

SMS Based Pump on/off control and notification

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in
Electronics and Communication Engineering

by

Gubbala Pradeep 38130074
Batthala Arun Kumar Reddy 38130027



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL – 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with “A” Grade by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Gubbala Pradeep (38130074), Batthala Arun Kumar Reddy (38130027) who carried out the project entitled “**SMS Based Pump on/off control and notification**” under our supervision from November 2021 to March 2022.

Internal Guide

Dr.T.BERNATIN.,M.E.,Ph.D.,

Head of the Department

Dr. T.RAVI.,M.E.,Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

Dr.S.EMALDA ROSLIN, M.E, Ph.D

External Examiner

Dr.N.R.Krishnamoorthy, M.E, Ph.D

DECLARATION

We **Gubbala Pradeep(RegNo.38130074)** and **Batthala Arun Kumar Reddy (RegNo.38130027)** hereby declare that the Project Report entitled " **SMS Based Pump on/off control and notification**" done by us under the guidance of **Dr.S.Emalda Roslin, M.E, Ph.D** and **Dr. N.R.Krishnamoorthy, M.E, Ph.D** is submitted in complete fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. **N.M. NANDITHA, M.E., Ph.D.**, Dean, School of **ELECTRONICS AND COMMUNICATION ENGINEERING** and Dr. **T. RAVI, M.E., Ph.D.**, Head of the Department, Dept. of **ELECTRONICS AND COMMUNICATION ENGINEERING** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Mrs. **Dr.T.BERNATIN.,M.E.,Ph.D.**, **Assistant Professor** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **ELECTRONICS AND COMMUNICATION ENGINEERING** who were helpful in many ways for the completion of the project.

ABSTRACT

This project aims to facilitate the owner to control water pump machines in the Agriculture field. The current control system still has many limitations in controlling the water pump machine in the Agriculture field as only It can be controlled only by sending SMS. The water machine control system was designed with a device that can control missed calls and get notified status through SMS, this device create the development of Arduino UNO microcontroller technology that allows the reading of the GSM Module and connected with SMS Gateway, apart from the use of the controls using smartphones that we need to type the message to control, the SMS Gateway as a solution also has remote access as the information will be sent automatically to the owner of the water machine that the water machine status on/off, if it is shut down manually then it will send on together with Arduino UNO microcontroller. The Arduino UNO microcontroller is used to turn on and off the water machine by a smartphone using missed calls as wireless communication to Arduino UNO controller and get notified the status by sms Experiment results facilitate real-time water control and much easy for the owner to control the motors in the Agriculture field.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	vii
	LIST OF TABLES	viii
1	INTRODUCTION	1
	1.1 OVERVIEW	1
2	LITERATURE SURVEY	3
3	AIM AND SCOPE OF THE PROJECT	5
	3.1 AIM AND OBJECTIVE	5
	3.2 PROPOSED SYSTEM	6
	3.3 WORKING	6
4	METHODOLGY	9
	4.1 INTRODUCTION	9
	4.2 HARDWARE REQUIRED	9
	4.2.1 Arduino UNOR3	9
	4.2.2 GSM SIM 800L-MODULE	18
	4.2.3 LCD DISPLAY	28
	4.2.4 SINGLE CHANNEL RELAY	33
	4.2.5 POWER SUPPLY	35
	4.3 HARDWARE IMPLEMENTATION	37
	4.4 SOFTWARE IMPLEMENTATION	39
5	RESULTS AND DISCUSSION	41
	5.1 RESULTS	41
	5.2 DISCUSSION	42
6	CONCLUSION AND FUTURE SCOPE	43
	6.1 CONCLUSION	43
	6.2 FUTURE SCOPE	43

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	Block diagram of prototype	5
3.2	Pump on off control system	6
3.3	Work flow chart	7
3.4	Simulation diagram	8
4.1	Arduino UNOR3	9
4.2	Arduino UNOR3 pin dsescription	10
4.3	Arduino UNOR3 pin Reference	17
4.4	Gsm 800L module	18
4.5	SIM 800-L GSM cellular	19
4.6	GSM 800-L Sim	19
4.7	Helical GSM antenna	20
4.8	GSM antenna with a U.FL to SMA adapter	21
4.9	SIM 800-L PinOut	22
4.10	LCD display	28
4.11	Interfacing a LCD with a microcontroller	31
4.12	LCD pin description	33
4.13	Single channel relay	33
4.14	relay working	34
4.15	5v Relay module pins	34
4.16	Circuit diagram of power supply	35
4.17	Block diagram of power supply	36
4.18	Regulator LM7805	37
4.19	Schematic diagram	37
4.20	Block diagram	38
5.1	Experiment Prototype and Control Water Pump	41

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
4.1	Technical specs	15
4.2	ARDUINO UNO PIN TYPE DEFINITIONS	17
4.3	power modes of SIM800L module	21
4.4	Command syntax	24
4.5	5 pin description of lcd	29
4.6	PIN symbols and descriptions	30
4.7	Description to connect the Arduino with LCD module	38

CHAPTER-1

INTRODUCTION

Agriculture is the backbone of the Indian Economy. Because without agriculture life is impossible since it is the main source of food for us. The farmer has to toil himself to produce the crop which brings him little revenue, so he has to try some other options for his sustenance, also today the availability of labor for carrying out agricultural activities is less, therefore the automation in the agricultural process is needed. Thus this project has proposed a system so that even after devoting less time to the field, the farmer can carry out his agricultural activities efficiently from remote places. The framework is outfitted with the most recent electronic gadgets, including a microchip and its capacities. The primary motivation behind the chip is to work on a framework plan and give straightforward capacities. Having a microchip in a gadget implies that eliminating, altering, or adding new elements makes it hard to compose programming that controls the gadget. As such, an inserted PC framework is an electronic framework that contains a microcomputer that runs a particular program. The PC is concealed in the item. Underlying frameworks are all over. Consistently, few PCs emerge from the processing plant, which enters our day-by-day business. The underlying framework is the main application that coordinates part of the equipment. PCs regularly have an assortment of programming and applications for an assortment of uses, yet the framework is essentially introduced with fixed capacities that can't be changed without the utilization of a body. One more method for pondering an introduced framework is to have a very much planned PC framework that permits you to play out specific assignments as fast as could really be expected.

There are two primary parts that separate among cost and energy utilization. With such countless frameworks costing thousands to millions of dollars, cost decrease is nothing to joke about. Inherent frameworks frequently use (moderately) slow handling and a limited quantity of memory to diminish costs. Delays are not only the speed of time. To lessen costs, all PC structures are purposely rearranged. For instance, an introduced framework regularly utilizes extra gadgets that are

constrained by a coordinated sequential point of interaction, which is ten to multiple times slower than an adornment contrasted with a PC. Introduced framework applications frequently require constant testing and not many assets: typically, no circle, no working framework, no console or screen. A glimmer drive can supplant a pivoting medium, and a little tap with a LCD screen can be utilized rather than a PC or PC screen. Programming is the name of the product introduced on the equipment inside at least one ROM/Flash memory IC chips. Standard frameworks are intended to be 100 percent solid when worked constantly for extensive stretches of time, now and then estimated over numerous years. Programming is regularly evolved and tried past the overall prerequisites of the product, and this can be effortlessly begun in case of an issue.

The system consists of hardware and software and, finally, the integration of the two parts to provide the results. The hardware system consists of the drivers. In hardware design, we need all the components that are necessary to accomplish the project, and these components are Relay, AC water pump motor.

CHAPTER-2

LITERATURE SURVEY

Y. P. Patil[1]: In this author had proposed the automatic system based on ARM and for communication GSM technology is used. Automatic irrigation technique irrigated using wireless sensor network i.e. Zig-bee and internet technology. Soil moisture sensor, temperature sensors placed in root zone of plant. Soil moisture sensor sense water level and accordingly it turns OFF the motor when the water goes above the danger level and provide accurate supply to the fields. For measuring threshold values of temperature sensor and soil moisture sensor an algorithm is developed that was programmed into a microcontroller to control water quantity. To provide efficient power supply Photovoltaic cells are used. If the water level reaches at danger level, automatically motor will be turned off and it will provide notification to the farmer through GSM. Irrigation system provides adequate water supply in particular area which is real time. The system was based on ARM7TDMI core and GSM. GSM is an important part as communication between various systems occurs using GSM. GSM operate through SMS and is a link between ARM processor and centralized unit. The information send to user in the form of SMS and GSM modem is controlled with the help of standard set of AT (Attention) commands. These commands are used to control majority of the functions of GSM model. The idea was developed for improve irrigation system and reduced cost of irrigation water.

Karan Kansara [2]:In this author proposed an Irrigation Control System Using Android and GSM for Efficient Use of Water and Power. Automatic microcontroller based rain gun irrigation system in which the irrigation will take place only when there will be a need of water as a result it saves a large amount of water as it is avoiding wastage of water. Android is used for mobile devices that include an operating system This application makes use of the GPRS feature of mobile phone as a solution for irrigation control system. These systems covered lower range of agriculture land and not economically affordable. The system supports water management decision, used for monitoring the whole system with GSM (RS-232) module. The system continuously monitors the water level and

provide accurate amount of water required to the fields. The system continuously checks the temperature and humidity of soil in order to retain the nutrient composition of the soil so that. The system uses sensors for remote monitoring and controlling devices which are controlled via SMS using a GSM using android mobile. Automated irrigation system uses valves to turn motor ON and OFF and these valves are operating using microcontroller. These valves are automatically turned OFF and ON depending upon the need of supply. When there is a intense need of supply these valves will be ON and when water reaches at danger level these will be OFF. The system provides essential nutrients and enrich plant growth. Field is equipped with wireless communication sensors that avails better facilitated sensor communication and covers wider field area. The main aim of this paper is to provide automatic irrigation to the plants which helps in saving money and water. The entire system is controlled using 8051 micro controller which is programmed as giving the interrupt signal to the sprinkler. A wireless application of drip irrigation automation supported by soil moisture sensors. The author further added that if different kinds of sensors (that is, temperature, humidity, and etc.) are involved in such irrigation in future works, it can be said that an internet based remote control of irrigation automation will be possible.

