- # Abstract

Automation is taking over various aspects of our life and the most important aspect is the one dealing with quality of food produce and its consumption. With growing produce, it is our responsibility to manage our resources provided by nature effectively and appropriately. For this very much purpose we need a way to hand over this task into the hands of automation.

Seeing the modern availability of power saving and capable electronics we aim to make a system with ability to supply water mixed along with nutrients and fertilizer plants needs and use the approach of drip irrigation equipped with advanced sensors for the purpose of supplying.

- # Introduction

In this project we have made a dynamic drip irrigation system to scale down human errors as the processes are handled by microcontroller 8051. The project utilizes soil moisture sensor and water level sensor to monitor soil water presence level and threshold water stored in the water tank respectively.

A basic SPDT switch is directly connected to the DC motor connected to water tank to control the amount of water supply based on the soil type. An LCD Display also is used to notify the present status of the system such as when there is Low or High moisture detected in the soil and if there is sufficient water present in the water tank.

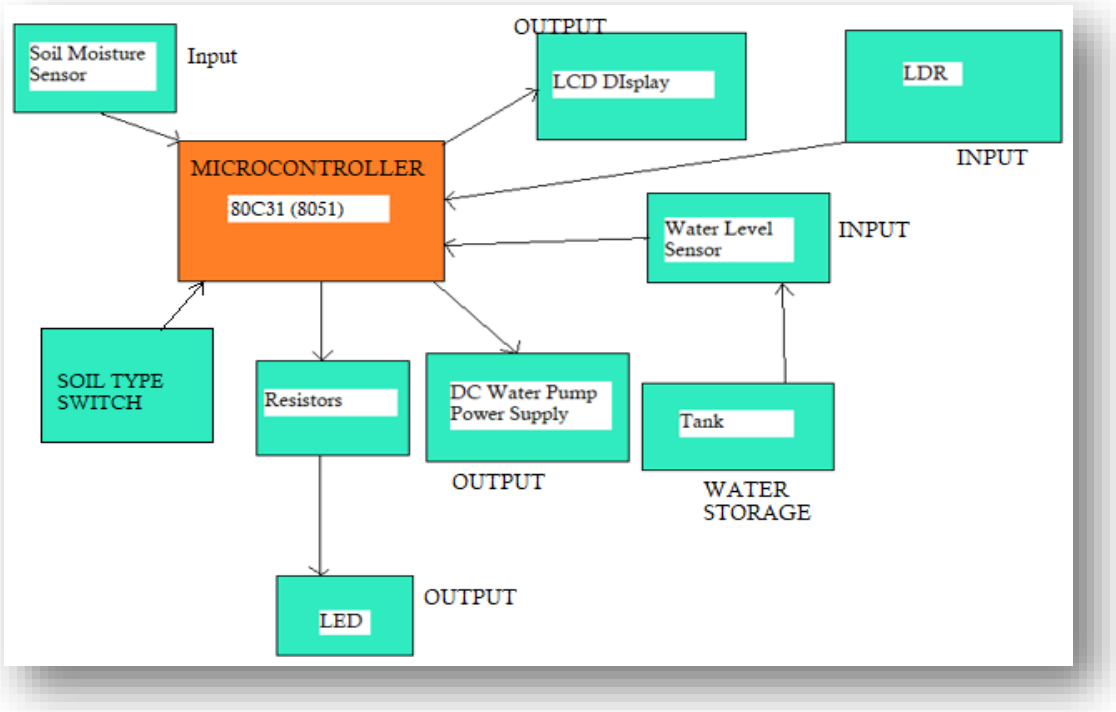
Additionally, an active buzzer has been included for the purpose of notifying if immediate water refill is required in the water tank.

