BLIND AND LIGHTING CONTROL TO MAINTAIN COMFORT LIGHT INTENSITY OF THE CLASSROOM UTILIZING MICROCONTROLLER ATMega8535

A final project report presented to the Faculty of Engineering

By

I Gusti L. Wahyudi Indrawan

002200900004

in partial fulfillment of the requirements of the degree Bachelor of Science in Electrical Engineering

President University April 2013
DECLARATION OF ORIGINALITY

I declare that this final project report, entitled “Blind and Lighting Control to Maintain Comfort Light Intensity of the Classroom” is my own original piece of work and, to the best of my knowledge and belief, has not been submitted, either in whole or in part, to another university to obtain a degree. All sources that are quoted or referred to are truly declared.

Cikarang, April 2013

I Gusti L. Wahyudi Indrawan
BLIND AND LIGHTING CONTROL TO MAINTAIN COMFORT LIGHT INTENSITY OF THE CLASSROOM UTILIZING MICROCONTROLLER ATMega8535

By

I Gusti L. Wahyudi Indrawan
002200900004

Approved by

Arthur Silitonga, M.Sc
Final Project Supervisor

Dr.-Ing. Erwin Sitompul
Head of Study Program
Electrical Engineering

Dr.-Ing. Erwin Sitompul
Dean of Faculty of Engineering
ACKNOWLEDGMENT

Foremost, I would like to praise God for His overflow of mercy and guidance, so that the compilation of this final project report could be finished. Thank you for the infinite blessing and faithfulness to support me in all the time of research and writing of this final project report.

This final project is very hard to be finished without any supports. Therefore I would like to express my special gratitude to my family. Thank you for believing and funding me. Through your encouragement for me, I put my best effort in working on this project. Without you, it is impossible for me to pass all of these challenges in my life.

My deep gratefulness I delivered to my final project advisor, Arthur Silitonga, M. Sc. Thank you for your precious supervision, time, and guidance. By your support, I am able to obtain plentiful amount of knowledge through the process of developing my final project.

I would also express my gratitude to the Head of Electrical Engineering Study Program, Dr.-Ing. Erwin Sitompul. Thank you for your all of your spirit and wise advice for me during my final project development.

Last but not least, I would like to thank all of my Electrical Engineering classmates in President University for the help, support, and solidarity. It is just great when we are together.

Cikarang, April 2013,

I Gusti L. Wahyudi Indrawan
ABSTRACT

Most of the artificial lighting systems are presented and operated based on the day and night only without taking any consideration of light contribution from the outside of the room, especially in the day. This situation can be inconvenient for the student inside a classroom to manually switch on or switch off the lamps several times according to the ambient light intensity level that may increase or drop significantly. Besides that, studying in a low light intensity will affect the eyes become tired faster than studying under a sufficient light intensity. On the other hand, doing presentation in a very bright classroom will not give a good display. From this point of view, the idea to develop and to implement this final project comes up. The main objective is to design and to model an automatic light intensity controller which will be used to maintain a proper lighting system for the classroom to provide a comfortable learning atmosphere for the student. A comfortable light intensity can be maintained by integrating automatic brightness controller with automatic blind system. Automatic brightness controller is built by adopting the principle of phase firing angle while blind will be driven by stepper motor. A well planned blind and lighting system in a classroom will give significant effect to the learning atmosphere and concentration of the students.

Keywords: automatic brightness controller, automatic blind controller, phase firing angle, stepper motor.
# TABLE OF CONTENT

TABLE OF CONTENT ........................................................................................................... i
LIST OF FIGURES ............................................................................................................. iv
LIST OF TABLES ................................................................................................................ vi
EQUATIONS ....................................................................................................................... vii

CHAPTER 1 INTRODUCTION .............................................................................................. 1
  1.1 Final Project Background ....................................................................................... 1
  1.2 Problem Statement ............................................................................................... 4
  1.3 Final Project Objectives ....................................................................................... 4
  1.4 Final Project Scopes and Limitations ................................................................. 5
  1.5 Final Project Methodology .................................................................................. 5
  1.6 Final Project Outline ........................................................................................... 6

CHAPTER 2 LITERATURE REVIEW .................................................................................. 7
  2.1 Illumination .......................................................................................................... 7
  2.2 Phase Firing Angle ............................................................................................. 8
  2.3 Main Components ............................................................................................... 12
    2.3.1 Light Dependent Resistor ........................................................................... 12
      2.3.1.1 Overview ............................................................................................ 12
      2.3.1.2 Characteristic of LDR ...................................................................... 12
    2.3.2 Microcontroller ATmega8535 .................................................................... 13
      2.3.2.1 Overview ............................................................................................ 13
      2.3.2.2 General Description of ATmega8535 .............................................. 14
      2.3.2.3 ATmega8535 Features ..................................................................... 15
      2.3.2.4 ATmega8535 Pin Configuration ...................................................... 17
      2.3.2.5 Interrupts ........................................................................................... 18
      2.3.2.6 Analog-to-Digital Converter (ADC) ............................................... 22
      2.3.2.7 Timer .................................................................................................. 25
    2.3.3 Liquid Crystal Display (LCD) LMB162AFC ............................................ 27
      2.3.3.1 Overview ............................................................................................ 27
2.3.3.2 Display Memory Map ................................................................. 29
2.3.4 Optocoupler ............................................................................. 30
  2.3.4.1 Overview ........................................................................... 30
  2.3.4.2 Liteon 4N26 Optocoupler ................................................. 30
  2.3.4.3 MOC 3020 Optocoupler ..................................................... 31
2.3.5 TRIAC ....................................................................................... 32
  2.3.5.1 Overview ........................................................................... 32
  2.3.5.2 Characteristic of Triac ......................................................... 33
2.3.6 Bipolar Stepper Motor .............................................................. 34
2.3.7 Stepper Motor Driver L293D ..................................................... 35
2.3.8 220 Volt AC Incandescent Lamp ............................................. 37

2.4 Supporting Components .............................................................. 38
  2.4.1 Voltage Regulator ................................................................. 38
  2.4.2 Diode ..................................................................................... 39
  2.4.3 Quartz Crystal 12 MHz .......................................................... 40

CHAPTER 3 DESIGN AND IMPLEMENTATION ............................................. 41
3.1 Basic Concept ............................................................................ 41
3.2 System Block Diagram ............................................................... 41
3.3 Hardware Configuration ............................................................. 42
  3.3.1 Power Supply Unit ................................................................. 42
  3.3.2 Minimum System ATmega8535 .............................................. 44
  3.3.3 Light Sensor .......................................................................... 45
  3.3.4 Zero Crossing Detector ......................................................... 47
  3.3.5 Brightness Controller Circuit ................................................ 49
  3.3.6 LCD - Microcontroller Connection (4bit) ............................ 50
  3.3.7 Stepper Motor Driver ............................................................ 51
3.4 Software ..................................................................................... 53
  3.4.1 Program Flow Chart .............................................................. 53
  3.4.2 Analog-to-Digital Converter Program ................................... 54
  3.4.3 Motor Stepper program ........................................................ 55
  3.4.4 External Interrupt and Timer Configuration .......................... 56
CHAPTER 4 RESULT AND TESTING .................................................................62

4.1 Result ........................................................................................................62
4.2 Testing ........................................................................................................63
  4.2.1 Zero Crossing Detector Testing .............................................................63
  4.2.2 Brightness Controller Circuit ...............................................................64

CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS .........................71

5.1 Conclusions ..............................................................................................71
5.2 Recommendations ....................................................................................71

REFERENCES .................................................................................................72

APPENDIX .........................................................................................................74
LIST OF FIGURES

Figure 1.1 The electromagnetic spectrum ................................................................. 1
Figure 2.1 Illumination ................................................................................................. 7
Figure 2.2 Zero crossing points .................................................................................. 8
Figure 2.3 Zero crossing points of 50 Hz Alternating Current .................................. 8
Figure 2.4 Triac firing ................................................................................................. 9
Figure 2.5 Output voltage vs. conduction time ......................................................... 11
Figure 2.6 Light dependent resistor and its symbol .................................................. 12
Figure 2.7 Resistance as a function of illumination .................................................. 13
Figure 2.8 Microcontroller Atmel ATmega8535 ....................................................... 14
Figure 2.9 ATmega8535 pin configuration ............................................................... 17
Figure 2.10 MCU Control Register ......................................................................... 18
Figure 2.11 MCU Control and Status Register ........................................................ 19
Figure 2.12 General Interrupt Control Register ....................................................... 20
Figure 2.13 General Interrupt Flag Register ............................................................ 20
Figure 2.14 ADC Multiplexer and Selection Register .............................................. 23
Figure 2.15 ADC Control and Status Register A ..................................................... 24
Figure 2.16 Timer Counter Control Register 0 ......................................................... 25
Figure 2.17 Timer/Counter Interrupt Mask Register ............................................... 27
Figure 2.18 LCD LMB162AFC ............................................................................... 28
Figure 2.19 Optocouplers ....................................................................................... 30
Figure 2.20 Liteon 4N26 Optocoupler ..................................................................... 31
Figure 2.21 MOC 3020 ............................................................................................ 31
Figure 2.22 Triac symbol ......................................................................................... 32
Figure 2.23 Characteristic of triac .......................................................................... 33
Figure 2.24 NMB-MAT PM42S-096 ......................................................................... 34
Figure 2.25 Physical shape and pin diagram of L293D ............................................. 35
Figure 2.26 L293D block diagram ........................................................................... 36
Figure 2.27 Incandescent lamp ................................................................................ 37
Figure 2.28 (a) L7805CV, (b) L7812CV, and (c) LM323T voltage regulator .......... 38
Figure 2.29 L78XX and LM323T application circuit .............................................. 39
LIST OF TABLES

