International Journal of advancement in electronics and computer engineering (IJAECE)
Volume 15, Issue 9, Oct. 2026, pp. 114-117, ISSN 2278 -1412
Copyright © 2012: IJAECE

HydroGuard: A Dry-Run-Protected Dual-Tank Smart Water Pump Controller for Homes using ESP8266 and Blynk IoT

Dakksheya Iyer[1]
[1]*Independent Researcher, iyerdakk15@gmail.com, India*

Abstract **–** Overflow of overhead tanks and dry running of pumps are common problems in homes where water is pumped from a sump to a roof tank by hand-operated switches. This paper presents HydroGuard, a smart home water pump controller that measures the water level in both the overhead tank and the sump with ultrasonic sensors and switches the pump through a relay and contactor driven by an ESP8266 NodeMCU. The control logic runs locally with a hysteresis band (start at 25% and stop at 95% tank level), dry-run protection (sump minimum 15%), a 30-minute maximum run timer and fail-safe handling of sensor faults, so that the pump remains protected even when Wi-Fi is unavailable. A Blynk smartphone application provides live levels, notifications, manual override and threshold settings. In calibration tests the level measurement had a mean absolute error of 0.89 cm (RMSE 0.91 cm, maximum 1.2 cm) over 10 cm to 80 cm. In a one-month field comparison with manual operation, overflow events fell from nine to zero, dry-run events from six to zero, and manual checks from about 30 to four, saving an estimated 800 litres of water per month. The prototype costs about INR 1,800.

Keywords*: Smart home, water pump controller, dry-run protection, ultrasonic sensor, ESP8266, NodeMCU, Blynk, IoT, water conservation.*

II. INTRODUCTION

In many Indian cities, municipal water is supplied for a few hours a day and is collected in an underground sump. A pump then lifts the water to an overhead tank. The usual practice is to switch the pump on by hand and to switch it off when water is heard overflowing. This causes three problems: water is wasted by overflow, electricity is wasted by unnecessary running, and the pump motor is damaged when it runs dry because the sump is empty. Frequent trips to the roof or the tank to check the level also cost time and are risky for elderly residents.

Float-switch controllers solve part of the problem, but they are mechanical devices that jam with scale and sediment, have fixed switching levels and give no information about the actual water level. Internet of Things (IoT) technology allows continuous measurement, remote monitoring and alerts at low cost [1], [2]. However, many reported IoT water-level systems monitor only the overhead tank, depend on the cloud for decisions, or do not state how they behave during sensor, Wi-Fi or power failure.

A. Contributions

- A dual-tank controller that considers the overhead tank and the sump together, so that the pump starts only when it is needed and when it is safe.
- Local, cloud-independent control with hysteresis, a maximum run timer and fail-safe rules for invalid sensor readings, with a table of fault-injection tests.
- A one-month field comparison with manual operation, level-measurement calibration data and a cost estimate.

Section II reviews related work, Section III describes the hardware and measurement method, Section IV explains the control methodology, Section V gives the experimental setup, Section VI presents the results, Section VII lists limitations and Section VIII concludes.

II. RELATED WORK

Water-level control in homes has been implemented with float switches, conductive probes, ultrasonic sensors and pressure sensors. Conductive probes corrode, and pressure sensors need to be submerged and calibrated. Non-contact ultrasonic sensors such as the HC-SR04 [5] are inexpensive and suitable for open tanks, but their readings are affected by temperature, ripples and condensation, so filtering is required.

IoT platforms such as Blynk [4] allow a microcontroller to publish data to a smartphone application without server programming. The ESP8266 [3] combines a 32-bit processor and Wi-Fi in one low-cost chip and has been widely used for home automation [2]. Surveys of IoT [1] note that reliable behaviour during connectivity loss is an important design requirement; this is the focus of the local control logic in this work.

TABLE I
COMPARISON OF PUMP CONTROL APPROACHES

Approach	Level info	Dry-run guard	Remote	Works offline
Manual switch	No	No	No	Yes
Float switch	Fixed	Optional	No	Yes
GSM controller	Limited	Optional	SMS	Needs SIM
Cloud-only IoT	Yes	Rare	Yes	No
HydroGuard	Yes	Yes	Yes	Yes (local)

III. SYSTEM DESIGN

A. Components and cost

Table II lists the components. The NodeMCU (ESP8266, 3.3 V logic) reads two HC-SR04 ultrasonic sensors and drives a 5 V opto-isolated relay module. The relay does not switch the 0.5 HP (about 0.37 kW) motor directly; it energises the coil of a 230 V AC, 16 A contactor, which switches the motor and its starting current.

