19L604 – EMBEDDED SYSTEMS & IOT

Project Report

Batch - 18

Project report submitted in partial fulfilment of the requirements for thedegree of

BACHELOR OF ENGINEERING

Branch: ELECTRONICS AND COMMUNICATION ENGINEERING

Of Anna University

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE– 641004.

Sheena S -20L141

Sankar Subramanian K V -20L157

Arun D -21L402

Jaya Murugan P -21L404

TABLE OF CONTENT:

INTRODUCTION:

One of the most vital natural resources for life is water. To fulfill their everyday basic requirements, humans depend on water. In addition to drinking, water is needed for a range of domestic tasks. The majority of us, though, are tired of the constant battle that is water management.

A practical way to keep track of and control the water level in the tanks is with an automatic water level controller. In the modern home, it is a need. Installing a smart water controller in your home has a number of significant benefits aside from the obvious one of conserving water and utilizing it wisely.

PROBLEM STATEMENT:

According to the research, the filling of water results in overflow andwater waste. In order to enable water level monitoring, the article discusses the development of Blynk IoT and PHP web-based programming. They also claimed that the system's regulation of the water level has a 2 cminaccuracy. They employed an ESP8266 microcontroller, which can connectto the internet, as their main control device. For monitoring water level, they employ an ultrasonic sensor.

HARDWARE MODULES & COMPONENTS REQUIRED

- Water Level Sensor or Submersible pressure transducer
- Ultrasonic Water Level Transmitters
- Optical liquid level sensor
- Float level sensors

SOFTWARE MODULES & TOOLS REQUIRED:

Software simulation:

HARDWARE SPECIFICATIONS:

1. **Water Level Sensor or Submersible pressure transducer:**

Level sensors are used to detect the level of substances that can flow. Such substances include liquids, slurries, granular material and powders. Level measurements can be done inside containers or it can be the level of a river or lake. Such measurements can be used to determine the number of materials within a closed container or the flow of water in open channels.

2. Ultrasonic Water Level Transmitters:

Ultrasonic Sensors use high frequency ultrasonic waves to detect the levels of any medium liquid or solid. This sensor/transmitter is mounted at the top of a tank and aimed downward. It transmits waves and measuresthe time it takes to receive the return signal back from the water to the sensor. The measured time is then used to output a level in terms of a precise voltage from 0-10 volts. The disadvantages include it costs a little more than other types of water level sensors, difficult to sense levels at long range and causes temporary inaccurate readings due to turbulence in water.

3. Optical liquid level sensor:

Optical liquid level sensors do not measure the liquid level, instead, they detect the presence or absence of liquid. Optical liquid level sensors consist of two main parts; An infra-red LED and phototransistor that is accurately positioned at the base of the sensor tip. When the tip is in air, infra-red light reflects internally around the tip of the phototransistor providing strong optical coupling between the two. When the sensor tip is immersed in liquid, the infrared light escapes from the tip causing a reduction in the amount of light at the phototransistor which makes the output change state.

4. Float level sensors:

Float level sensors are continuous level sensors featuring a magnetic float that rises and falls as liquid levels change. The movement of the float creates a magnetic field that actuates a hermetically sealed reed switch located in the stem of the level sensor, triggering the switch to open or close. Different variations of float switches are used for commercial and industrial applications involving water, oil, chemicals, and other liquid materials. The disadvantages include Float actuation relies on liquid contact, Thick or sticky fluids may prevent the float from moving.

5. Ultrasonic Sensor:

Ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e., the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has traveled to and from the target).

