

Air Quality Monitoring System (Front-End) using FPGA

Mohammad Shamsul Bin Taip
Faculty of Electrical Engineering
Universiti Teknologi MARA (UiTM)
Shah Alam Malaysia

ABSTRACT

This technical paper presents development of Air Quality Monitoring System that might helps government or any industries to monitor the amount of poisonous gases or air pollutants emitted in the air in order to have a safe environment. This system focuses on the development of Graphical User Interface (GUI) menu for user to activate certain sensors and display the results on FPGA (Field Programmable Gate Array) Development Board. The output from sensor circuit acts as input to the FPGA board. The FPGA, which is programmed by using VHSIC hardware description language (VHDL) code, displays the sensor's output thru LED and seven segment display.

Keywords: Air Quality Monitoring System, Graphical User Interface (GUI), Sensor, FPGA (Field Programmable Gate Array)

I. INTRODUCTION

Nowadays, the air quality is not consistent due to pollution and ozone depletion. Information regarding this issue should be made available to public because if they know the level of poisonous gas or air pollutants they might expose to, they will prohibit themselves to go out. Some of the health issues are related with the air quality such as asthmatic people will have problem if they are exposed in hazy environment or air that pollutes. In our country, the Air Pollution Index (API) is made known to public when there is serious haze problem and this information is only available from Department of Environment. By having a monitoring system, people will know the amount of "poisonous" gas such as carbon monoxide, sulfur and etc. in air due to emission from vehicles and factories. Factories can also use this monitoring system to control the emission of these gases according to the permitted value or standards. Thus, public can avoid themselves from being exposed to the hazardous environment.

Due to the nature of this system, which is related to air quality, temperature sensor and humidity sensor are used to measure the variables. Temperature plays an important role in keeping the earth balance and in order in term of climate. Changes in temperature which increase gradually over the years will have a significant effect on human physical health in a long run. Such increase will lead to a major climate change. Thus, increase the incidence of heat waves and extreme hot phenomenon. Extreme heat waves during the summer while less cold spells produced during the rainy season are among the effects experienced by the affected countries due to the increase in temperature.

Humidity is defined as the amount of moisture in the air measured as the percentage of total amount of moisture the air can hold at a particular temperature. The ideal relative humidity for a typical home is between 35% and 45%, too much above or below these levels will lead to problems. In concern of physical health, a relative humidity that is too low could lead to sore throats, sinus congestion and skin disorders for the occupants. At high relative humidity, it does not affect the human health directly but indirectly. At this high level, it can cause the growth of micro-organisms (mold, etc) – due to condensation inside the building shell – and increase chemical off-gassing, which can affect human health indirectly.

Due to the above stated reasons, the temperature and humidity of the surrounding are important issues in air quality monitoring system. Thus, the measurement of temperature and humidity are also included in the objectives of this work.

In this paper, a system that can activate a thermal sensor from a Graphical User Interface (GUI) menu, read the temperature value, and then display the results using FPGA (Field Programmable Gate Array) development board, is proposed here. The GUI menu is designed using JAVA because of it is

widely used especially in FPGA application [1] and FPGA is known to be a rapid prototype chip [2].

The description of the system is provided in the next section. The circuit and the algorithm for developing the system are described in section 3. The results and discussion are covered in Section 4. Lastly, Section 5 concludes the paper.

II. LITERATURE REVIEW

The proposed system provides GUI menu for user to activate the humidity and ozone sensors and display the results on LED and seven segment LED. Since ozone sensors are not costly, switches at Field Programmable Gate Array (FPGA) board are used to imitate the function of real sensors. This is to ensure that the developed system functions accordingly.

The complete Air Quality Monitoring System can be divided into two parts: front-end and back-end systems. The work will only concentrate on the front-end system, which consists of the GUI menu from the PC, interfacing circuit, FPGA development board and Sensor circuit as shown in Figure 1.

