INDIAN NATIONAL COMMITTEE ON SURFACE WATER (INCSW-CWC)

UID	DL-2011-101
Type (State whether final or draft report)	Final Report
Name of R&D Scheme	Development of an Indigenous Automated Micro Irrigations System
Name of PI & Co-PI	PI: Dr. (Ms.) Neelam Patel, Principal Scientist, Water Technology Centre, ICAR-IARI. Co-PI: Dr.T.B.S. Rajput, Emeritus Scientist, Water Technology Centre, ICAR-IARI.
Institute Address	Indian Agricultural Research Institute, New Delhi-110012
Circulation (State whether Open for public or not)	Open as per rule of Institute
Month & Year of Report Submission	August, 2020

©INCSW Sectt.
Central Water Commission
E-Mail: incsw-cwc@nic.in

Project Report
Indian National Committee for Irrigation & Drainage
Ministry of Water Resources
Sewa Bhavan, R.K.Puram New Delhi.

Title of the project: "Development of Indigenous Automated Micro-Irrigation System"

Project Scientists:

PI: Dr. (Ms.) Neelam Patel, Principal Scientist, Water Technology Centre, ICAR-IARI.

Co-PI: Dr.T.B.S. Rajput, Emeritus Scientist, Water Technology Centre, ICAR-IARI.

Research Station/ Institution

Name : Water Technology Center.

Address: Indian Agricultural Research Institute, New Delhi-110012.

Telephone: 011-25848703, 09868060359 Fax 011-25848703 e-mail: np_wtc@yahoo.com.

Date of sanction: **Dec, 2012** **Duration:** ***42 months***

Total sanctioned budget including any additional budget sanctioned: Rs. 17, 93,138 /-

Copy of budget sanction and Utilization certificates have enclosed in end of report.

Objectives:

1**.** To develop and test the soil moisture sensors.

2**.** To develop the software for scheduling of irrigation on real time basis taking into consideration of the field moisture conditions.

3. Integration of sensors, trigger mechanism, regulatory valves and control unit through embedded system to develop a completely indigenous automated micro irrigation system.

4. Testing of developed automated system at different climatic and operational scenarios.

1. EXECUTIVE SUMMARY:

The objective of the project is to build an efficient sensor based automated irrigation scheduling system. The system is driven by solar energy, by implementing the set of solar panel and the battery setup. As a preliminary step the system is divided into two modules, firstly Transmitter module and secondly Receiver module.

Transmitter module comprises of three important sections namely

1. Soil moisture sensor network comprising of Tensiometer type sensors for reading real time soil moisture conditions.

2. Microcontroller unit to analyze and compare the tensiometer readings with the preset reference values fused in the microcontroller firmware.

3. GSM section comprising of a GSM module, used to achieve communication between the transmitter module with the user.

Tensiometer type sensors are installed in the strategic locations of the field so that it covers the overall irrigation area, and water is filled to the reference mark in the tensiometer and interfaced with the microcontroller unit. Once the dry condition occurs over the irrigation field and if the dry condition is witnessed in majority of the tensiometer then the microcontroller will send request message in SMS format to the user to switch on the motor and irrigate the field through GSM module in the system.

Receiver module is the one which trips the motor, it is comprised of:

1. GSM module for receiving the commands from the user and transferring it to the microcontroller unit and for sending an acknowledgment messages to the user about the current operating condition of a motor.

2. Microcontroller unit for receiving the commands from the GSM module and acting upon it appropriately.

3. Two sensors namely Water level sensor and a power indicator modules are used to check the water and power availability to the system.

4.16x2 LCD module to display the operating condition messages of the system.

5. SPDT type relay circuit for tripping the motor according to the command sent by the microcontroller unit.

Once the transmitter module sends an request SMS to the user, As user being the master of the system, he/she must decide whether to switch on/ off the motor, Once the user wants to switch

on/off the motor, the appropriate message in the SMS is sent to the receiver section of the system. The receiver module has been programmed with three types of operational modes namely,

1. Unconditional ON/OFF mode.
2. Half an hour operational mode.
3. One hour operational mode

Once the receiver module receives the command to switch on the motor, the GSM module receives the command and transfers it to the microcontroller unit, the microcontroller unit checks for the authenticity of the message, once the message is proved to be from the authenticated user, then the microcontroller checks two criteria's namely,

1. Water availability in the reservoir.
2. Power supplied to the motor.

Once the water in the reservoir is at the acceptable level and the power supply to the motor is present then it proceeds to the next level of operation or else if any of the two conditions doesn't satisfy then the process of switching ON of the motor ends and the appropriate acknowledgement message regarding the reason for not starting the motor, whether due to insufficient water in the reservoir or the absence of power to the motor is sent to the user.

If the both the condition satisfies, then microcontroller checks for the command sent by the user, if the command is for unconditional switching ON of the motor, then the motor will be switched on .Many commands corresponding to different operational modes are fused to the firmware, the commands utilized in the project are listed below.

1	=	Unconditional switching ON of the motor.
0	=	Unconditional switching OFF of the motor.
2	=	30Mins operation of the motor.
3	=	1 hour operation of the motor

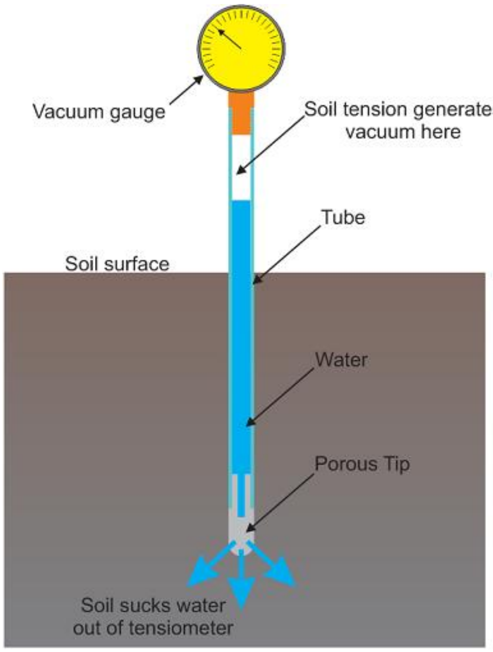
2. Materials and their Methods:-

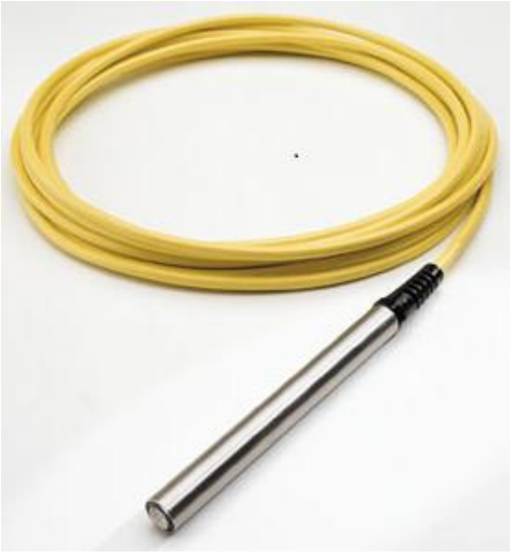
The system architecture of the project could be broadly subdivided into three sections namely,

1. Development of a sensor network.
2. PIC Microcontroller section.
3. GSM Communication interface section.

2.1 Development of a sensor network:-

The proposed PIC microcontroller has got 8 inbuilt ADC (Analog to Digital Converter) channels; and 32 digital I/O channels in the PIC16f877a controller. As per the requirement of a transmitter module, three tensiometer type sensors are interfaced using digital pins of the microcontroller and considering the receiver module of the system,


Vacuum gauge
Soil tension generates vacuum here
Tube
Soil surface
Water
Porous Tip
Soil sucks water out of tensiometer



Tensiometer Water Level Probe.

Fig.1 Sensors employed in the project.

Two sensors namely water level sensor **(WL400 WATER LEVEL SENSOR Submersible Pressure Transducer)** and a power detection sensor are used to check the water level of the reservoir and the power supplied to the motor before switching ON the motor.

2.2 PIC Microcontroller

In any embedded system, microcontroller acts as a brain of the whole system. As shown in Fig. 2, PIC16f877a is an 8-bit PIC16x series microcontroller, which possesses 8k bytes of flash program memory and 368 bytes of EEPROM data memory. The PIC16f877a microcontroller consists of 8 10-bit ADC ports, 3 Timers, 2 Serial communication modules and 15 Interrupts.

Fig.2 PIC microcontroller

All the controlling and decision were taken by the PIC microcontroller depending on the following data's,

(a) Moisture content of the soil.
(b) Water level in the storage tank.
(c) Power Supplied to the motor.
(d) The SMS received or sent by the GSM module.

Depending on these criteria's, the PIC microcontroller controls the motor operation.

2.3 GSM Module

GSM is an important and reliable technology used in wireless communication based systems. There are various kinds with different feature oriented GSM modules in the market. SIM900A module is being employed in the project, SIM900A GSM module is built with dual band GSM/GPRS based SIM900A modem from SIMCOM. It works on the frequencies 900/1800 MHz. The baud rate is configurable from 1200 to 115200 through AT (Used to communicate with GSM module) command. This is a complete GSM/GPRS module in a SMT type and designed with a very powerful single-chip processor integrating AMR926EJ-S core.

SIM900 Quad-band GSM / GPRS device, works on frequencies 850 MHZ, 900 MHZ, 1800 MHZ and 1900 MHZ. It is very compact in size and easy to use as plug in GSM Modem. The Modem is designed with 3V3 and 5VDC TTL interfacing circuitry, which allows User to directly interface with 5V Microcontrollers (PIC, AVR, Arduino, 8051, etc.) as well as 3V3 Microcontrollers (ARM, ARM Cortex XX, etc.). The baud rate can be configurable from 9600-115200 bps through AT (Attention) commands. This GSM/GPRS TTL Modem has internal TCP/IP stack to enable User to connect with internet through GPRS feature. It is suitable for SMS.