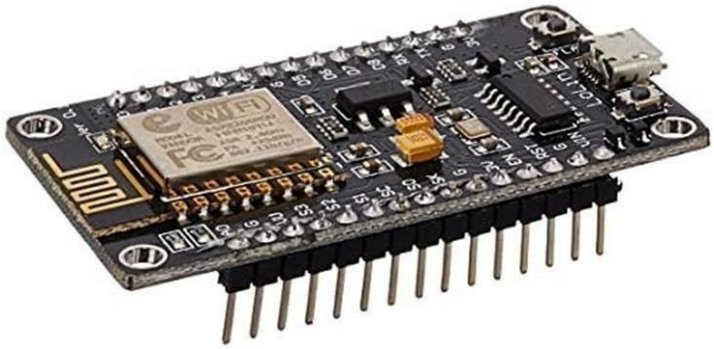
The HC-SR04 ultrasonic module is a module that can provide non-contact measurement within the range of 2cm to 400cm with ranging accuracy that can reach 3mm. It works on the principle of echolocation.

The ultrasonic sensor as a trigger and an echo pin. The Arduino provides a high signal of 10 microseconds to this pin. After the HC-SR04 is triggered, it sends out eight 40 kHz sound waves to the surface of the water. On getting to the surface of the water, the wave is echoed back to the sensor and the ESP8266 reads the echo pin to determine time spent between triggering and receiving of the echo. Since we know that the speed of sound is around 340m/s then we can calculate the distance using;

Distance = (time/2) * speed of sound

6. NodeMCU:

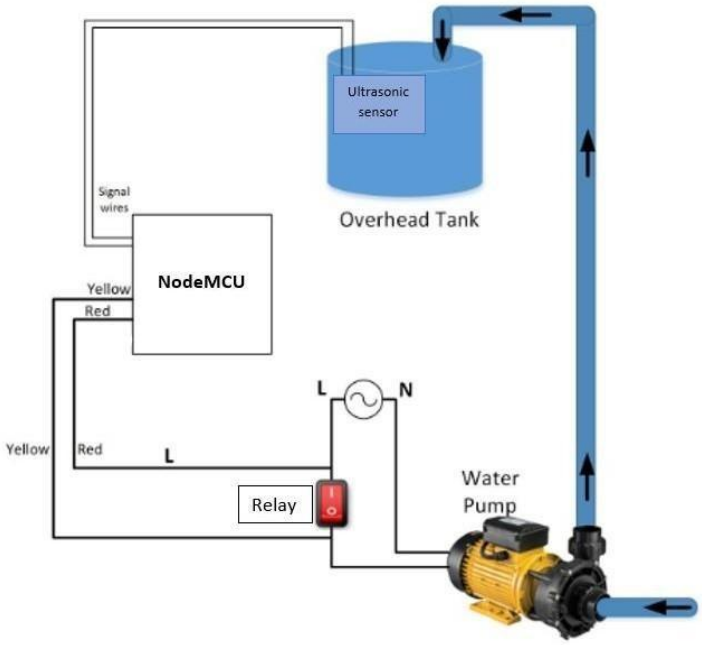
NodeMCU ESP8266-12E MCU is a development board with one analogue and many general-purpose input-output (GPIO) pins. It has 4MB flash memory, and can operate at a default clock frequency of 80MHz. The digital pin D4 of NodeMCU is used to control the Water Pump ON/OFF. And digital pin D1 &D2 of NodeMCU is used to read data of distance sensor HC-SR04.



NodeMCU is a low-cost open source IoT platform. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit).

