ICT 3143 Embedded Systems lab

Vth Sem B.Tech (CCE)

Miniproject report on

Proximity based Water Dispenser

Submitted by

220953438	**Aanya Shantaram**
220953428	**Ritika Salimath**
220953382	**B Abhijeeth**

Department of I & CT

MIT, Manipal

November, 2024

ABSTRACT

The proximity-sensing water dispenser is developed to dispense water when an object, such as a cup, goes into a predefined proximity range. The system uses an LPC1768 microcontroller, an ultrasonic HC-SR04 sensor for proximity detection, and relay-control of a pump so as to control the flow of water. The system monitors the depth of the water in the tank using a second ultrasonic sensor and displays the depth of water using the LCD. A "Please refill" message is shown when the water level drops to a specified critical threshold. This added feature increases the user friendliness of the system. The report describes the hardware components, software design, and techniques for implementation to achieve the goal of real-time functionality, particularly emphasizing on sensor integration, distance calculation, and interrupt handling for proximity detection.

Contents:

List of Figures:

A. Introduction:

The increasing demand for contactless devices is directly linked to growing hygiene and convenience aspects in daily usage. Products like automatic drinking water dispensers find their applications in houses as well as public areas.

The ultrasonic sensor, HC-SR04, gets input from the microcontroller LPC1768, which regulates the flow of water by controlling the water pump with a relay, and gives feedback to the user through an LCD display. The HC-SR04 ultrasonic sensor detects an object within a predefined distance, and the microcontroller interprets this signal as a request for water. Then the microcontroller activates a relay, which will switch on the water pump, the latter dispenses water and also activates the LEDs when there is an object within proximal range. The second HC-SR04 ultrasonic sensor is then used to measure and display the water level in the tank on an LCD screen by placing the sensor at the top of the water tank. In case of a reading that is below the critical water level, the system will display on the LCD screen a "Please refill" message.

Key issues addressed in this project include: accurate sensor time measurements, proximity sensing through interrupts, and easy integration of all the components used. This project illustrates an embedded system that demonstrates the feasibility of minimizing hardware solutions for intelligent and touchless systems.

B. Methodology:

a) Components Required and working principle of all the components

i) Ultrasonic sensor

The ultrasonic sensor is used to measure the distance of the object in front of it. It measures the distance by emitting high frequency sound waves and measuring the time the sound takes to reflect back to the sensor. The frequency of the sound waves emitted is too high and is inaudible to the human ear. The sensor emits such waves at regular intervals.

For this project we are using the HC-SR-04 ultrasonic sensor. It has a transmitter and a receiver for the sound waves. The following pins are present on the sensor:

1. VCC: The sensor needs to be supplied +5 V on this pin for its proper functioning
2. Trig: The trigger pin. The microcontroller has to give a 10us trigger pulse to the module.
3. Echo: Microcontroller monitors this pin to detect the presence of an obstacle in front of it.
4. GND: The ground pin

All kinds of materials can be detected irrespective of their color.

The working of the sensor can be described in the following 4 steps:

1. First a trigger pulse of atleast 10us has to be sent to the trigger pin.
2. The module then automatically sends eight 40kHz frequency sound waves and waits for the echo pin to generate a rising edge output.
3. When the rising edge is detected, we start the timer which helps us calculate the time taken by the sound waves to reflect off the object in front of the sensor and return. We then wait for a falling edge on the echo pin.
4. Once a falling edge is detected the timer is stopped and the count of the timer is read. This is then used in the calculation to measure how far the object is in front of the microcontroller. The following equation helps to calculate the required distance in cm where echoTime is the time taken for the sound wave to return to the sensor:

$$distance = \frac{(0.00343 * echoTime)}{2}$$

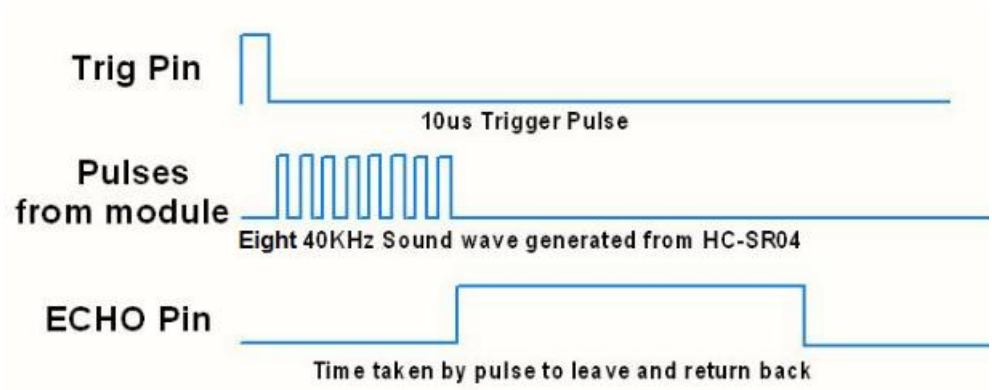

Fig2.1: Timing diagram for HC-SR-04 Ultrasonic sensor

Fig2.2: HC-SR-04 Ultrasonic Sensor

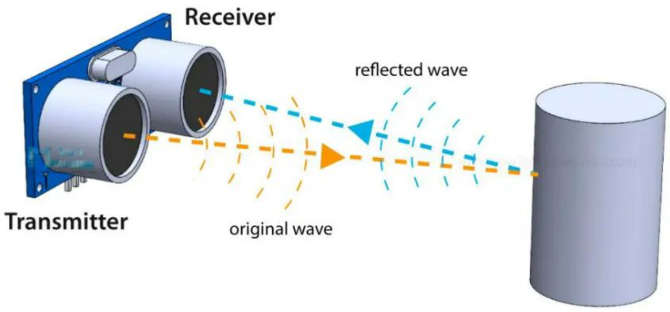

Fig2.3: Working of HC-SR-04 Ultrasonic Sensor

ii)Relay module

The relay module helps to control high-power devices with a small electrical signal by making use of the concept of electromagnetic induction. For this project the relay module is used to switch on and off our pump.

