KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

PROJECT ACTIVITY REPORT

Design of Water Level Indicator and Pump Controller

By:

Abhisek Adhikari

Adhish Bisht

Lakpa Tamang

Lokesh Pant

Submitted to:

Mr. Madhav Parsad Pandey

February, 2015

ABSTRACT

Most of the people in residential areas of capital of Nepal, Kathmandu face the problem of running out of water and overflow of water in overhead water tanks due to excess pumping of commercial water via pumps from reserve tanks. It becomes difficult for users to judge the level of water in water tanks. When the pump is turned ON, users can not realize that the water tank is filled, which often results in overflow. Water level indicator and pump controller system is used to sort out the issues associated with overhead water tank. It is possible to check the level of the water using water-philiac sensor so that the water does not corrode it and when water goes below a certain level, pump gets turned ON automatically. Also when there is overflow of water in water tank it detects the water level so that if the water level goes above, the pump gets turned off automatically. Further, it continuously monitors the level of water and gives the user an idea of the level of water and the user can manually intervene the system and control level of water as desired. The system basically prevents wastage of water, wastage of energy, it is very cost efficient and user friendly with manual and automatic modes available via easy user interface.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Department of Electrical and Electronics Engineering, Kathmandu University for providing the opportunity to undertake this project 'Design of Water Level Indicator and Pump Controller'.

We would like to thank our supervisor for his guidance and help provided to us throughout the semester which undoubtedly helped us to take our project to the state it is today.

LIST OF FIGURES

Figure 1 Block Diagram **2**

Figure 2 Float Sensor **3**

Figure 3 Ethernet Cable **4**

Figure 4 NPN and PNP Transistor **5**

Figure 5 Relay **5**

Figure 6 12V relay **6**

Figure 7 Microcontroller **8**

Figure 8 Diode **9**

Figure 9 LCD **9**

Figure 10 NPN transistor as a Switch Common Emitter Configuration **11**

Figure 11 PNP transistor as a Switch Common Emitter Configuration **11**

Figure 12 Overall Arrangement of the System and Circuit **14**

TABLE OF CONTENT

ABSTRACT i

ACKNOWLEDGEMENT ii

LIST OF FIGURES iii

TABLE OF CONTENT 1

CHAPTER 1 INTRODUCTION 1

1.1 Background 1
1.2 Objectives 1
1.3 Block Diagram 2
1.4 System Overview 2

CHAPTER 2 LITERATURE SURVEY 3

2.1 Float Sensor 3
2.2 Ethernet Cable 3
2.3 Transistor 4
2.4 Relay Unit 5
2.5 Microcontroller 7
2.6 Diode 8
2.7 Liquid Crystal Display(LCD) 9

CHAPTER 3 DISCUSSION & RESULTS 10

3.1 Details on Project Activity 10
3.2 Works Accomplished 17
3.3 Problems Encountered 17
3.4 Future Plans 18

CHAPTER 4 CONCLUSION 19

APPENDIX 20

CHAPTER 1 INTRODUCTION

1.1 Background

Water is essential in every hour of our lives. A study by World Bank suggests the median water consumption in Kathmandu is below WHO's basic sufficiency level. 50% of households have barely enough water to ensure that their basic hygiene and consumption requirements are met. The Nepalese Government Central Bureau of Statistics has provided the data that daily demand of water in Kathmandu is 360 million litres, but the supply is less than half and water wastage via leakages, overflow and spillage adds up to 35% of that being supplied. This gave us motivation to prevent the wastage of water where the supply is already very less. Water management is crucial in this situation and can help to prevent the inequity in water consumption in households of Kathmandu. Hence we decided to undertake the project "Design of Water Level Indicator and Pump Controller."

1.2 Objectives

There are some objectives need to be achieved in order to accomplish this project. These objectives will act as a guide and will restrict the system to be implemented for certain situations:

1. To develop water level indicator, the indicator can be leds of varying colors indicating different water levels or a LCD for digital display.
2. To choose a sensor that is water-philiac.
3. To develop an algorithm that can be custom tailored to control any number of pumps and monitor level of water in any number of tanks.
4. To design a control board that has robust relay unit, sensor input unit, indicator unit and an output unit.
5. To prepare a product that can be implemeneted in households.

1.3 Block Diagram

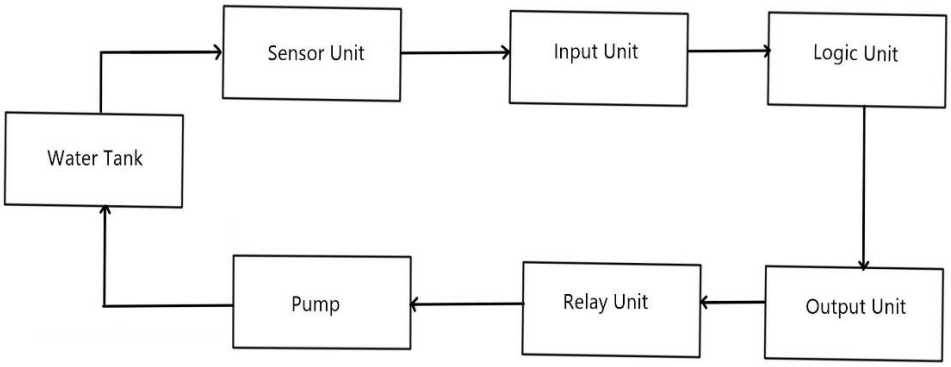

Figure 1 Block Diagram

1.4 System Overview

The system comprises of distinct sensor unit, input unit, logic unit, output unit and relay unit. The sensor unit includes plastic float sensors which are water-philiac. These float sensors detect the presence of water in the tank. Combination of float sensors is used to detect full, medium or empty as levels of water in the tank. The input unit is a ethernet wire and RJ-45 connector which is used to bridge the sensor inputs to the logic unit. As 5V input is fed to the float sensor, in the presence of water at a certain level, the float sensor inputs 5V to the logic unit via the ethernet wire. Ethernet cable can be used to connect maximum number of sensor inputs to the logic unit with almost 0V drop at a significant distance. Also the input unit includes pushbuttons for manual operation and selector switch to toggle manual and auto operation modes. Depending on the necessity of the workload, the logic unit is of 2 different types; transistor based logic unit to drive the indicator leds and single motor for single tank, micrcontroller based logic unit to detect water level in network of tanks and drive 2 or more number of pumps. Microcontroller based logic unit can be scaled to include wireless comunication between the units and is adaptable to changes as the logic can be programmed as and when needed by the user. The output unit is the indicator unit which indicates the level of water in the tank. This includes either leds or a LCD display

used to display levels of water in the tank. The relay unit is combination of relays used to drive the pumps.

CHAPTER 2 LITERATURE SURVEY

2.1 Float Sensor

Float sensors are side-mounted through a hermetically sealed hole on the side of a tank. When the level of the tank's liquid rises or falls past the level of the float, the float will rise and fall with it, activating the switch. Wires then send an electric signal to the outside of the tank through the watertight seal, making these components essential for accurate monitoring or regulation of water levels. Electric signal for the sensor is 5V in our case.

Figure 2 Float Sensor

2.2 Ethernet Cable

The Ethernet cables used are those used for connectivity in most office and home local area networks which rely on twisted wire pairs within an overall cable. Twisting of wires together enables the currents to balance, i.e in one wire the current is moving in one direction, and in the other wire of the pair the current is going in the other, enabling the overall fields around the twisted pair to cancel. This helps sensor inputs to be transmitted to considerable lengths without communication interferences. There are several categories of Ethernet cables e.g. Cat 5 cables, Cat 6 cables, Cat 7 and Cat 8 etc, but a Cat 5 cable suits our work since we just want minimal drop of voltage over the length of cables and speed is not our top priority.