Table 1.1 SNI 03-6575-2001 .......................................................... 3
Table 2.1 Interrupt 1 sense control ..................................................... 19
Table 2.2 Interrupt 0 sense control ..................................................... 19
Table 2.3 Interrupt Vectors .............................................................. 21
Table 2.4 Voltage Reference Selections for ADC ................................. 23
Table 2.5 Input Channel Selections .................................................... 24
Table 2.6 ADC prescalar selections ................................................... 25
Table 2.7 Timer prescalar selections .................................................. 26
Table 2.8 Terminal Functions ............................................................ 29
Table 2.9 Stepper table ................................................................. 36
Table 3.1 Stepper table ................................................................. 52
Table 3.2 Inverted input signal .......................................................... 52
Table 3.3 Reversed stepper table ....................................................... 52
Table 4.1 Calculated vs. measured load voltage ................................. 69
EQUATIONS

Equation 2.1 Illumination .............................................................................................................. 7
Equation 2.2 Firing delay .............................................................................................................. 9
Equation 2.3 Firing angle ............................................................................................................ 9
Equation 2.4 Average load voltage ............................................................................................. 10
Equation 2.5 Voltmeter measurement voltage ........................................................................... 10
Equation 2.6 Eight bits ADC .................................................................................................... 22
Equation 2.7 Ten bits ADC ....................................................................................................... 22
Equation 2.8 Maximum timer’s time range ................................................................................ 26
Equation 2.9 Maximum timer’s time range with prescalar ......................................................... 26
Equation 2.10 TCNT0 calculation ............................................................................................... 27
Equation 3.1 LDR output voltage .............................................................................................. 46
Equation 3.2 ADC to Lux normalization ..................................................................................... 46
CHAPTER 1
INTRODUCTION

1.1 Final Project Background

Light is electromagnetic radiation of wavelength that is visible to the human eye and is responsible for the sense of sight. The lower limit of wavelength of light is generally taken between 360 nm and 400 nm and the upper limit between 760 nm and 830 nm [1]. The same as all electromagnetic radiation, light radiates and can travel through empty spaces with unlimited distances at a speed of \(3 \times 10^8\) m/s.

![The electromagnetic spectrum](image)

**Figure 1.1 The electromagnetic spectrum [2]**

Visible light can be considered as the most important thing for human to visually recognize all the objects around. Without light, eyes will never be able to see anything since there is nothing that reflects off the objects. Moreover, in order to see objects clearly, eyes need a proper amount of light to illuminate them. In the daytime, sunlight is considered enough to help people to see any objects in the outside. The situation will be different when people...
are inside a dark room or a room with low light intensity. Because of that, an artificial lighting system is indispensable.

Most of the artificial lighting systems are presented and operated base on the day and night only without taking any consideration of light contribution from the outside, especially in the day. Moreover this setting cannot provide a comfortable working situation inside the room. For example, it will be harder to read or write when it is cloudy outside while lighting inside a room is completely off since the light intensity might significantly drop. For a short period of time it may be neglected, but it will not be the same when this situation happens continuously for a long period.

The previous case could be worse for students who study inside a classroom since they always need a proper lighting during the learning sessions. Studying in a low light intensity will affect the eyes become tired faster than studying under a sufficient light intensity. If this condition is ignored and the eyes are continuously forced to work under insufficient light intensity, it may also cause eye strain. By looking at this problem, then a lighting control system need to be developed in order to provide a constant light level inside a classroom. Besides that, the system may also be integrated with automatic blind control. A well planned blind and lighting system in a classroom will give significant effect to the learning atmosphere and concentration of the students. Besides that, it will also indirectly support the teacher while conducting the class session.

In another case, presentation using projector needs a lower light intensity so that students can see the slides clearly. At the same time, while watching the presentation student may need to make a note and it will be very uncomfortable to write in this condition. Actually, in this special case, lower light intensity is needed only at the front side of the room where the slides are reflected, not the whole room. Therefore, the blind and lighting control system is expected to be able to work in a real time to overcome these problems.

The blind and lighting control that will be developed is expected to be able to maintain light intensity inside the classroom both in normal mode and in presentation mode. As the point of reference, standard light intensity of the classroom will be based on the provision stated in National Standard of Indonesia (SNI) number 03-6575-2001 which is shown in the Table 1.1 below.
Table 1.1 SNI 03-6575-2001 [3]

<table>
<thead>
<tr>
<th>Room Functionality</th>
<th>Illumination Level (Lux)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dwelling House:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terrace</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>Living room</td>
<td>120 ~ 250</td>
<td></td>
</tr>
<tr>
<td>Dining room</td>
<td>120 ~ 250</td>
<td></td>
</tr>
<tr>
<td>Workroom</td>
<td>120 ~ 250</td>
<td></td>
</tr>
<tr>
<td>Bedroom</td>
<td>120 ~ 250</td>
<td></td>
</tr>
<tr>
<td>Bathroom</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>Kitchen</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>Garage</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td><strong>Office:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Director room</td>
<td>350</td>
<td></td>
</tr>
<tr>
<td>Workroom</td>
<td>350</td>
<td></td>
</tr>
<tr>
<td>Computer room</td>
<td>350</td>
<td>Use screen protector to prevent glare caused by screen reflection</td>
</tr>
<tr>
<td>Meeting room</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>Drawing room</td>
<td>750</td>
<td>Use local lighting on the drawing table</td>
</tr>
<tr>
<td>Archives warehouse</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>Active archives warehouse</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td><strong>Educational institution:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classroom</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>Library</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>Laboratory</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>Drawing room</td>
<td>750</td>
<td>Use local lighting on the drawing table</td>
</tr>
<tr>
<td>Canteen</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td><strong>Hotel &amp; Restaurant:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lobby, Corridor</td>
<td>100</td>
<td>Vertical lighting is very important to create a good impression</td>
</tr>
<tr>
<td>Ballroom</td>
<td>200</td>
<td>Lighting system should be made in such a way so that it matches with the situation. Switching and dimming system can be used to give more effects.</td>
</tr>
<tr>
<td>Dining room</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>Cafeteria</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>Bedroom</td>
<td>150</td>
<td>Additional lighting is required over the bed and over the mirror</td>
</tr>
<tr>
<td>Kitchen</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td><strong>Hospital:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wards</td>
<td>250</td>
<td></td>
</tr>
</tbody>
</table>
1.2 Problem Statement

Implementation of control system theory to maintain comfort light intensity inside the classroom by monitoring and controlling both the blind and the lighting system will be the main concern of the research. Related with this, there will be several problems have to be solved in this project and they are:

- How to monitor and detect changing of light intensity inside the classroom;
- How to display the measured light intensity of each sensor;
- How to build a zero-crossing detection circuit of Alternating Current;
- How to control the light intensity inside the classroom so that it will stay still around the determined set points;
- How to build dimmer circuit that has a capability to change or vary the brightness of the lamp;
- How to control the stepper motors that will handle the opening/closing process of the blind.

All of the problems defined above are then used as the basic guideline of the project development.

1.3 Final Project Objectives

Main objective of this final project development is to design an automatic light intensity controller which will be used to maintain a proper lighting system for the classroom with incandescent lamp to provide a comfortable learning atmosphere for the student. In order to achieve this final goal, then the following specific points need to be completed:

1. Calibration of sensing circuit which would detect the light intensity of the environment.
2. Development of zero-crossing detection circuit
3. Development of incandescent lamp brightness controller circuit
4. Development of stepper motor controller circuit
5. Development of controller of the system in order to be able to maintain proper light levels both in normal and presentation mode.
1.4 Final Project Scopes and Limitations

Scopes and limitations of the final project are:

1. Development of the control system is utilizing microcontroller Atmel ATmega8535.
2. Light intensity measurement is utilizing Light Dependent Resistor (LDR).
3. Utilizing three incandescent AC lamps.
4. Utilizing three stepper motors.
5. Miniature of the complete project is including the lamps and the blinds.

1.5 Final Project Methodology

This final project is compiled with several processes. These processes are performed in a consecutive order as in the life cycle development of project initiation, planning, implementation, and lastly evaluation. These processes are then executed through the following steps:

- **Initiation and planning**
  This is basically the preliminary step where topic, background, objective, and scope of the final project are clearly defined. Research is done in order to get as much as information and resources for supporting the idea.

- **Working principle study**
  This is the stage of analyzing and understanding basic concept that will be applied into the final project. It is including the study of sensing circuit, microcontroller, source code, dimming circuit, and the simulation software.

- **Execution (circuit development)**
  This is the stage of both software and hardware development. Furthermore, all of the gathered information from the previous section is implemented here. Besides that, early stage of debugging, bug hunting, circuitry designing, and packaging are also covered in this section.

- **Evaluation**
  This is the final stage of the processes that include circuit testing and troubleshooting activities. The circuit is divided into several blocks to be analyzed and checked one by one in order to ensure reliability of the whole system.
1.6 Final Project Outline

In general, the complete final project report is organized specifically into the following chapters:

Chapter 1: Introduction.

This chapter provides an overview of the overall topic and the aim of the project. It consists of problem background, final project statement, final project objective, final project scope and limitation, final project methodology, and final project outline.

Chapter 2: Literature Study

This chapter covers the basic knowledge and fundamental theories leading to implementation of the final project. Besides that, it also provides a complete review and summary of all pertinent components relevant to the final project.

Chapter 3: Design Implementation

This chapter contains the implementation of all the supporting principles and theories. Besides that, it covers the model and the detailed description of the utilized method and technique. Complete software and hardware configurations are also included in this chapter.

Chapter 4: Result and Discussion

This chapter presents the summary and full analysis of the developed project including the result and all necessary discussions related to it.

Chapter 5: Conclusions and Recommendations.

This chapter consists of conclusions obtained throughout the project. They are outlined according to the final result and the whole achieved objectives. Moreover, it is also containing possible recommendations for improvement in the future projects or applications.
CHAPTER 2
LITERATURE REVIEW

2.1 Illumination

The illumination or intensity of illumination of a surface is defined as the luminous flux per unit area incident on a given surface. It is also called illuminance and represented by $E$ [4]. More generally, it is the quantitative statement of light intensity that hits a given area in correlation with human perception of brightness level. This definition can be described by the Figure 2.1 below.

