Water Level Controller

Gaurav Gupta

Abstract

A Water level controller is designed to control the water level of tank. The system deals with monitoring of the water level of the tank and controlling the action of water pump in accordance. The pump is driven by the circuit to meet the demands like controlling water level.

1 Introduction

The Water Level Controller (WLC) is a system to control the water level of the tank either automatically or manually. The system when set to the auto mode will control water level of the tank in order to meet the user set required water level. The system is designed to avoid situations of spillage as well as absence of water.

1.1 Power Switch

The power switch as shown in Fig.1 is used to operate the motor with the control signals from the WLC main system. An alternate switch (big red one) is provided to override the actions of the controller i.e. manually switching on/off the motor. The red LED will glow when the motor is set to the on status by the controller.

Figure 1: Power switch driven by the controller to operate the motor

Control wires:

1. **Thick** wires to be used as a break point between any one out of two wires of the motor.
2. **Red**: Vcc.
3. **Orange**: Gnd.
4. **Brown**: Input signal.

1.2 Main Control

Control wires:

- To tank:
  1. **Black**: Vcc.
  2. **Brown**: Level-1.
  3. **Orange**: Level-2.
  4. **Red**: Level-3.
  5. **Yellow**: Level-4.
  6. **Green**: Level-5.
- To power switch:
  1. **Red**: Gnd.
  2. **Orange**: Vcc.
  3. **Brown**: Output signal.

Figure 2: Main control box of the WLC

2 Code

```
#include <mega16.h>
#include <stdio.h>
#include <delay.h>

// External Interrupt 2 service routine
int set_lev = 2;
int current_lev;
int IsAutoModeEnabled = 0;
int intrr_flag = 0;

interrupt [EXT_INT2] void ext_int2_isr(void)
{
// Place your code here
set_lev++;
if (set_lev > 5){
     set_lev = 2;
}

if (IsAutoModeEnabled){
     if (current_lev < set_lev){
       PORTC.5 = 1; // set = 1 ; turn on
       intrr_flag = 1;
     }
     else{
       PORTC.5 = 0; //set = 0;
       PORTC.6 = 1; //reset = 1; turn off
     }
}
}

// Declare your global variables here

void main(void)
{
// Declare your local variables here
```

```
int inp_lev_1;
int inp_lev_2;
int inp_lev_3;
int inp_lev_4;
int inp_lev_5;
int filling_flag = 0;

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTB=0x00;
DDRB=0xFF;

// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
DDRD=0xFF;

TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT2:  On
// INT2 Mode:  Rising Edge
GICR|=0x20;
MCUCR=0x00;
MCUCSR=0x40;
GIFR=0x20;
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x00;

ACSR=0x80;
SFIOR=0x00;

// Global enable interrupts
# asm("sei")

while (1)
      {
      // Place your code here
      inp_lev_1 = PINA.1;
      inp_lev_2 = PINA.2;
      inp_lev_3 = PINA.3;
      inp_lev_4 = PINA.4;
      inp_lev_5 = PINA.5;
      IsAutoModeEnabled = PINA.6;

      PORTD = 0x00;
      PORTC = 0x00;
      PORTB = 0x00;
      current_lev = 0;

      switch (set_lev){
        case 2:
          PORTB.3 = 1;
          break;
        case 3:
          PORTB.4 = 1;
          break;
        case 4:
          PORTB.5 = 1;
          break;
        case 5:
          PORTB.6 = 1;
          break;
      }

      if (!inp_lev_1 & !inp_lev_2 & !inp_lev_3 & !inp_lev_4 & !inp_lev_5){
        PORTC.0 = PINA.0;
        // current level is 0
      }
      if (inp_lev_1){
        PORTC.0 = 1;
        current_lev = 1;
      }
      if (inp_lev_2){
        PORTC.1 = 1;
        PORTC.0 = 1;
        current_lev = 2;
      }
      if (inp_lev_3){
        PORTC.2 = 1;
        PORTC.1 = 1;
        PORTC.0 = 1;
        current_lev = 3;
      }
      if (inp_lev_4){
        PORTC.3 = 1;
        PORTC.2 = 1;
```

```
      PORTC.1 = 1;
      PORTC.0 = 1;
      current_lev = 4;
    }
    if (inp_lev_5){
      PORTC.4 = 1;
      PORTC.3 = 1;
      PORTC.2 = 1;
      PORTC.1 = 1;
      PORTC.0 = 1;
      current_lev = 5;
    }

    if (IsAutoModeEnabled){
      if (current_lev >= set_lev){
        PORTC.5 = 0; //set = 0;
        PORTC.6 = 1; //reset = 1; turn off
        intrr_flag = 0;
        filling_flag = 0;
      }
      else if (!current_lev || current_lev < set_lev-1 || filling_flag){
        PORTC.5 = 1; // set = 1 ; turn on
        filling_flag = 1;
      }
      else{
        if (intrr_flag){
          PORTC.5 = 1; // set = 1 ; turn on
        }
        else{
          PORTC.5 = 0; //set = 0;
          PORTC.6 = 1; //reset = 1; turn off
        }
      }
    }
}
}
```