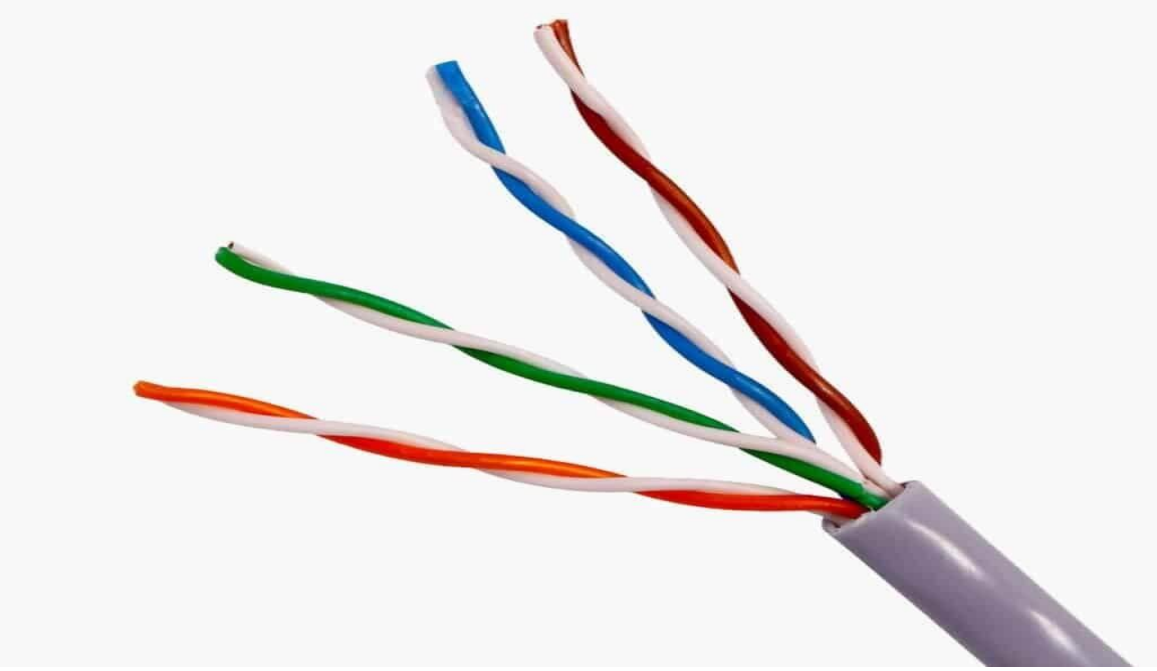

Figure 3 Ethernet Cable

2.3 Transistor

Transistor is a semiconductor device used to amplify or switch electric signals and power. Transistor in our project is used as a switch to build the logic to control the relay unit. BC547 and BC557 is used to build the transistor logic. BC547 is a NPN transistor with three pinouts; collector, base and emitter. BC557 is a PNP transistor with three pinouts; emitter, base and collector. In a NPN transistor, the collector and emitter is left open (Reverse biased) when the base pin is held at ground and is closed (Forward biased) when a signal is provided to base pin. In a PNP transistor, the collector and emitter is left open (Reverse biased) when the base pin is provided a signal greater than voltage provided at emitter and is closed (Forward biased) when base pin is held to ground or provided voltage less than the provided at emitter. When a transistor is used as a switch it is operated in the Saturation or Cut-off region. Both NPN and PNP transistors act as Open switch during Reverse Bias and as a closed switch during Forward Bias, this biasing can be achieved by supplying the required signal in the base pin. Since our control operation requires 5V to drive the output units, BC547 and BC557 suits well for our system.

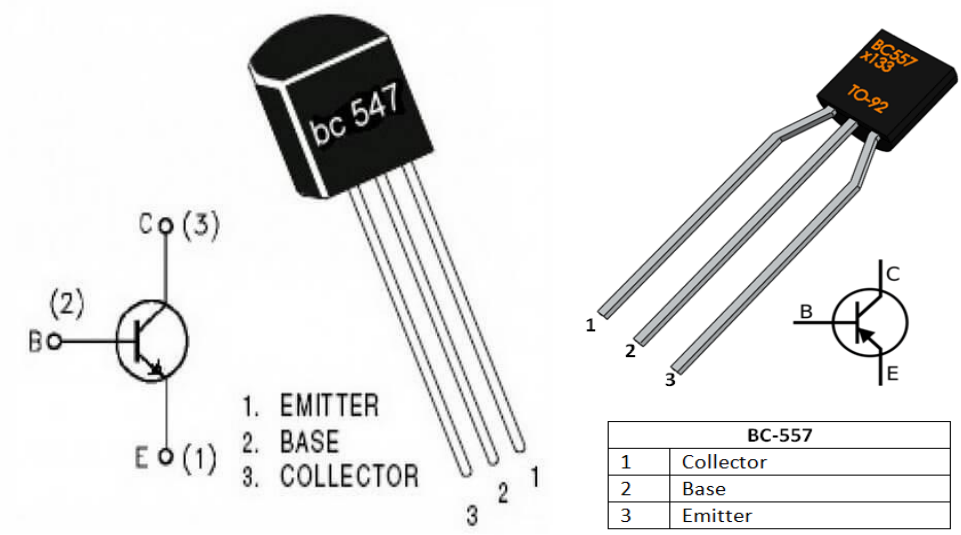

Figure 4 NPN and PNP Transistor

2.4 Relay Unit

Relay unit comprises of 5V relays and 12V relays, their number decided by the number of pumps needed to be controlled. A relay is an automatic switch that is commonly used in an automatic control circuit to control a high-current using a low-current signal. The input voltage of the 5V relay signal ranges from 0 to 5V and that of 12V relay ranges from 0 to 12V. The pin configuration of 5V relay is shown in the figure.

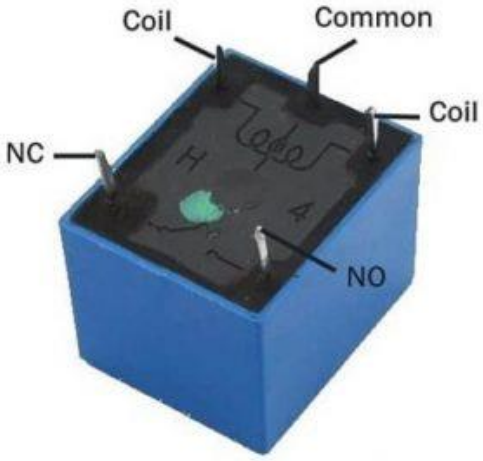

Figure 5 Relay

Coil: Coil represents the two ends of the coil in a relay, which is energised when operating voltage is applied at them. One end of the coil is connected to positive end of dc voltage while the other end is connected to ground.

Common: This pin is provided with the control signal the relay needs to transfer to activate the loads.

Normally Closed (NC): Normally Closed pin in the relay is already connected to the common pin of the realy before the relay is energised. One terminal of load is connected to either NC/NO pins.

Normally Open (NO):Normally Open pin as name suggests is that pin that is initially not connected to the common pin of the relay. It connects to the common pin of relay only when relay is energised.

Normal operating voltage for 5V relay unit is 5V, normal current is 70mA, maximum AC load current is 10A at 250V AC. Maximum DC load current is 10A at 30V DC.

Figure 6 12V relay

12V relays have similar coil and pin arrangements as 5V relay; however they have double the number of pins compared to 5V relays and have higher current and voltage ratings. Higher current and voltage ratings make 12V relays more reliable and secure for our project. 12V relay is actuated by the control signal from 5V relay unit. Then the load is actuated by the 12V relay unit.

2.5 Microcontroller

A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip. It is embedded inside of a system to control a singular function in a device. It does this by interpreting data it receives from its I/O peripherals using its central processor. The temporary information the microcontroller receives is stored in its data memory, where the processor accesses it and uses instructions stored in its program memory to decipher and apply the incoming data. The incoming data is fed by the sensor unit to the microcontroller. The core elements of a microcontroller are:

- The processor: The processor processes and responds to various instructions that direct the microcontroller's function. This involves performing basic arithmetic, logic and I/O operations, which communicate commands to other components in the larger embedded system.
- Memory: A microcontroller's memory is used to store the data that the processor receives and uses to respond to instructions that it's been programmed to carry out. It has two main memory types:

  Program Memory: It stores long-term memory information about the instructions that the CPU carries out. It is non-volatile memory, meaning it holds information over time without needing a power source.

  Data memory: It is temporary storage unit which holds the incoming data from the outside environment when the instructions are being executed. It is volatile, meaning the data in memory is maintained only in the availability of a power source.

- I/O peripherals: The input output devices are the interface for the processor to the external environment. The input ports receive information and send it to the processor in the form of binary data. The processor receives the data and sends the necessary instructions to output devices via the output ports of a microcontroller.

