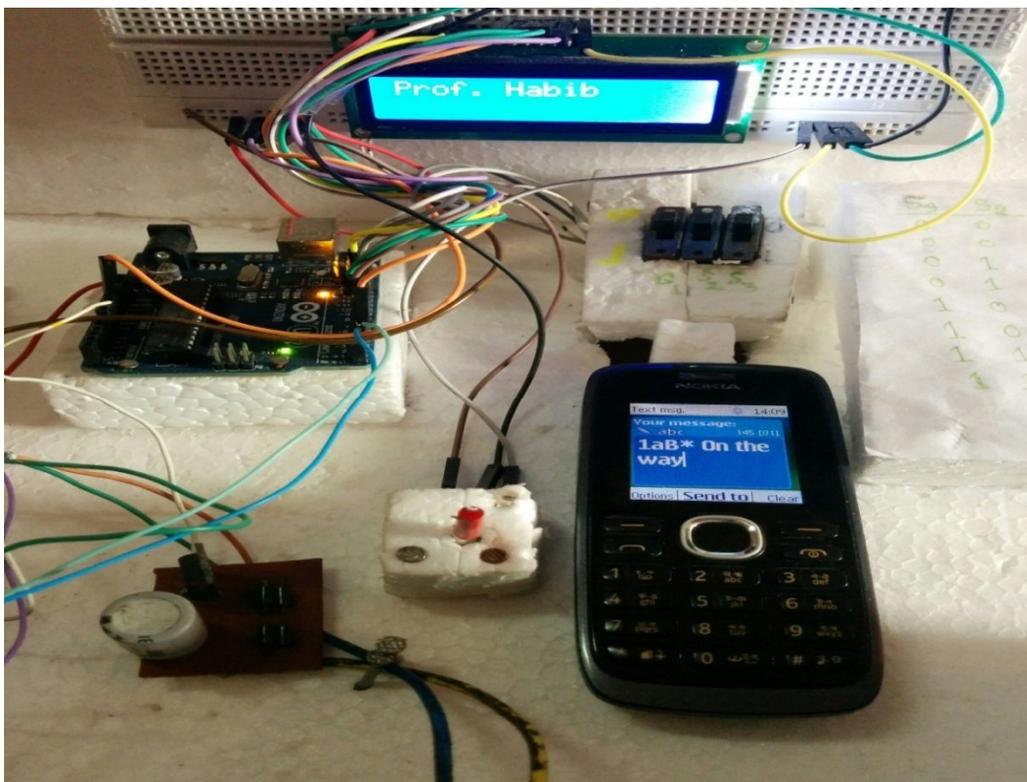
Result, Conclusion and **References**

Result:

Switch 000 is Mobile Mode.



When sending a message from mobile to GSM Module. Here “1aB*” is password. When we sending message from mobile to GSM module then we use this password.



Message shows in the LCD Display



When using switch ($S_3=0$, $S_2=1$, $S_1=1$) then this status shows in the LCD Display.



CONCLUSION

The prototype of the GSM based display toolkit was efficiently designed. This prototype has facilities to be integrated with a display board thus making it truly mobile. The toolkit accepts the SMS, stores it, validates it and then displays it in the LCD module. The SMS is deleted from the SIM each time it is read, thus making room for the next SMS. The major constraints incorporated are the use of the termination character of the SMS and the display of one SMS as a time. These limitations can be removed by the use of higher end microcontrollers and extended RAM. The prototype can be implemented using commercial display boards. In this case, it can solve the problem of instant a faculty teacher information transfer for the student.

REFERENCES

- <http://en.wikipedia.org/wiki/Wikipedia>
- <http://images.google.com>
- <https://www.arduino.cc/en/main/arduinoBoardUno>
- <http://www.slideshare.net/>
- [http:// learn.sparkfun.com](http://learn.sparkfun.com)
- <http://www.allaboutcircuits.com/projects/interface-an-lcd-with-an-arduino/>
- <http://www.instructables.com/id/How-to-use-an-LCD-displays-Arduino-Tutorial/>
- <https://arduino-info.wikispaces.com/LCD-Blue>
- <http://fritzing.org/projects/thermistor-output-to-lcd>
- <https://startingelectronics.org/beginners/start-electronics-now/tut14-arduino-LCD-thermometer/>
- http://scuola.arduino.cc/lesson/LVEVK1a/Using_SMS_messages_to_control_LED_color