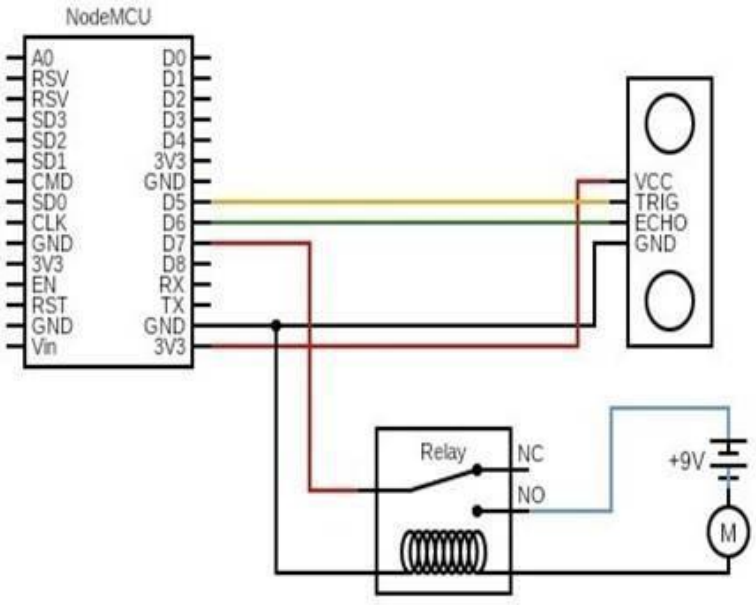
HARDWARE MODULE DESCRIPTION

Setup diagram:



An overhead tank is there at the top. An ultrasonic sensor is connected at the top of the overhead tank which is used to measure the water distance. The ultrasonic sensor is connected to the NodeMCU. A Relay switch is connected to the NodeMCU which acts as a switch that is responsible for turning on or off the motor according to the water level. Water pump is connected to the Relay.

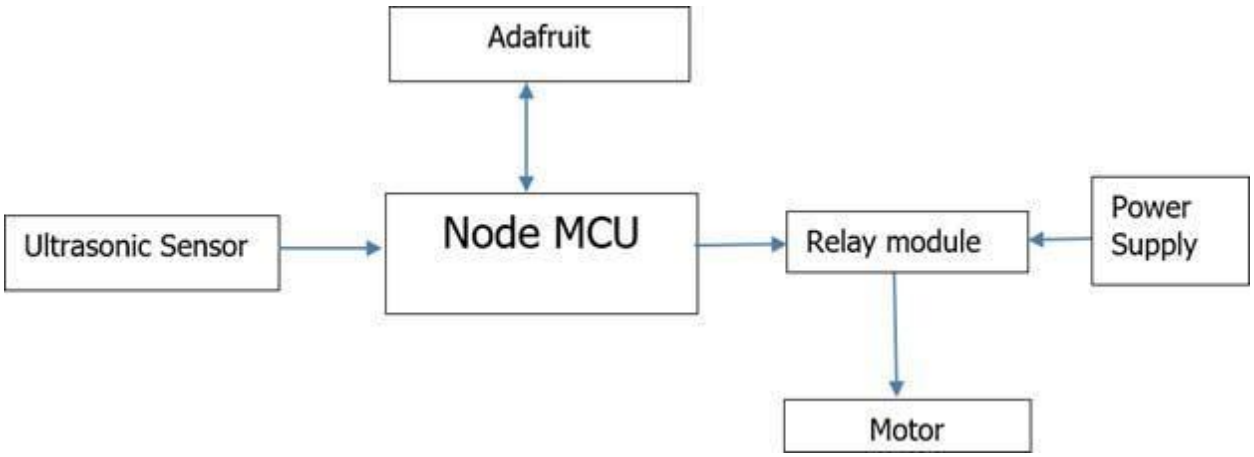
Circuit diagram:



Here, the VCC pin of the ultrasonic sensor is connected to the 3V3 pin of the NodeMCU. Trigger pin of the ultrasonic sensor is connected to the D5 pin NodeMCU. Echo pin of the ultrasonic sensor to the D6 pin of NodeMCU. Ground pin of the ultrasonic sensor is connected to the Ground pin of NodeMCU. Battery is connected to one terminal of the motor and the other end is connected to a relay. The positive terminal of the battery is connectedto NO pin of relay.

SOFTWARE MODULE DESCRIPTION:

Block diagram:



- NodeMCU is connected with an ultrasonic sensor and Relay module.
- Relay module is connected to the motor.
- It acts as a switch which is responsible for turning the motor on or off.
- NodeMCU is connected with Adafruit through MQTT protocol.