Graphical User Interface (GUI) menu is developed to make it possible for the users to choose type of information they want to have. The instructions are activated from the GUI menu and they are sent to FPGA for processing through the communication gateway interfacing between the FPGA and the computer. The FPGA as the controller circuit will select which sensor will be triggered. The results processed in FPGA are then displayed through the LEDs or seven segment displays on the board.

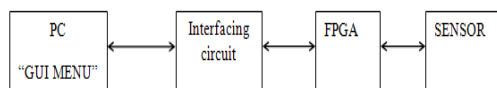


Figure 1 Air Quality Monitoring System overview

A. Graphic User Interface (GUI)

GUI for this project is developed using programming language, JAVA. JAVA is developed by Sun Microsystems and it can be used to develop many types of applications. There are simple text-based programs called console applications. To build this GUI menu, BlueJ software is used to write and compile the language code. These programs just

support text input and output to the computer screen. The menus, toolbars, buttons, scroll bars, and other controls will depend on the computer mouse for input [3][4].

GUI menu generated from the BlueJ software displays temperature, humidity, ozone buttons and some colors as indicator for Air Pollution Index (API).

B. FPGA: Altera Development Board DE2-70

FPGA is a RAM based array device from Xilinx, Altera, and others that has very fast growing technology that formed the basis for high-performance and affordable computing systems. FPGA based logic simulators can emulate complex logic designs at clock speeds of several orders of magnitude faster than even accelerated software simulators, while FPGA-based prototyping systems provide great flexibility in rapid prototyping and system verification[2]. However, besides FPGA pin configuration, existing FPGA-based systems also meet the problem of improving the routability of interconnect networks in the architecture design. A promising new alternative technology has emerged that enables designers to utilize a large FPGA that contains both memory and logic elements along with an intellectual property (IP) processor core to rapidly implement a computer and custom hardware for embedded systems[5].

In this project, Altera Development Board DE2-70 with FPGA Cyclone II, as shown in Figure 2, is used as the controller and the processing unit to the system. The instructions from the GUI menu will be processed according to the algorithm developed in VHDL language and the algorithm will be synthesized and implemented on the FPGA to perform as the controller and processing unit of the system. The display components from the FPGA board used in this project are LED, seven segment, expansion header and USB blaster.

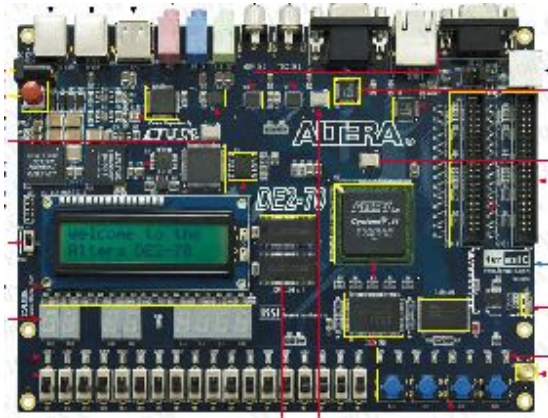


Figure 2 FPGA Board

The functions of the components used from the board are briefly described here:

1. LED – present the digital value output from sensor circuit in binary. LED lights up indicates as logic '1' and logic '0' when LED lights off.
2. Seven Segment – present the decimal value for current temperature and humidity. For this test, only three seven segment will be used.
3. USB- Blaster – to send the data from software to the FPGA board.
4. Header Expansion – receive a signal from the sensor circuit.

C. Sensor Circuit

The environment temperature or the humidity of the air will be measured from thermal and humidity sensor. Figure 3 shows the sensor type and its circuit in schematic diagram. For the sensor component, HSM-20G module is chosen because both humidity and thermal sensor are provided in the same circuit. This module uses standard input voltage i.e. +5V which provided from a voltage regulator. The output from this module presents the output value from the sensors in terms of voltage and it will be converted to digital value by ADC 0804.

The Environmental Relative Humidity sensor (HSM-20G) is used to detect humidity and temperature. It can measure the relative humidity level from 20 to 95 percent and from 0°C to 50°C for temperature [6]. By using this module, we do not have to use separate sensor and circuit for sensing temperature and humidity. The manual switch is used to activate any of two sensors from the sensor module. Figure 5

show the Environmental Relative Humidity sensor (HSM-20G).