GasoreGeoffrey[3]:In this author proposed The control circuit is supplied by 5 volts, when the IR transmitter transmit light beam to the IR receiver, is evidence that there is no obstacle between them.This is an indication that water has not reached the predetermined level at which the IR sensor is placed, with this condition, the input pin of the microcontroller is always high and the output pin of microcontroller is also high. Such output will activate the base of the transistor which connects the relay to the supplying voltage, whenever the relay is supplied; it closes its contact and connects the pump to supplying voltage, 24VDC. The pump fills the water in the tank whenever the above condition is fulfilled.

The red LED lights ON to indicate that the pump is pumping and on the LCD it is displayed "PUMP IS ON".If the moisture content of the soil is low at aa temperature below 15°C) the temperature detector sends the signal to the microcontroller to switch off the electro valve for stopping the irrigation process. The Green LED is off to indicate that the soil is wet and on the LCD it is displayed "VALUE IS OFF".

CHAPTER-3

AIM AND SCOPE OF PROPOSED SYSTEM

3.1 AIM AND OBJECTIVE: Currently, the use of water machine is less efficient because the use of water machine especially when operating the water machine turning on/off the water machine through sms takes some time , and base on this problem a water machine control system are created with microcontroller Arduino UNO and smartphone as a controller so we can control it by making phone calls.The proposed system has been performed for monitoring and remote accessing of the submersible pump by Automatic Turn ON & OFF Using Cellphone Missed Calls and get notified status through messages. The cell phone was used to control the irrigation pump. The panel performed includes Arduino Uno board, GSM 800I module, SPDT relay, Submersible pump, and LCD display.

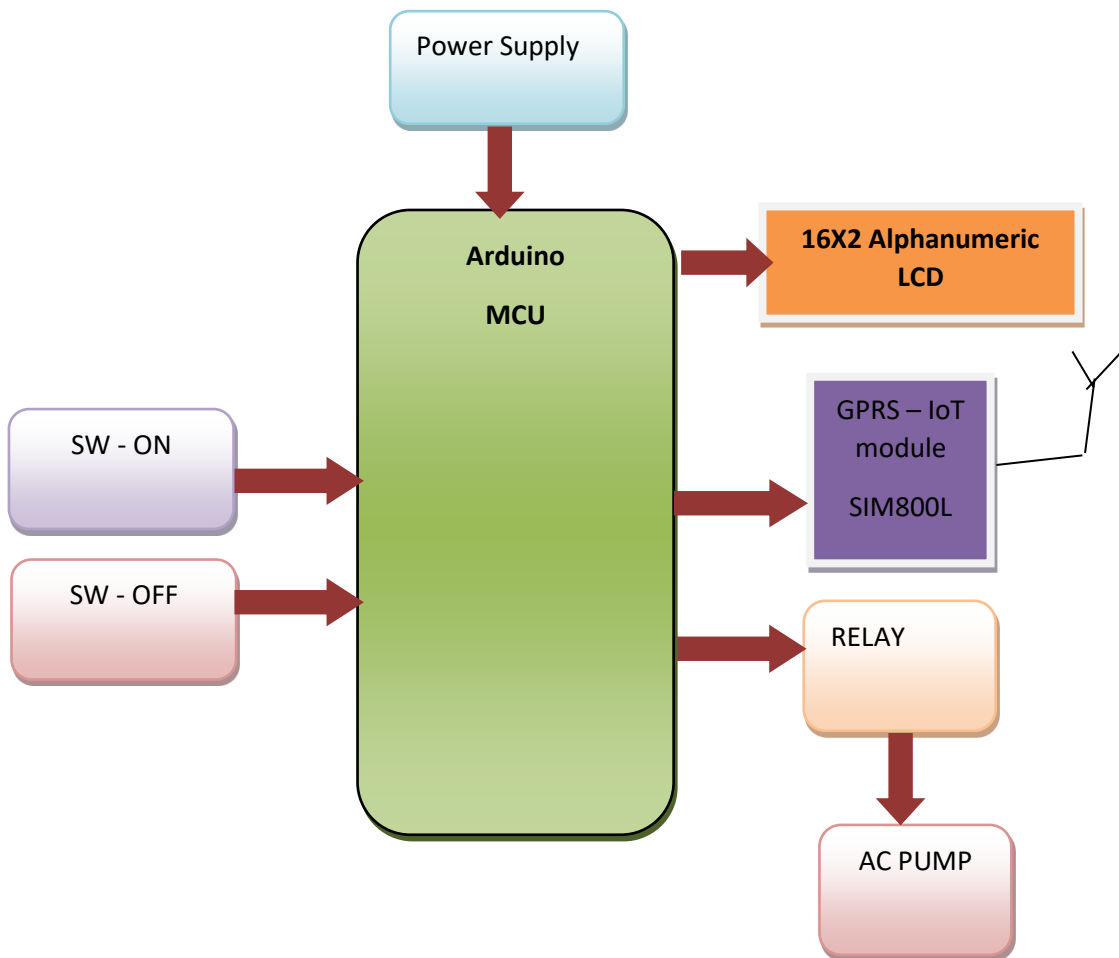


Figure 3.1: Block diagram of prototype

3.2 PROPOSED SYSTEM:

The proposed system has performed for monitoring and remote accessing of the submersible pump by Automatic Turn ON & OFF Using Cellphone Missed Calls and get notified status through messages. The cell phone was used to controlling irrigation pump. The panel performed includes: Arduino Uno board, GSM module, relay, LCD display. The farmer can use there time for other works.

3.3 WORKING :

Arduino board is connected to GSM modem in which transmitter of Arduino is connected to the receiver of GSM modem and receiver of Arduino is connected to the transmitter of GSM modem. GSM modem is working simply as a mobile phone when we call on the number of GSM modem it replies back and can be operated according to the coding which we have done in the microcontroller. The motor pump is connected to the Relay and it is connected to digital pins of an Arduino board. The LCD is connected to the digital pins of Arduino board to display the status of the water pump. 5v power supply is generated from USB cable which is connected to Arduino board. First of all GSM/GPRS module is initialized and waits until it connects to the network. 12v power supply is scheduled to the GSM/GPRS module through adaptor. The redlight indicator illustrates GSM/GPRS is ON.

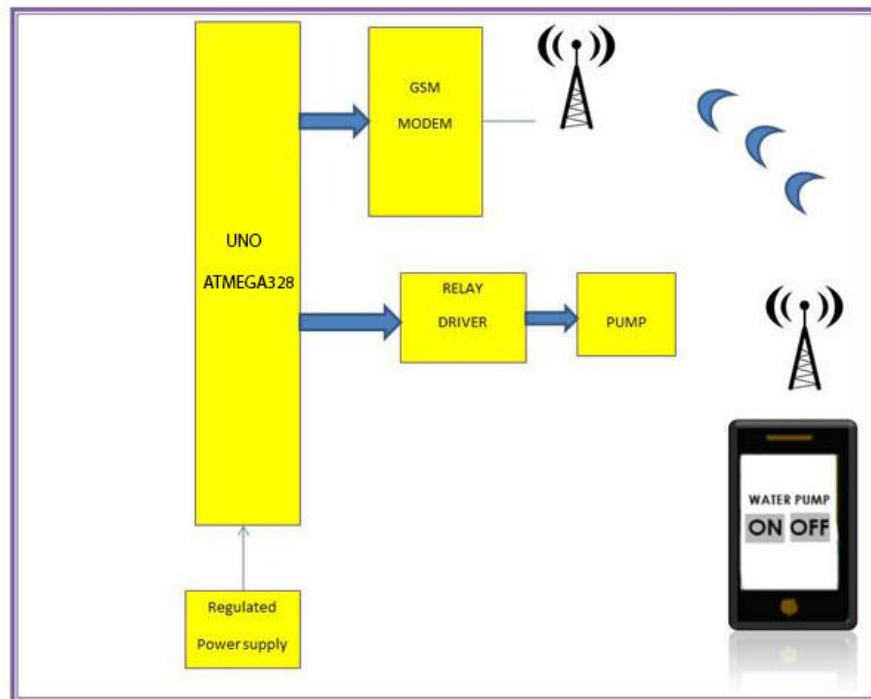


Figure 3.2 Pump on off control system

When we call to the number of the GSM then the digital signal is triggered from the Arduino board to relay then the red light indicator will glow and make the water pump ON. The LCD will display and also a SMS alert is sent to the user by GSM/GPRS module as pump off success. In similar way motor can be turned off by getting alert message as "PUMP OFF SUCCESS. Then the motor pump is switched ON by the user as soon as the message ON is received.

In this task, Arduino UNO R3 is utilized. Thus, to code it, Arduino IDE programming is utilized, and the writing computer programs is done in C. The code comprises of capacities like expressing pin out of the engines, checking for the approaching information on sequential pin from the GSM, Condition for switching on and off of the water pump and to display the status on the LCD display. The figure beneath clarifies the nitty gritty stream graph of the proposed model

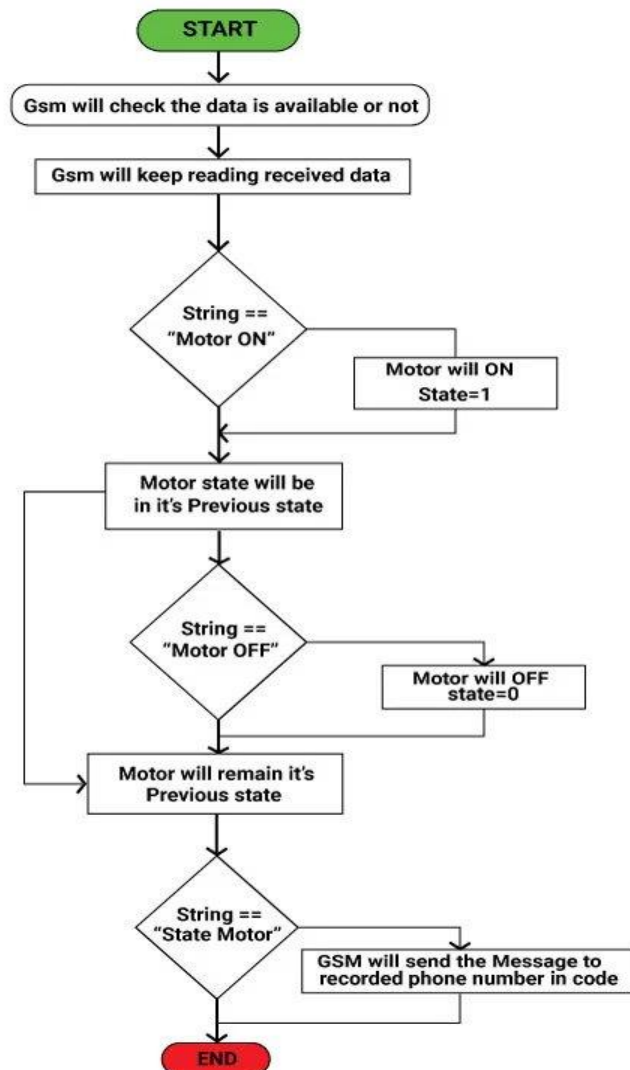


Figure 3.3 work Flowchart

For in-depth understanding of the working refer to the above Flow-Chart,



- AGRICULTURE AREAS
- Aqua Culture

- Ease of operation, Low maintenance cost
- No wastage of time, Fit and forget system

CHAPTER-4

METHODOLOGY

4.1 INTRODUCTION

In this chapter, the procedure for design and development of SMS Based water pump on off system is explained. The block diagram of the system is shown. The schematic of embedded control hardware circuit is shown in diagram. The hardware and software description of the data acquisition system for monitoring physical parameter are described below.

