COLLEGE OF ENGINEERING AND TECHNOLOGY, GHATIKIA, BHUBANESWAR

DEPARTMENT OF ELECTRONICS AND INSTRUMENTATION

MICROPROCESSORS AND MICROCONTOLLERS PROJECT REPORT ON

AUTOMATIC IRRIGATION SYSTEM

Submitted to:

Ms. Suman Bala Behera
Mr. Sidharth Das

SUBMITTED BY: GROUP-5

SOUMYASHREE RANJAN JENA	1901106544
SOURAV KUMAR MAHJHI	1901106545
SOURAV KUMAR SWAIN	1901106546
SHRUTI SARLA MAHAPATRA	1901106548
SUBHALAXMI SAHOO	1901106549

ACKNOWLEDGEMENT

It gives immense pleasure in bringing out this synopsis of the project entitled “AUTOMATIC IRRIGATION SYSTEM”.

The completion of this project could not have been possible without the participation and assistance of a lot of individuals contributing to this project. However, we would like to express our deep appreciation and indebtedness to the Head of Department, Instrumentation And Electronics, Dr.Sribatsa Behera sir, our teachers, Ms.SumanBalaBehera ma’am and Mr.Sidharth Das sirwhose insightful leadership and knowledge benefited us to complete this project within the prescribed time.

Last but not the least, we would like to thank everyone who is involved in the project directly or indirectly.

CONTENTS

INTRODUCTION

The world has evolved by the human beings starting from agriculture revolution starting from 10000 years back till date with advancement of industrial revolution 4.0. In the process of evolution, the world population reaps the benefits of science and technology in a different era. But there are vagaries associated with socio-economic advancement like water, air, noise, nuclear pollution, etc.

An increase in the human population and sizable diminishing of flora and fauna is exerting massive pressure on mother earth. As a result, pandemics broke out regularly, climate change is leading to enormous damage to mother earth. There is a forecast by the world economic forum that there will be drought and food shortage for about 100 million people.

Water is a very precious resource, only available on earth. Therefore, water management and distribution are very much essential to tackle the grave situation in near future. Here comes the science and technology which will regulate water demand and distribution. Agriculture has to be revamped with available water resources so that we can be able to feed 7 billion mouths which are increasing day by day.

PROBLEM STATEMENT

The project involves a blending of science and technology into the economy and livelihood of many. Due to climatic changes, there are many challenges to agriculture. Limitations in water have affected farmers worldwide. As the economy of many countries depends on agriculture, there is an urgent need for water management and distribution.

Earlier farmers had to be resourceful due to low rainfalls. They irrigated their lands to overcome water deficiency. But irrigation systems are inefficient as they waste about half of the irrigated water due to evaporation, wind, or runoff (which also causes soil erosion). To avoid this wastage of water and also to protect the earth, we need a much more efficient method of irrigating the land.

The project looks into developing an automated irrigation system that could be controlled through a mobile application. This system will work to control and save water and electricity, increase agricultural production using small quantities of water and minimize manual intervention in watering operations with increasing watering speed.

Benefits of Automated Irrigation System:

- Reduced labor
- Timely irrigation – plants being watered when needed
- Management of higher flow rates
- Accurate cut-off of water compared to manual checking
- Reduced runoff of water and nutrients
- Reduced costs for vehicles used to check irrigation.

In the traditional method of irrigation, farmers usually dig canals from the main water supply channel else carry buckets of water to their fields. This causes a delay in irrigation time and water is also wasted during the trip. Sometimes the canals may overflow with water else there can be a lack of water, both of which cause crop damage and leads to food deficiency and low economy.

Our idea is to check the water content of soil and accordingly water the plants so that there can be effective management and minimum wastage of water. The excess supply of water, soil damage and, lack of water supply can be managed effectively so that there is maximum yield.