- # Methodology

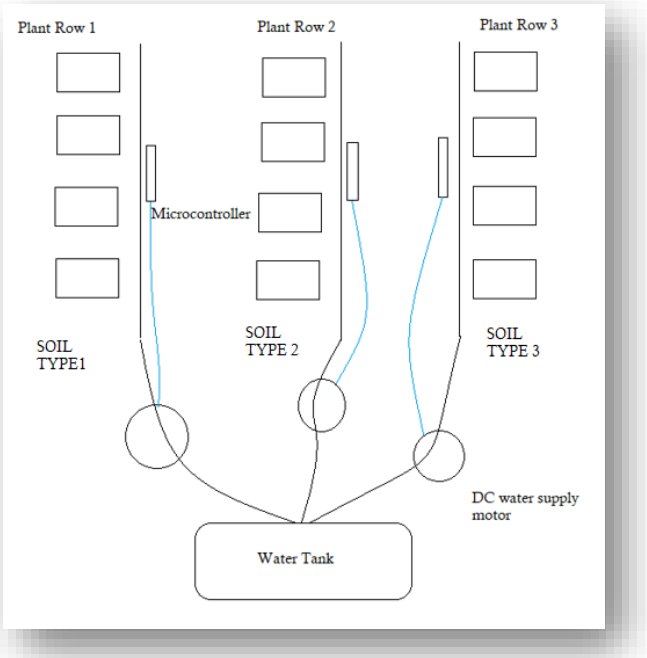
The circuit check for the threshold value of moisture in soil and current reading is taken with help of soil moisture sensor. If value is less, DC motor runs and at every step, water level in tank is measured using level sensor, if

it is low, the buzzer rings to notify. The DC motor stops once required moisture value is reached. All the results are displayed in LCD.

- **Overview of the Hardware**



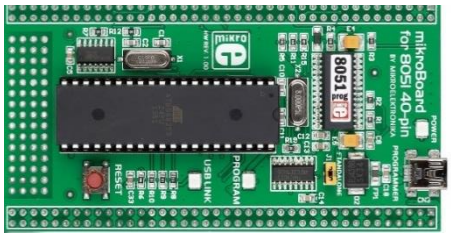
Input/output block diagram (Above) Proposed implementation (Below)



- **Components Used**

1) 80C31 (8051) Microcontroller

Microcontroller to handle all the tasks.

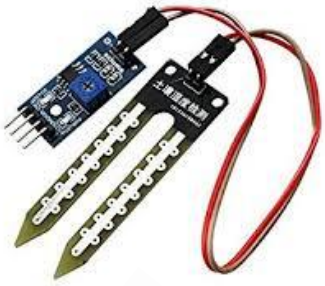


2) LCD Display

To display the current status of drip irrigation system 16x2 line LCD Display is used.

3) Soil Moisture Sensor

To sense the moisture in soil and interrupt microcontroller as soon as value drops below threshold



4) Water level sensor

To sense when water is running out in Water Tank and beep a buzzer.

5) DC Water pump

Motor to supply water to plants in a drip.

6) Power Supply

7) LED

LEDs for the purpose of lighting up when

1) Buzzer rings
2) Water Level LOW in Tank
3) Moisture Level LOW in Soil

8) Active Buzzer

Beeps when Water LOW detected in Tank

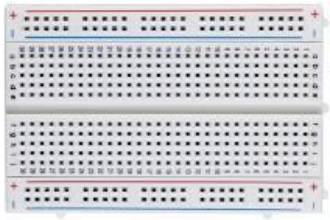
9) Optocoupler IC (4N35)

Since 8051 cannot power up the motor by itself, the optocoupler IC does this task by taking power from battery and switching the pump when microcontroller sends a HIGH to pump.