4.2 HARDWARE REQUIRED

- Arduino R3
- GSM 800-L module
- LCD display
- Spdt Relay
- Battery
- USB to serial converter

4.2.1 Arduino UNOR3:

The Arduino Uno R3 is a microcontroller panel dependent happening At mega 328. It works at a voltage of +5V. It comprises of 14 computerized I/O jots of which 6 stretch PWM harvest, a working clock recurrence of 16 MHz, a USB connotation, 6 humble information bases, a power knave then a rearrange key. The Arduino can be customized utilizing C or C++ through the Arduino IDE programming.



Figure 4.1 : Arduino UNOR3

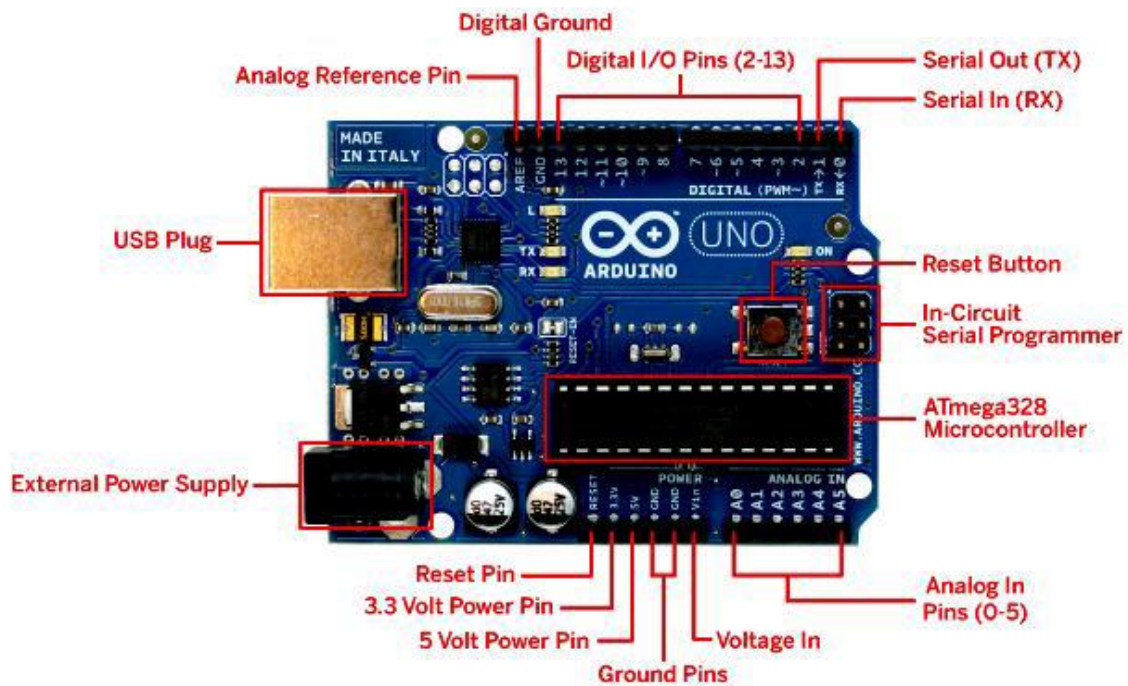


Figure 4.2 : Arduino UNOR3 pin dscription

Arduino UNO is a microcontroller board it has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connec-tion, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simp-ly connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started

4.2.1.1OVERVIEW:-

INTRODUCTION TO EMBEDDED SYSTEMS:

Embedded systems are electronic devices that incorporate microprocessors with in Their implementations. The main purposes of the microprocessors are to simplify the system design and provide flexibility. Having a microprocessor in the device means that removing the bugs, making modifications, or adding new features are only matters of rewriting the software that controls the device. Or in other words embedded computer systems are electronic systems that include a microcomputer to perform a specific dedicated application. The computer is hidden

inside these products. Embedded systems are ubiquitous. Every week millions of tiny computer chips come pouring out of factories finding their way into our everyday products.

Embedded systems are self-contained programs that are embedded within a piece of hardware. Whereas a regular computer has many different applications and software that can be applied to various tasks, embedded systems are usually set to a specific task that cannot be altered without physically manipulating the circuitry. Another way to think of an embedded system is as a computer system that is created with optimal efficiency, thereby allowing it to complete specific functions as quickly as possible.

Embedded systems designers usually have a significant grasp of hardware technologies. They used specific programming languages and software to develop embedded systems and manipulate the equipment. When searching online, companies offer embedded systems development kits and other embedded systems tools for use by engineers and businesses.

Embedded systems technologies are usually fairly expensive due to the necessary development time and built in efficiencies, but they are also highly valued in specific industries. Smaller businesses may wish to hire a consultant to determine what sort of embedded systems will add value to your organization.

CHARACTERISTICS :-

Two major areas of differences are cost and power consumption. Since many embedded systems are produced in the tens of thousands to millions of units range, reducing cost is a major concern. Embedded systems often use a (relatively) slow processor and small memory size to minimize costs.

The slowness is not just clock speed. The whole architecture of the computer is often intentionally simplified to lower costs. For example, embedded systems often use peripherals controlled by synchronous serial interfaces, which are ten to hundreds of times slower than comparable peripherals used in PCs.

Programs on an embedded system often must run with real-time constraints with limited hardware resources: often there is no disk drive, operating system, keyboard or screen.

A flash drive may replace rotating media, and a small keypad and LCD screen may be used instead of a PC's keyboard and screen.

Firmware is the name for software that is embedded in hardware devices, e.g. in one or more ROM/Flash memory IC chips.

Embedded systems are routinely expected to maintain 100% reliability while running continuously for long periods, sometimes measured in years. Firmware is usually developed and tested to much stricter requirements than is general-purpose software, which can usually be easily restarted if a problem occurs.

Platform

There are many different CPU architectures used in embedded designs. This in contrast to the desktop computer market, which as of this writing (2003) is limited to just a few competing architectures, mainly the Intel/AMD x86, and the Apple/Motorola/IBM PowerPC, used in the Apple Macintosh.

One common configuration for embedded systems is the *system on a chip*, an application-specific integrated circuit, for which the CPU was purchased as intellectual property to add to the IC's design.

TOOLS :-

Like a typical computer programmer, embedded system designers use compilers, assemblers and debuggers to develop an embedded system.

Those software tools can come from several sources:

Software companies that specialize in the embedded market Ported from the GNU software development tools.

Sometimes, development tools for a personal computer can be used if the embedded processor is a close relative to a common PC processor.

Embedded system designers also use a few software tools rarely used by typical computer programmers.

Some designers keep a utility program to turn data files into code, so that they can include any kind of data in a program.

Most designers also have utility programs to add a checksum or CRC to a program, so it can check its program data before executing it.

OPERATING SYSTEM

They often have no operating system, or a specialized embedded operating system (often a real-time operating system), or the programmer is assigned to port one of these to the new system.

DEBUGGING

Debugging is usually performed with an in-circuit emulator, or some type of debugger that can interrupt the micro controller's internal microcode.

The microcode interrupt lets the debugger operate in hardware in which only the CPU works. The CPU-based debugger can be used to test and debug the electronics of the computer from the viewpoint of the CPU. This feature was pioneered on the PDP-11.

Developers should insist on debugging which shows the high-level language, with breakpoints and single stepping, because these features are widely available. Also, developers should write and use simple logging facilities to debug sequences of real-time events. PC or mainframe programmers first encountering this sort of programming often become confused about design priorities and acceptable methods. Mentoring, code-reviews and ego less programming are recommended.

THE CONTROL LOOP

In this design, the software simply has a loop. The loop calls subroutines. Each subroutine manages a part of the hardware or software. Interrupts generally set flags, or update counters that are read by the rest of the software.

A simple API disables and enables interrupts. Done right, it handles nested calls in nested subroutines, and restores the preceding interrupt state in the outermost enable. This is one of the simplest methods of creating an exokernel.

Typically, there's some sort of subroutine in the loop to manage a list of software timers, using a periodic real time interrupt. When a timer expires, an associated subroutine is run, or flag is set.

Any expected hardware event should be backed-up with a software timer. Hardware events fail about once in a trillion times. That's about once a year with modern hardware. With a million mass-produced devices, leaving out a software timer is a business disaster.

State machines may be implemented with a function-pointer per state-machine (in C++, C or assembly, anyway). A change of state stores a different function into the pointer. The function pointer is executed every time the loop runs.

Many designers recommend reading each IO device once per loop, and storing the result so the logic acts on consistent values.

Many designers prefer to design their state machines to check only one or two things per state. Usually this is a hardware event, and a software timer.

Designers recommend that hierarchical state machines should run the lower-level state machines before the higher, so the higher run with accurate information.

Complex functions like internal combustion controls are often handled with multi-dimensional tables. Instead of complex calculations, the code looks up the values. The software can interpolate between entries, to keep the tables small and cheap.

One major weakness of this system is that it does not guarantee a time to respond to any particular hardware event. Careful coding can easily assure that nothing disables interrupts for long. Thus interrupt code can run at very precise timings. Another major weakness of this system is that it can become complex to add new features. Algorithms that take a long time to run must be carefully broken down so only a little piece gets done each time through the main loop.

This system's strength is its simplicity, and on small pieces of software the loop is usually so fast that nobody cares that it is not predictable.

