An Investigation of a Low-cost CO2 Indoor Air Quality Monitor

Thesis · May 2014
DOI: 10.13140/RG.2.1.3901.1364

1 author:

Chen Shen
University College London

UNIVERSITY OF NOTTINGHAM

DEPARTMENT OF ARCHITECTURE AND BUILT ENVIRONMENT

K133RP -Final Report

An investigation of a low-cost CO_2 indoor air quality monitor

Chen Shen 13 May 2014

A report submitted in partial fulfilment of the regulations for the Degree of Bachelor of Engineering in Architectural Environment Engineering at the University of Nottingham, 2014.

Abstract

Carbon dioxide (CO2) sensors are increasingly used in areas of Indoor Air Quality (IAQ) monitoring and demand ventilation control. Insufficient ventilation rates can raise complains from occupant as well as cause discomfort and healthy problems, whereas over-ventilation contributes to unnecessary energy consumption and hence is not environmental or economic friendly. The existing monitors on the market are relatively expensive in UK. Therefore, there is an overwhelming demanding of developing low-cost CO_2 monitoring systems. The project is to construct and investigate a new prototype Carbon Dioxide IAQ monitor device designed by Liddament (2013). It is notable that the new monitor design using MG811 sensors benefits from the low cost. However, comparing its performance with the existing models is of great importance, as the result can determine whether it is marketable.

Acknowledgement

I wish to express my sincere thanks to my supervisor Dr Benjamin Jones for his great support throughout my final year in university.

I would also like to thank the University of Nottingham for funding most of the components required.

My thanks are extended to Martin Liddament for his extraordinary work which makes the success of this project possible.

I am also grateful to Dr ED Cooper and Dr Benjamin Jones who both agreed to be my referee as well as offering me the opportunities to attend CIBSE events. I gained a lot of inspiration from those events.

Contents

List of Figures

List of Tables

1. Introduction

A qualified indoor environment is essential for occupants. Studies (Chapin. 1974, and Szalai. 1972) have shown that typical modern populations spend more than 90% of their time indoor on average (Leech et al. 2004). EPA[1] researches indicate that indoor air quality has remarkable impact on occupants' comfort, productivity, health and wellbeing. Poor IAQ can lead to a reduction in the performance of office work by 6-9% (Wyon 2004) and an increase in school absence rate (Currie et al. 2007). While an investigation by Wargocki et al. (2005) suggests that the performance of school work can be improved by over 15% with the ventilation rate increasing from 5 to 10 l/s. Furthermore, occupants' exposure to indoor air pollution can cause various health problems, such as Sick Building Syndrome (SBS) and Building Related Illness (BRI) (Jones, 1999). EPA explains that the term 'Sick Building Syndrome' is used to describe situations in which building occupants experience acute health and comfort effects that appear to be linked to time spent in a building, but no specific illness or cause can be identified. Indicators of SBS include headaches, respiratory tract irritations (coughing, dry, itchy or sore nose, etc.), eye irritation, dry or itchy skin etc. Whereas, the term 'Building Related Illness' (BRI) is used when symptoms of diagnosable illness are identified and can be attributed directly to airborne building contaminants. Symptoms of BRI include cough, chest tightness, fever, chills, and muscle aches (HSE[2]).

The occurrence of illnesses related with poor ventilation has driven an increasing attention towards indoor air quality monitoring (Zampolli et al. 2004). Carbon dioxide in indoor environment normally comes from occupants' respiration and indoor combustion process. Although the CO_2 concentration in most buildings will not cause health problems as it is not itself a pollutant, it is commonly used as an indicator of indoor air quality and can be employed to estimate the adequacy of ventilation (CIBSE, Brown, 2004). Indoor CO_2 concentrations above about 1000 ppm are generally regarded as an indication of unacceptable ventilation rate (Daisey, 2003). Various national guidelines give specifications of different CO_2 concentration standards. BS EN13779(2007) classifies IAQ into 4 categories (see Table 1-1) - special, high, medium and low corresponding to 350ppm, 500ppm, 800ppm and 1200ppm of the indoor CO_2 concentration above the ambient (Jones and Kirby, 2012). Whereas, ASHARE Standards 62.1 recommends that the

[1] Environmental Protection Agency, United States
[2] Health and Safety Executive, United Kingdom

CO_2 concentration should not exceed 1000ppm, and levels less than 800ppm indicate the sufficient fresh air supply. CIBSE GUIDE A adapts the same standards of EN13779, based on the outdoor CO_2 concentration lies in the range of 350-400ppm.

Table 0-1 General classification of IAQ and CO2 concentration levels (EN13779:2007)

Category	CO_2-level above level of outdoor air in ppm	
	Typical range	Default value
Special indoor air quality	≤400	350
High indoor air quality	400-600	500
Medium indoor air quality	600-1000	800
Low indoor air quality	>1000	1200

There is an increasing use of CO_2 monitors by demand-controlled ventilation (DCV) systems in HVAC designs. Demand controlled ventilation (DCV) is a strategy that attempts to modulate the outside airflow rates based on real-time occupancy (Emmerich, 1997). The most common method to incorporate DCV into the design of an HVAC system is to adjust the amount of outdoor ventilation based on the level of CO2 in the building air. (Lawrence, 2004). The benefits offered by DCV include saving energy by avoiding the heating, cooling, and dehumidification of more ventilation air than is needed while maintaining required levels of indoor air quality (Dwyer, 2014).

There are many carbon dioxide IAQ monitors available on the market. They are well developed but quite expensive. The aim of this project is to build and test a new prototype low-cost indoor air quality monitor designed by Liddament (2013). The construction and calibration procedures are demonstrated by a detailed explanation. Meanwhile, the report also presents a performance evaluation and cost analysis of the monitor.

The aim will be achieved by meeting the following objectives:

1. Acquire parts for the monitor
2. Build the monitor
3. Calibrate the monitor
4. Test the monitor for performance evaluation
5. Determine if the sensor is usable, reliable and repeatable.
6. Recommend improvements.

2. Literature Review

Early investigation- Hooker's research in 1875

In his article published in 1875, Hooker observed that the amount of organic impurity is in proportion to the amount of CO_2 produced by the inhabitant of an air-space. Accordingly, he suggested taking CO_2 concentration as a measure of the condition of the air space.

In order to calculate the amount of air that is required to reduce CO_2 concentrations to an acceptable level (defined as no sensible difference between inhabited space and outdoor air), he conducted a series of experiment in barracks and hospitals, involving hundreds of observers. Notes of their sensation were made on their first entering to the room from the out air. The air quality was classified as 'fresh', 'rather close', 'close', 'very close' and 'extremely close' based on the records of observers' expressions. The air of each room tested in the experiment was collected using jars or bottles, and the concentration of CO_2 was then measured and compared. Dry bulb, wet bulb temperature and humidity were also measured as they are thought to be related to occupants' comfort.

According to the result of observation and calculation, Hooker (1875) recommended a standard that the respiratory impurity (estimated as CO_2) ought not to exceed 0.2000 per 1000 volume (equivalent to 200ppm in ideal gas condition). Taking the mean ambient CO_2 concentration at 400ppm as Hooker determined, this gives a mean internal CO_2 concentration 600ppm.

Compare to the recommended values given by ASHARE standards (700 ppm above outside CO_2 concentration) and CIBSE Guide A (400-600 ppm, see Table 1), the standard suggested by Hooker seems to be much more strict.

It is hard to determine the exact reasons for the above result due to the limited information presented in Hooker's article, although the technologies to test CO_2 levels in 19th century are suspected to be related. Of course there are other factors make the experiment can hardly be reviewed as perfect in today's perspective. For instance, there was tobacco-smoking present during the test, and the results were greatly affected by the subjectivity of observers. However, Hooker's research is still of great importance. Not only does it show scientists' awareness of the importance of IAQ one and a half centenary ago, but also gives an insight into the early development of indoor air quality investigation.

NDIR Sensors

There are two types of electronic sensors can be used in the detector modules in order to detect carbon dioxide (Lee. et al 2012). They sensors are categorised according to the different working mechanism they based on. One type is called NDIR (non-dispersive infrared sensor) while the other one is generally referred as chemical based CO_2 sensors.

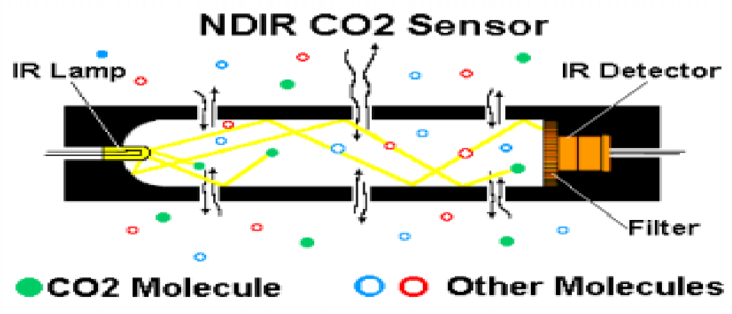

Figure 0-1 Principal diagram of a typical NDIR carbon dioxide sensor

NDIR sensor is currently the most used type of carbon dioxide sensor in the industry (Lee. et al. 2012). A typical NDIR CO_2 sensor consists of a tube or a chamber in which a source of infrared light is placed at one end and at the other end there is a detector. The IR light is directed through the sensing chamber which is filled of gas. There is a filter placed in front of the detector, which eliminates all lights except the particular wavelength (4.26 µm) that only CO_2 molecules can absorb. The concentration of carbon dioxide is then determined from the amount of light reaching the detector.

As Kaneyasu et al. (2000) indicates, although infrared spectroscopic analysers are commonly used, they have several disadvantages such as their large size and high cost. And therefore, there is a need to develop inexpensive CO_2 monitors, such as chemical carbon dioxide sensors. The most common type is the one used in the new proposed IAQ monitor design - MG811 model. The details of its working principle are explained in Chapter 4.

Existing devices on the market

There are many CO_2 indoor air quality monitor devices available on the market. Table 2-2 shows some popular ones and their characteristics.

Table 2-1 Information of existing NDIR CO2 monitors on the market[3]

Manufacturer & Model	**Testo-535**	**TSI IAQ7515**	**Q-Trak 8550**	**EXTECH-CO250**	**ANTON-IAQ8494**
Sensor type	NDIR	NDIR	NDIR	NDIR	NDIR
Measurement Range (ppm)	0 - 9999	0 - 5000	0-5000	0-5000	0-9999
Accuracy	±50ppm/±2% (1)	±50ppm/±3% (2)	±50ppm/±3%	±50ppm/±5%	±75ppm/±5%
Resolution (ppm)	1	1	1	1	2
Operating Temp (℃)	0 - 50	5 - 45	5-45	0 - 50	-15 - 60
RH Range	0~99%	Up to 80%	5~95%	0~99%	0~95%
Other parameters detectable	None	None	Temperature Humidity	Temperature Humidity	Temperature
Price	£444.00 (VAT excluded)	£374 (Estimated)	Not available	£303.60 (VAT included)	£228.00 (VAT included)

(1) When CO_2 concentration < 5000ppm. If CO_2 concentration is over 5000ppm, the accuracy is ±100ppm or ±3% mv.

(2) The error should be taken as whichever is greater. This applies to all sensors in the table.

As can been seen from Table 2-1, all the models are constructed of NDIR sensors. Two monitors have a measurement range of 0- 9,999ppm, while the other three have the upper limits at 5000ppm. Accuracy of most monitors is limited in ±50ppm, whereas the error of

[3] Check product specifics for more details. Links are attached in reference list.

ANTON IAQ8494 can reach ±75ppm. ANTON IAQ8494 also has a higher resolution of 2ppm, while other models' value of 1 ppm. However, it also has a broader operating temperature range, which makes it more competitive for applications in hot or cold climates. As shown in the table, all models have high tolerance of humidity, but condensation should be avoided in all cases.