The relay module has the following pins/terminals:

1. Power pins:

- GND: It is the common ground pin
- VCC: A voltage of 5V needs to be supplied to this pin to power up the relay module

2. Control pin:

IN: this pin is used to control the relay. This pin is an active low pin. To activate the relay we need to give a logic low to the pin, and to deactivate the relay a logic high needs to be given.

3. Output terminals:

- COM: The common terminal. It is used to connect to the device to be controlled, in our project it would be the pump
- NC: Normally closed terminal. It is connected to the COM when the relay is not activated
- NO: Normally open terminal. It is connected to COM when the relay is activated

The working of the relay module can be described in the following steps:

1. A low power signal (control signal) from the microcontroller energizes the relay coil which creates a magnetic field.
2. The magnetic field attracts the armature inside the module which moves the contacts to either open or close the circuit.
3. When the contacts close the circuit gets energized and a current flows through it.

4. When the control signal is removed the electromagnet gets deactivated which opens the contacts and interrupts the flow of current.

Fig2.4: Relay module

iii)Pump

The project uses a 6V pump to dispense the water from the container. Pumps work by converting mechanical energy into pressure to move fluids from one place to another.

The working of the pump is described as follows:

1. When the circuit is completed and power is supplied to the pump, its moving part, like a piston, pushes air out of the way and creates a partial vacuum.
2. This partial vacuum is filled with the fluid in which the pump is immersed in.
3. The rotating impeller in the pump accelerates the fluid and as the pump’s casing expands the fluid’s velocity is converted into pressure.
4. The pump delivers flow by creating a pressure.

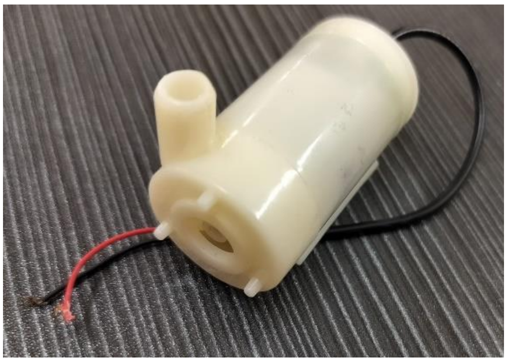

Fig2.5: Water pump

iv)Bidirectional Logic Level Converter

The project requires a bidirectional logic level converter in order to operate the relay module. The high voltage for the GPIO pin on the LPC1768 microcontroller is 3.3V but the relay module requires a control signal of 5V. Hence the logic level converter is used which helps us convert the 3.3V high of the GPIO pin to a 5V signal to be given to the relay module.

The bidirectional logic level converter uses N-channel MOSFET and pull up resistors to help convert the voltage levels. It connects a high voltage (HV) side to a low voltage (LV) side, and allows the signals to flow in either direction.

Fig2.6: Bidirectional Logic Level Converter

v)Breadboard

The breadboard allows us to connect all the electronic components with wires to create temporary circuits. The breadboard has metal strips under the plastic surface with holes provided for wires. These metal strips conduct electricity. The wires in the same row are electrically connected.

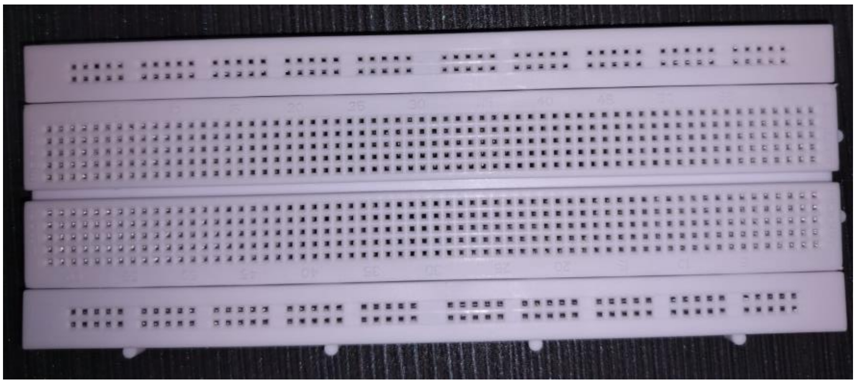

Fig2.7: Breadboard

vi) Jumper wires

Jumper wires are electrical wires which have connector pins on each end that allows us to connect two points without the need for soldering. There are 3 types of jumper wires:

1. Male to Male: both ends are male

2. Female to female: both ends are female
3. Male to female: one end is male and the other is female

In this project we make use of female- to-female, male-to-male and male-to-female jumper wires.

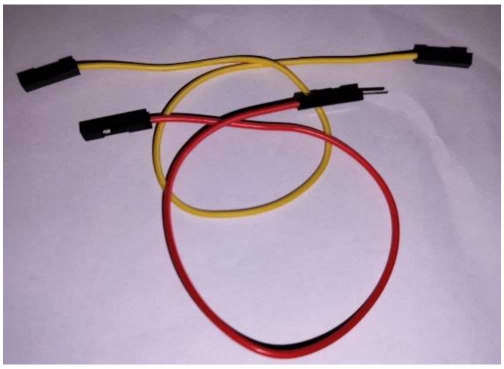

Fig2.8: Female to Female and Male to Female jumper wires

vii) LPC1768

LPC 1768 is the microcontroller being used in this project. All the above-mentioned components are connected to it. It is an ARM 32-bit Cortex-M3 microcontroller by NXP. It operates at speeds up to 100MHz. It has a wide range of peripherals including I2C, UART and SPI interfaces, PWM, ADC and DAC. It has 512KB of flash memory and 64KB of RAM. This microcontroller is known for its low power consumption and high performance. This makes it one of the top choices to be used as a microcontroller.

viii)9V transistor battery

A 9V transistor battery is used to provide the power supply required for the proper functioning of the pump. It is part of the circuit made between the relay and the pump. When the relay closes the circuit, the battery provides the necessary power to the pump which then helps dispense the water.

