FAN CONTROL USING ARDUINO MEGA 2560

A Final Project
Presented to
The Engineering Faculty

By
Edwin Gunstanto
ID No. 002200800002

In partial fulfilment
Of the requirements of the Degree
Bachelor of Science in Electrical Engineering

President
University
January 2012
DECLARATION OF ORIGINALITY

I declare that this final project, entitled “Fan Control Using Arduino Mega 2560” is, to the best of my knowledge and belief, an original piece of work that has not been submitted, either in whole or in part, to another university to obtain a degree.

Cikarang, Indonesia, January 4, 2012

_______________________________
Edwin Gunstanto
ID No. 00220020008
FAN CONTROL USING ARDUINO MEGA 2560

By

Edwin Gunstanto
ID No. 002200800002

Approved by:

________________________________________
Dr.-Ing. Erwin Sitompul
Final Project Advisor

________________________________________
Dr.-Ing. Erwin Sitompul
Acting Dean of Faculty of Engineering

________________________________________
Dr.-Ing. Erwin Sitompul
Program Head of Electrical Engineering
The fear of the LORD is the beginning of knowledge: but fools despise wisdom and instruction.

(Proverb 1:7)
ACKNOWLEDGMENT

First of all, I would like to give my best gratitude to my God, Jesus Christ for His blessing, directions, and love so that I could be in the final term of my undergraduate program at PresidentUniversity, which is the Final Project term.

This final project is dedicated to all people who have given me a lot of support and motivation. I would thank my beloved parents, Mr. Trisnahadi Tanujaya and Mrs. Yuyut Indrawati. Through my parent’s motivation for me, I put my best effort in working on this final project. They have taught me to survive and struggle to get the best result in every part of my life.

I would also thank my final project advisor, Dr.-Ing. Erwin Sitompul who is also acting as Dean of Faculty of Engineering, for his precious supervision, advice, and guidance. By his support for me, I am able to obtain plentiful amount of knowledge in Electrical Engineering through the process of preparing the final project.

I wish to deliver my great thanks to all my best friends who have supported me with a lot of prayer, attention, and care. I am also grateful to all of seniors in PresidentUniversity, who gave me support during the final project working process. Last but not least, I thank everyone, whose name could not be mentioned one by one in this final project, for their assistance to me.

January 2012

The author
ABSTRACT

There are some tools that can be used to influence the air temperature to be more comfortable, such as: fan, air conditioner, etc. In the industrial case, fan is preferred to be used as a tool to improve the working comfort in the worker’s area because the price of a fan is cheaper than the price of an air conditioner. During the internship conducted by the author in an automotive company, many of the fans are broken for so many times due to extensive use without rest.

From these points of view, the idea to design and to implement a “Fan Control Using Arduino Mega 2560” came up. This final project emphasizes in designing the hardware and the software that will work as an integrated system in the form of a circuit that will control the fan. Through the controller, the auto on-off fan can be realized based on human presence. An infrared sensor is used to control the state of the fan. The fan will be on if there is human presence. The other task of the control system is to control the rotation speed of the fan based on the air temperature. The test results of the whole functions show that the fan control using Arduino Mega 2560 circuit can operate well and fulfil the preset objectives.
TABLE OF CONTENT

DECLARATION OF ORIGINALITY.................................................................i
APPROVAL PAGE.........................................................................................ii
INSPIRATIONAL QUOTE .............................................................................3
ACKNOWLEDGMENT ....................................................................................4
ABSTRACT .................................................................................................5
TABLE OF CONTENT.....................................................................................6
LIST OF FIGURES..........................................................................................8
CHAPTER 1 ..................................................................................................10
  1.1 Problem Background ...........................................................................10
  1.2 Final Project Statement ......................................................................10
  1.3 Final Project Objective .......................................................................11
  1.4 Final Project Scope .............................................................................11
  1.5 Final Project Methodology .................................................................12
  1.6 Final Project Outline ..........................................................................12
CHAPTER 2 ..................................................................................................14
  2.1 Constructing the Proposed Solution..................................................14
  2.2 Directory of Components ...................................................................15
  2.3 Main Components ..............................................................................16
    2.3.1 Microcontroller Arduino Mega 2560 ............................................16
    2.3.2 Liquid Crystal Display (LCD)......................................................19
    2.3.3 Infrared Sensor ..........................................................................21
    2.3.4 Temperature Sensor (LM35) .......................................................23
    2.3.5 12V DC Fan ..............................................................................25
    2.3.6 Keypad ......................................................................................26
    2.3.7 Perfboard ..................................................................................27
  2.4 Supporting Tools .................................................................................28
    2.4.1 BD139 NPN Power Transistor ...................................................28
    2.4.2 Resistors .....................................................................................29
    2.4.3 Adapter ......................................................................................30
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.4</td>
<td>Voltage regulator 7805</td>
<td>31</td>
</tr>
<tr>
<td>2.4.5</td>
<td>Capacitor</td>
<td>33</td>
</tr>
<tr>
<td>2.5</td>
<td>Software of the Circuit</td>
<td>34</td>
</tr>
<tr>
<td>CHAPTER 3</td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>3.1</td>
<td>Full Circuit Diagram</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>Component System Testing</td>
<td>36</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Testing Infrared Sensor</td>
<td>37</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Testing LM35</td>
<td>38</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Testing Keypad</td>
<td>39</td>
</tr>
<tr>
<td>3.2.5</td>
<td>Testing Fan DC 12 V</td>
<td>40</td>
</tr>
<tr>
<td>3.2.6</td>
<td>Testing LCD</td>
<td>41</td>
</tr>
<tr>
<td>3.3</td>
<td>Software Analysis</td>
<td>42</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Main System</td>
<td>42</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Starting</td>
<td>43</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Choosing Minimum Temperature by Using Keypad</td>
<td>45</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Sensor Reading</td>
<td>48</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Running the Fan According to the Requirements</td>
<td>49</td>
</tr>
<tr>
<td>3.4</td>
<td>System Testing</td>
<td>50</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Choose the Minimum Temperature</td>
<td>51</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Run the System with Several Conditions</td>
<td>51</td>
</tr>
<tr>
<td>CHAPTER 4</td>
<td></td>
<td>58</td>
</tr>
<tr>
<td>4.1</td>
<td>Analysis and Discussion</td>
<td>58</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Accuracy of the Sensor</td>
<td>58</td>
</tr>
<tr>
<td>4.1.2</td>
<td>The Use of the Adaptor</td>
<td>59</td>
</tr>
<tr>
<td>4.2</td>
<td>Strengths and Weaknesses</td>
<td>61</td>
</tr>
<tr>
<td>CHAPTER 5</td>
<td></td>
<td>62</td>
</tr>
<tr>
<td>5.1</td>
<td>Conclusions</td>
<td>62</td>
</tr>
<tr>
<td>5.2</td>
<td>Recommendations</td>
<td>62</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
<td>64</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 2.1 Block Diagram......................................................... 16
Figure 2.2 Arduino Mega 2560...................................................... 16
Figure 2.3 LCD 16 X 2 ............................................................. 20
Figure 2.4 LCD Circuit ............................................................ 20
Figure 2.5 LCD Diagram .......................................................... 21
Figure 2.6 Infrared Sensor .......................................................... 22
Figure 2.7 Infrared Sensor Circuit ................................................. 22
Figure 2.8 The Way of Working Infrared Sensor ............................. 23
Figure 2.9 LM35 ....................................................................... 24
Figure 2.10 12V DC Fan ............................................................. 26
Figure 2.11 Keypad ................................................................... 26
Figure 2.12 Perfboard ............................................................... 27
Figure 2.13 BD139 NPN Power Transistor ..................................... 28
Figure 2.14 Power Transistor Circuit .............................................. 28
Figure 2.15 Resistor ................................................................... 30
Figure 2.16 Adaptor ................................................................... 31
Figure 2.17 Voltage Regulator ...................................................... 32
Figure 2.18 Capacitor .................................................................. 33
Figure 2.19 Display on Program .................................................. 35
Figure 3.1 Full Circuit Diagram .................................................... 36
Figure 3.2 Blink Program .............................................................. 37
Figure 3.3 Infrared Circuit .......................................................... 38
Figure 3.4 Infrared Program ......................................................... 38
Figure 3.5 LM35 Circuit .............................................................. 39
Figure 3.6 LM35 Program ............................................................ 39
Figure 3.7 Keypad ...................................................................... 41
Figure 3.8 Diagram System ........................................................ 43
Figure 3.9 The Overall System of Fan Control ............................... 50
Figure 3.10 Starting LCD ............................................................ 51
Figure 3.11 Input Voltage for Category One .................................................. 52
Figure 3.12 Fan Voltage for Category One .................................................. 52
Figure 3.13 Input Voltage for Category Two ............................................... 53
Figure 3.14 Fan Voltage for Category Two .................................................. 53
Figure 3.15 Input Voltage for Category Three .............................................. 54
Figure 3.16 Fan Voltage for Category Three ............................................... 54
Figure 3.17 Input Voltage for Category Four ............................................... 55
Figure 3.18 Fan Voltage for Category Four ............................................... 55
Figure 3.19 Sample Graph Temperature vs. Time of the System............... 56
Figure 3.20 Sample Graph Voltage vs. Time of the System ....................... 57
Figure 4.1 Voltage Regulator Circuit ......................................................... 58
Figure 4.2 Hysteresis .................................................................................. 59
Figure 4.3 Multi Voltage Adaptor ............................................................... 60
Figure 4.4 12V DC Fan ........................................................................... 60
CHAPTER 1
INTRODUCTION