SIMULATION TOOL USED:

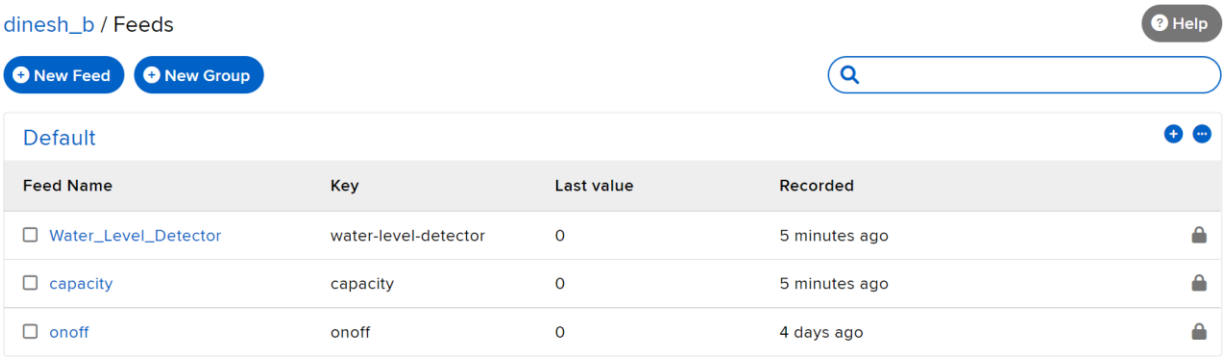
Software simulation:

ARDUINO

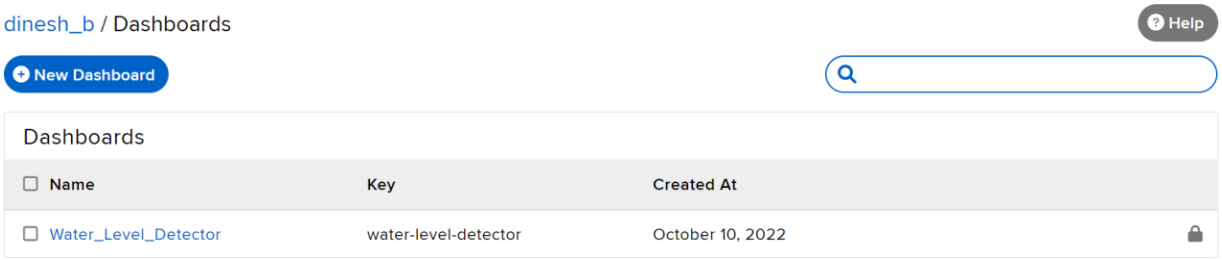
adafruit

SIMULATION RESULT:

