



Multiplexer Performance Testing For IoT-Based Air Quality Monitoring System

Bakti Dwi Waluyo¹, Harvei Desmon Hutahaean², Agus Junaidi³

^{1, 2, 3}Department of Electrical Engineering, Universitas Negeri Medan, Medan, Indonesia

E-mail: bakti_dw@unimed.ac.id, harvei.hutahaean@gmail.com, agusjunaidi@unimed.ac.id

ARTICLE INFO

ABSTRACT

Article history:

Received: 26 February 2020

Revised: 22 Maret 2020

Accepted: 01 May 2020

Keywords:

multiplexer, ESP8266, gas sensor, IoT

Technology-based on the Internet of Things (IoT) which is integrated with the NodeMCU ESP8266 device has produced intelligent equipment. The smart equipment is even able to display data in real-time through computer devices and smartphones while connected to the internet. However, NodeMCU ESP8266 has limitations on analog input channels, so sensors with analog output cannot be connected more than one. Therefore, the multiplexer in the form of IC 4051 is used to add analog channels in NodeMCU ESP8266. IC 4051 has eight analog channels which are controlled by selector pins. Sensors connected to IC 4051 are temperature (LM35), humidity (HIH-4030), gas (MQ-2 and MQ-135). NodeMCU ESP8266 receives values from sensor readings and sends to the Firebase database via internet connection. The Firebase database can be monitored via a smartphone by making its application on the MIT App Inventor. This experiment has succeeded in acquiring data from the four sensors using a multiplexer, then the data can be viewed via a smartphone.

Copyright © 2019 Jurnal Mantik.
All rights reserved.

1. Introduction

At present Internet of Things (IoT) has more attention from researchers [1], because IoT forms a system where machines equipped with various sensors and actuators can communicate with humans when connected to the internet. There are several devices that can connect humans and machines through the internet, one of which is NodeMCU. NodeMCU is a provider of IoT device that is equipped with ESP8266 chip [2][3].

NodeMCU ESP8266 only has one Analog to Digital Converter (ADC) pin with a range of 10bit namely pin A0. The limitations of analog input on the NodeMCU ESP8266 can be overcome by using the ADS1115 module. ADS1115 module is ADC with a range of 16bit with I2C [15]. But the ADS1115 module is only able to add as many as 4 ADC channels [1][3]. Therefore, the ADS1115 module is replaced with IC Multiplexer 4051 with 8 analog input channels [4][16].

This research uses IC Multiplexer 4051 which is connected with two air quality sensors (MQ-2 and MQ-135), LM35 temperature sensor and HIH-4030 humidity sensor. Overall sensor data is used to monitor air quality, temperature and humidity. Sensor data is sent through the internet so that it can be monitored through hardware that uses the Android operating system. MIT App Inventor is an open source web-based application for creating applications for Android devices [5].

2. System Overview

a. Hardware

Figure 1 is a block of the NodeMCU ESP8266 system with IC 4051 which is connected with a temperature sensor, a humidity sensor and two air quality sensors. NodeMCU ESP8266 must be connected to the internet, so that the sensor reading data can be sent to the cloud. Sensor data stored in the





cloud can be monitored using an Android smartphone connected to the internet[7].

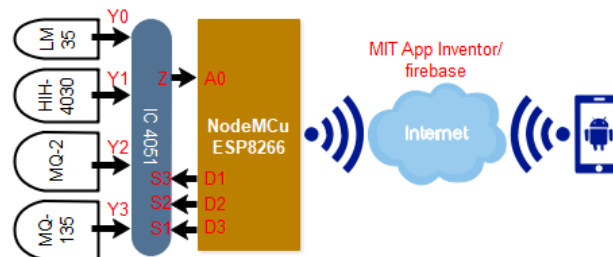


Figure. 1. System block IoT-based air quality monitoring.

The hardware system implementation is shown in Figure 2. Where in this case NodeMCu ESP8266 lacks ADC pins, then the multiplexer can be used to add analog inputs. One-off IC multiplexer used is 4051, where IC 4051 has 8 analog channels.