While the price of a single Q-Trak 8550 monitor is unknown, it can be seen that the prices of other IAQ monitors vary from £289 (include VAT) to £444 (exclude VAT). The costs of those devices are evidentially high compare to the new IAQ monitor design which Liddament (2013) claims to be less than £55. However, apart from cost, reliability is also another key factor that could determine whether it is usable or not. This requires further investigation of the device's performance and the comparison to other models.

Testo-535

TSI IAQ7515

Q-Trak 8550

EXTECH-CO250

Anton IAQ 8494

Figure 2-2 Existing NDIR monitors on the market

3. Theory

Martin Liddament (2013) designed a new prototype CO_2 indoor air quality monitor with a low cost under £55. The details of the design including explained guidance of building the device are all available on Veetech.co.uk. The completed device consists of a MG811 carbon dioxide sensor, an Arduino R3 processor, an Arduino Ethernet Shield data logger, a digital display and other accessories such as power supply chargers and connecting wires.

Components

CO_2 Sensor Details

This new IAQ monitor design is based on a MG-811 Metal Oxide sensor module manufactured by Sandbox electronics. Full details of the sensor can be easily found online or following the link MG-811 datasheet [Viewed 7 May 2014]. MG-811 is the most common type of chemical carbon dioxide sensors (See Figure 3-1). Its ability to generate a liner voltage output makes it an ideal choice for application in indoor air quality controlling. But it also has other highlighting features. Those include:

- **High sensitivity and selectivity to CO_2.** The MG-811 sensor is able to detect the CO_2 concentration ranging from 0 – 10,000 ppm. While being sensitive to CO_2, it is less sensitive to alcohol and CO which are substances usually existing in indoor environment.
- **Low humidity and temperature dependence.** The manufacturer claims that it can be operated within the temperature range of -20–70 ℃ as well as generating relative stable outputs at various humidity levels (Hanwei Electronics Co. Ltd).
- **Low cost.** The main advantage of MG-811 sensor is its fairly low price compare to other sensor modules. Although the price varies between suppliers, it is possible to get a single sensor unit (without amplifier) for only £15 approximately (Ali Express.com)

Figure 0-1 MG811 CO_2 Gas Sensors without circuit broads (Ali Express, 2014)

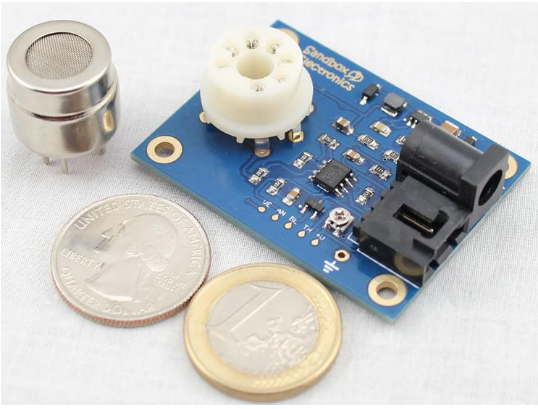

Figure 0-2 Sandbox MG-811 CO_2 Gas Sensor Module (Sandbox Electronics, 2014)

The MG811 CO_2 sensor works through an electrochemical reaction that occurs when carbon dioxide passes over the sensor. The chemical reactions can be written as

Catholic： $2Li^+ + CO_2 + 1/2O_2 + 2e^- = Li_2CO_3$

Anodic： $2Na^+ + 1/2O_2 + 2e^- = Na_2O$

Overall reaction： $Li_2CO_3 + 2Na^+ = Na_2O + 2Li^+ + CO_2$

Once power is applied, the sensor begins warming up to aid in the chemical reaction in 30-60 seconds. When the surface temperature is high enough, the sensor will generate an output voltage in the range of 100-600 mV which is equivalent to CO_2 concentration of 400-10,000 ppm (Sandbox Co. Ltd). The low voltage then needs to be amplified by applying relative equations so it can be read by Arduino modules (see next section). In the circuitry of the Sandbox module (see Figure 3-2), an operational amplifier and a stabilised 6 volt output chip are attached to ensure the operation and stability of the unit.

Arduino modules

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software (Arduino.cc). On the official website it is defined as 'a tool for making computers that can sense and control more of the physical world than a desktop computer'. Arduino was originally developed as an aid for teaching students, but then developed commercially by Massimo Banzi and David Cuartielles in 2005. Its ease of use and durability make it enormously successful with students, artists and designers (Monk 2012).

The Arduino can sense the environment by receiving inputs from a variety of sensors and reacting with controlled outputs. The heart of the Arduino is a microcontroller. This allows the broad to communicate with computer through programming. The programming codes for Arduino are called sketches. Sketches can be written and uploaded using Arduino software, i.e. Integrated Development Environment (IDE). A USB lead is required for programming to connect the device with computer.

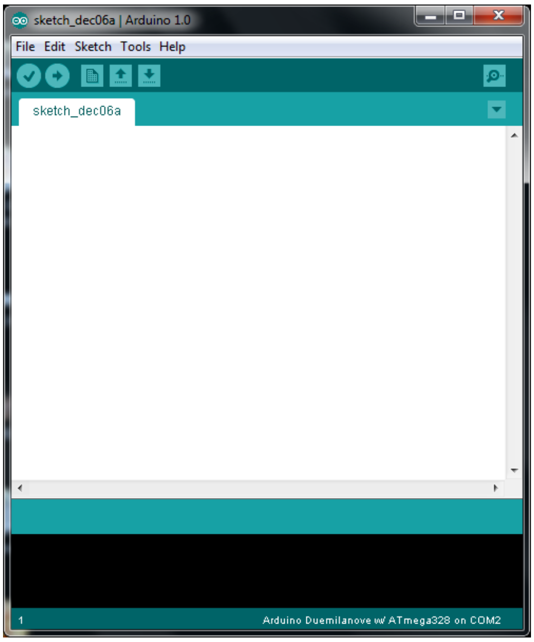

Figure 0-3 Arduino Integrated Development Environemnt (IDE)

There are a range of Arduino products for different functions while two of them are used in this carbon dioxide IAQ monitor design. One is the Arduino Uno board (see Figure 3-2) and the other one is the Arduino Ethernet shield (see Figure 3-3). Although a brief introduction of their roles will be present later, this article will not cover the technologies of the two Arduino modules in full details. First it is beyond the author's expertise. On the other hand, although a working knowledge of the Arduino is required for completing the construction of the monitor, it is not necessary for users to gain professional understanding because of its ease to use.

Figure 0-4 Photo of unpacked Arduino UNO R3

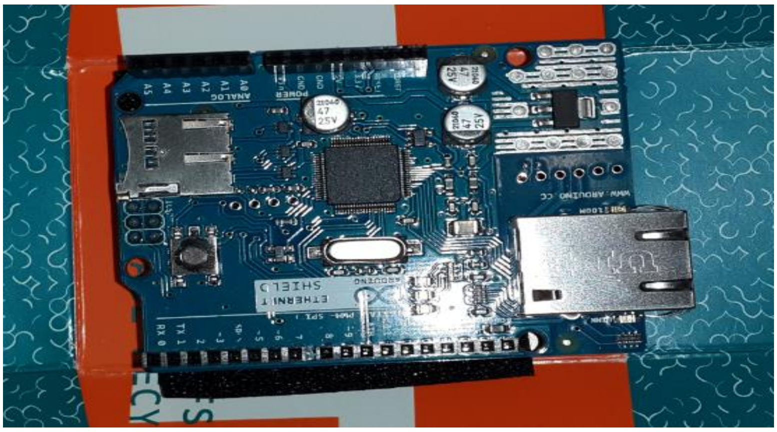

Figure 0-5 Photo of unpacked Arduino Ethernet Shield

For an easy understanding, the Arduino Uno here basically serves as a microprocessor. It receives the input signals from the MG811 sensor module, and then produces output voltages that are translated into numbers shown by the digital display.

The Arduino Ethernet Shield has an on-board micro-SD card slot and can be plugged on top of the Arduino Uno broad. The shield gives the potential to connect the monitor to the internet so the readings can be shown constantly on a computer. However, the main function of the shield here is to record and store the CO_2 measurement data so it can be analysed later on a computer.

Digital display

A four-digit, seven-segment Display is used to show the readings of CO_2 concentration level. Full details can be found from the datasheet (Accessed on 7 May 2014). As shown in pin layout (see Figure 3-6), there are 12 pins in total, 6 pins each aligned on the top and bottom side of the display. Those pins are numbered in the way shown in Figure 3-6 as a conventional practice.

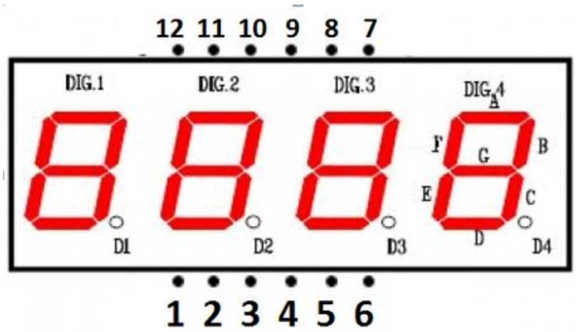

Figure 0-6 Display pin layout and numbering (Limitless, 2013)

Each pin controls a digit or a segment. The corresponding relationship between the pins and the segment/digit that they control is illustrated in Figure 3-7. When connecting to the Arduino, it is important to have 220 ohm limiting resistors (see Figure 3-8) inserted in series with each segment of the display. Otherwise, the display will be burned and damaged quickly.

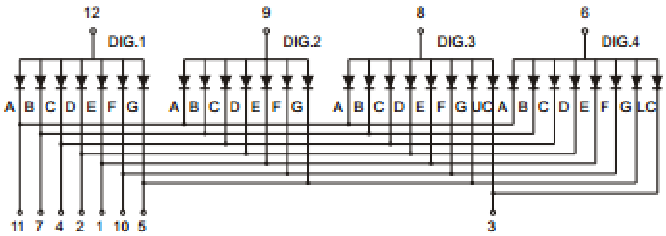

Figure 3-7 Internal Circuit Diagram 4-digit-7-segment diplay (Datasheet)

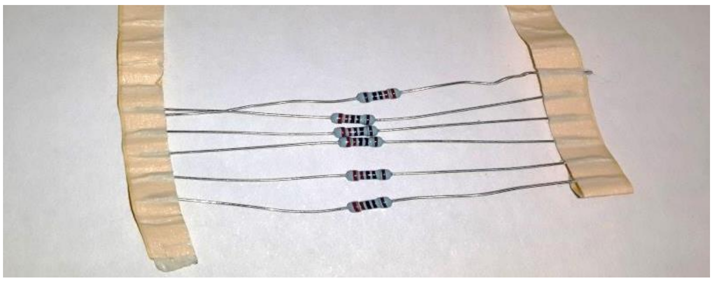

Figure 03-8 Figure 3 8 Photo of 220 ohm Metal Film Resistor

Additional items

Two 9-12 Volt stabilised DC power supplies (see Figure 3-10) rated at a minimum of 1 Amp with a 2.1 mm connector are needed. One is used for heating the sensor while the other one is used to power the Arduino. Apart from the chargers, other items required include connection wires (see Figure 3-11), a USB cable to connect the Arduino to a computer (see Figure 3-12) and a breadboard (See Figure 3-13).

Figure 0-10 9-12 Volt DC power supply

Figure 0-9 Jump wire kit

Figure 03-10 USB connection cable

Figure 03-13 Photo of a Breadboard

Construction of the monitor

Table 3-1 shows the Arduino pins that are used and their corresponding connection.