![Figure 2.1 Illumination](image)

From the Figure 2.1 above, luminous flux that falls on the specific area from a light source of intensity $I$ is expressed by the following equation:

$$E = \frac{I \cdot \cos \theta}{r^2} \quad \text{Equation 2.1}$$

Where:

- $E$ : Illuminance (Lux)
- $I$ : Light source intensity (Candela)
- $\theta$ : Angle of incident
- $r$ : Distance between light source and the area (Meter)
2.2 Phase Firing Angle

Phase firing angle is a time proportioning method to control the amount of power that goes into a load (either resistive or inductive load) by cycle basis. In this concept, there is a term called current conducting time which means the time when current starts to flow into the load until the next nearest zero crossing point. By varying the portion of conduction time will determine the total voltage being delivered. This can be achieved by utilizing a low frequency switch to chop the alternating current sine wave.

In order to get precise gating time, a fixed reference point is required. This is called as zero crossing point. Zero crossing is the instantaneous point when alternating current sinusoidal wave crosses zero volt. Zero crossing occurs twice in every full cycle as shown by the Figure 2.2 below.

![Figure 2.2 Zero crossing points](image)

Time interval of zero crossing points is determined by the frequency of AC sinusoidal wave. In Indonesia, the State Electricity Company (PLN) distributes 220 volts alternating current at a fundamental frequency of 50 Hz for each full cycle which gives a period of 20 ms. Since zero crossing point occurs twice in every full cycle, then it will occur every 10 ms as shown by the Figure 2.3 below.

![Figure 2.3 Zero crossing points of 50 Hz Alternating Current](image)
By knowing the fundamental properties of AC sinusoidal wave, especially the zero crossing point of each full cycle, now the conduction time can be precisely determined for each half cycle. Theoretically, the minimum allowed conduction time is 0 ms and the maximum is 10 ms. The way to implement this concept in practical will be discussed more deeply in the next chapter.

In order to determine the precise conduction time, another term is required and it is called as firing delay. If conduction time is the time when current is fed into the load, then firing delay is the time while current is blocked right after the zero crossing point up to the conduction point or it is equal to the zero crossing intervals subtracted by the conduction time.

\[ \text{Firing delay} = 10\text{ms} - \text{Conduction time} \quad \ldots\ldots \text{Equation 2.2} \]

Theoretically, by knowing the firing delay, then phase firing angle control can be done by simply varying it after each zero crossing point between an interval of 0 up to 10 ms. In this case, the one that actually changes is the firing angle proportional to time delay. As explained previously, a half cycle of 50 Hz AC voltage takes exactly 10 ms. If this interval is divided into 180 time slots, then each time slot will be equal to 1 degree. From this information, if \( \alpha \) is assumed as the firing angle and \( t_d \) is the firing delay, then the relation of these variables is simply determined by the formula below.

\[ \alpha = 18t_d \text{ ms} \quad \ldots\ldots\ldots\ldots\ldots\text{Equation 2.3} \]

Next, the value of \( \alpha \) (firing angle) will be used to determine the average voltage of the load.

\[ \text{Figure 2.4 Triac firing} \]
From the Figure 2.4 above, the average voltage of the load can be calculated by finding the total shaded area under the curve divided by appropriate interval. Since the waveform of Alternating Current is symmetrical, a quarter or a half cycle is considered enough to represents the full cycle. Therefore, it is acceptable to take the first half cycle only which goes from $\alpha$ through $180^\circ$. For alternating current (AC), the input voltage oscillates in a sine wave pattern $v_i(t) = V_p \sin \omega t$ with $V_p$ as the peak voltage. The load average voltage integration is showed below.

\[
V_{av} = \frac{1}{180} \int_{\alpha}^{180} V_p \sin \left(\frac{\pi}{180} t\right) dt
\]

\[
V_{av} = \frac{V_p}{180} \times \frac{180}{\pi} \int_{\alpha}^{180} \sin \theta d\theta
\]

\[
V_{av} = \frac{V_p}{\pi} (\cos \theta)_{\alpha}^{180}
\]

\[
V_{av} = \frac{V_p}{\pi} \left[ (\cos 180) + \cos \alpha \right]
\]

\[
V_{av} = \frac{V_p}{\pi} (1 + \cos \alpha)
\]

\[
V_{av} = \frac{V_p}{\pi} (\cos \alpha + 1) \quad \text{………………Equation 2.4}
\]

Next, if a non-RMS Voltmeter is used for the measurement, it will never give the average voltage but it will multiply the average voltage with a factor of $\left(\frac{\pi}{2\sqrt{2}}\right)$ or 1.1107. Because of this, it can be stated that the reading of non-RMS Voltmeter will follow the result of the formula below.

\[
V_{voltmeter} = V_{av} \times \left(\frac{\pi}{2\sqrt{2}}\right) \quad \text{………………Equation 2.5}
\]
The relation between the average values with the Voltmeter reading values is showed by the Figure 2.5 below.

![Figure 2.5 Output voltage vs. conduction time](image-url)

**Figure 2.5 Output voltage vs. conduction time**
2.3 Main Components

2.3.1 Light Dependent Resistor

2.3.1.1 Overview

Light dependent resistor (LDR) is actually a light level detection device whose resistance changes inversely with the changing of light intensity falling on its surface. Normally, the resistance of LDR is very high but it will drop significantly at the time it is illuminated with light.

When light level falling on the device is of a very low frequency, then there will be only a few free electrons moving in the device that carry electrical charge. This means, if the light intensity is low, light dependent resistor will become a very bad conductor or it can be said that the light dependent resistor is having high resistance when there is not enough light falling on its surface. On the other hand, the brighter the light, the more free electrons move in the semiconductor and light dependent resistor become more conductive. This means the light dependent resistor is having lower resistance when the light is brighter.

2.3.1.2 Characteristic of LDR

Light dependent resistor has a particular property where it can remember the lighting conditions in which they have been stored. Light storage reduces equilibrium time to reach steady resistance values.

Example of light dependent resistor that is used in this project and its symbol is shown by the Figure 2.6 below.

![Figure 2.6 Light dependent resistor and its symbol](image-url)
Relation between LDR’s resistance and light intensity that falls on its surface is described by the following graph.

![Graph showing the relationship between Resistance (kΩm) and Lux](image-url)

**Figure 2.7 Resistance as a function of illumination [5]**

### 2.3.2 Microcontroller ATmega8535

#### 2.3.2.1 Overview

Microcontroller unit (MCU) is an integrated circuit (IC) chip that contains many of the functions found in a standard computer system such as read only memory (ROM), random access memory (RAM), and programmable input/output peripherals. Read only memory (ROM) is a place provided to save the program being executed permanently. Size of the program that can be written into this memory is vary depends on its capacity. Next, random access memory (RAM) is a place where all temporary data and intermediate results during the operation of the microcontroller are stored. When the power supply to the microcontroller unit is off, all the content of this memory is automatically cleared. Besides read only memory and random access memory, each microcontroller unit is completed at least by one register or usually called as port which is connected to the microcontroller pins. The functions or directions (input/output) of those pins can be changed freely according to the needs of the programmer.
Microcontroller is widely designed for embedded applications where only some simple routine tasks are required. All the operations inside the microcontroller unit are processed at high speed. Besides that, it can be programmed to perform various functions according to the system’s requirement. Overall, this small device is found very useful in any automatically controlled application that requires many decisions and calculations.

### 2.3.2.2 General Description of ATmega8535

Microcontroller ATmega8535 belongs to AVR generation with 8 kilobytes of In-System Programmable Flash produced by Atmel. AVR microcontroller itself is based on enhanced RISC (Reduced Instruction Set Computing) architecture. This is proven by its capability to execute instruction only in a single clock cycle. Microcontroller ATmega8535 becomes a very powerful integrated circuit that provides flexibility and cost effective solution to many embedded control applications.

The Atmega8535 AVR is supported with a full suite of program and system development tools including C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits [6]. Figure 2.8 below shows an example of ATmega8535 that is used in this project.

![Figure 2.8 Microcontroller Atmel ATmega8535](image-url)
ATmega8535 is chosen as the main processing unit for this final project because of several reasons. The first, Atmega8535 provides a faster program execution speed since the instruction can be executed within a single clock cycle. This is much faster if it is compared with MCS51 microcontroller family which requires 12 clock cycle to execute an instruction. The second, ATmega8535 has already completed with on-chip Analog-to-Digital Converter, either 8-bit or 10-bit. By using this feature, any detecting devices with analog output signal can be directly interfaced with the ATmega8535 I/O pins without any additional Analog-to-Digital Converter chip. Lastly, this microcontroller is equipped with 32 I/O lines and the programmer can determine the data direction registers (DDRx) freely based on the project requirement. Data direction registers configures the data direction of port pins which means its setting will determines whether port pins will be used for input or output. By simply writing it to 0 will makes the corresponding port pin as input while writing it to 1 will makes the corresponding port pin as output.

2.3.2.3 ATmega8535 Features
According to the datasheet, ATmega8535 is equipped with the following features:

- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 8K Bytes of In-System Self-Programmable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 512 Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 512 Bytes Internal SRAM
  - Programming Lock for Software Security
Peripheral Features

- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Four PWM Channels
- 8-channel, 10-bit ADC
  - 8 Single-ended Channels
  - 7 Differential Channels for TQFP Package Only
  - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only
- Byte-oriented Two-wire Serial Interface
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator

Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

I/O and Packages

- 32 Programmable I/O Lines
- 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MLF

Operating voltages

- 4.5 - 5.5V for ATmega8535

Speed Grades

- 0 - 16 MHz
2.3.2.4 ATmega8535 Pin Configuration

![ATmega8535 Pin Configuration Diagram](image)

**Figure 2.9 ATmega8535 pin configuration [7]**

**Pin Descriptions**

- **VCC**: Digital supply voltage
- **GND**: Ground
- **Port A (PA7…PA0)**: Port A is served as bi-directional input output port. The port pins can also provide internal pull-up resistors (selected for each bit). Besides that, port A output buffers have drive characteristics with both high sink and source capability. In the other hand, Port A can also be used as the analog inputs to the Analog-to-Digital Converter. Any analog inputs that need to be converted to digital signals should be connected to this port.
- **Port B (PB7…PB0)**: Port B, port C, and port D are served as bi-directional input output port. The port pins can also provide internal pull-up resistors (selected for each bit). Besides that, port B, port C, and port D output buffers have drive characteristics with both high sink and source capability.
- **Port C (PC7…PC0)**
- **Port D (PD7…PD0)**
RESET : Reset input. A low pulse that is given to this pin will generate a reset.