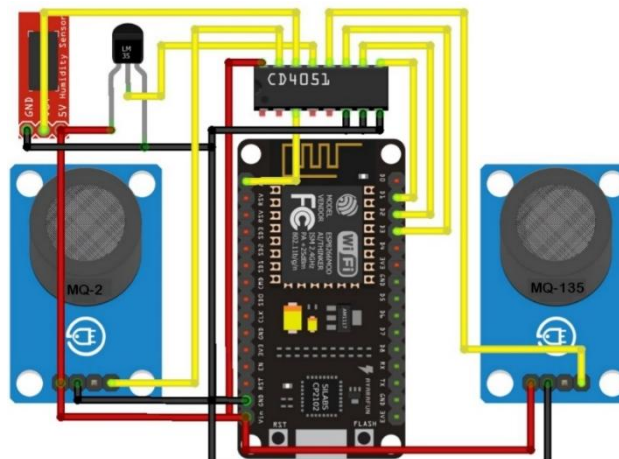


Figure. 2. Implementation of hardware systems

Connections on IC 4051, NodeMCu Esp8266, and sensors are shown as Figure 3.

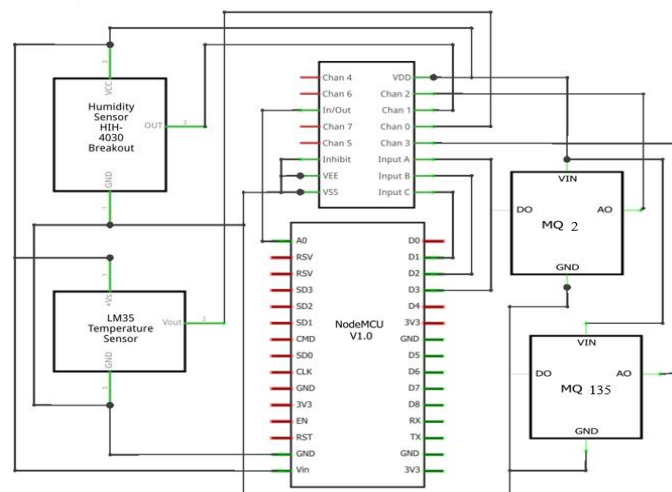




Figure. 3. Connections on IC 4051, NodeMCU Esp8266 and Sensors

Pins chan 0 to chan 7 are analog inputs from each sensor. Input A, input B, and input C are pin selector connected to the digital pins D1, D2, and D3 on NodeMCU ESP8266, in other word pins A, B, and C are used as a determinant of which channel input will be used. Determination of input channels (0 to 7) by giving values of 0 (low) and 1 (high) on each selector pin. Table 1 is the changes in the selector pin values to determine the channel used. Pin Z (in / out) is connected to analog pin A0 NodeMCU ESP8266 which is used to send sensor reading data from IC 4051.

Table 1
Determination of the channel used

\bar{E}	C	B	A	Chanel	Sensor
0	0	0	0	0	LM35
0	0	0	1	1	HIH-4030
0	0	1	0	2	MQ-2
0	0	1	1	3	MQ-135
0	1	0	0	4	-
0	1	0	1	5	-
0	1	1	0	6	-
0	1	1	1	7	-

Sensors that are commonly used to measure environmental parameters in this study are shown in Table 2. MQ-135 gas sensor was chosen to measure the concentration of carbon dioxide (CO₂), ammonia (NH₃), nitrogen oxide (NO_x), benzene, and smoke with a concentration of 10-1000ppm[3][8][9]. MQ-2 gas sensor has a high sensitivity to liquified petroleum gas (LPG), propane, methane and hydrogen with concentrations of 300-10000ppm[10][11][12].

LM35 is a temperature sensor that has high precision and linear output voltage, without calibration or additional components can show the accuracy of $\pm \frac{1}{4}^{\circ} \text{C}$ at room temperature[13]. HIH-4030 sensor is a humidity sensor with linear voltage output with a small current consumption of 200 μA [14].

Table 2
Sensors and measurement parameters

Sensor	Parameter
LM35	Temperature
HIH-4030	Humidity
MQ-2	LPG, Propana, Metana, Hidrogen
MQ-135	CO ₂ , NH ₃ , NO _x , Benzena, Smoke



Figure 5. Example of six blocks of computational concepts

b. Software

Software design starts with making a flowchart (Figure 4), then make a program on ESP8266 and make an application on the MIT App Inventor (MAI). Program on NodeMCU ESP8266 is made using the Arduino IDE(Integrated Development Environment).ESP8266 program is written in C language, including program needed to integrate sensors and cloud on the internet. Android IDE helps write and compile programs and upload to ESP8266 flash memory[2].





