Water Level Monitoring and Pump Control

Name: JEROME PUNNOOSE - 34
RUBEN JOEMON - 49

Semester & Branch: S3 – ER

Department: ELECTRONICS

College: SAINTGITS COLLEGE OF ENGINEERING

1 Introduction

Water management is a key challenge in both domestic and agricultural contexts. Manual pump control often leads to water wastage when tanks overflow or causes pump dry-run when water levels are low. These inefficiencies not only waste valuable resources but also increase electricity consumption and reduce the lifespan of the pump.

This microproject demonstrates an **IoT-based water level monitoring and smart pump control system** developed using the ESP8266 NodeMCU microcontroller, HC-SR04 ultrasonic sensor, and the Blynk IoT platform. The system continuously monitors the water level in the tank and automatically controls the pump to maintain optimal levels. Users can also track water levels and control the pump remotely through a mobile application, providing both convenience and efficiency in water management.

Objectives:

- To measure the water level in the tank accurately using an ultrasonic sensor interfaced with NodeMCU.
- To automate the pump's ON/OFF operation based on predefined threshold water levels, preventing overflow and dry-run.
- To enable real-time monitoring and remote control of the water tank and pump through the Blynk IoT mobile application.
- To support both **manual** and **automatic** operating modes, allowing users to either rely on automatic control or manually operate the pump as required.

2 Block / Functional Diagram

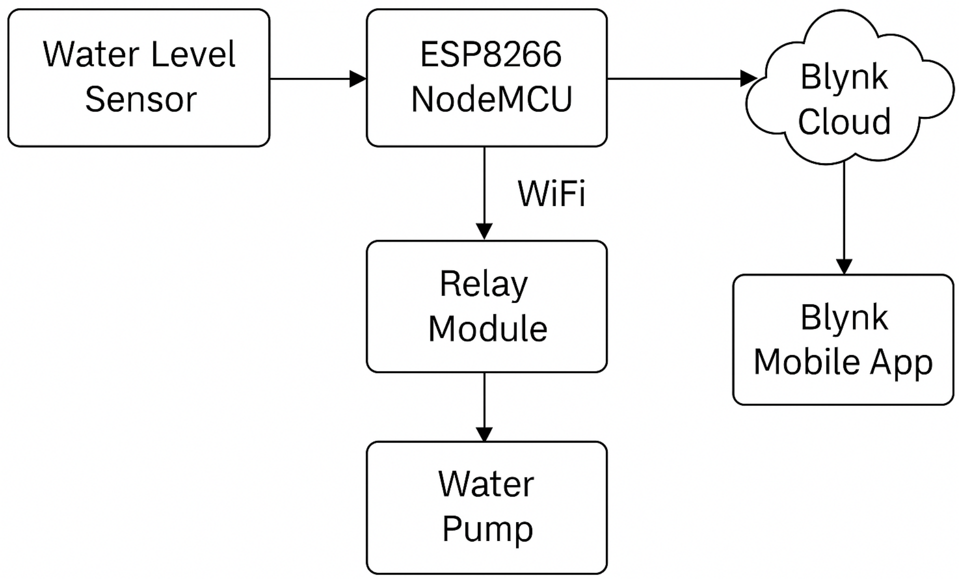

Figure 1: Functional Block Diagram

Explanation: The block diagram illustrates the working of the IoT-based Water Level Monitoring and Pump Control System.

The Water Level Sensor (HC-SR04) measures the water level in the tank and sends distance data to the ESP8266 NodeMCU microcontroller. Based on the measured level, the NodeMCU processes the data and controls the Relay Module, which in turn switches the Water Pump ON or OFF automatically.

The NodeMCU also connects to the Blynk Cloud via Wi-Fi, enabling real-time monitoring and control. The Blynk Mobile App allows users to view the current tank level, receive alerts, and manually operate the pump remotely through the internet