The ADC0804 is a CMOS 8-bit successive approximation A/D converter that uses a differential potentiometric ladder. This converter is designed to allow operation derivative control bus with TRI-STATE output latches directly driving the data bus. The A/D appears like memory locations or I/O ports to the microprocessor and no interfacing logic is needed. Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution. This means that ADC 0804 converts the reading into digital value.

In order to get constant voltage supplied 5V to the ADC0804 IC, a simple voltage regulator circuit is built using voltage regulator IC - LM78M05 and two capacitors 1000 μ F. Figure 4 shows the voltage regulator developed for this system.

This circuit also has eight LEDs component to detect each digital value. This digital value is obtained from the FPGA board through the expansion header connector which is connected to the board.

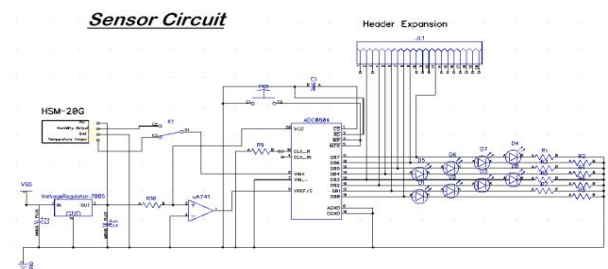


Figure 3 Schematic sensor circuit.

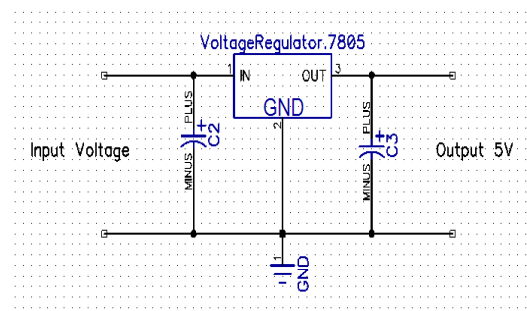


Figure 4 Voltage Regulator

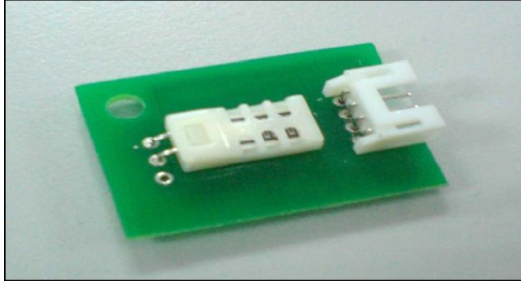


Figure 5 Environmental Relative Humidity sensors (HSM-20G)

D. Communication Interfacing

This part is developed by using JAVA programming language. The serial communication port, USB- RS 232, is used to interface the PC and the FPGA board. When the button on the GUI menu is activated, the signal will be sent to the FPGA board. This work has been based on a platform independent virtual interface as suggested in [5].

The virtual interface has two functions: Firstly, it provides platform independent API calls to prepare a configuration interface for the FPGAs to be configured. Secondly, it provides platform independent API calls for communication (reading and writing) between the software implemented on CPU and the hardware implemented on FPGA. The software (Java program) only communicates with the virtual software interface, whereas the hardware (FPGA) only communicates with virtual hardware [7].

III. METHODOLOGY

The project will be divided into a series of milestones. Each milestone will require design, simulation, or verification for one or more components of the overall system. This project is intended to introduce or reinforce the following concepts:

A. GUI development

To develop GUI menu using JAVA, one needs to understand the Java event-driven programming concepts, terminology, and available Swing controls. The fundamentals of designing, implementing, and distributing wide variety of Java GUI applications are very important to create this GUI menu. In such

applications, users interact with a set of visual controls (buttons, labels, text boxes, tool bars, and menu items) to make an application do its required tasks. The applications have a familiar appearance to the user. Figure 6 shows the process to create the GUI by JAVA.