PROJECT APPROACH

Materials Used:

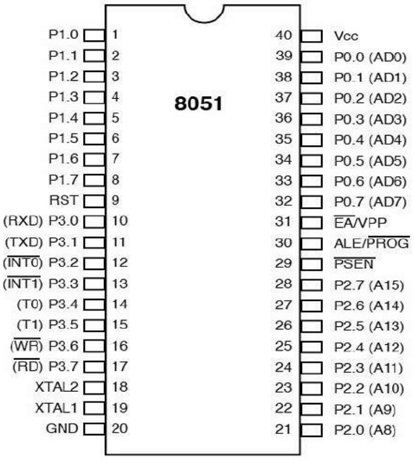
COMPONENT	QUANTITY
8051 Microcontroller	1
LCD 16 X 2	1
Motor	1
Power Supply (5v)	3
Ground	2
ADC 0808 (8BIT)	4
Proteus 8	_
Keil uVision	_
Potentiometer	1
Relay Switch	1

1) 8051 Microcontroller:

8051 microcontroller is **an 8-bit microcontroller** created in 1981 by Intel Corporation. It has an 8-bit processor that simply means that it operates on 8-bit data at a time. It is among the most popular and commonly used
microcontroller. The 8051-assembly language programming is **based on the memory registers.** If we want to manipulate data to a processor or controller by performing subtraction, addition, etc., we cannot do that directly in the memory, but it needs registers to process and to store the data.

An 8051 microcontroller has the following 12 major components:

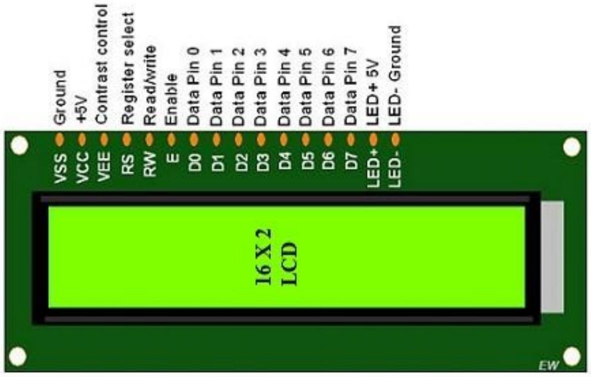
- ALU (Arithmetic and Logic Unit)
- PC (Program Counter)
- Registers.
- Timers and counters.
- Internal RAM and ROM.
- Four general purpose parallel input/output ports.
- Interrupt control logic with five sources of interrupt.
- Serial date communication.



- **Pins 1 to 8** – These pins are known as Port 1. This port doesn’t serve any other functions. It is internally pulled up, bi-directional I/O port.

- **Pin 9** – It is a RESET pin, which is used to reset the microcontroller to its initial values.
- **Pins 10 to 17** – These pins are known as Port 3. This port serves some functions like interrupts, timer input, control signals, serial communication signals RxD and TxD, etc.
- **Pins 18 & 19** – These pins are used for interfacing an external crystal to get the system clock.
- **Pin 20** – This pin provides the power supply to the circuit. • **Pins 21 to 28** – These pins are known as Port 2. It serves as I/O port. Higher order address bus signals are also multiplexed using this port.
- **Pin 29** – This is PSEN pin which stands for Program Store Enable. It is used to read a signal from the external program memory.
- **Pin 30** – This is EA pin which stands for External Access input. It is used to enable/disable the external memory interfacing.
- **Pin 31** – This is ALE pin which stands for Address Latch Enable. It is used to demultiplex the address-data signal of port.
- **Pins 32 to 39** – These pins are known as Port 0. It serves as I/O port. Lower order address and data bus signals are multiplexed using this port.
- **Pin 40** – This pin is used to provide power supply to the circuit.

2) 16 X 2 LCD :

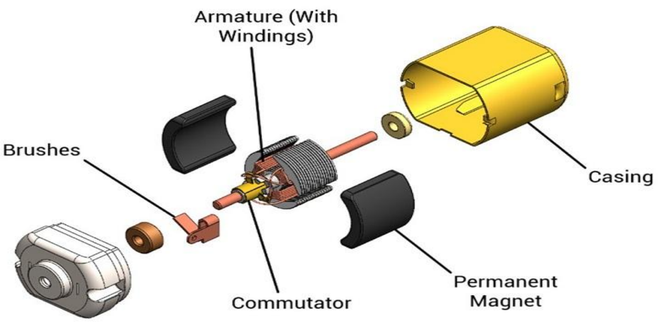


A 16x2 LCD means it **can display 16 characters per line and there are 2 such lines.** In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. The **LCD controller** uses RS and RW lines along with E to operate the LCD. Resister Select (RS): Determines whether a command (RS = 0) is sent (to set up the display) or actual data (RS=1) is sent. Read/Write RW=0; writes to the LCD. RW=1; Reads from the LCD. **The E (enable) bit tells the LCD when it should read the data lines.**

All of these bits need to be connected to different pins on the board.