Fig2.9: 9V Transistor battery

b) Circuit diagram

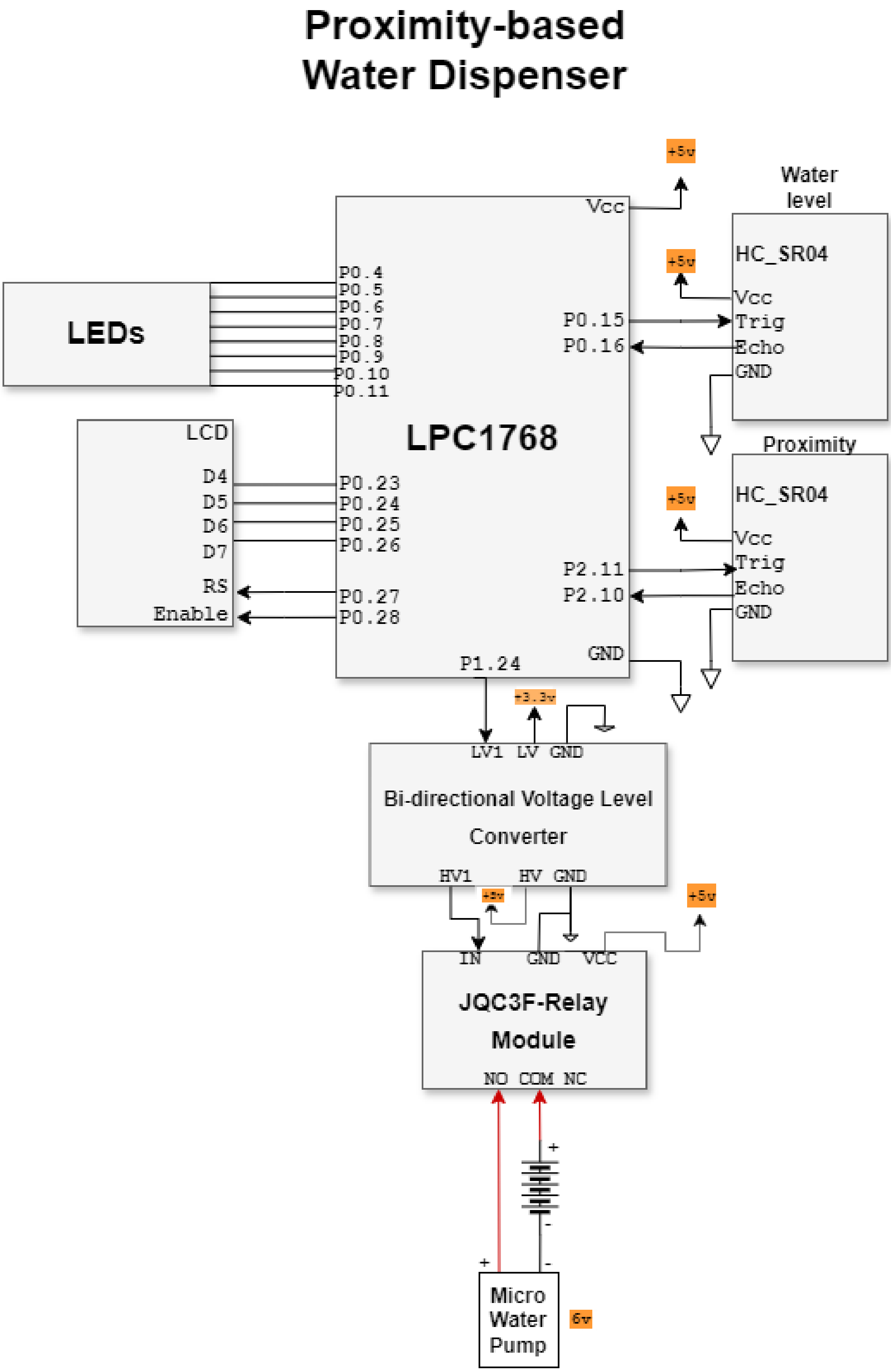

Fig2.10: Circuit Diagram

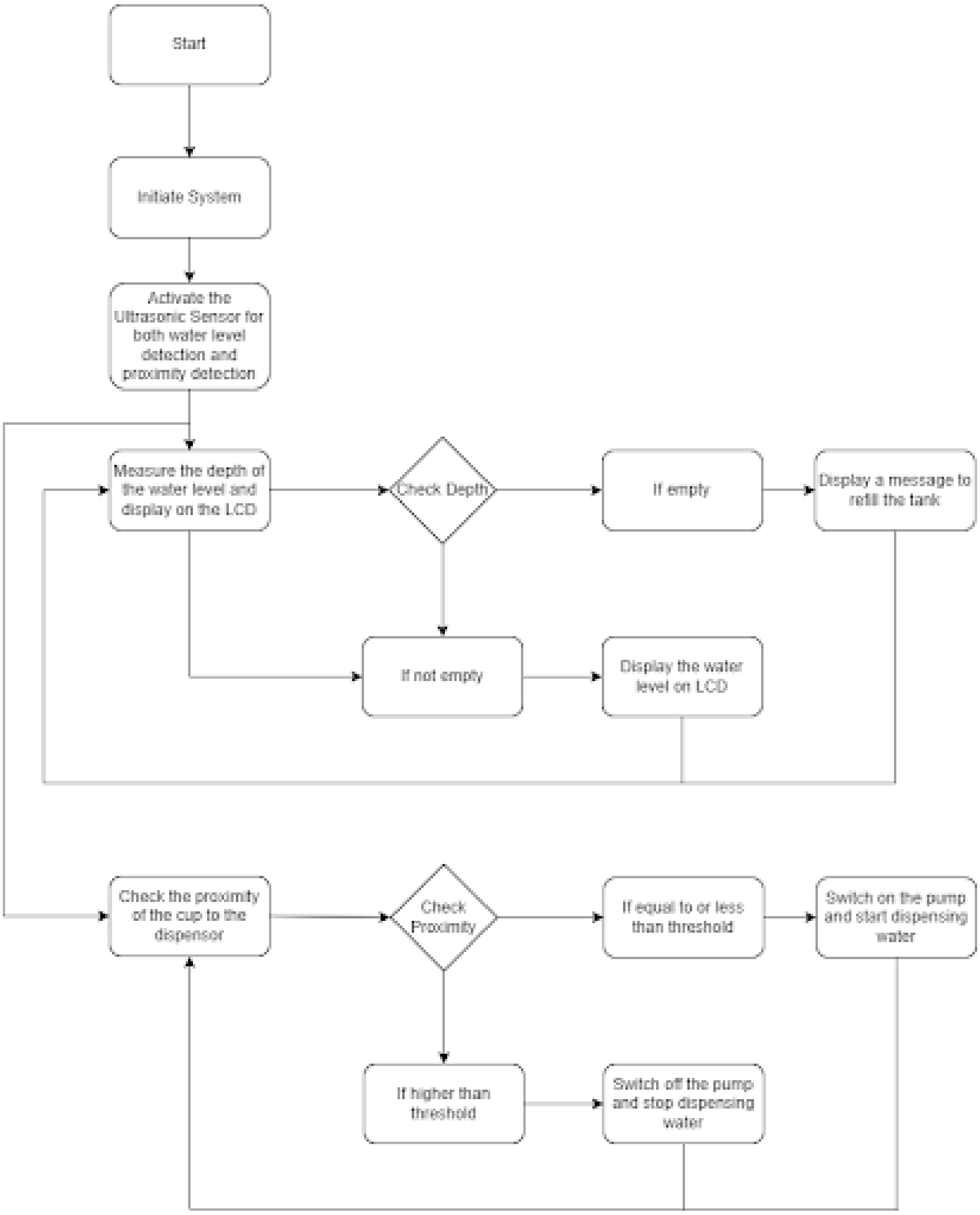

Fig2.11: Flow Diagram

c) Description about the connection

This project uses an LPC1768 microcontroller connected to the following components:

- HC-SR04 ultrasonic sensors (x2)
- LCD (part of the LPC1768 development board)
- Bidirectional voltage level shifter
- LED array (part of the LPC1768 development board)
- Micro water pump through a relay module

i. Ultrasonic Sensor for Water Level Measurement:

- Trigger pin of sensor connected to P0.15 of MCU (CNC)
- Echo pin of sensor connected to P0.16 of MCU (CNC)
- Vcc pin of sensor connected to 5v power supply
- GND pin of sensor connected GND of the MCU circuit

This sensor monitors the water level in the container by calculating the distance between the sensor and the surface of the water in the container.

ii. Ultrasonic Sensor for Proximity Detection:

- Trigger pin of sensor connected to P2.11 of MCU (CNB)
- Echo pin of sensor connected to P2.10 of MCU (CNB)
- Vcc pin of sensor connected to 5v power supply
- GND pin of sensor connected GND of the MCU circuit

This sensor detects the presence of an object, which in this case is a cup/container which will activate the pump when it is within a specified range of this sensor.

iii. Bidirectional Voltage Level Shifter:

