

# OPC Protocol Application for Real-Time Carbon Monitoring System for Industrial Environment

Mohd Faiz Rohani, Noor Azurati Ahmad, Shamsul Sahibuddin, Salwani Mohd Daud

Advanced Informatics School, Universiti Teknologi Malaysia, Malaysia

---

## Article Info

### Article history:

Received Jan 9, 2017

Revised Mar 15, 2017

Accepted Mar 30, 2017

---

### Keyword:

Carbon emission

Carbon monitoring system

CO<sub>2</sub> sensor

OPC architecture

---

## ABSTRACT

Global warming is referred to the rise in average surface temperatures on earth primarily due to the Greenhouse Gases (GHG) emissions such as Carbon Dioxide (CO<sub>2</sub>). Monitoring the emissions, either direct or indirect from the industrial processes, is important to control or to minimize their impact on the environment. Most of the existing environmental monitoring system is being designed and developed for normal environment monitoring. Hence, the aim of this project is to develop industrial CO<sub>2</sub> emission monitoring system which implements industrial Open Platform Communications (OPC) protocol in an embedded microcontroller. The software algorithm based on OPC data format has been designed and programmed into the Arduino microcontroller to interface the sensor data to any existing industrial OPC compliant Supervisory Control and Data Acquisition (SCADA) system. The system has been successfully tested in a lab with the suitable environment for real-time CO<sub>2</sub> emissions measurement. The real-time measurement data has been shown in an industrial SCADA application which indicates successful implementation of the OPC communications protocol.

*Copyright © 2017 Institute of Advanced Engineering and Science.  
All rights reserved.*

---

## Corresponding Author:

Noor Azurati Ahmad,

Advanced Informatics School,

Universiti Teknologi Malaysia,

Jalan Sultan Yahya Petra, 54100, Kuala Lumpur, Malaysia.

Email: [azurati@utm.my](mailto:azurati@utm.my)

---

## 1. INTRODUCTION

The earth climate is continuously evolving; with part of it, is caused by human activities resulting in the emissions of Greenhouse Gases (GHG) such as carbon dioxide, methane and many other gases which lead to global warming. According to the Inter-Governmental on Climate Change (IPCC), global warming is the increase of Earth's average surface temperature due to the effect of greenhouse gases such as fossil fuels or from deforestation. These gases will trap heat into the atmosphere on which one of the GHG emissions that contributed the most to the global warming is CO<sub>2</sub> with a total of 76% of total emissions. Hence, monitoring and managing carbon emissions from industrial processes are important to maintain a low level of emission. Industrial processes contributed to 21 % of GHG emissions globally [1]. Current concerns regarding the manufacturing processes are not limited to the production quality, but also include the ecological footprint. Therefore, the system is aimed to measure the eco-efficiency and environmental impact of each manufacturing processes within production systems. A real-time carbon real-time monitoring system to monitor carbon emissions released at any particular area or process is required for the purpose.

## 1.1. RELATED WORK

In this study, embedded monitoring system using a microprocessor with an environmental sensor is being investigated to monitor the carbon level of the actual environment. The system requires a microcontroller with CO<sub>2</sub> and other GHG gases sensor, with a wireless module for wireless communication.