Features:

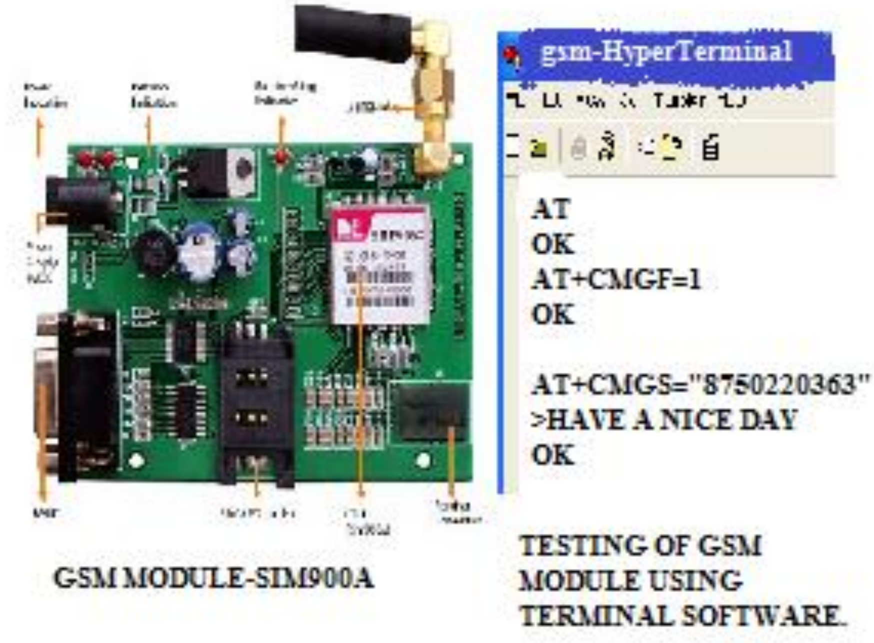
- Quad Band GSM/GPRS : 850 / 900 / 1800 / 1900 MHz
- Built in RS232 to TTL or viceversa Logic Converter (MAX232)
- Configurable Baud Rate
- SMA (SubMiniature version A) connector with GSM L Type Antenna
- Built in SIM (Subscriber Identity Module) Card holder
- Built in Network Status LED
- Inbuilt Powerful TCP / IP (Transfer Control Protocol / Internet Protocol) stack for internet data transfer through GPRS (General Packet Radio Service)
- Audio Interface Connectors (Audio in and Audio out)
- Most Status and Controlling pins are available
- Normal Operation Temperature : -20 °C to +55 °C
- Input Voltage : 5V to 12V DC
- LDB9 connector (Serial Port) provided for easy interfacing

A universal asynchronous receiver/transmitter (usually abbreviated UART) is a type of a “synchronous receiver/transmitter", a piece of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with other communication standards such as EIA RS-232. The "universal" designation indicates that the data format and transmission speeds are configurable. A UART is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port. The Universal Asynchronous Receiver/Transmitter (UART) takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. Each UART contains a shift register which is the fundamental method of conversion between serial and parallel forms. Serial transmission of digital information (bits) through a single wire or other medium is much more cost effective than parallel transmission through multiple wires. The UART usually does not directly generate or receive the external signals used between different items of equipment. Separate interface devices are used to convert the logic level signals of the UART to and from the external signaling levels. External signals may be of many different forms. Communication may be "full duplex" (both send and receive at the same time) or "half duplex" (devices take turns transmitting and receiving).

A UART usually contains the following components:

- A clock generator, usually a multiple of the bit rate to allow sampling in the middle of a bit period.
- Input and output shift registers
- Transmit/receive control
- Read/write control logic
- Transmit/receive buffers (optional)
- Parallel data bus buffer (optional)
- First-in, first-out (FIFO) buffer memory (optional)

PIC16f877A is the microcontroller used in the system. The PORTA of microcontroller is used as analog inputs. Three analog input pins AN0, AN1, AN2 are used to receive analog output of three channels of accelerometer. The RX pin of microcontroller is used to receive SMS from GSM modem through MAX232 and GPS data from GPS receiver using multiplexing technique. The TX pin of microcontroller is used to transmit data from microcontroller to web server or to our mobile phone through GPS modem using GPRS link.



```
AT
OK
AT+CMGF=1
OK

AT+CMGS="8750220363"
>HAVE A NICE DAY
OK
```

Fig.3 SIM900A GSM Module

Fig. 3 shows the schematic of the SIM900A GSM module. In simple words, GSM module is a raw form of a mobile phone, where no external peripherals like touch-screen or volume adjustment features are interfaced. In present study, the GSM module has been interfaced with the PIC microcontroller through UART communication protocol. The GSM module is operated with the help of AT commands. The working of the module is first tested with the AT commands using terminal software. Once the module works well with the terminal software and then, the module is interfaced with the PIC controller using UART (Universal Asynchronous Receiver Transmitter) communication protocol. Once the connection between the PIC microcontroller and SIM900A GSM module is established successfully, it inherently means that the communication between the user and the system is established. After that both the user and the system can interchange messages and appropriate action can be taken by the system according to the message sent by the user.

MPLAB software was used to coding. MPLAB is a proprietary freeware integrated development environment for the development of embedded applications on PIC and ds PIC microcontrollers, and is developed by Microchip Technology. MPLAB is designed to work for programming and debugging PIC microcontrollers using a personal computer. PIC Kit programmers are also supported by MPLAB.

3. METHODOLOGY:-

The overall function of the system could be best explained through the block diagram approach as shown in fig 4 and schematic diagram shown in the Fig.5.

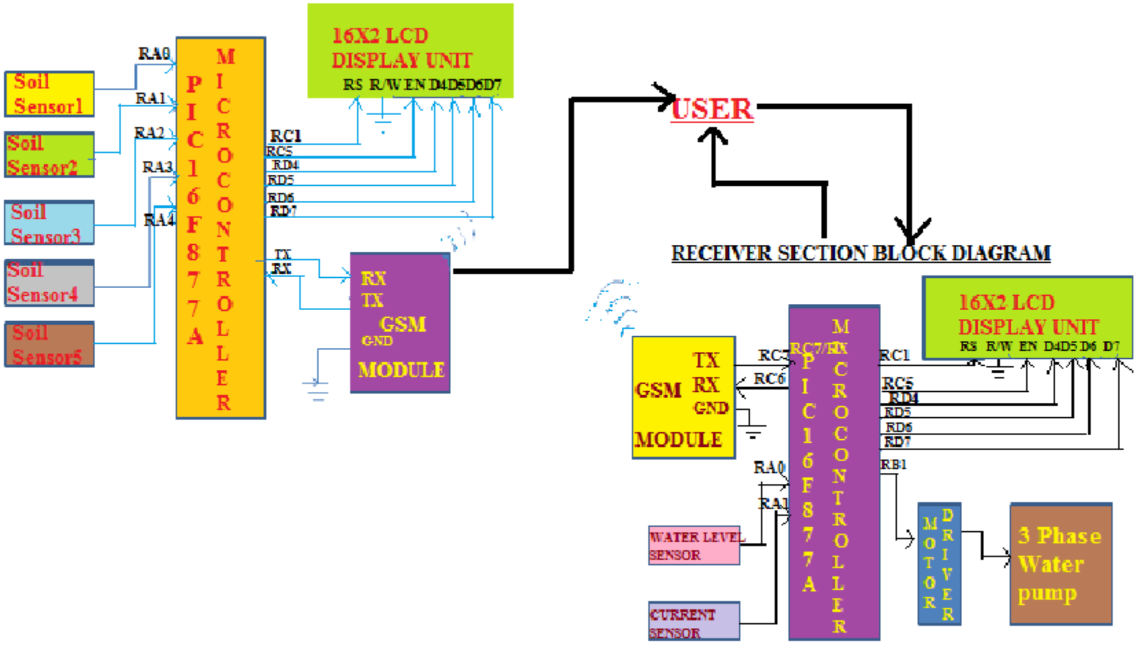


Fig 4. Block diagram Representation of the system functionality.

As shown in the schematic diagram, the system operation starts with the transmitter module sending request SMS to the user through PIC microcontroller using GSM technology and ends with the microcontroller switching ON/OFF the water pump for irrigation.

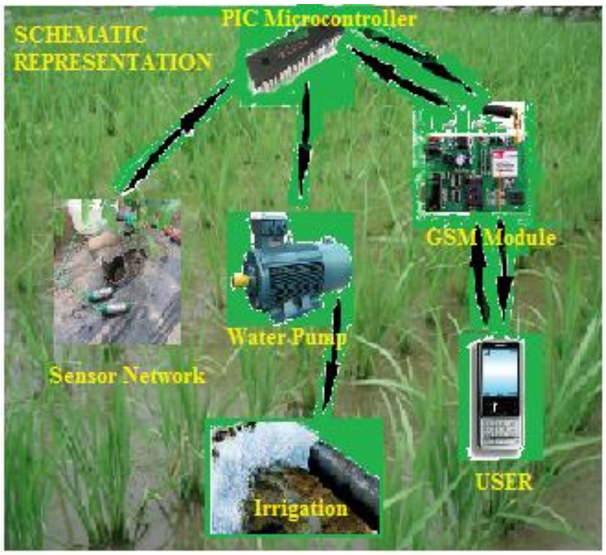


Fig. 5 Schematic Representation of Operation

As shown in the block diagram in Fig. 6, three soil moisture sensors (Tensiometer type) are fixed at the strategic locations of the irrigation field in such a way that it covers the whole field appropriately.

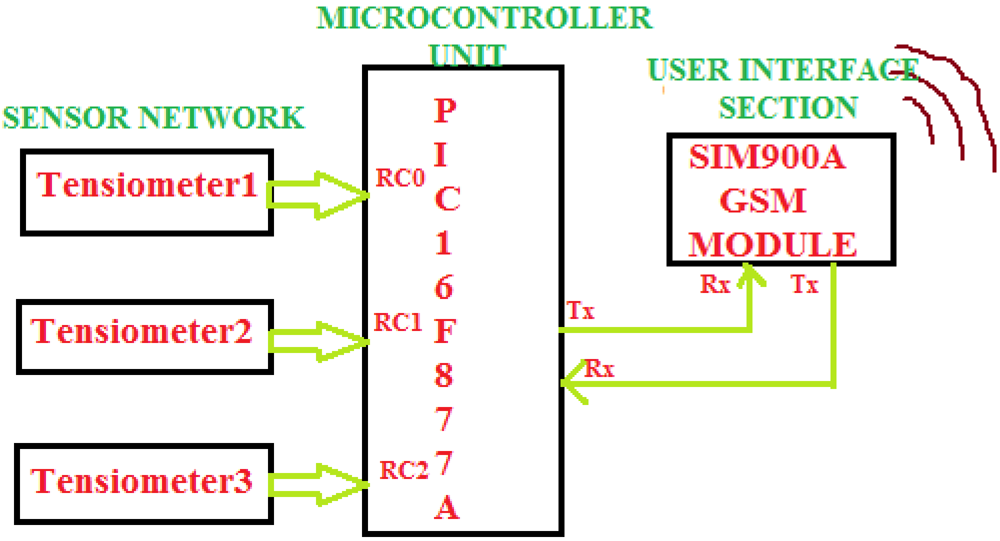
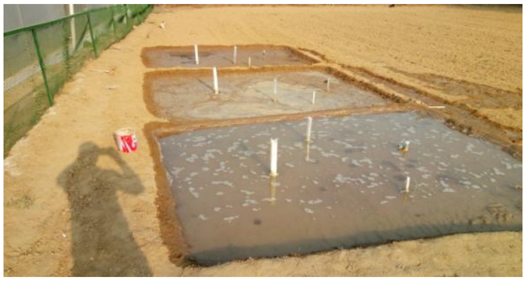
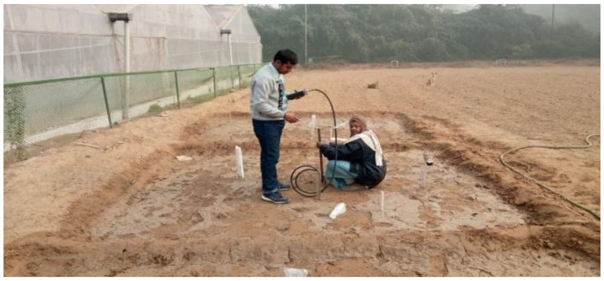
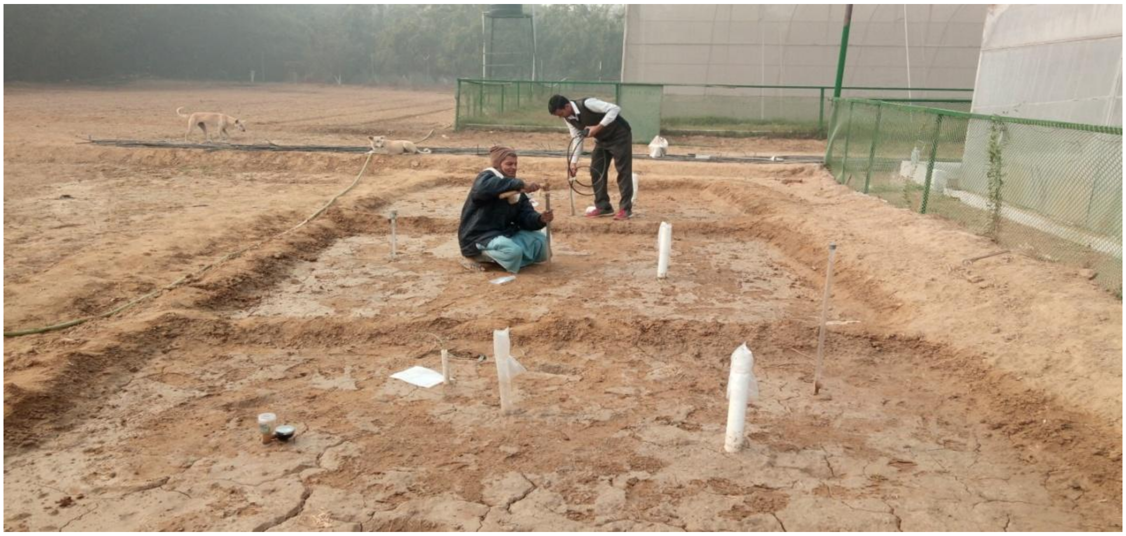


Fig. 6 Block diagram of the Transmitter section

Sensor was installed at the PFDC farm of Water Technology Centre, IARI, New Delhi (28°08′N, 77°12′E, 228.61 m above mean sea level).