- HV pin (High Voltage Reference) of shifter connected to 5v power supply
- LV pin(Low Voltage Reference) of shifter connected to 3.3v power supply
- GND pin of shifter connected GND of the MCU circuit
- LV1 pin of shifter connected to P1.24 of the LPC1768, which will provide the 3.3v signal for switching on/off the relay.
- HV1 pin of shifter will provide the 5v converted signal from the P1.24 pin of the LPC1768, which is now connected to the IN pin of the relay module

This level shifter converts the incoming 3.3volt signal from the LPC1768 and converts it into a 5volt signal which is required for turning on/off the relay module.

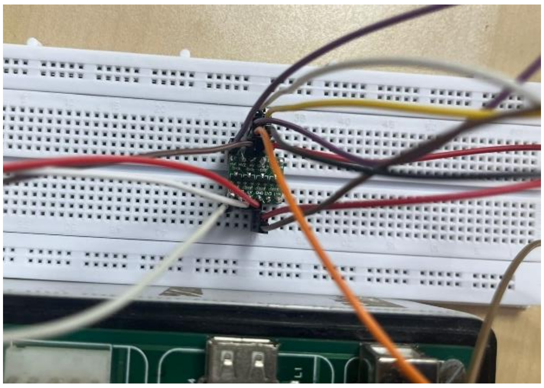

Fig2.12: Connections to the bidirectional voltage level shifter

iv. Relay Module:

- IN pin of relay connected to HV1 pin of voltage level shifter which provides converted 5v signal from pin P1.24 of the LPC1768
- GND pin of relay connected GND of the MCU circuit
- Vcc pin of relay connected to 5v power supply
- NO (Normally Open) terminal of relay connected to positive power supply wire of the pump
- COM terminal of relay connected to positive terminal of +9v transistor battery

The relay module controls the micro water pump, which is powered by a separate 9V transistor battery. This helps the low power control circuit (LPC1768) to control the high-power component (the pump).

The relay switches it on/off based on proximity detection and water level conditions.

