

# MICROCONTROLLER BASED WATER LEVEL CONTROL SYSTEM

The present concept implements controlling of pump which pumps water from the sump (underground tank) to the overhead tank, using 8951 microcontroller.

The control panel, i.e. the main control unit of the system which consists of the primary control switches, pump indicator, siren and level indicators. The visual example of how switches And the indicators can be placed as shown the figure below.

Power ON/OFF			Manual/Auto
Indicators			Pump ON/OFF
Levels	Tank	Sump	
F	LED	LED	Pump ON Indicator
H	LED	LED	
L	LED	LED	Siren
E	LED	LED	

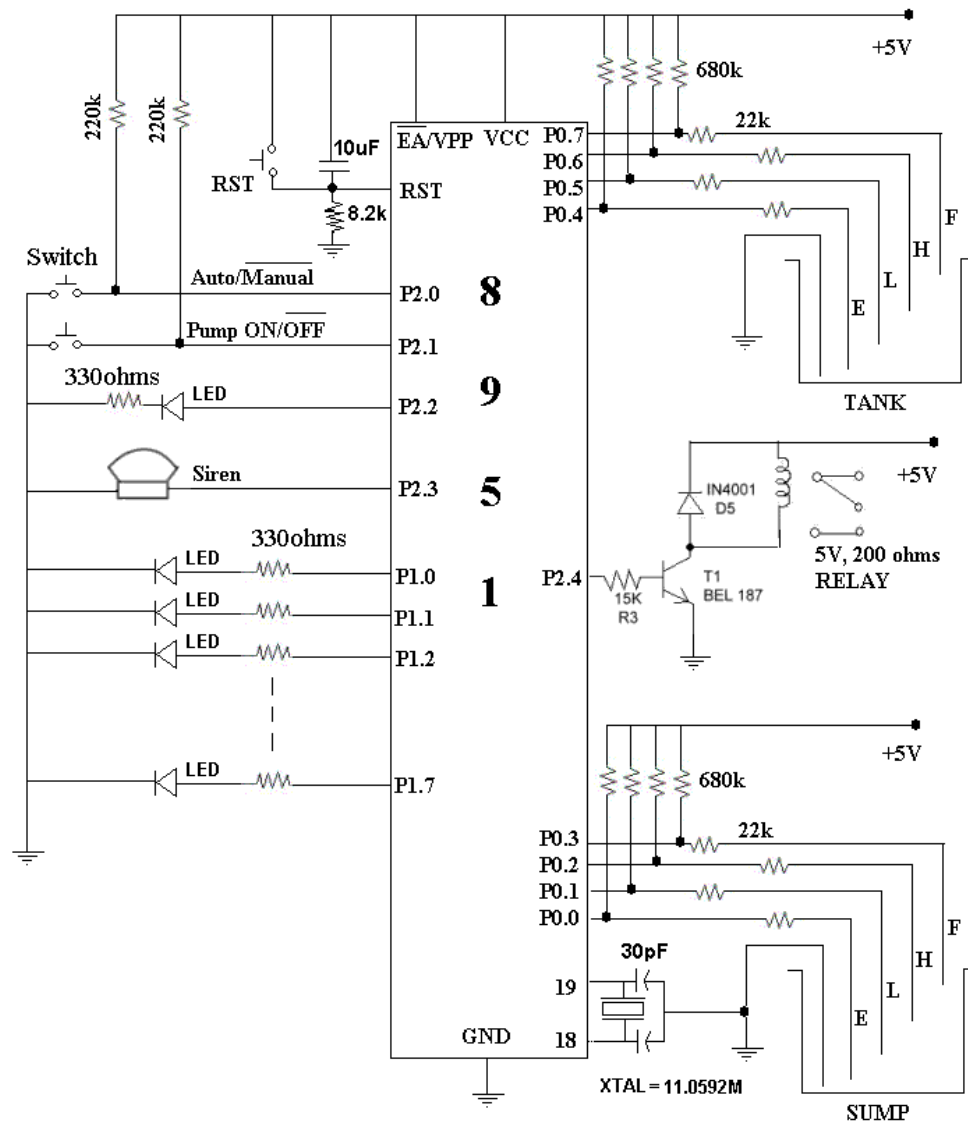
Front view of the model

In the figure shown above there are total of nine LEDs, four of which indicates the water level in the tank, another four indicates the water level in the sump and one LED indicates whether the pump is ON/OFF. It also consists of three switches.

1. Switch 1 is the main power switch which is used to activate the system.
2. Switch 2 is used to select whether to operate the pump in AUTO or MANUAL mode.
3. Switch 3 comes to picture only when the system is operated in MANUAL mode. It controls the direct activation of the pump.

## Description :

This system is built around an 8951 microcontroller and the circuit diagram is as given below.



<http://electroguys.com>

### Circuit diagram of Water level indicator cum controller

As you can see in the above diagram, port 0 is exclusively used as an input port which takes the information regarding the water level in the sump as well as in the overhead tank.

Port 1 is used as output port which is connected to the indicator that indicates the water level in both the tanks.

Port 2 is used as in/out port, it takes the input from switch 2 and switch 3 and gives the output which is connected to pump indicator, siren and the relay which controls the switching of the pump.

#### Working of the system:

There are two modes of working for the system

1. Manual mode
2. Auto mode

Which is controlled by switch 2 (refer control panel diagram)

#### Manual mode:

When the system is active and in manual mode, it only indicates the water levels in the tanks and it doesn't control any working of the pump. To activate the pump in manual mode switch 3 is used.

In this mode the operator should manually control the working of the pump. As in case if the tank is full, operator should switch of the pump which is not the case when compared to auto mode.