3) DC Motor

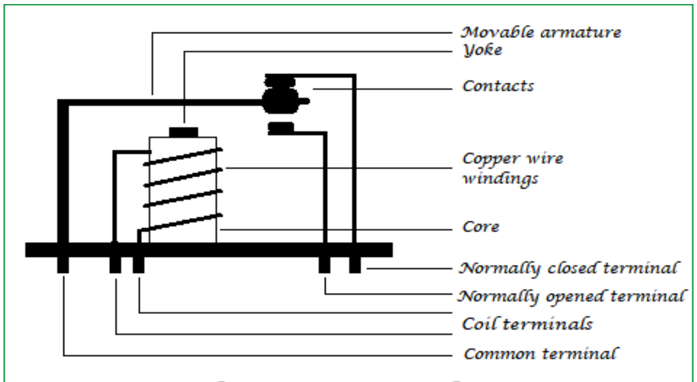
When a current-carrying conductor is placed in a magnetic field, it experiences a torque and has a tendency to move.
In other words, when a magnetic field and an electric field interact, a mechanical force is produced. The **DC motor** or **direct current motor** works on that principle. This is known as motoring action.
The direction of rotation of this motor is given by Fleming's left hand rule, which states that if the index finger, middle finger, and thumb of your left hand are extended mutually perpendicular to each other and if the index finger represents the direction of the magnetic field, middle finger indicates the direction of the current, then the thumb represents the direction in which force is experienced by the shaft of the **DC motor**.



4) Relay Switch

A Relay is an electromechanical device that can be used to make or break an electrical connection. It consists of a flexible moving mechanical part which can be controlled electronically through an electromagnet, basically, a relay is just like a mechanical switch but you can control it with an electronic signal instead of manually turning it on or off.

Construction of Relay



An Electromagnetic Relay

Electromagnet: An Electromagnet plays a major role in the **working of a relay**. It is a metal which doesn’t have magnetic property but it can be converted into a magnet with the help of an electrical signal. We know that when current passes through the conductor it acquires the properties of a magnet. So, when a metal winded with a copper wire and driven by the sufficient power supply, that metal can act as a magnet and can attract the metals within its range.

Movable Armature: A movable armature is a simple metal piece which is balanced on a pivot or a stand. It helps in making or breaking the connection with the contacts connected to it.

Contacts: These are the conductors that exist within the device and are connected to the terminals.

Yoke: It is a small metal piece fixed on a core in order to attract and hold the armature when the coil is energized.

Spring (optional): Few relays don’t need any spring but if it is used, it is connected to one end of the armature to ensure its easy and free movement. Instead of a spring, a metal stand like structure can be used.

4) ADC 0808

ADC0808 is an 8 bit **analog to digital converter** with eight input analog channels, *i.e.*, it can take eight different analog inputs. The input which is to be converted to digital form can be selected by using three address lines. The voltage reference can be set using the Vref+ and Vref- pins. The step size is decided based on set reference value. Step size is the change in analog input to cause a unit change in the output of ADC. The default step size is 19.53mV corresponding to 5V reference voltage. **ADC0808** needs an external clock to operate unlike ADC0804 which has an internal clock. The ADC needs some specific control signals for its operations like start conversion and bring data to output pins. When the conversion is complete the EOC pins goes low to indicate the end of conversion and data ready to be picked up.

Pin Description:

Pin No	Function	Name
1	Analog input pins	IN3
2		IN4
3		IN5
4		IN6
5		IN7
6	Start conversion; input pin; a low to high pulse is given	SC
7	End of conversion; output pin; goes low when the conversion is over	EOC
8	Digital output bit 4	D3
9	Input pin; a low to high pulse brings data to output pins from the internal registers at end of conversion	Output enable
10	Clock input; to provide external clock	Clock input
11	Supply voltage; 5V	Vcc
12	Positive reference voltage	Vref(+)
13	Ground ()v)	GND
14	Digital output bit	D1
15		D2
16	Negative reference voltage	Vref(-)
17	Digital output bits	D0
18		D4

19		D5
20		D6
21		D7
22	Address latch enable; Input pin; low to high pulse is required to latch in the address	ALE
23	Address lines	AddressC
24		AddressB
25		AddressA
26	Analog inputs	IN0
27		IN1
28		IN2

WORKING

The idea of the project is to implement an automatic irrigation system by sensing the moisture of the soil. The working of the circuit is as follows.