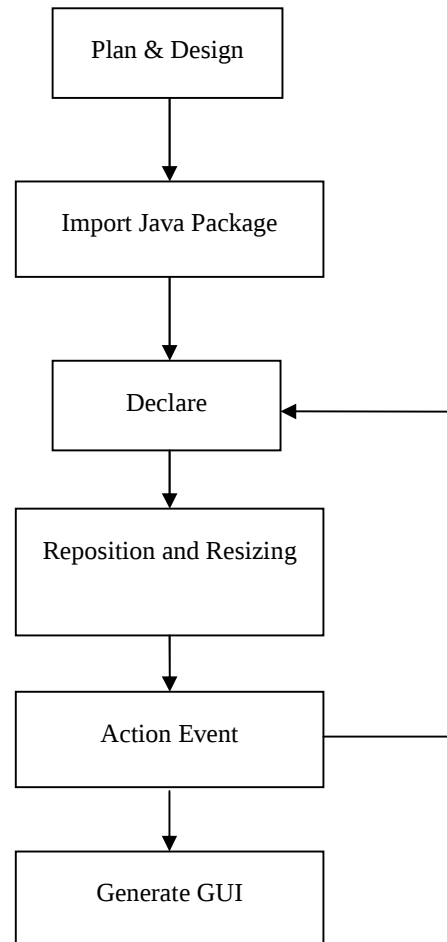


Figure 6 Flow chart to create the GUI menu by JAVA
For this initial GUI menu development, designer needs to plan and design the GUI menu by sketching. The number of button, label, and tool bars is determined at earlier plan.

In programming process, the designer needs to import java package in order to enable the selected function. For example, **import java.awt.event.*;** is a package to indicate that the user wants some action to occur like button exit. The other packages also are important to make the GUI menu become real.

After importing the package, the declaration of object is needed before repositioning or arrangement of

object in a grid like button, text and etc. Example of declaration is

```
static JButton temperatureButton = new JButton();
```

These declarations will be prefixed with the keyword static for temperature button. After declaration, the button or other object can be arranged like size of button, colour, and position and so on. The example of reposition of the temperature button are shown below.

```
temperatureButton.setIcon(temperature);
temperatureButton.setBackground(Color.CYAN);
gridConstraints.gridx = 0;
gridConstraints.gridy = 1;
getContentPane().add(temperatureButton,
gridConstraints);
```

One of the important parts in developing the GUI menu is action event. One of examples is exit button which can close the window after the user click at the exit button..

B. System Testing

The Air Quality Monitoring System is tested at several levels. First, the sensor circuit is tested for its functions. Since the circuit sensor detects temperature and humidity environment, heat source is applied near to the circuit and the output voltage measured using voltmeter should show increasing value. The results can be read from the LEDs where the LED presents binary value of the voltage and the value should be increased by bits.

After testing the sensor circuit, the interfacing between circuit sensor with FPGA board has to be checked. This indicates the VHDL program used to control the board and sensors are functioning as expected. The VHDL code is compiled and synthesized using Quartus II 7.2 software. The code includes the method on how the output binary from circuit sensor is converted to decimal and the algorithm used to develop this system. The Quartus II 7.2 software generates *sof* file for the synthesized circuit and the file is downloaded to the FPGA through the USB-Blaster. Joint Test Action Group (JTAG) mode is used to configure FPGA devices. The JTAG interface also allows boundary-scan test using third-party boundary scan tools [8][9].

Figure 8 shows the process on how to separate each digit of decimal output and it consists of several operations such as multiplication, division and subtraction. The digit is converted to seven-segment code using the decoder as show in Figure 9. Figure 10 shows the hardware part and the application software needed to program the hardware.