Table 3-1 Arduino pin layout to the sensor and digital display

Arduino Pin	Connection	Notes
GND	CO2 Sensor Black Connector (0V)	
5V	CO2 Sensor Red Connector (5V)	
A0	CO2 Sensor Yellow Connector (Analogue Output)	
A1	Display Anode 1	See Display Data Sheet
A2	Display Anode 2	
A3	Display Anode 3	
A4	Display Anode 4	
D2	Display Segment A (VIA a 220 Ohm Resistor*)	See Display Data Sheet
D3	Display Segment B (VIA a 220 Ohm Resistor*)	
D5	Display Segment C (VIA a 220 Ohm Resistor*)	
D6	Display Segment D (VIA a 220 Ohm Resistor*)	
D7	Display Segment E (VIA a 220 Ohm Resistor*)	
D8	Display Segment F (VIA a 220 Ohm Resistor*)	
D9	Display Segment G (VIA a 220 Ohm Resistor*)	
NOTE: To avoid damage to the display 220 Ohm Resistors are essential		

The construction process can be divided into four steps:

1) Mount the Arduino Ethernet Shield on top of the Arduino Uno R3 broad
2) Insert the MG811 sensor head into the module bed.
3) Connect the CO_2 sensor module to the Arduino module using jump wires according to the pin layout listed in Table 3-1. Figure 3-14 illustrates the connecting relations of the circuit between the Arduino and the sensor head.
4) Connect to the Arduino modules to the digital display using jumping wires according to Table 3-1. The connections are illustrated in Figure 3-15[4].

A photo of the completed device is taken and shown in Figure 3-16.

[4] Figure 3-15 shows the 220 Ohm resistors are inserted in series with each digit while Table 3-1 indicates to be with each segments of the display. Both ways should work.

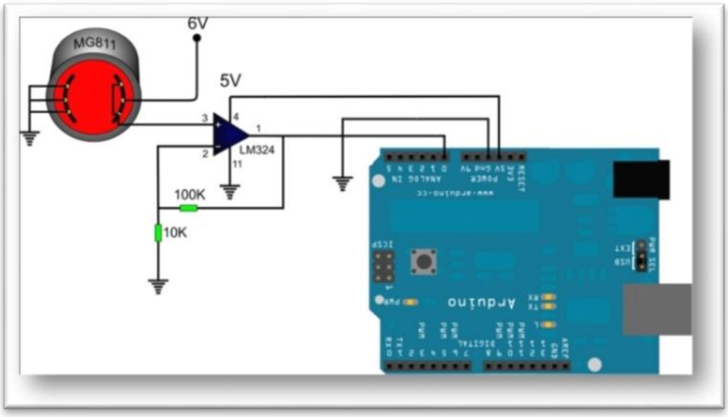

Figure 3-114 Connection diagram between sensor and the Arduino ([Reference](#))

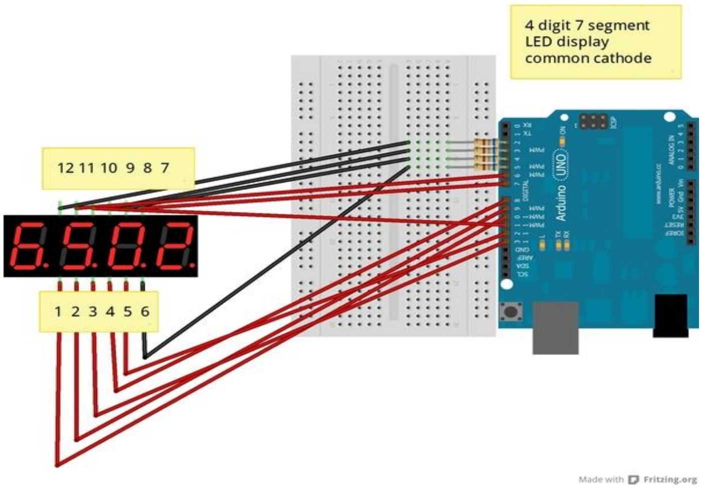

Figure 03-125 Connection between the digital display and Arduino ([Rothen, 2013](#))

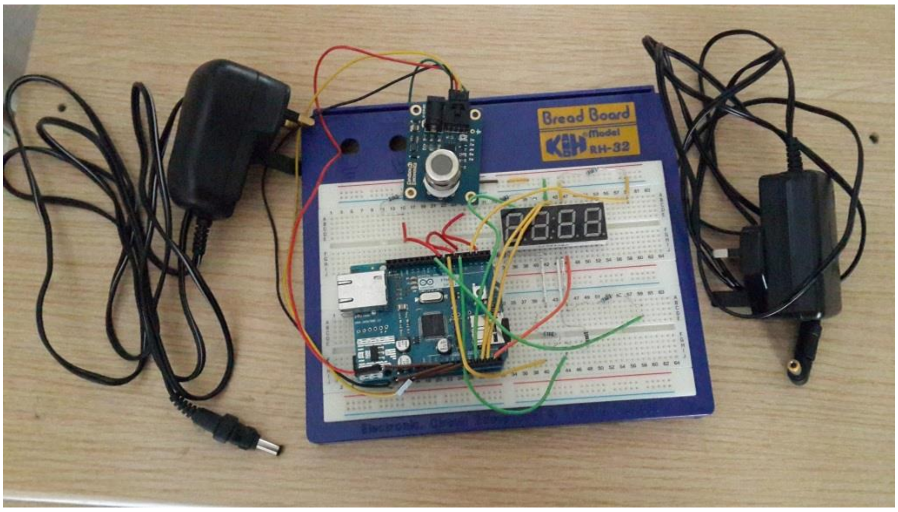

Figure 03-136 A photo of the completed MG811 carbon dioxide IAQ monitor

The whole procedure of building the device can be illustrated using a schematic diagram as shown in Figure 3-16.

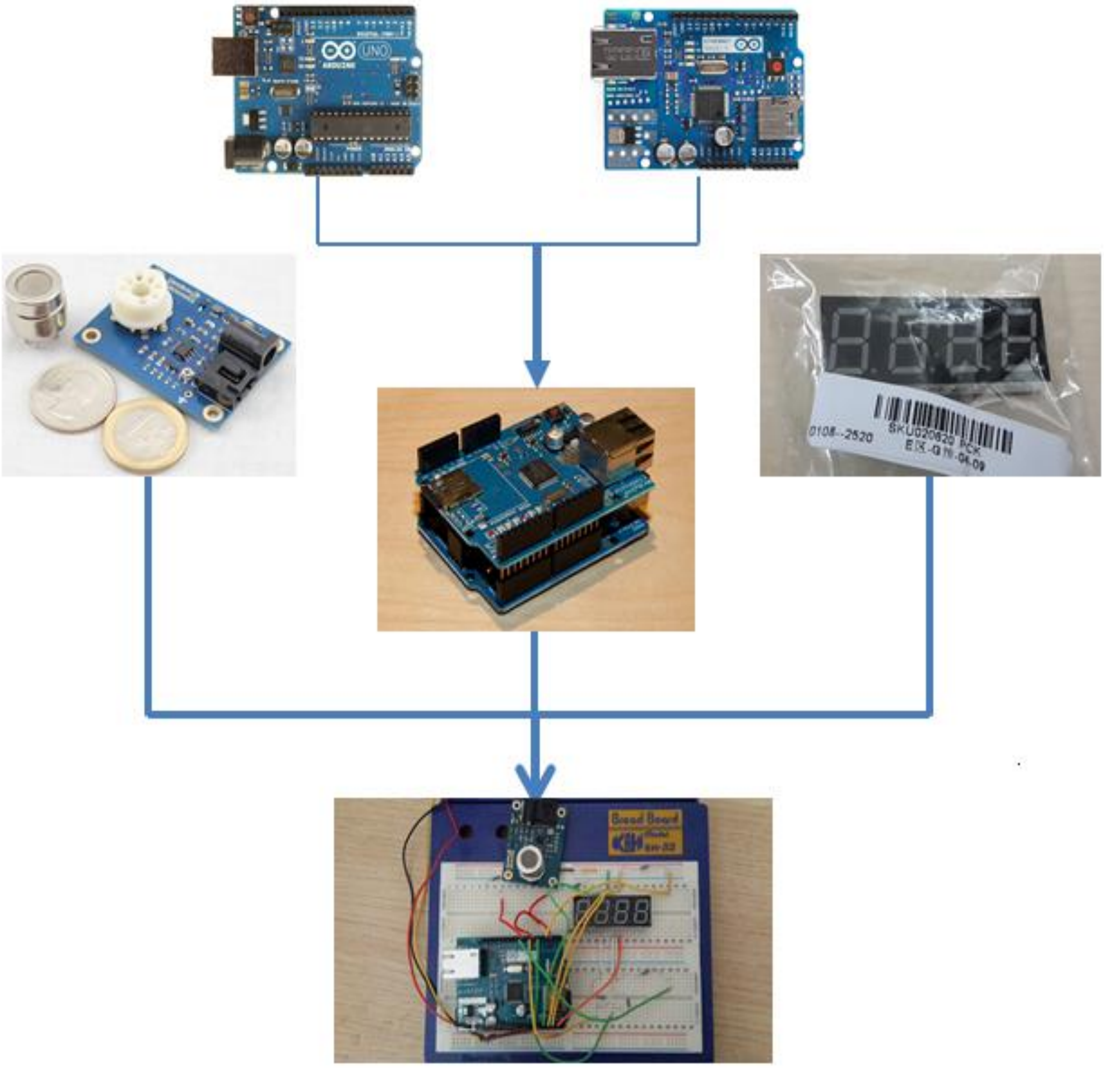

Figure 03-147 Photographic Illustration of the Co_2 Monitor Construction Process

4. Method

Calibration process

The MG-811 sensor has to be calibrated before taking any further measurements. Without the process, it will only give meaningless results. Two sources at known CO_2 concentration levels are required to achieve an accurate calibration result. In the data sheet it is suggested to calibrate the MG811 sensor for 400 ppm and 1000 ppm. However, while the outdoor ambient environment can be treated as roughly a 400 ppm source, it is not easy for most users to get access to a reliable 1000 ppm source. Liddament (2013) found that the sensor can still be sensitive at high concentration levels. Therefore he proposed to use the exhaled air as an alternative source which gives a second calibration point at 40,000 ppm.
Liddament (2013) showed the calibration steps in details and also identified possible causes for failure on the VEETECH website. The calibration procedures practiced in this research generally followed Liddament's description, but alternations are also made according to resources available.

1. **Load the calibration program into Arduino**

Both sketches of the calibration program and operation program were written by Liddament (2013) and can be found in Appendix I. To start the calibration, first load the calibration sketch into the Arduino UNO processor via the Arduino IDE software. The function of this sketch is basically 5000mV (5V) voltmeter. Liddament (2013) underlined that it is important to use this voltmeter rather that attaching an external metre. This is because any error between the Arduino value and the externally measured value will seriously impair the calibration.

2. **Check that the calibration program is yielding a reliable result**

The Arduino can give a plausible but incorrect result if it does not receive sufficient power from the power supply. To check for correct operation connect power to the Arduino and the CO_2 sensor module. Also make sure that the display is powered via the Arduino logic pins. While the device is operating under full load, carefully disconnect the yellow CO2 sensor output wire from A0 of the Arduino and connect a jump wire from 3.3V to A0. Check that the display reads 3.3 Volts to within 1% error, i.e. 2700-3300 mV. Any deviation usually suggests that the power supply to the Arduino is insufficient. This may be found if the Arduino is only power by a computer.