As the sensors are interfaced with the PIC microcontroller, the values of the sensors will be read at the regular intervals. With the application of the programming language (C Programming Language), the threshold level of dry and wet conditions of the field are programmed and fused to the microcontroller. Once the majority i.e 2 out of 3 sensors read dry condition then the microcontroller sends a request SMS message to the user using GSM wireless technology. The operation of the transmitter module could be well explained with the help of the flow chart as shown in the fig 7 below.

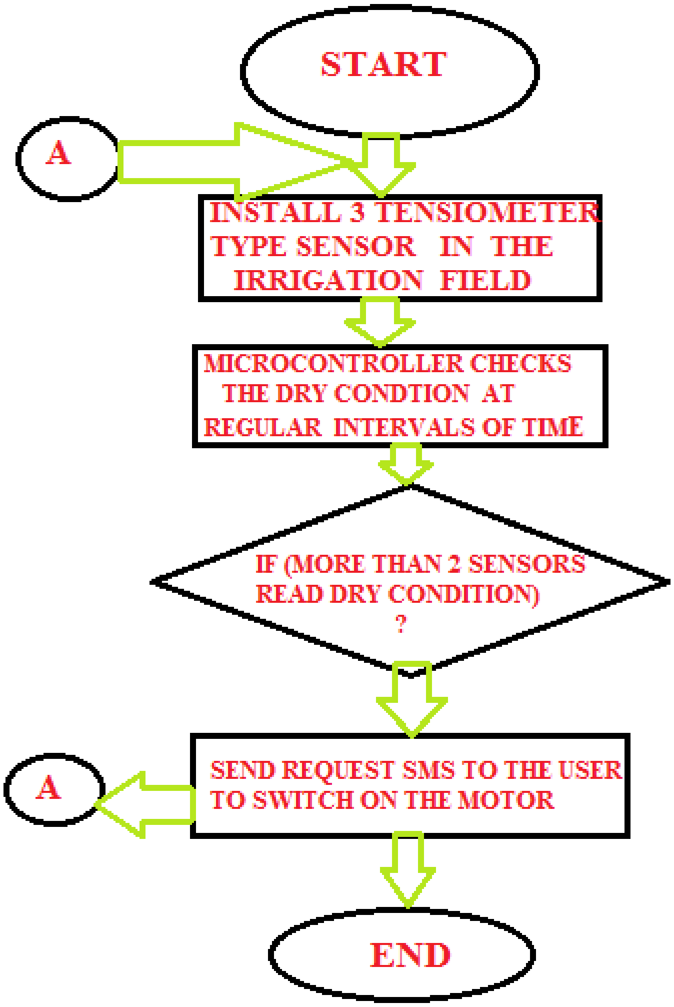

START
A
INSTALL 3 TENSIOMETER TYPE SENSOR IN THE IRRIGATION FIELD
MICROCONTROLLER CHECKS THE DRY CONDTION AT REGULAR INTERVALS OF TIME
IF (MORE THAN 2 SENSORS READ DRY CONDITION) ?
SEND REQUEST SMS TO THE USER TO SWITCH ON THE MOTOR
A
END

Fig. 7 Flowchart of transmitter section.

As the GSM module has the advantage of working with the location independent and can operate the irrigation system from anywhere under the satellite coverage area in the globe. Therefore, we can develop a reliable and efficient wireless sensor based irrigation scheduling system. The operation of the Receiver section is best explained with the flowchart as shown in the Fig. 9.

The block diagram of the receiver module is as shown in fig 8. below.

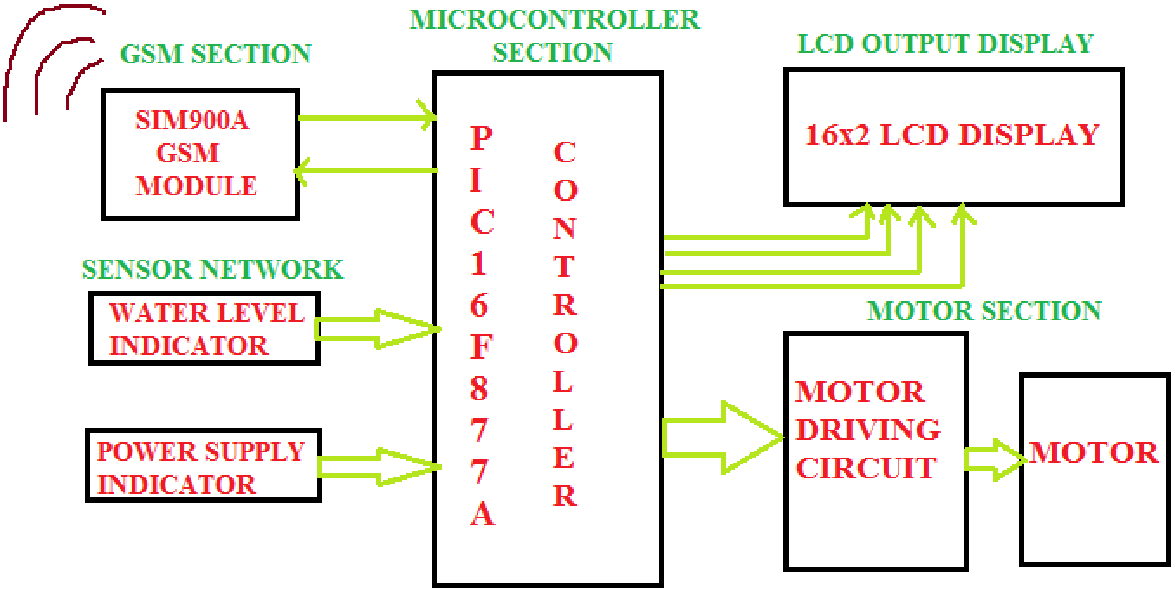


Fig.8 Block diagram of receiver section

Once the user responds to the request of the system to switch on the water pump. The GSM module receives the SMS from the user and immediately transfers it to the microcontroller. Receiving the data, the PIC microcontroller puts the received message under three main verification process namely,

(i) Check authentication of the received message.

(ii) Checking the water level in the storage tank.

(iii) The ampere rating of the power provided to the 3phase water pump.

3.1 Format Check:-

The format check on the received message is done to verify whether the SMS is sent by the authenticated user or not. Performing the format check, two significant mistakes could be avoided, first one is the misuse of the system by the unknown person and the second one is to ignore the service messages sent by the concerned network. As shown in the flowchart of the receiver section, once the format check is performed and message proved authenticated, then it is transferred to the next level of verification process.

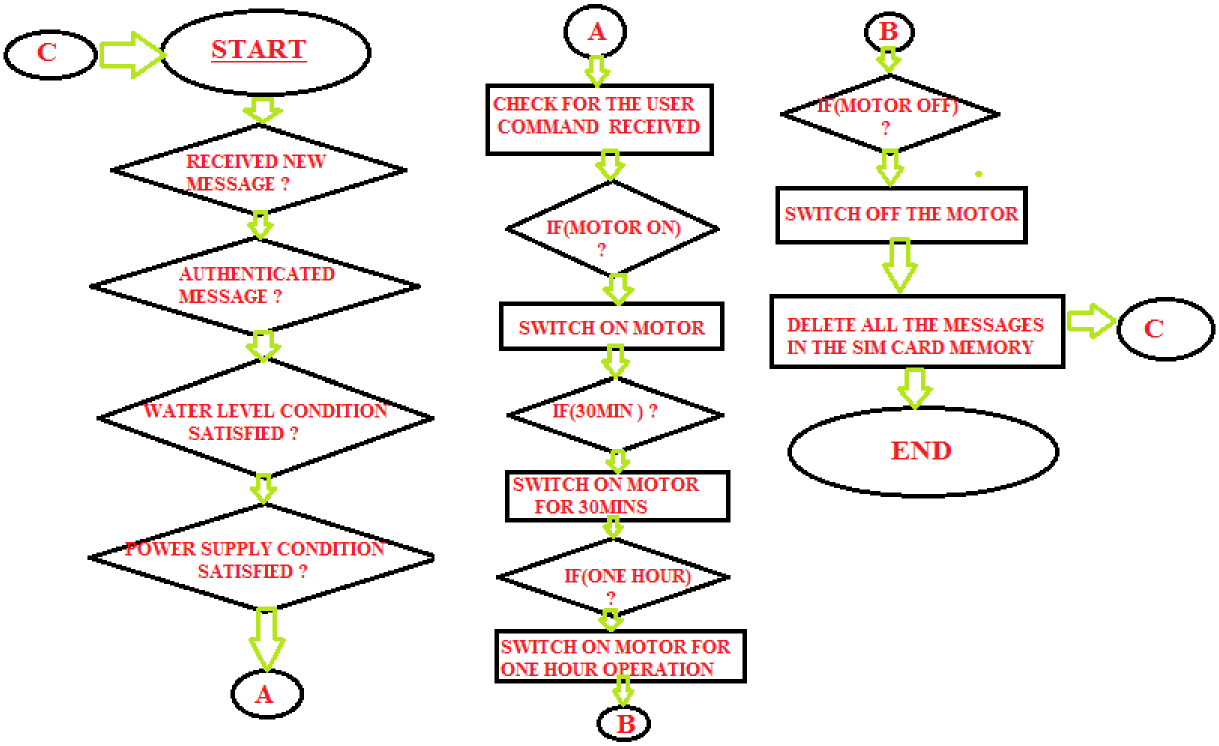

C
START
RECEIVED NEW MESSAGE ?
AUTHENTICATED MESSAGE ?
WATER LEVEL CONDITION SATISFIED ?
POWER SUPPLY CONDITION SATISFIED ?
A
A
CHECK FOR THE USER COMMAND RECEIVED
IF(MOTOR ON) ?
SWITCH ON MOTOR
IF(30MIN) ?
SWITCH ON MOTOR FOR 30MINS
IF(ONE HOUR) ?
SWITCH ON MOTOR FOR ONE HOUR OPERATION
B
B
IF(MOTOR OFF) ?
SWITCH OFF THE MOTOR
DELETE ALL THE MESSAGES IN THE SIM CARD MEMORY
C
END

Figure.9 flowchart of Receiver Section

3.2 Checking the water level in the storage tank:-

This is one of the significant checks that the PIC microcontroller does before leading to the motor operation, if the motor is operated with insufficient water in the storage tank; it leads to the dry running and damage of the water pump. Therefore, checking the water level is considered as one of the important verification process in the present study.