Another advantage is that this system guarantees that the software will run. There is no mysterious operating system to blame for bad behavior.

Table 4.1 Technical specs

Microcontroller	Atmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

4.2.1.2 BASIC ARDUINO CODE DEFINITIONS:

setup(): A function present in every Arduino sketch. Run once before the loop() function. Often used to set pinmode to input or output. The setup() function looks like:

```
void setup(){  
  
    //code goes here  
  
}
```

loop(): A function present in every single Arduino sketch. This code happens over and over again. The loop() is where (almost) everything happens. The one exception to this is setup() and variable declaration. ModKit uses another type of loop called “forever()” which executes over Serial. The loop() function looks like:

```
void loop( ) {  
  
    //code goes here  
  
}
```

input: A pin mode that intakes information.

output: A pin mode that sends information.

HIGH: Electrical signal present (5V for Uno). Also ON or True in boolean logic.

LOW: No electrical signal present (0V). Also OFF or False in boolean logic.

digitalRead: Get a HIGH or LOW reading from a pin already declared as an input

digitalWrite: Assign a HIGH or LOW value to a pin already declared as an output.

analogRead: Get a value between or including 0 (LOW) and 1023 (HIGH). This allows you to get readings from analog sensors or interfaces that have more than two states.

analogWrite: Assign a value between or including 0 (LOW) and 255 (HIGH). This allows you to set output to a PWM value instead of just HIGH or LOW.

PWM: Stands for Pulse-Width Modulation, a method of emulating an analog signal through a digital pin. A value between or including 0 and 255. Used with analogWrite.

Table 4.2 ARDUINO UNO PIN TYPE DEFINITIONS: (TAKE A LOOK AT YOUR ARDUINO BOARD)

Reset	3v3	5v	Gnd	Vin	Analog In	RX/TX	Digital	PWM(~)	AREF
Resets Arduino sketch on board	3.3 volts in and out	5 volts in and out	Ground	Voltage in for sources over 7V (9V - 12V)	Analog inputs, can also be used as Digital	Serial comm. Receive and Transmit	Input or output, HIGH or LOW	Digital pins with output option of PWM	External reference voltage used for analog

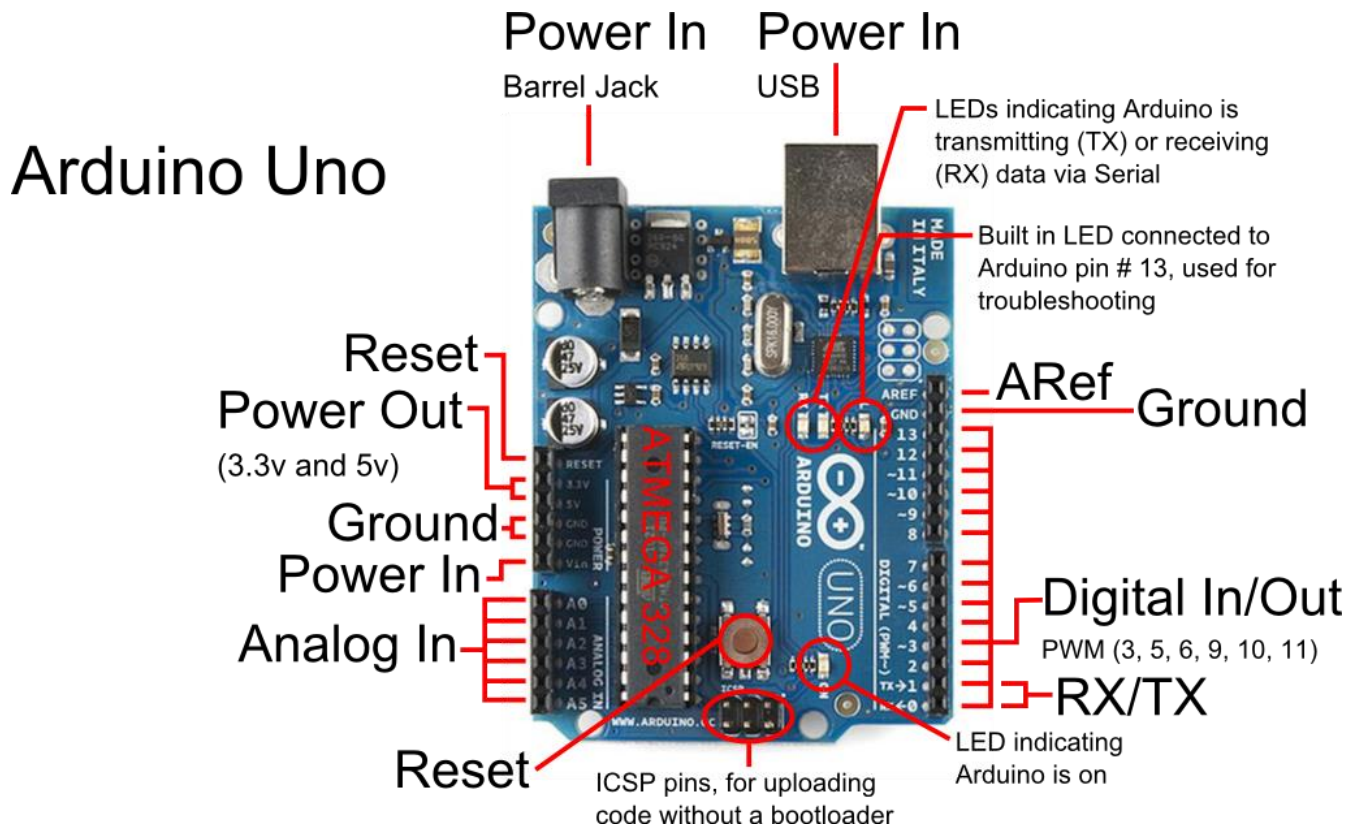


Figure 4.3 : Arduino UNOR3 pin Reference

4.2.2.GSM SIM 800L-MODULE :

SIM800L is a miniature cellular module which allows for GPRS transmission, sending and receiving SMS and making and receiving voice calls. It can do basic functions which can be done via normal phone like calling, send sms. The GSM needs 5V power supply to operate.



Figure 4.4: Gsm 800L module

4.2.2.1 Introduction to SIM800L GSM/GPRS module

At the heart of the module is a SIM800L GSM cellular chip from SimCom. The operating voltage of the chip is from 3.4V to 4.4V, which makes it an ideal candidate for direct LiPo battery supply. This makes it a good choice for embedding into projects without a lot of space.



Figure 4.5: SIM 800-L GSM cellular

All the necessary data pins of SIM800L GSM chip are broken out to a 0.1" pitch headers. This includes pins required for communication with a microcontroller over UART. The module supports baud rate from 1200bps to 115200bps with Auto-Baud detection.

The module needs an external antenna to connect to a network. The module usually comes with a Helical Antenna and solders directly to NET pin on PCB. The board also has a U.FL connector facility in case you want to keep the antenna away from the board.

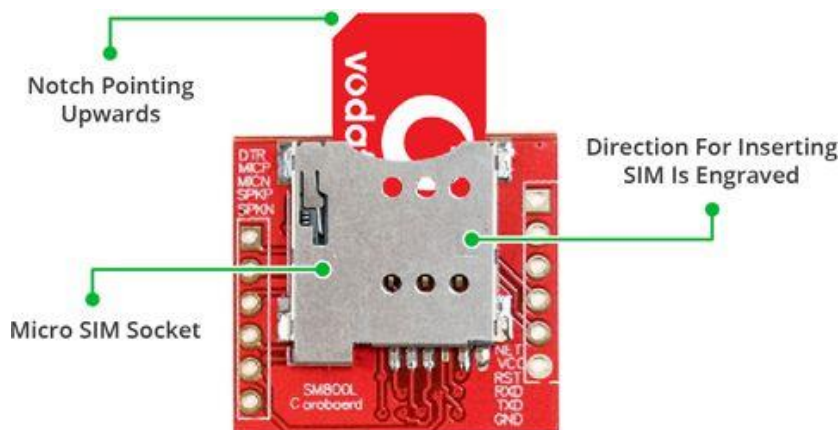


Figure 4.6 : GSM 800-L Sim

There's a SIM socket on the back! Any activated, 2G micro SIM card would work perfectly. Correct direction for inserting SIM card is normally engraved on the surface of the SIM socket.

This module measures only 1 inch² but packs a surprising amount of features into its little frame. Some of them are listed below:

- Supports Quad-band: GSM850, EGSM900, DCS1800 and PCS1900

- Connect onto any global GSM network with any 2G SIM
- Make and receive voice calls using an external 8Ω speaker & electret microphone
- Send and receive SMS messages
- Send and receive GPRS data (TCP/IP, HTTP, etc.)
- Scan and receive FM radio broadcasts
- Transmit Power:
 - Class 4 (2W) for GSM850
 - Class 1 (1W) for DCS1800
- Serial-based AT Command Set
- FL connectors for cell antennae
- Accepts Micro SIM Card

4.2.2.2. Selecting Antenna

An antenna is required to use the module for any kind of voice or data communications as well as some SIM commands. So, selecting an antenna could be a crucial thing. There are two ways you can add an antenna to your SIM800L module.

The first one is a Helical GSM antenna which usually comes with the module and solders directly to NET pin on PCB. This antenna is very useful for projects that need to save space but struggles in getting connectivity especially if your project is indoors.



Fig 4.7 : Helical GSM antenna

The second one is any 3dBi GSM antenna along with a U.FL to SMA adapter which can be obtained online for less than \$3. You can snap-fit this antenna to small u.fl connector located on the top-left corner of the module. This type of antenna has a better performance and allows putting your module inside a metal case – as long the antenna is outside.



Fig 4.8 : GSM antenna with a U.FL to SMA adapter

4.2.2.3 Supplying Power for SIM800L module

One of the most important parts of getting the SIM800L module working is supplying it with enough power.

Depending on which state it's in, the SIM800L can be a relatively power-hungry device. The maximum current draw of the module is around 2A during transmission burst. It usually won't pull that much, but may require around 216mA during phone calls or 80mA during network transmissions. This chart from the datasheet summarizes what you may expect:

Table 4.3 power modes of SIM800L module

Modes	Frequency	Current Consumption
Power down		60 uA
Sleep mode		1 mA

Stand by		18 mA
Call	GSM850	199 mA
	EGSM900	216 mA
	DCS1800	146 mA
	PCS1900	131 mA
GPRS		453 mA
Transmission burst		2 A

Since SIM800L module doesn't come with onboard voltage regulator, an external power supply adjusted to voltage between 3.4V to 4.4V (Ideal 4.1V) is required.