10) Push button

Pull up button to reset the microcontroller

11) Breadboard and jumpers

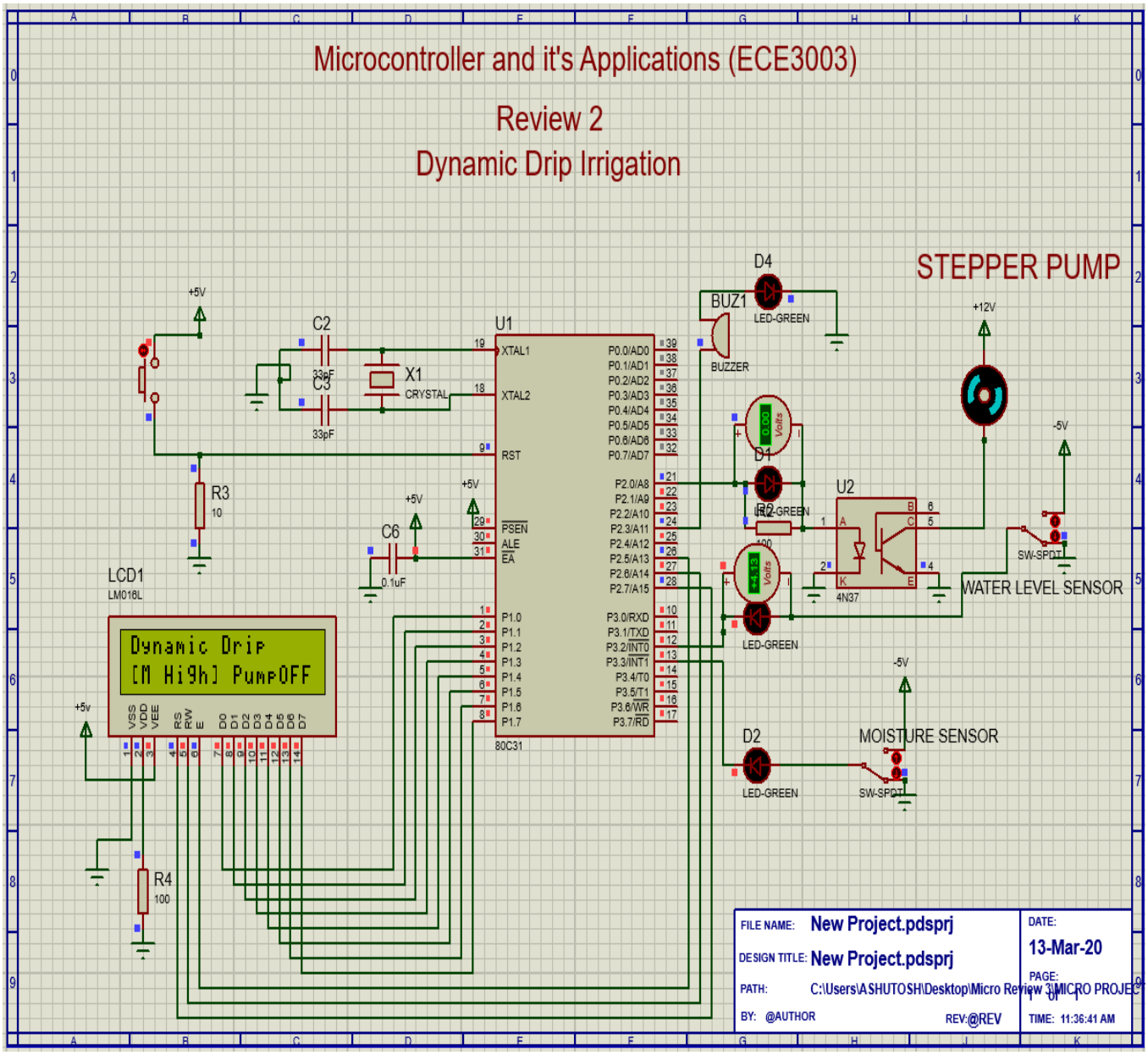


12) SPDT Switch

Used to supply a predefined voltage to the DC Motor depending on the soil type water requirement.

- **Circuit Diagram (Proteus Simulation)**

Proteus Professional 8 software is used to build the circuit diagram and perform the simulation.



Dynamic Drip Irrigation System circuit diagram including 8051(80C31) at the center.