The use of a wireless system will become more important because of wireless sensor network's compatibility for distributed data collection and monitoring in the harsh environment [2]. Previous related research work is being reviewed so that the research gap can be identified. A low-power microcontroller and wireless sensor system has been designed with low power consumption for long operating life [3]. The system measured carbon dioxide, humidity and temperature connected to a 32-bit microcontroller. It consists of two main components: data logger and data interpreter. The sensing unit components developed with a sensor, attached to Arduino microcontroller unit based on ATmega328 low-power processor. The ZigBee wireless board has been used to transmit the sensing unit sensors data to the base station. The wireless technology has been selected because it uses 2.4GHz ISM band, with direct sequence spread spectrum (DSSS) which has the benefits of resistance against intended and unintended jamming, sharing of a single channel for multiple users, and has the capability of reducing background noise level which hampers the interception. The study focuses on low-cost and low-speed communication between devices with low power consumption with receiver sensitivity of -100dBm (at 1% packet loss rate). The system only focuses on the collection of environmental data in the test field. Another work that has been reviewed developed a system using open source hardware platforms connected to a wireless module. They have developed a hardware platform using Raspberry Pi connected to a wireless module. ZigBee had been chosen as the wireless technology since it offers low power consumption, and scalable in terms of the number of nodes. It makes the system well suited for a wide variety of applications related to the environment. The system consists of a sensing unit, a base station and a number of distributed wireless sensor nodes connected using the ZigBee wireless protocol. The Raspberry Pi was used because of its low cost and low power consumption which runs on ARM11 microprocessor. It also supports a number of operating systems including Linux and the latest Windows 10 IOT version. The main controller manages several nodes of sensors. The sensor nodes consist of sensors, controller and transceiver where the main function is data collection as well as data distribution. Furthermore, the user application program on each sensor node is responsible for data handling from sensors within defined communication protocol with the base station [4]. With current related works only focus on environmental monitoring in the non-industrial area, this project will investigate low power consumption embedded system, and a scalable wireless sensor network for environmental parameters, especially CO<sub>2</sub> in an industrial environment. The system will be deployed using wireless technology such as ZigBee that has good performance in harsh and potentially damaging conditions including extreme temperatures and temperature cycles, ingress of particulates, electrostatic discharge (ESD), electromagnetic interference (EMI), vibrations, as well as physical impact [4]. The integration to the existing industrial systems through unified protocol is to support data exchange between the carbon monitoring system and the industrial Supervisory Control and Data Acquisition/Human Machine Interface (SCADA/HMI). The system has the capability to monitor real-time environment carbon emissions and interfaces to the industrial system through the communication protocol. The development of the monitoring system includes several components, namely the main controller for central processing and the wireless communication module for data communication and sensors as the hardware architecture. Meanwhile, the controller programming and the system interface are considered as the system software architecture. The monitoring system software has several important requirements. The microcontroller of sensor node must be able to manage and send the sensor data within specific of time frame. At the receiving end, the coordinator node needs to receive the data in a specific manner, able to collect the data from its subsystem and converts the data to OPC format. The use of OPC as the communication protocol between server and all SCADA-based machines has spread widely these days. In the next section, we will discuss further on the OPC architecture.

## 1.2. OPEN PLATFORM COMMUNICATIONS (OPC)

Formerly known as Object linking and embedding for Process Control, the name had been changed to Open Platform Communications in 2011. This is due to the protocol has grown beyond the original object linking and embedding implementation, as well as support in another platform other than Windows for their OPC Unified Architecture (UA) specifications. Most of the equipment have their own built-in drivers and cannot communicate to each other directly. The challenges are to standardize the interfaces between the components and the difficulty in integrating these diverse components. To solve this problem, OPC specification is developed by the industry task force to introduce a standard mechanism for communication between the various devices.

Figure 1 illustrates the conventional and OPC standard communications. As shown in Figure 1(a), conventional communication requires different drivers for different devices. While in Figure 1(b), using OPC

standard communications, the devices do not need to have different drivers to communicate to different devices, and only uses a single driver for communication. This will significantly reducing the network traffic since the applications only made requests to the OPC server which manages the data exchange without accessing the devices [5]. The application of OPC specification enables the connections to real-time equipment and processing the data acquired from the equipments's is possible. OPC is a single or unified protocol mainly used for communication in industrial process control systems which specifies the real-time plant data between control devices from different manufacturers. It eliminates the need of custom drivers for each device and application. It can interface to other industrial protocol such as Modbus, Profibus, etc. and translate to single OPC protocol that can be used by any upper layer applications. Since industrial sensors, actuators and devices can be connected using this protocol; it is possible to design a system that can retrieve equipment and processes information, perform an analysis and control industrial processes so that monitoring and controlling carbon emission is achievable. There are three layers of communication namely the device, data source and also the data sink. The data source is where the devices reside, such as programmable logic controllers, sensors, electrical motors and drivers which are located at the plant production floor or remote sites. They can be any device from any manufacturer, which uses different custom drivers or communication protocols. These devices must have a common protocol to interact with each other. OPC Server is based on client-server architecture and provides unified interfacing to devices drivers from different protocols. It provides connection and data link to the upper-level layer in plant system architecture. While data sink is the application that supports OPC architecture and acts as an OPC Client. The application makes a request to OPC server for specific data required for a certain period of time [6]. OPC has two major specifications, mainly the OPC Classic based on Windows Component Object Model (COM)/Distributed (COM) that has been implemented as early as 1996 and the newer technology OPC Unified Architecture (OPC UA) which is based on web technology. The OPC has data format such as value, timestamp as well as data quality based on the specifications. There are three major specifications namely as OPC Data Access (OPC DA), Historical Data Access (OPC HDA) and Alarm and Events (OPC A&E). OPC classic is still considered de facto for industrial standard and widely used in the industry, even though OPC UA is gaining popularity due to its flexibility [6], [7].

