WATER PUMP CONTROL AND WATER LEVEL MONITORING

APPLIED ELECTRONICS AND INSTRUMENTATION - 2nd Year

Arpit Das	44
Shankha Bhattacharya	21
Shubham Bhardwaj	18
Ananya Thakur	50
Suhan Roychowdhuri	04
Bala Sourvendra	41

OBJECTIVE

Our main objective for this project is to check the water level in the tank and switch on the water pump if water level is low and switch it off when water level reaches a high point. Here we are using an **Ultrasonic Sensor** using which we are calculating the distance of the water level from the sensor. We are also using a relay that acts as a switch between the **Arduino** and the high voltage **Water Pump**. The Ultrasonic sensor sends the data to the Arduino in which the program calculates the distance and tells the Arduino to turn the relay on or off accordingly.

List of Components

- Arduino UNO (x1)
- Ultrasonic Sensor - HC SR04 (x1)
- Relay (x1)
- Bread Board (x1)
- Connecting Wires (x1Lot)
- LED (x1)

Key Components

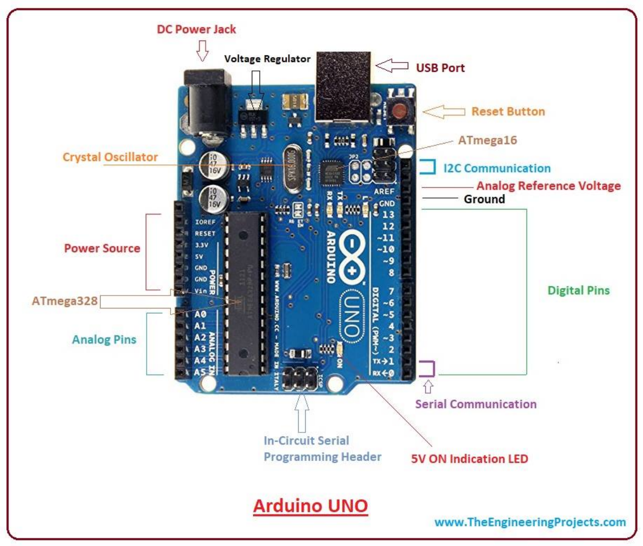
- Arduino UNO

  The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits. The Arduino Uno pinout consists of 14 digital pins, 6 analog inputs,a power jack, USB connection and ICSP header.
  The versatility of the pinout provides many different options such as driving motors, LEDs, reading sensors and more.

  The pin configuration is such as :

  A0 - A5 : Analog Input Pins
  Rx - 13 : Digital Input/Output Pins (Tx/Rx - Serial communication)
  DC Power Jack : To power the Arduino board
  USB Port : To upload the code on the Arduino board

Detailed pin configuration

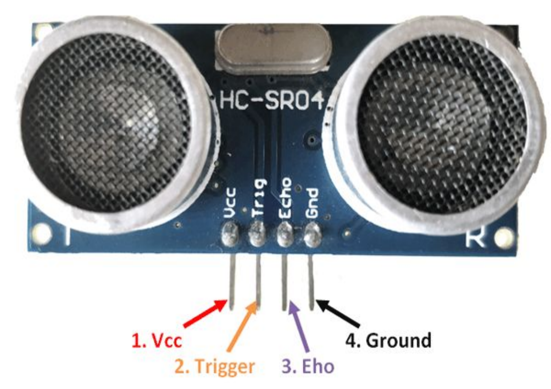


Key Components

- Ultrasonic Sensor

  Ultrasonic sensors measures distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

1. **VCC** - The Vcc pin powers the sensor, typically with +5V
2. **Trigger** - Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3. **Echo** - Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the final time taken for the US wave to return back to the sensor
4. **Ground** - This pin is connected to the Ground of the system.



Detailed pin configuration

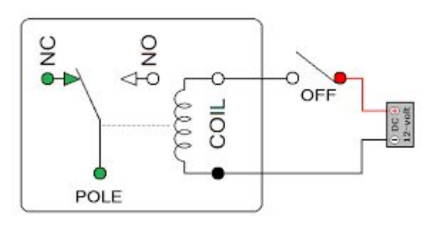
Key Components

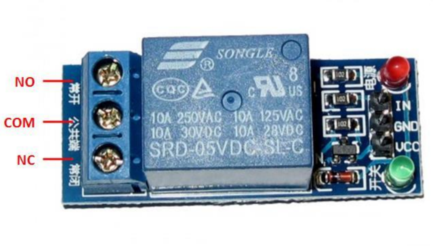
- Relay

A **relay** is an electrically operated switch. It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The switch may have any number of contacts in multiple contact forms, such as make contacts, break contacts, or combinations thereof.