The soil moisture sensor is inserted in the.Depending on the quality of the sensor, it must be inserted near the roots of the plant. The soil moisture sensor measures the conductivity of the soil.

Wet soil will be more conductive than dry soil. The soil moisture sensor module has a comparator in it.

The voltage from the prongs and the predefined voltage are compared and the output of the comparator is high only when the soil condition is dry.

Due to the unavailability of the soil moisture sensor library in Proteus Simulation Software we have used 10k ohm potentiometer.

This output from the soil moisture sensor is given to the analogue input pin (Port -1) of the microcontroller. The microcontroller continuously monitors the analogue input pin.

When the moisture in the soil is above the threshold, the microcontroller displays a message mentioning the same and the motor is off.

When the output from the soil moisture sensor is high i.e. the moisture of the soil is less. This will trigger the microcontroller and displays an appropriate message on the LCD and the output of the microcontroller, which is connected to the base of the transistor is high.

When the transistor is turned on, the relay coil gets energized and turns on the motor.

When the moisture of the soil reaches the threshold value, the output of the soil moisture sensor is low and the motor is turned off.

The system is also designed to warn when the moisture is very high than the threshold and the soil is too wet, which is dangerous for the plant.

Program Code

HEADER FILE FOR LCD

```
#ifndef _LCD_H
#define _LCD_H
void LCD_init();
void LCD_Cmdwrite(unsigned char);
void LCD_Datawrite(unsigned char);
void LCD_Clear();
void LCD_GoToLINEone();
void LCD_GoToLINEtwo();
void LCD_GoToXY(char , char);
void LCD_DisplayString(unsigned char *);
void LCD_DisplayNumber(unsigned int);
#endif
```

HEADER FILE FOR ADC

```
#ifndef _ADC_H
#define _ADC_H
void ADC_init();
unsigned char ADC_StartConversion(char);
#endif
```

HEADER FILE FOR DELAY

```
#ifndef _DELAY_H
#define _DELAY_H
void delay_us(unsigned int);
void delay_ms(unsigned int);
void delay_sec(unsigned char);
#endif


#include<reg51.h>

#include "lcd.h"
#include "delay.h"


void delay_us(unsigned int us_count)
 {
    while(us_count!=0)
      {
         us_count--;
       }
   }

       void delay_ms(unsigned int ms_count)
 {
        while(ms_count!=0)
         {
            delay_us(112);   //delay_us is called to generate 1ms delay
             ms_count--;
            }
   }

       void delay_sec(unsigned char sec_count)
 {


    while(sec_count!=0)
     {
        delay_ms(1000);    //delay_ms is called to generate 1sec delay
        sec_count--;
       }
 }
#include<reg51.h>
#include "lcd.h"
#include "delay.h"

#define databus  P0        // LCD DATABUS IS CONNECTED TO PORT 0

sbit rs = P2^0;        //  REGISTER SELECT IS CONNECTED TO PORT 2.0
sbit rw = P2^1;       //  READ / WRITE PIN IS CONNECTED TO THE PORT 2.1
sbit en = P2^2;      // ENABLE PIN IS CONNECTED PORT 2.2

/* 16*2 LCD SPECIFICATIONS */

#define LCDMAXLINES 2
#define LCDMAXCHARS 16
#define LINEone 0x80
#define LINEtwo 0xc0
#define Blankspace ' '


void LCD_enablePulse()
{

      en = 1;
      delay_us(100);  // pulse width > 230nsec
      en = 0;
      delay_us(10);  // hold time
}

void LCD_init()
{

      delay_us(5000);
      databus = 0x00; // SET IT AS THE OUTPUT PORT;


      LCD_Cmdwrite(0x38);   // LCD 2 LINES , 5*7 MATRIX
      LCD_Cmdwrite(0x0E);   // DISPLAY ON CURSOR ON;
      LCD_Cmdwrite(0x01);    // clear the LCD
      LCD_Cmdwrite(LINEone);    // MOVE THE CURSOR T0 THE LINE FIRST POSITION

}


void LCD_Cmdwrite(unsigned char cmd )
```