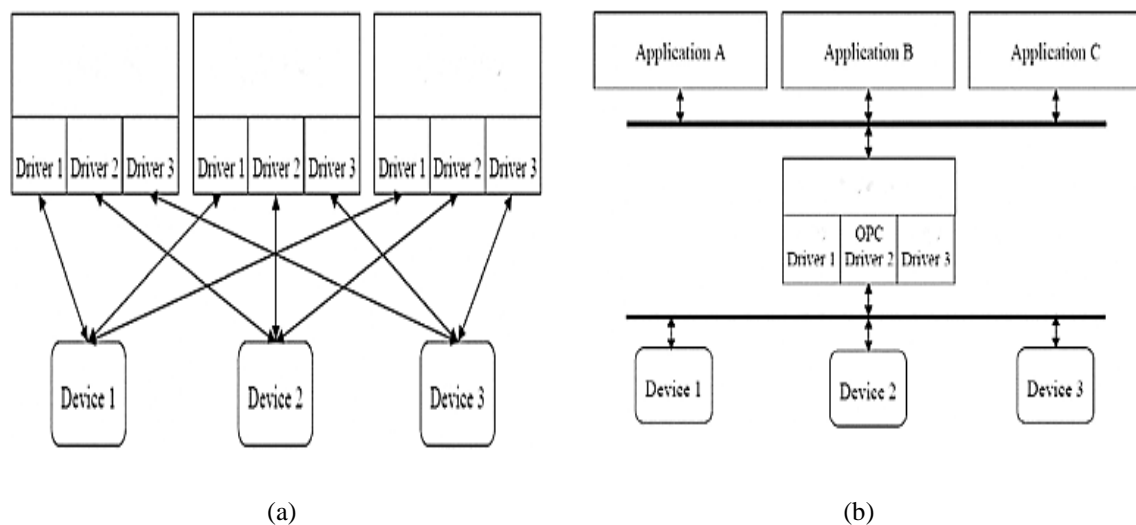


Figure 1. (a) Conventional communication and (b) OPC standard communication

## 2. SYSTEM AND SOFTWARE DESIGN

Figure 2 depicts the proposed architecture of the carbon monitoring system for industrial processes. CO<sub>2</sub> and other sensors will be interfaced to an Arduino microcontroller board to measure the respective parameters in the production floor. The setup established a connection from sensors to the microcontroller in collecting the sensors parameter data at a set interval of time. Temperature and humidity sensor, connected to the Digital input while CO<sub>2</sub> and combustible gas sensors are connected to Analog input 0 and 4 respectively. ZigBee wireless communication will be used as a medium to transfer the data to the main microcontroller. The main node Arduino board connected to the main computer through a USB-Serial Port serial communication. Software algorithm of the sensing and main node is shown in the following Figure 3.