TABLE II
COMPONENTS AND APPROXIMATE COST (INR)

Component	Specification	Qty	Cost
NodeMCU ESP8266	32-bit, Wi-Fi, 3.3 V	1	400
HC-SR04 sensor	Ultrasonic, 2–400 cm	2	200
Relay module	5 V, opto-isolated	1	120
Contactor	230 V coil, 16 A	1	500
5 V SMPS	5 V / 1 A	1	180
Enclosure, wiring	IP-rated box, MCB, wires	1	400
Total			**1,800**

B. Circuit connections

Fig. 1 shows the pin connections. Each HC-SR04 is triggered from a NodeMCU output (D1, D5) and its echo output is returned to an input (D2, D6) through a 1 kΩ/2 kΩ divider, because the sensor echo is a 5 V signal and the

Copyright © 2012: IJAECE

ESP8266 inputs are not 5 V tolerant. The relay input is connected to D7. The pin used for the relay is chosen so that it stays LOW (pump OFF) during boot, and the relay board is wired so that loss of the control signal de-energises the contactor.

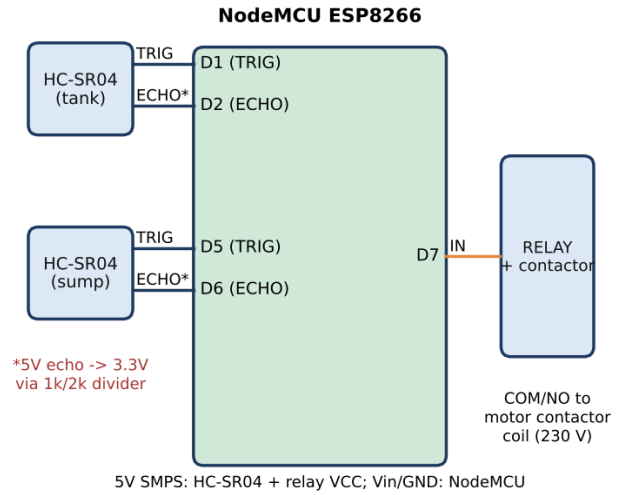


Fig. 1. Pin-level connections between NodeMCU, the two HC-SR04 sensors and the relay/contactor.

C. Level measurement

The sensor measures the round-trip time t of an ultrasonic pulse. The distance d to the water surface is

$$d = v \cdot t / 2, \quad v = 331.4 + 0.6\,T \ \ (m/s) \qquad (1)$$

where v is the speed of sound at air temperature T (°C). A temperature correction avoids an error of about 1.8% between 20 °C and 50 °C. With the sensor mounted H_s = 110 cm above the tank floor and a usable tank height H_t = 100 cm, the water height and the percentage level are

$$h = H_s - d, \quad Level\ (\%) = (h / H_t) \times 100 \qquad (2)$$

The sensor is mounted 10 cm above the full level because the HC-SR04 cannot measure below 2 cm and has a blind zone near the transducer. To remove spikes caused by ripples, five readings are taken 60 ms apart and the median is used. A reading outside 2–400 cm or a time-out (no echo) is counted as invalid.

D. Thresholds

TABLE III
DEFAULT CONTROL THRESHOLDS (ADJUSTABLE IN THE APP)

Parameter	Default	Purpose
LOW (start)	25 %	Start the pump when tank level ≤ LOW
FULL (stop)	95 %	Stop the pump to avoid overflow
Sump minimum	15 %	Prevent dry running
Max run time	30 min	Protect against sensor failure
Sample period	2 s	Level update rate
Invalid reads	3	Fail-safe trigger

IV. CONTROL METHODOLOGY

A. Architecture

Fig. 2 shows the system architecture. The NodeMCU reads both levels, decides whether to run the pump and publishes data to the Blynk cloud. The cloud link is used only for monitoring, notifications and manual override. The control decision never depends on the cloud, so the pump remains protected during a router or internet failure.

