Proceeding ICMETA: Volume 1/2017, June 27th 2017 ISBN 978-602-397-058-2

Gas measurement system

Haryono, Dian Yudha R
Balai Pengamatan Atmosfer dan Antariksa - LAPAN
Pasuruan, East Java, Indonesia

E-mail: haryono@lapan.go.id, dian_yudha@yahoo.com

ABSTRACT. Many industries are established, large-scale waste combustion and increasing of motor vehicles. They are effecting in increasing air pollution. It is required an effort to control the air pollution which is increasing day to day. To control the air pollution cannot be done without an existing of accurate data. The data measurement can be used by a stakeholder in order to get quick decisions. There are several tools to take measurements of gas. The tools are generally placed at a certain point that requires a good infrastructure when it would like to take measurements in different locations requires a process which is not simple. Sometimes data is not ready to be accessed because still be kept in the sensor device need to take the data manually. It is required to measure the gas which can be moved in other location easily and easy to be accessed. Therefore this research is conducted: how to measure gas which has high mobility, compact, accessible and near real time. The sensor is placed in the certain area, observational data and the location data of the sensor will be sent to the server as fast as possible. From the data obtained can be accessed from the website, make it easy to be accessed in anywhere at anytime. The system that has been made showed: measurement of gas can be done easily, the device to sensor the gas is relatively small and compact, it can be moved easily and the data is accessible via the website.

Keywords: Gas Sensor, Arduino, Android, Bluetooth, Web server.

1. Introduction

Many air pollution measurement systems are available in the market, but most of them are not designed in a small size. It is because they would like to cover a requirement user as many as possible so that many components are placed in the measurement system. Such user has not even used some features which are available on some devices. Therefore to make it movable is not easily. It is not being designed to be placed in such location where limitations in term of size, weight and power consumption are there. Another issue is about difficulting in accessing the data, the data sometimes is not up to date because the device is not designed to upload to the server automatically.

Many types of research and speaking about Air Pollution Measurement System in the term theoretically have been done. Such as in [1] talks the overview about Air Pollution Measurement System. In specifically also has been talked about measuring of the Particulate Pollution [2] and [3]. Comparability of Air Pollution Measurements is also has been done in the research [4]. A city-wide mobile monitoring campaign was combined with passive samplers [5].

Unlike mentioned research early, this research is developing to the measurement system to achieve in the acquition data from sensors and serve to the user via web browser. All are done automatically, the data is ready in front of a computer that can be accessed from anywhere at anytime.

 ISBN 978-602-397-058-2

It is required to develop the measurement system which is lighter, movable, compact, near real time, can be placed in a certain location easily, accessible and less power consumption. Therefore to overcome above issues this research is conducted.

2. Method

The method to achieve the objective is:

1. Understanding about the air pollution.
2. Finding small and light sensors for gas pollution such as CO.
3. Finding the devices that will be used to gather the data from the sensor.
4. Finding the communication way which can support the communication between the sensor system and server.s
5. Design and Implement: Design the hardware and software, then doing software development for the measurement system.
6. Testing

3. System design

System design of the measurement system is shown in Figure 1. It shows the flow of the data from sensor until the browser. Gas Sensors are CO, Temperature, Humidity, and Pressure. The data will be sent to the Onboard Computer (OBC) to be processed. After the data is processed and packaged and ready to be sent, the OBC will send the data to Android Device. After that, the Android Device will send the data via the internet to the server. The OBC will handle how to package all data from the sensor and send to Android Device. Web server is Windows based. The web server receives the data from Android Device via the internet and will be saved to the database. GPS data and Gas Sensor Data will be processed and ready to be requested by the browser.