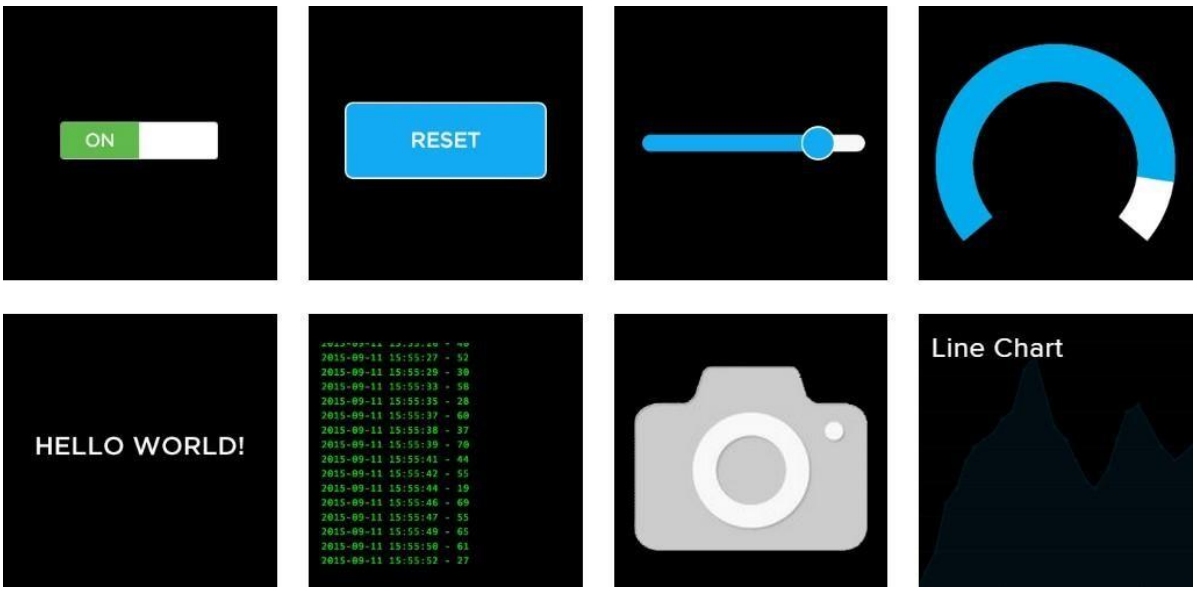
❖ Feeds



❖ Dashboards



❖ Blocks



/

CODE WITH COMMENTS:

```
// defines pins numbers
#include <ESP8266WiFi.h> //Nodemcu wifi
// Libraries for mqtt
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

// Wifi name and password
#define _SSID "Username"
#define _PASS "password"

// Credentials for adafruit - broker
#define AIO_SERVER "io.adafruit.com"
#define AIO_SERVERPORT 1883
#define AIO_USERNAME "dinesh_b"
#define AIO_KEY "aio_odgS63LsTmvJW3Kfq4TOHmiuk9EP"

void MQTT_connect();
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);
```

```
Adafruit_MQTT_Publish usSensor = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/water_level_detector");

Adafruit_MQTT_Publish capacity = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/capacity");

Adafruit_MQTT_Subscribe onoff = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feed/onoff");

#define trigPin 12 //D6

#define echoPin 14 //D5

#define motorPin 13 //D7


// defines variables

long duration = 0;

int distance;

int output,output_value;


void MQTT_connect() {

  int8_t ret;


  // Stop if already connected.

  if (mqtt.connected()) {

    return;

  }
```

```
  Serial.print("Connecting to MQTT... ");
  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for
connected
       Serial.println(mqtt.connectErrorString(ret));
       Serial.println("Retrying MQTT connection in 5 seconds...");
       mqtt.disconnect();
       delay(5000); // wait 5 seconds
       retries--;
       if (retries == 0) {
         // basically die and wait for WDT to reset me
         while (1);
       }
  }
  Serial.println("MQTT Connected!");
}
```