3 Schematic

3.1 Power Switch

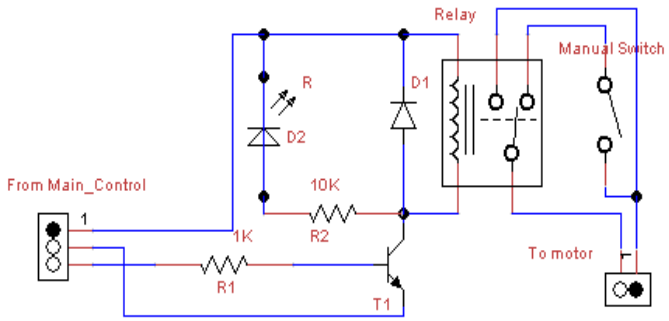

Figure 3: Schematic of power switch

3.2 Main Control

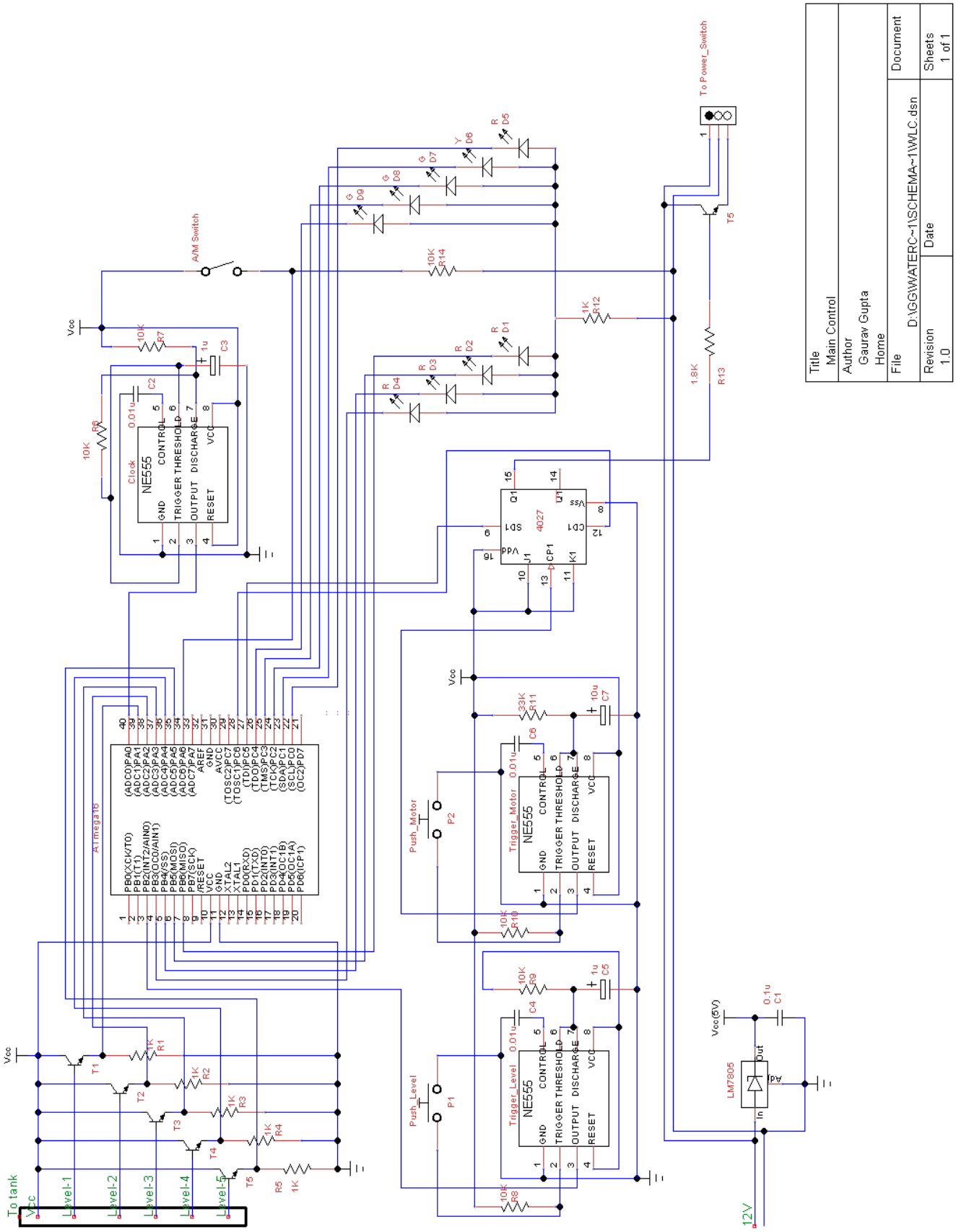

Figure 4: Schematic of main control box.