IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

# Prototype of wireless sensor network based on open industrial protocol: modbus

# Prototype of wireless sensor network based on open industrial protocol: modbus

**D Y Tadeus*, Yuniarto and T Yuwono**

Industrial Technology Department, Vocational School, Diponegoro University, Semarang, Indonesia

*Email: distayoel@live.undip.ac.id

**Abstract.** Wireless Sensor Network (WSN) is generally implemented using multiple sensor nodes located in remote locations or relatively far from the control room and monitoring. WSN conventionally has its own communication protocol architecture that requires a specific interface system in accordance with its application. The software used in WSN interface system development is very varied and requires deep knowledge of programming skills because generally the interface is built from scratch so it will be difficult for quick and wide implementation requirements. This research provides solutions to these problems through a new design that is alternatively different from conventional WSN. The main difference lies in the communication protocol used, namely the open industrial protocol: Modbus. Implementing this protocol on WSN will provide access to industry-standard interfaces to be used as a WSN interface system. The industry standard interface system does not requires any complicated programming skills because it has been provided by ready-to-use graphical components so project implementation can be executed more quickly and easily. The Wifi ESP8266 module is used as a wireless transceiver component on the node sensors embedded with Modbus TCP protocol over TCP/IP network protocol. Interface system is built using Microsoft Visual Studio Community version and AdvancedHMI as an integrated development environment. The prototype was built and tested in an environment temperature monitoring system.

## 1. Introduction

Wireless Sensor Network (WSN) has enormous potential for implementation, including: in the military field for tracking targets and surveillance, prevention and mitigation of risks from natural disasters, medical health monitoring, limited and dangerous environmental exploration, and measurement of environmental variables within fields of agriculture, animal husbandry, fisheries, marine, and climate as shown on Fig. 1[1][2]. Another example, WSN is used in conjunction with a water sprinkling tractor to form an integrated agricultural land irrigation system [3], implementation of WSN on security systems using Bluetooth protocols[4] tracking system architecture of sports team players by implementing hybrid location technology for better accuracy and aggregation of data for energy efficiency[5]. There are bins that monitored by Wireless Sensor Network (WSN) to get information about unfilled level of bins through a central monitoring station and evaluation of wireless links [6]. In every bins the Wireless Monitoring Unit (WMU) are installed with sensor and transmit data measured from WMU to the to the Wireless Access Point Unit (WAPU). It receives data from the WMU's and transmit data to central monitoring station, and then forwarded to gateway system, the level of the bins is monitored by human machine interface/application.

[7] identifies the wireless sensor network communication on potential effects associated with the operation, environment and structure of wind turbines, a reliable routing protocol for wireless sensor networks is proposed and evaluated in the boundaries of wind turbines application. [8] describes the application of WSN in environmental monitoring, with particular on water quality, monitored

parameters such as microcontroller units (MCU) and wireless communication standards, data security implementation, power supply architectures, and potential application scenarios is compared and evaluated that is proposed by various authors. WSN is typically implemented using multiple sensor nodes. Each sensor node has transducers, processor, memory, power supply, transceiver, and actuators. Unlike traditional networks, WSN has its own standard and design architecture and resource constraints such as limited energy sources, short communication distances, low bandwidth, and limited processing and storage capabilities in each sensor node [2]. Table I. provides a technical comparison of the commonly employed WSN standards [8].There are limitations to the communication protocols used as well as the acquisition, presentation, and data management systems. WSN has communications protocol architecture that requires a specific interface system in accordance with its application. Some examples are the WSN interface built by[9] using MoteWorksver software. 2.0F, Power Calculator Spread Sheet PCSS, Microsoft Excel, MATLAB. The WSN interface built by [10] uses Microsoft VB.NET software. Both examples show that the software used in the development of WSN interface systems can vary greatly and differ in their usage and also require adequate programming knowledge and skills that will make it difficult for rapid and extensive implementation.