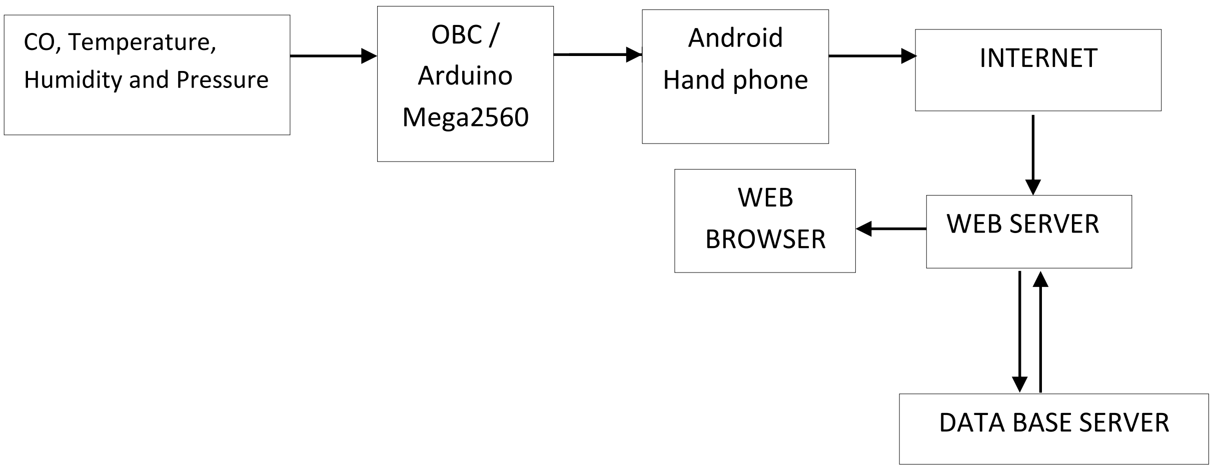


Figure 1. System Design

3.1. Gas Sensor

In the initial development the CO sensor, Temperature Sensor, Humidity Sensor, Air Pressure are included, another gas sensor will be done for the next development. There are many CO Sensor in the Market, such as MQ7, ZE07-CO, DT-Sense carbon monoxide sensor etc.

The MQ7 component is Carbon Monoxide Sensor component that very small and light which requires 5 V DC. As shown in Figure 2, how small it is compared to the finger. The component needs 2 power sources, one is for the component and the other one is for the heater of the sensor. The heater of the sensor can be supplied power in one cycle or two cycles, to get more sensitive sensor two cycles

Proceeding ICMETA: Volume 1/2017, June 27th 2017 ISBN 978-602-397-058-2

is recommended. The two cycles are by supplying a power to the heater at 5 V for 60 seconds and then 1.4Volt for 90 seconds. The component schema can be seen in Figure 3.

There is a board that already available in the market that can generate the data from the sensor via RS 232 output. The name is DT-SENSE Carbon Monoxide Sensor. It makes the jobs is more simple because the output can be directly connected to the PC or Onboard computer via RS 232 port. Figure 3 shows the DT-Sense Carbon Monoxide Sensor module, it contains the MQ7 co sensor, the AT Mega Microcontroller, and resistor conditioner. But with this module we need to convert it to PPM, there is other Jobs need to do with this kind of module, like calibration, as the different temperature is having an influence on this measurement.

To make it development fast, ZE07-CO is used. This module is more expensive compared to the two devices above but it gives slightly simple to get the value of the Carbon Monoxide. It outputs the UART as a digital and needs to convert to PPM with the formula that is given. Figure 4 is the ZE07-CO sensor.

For the Humidity and Temperature Sensor DHT 11 component is used, whereas for the pressure sensor, BMP180 is used. Figure 5 is the DHT11 and Figure 6 is a BMP180 sensor . There is Arduino library that can use directly to calculate the Temperature and Humidity [7] and to calculate the pressure [8].

Figure 2. Carbon Monoxide Sensor.

Figure 3. DT-Sense Carbon Monoxide Sensor [6]

Figure 4. the ZE07-CO sensor

Figure 5. the DHT 11 sensor

Figure 6. the BMP180 sensor

3.2. On Board Computer (OBC)

Many OBCs are available in the market, it needs to select properly according to the requirement. The requirement of the system should be less power consumption and as light as possible.

Raspberry can handle the communication via Internet easily as Operating System can be installed on the board. Raspberry also has serial communication that can be used and program easily. Figure 7 is the Raspberry that is selected. The type of Raspberry that is selected is Raspberry Pi 2 Model B, it is because can be installed the Windows IOT, it is expected can more easy in the programming of the device. Raspberry Pi 2 Model B contains 900MHz quad-core ARM Cortex-A7 CPU with 1GB RAM. Like the (Pi 1) Model B+, it also has 4 USB ports, 40 GPIO pins, Full HDMI port, Ethernet port, Micro SD card slot etc.

The other small factor of onboard computer is Arduino Pro Mini. It is using ATMega328 microcontroller. Table 2 is the detail specification of the Arduino Pro Mini and Figure 8 is the board of Arduino.