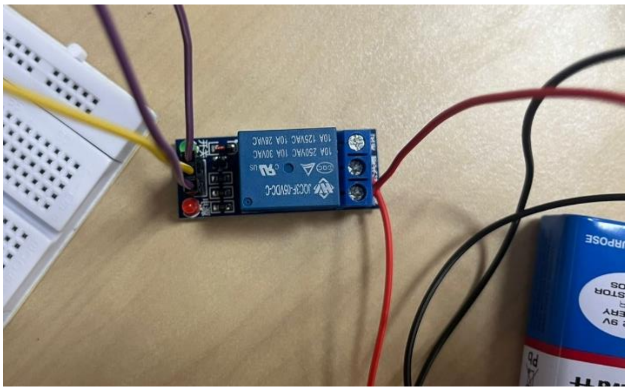

Fig2.13: Connections to the relay module

v. Battery:

- Positive terminal of battery connected to COM terminal of relay
- Negative terminal of battery connected to negative terminal of pump

This 9v battery is used to power the 6v pump.

vi. Micro pump

- Positive terminal of pump is connected to NO terminal of relay
- Negative terminal of pump is connected to negative terminal of 9v transistor battery

vii. LCD Display:

- Data pins D4, D5, D6, D7 of the LPC are connected respectively to pins P0.23, P0.24, P0.25, P0.26 of the MCU

- Register Select of LCD connected to P0.27 of MCU
- Enable pin of LCD connected to P0.28 of MCU

- Used for displaying depth of the water in the container, displaying the message “Please refill” if the container is empty.

viii. LED Array

- LEDs are connected to pins P0.4 to P0.11 of MCU

LEDs are used to indicate when the distance between the proximity ultrasonic sensor and the object (cup) is less than 40cm.

d) Method

Part 1: Ultrasonic Sensor Setup:

- The MCU generates a 10us pulse to the trigger pins of both ultrasonic sensors. For the proximity sensor, pulse is sent only when the trigger flag is 1 i.e. after an iteration of proximity detection is done.
- The pulse causes the sensors to emit ultrasonic waves at a frequency of 40 kHz. It is an 8 cycle ultrasonic burst traveling at the speed of sound.

Part 2: Distance Calculation:

- The Echo pins become high right away after the 8cycle ultrasonic burst is sent. The timer starts at this time. The ultrasonic wave travels to an object and gets reflected from it back to the sensor, which is now received by the echo pin. On receiving the wave, the echo pins become low again and the timer stops. The time interval is measured based on the time the echo pin is high and is used to calculate the distance of the object from the sensor.
- Using the formula:

$$distance = \frac{(0.00343 * echoTime)}{2}$$

The distance based on the time taken for the pulse to return is calculated.

Part 3: Interrupt Handling

- A GPIO Interrupt is configured on P2.10 (echo pin of the proximity sensor). This interrupt triggers on the rising edge of the proximity sensor’s echo pin.

Part 4: Object Proximity Detection:

- In the ISR (Interrupt Service Routine) of the GPIO interrupt, trigger flag will be set to 1 and the interrupt flag is reset. It starts a timer, measures the echo time and calculates the distance. Based on the distance measured between an object and the proximity ultrasonic sensor, the relay is controlled. When distance is less than 40cm, relay is turned on by sending logic 0 to the P1.24 pin connected to the IN pin of the relay (LEDs are also turned on at this time); otherwise, the relay is turned off by sending logic 1 to the IN pin.

- When the relay is switched on, the pump will dispense water else the pump will remain off.

Part 5: Water Level Detection:

- The second ultrasonic sensor, placed at the top of the water container is used to continuously monitor water levels. The water level is measured and then displayed (in cm) on the LCD.
- If the distance measured by this sensor is the height of the container or more (7.5cm or more in this case), the LCD displays a "Please Refill" message, alerting the user to add more water to the container.

C. Results and Discussion:

1. Real time display of the depth of water

The LCD efficiently displays the depth at which the water is present in the container with continuous updates. When the water is about to be depleted the LCD flashes a message to alert the user to refill the container.

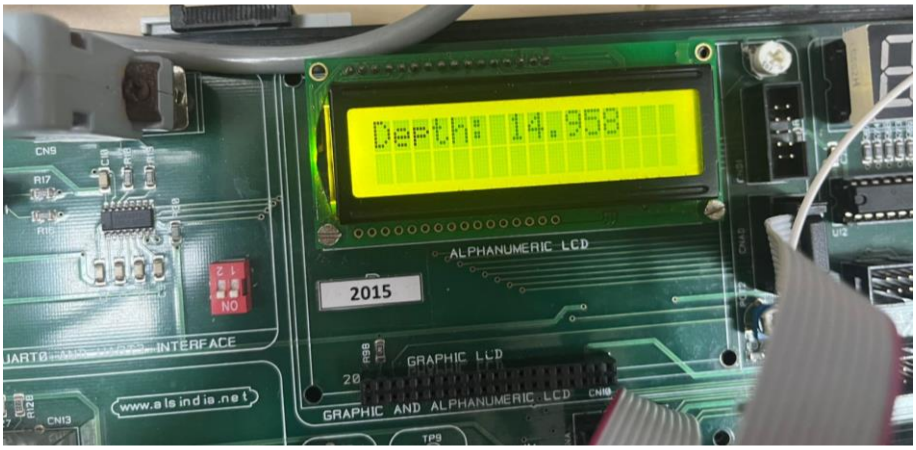

Fig2.14: LCD Displaying Depth of Water

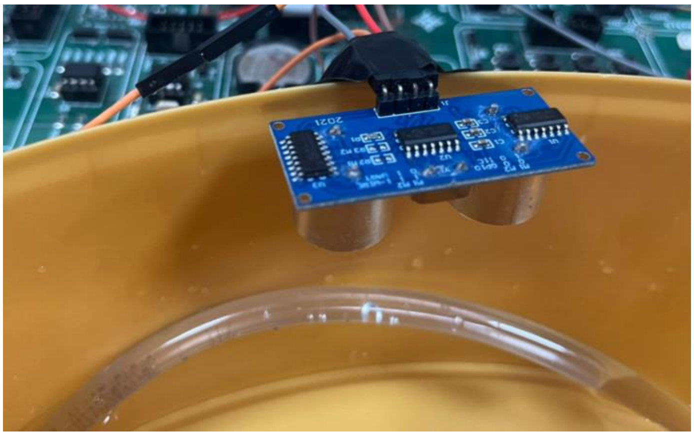

Fig2.15: Ultrasonic Sensor used for water level detection

2. Detection of cup in front of container

With the help of GPIO interrupts, water is dispensed as and when the ultrasonic sensor placed in front of the container detects a cup.