4.2.2.4 SIM800L GSM Module Pinout

The SIM800L module has total 12 pins that interface it to the outside world. The connections are as follows:



Fig 4.9 : SIMM 800-L PinOut

NET is a pin where you can solder Helical Antenna provided along with the module.

VCC supplies power for the module. This can be anywhere from 3.4V to 4.4 volts. Remember connecting it to 5V pin will likely destroy your module! It doesn't even run on 3.3 V! An external power source like Li-Po battery or DC-DC buck converters rated 3.7V 2A would work.

RST (Reset) is a hard reset pin. If you absolutely got the module in a bad space, pull this pin low for 100ms to perform a hard reset.

RxD (Receiver) pin is used for serial communication.

TxD (Transmitter) pin is used for serial communication.

GND is the Ground Pin and needs to be connected to GND pin on the Arduino.

RING pin acts as a Ring Indicator. It is basically the 'interrupt' out pin from the module. It is by default high and will pulse low for 120ms when a call is received. It can also be configured to pulse when an SMS is received.

DTR pin activates/deactivates sleep mode. Pulling it HIGH will put module in sleep mode, disabling serial communication. Pulling it LOW will wake the module up.

MIC± is a differential microphone input. The two microphone pins can be connected directly to these pins.

SPK± is a differential speaker interface. The two pins of a speaker can be tied directly to these two pins.

4.2.2.5 GSM Network Operators

T-Mobile and Cingular operate GSM networks in the United States on the 1,900 MHz band. GSM networks in other countries operate at 900, 1,800, or 1,900 MHz

4.2.2.6 GSM carrier frequencies

GSM networks operate in a number of different carrier frequency ranges (separated into GSM frequency ranges for 2G and UMTS frequency band for 3G), with most 2G GSM networks operating in the 900 MHz or 1800 MHz bands. Where these bands were already allocated, the 850 MHz and 1900 MHz bands were used instead (for example in Canada and the United states). In rare cases the 400 and 450 MHz frequency bands are assigned in some countries because they were previously used for first-generation systems.

Most 3G networks in Europe operate in the 2100 MHz frequency band. Regardless of the frequency selected by an operator, it is divided into time slots for individual phones to use. This allows eight full-rate or sixteen half-rate speech channels per Radio frequencies. These eight radio timeslots (or eight burst periods) are grouped into a TDMA frame. Half rate channels use alternate frames in the same timeslot. The channel data rate for all 8 channels is 270.833 Kbit/s, and the frame duration is 4.615 ms. The transmission power in the handset is limited to a maximum of 2 watts in GSM850/900 and 1 watt in GSM1800/1900.

4.2.2.7 GSM ATCommands

AT Commands are used to perform different operations is GSM module

Short message commands

Preferred Message Format +CMGF

Description:

The message formats supported are *text mode* and *PDU mode*. In PDU mode, a complete SMS Message including all header information is given as a binary string (in hexadecimal format). Therefore, only the following set of characters is allowed: {'0','1','2','3','4','5','6','7','8','9', 'A','B','C','D','E','F'}. Each pair of characters are converted to a byte (e.g.: '41' is converted to the ASCII character 'A', whose ASCII code is 0x41 or 65). In Text mode, all commands and responses are in ASCII characters. The format selected is stored in EEPROM by the +CSAS command.

Table 4.4 Command syntax:

AT+CMGF

Command	Possible responses
AT+CMGF?	+CMGF=1
Note :possible message format	Ok Note :Text mode

Send message + CMGS

To send a message in text mode **CMGS** command used

Description:

The <address> field is the address of the terminal to which the message is sent. To send the message, simply type, <ctrl-Z> character (ASCII 26). The text can contain all existing characters except <ctrl-Z> and <ESC> (ASCII 27). This command can be aborted using the <ESC> character when entering text. In PDU mode, only hexadecimal characters are used ('0'...'9','A'...'F').

Syntax:

In SMS text mode, the syntax of the +CMGS AT command is: (Optional parameters are enclosed in square brackets.)

+CMGS=address[,address_type]<CR>sms_message_body<Ctrl+z> Before we discuss each of the parameters, let's see an example that gives you some idea of how an actual command line should look like: AT+CMGS="+85291234567",145<CR>This is an example for illustrating the syntax of the +CMGS AT command in SMS text mode.<Ctrl+z>

The address Parameter

The first parameter of the +CMGS AT command, *address*, specifies the destination address to send the SMS message to. Usually it is a mobile number formatted using the typical ISDN / telephony numbering plan (ITU E.164/E.163). For example, "+85291234567", "91234567", etc. Note that the value passed to the *address* parameter should be a string, i.e. it should be enclosed in double quotes.

The address_type Parameter

The second parameter of the +CMGS AT command, *address_type*, specifies the type of the address assigned to the *address* parameter. Two values are commonly used. They are 129 and 145:

As *address_type* is an optional parameter, it can be omitted. If you do so, the GSM/GPRS modem or mobile phone will use the default value of the *address_type* parameter, which is:

- 129 if the value of *address* does not start with a "+" character. For example, "85291234567".
- 145 if the value of *address* starts with a "+" character. For example, "+85291234567".

The <CR> Character

<CR>, which represents the carriage return character, follows the *address_type* parameter. When the GSM/GPRS modem or mobile phone receives the carriage return character, it will send back a prompt formed by these four characters: the carriage return character, the linefeed character, the ">" character and the space character.

The sms_message_body Parameter

The third parameter of the +CMGS AT command, *sms_message_body*, specifies the body of the SMS message to be sent. Entering the <Esc> character will cancel the +CMGS AT command.

The <Ctrl+z> Character

When you finish entering the SMS message body, you have to enter the <Ctrl+z> character to mark the end of the SMS message body. The GSM/GPRS modem or mobile phone will then attempt to send the SMS message to the SMS center

Read message +CMGR

The AT command +CMGR (command name in text: Read Message) is used to read a message from a message storage area. The location of the message to be read from the message storage area is specified by an index number. The message to be retrieved by the AT command +CMGR does not necessarily have to be an SMS message. It can be of other message types such as status reports and cell broadcast messages, but we will only focus on SMS messages here.

Format of the Information Response of the +CMGR AT Command in SMS Text Mode.

If the GSM/GPRS modem or mobile phone successfully reads the SMS message from message storage, it will return an information response to the computer / PC. In SMS text mode, the format of the information response of the +CMGR AT command is different for different message types. In the sections that follow, we

assume the message to be read is an SMS message but not of other message types like status reports and cell broadcast messages.

Incoming SMS Messages and Outgoing SMS Messages

If the SMS message retrieved is an SMS message received from the SMS center (i.e. incoming SMS message), the information response of the +CMGR AT command in SMS text mode has the following format: (Optional fields are enclosed in square brackets.)

```
+CMGR:message_status,address,[address_text],service_center_time_stamp[address_type,TPDU_first_octet,protocol_identifier,data_coding_scheme,service_center_address,service_center_address_type,sms_message_body_length]<CR><LF>
sms_message_body
```

For incoming SMS messages:

```
+CMGR:"RECREAD","+85291234567",,"07/04/20,10:08:02+32",145,4,0,0,"+85290000000",145,49
```

It is easy to read text messages via AT Commands

To place a call following AT commands are used:

ATD+ +ZZxxxxxxxxxx; – Dials a specified number. The semicolon (;) modifier at the end separates the dial string into multiple dial commands. All but the last command must end with the semicolon (;) modifier.

ATH – Hangs up the call

Arduino Code – Receiving Call

Receiving call doesn't require any special code; you just have to keep listening to the SIM800L module. Yet, you may find this sketch very useful, when you need to trigger an action when a call from specific phone number is received.

Incoming call is usually indicated by 'RING' on serial monitor followed by phone number and caller ID. To accept/hang a call following AT commands are used:

ATA – Accepts incoming call.

ATH – Hangs up the call. On hanging up the call it sends NO CARRIER on the serial monitor indicating call couldn't connect.

Below output on serial monitor shows call received by SIM800L GSM module.

4.2.3 LCD DISPLAY:

LCD's typically have 16 data pins and 2 for the LED backlight. Character LCDs use a standard 14-pin interface and those with backlights have 16 pins. There may also be a single backlight pin, with the other connection via Ground or VCC pin. The two backlight pins may precede the pin 1.



Fig 4.10 LCD display

Liquid crystal display a type of display used in digital watches and many portable computers.

LCD displays utilize two sheets of polarizing material with a liquid crystal solution between them. An electric current passed through the liquid causes the crystals to align so that light cannot pass through them. Each crystal, therefore, is like a shutter, either allowing light to pass through or blocking the light. The liquid crystals can be manipulated through an applied electric voltage so that light is allowed to pass or is blocked.

By carefully controlling where and what wavelength (color) of light is allowed to pass, the LCD monitor is able to display images. A back light provides LCD monitor's brightness. Other advances have allowed LCD's to greatly reduce liquid crystal cell response times.

Response time is basically the amount of time it takes for a pixel to “change colors”. In reality response time is the amount of time it takes a liquid crystal cell to go from being active to inactive Here the LCD is used at both the Transmitter as well as the receiver side.

The input which we give to the microcontroller is displayed on the LCD of the transmitter side and the message sent is received at the receiver side which displays at the receiver end of the LCD and the corresponding operation is performed.

They make complicated equipment easier to operate. LCDs come in many shapes and sizes but the most common is the 16 character x 4 line display with no backlight.

It requires only 11 connections – eight bits for data (which can be reduced to four if necessary) and three control lines (we have only used two here). It runs off a 5V DC supply and only needs about 1mA of current.

The display contrast can be varied by changing the voltage into pin 3 of the display.

4.2.3.1Pin description of LCD:

From this description, the interface is a parallel bus, allowing simple and fast reading/writing of data to and from the LCD. This waveform will write an ASCII Byte out to the LCD's screen.

Table 4.5 pin description of lcd

Pins	Description
1	Ground
2	Vcc
3	Contrast Voltage
4	"R/S" _Instruction/Register Select
5	"R/W" _Read/Write LCD Registers
6	"E" Clock
7 - 14	Data I/O Pins

Table 4.6 PIN symbols and descriptions

Vcc, Vss and Vee

While Vcc and Vss provide +5V and ground respectively, Vee is used for controlling LCD contrast.