Proceeding ICMETA: Volume 1/2017, June 27th 2017 ISBN 978-602-397-058-2

Table 2. Arduino Pro Mini specification [12]

Microcontroller	ATmega328 *
Board Power Supply	3.35 -12 V (3.3V model) or 5 - 12 V (5V model)
Circuit Operating Voltage	3.3V or 5V (depending on model)
Digital I/O Pins	14
PWM Pins	6
UART	1
SPI	1
I2C	1
Analog Input Pins	6
External Interrupts	2
DC Current per I/O Pin	40 mA
Flash Memory	32KB of which 2 KB used by bootloader *

Another option for the On Board Computer is Arduino Mega 2560, figure 9. This board is more expensive than Arduino Pro Mini but has many features. The most important for this scenario is having 4 UARTS that very important in this scenario because it needs to handle many sensors to get the data. Table 3 is the detail specification of the Arduino Mega 2560 and Figure 10 is the board of Arduino Mega 2560.

Table 3 Specification of the Arduino Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

In order to achieve the scenario of this system, Arduino Mega 2560 is selected. It will work together with the Android Handphone to send the data to the Web Server. Android Headphone will be used to send the data to Server as in the Android Handphone is available 3G connection. Beside that the Data will be displayed in the Device so that using Handphone will be easy to display.

3.4. Android Device

Android Device is used to send the data to the server via the internet. Android LG 20 Handphone as it is small, light and cheaper. It is selected to make an essay in sending the data to the server because it has the 3G device. Besides, it has GPS inside the device. Using this device can show the data to the display, so that when monitoring to the location point is easy. Figure 10 is the Android Handphone that is used.

Proceeding ICMETA: Volume 1/2017, June 27th 2017 ISBN 978-602-397-058-2

3.5. Bluetooth HC06

In order to send the data from Arduino board to Android device, Bluetooth HC06 is used. It is showed in figure 11. It is relatively small. HC 06 has Tx, Rx pin that can be connected to the Microcontroller to send or receive data.

Figure 7. Raspberry Pi 2 Model B [9].

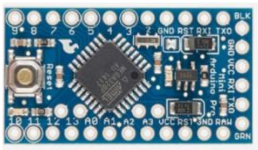


Figure 8. Arduino Pro Mini

Figure 9. Arduino Mega 2560

Figure 10. Android Handphone LG 20

Figure 11. Bluetooth HC06

4. Implementation

The devices need to be programmed is divided into three categories they are:

1. OBC: Arduino Mega 2560
2. Android Device
3. Web API and Web Application

The tool to program above devices is using Visual Studio 2015. implementation in detail will be talked separately in other paper.

4.1. Programming the Arduino Mega 2560

Arduino is based on Atmel microcontroller, it can be programmed using Atmel studio, Arduino software, or Visual Studio 2015. Visual Studio 2015 is selected because the writer has experience with this tool and easy to use. Visual Studio opens quickly and provides high-speed intellisense [12].

It works together with Arduino IDE for Visual Studio. Arduino IDE for Visual Studio is freeware, fully compatible (unless to break the rules) Arduino development environment with optional debugger/pro pack [13]. In Figure 12 shows the interface configuration to send the compiled program to Arduino successfully. It needs to select the Arduino board that is used, at here is Atmega2560 and COM6 are the port that is used to connect to the Arduino. It can be checked in device manager in windows in order to know the port being used. The structure code in Visual Studio 2015 is exactly when programming in Arduino IDE.

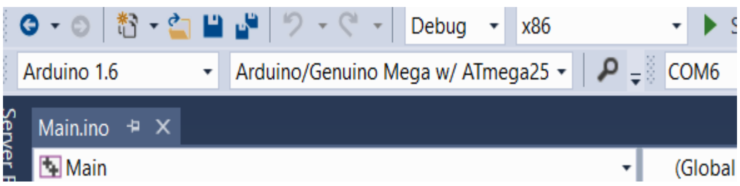


Figure 12. The interface configuration tool to connect the Arduino Mega 2560 using Visual Studio 2015

4.2. Programming the Android Handphone

There are many ways to program the Handphone Android. Most popular language is Java or C#. The IDE tool can use Eclipse, Android Studio or Visual Studio. Eclipse and Android Studio IDE use Java to program the android, whereas Visual Studio using C#. For the .Net programmer, it is easier to program using Visual Studio. The tools and language have advantage and disadvantage. Visual Studio works together with Xamarin in order to program the Android Handphone. Using Visual studio and Xamarin programming with C# and XAML can be implemented. As many people like programming this two tools because many features can be offered. C# and XAML offered binding with MVVM pattern. With this such tool, programming can be faster and enjoy. Therefore Visual Studio is selected.