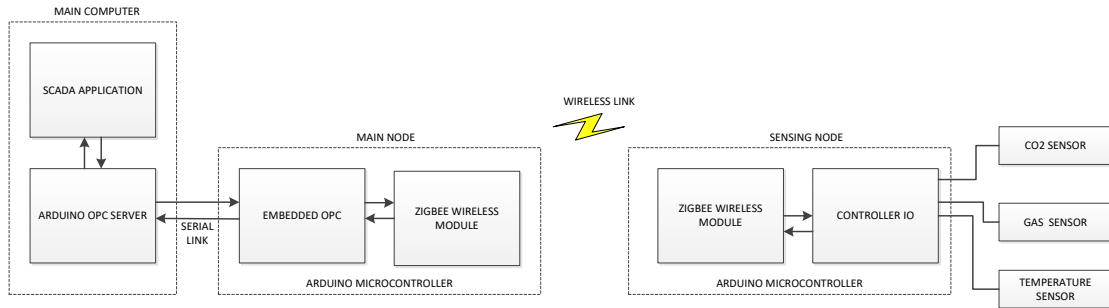


Figure 2. Proposed system architecture

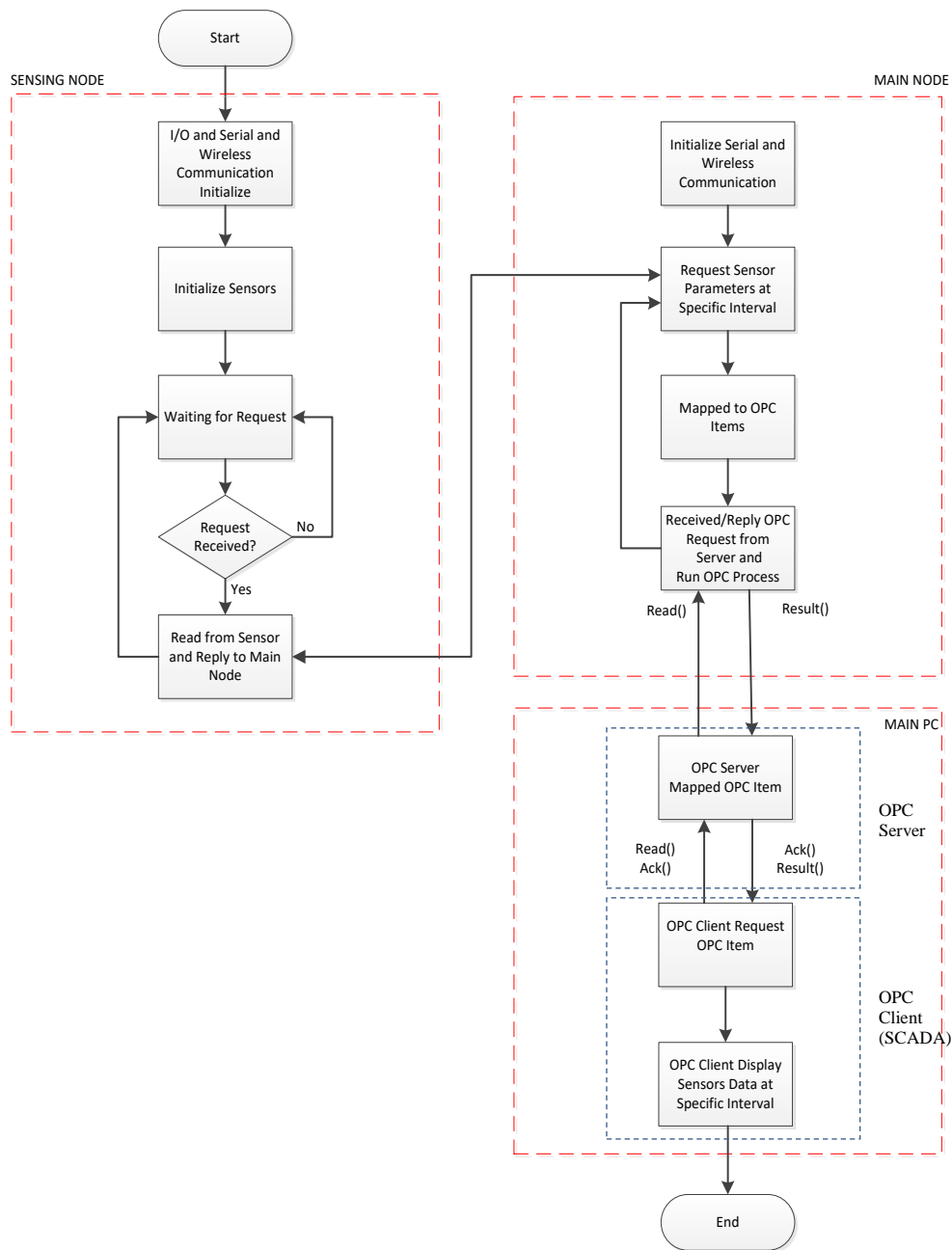
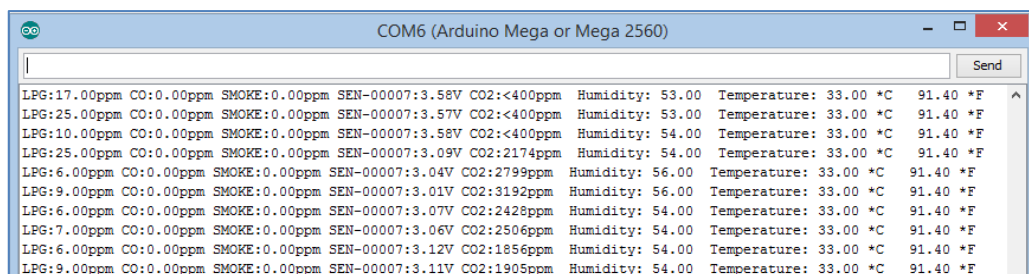