1.1 Problem Background

As we know, Indonesia is located in the equatorial line. That condition makes the temperature in Indonesia quite high during the day. The high temperature makes many people try to use tool that can reduce the air temperature or at least make it more comfortable to do work. There are some tools that can be used to do that such as: fan, air conditioner, etc. In the industrial case, they prefer to use fan as the tool to improve working comfort in the worker’s area because the price of a fan is cheaper than the price of an air conditioner. Reducing the temperature with using fan can make working condition more comfortable for the workers and it also can reduce the overheating problem in the machine.

During the internship conducted by the author in an automotive company, many of the fans are broken for so many times. After some observations, the author concludes that this is caused by the extensive use of the fans without rest. The extensive use of the fans can happen because sometimes the operator forgets to turn off the fan and just leave it in on condition. When the fans are broken, the operator cannot work comfortably and some machines become hot and finally get overheated.

From these points of view, the idea to design and to implement a “Fan Control Using Arduino Mega 2560” came up.

1.2 Final Project Statement

This final project emphasizes on designing the hardware and the software that will work as an integrated system to control the fan in the form of a circuit. The controller (Arduino Mega 2560) uses two sensed variables: human presence and air temperature sensor. The value delivered by both sensors can affect the output of the controller.
Through the controller, an auto on-off fan can be realized based on the human presence. The author used an infrared sensor to control the state of the fan. This function can ease the worker in using the fan. When there is a worker in the work place, the fan will be on, and vice versa.

The other task of the controller is to control the rotation speed (commonly referred to as rotations per minute, RPM) of the fan based on the air temperature. When the air temperature is very high in the certain limit, the fan will rotate in the maximum speed and when the temperature is low, the fan will rotate in the low speed. This function can reduce the cost of paying electricity.

1.3 Final Project Objective
The objective of this final project is

- To increase the life time of the fan (the fan should not always be on)
- To reduce the cost of paying electricity (less power consumption, cheaper cost)
- To minimize unworthy activity of the workers

1.4 Final Project Scope
Due to of the limitation of time and resources in the final project preparation, the scope of the final project can be listed as below:

- This final project will discuss on how to realize a fan control using Arduino Mega 2560. Several other electronic components are used, such as LCD, transistor, resistors, keypad, infrared sensor, LM35, adaptor, capacitors, voltage regulators.

- The interface among Arduino Mega 2560, transistor, resistor, and the circuit output components such as LCD and fan, will be developed by using C programming language which is high level language that provides low level access to memory.
1.5 Final Project Methodology

- Studying of working principles.
  This step is initiated by a study about the basic concept of Arduino Mega 2560 and how to use programming C language in the Arduino. Furthermore, the working principle of the infrared sensor and temperature sensor are learned.

- Designing and developing the circuit and the fan control system. In this stage, the author explored and decided the hardware and the software that would be used in the assembly of the fan control system.

- Conducting the testing and the trouble-shooting activity and analysing the circuit system.
  Furthermore, in this step the author checked the integration between the hardware and the software. Finally, the part by part testing was conducted before continuing the circuit packaging process. This activity should be done to make sure that all of the circuits do their functions, including whether the software application runs accordingly at the optimal condition.

1.6 Final Project Outline

Here is the description of the final project outline:

1. CHAPTER 1 INTRODUCTION
This chapter consists of problem background, final project statement, final project objective, final project scope, final project methodology, and final project outline.

2. CHAPTER 2 RESEARCH METHODOLOGY
This chapter lists the hardware (main components and supporting components) and the software used in the project. The descriptions of the functions are also
elaborated. This chapter also discusses how the circuit system works as the interface among the hardware components.