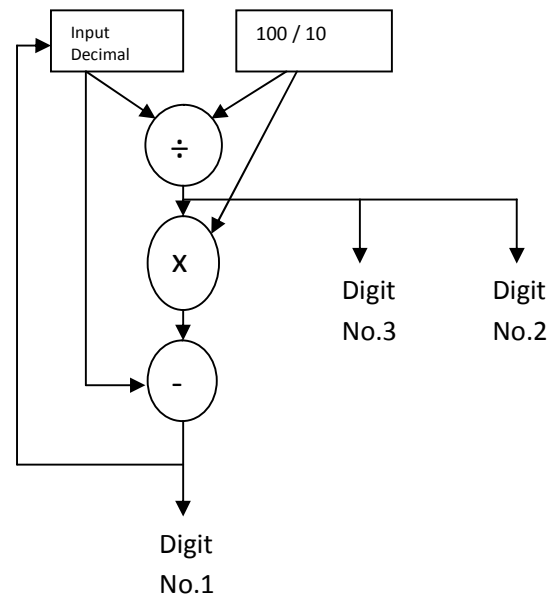


Figure 8 Separate the decimal

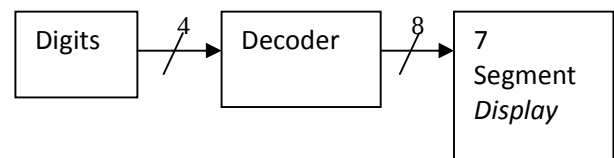


Figure 9 Locating the Decimal

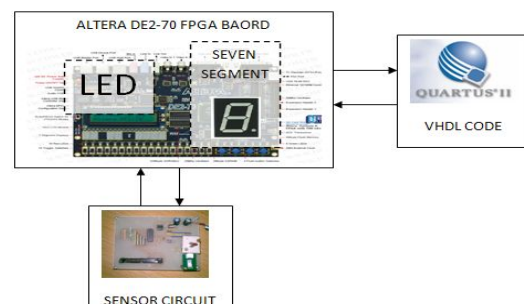


Figure 10 Test using VHDL

IV. RESULTS AND DISCUSSION

A. GUI menu

Figure 11 shows the GUI menu that is successfully created using BlueJ software and it contains graphical or special graphical element along with text labels, which enables user to interact with the program. The menu also includes some coloured indicator to show the API status which will be developed in the future work. The *exit button* functions as an exit from the GUI menu.

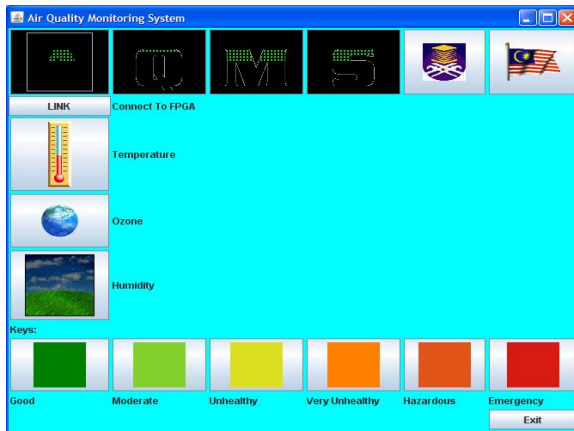


Figure 11 GUI menu

B. Testing Circuit

Figure 12 shows one of the results for testing circuit sensor. There is change by bits if the temperature and humidity changes and they are shown by the LEDs.



Figure 12 LED results.

The output voltages at each sensor are also taken as shown in Table 1 and Table 2 for output voltage at current temperature and current humidity.

Current Temperature (°C)	Output Voltage (V)
T ₁ = 25	V ₁ = 3.78
26	3.83
27	3.88
30	4.02
35	4.27
40	4.51
45	4.78
T ₂ = 50	V ₂ = 5.00

Table 1 Results for Current Temperature and Output voltage

For the value in the range between 26 °C to 45 °C , the output value is calculated using the formula below which indicates that by increasing 1 °C, the voltage is increased by average of 49mV.

$$1^{\circ}\text{C} = \frac{V_2 - V_1}{T_2 - T_1} = \frac{5.0 - 3.78}{50 - 25} = 49\text{mV}/^{\circ}\text{C}$$

Current Humidity (%)	Output Voltage (V)
RH ₁ = 76	V ₁ = 2.70
77	2.821
78	2.942
79	3.063
80	3.184
85	3.789
90	4.394
RH ₂ = 95	V ₂ = 5.00

