IoT Management Solution: A Case Study of Fill Level Monitoring for Gravel Silos

Saif Aldeen Saad Obayes Al-Kadhim¹, Ibtesam R. K. Al-Saedi²

¹ Shiite Endowment Diwan Babylon, Iraq.
² Communication Eng. Dept. University of Technology (UoT) Baghdad IRAQ.

Saifuldeen.Saad@ieee.org
drkarhiy@yahoo.com

Abstract. In order to make ubiquitous system, technologist uses different type of equipment in perception layer, for the environment sensing and sensory data processing. This equipment’s are sensors, data processing and sending devices. Sensors sense the environment and create the data accordingly. Sensor generated data need to be processed with some processing device and can be send over the network. This research paper introduced the basic application based on IoT. The application includes Ultrasonic distance sensor and NodeMCU for obtaining the fill level of the gravel silos. We are publishing our data over the web using Zerynth studio portal and showing the analysis graph of application near real time using the Internet of Things (IoT) concept. Paper also describes server configuration on the NodeMCU and real-time response to web browser over the WLAN.

Keywords. internet of thing, pervasive system, NodeMCU, sensors.

1. Introduction

The basic idea for the IoT is to connect anything anytime anywhere and the automation is one of the parts of the ubiquitous system which involves all most all the domains [1]. As IoT Technology emerges new domains, the challenges and the facilities of new hardware and software came into existence[2]. Different type of hardware and software are now available that can help for the automation and environment sensing [3]. Different kinds of simple industrial systems which are not automated needs automation to improve the quality of work, moreover, these systems should be responsive whether there is someone in office or not. One of the systems that can be automated is fill level of the gravel silos.

There are many gadgets available in the market for measuring the fill level of an industrial fill silo, but the capabilities to connect to the internet is the limitation of those systems, and are not adaptable to extend other home appliances or systems for automation. NodeMCU (Node Micro Controller Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266, which is capable of sensing the environment using its General-Purpose Input Output (GPIOs) through sensors. We can use simple and economical sensor devices connected to the GPIO pins of NodeMCU. The NodeMCU computer board holds a CPU, RAM,
networking (Wi-Fi), and even a modern operating system and SDK. When purchased at bulk, the ESP8266 chip costs only $2 USD a piece [4]. That makes it an excellent choice for IoT projects of all kinds, which consumes low electricity but very powerful in processing. A variety of interfaces is given for external devices. Some of these devices are essential like display and others are optional.

NodeMCU GPIO ports can interact with sensors or actuators and can also be programmed for a specific need. Different operating systems are available for NodeMCU and most of which are open source [5]. In this paper, all the experiments presented, are done on eLua (embedded Lua), which is a very simple and fast scripting language with an established developer community [6]. The sensor we have used for the experimental purpose is HC-SR04 Ultrasonic distance sensor, which can be connected to GPIOs of NodeMCU to measure the distance between the object and sensor. The Ultrasonic distance sensor emits the sound pulses and receives the bounce back pulses from the object. By calculating the time taken from an emission of the pulse and receiving back, we can measure the distance of the object that comes in front of this sensor. NodeMCU is rich from the programming point of view. It supports C/C++, Python and Java programming languages. In our experiment Python scripting language has been used, which is easy to program with GPIOs and has an active community for discussion.

Our experiment includes the basic infrastructure to create the pervasive environment for the Internet of Things (IoT) [7]. The basic IoT involves the infrastructure like networks, data analytics, data management, security etc., and make the IoT system powerful [8].

The structure of this paper is as follows: Section 2 deals with technical detail of NodeMCU v3 ESP8266, in the Section 3 System design, Hardware & Software requirement has been discussed, Section 4 describes Experiments and results, section 5 covers Conclusion and at last references.

2. NODEMCU 3 ESP8266

The NodeMCU v3 ESP8266 (Figure 1) is a Tensilica Xtensa LX106 System on Chip (SoC) with 80 MHz CPU. It contains 64 KiB of instruction RAM, 96 KiB of data RAM. External QSPI flash – 512 KiB to 4 MiB (up to 16 MiB is supported). Energy saving devices are much appreciable when we talk about IoT, because lots of IoT devices takes less human interaction and makes machine to machine communication; 32-bit RISC architecture [9] work on low energy and makes NodeMCU efficient. It can also work as a WSN node. Dimension of the NodeMCU is credit card size but includes lots of external interfaces like four USB ports, IEEE 802.11 b/g/n Wi-Fi [10], SPI, FC, FS interfaces with DMA (sharing pins with GPIO), UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2, 10-bit ADC,16 GPIO pins and micro USB power input that can handle 2.5 Amps [11].