XTAL1 : Input to the inverting Oscillator amplifier and input to the internal clock operating circuit

XTAL2 : Output from the inverting Oscillator amplifier.

AVCC : AVCC is the voltage source pin for the microcontroller and the Analog-to-Digital Converter. It should be externally connected to VCC, even if the ADC is not used. At the time the ADC is used, it must be connected to VCC.

AREF : AREF is the analog reference pin for the A/D Converter.

### 2.3.2.5 Interrupts

The ATmega8535 has a capability to detect multiple interrupt sources, both internal and external interrupts. Internal interrupt may happen if an internal peripheral is reaching a certain state, (i.e. timer1 overflow). In contrary, external interrupt is caused by an external source that produces a certain level on a pin. Any interrupt source will affect a jump to a specific location in the memory based on the required action in the program or code. At the time the microcontroller exits from an interrupt state, it will always return to the main program.

On the ATmega8535 chip, there are three pins for external interrupt, which are INT0 (pin 16), INT1 (pin 17), and INT2 (pin 3). This external interrupt can be activated by giving an interrupt signal directly into each of this interrupt pin. The setting for these interrupt signals depicted in the Figure 2.10 are handled by MCU Control Register (MCUCR) and MCU Control and Status Register (MCUCSR).

![Figure 2.10 MCU Control Register](image-url)
MCU Control Register contains configuration bits that determine what kind of signal could trigger the INT0 and INT1.

1. Bit 3,2 – ISC11, ISC10
   Both ISC11 and ISC10 together determine the level and edge on external interrupt INT1 pin. The complete combination is given by the Table 2.1 below.

   **Table 2.1 Interrupt 1 sense control**

<table>
<thead>
<tr>
<th>ISC11</th>
<th>ISC10</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>The low level of INT1 generates an interrupt request.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Any logical change on INT1 generates an interrupt</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>The falling edge of INT1 generates an interrupt</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>The rising edge of INT1 generates an interrupt request.</td>
</tr>
</tbody>
</table>

2. Bit 1,0 – ISC01, ISC00
   Both ISC01 and ISC00 together determine the level and edge on external interrupt INT1 pin. The complete combination is given by the Table 2.2 below.

   **Table 2.2 Interrupt 0 sense control**

<table>
<thead>
<tr>
<th>ISC01</th>
<th>ISC00</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>The low level of INT0 generates an interrupt request.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Any logical change on INT0 generates an interrupt</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>The falling edge of INT0 generates an interrupt</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>The rising edge of INT0 generates an interrupt request.</td>
</tr>
</tbody>
</table>

Next, there is only one bit to set the interrupt sense control 2 (ISC2). This bit will configure what type of event could triggers the INT2.

**Figure 2.11 MCU Control and Status Register**
1. **Bit 6 – ISC2**  
   Writing logic 0 to this bit will make the external interrupt 2 actives at the falling edge while writing logic 1 will make it actives at the rising edge.

External interrupt activation should be done through General Interrupt Control Register (GICR) which contains the interrupt enabled bit for all the external interrupts of the ATmega8535 as shown by the Figure 2.12 below.

![General Interrupt Control Register](image)

**Figure 2.12 General Interrupt Control Register**

1. **Bit 7 – INT1** : External interrupt request 1 will be enabled when this bit set to 1 and the I-bit in the status register is also 1.
2. **Bit 6 – INT0** : External interrupt request 0 will be enabled when this bit set to 1 and the I-bit in the status register is also 1.
3. **Bit 5 – INT2** : External interrupt request 2 will be enabled when this bit set to 1 and the I-bit in the status register is also 1.

The last setting for external interrupt is done through General Interrupt Flag Register (GIFR) as shown by the Figure 2.13 below.

![General Interrupt Flag Register](image)

**Figure 2.13 General Interrupt Flag Register**
In this register, only three bits of the GIFR are utilized while the other bits are reserved. Each of this bit will be set individually to logic 1 when the interrupt event occurs at the corresponding pin. If the I-bit in status register and the corresponding bit in GICR are set to 1, the microcontroller will jump to the corresponding interrupt vector.

1. **Bit 7 – INTF1**: External interrupt flag 1
2. **Bit 6 – INTF0**: External interrupt flag 0
3. **Bit 5 – INTF2**: External interrupt flag 2

Regarding to the interrupt vectors, the Table 2.3 shows a complete list of them from the highest priority to the lowest one.

### Table 2.3 Interrupt Vectors [8]

<table>
<thead>
<tr>
<th>Vector No.</th>
<th>Source</th>
<th>Interrupt Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RESET</td>
<td>External Pin, Power-on Reset, Brown-out Reset</td>
</tr>
<tr>
<td>2</td>
<td>INT0</td>
<td>External Interrupt Request 0</td>
</tr>
<tr>
<td>3</td>
<td>INT1</td>
<td>External Interrupt Request 1</td>
</tr>
<tr>
<td>4</td>
<td>TIMER2 COMP</td>
<td>Timer/Counter2 Compare Match</td>
</tr>
<tr>
<td>5</td>
<td>TIMER2 OVF</td>
<td>Timer/Counter2 Overflow</td>
</tr>
<tr>
<td>6</td>
<td>TIMER1 CAPT</td>
<td>Timer/Counter1 Capture Event</td>
</tr>
<tr>
<td>7</td>
<td>TIMER1 COMPA</td>
<td>Timer/Counter1 Compare Match A</td>
</tr>
<tr>
<td>8</td>
<td>TIMER1 COMPB</td>
<td>Timer/Counter1 Compare Match B</td>
</tr>
<tr>
<td>9</td>
<td>TIMER1 OVF</td>
<td>Timer/Counter1 Overflow</td>
</tr>
<tr>
<td>10</td>
<td>TIMER0 OVF</td>
<td>Timer/Counter0 Overflow</td>
</tr>
<tr>
<td>11</td>
<td>SPI, STC</td>
<td>Serial Transfer Complete</td>
</tr>
<tr>
<td>12</td>
<td>USART, RXC</td>
<td>USART, Rx Complete</td>
</tr>
<tr>
<td>13</td>
<td>USART, UDRE</td>
<td>USART Data Register Empty</td>
</tr>
<tr>
<td>14</td>
<td>USART, TXC</td>
<td>USART, Tx Complete</td>
</tr>
<tr>
<td>15</td>
<td>ADC</td>
<td>ADC Conversion Complete</td>
</tr>
<tr>
<td>16</td>
<td>EE_RDY</td>
<td>EEPROM Ready</td>
</tr>
<tr>
<td>17</td>
<td>ANA_COMP</td>
<td>Analog Comparator</td>
</tr>
<tr>
<td>18</td>
<td>TWI</td>
<td>Two-wire Serial Interface</td>
</tr>
<tr>
<td>19</td>
<td>INT2</td>
<td>External Interrupt Request 2</td>
</tr>
<tr>
<td>20</td>
<td>TIMER0 COMP</td>
<td>Timer/Counter0 Compare Match 2</td>
</tr>
<tr>
<td>21</td>
<td>SPM_RDY</td>
<td>Store Program Memory Ready</td>
</tr>
</tbody>
</table>
2.3.2.6 Analog-to-Digital Converter (ADC)

Analog-to-Digital Converter (ADC) is used to convert the continuous signal of varying voltage level to a digital number or value that represents the portion of that sampled analog signal which will be read by the microcontroller. ATmega8535 is completed already with built-in Analog-to-Digital Converter which has the following features:

- 8-bit and 10-bit Resolution
- 0.5 LSB Integral Non-linearity
- ±2 LSB Absolute Accuracy
- 65 - 260 μs Conversion Time
- Up to 15 kSPS at Maximum Resolution
- 8 Multiplexed Single Ended Input Channels
- 7 Differential Input Channels
- 2 Differential Input Channels with Optional Gain of 10x and 200x(1)
- Optional Left Adjustment for ADC Result Readout
- 0 - VCC ADC Input Voltage Range
- Selectable 2.56V ADC Reference Voltage
- Free Running or Single Conversion Mode
- ADC Start Conversion by Auto Triggering on Interrupt Sources
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

The digital number resulted in the conversion is represented in 8 or 10 bit resolution. An 8-bit converter will generate output with the range of 0 up to $2^8 - 1$ (255) while 10-bit converter will provide output with the range of 0 up to $2^{10} - 1$ (1023). The ADC value of each resolution can be calculated using the following formula.

\[
\text{Eight bits ADC: } \quad ADC = \frac{V_{IN} \cdot 255}{V_{REF}} \quad \text{....................Equation 2.6}
\]

\[
\text{Ten bits ADC: } \quad ADC = \frac{V_{IN} \cdot 1023}{V_{REF}} \quad \text{...................Equation 2.7}
\]
A. ADC Prescalar

Conversion of analog signal into digital requires some regular interval. This interval will be based on the clock frequency. According to the datasheet, ADC will operate properly within a frequency range of 50 kHz to 200 kHz. However, frequency up to 1 MHz does not reduce the ADC resolution significantly. In the other hand, CPU clock frequency might be much higher than this range. Because of that, the clock frequency division is required to achieve the previous frequency range and prescalars are presented as the division factors. For ATmega8535 microcontroller, some of the predefined division factors are 2, 4, 8, 16, 32, 64, and 128.

B. ADC Registers

To use the ADC peripheral of ATmega8535, certain registers need to be configured. The first register is ADC Multiplexer and Selection Register (ADMUX).

1. **Bit 7:6 – REFS1:0**

These bits are the reference selection bits. They are used to determine the reference voltage of the ADC conversion. There are four combination can be made by these bits.