3. Allow the system to reach steady state conditions

If the power supply is proven to be sufficient, restore the yellow wire to A0. Typically the display will give readings between 2500 and 4500. The CO2 sensor now need be left to 'age' under full operating conditions so that the system can reach steady state. This process can take as much as 48 hours for a new MG811 CO_2 sensor. Ideally leave a new sensor to operate for two days in an unoccupied room. The heating element should be allowed to operate for at least 2 hours before taking readings in subsequent restarts. Liddament (2013) mentioned that the sensitivity of the sensor may change over several weeks. However, this can be easily spotted by observing the output for 'outdoor' conditions.

4. Determine the CO2 sensor output voltage at 400 ppm (v400)

Once the system has reached 'steady state' it is now time to determine the output voltage of the sensor at 400 ppm. Liddament (2013) suggested the way by letting the device operate in a totally unoccupied and well ventilated space. To ensure the room has enough ventilation the windows need to be left open and the premises need to be left unoccupied for as long as possible (e.g. 2-3 hours). Ideally the operating temperature should be at normal room value. Then observe the voltage without breathing on the sensor. Vacate the space for another hour and check again. If the voltage is substantially unchanged, take a note of the reading. This gives the value 'v400ppm' in the Arduino operational sketch.

Liddament's method requires a room or space that can be unoccupied with windows opening for a long time. This may raise safety concerns to some users. In addition, it is hard to tell whether the space is sufficiently ventilated so that the indoor CO_2 concentration levels are identical to outdoor ambient. One may wonder why not simply exposing the sensor in the outdoor environment to determine the output voltage at 400ppm. This is mainly due to the concerns of raining in UK climate which may lead to wrong results or even serious damage to the electronic circuit. However, for climates where raining is not a concern, this can be an easier and more practical method. Another alternative is to place the sensor at the edge of the window opening as long as there is accessible power supply nearby. Both the methods have been proven to be practicable during the research and they give close results.

No matter which method is used, as Liddament(2013) underlined, the ultimate accuracy of the sensor is dependent on the effort spent in determining the value of v400ppm.

5. Determine the CO2 sensor output voltage at 40000 ppm (exhalation concentration)

This is the most challenging but yet crucial step in the calibration process and will substantially determine the final success. First a bag of suitable size is needed. Liddment(2013) suggested a bag of length about 30cm and width about 22cm, but for most users the easiest one to get would be a medium size sealable sandwich bag. Once all are set, the following steps should be taken.

1) Snip one corner of the bag to leave a tiny hole as the ‘exhaust vent’;
2) Inflate the bag as best as possible and then carefully place the complete unit in the bag without switching the unit off or disturbing the wiring. Make sure there is no direct connect between the sensor head and the bag surface because the high temperature of heated sensor could melt the plastic. Insert the device in an open top plastic box or rigid if necessary. Connecting wires to the power supplies will be through the open end of the bag;
3) Seal the open end of the sandwich bag as best as possible and exhale into the bag through the open end very gently (may be easier with a straw);
4) While taking fresh breath, pinch the exhaust vent completely to prevent ambient air entering the bag. When exhaling air into the bag, the exhaust hole should be semi restricted to allow the bag to inflate. Within a few minutes the bag will be virtually entirely filled with exhaled air at 40000 ppm;
5) When the reading on the display remains constant for approximately 30 seconds then steady state has been reached;
6) Keep exhaling for another 30 seconds to make sure and then take a note of the reading. This is the output voltage at 40000 ppm and normally would be at about 2-2.5 volts below the 400 ppm voltage.

Liddament (2013) identified the biggest drawback during the process, i.e. the relative humidity in the bag will be 100%. But he also underlined that this does not seem to be a problem other than making it difficult to read the display, mainly because the MG811 sensor has a high tolerance to operation humidity level. The whole process is not straightforward so it is recommended to have two people working together.

It is encouraged that the users adopt adjustments to the calibration process according to their own circumstances. For example, when determining the v40000 ppm voltage in the actual practise, instead of putting the whole unit into the bag as Liddament (2013) suggested, only the CO_2 sensor module was placed into the bag. This has been proven to be feasible and more convenient as a large breadboard was used. It is important to keep the unit switched on during the periods of determining output voltages. A correct calibration result is crucial to the success of the monitor’s measurement. So the more effort put into it, the better will be the accuracy of the monitor.

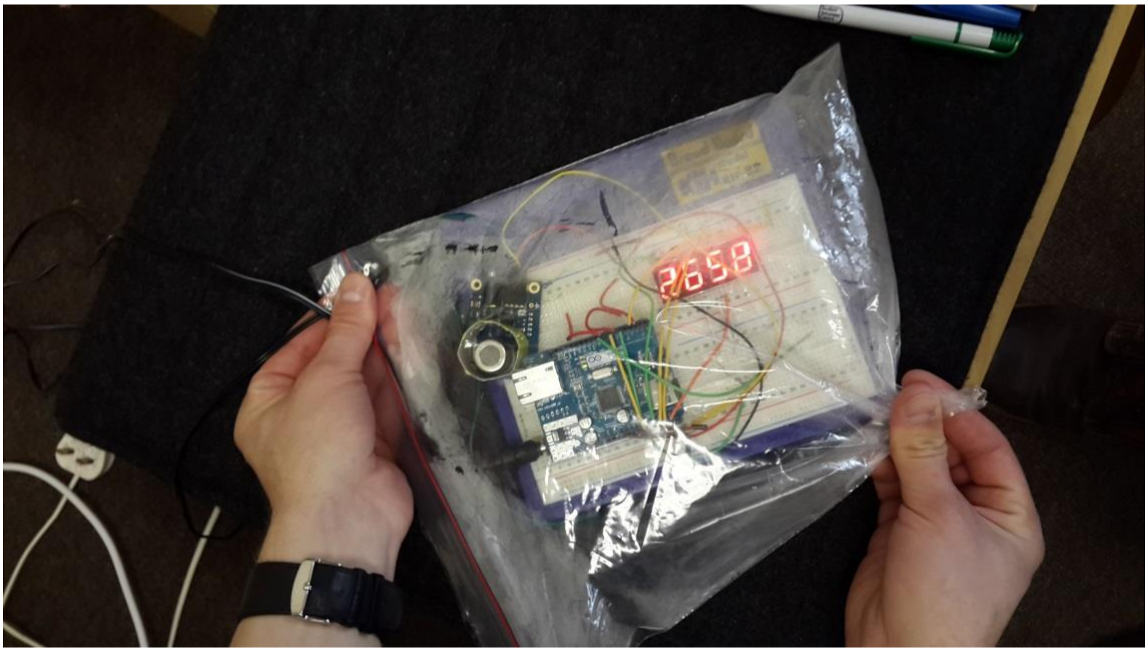

Figure 40-1 Determining the output voltage at 40000 ppm

Testing operation

After the output voltage at 400 ppm and 40000 ppm are determined from the calibration, replace v400 and v40000 in sketch 2 with the new values, and then upload the uploaded sketch into Arduino via IDE software. The operation program allows the device to perform the following tasks:

1. Monitors the CO_2 concentration value every 10 seconds
2. Displays the reading of CO_2 concentration in ppm on the Four-digit Display continuously
3. Writes the CO_2 value to the SD card approximately every 10 minutes
4. Write the CO_2 value to the serial monitor every 10 seconds (if connected)

Once the sketch is uploaded successfully, the display will immediately give the reading of CO_2 concentration levels in ppm. The equation used to convert a voltage to a CO_2 concentration can be deducted from the sketch:

$$CO_2\ Concentration = 10^{\,Voltage - v400ppm / v400ppm - v40000ppm} + \log_{10} 400$$

Where 'CO_2 Concentration' is the reading shown on the display and 'Voltage' means output voltage from CO_2 module.

In order to evaluate the performance of the IAQ monitor, an operational testing was run. Space with a regular occupation pattern is required so that the CO_2 variation can be verified with predications. The room that has been chosen as the testing environment is room A1 located in Dr located on the ground floor of Marmont Centre at the University of Nottingham. The reason for choosing this room is because it is used as a private office and hence has a regular and predicable occupation pattern. A simple sketch of testing environment is presented below:

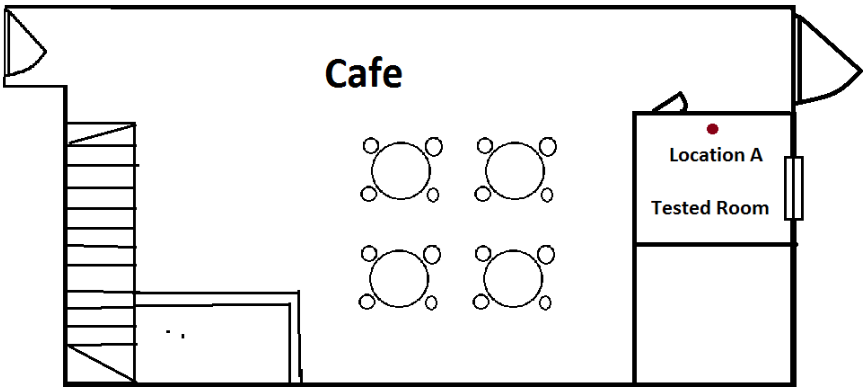

Figure 4-2 Testing environment layout

As shown in the layout, there is a set of windows in the external wall of the room and also a door that separates the office and the café. The building is natural ventilated so there is no mechanical ventilation in the room. Additionally, during the testing period all the windows were kept closed and the door was only opened for a few times when people need to get through. This means the room is mainly ventilated by air leakage through fabric and window opening gaps, although small amount of air exchange will take place between the café and the tested room through door openings or gaps.

First the IAQ monitor device was placed at location A and then switched on. It was left in a constant operation from the Wednesday (2 April 2014) midday till about 1 pm of the next Monday (5 April 2014). The CO_2 concentration levels were collected every 10 minutes and stored in the SD card. The data was written into a simple text file which can be read directly by Excel. The data can be easily read and edited with no complex knowledge involved.

Figure 4-3 A Photo of the Carbon Dioxide IAQ monitor under operation

5. Results

Results of the calibration experiments show that the output voltage of the CO_2 sensor module to be **1749mV** at concentration of 400 ppm and **3064mV** at concentration of 40000 ppm.

The calibrated IAQ monitor operated for about five days starting from noon on 2nd April to 7th April. The indoor CO_2 concentration levels of the room were recorded every 10 minutes. Once the device is switched on, the CO_2 sensor module starts to warm up. It takes around 2-3 hours for the sensor to reach steady state. During this process, the monitor gives unstable and inaccurate readings which are typically unrealistic high. When editing the data, the measurements taken during this period are disregarded.

The data sampled for performance evaluation starts at a point after the monitor are stabilized. Figure 5-1 shows the variation of CO2 concentration levels in the room during the whole testing period (2nd April – 7th April). The time period covered starts from 14:00 pm on Wednesday to 12:30 pm on the following Monday.