This research provides a solution through a new design approach especially on utilizing open industrial protocol: Modbus for communication protocol of WSN. Modbus is one of the industry-standard communication protocols, it is supported by many application programs that no longer require deep knowledge of programming, as researchers have done in interface system design for the development of Scada training modules [11]. This research utilizes an ESP8266 module as a wireless transceiver component on the sensor nodes embedded with Modbus TCP protocol over TCP/IP network protocol. The interface system is built using Microsoft Visual Studio Community version and AdvancedHMI as an integrated development environment. The prototype was built and tested in a temperature monitoring system environment.

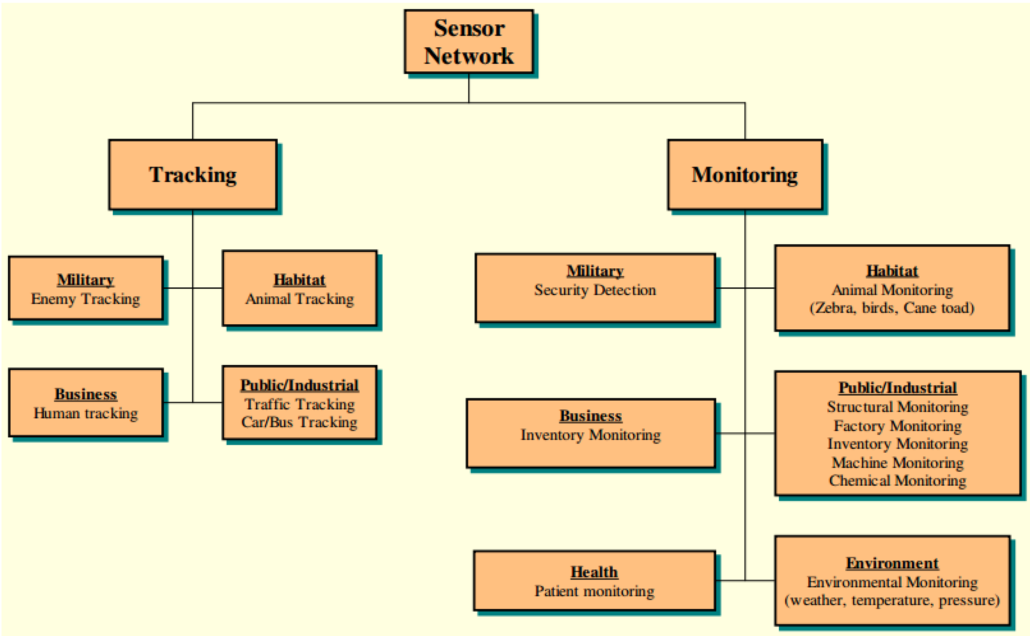


**Figure 1.** Potential applications of WSN on multidiscipline sector.

IOP Conf. Series: Materials Science and Engineering **801** (2020) 012140 doi:10.1088/1757-899X/801/1/012140

**Table 1.** Technical comparison of the commonly employed WSN standards.

| Wireless standards | Frequency band (MHz) | Comparative power consumption | Range capability (LoS) | Network topology | Data rates (kbps) |
|---|---|---|---|---|---|
| IEEE 802.15.4 | 2,400, 915, 868 | Low | 100 m+ | Star | 250, 40, 20 |
| ZigBee | 2,400, 915, 868 | Low | 100 m+ | Star, tree and mesh | 250, 40, 20 |
| Bluetooth | 2,400 | High | 10-100 m | P2P and star | 720 |
| LoRaWAN | 915, 868 | Very low | 10 km+ | Star | 50 |
| SigFox | 902, 868 | Very low | 40 km | Star | 1 |

## 2. Experimental details

The sensor nodes are built with battery, temperature sensor TMP36, and ESP8266 Wifi module as a main processing unit, data acquisition, and wireless communication as shown on Fig. 1. Fig. 2 shows multiple sensor nodes and interface system are connected to local network through a wireless access point. Interface system acts as client and sensor node act as server.

TMP36 will produce analogue voltage equivalent to temperature readings. It has 10mV/0C output voltage scaling, 0.5V offset voltage, and produces 750mV output voltage at 250C. Sensor's output voltage will be converted to digital form by 10 bit embedded Analog to Digital Converter (ADC) on ESP8266 with 3.3V reference voltage on terminal Analog Input as shown on Fig. 3. The converted value will be in the range of 0 – 1023 decimal and stored in analogue holding register 0 address.