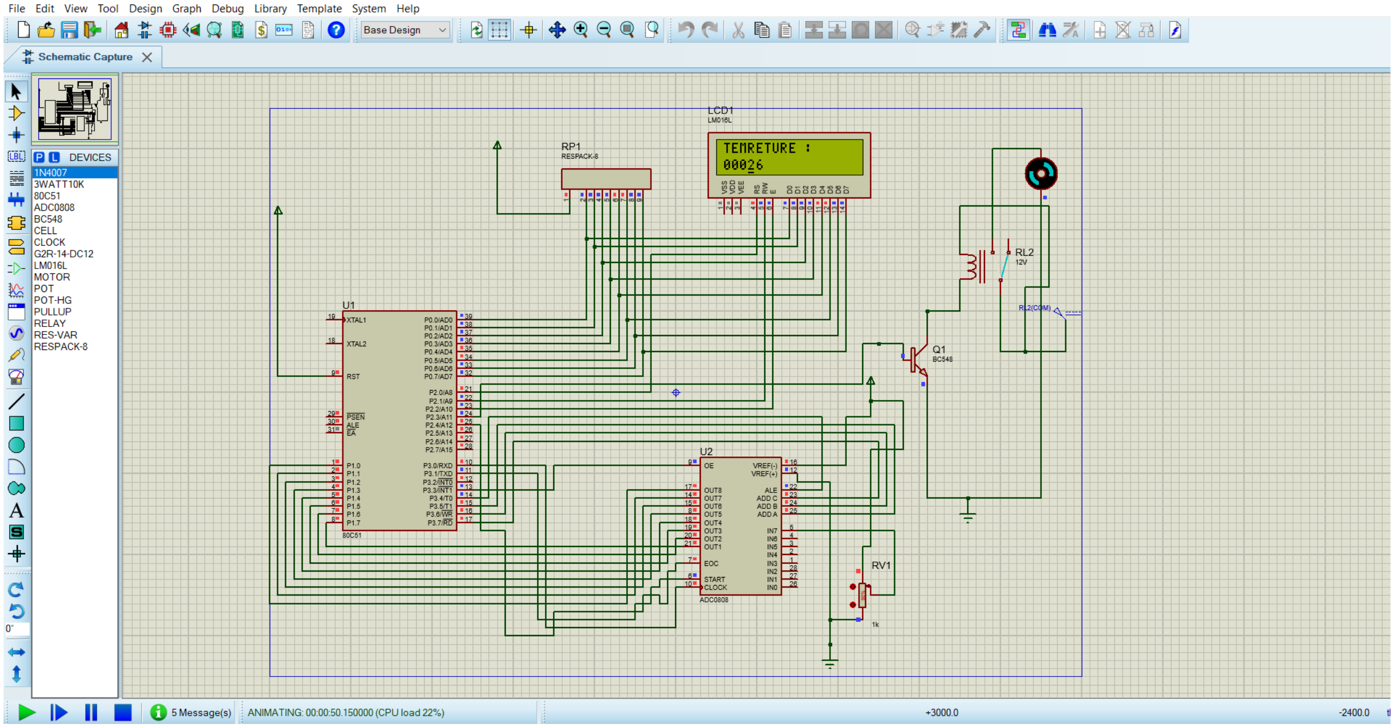
```
{
      databus = cmd;   // SEND THE COMMAND TO THE LCD
      rs = 0;          // SELECT THE COMMAND REGISTER
      rw = 0;          //SELECTNG THR WRITE OPERATION

      LCD_enablePulse();

      delay_ms(1);

}

void LCD_Datawrite( unsigned char dat )
{
      databus = dat;
      rs = 1; // to select the data register
      rw = 0;   // write operation
      LCD_enablePulse();

      delay_ms(1);
}


void LCD_Clear()
{
      LCD_Cmdwrite(0x01);     // CLEAR THE LCD AND GO TO FIRST LINE FIRST POSITION;
      LCD_Cmdwrite(LINEone);

}


void LCD_GoToLINEone()
{
      LCD_Cmdwrite(LINEone);    // MOVE THE CURSOR TO THE FIRST LINE FIRST POSITION
}

void LCD_GoToLINEtwo()
{
      LCD_Cmdwrite(LINEtwo);    // MOVE THE CURSOR TO THE LINE TWO
}


void LCD_GoToXY(char row, char col)
{
      char pos;    //THIS WILL HAVE THE COMPUTED HEXADECIMAL VALUE OF THE POSITION TO
WHICH THE CURSOR IS TO BE MOVED

      if( row< LCDMAXLINES )
      {
            pos = LINEone | (row<<6);  // IF THE ROW->0 THEN THIS STATEMENT WILL MOVE
IT TO THE POS = 0X80 AND IF ROW->1 THEN POS = 0XC0

            if( col < LCDMAXCHARS)
            {
                   pos = pos + col;
            }
      }

      LCD_Cmdwrite(pos);
}


void LCD_DisplayString(unsigned char *string_ptr)
{
      while(*string_ptr!='\0')
      {
            LCD_Datawrite(*string_ptr);
            string_ptr++;
      }
}



void LCD_DisplayNumber(unsigned int num)
{
     LCD_Datawrite((num/10000)+0x30);
         num = num%10000;

         LCD_Datawrite((num/1000)+0x30);
         num = num%1000;

         LCD_Datawrite((num/100)+0x30);
         num = num%100;

         LCD_Datawrite((num/10)+0x30);

         LCD_Datawrite((num%10)+0x30);

}
#include<reg51.h>

#include "delay.h"
#include "adc.h"


#define adc_databus P1

sbit adc_Start = P3^1;   // ADC START CONVERSION PIN
sbit adc_EOC   = P2^4;  // ADC EOND OF CONVERSION
sbit adc_OE    = P3^3;  //ADC OUTPUT ENABLE PIN
sbit adc_ALE   = P3^4;   // ADC LATCH EABLE
sbit adc_A0    = P3^5;
sbit adc_A1    = P3^6;     // THIS THREE PIN ARE THE ADRESS LINE PINS
sbit adc_A2    = P3^7;

void ADC_init()
{
      adc_Start = 0;        //INITIALIZE ALL THE CONTORL PINS TO ZERO
      adc_ALE = 0;
      adc_OE = 0;