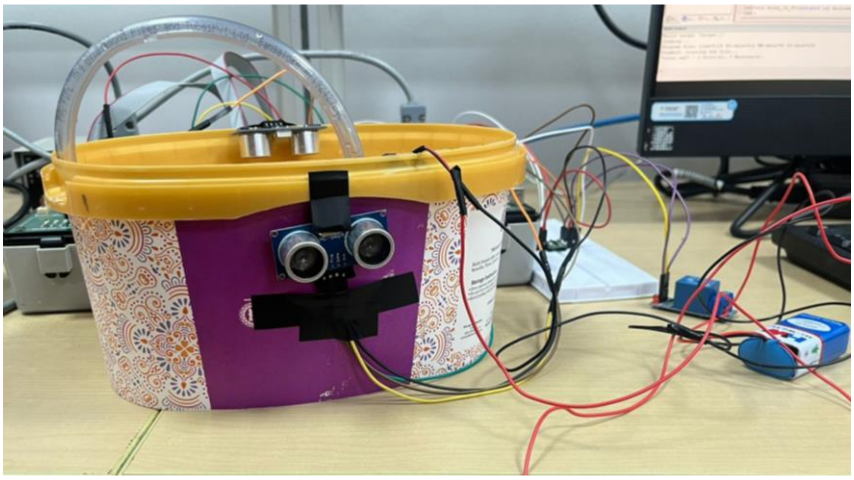

Fig2.16: Ultrasonic sensor used for detecting the proximity of an object

3. Immediate response of the relay module

When a cup is brought in front of the ultrasonic sensor, the microcontroller immediately sends a signal to the relay module which then switches on the pump and dispenses water. As soon as the cup is taken away from the sensor a signal is sent to the relay to stop the flow of water by switching off the pump.

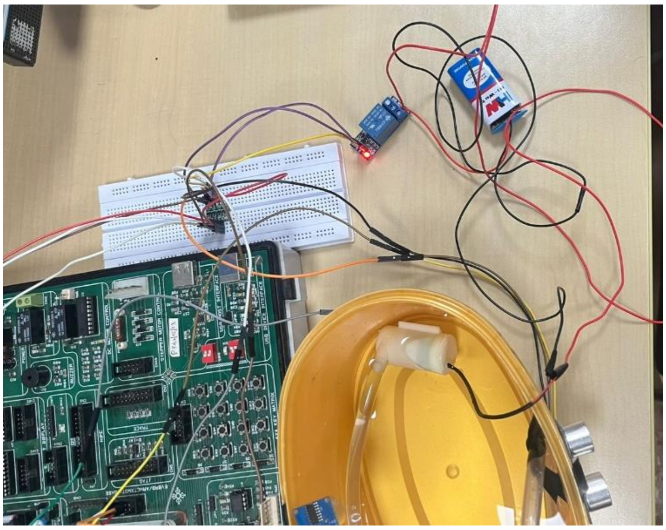

Fig2.17: Relay Module being switched on

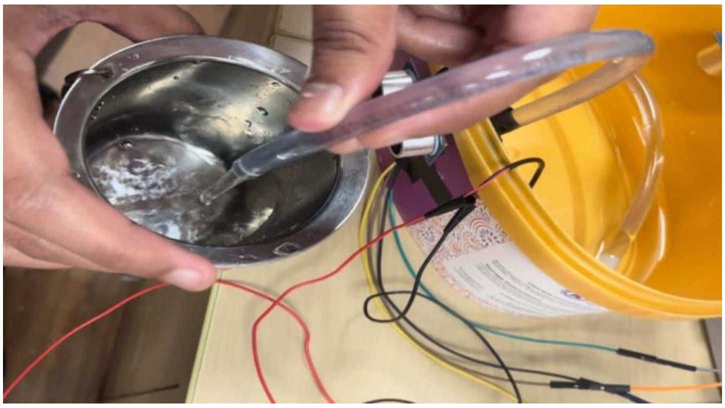

Fig2.18: Pump dispensing water

4. Potential Improvements and limitations

The current proximity sensor water dispenser design enables contactless operation and water level monitoring successfully, although there are certain limitations and areas for improvement. The use of two HC-SR04 sensors can suffer from timing interference which would give inconsistent readings. With time, the relay module introduces noise and reliability issues. Because the sensors and LCD are in constant use, high power consumption becomes an issue.

Improvements would include timing the sensor so that readings are consistent with each other, adding an infrared sensor so that proximity detection is enhanced without interference from the outputs of the sensor, and implementing power saving modes that would make the system last longer with a given battery.