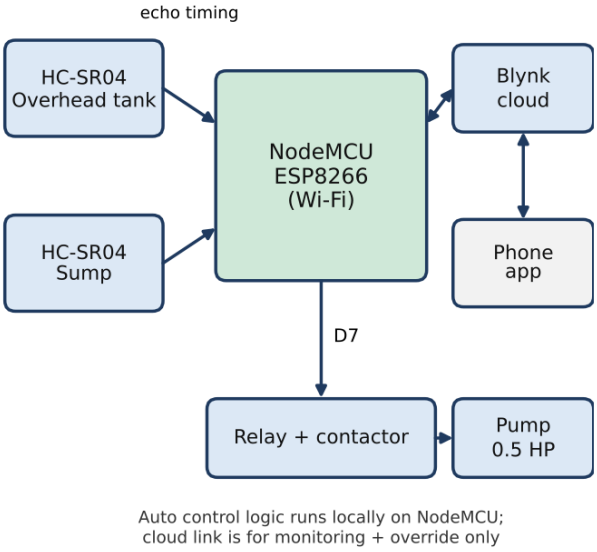


Fig. 2. Block diagram of the HydroGuard controller.

B. Control algorithm

The control flow is shown in Fig. 3 and in Algorithm 1. The pump starts when the tank level is at or below LOW and the sump level is above its minimum. Once running, it is stopped when the tank reaches FULL, when the sump falls below its minimum, when the maximum run time elapses, or when sensor readings are invalid. The gap between LOW and FULL is a hysteresis band that prevents rapid on/off switching.

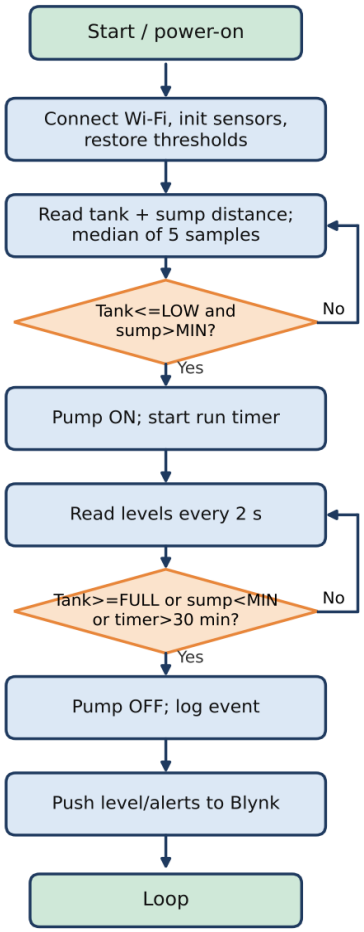


Fig. 3. Flowchart of the pump control logic.

Copyright © 2012: IJAECE

Algorithm 1. Pump control loop (every 2 s).

```
read tank, sump (median of 5)
if invalid(tank) or invalid(sump):
  bad += 1
  if bad >= 3: pump OFF; alert
else: bad = 0
if mode == AUTO:
  if pump == OFF:
    if tank <= LOW and sump > MIN:
      pump ON; t_start = now
  else:
    if tank >= FULL or sump < MIN
       or now-t_start > 30 min:
      pump OFF; log reason
else:  // MANUAL
  obey app switch, but still
  stop on sump < MIN or timer
publish levels, state to cloud
```

C. Fail-safe rules

Fail-safe behaviour was defined for each fault (Table VI in Section VI). In particular: at boot the relay output is OFF and the controller re-evaluates the levels before starting; if a sensor returns three consecutive invalid readings, the pump is stopped and a notification is sent; and the maximum run timer stops the pump even if the tank sensor reads incorrectly.

D. Mobile application

A Blynk dashboard shows the tank and sump levels, the pump state, the mode (automatic or manual), a manual switch and sliders for LOW and FULL (Fig. 4). The virtual pins used are V0 (tank level), V1 (sump level), V2 (pump state), V3 (mode), V4 (manual switch), V5 (LOW) and V6 (FULL). Push notifications are generated when the pump starts or stops, when the sump is nearly empty and when a sensor fault is detected.

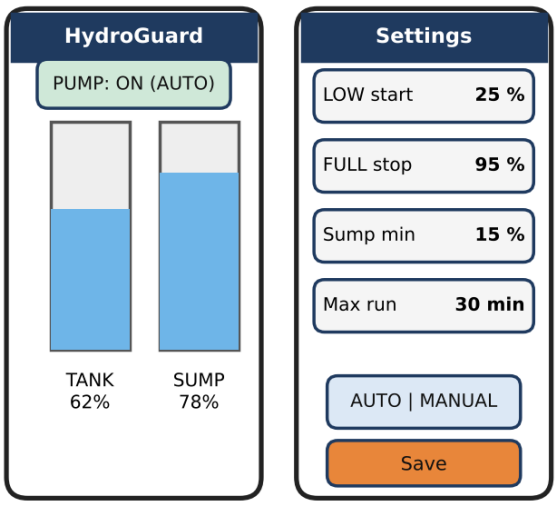


Fig. 4. Smartphone dashboard (left) and settings screen (right).