Modbus TCP protocol is developed on Arduino Integrated Development Environment (IDE) to be transferred to ESP8266 and each of them is set with unique IP address. It was done by configuring reserved IP address on the access point wireless Local Area Network settings according to Media Access Control (MAC) address of each sensor node. Dynamic Host Configuration Protocol (DHCP) server is used to distribute IP address dynamically, so new sensor node can be added to network easily. Each sensor node is set with 192.168.0.150, 192.168.0.151, 192.168.0.152 IP address respectively as shown on Fig. 5, Fig. 6, and Fig. 7.

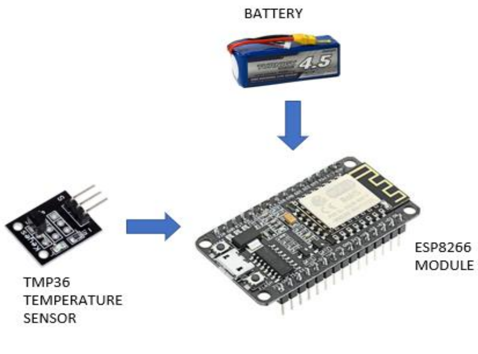


**Figure 2.** Sensor node's components.

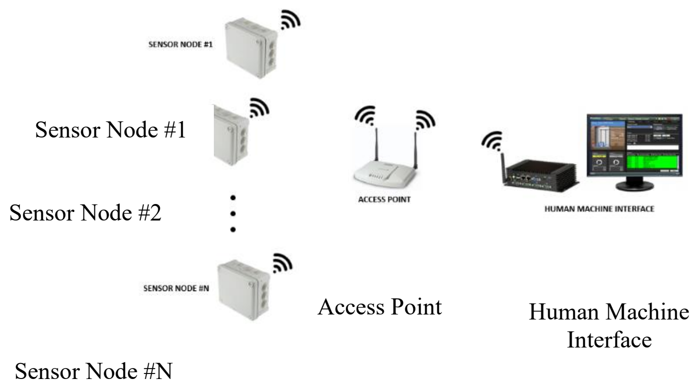


**Figure 3**. WSN topology design.

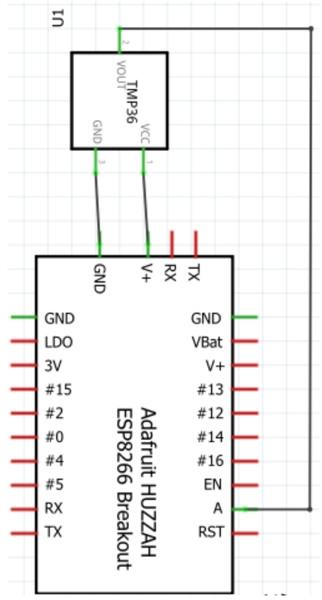


**Figure 4.** Schematic diagram for TMP36 temperature sensor connection to ESP8266.

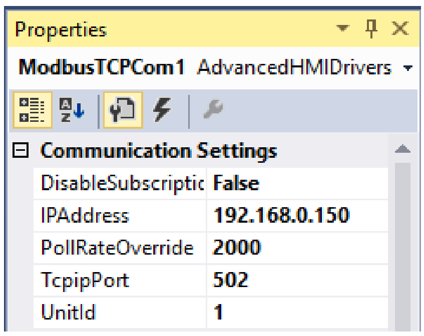


**Figure 5.** IP address setting for sensor node #01.

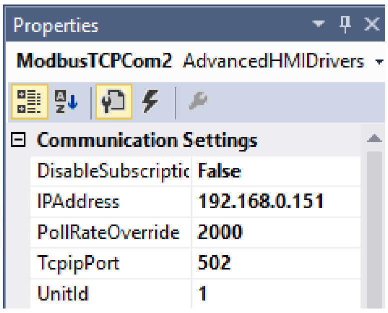


**Figure 6**. IP address setting for sensor node #02.

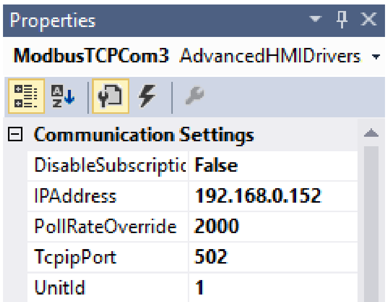


**Figure 7.** IP address setting for sensor node #03.

**Table 2.** Modbus addressing format.

| Coil Numbers | Type | Name |
|---|---|---|
| 00001-09999 | read/write | Discrete Output Coils |
| 10001-19999 | read only | Discrete Input Coils |
| 30001-39999 | read only | Analog Input Registers |
| 40001-49999 | read/write | Analog Output Holding Registers |

The interface system is developed on Microsoft Visual Studio Community Version and supported by AdvancedHMI as a ready to use graphical library and Modbus TCP client driver. Modbus TCP client driver component is added and configured for each sensor node's IP address. It used native format Modbus addressing as shown on Table. II. Temperature readings from sensor nodes is read with address format 40001 for accessing analog holding register 0 (offset +1) and displayed on interface system by seven segment visual components along with periodically two dimensions data trends and data exporter (.log format) for further purposes. The value received need to be converted back to Celsius temperature unit by adjusting scale and offset value parameter provided on visual component's properties dialog box. Equation (1) and (2) is used to convert digital value to Celsius temperature unit,