Figure 3. Software algorithm of real-time carbon monitoring system

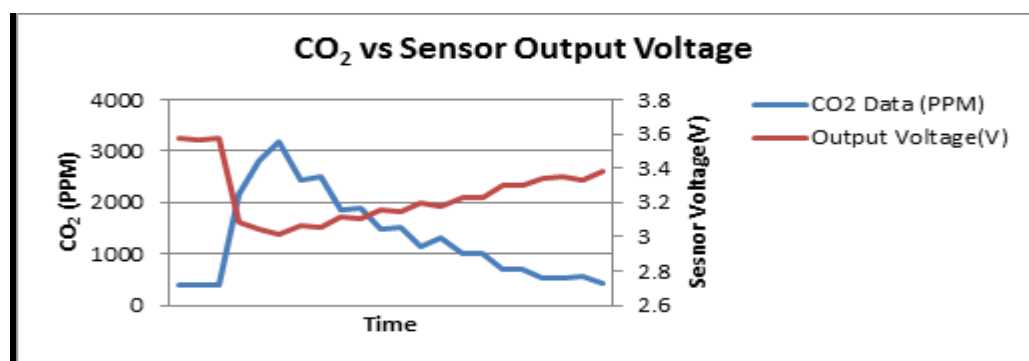
An open source Arduino OPC Server in the main computer mapped the received data (in OPC format) and act as an interface to the OPC Client [8]. The OPC compliant systems such as SCADA system will be used to display the sensing data to users. Following the standard OPC Data Access (DA) Protocol, a client will establish a connection to a server by creating *OPCServer* Object which is considered as a top-level object of the hierarchy. The client will group the OPC Items with the identical setting in an *OPCGroup* Object. Several values are read simultaneously with a single call via methods provided by the *OPCGroup* Object. By using asynchronous polling, the client will send a read request, and once received by the OPC Server, it will reply with an acknowledgement (ACK) and the result. The client will acknowledge the received data with a return acknowledgement to the server [6]. When the cycle starts, the initialization of serial communication as well as the sensors (I/O) initialization will be started. The controller will wait for a request to read the data from sensors which has been set at a 1000ms (1 sec) interval. The collected data will be added to OPC Items (by registering the data to OPC format) and this process will continue with new updated real time data. The connection and read request are made from the OPC server to the main controller, based on the communication initiated by the OPC Client to the Arduino OPC Server. The link between sensors and controllers as well as the OPC format data will be tested in the unit or components testing. Following, Wireless link and complete system test will be performed to ensure data acquired by the sensors can be transmitted and received by the main node controller at the other end.

### 3. EXPERIMENTAL RESULTS AND DISCUSSION

The experimental works began with a unit test in a lab environment. The sensors were connected to the Arduino board with a test algorithm interfaced to the computer using Serial communications. This was performed to ensure that the controller hardware and the sensors are working before being tested in the field. The results as shown in Figure 4(a) showed that the controller received the sensors data in text form which is recorded by the serial monitoring software available in Arduino Sketch software. The sensor is highly sensitive to CO<sub>2</sub> and the module output voltage falls as the concentration of the CO<sub>2</sub> increases as shown in Figure 4(b). From the result, the CO<sub>2</sub> is 400PPM (Parts per Million) which is healthier when it is lower than the standard of 600PPM described by the US National Institute for Occupational Safety and Health (NIOSH) [9]. Other gas parameters were set lower as being expected in normal condition while temperature and humidity level reading are considered acceptable.



(a)



(b)

Figure 4. (a) (top) sensor parameters data in normal text form and (b) (bottom) co<sub>2</sub> vs sensor output voltage data over time

With the successful testing; further testing is performed to test the sensor parameters in an OPC format data. The software algorithm for the normal test had been updated so that the received data will be converted to OPC data. The received data had been visualized using OPC Expert (OPC Client) by the OPC Training Institute (OPCTI); where OPCTI is a supporting member of the OPC Foundation [10] as shown in Figure 5 below: The interface shows a list of the available and connected server at the left top and a list of registered OPC items on the right. OPC items value will be updated at a 100ms rate which was the default software configuration. For the Other related information is shown in the bottom window, while the connection is still active.

