SPEED CONTROLLER DESIGN FOR SPEED CONTROL OF SINGLE-PHASE INDUCTION MOTOR WITH THE HELP OF GSM AND ANDROID PHONE

PROJECT REPORT

Submitted by

N Sravan Kumar Reddy (0017117031)

T Lakshman Kumar (0017117041)

S Venkata Rami Reddy (0017117030)

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

Department of Electrical and Electronics

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SCHOOL OF ELECTRICAL SCIENCE

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

PADUR, CHENNAI-603 103

APRIL 2021

BONAFIDE CERTIFICATE

Certified that this project report titled **"SPEED CONTROLLER DESIGN FOR SPEED CONTROL OF SINGLE-PHASE INDUCTION MOTOR WITH THE HELP OF GSM AND ANDROID PHONE"** is the bonafide work of **"** N. SRAVAN KUMAR REDDY**(17117031),** T Lakshman Kumar (0017117041), S. VENKATA RAMI REDDY**(17117030)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertations on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr.A.K.PARVATHY P.hD
HEAD OF THE DEPARTMENT
Professor
Department of Electrical & Electronics Engineering
Hindustan University, OMR Road
Padur, Chennai

Dr.K.NAGESWARA RAO P.hD
SUPERVISOR
Assistant Professor
Department of Electrical &Electronics Engineering
Hindustan University, OMR Road,
Padur, Chennai

The Project Viva - Voce Examination is held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Induction Motors account for more than 85% of all motors used in industry and domestic applications. In the past they have been used as constant-speed motors as traditional speed control methods have been less efficient than speed control methods for DC motors. However, DC Motors require commutators and brushes which are hazardous and require maintenance. Thus Induction Motors are preferred. As many of the industries use induction motors. So, controlling of induction motor plays a very vital role. So, our project concentrates on controlling the speed of induction motor using Android phone remotely by the help of the GSM technology.

GSM based motor is able to control the motor from remote place, looking over its operating conditions, it gets feedback from the motor itself. The target here is to control the motor from distant place by mobile and also to get feedback by SMS while it is in ON or OFF condition. It also ensures the safe operation of the motor by detecting the voltage of the source and ensures feedback from system while it is over or under voltage.

It can also receive these feedbacks by SMS as GSM network is everywhere around the globe and that's why it is highly convenient. With the help of GSM network to operate motor, it also transfers feedback information through. Also implementation of GSM network is very fruitful because it does not need extra equipment for networking.

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

It's a matter of pride and privilege for me to express my deep gratitude to the management of HITS for providing me the necessary facilities and support.

I am highly elated in expressing my sincere and abundant respect to the Vice Chancellor Dr. S. N. SRIDHARA, for giving me this opportunity to bring out and implement my ideas in this project.

I wish to express my heartfelt gratitude to Dr. A. K. PARVATHY, Head of the Department of Electrical and Electronics Engineering for much of her valuable support encouragement in carrying out this work.

I would like to thank my internal guide Dr. K. NAGESWARA RAO, for continuously guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank my Project Co-Ordinator Ms. P. GETZIAL ANBU MANI, for continuously guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the Electrical and Electronics Department, who extended directly or indirectly all support.

Last, but not the least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF CONTENTS

CHAPTER 1

INTRODUCTION, SCOPE & OBJECTIVES OF THE INVESTIGATION

1.1 INTRODUCTION

The “Design and speed control of Induction Motor Using GSM and Android phone” implements the emerging applications of the GSM technology. With GSM networks, a control system has been proposed which acts as an embedded system and it can monitor or control appliances and other devices locally using various peripherals.

It is a very good example of integrated embedded system as all its operations are controlled by the pre-planned program inside the Atmega-microcontroller. Target here is to ON/OFF and speed control of single-phase induction motor, the electrical or electronic appliances connected to this system. System allows user to effectively monitor and control motor via mobile phone by sending commands in form of SMS and receive the status of appliance.

The main concept behind is to receive the sent SMS and process it further as required to perform various operations. Kind of the operation to be performed depends on the nature of the SMS sent and by making necessary changes, operation can be changed or improved. In whole operation, the sent SMS is stored and analyzed from the receiver mobile station and soon the required control signal is generated and later sent to the intermediate hardware will be designed and its nature will be according to command received from sent message.

A particular kind of GSM model is the soul of whole working operation. The sent message containing command motions the process later to perform the required task accordingly. The system here provides method to control the motor from a remote distance with the help of microcontroller and the program.

Electric induction motors run at fixed speed and are ideally suited to application where a constant motor output speed is required. However there are some application where varying motor output speed.

While equipment like conveyors may be fine for a fixed speed there are some applications which are better suited to running at variable speeds such as fan, pumps, winders and precision tools. A recent trend among customers required automation, to develop the motor varying the speed automatically in this project by using android Bluetooth. The AC induction motor is the most popular motor use in consumer and industrial application.

There are various method of controlling the speed of AC motor. There are several of method is available for speed control of ac motor one of the method is two vary frequency and voltage of motor. Speed modulation of a single-phase motor is usually achieved either by some electrical means, such as reducing supply voltage by auto-transformer, or by switching windings to change the number of motor poles for different operating condition as required.

Voltage control is best method, but it allows only limited speed range to be obtained. Now frequency acts as interesting alternative to voltage control, In frequency control method when we control the frequency of the motor the air gap flux is saturate and hence to maintain the air gap flux. Therefore, the stator voltage should also be reduced in proportional to the frequency so as to maintain the air-gap flux constant.

The magnitude of the stator flux is proportional to the ratio of the stator voltage and the frequency. Hence, if the ratio of voltage to frequency is kept constant, the flux remains constant. In our project the speed of the induction motor control by using GSM technique.

We get wide range of speed in optimum output by using android application. The present world of rapid technological changes there is an urgent demand for the best quality product and services, that can achieved by automation in industries. Android is the open source software, manufacturers can modified the operating system to suit their current need and phones. This become cheep and feasible alternative for the manufacturer.

Here the proposed system is designed to controlling the speed of induction motor remotely. Android Mobile acts as a transmitter and the received by Global Serial Module (GSM) which is connected to Atmega-Microcontroller. This system comes under wireless technology.

1.2 OBJECTIVES

1. To achieve wireless communication using Global Serial Module (GSM) in between Android phone and Atmega-Microcontroller for speed control of single-phase induction motor.
2. For advancement we are using the latest GSM model (SIM800I) for better and good performance to the project.
3. This system can be used instead, where the presence of human being is must and should.

CHAPTER 2

LITERATURE REVIEW

1. **Prof. R.R Jadhav, Prathmesh P Pandit, Shubham D. Pal, Vineet H. Risbud. "THREE PHASE MOTOR CONTROL USING GSM" M.Tech (E&TC), Assistant Professor (ETRX Department), COE Manjari (Bk), Pune, India1 BE, ETRX Department, COEM, Pune, India.**

   - ✔ In this paper authors designed this model to protect the motor against frequent power cut, over voltage, over current, three phase detection & dry-run helps to off motor during dry condition.
   - ✔ Consumer can check all the conditions and on/off the motor by sending text message by registered mobile number. If any abnormal condition is occurred motor automatically switch off & system gives information about fault to the user threw message.
   - ✔ The message send to the GSM controller perform the proper operation based on the motor condition and given task. The signals sent to the controller to switch on/off the motor through the starter the relays. The relays is controlled by the microcontroller.
   - ✔ Used ATMEGA 328 microcontroller for checking all conditions of motor.

2. **Chen, P.J. and Jiang, X.H. (2008) Design and Implementation of Remote Monitoring System Based on GSM. Pacific-Asia Workshop on Computational Intelligence and Industrial Application, PACIIA'08, IEEE, Wuhan.**

- ✔ Remote monitoring system based on SMS of GSM is presented in this journal by the authors.
- ✔ The system includes two parts which are the monitoring center and the remote monitoring station. The monitoring center consists of a computer and a TC35 communication module of GSM. The computer and TC35 are connected by RS232. The remote monitoring station includes a TC35 communication module of GSM, a MSP430F149 MCU, a display unit, various sensors, data gathering and processing unit.
- ✔ The software of the monitoring center and the remote monitoring station is designed by using '**VB'**.
- ✔ **Visual Basic** is a programming language and development environment created by Microsoft. The Visual Basic program also includes features like "IntelliSense" and "Code Snippets," which automatically generate code for visual objects added by the programmer.

CHAPTER 3

ATMEGA 328

ARDUINO NANO

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

3.1 Ardunio Diagram

1. Power USB

Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection.

2. Power (Barrel Jack)

Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.

3. Voltage Regulator

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4. Crystal Oscillator

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

5, 17. Arduino Reset

You can reset your Arduino board, i.e., starts your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button on the board. Second, you can connect an external reset button to the Arduino pin labeled RESET.

6,7,8,9 .Pins (3.3, 5, GND, Vin)

- 3.3V (6) − Supply 3.3 output volt
- 5V (7) − Supply 5 output volt
- Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.
- GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.

10. Analog pins

The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

11. Main microcontroller

Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

12. ICSP pin

Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

13. Power LED indicator

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

14. TX and RX LEDs

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins

responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15. Digital I/O

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.

16. AREF

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

3.1 ARDUINO IDE SETUP

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.

Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Step 5 − Open your first project.

Once the software starts, you have two options −

- Create a new project.
- Open an existing project example.

To create a new project, select File → **New**.

To open an existing project example, select File → Example → Basics → Blink.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to **Tools → Serial Port**menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note − If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software

In this chapter, we will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

Sketch − The first new terminology is the Arduino program called "**sketch**".

3.2 ARDUINO STRUCTURE:

Arduino programs can be divided in three main parts: **Structure, Values**(variables and constants), and **Functions**. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the **Structure**. Software structure consist of two main functions −

- Setup( ) function
- Loop( ) function

```
Void setup ( ) {

}
```

- **PURPOSE** − The **setup()** function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.
- **INPUT** − -
- **OUTPUT** − -
- **RETURN** − -

```
Void Loop ( ) {

}
```

- **PURPOSE** − After creating a **setup()** function, which initializes and sets the initial values, the **loop()** function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.
- **INPUT** − -
- **OUTPUT** − -
- **RETURN** − -

3.3 ARDUINO DATA TYPES

Data types in C refers to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in the storage and how the bit pattern stored is interpreted.

The following table provides all the data types that you will use during Arduino programming.

Void	Boolean	char	Unsigned char	byte	int	Unsigned int	word
Long	Unsigned long	short	float	double	array	String-char array	String-object

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.