3.3 Checking the power supplied to the water pump:-

As the system is solar power driven so the system is capable of checking the presence of the power supply to the motor even in the absence of the AC power supply. This test makes user be aware of the power supply status of his/her irrigation field. This is done by taking an tapping from the phase line of the power supply line of the motor, and stepping it down to 12v ac power using a step down transformer and converting it into a dc power of 5v using a wheatstone bridge and an voltage regulation circuit and this phase line is provided to the input channel of the microcontroller and if the power is available then the input channel will be ‘high’ and ‘low’ in the absence of the power. So checking the status of input channel of the microcontroller, the power supply status of the irrigation field is found. In the process of switching ON the motor, the process is continued only when the both water level and the power supply condition are satisfied, if either of the condition fails, then the process gets stopped abruptly and the acknowledgement message along with the reason is messaged to the user.

Once the received message from the user passes all the verification process and proves to be an authenticated message, then the microcontroller checks the command sent by the user. The PIC microcontroller checks the command and operates the motor according to the command passed by the user. According to the firmware code fused, three operational modes are programmed namely:

1. Unconditional Switch ON/OFF of the motor.
2. Half an hour operational mode of the motor.
3. One hour operational mode of the motor.

The motor in the irrigation field is operated according to the specific command according to the required mode of the operation sent by the user. The appropriate acknowledgement messages are assigned along with the operation of the motor. Once completing the operation then the microcontroller get back to the initial stage and start checking for the new message.

4. Developmental Stages of the System:-

4.1 Development of transmitter module:-

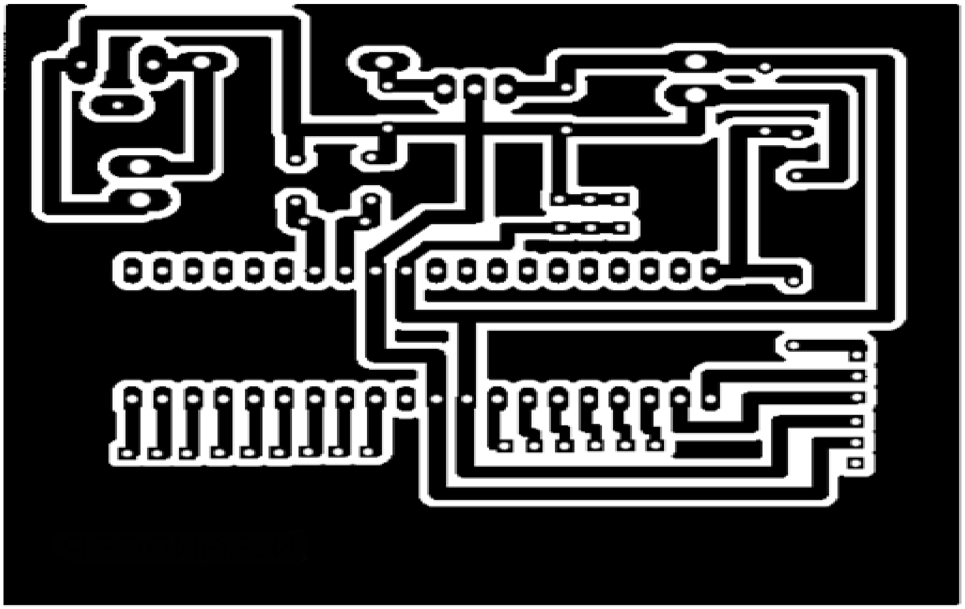


Fig 10. PCB Layout Design of a Transmitter module

Once the functionality of the system is tested proven in the ready-made microcontroller boards then our objective was to build our own PCB boards indigenously. The first step towards it was to design a hardware PCB layout of the system. The transmitter module is designed according to the requirement of having a sensor network along with microcontroller and GSM communication sections. The PCB layout of transmitter module is designed as shown in the fig.10.

4.2 Development of a Receiver module:-

Before starting with the hardware PCB layout design of the receiver module, the components and their functionalities were checked and finalized. All the sensors which are to be employed in the system are interfaced and checked their operations using the dotted boards and using simulation software. Then the dimensions and component placements were decided by making schematic designs using the Eagle designing software tool. Once all the pre-requisite works are completed at the satisfactory level, then the PCB designing of the receiver module are carried out, after different levels of correction stages, the final PCB layout been finalized. The final receiver module PCB design is as shown in fig.11.

Once the PCB layout is finalized, then in the course of board development, following steps are followed:

1. The one sided copper boards which contains of copper surface on one side and the plastic surface on the other side, which are used in building the prototype systems.
2. The PCB layout is printed in the laser printer and fixed on the copper side of the board.
3. In view of pasting the toner part of the print on the copper side, the copper side was heated using an iron box for around 20-30 minutes.
4. Once the toner part of the design is well fixed on the copper surface, then the board is dipped in the water for around 15mins.
5. Then the board is taken out of the water and the paper part was peeled off leaving the toner part unaffected.
6. The etching solution containing of 100grams of Ferric chloride + 500ml of water is prepared in the non metallic container.

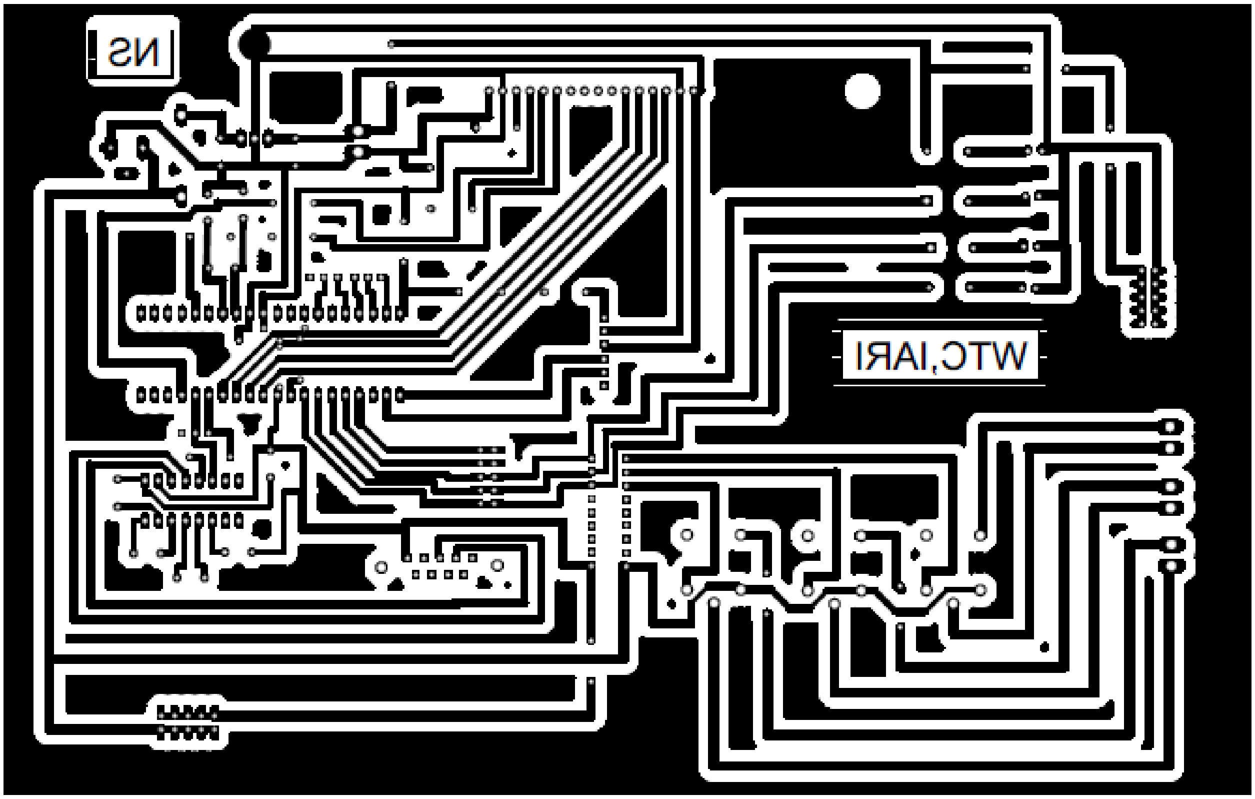


Fig 11.The PCB layout of the proposed system is developed

7. Then the board is dipped in the ferric chloride solution, as the ferric chloride is an efficient etching agent, which etches off all the copper part of the board leaving the toner part unaffected.

8. Once the board is taken off the solution we have the board with only the toner part in it.

9. Then the toner part is removed using water and metallic scrubber.

10. Then on the copper side of the PCB board, we now have only the tracks of our layout in copper and remaining in plastic.

11. And then the drilling works on the board are carried out and the respective tested components are placed and soldered.

12. Then again the functionality of the board is tested and the tested systems are fixed in the irrigation field.

13. The prototype system is as shown in fig12.

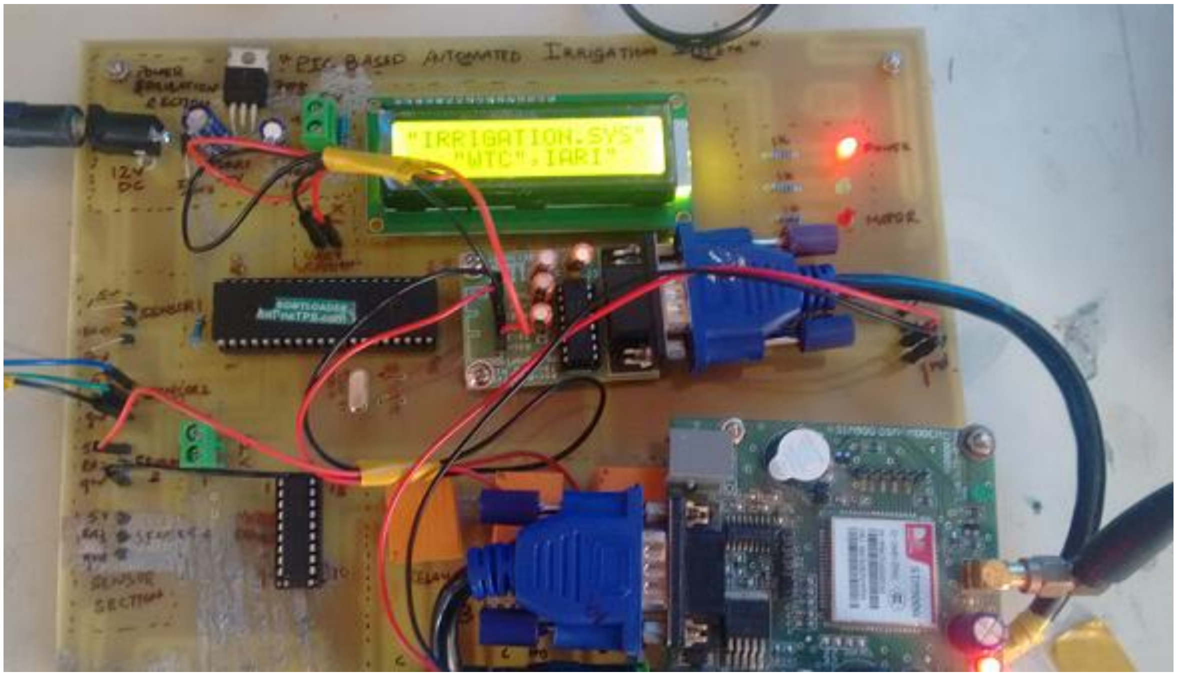


Fig12.Working Receiver Prototype Model.