GPIOs pins are the interface for the outer world, using these pins, sensors can sense the nearby environment thus it can be used in electronics projects. The NodeMCU can be used as a simple computer for many purposes like word processing, internet surfing, enjoying games by attaching a keyboard and mouse.

![Figure 1. NodeMCU v3 ESP8266 [11].](image)

NodeMCU can send sensory data through gateway or coordinator and over the network as well. It can be used as a wireless sensor node and also as a gateway [12]. This device is capable of working as a node, or act as a gateway, or even as coordinator of wireless sensor network [13]. There are other devices like Arduino, BeagleBoard, Intel Galileo, Raspberry Pi available to substitute the
NodeMCU, but NodeMCU is low-cost device and has lots of functionality as compared to other devices.

3. SYSTEM DESIGN: HARDWARE & SOFTWARE

This paper describes the implementation of automatic fill level detection of filling gravel silo using NodeMCU and ultrasonic distance sensor. Python is used as programming language because of its simplicity, available support and popularity in sensory projects. The NodeMCU supports a number of operating systems and we are using eLua. We can further extend the system in the context of the wireless sensor network with IoT using Xbee, RFID, sensor-network, M2M connection [14], but for this paper, we are concentrating on ‘to build a base for one of the IoT system device’. Enabling the Wireless Sensor Network (WSN) using Xbee based on ZigBee protocol for sending the information in RF (Radio frequency) form is time-consuming and costly [15], but WSN has its own advantages and facilities, if the data need in near real-time.

3.1. NEED OF THE SYSTEM

Daily factory activity consists of several kinds of tasks, which needs to be automated. One of them is checking the fill level of the fill silo while filling. Automation of the system brings different facilities to the user and improves the style of work. The proposed system is not only used in the given context but can also be used in the different applications like in the underground sump-pit to avoid overflow. It can also be mounted in car washing system for detecting and washing the car automatically.

3.2. PROPOSED SYSTEM

The objective of this paper is to enable the basic concepts of IoT in order to build a pervasive system. The Fill Level Monitoring (FLM) is Ultrasonic Distance Sensor based device that measures the fill level of the gravel silo using NodeMCU. The FLM system alerts when the silo is empty or when fill, based on the Threshold value, which is calculated dynamically. The threshold value is different for the different silo volumes. The goals of our system are:

- Regular readings of the fill depth in the gravel silo.
- Alerts when the silo is empty or when full based on the threshold value.
- The threshold value is calculated using the volume of the silo, it is not based on the distance and it is generated dynamically for the different volume silo.
- On-site readings of the current fill level.
- Off-site (LAN based) Web-based information of FLM activity.
- Automatic restart of the process after an unexpected failure.

3.3. HC-SR04 ULTRASONIC DISTANCE SENSOR

Ultrasonic distance sensors are designed to measure the distance between the source and target using ultrasonic waves. We use this sensor because it uses ultrasonic waves and it is relatively accurate across short distances and doesn't cause disturbances because they are inaudible to the human ear. HC-SR04 is a commonly used sensor for non-contact distance measurement with distances from 2cm to 400cm [16]. The ultrasonic sensor uses sonar to scale distance with accuracy and stable readings. It consists of an ultrasonic transmitter (trig), receiver (echo) and a control circuit. The transmitter transmits short bursts which get reflected by the target and are picked up by the receiver. The time difference between transmission and reception of ultrasonic signals is calculated. Using the speed of sound and “Speed = Distance/Time” equation, the distance between the source and target can be easily calculated.

3.4. CIRCUIT DESIGN

The NodeMCU can be easily connected with different type of sensors using General Purpose Input Output (GPIO) pins. GPIO pins are the interface to the outer world for the NodeMCU; sensor senses the
environment specifically for which it made and output the data accordingly. The HC-SR04 Ultrasonic distance sensor has been used in this experiment and has four pins: ground (Gnd), output pulse echo (Echo), input pulse trigger (Trig), and 5V supply (Vcc). The Gnd pin is used for voltage connection to the earth, Trig and Echo pins are output and input pins for NodeMCU respectively which are handled by Boolean value. The Vcc pin activates the sensor by applying 5V. The NodeMCU command the Trig pin to send the sonic pulses by making it ‘TRUE’ and when echo receives the bounce back pulses after colliding from the hurdle, it makes the input pin ‘TRUE’. The experiment is done using the python scripting and object-oriented methodology is used for the programming. The General-Purpose Input Output (GPIO) pins are digital pins on the NodeMCU board. Each pin can have only two states: a low voltage state and a high-voltage state, representing 0 and 1 respectively. From the NodeMCU Lua application, can read the state from each pin, and then set the state. In the scripting, two pins are assigned as Trig (GPIO.OUT) and Echo (GPIO.IN) and other pins are used for power 5V the sensor and for ground. First, the trig pin is set with ‘FALSE’ status for settling the circuitry and then activated with ‘TRUE’ signal for 10µS and then set with ‘FALSE’ status so trig will trigger the sonic pulses. Immediately after this activity we activated echo to ‘TRUE’ and measure the timing between two events; when the echo pin was false and when immediately the echo pin gets true value.