**Auto mode:**

When the system is active and in auto mode, it only indicates the water levels in the tanks and it controls the working of the pump.

Programm:

```
.*****  
,  
*****  
;Objective : Water Level Indicater and Motor Protection  
;Name of the Programmer : Prateek Kaura  
;Hardware Used : CPU Board, Led Card, Power Supply, 8 pin connector  
; Two pin connector  
;Program Description ; Crystal=11.0592kHz  
; Port0 is used as LCD  
; Port1 is used for Motor  
; Port2 is used for LED's  
; Port3 is used for RF Module  
;Status :  
*****Hardware  
,  
declaration*****,  
lcd equ p0  
RS BIT P2.5  
EN BIT P2.4  
buzzer bit p2.3  
GREEN bit p2.0  
YELLOW bit p2.1  
RED bit p2.2  
senser1 bit p3.0  
senser2 bit p3.1  
senser3 bit p3.2  
senser4 bit p3.3  
relay1 bit p1.0  
key1 bit p1.3  
*****  
,  
*****,  
;  
ORG 0000H ;RESET OPERATION
```

```

SJMP POWERON ;GO TO POWERON LABEL
ORG 0003H ;EXTERNAL0 INTERRUPT

RETI ;RETURN FROM THE INTERRUPT
ORG 000BH ;TIMER0 INTERRUPT
RETI ;RETURN FROM THE INTERRUPT
ORG 0013H ;EXTERNAL1 INTERRUPT

RETI ;RETURN FROM THE INTERRUPT
ORG 001BH ;TIMER1 INTERRUPT
; ljmp timer1

RETI ;RETURN FROM THE INTERRUPT
ORG 0023H ;SERIAL COMMUNICATION INTERRUPT
RETI ;RETURN FROM THE INTERRUPT
ORG 0030H

POWERON: MOV SP,#70H ; Move the stack pointer at 70h location
MOV IE,#00H ; Disable all the interrupts
MOV IP,#00H ; Disable the interrupt priority register
MOV P0,#0FFH ; Move 0FFH in in port 0
MOV P1,#0FFH ; Move 0FFH in in port 1
MOV P2,#0FFH ; Move 0FFH in in port 2
MOV P3,#0FFH ; Move 0FFH in in port 3

;*RAM variable declaration ,30h-7Fh
;.....
;.....
;*****
;*****
;*****POWER ON
;*****
;*****RAM
CLEARANCE*****
MOV R0,#7FH
RAM_CLR: MOV @R0,#00H ; Clear the RAM from 00 to 7FH
DJNZ R0,RAM_CLR
..*****MAIN
PROG*****

setb red
setb yellow
setb green
clr relay1
mov lcd,#38h
acall command

```

```
mov lcd,#01h
acall command
mov lcd,#0eh
acall command
mov lcd,#80h
acall command
MOV DPTR,#DATA1
ACALL DISPLAY_
main1:
setb red
setb yellow
setb green
mov r2,#70d
main3:
setb green
setb RELAY1
acall delay
jnb senser1,check1
djnz r2,main3
mov r2,#50d
sjmp main5
main11:
jnb p1.3,main3
sjmp main11
main:
setb buzzer
setb green
setb red
setb yellow
jnb senser1,check1
sjmp main
main5:
clr relay1
acall delay
setb green
setb red
setb yellow
mov lcd,#0c0h
acall command
MOV DPTR,#level4
ACALL DISPLAY_
```

```
acall command
sjmp main11
check1:
clr green
setb red
setb YELLOW
jb senser1,main3
jnb senser2,check2
mov lcd,#0c0h
acall command
MOV DPTR,#level1
ACALL DISPLAY_
djnz r2,check1
mov r2,#20
sjmp main
check2:
clr YELLOW
clr green
setb red
;jnb p3.2,main3
jnb senser3,check3
jb senser2,main
mov lcd,#0c0h
acall command
MOV DPTR,#level2
ACALL DISPLAY_
sjmp check2
check3:
clr red
mov r3,#04
clr YELLOW
clr green
clr relay1
buzzer_on:
clr buzzer
acall delay
setb buzzer
acall delay
jb senser3,main
mov lcd,#0c0h
acall command
MOV DPTR,#level3
```

```

ACALL DISPLAY_
djnz r3,buzzer_on
check34:
setb buzzer
acall delay
clr red
clr YELLOW
clr green
clr relay1
jnb senser3,main7
mov lcd,#0c0h
acall command
MOV DPTR,#level3
ACALL DISPLAY_
sjmp check34
main7:
ljmp main
DISPLAY_:
CLR A
MOVC A,@A+DPTR
MOV LCD,A
ACALL DATA
INC DPTR
JZ EXIT
SJMP DISPLAY_
EXIT:
RET
;*****command*****
command:
clr RS
setb EN
nop
nop
nop
nop
nop
clr EN
ACALL DELAY
ret
;*****data*****
data:
setb RS
setb EN

```

```

nop
nop
nop
nop
nop
clr EN
ACALL DELAY
ret
;*****delay*****
delay:
mov 33h,#50d
loop1: mov 34h,#50d
loop2: mov 35h,#20d
loop3: djnz 35h,loop3
djmp 34h,loop2
djmp 33h,loop1
ret
;*****data*****
byte
ORG 300H
DATA1:
DB 'WATER LEVEL ',0
level1:
DB 'Low Level ',0
level2:
DB 'Medium Level ',0
level3:
DB 'High Level ',0
level4:
DB 'Input Line Fail ',0

```

Source : <http://todayscircuits.blogspot.com/2012/01/microcontroller-based-water-level.html#.VHB64jSUdhg>