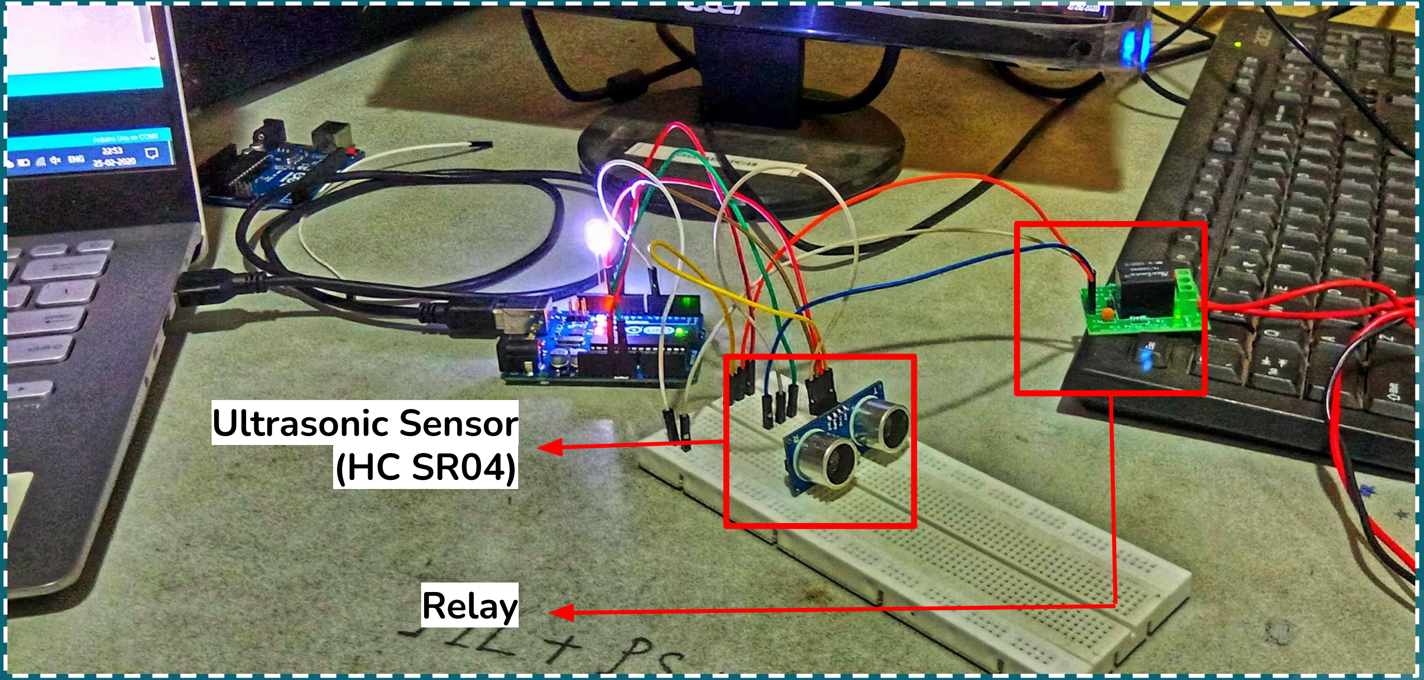
Pin Configuration of Relay :

1. **IN -** The input given to the relay through the Arduino
2. **GND -** For ground connection
3. **VCC -** To supply voltage to the relay
4. **NC -** The other end of the load is either connected to NO or NC. If connected to NC the load remains connected before trigger.
5. **NO -** The other end of the load is either connected to NO or NC. If connected to NO the load remains disconnected before trigger.
6. **COM -** Common is connected to one End of the Load that is to be controlled