Four GPIO pins of the NodeMCU is used, to power the sensor GPIO 5V (pin 2) connected to Vcc, GPIO GND (Pin 6) is used for ground the circuitry, Trigger the sonic pulse GPIO 23 (Pin 16) plugged into trig which is GPIO output and GPIO 24 (Pin 18) is used for ECHO which is GPIO Input. For the experimental flexibility point of view, the connection is built using the breadboard and jumper wires that can also be fixed permanently as per the need. The snapshot of the complete system is given in figure 2. The time taken by this process can give the distance of the object.

![Figure 2. The FLM system.](image)

Used sensor needs a 10µS pulse to enable the trig pin, activating the trig pin for 10µS emits the required sonic pulses. Receiver pin (echo) is indicating ‘FALSE’ until it does not receive the bounce back pulses which transmitted by the trig. Once echo shows signal ‘TRUE’ the python program calculates the distance between sensor and fill level. Echo pin release 5V as the output pulse and NodeMCU GPIO pins can have 3.3V for reading the signal, so applying 5V can damage the NodeMCU. Therefore, the circuit needed voltage divider register (Figure 2), and it can be easily calculated using the voltage divider formula.

\[ V_{out} = V_{in} \times \frac{R_2}{R_1+R_2} \] (1)

Value of \( V_{out} \) is 3.3V and \( V_{in} \) is 5V. The value of register \( R_1, R_2 \) can be determined by assuming one of the values of the register [17]. In our experiment, we took \( R_1 \) as 1kΩ and calculation give the value of \( R_2 \) approx. 2kΩ. So voltage divider circuitry places 1kΩ register in between echo and NodeMCU GPIO pin, and 2kΩ register between the ground and GPIO pin used for echo (Figure 3).
4. EXPERIMENT AND RESULTS

Python scripting has been done for the experiment, handling GPIOs using the python is quite easy and less complex as compare to other languages. The paper [18] is about API developed for the NodeMCU GPIOs for the different type of sensors, but we have used the python programming from the scratch so that it will be easy to customize the application and will have in-depth knowledge to handle sensor.

The distance between the ultrasonic distance sensor and the surface of the fill is calculated using the simple formula “Speed = Distance/Time”.

\[ \text{speed_of_sound} = \frac{34300}{\text{in cm/sec}} \quad \text{duration} = \text{pulse_end} - \text{pulse_start} \]

\[ \text{distance} = \text{duration} \times \left(\frac{\text{speed_of_sound}}{2}\right) \quad (2) \]

The speed of the sound is 34300 cm/sec and time is calculated from the pulse_start and pulse_end. The pulse_start is the time when trig pin is activated by ‘TRUE’ and pulse_end is the time when bounce back pulse encounters to echo. Activating and deactivating the GPIO pins using python scripting pulse_start time and pulse_end time is determined. The speed of sound is divided by two because of the ultrasonic pulse travel the distance twice, first from the sensor to the fill surface and then bounce back to the sensor.

The threshold value for overflow alarm is dynamically generated. The volume of the silo has been determined using below formula:

\[ v = \pi \times r^2 \times h \quad (3) \]

Where ‘\( \pi \)’ is constant with value 22/7, ‘\( r \)’ radius of the cylinder and ‘\( h \)’ is the height of the cylinder. So using the distance as ‘\( h \)’ the volume of the silo is easily determined.

Most of all fill silos are cylindrical in shape so we took this formula, but as per the requirement, the formula to calculate volume can be changed. Threshold value is taken using volume formula of the cylinder and since the fill silos are not in standard cylindrical shape always, so for the safe margin 75% of the original volume is taken as full volume for the experiment, and also volume of the silo is subject to customization at the deployment time of the system. When the silo is filled up to threshold value of the silo, our system shows the message of overflow (Figure 4). It indicates the underflow if the silo is empty.