3. **CHAPTER 3 RESULTS**
This chapter reports about how the test of each part of the hardware, software, and also the final simulation test of the system are conducted.

4. **CHAPTER 4 DISCUSSIONS**
This chapter generally cover the analysis of the results reported in the previous chapter. It also explains the strengths and the weaknesses of the fan control system as the end product of this final project.

5. **CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS**
This chapter concludes the whole contents of the final project and gives therecommendationfor the future development of the current result of the final project.
CHAPTER 2
DESIGN DEVELOPMENT

2.1 Constructing the Proposed Solution

Based on the problem background that has already been explained in the Chapter 1, the author came up with an idea to control the fan by integrating the output of an infrared sensor and a temperature sensor (LM 35). Thus, the fan can be controlled based on the human presence and current air temperature. A minimum air temperature is the next input for the system. It should be tipped in through a keypad and will decide the operation range of the proposed system.

In order to construct that system, the author chose some components to fulfil the system needs. There are some components those are used:

- The author used numerical keypad 3x4 to start and to input the minimum temperature in the system. The author chose numerical keypad 3x4 because it is fit with the system’s needs and practice to be used than using other keypad such as keypad 4x4 or alphabetical keypad.

- The author used infrared sensor to sense the human presence, the author used infrared sensor. The author used the infra-red sensor because it can be calibrated easily, it can be plugged directly to Arduino with only simple circuit and also it has cheaper price than using photo transistor sensor.

- The author used LM 35 to sense the air temperature. The author used LM 35 because it has some more benefits than other temperature sensors. This LM 35 can measure from -55 °C – 150 °C range. Beside of that, LM 35 can be calibrated directly to the °C easily. The most important thing, LM 35 is low cost temperature sensor.

- The author used Arduino Mega 2560 as the main brain. The author chose it because i can be programmed in C language without using any additional program, it has many pins for analogue or digital inputs and also it can be supplied directly using computer and adapter.
The author used LCD 16X2 to display the air temperature and the commands. The author used it because it can fulfil the system’s needs in displaying the 16X2 characters. Beside of that this LCD also has the backlight that can show more clearly the commands and the air temperature.

The author used 12V DC fan as the actuator, the author used 12V DC fan. The author used it because the fan is big enough to give effects on the temperature. Prior to this decision, the author already tried to use a 5V DC fan which is considered too small. Beside of that, by using the 12V DC fan, the change of the fan speed can be sensed directly than using 5V DC fan.

The author used multivoltage adapter as the second power supply for the fan and the circuits. The author used it because it can be more convenient for the operator if the operator can select the voltage to determine the highest voltage that will be given to the fan than just using 12V DC adapter.

The author used transistor BD 139 NPN to be a switch and amplifier in controlling the fan. The author chose it because it can amplify the current fit with the system’s needs that can give near 12V DC supply to the fan. Beside of that, this transistor can act as the switch also that is very important is controlling this fan.

Those are the reason why the author chose those components as fan control’s component. The detail and the more explanations about those components can be seen in the next sub chapter.

2.2 Directory of Components

In this final project, the author constructed the fan control system by using the microcontroller Arduino Mega 2560. The block diagram of this project is shown in Figure 2.1.
The block diagram describes the main circuit components which are used. Infrared Sensor, keypad and LM 35 perform as the inputs of the system, whereas the LCD (Liquid Crystal Display) and Fan perform as the outputs of the system. The microcontroller Arduino Mega 2560 performs as the processor of the digital signals sent from the system inputs to produce the digital signals for activating the system outputs. In the next sections, the author will discuss about main components, supporting components, and the software of the circuit. Additionally, the complete circuit diagram can be seen in the Appendix 1.

2.3 Main Components

2.3.1 Microcontroller Arduino Mega 2560
The Arduino Mega 2560 (Figure 2.2) is a microcontroller board based on the ATMEGA2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the ArduinoDuemilanove or Diecimila.[Arduino. December 19 2011. <http://Arduino.cc/en/Main/ArduinoBoardMega2560>.

**Summary**

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>ATMEGA2560</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
<td>5V</td>
</tr>
<tr>
<td>Input Voltage (recommended)</td>
<td>7-12V</td>
</tr>
<tr>
<td>Input Voltage (limits)</td>
<td>6-20V</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
<td>54 (of which 14 provide PWM output)</td>
</tr>
<tr>
<td>Analog Input Pins</td>
<td>16</td>
</tr>
<tr>
<td>DC Current per I/O Pin</td>
<td>40 mA</td>
</tr>
<tr>
<td>DC Current for 3.3V Pin</td>
<td>50 mA</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>256 KB of which 8 KB used by bootloader</td>
</tr>
<tr>
<td>SRAM</td>
<td>8 KB</td>
</tr>
<tr>
<td>EEPROM</td>
<td>4 KB</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>16 MHz</td>
</tr>
</tbody>
</table>
The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). Voltage can be supplied through this pin, or, if supplying voltage via the power jack, the voltage can be accessed through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

**Memory**

The ATMEGA2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

**Input and Output**

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kΩ. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATMEGA8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can
be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

- **PWM: 0 to 13.** Provide 8-bit PWM output with the `analogWrite()` function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **TWI: 20 (SDA) and 21 (SCL).** Support TWI communication using the wire library. Note that these pins are not in the same location as the TWI pins on the Duemilanove or Diecimila.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

### 2.3.2 Liquid Crystal Display (LCD)

Liquid Crystal Display (LCD) is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly.

LCDs therefore need a light source and are categorized as "passive" displays. Some types can use ambient light such as sunlight or room lighting. There are many types of LCDs that are designed for both special and general uses. They can be optimized for static text, detailed still images, or dynamic, fast-changing, video content. At this project, the author used the LCD 16x2 (sample picture: Figure 2.3). It means the LCD has 16 characters per line by 2 lines with
the total of 32 characters. This LCD is supported with LED backlight, thus the user is able to turn on/off the light of LCD display. The appearance of LCD 16x2 is shown in the figure below.[Wikipedia. 17 December 2011. <http://en.wikipedia.org/wiki/Liquid_crystal_display>.

![Figure 2.3 LCD 16 X 2](image)

To wire your LED screen to your Arduino, the following connection must be done:

- LCD RS pin to digital pin 12
- LCD Enable pin to digital pin 11
- LCD D4 pin to digital pin 5
- LCD D5 pin to digital pin 4
- LCD D6 pin to digital pin 3
- LCD D7 pin to digital pin 2

Additionally, wire a 10K pot to +5V and GND, with it's wiper (output) to LCD screen’s VO pin (pin3).