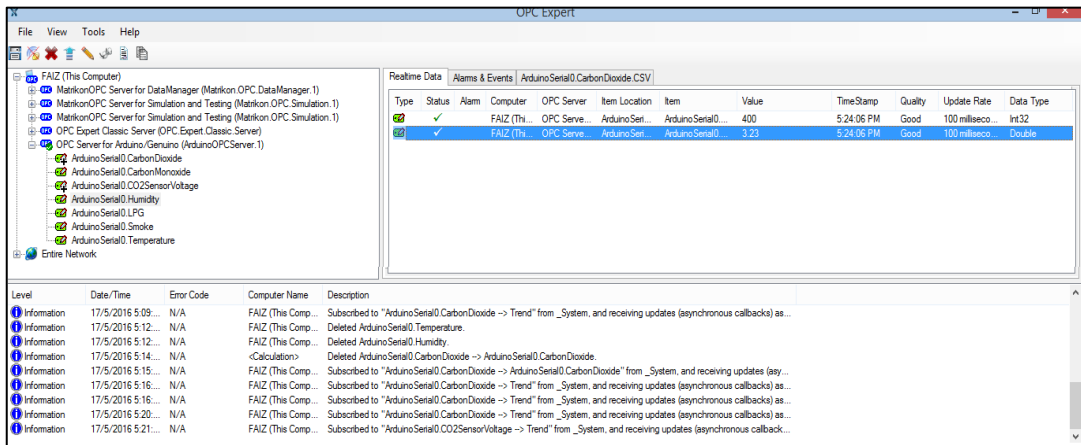


Figure 5. CO<sub>2</sub> data in OPC format

The data has been further visualized in data trending chart as showed in Figure 6(a) and being recorded in CSV file (Figure 6(b)) which will allow data tracking over the course of testing. During normal ambience, the result showed that the sensor detection is around 400PPM while exhaling towards the sensor will cause the sensor to fluctuate above the level at a peak of 1880PPM as seen in Figure 6(b).

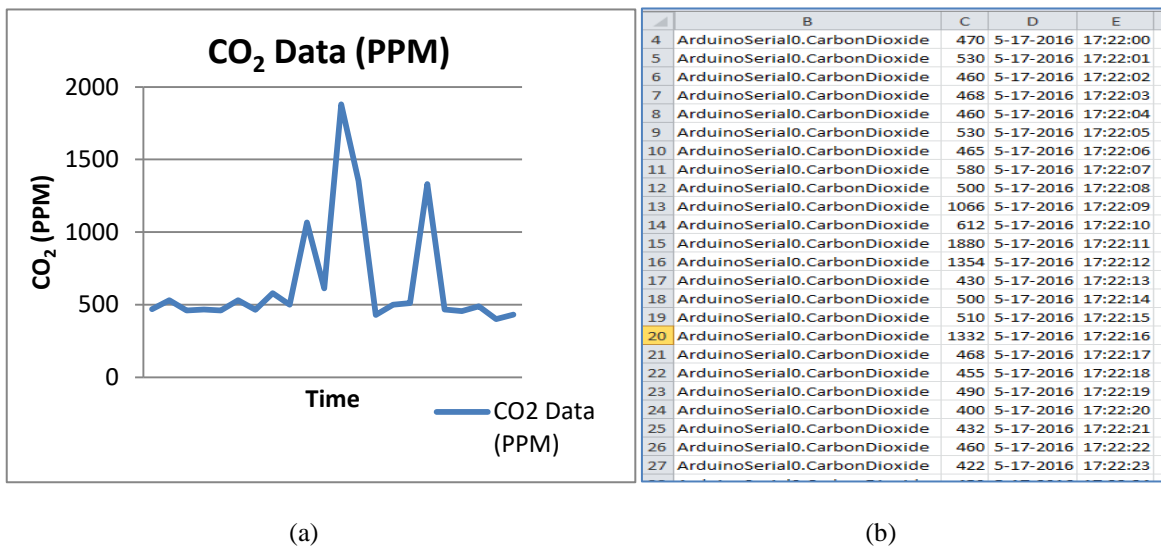


Figure 6 (a) CO<sub>2</sub> data trending and (b) CO<sub>2</sub> recorded data in CSV format

Subsequent to the unit test, a complete system with software algorithm as shown in Figure 3 has been run in the Vehicle Diesel Engine Testing Laboratory in Malaysia-Japan International Institute of Technology (MJIT), Universiti Teknologi Malaysia (UTM) Kuala Lumpur which is a suitable testing



environment to measure CO<sub>2</sub> emissions. The lab, as shown in Figure 7(a) below is used for testing diesel engine performance, and suitable for CO<sub>2</sub> testing since the engine combustion will release a significant amount of carbon emissions. The testing has been done using the particular setup as shown in Figure 7(b). The system sensing node will wait for the command from the main node for the data acquiring process. Once received, the controller will read the required data and send them to the main node for data conversion to OPC format. The Sensing node has been placed nearby the engine exhaust which will be opened to release the CO<sub>2</sub> inside the laboratory instead of channelling it out of the room. This is to measure the amount of CO<sub>2</sub> that will be released during the test. While in the monitoring area, the main node is setup approximately 10 metres apart from the sensing node.