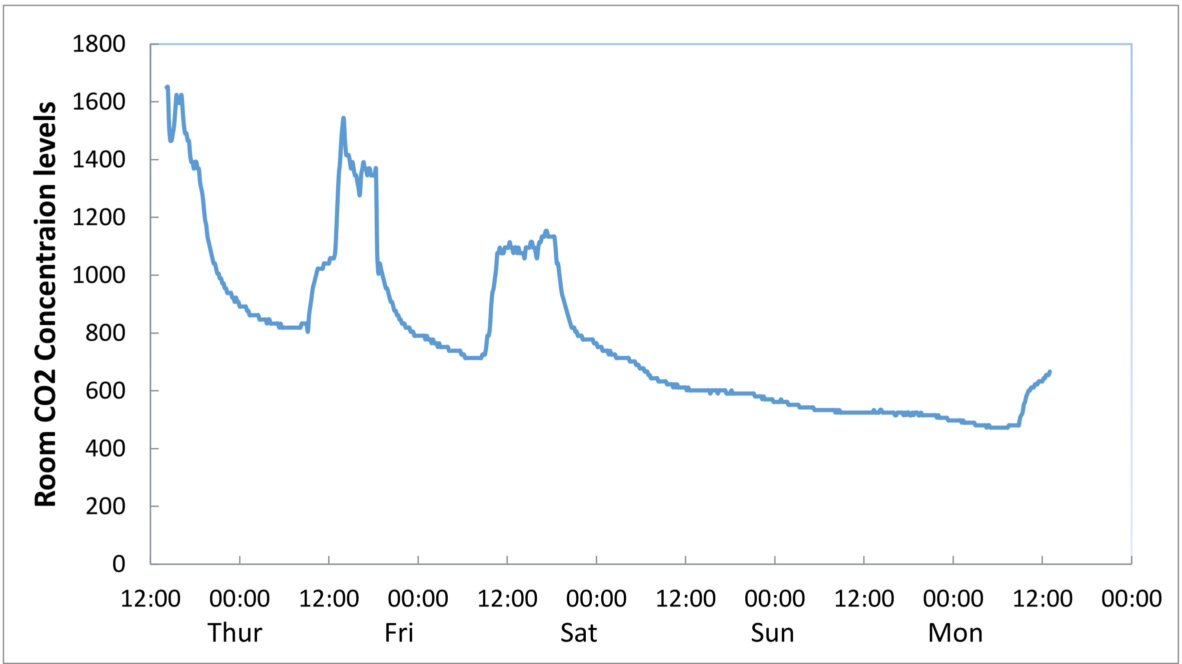

Figure 5-1 CO_2 concentration levels of the tested room

To give clearer images of how the CO_2 concentration levels in the room varied over specific days, the data is broken down to produce detailed charts that cover shorter periods.

Figure 5-2 and 5-3 show the indoor CO_2 concentration levels on Thursday and Friday while Figure 5-4 shows the variation of CO_2 concentration levels over the weekend till Monday noon.

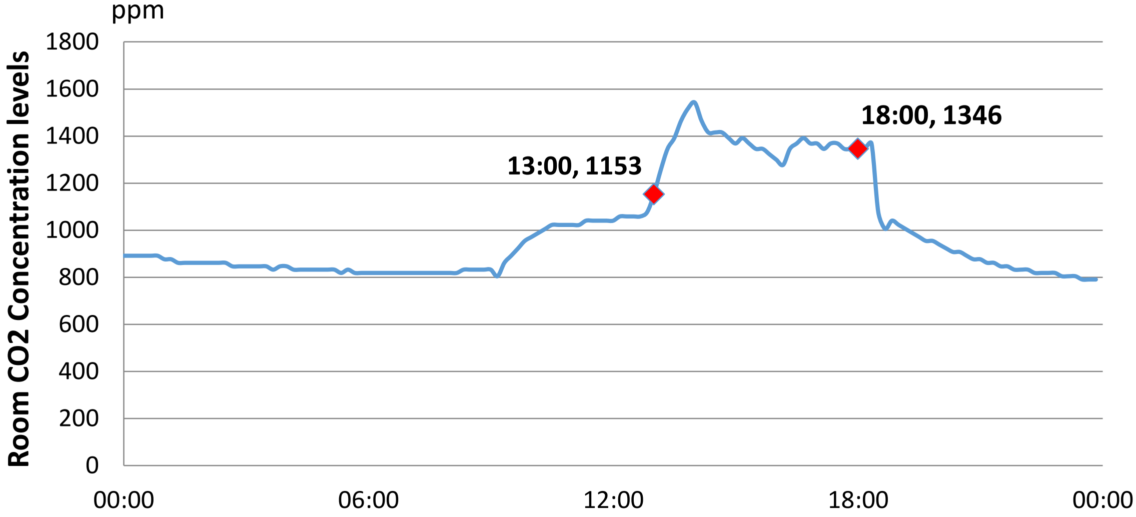

Figure 05-2 Indoor CO2 concentration levels on Thursday

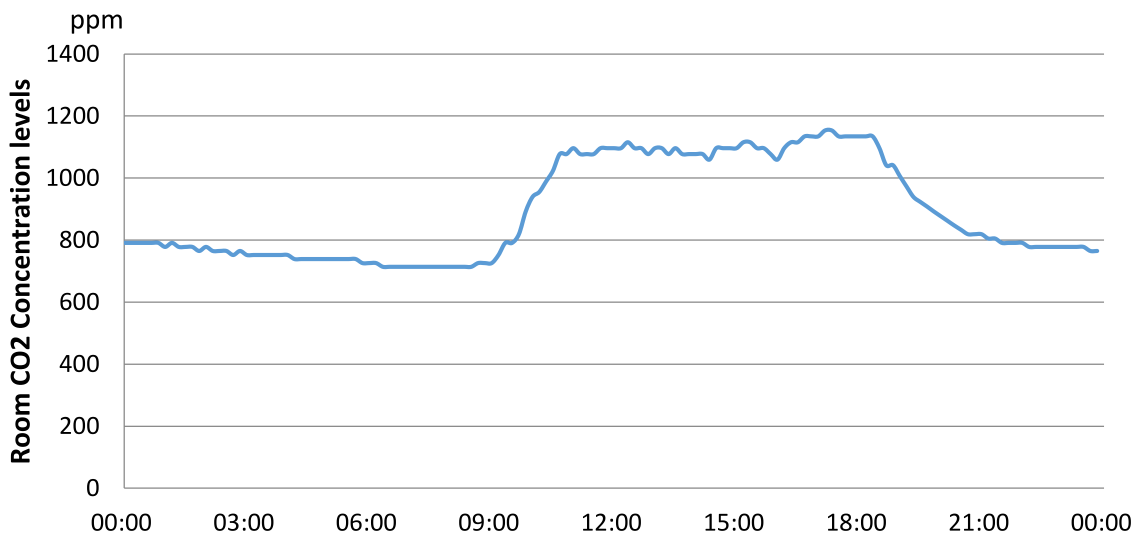

Figure 5-3 Indoor CO2 concentration levels on Friday

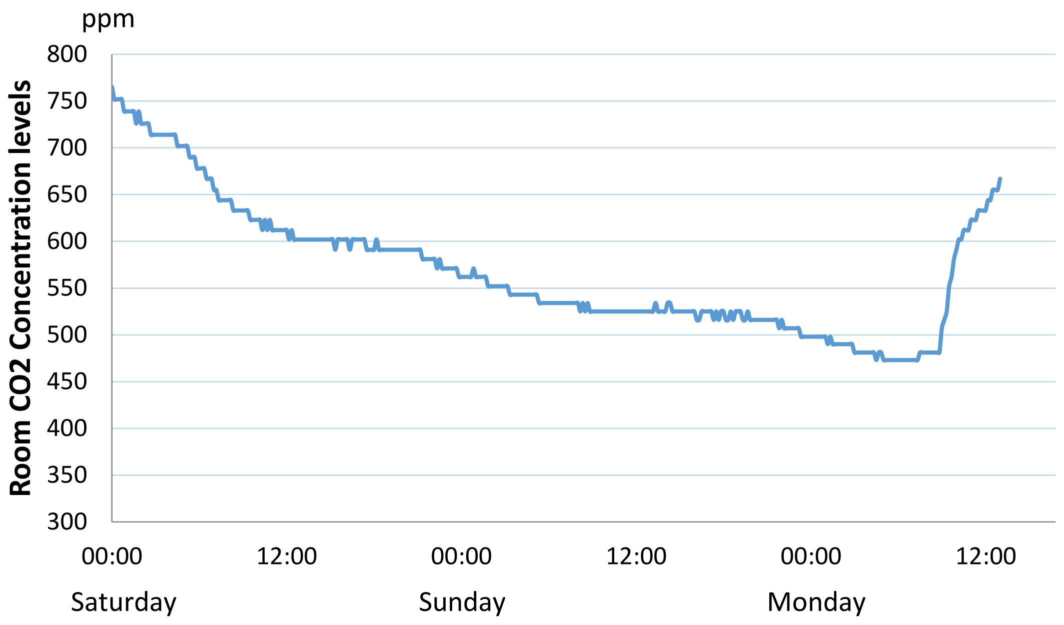

Figure 50-4 Indoor CO_2 concentration levels over weekends till Monday noon

6. Discussion

Usability and Reliability

The first step of performance evaluation is to look at whether the CO_2 IAQ monitor is fairly useable. Being useable means the measurements taken by the CO_2 monitor are reasonable, sensitive and consonant with the actual room usage facts. Otherwise, it indicates there are some problems in pervious construction or calibration process.

The indoor CO_2 concentration data collected by the monitor is reviewed from the following aspects:

- Do the CO_2 levels detected by the monitor lie in a realistic range?
- Does the data agree with actual room usage facts during the testing period including occupation time and occupancy levels, etc.?

Answers to these questions can immediately lead to a conclusion whether the IAQ monitor device is usable or not.

1) Does the data lie in a realistic range?

Theoretically, readings shown on the display can be ranging from 0 ppm up to 9999 ppm. However, concentrations of CO_2 in occupied indoor spaces are always higher than concentrations outdoors because human produces and exhale carbon dioxide. Therefore, the minimum indoor CO_2 concentration measurement by the monitor should not be less than 400 ppm. On the other hand, it is also impossible for the indoor CO_2 concentrations to be extremely high. Occupational Safety and Health Administration (OSHA) has set a permissible exposure limit (PEL) for CO2 of 5,000 ppm over an 8-hour work day. It is unlikely for CO_2 concentration of the tested office to reach such a high level, considering there is only one regular occupant in the room. According to ASHARE standards, indoor environments with good air quality require the CO2 concentrations to be maintained below 1000 ppm. Since ventilation is restricted during the testing period, low air quality was encountered when there were meetings. Studies show that adverse health effects are expected if the occupants are exposed in spaces with indoor CO_2 levels higher than 2500ppm for a long time (Satish et al). During the testing period, no health effect or evidential physical discomfort has been reported. Accordingly, the maximum indoor CO_2

concentration is estimated to be no higher than 2500 ppm. Therefore, the CO_2 concentration data collected by the monitor should be in the range of 400-2500ppm. Otherwise, it suggests the monitor is giving inaccurate readings.

In the failed first attempt at taking measurements, indoor CO_2 concentration levels measured by the monitor are unrealistically high, ranging from 3800ppm to 6000ppm. There are several reasons that can lead to unrealistic results like this. The most possible one is poor calibration. If this is identified as the problem, the device has to be recalibrated using revised calibration methods. To recalibrate the CO_2 monitor, first reload the calibration program (Sketch 1) into the Arduino, and then simply repeat the calibration steps described in Chapter 4 to determine V400ppm and V40000ppm. Again, the more efforts put into the calibration, the better the results would be. However, if the CO_2 monitor still gives unrealistic readings, then other potential problems need to be considered. Those include incorrect wiring, insufficient power supply, electrical short circuits or MG-811 sensor failure.

The testing results indicate that the minimum CO_2 concentration measured in the office room was 473 ppm while the maximum measured was 1652 ppm. In other words, the indoor CO_2 concentration levels of the office varied in a range of 450ppm -1700ppm approximately during the testing period. The result can be considered to be a reliable and convincing indication of the CO_2 concentration, as the lowest value is slightly above 400 ppm while an indoor CO_2 concentration level of 1652ppm is achievable during group meetings (more occupants in the space).

2) Does the data match the occupation time pattern and occupancy levels?

After the readings are verified to be realistic, the next step of performance evaluation is to check whether the results are able to reflect the actual facts of room usage.
The following predications can be made:

- The indoor CO_2 levels are higher during office hours. This is because of the exhalation from occupants and lack of ventilation (windows were closed);
- The indoor CO_2 levels decrease when the room is unoccupied. This is due to effects of air leakage.