14. Then to install and check the functionality of the system, the appropriate cabinets and installing stands were made as shown in the fig 13.

15. First the dimensions of the systems like solar panel, receiver PCB board, and the cabinet for batter setup are noted down.

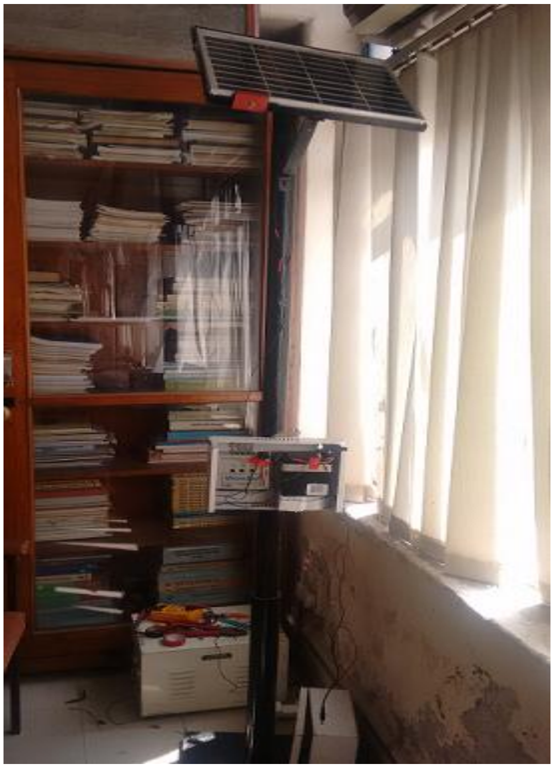
16. The cabinet for the receiver module are made of the transparent fiber sheet so that the operation of the module can be seen and also be rigid.

17. For placing the solar panels, the MS sheets are employed and the aluminum pole is used instead of iron pole to reduce the weight of the stand.

18. Separate iron basement is been designed and fixed to the field surface with the help of iron rods injected through the drilled holes of the stand, as the system attached to the aluminum pole is not permanently fixed to the iron basement ,so that the system could be separately taken off for the service purposes.

Stage 1.

Stage 2.



Stage 3.

Stage 4.

Stage 5 **Stage 6.**

Fig .13 Developmental Stages of the system

19. Once the system is well fixed with the cabinet setup, the system is fixed in the field and tested its functionality.

5. Conclusion:-

The proposed automation system is capable of filling the technical gaps of the previous agricultural automation systems like the user had got no control over the motor operation for irrigation of crop. With the application of the GSM technology user can have the control over his / her irrigation and water pump can start from anywhere under the satellite coverage area in the globe. GSM based wireless automated irrigation system overcomes many milestones and obstacles and proves to be a reliable irrigation system for automation domain.

The proposed system can save a lot of water and manpower. The system can achieve more food production with the limited usage water and man power. With the application of the proposed project, obstacles like lack of manpower and water scarcity could be strategically handled and achieve a good food production and in-turn can make agriculture a real backbone of our Indian economy.

5. Field tests:

5.1 Transmitter module:

As shown in the fig10, the transmitter module is built and installed in the field and in view of checking the functionality of the sensors, the tensiometer is fixed in the field and readings are taken at regular intervals of time as shown in the figure 14.

Date	10th May 2016	11th May 2016	12th May 2016	13th May 2016	18th May 2016	23th May 2016	24th May 2016	25th May 2016	26th May 2016	27th May 2016	28th May 2016	30th May 2016	31st May 2016
Tensio-meter Readings	20	20	21	21	24	26	26	26	27	27	27	30	30

Fig 14. Tabulation of the tensiometer readings.

The behavior of the tensiometer fixed in the field is graphically represented as in figure 15.

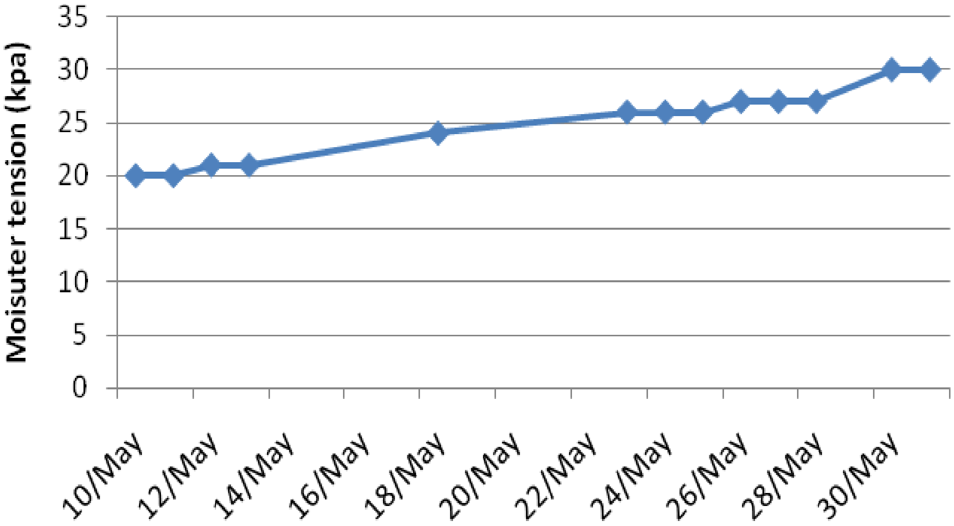
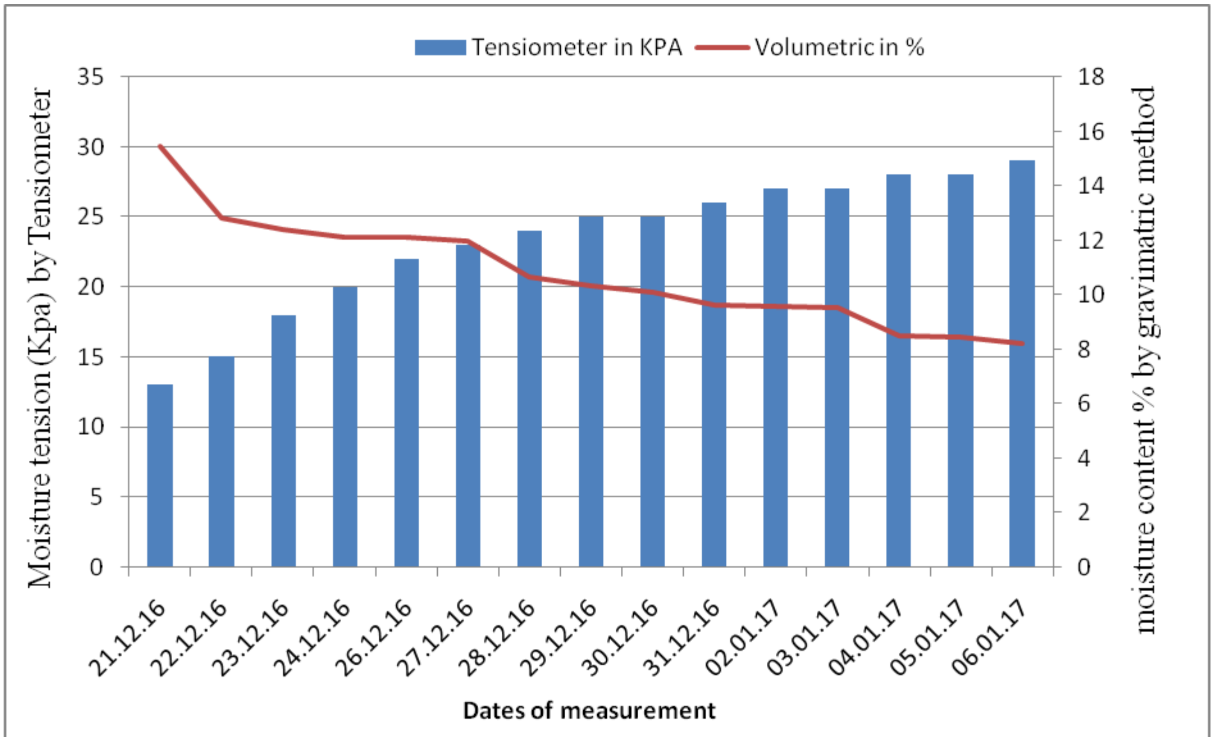


Fig 15.Graphical Representation of tensiometer sensor

The unit of reading tensiometer in kpa. Water level sensor was used to measure water level and in this there is no need to convert pressure gauge reading. This sensor automatic detects the pressure developed in the tensiometer.

Yes tensiometer gives matric potential having vacuum gauge. We have not used this vacuum gauge as reading of gauge reading is very tedious. We have modified the tensiometer and removed this gauge. Drop in water level inside the tube was measure by using the level sensor. This device was selected for Innovative Irrigation Show organized by Irrigation Association in

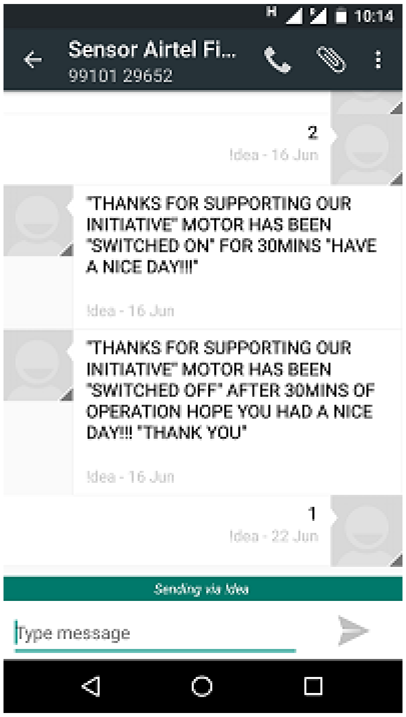
USA. First regular tensiometer was tested and moisture content was determined against tensiometer kpa. After this, relationship was drawn against drop in water level and moisture content. Program coding was written to send the message against the drop in water level which ultimately shows the moisture content for making decision about the time of irrigation. Cost of regular tensiometer varies from 12000 to 15000 per unit and using the tensiometer enhances the cost of the system.



Calibration of soil moisture tension with gravimatric method

5.2 Receiver module:

As shown in the fig 12. The system is developed and installed in the field and tested its functionality for the period of 3 months. The system proves to be working well under various climatic and different operational modes. As you can notice the modes of operation in the below screenshots, As in the first screenshot, the 30 mins operation of motor is tested, the results are as follows, As i send the command namely numeric constant ‘2’ command which corresponds to the 30mins operation fused to the firmware , the receiver module received the message and switch on the motor for the 30 mins operational mode, and corresponding acknowledgement is sent from the receiver module saying the motor switched on for the duration of 30mins and at the end of the prescribed duration of 30 mins, the receiver module switches off the motor automatically and sends an intimation message to the user saying the motor is switched off successfully, so that the user can be unambiguous over the current operating status of his motor.