$$V_o=(digital\ value/1023)*3.3 \quad (1)$$

$$Temp\ (^0C) = (V_o - 0.5)/0.01 \quad (2)$$

1023 is decimal number obtain from 10 bit ADC resolution ($2^{10}$), 3.3 is the voltage reference used by ESP8266. 0.5 is sensor's voltage offset and 0.01 is sensor's voltage output scaling  Scale and offset value properties can be obtained by combining (1) and (2), thus

$$Temp\ (^{0}C) = digital\ value*0.32258 - 50 \tag{3}$$

Equation (3) shows the standard format accommodating scale and offset, the theoretical scale value is 0.32258 and the offset value is -50.

## 3. Results and discussions

Figure 8 shows the result of local network scanning application running on smartphone android operating system, it displays all connected devices on network including all three sensor nodes. It confirms that each sensor node has successfully connect to wireless access point with predefined IP address. Fig.9 shows the interface system running on Windows based operating system. There are temperature readings displayed on the screen from three sensor nodes. The seven segment visual component display temperature nominal in numeric form, needle gauge visual component display temperature in analogue meter format to emphasize safe operating range of the measurements, data trend visual component display temperature in historical format to emphasize changes in temperature dynamic. The temperature value obtained from sensor nodes need to be calibrated with another temperature sensor for correct readings. The common digital room temperature sensor is used for that purpose as shown on Fig. 10 Correction action is taken by adjusting offset value properties on each visual component. Each sensor node has different offset value from another as shown on Fig. 9. It shows that in field calibration action is mandatory even all sensor has calibrated before by manufacturer. Fig. 10 show the temperature readings log generated by DataLogger visual components, it produced time stamped format data log with its comma separated value so it can be imported easily by common spreadsheet application like Microsoft Excel.  Fig. 9 shows the interface system application with all visual components successfully read data from analogue holding register 0 located on each sensor node through Modbus TCP protocol.