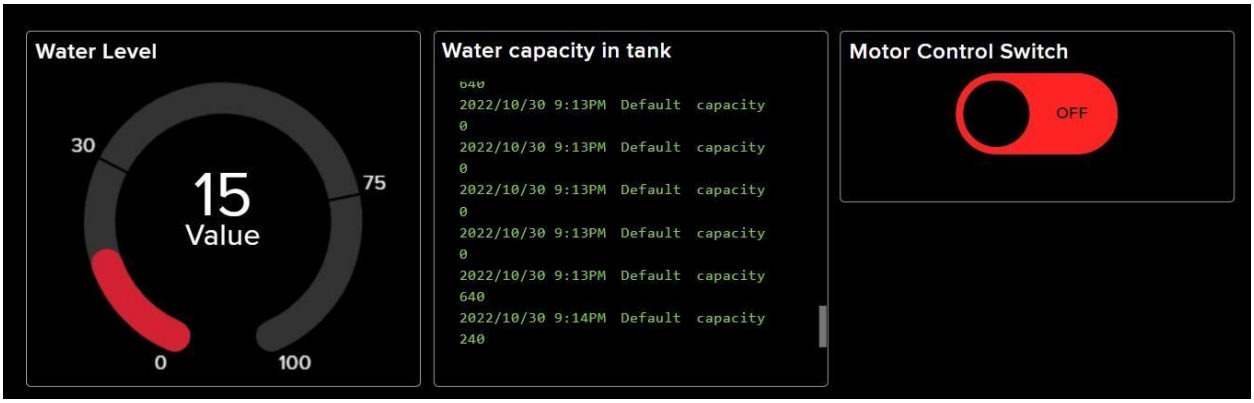
```
void setup()
{
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(motorPin, OUTPUT); // Sets the motorPin as an Output
  Serial.begin(115200); // Starts the serial communication
  Serial.print("Connecting...");
  WiFi.begin(_SSID,_PASS);
  while(WiFi.status()!=WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  mqtt.subscribe(&onoff);
}
```

```
void loop() {
  MQTT_connect();
  Adafruit_MQTT_Subscribe *subscription;
  while((subscription = mqtt.readSubscription(1000))) {
    if(subscription == &onoff) {
      Serial.print(F("Motor: "));
      Serial.println((char*)onoff.lastread);
    }
  }
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH,30000);
```

```
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
if (distance<=200)
{
  Serial.print("Distance: ");
  Serial.print(distance);
  distance = 100 - (distance/20.0 * 100);
  if(distance<0)
  {
    distance = 0;
  }
  Serial.print("cm");
  if(usSensor.publish(distance))
  {
    capacity.publish(distance*20*10*8/100);
    Serial.print("- published");
  }
  else
  {
    Serial.print("- failed");
```

```
    }
    Serial.print("\n");
    delay(5000);
  }
  else
  {
    Serial.println("Out of Range");
  }
  if(distance <= 40)
    {
      digitalWrite(motorPin, HIGH);
    }
  if(distance >= 70)
    {
      digitalWrite(motorPin, LOW);
    }
  if(!mqtt.ping()) {
    mqtt.disconnect();
  }
}
```

RESULT ANALYSIS:



The gauge on the left shows the water level, while the Water capacity in tank window shows the estimated water capacity of the tank.

CONCLUSIONS AND FUTURE SCOPE:

From the results, the capacity of water in the tank is expressed in milliliters as **1 cm^3 = 1 ml**.

Here, the dimensions of the tank considered are 20cm x 10cm x 8cm. Hence, the volume is calculated as 1600 cm^3, which is 1600 ml.

For 75% capacity, it is estimated that the tank contains 1200 milliliters of water (fuel in case of fuel tank). This analogy can be used for Fuel tank level gauging.

Benefits:

The water level in the tanks may be monitored and controlled with the help of an automatic water level controller. In the modern home, it is a need. Installing a smart water controller in a home has a number of significant benefits in addition to the obvious one of conserving water and utilizing it wisely.

Advantages of using a water level controller:

Saves Money: Water level controllers help conserve energy and eventually helps to save money. With the use of these technologies, water is managed, which means that wasted water and electricity are kept to a minimum.

Conserves Water: An automated water level controller can help you get the most out of your water. During the day, water pumps are typically used more frequently. These systems are advantageous because they constantly deliver more water during the day and less water at night, ensuring that the water level is maintained at all times.

Low-Maintenance: Water level controllers are low-maintenance devices that last a long time and require little maintenance. They have stainless conductive electrodes built in that are coated in plastic. They are known for lasting a long time and requiring little care and cleaning.

REFERENCES:

https://www.slideshare.net/gaurav9616rai/water-level-monitoring-system
https://www.biz4intellia.com/water-level-monitoring/

https://youtu.be/9geREeE13jc

https://electronics-project-hub.com/iot-based-water-level-monitoring-system/

https://www.biz4intellia.com/water-level-monitoring/#:~:text=What%20is%20an%20IoT%2Dbased,status%20of%20the%20liquid%20levels

https://www.electronicshub.org/water-level-indicator/

https://circuitdigest.com/microcontroller-projects/water-level-indicator-project-using-arduino

https://www.instructables.com/1-AUTOMATIC-WATER-LEVEL-CONTROLLER/