The room was occupied from 9 am to 18 pm on weekdays and vacant during non-office hours and weekends. According to Figure 5-2 and Figure 5-3, it can be clearly seen that the indoor CO_2 concentrations were about 700 ppm-800 ppm at around 9am on Thursday and Friday. The concentration line then started rising up rapidly after the office user arrived. In two to three hours, the CO_2 concentration levels in the room exceeded 1000 ppm and remain higher through the rest working time. As shown in both figures, the concentration level of CO_2 in the room starts decreasing at around 6pm when the occupant left the room. It can also be found that the variation patterns of indoor CO_2 concentration on Thursday and Friday are significantly different. This is mainly because of different indoor activities taking place in the space.

Figure 5-4 shows the variation of indoor CO_2 levels during the weekends when there was no occupants in the office and the café. It can be clearly seen that the CO_2 levels were decreasing constantly over the weekends and reached the lowest point to about 450ppm at around 9am on Monday. After that the room was reoccupied and the concentration line started to rise up again, as a result of carbon dioxide production by occupants.

Apart from showing time period during which the office was used, the CO_2 concentration monitor is also able to indicate the number of occupants in the office. As can be read from Figure 5-2 and Figure 5-3, the indoor CO_2 concentrations were normally ranging between levels of 1000-1200ppm during most of the office hours except from the Thursday afternoon (see Figure 5-2). As can be found from the plot, the indoor CO_2 levels were evidentially higher between around 13:00-18:30 in Thursday afternoon in comparison to the rest working time. The data shows that the concentration reached a peak value of 1516 ppm at around 2 pm. Interview with the office user has confirmed that there was a meeting taken place during the period which involved two more adults. This evidence proved that the IAQ monitor is able to indicate the rising of indoor CO_2 concentration due to increased occupancy. On the other hand, during the night time when the office is empty, a gradual decrease in the indoor CO_2 level can be observed. This is caused by small amount of ventilation by air leakage. However, the CO_2 concentration can only fall down to 700-800ppm, because the office was re-occupied soon at around 9 am of next morning, after which it raise up again as a result of respiration from the occupants. Whereas Figure 5-4 shows that during the weekends when there was more time for the indoor/outdoor air exchange, the indoor CO_2 concentration levels dropped significantly. The lowest value was about 450ppm which is quite close to the CO_2 concentration of the outdoor air. This shows that the IAQ monitor is able to indicate the decrease of indoor CO_2 levels when there is no occupant in the space.

To summarize, the measurement by the CO_2 monitor is shown has been proven to provide a convincing indication of indoor CO_2 concentrations. The result is consistent with the actual facts of room usage including the occupation period and the occupancy levels. Therefore, it can be concluded that the CO_2 monitor is usable and reliable in indicating the indoor air quality.

Cost analysis

- **Cost overview**

Table 6-1 Prices of Required Components

Items	Liddament's research	This research project
MG-811 CO_2 sensor	£36	£36
Arduino UNO R3	£7	£17.99
Arduino Ethernet Shield (with SD card slot)	£6.12	£25.69
Four digit display	£2.14	£1.99
Additional components cost	Unknown	£81.67
Total	> £51.26	£137.19

On his website, Liddament claims that he managed to build his prototype MG-811 CO_2 monitor for a cost under £55. Whereas in this research, the total amount of money spent for building the device has reached around £137. There are many reasons responsible for this.

First, it can be clearly seen from the table that there is huge price difference between the Arduino broads used by Liddament and those used in this research project. One direct reason for this is that different Arduino brands were chosen for the researches. The original Arduino hardware is manufactured by an Italian company called 'Smart Projects', and the current prices run around £22 for a single Arduino UNO R3 and £25 for an Ethernet Shield from Amazon [Amazon]. Whereas there are also many non-official products, sometimes referred as 'Clones', produced by other manufacturers available from online source, such as eBay. Those ‘clones’, mainly of which are supplied from China, are considerably cheaper than the official products of the same models. Both the Arduino UNO R3 and Arduino Ethernet Shield use by Liddament to build his monitor are those non-official products, which cost only £7.00 and £6.12 respectively including postage to the UK according to Liddament (2013). What needs to pointed out here is, despite those non-official Arduino brand are not attested, it does not necessarily mean that they won’t be able to provide correct functions because the wiring platform of Arduino hardware is open-sourced. Since Liddament has successfully built the device, it is assumed that the capability of those modules has already been proven. .

In contrast, the Arduino brand employed in this research are official products from accredited sellers at Amazon. Such a purchase option leaded to a considerable higher cost, but the decision is made out of several reasons. One important reason is that the shopping options were limited due to the University’s purchasing policies. Secondly, as the primary objective of the research is to construct a MG-811 CO_2 indoor air quality monitor according to Liddament’s description and test its performance, the priority has been given to examining the theory rather than minimizing the expense. Therefore, the official Arduino brand with promising quality and warrants are used despite they are much more expensive. Another reason for choosing the original Arduino products is to control the delivery time under a busy research schedule, as the cheaper ones on eBay could take more than one month to be delivered to UK as they were to be shipped from China.

It is necessary for future researchers to understand the advantages and disadvantages of choosing official or non-official Arduino brands and the corresponding impacts on equipment cost and results. One initial driven motivation of Liddament’s new design is to develop a carbon dioxide IAQ monitor that is usable but at lower cost than current monitors on the market. Therefore, a sufficient amount of consideration should be taken in order to reduce the product cost as much as possible without comprising its durability and stability. There is no research so far conducted to compare the CO_2 monitor performance between devices using official Arduino broads and devices using non-official ones. Therefore future researchers are encouraged to decide whether to use official or non-official Arduino hardware subject to their own needs, budget circumstance and/or other individual concerns.

Apart from the essential parts (sensor, Arduino and display), 2/3 of the total expense were spent on some additional items. Those include two power supply chargers, a breadboard and a jump wire kit. Other necessary small items like the Micro SD card, USB connection wire were found from disused computer or personal electronic devices. Liddament did not give a detailed price list because those items can be easily found in daily working environment. However, for users don't have spare ones, those will add a small extra cost. The amount of invest on those items is much dependent on where the users get them from. This is to be explained with more details in the following sections.

For those that may be interested, these Arduino broads and all the other components needed for the monitor are also available in high-street retailers, such as Maplin, but will normally costs more than from online sellers.

- **Potential cost saving**

Although the amount of money spent in constructing the IAQ monitor in the research has reached £137, there is great space for practical cost reduction. For example, a whole pack of jump wire kit costs about £15, but only a few of those connecting wires included are used to layout the components. Therefore, if only calculating the cost for the number of wires used in one monitor, the actual amount can be ignorable. What's more, the breadboard used in the research, which costs £15, is extensively oversized than needed. This is because only the digital display unit actually sits on the base and the area it takes is quite small. Therefore, if an appropriate compact breadboard is used instead, the cost can be greatly reduced to under £2 (Amazon, 2014).

For individual researchers, some of the required items can be easily found from office or domestic supplies. For instance, the USB cable for normal printers can also be used to connect Arduino to a computer as well as providing power supply. More costs may also be avoided by using an extra SD card from old phones or power chargers for cameras and mobile phones (if the connectors are capable).

If sufficient efforts are made, the cost for a single MG-811 carbon dioxide sensor can be reduced to under £55 as Liddament claimed, or around £85 if official Arduino broads are to be used. In addition, the information from the Arduino store shows that the modules will be sold at discounted price (about 10% off) for large orders. This means if the MG-811 IAQ sensors are commercialized and put into massive production, the average cost is expected to be further decreased.

In Chapter 2.4, it has been shown that the prices for a typical NDIR CO_2 monitor runs from £230 to £440 approximately. Compare to those existing devices on the market, the price of a MG-811 carbon dioxide IAQ monitor is, with no doubt, considerably affordable. The great advantage in price makes the MG-811 IAQ monitor very competitive in the potential market, but its performance against the existing devices still needs to be further investigated.

7. Conclusion

A prototype MG811 carbon dioxide indoor air quality monitor is successfully constructed based on a new-proposed system design. The performance of the completed device is evaluated and the construction cost is also analysed. Results of the operational testing show that the IAQ monitor is usable under room conditions and able to indicate the variation of indoor CO_2 concentration levels. Measurements also prove that it operates effectively in the 400 ppm -2000 ppm range. Therefore, it can be concluded that the MG811 monitor is a useful tool for estimating the indoor air quality and ventilation rates, although it is not appropriate for academic study and settling contractual issues. In terms of marketability, it is achievable to build a single device for under £55. The cost is only about 1/4 to 1/8 of the price for a typical NDIR sensor on the market and hence makes the MG811 model quite attractive. However, it is necessary to compare the performance of both sensor models in order to determine which one is more competitive.

To give an overview of the project, the new-proposed MG811 carbon dioxide IAQ monitor design has been proven to be both useful and practical. All components are affordable and easy to get while the construction is simple and can be accomplished in domestic environment. Nevertheless, sufficient efforts are expected to be put in device calibration, which is crucial to its final success and for the future of the device. What needs to be underlined is that the work on this device is not finished and further investigation is required.

8. Further work

Although results from the operation testing have proven that the monitor is able to indicate the indoor CO_2 concentration level change trends correctly, there are still questions about the accuracy of its measurements. The easiest and most practical way of evaluating the monitor's accuracy is to compare its measurements against the measurements performed by a high-quality accurate NDIR monitor. Simply place the MG-811 monitor and the NDIR monitor in the same testing environment and compare the CO_2 concentration readings displayed. If a NDIR monitor also has a data storing function, then the CO_2 concentration data collected by both monitors need to be compared. Accordingly, a comparison resultant curve across the important 400 ppm – 2000 ppm range can be drawn. Figure 8-1 shows an example of comparison done by Liddament.

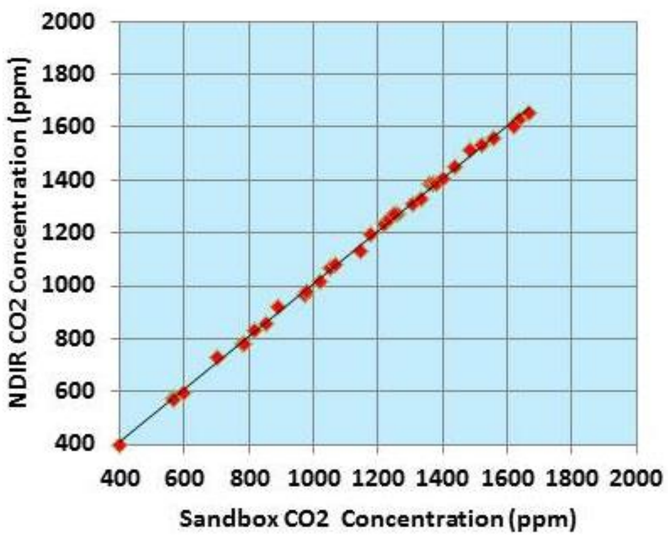

Figure 8-1 Calibrated Sandbox MG811 CO2 sensor vs NDIR (Liddament, 2013)

As can be seen from the plot above, the CO_2 concentration measurements by the MG811 sensor are quite close to those by the NDIR sensor. But again, the result is dependent on the accuracy of calibration. The calibration must be improved to industry standards if the device is to be used widely.

Liddament(2013) also underlined that the sensor accuracy may drift over time. Therefore, it is necessary to test the repeatability of the calibrated MG-811 carbon dioxide monitor device. Repeatability is defined as the ability to obtain consistent results when measuring the same part with the same measuring instrument. Where in this case, it means that whether the MG-811 CO_2 monitor will still give reliable and accurate results after being turned on/off for many times or under a constant long-time operation. This can be investigated by assessing the accuracy of the MG-811 monitor using an accurate NDIR sensor regularly, say after every 10 days, and checking if the accuracy will stay constant. It is clear that work on the MG811 monitor is nor finished and requires further efforts input.