MAI is used to design and build applications on smartphones. MAI is a project whose resources are open to the public so that it can be downloaded and run on a different server. MAI provides code in the form of blocks that are very easy to set up[15]. In general, MAI has six computational block concepts that are often used including procedures, variables, logic, loops, conditionals, and lists[5]. Figure 5 show an example of six computational block concepts.

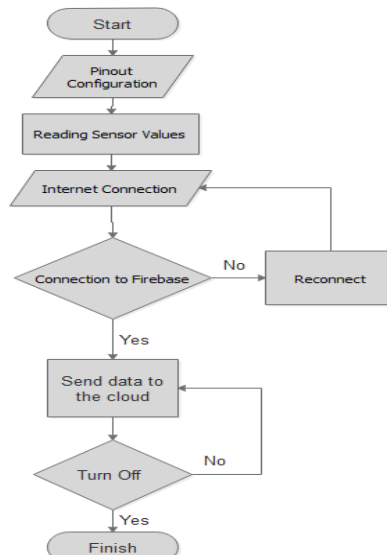


Figure 4. Flow diagram of sensor data retrieval using NodeMCU ESP8266

3. Implementation

Implementation of IC Multiplexer 4051 as a solution to add analog channels on NodeMCU ESP8266. IC 4051 is connected with sensors MQ-2, MQ-135, LM35, and HIH-4030. sensor data sent by ESP8266 via the internet can be viewed through an android-based smartphone. In other word this tool uses the concept of the Internet of Things (IoT). Figure 6 is the arrangement of programs on the Arduino IDE to connect ESP8266 to the internet.

```
#include <FirebaseESP8266.h>
#include <ESP8266WiFi.h>
#define FIREBASE_HOST
#define FIREBASE_AUTH
#define WIFI_SSID "xxx"
#define WIFI_PASSWORD "123456789"
void setup() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED)
    delay(500);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);
  int n = 0;
  void loop() {
    if (WiFi.status() != WL_CONNECTED ) {
      WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
      while (WiFi.status() != WL_CONNECTED)
        delay(500);
      Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
      Firebase.reconnectWiFi(true);
    }
  }
}
```

Figure 6. List program to connect ESP8266 to the internet

ESP8266 must be programmed for the selection of input channels through IC 4051. Figure 7 is the list





programs for selecting input channels.

```
#define MUX_A D4
#define MUX_B D3
#define MUX_C D2
#define ANALOG_INPUT A0
void setup() {
  pinMode(MUX_A, OUTPUT);
  pinMode(MUX_B, OUTPUT);
  pinMode(MUX_C, OUTPUT); }
void changeMux(int c, int b, int a) {
  digitalWrite(MUX_A, a);
  digitalWrite(MUX_B, b);
  digitalWrite(MUX_C, c); }
void loop() {
  float value;
  changeMux(LOW, LOW, LOW);
  value = analogRead(ANALOG_INPUT); //sensor LM35
  changeMux(LOW, LOW, HIGH);
  value = analogRead(ANALOG_INPUT); //sensor HIH-4030
  changeMux(LOW, HIGH, LOW);
  value = analogRead(ANALOG_INPUT); //sensor MQ-2
  changeMux(LOW, HIGH, HIGH);
  value = analogRead(ANALOG_INPUT); //sensor MQ-135}
```

Figure 7. List program for Selecting input channels

ESP8266 needs to be integrated with MAI so that sensor data can be stored and monitored. Figure 8 is a configuration for connecting to the Firebase database. Figure 9 sensor values in the firebase database to display. Figure 10 where the sensor values are updated into the database.