4.3. Programming the Web API andWeb Application

In order to broadcast or share the data to the global easily, Web Application is created. There is two part of implementing this system, first, is creating the web service and second web application. Web API will handle in receiving the data from android via internet and Web Application will serve to the user in giving the information. The tool that is used Visual Studio 2015 with latest technology ASP MVC and web API services. When creating the project ASP.NET Web Application is selected. It is to create the web API services and Web application using ASP MVC framework. Figure 13 is the first interface to create the Web application using ASP MVC framework.

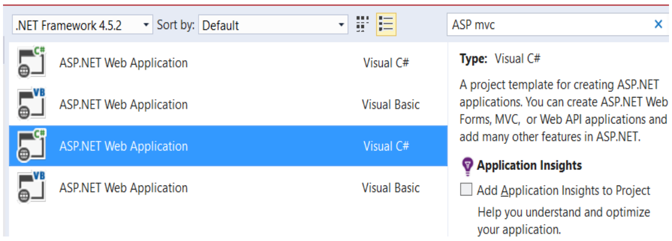


Figure 13. First interface to create the Web application using ASP MVC framework in Visual Studio 2015.

5. Testing and analysis

The scenario to test the system is list below:

1. Testing collecting data from sensor in OBC: CO sensor, Temperature, humidity, and Pressure
2. Testing communication data between OBC and Android Handphone.
3. Testing in sending data from Android to the Database Server
4. Showing the data in the browser

The emphasis at here is testing the whole system whether work as expected or not. About calibration is not talked here. The calibration will be planned in the next research. This research is to concentrate on the creation of the whole system so that when the data is available from the gas sensor can be read easily.

5.1. Testing collecting data from a sensor in OBC.

There four sensors which have been acquired, they are CO sensor, Temperature, humidity, and Pressure. This has not done any calibration yet because of limitation of the devices. In the future to be expected to have a lab that supports in doing the calibration. What has been done is by comparing the data to the existing simple device. For example for CO sensor has been compared to the Portable CO detection. The result that has been gotten is relatively same. The whole result is shown in the Serial window communication in Figure 14.

5.2. Testing communication data between OBC and Handphone Android.

The format data is shown below:

```
#|0.50|48.00|27.10|1008.56|
```

Proceeding ICMETA: Volume 1/2017, June 27th 2017 ISBN 978-602-397-058-2

It is separated by "|" delimiter, first is data CO, second is humidity, third is temperature, the fourth is pressure. The data from OBC is sent via Bluetooth and received by android handphone. The data will be extracted by the android handphone and showing to the android display. Figure 15 is the result of communication between OBC and android handphone. As it is shown the data row is displayed in the very above, and followed by the extracted data (CO, Humidity, Temperature, and Pressure). The data is handled in the class and ready to be sent to the Server.

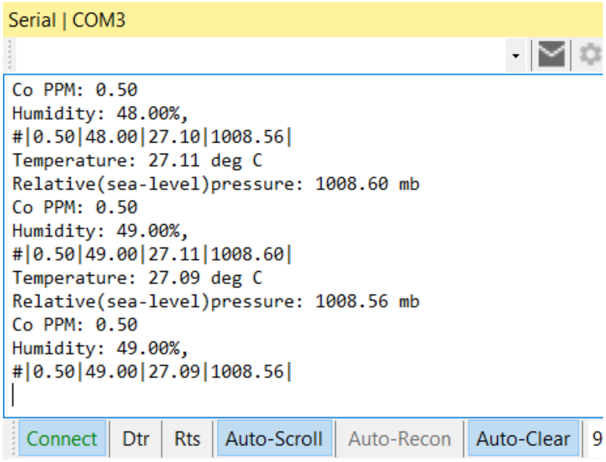
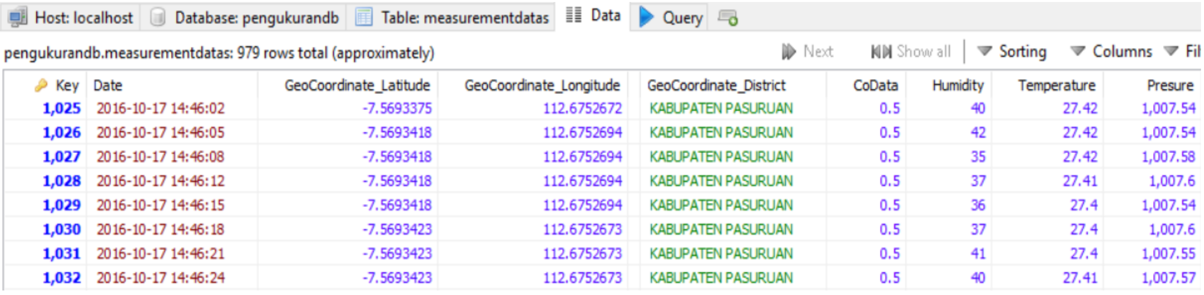