Example

```
Void Loop ( ) {
  // rest of the code
}
```

Boolean

A Boolean holds one of two values, true or false. Each Boolean variable occupies one byte of memory.

Example

```
boolean val = false ; // declaration of variable with type boolean and initialize it with false
boolean state = true ; // declaration of variable with type boolean and initialize it with true
```

Char

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

However, characters are stored as numbers. You can see the specific encoding in the ASCII chart. This means that it is possible to do arithmetic operations on characters, in which the ASCII value of the character is used. For example, 'A' + 1 has the value 66, since the ASCII value of the capital letter A is 65.

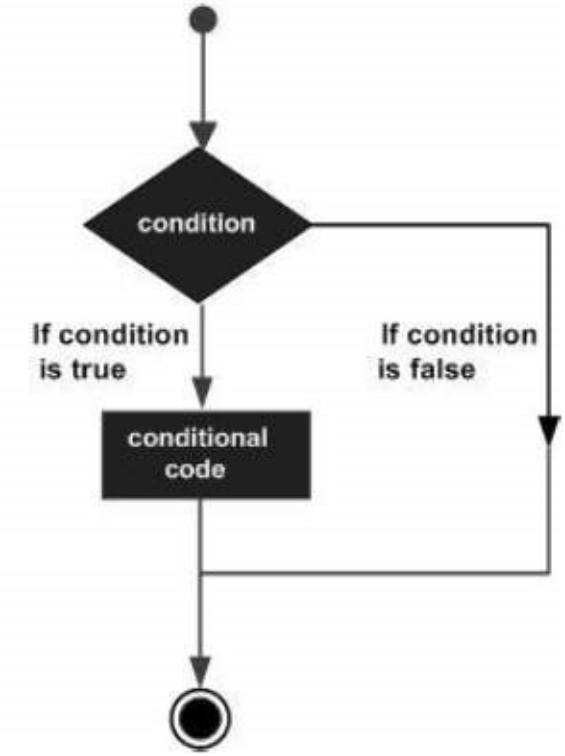
Example

```
Char chr_a = ‘a’ ;//declaration of variable with type char and initialize it with character a
Char chr_c = 97 ;//declaration of variable with type char and initialize it with character 97
```

3.4 ARDUINO CONTROL STATEMENTS:

Decision making structures require that the programmer specify one or more conditions to be evaluated or tested by the program. It should be along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

Following is the general form of a typical decision making structure found in most of the programming languages −



Control Statements are elements in Source Code that control the flow of program execution. They are −

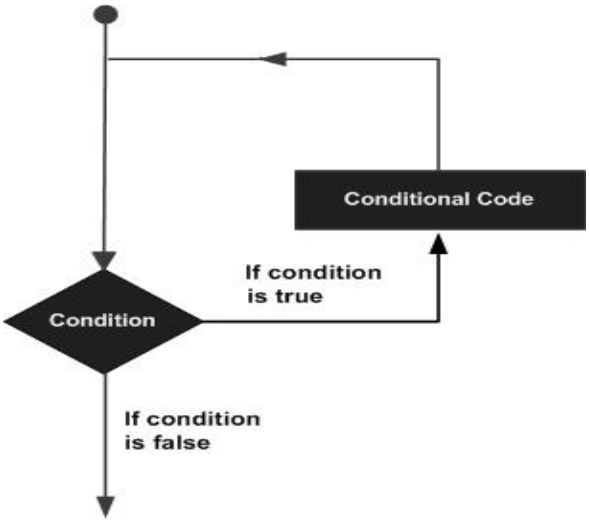
S.NO.	Control Statement & Description
1	**If statement** It takes an expression in parenthesis and a statement or block of statements. If the expression is true then the statement or block of statements gets executed otherwise these statements are skipped.
2	**If …else statement** An **if** statement can be followed by an optional else statement, which executes when the expression is false.
3	**If…else if …else statement** The **if** statement can be followed by an optional **else if...else**statement, which is very useful to test various conditions using single if...else if statement.

4	**switch case statement** Similar to the if statements, **switch...case** controls the flow of programs by allowing the programmers to specify different codes that should be executed in various conditions.
5	**Conditional Operator ? :** The conditional operator ? : is the only ternary operator in C

3.5 ARDUINO LOOPS

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages −



C programming language provides the following types of loops to handle looping requirements.

S.NO.	Loop & Description

1	**while loop** 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.
2	**do…while loop** The **do…while** loop is similar to the while loop. In the while loop, the loop-continuation condition is tested at the beginning of the loop before performed the body of the loop.
3	**for loop** A **for loop** executes statements a predetermined number of times. The control expression for the loop is initialized, tested and manipulated entirely within the for loop parentheses.
4	**Nested Loop** C language allows you to use one loop inside another loop. The following example illustrates the concept.
5	**Infinite loop** It is the loop having no terminating condition, so the loop becomes infinite.

3.6 ARDUINO STRINGS:

Strings are used to store text. They can be used to display text on an LCD or in the Arduino IDE Serial Monitor window. Strings are also useful for storing the user input. For example, the characters that a user types on a keypad connected to the Arduino.

There are two types of strings in Arduino programming −

- Arrays of characters, which are the same as the strings used in C programming.
- The Arduino String, which lets us use a string object in a sketch.

In this chapter, we will learn Strings, objects and the use of strings in Arduino sketches. By the end of the chapter, you will learn which type of string to use in a sketch.

String Character Arrays

The first type of string that we will learn is the string that is a series of characters of the type **char**. In the previous chapter, we learned what an array is; a consecutive series of the same type of variable stored in memory. A string is an array of char variables.

A string is a special array that has one extra element at the end of the string, which always has the value of 0 (zero). This is known as a "null terminated string".

String Character Array Example

This example will show how to make a string and print it to the serial monitor window.

Example

```
void setup() {
  char my_str[6]; // an array big enough for a 5 character string
  Serial.begin(9600);
  my_str[0] = 'H'; // the string consists of 5 characters
  my_str[1] = 'e';
  my_str[2] = 'l';
  my_str[3] = 'l';
  my_str[4] = 'o';
  my_str[5] = 0; // 6th array element is a null terminator
  Serial.println(my_str);
}

void loop() {

}
```

The following example shows what a string is made up of; a character array with printable characters and 0 as the last element of the array to show that this is where the string ends. The string can be printed out to the Arduino IDE Serial Monitor window by using **Serial.println()** and passing the name of the string.

This same example can be written in a more convenient way as shown below −

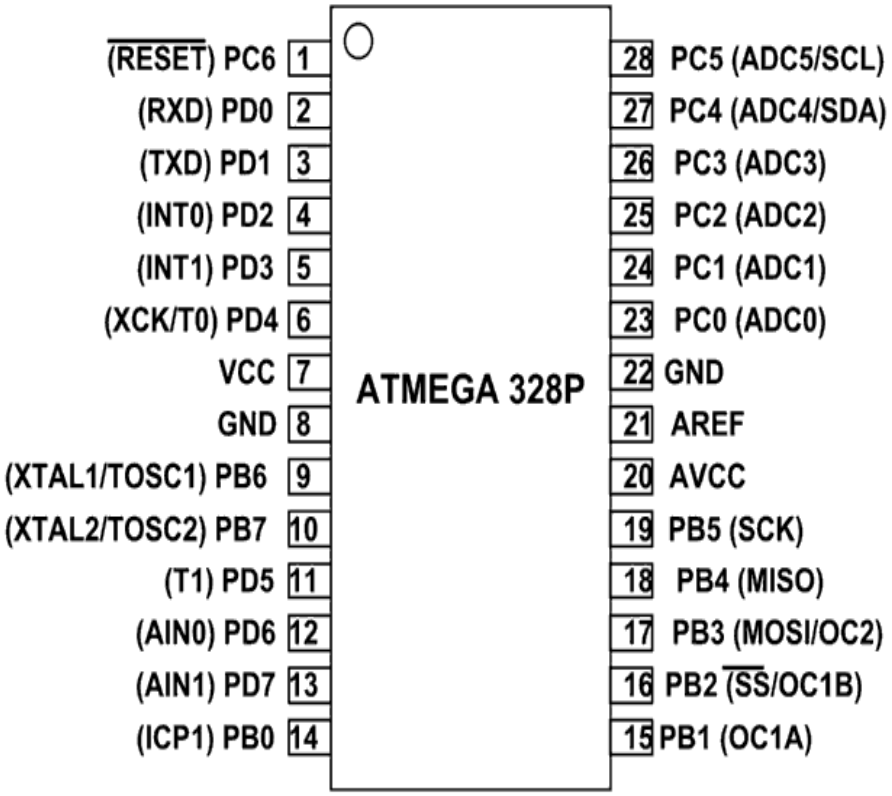
Example

```
void setup() {
  char my_str[] = "Hello";
  Serial.begin(9600);
  Serial.println(my_str);
}

void loop() {


}
```

In this sketch, the compiler calculates the size of the string array and also automatically null terminates the string with a zero. An array that is six elements long and consists of five characters followed by a zero is created exactly the same way as in the previous sketch.

3.7 ATMEGA328 CONTROLLER:

ATMEGA328P is high performance, low power controller from Microchip. ATMEGA328P is an 8-bit microcontroller based on AVR RISC architecture. It is the most popular of all AVR controllers as it is used in ARDUINO boards.



3.7 MICRO CONTROLLER CIRCUIT DIAGRAM

- List the functions to be executed by controller.
- Write the functions in programming language in IDE programs.

You can download the IDE program for free in company websites. IDE program for AVR controllers is 'ATMEL STUDIO'. Link for ATMEL STUDIO is given below.