3 Circuit Diagram and Working Principle

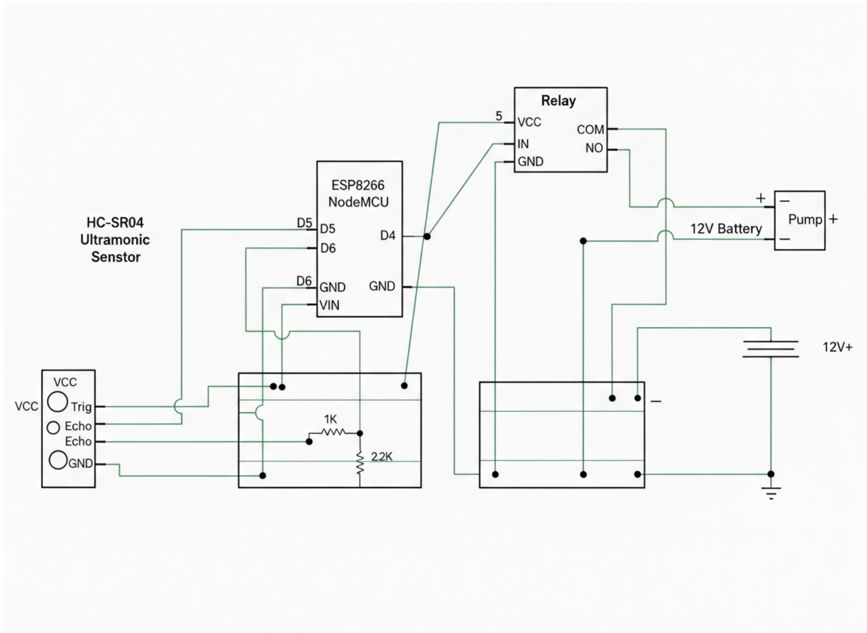

Figure 2: Circuit Diagram of Water Level Monitoring System

Working Principle: The HC-SR04 sensor measures the distance between the water surface and the sensor. The NodeMCU computes the water level percentage as:

$$Level(\%) = \frac{(D_{max} - D_{measured})}{(D_{max} - D_{min})} \times 100$$

where $D_{max} = 9.5$ cm (tank empty) and $D_{min} = 4.3$ cm (tank full). In AUTO mode, the pump turns ON when the level drops below 20% and turns OFF above 80%. In MANUAL mode, the user can control the pump directly via the Blynk switch.

Components:

- ESP8266 NodeMCU
- HC-SR04 Ultrasonic Sensor
- Relay Module (JQC3F 05VDC-C)
- Water Pump (12V, with 9V battery for power)

4 Detailed Circuit Connections

4.1 ESP8266 NodeMCU Pinout

Function	NodeMCU Pin	Connected To
HC-SR04 Trig	D5 (GPIO14)	Trig on HC-SR04
HC-SR04 Echo	D6 (GPIO12)	Echo on HC-SR04
Relay IN	D4 (GPIO2)	Relay module IN
HC-SR04 VCC	5V or VIN	HC-SR04 VCC
HC-SR04 GND	GND	HC-SR04 GND
Relay VCC	VIN (or 5V)*	Relay module VCC
Relay GND	GND	Relay module GND
Breadboard power rail	VIN	Powered by 9V/12V supply

Table 1: ESP8266 NodeMCU main connections

4.2 HC-SR04 Ultrasonic Sensor

Pin	Connected To
VCC	VIN (through breadboard)
GND	GND
Trig	NodeMCU D5
Echo	NodeMCU D6

Table 2: HC-SR04 Ultrasonic Sensor Connections

4.3 Relay Module

Relay Pin	Connected To
IN	NodeMCU D4 (GPIO2)
VCC	VIN on NodeMCU/breadboard
GND	GND on NodeMCU
Common (COM)	Pump + terminal
NO (Normally Open)	+9V/+12V battery supply
NC (Normally Closed)	Not connected

Table 3: Relay Module Connections

Relay Contacts (Switching the Pump)

- COM → + terminal of water pump.

- NO $\rightarrow$ + terminal of battery (9V or 12V for pump).
- Water pump $-$ terminal $\rightarrow$ $-$ terminal of battery.

4.4 Water Pump

Pump Terminal	Connected To
+ (Positive)	Relay COM
– (Negative)	Battery – (GND)

Table 4: Water Pump Connections

4.5 Power Supply

- NodeMCU: USB port OR external 5V.
- Pump: Separate 9V or 12V battery.
- **All grounds (battery, NodeMCU, relay) must be connected together.**

Description of Developed Code and Interfacing Logic

The developed code integrates hardware components and IoT functionality to achieve real-time water level monitoring and smart pump control. The system is built around the ESP8266 NodeMCU microcontroller, which connects to the internet through Wi-Fi and communicates with the Blynk Cloud for remote operation.