Figure 14. Data sensor CO, Humidity, Temperature and Pressure

Figure 15. Communication results between OBC and android handphone

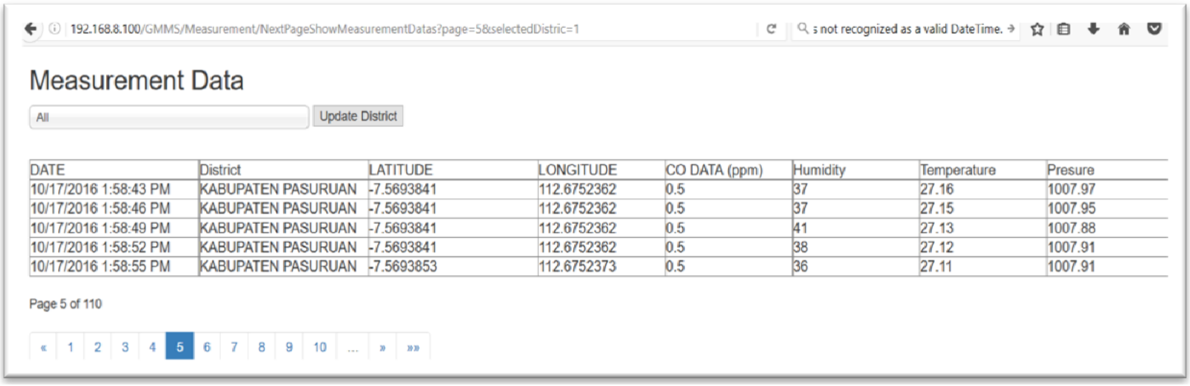
5.3. Testing in sending data from Android to the Database Server.
In this test scenario would like to know whether the data can be sent to the server via Web Service and can be saved to the database or not. The row data have been sent to the server for more than 1000 times, and the data is continuously saved to database successfully. Figure 16. is the data rows that have been save to the database. The web service also detects automatically the district based on the longitude and latitude which has been received, hence the row can be certificated based on the district. From the data that have been received can be analyzed that the fastest data that can be saved to the server from Sensor device is about 3 seconds. The most delay occurs when reading location data and when sending the data to the server.

5.4. Showing the data in the browser
Finally, the test is to know whether the data can be shown to the browser or not. As figure 17. Showed the data being displayed in the browser. It just read the data from the database and shows it in the browser. The code using technology ASP MVC has done successfully.



Key	Date	GeoCoordinate_Latitude	GeoCoordinate_Longitude	GeoCoordinate_District	CoData	Humidity	Temperature	Presure
1,025	2016-10-17 14:46:02	-7.5693375	112.6752672	KABUPATEN PASURUAN	0.5	40	27.42	1,007.54
1,026	2016-10-17 14:46:05	-7.5693418	112.6752694	KABUPATEN PASURUAN	0.5	42	27.42	1,007.54
1,027	2016-10-17 14:46:08	-7.5693418	112.6752694	KABUPATEN PASURUAN	0.5	35	27.42	1,007.58
1,028	2016-10-17 14:46:12	-7.5693418	112.6752694	KABUPATEN PASURUAN	0.5	37	27.41	1,007.6
1,029	2016-10-17 14:46:15	-7.5693418	112.6752694	KABUPATEN PASURUAN	0.5	36	27.4	1,007.54
1,030	2016-10-17 14:46:18	-7.5693423	112.6752673	KABUPATEN PASURUAN	0.5	37	27.4	1,007.6
1,031	2016-10-17 14:46:21	-7.5693423	112.6752673	KABUPATEN PASURUAN	0.5	41	27.4	1,007.55
1,032	2016-10-17 14:46:24	-7.5693423	112.6752673	KABUPATEN PASURUAN	0.5	40	27.41	1,007.57