![Figure 2.4 LCD Circuit](image)
In this project, the author used a LCD to display the minimum air temperature that worker chooses for the fan. Besides, LCD also shows the change of air temperature in a certain period of time. The LCD connection to the Arduino and the LCD diagram can be seen in Figure 2.4 and Figure 2.5. By this way, the worker can see the change of air temperature according the real value measured by the sensor.

2.3.3 Infrared Sensor

IR emitter and IR phototransistor

An infrared emitter is an LED (see Figure 2.6) made from gallium arsenide, which emits near-infrared energy at about 880nm. The infrared phototransistor acts as a transistor with the base voltage determined by the amount of light hitting the transistor. Hence it acts as a variable current source. Greater amount of IR light cause greater currents to flow through the collector-emitter leads. As shown in the diagram below, the phototransistor is wired in a similar configuration to the voltage divider. The variable current traveling through the resistor causes a voltage drop in the pull-up resistor. This voltage is measured as the output of the device. [Irbasic.17 December 2011.<http://irbasic.blogspot.com/>.]
Figure 2.6 Infrared Sensor

![Infrared Sensor](image)

Figure 2.7 Infrared Sensor Circuit

**Photo**

IR reflectance sensors contain a matched infrared transmitter and infrared receiver pair. The infrared circuit can be seen in Figure 2.7. These devices work by measuring the amount of light that is reflected into the receiver. Because the receiver also responds to ambient light, the device works best when well shielded from ambient light, and when the distance between the sensor and the reflective surface is small (less than 5mm). [Irbasic.17 December 2011. <http://irbasic.blogspot.com/>.]

An IR emitter  An IR phototransistor
In this project, the author used an infrared sensor to determine whether there is a person or not in some range around the sensor. This infrared sensor will give some value when there is a person to the Arduino and it will give order to Arduino to on or off the fan.

2.3.4 Temperature Sensor (LM35)

The LM35 (Figure 2.9) series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1/4 °C at room temperature and ±3/4°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35’s low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The

![Figure 2.9 LM35](image)

**Features**

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guarantee able (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1/4°C typical
- Low impedance output, 0.1 W for 1 mA load
In this project, the author used the LM35 to work together with infrared sensor in determining when the fan should be turned on or off. The LM35 will deliver the value just like the explanation previously given and the value will be converted to the degree Celsius by the software. The degree Celsius from LM35 will be used in the decision to turn on or off the according to the minimum air temperature given by the worker and human presence as given by the infrared sensor.

2.3.5 12V DC Fan

A mechanical fan is a machine used to create flow within a fluid, typically a gas such as air. A fan consists of a rotating arrangement of vanes or blades which act on the air. Usually, it is contained within some form of housing or case. This may direct the airflow or increase safety by preventing objects from contacting the fan blades. Most fans are powered by electric motors, but other sources of power may be used, including hydraulic motors.

A standalone fan is typically powered with an electric motor. Fans are often attached directly to the motor's output, with no need for gears or belts. The electric motor is either hidden in the fan's centre hub or extends behind it. For big industrial fan, three-phase asynchronous motors are commonly used, placed near the fan and driving it through a belt and pulleys. Smaller fan are often powered by shaded pole AC motor or brushed or brushless DC motors. AC-powered fan usually use mains voltage, while DC-powered fan use low voltage, typically 24 V, 12 V or 5 V. Cooling fan for computer equipment exclusively use brushless DC motors, which produce much less electromagnetic interference. Brushless DC motors (BLDC motors, BL motors) also known as electronically commutated motors (ECMs, EC motors) are electric motors powered by direct-current (DC) electricity and having electronic commutation systems, rather than mechanical commutators and brushes. The current-to-torque and frequency-to-speed relationships of BLDC motors are linear. [Wikipedia. 15December 2011. <http://en.wikipedia.org/wiki/Mechanical_fan>.]
In this project, the author used the 12V DC fan to become the main actuator of this project. The fan will help to reduce the temperature by varying the rotation speed through the Arduino. The author used the 12 VDC fan because the size is enough to be able to reduce the air temperature.

### 2.3.6 Keypad

A keypad (Figure 2.11) is a set of buttons arranged in a block or "pad" which usually bear digits, symbols and usually a complete set of alphabetical letters. If it mostly contains numbers then it can also be called a numeric keypad. [Wikipedia. 15 December 2011. <http://en.wikipedia.org/wiki/Keypad>.]
In this project, the author used keypad as a medium to input the minimum temperature. This keypad also has the function to start the mechanism of the fan control in this project.

2.3.7 Perfboard

Perfboard (Figure 2.12) is a material for prototyping electronic circuits. It is a thin, rigid sheet with holes pre-drilled at standard intervals across a grid, usually a square grid of 2.54 mm (0.1") spacing. These holes are ringed by round or square copper pads. Inexpensive perfboard may have pads on only one side of the board, while better quality perfboard can have pads on both sides (plate-through holes). Since each pad is electrically isolated, the builder makes all connections with either wire wrap or miniature point to point wiring techniques. Discrete components are soldered to the prototype board such as resistors, capacitors, and integrated circuits. The substrate is typically made of paper laminated with phenolic resin (such as FR-2) or a fiberglass-reinforced epoxy laminate (FR-4). The 0.1" grid system accommodates integrated circuits in DIP packages and many other types of through-hole components. Perfboard is not designed for prototyping surface mount devices. [Wikipedia. 3 January 2012. <http://en.wikipedia.org/wiki/Perfboard>.]
In this project, the author used perfboard as the media to connect all of the tools that are used for the fan control system. This board only needs to be soldered to connect each component to other. It can be used for many projects that need to accommodate many little parts such as capacitors and resistors.