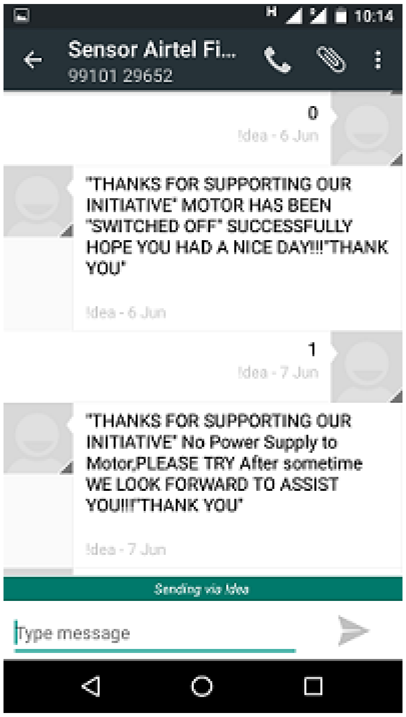


2
Idea - 16 Jun

"THANKS FOR SUPPORTING OUR INITIATIVE" MOTOR HAS BEEN "SWITCHED ON" FOR 30MINS "HAVE A NICE DAY!!!"
Idea - 16 Jun

"THANKS FOR SUPPORTING OUR INITIATIVE" MOTOR HAS BEEN "SWITCHED OFF" AFTER 30MINS OF OPERATION HOPE YOU HAD A NICE DAY!!! "THANK YOU"
Idea - 16 Jun

1
Idea - 22 Jun



0
Idea - 6 Jun

"THANKS FOR SUPPORTING OUR INITIATIVE" MOTOR HAS BEEN "SWITCHED OFF" SUCCESSFULLY HOPE YOU HAD A NICE DAY!!!"THANK YOU"
Idea - 6 Jun

1
Idea - 7 Jun

"THANKS FOR SUPPORTING OUR INITIATIVE" No Power Supply to Motor,PLEASE TRY After sometime WE LOOK FORWARD TO ASSIST YOU!!!"THANK YOU"
Idea - 7 Jun

Fig 16. Screenshots of switching on the motor through SMS.

As you can notice in the second screenshot, the motor operating mode is tested, as pre planned, we cut the supply to the motor and tried switching on the motor unconditionally by sending the corresponding command '1' to the receiver module, As the receiver module receives the authenticated message, as the water level condition is satisfied, then it moved over to checking of the power supply to the motor, once it identified the absence of the power supply to the motor, it stops the verification process and goes back to the starting of the program without switching on the motor and the appropriate acknowledgement message is sent and the reason for the failure of the operation is sent to the user as shown in the second screenshot of the figure below.

Bill of materials

Following is the list of components required to build receiver module:-

PIC16f877A -1
Power adaptor 12V/2A-1 OR Transformer (12V/2A)
GSM module (SIM-900A)-1
12V voltage regulator-1
Diode IN4007
22pf capacitor
11.0592MHz crystal-1
Max-232-1
10uf capacitor-5
Db9 connector-1
Uln2003a-1
Resistor-1k -5
Relay -3
Lcd 16x2-1
7805 voltage regulator-1
40 pin IC base-1
18 pin IC base
Led (different colors)-5
104 capacitor-1
Copper clad-1

Cost

The main motive of this project is to reduce the cost of this project to bare minimum so that it will be easily affordable by the farmers. The cost of Operation for sim is Rs. 45 per month as it required message pack only for operations this cost may be vary with the plan of service provider. The total cost of the receiver module is 1800/-.

PROGRAMMING CODES

1. **Display unit header file:**

```
********************************************************************************
***************LCD OUTPUT DEVICE INTERFACING FUNCTIONS**************
********************************************************************************
void Lcd4_Port(char a)
{
    if(a & 1)
        D4 = 1;
    else
        D4 = 0;
    if(a & 2)
        D5 = 1;
    else
        D5 = 0;
    if(a & 4)
        D6 = 1;
    else
        D6 = 0;
    if(a & 8)
        D7 = 1;
    else
        D7 = 0;
}
void Lcd4_Cmd(char a)
{
    RS = 0;          // => RS = 0
    Lcd4_Port(a);
    EN  = 1;          // => E = 1
 __delay_ms(4);
 EN  = 0;         // => E = 0
}
/*Command for clear screen*/
Lcd4_Clear()
{
    Lcd4_Cmd(0);
    Lcd4_Cmd(1);
}
/*Function to set the curser at the specified bit of the LCD*/
void Lcd4_Set_Cursor(char a, char b)
{
    char temp,z,y;
    if(a == 1)
    {
     temp = 0x80 + b;
        z = temp>>4;
```

```
                y = (0x80+b) & 0x0F;
                Lcd4_Cmd(z);
                Lcd4_Cmd(y);
        }
        else if(a == 2)
        {
                temp = 0xC0 + b;
                z = temp>>4;
                y = (0xC0+b) & 0x0F;
                Lcd4_Cmd(z);
                Lcd4_Cmd(y);
        }
}

/*LCD  initialization function*/
void Lcd4_Init()
{
Lcd4_Port(0x00);
  __delay_ms(20);
 Lcd4_Cmd(0x03);
        __delay_ms(5);
 Lcd4_Cmd(0x03);
        __delay_ms(11);
 Lcd4_Cmd(0x03);
 /////////////////////////////////////////////////////
 Lcd4_Cmd(0x02);
        Lcd4_Cmd(0x02);
 Lcd4_Cmd(0x08);
        Lcd4_Cmd(0x00);
        Lcd4_Cmd(0x0C);
 Lcd4_Cmd(0x00);
 Lcd4_Cmd(0x06);
}
/*Function to write characters on LCD*/
void Lcd4_Write_Char(char a)
{
  char temp,y;
  temp = a&0x0F;
  y = a&0xF0;
         RS = 1;         // => RS = 1
  Lcd4_Port(y>>4);          //Data transfer
         EN = 1;
         __delay_ms(5);
         EN = 0;
         Lcd4_Port(temp);
         EN = 1;
```

```
             __delay_ms(5);
             EN = 0;
}
/*Function to write the string on the LCD*/
void Lcd4_Write_String(char *a)
{
            int i;
            for(i=0;a[i]!='\0';i++)
             Lcd4_Write_Char(a[i]);
}
void Lcd4_Shift_Right()
{
            Lcd4_Cmd(0x01);
            Lcd4_Cmd(0x0C);
}
void Lcd4_Shift_Left()
{
            Lcd4_Cmd(0x01);
            Lcd4_Cmd(0x08);
}
void lcd4_putch(char a)
{
            RS = 1;             // write characters
            Lcd4_Write_Char( a);
}
/*Function to write the string on the LCD*/
void lcd4_puts(const char * a)
{
            RS = 1;             // write characters
            while(*a)
                        Lcd4_Write_Char(*a++);
                        //lcd_write(*s++);
}
/*Delay function used  in the LCD header functions*/
void lcd4_delayms(unsigned int k)
{
            int j,m,l;
            for(l=0;l<=k;l++)
            {
                        for(m=0;m<=29;m++)
                        {
                                    for(j=0;j<=10;j++)
                                    {           }
                        }
            }
}
```

```
*******************************************************************************
*******************END OF LCD HEADER FILE ********************************
*******************************************************************************
```

2. Transmitter Module Code:

```
*******************************************************************************
****************TRANSMITTER MODULE CODE STARTS***********************
*******************************************************************************
/*Preprocessor Directives*/
#define _XTAL_FREQ 11059200
#include<pic.h>
#include<stdio.h>
#include<stdlib.h>
#include <htc.h>
#include <string.h>
#define BAUDRATE 9600
/*User defined functions Declaration*/
void interrupt ISR(void);
void UART_Init(void);
void UART_Write(const unsigned char);
unsigned char UART_Read(void);
void UART_Write_Text(const unsigned char*);
void delaySec(unsigned int sec);
void sendSMS();
unsigned char uart_rd[20], k = 0, m = 0;
void delay(unsigned int k);
void recsms();
void sendSMS();
void sendSMS1();
void sendSMS2();
char lookup(int);
void sensor1();
void sensor();
void sensor2();
void sensor3();
void sensor4();
/*Variable declaration section*/
char x[65];
char y[69];
char UART_Data_Ready();
char UART_Read();
void UART_Write_Text(const unsigned char *uartString);
int atoi(const char*s);
__CONFIG_3F31;
unsigned  p,q,a=0x100,z;
  float tmpr,tmpr1,r,f;
   unsigned char del;
```

```
/*User Defined functions Initializations*/
void interrupt ISR(void)
{
if(RCIF)
{
if(OERR)
{
CREN = 0;
CREN = 1;
}
uart_rd[m] = RCREG;
if(uart_rd[m] == '"'){
k++;
m = 0;
}
if(k == 6){
uart_rd[m] = '\0';
k = 0;
m = 0;
}
}
}
/*GSM initialization function*/
void UART_Init(void)
{
TRISC = 0xC0;
TXSTA = 0x24;
RCSTA = 0x92;
//SPBRG = 0x67;// Fosc = 16 MHz
SPBRG = 129 ;  //Fosc = 20 MHz;
}
/*GSM write function*/
void UART_Write(unsigned char data)
{
while(!TXIF);
TXREG = data;
}
/*Checking the microcontroller buffer */
char UART_Data_Ready()
{
  return RCIF;
}
/*UART read function*/
char UART_Read()
{
if(OERR)
```

```
{
CREN = 0;
CREN = 1;
}
  while(!RCIF);  //while(RCIF) didnt work as expected with led testing
  return RCREG;
}
/*UART string receiving function*/
void UART_Read_Text(char *Output, unsigned int length)
{
        int i;
        for(int i=0;i<length;i++)
                Output[i] = UART_Read();
}

/*unsigned char UART_Read(void)
{
if(OERR)
{
CREN = 0;
CREN = 1;
}
while(!RCIF)
return RCREG;
} */
/*UART string writing function*/
void UART_Write_Text(const unsigned char *uartString)
{
while(*uartString)
UART_Write(*uartString++);
}
/*Delay functions*/
void delaySec(unsigned int sec){
unsigned int i = 0;
for(i = 0; i < sec; i++){
__delay_us(200);
__delay_us(200);
__delay_us(200);
__delay_us(200);
__delay_us(200);
}
}
void delay(unsigned int k)
{
 int j,m,l;
for(l=0;l<=k;l++)
```

```
 {
  for(m=0;m<=29;m++)
   {
    for(j=0;j<=10;j++)
    {}
  }
 }
}
/*Function to send request SMS to the user regarding the dry condition*/
void sendSMS(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(500);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(500);
UART_Write_Text("AT+CMGS=\"9868060359\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(500);
UART_Write_Text("FIELD IS DRY, PLEASE SWITCH ON THE  MOTOR");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(1000);
}
/*Main function starts*/
void main()
{
//unsigned char data;
TRISC = 0xFF;
TRISB = 0x00;
//UART_Init();
delay(500);
GIE = 1;
PEIE = 1;
//RCIE = 1;
//CREN=1;FERR=0;OERR=1;
char i,c;
int j,k,l;
delay(500);
while(1)
{
/*Checking dry condition of the soil*/
if((RC0==0&&RC1==1&&RC2==1)|| (RC0==1&&RC1==0&&RC2==1)|| (
RC0==1&&RC1==1&&RC2==0)&&(RC0==1&&RC1==1&&RC2==1))
{
          RB1==1;   //DRY INDICATION LED
          sendSMS();   //SENDING REQUEST MESSAGE TO THE USER TO SWITCH ON
THE MOTOR
}
   }
     }
***********************************************************************************
********************TRANSMITTER MODULE CODE ENDS********************
```