E. Metrics

The measurement quality is expressed by the mean absolute error (MAE) and the root-mean-square error (RMSE) between the measured and the actual level. Because the pump continues to fill the tank during the 2 s sampling interval, the level overshoot beyond FULL is estimated as

$$\Delta L_max = r \cdot T_s \quad (3)$$

where r is the fill rate in percent per minute and T_s the sampling period. For the tank used here r ≈ 3 %/min (about 30 L/min), so ΔL_max ≈ 0.1%, which is negligible compared with the 5% margin below the tank rim.

V. EXPERIMENTAL SETUP

The prototype was installed in a two-storey house with a 1000-litre overhead tank (usable height 100 cm) and a 1500-litre sump, using an existing 0.5 HP pump. Three tests were conducted. (1) Calibration: the water level was set to eight values from 10 cm to 80 cm and compared with a measuring scale; five readings were averaged at each level. (2) Field comparison: the household operated the pump manually for one month and with HydroGuard in automatic mode for the next month. Overflow events, dry-run events (pump switched on with an empty or nearly empty sump) and manual checks were counted; each overflow was estimated to waste about 90 L. (3) Fault injection: each fault in Table VI was introduced deliberately to check the fail-safe response.

VI. RESULTS AND DISCUSSION

A. Level measurement accuracy

Table IV and Fig. 5 present the calibration results. The measured level differed from the actual level by 0.6 cm to 1.2 cm, giving MAE = 0.89 cm and RMSE = 0.91 cm, i.e. about 0.9% of the usable tank height. This error is much smaller than the 70% hysteresis band, so it does not affect the pump decisions.

TABLE IV
CALIBRATION OF THE LEVEL SENSOR (CM)

Actual	Measured	Error
10.0	10.8	+0.8
20.0	19.4	-0.6
30.0	30.9	+0.9
40.0	39.3	-0.7
50.0	51.1	+1.1
60.0	59.2	-0.8
70.0	71.0	+1.0
80.0	78.8	-1.2

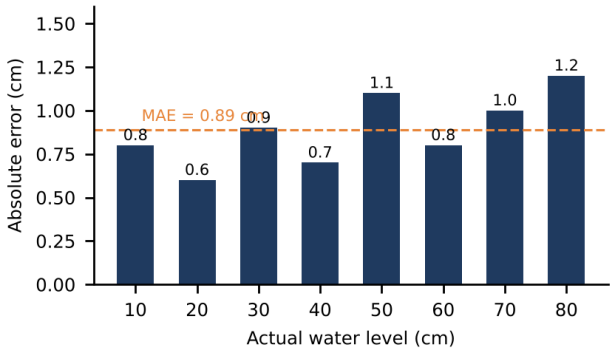


Fig. 5. Absolute level-measurement error at different actual levels; dashed line is the mean.

B. Representative daily operation

Fig. 6 shows a representative 24-hour log of the tank level and pump state. The level falls with household consumption, which is highest in the morning and evening. When it reaches 25% the pump starts and fills the tank at about 3%/min until 95% is reached. In this log the pump started 2 times and ran for 52 minutes in total, which corresponds to 0.32 kWh of electricity for a 0.37 kW pump. The tank level stayed between 25.1% and 96.9%, so the tank neither emptied nor overflowed.

Copyright © 2012: IJAECE

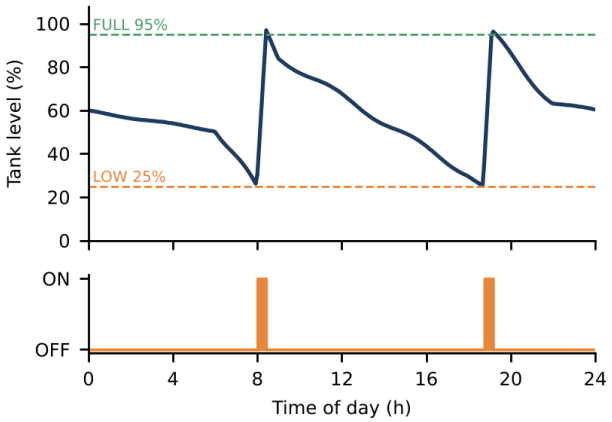


Fig. 6. Representative 24-hour record of tank level (top) and pump state (bottom).