  Depending on the scale of load, flexibility requirements and complexity of project, we also opted to keep microcontroller based logic as our logic unit. Arduino Mega 2560 is used as our microcontroller board. It has higher performance, more number of input output pins that meet our project needs so Mega 2560 was optimal for our project over other Arduino boards. It has 54 digital I/O pins, 16 analog inputs, 4 UARTs, 16MHz oscillator, a USB connection, a power jack and a reset button. It operates in 5-12V range, has 256KB of flash memory, 8KB of SRAM, 4KB of EEPROM.

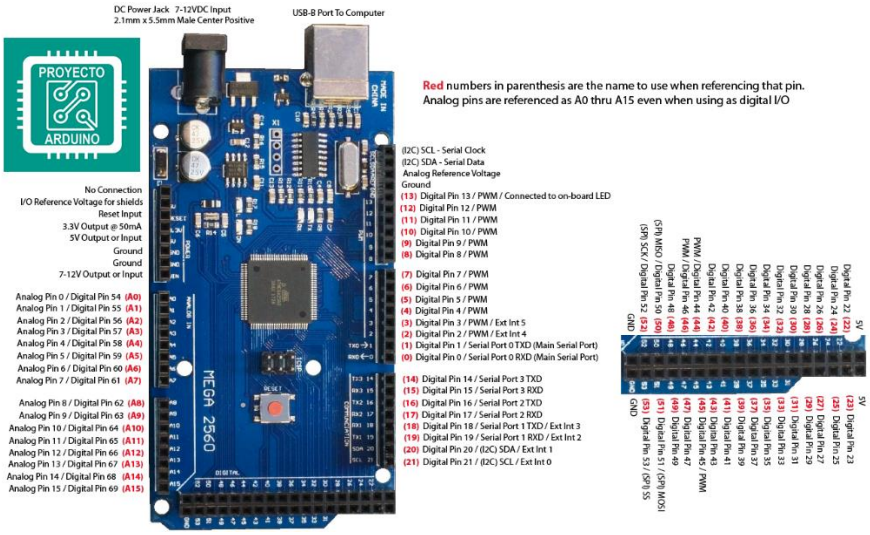

Figure 7 Microcontroller

2.6 Diode

Diode is an electrical component that allows the flow of current in only one direction. The most common type of diode uses p-n junction. In this type of diode, one material (n) has electrons as charge carriers, while second material (p) has holes (places depleted of electrons which act as positively charged particles) as charge carriers. At the interface of these two materials, a depletion region is formed across which electrons diffuse to fill holes in the p material. The depletion region restricts the flow of electrons to the p side. When the junction is forward biased (that is positive voltage is applied to p side), electrons can easily move across the junction to fill the holes, and a current flows through the diode. When the junction is reverse biased, (that is, a negative voltage is applied to the p side), the depletion region widens and electrons can not easily move across. The current remains significantly small and does not increase until a breakdown voltage is reached. Diode in our project is used as a flyback diode that eliminates the flyback, that is it eliminates the sudden voltage spike seen across an inductive load when its supply current is suddenly reduced or interrupted. Since our load is motor (pump), it is an inductive load and it is controlled by relays which itself is inductive. So, a diode is implemented as flyback component in our circuit. 1N4007 has maximun current capacity of 1A. The power dissipation is 3W.

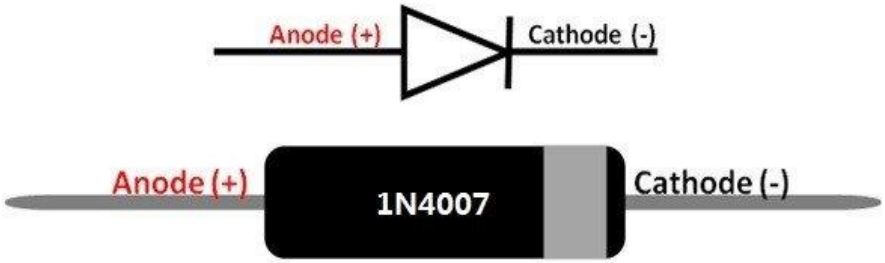

Figure 8 Diode

2.7 Liquid Crystal Display(LCD)

LCDs are parallel interface display units compatible with arduinos that outputs individual ASCII characters with fixed size. Using these individual characters we can form a text. The display is composed of small rectangular areas of 5*8 pixels grid. Each pixel can light up individually, so we can generate characters within each grid. The number of rectangular areas define size of the LCD. Our project employs 16*2 LCD grid for single tank, single pump systems and 20*4 LCD grids for multiple tanks, multiple pumps systems. Since LCDs have parallel interface, microcontroller must manipulate several pins at once to control the display. LCDs interface has following pins:

- Register Select (RS) pin- enables where in LCDs memory we are writing data to.
- Read/Write pin: selects reading or writing mode.
- Enable pin: enables writing to registers
- 8 data pins (D0-D7): states of these pins (high or low) determines what data we are writing to the lcd data register.

Figure 9 LCD

CHAPTER 3 DISCUSSION & RESULTS

3.1 Details on Project Activity

At the very beginning, we were focused on logic unit of our system. The logic unit implemented PNP and NPN transistors as switches to derive the logic. We started our project with operating transistors as switches, it was a learning procedure so we operated both BC547 and BC557 in cut off and saturation modes, and understood the forward and reverse biased characteristics of both the transistors. To operate transistors as switches in common emitter configuration, both collector base and emitter base junction must either be forward biased or reverse biased. A simple led was used as an indicator to realize the switching nature of transistors. NPN transistor required a certain bias voltage that would inject a bias current from base to emitter enough to operate the transistor in saturation mode. In the saturation mode, NPN transistor (BC547) behaved as a closed switch. While in the absence of enough biasing current, NPN transistor operated in cut off region. In the cut off region, it behaved as a open switch. The major takeway was it is always preferred to connect the load in collector terminal of NPN transistor. If the load is connected at the emitter terminal of NPN transistor, it limits the biaisng current fed by the biasing voltage, thus the resulting current might not be enough to switch on the transistor. And further analysis of circuit might be required. Therefore in NPN transistor common emitter configuration, we opted to connect the load in collector terminal and implementing KVL, the biasing current requierd was figured out. In PNP transistor, the situation is different, PNP transistor (BC557) requires ground to be connected to the base terminal of transistor so that injecting current from emitter(P) flows to base(N) of the transistor and it operates in saturation mode thus transistor is switched ON or closed. While in the absence of ground in the base terminal of PNP transistor, there would be no injecting current from emitter to base. This leads transistor to operate in cut off region and the switch is open. The major takeaway was to connect the load in collector terminal of PNP transistor common emitter configuration, load in the emitter terminal of PNP would limit the biaising current and more voltage must be fed to reach the requierd biasing current. This disturbs the circuit stability.

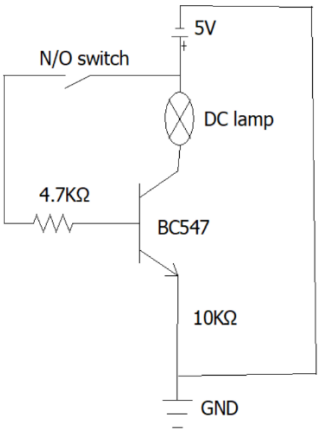

Figure 10 NPN transistor as a Switch Common Emitter Configuration

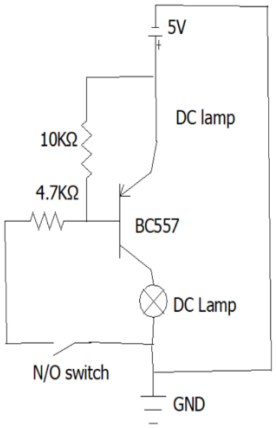

Figure 11 PNP transistor as a Switch Common Emitter Configuration

After the transistors switching states and characteristics were assesed, we derived the logic we wanted to implement in the project. The goal was to automatically switch ON the pump when water in the overhead tank falls below lower sensor level and turn the pump OFF only when water in the overhead tank reaches the upper sensor level. However, we opted to address further practical issues seen in water consumption scenario of Kathmandu. Almost every house has a reserve tank to store the water which is then pumped to the overhead tank. In the absence of water in the reserve tank, the pump is switched OFF even if the water in overhead tank falls below lower sensor level. To include this scenario, we implemented two NPN transistors and one float sensor, the float sensor was side mounted in the reserve tank at a certain level such that it is enough to fill the overhead tank. However if water falls below that sensor level in the reserve tank, no water is pumped in overhead tank. The system is reserve tank could be made more stable using two sensor levels however a large reserve tank compared to pump's output and size of average overhead tank, one

sensor arrangement operates equally well. Thus reserve tank and overhead tank's dependency on it was addressed to incorporate practical scenario.