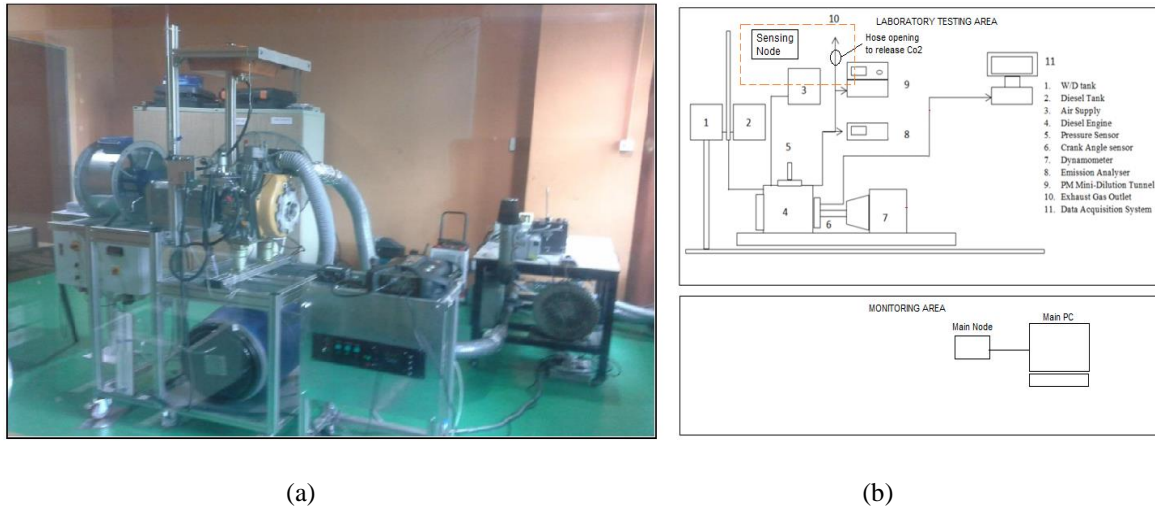


Figure 7. (a) Vehicle Engine Testing Lab and (b) Test Setup

From the testing, measurement of the CO<sub>2</sub> parameter has been successfully acquired as shown in an Industrial SCADA system for the data display to the user. For this particular test, data received will be monitor by OPC explorer application (OPC Server Side) with data showed by the HMI software to ensure data being received correctly. This indicates that the system had been integrated successfully with the existing SCADA system where the CO<sub>2</sub> data was displayed in the monitoring system. Furthermore, the alarm was configured to the system, which will be triggered if the level of CO<sub>2</sub> is higher than the configured level 1000.

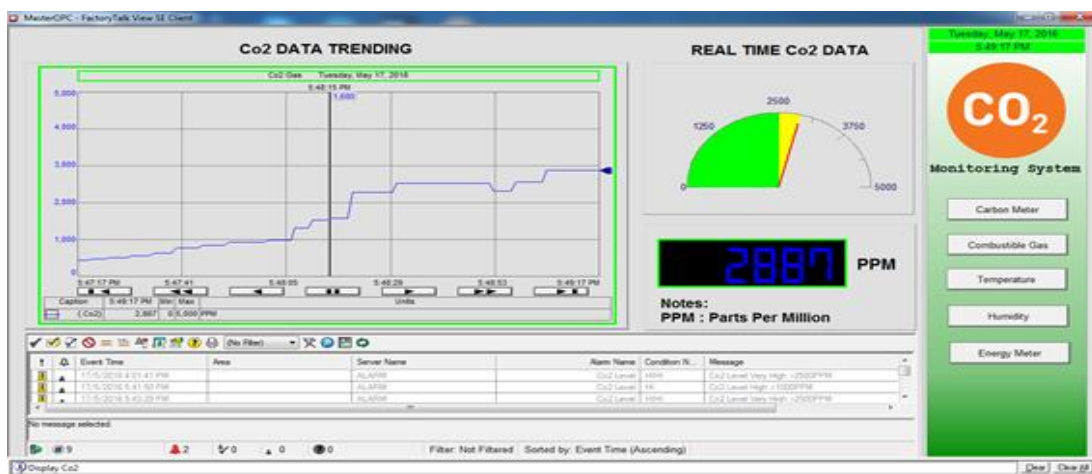


Figure 8. User Interface for Real-Time Carbon Emission Monitoring System using OPC Compliant SCADA Application PPM (High) and 2500PPM (High High) as shown in the following Figure 8.

#### 4. CONCLUSION

Integration with the existing industrial system through Open Platform Communications (OPC) protocol to support data interchange between the carbon monitoring system and industrial SCADA/HMI had been successfully developed and tested. An Arduino microcontroller with embedded programmed OPC code is used and interfaced with CO<sub>2</sub> and combustible gas sensors for data feeding to OPC Compliant system or interfaces. In future, we will consider testing in the real environment of production in the industrial area and connect all sensors to each industrial process via wireless technology. The OPC communication data traffic will be investigated closely to ensure the data is received without any data lost and corrupted. Furthermore, Cyclic Redundancy Code (CRC) algorithm will be used to detect any accidental changes of data transferred from sensing to the main node microcontroller. A web-based OPC Unified Architecture (OPC UA) architecture will be considered to support IOT technology data transfer to the Internet.