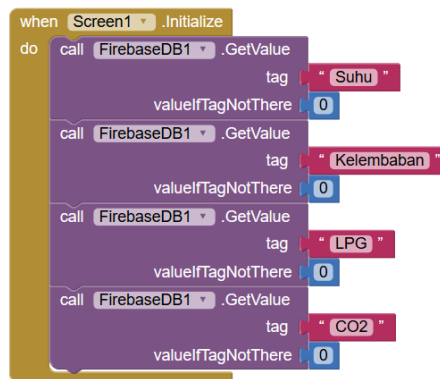


Figure 8. Program block to connect to the Firebase database

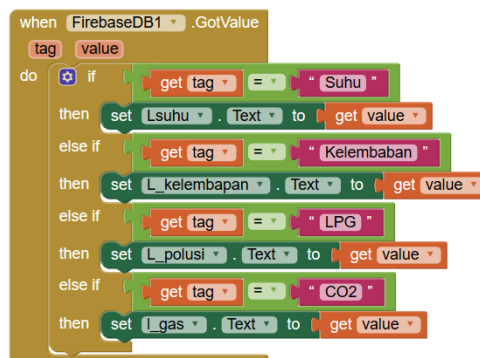




Figure 9. Program block obtains sensor values

```
when FirebaseDB1 .DataChanged
  tag value
do
  if get tag == " Suhu "
  then set Lsuhu . Text to get value
  else if get tag == " Kelembaban "
  then set L_kelembaban . Text to get value
  else if get tag == " LPG "
  then set L_polusi . Text to get value
  else if get tag == " CO2 "
  then set L_gas . Text to get value
```

Figure 10. Program block updating data into a database

After doing the ESP8266 and MAI programming process, then testing with an android smartphone connected to the internet. Download MIT AI2 Companion application through the Google Play Store using an android smartphone. Install AI2 Companion so that the application display as shown in Figure 11, then choose scan QR code. On the MAI web page, choose *build* then *provide QR code for.apk*, QR code display can be seen in Figure 12. Next scan the QR code on the MAI with a smartphone, download option appears and after it's finished then install. The application that has been built can be tried with a smartphone and looks like Figure 13.

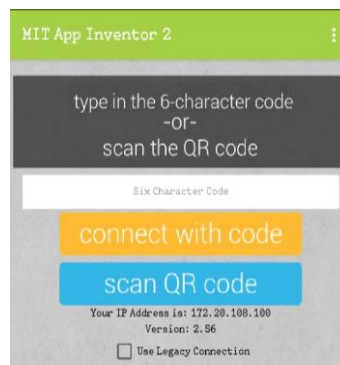


Figure 9. Display of the MIT AI2 Companion application

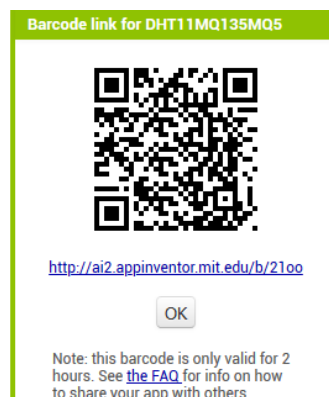




Figure 10. Display QR code on MAI

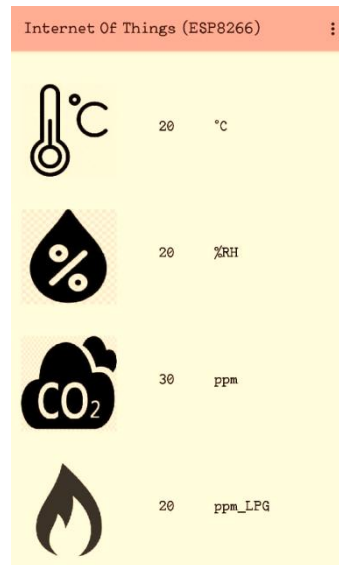


Figure 11. Display application integration of ESP8266 and MIT App Inventor

Table 3 shows that the levels of carbon monoxide and carbon di-oxide have different variations in each room condition and the number of people in it. Where in a room with an area of 200 M² with a total of 5, 10, 20, 30 and 45 people.