Code Description

The program begins by including the required libraries 'ESP8266WiFi.h' and 'BlynkSimpleEsp8266.' to enable Wi-Fi and cloud connectivity. The Blynk Template ID, Authentication Token, and Wi-Fi credentials are defined to link the NodeMCU with the user's Blynk account.

Pin configurations are assigned for the HC-SR04 ultrasonic sensor (`TRIG_PIN = D5`, `ECHO_PIN = D6`) and the relay module (`RELAY_PIN = D4`). The relay operates in an active-LOW configuration, meaning the pump turns ON when the pin is set LOW.

The function `readWaterLevel()` triggers the ultrasonic sensor and measures the echo time to calculate distance. This distance is converted to a percentage representing the tank's water level using calibrated full and empty values. The computed level is sent to the Blynk gauge widget (`V0`) for live visualization in the mobile application.

The program supports two modes of operation:

- **Automatic Mode (V3 = ON):** The pump automatically turns ON when the water level falls below 30%, and turns OFF when the level exceeds 80%, preventing overflow and dry-running.

- **Manual Mode (V3 = OFF):** The user can manually control the pump through the Blynk Mobile App using virtual pin `V1`, useful during maintenance or manual intervention.

A timer function executes `readWaterLevel()` every two seconds, ensuring continuous level updates. The main loop continuously runs `Blynk.run()` and `timer.run()` to maintain cloud communication and responsiveness.

Interfacing Logic

The interfacing between hardware components is modular and efficient:

- The **HC-SR04 ultrasonic sensor** sends ultrasonic pulses, and the NodeMCU measures the echo time to determine the distance to the water surface.
- The **Relay Module** receives control signals from the NodeMCU and switches the **Water Pump** ON or OFF accordingly.
- The **ESP8266 NodeMCU** connects to the **Blynk Cloud** via Wi-Fi, transmitting real-time water level data and receiving user commands.
- The **Blynk Mobile App** displays the tank level, allows mode selection (manual/auto), and provides remote pump control.

This integration ensures seamless interaction between sensing, control, and IoT monitoring. The combination of automation and cloud connectivity makes the system efficient, user-friendly, and reliable for both domestic and agricultural water management.

5 Image of Working Model

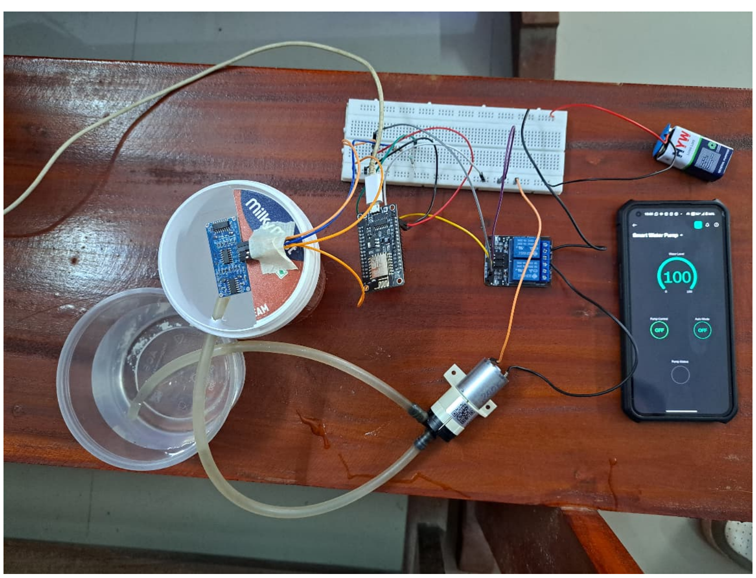

Figure 3: Built Prototype: NodeMCU, sensor, relay, and water tank

6 Expected and Observed Results

Parameter	Expected	Observed
Water level display	Real-time on Blynk	Accurate (±2%)
Pump control	Auto and Manual	Fully functional
Pump status LED	ON/OFF matches relay	Works correctly
Wi-Fi transmission	Stable and responsive	Slight latency if Wi-Fi weak

GitHub Repository: View Project on GitHub

7 Conclusion

This project demonstrates a reliable IoT-based water level monitoring and automated pump control node. Using NodeMCU and Blynk, it supports domestic/agricultural use, with real-time visibility and flexible control modes.