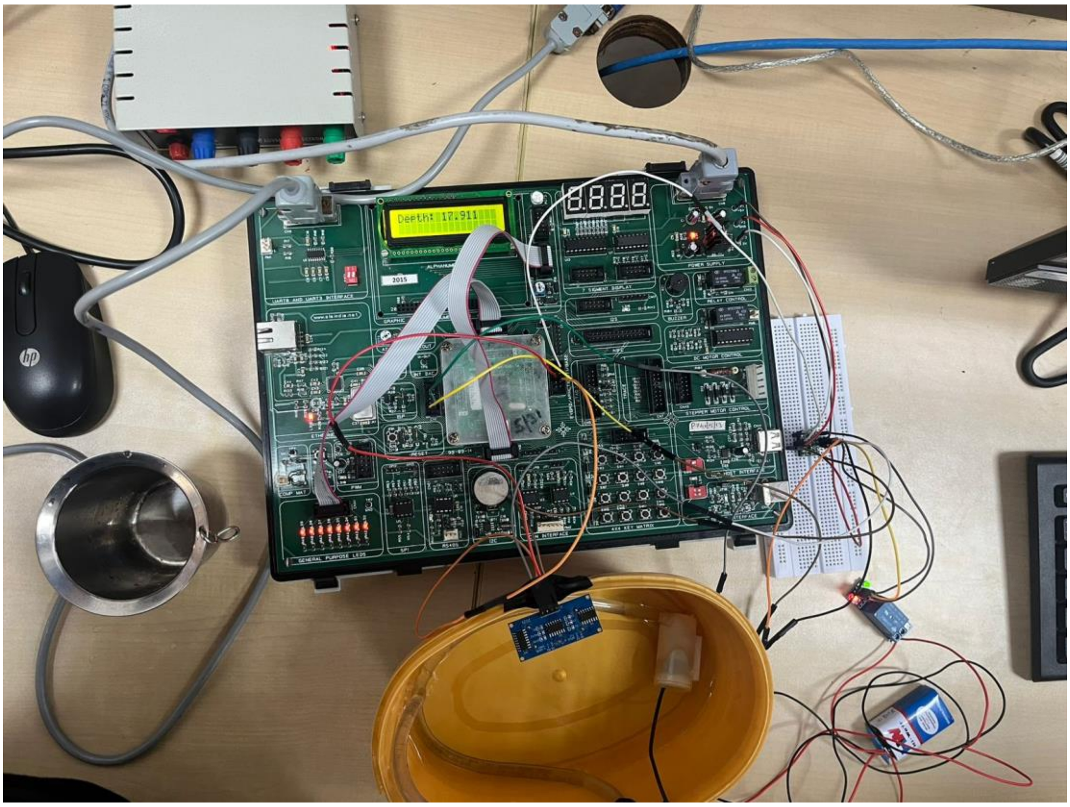

Fig2.19: Top View of the Project

D. References

[1] NXP Semiconductors, "LPC1769/68/67/66/65/64/63 Product Data Sheet,". [Online]. Available: https://www.nxp.com/docs/en/data-sheet/LPC1769_68_67_66_65_64_63.pdf

[2] Keil, "LPC17xx User Manual,". [Online]. Available: https://www.keil.com/dd/docs/datashts/philips/lpc17xx_um.pdf

[3] HC-SR-04 Ultrasonic Sensor, "Ultrasonic Ranging Module HC - SR04 Data Sheet,". [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[4] Relay Module, "Understanding How a Single Channel Relay Module Works and How to Use it with Arduino to Control AC Loads". [Online]. Available: https://circuitdigest.com/microcontroller-projects/interface-single-channel-relay-module-with-arduino

[5] Bidirectional Logic Level Converter, "Bidirectional Logic Level Converter Data Sheet,". [Online]. Available: https://www.microbot.it/documents/mr002-005_datasheet.pdf