Table 3
Carbon monoxide and carbon di-oxide levels in various situations

Number of People	Area (M ²)	CO (ppm)	CO ₂ (ppm)
5	200	41	400
10	200	44	476
20	200	49	510
30	200	53	690
45	200	58	780

4. Conclusion

In this research a device for monitoring IoT-based air quality has been built and realized. The main components utilized are NodeMCU ESP8266, Multiplexer IC 4051, LM35, HIH-4030, MQ-2 and MQ-135. The design of this device focuses on using IC 4051 as an analog input enhancer for NodeMCU ESP8266. The result is that IC 4051 is able to integrate sensor data into ESP8266. Then the sensor data is sent to the Firebase (cloud) database via the internet. The database from Firebase can be monitored via a smartphone by first making the application on the MIT App Inventor. Finally, device for monitoring temperature, humidity and air quality are available on smartphones. The equipment can be integrated if ESP8266 and smartphone are connected to the internet.

5. References

- [1] S. Chanthakit and C. Rattanapoka, "Mqtt based air quality monitoring system using node MCU and node-red," *Proceeding 2018 7th ICT Int. Student Proj. Conf. ICT-ISPC 2018*, pp. 1–5, 2018.
- [2] S. Saha and A. Majumdar, "Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system," *Proc. 2nd Int. Conf. 2017*





- Devices Integr. Circuit, DevIC 2017*, pp. 307–310, 2017.
- [3] Z. Wan, Y. Song, and Z. Cao, “Environment dynamic monitoring and remote control of greenhouse with ESP8266 NodeMCU,” *Proc. 2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2019*, no. Itnec, pp. 377–382, 2019.
- [4] N. X. P. Semiconductors, “Datasheet IC Multiplexer 4051B,” no. September. pp. 1–23, 2014.
- [5] B. Xie and H. Abelson, “Skill progression in MIT app inventor,” *Proc. IEEE Symp. Vis. Lang. Human-Centric Comput. VL/HCC*, vol. 2016-Novem, pp. 213–217, 2016.
- [6] H. D. Hutahaean, B. D. Waluyo, M. A. Rais, U. N. Medan, and S. Utara, “Teknologi Identifikasi Objek Berbasis Drone Menggunakan Algoritma Sift Citra Digital,” vol. 04, pp. 193–198, 2019.
- [7] M. F. Nur, M. A. Murti, C. Setianingsih, F. T. Elektro, and U. Telkom, “Design of Home Electrical Appliances Control and Monitoring System Based on Android,” *e-Proceeding Eng. Vol.6, No.1 April 2019*, vol. 6, no. 1, pp. 125–134, 2019.
- [8] A. A. Ibrahim, “Carbon Dioxide and Carbon Monoxide Level Detector,” *2018 21st Int. Conf. Comput. Inf. Technol. ICCIT 2018*, pp. 1–5, 2019.
- [9] R. Tem, “Datasheet MQ-135 Sensor,” vol. 1. pp. 3–4, 2018.
- [10] T. Data, “Datasheet MQ-2 Sensor,” *Pololu*. p. 2, 2016.
- [11] J. M. Montoya and J. Chilo, “Real Time Wireless Monitoring System of CO₂ and CH₄ in Juliaca-Perú,” *10th IEEE Int. Conf. Intell. Data Acquis. Adv. Comput. Syst. Technol. Appl. 18-21 Sept. 2019, Metz, Fr.*, pp. 464–467, 2020.
- [12] S. Panpaeng, P. Phanpeang, and E. Metharak, “Cigarette smoke detectors for non-smoking areas in the building,” *2018 22nd Int. Comput. Sci. Eng. Conf. ICSEC 2018*, pp. 1–4, 2018.
- [13] C. Liu, W. Ren, B. Zhang, and C. Lv, “The application of soil temperature measurement by LM35 temperature sensors,” *Proc. 2011 Int. Conf. Electron. Mech. Eng. Inf. Technol. EMEIT 2011*, vol. 4, pp. 1825–1828, 2011.
- [14] Honeywell, “Datasheet HIH-4030/31 Series,” *Humidity Sensors*. p. 8, 2008.
- [15] B. Z. Barkana and W. McDonough, “AP Computer Science Principles: Designing the hourly App in MIT App Inventor,” *2019 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2019*, pp. 1–6, 2019.
- [16] ADS1115 16-BIT ADC - 4 CHANNEL WITH PROGRAMMABLE GAIN AMPLIFIER Available: <https://www.adafruit.com/product/1085>
- [17] <https://www.theengineeringprojects.com/2017/08/introduction-to-cd4051.html>