![Figure 3. HC-SR04 sensor with voltage divider circuitry.](image-url)

![Figure 4. Experiment result.](image-url)
4.1. SERVER CONFIGURATION
The Server-side scripting has been done for the FLM in the NodeMCU, so it acts like sensor node as well as coordinator itself. The lightweight server ‘Flask’ is used that work fine on the NodeMCU. An HTML page with jQuery scripting is developed for showing the result of the experiment so the server can be updated with the near real-time processed data of the sensor. The HTML page with jQuery is deployed on the server. The deployment process is simple web resource deployment and explanation of it, is out of the scope of this paper. The user can observe the compiled information from the local-host. The script is written only for localhost and it meant to be extended.

4.2. ZERYNTH PLATFORM
The Zerynth studio is an IoT platform that allows analytics, collection and act on the sensory data. We have configured our project through IoT based www.Zerynth.com portal, where IoT project can be linked. We created a daily-base chart of the filling activity of the gravel silo. The graph (figure 5) showed here the range of fill in the silo which dynamically change over the data.

Anyone can access the graph and reports of the project on Zerynth platform if it is made public; all we need is channel number. Our project is public to all in the Zerynth platform. The Zerynth studio allows us to do analysis of sensory data of the application and can does mathematics on it. It also stores sensory data and provide XML, Json, csv file formats of the sensory data. In the different applications these file formats can be used to get benefited of our project.

![Figure 5. Daily analysis of fill level on Zerynth studio.](image)

5. CONCLUSION
In order to make the pervasive system, a small concrete plant system needs to be automatic and must be part of the pervasive system. The proposed system is cheap, easily customizable and accurate. It can easily upgrade to IoT ecosystem. The server configuration is done in this project but it limited to local area network. In order to overcome this problem, we linked our application to Zerynth platform. To make an application, internet of thing (IoT), truly we need to connect this system to the internet.

References
[1] M. Weiser, “Hot topics-ubiquitous computing,” Computer (Long. Beach. Calif.), vol. 26, no. 10, pp. 71–72, 1993.
[2] S. G. Glisic, Advanced Wireless Communications and Internet: Future Evolving Technologies: Third Edition. 2011.
[3] P. Vermesan, O., & Friess, Internet of Things: Converging Technologies for Smart Environments. Aalborg: River Publishers, 2013.
[4] Divyah Bala, Esp8266: Step by Step Tutorial for Esp8266 Iot, Arduino Nodemcu Dev Kit. Independently Published, 2018, 2018.
[5] A. Carlos Bento, “IoT: NodeMCU 12e X Arduino Uno, Results of an experimental and comparative survey,” Int. J. Adv. Res. Comput. Sci. Manag. Stud., vol. 6, no. 1, 2018.
[6] A. Costin, “Lua Code: Security overview and practical approaches to static analysis,” in Proceedings - 2017 IEEE Symposium on Security and Privacy Workshops, SPW 2017, 2017.
[7] L. Shkurti, X. Bajrami, E. Canhasi, B. Limani, S. Krrabaj, and A. Hulaj, “Development of ambient
environmental monitoring system through wireless sensor network (WSN) using NodeMCU and ‘WSN monitoring,’” in 2017 6th Mediterranean Conference on Embedded Computing, MECO 2017 - Including ECYP 2017, Proceedings, 2017.

[8] R. Nabati and S. Taheri, “THE INTERNET OF THINGS (IOT) A SURVEY,” TURKISH ONLINE J. Des. ART Commun., 2016.

[9] A. H. Limited, “Architecture Reference Manual,” Arm Ddi E, 2000.

[10] IEEE Computer Society, “Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs),” IEEE Stand., 2006.

[11] EINSTRONIC, “Nodemcu,” Einstronic, 2017.

[12] M. F. Wicaksono, “Implementasi Modul Wifi Nodemcu Esp8266 Untuk Smart Home,” J. Tek. Komput. Unikom, 2017.

[13] K. S. Mohamed and K. S. Mohamed, “IoT Physical Layer: Sensors, Actuators, Controllers and Programming,” in The Era of Internet of Things, Cham: Springer International Publishing, 2019, pp. 21–47.

[14] G. R. Sakthidharan and A. Punitha, “Wireless Sensor Network in Automation and Internet of Things,” 2017.

[15] N. Gupta and Z. Hashim, “Recent Trends and Applications of Materials used in Fabrication of Devices for Wireless Sensor Network,” in Materials Today: Proceedings, 2017, vol. 4, no. 9, pp. 10332–10335.

[16] Indoware, “Ultrasonic Ranging Module HC - SR04,” Datasheet, 2013.

[17] D. McBrearty, “Voltage dividers,” in Electronics Calculations Data Handbook, 1998.

[18] E. M. F. De Jesus and M. C. Marques Neto, “LibsensorPy: A library to improve the development of ubiquitous applications on raspberry Pi,” in WebMedia 2015 - Proceedings of the 21st Brazilian Symposium on Multimedia and the Web, 2015.