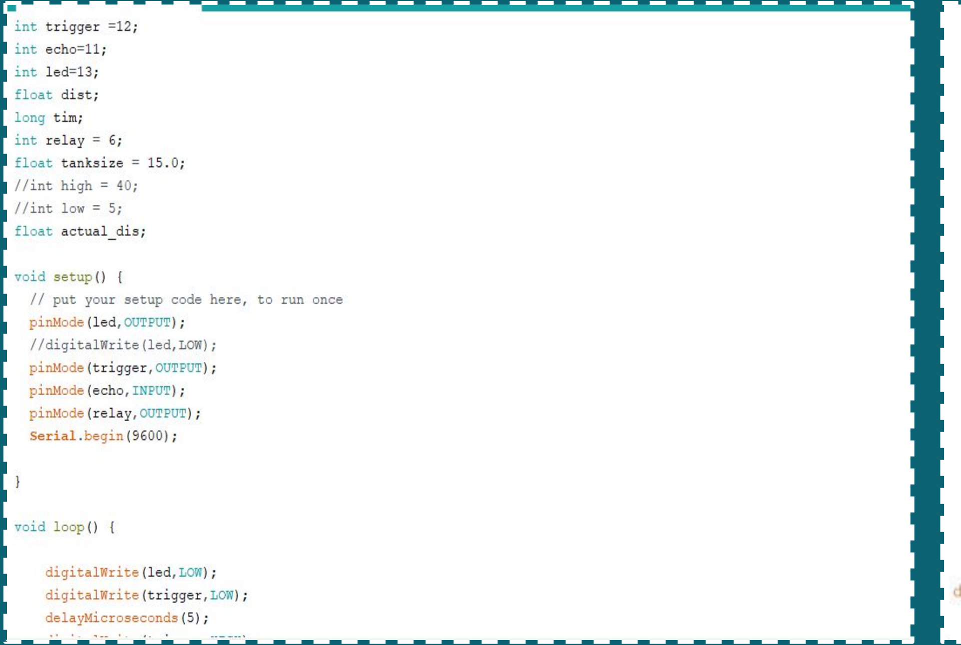
Circuit Diagram



Ultrasonic Sensor (HC SR04)

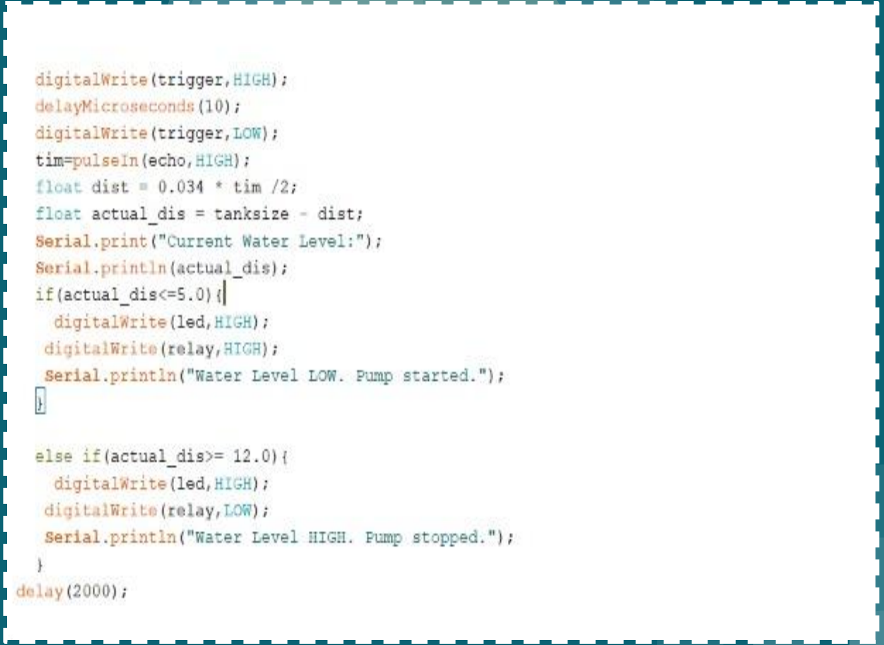
Relay

Program Code



```
int trigger =12;
int echo=11;
int led=13;
float dist;
long tim;
int relay = 6;
float tanksize = 15.0;
//int high = 40;
//int low = 5;
float actual_dis;

void setup() {
  // put your setup code here, to run once
  pinMode(led,OUTPUT);
  //digitalWrite(led,LOW);
  pinMode(trigger,OUTPUT);
  pinMode(echo,INPUT);
  pinMode(relay,OUTPUT);
  Serial.begin(9600);

}

void loop() {

    digitalWrite(led,LOW);
    digitalWrite(trigger,LOW);
    delayMicroseconds(5);
```