PIN	SYMBOL	I/O	DESCRIPTION
1	Vss	--	Ground
2	Vcc	--	+5V power supply
3	Vee	--	Power supply to control contrast
4	RS	I	RS=0 to select command register RS=1 to select data register
5	R/W	I	R/W=0 for write R/W=1 for read
6	EN	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

The ASCII code to be displayed is eight bits long and is sent to the LCD either four or eight bits at a time. If four bit mode is used, two "nibbles" of data (Sent high four bits and then low four bits with an "E" Clock pulse with each nibble) are sent to make up a full eight bit transfer. The "E" Clock is used to initiate the data transfer within the LCD. Deciding how to send the data to the LCD is most critical decision to be made for an LCD interface application.

Eight-bit mode is best used when speed is required in an application and at least ten I/O pins are available. The "R/S" bit is used to select whether data or an instruction is being transferred between the microcontroller and the LCD.

If the Bit is set, then the byte at the current LCD "Cursor" Position can be reader written. When the Bit is reset, either an instruction is being sent to the LCD or the execution status of the last instruction is read back.

4.2.3.2.INTERFACING LCD WITH CONTROLLER:

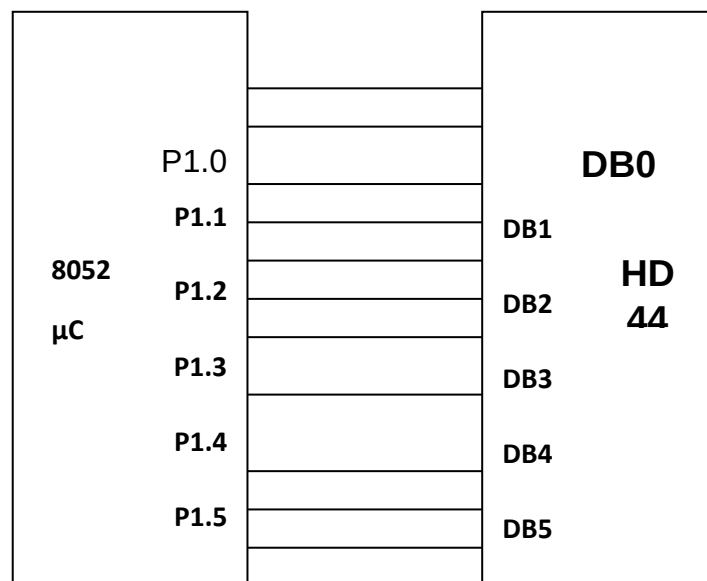


Fig 4.11 Interfacing a LCD with a microcontroller.

4.2.3.3 Basic commands of LCD:

When LCD is powered up, the display should show a series of dark squares, possibly only on part of display.

These characters are actually in their off state, so the contrast control should be adjusted anti-clockwise until the squares are just visible.

The display module resets itself to an initial state when power is applied, which curiously the display has blanked off so that even if characters are entered, they cannot be seen.

It is therefore necessary to issue a command at this point, to switch the display on.

4.2.3.4 Prototype circuit:

For a LCD module to be used effectively in any piece of equipment, a microprocessor or a micro controller is usually required to drive it.

However, before attempting to wire the two together, some initial experiments can be performed by connecting a series of switches to the pins of the module.

This can be a quite beneficial step, if even you are thoroughly conversant with the workings of microprocessors.

4.2.3.5 Circuit description of LCD experiment:

The circuit can be wired up on a “plug-in-style” prototyping board, using dual-in-line switches for the data lines (S1-S8)

A toggle switch for the RS input (S10) and a momentary action switch (or macro switch) for usage.

Most of the LCD modules conform to a standard interface specification. A 14pin access is provided having eight data lines, three control lines and three power lines.

The connections are laid out in one of the two common configurations, either two rows of seven pins, or a single row of 14 pins.

One of the pins are numbered on the LCD's print circuit board (PCB), but if not, it is quite easy to locate pin1.

Since this pin is connected to ground, it often has a thicker PCB track, connected to it, and it is generally connected to metalwork at same point.

4.2.3.6 Pin description:

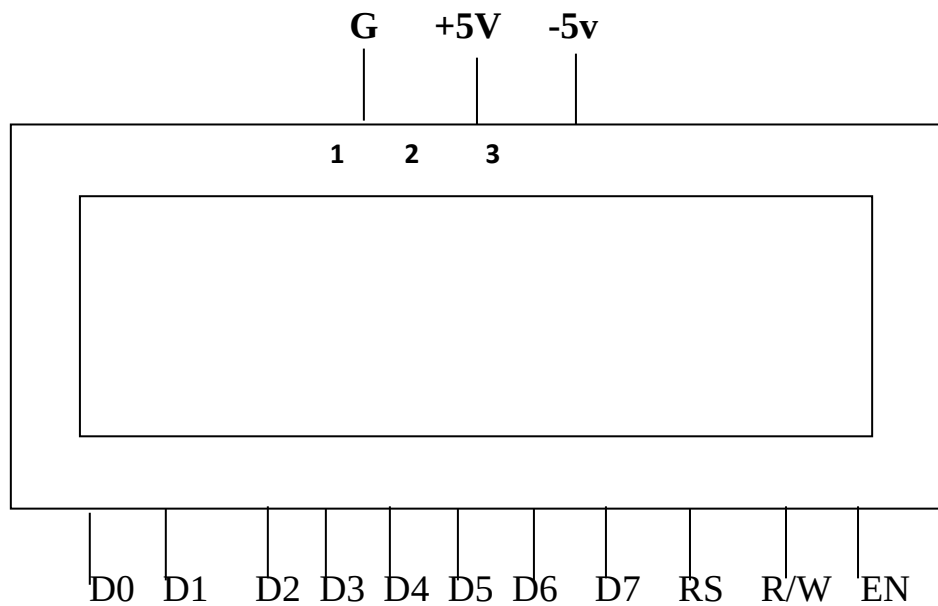


Fig 4.12 LCD pin description

The LCD plays a major role in the entire operation as it has the ability to display the certain data that the user has entitled. LCD display varies from input to input as there is no specific outline for it to operate.

4.2.4 SINGLE CHANNEL RELAY:

The Single Pole Double Throw relay is quite useful in certain applications because of its internal configuration. It has one common terminal and 2 contacts in 2 different configurations: one can be Normally Closed and the other one is opened or it can be Normally Open and the other one closed. So basically you can see the SPDT relay as a way of switching between 2 circuits: when there is no voltage applied to the coil one circuit “receives” current, the other one doesn’t and when the coil gets energised the opposite is happening.



Fig 4.13 Single channel relay

Relays consist of three pins normally open pin , normally closed pin, common pin and coil. When coil powerd on magntic field is generated the contacts connected to each other.

4.2.4.1 features:

- Contact current 10A and 250V AC or 30V DC.
- Each channel has indication LED.
- Coil voltage 12V per channel.
- Kit operating voltage 5-12 V ,
- Input signal 3-5 V for each channel.
- Three pins for normally open and closed for each channel.

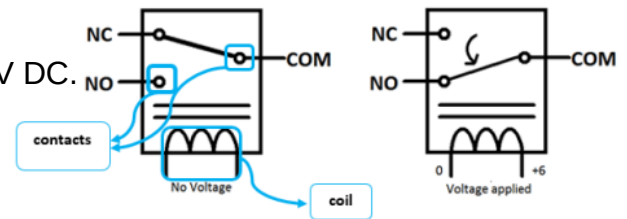


Fig 4.14 relay working

How to connect relay module with Arduino

As shown in relay working idea it depends on magnetic field generated from the coil so there is power isolation between the coil and the switching pins so coils can be easily powered from Arduino by connecting VCC and GND pins from Arduino kit to the relay module kit after that we choose Arduino output pins depending on the number of relays needed in project designed and set these pins to output and make it out high (5 V) to control the coil that allow controlling of switching process

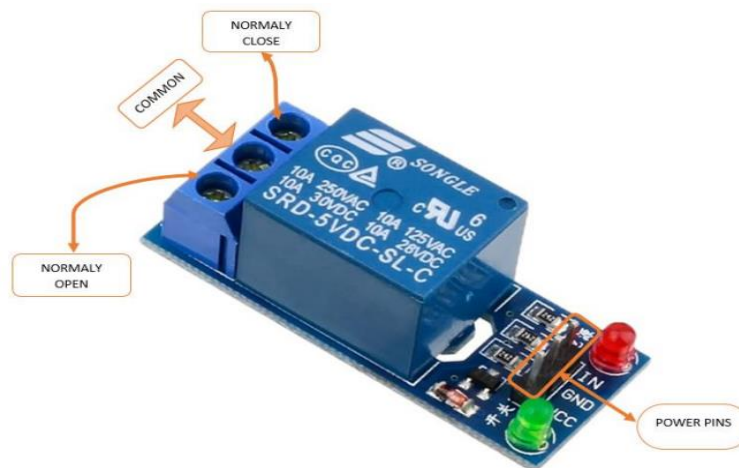


Figure: 4.15 5v Relay module pins

4.2.5 POWER SUPPLY:

In my project I used 9 volts transformer for continuous power supply. Why I am using this means to continuous power will come. Otherwise If I use a battery some times the total currents will loss so that's way I am using A.C Transformer. A.C transformer is giving the input to Bridge Rectifier. Bridge Rectifier converts A.C to D.C. After that we are using one filter capacitor 1000uf/25v electrolytic capacitor .We connecting this capacitor in parallel section. The main purpose of this capacitor is if there is any alternate peaks we need to reduce that peaks. Nothing but a filtering that repull's. After that we are using LM7805 Regulator Most digital logic circuits and processors need a 5 volt power supply. To use these parts we need to build a regulated 5 volt source. We make a 5 volt power supply, The LM7805 is simple to use. First connect the positive lead of our unregulated DC power supply Input pin, connect the negative lead to the Common pin and then when we turn on the power, we get a 5 volt supply from the Output pin. Here we are using one red color led to indicate the power.