Table 2 Results for Current Humidity and Output voltage

Meanwhile for the humidity, the value in the range of 77% to 90% , the average voltage for each 1%RH is 0.121V

$$1\% = \frac{V_2 - V_1}{RH_2 - RH_1} = \frac{5.0 - 2.7}{95 - 76} = 0.121\text{V}/\text{RH}\%$$

C. Interfacing with VHDL

The testing on FPGA board using VHDL gives some interesting output. The temperature output is shown in binary value at LEDs and decimal value for current humidity at seven segments. Since the output binary from ADC0804 is large , so to be able to display in

LED and seven segments, the data are normalized by subtracting 140 from the original value.

D. Interfacing with JAVA

The JAVA and the FPGA interfacing part are still under development. By having correct interfacing, the system is expected to be able to activate the sensor from the GUI menu.

V. CONCLUSION

This project shows that a monitoring system can be developed using JAVA and FPGA. For Java applications, each interactive GUI component needs to register to an event listener and needs to implement an event handler. BlueJ is able to handle hooking up the event listener for language programmers, so the JAVA creator can concentrate on implementing the actual business logic that should be triggered by the event. While, the FPGA can be used as a control device which can activate the system and it also can be used as a processing unit to the system.

In future, work will include completing the implementation of all trap functions and a bigger subset of the Java programming communicates interfacing. With these prerequisites, more complex applications will be possible.

ACKNOWLEDGEMENT

The author would like to express his gratitude and much appreciation to his project supervisor, Dr. Azilah Bt Saparon for her supervision, guidance, invaluable advice and suggestions upon completing this project.

REFERENCE

[1] Hutchings, B.; Nelson, B., "Developing and debugging FPGA applications in hardware with JHDL," *Signals, Systems, and Computers, 1999. Conference Record of the Thirty-Third Asilomar Conference on*, vol.1, no., pp.554-558 vol.1, 1999

[2] Li, J.; Cheng, C.-K., "Routability improvement using dynamic interconnect architecture," *FPGAs for Custom Computing Machines, 1995. Proceedings. IEEE Symposium on*, vol., no., pp.61-67, 19-21 Apr 1995

[3] Weimin Zhang; Devgan, S.S., "Java based graphical user interface for Livingston," *Southeastcon 2000. Proceedings of the IEEE*, vol., no., pp.197-200, 2000

[4] Chen, J.; Subramaniam, S., "A GUI environment to manipulate FSMs for testing GUI-based applications in Java," *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, vol., no., pp. 10 pp.-, 3-6 Jan. 2001

[5] El Medany, W.M., "FPGA remote laboratory for hardware e-learning courses," *Computational Technologies in Electrical and Electronics Engineering, 2008. SIBIRCON 2008. IEEE Region 8 International Conference on*, vol., no., pp.106-109, 21-25 July 2008

[6] Chia-Yen Lee; Gwo-Bin Lee, "MEMS-based humidity sensors with integrated temperature sensors for signal drift compensation," *Sensors, 2003. Proceedings of IEEE*, vol.1, no., pp. 384-388 Vol.1, 22-24 Oct. 2003

[7] Haubelt, C.; Schlichter, T.; Keinert, J.; Meredith, M., "SystemCoDesigner: Automatic design space exploration and rapid prototyping from behavioral models," *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, vol., no., pp.580-585, 8-13 June 2008

[8] Be Van Ngo; Law, P.; Sparks, A., "Use of JTAG boundary-scan for testing electronic circuit boards and systems," *AUTOTESTCON, 2008 IEEE*, vol., no., pp.17-22, 8-11 Sept. 2008

[9] de la Torre, E.; Garcia, M.; Riesgo, T.; Torroja, Y.; Uceda, J., "Nonintrusive debugging using the JTAG interface of FPGA-based prototypes," *Industrial Electronics, 2002. ISIE 2002. Proceedings of the 2002 IEEE International Symposium on*, vol.2, no., pp. 666-671 vol.2, 2002