In the overhead tank, two sensors were used, high level sensor to indicate upper level of water in the tank, low level sensor to indicate lower water level in the tank. Two transistor one NPN and other PNP were used for both high and low level sensors. From the figure below, we can see how the logic we wanted in our system is derived. Once the reserve tank's sensor detects enough water level, it feeds the biasing voltage to the BC547 (1). The 4.7KΩ resistor limits the current to required biasing current enough to operate the transistor(1) in saturation mode. Once this transistor(1) operates in saturation mode, collector and emitter are short circuited. Thus, 6V from collector of transistor(1) switches to its emitter which is then fed to BC547(2) via 4.7KΩ resistor. This injects enough biasing current to operate transistor(2) in saturation mode. Then the collector and emitter of transistor(2) are short circuited. 6V from collector of transistor(2) reaches to its emitter which is fed to emitter of transistor BC557(4). However this 6V from BC557(4) emitter is not yet switched to its collector. Now, lets observe the overhead tank, in the overhead tank, if the tank is full i.e, if the upper sensor level is met by water level then the 6V from sensors are fed to the BC547(3), which injects biasing current to the transistor(3) and the collector and emitter of transistor(3) is short circuited. Thus 6V from collector of transistor(3) reaches to emitter which is fed to the base of transistor (4) BC557. Since BC557 is PNP transistor, due to 6V at its base terminal and 6V which initialy arrived at its emitter terminal via reserve tank's transistors, it won't switch ON. BC557 being a PNP transistor, only switches ON when there is ground at its base terminal. So, when the water level in overhead tank falls below upper sensor level, the upper level sensor now does not feed 6V to BC547(3). This leads BC547(3) to switch off, it now operates in cut off region. Thus ground which is already connected to emitter of BC547(3) via 10KΩ resistor is now fed to base terminal of BC557(4). Ground at base terminal of BC557(4) and 6V at its emitter causes it to inject enough biasing current from emitter to its base, now BC557(4) operates in saturation mode and is switched ON. Its emitter and collector are short circuited, 6V from its emitter reaches to its collector. And then 6V reaches to emitter of BC557(6) as its emitter is physically tied to collector of BC557(4). However this 6V from emitter of BC557(6) deoes not reach its collector and to the realy. To achieve this, the water level in the tank must fall below lower sensor level. Once the water in the tank falls below lower sensor

level, BC547(5) is switched off, ground tied to its emitter via 10KΩ resistor is fed to BC557(6). BC557(6) being a PNP transistor, has ground at its base and 6V at its emitter terminal so it injects biasing current from emitter to its base and operates in saturation mode. It is switched ON. Thus 6V from BC557(6)'s emitter reaches to its collector and then to the coil of relay as shown in the figure below. 6V and ground is between relays coil, the relay's Common point switches from Normally Cosed (N/C) to Normally Open (N/O). The 6V relay has a pair of Common N/C N/O arrangements. In the upper Common N/C N/O arrangement, as we see in figure below, the N/O terminal already has 6V from BC557(4)'s collector terminal and its common terminal is connected to relay's coil. This arrangement is done to latch the relay even if water reaches the lower sensor level and keep the pump turned ON until water reaches upper sensor level. So, once the relay turns ON, the 220VAC from lower common terminal of relay connects to lower N/O terminal of relay and 220VAC is fed to the pump and pump turns ON and fills the overhead tank. First the water level reaches the lower sensor level, this causes lower sensor level to feed 6V to base terminal of BC547(5). BC547(5) now operates as closed switch, it feeds 6V from collector to its emitter which reaches to base terminal of BC557(6). 6V at base terminal of BC557(6) causes it to operate in cut off mode. So 6V is no longer available at its collector terminal and relay's coil does not get its 6V from there. But pump needs to remain ON until water reaches upper sensor level. So the previously discussed latching of relay is done. As we see in figure, once the relay was switched ON, its common and N/O were connected, the upper N/O is physically connected to collector of BC557(4) which still has 6V. This 6V is fed to relay's coil via N/O to Common terminal to relay's coil as seen in figure. Thus, even if water level goes above lower sensor level, pump does not turn off. Now pump gradually fills the overhead tank, and as upper sensor level is reached, 6V from sensor is fed to base of BC547(3) and it injects biasing current enough to operate BC547(3) in saturation mode. 6V from its collector reaches its emitter and this 6V from its emitter reaches to base of BC557(4) which causes BC557(4) to switch off. Now its collector and emitter are opened so 6V is not reaching its collector and upper N/O terminal of relay has no 6V thus upper common terminal of relay has no 6V which leads 0V at relay's coil. This turns off the relay, common terminals of relay are now connected to N/C terminals and pump no longer gets its 220VAC. The pump is turned off and water is not fed to overhead tank and prevents the overflow. This way we got our logic implemented in the system. The sensor wires connecting sensors and trnasistors are ethernet cables.