2.4 Supporting Tools

2.4.1 BD139 NPN Power Transistor

BD139 NPN Power Transistor (Figure 2.13) features collector dissipation (Pc(max): 12.5 W), collector current (Ic(max): 1.5 A), and collector-emitter voltage (Vceo(max): 80V). This transistor is used to be implemented in the power transistor circuit. The design of this circuit is described in the figure 2.13. [100y.15 December 2011.<http://us.100y.com.tw/PNoInfo/27171.htm>.

\[ I_E = I_B + I_C \] (2.1)
For this project, the author only needed one power transistor circuit to amplify the current supplied by the microcontroller (Arduino Mega 2560 – as the input signal (voltage)) to the outputs of the circuit system which is the fan. By using resistors $R_1 = 3 \, k\Omega$ and $R_2 = 114 \, k\Omega$ in the circuit as can be seen on Figure 2.13, the circuit receive input voltage between 1 – 5 V from the Arduino to be amplified in to certain voltage. Equation 2.1 until 2.5 are equations to calculate the current in the base and also the gain $\beta$ based on the input voltage $V_{CC}$.

\[ V_{CE} = -V_{BC} + V_{BE} \quad (2.2) \]

\[ I_B = (V_{CC} - V_{BE})/R_1 \quad (2.3) \]

\[ \alpha = \frac{I_C}{I_E} \quad (2.4) \]

\[ \beta = \frac{I_C}{I_B} \quad (2.5) \]

For this project, the author only needed one power transistor circuit to amplify the current supplied by the microcontroller (Arduino Mega 2560 – as the input signal (voltage)) to the outputs of the circuit system which is the fan. By using resistors $R_1 = 3 \, k\Omega$ and $R_2 = 114 \, k\Omega$ in the circuit as can be seen on Figure 2.13, the circuit receive input voltage between 1 – 5 V from the Arduino to be amplified in to certain voltage. Equation 2.1 until 2.5 are equations to calculate the current in the base and also the gain $\beta$ based on the input voltage $V_{CC}$.

\[ \alpha = \frac{I_C}{I_E} \quad (2.4) \]

\[ \beta = \frac{I_C}{I_B} \quad (2.5) \]

2.4.2 Resistors

Resistors (Figure 2.15) are common elements of electrical networks and electronic circuits and are ubiquitous in most electronic equipment. Practical resistors can be made of various compounds and films, as well as resistance wire (wire made of a high-resistivity alloy, such as nickel-chrome). Resistors are also implemented within integrated circuits, particularly analog devices, and can also be integrated into hybrid and printed circuits. [Wikipedia. 16 December 2011. <http://en.wikipedia.org/wiki/Resistor>.

The current through a resistor is in direct proportion to the voltage across the resistor's terminals. Thus, the ratio of the voltage applied across a resistor's
terminals to the intensity of current through the circuit is called resistance. This relation is represented by Ohm's law:

\[ I = \frac{V}{R} \]  

(2.6)

2.4.3 Adapter

An adapter or adaptor (Figure 2.16) is a device that converts attributes of one device or system to those of an otherwise incompatible device or system.

The AC adapter, AC/DC adapter or AC/DC converter is a type of external power supply, often enclosed in what looks like an over-sized AC plug. AC adapters are used with electrical devices that require power but do not contain internal components to derive the required voltage and power from mains power. The internal circuitry of an external power supply is very similar to the design that would be used for a built-in or internal supply.
External power supplies are used both with equipment with no other source of power, and with battery-powered equipment, where the supply both charges the battery and powers the equipment when plugged in.[Wikipedia. 16 December 2011. <http://en.wikipedia.org/wiki/Adapter>.]

Figure 2.16 Adaptor

In this project, the author used a multivoltage adapter. This adapter can ease the change of voltage as desired by the user. The use of multivoltage adapter in this project is to supply the voltage to the fan according to the order from the Arduino through current amplifier.

2.4.4 Voltage regulator 7805

A voltage regulator(Figure 2.17) is an electrical regulator designed to automatically maintain a constant voltage level. A voltage regulator may be a simple "feed-forward" design or may include negative feedback control loops. It may use an electromechanical mechanism, or electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages.
Voltage Regulator (regulator), usually having three legs, converts varying input voltage and produces a constant regulated output voltage. They are available in a variety of outputs.

The most common part numbers start with the numbers 78 or 79 and finish with two digits indicating the output voltage. The number 78 represents positive voltage and 79 negative one. The 78XX series of voltage regulators are designed for positive input. And the 79XX series is designed for negative input. [Wikipedia.15 December 2011.<http://en.wikipedia.org/wiki/Voltage_regulator>.]

Examples:
- 5V DC Regulator Name: LM7805 or MC7805
- -5V DC Regulator Name: LM7905 or MC7905
- 6V DC Regulator Name: LM7806 or MC7806
- -9V DC Regulator Name: LM7909 or MC7909

![Figure 2.17 Voltage Regulator](image)

In this project, the author used voltage regulator 7805 to regulate the 12 volt from the adapter to 5 volt. It can guarantee the constant 5 volt for the system; in this case will be used by the infrared sensor and the LM35 temperature sensor. The adapter is needed because the voltage from Arduino sometimes is no stable.
2.4.5 Capacitor

A capacitor (formerly known as condenser) is a passive two-terminal electrical component used to store energy in an electric field. The forms of practical capacitors vary widely, but all contain at least two electrical conductors separated by a dielectric (insulator); for example, one common construction consists of metal foils separated by a thin layer of insulating film. Capacitors are widely used as parts of electrical circuits in many common electrical devices.[Wikipedia. 16 December 2011. <http://en.wikipedia.org/wiki/Capacitor>.]

Figure 2.18 Capacitor

\[ C = \frac{Q}{V} \quad (2.7) \]

\[ C = \frac{\varepsilon A}{d} \quad (2.8) \]

In this project, author used capacitor to help the voltage regulator in order to regulating the 12 volt voltage from adapter to 5 volt voltage that is used to supply LM35 and infrared sensor. Equation 2.7 and 2.8 are the general equations.
to calculate the capacitance based on \( d \) (length), \( A \) (area), \( \varepsilon \) (permittivity). The capacitor used in this project is 0.1 \( \mu \)F and 0.33 \( \mu \)F.

2.5 Software of the Circuit

Software in this project plays an important role as the hardware. The author used C language as the language to program the microcontroller Arduino Mega 2560. The display of the program can be seen in Figure 2.19. This language is high level language that can be easily understood by people in making that program. Arduino uses the ATMEGA 2560 as the brain in controlling the system. It means that the Arduino itself has the compiler to compile the C language to the AVR language. Software to program the Arduino is already in one package with the compiler without any requirement for additional program. The actual program will be explained in the Chapter 3.
```cpp
#define pinDataLM35 15

#include <Keypad.h>
String key1 ;
const byte ROWS = 4; // four rows
const byte COLS = 3; // three columns
char keys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}
};
byte rowPins[ROWS] = {14, 15, 16, 17}; // connect to the row pinout
byte colPins[COLS] = {18,19,20}; // connect to the column pinouts of

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS,
int countone = -1;
int counttwo= 0;
String stringone = String(""');
int z=1;

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

Figure 2.19 Display on Program
CHAPTER 3

RESULTS

3.1 Full Circuit Diagram

According to the components those have been chosen by the author, the full diagram circuit will be in Figure 3.1 below.

![Figure 3.1 Full Circuit Diagram](image-url)
3.2 Component System Testing

3.2.1 Testing the Microcontroller’s Software

Firstly, the compiler program should be installed in a computer and set by using the provided USB output. In order to check the microcontroller software we can use the example program that is provided by the compiler. Before conducting the test, the user should prepare the LED as the test tool. Blink LED program (Figure 3.1) is one of the programs that can be used to check the microcontroller.