E. C code:

```
#include <stdio.h>

#include <LPC17xx.h>

#include <string.h>

// Defining Constants
```

```
#define LED_Pinsel 0xff // P0.4-0.11 (LEDs)
#define TRIGGER_PIN_W (1 << 15) // P0.15 (Trigger Pin) WATER LEVEL
#define ECHO_PIN_W (1 << 16) // P0.16 (Echo Pin) WATER LEVEL
#define TRIGGER_PIN_P (1<<11) //P2.11 (Trigger pin) PROXIMITY
#define ECHO_PIN_P (1<<10) //P2.10 (Echo pin) PROXIMITY

void EINT3_IRQHandler(void);

// Variable Declarations
int temp, temp1, temp2 = 0; int flag = 0, flag_command=0;
int flag1,flag2;
int empty=0;
int trig_flag=1; //used to see if a trigger pulse is to be sent to the ultrasonic sensor used to detect proximity
int i, j, k, l, r, echoTime = 5000,echoT=5000; float distance = 0,dist=0;
int lcd_init[]={0x30,0x30,0x30,0x20,0x28,0x0C,0x06,0x01,0x80}; //Command codes to initialize LCD
char s[20]="";
//Function Declarations
void delay(int r1);
void timer_start(void);
float timer_stop(void);
void timer_init(void);

void timer_prox_start(void);
float timer_prox_stop(void);
void timer_prox_init(void);

void delay_in_US(unsigned int microseconds);
void delay_in_MS(unsigned int milliseconds);

//Functions
void port_write()
```

```
{
	int a;
	LPC_GPIO0->FIOMASK=0xE07FFFFF;
	LPC_GPIO0->FIOPIN=temp2<<23;

	if(flag1==0)
	{
		LPC_GPIO0->FIOCLR=1<<27;
	}
	else
	{
		LPC_GPIO0->FIOSET=1<<27;
	}
	//enable LCD
	LPC_GPIO0->FIOSET=1<<28;
	for(a=0;a<20;a++);//delay
	LPC_GPIO0->FIOCLR=1<<28;
	LPC_GPIO0->FIOMASK=0;
	for(a=0;a<300000;a++);//delay for lcd to respond

}
void lcd_write()
{

	flag2 = (flag1 == 1) ? 0 :((temp1 == 0x30) || (temp1 == 0x20)) ? 1 : 0;



	temp2=temp1&0xF0;
	temp2=temp2>>4;
	port_write();
	if(flag2==0)
	{
```

```
                    temp2=temp1&0xF;
                    port_write();
          }
}
int main()
{
  SystemInit();
  SystemCoreClockUpdate();
          timer_init();
          timer_prox_init();
  LPC_PINCON->PINSEL0 &= 0xfffff00f; // Interface LEDs P0.4-P0.11
  LPC_PINCON->PINSEL0 &= 0x3fffffff; // Interface TRIG P0.15
          LPC_PINCON->PINSEL1 =0;
          LPC_PINCON->PINSEL3 =0;//for p1.24

          //for the GPIO interrupt using P2.10
          LPC_PINCON->PINSEL4|=0; //P2.10 and P2.11 in function 0
          LPC_GPIO2->FIODIR|=1<<11; //P2.11 in output mode (trigger)

          LPC_GPIOINT->IO2IntEnR=1<<10; //Rising edge P2.10

          NVIC_EnableIRQ(EINT3_IRQn); //enable the interrupt

          LPC_GPIO1->FIODIR|=1<<24;//p1.24

  LPC_GPIO0->FIODIR |= TRIGGER_PIN_W | 1 << 17; // Direction for TRIGGER pin
          LPC_GPIO0->FIODIR|=0<<16|3<<27 | 0xF<<23;
  LPC_GPIO0->FIODIR |= LED_Pinsel << 4; // Direction for LED
  LPC_PINCON->PINSEL1 |= 0;

  i = 0;
  flag = 1;
```

```
LPC_GPIO0->FIOCLR |= TRIGGER_PIN_W;
while (1)
{
        if(trig_flag==1)
        {
            LPC_GPIO2->FIOSET=1<<11; //Send high on P2.11 trigger pin
            delay_in_US(10);
            LPC_GPIO2->FIOCLR=1<<11; //Pull down the P2.11 pin to a low
            trig_flag=0;
        }

 LPC_GPIO0->FIOSET = 0x00000800;
 // Output 10us HIGH on the TRIGGER pin
     LPC_GPIO0->FIOMASK = 0xFFFF7FFF;
     LPC_GPIO0->FIOPIN |= TRIGGER_PIN_W;
 delay_in_US(10);
 LPC_GPIO0->FIOCLR |= TRIGGER_PIN_W; LPC_GPIO0->FIOMASK = 0x0;

        while (!(LPC_GPIO0->FIOPIN & ECHO_PIN_W));// Wait till ECHO PIN
becomes high
 timer_start();
 while (LPC_GPIO0->FIOPIN & ECHO_PIN_W); // Wait till ECHO PIN becomes low
 echoTime = timer_stop(); // Store the time taken on stopping the timer
 distance = (0.00343 * echoTime) / 2; //Calculations of Distance in cm
        if(distance<=7.5)
        {
            empty=1;
                        flag1=0;
        for(i=0;i<9;i++)
        {
            temp1=lcd_init[i];
            lcd_write();
```

```
}
flag1=1;
sprintf(s,"Please refill");

for(i=0;s[i]!='\0';i++)
{
	if(i==16)
	{
		flag1=0;
		temp1=0xC0;
		lcd_write();
		flag1=1;
	}
	temp1=s[i];
	lcd_write();
}
continue;
}
else
	empty=0;
flag1=0;
for(i=0;i<9;i++)
{
	temp1=lcd_init[i];
	lcd_write();
}
flag1=1;
sprintf(s,"Depth: %.3f",distance);
for(i=0;s[i]!='\0';i++)
{
	if(i==16)
	{
```

```
					flag1=0;
					temp1=0xC0;
					lcd_write();
					flag1=1;
				}
				temp1=s[i];
				lcd_write();
			}

   delay(88000);
}
}

void delay_in_US(unsigned int microseconds)
{
	LPC_TIM0->TCR = 0x02;
  LPC_TIM0->PR = 0; // Set prescaler to the value of 0
	LPC_TIM0->MR0 = microseconds - 1; // Set match register for 10us
	LPC_TIM0->MCR = 0x01; // Interrupt on match
  LPC_TIM0->TCR = 0x01; // Enable timer
  while ((LPC_TIM0->IR & 0x01) == 0); // Wait for interrupt flag
	LPC_TIM0->TCR = 0x00; // Stop the timer
  LPC_TIM0->IR = 0x01; // Clear the interrupt flag
}
void delay_in_MS(unsigned int milliseconds) // Using Timer0
{
  delay_in_US(milliseconds * 1000);
}
void timer_init(void)
{
// Timer for distance
	LPC_TIM0->CTCR = 0x0;
```

```
  LPC_TIM0->PR = 11999999; //To maintain 12Mhz as per specified for LPC 1768
}
void timer_start(void)
{
  LPC_TIM0->TCR = 0x02; // Reset Timer
        LPC_TIM0->TCR = 0x01; // Enable timer
}
float timer_stop()
{
  LPC_TIM0->TCR = 0x0;
  return LPC_TIM0->TC;
}
void delay(int r1)
{
  for (r = 0; r < r1; r++);
}

void timer_prox_init()
{
        LPC_TIM1->CTCR = 0x0;
  LPC_TIM1->PR = 11999999; //To maintain 12Mhz as per specified for LPC 1768

}
void timer_prox_start(void)
{
  LPC_TIM1->TCR = 0x02; // Reset Timer
        LPC_TIM1->TCR = 0x01; // Enable timer
}
float timer_prox_stop()
{
  LPC_TIM1->TCR = 0x0;
  return LPC_TIM1->TC;
```

```
}

void EINT3_IRQHandler()
{
                        LPC_GPIOINT->IO2IntClr=1<<10; //Clears the interrupt
    timer_start();
    while (LPC_GPIO2->FIOPIN & ECHO_PIN_P); // Wait till ECHO PIN becomes low
    echoT = timer_stop(); // Store the time taken on stopping the timer
    dist = (0.00343 * echoT) / 2; //Calculations of Distance in cm



    if (dist < 40)
      {
          LPC_GPIO0->FIOSET = LED_Pinsel << 4;

                                                            LPC_GPIO1->FIOCLR = 1 << 24;//turn on relay
      }
    else
      {
          LPC_GPIO0->FIOCLR = LED_Pinsel << 4;

                                                  LPC_GPIO1->FIOSET = 1 << 24; //turn off relay
      }
                        trig_flag=1;
}
```