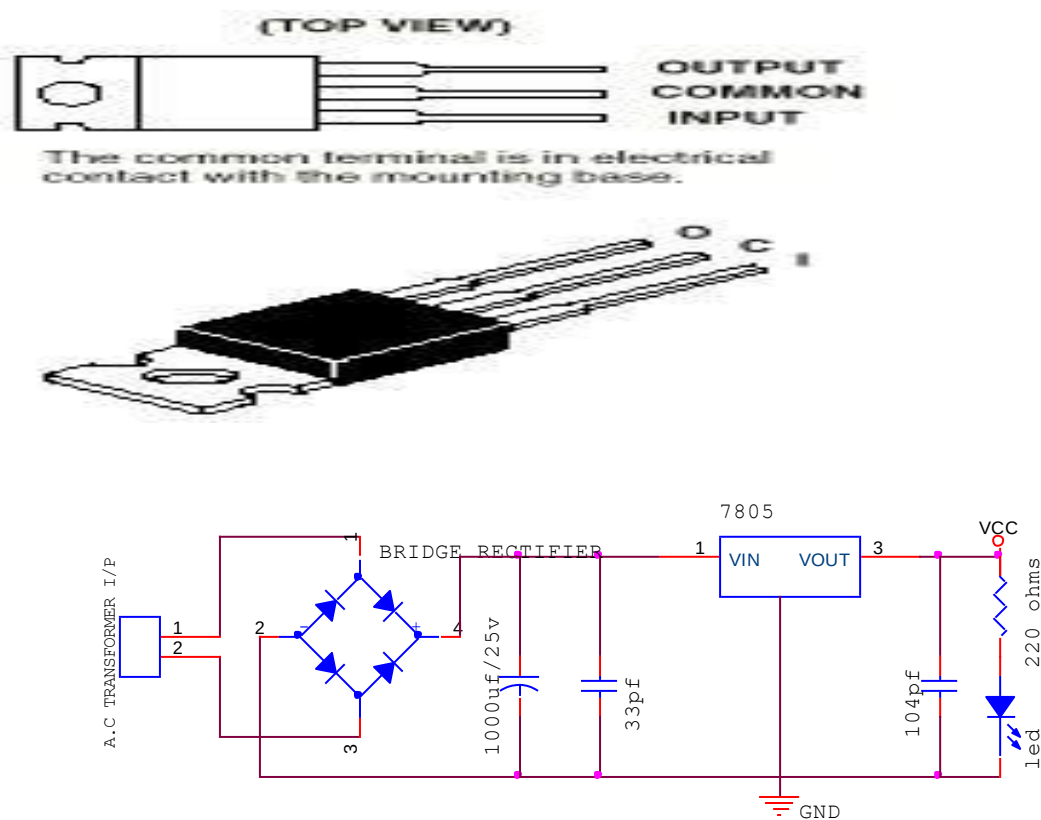


Fig 4.16 Circuit diagram of power supply

We are using General purpose PCB. To mount or place the components on the PCB. After that

The LM7805 is simple to use. You simply connect the positive lead of your unregulated DC power supply (anything from 9VDC to 24VDC) to the Input pin, connect the negative lead to the Common pin and then when you turn on the power, you get a 5 volt supply from the Output pin.

4.2.5.1 CIRCUIT FEATURES:

- Brief description of operation: Gives out well regulated +5V output, output current capability of 100 mA
- Circuit protection: Built-in overheating protection shuts down output when regulator IC gets too hot
- Circuit complexity: Very simple and easy to build
- Circuit performance: Very stable +5V output voltage, reliable operation
- Availability of components: Easy to get, uses only very common basic components
- Design testing: Based on datasheet example circuit, I have used this circuit successfully as part of many electronics projects
- Applications: Part of electronics devices, small laboratory power supply
- Power supply voltage: Unregulated DC 8-18V power supply
- Power supply current: Needed output current + 5 mA

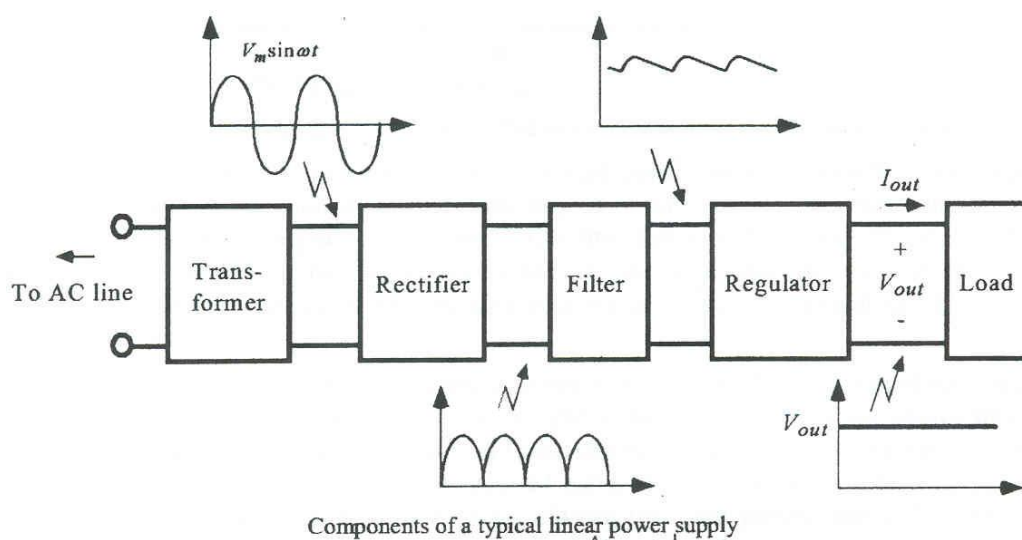


Fig 4.17 Block diagram of power supply

The power supply unit is used to provide constant 5 V dc supply to the peripherals. The 230 V ac is converted into 9 V ac by using a transformer and then a bridge rectifier rectifies it to a 9 V dc with ac ripples. This is then filtered by electrolytic capacitors used across the rectifier output. LM7805 regulator is employed to obtain a constant 5 V dc at the output.

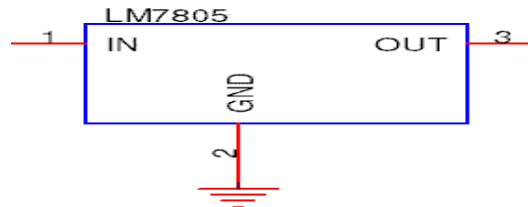


Fig 4.18 Regulator LM7805.

4.3.Hardware implementation

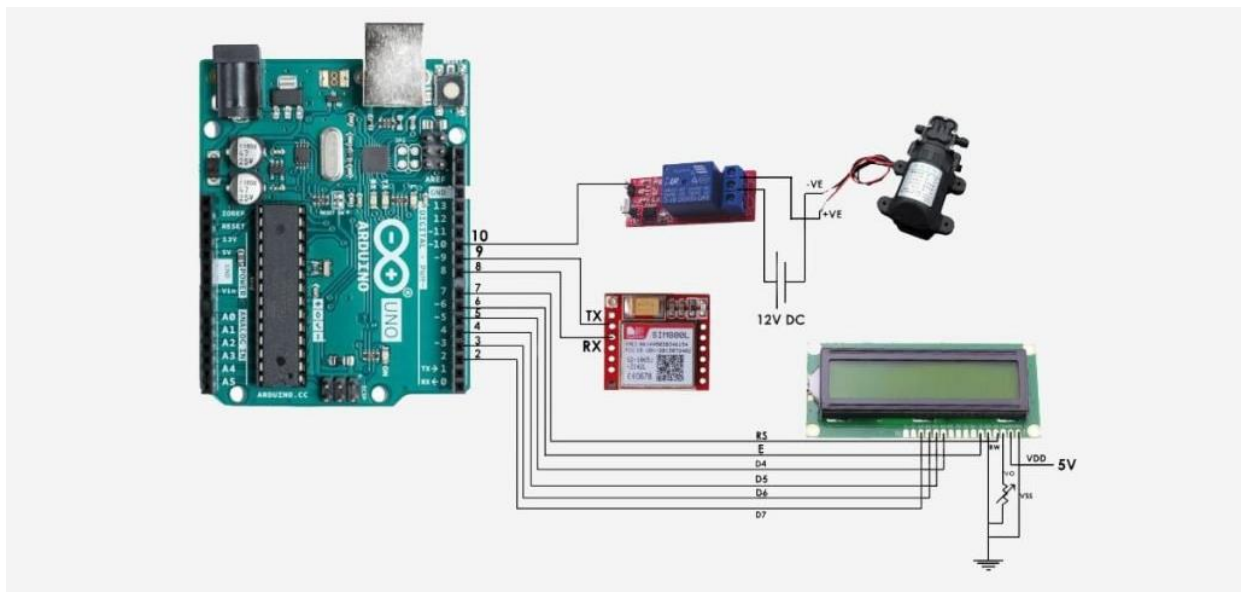


Figure 4.19: Schematic diagram

The digital pins of the Arduino UNO R3 board is connected to the LCD module, GSM module and Relay, The relay is connected to the water pump. The connections to the pins are explained below.

This is a digital IC used as a regulator for regulating the output. LM7805 provides a constant 5V dc supply as output. For this it needs at least 8V dc as input.

4.3.1 Connecting LCD Module

Refer to the following table to connect the Arduino with the LCD module.

Table 4.7 PINs to connect the Arduino with the LCD module.

Arduino Pin	LCD
Pin 7	RS
Pin 6	EN
Pin 5	D4
Pin 4	D5
Pin 3	D6
Pin 2	D7

4.3.2 Connecting GSM Module

Connect the Vcc of the GSM module to the 5V supply on the Arduino board.

Next, connect the ground of the GSM module to the ground pin of Arduino.

Further,

- Rx of GSM – Tx of the Arduino (Pin 8)
- Tx of GSM – Rx of the Arduino (Pin 9)

That's it for the GSM module.

4.3.3 Connecting Motor (Water Pump)

The Arduino is connected to the water pump through the relay.

So before proceeding further, we need to connect the relay board to Arduino.

Hence, connect the relay pin to pin-10 of the Arduino board.

Next, the relay is connected to the water pump and a power supply.

Connect the relay NO pin to the one terminal of the water pump. Another terminal of the water pump should be connected to the +ve of 12V power supply.

At last, connect the common pin of the relay to the –ve of 12V power supply.