```cpp
/* Blink

Turns on an LED on for one second, then off for one second, loops.

This example code is in the public domain.
*/

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000); // wait for a second
}
```

**Figure 3.2 Blink Program**

After setting the simple circuit and uploading the blink program to the Arduino, LED will blink every one second if the microcontroller software is correctly working. There are a lot of simple examples program that can be used to check the microcontroller, provided in Arduino website (www.Arduino.cc).
3.2.2 Testing Infrared Sensor

In order to test the infrared sensor, Arduino can also be used. Firstly, the circuit as seen on Figure 3.2 must be set up.

![Figure 3.3 Infrared Circuit](image)

![Figure 3.4 Infrared Program](image)

After the connection is completed and the program is written (see Figure 3.3) and uploaded into the compiler, the value of the sensor can be seen. If the
infrared sensor is working, the serial monitor will show a value that will change when an object is moved near the infrared sensor.

3.2.3 Testing LM35

In checking LM35 as the temperature sensor, we can use Arduino can be utilized to show the measured temperature in the serial monitor of the Arduino compiler. The circuit is shown in the Figure 3.4.

![Figure 3.5 LM35 Circuit](image)

After the connection as in Figure 3.5 is done, the program can be uploaded into the Arduino. The value of the sensor can be seen on the serial monitor. If the
infrared sensor is working well, the serial monitor will show the current temperature value.

### 3.2.4 Testing Keypad

In order to check keypad whether it is working or not, the keypad should be connected to the Arduino in the way shown in Figure 3.6. After that, a simple programming can be written to define each button to be like in the keypad and the program.

Program:

```c
#include <Keypad.h>

const byte ROWS = 4; // four rows
const byte COLS = 3; // three columns

char keys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'#','0','*'}
};

byterowPins[ROWS] = {5, 4, 3, 2}; // connect to the row pinouts of the keypad
bytecolPins[COLS] = {8, 7, 6}; // connect to the column pinouts of the keypad

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    char key = keypad.getKey();
    if (key != NO_KEY)
    {
        Serial.println(key);
    }
}
```
After all are already prepared, the program can be uploaded. By observation on the serial monitor, it can be seen whether the keypad is working or not, just by pushing every button and seeing the result whether the result is correct or not.

3.2.5 Testing Fan DC 12 V

In order to check the fan, it can be connected to the 12 V output of the AC/DC adapter. Thing that should be remembered is the polarity of the adapter. The polarity of the connecting cable must be taken care. If the fan starts rotating, then it is working well.

3.2.6 Testing LCD

In order to test LCD, it can be directly connected to the voltage source (5 volt). The LCD is working correctly if the LCD is turned on in every part of its surface. The other way to check whether LCD works or not is using Arduino. A
connection as shown in the Figure 2.4 should be constructed, along with the following program uploaded into the Arduino.

**Program:**

```c
#include <LiquidCrystal.h>
LiquidCrystalLCD(12,11,5,4,3,2);
void setup()
{
  LCD.begin(16,2);
  LCD.print("hello, world!");
}
void loop()
{
  LCD.setCursor(0,1);
  LCD.print(millis()/1000);
}
```

If there is a “hello world” word written on the LCD, it means that the LCD is working.

### 3.3 Software Analysis

In this project, the author wrote the software in C language. This C language is supported by Arduino to control all the process in running the control system in this project. This software is uploaded to Arduino by using arduino -0022.

#### 3.3.1 Main System

In this system, the author wrote the primary program to control the rotation speed of the fan, based on detection of the infrared sensor and the temperature sensor. The way how the program control the fan is explained in the flow chart below.
As can be seen on the flowchart, the system consists of three subsystems: the keypad, the sensor reading, and the fan. There is also fan speed control that needs to be defined and decided by the author. Beside three primary subsystems, the author also used LCD to show the air temperature that the worker wants.

3.3.2 Starting

In the starting of the program, the program content of the initialization of pin LM35, keypad, LCD, infrared sensor, and fanspeed which is controlled by the pulsewidth of the voltage. Those initiations are functioned to introduce the tools to the circuit so that the program will know about that circuit and what the starting point of that electrical components. The other thing included in the starting section is, that there is void set up to set the output and LCD.
Program:

#define pinDataLM35 15

//initialize keypad
#include <Keypad.h>
String key1 ;
const byte ROWS = 4;  //four rows
const byte COLS = 3;  //three columns
char keys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}
};
byterowPins[ROWS] = {14, 15, 16, 17}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {18,19,20}; //connect to the column pinouts of the keypad
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS );

String stringone = String("" );
int z=1;

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystalLCD(12, 11, 5, 4, 3, 2);

//initialize temperature reading
float temperature;
int samples[5];
float tempa;
float tempc = 0;
int i;

//fanpeed type
int fanpeed;
//variable type
int sensor1;
int a;
int b;
int c;
int x1;
int x2;
int x3;
int x4;
int sensor;

void setup() {
  pinMode(13, OUTPUT);
  Serial.begin(9600); // buka serial port, set baud rate 9600 bps
  LCD.begin(16, 2);
  Serial.println("Input the minimum temp");
  LCD.print("FAN CONTROLLER");
  delay(2000);
  LCD.clear();
  LCD.print("made by");
  delay(2000);
  LCD.clear();
  LCD.print("Edwin/EE");
  LCD.setCursor(0,1);
  LCD.print("002200800002");
  delay(3000);
  LCD.clear();
  LCD.setCursor(0,0);
  LCD.print("1. Input min temp");
  LCD.setCursor(0,1);
  LCD.print("2. Press *");
  delay(3000);
  LCD.clear();
  LCD.print("temp");
}

3.3.3 Choosing Minimum Temperature by Using Keypad

After the initialization of the keypad by using the starting program, now each button of the keypad should be assigned. The number 1 until 9 only function as the number. But, it can also be used to determine the minimum air temperature limit to the output. For *, the author set it to display what the worker already pushed in the keypad, display it on the LCD, and move to the sensor reading.
Program:

void loop() {
  char key = keypad.getKey();

  if (z==1) {
    if(key != NO_KEY) {
      switch (key) {
        case '#':
          Serial.println("reset");
          LCD.print("reset");
          delay(500);
          z=1;
          break;
        case '0':
          Serial.print(key);
          LCD.print(key);
          stringone = stringone + key;
          break;
        case '1':
          Serial.print(key);
          LCD.print(key);
          stringone = stringone + key;
          break;
        case '2':
          Serial.print(key);
          LCD.print(key);
          stringone = stringone + key;
          break;
        case '3':
          Serial.print(key);
          LCD.print(key);
          stringone = stringone + key;
          break;
        case '4':
          Serial.print

Serial.print(key);
LCD.print(key);
stringone = stringone + key;
break;
case '5':
Serial.print(key);
LCD.print(key);
stringone = stringone + key;
break;
case '6':
Serial.print(key);
LCD.print(key);
stringone = stringone + key;
break;
case '7':
Serial.print(key);
LCD.print(key);
stringone = stringone + key;
break;
case '8':
Serial.print(key);
LCD.print(key);
stringone = stringone + key;
break;
case '9':
Serial.print(key);
LCD.print(key);
stringone = stringone + key;
break;
Serial.print(key);
case '*':
Serial.println('');
Serial.println("Your choice");
Serial.println(stringone);
LCD.clear();
LCD.print("Your choice: ");
LCD.print(stringone);
z = 2;
3.3.4 Sensor Reading

In this part, the program orders the Arduino to read the output of the sensor. For infrared sensor, the Arduino reads the values when there is an obstacle and when there is no obstacle. For LM35, Arduino reads the output value and converts it into degree Celsius so that the user can know it easily.

**Program:**