Figure 16. Data rows that have been save to the database



DATE	District	LATITUDE	LONGITUDE	CO DATA (ppm)	Humidity	Temperature	Presure
10/17/2016 1:58:43 PM	KABUPATEN PASURUAN	-7.5693841	112.6752362	0.5	37	27.16	1007.97
10/17/2016 1:58:46 PM	KABUPATEN PASURUAN	-7.5693841	112.6752362	0.5	37	27.15	1007.95
10/17/2016 1:58:49 PM	KABUPATEN PASURUAN	-7.5693841	112.6752362	0.5	41	27.13	1007.88
10/17/2016 1:58:52 PM	KABUPATEN PASURUAN	-7.5693841	112.6752362	0.5	38	27.12	1007.91
10/17/2016 1:58:55 PM	KABUPATEN PASURUAN	-7.5693853	112.6752373	0.5	36	27.11	1007.91

Figure 17. Displaying data in the web browser

6. Conclusion and future work

The conclusion that can be obtained is the system has been made successfully. The data from Sensor is packaged by OBC and send to Server successfully, all data is accessible, it can be accessed at any time and anywhere. The measurement gas can be done near real time, movable and the sensor gas is relatively small and compact. Future work would be adding the other important gas sensor and package in to one PCB to make it as small as possible.

Acknowledgments

This project is supported by Balai Pengamatan Atmosfer dan Antariksa - LAPAN, Mr. Dian Yudha Risdianto, ST., M.T as Head of Balai Pengamatan Atmosfer dan Antariksa LAPAN, and my colleges Mr. Noi, Mr. Imron and Mr. Joyo. We would like to acknowledge for their support in this project.

References

[1] P. K. Yennawar, "Air Pollution Measurement System in India," in *Studies in Environmental Science*, vol. 2, H. W. de K. and L. J. B. T. Schneider, Ed. Elsevier, 1978, pp. 91–93.

[2] E. Muylle, D. Hachez, and G. Verduyn, "An Evaluation of Measuring Methods for Particulate Matter," in *Studies in Environmental Science*, vol. 2, H. W. de K. and L. J. B. T. Schneider, Ed. Elsevier, 1978, pp. 113–125.

[3] S. Jones, N. Richardson, M. Bennett, and S. R. Hoon, "The application of magnetic measurements for the characterization of atmospheric particulate pollution within the airport environment," *Science of The Total Environment*, vol. 502, pp. 385–390, Jan. 2015.

[4] H. Hatano, "Comparability of Air Pollution Measurements and the use of Reference Methods," in *Studies in Environmental Science*, vol. 2, H. W. de K. and L. J. B. T. Schneider, Ed. Elsevier, 1978, pp. 31–39.

[5] E. A. Riley *et al.*, "Correlations between short-term mobile monitoring and long-term passive sampler measurements of traffic-related air pollution," *Atmospheric Environment*, vol. 132, pp. 229–239, May 2016.
[6] "DT-Sense Carbon Monoxide Sensor," *Digiware Store*. [Online]. Available: http://digiwarestore.com/en/gas/dt-sense-carbon-monoxide-sensor-991921.html. [Accessed: 20-Sep-2016].
[7] "dht11.pdf." [Online]. Available: http://www.micropik.com/PDF/dht11.pdf. [Accessed: 07-Oct-2016].
[8] "BMP180 Barometric Pressure Sensor Hookup - learn.sparkfun.com." [Online]. Available: https://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup-/installing-the-arduino-library. [Accessed: 20-Sep-2016].
[9] "Raspberry Pi 2 Model B," *Raspberry Pi*. .
[10] "Introducing the Intel® Compute Stick," *Intel*. [Online]. Available: http://www.intel.com/content/www/us/en/compute-stick/intel-compute-stick.html. [Accessed: 04-Apr-2016].
[11] "Intel's Compute Stick gets bigger -- and better -- for its second generation," *CNET*. [Online]. Available: http://www.cnet.com/products/intel-compute-stick-2016/. [Accessed: 04-Apr-2016].
[12] "Visual Studio or Atmel Studio? - Arduino IDE for Visual Studio and Atmel Studio." [Online]. Available: http://www.visualmicro.com/page/User-Guide.aspx?doc=Getting-started-which-IDE.html. [Accessed: 11-Oct-2016].
[13] "Arduino IDE for Visual Studio extension." [Online]. Available: https://visualstudiogallery.msdn.microsoft.com/069a905d-387d-4415-bc37-665a5ac9caba. [Accessed: 11-Oct-2016].
[14] "PrismLibrary/Prism," *GitHub*. [Online]. Available: https://github.com/PrismLibrary/Prism. [Accessed: 11-Oct-2016].