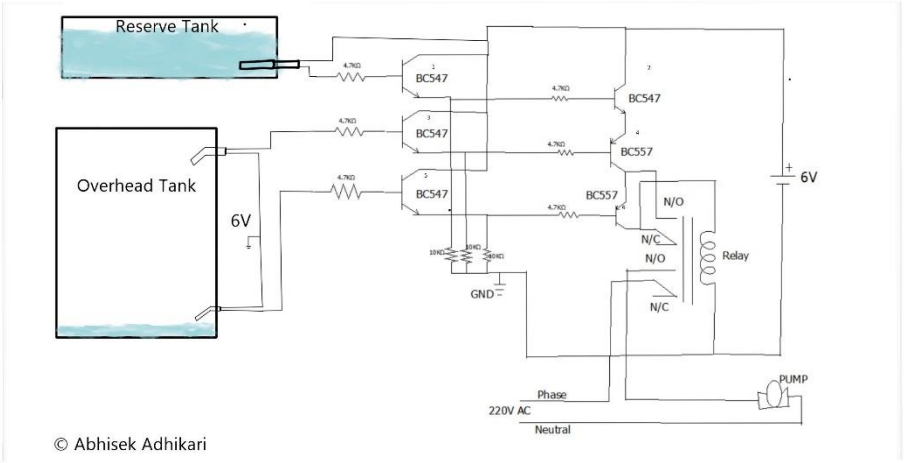

Figure 12 Overall Arrangement of the System and Circuit

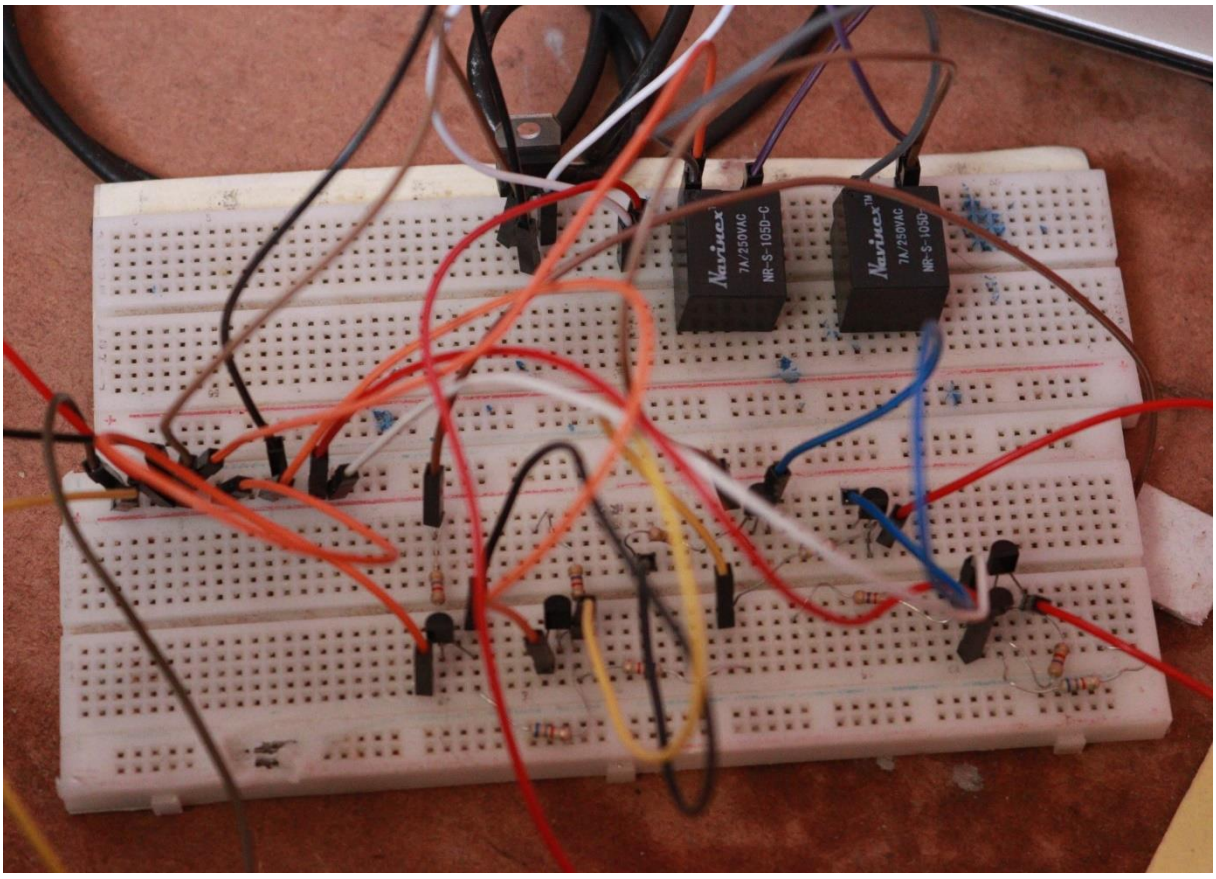

Figure 13 Transistor Logic Circuit

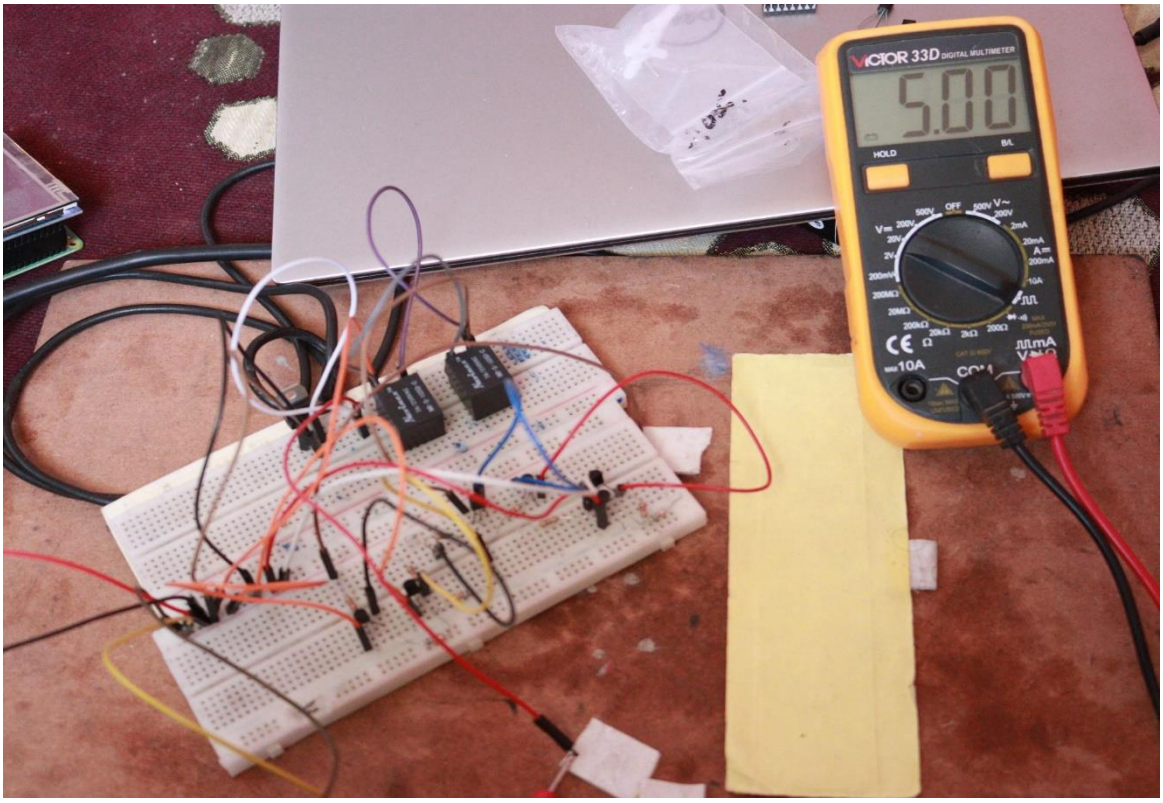

Figure 14 Live Circuit

There was always a problem of being less flexible to the changes when we adopt hardware based logic. That was encountered with transistor logic system. It was difficult to change the logic if required. Also, it consumed time to rectify problems if they were encountered. System was also more susceptible to failure if any unwanted disturbances occurred. Therefore, we also developed programmed logic to our system. The progrommable logic is flexible to changes, more stable but bit expensive. However flexibilty is a good trade off. A control board implementing power unit to the microcontroller unit, relay unit, sensor unit was designed using KiCAD.