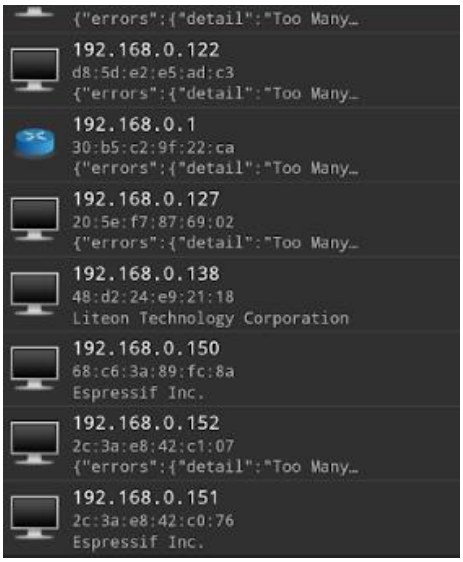


**Figure 8**. Result from IP scanner android application.

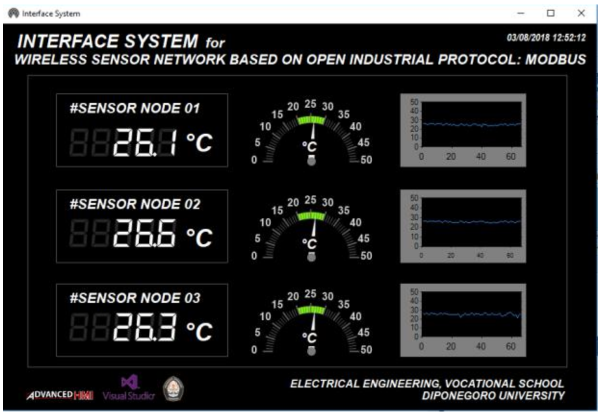


**Figure 9.** Interface system running on Windows based PC.


**Figure 10.** Digital thermometer for calibration purpose.

## 4. Conclusions

The open industrial protocol: Modbus had implemented successfully on WSN application. It offers scalability, simplicity, less time consuming on developing interface system because it did not require deep programming skills, simply using drag and drop concept. This research may be expanded to implementation on another platform and operating system such as embedded PC, Raspberry, Linux based operating system, and Mac OS. Because Modbus is one of industrial standard protocol, then WSN is feasible to be integrated on existing industrial applications.

## Acknowledgement

The authors would like to express heartfelt thanks to Electrical Engineering, Department of Industrial Technology, School of Vocation, Diponegoro University.

## References

[1] Othman M F and Shazali K 2012 *Wireless sensor network applications: A study in environment monitoring system*, in Procedia Engineering, **41** pp 1204–1210.
[2] Yick J , Mukherjee B, and Ghosal D, 2008 “Wireless sensor network survey,” *Comput. Networks*, **52**(12), pp. 2292–2330.
[3] Kim Y, Evans R G, and Iversen W M, 2008 “Remote Sensing and Control of an Irrigation System Using a Distributed Wireless Sensor Network,” *IEEE Trans. Instrum. Meas.*, **57**(7), pp. 1379–1387.
[4] Choi S H, Kim B K, Park J, Kang C H, and Eom D S, 2004 *An implementation of wireless sensor network for security system using Bluetooth*, Ieee Trans. Consum. Electron. **50**(1), pp. 236–244.

[5] Bräysy V, Hurme J, Teppo H, Korpela T, and Karjalainen M 2010 *Movement tracking of sports team players with wireless sensor network*, in *2010* Ubiquitous Positioning Indoor Navigation and Location Based Service, UPINLBS 2010.

[6] Ramson S R J and Moni D J 2017 *Wireless sensor networks based smart bin*, Comput. Electr. Eng. **64**, pp. 337–353.

[7] Simon C R, Simon C E, Basu B, and McGoldrick C 2017 Routing in wireless sensor networks for wind turbine monitoring, *Pervasive Mob. Comput.*, **39**, pp. **1–35**.

[8] Pule M, Yahya A, and Chuma J 2017 "Wireless sensor networks: A survey on monitoring water quality," *J. Appl. Res. Technol.*, **vol. 15**, no. 6, pp. 562–570.

[9] Abed A A, Ali A A, and Aslam N, 2010 *Building an HMI and demo application of WSN-based industrial control systems*, *2010 1st Int. Conf.* Energy, Power Control, **7**(2), pp. 302–306.

[10] Panda, Chinmaya and Patil R N 2016 *Design of Zigbee Based Wireless Sensor Network (WSN) and Human Machine Interface (HMI) for Industrial, Utility Air Supply Section*. International Journal of Innovative Research in Science, Engineering and Technology (**5**, Issue 11, November 2016).

[11] Tadeus, Dista Y, Yuniarto, dan Teguh Y 2017 *Design and Implementation of SCADA Training Module: Human Machine Interface (HMI) Based on Open Software.*