<table>
<thead>
<tr>
<th>REFS1</th>
<th>REFS0</th>
<th>Voltage Reference Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>AREF, Internal Vref turned off</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>AVCC</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Internal 2.56V Voltage Reference</td>
</tr>
</tbody>
</table>
2. **Bit 4:0 – MUX 4:0**

These bits are used to choose the ADC channel that will be used since there are 8 ADC channels. Table 2.5 below shows the detailed settings.

<table>
<thead>
<tr>
<th>MUX 4..0</th>
<th>Single Ended Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>ADC0</td>
</tr>
<tr>
<td>00001</td>
<td>ADC1</td>
</tr>
<tr>
<td>00010</td>
<td>ADC2</td>
</tr>
<tr>
<td>00011</td>
<td>ADC3</td>
</tr>
<tr>
<td>00100</td>
<td>ADC4</td>
</tr>
<tr>
<td>00101</td>
<td>ADC5</td>
</tr>
<tr>
<td>00110</td>
<td>ADC6</td>
</tr>
<tr>
<td>00111</td>
<td>ADC7</td>
</tr>
</tbody>
</table>

The second register is ADC Control and Status Register A (ADCSRA). The ADCSRA register is as follows.

![Figure 2.15 ADC Control and Status Register A](image)

#### Table 2.5 Input Channel Selections

1. **Bit 7 – ADEN**

ADC Enable is the bit that needs to be set so the ADC is enabled.

2. **Bit 6 – ADSC**

ADC Start Conversion bit should be set to 1 before and during the conversion progress. It should return to 0 when ADC conversion is done.

3. **Bit 4 – ADIF**

This bit is used to check whether the conversion has finished or not. It will be set to 1 automatically when the conversion is complete.
4. Bit 2:0 – ADPS 2:0

These bits determine what prescalar is used. This can be determined by selecting one of the combinations listed below.

<table>
<thead>
<tr>
<th>ADPS2</th>
<th>ADPS1</th>
<th>ADPS0</th>
<th>Division</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>128</td>
</tr>
</tbody>
</table>

Table 2.6 ADC prescalar selections

The third registers for ADC configuration are ADC Low byte (ADCL) and ADC High byte (ADCH). These registers are the place for ADC conversion result. Using CodevisionAVR compiler, the 10 bit conversion result can be taken by writing ADCW and if 8 bit mode is used, than it can be taken by writing ADCH.

2.3.2.7 Timer

Timer is a standard feature that most of microcontroller has. Timer contains a register that always increment its value at a predetermined rate start from zero up to its maximum limit. An 8-bit timer has a maximum value of 255 while 16 bit timer has a maximum value of 65535. In addition, timer operation does not need CPU intervention which means it works independently of CPU.

ATmega8535 is completed with two 8-bit resolution timers (timer0 and timer2) plus one 16-bit resolution timer1. Since the project is utilizing timer0 only, the other two timers will not be discussed here. As usual, several registers need to be configured to utilize the timer0 feature. The first register is Timer Counter Control Register 0 (TCCR0).

Figure 2.16 Timer Counter Control Register 0
Bit 2, 1, and 0 of this register are used to determine the prescalar of the timer. In this case prescalar is needed to make a longer range timer. For example, by utilizing 8 bit timer0 with 8 MHz external crystal then the maximum time range of this timer will be as follow.

\[
\text{tMAX} = \frac{1}{f\text{CLK}} \times (FFh+1) \quad \text{.................... Equation 2.8}
\]

\[
= \frac{1}{8000000} \times (255+1)
= 0.032\text{ms}
\]

Where:

* \( t\text{MAX} \): Maximum timer’s time range
* \( f\text{CLK} \): Crystal frequency
* \( FFh \): Timer’s maximum value (for 8 bit timer it is 255)

Next, a longer time range can be got by using a prescalar of 1024 (N = 1024) as shown by the calculation below.

\[
\text{tMAX} = \frac{1}{f\text{CLK}} \times (FFh+1) \times N \quad \text{...................... Equation 2.9}
\]

\[
= \frac{1}{8000000} \times (255+1) \times 1024
= 32.768\text{ms}
\]

In order to choose which prescalar will be used, one of the following bit combinations of the TCCR0 register can be implemented.

**Table 2.7 Timer prescalar selections**

<table>
<thead>
<tr>
<th>CS02</th>
<th>CS01</th>
<th>CS00</th>
<th>Prescalar</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Timer stopped</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>No prescaling</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>256</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1024</td>
</tr>
</tbody>
</table>
The next register that needs to be configured is Timer/Counter Interrupt Mask Register (TIMSK). The TIMSK register is as follow.

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read/Write</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>Initial Value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 2.17 Timer/Counter Interrupt Mask Register

By enabling bit 0 – TOIE0 (Timer Overflow Interrupt Enable) will increment a variable count at every interrupt (Overflow).

The last setting that needs to be done is for the Timer/Counter Register 0 (TCNT0). It is better to write its value in hexadecimal. TCNT0 can be found by using the following formula.

\[
TCNT0 = \left(1 + FFh\right) - \frac{T_{\text{timer}} \times f_{\text{CLK}}}{N}
\]

\[\text{………… Equation 2.10}\]

Where:

* \(T_{\text{timer}}\) : Needed timer range(s)

2.3.3 Liquid Crystal Display (LCD) LMB162AFC

2.3.3.1 Overview

Liquid Crystal Display (LCD) is a low-power, flat-panel display made of liquid containing crystals that are affected by electric current, sandwiched between filtering layers of glass or plastic [9]. LCD is based on the principle of blocking light, rather than emitting it, therefore it require less power to operate. Since it does not emit light directly, LCD always needs a light source and classified into passive display. Some of LCDs may use only ambient light source such as sunlight to operate.
This final project is utilizing 16 x 2 LCD LMB162AFC as shown by the Figure 2.18 above. 16 x 2 means the LCD has a capability to display 16 characters per line by 2 lines. Therefore, the maximum character that can be displayed on this type of LCD is 32. LCD LMB162AFC requires three control lines and eight lines data bus for 8-bit mode operation while in 4-bit mode operation it requires only three control lines and four lines for the data bus. The three control lines here are pin E, RS, and RW. E is the enable pin in which it always set to high before the data is sent to the LCD. Soon after that, the other control lines are set and all the data are placed into the data lines. When everything is ready, E is set back to low and this transition will generate a command to LCD to call all the previous data both in control and data lines. RS is the register select pin. All the commands or instructions will be sent when RS is low. In the other hand, all necessary data that will be displayed on the LCD are sent when RS is high. Lastly, RW is read/write pin. When RW is low, that means the data lines are still sending data (write) and if RW is high, that means the data lines are in the read stage.

The complete terminal functions of LCD LMB162AFC are described by the Table 2.8 below.
Table 2.8 Terminal Functions [10]

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>Pin Name</th>
<th>I/O</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VSS</td>
<td>Power</td>
<td>Power supply, Ground (0V)</td>
</tr>
<tr>
<td>2</td>
<td>VDD</td>
<td>Power</td>
<td>Positive power supply</td>
</tr>
<tr>
<td>3</td>
<td>NC</td>
<td>-</td>
<td>No Connection, leave open</td>
</tr>
<tr>
<td>4</td>
<td>RS</td>
<td>Input</td>
<td>Register Select</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RS=HIGH: transferring display data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RS=LOW: transferring instruction data</td>
</tr>
<tr>
<td>5</td>
<td>R/W</td>
<td>Input</td>
<td>Read / Write Control bus:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R/W=HIGH: Read mode selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R/W=LOW: Write mode selected</td>
</tr>
<tr>
<td>6</td>
<td>E</td>
<td>Input</td>
<td>Data Enable</td>
</tr>
<tr>
<td>7</td>
<td>DB0</td>
<td>I/O</td>
<td>Bi-directional tri-state Data bus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In 8 bit mode, DB0 ~ DB7 are in use</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In 4 bit mode, DB4 ~ DB7 are in use, DB0~DB3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>leave open</td>
</tr>
<tr>
<td>14</td>
<td>DB7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>BLA</td>
<td>Power</td>
<td>Backlight positive supply</td>
</tr>
<tr>
<td>16</td>
<td>BLK</td>
<td>Power</td>
<td>Backlight negative supply</td>
</tr>
</tbody>
</table>

2.3.3.2 Display Memory Map

There are two main memory areas in the LCD module for display. Those memory areas are Character Generator RAM (CGRAM) and Display Data RAM (DDRAM). Character Generator Ram is specially used for storing the user defined characters (5x8 dots font). All the characters here are free to be modified by the user. Once the power supply is off, all of the contents of this memory will be lost. On the other hand, Display Data RAM is the place where ROM characters are written. It defines the position of the character on the LCD (column and row).
2.3.4 Optocoupler

2.3.4.1 Overview

Optocoupler or also called as opto-isolator is an electronic device designed to transfer electrical signals by utilizing light waves to provide coupling with electrical isolation between its input and output [11]. It contains a pair of transmitter and receiver which are separated by a transparent barrier. The transmitter side consists of an infrared light emitting diode (LED). If compared with ordinary LED, infrared LED is having a better endurance to the visible signal. The light that is emitted by infrared LED cannot be seen through naked eyes. Next, the receiver side contains photo-sensor that has a capability to detect the emitted light by infrared LED. Several examples of optocouplers are shown in the Figure 2.19 below.

![Figure 2.19 Optocouplers](image)

Essentially, optocoupler is a switching and protective device that specially designed to separates the control circuit with the power circuit in order to protect them against damage that could be caused by current surges or another abnormal transition. Its working principle is quite simple. When the infrared LED in the transmitter side is conducting current, it will emit light that will pass through the transparent separator. Once the photo-sensor in the receiver side detects this emitted light, it will directly convert it back into electrical signal either high or low.