(Usually Atmel Studio 6.0 for Windows7 [http://atmel-studio.software.informer.com/6.0/],

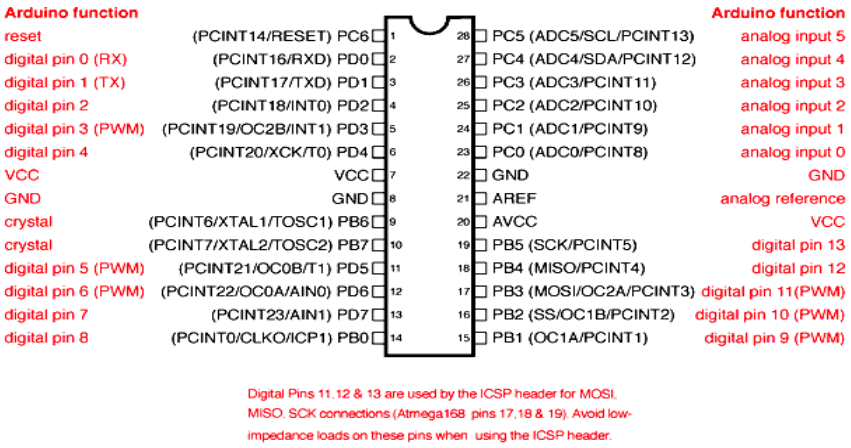
Atmel Studio 7 for Windows10 [https://www.microchip.com/avr-support/atmel-studio-7])

- ATMEGA328P programming can also be done in ARDUINO IDE.
- After writing the program, compile it to eliminate errors.
- Make the IDE generate HEX file for the written program after compiling.
- This HEX file contains the machine code which should be written in controller flash memory.
- Choose the programming device (usually SPI programmer made for AVR controllers) which establishes communication between PC and ATMEGA328P. You can also program ATMEGA328P using ARDUINO UNO board.
- Run the programmer software and choose the appropriate hex file.
- Burn the HEX file of written program in ATMEGA328P flash memory using this program.
- Disconnect the programmer, connect the appropriate peripherals for the controller and get the system started.

How to Use ATMega328P using Arduino

Since **ATmega328P** is used in Arduino Uno and Arduino nano boards, you can directly replace the arduino board with ATmega328 chip. For that first you need to install the **Arduino bootloader** into the chip (Or you can also buy a chip with bootloader – ATMega328P-PU). This IC with bootloader can be placed on Arduino Uno board and burn the program into it. Once Arduino program is burnt into the IC, it can be removed and used in place of Arduino board, along with a Crystal oscillator and other components as required for the project. Below is the **pin mapping between Arduino Uno and ATmega328P chip**.

ATMega328P and Arduino Uno Pin Mapping

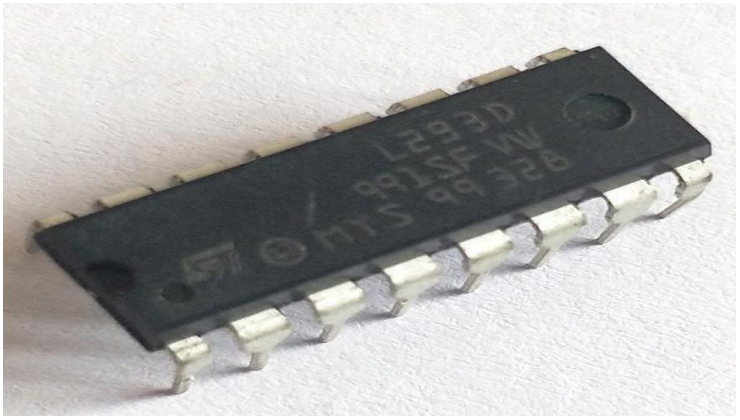


Applications

There are hundreds of applications for ATMEGA328P:

- Used in ARDUINO UNO, ARDUINO NANO and ARDUINO MICRO boards.
- Industrial control systems.
- SMPS and Power Regulation systems.
- Digital data processing.
- Analog signal measuring and manipulations.
- Embedded systems like coffee machine, vending machine.
- Motor control systems.
- Display units.
- Peripheral Interface system.

3.8L293D Motor Driver IC



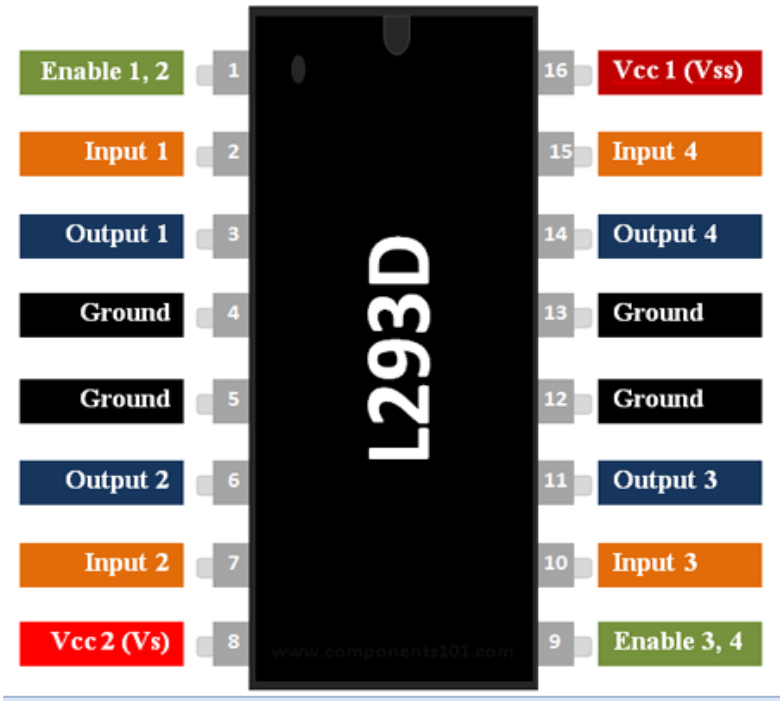


FIG 3.8 L293D Pin Configuration

Pin Number	Pin Name	Description
1	Enable 1,2	This pin enables the input pin Input 1(2) and Input 2(7)
2	Input 1	Directly controls the Output 1 pin. Controlled by digital circuits
3	Output 1	Connected to one end of Motor 1
4	Ground	Ground pins are connected to ground of circuit (0V)
5	Ground	Ground pins are connected to ground of circuit (0V)
6	Output 2	Connected to another end of Motor 1
7	Input 2	Directly controls the Output 2 pin. Controlled by digital circuits
8	Vcc2 (Vs)	Connected to Voltage pin for running motors (4.5V to 36V)

9	Enable 3,4	This pin enables the input pin Input 3(10) and Input 4(15)
10	Input 3	Directly controls the Output 3 pin. Controlled by digital circuits
11	Output 3	Connected to one end of Motor 2
12	Ground	Ground pins are connected to ground of circuit (0V)
13	Ground	Ground pins are connected to ground of circuit (0V)
14	Output 4	Connected to another end of Motor 2
15	Input 4	Directly controls the Output 4 pin. Controlled by digital circuits
16	Vcc2 (Vss)	Connected to +5V to enable IC function

Features

- Can be used to run Two DC motors with the same IC.
- Speed and Direction control is possible
- Motor voltage Vcc2 (Vs): 4.5V to 36V
- Maximum Peak motor current: 1.2A
- Maximum Continuous Motor Current: 600mA
- Supply Voltage to Vcc1(vss): 4.5V to 7V
- Transition time: 300ns (at 5Vand 24V)
- Automatic Thermal shutdown is available
- Available in 16-pin DIP, TSSOP, SOIC packages

3.8.1 L293D Equivalent Dual Timer IC

LB1909MC, SN754410, ULN2003

Where to use L293D IC

The L293D is a popular 16-Pin **Motor Driver IC**. As the name suggests it is mainly used to drive motors. A single **L293D IC** is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Micron rollers like Arduino, PIC, ARM etc.. this IC will be the right choice for you.

How to use a L293D Motor Driver IC

Using this L293D motor driver IC is very simple. The IC works on the principle of **Half H-Bridge**, let us not go too deep into what H-Bridge means, but for now just know that H bridge is a set up which is used to run motors both in clock wise and anti clockwise direction. As said earlier this IC is capable of running two motors at the any direction at the same time, the circuit to achieve the same is shown below.

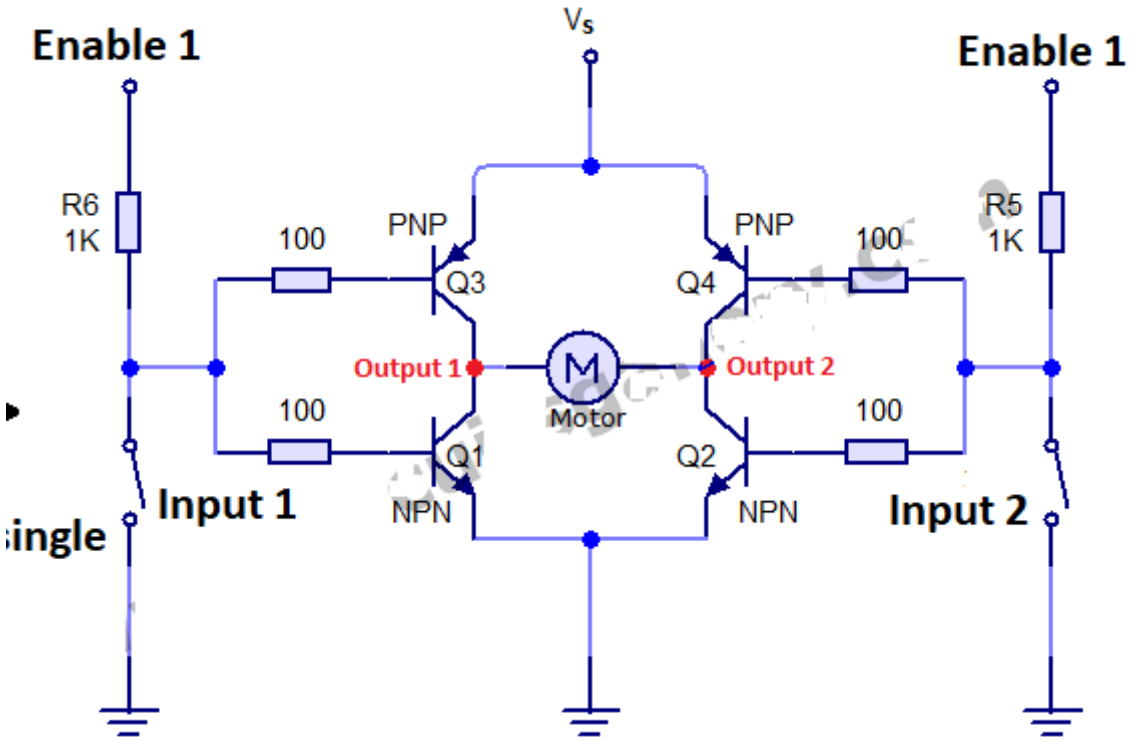


FIG 3.8.1 CIRCUIT DIAGRAM

All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss(Vcc1) which provides the voltage for the IC to work, this must be connected to +5V. The other is Vs(Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

The Enable pins (Enable 1,2 and Enable 3,4) are used to Enable Input pins for Motor 1 and Motor 2 respectively. Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. The input pins Input 1,2 are used to control the motor 1 and Input pins 3,4 are used to control the Motor 2. The input pins are connected to the any Digital circuit or microcontroller to control the speed and direction of the motor. You can toggle the input pins based on the following table to control your motor.