```c
if (z==2)
{
    for(i = 0;i<=4;i++)
    {
        temperature = analogRead(pinDataLM35);
        temperature = analogRead(pinDataLM35);
        temperature = analogRead(pinDataLM35);
        temperature = analogRead(pinDataLM35);
        temperature = analogRead(pinDataLM35);
        samples[i] = temperature;
        tempc = tempc + samples[i];
        delay(800);
    }
    tempc = tempc/5;
    tempc = tempc * (5.0 * 100.0/1024.0);
    tempa = (tempc*1.8)+32 ;
    Serial.print("Suhusatini (Celcius) : ");
    Serial.println(tempc);
    LCD.setCursor(0,0);
    LCD.clear();
    LCD.print("TempC: ");
    LCD.print(tempc);
    lcd.setCursor(0,1);
    lcd.print("TempF:");
    lcd.print(tempa);
    sensor1=analogRead(A3);
    sensor1=analogRead(A3);
    Serial.println(sensor1);
}
```
3.3.5 Running the Fan According to the Requirements

After the output values of sensors have been out, those outputs will be used for the limit when the fan will be off or will be on and how fast the fan will rotate. Those requirements will determine the final output in the fan. The fan can automatically work based on human presence and temperature. There are four categories for the fan speed. The category one is for the slowest speed and the category four is for the fastest speed. It means that the input voltage is also the lowest for category one and the highest for category four. Those four categories are made based on the temperature different. The minimum temperature limit will be chosen by the user. For the category one, the range temperature will be between the minimum temperature and minimum temperature + 4 °C. The range between each category is 4 °C. The example can be seen in the Figure 3.18 and Figure 3.19.

Program:

```cpp
long int one = stringToLong (stringone);
a = intone+4;
b = intone+8;
c = intone+12;

if (sensor1<1020){
    if (tempc<(intone-0.4))
        {x1=0;}
    if (tempc>(intone+0.4))
        {x1=180;}
    if (tempc<(a-0.4))
        {x2=0;}
    if (tempc>(a+0.4))
        {x2=25;}
    if (tempc<(b-0.4))
        {x3=0;}
    if (tempc>(b+0.4))
        {x3=25;}
    if (tempc<(c-0.4))
        {x4=0;}
    if (tempc>(c+0.4))
        {x4=25;}