- **Assembly Code**

```
 RS EQU P2.7
RW EQU P2.6
E  EQU P2.5
BB EQU P2.3

ORG 00H
LJMP GO                          ;Skip Vector Table
ORG 0003H                        ;ISR for EX1
    CLR P2.0
    SETB BB
    ACALL DINT
        ACALL TEXT1               ;Dynamic Drip
    ACALL LINE2                  ;Change Line
    ACALL TEXT4                  ;Tank Empty
    EXIT2:ACALL DELAY1
    RETI
ORG 0013H                        ;Interrupt Service Routine for EX0
                                ;Turn on the PUMP
    SETB P2.0
    ACALL DINT
    ACALL TEXT1                  ;Dynamic Drip
    ACALL LINE2                  ;Change Line
    ACALL TEXT3                  ;[Low} Pump ON
    MOV R0,#100D
    EXIT1:ACALL DELAY
    DJNZ R0,EXIT1
    CLR P2.3
    RETI

ORG 0100H
GO:NOP
SETB P3.3
SETB P3.4
MOV IE,#10000101B
```

```
MOV TMOD,#00000001B                ;Timer 0, Mode 1
MAIN:ACALL DINT
    ACALL TEXT1
    JNB P3.3, NEXT
    ACALL LINE2
    ACALL TEXT2
    CLR P2.0
    SJMP EXIT
NEXT:ACALL LINE2
    ACALL TEXT3
    SETB P2.0
    EXIT:ACALL DELAY1
SJMP MAIN


DELAY:
BACK1: MOV TH0,#0AAH
      MOV TL0,#00000000B
      SETB TR0
HERE2: JNB TF0,HERE2
      CLR TR0
      CLR TF0
      RET


TEXT1: MOV A,#"D"
   ACALL DISPLAY
   MOV A,#"y"
   ACALL DISPLAY
   MOV A,#"n"
   ACALL DISPLAY
   MOV A,#"a"
   ACALL DISPLAY
   MOV A,#"m"
   ACALL DISPLAY
   MOV A,#"i"
   ACALL DISPLAY
```

```
    MOV A,#"c"
    ACALL DISPLAY
    MOV A,#" "
    ACALL DISPLAY
    MOV A,#"D"
    ACALL DISPLAY
    MOV A,#"r"
    ACALL DISPLAY
    MOV A,#"i"
    ACALL DISPLAY
    MOV A,#"p"
    ACALL DISPLAY
    RET

TEXT2: MOV A,#"["
    ACALL DISPLAY
       MOV A,#"M"
    ACALL DISPLAY
    MOV A,#" "
    ACALL DISPLAY
     MOV A,#"H"
    ACALL DISPLAY
    MOV A,#"i"
    ACALL DISPLAY
    MOV A,#"g"
    ACALL DISPLAY
    MOV A,#"h"
    ACALL DISPLAY
    MOV A,#"]"
    ACALL DISPLAY
    MOV A,#" "
    ACALL DISPLAY
    MOV A,#"P"
    ACALL DISPLAY
    MOV A,#"u"
    ACALL DISPLAY
    MOV A,#"m"
```

```
    ACALL DISPLAY
    MOV A,#"p"
    ACALL DISPLAY
    MOV A,#"O"
    ACALL DISPLAY
    MOV A,#"F"
    ACALL DISPLAY
    MOV A,#"F"
    ACALL DISPLAY
    RET

TEXT3: MOV A,#"["
    ACALL DISPLAY
       MOV A,#"M"
    ACALL DISPLAY
    MOV A,#" "
    ACALL DISPLAY
    MOV A,#"L"
    ACALL DISPLAY
    MOV A,#"o"
    ACALL DISPLAY
    MOV A,#"w"
    ACALL DISPLAY
    MOV A,#"]"
    ACALL DISPLAY
    MOV A,#" "
    ACALL DISPLAY
    MOV A,#"P"
    ACALL DISPLAY
    MOV A,#"u"
    ACALL DISPLAY
    MOV A,#"m"
    ACALL DISPLAY
    MOV A,#"p"
    ACALL DISPLAY
    MOV A,#" "
    ACALL DISPLAY
```