Input 1 = HIGH(5v)	Output 1 = HIGH	Motor 1 rotates in Clock wise Direction
Input 2 = LOW(0v)	Output 2 = LOW	
Input 3 = HIGH(5v)	Output 1 = HIGH	Motor 2 rotates in Clock wise Direction
Input 4 = LOW(0v)	Output 2 = LOW	

Input 1 = LOW(0v)	Output 1 = LOW	Motor 1 rotates in Anti-Clock wise Direction
Input 2 = HIGH(5v)	Output 2 = HIGH	
Input 3 = LOW(0v)	Output 1 = LOW	Motor 2 rotates in Anti -Clock wise Direction
Input 4 = HIGH(5v)	Output 2 = HIGH	

Input 1 = HIGH(5v)	Output 1 = HIGH	Motor 1 stays still
Input 2 = HIGH(5v)	Output 2 = HIGH	
Input 3 = HIGH(5v)	Output 1 = LOW	Motor 2 stays still
Input 4 = HIGH(5v)	Output 2 = HIGH	

Applications

- Used to drive high current Motors using Digital Circuits.
- Can be used to drive Stepper motors.
- High current LED's can be driven.
- Relay Driver module (Latching Relay is possible).

3.8.2 2D Model of L293D (PDIP)

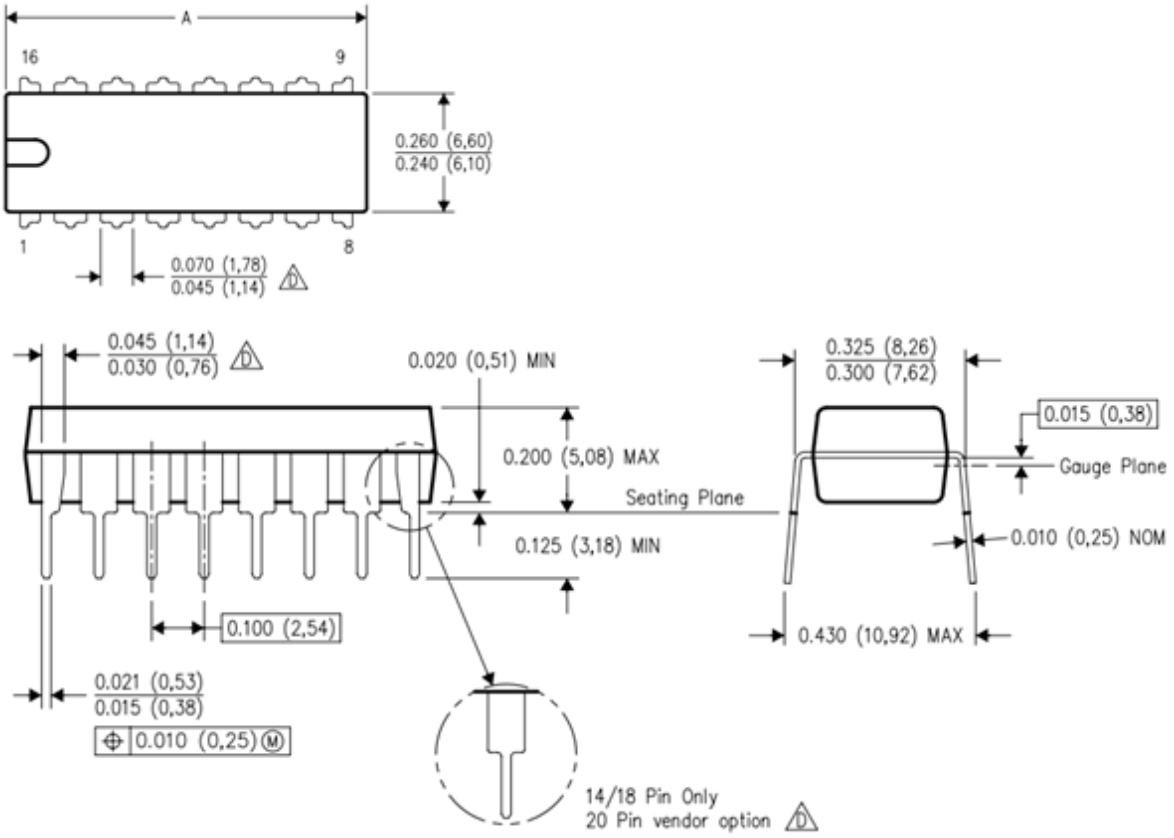


FIG 3.8.2 2D MODEL OF F293D

CHAPTER 4
LIGHT CRYSTAL DISPLAY

Introduction:

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

apes and S

ilable. Line lengths of 8, 16, 20, 24, 32 and 40 characters are all standard, in one, two

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

4.1 Liquid crystal display

Liquid crystal display is very important device in embedded system. It offers high flexibility to user as he can display the required data on it. But due to lack of proper approach to LCD interfacing many of them fail. Many people consider LCD interfacing a complex job but according to me LCD interfacing is very easy task, you just need to have a logical approach. This page is to help the enthusiast who wants to interface LCD with through understanding. Copy and Paste technique may not work when an embedded system engineer wants to apply LCD interfacing in real world projects.

Fig 4.1: LCD’S

4.2 FEATURES

- High contrast lcd supertwist display.
- ea dip162-dnled: yellow/green with led backlight.
- ea dip162-dn3lw and dip162j-dn3lw with white led b/l., low power.
- incl. Hd 44780 or compatible controller.
- interface for 4- and 8-bit data bus.
- power supply +5v or ±2.7v or ±3.3v.
- operating temperature 0~+50°c (-dn3lw, -dhnled: -20~+70°c).
- led backlight y/g max. 150ma@+25°c.
- led backlight white max. 45ma@+25°c.
- some more modules with same mechanic and same pinout.
- -dotmatrix 1x8, 4x20.
- -graphic 122x32.
- no screws required: solder on in pcb only.
- detachable via 9-pin socket ea b200-9 (2 pcs. Required).

You will be knowing about the booster rockets on space shuttle. Without these booster rockets the space shuttle would not launch in geosynchronous orbit. Similarly to understand LCD interfacing you need to have booster rockets attached! To get it done right you must have general idea how to approach any given LCD.This page will help you develop logical approach towards LCD interfacing.

First thing to begin with is to know what LCD driver/controller is used in LCD.Yes, your LCD is dumb it does not know to talk with your microcontroller. LCD driver is a link between the microcontroller and LCD. You can refer the datasheet of LCD to know the LCD driver for e.g. JHD 162A is name of LCD having driver HD44780U.You have to interface the LCD according to the driver specification. To understand the algorithm of LCD interfacing user must have datasheet of both LCD and LCD driver. Many people ignore the datasheets and end up in troubles. If you want to interface LCD successfully you must have datasheets. Why people ignore datasheets? Most of us do not like to read 100 pages of datasheet. But for a accurate technical specification datasheets are must. I will show you a technique to manipulate a datasheet within minutes.

First thing to find out in datasheet is the features viz. operating voltage, type of interface, maximum speed for interface in MHz, size of display data RAM, number of pixels, bits per pixel, number of row and columns. You must have the pin diagram of LCD.Pin diagram of LCD

Fig4.2: LCD alarm clock.

Each pixel of an LCD typically consists of a layer of molecules aligned between two transparent electrodes, and two polarizing filters, the axes of transmission of which are (in most of the cases) perpendicular to each other. With no actual liquid crystal between the polarizing filters, light passing through the first filter would be blocked by the second (crossed) polarizer.

The surfaces of the electrodes that are in contact with the liquid crystal material are treated so as to align the liquid crystal molecules in a particular direction. This treatment typically consists of a thin polymer layer that is unidirectional rubbed using, for example, a cloth. The direction of the liquid crystal alignment is then defined by the direction of rubbing. Electrodes are made of a transparent conductor called Indium Tin Oxide (ITO).

Before applying an electric field, the orientation of the liquid crystal molecules is determined by the alignment at the surfaces. In a twisted nematic device (still the most common liquid crystal device), the surface alignment directions at the two electrodes are perpendicular to each other, and so the molecules arrange themselves in a helical structure, or twist. This reduces the rotation of the polarization of the incident light, and the device appears grey. If the applied voltage is large enough, the liquid crystal molecules in the center of the layer are almost completely untwisted and the polarization of the incident light is not rotated as it passes through the liquid crystal layer. This light will then be mainly polarized perpendicular to the second filter, and thus be blocked and the pixel will appear black. By controlling the voltage applied across the liquid crystal layer in each pixel, light can be allowed to pass through in varying amounts thus constituting different levels of gray.

The optical effect of a twisted nematic device in the voltage-on state is far less dependent on variations in the device thickness than that in the voltage-off state. Because of this, these devices are usually operated between crossed polarizer's such that they appear bright with no voltage (the eye is much more sensitive to variations in the dark state than the bright state). These devices can also be operated between parallel polarizer's, in which case the bright and dark states are reversed. The voltage-off dark state in this configuration appears blotchy, however, because of small variations of thickness across the device.

Both the liquid crystal material and the alignment layer material contain ionic compounds. If an electric field of one particular polarity is applied for a long period of time, this ionic material is attracted to the surfaces and degrades the device performance. This is avoided either by applying an alternating current or by reversing the polarity of the electric field as the device is addressed (the response of the liquid crystal layer is identical, regardless of the polarity of the applied field).

When a large number of pixels are needed in a 2display, it is not technically possible to drive each directly since then each pixel would require independent electrodes. Instead, the display is multiplexed. In a multiplexed display, electrodes on one side of the display are grouped and wired together (typically in columns), and each group gets its own voltage source. On the other side, the electrodes are also grouped (typically in rows), with each group getting a voltage sink. The groups are designed so each pixel has a unique, unshared

combination of source and sink. The electronics or the software driving the electronics then turns on sinks in sequence, and drives sources for the pixels of each sink.