    fanpeed = x1 + x2 + x3 + x4;
    Serial.print(fanpeed);
    analogWrite(13,fanpeed);
}
if(sensor1>1020){analogWrite(13,0);}
tempc=0;
```
3.4 System Testing

After checking all the hardware and the software, the author assembled all components together and tested them in a series of simulation (see Figure 3.8 and Figure 3.9). The procedure of the testing is listed as follows:

1. Choose the minimum temperature by using the keypad.
2. Run the system with several conditions.

Figure 3.9 The Overall System of Fan Control
The author set the simulation to be similar with the condition in the factory where the locations of the sensors and the fans are adapted from the room shape in the manual welding sector.

### 3.4.1 Choose the Minimum Temperature

In order to set the minimum temperature, the power supply should be plugged in and the button “*” should be pressed. The system is now starting.

![Starting LCD](image)

**Figure 3.10 Starting LCD**

After the system is initiated, the air temperature will be shown in the LCD. There are two values displayed while the system is running, which are the temperature in degree Celsius and degree Fahrenheit.

### 3.4.2 Run the System with Several Conditions

The system will keep monitoring the presence of human and also the of the air temperature. The main point of the developed system is the human presence. When there is no person in front of the machine, the infrared sensor will send the command that the fan should always be off. When there is a person standing in front of the machine, the fan will be on and the temperature sensor will define how fast the speed of the fan should be, according to the measured temperature.

In case there is human presence in front of the machine, the rotation speed of the fan will be categorized in to four different categories.
For category one, the input voltage to the transistor is 3.42 volt, see Figure 3.42.

The fan voltage for category one from transistor is 8.3 volt, see Figure 3.12.
For category two, the input voltage to the transistor is 3.9 volt, see Figure 3.13.

![Figure 3.13 Input Voltage for Category Two](image1)

The fan voltage for category two from transistor is 9.43 volt, see Figure 3.14.

![Figure 3.14 Fan Voltage for Category Two](image2)
For category three the input voltage to the transistor is 4.38 volt, see Figure 3.15.

![Figure 3.15 Input Voltage for Category Three](image1)

The fan voltage for category three from transistor is 10.43 volt, see Figure 3.16.

![Figure 3.16 Fan Voltage for Category Three](image2)
For category four, the input voltage to the transistor is 4.85 volt, see Figure 3.15.

Figure 3.17 Input Voltage for Category Four

The fan voltage for category four from transistor is 11.6 volt, see Figure 3.16.

Figure 3.18 Fan Voltage for Category Four
Figure 3.18 shows the relation between the temperature and the time. In the range between \( t = 4 \text{s} \) until \( t = 40 \text{s} \), the graph shows the normal air temperature (there are some fluctuations of 0.2 – 0.3 °C). In the time between \( t = 44 \text{s} \) and \( t = 80 \text{s} \), the author used a solder to increase the air temperature. The author moved the solder close enough to the temperature sensor until the measured air temperature reached a certain desired high temperature.

Figure 3.19 shows the relation between the output voltages to the fan as a function of the time. This figure is directly related with Figure 3.18. As the temperature change in Figure 3.18, the output voltage also will change. For example, the author chose 25 °C as the minimum temperature. In the range between 25 °C - 29 °C, the output voltage will be (between \( t = 4 \text{s} \) and \( t = 32 \text{s} \)) 8.3 volt. In the range between 29 °C - 33 °C, the output voltage will be (between \( t = 36 \text{s} \) and \( t = 44 \text{s} \)) 9.43 volt. In the range between 33 °C - 37 °C, the output voltage will be (between \( t = 48 \text{s} \) and \( t = 60 \text{s} \)) 10.43 volt. In the range higher than 37 °C, the output voltage will be (between \( t = 60 \text{s} \) and \( t = 80 \text{s} \)) 11.60 volt. The range that has been made by the author has a hysteresis to reduce the alternating condition of output voltage. The hysteresis will be explained in the Chapter 4.
Figure 3.20 Sample Graph Voltage vs. Time of the System
CHAPTER 4
DISCUSSIONS

4.1 Analysis and Discussion

4.1.1 Accuracy of the Sensor

The author found that each part of the fan control system works normally, but there is a problem when all parts are set to work together, especially when the infrared sensor and the temperature sensor (LM35) turn on the fan. The accuracy of temperature sensor will deteriorate because the voltage supply for two sensors will be not stable when the fan is on. That condition is caused by the sensitivity of the LM35 sensor. For every deviation of 10 millivolt, the temperature will deviate for 1 °Celsius. This value is very significant compared to the exchange of the voltage and the current in the system. In order to handle this, the author used a voltage regulator and two capacitors to stabilize the power supply to the temperature sensor.

Voltage regulator circuit:

![Voltage Regulator Circuit](image)

Figure 4.1 Voltage Regulator Circuit

The circuit that has been arranged can reduce the error of the LM35 measurement but there is still a remaining error that should be handled. In order to further reduce the effect of unstable voltage, because of unstable sensor
measurement, the author used hysteresis as the program guideline. Hysteresis itself is the system that depends on not only current environment but also on its past environment. This dependence arises because the system can be in more than one condition. To predict its future evolution, either its internal state or its history must be known. If a given input alternately increases and decreases, the output tends to form a loop as in Figure 4.2. However, loops may also occur because of a dynamic lag between input and output. Often, this effect is also referred to as hysteresis, or rate-dependent hysteresis. This effect disappears as the input changes more slowly; so many experts do not regard it as true hysteresis. The program that author used can be seen in the sensor reading in Chapter 3.[Wikipedia. 20 December 2011. <http://en.wikipedia.org/wiki/Histeresis>.]

Hysteresis in this fan control system is to avoid unstable or alternating on off to the fan on the border between the categories, than the hysteresis issued.

![Figure 4.2 Hysteresis](image)

**Figure 4.2 Hysteresis**

### 4.1.2 The Use of the Adaptor

Firstly, the author used a 5 volt adapter and 5 volt DC fan to run the system. After a couples times in the improvement cycle of the system, the 5 volt DC is too small to run the system. It cannot give effective effect to change the temperature. It needs a bigger fan size. So, the author chose to use a 12V DC fan. It means that
author also used the 12 V adapter to support the fan. Finally, the author used the multivoltage adapter that can be used for several values of voltage. It eases the user when they want to use the other DC fan with different voltage based on their needs.

![Multi Voltage Adaptor](image)

**Figure 4.3 Multi Voltage Adaptor**

![12V DC Fan](image)

**Figure 4.4 12V DC Fan**

The author has set the transistor with followings specifications:

BD139 NPN Power Transistor features, that collector dissipation (Pc(max): 12.5 W), collector current (Ic(max): 1.5 A), and collector-emitter voltage (Vceo(max): 80V). It means that the user only can implement the circuit
with maximum 80 V voltages. The circuit will be burned, especially the transistor. So the author warns the use of the fan to be within 5 – 24 V range, so that the use of the system is save for the user.

4.2 Strengths and Weaknesses

The strengths of the system:

- The system can be implemented for real room or the real factory with the development design
- It can increase the effectiveness of using the fan (reducing cost, reducing not valuable work).
- It is more compact than the use analogue design in controlling the fan.

The weaknesses of the system:

- It only can use DC fan (in order to use an AC fan, additional circuit must be added)
- It has some noises that can affect the measurement. (Especially the noise that affects the LM35. This noise can influence the reading of the sensor and will give the unstable result)
CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions
Based on the testing results of the whole functions of fan controller with using Arduino Mega 2560, there are several points that can be concluded, which are:

1. Reducing fan speed also decreases the wear on the fan. Fan wear is roughly a function of the absolute number of revolutions of the fan. Reduced wear translates into increased lifetime. So, it is clear the system will increase the life time of the fan because it will be on; only when it is needed (there is a person). It is better than the old condition that sometimes the fan is always turned on.

2. The fan will consume power based on air the temperature and human presence. Thus, the electricity will be not used as much as the old condition, where the fan is always operated in the maximum speed. So, this system reduces the use of electricity and reduces the cost of electricity.

3. This resulting system utilizes the presence of the worker as the parameter whether the fan should be turned on or off. So, the worker does not need to push the button in order to activate the fan. By this, it can be expected that the effectiveness of the worker will be increased. (The worker only needs to set the fan control system once by the first time it is initialized.)

5.2 Recommendations
In order to develop the existing final project of fan control using arduino Mega 2560 circuit, there are two recommendations future improvements which are:

a. Applying more functions for keypad.
   In the existing keypad function, it is only used to set minimum temperature. For the proposed improvement, the keypad should be used
for some functions that can be run in the fan control system. It can be used to determine the range of the category or the maximum temperature. Another possibility is that the keypad also can be used to delete the input when the user types the input wrongly without restarting the system.

b. Applying circuit for control the AC Fan.
In the existing system, the microcontroller only controls the DC fan. It can be utilized more to control the AC fan that is commonly used. It will need different circuits and also different power supply. It needs the transistor and the power supply DC to AC. This improvement can be very useful so that the commonly used fan in houses or factory can be integrated directly.

c. Applying buzzer as the alarm.
When the air temperature is high and reaches the dangerous temperature to the machine and human, it can be designed to give warning signal to the people to check the machine if there is overheating situation or any other circumstances that influence the air temperature. It also can takes role as the warning system if there is fire or any other malfunction in the environment that causes the extreme temperature raise.
REFERENCES