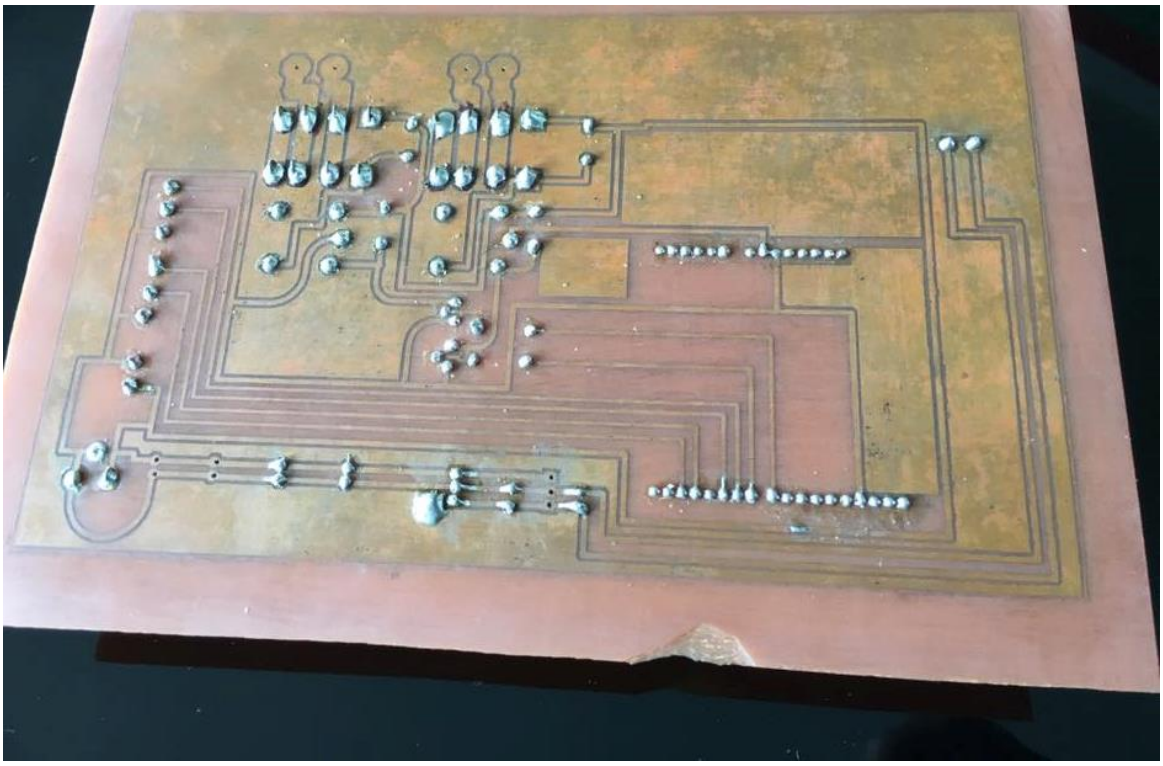

Figure 15 PCB Layout of Circuit

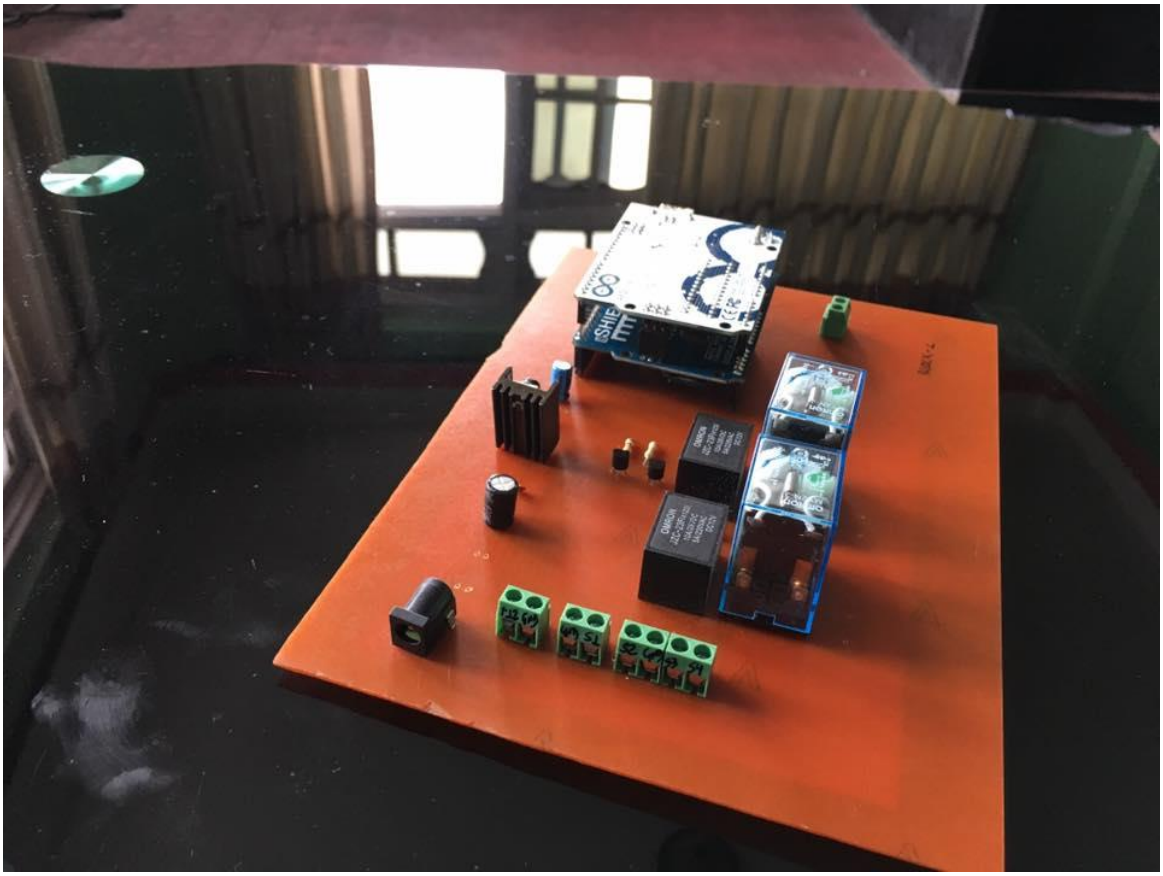

Figure 16 Demo Circuit with Arduino Uno and Ethernet Shield

Still, several improvements can be made in programmable logic system. PCB layout can be finer, inclusive of sensor inputs, can still accommodate a power supply unit within it such that system can be commercialized as a complete product. However, the above demonstrated sysem worked equally fine for the project.

Figure 17 Display Status

3.2 Works Accomplished

- The logic for the system was successfully developed.
- Logic was effectively implemented in the system.
- Successfull testing of circuit was done.
- Pump switching under different scenarios were tested.
- Basic PCB layout of circuit was made and tested.
- Improvements an be done to the system was realised.

3.3 Problems Encountered

- Common Grounding of system's power units was not effectively implemented, it was a major issue in the beginning.
- Implementing PNP and NPN transistors together and loading either emitter or collector was a confusion.
- Flywheel Diode's role was not noticed in the beginning.
- Transistor's loading capacity was an issue.
- PCB design was not effective.

3.4 Future Plans

The circuit implemented basic load switching algorithm, however system can further include wireless load switching, the system can be a standalone system with its own power supply unit. The PCB design of control board can be made more robust, it can include multiple relay units, sensor input units. Our future plans include improvising our system addressing above mentioned shortcomings, and making our system a stable product.

CHAPTER 4 CONCLUSION

This project is an outcome of group effort under the supervision and regular guidance of the supervisor along with the suggestions from teachers and seniors. This project would not have been up to this level without their parts.

A whole year has come to an end doing this project. Many things were taught in the classroom about electronic devices and logics. It is due to this project, many of those teachings are implemented and the level of understanding has come to net level. As mentioned earlier, many problems were encountered during the course of this project and many things were learned while solving those problems. Many things will be learned in days to come when the unsolved problems are solved. Moreover, time management became a denting task in the beginning and that was solved up to some extent when all the members cooperated at group works. Along with technical learning, team work was also a major learning. To complete the project as per the plans throughout the year has been quite an achievement.

It is expected that the project will be improvised in a way that can solve water wastage problems and can aid to reduce the inequity in water consumption. With the combine efforts from the group members, supervisors and teachers, it is hoped this Water Level Indicator and Pump Controller will be launched in the market as a commercial product.