4.4 Software implementation

In this task, Arduino UNO R3 is utilized. Thus, to code it, Arduino IDE programming is utilized, and the writing computer programs is done in C. The code comprises of capacities like expressing pin out of the engines, checking for the approaching information on sequential pin from the GSM, Condition for switching on and off of the water pump and to display the status on the LCD display. The figure beneath clarifies the nitty gritty stream graph of the proposed model

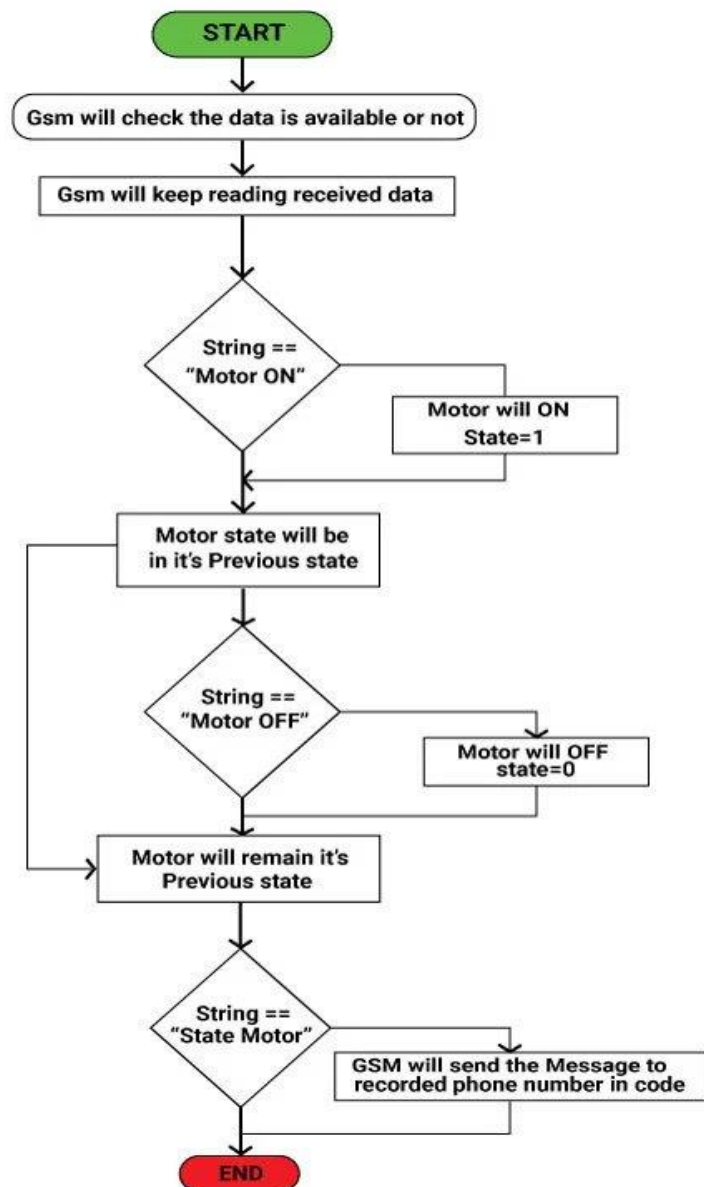


Fig 4.20 Block diagram

String "textMessage" is used to store the string data which gets received from a GSM module. We have added a condition if a string data is get matched with the string data we have added in "if" condition then motor get turned on. For eg., if the GSM module receives a string data "motor on" then the condition is get satisfied and the motor/ water pump will get tur on, Similarly for text "motor off" motor/ water pump will get turn off.

Steps to Use the System

Step 1 – Turn on the System. This will trigger an acknowledgement message to your number.

Step 2 – To ON the water pump call to the number 995****358.

Step 3 – It will be received by GSM and sent to Arduino.

Step 4 – Arduino will turn ON the water pump through the relay.

Step 5 – The pump will start.

Step 6 – Now, after a perticular time, Call to the number 995****358.

Step 7 – Here the Step-3 will be repeated, and the process in step-4 will be reversed, and the motor will be turned OFF.

CHAPTER-5

RESULTS AND DISCUSSION

5.1 RESULTS

Arduino UNO on this prototype is the central part of the overall control system of input and output connected to various other devices, below is the result of prototype and experiment perform. (See Figure)

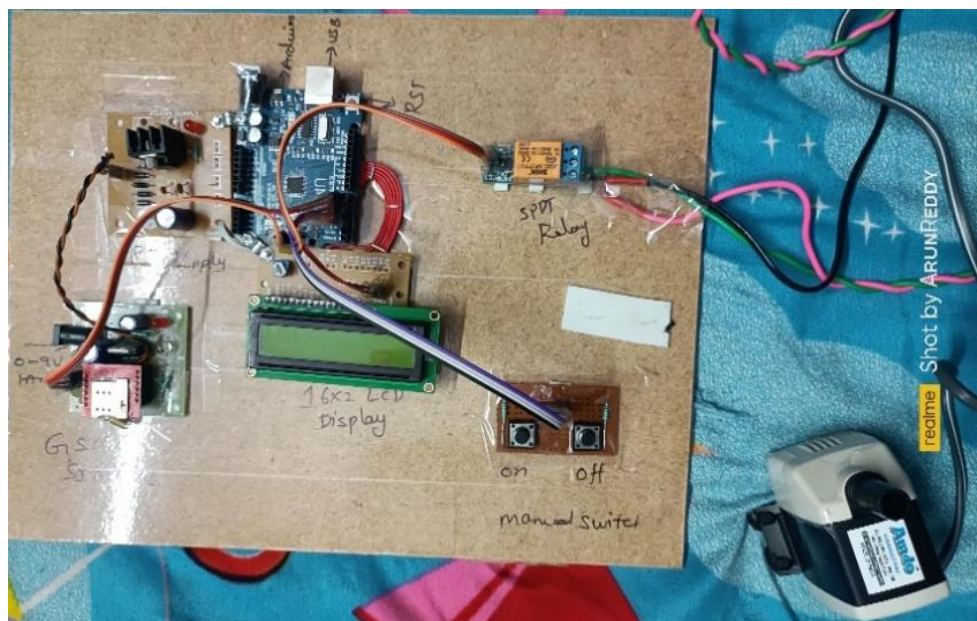


Figure 5.1. Experiment Prototype and Control Water Pump.

The experiment results of GSM SIM900A module is performed when we need to switch on the water pump by calling we can switch on the water pump and we get the notification message as the “PUMP ON SUCCESS”. In similar way motor can be turned off by getting notification message as “PUMP OFF SUCCESS”.

This prototype is tested in the agriculture fields and the status of water pump received through the smartphone.

The test perform on the Agriculture field, from some experiment that obtained good results can be controlled with a considerable distance due to connect through a GSM network that is wider than WiFi module or ethernet shield, and for the security of communication is filter from SMS number so not just any SMS message can be process.

Below is the list of Messages .which is used to send Pump On and Off Status.

SL.NO	Operation	SMS from phone
1	Calling to GSM number to ON	PUMP ON SUCCESS
2	Calling to GSM number to OFF	PUMP OFF SUCCESS
3	Pump ON manually	MANUAL PUMP ON
4	Pump ON manually	MANUAL PUMP OFF

5.2 DISCUSSION

In coastal area they only use the canal water for agriculture land,the canal water will be available for particular days only.Mostly 2-3 days.Some times,the neighbour farmers will switch off the water pump if the farmer is not there in agriculture field.

By using the sms based remote agriculture pump on off control and notification the farmer can switch on and off the water pumpby just doing the phone call and the farmer will get the status of the water pump through sms.If it is manually switched off then the farmer will get notification message as it is switched off manually.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

An experiment of water pump control prototype got a good result, and we can control the water pump in the agriculture fields by calling to the number and the status of water pump get notified through sms, Instead typing and sending the sms. If the pump is switch off manually then we will get the message as the pump is switched off manually and If the pump is switch on manually then we will get the message as the pump is switched on manually.

6.2 FUTURE SCOPE

- In present, one water pump is controlled. In future, we can implent it to control two or more water pumps .
- We can implement it in the Aquaculture.

REFERENCES

- [1]. Kay, M., "Smallholder irrigation technology: Prospects for sub-Saharan Africa" International Program for Technology and Research in Irrigation and Drainage, FAO, Rome, 2001, pp. 1–25.
- [2]. N. Shah and I. Das, "Precision Irrigation Sensor Network Based Irrigation", a book on Problems, Perspectives and Challenges of Agricultural Water Management, IIT Bombay, India, pp. 217–232, April 2008
- [3]. Fangmeier, D. D., Garrot, D. J., Mancino, C.F and Husman, S. H., "Automated irrigation systems using plant and soil sensors", American Society of Agricultural Engineers, ASAE Publication, 1990, pp. 533-537.
- [4]. Benzekri, A., Meghriche, K., and Refoufi, L., PC-based automation of a multi-mode control for an irrigation system Proceedings of International symposium on industrial embedded systems, Lisbon, July 2007, pp. 310 -315.
- [5]. Shinghal, K., Noor, A., Srivastava, N., and Singh, R., Wireless sensor networks in agriculture for potato farming International Journal of Engineering, Science and Technology, Vol. 2, No. 8, 2010, pp. 3955 -3963.
- [6]. Gautam, I., and Reddy, S. R. N., Innovative GSM-Bluetooth based remote controlled embedded system for irrigation, International Journal of Computer Applications, Vol. 47, No. 8, 2012, pp. 1.
- [7]. Zhang, F., Yang, M., and Ying, H., The application of GSM communication in agricultural automation, Journal of Technology for Agriculture, Vol. 1, No. 1, 2004, pp. 39-41.
- [8]. Shen Jin, Song Jingling, Han Qiuyan, Wang Shengde, and Yang Yan, School of Electric and Electronic Engineering, A Remote Measurement and Control System for Greenhouse Based on GSM-SMS, IEEE 8th International Conference on Electronic Measurement and Instrument, 2007, pp. 45-82

- [9]. Webin Huang, Guanglong Wang, Research of Wireless Sensor Networks for an Intelligent Measurement System Based on ARM , pp.1074-1079, 2011.
- [10]. Sezen SM, Yazar A, Irrigation Management on Yield And Quality Of Tomatoes Grown in different Soilless Media in Glasshouse, 41-48, 2010.
- [11]. Daniel K.Fisher and HirutKebede, a Low Cost Microcontroller-Based System to Monitor Crop Temperature and Water Status, pp. 168-173, 2010.