3.**RECEIVER MODULE CODE:**

```
***********************************************************************************
** ***************RECEIVER MODULE WORKING CODE***********************
***********************************************************************************
/*Objective: To switch ON/OFF the motor receiving command from the user.
  Author Institute:  Water Technology Center, ICAR-IARI.
 Description: 1.The firmware fused code is designed in a such a way to accept the commands of
only the authenticated user and programmed to work in three different modes namely,
     (i)  Unconditional ON/OFF the motor.
     (ii) Half an hour  operational mode.
     (iii)One hour operational mode.
               Command codes used for the respective operations are as follows,
               1    =  Unconditional Switching ON of the motor.
               2    = Half an hour operation of the motor.
               3    = One hour operation of the motor.
               0    = Unconditional Switching OFF of the motor. */
***************************Preprocessor directives*****************************
#define _XTAL_FREQ 11059200   // operating frequecy required for using standard delay
function
#include<pic.h>
#include<stdio.h>
#include<stdlib.h>
#include <htc.h>
#include <string.h>
#define BAUDRATE 9600
******************** User Defined Functions************************************
void interrupt ISR(void);
/*UART communication functions*/
void UART_Init(void);
void UART_Write(const unsigned char);
unsigned char UART_Read(void);
void UART_Write_Text(const unsigned char*);
char UART_Data_Ready();
char UART_Read();
```

```
void UART_Write_Text(const unsigned char *uartString);


*********************GSM communication function decleration********************
void recsms();
void sendsandyucon();
void sendsandyucoff();
void sendsandy30on();
void sendsandy30off();
void sendsandy1hron();
void sendsandy1hroff();
void sendsandyet();
void sendnpucon();
void sendnpucoff();
void sendnp30on();
void sendnp30off();
void sendnp1hron();
void sendnp1hroff();
void sendnpet();
void sendnppr();
void sendsandypr();
/*Delay functions but mainly lcd4_delayms() function and __delay_ms() std functions are used
in the code*/
void delaySec(unsigned int sec);
void delay(unsigned int k);
void cal(unsigned char*phoneno);
/*Display function*/
void disp(float num);
void disp1(float num);
/*decimal to Ascii conversion table*/
char lookup(int);
******************************************************************************
********************Variable decleration section**********************************
unsigned char uart_rd[20], k = 0, m = 0;
char data;
float s;
char x[65];
char y[69];
unsigned  p,q,a=0x100,z;
  float tmpr,tmpr1,r,f;
   unsigned char del;
/*std ascii to integer function which is not used in the code*/
int atoi(const char*s);
**************************LCD pins assignment*******************************
#define RS RC5
#define EN RC1
```

```
#define D4 RD4
#define D5 RD5
#define D6 RD6
#define D7 RD7
/*Including lcd header file in the same directory*/
#include "lcd.h"
/*Function to read water level*/
void sensor();
/*configuration bit selection*/
__CONFIG_3F31;
/*Interrupt Function*/
void interrupt ISR(void)
{
if(RCIF)
{
if(OERR)
{
CREN = 0;
CREN = 1;
}
uart_rd[m] = RCREG;
if(uart_rd[m] == ""){
k++;
m = 0;
}
if(k == 6){
uart_rd[m] = '\0';
k = 0;
m = 0;
}
}
}
/*UART communication detail functions*/
void UART_Init(void)
{
TRISC = 0xC0;
TXSTA = 0x24;
RCSTA = 0x92;
//SPBRG = 0x67;// Fosc = 16 MHz
SPBRG = 129 ;  //Fosc = 20 MHz;
}
/*GSM function to send the data to the user*/
void UART_Write(unsigned char data)
{
while(!TXIF);
TXREG = data;
```

```
}
char UART_Data_Ready()
{
   return RCIF;
}
/*GSM read function*/
char UART_Read()
{
if(OERR)
{
CREN = 0;
CREN = 1;
}
  while(!RCIF);  //while(RCIF) didnt work as expected with led testing
  return RCREG;
}
/*GSM read function*/
void UART_Read_Text(char *Output, unsigned int length)
{
            int i;
            for(int i=0;i<length;i++)
                        Output[i] = UART_Read();
}
/*unsigned char UART_Read(void)
{
if(OERR)
{
CREN = 0;
CREN = 1;
}
while(!RCIF)
return RCREG;
} */

/*GSM string writing function*/
void UART_Write_Text(const unsigned char *uartString)
{
while(*uartString)
UART_Write(*uartString++);
}
/*Delay functions*/
void delaySec(unsigned int sec){
unsigned int i = 0;
for(i = 0; i < sec; i++){
__delay_us(200);
__delay_us(200);
__delay_us(200);
__delay_us(200);
__delay_us(200);
}
}
void delay(unsigned int k)
{
 int j,m,l;
for(l=0;l<=k;l++)
 {
  for(m=0;m<=29;m++)
   {
    for(j=0;j<=10;j++)
    {}
  }
 }
}
/*Display function to display water level on lcd*/
void disp1(float num)
{
int x1,x2,x3,x4,n,temp;
 static float f1;
char ch1,ch2,ch3,ch4,dot='.';
//char P,Q,R,S,dot='.';
n=num;
temp=num;
x1=n%10;
n=n/10;
x2=n%10;
n=n/10;
x3=n%10;
f1=num-temp;
x4=f1*10;
ch1=lookup(x3);
ch2=lookup(x2);
ch3=lookup(x1);
ch4=lookup(x4);
//lcd_goto(0);
Lcd4_Clear();
Lcd4_Set_Cursor(1,0);
lcd4_putch('W');
lcd4_putch('A');
lcd4_putch('T');
lcd4_putch('E');
lcd4_putch('R');
lcd4_putch('_');
```

```
lcd4_putch('L');
lcd4_putch('V');
lcd4_putch('L');
lcd4_putch(':');
lcd4_putch(ch2);
lcd4_putch(ch3);
lcd4_putch(dot);
lcd4_putch(ch4);
//lcd4_puts("Amps");
 delay(1000);
}
/*Function to read water level in the reservoir*/
void sensor()
{
ADCON0=0x89;    //CH1
 __delay_ms(5);
 ADCON1=0x84;
 for (del=20;del--;);
 ADGO=1;
 while (ADGO);
 p=ADRESH;
 q=ADRESL;
 PIR1=PIR1&~64;  //ADC interrupt flag is cleared
 z=p*a+q;       //combined 10 bit number formation
 f=z;           //unsigned to float conversion
 r=(f*5)/1023;    //equivalent voltage calculation from ADC output
 tmpr1=(r*100);      //sensor output voltage to degree Celsius conversion
 delay(1000);
disp1(tmpr1);      //call of disp(float) to display the temperature
 delay(200);
}
/*Lookup table for Ascii conversion*/
char lookup(int no)
{
char cha;
switch(no)
{
case 0:
cha='0';
break;
case 1:
cha='1';
break;
case 2:
cha='2';
break;
case 3:
cha='3';
break;
case 4:
cha='4';
break;
case 5:
cha='5';
break;
case 6:
cha='6';
break;
case 7:
cha='7';
break;
case 8:
cha='8';
break;
case 9:
cha='9';
break;
}
return cha;
}
/*SMS functions*/
void recsms(){
UART_Write_Text("AT+CMGR=1");
UART_Write(0x0D);
UART_Write(0x0A);
delay(2000);
}
/*SMS codes for sending to Dr.Neelam patel*/
/*SMS corresponds for unconditional switching ON*/
void sendnpucon(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"9868060359\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
```

```
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED ON\"SUCCESSFULLY");
UART_Write_Text("\"HAVE A NICE DAY!!!\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}

/*SMS corresponds for unconditional switching off*/
void sendnpucoff(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"9868060359\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED OFF\" SUCCESSFULLY ");
UART_Write_Text(" HOPE YOU HAD A NICE DAY!!!\"THANK YOU\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}

/*Empty tank sms*/
void sendnpet(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"9868060359\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
```

```
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" YOUR WATER RESERVOIR IS \"EMPTY\" PLEASE FILL IT AND
TRY AGAIN ");
UART_Write_Text(" WE LOOK FORWARD TO ASSIST YOU!!!\"THANK YOU\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}
/*SMS to intimate the switching on of the motor for 30MINS*/
void sendnp30on(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"9868060359\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED ON\" FOR 30MINS");
UART_Write_Text("\"HAVE A NICE DAY!!!\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}

/*SMS to intimate that the motor has been switched off after the duration of 30MINS*/
void sendnp30off(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"9868060359\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
```

```
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED OFF\" AFTER 30MINS OF
OPERATION");
UART_Write_Text(" HOPE YOU HAD A NICE DAY!!! \"THANK YOU\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}

/*SMS to intimate sandy number for switching on motor for one hour*/
void sendnp1hron(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"9868060359\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED ON\"FOR THE DURATION OF 1
HOUR");
UART_Write_Text("\"HAVE A NICE DAY!!!\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}

/*SMS to intimate sandy number for switching off motor after one hour*/
void sendnp1hroff(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"9868060359\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED OFF\"AFTER 1 HOUR OF OPERATION");
UART_Write_Text(" HOPE YOU HAD A NICE DAY!!!\"THANK YOU\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}

/*Checking the AC power*/
void sendnppr(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"9868060359\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" No Power Supply to Motor,PLEASE TRY After sometime ");
UART_Write_Text(" WE LOOK FORWARD TO ASSIST YOU!!!\"THANK YOU\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}

/*SMS codes for sending to sandeep*/
/*SMS corresponds for unconditional switching on*/
void sendsandyucon(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
```

```
UART_Write_Text("AT+CMGS=\"8750220363\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED ON\"SUCCESSFULLY");
UART_Write_Text("\"HAVE A NICE DAY!!!\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}

/*SMS corresponds for unconditional switching off*/
void sendsandyucoff(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"8750220363\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED OFF\" SUCCESSFULLY ");
UART_Write_Text(" HOPE YOU HAD A NICE DAY!!!\"THANK YOU\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}
/*Empty tank SMS*/
void sendsandyet(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"8750220363\"");
```

```
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" YOUR WATER RESERVOIR IS \"EMPTY\" PLEASE FILL IT AND
TRY AGAIN ");
UART_Write_Text(" WE LOOK FORWARD TO ASSIST YOU!!!\"THANK YOU\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}
/*SMS to intimate the switching on of the motor for 30MINS*/
void sendsandy30on(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"8750220363\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED ON\" FOR 30MINS");
UART_Write_Text(" \"HAVE A NICE DAY!!!\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}
/*SMS to intimate that the motor has been switched off after the duration of 30MINS*/
void sendsandy30off(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"8750220363\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED OFF\" AFTER 30MINS OF OPERATION");