#### ACKNOWLEDGMENTS

The work has been supported by the Encouragement Grant, Vote No. (11J37). Hence, the authors would like to thank the Advanced Informatics School, Universiti Teknologi Malaysia (UTM) for supporting the research work.

#### REFERENCES

- [1] "IPCC Fourth Assessment Report: Climate Change 2007," Intergovernmental Panel on Climate Change (IPCC), Retrieve on 11 April 2016, [http://www.ipcc.ch/publications\\_and\\_data/ar4/syr/en/contents.html](http://www.ipcc.ch/publications_and_data/ar4/syr/en/contents.html)
- [2] D. G. Romeo, *et al.*, "An electronic interface for measuring CO<sub>2</sub> emissions in embedded systems," in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, pp. 417-420, 2012.
- [3] A. Abdullah, *et al.*, "Development of wireless sensor network for monitoring global warming," in *Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference on*, pp. 107-111, 2012.
- [4] S. G. Nikhade, "Wireless sensor network system using Raspberry Pi and zigbee for environmental monitoring applications," in *Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on*, pp. 376-381, 2015.
- [5] S. Mai, *et al.*, "An OPC UA client development for monitoring and control applications," in *Strategic Technology (IFOST), 2011 6th International Forum on*, vol.2, pp.700-705, 2011
- [6] M. H. Schwarz and J. Borcsok, "A survey on OPC and OPC-UA: About the standard, developments and investigations," in *Information, Communication and Automation Technologies (ICAT), 2013 XXIV International Symposium on*, pp. 1-6, 2013.
- [7] Y. Yao, *et al.*, "The Remote Monitoring System Based on the OPC Technology," in *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*, pp. 1-3, 2009
- [8] Arduino OPC Server, "Software Tools for Makers," Retrieve on 15 April 2016, <http://www.st4makers.com/>
- [9] Indoor Air Quality Investigation, "Occupational Safety & Health Administration," United States Department of Labour, Retrieve on 30 April 2016, [https://www.osha.gov/dts/osta/otm/otm\\_iii/otm\\_iii\\_2.html](https://www.osha.gov/dts/osta/otm/otm_iii/otm_iii_2.html)
- [10] OPC Expert, "The OPC Training Institute (OPCTI)," Retrieve on 3 May 2016, <https://opcexpert.com/>

#### BIOGRAPHIES OF AUTHORS



**Mohd Faiz** is a software and system engineer and currently a PHD Candidate at Advanced Informatics School, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia. He received his B.CS. (Software Engineering) in 2014 and his M.SC.(Computer System Engineering) in 2016 from the same university. His research interests include embedded systems, system engineering, Open Platform Communications (OPC) protocol, industrial communications protocol such as Modbus and embedded software architecture.





**Dr. Noor Azurati binti Ahmad** serves as a senior lecturer at Advanced Informatics School (UTM AIS), Universiti Teknologi Malaysia Kuala Lumpur. She obtained her B.Eng. in Computer Engineering in 2001 and Master of Electrical Engineering in 2006 from Universiti Teknologi Malaysia. She graduated with a PhD in Embedded Systems from University of Leicester in 2013. She is a Certified Tester Foundation Level (CTFL) under Malaysian Software Testing Board (MSTB) and Certified Professional for Requirements Engineering (CPRE) under International Requirements Engineering Board (IREB). She has also served as an engineer in Sapura Secured Technologies Sdn. Bhd under the Network Centric Operations (NCO) project. She involved in SIRIM qualification test for Sapura's military products. She is a member of Institute Electrical Electronic Engineer (IEEE), IEEE Computer Society and Registered Graduate Engineer with the Board of Engineers Malaysia (BEM). She has been actively involved in research related to design, software architecture and testing for embedded real-time systems and mobile and pervasive computing.



**Shamsul Sahibuddin** is a Professor of Software Engineering at Universiti Teknologi Malaysia (UTM). He is currently the Dean of Advanced Informatics School in UTM. He graduated from Aston University, United Kingdom with Ph.D. in Computer Science. He also holds masters and bachelor degree in computer science. His research interest is in software and system engineering, software quality, software process improvement, secure software development, requirement engineering, and information technology service management. He has twenty nine years experiences in software development and lecturing. He also is part of the Asia Pacific Software Engineering Conference Steering Committee since 2009. He has been the Chairman of the Technical Committee 11 on Software Engineering Standard at SIRIM, Malaysia since 2000.