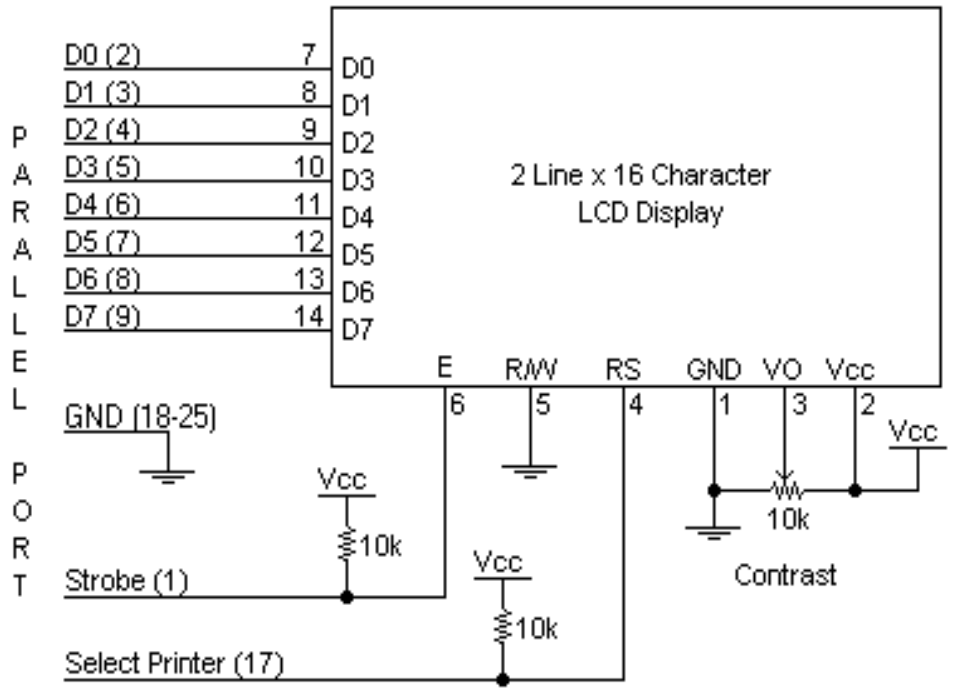


Fig4.2.1 2x16 LCD module

Above is the quite simple schematic. The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there are a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors. We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to

5v or use a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

CHAPTER 5

GSM

1. Overview

What is GSM? If you are in Europe or Asia and using a mobile phone, then most probably you are using GSM technology in your mobile phone.

- GSM stands for Global System for Mobile Communication. It is a digital cellular technology used for transmitting mobile voice and data services.
- The concept of GSM emerged from a cell-based mobile radio system at Bell Laboratories in the early 1970s. GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard.
- GSM is the most widely accepted standard in telecommunications and it is implemented globally.
- GSM is a circuit-switched system that divides each 200 kHz channel into eight 25 kHz time-slots. GSM operates on the mobile communication bands 900 MHz and 1800 MHz in most parts of the world. In the US, GSM operates in the bands 850 MHz and 1900 MHz
- .GSM owns a market share of more than 70 percent of the world's digital cellular subscribers
- . GSM makes use of narrowband Time Division Multiple Access (TDMA) technique for transmitting signals.
- GSM was developed using digital technology. It has an ability to carry 64 kbps to 120 Mbps of data rates.
- Presently GSM supports more than one billion mobile subscribers in more than 210 countries throughout the world.
- GSM provides basic to advanced voice and data services including roaming service. Roaming is the ability to use your GSM phone number in another GSM network. GSM digitizes and compresses data, then sends it down through a channel with two other streams of user data, each in its own timeslot.

2. Why GSM?

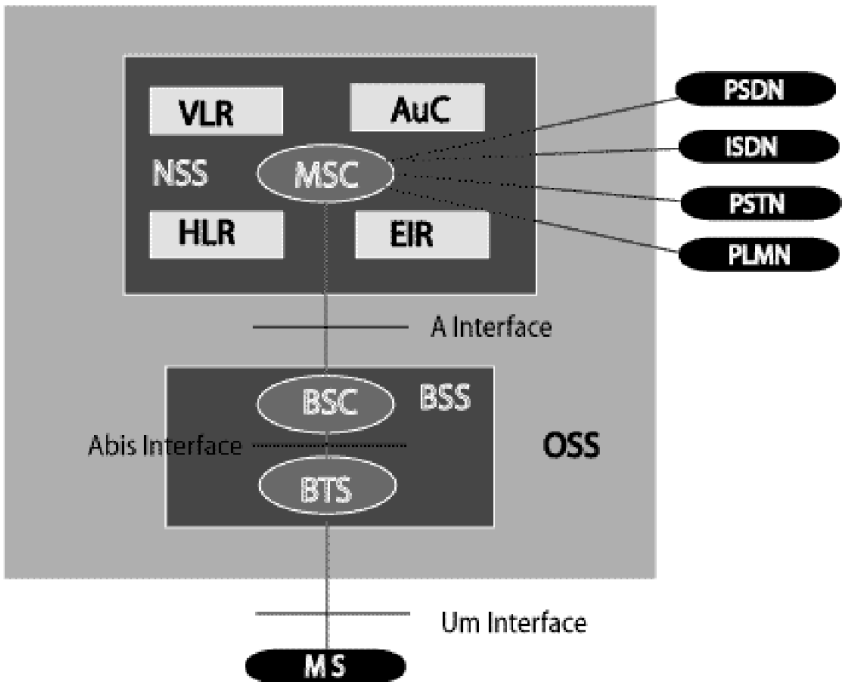
Listed below are the features of GSM that account for its popularity and wide acceptance

- Improved spectrum efficiency.
- International roaming.
- Low-cost mobile sets and base stations (BSs).
- High-quality speech.
- Compatibility with Integrated Services Digital Network (ISDN) and other telephone company services.
- Support for new services.

5.1 GSM Architecture

A GSM network comprises of many functional units. These functions and interfaces are explained in this chapter. The GSM network can be broadly divided into:

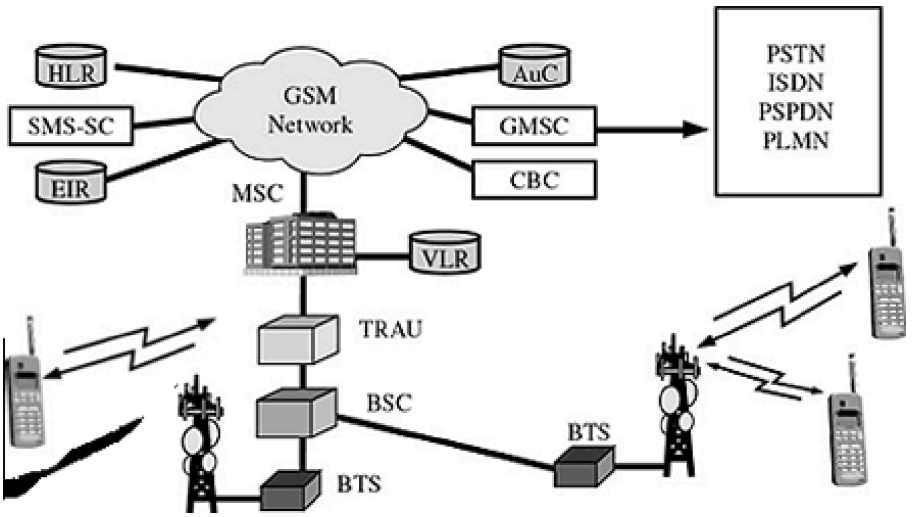
- The Mobile Station (MS)
- The Base Station Subsystem (BSS)
- The Network Switching Subsystem (NSS)
- The Operation Support Subsystem (OSS) Given below is a simple pictorial view of the GSM architecture.



5.1 GSM ARCHITECTURE

The additional components of the GSM architecture comprise of databases and messaging systems' functions:

- Home Location Register (HLR) , Visitor Location Register (VLR)
- Equipment Identity Register (EIR)
- Authentication Center (AUC)
- SMS Serving Center (SMS SC)
- Gateway MSC (GMSC)
- Chargeback Center (CBC)
- Transcoder and Adaptation Unit (TRAU) The following diagram shows the GSM network along with the added elements.



The MS and the BSS communicate across the Um interface. It is also known as the air interface or the radio link. The BSS communicates with the Network Service Switching (NSS) center across the A interface.

The additional components of the GSM architecture comprise of databases and messaging systems' functions:

- Home Location Register (HLR)
- Visitor Location Register (VLR)
- Equipment Identity Register (EIR)
- Authentication Center (AuC)
- SMS Serving Center (SMS SC)
- Gateway MSC (GMSC)
- Chargeback Center (CBC)
- Transcoder and Adaptation Unit (TRAU)

GSM Network Areas

GSM Network Areas In a GSM network, the following areas are defined:

1) Cell: Cell is the basic service area; one BTS covers one cell. Each cell is given a Cell Global Identity (CGI), a number that uniquely identifies the cell

2) Location Area: A group of cells form a Location Area (LA). This is the area that is paged when a subscriber gets an incoming call. Each LA is assigned a Location Area Identity (LAI). Each LA is served by one or more BSCs.

3) MSC/VLR Service Area: The area covered by one MSC is called the MSC/VLR service area

4) PLMN: The area covered by one network operator is called the Public Land Mobile Network (PLMN). A PLMN can contain one or more MSCs.

5.2 GSM SPECIFICATIONS

The requirements for different Personal Communication Services (PCS) systems differ for each PCS network. Vital characteristics of the GSM specification are listed below.

Modulation:

Modulation is the process of transforming the input data into a suitable format for the transmission medium. The transmitted data is demodulated back to its original form at the receiving end. The GSM uses Gaussian Minimum Shift Keying (GMSK) modulation method.

Access Methods:

Radio spectrum being a limited resource that is consumed and divided among all the users, GSM devised a combination of TDMA/FDMA as the method to divide the bandwith among the users. In this process, the FDMA part divides the frequency of the total 25 MHz bandwidth into 124 carrier frequencies of 200 kHz bandwidth.

Each BS is assigned with one or multiple frequencies, and each of this frequency is divided into eight timeslots using a TDMA scheme. Each of these slots are used for both transmission as well as reception of data. These slots are separated by time so that a mobile unit doesn't transmit and receive data at the same time.

Transmission Rate: The total symbol rate for GSM at 1 bit per symbol in GMSK produces 270.833 K symbols/second. The gross transmission rate of a timeslot is 22.8 Kbps. GSM is a digital system with an over-the-air bit rate of 270 kbps.

Frequency Band: The uplink frequency range specified for GSM is 933–960 MHz (basic 900 MHz band only). The downlink frequency band is 890–915 MHz (basic 900 MHz band only).

Channel Spacing:

Channel spacing indicates the spacing between adjacent carrier frequencies. For GSM, it is 200 kHz.