UART_Write_Text(" HOPE YOU HAD A NICE DAY!!! \"THANK YOU\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}
/*SMS to intimate sandy number for switching on motor for one hour*/
void sendsandy1hron(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"8750220363\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text("MOTOR HAS BEEN \"SWITCHED ON\"FOR THE DURATION OF 1 HOUR");
UART_Write_Text("\"HAVE A NICE DAY!!!\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);}
/*SMS to intimate sandy number for switching off motor after one hour*/
void sendsandy1hroff(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"8750220363\"");
UART_Write(0x0D);
UART_Write(0x0A);
```

```
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" MOTOR HAS BEEN \"SWITCHED OFF\"AFTER 1 HOUR OF
OPERATION");
UART_Write_Text(" HOPE YOU HAD A NICE DAY!!!\"THANK YOU\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}
/*Checking the AC power*/
void sendsandypr(){
UART_Write_Text("AT");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGF=1");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("AT+CMGS=\"8750220363\"");
UART_Write(0x0D);
UART_Write(0x0A);
lcd4_delayms(2000);
UART_Write_Text("\"THANKS FOR SUPPORTING OUR INITIATIVE\"");
UART_Write_Text(" No Power Supply to Motor,PLEASE TRY After sometime ");
UART_Write_Text(" WE LOOK FORWARD TO ASSIST YOU!!!\"THANK YOU\"");
UART_Write(0x0D);
UART_Write(0x0A);
UART_Write(0x1A);
lcd4_delayms(4000);
}

********************************************************************************
************************User defined functions ends********************************

/*Main function starts*/
void main()
{
//unsigned char data;
ADCON1=0x84;
TRISA=0xCF;
TRISC = 0xC0;
TRISB = 0x00;
TRISD = 0x00;
PORTC = 0x00;
```

```
PORTD = 0x00;
PORTB = 0x00;

UART_Init();
Lcd4_Init();
delay(500);
GIE = 1;
PEIE = 1;
//RCIE = 1;
//CREN=1;FERR=0;OERR=1;
char i,c;
int j,k,l;

delay(6000);
RCREG=0;
UART_Write_Text("AT+CNMI=2,1,0,0,0");
UART_Write(0x0D);
UART_Write(0x0A);

/*Initial lines written on the LCD screen*/
Lcd4_Clear();
Lcd4_Set_Cursor(1,0);
lcd4_putch('"');
lcd4_putch('I');
lcd4_putch('R');
lcd4_putch('R');
lcd4_putch('I');
lcd4_putch('G');
lcd4_putch('A');
lcd4_putch('T');
lcd4_putch('I');
lcd4_putch('O');
lcd4_putch('N');
lcd4_putch('.');
lcd4_putch('S');
lcd4_putch('Y');
lcd4_putch('S');
lcd4_putch('"');
Lcd4_Set_Cursor(2,3);
lcd4_putch('"');
lcd4_putch('W');
lcd4_putch('T');
lcd4_putch('C');
lcd4_putch('"');
lcd4_putch(',');
lcd4_putch('I');
```

```
lcd4_putch('A');
lcd4_putch('R');
lcd4_putch('I');
lcd4_putch("");
RCREG=0;

while(1)
{
if(RCIF)                        // Checking whether the message is received.
{
for(i=0;i<=10;i++)
{
x[i]=UART_Read();            // Collecting the received message in the array.
}
}

Lcd4_Clear();
Lcd4_Set_Cursor(1,0);              //  setting the cursor at the first bit of the first line
lcd4_putch(x[4]);
lcd4_putch(x[5]);
lcd4_putch(x[6]);
lcd4_putch(x[7]);

if(x[4]=='C'||x[4]=='+')                  // format checking.
{
lcd4_putch('-');
lcd4_putch('R');
lcd4_putch('E');
lcd4_putch('A');
lcd4_putch('D');
lcd4_putch('Y');
__delay_ms(10);

 RCREG = 0;  // Great job sandy.
__delay_ms(15);

 UART_Write_Text("AT+CMGR=1");    // reading the first message in the sim memory.
 UART_Write(0x0D);
 UART_Write(0x0A);

 for(i=0;i<=69;i++)                      //collecting the first message in the array of the microcontroller
 {
  y[i]=UART_Read();
 }
   RB2=1;
```

```
__delay_ms(15);
    Lcd4_Set_Cursor(2,1);
    lcd4_putch(y[65]);
            lcd4_putch(y[66]);
            lcd4_putch(y[67]);
            lcd4_putch(y[68]);
            lcd4_putch('N');
    lcd4_putch('S');

if(y[66]=='1')                          //checking the command of operation
{

  sensor();      //water level sensor-RA0
      if(tmpr1>25)    //checking the sufficiency of water level
  {
   if(RA2==1)      //checking the ac power supply for motor.
     {
            Lcd4_Set_Cursor(2,1);
            lcd4_putch('P');
            lcd4_putch('W');
            lcd4_putch('R');
            lcd4_putch('.');
            lcd4_putch('P');
    lcd4_putch('S');
            lcd4_putch('T');
      RB1=1;  //motor on
      sendnpucon();      //sms Dr. Neelam patel  about unconditional switching on of the motor
      sendsandyucon();    //sms to sandeep about unconditional switching on of the motor
      Lcd4_Clear();
      Lcd4_Set_Cursor(1,1);
      lcd4_putch('M');
      lcd4_putch('O');
      lcd4_putch('T');
      lcd4_putch('O');
      lcd4_putch('R');
       lcd4_putch('_');
      lcd4_putch('S');
      lcd4_putch('T');
      lcd4_putch('A');
      lcd4_putch('R');
       lcd4_putch('T');
         }
   else
    {
      sendnppr();
      sendsandypr();
```

```
        Lcd4_Clear();
        Lcd4_Set_Cursor(1,0);
        lcd4_puts("No Power");
        Lcd4_Set_Cursor(2,0);
        lcd4_puts("Try Aftr Smetme");
          }
  }
else
 {
        sendnpet();
        sendsandyet();
        Lcd4_Clear();
        Lcd4_Set_Cursor(1,0);
        lcd4_puts("Tank Empty");
                Lcd4_Set_Cursor(2,0);
                lcd4_puts("Please fill");
        }
}

else if(y[66]=='0')          //checking whether command is to switch off the motor.
{
RB1=0;
sendnpucoff();
sendsandyucoff();
Lcd4_Clear();
Lcd4_Set_Cursor(1,1);
lcd4_putch('M');
lcd4_putch('O');
lcd4_putch('T');
lcd4_putch('O');
lcd4_putch('R');
lcd4_putch('_');
lcd4_putch('S');
lcd4_putch('T');
lcd4_putch('O');
lcd4_putch('P');
}

else if(y[66]=='2')              //Command for 30 Mins operation.
{
   sensor();
        if(tmpr1>25)
   {
    if(RA2==1)
    {
    Lcd4_Set_Cursor(2,1);
```

```
lcd4_putch('P');
 lcd4_putch('W');
 lcd4_putch('R');
 lcd4_putch('.');
 lcd4_putch('P');
 lcd4_putch('S');
 lcd4_putch('T');
    RB1=1;
        sendnp30on();
        sendsandy30on();
        Lcd4_Clear();
        Lcd4_Set_Cursor(1,0);
        lcd4_putch('M');
        lcd4_putch('.');
        lcd4_putch('S');
        lcd4_putch('R');
        lcd4_putch('T');
        lcd4_putch('-');
        lcd4_putch('3');
        lcd4_putch('0');
        lcd4_putch('M');
        lcd4_putch('I');
        lcd4_putch('N');
        lcd4_putch('S');
        for(i=0;i<30;i++)      //30 Mins operational mode.
        {
        for(j=0;j<6000;j++){
        __delay_ms(10);
        }
        }
        RB1=0;
        sendnp30off();
        sendsandy30off();
        Lcd4_Clear();
        Lcd4_Set_Cursor(1,0);
        lcd4_putch('M');
        lcd4_putch('.');
        lcd4_putch('S');
        lcd4_putch('T');
        lcd4_putch('P');
        lcd4_putch('-');
        lcd4_putch('3');
        lcd4_putch('0');
        lcd4_putch('M');
        lcd4_putch('I');
        lcd4_putch('N');
```

```
         lcd4_putch('S');
         }
   else
    {
   sendnppr();
   sendsandypr();
         Lcd4_Clear();
   Lcd4_Set_Cursor(1,0);
   lcd4_puts("No Power");
   Lcd4_Set_Cursor(2,0);
   lcd4_puts("Try Aftr Smetme");
         }
  }
         else
         {
         sendnpet();
   sendsandyet();
         Lcd4_Clear();
   Lcd4_Set_Cursor(1,0);
   lcd4_puts("Tank Empty");
   Lcd4_Set_Cursor(2,0);
   lcd4_puts("Please fill");
         }
}

else if(y[66]=='3')                  //command for One hour operation
{
   sensor();
         if(tmpr1>25)
    {
    if(RA2==1)
     {
      Lcd4_Set_Cursor(2,1);
                  lcd4_putch('P');
                  lcd4_putch('W');
                  lcd4_putch('R');
                  lcd4_putch('.');
                  lcd4_putch('P');
      lcd4_putch('S');
                  lcd4_putch('T');
         RB1=1;
 //      sendnp1hron();
//       sendsandy1hron();
         Lcd4_Clear();
         Lcd4_Set_Cursor(1,0);
         lcd4_putch('M');
```

```
            lcd4_putch('.');
            lcd4_putch('S');
            lcd4_putch('R');
            lcd4_putch('T');
            lcd4_putch('-');
            lcd4_putch('1');
            lcd4_putch('.');
            lcd4_putch('H');
            lcd4_putch('O');
            lcd4_putch('U');
            lcd4_putch('R');
            for(i=0;i<60;i++)      //Delay of One hour operation
            {
            for(j=0;j<6000;j++){    //1 min(60x100=6000)
            __delay_ms(10);
            }
            }
            RB1=0;
//          sendnp1hroff();
//          sendsandy1hroff();
            Lcd4_Clear();
            Lcd4_Set_Cursor(1,0);
            lcd4_putch('M');
            lcd4_putch('.');
            lcd4_putch('S');
            lcd4_putch('T');
            lcd4_putch('P');
           lcd4_putch('-');
            lcd4_putch('1');
            lcd4_putch('.');
            lcd4_putch('H');
            lcd4_putch('O');
            lcd4_putch('U');
            lcd4_putch('R');
            }
     else
     {
   // sendnppr();
  //  sendsandypr();
            Lcd4_Clear();
    Lcd4_Set_Cursor(1,0);
    lcd4_puts("No Power");
    Lcd4_Set_Cursor(2,0);
    lcd4_puts("Try Aftr Smetme");
            }
    }
```

```
            else
            {
  //  sendnpet();
  //  sendsandyet();
            Lcd4_Clear();
    Lcd4_Set_Cursor(1,0);
    lcd4_puts("Tank Empty");
    Lcd4_Set_Cursor(2,0);
    lcd4_puts("Please fill");
            }
   }
}
if(OERR)
{
CREN = 0;
CREN = 1;
}
UART_Write_Text("AT+CMGDA=\"DEL ALL\"");
UART_Write(0x0D);
UART_Write(0x0A);
delay(200);
RCREG=0;
delay(200);
RB2=0;
RB3=0;
}       }
```

****
**********************RECEIVER CODE ENDS*******************************

Suggestion and future scope

The project is aimed to design and implement an “Indigenous Automated Micro-Irrigation System” for agricultural fields. After doing different tests and programming different codes, eventually the obliged outcome is put forward. It is a fast and efficient approach to control the irrigation by setting different timers depending on the water level. This whole system is driven by the most popular GSM Technology. To make use of these whole system farmers should be capable of typing messaging on mobile phones. So to make the easy access of this technology this whole system can be made application based. By doing this farmers can switch on/off the motor by just one click on mobile phones and can set timers also by just single click.

As we know that agricultural lands are located at remote locations, so there are chances that the mobile network is not available at certain time and at certain places. So this will create problem to the whole system. To tackle with this, the whole system can be implemented by using wireless wifi technology. At last, this whole system can be made mobile application based so that farmers can easily make use of this technology for irrigation purpose.

Signature of Principal Investigator   Signature of Head of Department