2.3.4.2 Liteon 4N26 Optocoupler

Liteon 4N26 belongs to NPN type optocoupler. It is a general purpose optocoupler which consists of a gallium arsenide infrared emitting diode that drive a silicon phototransistor in a 6-pins package. The Physical shape and its symbol are shown in the Figure 2.20 below.
2.3.4.3 MOC 3020 Optocoupler

The MOC 3020 is optically isolated, non zero crossing triac driver devices. It contains a GaAs infrared emitting diode and a light activated silicon bilateral switch, which functions like a triac. This device is specially designed for interfacing between electronic control signals and power triac with a capability to control both resistive and inductive loads for 220-240 VAC operations. The Physical shape and symbol of MOC 3020 are shown in the Figure 2.21 below.
2.3.5 TRIAC

2.3.5.1 Overview

The triac or Triode for Alternating Current is a three-terminal semiconductor device that is used to switch AC outputs of relatively high current at high voltages [12]. Triac conducts electricity in either direction when triggered by a positive or negative signal at the gate. By having the capability to control the current flow in both halves of the cycle, as such this three terminals semiconductor device is very ideal solution for alternating current switching applications. Triac switching action is much faster and provides more accurate control if it is compared with conventional relays.

![Triac Symbol](image)

As a bilateral device, the terminals of triac are simply named MT1, MT2, and G since the terms anode and cathode are having no meaning in this case. MT1 and MT2 are current carrying terminals while G is the gate terminal used for triggering the triac. Generally MT1 is used as the reference point and current is flowing between MT1 and MT2 in both directions.
2.3.5.2 Characteristic of Triac

In general, the characteristic of triac is shown by the Figure 2.23 below.

![Figure 2.23 Characteristic of triac](image)

The first quadrant (I) is the region where terminal MT2 is always positive with respect to MT1 while the third quadrant (III) is the region where MT2 is always negative. The gate (G) is the control terminal of the triac that will set the firing angle according to the triggering signal. Since the triac can conduct current in either direction, then there will be four possibilities for triac triggering modes, which are:

- Quadrant I: Positive voltage and positive gate current
- Quadrant II: Positive voltage and negative gate current
- Quadrant III: Negative voltage and negative gate current
- Quadrant IV: Negative voltage and positive gate current

When alternating current is applied into both of the triac’s terminals (MT1 & MT2) while the gate (G) is open, the triac will never become conductive (OFF) till $V_{DRM}$ is reached. $V_{DRM}$ is the break over voltage of the device and is the highest voltage that the triac may be allowed to block in either direction [14]. By applying various currents into the gate, the
break over voltage may be lowered. The more current is injected, the lower the break over voltage and triac become conductive (ON). Once the triac is triggered, it will remain “ON” till the voltage in MT1 and MT2 reach 0 volt. When it is “OFF”, then it will also remain “OFF” till the gate is triggered back and break over voltage is reached.

2.3.6 Bipolar Stepper Motor

Stepper motor is a type of electric motor that rotates step by step with fixed and very accurate increments. It divides a complete rotation into some equal steps and these steps are commonly very small. Two phase stepper motor comes in two different electromagnetic coils arrangements, they are uni-polar stepper motor and bipolar stepper motor. The uni-polar stepper motor has two coils per phase while the bipolar stepper motor has only one coil per phase. In this final project, the uni-polar stepper motor is not used so it will not be discussed here.

Bipolar stepper motor has two independent set of coils inside of it. Besides that, it is usually completed also with four wires with different colors. Both of the dependent set of the coils can be distinguished by measuring one by one the resistance between each wire. The two wires can be said in pairs if they are having the same resistance. The unpaired wires will give an infinite resistance.

This final project utilizes NMB-MAT PM42S-096 bipolar stepper motor as the roller blind driver. This motor has a maximum driving voltage of 24 V. It rotates 3.75° per step which mean to accomplish one full rotation, it should move 96 steps. Examples of NMB-MAT PM42S-096 bipolar stepper motor and its symbol are shown in the Figure 2.24 below.

![Figure 2.24 NMB-MAT PM42S-096](image)
2.3.7 Stepper Motor Driver L293D

In order to operate properly, the stepper motor required a driver to strengthen or buffer the microcontroller signal so it will be strong enough to drive such a large load. For this final project, three L293D motor drivers are used to drive three stepper motors. This type of chip provides the following features:

- Wide supply-voltage range: 4.5 V to 36 V
- Separate input-logic supply
- Internal ESD protection
- Thermal shutdown
- High-noise-immunity inputs
- Output current 600 mA per channel
- Peak output current 1.2 A per channel

L293D is a dual H-bridge, high-current, four channel driver. This 16-pin integrated circuit (IC) has a capability to receive control signal with low-current from the microcontroller and then provide a higher-current signal to drive or control the motor’s rotation. The physical shape and pin diagram of L293D is shown by the Figure 2.25 below.

![Figure 2.25 Physical shape and pin diagram of L293D][15]
The block diagram of L293D is shown by the Figure 2.26 below.

![L293D Block Diagram](image)

**NOTE:** All output diodes are internal in L293D.

**Figure 2.26 L293D block diagram**

From the block diagram above, the enable pin 1 and 9 should be set to high for motor to rotate. This means when pin 1 and 9 are high, the associated drivers are enabled. In contrary, if these enable pin is set to low, the associated drivers are disabled which makes the motor stop rotating and the output stays in the high-impedance state. Next, the stepper motor operation can be controlled through pin 2, 7, 10, and 15 (these pins are connected to the microcontroller). Here is how they should be pulsed to produce a single cycle.

<table>
<thead>
<tr>
<th>Table 2.9 Stepper table [16]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STEP 1</strong></td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>STEP 1</td>
</tr>
<tr>
<td>STEP 2</td>
</tr>
<tr>
<td>STEP 3</td>
</tr>
<tr>
<td>STEP 4</td>
</tr>
</tbody>
</table>
Pin 3, 6, 11, and 14 are the output pins that should be connected to the motors according to the motor’s wire pairs as explained in the previous section. Pin 8 is the motor power supply which can range from 4V to 34V. It is better to separate this power supply with the power supply for the logic circuit. Lastly, pin 16 is the power for the working of the logic control and it is better to use 5V.

2.3.8 220 Volt AC Incandescent Lamp

Incandescent lamp is a lamp that produces light by heating up a filament of wire inside a bulb with an electric current, causing incandescence. The glass bulb containing the filament is filled with a nonreactive gas [17]. Brightness level of an incandescent lamp is strongly affected by its input voltage so thus it can be simply changed by only varying the input voltage. Figure 2.27 below shows the construction of an incandescent lamp.

![Incandescent lamp diagram]

**Figure 2.27 Incandescent lamp [18]**
2.4 Supporting Components

2.4.1 Voltage Regulator

Voltage regulator is a specially designed electronic device used to maintain a consistent voltage level, independent of the loads. In some applications, voltage regulator is required to keep the voltages within any specific ranges that some electrical components may tolerate. Besides that, voltage regulator can also eliminate power surges and brownout that possibly harm any sensitive components contained in the electronic circuit. Therefore, any voltage regulators will keep the electrical circuits operate efficiently and lasting longer.

In general, voltage regulator operates by comparing the actual output voltage to a fixed reference voltage. Any detected difference is directly amplified and used to control the regulation element to minimize or eliminate the voltage error. For example, if the load current is increasing or the input voltage is reducing, then the output voltage may become very low. When this voltage error is detected, the regulation element will operate in such a way, up to a specific point, so that the higher voltage is produced at the output. This can be done either by dropping less of the input voltage or by drawing input current for a longer period. By doing this, a stable voltage level will always be achieved and safe to be used.

In this project, three-terminal voltage regulators are used in the power supply unit to provide constant voltage at 5 V in which they stabilize the DC voltages that will be fed into the MCU minimum system and the other logic circuits. These voltage regulators are L7812CV, L7805CV, and LM323T. Each type employs internal current limiting, thermal shut-down, and safe area protection, making it essentially indestructible [19]. Physical appearance and application circuit of these regulators are shown in the Figure 2.28 and Figure 2.29 below.

Figure 2.28 (a) L7805CV, (b) L7812CV, and (c) LM323T voltage regulator
2.4.2 Diode

A diode is like a one-way street: it only allows electricity to flow in one direction and not the other. This means that by definition diodes are polarized, meaning that they can only be placed in a circuit in one direction [20]. Generally, a diode has a low resistance (ideally zero) to electric current that allows it to flows in one direction and a high resistance (ideally infinite) that block the current in reverse direction. By having this property, general purpose diode is found to be very useful if functioned as rectifier which converts the alternating current into direct current. This can be done by arranging four diodes in a special configuration called as a diode bridge. It will operate to transform alternating current into direct current using both phases of the alternating current.

In this final project, the two bridge rectifier arrangements are utilizing four 1N5392GP diodes and four 1N5408 diodes. The physical appearance of these diodes and their arrangement are shown in the Figure 2.30 below.

Figure 2.29 L78XX and LM323T application circuit

Figure 2.30 Diode 1N5392GP, 1N5408, and their arrangement
2.4.3 Quartz Crystal 12 MHz

A quartz crystal is an electronic oscillator that capable to provide a constant frequency output. The size of quartz crystal in a crystal oscillator will determine the final frequency of oscillations. Generally its frequency is very stable and range from a few tens of kilohertz up to tens of megahertz. Physical shape of quartz crystal and its symbol is shown by the Figure 2.31 below.

![Quartz Crystal Image](image)

(a) (b)

**Figure 2.31 Quartz crystal: (a) physical appearance and (b) the symbol.**

Quartz crystal always produces an output that repeats regularly. By having this property, quartz crystal is normally used as the frequency determining element in the oscillator circuit.
CHAPTER 3
DESIGN AND IMPLEMENTATION

3.1 Basic Concept

The overall system is expected to work automatically to monitor, control, and maintain the light intensity level inside the room. All light dependent resistors will measure the light intensity falling on their surfaces by converting the resistances into voltages so the microcontroller can read them. All of these values will be displayed on the LCD and also compared with the user predetermined setpoint. In this system, the light intensity can be controlled through two controlled devices, which are curtains and lamps. Curtains will have higher priority than the lamps. It means when the light intensity level falls below the predetermined setpoint, the curtain will be opened first and if curtain already opened and light intensity does not reach the set point yet, then the lamp will start brightening bit by bit till it reaches the maximum brightness. In contrary, if the light intensity level measurement shows that it is above the setpoint, the lamp will be dimmed bit by bit till it completely OFF. When the lamp is OFF but the light intensity still greater than the setpoint, then the curtain will be completely closed.