9. Reference

Anon. *MG811 Carbon Dioxide (CO2) Sensor* [Online]. 2011 (Viewed 12 Dec 2013) Available from: http://www.spencerpages.com/wiki/index.php

Anon: How does an NDIR CO_2 Sensor Work? Available from: http://www.co2meter.com

Brown, S.K., *Indoor air quality, Australia: State of the Environment Technical Paper Series (Atmosphere)*, Department of the Environment, Sport and Territories, Canberra, 1997.

BS EN 13779:2007 *Ventilation for Non-residential Buildings – Performance*

BS EN ISO 1600-26:2012 Indoor Air part 26: Sampling strategy for carbon dioxide. British Standard Institution UK, 2012.

Chapin F.S. *Human Activity Patterns in the City: Things People Do in Time and Space*. John Wiley and Sons, New York. 1974

CIBSE Guide A: *Environmental design* The Chartered Institution of Building Service Engineers London 2006

Currie J. et al. *Does Pollution Increase School Absences?* 2007. NBER Working Paper No.13252.

Daisey J.M, Angell W.J, Apte M.G. Indoor air quality, ventilation and health symptoms in schools: an analysis of existing information. Indoor Air [Online] 2003. Volume 13: 53-64. [Viewed 11 Dec 2013] Available from: doi:10.1034/j.1600-0668.2003.00153.

Dwyer T. *Sensing the need for demand control ventilation* The CIBSE Journal CPD Programme September [Viewed 13 May 2014] Available from: http://www.cibsejournal.com/cpd/2009-09/

Emmerich. *Literature Review on CO_2-Based Demand-Ventilation Control.* National Institution of Standards and Technology [View 13 May 2014]. Available from: http://fire.nist.gov/bfrlpubs/build97/PDF/b97015.pdf

EPA: *Indoor Air Facts No.4: Sick Building Syndrome*, 1991. United Stated Environmental Protection Agency.

EPA: *Report to Congress on Indoor Air Quality, Volume II: Assessment and Control of Indoor Air Pollution.* 1989. United States Environment Protection Agency.

Federal Technology Alert: *Demand-Controlled Ventilation Using CO_2 Sensors.* [Online] Available from: http://www.habmigern2003.info/PDF/fta_co2.pdf. U.S. Department Of Energy, Federal Energy Management Program

Hooker J.D. *On the Theory of Ventilation: An Attempt to Establish a Positive Basis for the Calculation of the Amount of Fresh Air Required for an Inhabited Air-Space*. January 1875 [viewed 12 Dec 2013]. Proceedings of the Royal Society of London [Online], Available from: http://rspl.royalsocietypublishing.org/content/23/156-163/187

HSE: *How to deal with sick building syndrome (SBS): Guidance for employers, building owners and building managers 3rd*. 2000. Healthy and Safety Executive, United Kingdom.

Jones A.P. *Indoor Air Quality and Health. Atmospheric Environment* [ONLINE] Elsevier. December 1999, Volume 33. Issue 28. 4535-4564. [Viewed 09 Dec 2013] Available from: http://www.sciencedirect.com.

Jones B.M and Kirby R Indoor Air Quality in U.K. School Classrooms Ventilated by Natural Ventilation Windcatchers. *International Journal of Ventilation.* March 2012, Volume 10, No 4.

Kaneyasu K, et al. A carbon dioxide gas sensor based on solid electrolyte for air quality control. *Sensors and Actuators B: Chemical* [Online]. June 2000, Vol.66, 1-3, pp56-58. [Viewed 12 Dec 2013] Available from: http://www.sciencedirect.com

Lawrence. Demand-Controlled Ventilation and Sustainability. ASHARE Journal December 2004 [Viewed 12 Dec 2013]. Available from: https://www.ashrae.org

Lee T.J. et al. *CO_2 sensor with data logger system*. System, Applications and Technology Conference, Farmingdale, NY. IEEE Long Island, May 2012, Available from: DOI: 10.1109/LISAT.2012.6223210

LEECH J.A et al. it's about time: A comparison of Canadian and American time–activity patterns. *Journal of Exposure Analysis and Environmental Epidemiology* [Online] 2002 Issue 12, P427- 432 .Available from: http://www.nature.com/jes/journal/v12/n6/full/7500244a.html. [Accessed: 13 May 2014].

Liddament M.W. *A Guide to Energy Efficient Ventilation*. Air Infiltration and Ventilation Centre, 1996.

Lombard L.P, Ortiz J and Pout Christine A review on buildings energy consumption information. *Energy and Buildings* [Online] 2008 [Viewed 12 Dec 2013]. Vol40, Issue 3, P394-398. Available from: DOI: 10.1016/j.enbuild.2007.03.007

Mandayo G.G et al. Performance of a CO_2 Impedimetric Sensor Prototype for Air Quality Monitoring. *Sensors* May 2011, 11, 5047-5057. Available from: doi:10.3390/s110505047

Monk S. *Programming Arduino: getting started with sketches*. New York: McGraw-Hill, 2011

Requirements for Ventilation and Room-conditioning Systems. British Standard Institution , 2007.

Satish et al. *Is CO2 an Indoor Pollutant? Direct Effects of Low-to-Moderate CO2 Concentrations on Human Decision-Making Performance*. Environmental Health Perspectives [Viewed 12 Dec 2013]. Available from: http://ehp.niehs.nih.gov/wp-content/uploads/120/12/ehp.1104789.pdf

Schell M. and Inthout D. *Demand Control Ventilation Using CO_2*. [Viewed 09 Dec 2013] Available from: https://www.airtest.com/support/reference/article2.pdf.

Sundell J. (2011) *Commemorating 20 Years of Indoor Air Ventilation rates and health: multidisciplinary review of the scientific literature* [Online] Wiley Online Library [Viewed 13 May 2014] Available from: doi:10.1111/j.1600-0668.2010.00703.

Szalai A. (Ed.) *The Use of Time: Daily Activities of Urban and Suburban Populations in Twelve Countries*. Mouton, The Hague. 1972.

Vora S, Tiwari M and Singh J GSM Based Remote Monitoring of Waste Gas at Locally Monitored Gui With The Implementation of Modbus Protocol and Location Identification Through GPS. *Internal Journal of Advanced Research in Engineering and Technology* Volume 3, Issue 2, pp. 52-59. 2012.

WHO: *Air Quality Guidelines (Global Update)* 2005. Issued by World Health Organization, Geneva

Wyon D.P. The effects of indoor air quality on performance and productivity. *Indoor Air* [Online]. Wiley Online Library, 2004 [Viewed 11 Dec 13]. Available from: DOI: 10.1111/j.1600-0668.2004.00278.x

Yasuda T, Yonemura S and Tani A. Comparison of the Characteristics of Small Commercial NDIR CO2 Sensor Models and Development of a Portable CO2 Measurement Device. *Sensors.* March 2012. Vol.12 3641-3655. [Viewed 12 Dec 2013] Available from: doi:10.3390/s120303641

Zampolli S. et al. (2004) *An electronic nose based on solid state sensor arrays for low-cost indoor air quality monitoring applications*, Sensors and Actuators B: Chemical, Feb 2004 [Viewed 12 Dec 2013], Vol101, Issues 1–2, Pages 39-46, ISSN 0925-4005. Available from: *http://dx.doi.org/10.1016/j.snb.2004.02.024*.

Zhao X.Q and Zhang ZH. *Development of Remote Waste Gas Monitor System.* International Conference on Measuring Technology and Mechatronics Automation, Vol1, pp1105-1108 Changsha City, March 2010 [Viewed 12 Dec 2013] Available From: 10.1109/ICMTMA.2010.529

10. Appendices

Appendix 1: Components shopping list
Appendix 1: Sketch 1- calibration program
Appendix 2: Sketch 2- operation program

Appendix 1

Component shopping list

Items	Number required	Liddament's research		Shen's research	
		Price	Source	Price	Source
MG-811 CO_2 sensor	1	£36	Sandbox Electronics	£36	Sandbox Electronics
Arduino UNO R3	1	£7	eBay	£17.99	Amazon
Arduino Ethernet Shield with SD card slot	1	£6.12	eBay	£25.69	Cool Components
Four digit display	1	£2.14	eBay	£1.99	Amazon
9 -12V DC power supply	2	NA		£7.95 £13.99	Amazon Maplin
Breadboard	1	NA		£14.99	Maplin
220ohm resistors	At least 7	NA		£0.36 each Available as a pack of 10	Maplin
Connection jump wires	Many	NA		£14.99	Maplin
Micro SD card	1	NA		NA	
USB connector for Arduino	1	NA		NA	

1. The names of online source are linked to the store websites.
2. The prices are used as guidance. They are not guaranteed to be the lowest available.