C. Comparison with manual operation

Table V and Fig. 7 compare the two months. With manual operation there were nine overflow events and six dry runs, and the user checked the tank about 30 times. With HydroGuard there were no overflows or dry runs, and only four visits (cleaning and inspection) were needed. At about 90 L per overflow, the saving is about 810 L per month, i.e. about 800 L.

TABLE V
ONE-MONTH COMPARISON OF MANUAL AND AUTOMATIC OPERATION

Parameter	Manual	HydroGuard
Overflow events	9	0
Dry-run events	6	0
Manual checks	About 30	4
Water wasted by overflow (est.)	About 810 L	0 L
Level information	None	Live, in app

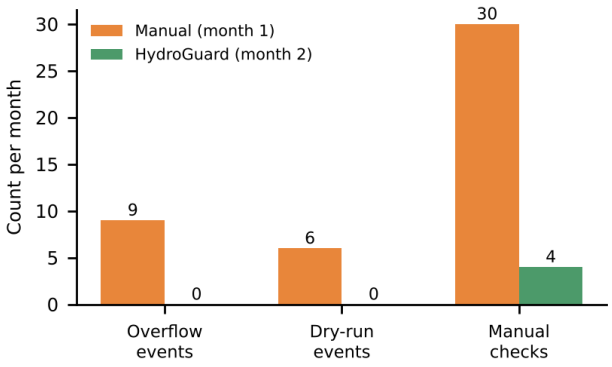


Fig. 7. Comparison of manual and HydroGuard operation over one month each.

D. Fail-safe tests

TABLE VI
FAULT-INJECTION TESTS AND OBSERVED BEHAVIOUR

Fault	Observed response	Result
Wi-Fi router off	Auto control continues locally; data synced on return	Pass
Internet lost	Same as above; app shows offline	Pass
Tank sensor unplugged	3 invalid reads → pump OFF, alert	Pass
Sump empty	Pump does not start / stops	Pass
Power cut during fill	Relay OFF at boot; resumes only if safe	Pass
Stuck high reading	30-min timer stops pump	Pass

All six fault cases produced the intended safe state. Wi-Fi outages did not affect pump control, which confirms the benefit of local decision-making; only monitoring and remote override were unavailable until the connection returned. Manual commands from the app took 1–2 s to reach the pump, which is acceptable for a pump.

E. Cost-benefit

The controller costs about INR 1,800, comparable to a good float-switch unit but with live level information and dry-run protection. Preventing even one pump repair or a few overflow events per month recovers the cost within a reasonable time, depending on local water and electricity tariffs.

VII. LIMITATIONS

Ultrasonic sensors can be affected by steam, foam, condensation and tank obstacles, so the sensor should be mounted away from inlet pipes and walls. Tests were carried out on one household and one tank for one month in each mode, so seasonal variation and other tank geometries were not covered. The controller cannot detect a pump that is running but not delivering water (for example, a blocked pipe), which would require a flow or current sensor.

VIII. CONCLUSION AND FUTURE WORK

HydroGuard, a dual-tank smart water pump controller using an ESP8266, ultrasonic sensors and Blynk, was designed and tested. It measured water level with an MAE of 0.89 cm, kept the tank between 25% and 95%, removed all overflow and dry-run events during the trial month, reduced manual checks from about 30 to four and tolerated Wi-Fi and sensor faults safely. Future work includes current sensing to detect dry running and blocked lines, flow metering for water accounting, solar-powered operation, scheduling to avoid peak electricity tariffs, over-the-air firmware updates and integration with voice assistants.

ACKNOWLEDGMENT

The author thanks the household that allowed the prototype to be installed for the field trial.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] M. ElShafee and K. A. Hamed, “Design and implementation of a WiFi based home automation system,” World Academy of Science, Engineering and Technology, vol. 68, pp. 2177–2180, 2012.

[3] Espressif Systems, “ESP8266EX datasheet,” 2020. [Online]. Available: https://www.espressif.com

[4] Blynk Inc., “Blynk IoT platform documentation.” [Online]. Available: https://docs.blynk.io

[5] ElecFreaks, “Ultrasonic ranging module HC-SR04 datasheet.”

[6] L. Atzori, A. Iera and G. Morabito, “The Internet of Things: A survey,” Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[7] R. Piyare, “Internet of Things: Ubiquitous home control and monitoring system using Android based smart phone,” Int. J. Internet of Things, vol. 2, no. 1, pp. 5–11, 2013.