In the first chapter, it is also explained that there will be two modes, normal mode and presentation mode. For switching the mode, it should be manually selected by the user through a switch. The whole concept is still following the previous explanation. The difference is only on the setpoint of the light intensity level at the front side, the rest are exactly the same.

3.2 System Block Diagram

In general, system block diagram is shown by the Figure 3.1 below. Base construction of this final project is utilizing the microcontroller unit Atmel ATmega8535. It is the brain that handles and controls the whole actions, start from head up to the tail of the system. This integrated circuit will operate according to the predetermined program that is downloaded into it.
There are two power supply units that provide 5 volts DC with different current ratings. A lower current rating power supply is used to supply all of the logic circuits while the higher one is specially used to supply the stepper motors only.

All of these components and circuits will be discussed more deeply, one by one, and completed with their basic working principle, calculation, and specific function.

3.3 Hardware Configuration

3.3.1 Power Supply Unit

As mentioned previously, the power supply units are built to supply direct current into all the logic circuits and the stepper motors. This circuit is specifically designed in order to produce two DC outputs, which are 5 volts up to 1.5 Amperes and 5 volts up to 3 Amperes. The 5 volts DC 1.5 A will be used to supply the minimum system of microcontroller unit, light dependent resistors (LDRs), liquid crystal display (LCD), and the optocouplers, while the 5 volts DC 3 A will be used to supply the stepper motors only. The stepper motors require a separated power supply since they need high current to operate together. When the load need a higher current than the power supply can produce, the output voltage of the power supply might drop significantly and disturb the other circuit. If this problem occur, it...
may cause the logic circuits completely OFF and/or do not operate properly because do not get enough power. Therefore, to avoid this problem then two power supplies are built separately.

The two power supplies are L7805CV based and LM323T based. The L7805CV based power supply can provide 5V 1.5A DC while LM323T based power supply can provide 5V 3A DC.

The complete circuits of both power supply units are shown by the Figure 3.2 below.

![Power supply circuit](image)

(a)

(b)

Figure 3.2 Power supply circuit, (a) for logic circuits (b) for stepper motors
In Figure 3.2 (a), one Ampere step down transformer is used to decrease the $220\text{V}_\text{AC}$ source into $12\text{V}_\text{AC}$. Next, four diodes 1N5392GP is used as a bridge rectifier to rectify the alternating current into direct current. However, the rectifiers output is still having ripple voltage and that is the reason why $2200 \ \mu\text{F}$ smoothing capacitor is installed right after these rectifiers configuration. Well, the 330 nF and 100nF capacitors are chosen according to the datasheet and they are included in the voltage regulation circuits to maximize the stability of the LM7812 and LM7805 ICs to produce either 12 volts or 5 volts direct current (DC), max 1.5 A. Next, 100 uF capacitors are used as output bulk capacitors. Lastly, LED connection on the tail of the power supply circuit is used only as an indicator in order to make sure that there is voltage passing through it.

In Figure 3.2 (b), the application is quite similar. The differences are only on the values or ratings of the components being used. Here the transformer is 3A step down transformer and the bridge rectifier is utilizing 1N5408 diodes. The voltage regulating circuit itself is utilizing LM323T which provides 5V DC up to 3A in the output. Next, a quite big output bulk capacitor is used ($2200 \ \mu\text{F}/50\text{V}$) since the stepper motors might be switching ON/OFF frequently. When they are switching ON, they will draw more current.

In addition, note that one diode on the right side of the bridge circuit in Figure 3.2 (a) is omitted in the Figure 3.2 (b). This is because the zero crossing detection circuit is connected directly to this L7805CV based power supply. Since the input of zero crossing circuit is taking the pulsating DC output from the bridge, then this diode should be installed before the filtering capacitor C1.

### 3.3.2 Minimum System ATmega8535

Minimum system is the standard circuit configuration required by the microcontroller unit so that it can operate properly. Actually, ATmega8535 is completed already with internal oscillator so that it does not need any external resonator or crystal as the CPU clock source. Simply connect the VCC, AVCC, and GND pins to the power supply and it is ready to operate. However, the maximum value of the internal oscillator is only 8MHz and need to set the fuse bit to change it. Therefore, it is better to use external crystal for precision reason and build a complete minimum system for the microcontroller so that it can operate properly. The minimum system of ATmega8535 is shown by the Figure 3.3 below.
Pin 12 and pin 13 are connected with crystal 12 MHz and two capacitors 22 pF. They are set together as a clock source for the microcontroller unit. The chosen crystal value will affect the microcontroller time precision. Next, pin 9 is connected with 4.7 \( \mu \)F capacitor and 10 k\( \Omega \) resistor. Besides that, it is also connected with a push button that acts as reset button. When this button is pushed, the microcontroller will go back to its initial setting which means the microcontroller will start to read and process the program from the beginning.

### 3.3.3 Light Sensor

Light dependent resistor (LDR) is used as the illumination measurement unit. It is built in combination with 10 k\( \Omega \) resistor in a voltage divider configuration as shown by the Figure 3.4 below. This light sensor circuit will provide information related to the illumination value around its surface directly into the microcontroller unit. The output of this measurement circuit is an analogue signal. Since the project utilizes ATmega8535 which is completed already with analogue to digital converter (ADC), then external ADC IC is not required. The microcontroller operation will be based on the input given by these LDRs and comparing them with the predetermined setpoint.
The configuration of light intensity measurement circuit is shown by the Figure 3.4 below.

By setting up the components into voltage divider configuration, the circuit will directly obtain a voltage that varies with the measured values of light intensity, instead of a resistance change. By adapting the concept, then the output voltage of this configuration can be calculated using the formula below.

\[
V_{\text{out}} = 5x \left( \frac{10^3}{LDR + (10 \times 10^3)} \right)
\]

............... Equation 3.1

Base on the formula above, when light level is low, it means that output voltage will also low since the resistance of LDR is high and vice versa. Next, the analog value resulted by this circuit configuration will be converted by Analog-to-Digital Converter into digital value. For this project, 10-bit ADC is used instead of 8-bit ADC in order to get more accurate digital value. Since 10-bit ADC is used, it means the digital value will range from 0 to 1023. Next, this range is normalized in order to convert it into Lux. In this case the Lux range is assumed start from 0 to 1000. By knowing these properties, then the Lux can be found using the formula below.

\[
\text{Lux} = \frac{\text{ADC} \times 1000}{1023}
\]

............... Equation 3.2
3.3.4 Zero Crossing Detector

Zero crossing detector is a set of circuit that has a capability to detect instantaneous point at which there is no voltage present or at the time when the alternating current sinusoidal wave is changing from positive into negative and vice versa. As explained in the previous chapter, for the case of 50 Hz alternating current (AC) waveform, zero crossing normally occurs twice for each full cycle as shown by the Figure 3.5 below.

![Figure 3.5 50 Hz AC waveform with zero crossing points](image)

Zero crossing detection in this project is utilizing a step down transformer, five diodes 1N5392GP, optocoupler NPN, and two resistors. Practically, this circuit can be connected directly into the power supply circuit so it utilizes the same transformer and bridge rectifier circuit as the power supply used. Complete configuration of these components is shown in the Figure 3.6 below.

![Figure 3.6 Zero crossing detector](image)
Firstly, the 12V alternating current is rectified into direct current by the bridge rectifier circuit. The output of this rectifier is not pure direct current, but it is a pulsating positive DC voltage as shown by the waveform in the Figure 3.7 below.

![Figure 3.7 Pulsating positive DC voltage](image)

This pulsating positive voltage is used to drive the infrared LED inside the NPN optocoupler and a 1.2 kΩ resistor is used to limit the current through it. By doing this, the optocoupler LED will stay ON for most of the cycle except at the time when the AC sinusoidal wave crosses 0 V (or about to cross 0 V since the LED already OFF). If this LED is ON, then the transistor inside the optocoupler will also ON. This condition will pull the external interrupt pin of the microcontroller low, in this case external interrupt 0 PIND.2 (EXT INT0). In contrary, when the LED is OFF, then the external interrupt PIND.2 will be pulled high. Theoretically, based on the working principle above, output of this zero crossing detector should look like the red waveform in the following figure.

![Figure 3.8 Output of zero crossing detector](image)
In the Figure 3.8 above, the output of zero crossing detector is always low during most of the cycle and it will go high only if the AC sinusoidal wave is about to cross zero. This transition is called as rising edge. When AC sinusoidal wave has passed the 0 V, the output of zero crossing detector will go back to low state again and this transition from high to low is called as falling edge.

By the presence of this interrupt signal into PIND.2, continuously the microcontroller will interrupted every 10 ms and it will be able to recognize when the wave pass through the zero line. Next, this will be used as reference point to start the triac firing delay precisely.

### 3.3.5 Brightness Controller Circuit

Brightness controller circuit has the responsibility to maintain the brightness of the lamp. This circuit consists of triac BT139, optocoupler MOC3020 as a triac driver, and several current limiting resistors. Here MOC3020 is used also as a protection for the whole logic circuits if there is unwanted high voltage or high current occur in the load. By using this optocoupler, all of the logic circuit is completely separated from the AC load (lamp). A complete circuit of the brightness controller is shown by the Figure 3.9 below.

![Figure 3.9 Brightness controller circuit](image)

Triac gating through the MOC3020 is done by giving a low pulse into optocoupler pin 2. Active low is chosen so that the microcontroller does not need to provide voltage for the circuit. This condition is generally known as sinking where the digital I/O only provides ground needed in the circuit. Sinking is chosen instead of sourcing because of the optocoupler current rating consideration. When a low pulse is given to optocoupler pin 2, then the bilateral switch inside MOC3020 will directly ON and triac is latched until it reaches the zero crossing point.
3.3.6 LCD - Microcontroller Connection (4bit)

Actually there are different ways of interfacing 16x2 LCD with the microcontroller. They are 4-bit mode and 8-bit mode. In this project, 4-bit mode is chosen instead of 8-bit mode. This is because the 8-bit mode needs a total of 11 data lines (3 control lines plus 8 lines for the data bus) while 4-bit mode requires only 7 data lines (3 control lines plus 4 lines for the data bus). On 4-bit mode, approximately 50 µs is needed to transmit a character. However, this very short period of time will be unseen by the human eyes.