Appendix 2

Sketch 1- Calibration program

```
/*

/*
CO2 Monitor and Data Logger
SKETCH 1 Calibration
Code by Martin Liddament

- Used to determine the CO2 sensor voltage output at 400 ppm and 40000ppm
- Follow website instructions
- Accurate calibration of the CO2 module is essential.

IMPORTANT

- For demonstration only;
- Not verified or guaranteed to be free of errors
- Not suitable for critical applications
- Use of this code is entirely at user's risk
- This code is in the public domain.
- Copyright VEETECH Ltd 2013 www.veetech.co.uk

*/
// Setting some initial parameters

int digit[4]; // stores measured voltage for display

// setting digital and analog pins plus delay time
int anode1 = A1;
int anode2 = A2;
int anode3 = A3;
int anode4 = A4;
int ledRed = 13;
int segA = 2;
int segB = 3;
int segC = 5;
int segD = 6;
int segE = 7;
int segF = 8;
int segG = 9;
int analogPin = A0;
int delayTime = 2;


void setup()
{
  Serial.begin(9600);
 // initialize the digital pins as outputs.
  pinMode(anode1, OUTPUT);
  pinMode(anode2, OUTPUT);
  pinMode(anode3, OUTPUT);
  pinMode(anode4, OUTPUT);
  pinMode(segA, OUTPUT);
  pinMode(segB, OUTPUT);
  pinMode(segC, OUTPUT);
  pinMode(segD, OUTPUT);
  pinMode(segE, OUTPUT);
  pinMode(segF, OUTPUT);
```

```
  pinMode(segG, OUTPUT);
  pinMode(ledRed, OUTPUT);
  pinMode(analogPin, INPUT);

}
void loop()
{
  int data = analogRead(analogPin); //read output
  float voltage = data/204.6;  //convert output to voltage
  int co2 = voltage * 1000;  // convert voltage to 4 digit integer

  displayAssign(co2); // assign 4 digit integer to display array

  // Reset Display to off

  digitalWrite(anode1, LOW);
  digitalWrite(anode2, LOW);
  digitalWrite(anode3, LOW);
  digitalWrite(anode4, LOW);
  digitalWrite(segA, HIGH);  // HIGH switches segment off
  digitalWrite(segB, HIGH);
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);
  digitalWrite(segE, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);

 int i =0;          // each sampling loop takes approx .1 secs
 while (i < 1000)   // to prevent lower digit lights flashing
                    // a while loop (about 10 seconds) is added
  {
 // activate display

  digitalWrite(anode1, HIGH);  // switch on anode 1
  displayDigit(0);             // display the value of the first digit
  digitalWrite(anode1, LOW);   // switch off anode 1
  digitalWrite(anode2, HIGH);
  displayDigit(1);             // display the value of the second digit
  digitalWrite(anode2, LOW);
  digitalWrite(anode3, HIGH);
  displayDigit(2);             // display the value of the third digit
  digitalWrite(anode3, LOW);
  digitalWrite(anode4, HIGH);
  displayDigit(3);             // display the value of the fourth digit
  digitalWrite(anode4, LOW);

  i ++;  // increment while loop
  }      // end while loop
 }       // end sampling loop

void displayAssign(int d)    // assigns four figure co2 conc to
individual display digits
{
  int number = d;
  for(int i = 3; i >= 0; i--)
{
  digit[i] = number % 10; // %=modulus remainder from division
  number /=10;            // integer divide
}  // end for loop
}  // enf function
```

```
void displayDigit(int dis) // assign value of each digit to the display
{
if (digit[dis] == 0)
{
disZero();
}
else if (digit[dis] == 1)
{
disOne();
}
else if (digit[dis] == 2)
{
disTwo();
}
else if (digit[dis] == 3)
{
disThree();
}
else if (digit[dis] == 4)
{
disFour();
}
else if (digit[dis] == 5)
{
disFive();
}
else if (digit[dis] == 6)
{
disSix();
}
else if (digit[dis] == 7)
{
disSeven();
}
else if (digit[dis] == 8)
{
disEight();
}
else
{
disNine();
}
}


void disZero()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segD, LOW);
  digitalWrite(segE, LOW);
  digitalWrite(segF, LOW);

  delay(delayTime);

  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
```

```
  digitalWrite(segD, HIGH);
  digitalWrite(segE, HIGH);
  digitalWrite(segF, HIGH);

}


void disOne()

{
  digitalWrite(segB, LOW); // LOW switches segment on
  digitalWrite(segC, LOW);
  delay(delayTime);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
}

void disTwo()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segG, LOW);
  digitalWrite(segE, LOW);
  digitalWrite(segD, LOW);

  delay(delayTime);

  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segG, HIGH);
  digitalWrite(segE, HIGH);
  digitalWrite(segD, HIGH);

}

void disThree()
{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segG, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segD, LOW);

  delay(delayTime);

  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segG, HIGH);
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);

}

void disFour()

{

  digitalWrite(segB, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segF, LOW);
```

```
  digitalWrite(segG, LOW);

  delay(delayTime);

  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);

}

void disFive()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segC, LOW);
  digitalWrite(segD, LOW);
  digitalWrite(segF, LOW);
  digitalWrite(segG, LOW);

  delay(delayTime);

  digitalWrite(segA, HIGH);
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);

}

void disSix()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segC, LOW);
  digitalWrite(segD, LOW);
  digitalWrite(segE, LOW);
  digitalWrite(segF, LOW);
  digitalWrite(segG, LOW);

  delay(delayTime);

  digitalWrite(segA, HIGH);
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);
  digitalWrite(segE, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);

}

void disSeven()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segF, LOW);

  delay(delayTime);
```

```
  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
  digitalWrite(segF, HIGH);

}

void disEight()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segD, LOW);
  digitalWrite(segE, LOW);
  digitalWrite(segF, LOW);
  digitalWrite(segG, LOW);

  delay(delayTime);

  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);
  digitalWrite(segE, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);

}

void disNine()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segF, LOW);
  digitalWrite(segG, LOW);

  delay(delayTime);

  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);

}
```

Appendix 3

Sketch 2- Operation Program

```
/*

VEETECH CO2 Monitor and Data Logger
SKETCH 2 Operation
Code by Martin Liddament

Used to determine:
- CO2 Concentration;
- Display concentration on 4-digit display in ppm;
- Record concentration on an SD Card at 10 minute intervals.

IMPORTANT

- For demonstration only;
- Not verified or guaranteed free of errors;
- Not suitable for critical applications;
- Use of this code is entirely at user's risk;
- Accurate calibration of the CO2 module is essential.

This code is in the public domain.
Copyright VEETECH Ltd 2013 www.veetech.co.uk.

*/

// Setting some initial parameters
#include <SD.h>
#include <math.h>
int digit[4]; // stores the co2 concentration data for display
File co2Data;

int co2 = 9999; // co2 is the co2 concentration. Preset value for code
checking
int cycleTotal = 0; // Total number of 10 second cycles from start.
int timeSpan = 0; //  Time record for SD Card Recording
int timePeriod = 0;

int anode1 = A1; //   The display anodes - see display datasheet
int anode2 = A2;
int anode3 = A3;
int anode4 = A4;
int segA = 2;    // display segments - see display datasheet
int segB = 3;
int segC = 5;
int segD = 6;
int segE = 7;
int segF = 8;
int segG = 9;

// analog input

int analogPin = A0; // voltage input from sensor

// Pins 4,10,11,12 and 13 are used by the SD Card and Ethernet Shield

// setting display illumination time (msecs)
int delayTime = 2;

// Setting Sensor Calibration Constants
```

```
 float v400ppm = 4.535;   //MUST BE SET ACCORDING TO CALIBRATION
 float v40000ppm = 3.206; //MUST BE SET ACCORDING TO
CALIBRATION``````````````````````
 float deltavs = v400ppm - v40000ppm;
 float A = deltavs/(log10(400) - log10(40000));
 float B = log10(400);

void setup()
{
  Serial.begin(9600);
 // initialize pins as outputs and inputs.
  pinMode(anode1, OUTPUT);
  pinMode(anode2, OUTPUT);
  pinMode(anode3, OUTPUT);
  pinMode(anode4, OUTPUT);
  pinMode(segA, OUTPUT);
  pinMode(segB, OUTPUT);
  pinMode(segC, OUTPUT);
  pinMode(segD, OUTPUT);
  pinMode(segE, OUTPUT);
  pinMode(segF, OUTPUT);
  pinMode(segG, OUTPUT);
  pinMode(analogPin, INPUT);
  pinMode(10, OUTPUT); //write to sd card

 // Initialize sd card
 if (!SD.begin(4)) {
        Serial.println("initialization failed!");
        return;
  }
  Serial.println("initialization done.");

}
void loop()
{

// Read co2 data from sensor

  int data = analogRead(analogPin); //digitise output from c02 sensor
  float voltage = data/204.6;       //convert output to voltage

// Calculate co2 from log10 formula (see sensor datasheet)

  float power = ((voltage - v400ppm)/A) + B;
  float co2ppm = pow(10,power);
  co2 = co2ppm;

// Assign co2 value to digital display

  displayAssign(co2); // assign 4 digit integer to display array


 // Reset Display to off

  digitalWrite(anode1, LOW); //Low switches anode off
  digitalWrite(anode2, LOW);
  digitalWrite(anode3, LOW);
  digitalWrite(anode4, LOW);

  digitalWrite(segA, HIGH);  // HIGH switches segment off
```

```
  digitalWrite(segB, HIGH);
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);
  digitalWrite(segE, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);

// Print output to serial monitor (if connected)

Serial.print (cycleTotal);
Serial.print("\t");
Serial.println (co2);

// check on time for SD Card recording
timePeriod = timePeriod + 1;
if (timePeriod == 1)
{
// write to sdcard

// Open SD Card for writing
co2Data = SD.open("co2Data.txt", FILE_WRITE);
co2Data.print (timeSpan); // timePeriod of 6 gives 1 approx 1 minute
recording
co2Data.print ("\t");
co2Data.println (co2);

// close the file:
co2Data.close();

// write to serisl monitor (if connected)
Serial.print (cycleTotal);
Serial.print("\t");
Serial.println (co2);

timeSpan = timeSpan + 1;
}
else if (timePeriod ==60) //set this according to recording period
required
{
timePeriod = 0;
}

 int sdTiming =0;           // Each sampling loop takes approx .08 secs.
 while (sdTiming < 1218)    // To prevent lower digit lights flashing
                            // a while loop (of about 10 seconds) is
added.
                            // Parameter sdTiming must be adjusted to make
this
                            // 10 second loop as accurate as possible
                            // otherwise the sdcard recording time
                            // will drift.
  {

 // activate display

  digitalWrite(anode1, HIGH);  // switch on anode 1
  displayDigit(0);             // display the value of the first digit
  digitalWrite(anode1, LOW);   // switch off anode 1
  digitalWrite(anode2, HIGH);
  displayDigit(1);             // display the value of the second digit
  digitalWrite(anode2, LOW);
```

```
  digitalWrite(anode3, HIGH);
  displayDigit(2);             // display the value of the third digit
  digitalWrite(anode3, LOW);
  digitalWrite(anode4, HIGH);
  displayDigit(3);             // display the value of the fourth digit
  digitalWrite(anode4, LOW);

  sdTiming ++;  // increment while loop
  }      // end while loop

cycleTotal ++;
} // end sampling loop

void displayAssign(int d)    // assigns four figure co2 conc to
individual display digits
{
  int number = d;
  for(int i = 3; i >= 0; i--)
{
  digit[i] = number % 10; // %=modulus remainder from division
  number /=10;            // integer divide
}  // end for loop
}  // enf function


void displayDigit(int dis) // assign value of each digit to the display
{
if (digit[dis] == 0)
{
disZero();
}
else if (digit[dis] == 1)
{
disOne();
}
else if (digit[dis] == 2)
{
disTwo();
}
else if (digit[dis] == 3)
{
disThree();
}
else if (digit[dis] == 4)
{
disFour();
}
else if (digit[dis] == 5)
{
disFive();
}
else if (digit[dis] == 6)
{
disSix();
}
else if (digit[dis] == 7)
{
disSeven();
}
else if (digit[dis] == 8)
{
```

```
disEight();
}
else
{
disNine();
}
}


void disZero()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segD, LOW);
  digitalWrite(segE, LOW);
  digitalWrite(segF, LOW);

  delay(delayTime);

  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);
  digitalWrite(segE, HIGH);
  digitalWrite(segF, HIGH);
}

void disOne()

{
  digitalWrite(segB, LOW); // LOW switches segment on
  digitalWrite(segC, LOW);
  delay(delayTime);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
}

void disTwo()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segG, LOW);
  digitalWrite(segE, LOW);
  digitalWrite(segD, LOW);
  delay(delayTime);
  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segG, HIGH);
  digitalWrite(segE, HIGH);
  digitalWrite(segD, HIGH);

}

void disThree()
{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segG, LOW);
```

```
  digitalWrite(segC, LOW);
  digitalWrite(segD, LOW);
  delay(delayTime);
  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segG, HIGH);
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);
}

void disFour()

{

  digitalWrite(segB, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segF, LOW);
  digitalWrite(segG, LOW);
  delay(delayTime);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);
}

void disFive()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segC, LOW);
  digitalWrite(segD, LOW);
  digitalWrite(segF, LOW);
  digitalWrite(segG, LOW);
  delay(delayTime);
  digitalWrite(segA, HIGH);
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);
}

void disSix()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segC, LOW);
  digitalWrite(segD, LOW);
  digitalWrite(segE, LOW);
  digitalWrite(segF, LOW);
  digitalWrite(segG, LOW);
  delay(delayTime);
  digitalWrite(segA, HIGH);
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);
  digitalWrite(segE, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);
}

void disSeven()
```

```
{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segF, LOW);
  delay(delayTime);
  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
  digitalWrite(segF, HIGH);
}

void disEight()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segD, LOW);
  digitalWrite(segE, LOW);
  digitalWrite(segF, LOW);
  digitalWrite(segG, LOW);
  delay(delayTime);
  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
  digitalWrite(segD, HIGH);
  digitalWrite(segE, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);
}

void disNine()

{
  digitalWrite(segA, LOW); // LOW switches segment on
  digitalWrite(segB, LOW);
  digitalWrite(segC, LOW);
  digitalWrite(segF, LOW);
  digitalWrite(segG, LOW);
  delay(delayTime);
  digitalWrite(segA, HIGH);
  digitalWrite(segB, HIGH); // High switches segment off
  digitalWrite(segC, HIGH);
  digitalWrite(segF, HIGH);
  digitalWrite(segG, HIGH);
}
```