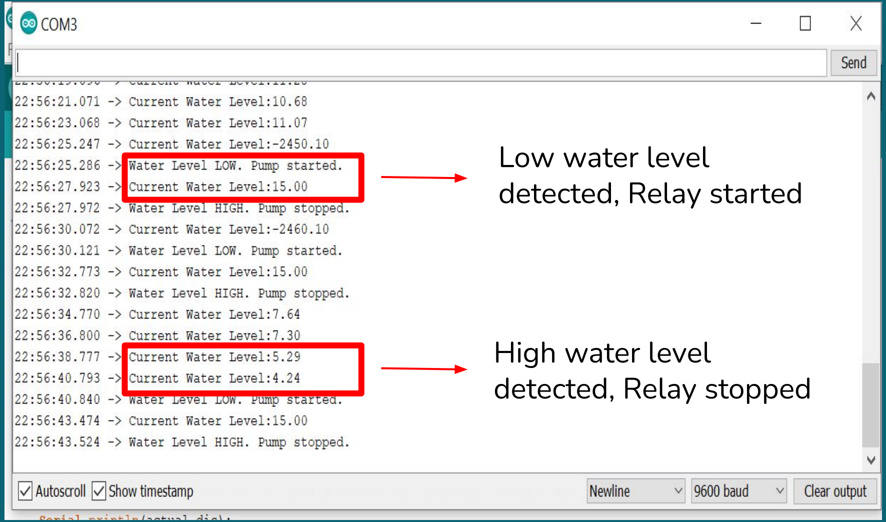
1



```
digitalWrite(trigger,HIGH);
delayMicroseconds(10);
digitalWrite(trigger,LOW);
tim=pulseIn(echo,HIGH);
float dist = 0.034 * tim /2;
float actual_dis = tanksize - dist;
Serial.print("Current Water Level:");
Serial.println(actual_dis);
if(actual_dis<=5.0){
  digitalWrite(led,HIGH);
 digitalWrite(relay,HIGH);
 Serial.println("Water Level LOW. Pump started.");
}

else if(actual_dis>= 12.0){
  digitalWrite(led,HIGH);
 digitalWrite(relay,LOW);
 Serial.println("Water Level HIGH. Pump stopped.");
}
delay(2000);
```

2

Result



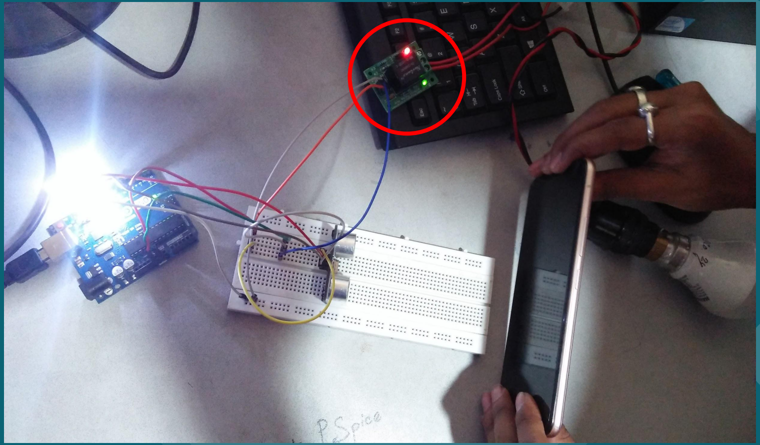
```
22:56:19.090 -> Current Water Level:11.26
22:56:21.071 -> Current Water Level:10.68
22:56:23.068 -> Current Water Level:11.07
22:56:25.247 -> Current Water Level:-2450.10
22:56:25.286 -> Water Level LOW. Pump started.
22:56:27.923 -> Current Water Level:15.00
22:56:27.972 -> Water Level HIGH. Pump stopped.
22:56:30.072 -> Current Water Level:-2460.10
22:56:30.121 -> Water Level LOW. Pump started.
22:56:32.773 -> Current Water Level:15.00
22:56:32.820 -> Water Level HIGH. Pump stopped.
22:56:34.770 -> Current Water Level:7.64
22:56:36.800 -> Current Water Level:7.30
22:56:38.777 -> Current Water Level:5.29
22:56:40.793 -> Current Water Level:4.24
22:56:40.840 -> Water Level LOW. Pump started.
22:56:43.474 -> Current Water Level:15.00
22:56:43.524 -> Water Level HIGH. Pump stopped.
```

☑ Autoscroll ☑ Show timestamp Newline 9600 baud

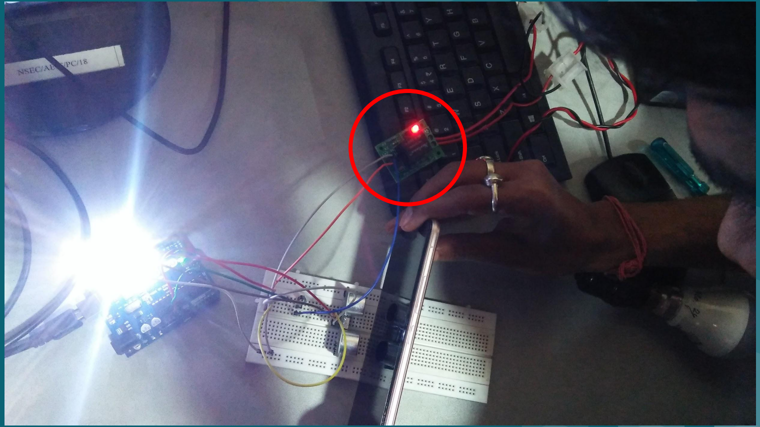
```
Serial.println(actual_dis);
```

→ Low water level detected, Relay started

→ High water level detected, Relay stopped



Relay started



Relay stopped