APPENDIX

Arduino Code

```
    /* Tanks
 Half Orange            35/tank1e
 Orange                37/tank1f
 Half Green (sensor unit) 33
 Blue                  Ground
 Half Blue              31
 Green                 25
 Half Brown             29
 Brown                 27  */


#include <LiquidCrystal.h>


        LiquidCrystal lcd(4, 5, 9, 10, 11, 12); // rs, E , d4-d7

        const int  buttonPinmode = A15;
        int buttonState = 0;

        int manual = 0;
        int automode = 0;

        int buttonPushCounter1;
        int buttonState1;
        int lastButtonState1;

        int buttonPushCounter2;
        int buttonState2;
        int lastButtonState2;

        int buttonPushCounter3;
        int buttonState3;
        int lastButtonState3;
```

```
String a, b, c, d;
String a1, b1, c1,d1;
int togglesystem = 1;
int togglesystem1 = 1;
int togglesensor;

int togglebuzzer;


unsigned long timesystem = millis();
unsigned long timesystem1 = millis();
unsigned long timesystemdisplay = millis();
unsigned long timesystemsensor = millis();

 unsigned long timebuzzer = millis();

  int buzz=500;
int l;

const int Systemindicator = 47;

int i;
long j = 0;
long k = 0;



                //INPUT SWITCHES for INDIVIDUAL PUMPS
int manual_on1 = A11;
int manual_on2 = A12;
int manual_on3 = A13;
int sensor_indication = 33;



                        //OUTPUT
int relaypin1 = A0;
int relaypin2 = A1;
```

```
int relaypin3 = A2;
int led_1 = 23;
int led_2 = 39;
 int buzzer = 41;



                //SENSOR TANK 1
int sensor_empty1 = 35;
int sensor_full1 = 37;

                //SENSOR TANK 2
int sensor_empty2 = 31;
int sensor_full2 = 25;

                //SENSOR TANK 3
int sensor_empty3 = 29;
int sensor_full3 = 27;

                //SENSOR UNDERGROUND

int sensor_reserve = 45;
void setup()
{


    lcd.begin(20, 4);
    lcd.clear();

    lcd.setCursor(0, 0);
    lcd.print("     Automatic    ");
    lcd.setCursor(0, 1);
    lcd.print("     Water Pump ");
    lcd.setCursor(0, 2);
    lcd.print("     Controller  ");

    delay(2000);
```

```
 lcd.clear();
 lcd.setCursor(0, 1);
 lcd.print("Starting System");
 delay(900);
 lcd.setCursor(0, 2);
 lcd.print("Please Wait");
 delay(800);


lcd.setCursor(11, 2);
lcd.print(".");
delay(700);

lcd.setCursor(12, 2);
lcd.print(".");
delay(700);

lcd.setCursor(13, 2);
lcd.print(".");
delay(700);

lcd.setCursor(14, 2);
lcd.print(".");
delay(600);

lcd.setCursor(15, 2);
lcd.print(".");
delay(500);

lcd.setCursor(16, 2);
lcd.print(".");
delay(400);

lcd.setCursor(17, 2);
lcd.print(".");
delay(300);

lcd.setCursor(18, 2);
```

```
 lcd.print(".");
 delay(200);

 lcd.setCursor(19, 2);
 lcd.print(".");
 delay(100);

 lcd.setCursor(20, 2);
 lcd.print(".");
 delay(100);


lcd.clear();
Serial.begin(9600);

pinMode(buttonPinmode, INPUT);
digitalWrite(buttonPinmode, HIGH);

pinMode(Systemindicator, OUTPUT);
digitalWrite(Systemindicator, LOW);

pinMode(relaypin1, OUTPUT);
digitalWrite(relaypin1, LOW);

pinMode(relaypin2, OUTPUT);
digitalWrite(relaypin2, LOW);

pinMode(relaypin3, OUTPUT);
digitalWrite(relaypin3, LOW);

pinMode(led_1, OUTPUT);
digitalWrite(led_1, LOW);

pinMode(led_2, OUTPUT);
digitalWrite(led_2, LOW);

   pinMode(buzzer, OUTPUT);
digitalWrite(buzzer, LOW);
```

```
pinMode(manual_on1, INPUT);
digitalWrite(manual_on1, HIGH);

pinMode(manual_on2, INPUT);
digitalWrite(manual_on2, HIGH);

pinMode(manual_on3, INPUT);
digitalWrite(manual_on3, HIGH);

            //Initialization of Sensor of tank 1

 pinMode(sensor_full1, INPUT);
 digitalWrite(sensor_full1, HIGH);

 pinMode(sensor_empty1, INPUT);
 digitalWrite(sensor_empty1, HIGH);

            //Initialization of Sensor of tank 2

 pinMode(sensor_full2, INPUT);
 digitalWrite(sensor_full2, HIGH);

 pinMode(sensor_empty2, INPUT);
 digitalWrite(sensor_empty2, HIGH);

            //Initialization of Sensor of tank 3

 pinMode(sensor_full3, INPUT);
 digitalWrite(sensor_full3, HIGH);

  pinMode(sensor_empty3, INPUT);
  digitalWrite(sensor_empty3, HIGH);


          //Initialization of Reserve Tank's Sensor

    pinMode(sensor_reserve, INPUT);
```

```
            digitalWrite(sensor_reserve, HIGH);

              pinMode(sensor_indication, INPUT);
              digitalWrite(sensor_indication, HIGH);

            }

     void loop()
    {

     if ((automode == 1 || manual == 1 || digitalRead(sensor_reserve) == HIGH) &&
(millis() - timesystem1 > 150))
    {
     togglesystem1 = ! togglesystem1;

     Serial.print(" togglesystem1: ");
       Serial.println(togglesystem1);
     digitalWrite(Systemindicator, togglesystem1);
     timesystem1 = millis();
    }

     if (digitalRead(sensor_indication) == HIGH)
   {

     togglesensor = 2;
     digitalWrite(led_1, LOW);
     digitalWrite(led_2, LOW);
   }

    if (millis() - timesystemsensor > 150 && digitalRead(sensor_indication) == LOW)
  {

     togglesensor = ! togglesensor;
     timesystemsensor = millis();
   }

    if (togglesensor == 0)
   {
```

```
    digitalWrite(led_1, HIGH);
    digitalWrite(led_2, LOW);
}
  else if (togglesensor == 1)
{
    digitalWrite(led_2, HIGH);
    digitalWrite(led_1, LOW);
}




  buttonState = digitalRead(buttonPinmode);

  if (buttonState == HIGH)
{
      automode = 1 ;
      manual = 0;
      j++;
      k = 0;


 }

    else if (buttonState == LOW)
 {
       manual = 1;
       automode = 0;
       k++;
       j = 0;
 }
```

```
    if (digitalRead(sensor_reserve) == HIGH)
  {
     digitalWrite(relaypin1, LOW);
     digitalWrite(relaypin2, LOW);
     digitalWrite(relaypin3, LOW);
     buttonPushCounter1 = 0;
     buttonPushCounter2 = 0;
     buttonPushCounter3 = 0;
     a1 = "OFF    ";
     b1 = "OFF    ";
     c1 = "OFF    ";

     lcd.setCursor(0, 0);
     lcd.print("   !!!Warning!!!  ");
     lcd.setCursor(0, 1);
     lcd.print("  MOTOR TURNED OFF");
     lcd.setCursor(0, 2);
     lcd.print("    Reserve Tank  ");
     lcd.setCursor(0, 3);
     lcd.print("       EMPTY        ");



      if (buzz<=500 && buzz != 450 && (millis() - timebuzzer > 1500)&&
digitalRead(manual_on1)==HIGH)
      {
       togglebuzzer =! togglebuzzer;
       digitalWrite(buzzer,togglebuzzer);
        timebuzzer = millis();
       Serial.print("hehe");
     Serial.println(buzz);
      buzz--;
       }

        else if (digitalRead(manual_on1)==LOW)
```

```
      {
       digitalWrite(buzzer,LOW);
       buzz=501;
            buttonPushCounter1 = 0;
      }

       if (buzz==450)
      {

        digitalWrite(buzzer,LOW);
        Serial.println("haha");

      l++;
       Serial.print("l");
       Serial.println(l);
      if (l==500)
      {
       buzz=500;
       l=0;
      }
      }
}




 else
 {
   buzz=500;
 digitalWrite(buzzer,LOW);

  if (millis() - timesystem > 2800 )
 {
    lcd.clear();
    togglesystem = ! togglesystem;
    Serial.print(" togglesystem: ");
```

```
    Serial.println(togglesystem);
    i++;
    timesystem = millis();
  }




                    //Sensor Check//
   if (digitalRead(sensor_full1) == LOW && digitalRead(sensor_empty1) == LOW)
 {
      a = "FULL        ";

 }
   else if (digitalRead(sensor_full1) == HIGH && digitalRead(sensor_empty1) ==
LOW)
 {
      a = "MEDIUM      ";

 }

   else if (digitalRead(sensor_empty1) == HIGH)