Speech Coding:

For speech coding or processing, GSM uses Linear Predictive Coding (LPC). This tool compresses the bit rate and gives an estimate of the speech parameters. When the audio signal passes through a filter, it mimics the vocal tract. Here, the speech is encoded at 13 kbps.

Duplex Distance

Duplex distance is the space between the uplink and downlink frequencies. The duplex distance for GSM is 80 MHz, where each channel has two frequencies that are 80 MHz apart.

Miscellaneous

- Frame duration: 4.615 Ms
- Duplex Technique: Frequency Division Duplexing (FDD) access mode previously known as WCDMA.
- Speech channels per RF channel: 8.

5.3 GSM Addressing and Identifiers

GSM treats the users and the equipment in different ways. Phone numbers, subscribers, and equipment identifiers are some of the known ones. There are many other identifiers that have been well-defined, which are required for the subscriber's mobility management and for addressing the remaining network elements. Vital addresses and identifiers that are used in GSM are addressed below.

International Mobile Station Equipment Identity

The International Mobile Station Equipment Identity (IMEI) looks more like a serial number which distinctively identifies a mobile station internationally. This is allocated by the equipment manufacturer and registered by the network operator, who stores it in the Entrepreneurs-in-Residence (EIR). By means of IMEI, one recognizes obsolete, stolen, or non-functional equipment.

Following are the parts of IMEI:

- Type Approval Code (TAC): 6 decimal places, centrally assigned
- . Final Assembly Code (FAC): 6 decimal places, assigned by the manufacturer
- . Serial Number (SNR): 6 decimal places, assigned by the manufacturer.
- Spare (SP): 1 decimal place. Thus, IMEI = TAC + FAC + SNR + SP. It uniquely characterizes a mobile station and gives clues about the manufacturer and the date of manufacturing.

International Mobile Subscriber Identity

Every registered user has an original International Mobile Subscribter Identity (IMSI) with a valid IMEI stored in their Subscriber Identity Module (SIM).

IMSI comprises of the following parts:

- Mobile Country Code (MCC): 3 decimal places, internationally standardized.
- Mobile Network Code (MNC): 2 decimal places, for unique identification of a mobile network within the country.
- Mobile Subscriber Identification Number (MSIN): Maximum 10 decimal places, identification number of the subscriber in the home mobile network.

Mobile Subscriber ISDN Number

The authentic telephone number of a mobile station is the Mobile Subscriber ISDN Number (MSISDN). Based on the SIM, a mobile station can have many MSISDNs, as each subscriber is assigned with a separate MSISDN to their SIM respectively.

Listed below is the structure followed by MSISDN categories, as they are defined based on international ISDN number plan:

- Country Code (CC) : Up to 3 decimal places.
- National Destination Code (NDC): Typically 2–3 decimal places.
- Subscriber Number (SN): Maximum 10 decimal places.

Mobile Station Roaming Number

Mobile Station Roaming Number (MSRN) is an interim location dependent ISDN number, assigned to a mobile station by a regionally responsible Visitor Location Register (VLA). Using MSRN, the incoming calls are channeled to the MS.

The MSRN has the same structure as the MSISDN.

- Country Code (CC) : of the visited network.
- National Destination Code (NDC): of the visited network. 🎬 Subscriber Number (SN): in the current mobile network.

Location Area Identity

Within a PLMN, a Location Area identifies its own authentic Location Area Identity (LAI). The LAI hierarchy is based on international standard and structured in a unique format as mentioned below:

- Country Code (CC): 3 decimal places.
- Mobile Network Code (MNC): 2 decimal places.
- Location Area Code (LAC): maximum 5 decimal places or maximum twice 8 bits coded in hexadecimal (LAC < FFFF).

Cell Identifier Using a Cell Identifier (CI) (maximum 2 × 8) bits, the individual cells that are within an LA can be recognized. When the Global Cell Identity (LAI + CI) calls are combined, then it is uniquely defined.

GSM Operations

Once a Mobile Station initiates a call, a series of events takes place. Analyzing these events can give an insight into the operation of the GSM system.

Mobile Phone to Public Switched Telephone Network (PSTN)

When a mobile subscriber makes a call to a PSTN telephone subscriber, the following sequence of events takes place:

1. The MSC/VLR receives the message of a call request.

2. The MSC/VLR checks if the mobile station is authorized to access the network. If so, the mobile station is activated. If the mobile station is not authorized, then the service will be denied.

3. MSC/VLR analyzes the number and initiates a call setup with the PSTN.

4. MSC/VLR asks the corresponding BSC to allocate a traffic channel (a radio channel and a timeslot).

5. The BSC allocates the traffic channel and passes the information to the mobile station.

6. The called party answers the call and the conversation takes place.

7. The mobile station keeps on taking measurements of the radio channels in the present cell and the neighboring cells and passes the information to the BSC. The BSC decides if a handover is required. If so, a new traffic channel is allocated to the mobile station and the handover takes place. If handover is not required, the mobile station continues to transmit in the same frequency.

PSTN to Mobile Phone

When a PSTN subscriber calls a mobile station, the following sequence of events takes place:

1. The Gateway MSC receives the call and queries the HLR for the information needed to route the call to the serving MSC/VLR.

2. The GMSC routes the call to the MSC/VLR.

3. The MSC checks the VLR for the location area of the MS.

4. The MSC contacts the MS via the BSC through a broadcast message, that is, through a paging request. 5. The MS responds to the page request

6. The BSC allocates a traffic channel and sends a message to the MS to tune to the channel. The MS generates a ringing signal and, after the subscriber answers, the speech connection is established.

7. Handover, if required, takes place, as discussed in the earlier case.

To transmit the speech over the radio channel in the stipulated time, the MS codes it at the rate of 13 Kbps. The BSC transcodes the speech to 64 Kbps and sends it over a land link or a radio link to the MSC. The MSC then forwards the speech data to the PSTN. In the reverse direction, the speech is received at 64 Kbps at the BSC and the BSC transcodes it to 13 Kbps for radio transmission.

GSM supports 9.6 Kbps data that can be channeled in one TDMA timeslot. To supply higher data rates, many enhancements were done to the GSM standards (GSM Phase 2 and GSM Phase 2+).

Explanation of commonly used AT commands:

1) **AT -** This command is used to check communication between the module and the computer.
For example,
AT
OK
The command returns a result code OK if the computer (serial port) and module are connected properly. If any of module or SIM is not working, it would return a result code ERROR.

2) +CMGF - This command is used to set the SMS mode. Either text or PDU mode can be selected by assigning 1 or 0 in the command.
SYNTAX: AT+CMGF=<mode>
0: for PDU mode
1: for text mode

The text mode of SMS is easier to operate but it allows limited features of SMS. The PDU (protocol data unit) allows more access to SMS services but the operator requires bit level knowledge of TPDUs. The headers and body of SMS are accessed in hex format in PDU mode so it allows availing more features.

For example,
AT+CMGF=1
OK

3) +CMGW - This command is used to store message in the SIM.
SYNTAX: AT+CMGW=" Phone number"> *Message to be stored* Ctrl+z
As one types AT+CMGW and phone number, '>' sign appears on next line where one can type the message. Multiple line messages can be typed in this case. This is why the message is terminated by providing a 'Ctrl+z' combination. As Ctrl+z is pressed, the following information response is displayed on the screen.
+CMGW: Number on which message has been stored

4) +CMGS - This command is used to send a SMS message to a phone number.
SYNTAX: AT+CMGS= serial number of message to be send.

As the command AT+CMGS and serial number of message are entered, SMS is sent to the particular SIM.

For example,

AT+CMGS=1

OK

5) ATD - This command is used to dial or call a number.

SYNTAX: ATD<Phone number>;(Enter)

For example,

ATD123456789;

6) ATA - This command is used to answer a call. An incoming call is indicated by a message 'RING' which is repeated for every ring of the call. When the call ends 'NO CARRIER' is displayed on the screen.

SYNTAX: ATA(Enter)

As ATA followed by enter key is pressed, incoming call is answered.

For example,

RING

RING

ATA

7) **ATH** - This command is used to disconnect remote user link with the GSM module.

SYNTAX: ATH (Enter)

List of AT commands:

The AT commands for both, GSM module and the mobile phone, are listed below. Some of these commands may not be supported by all the GSM modules available. Also there might be some commands which won't be supported by some mobile handsets.

Testing:

Command	Description
AT	Checking communication between the module and computer.

Call control:

Command	Description
ATA	Answer command
ATD	Dial command
ATH	Hang up call
ATL	Monitor speaker loudness
ATM	Monitor speaker mode
ATO	Go on-line
ATP	Set pulse dial as default
ATT	Set tone dial as default

AT+CSTA	Select type of address
AT+CRC	Cellular result codes

Data card Control:

Command	Description
ATI	Identification
ATS	Select an S-register
ATZ	Recall stored profile
AT&F	Restore factory settings
AT&V	View active configuration
AT&W	Store parameters in given profile
AT&Y	Select Set as power up option
AT+CLCK	Facility lock command
AT+COLP	Connected line identification presentation
AT+GCAP	Request complete capabilities list
AT+GMI	Request manufacturer identification
AT+GMM	Request model identification
AT+GMR	Request revision identification
AT+GSN	Request product serial number identification (IMEI)

Phone control:

Command	Description
AT+CBC	Battery charge
AT+CGMI	Request manufacturer identification
AT+CGMM	Request model identification
AT+CGMR	Request revision identification
AT+CGSN	Request product serial number identification
AT+CMEE	Report mobile equipment error
AT+CPAS	Phone activity status
AT+CPBF	Find phone book entries
AT+CPBR	Read phone book entry
AT+CPBS	Select phone book memory storage
AT+CPBW	Write phone book entry
AT+CSCS	Select TE character set
AT+CSQ	Signal quality

Computer data interface:

Command	Description
ATE	Command Echo
ATQ	Result code suppression
ATV	Define response format
ATX	Response range selection
AT&C	Define DCD usage
AT&D	Define DTR usage
AT&K	Select flow control
AT&Q	Define communications mode option
AT&S	Define DSR option
AT+ICF	DTE-DCE character framing
AT+IFC	DTE-DCE Local flow control
AT+IPR	Fixed DTE rate

Service:

Command	Description
AT+CLIP	Calling line identification presentation
AT+CR	Service reporting control
AT+DR	Data compression reporting
AT+ILRR	DTE-DCE local rate reporting

Network Communication parameter:

Command	Description
ATB	Communications standard option
AT+CBST	Select bearer service type
AT+CEER	Extended error report
AT+CRLP	Radio link protocol
AT+DS	Data compression

Miscellaneous:

Command	Description
A/	Re-execute command line
AT?	Command help
AT*C	Start SMS interpreter
AT*T	Enter SMS block mode protocol
AT*V	Activate V.25bis mode
AT*NOKIATEST	Test command
AT+CESP	Enter SMS block mode protocol

SMS Text mode:

Command	Description
AT+CSMS	Select message service
AT+CPMS	Preferred message storage
AT+CMGF	Message format
AT+CSCA	Service centre address
AT+CSMP	Set text mode parameters
AT+CSDH	Show text mode parameters
AT+CSCB	Select cell broadcast message types
AT+CSAS	Save settings
AT+CRES	Restore settings
AT+CNMI	New message indications to TE
AT+CMGL	List messages
AT+CMGR	Read message
AT+CMGS	Send message
AT+CMSS	Send message from storage
AT+CMGW	Write message to memory
AT+CMGD	Delete message

SMS PDU mode:

Command	Description
AT+CMGL	List Messages
AT+CMGR	Read message
AT+CMGS	Send message
AT+CMGW	Write message to memory

9. SIM800L

SIM800L is a miniature cellular module which allows for GPRS transmission, sending and receiving SMS and making and receiving voice calls. Low cost and small footprint and quad band frequency support make this module perfect solution for any project that require long range connectivity. After connecting power module boots up, searches for

cellular network and login automatically. On board LED displays connection state (no network coverage - fast blinking, logged in - slow blinking).