      adc_EOC = 1;           // CONFIGURE THE EOC PIN AS I/P
      adc_databus = 0xff;   //  CONFIGURE adc_databus as I/P
}

//compiler ----sourav
unsigned char ADC_StartConversion(char channel)
{
       unsigned char adc_result; // to store the adc converted value

      // THIS IS DONE FOR SELECTING THE APPROPIATE CHANNEL TO BY GETTING THE CHANNEL
ADDRESS
      adc_A0 = ((channel>>0) & 0x01 );
      adc_A1 = ((channel>>1) & 0x01 );
      adc_A2 = ((channel>>2) & 0x01 );




      adc_ALE = 1; // LATCH THE ADDRESS BY MAKING ALE 1
      delay_us(50);   // DELAY >50ns
      adc_Start = 1;
      delay_us(25);   // DELAY >25ns


      adc_ALE = 0;
      delay_us(50);
      adc_Start = 0; // PULL DOWN THE START LINE TO ZERO AFTER STARTING THE CONVERSION


      /*WAIT UNTIL THE CONVERSION IS COMPLETE
      EOC WILL BE WILL BE AGIN PULLED TO HIGH BY THE HARDWARE ADC0808 ONCE THE
CONVERSION IS COMPLETE*/

      while(adc_EOC==0);


      adc_OE = 1;   // OUTPUT ENABLKE HIGH TO BRING THE DATA TO PORT PINS
      delay_us(25);
      adc_result = adc_databus;  // READ DATA FROM THE ADC DATA BUS
      adc_OE = 0;  // AFTER READING THE DATA OUTPUT ENABLE IS AGAIN MADE 0


  return (adc_result);   //0x10

}
#include<reg51.h>

#include "lcd.h"
#include "delay.h"
#include "adc.h"
```

```
unsigned char adc_result = 0;
unsigned int adc_temp = 0;
sbit adc_clk   = P3^0;     // THIS IS FOR THE CLOCK FREQUENCY PROVIDED  TO THE ADC
sbit relay_pin = P2^3;


void timer_0() interrupt 1
{
      adc_clk = ~ adc_clk;
}


void main()
{
      relay_pin = 0;

      TMOD = 0x02;
      TH0 = 0xFD;
      LCD_init();  //INITIALISE THE LCD

      LCD_DisplayString("AUTOMATIC");

      // GO TO LINE TWO
      LCD_GoToXY(1,0);
      LCD_DisplayString("IRRIGATION SYSTEM");

      delay_ms(500);

      LCD_Clear();  // CLEAR THE LCD SCREEN
      LCD_GoToLINEone();

      // INITILAISING THE ADC

            ADC_init();
  LCD_DisplayString("TEMRETURE : ");

      IE=0x82;

      TR0 = 1;  // START THE TIMER;


      while(1)
      {

            //compiler ---- sourav
            // GET THE ADC VALUE OF CHANNEL 8
            adc_result = ADC_StartConversion(7);   // 0x10

            // GO TO THE LINE TWO AND DISPLAY THE ADC VALUE
            LCD_GoToLINEtwo();
            //LCD_DisplayNumber(adc_result);
```