 {
      a = "LOW         ";
 }

     if (digitalRead(sensor_full2) == LOW && digitalRead(sensor_empty2) ==
LOW)
 {
      b = "FULL        ";
 }
     else if (digitalRead(sensor_full2) == HIGH && digitalRead(sensor_empty2) ==
LOW)
 {
       b = "MEDIUM      ";

 }
      else if (digitalRead(sensor_empty2) == HIGH)
```

```
    {
          b = "LOW          ";
    }

          if (digitalRead(sensor_full3) == LOW && digitalRead(sensor_empty3) ==
LOW)
    {
           c = "FULL         ";
    }
          else if (digitalRead(sensor_full3) == HIGH && digitalRead(sensor_empty3) ==
LOW)
    {
            c = "MEDIUM       ";

    }
         else if (digitalRead(sensor_empty3) == HIGH)
    {
            c = "LOW          ";
     }


           if (digitalRead(sensor_reserve) == LOW )
     {
           d = "LEVEL OK     ";
     }


  if (togglesystem == 1  )
  {

    lcd.setCursor(0, 0);
    lcd.print("Tank 1:");
    lcd.setCursor(7, 0);
    lcd.print(a);

    lcd.setCursor(0, 1);
    lcd.print("Tank 2:");
    lcd.setCursor(7, 1);
```

```
  lcd.print(b);

  lcd.setCursor(0, 2);
  lcd.print("Tank 3:");
  lcd.setCursor(7, 2);
  lcd.print(c);

  lcd.setCursor(0, 3);
  lcd.print("Reserve:");
  lcd.setCursor(8, 3);
  lcd.print(d);
 }



 else if (togglesystem == 0 && automode ==1 && digitalRead(sensor_indication) ==
LOW )
 {

  lcd.setCursor(0, 0);
  lcd.print("Auto MOTOR 1:");
  lcd.setCursor(13, 0);
  lcd.print(a1);

  lcd.setCursor(0, 1);
  lcd.print("    MOTOR 2:");
  lcd.setCursor(13, 1);
  lcd.print(b1);

  lcd.setCursor(0, 2 );
  lcd.print("    MOTOR 3:");
  lcd.setCursor(13, 2);
  lcd.print(c1);

    lcd.setCursor(0, 3);
  lcd.print("Sensor Detected     ");
```

```
  }
  else if (togglesystem == 0 && manual ==1 && digitalRead(sensor_indication) ==
LOW)
  {

   lcd.setCursor(0, 0);
   lcd.print("Manual MOTOR 1:");
   lcd.setCursor(15, 0);
   lcd.print(a1);

   lcd.setCursor(0, 1);
   lcd.print("       MOTOR 2:");
   lcd.setCursor(15, 1);
   lcd.print(b1);

   lcd.setCursor(0, 2);
   lcd.print("       MOTOR 3:");
   lcd.setCursor(15, 2);
   lcd.print(c1);

     lcd.setCursor(0, 3);
   lcd.print("Sensor Detected     ");




  }

  if (togglesystem == 0 && automode ==1 && digitalRead(sensor_indication) ==
HIGH )
  {

   lcd.setCursor(0, 0);
   lcd.print("Auto MOTOR 1:");
   lcd.setCursor(13, 0);
   lcd.print(a1);
```

```
 lcd.setCursor(0, 1);
 lcd.print("    MOTOR 2:");
 lcd.setCursor(13, 1);
 lcd.print(b1);

 lcd.setCursor(0, 2 );
 lcd.print("    MOTOR 3:");
 lcd.setCursor(13, 2);
 lcd.print(c1);

   lcd.setCursor(0, 3);
 lcd.print("Sensor Not Detected ");



}
 if (togglesystem == 0 && manual ==1 && digitalRead(sensor_indication) == HIGH)
{

 lcd.setCursor(0, 0);
 lcd.print("Manual MOTOR 1:");
 lcd.setCursor(15, 0);
 lcd.print(a1);

 lcd.setCursor(0, 1);
 lcd.print("       MOTOR 2:");
 lcd.setCursor(15, 1);
 lcd.print(b1);

 lcd.setCursor(0, 2);
 lcd.print("       MOTOR 3:");
 lcd.setCursor(15, 2);
 lcd.print(c1);

  lcd.setCursor(0, 3);
 lcd.print("Sensor Not Detected ");
```

```
    }
    if (i == 50  && millis() - timesystemdisplay > 3000 &&  (togglesystem == 0 ||
togglesystem == 1))
   {
    lcd.clear();
    togglesystem = 2;
    lcd.setCursor(0, 0);
    lcd.print("   Contact us at:   ");
    lcd.setCursor(0, 1);
    lcd.print("     9851232434     ");
    lcd.setCursor(0, 2);
    lcd.print("       GUNINA       ");
    lcd.setCursor(0, 3);
    lcd.print("    TECHNOTRONIX    ");
    i = 0;
    timesystemdisplay = millis();
   }


   if (automode == 1)
   {

      if ((j <= 20)  &&    (togglesystem == 0 || togglesystem == 1 || i == 0))

     {
      Serial.print("j ");
      Serial.println(j);
      lcd.clear();
      togglesystem = 2;

      lcd.setCursor(0, 1);
      lcd.print("      MOTOR 1       ");
      lcd.setCursor(0, 2);
      lcd.print("    Auto Mode ON    ");
     }
```

```
 if ( digitalRead(relaypin1) == LOW)
  {
    a1 = "OFF    ";

  }
  if ( digitalRead(relaypin2) == LOW)
  {
    b1 = "OFF    ";

  }

  if ( digitalRead(relaypin3) == LOW)
  {
    c1 = "OFF    ";

  }



if (digitalRead(sensor_full1) == LOW)
 {

   digitalWrite(relaypin1, LOW);

   a1 = "OFF    ";
 }

 else if ( digitalRead(sensor_empty1) == HIGH)
 {

   digitalWrite(relaypin1, HIGH);
   a1 = "ON     ";

 }

 buttonPushCounter1 = 0;
```

```
}


else if (manual == 1)


{
 if ( digitalRead(relaypin1) == LOW)
 {
  a1 = "OFF  ";

 }
 if ( digitalRead(relaypin2) == LOW)
 {
  b1 = "OFF  ";

 }

 if ( digitalRead(relaypin3) == LOW)
 {
  c1 = "OFF  ";

 }

 if (( k <= 10) && (togglesystem == 0 || togglesystem == 1 || i == 0))

 {
  lcd.clear();
  togglesystem = 2;

  lcd.setCursor(0, 1);
  lcd.print("      MOTOR 1       ");
  lcd.setCursor(0, 2);
  lcd.print("   Manual Mode ON   ");


 }
```

```
if ( digitalRead(relaypin1) == LOW)
{
  a1 = "OFF  ";

 }
 if ( digitalRead(relaypin2) == LOW)
 {
  b1 = "OFF  ";

 }

 if ( digitalRead(relaypin3) == LOW)
 {
  c1 = "OFF  ";

 }

  buttonState1 = digitalRead(manual_on1);
  delay(50);

 if (buttonState1 != lastButtonState1)
 {

  if (buttonState1 == LOW)
  {
    if (buttonPushCounter1 == 2)
    {
      buttonPushCounter1 = 0;
    }
    buttonPushCounter1++;
  }
  else
  {
    Serial.println("off");
  }
     }

 lastButtonState1 = buttonState1;
```

```
    if (buttonPushCounter1 == 2 || digitalRead(sensor_full1) == LOW )
    {
      a1 = "OFF  ";
      digitalWrite(relaypin1, LOW);
      buttonPushCounter1 = 0;
    }
    else if ( buttonPushCounter1 == 1  && (digitalRead(sensor_empty1) == HIGH ||
digitalRead(sensor_empty1) == LOW))
    {
      digitalWrite(relaypin1, HIGH);
      a1 = "ON   ";
    }

  }

  if (automode == 1 || manual == 1)
  {

      buttonState2 = digitalRead(manual_on2);
      delay(50);
    if (buttonState2 != lastButtonState2)
    {

      if (buttonState2 == LOW)
      {
        if (buttonPushCounter2 == 2)
        {
          buttonPushCounter2 = 0;
        }
        buttonPushCounter2++;
      }
      else
      {
        Serial.println("off");
      }
     }

    lastButtonState2 = buttonState2;
```

```
    if (buttonPushCounter2 == 2 || digitalRead(sensor_full2) == LOW )
    {
      b1 = "OFF  ";
      digitalWrite(relaypin2, LOW);
      buttonPushCounter2 = 0;
    }

    else if ( buttonPushCounter2 == 1  && (digitalRead(sensor_empty2) == HIGH ||
digitalRead(sensor_empty2) == LOW))
    {
      digitalWrite(relaypin2, HIGH);
      b1 = "ON   ";

    }


    buttonState3 = digitalRead(manual_on3);
    delay(50);
    if (buttonState3 != lastButtonState3)
    {

      if (buttonState3 == LOW)
      {

        if (buttonPushCounter3 == 2)
        {
          buttonPushCounter3 = 0;
        }
        buttonPushCounter3++;



      }
      else
      {
        Serial.println("off");
      }

          }
```

```
      lastButtonState3 = buttonState3;
      if (buttonPushCounter3 == 2 || digitalRead(sensor_full3) == LOW )
      {
        c1 = "OFF  ";
        digitalWrite(relaypin3, LOW);
        buttonPushCounter3 = 0;
      }
      else if ( buttonPushCounter3 == 1  && (digitalRead(sensor_empty3) == HIGH ||
digitalRead(sensor_empty3) == LOW))
      {
        digitalWrite(relaypin3, HIGH);
        c1 = "ON   ";
      }
      }
     }
  }
```