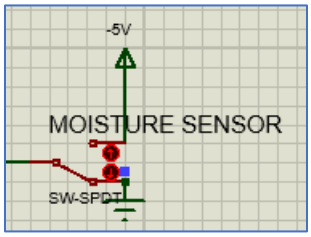
```
    MOV A,#"O"
    ACALL DISPLAY
    MOV A,#"N"
    ACALL DISPLAY
    RET

TEXT4: MOV A,#"T"
    ACALL DISPLAY
    MOV A,#"a"
    ACALL DISPLAY
    MOV A,#"n"
    ACALL DISPLAY
    MOV A,#"k"
    ACALL DISPLAY
    MOV A,#" "
    ACALL DISPLAY
    MOV A,#"E"
    ACALL DISPLAY
    MOV A,#"m"
    ACALL DISPLAY
    MOV A,#"p"
    ACALL DISPLAY
    MOV A,#"t"
    ACALL DISPLAY
    MOV A,#"y"
    ACALL DISPLAY
    MOV A,#"!"
    ACALL DISPLAY
    RET


 DINT:MOV A,#0CH            ;DIsplay ON,Cursor OFF
    ACALL CMD
    MOV A,#01H              ;Clear Display
    ACALL CMD
    MOV A,#06H              ;Increment Cursor
    ACALL CMD
```

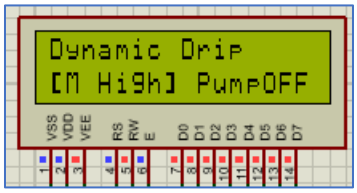
```
    MOV A,#80H              ;Force cursor to go to first line
    ACALL CMD
    MOV A,#3CH              ;Activate Second Line
    ACALL CMD
    RET

LINE2:MOV A,#0C0H
    ACALL CMD
    RET

CMD: MOV p1,A               ;Command Register
    CLR RS
    CLR RW
    SETB E
    CLR E
    ACALL DELAY
    RET

DISPLAY:MOV p1,A
    SETB RS
    CLR RW
    SETB E
    CLR E
    ACALL DELAY
    RET
DELAY1: CLR E
    CLR RS
    SETB RW
    MOV p1,#0FFH
    SETB E
    MOV A,p1
    JB ACC.7,DELAY1
    CLR E
    CLR RW
    RET

 END
```

- # Results

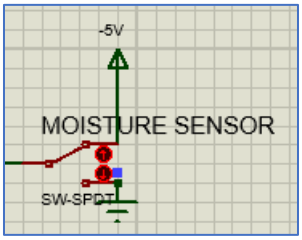
The software simulation performed on Proteus 8 yields the following results upon execution.

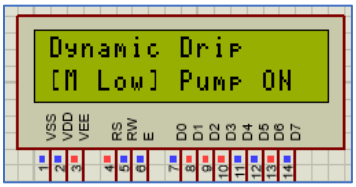
1) Scenario A
When the moisture content in the soil is HIGH, the system is in idle mode and LCD display has the following output.



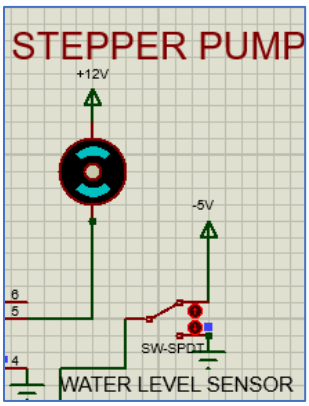


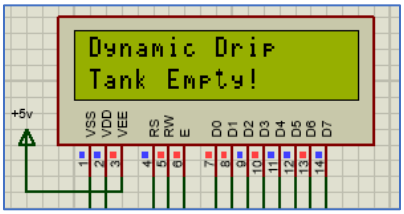
2) Scenario B
When the moisture content in the soil is LOW as monitored by MOISTURE SENSOR, the system is in idle mode and LCD display has the following output.

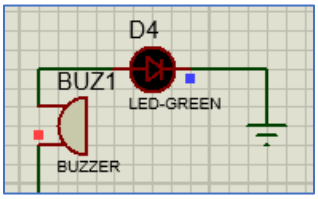




3) Scenario C
When the water is tank has started depleting and then buzzer goes high and LCD notifies.







- # Conclusion

A dynamic drip irrigation system has been successfully implemented which deals with the problem of water management and if applied on a large scale would make its crucial stance in dealing with environmental problems. The project has the potential to grow and scale with implementation of IoT technology.

- # References

www.hackster.io

www.wikipedia.org

www.google.com

www.circuitstoday.com