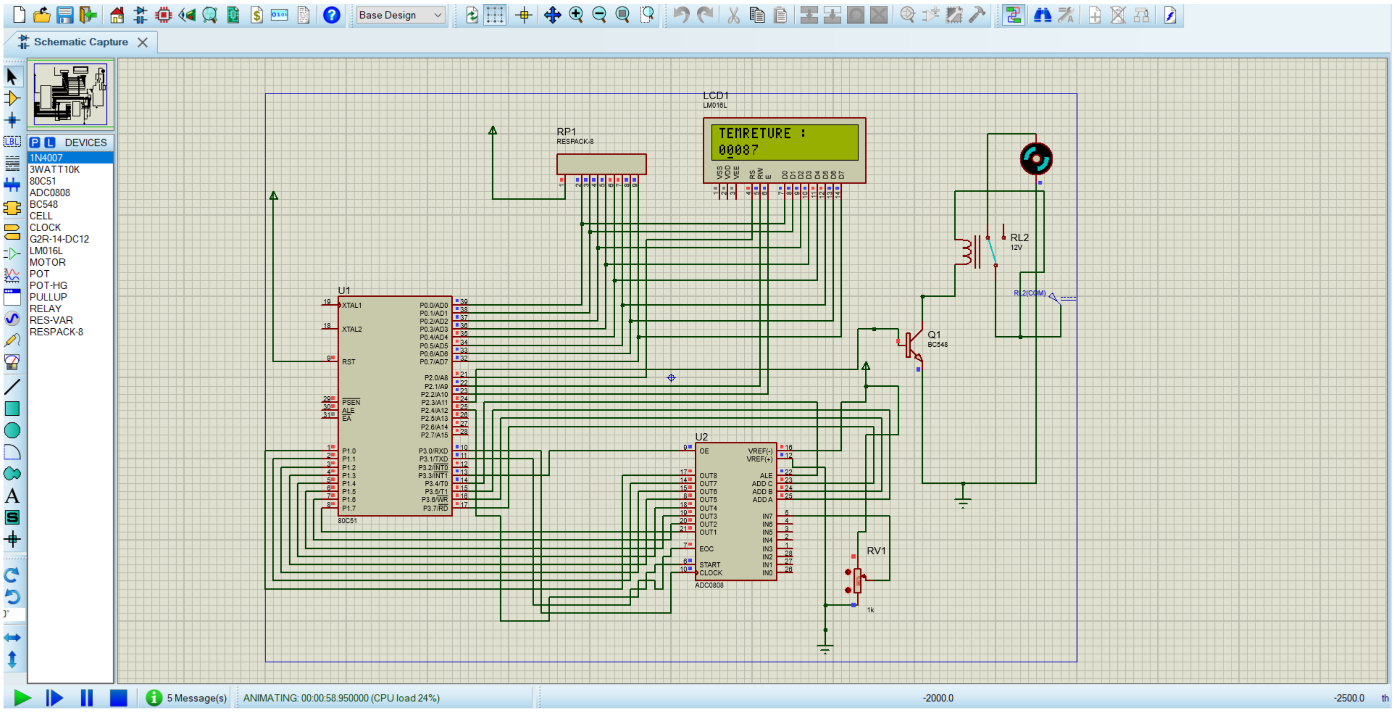
```
   adc_temp = (adc_result);   //0x20



            if((adc_temp)<50)
            {
                  relay_pin =  0;
            }
            else
            {
                  relay_pin = 1;
            }

   LCD_DisplayNumber(adc_temp);
      }
}
```

OBSERVATION

(TEMP < 50 Deg (in Celsius) Motor -- OFF)



(TEMP > 50 Deg (in Celsius) Motor -- ON)



CONCLUSION

It can be seen that the combination of hardware and software provides a irrigation controller that can be implemented at relatively low cost and which is extremely user friendly. From the point of view of working at remote place the developed microcontroller based irrigation system can work constantly for indefinite time period, even in certain abnormal circumstances. If the plants get water at the proper time then it helps to increase the production from 25 to 30 %. This system can be used to irrigate very large areas as it only needs to divide the whole land into number of sectors and single microcontroller can control the whole process. It saves human energy, time, cost, etc.