NOTICE: Be prepared to handle huge power consumption with peek up to 2A. Maximum voltage on UART in this module is 2.8V. Higher voltage will kill the module.

This module have two antennas included. First is made of wire (which solders directly to NET pin on PCB) - very useful in narrow places. Second - PCB antenna - with double sided tape and attached pigtail cable with IPX connector. This one have better performance and allows to put your module inside a metal case - as long the antenna is outside.

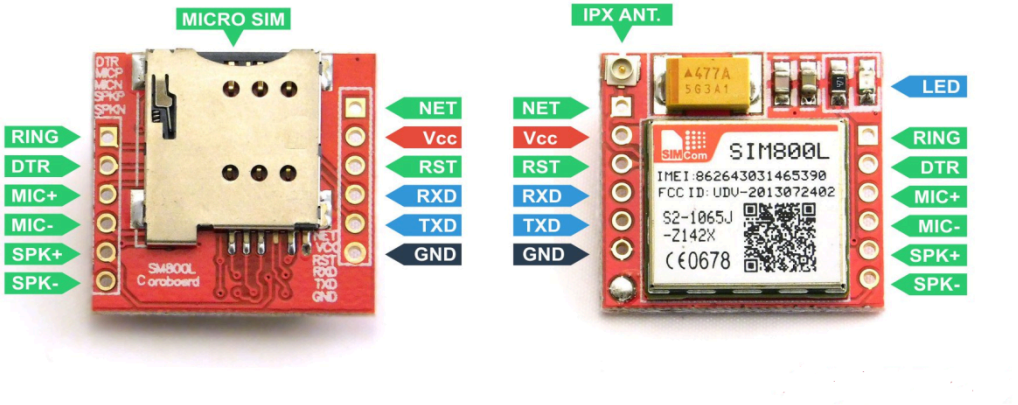
Specification

- Supply voltage: 3.8V - 4.2V
- Recommended supply voltage: 4V
- Power consumption:
    - sleep mode < 2.0mA
    - idle mode < 7.0mA
    - GSM transmission (avg): 350 mA
    - GSM transmission (peek): 2000mA
- Module size: 25 x 23 mm
- Interface: UART (max. 2.8V) and AT commands
- SIM card socket: microSIM (bottom side)
- Supported frequencies: Quad Band (850 / 950 / 1800 /1900 MHz)
- Antenna connector: IPX
- Status signaling: LED
- Working temperature range: -40 do + 85 ° C

Set includes:

- SIM800L module.
- Gold pin headers.
- Wire antenna.
- PCB antenna with pigtail and IPX connector.

Module pinout



Pin out (bottom side - left):

- RING (not marked on PBC, first from top, square) - LOW state while receiving call
- DTR - sleep mode. Default in HIGH state (module in sleep mode, serial communication disabled). After setting it in LOW the module will wake up.
- MICP, MICN - microphone (P + / N -)
- SPKP, SPKN - speaker (P + / N -)

Pin out (bottom side - right):

- NET - antenna
- VCC - supply voltage
- RESET - reset
- RXD - serial communication
- TXD - serial communication
- GND - ground

CHAPTER 6

TEMPERATURE SENSOR

Introduction:

Temperature sensors are vital to a variety of everyday products. For example, household ovens, refrigerators, and thermostats all rely on temperature maintenance and control in order to function properly. Temperature control also has applications in chemical engineering. Examples of this include maintaining the temperature of a chemical reactor at the ideal set-point, monitoring the temperature of a possible runaway reaction to ensure the safety of employees, and maintaining the temperature of streams released to the environment to minimize harmful environmental impact.

While temperature is generally sensed by humans as “hot”, “neutral”, or “cold”, chemical engineering requires precise, quantitative measurements of temperature in order to accurately control a process. This is achieved through the use of temperature sensors, and temperature regulators which process the signals they receive from sensors.

From a thermodynamics perspective, temperature changes as a function of the average energy of molecular movement. As heat is added to a system, molecular motion increases and the system experiences an increase in temperature. It is difficult, however, to directly measure the energy of molecular movement, so temperature sensors are generally designed to measure a property which changes in response to temperature. The devices are then calibrated to traditional temperature scales using a standard (i.e. the boiling point of water at known pressure). The following sections discuss the various types of sensors and regulators.

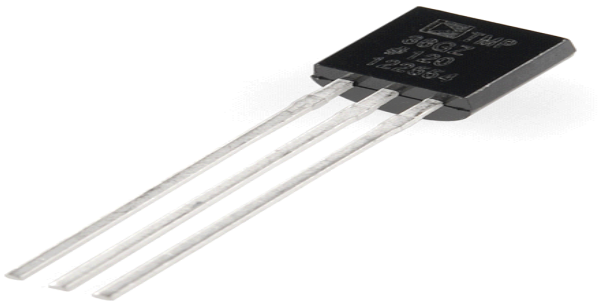


Fig:6 Temp sensor

The most commonly used type of all the sensors are those which detect Temperature or heat. These types of temperature sensor vary from simple ON/OFF thermostatic devices which control a domestic hot water heating system to highly sensitive semiconductor types that can control complex process control furnace plants.

Temperature Sensors:

Temperature sensors are devices used to measure the temperature of a medium. There are 2 kinds on temperature sensors:

1) Contact sensors and

2) Noncontact sensors.

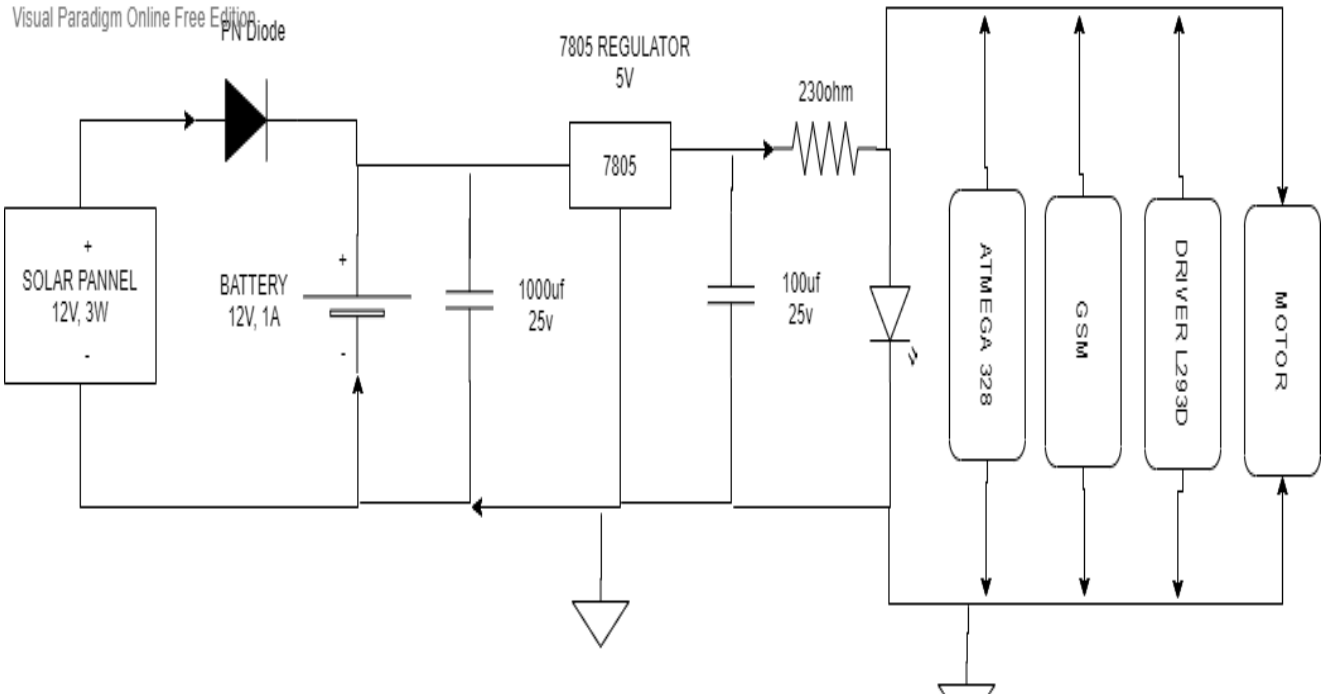
However, the 3 main types are thermometers, resistance temperature detectors, and thermocou ples. All three of these sensors measure a physical property (i.e. volume of a liquid, current through a wire), which changes as a function of temperature. In addition to the 3 main types of temperature sensors, there are numerous other temperature sensors available for use.

CHAPTER 7

RESULTS

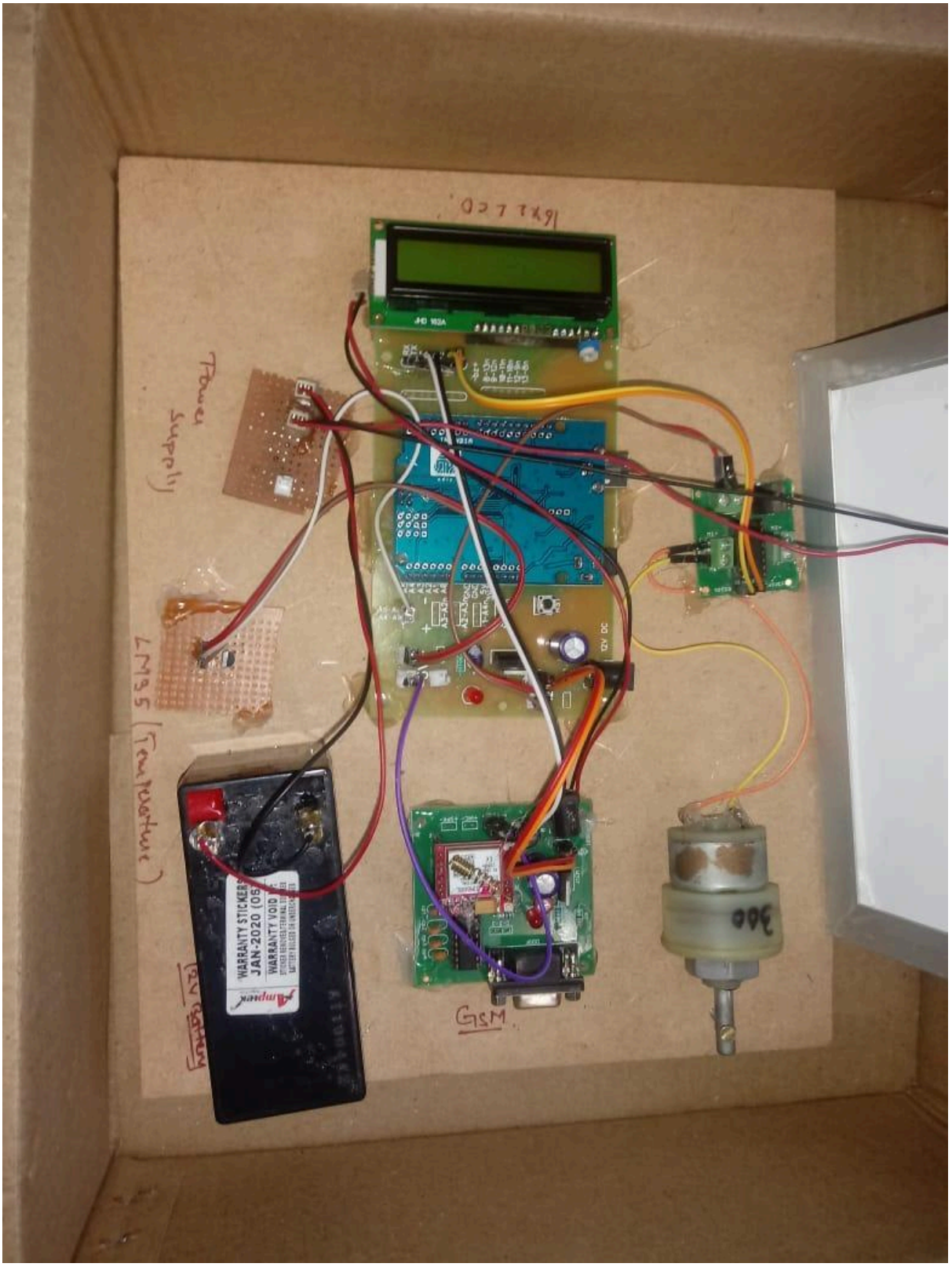
CIRCUIT DIAGRAM AND OUTPUT

7.1 CIRCUIT DIAGRAM



7.1 CIRCUIT DIAGRAM

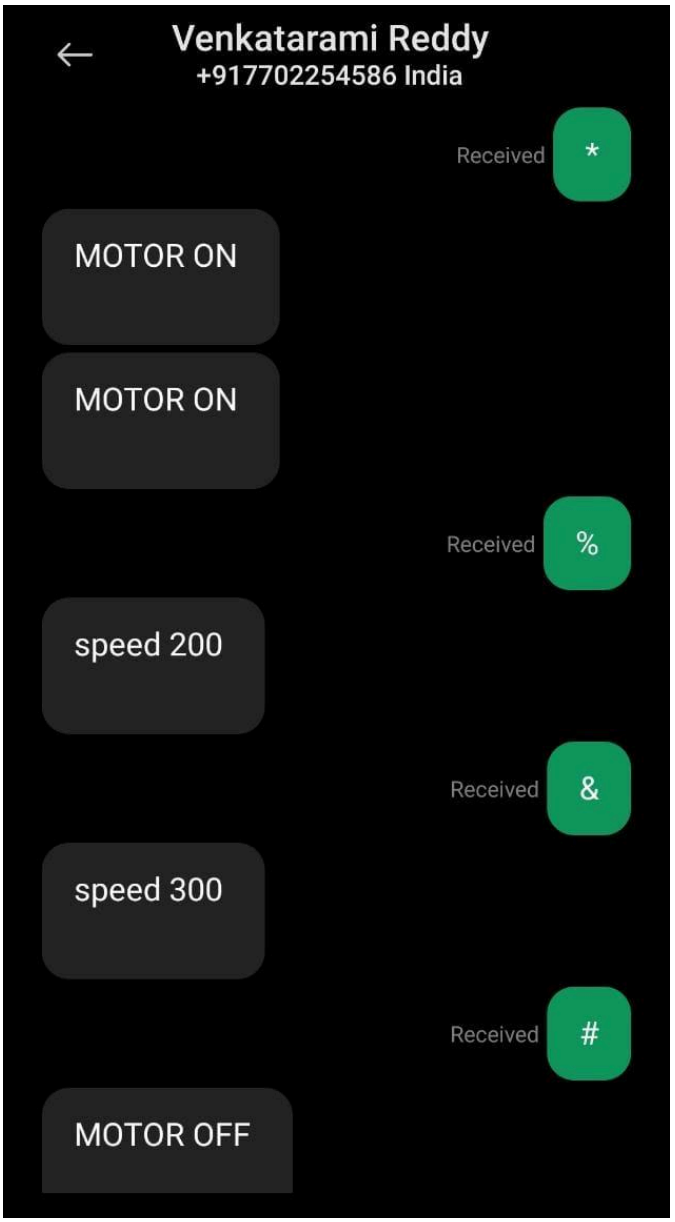
7.2 OUTPUT


Power supply
LM35 (Temperature)
GSM
WARRANTY STICKER
JAN-2020
WARRANTY VOID

7.2 OUTPUT

7.3 Video Link

https://drive.google.com/file/d/1_hdeYvzatwJJMxTvSZTUUM6lubNCzsfy/view?usp=sharing



CHAPTER 8

CONCLUSION AND FUTURE SCOPE

Conclusion: Agriculture:

This project can be widely used in agriculture:

- ✔ Checking the presence of three phase current in the phone is a very time saving approach.
- ✔ This project can also be used to irrigate small and large farms.
- ✔ The irrigation system of the farm can be controlled from the farmer's figure tips from anywhere.
- ✔ This project helps farmers to not only protect the motor but also saves power by turning it off anytime.
- ✔ Rain sensor detects the rain and accordingly the motor can be turned on or turned off from any convenient place in time.

Green House / Nursery:

- ✔ Many companies are doing Green House Projects and they can check the Water management through our Mobile application.
- ✔ Since this project is cost efficient it can be implemented in small scale and individual green house projects is benefitted.
- ✔ By the help of this project the irrigation system can be taken care of even from the person's office.
- ✔ Nursery water management can be implemented through our Mobile based water management system.

Water supply system:

- ✔ The motor is used to supply water to houses.
- ✔ This is also used to supply drinking water to houses in the rural areas.

Future scope:

India is heading towards digital India, modernization in agriculture is important. Farmer required to utilize every drop of water. Our project is suitable for any size of farm fields or any type of crop. This project automates the irrigation system like normal farm, green house, nursery etc., Making use of this Mobile Application farmer can protect his motor and thereby avoiding future damages and losses. This is a very cost effective project which is easily affordable by the Farmers, Engineering Industries, Water Departments and Hazardous Industries.

CHAPTER 9

REFERENCES

- ✔ “Single Power-Conversion AC--DC Converter with High Power Factor and High Efficiency”, Yong-Won Cho, JungMin Kwon, Bong-Hwan Kwon
- ✔ “Smart automation system using arduino and rain drop sensor”, Prof. Abhijit G Kalbande
- ✔ “Control of Three Phase Induction Motor by Ardiuno with IOT”, Anbuganapathy S. 1 Sivanantham N.
- ✔ “Literature Review on Home Automation System”, Neha Malik, Yogita Bodwade.
- ✔ S.Funabik.(1990). “A single-phase PWM controlled AC to DC converter based on control of unity displacement powers factor”, IEEE-IAS Conf. Rec., pp. 1012-1016.
- ✔ R. Itoh, Eng, K. Ishizaka,(1991). “Three-phase fly back ACDC converter with sinusoidal supply currents”, Proc. Inst. Elec. Eng. B. Elect. Power Appl., vol. 138, pp. 143- 151, .
- ✔ Lokesh Heda, Pritesh Bhutada, Rinkesh Thakur, Piyush Bhattad, Vinita Singh.(2016) “Fault Monitoring and Protection of Three Phase Devices”, International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, Vol. 4, Iss. 4, 208-210.
- ✔ S.Sumeetha, D.Sharmila.(2012) “Embedded based Remote Control Application using Mobile Phone in Irrigation”, International Journal of Power Control Signal and Computation, Vol. 3, Iss. 1, 76-79.
- ✔ V. Bhaskar, T. Gowri Manohar. (2011). “GSM Based Motor Monitoring and Speed Control”, International Journal of Mechanical and Industrial Engineering, Vol. 1, Iss. 2, 87-92.
- ✔ Prof.R.R.Jadhav, Prathmesh P Pandit, Shubham D.Pal, Vineet H.Risbud.(2015). “Three Phase Motor Control Using GSM”, International journal of innovative research in electrical, Electronics, instrumentation and control engineering volume 3, Issue 5.

- ✔ Wani SurajR, Ghaywat VivekV, Naik AkshayD, Mandlik Sachin.B(2015). “IVR System for three phase motor protection, Control and Alert system using GSM”, international journal of engineering and computer science volume 4 Issue 2 Feb.
- ✔ Ransing Rasika, Manita Rajput (2015) “Smart home for elderly care based on wireless sensor network”, Nascent Technologies in the Engineering Field (ICNTE) 2019.