In this case the LCD data lines RS, RW, E, DB4, DB5, DB6, and DB7 are connected into PORTC of the microcontroller ATmega8535, VDD is given a 5 V supply, while VSS and VEE are both connected to the ground. The complete connection of 16x2 LCD with the microcontroller in 4-bit mode is shown by the Figure 3.10 below.

![Figure 3.10 LCD interfacing (4-bit mode)](image-url)
### 3.3.7 Stepper Motor Driver

Stepper motor driver is used as an interface between microcontroller with the stepper motor itself. The driver being used is dual H-bridge, high current, 16-pins L293D. In this final project, since the motor requires quite high current to operate than it is better to separate this motor driver with the other logic circuits. For this purpose TLP521 optocoupler is used to separate the motor driver with the microcontroller. Related with this, the power supply for the motor and the driver are also separated so that in case the load draws high current, then the other circuits will not be disturbed. The complete configuration is shown by the Figure 3.11 below.

![Figure 3.11 Bipolar stepper motor interfacing](image)

From the Figure 3.11 above, it is clearly seen that the microcontroller is completely separated with the controlled load and its driver. Note that active low is chosen so that the microcontroller is sinking instead of sourcing. When microcontroller gives low pulses into the optocoupler, then the internal LED will ON and emitting infrared light into the receiver on the other side. This condition will automatically pull the output signal low. In the other hand, if microcontroller gives high pulses then the internal LED will OFF and pull the output signal high.
Actually, by utilizing NPN optocoupler configuration will invert the input signals into its opposite condition. According to the explanation given in previous chapter, to make a single cycle then the following step must be fulfilled.

<table>
<thead>
<tr>
<th>Table 3.1 Stepper table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pin 2</strong></td>
</tr>
<tr>
<td><strong>STEP 1</strong></td>
</tr>
<tr>
<td><strong>STEP 2</strong></td>
</tr>
<tr>
<td><strong>STEP 3</strong></td>
</tr>
<tr>
<td><strong>STEP 4</strong></td>
</tr>
</tbody>
</table>

Since the utilization of optocoupler is inverting signal into its opposite condition, then the input signals should be inverted first as shown by the Table 3.2 below. By applying this condition, then the input signals of L293D driver will follow the value listed in the stepper table above.

<table>
<thead>
<tr>
<th>Table 3.2 Inverted input signal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input 1</strong></td>
</tr>
<tr>
<td><strong>STEP 1</strong></td>
</tr>
<tr>
<td><strong>STEP 2</strong></td>
</tr>
<tr>
<td><strong>STEP 3</strong></td>
</tr>
<tr>
<td><strong>STEP 4</strong></td>
</tr>
</tbody>
</table>

In contrary, active low method has indirectly manipulated this condition. It means, by applying active low method, the input signal that is given into the optocoupler will look like the same as the output signal without any inversion. Since active low method is already chosen for this project, the input signal can be simply set and follow the stepper table above. In addition, to change the direction of the rotation can be done by simply reverse the step. It means step 4 become step 1, step 3 become step 2, step 2 become step 3, and step 1 become step 4. This reversed step order is shown by the Table 3.3 below.

<table>
<thead>
<tr>
<th>Table 3.3 Reversed stepper table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pin 2</strong></td>
</tr>
<tr>
<td><strong>STEP 1</strong></td>
</tr>
<tr>
<td><strong>STEP 2</strong></td>
</tr>
<tr>
<td><strong>STEP 3</strong></td>
</tr>
<tr>
<td><strong>STEP 4</strong></td>
</tr>
</tbody>
</table>
3.4 Software

3.4.1 Program Flow Chart

Figure 3.12 Program flow chart
3.4.2 Analog-to-Digital Converter Program

The part of code to access the internal ADC is showed by the example below. This ADC value will be used to determine the triac firing delay in the next section.

```c
#include <stdio.h>
#include <delay.h>

unsigned int ADC;

#define ADC_VREF_TYPE 0x40

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

void main(void)
{
    // IO initialization
    PORTA=0xFF;
    DDRA=0x00;

    // ADC initialization
    ADMUX=ADC_VREF_TYPE & 0xff;
    ADCSRA=0x84;
    SFIOR&=0xEF;

```
while (1)
{
    ADC = read_adc(0);
}

3.4.3 Motor Stepper program

The following code is used to control the stepper motor. This code will be combined with
triac firing code in the next section to maintain the light intensity level.

#include <mega8535.h>
#include <delay.h>
#define ALEFT 0
#define ARIGHT 1

unsigned int motor_stepA[4] = {0b00110000,0b00101000,0b01001000,0b01010000};

void turn_A(unsigned char directionA, int stepA)
{
    static unsigned char i = 0;
    unsigned int j = 0;
    while(j <= stepA)
    {
        j++;
        if(directionA)
        {
            i++;
        }
        else
        {
            i--;
        }
    }
if(i == 255)
{
    i = 3;
}
else
    if(i == 4)
    {
        i = 0;
    }
PORTB = (PORTB & 0b11110000) | (motor_stepA[i] & 0b00001111);
delay_ms(50);
}

void main(void)
{
    PORTB = 0x00;
    DDRB = 0xFF;
    while(1)
    {
        turn_A(ARIGHT, 23);
    }
}

3.4.4 External Interrupt and Timer Configuration

In this project, external interrupt is used to activate the timer to produce low pulse so that the triac is triggered precisely. When zero crossing occurs, there will be two edges, rising edge and falling edge. For a proper operation, external interrupt is set in such a way so that it activates the timer at the falling edge and ignores the rising edge. Next, when the timer is started, it will always increment by 30 µs till the firing pulse is cleared. In this part, timer will be set to fire the triac at a specific time delay with a 30 µs firing pulse width. The example of how external interrupt is set to start the timer to fire the triac plus stepper motor is given below.
```c
#include <mega8535.h>
#include <stdio.h>
#include <delay.h>
#include <stdint.h>
#define ALEFT 0
#define ARIGHT 1

volatile unsigned int ADC;
unsigned int count = 0;
volatile unsigned int x = 280;
unsigned char motor_stepA[4] = {0b00000101, 0b00000110, 0b00001010, 0b00001001};

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // Reset Timer
    TCNT0 = 0;
    // Start Timer 0
    TCCR0 = 0x02;
}

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    // Reinitialize Timer 0 value
    TCNT0 = 0xD3;
    // Increment count every 50 us
    count++;

    // Front lamp
    if(count == x)
    {
        PORTD.4 = 0;
    }
}
```
else
    if(count == x + 1)
        {
            PORTD.4 = 1;
            // Clear count
            count = 0;
            // Stop the timer
            TCCR0 = 0;
            // Reset the timer
            TCNT0 = 0;
        }
    }

#define ADC_VREF_TYPE 0x40
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

void turn_A(unsigned char directionA, int stepA)
{
    static unsigned char i = 0;
    unsigned int j = 0;
    while(j <= stepA)
    {

j++;
if(directionA)
{
    i++;
}
else
{
   i--;
}
if(i == 255)
{
    i = 3;
}
else
  if(i == 4)
  {
    i = 0;
  }
PORTB = (PORTB & 0b11110000) | (motor_stepA[i] & 0b00001111);
delay_ms(50);

void get_adc()
{
    ADC=read_adc(0);
}

void change_intensity()
{
    static unsigned int motor_flagA = 0;
    // If light intensity below the setpoint, open blinds, and shorten the firing delay
    if((ADC <= 250) && (motor_flagA < 20))
    

turn_A(ARIGHT,1);
motor_flagA++;
}
#asm("cli")
if((PORTD.4 == 1) && (ADC <= 250) && (motor_flagA == 20))
{
    if(x > 1)
    {
        x = x - 1;
    }
}

// If light intensity greater than the setpoint, increase the firing delay, and close blinds
if((PORTD.4 == 1) && (ADC >= 300))
{
    if(x < 280)
    {
        x = x + 1;
    }
}
#asm("sei")
if((ADC >= 300) && (x == 280) && (motor_flagA > 0))
{
    turn_A(ALEFT,1);
motor_flagA--;
}
}

void main(void)
{
    // Input/Output Ports initialization
    PORTA=0x80;
    DDRA=0x78;
    PORTB=0x00;
DDRB=0xFF;
PORTC=0x00;
DDRC=0x00;
PORTD=0xF0;
DDRD=0xF0;

// Timer/Counter 0 initialization
TCCR0 = 0x02;
TCNT0 = 0xD3;
OCR0 = 0x00;

// External Interrupt(s) initialization
GICR|=0x40;
MCUCR=0x02;
MCUCSR=0x00;
GIFR=0x40;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x01;

// ADC initialization
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;
SFIOR&=0xEF;

// Global enable interrupts
#asm("sei")

while (1)
{
    get_adc();
    change_intensity();
}

CHAPTER 4
RESULT AND TESTING

4.1 Result

Figure 4.1 below shows the appearance of the final configuration of the device.

![Image of a device with labeled parts]

Figure 4.1 Up side view of the device

1. Step down transformers
2. High current power supply for the three stepper motors only
3. Power supply for all of the logic circuit
4. Stepper motor drivers configuration
5. ATmega8535 minimum system
6. AC lamps brightness controller
7. Liquid Crystal Display (LCD)
8. Light Dependent Resistor (LDR)
9. Mode selection button
10. AC lamps
11. Bipolar stepper motors
12. Blinds

The device has successfully operated based on the flow chart given in the previous chapter. In the normal mode, if the light intensity drops below the predetermined setpoint then the stepper motor opens the blind while the lamp stays OFF. If the blind has already opened and the light intensity still below the setpoint, the lamp will ON and get brighter until the setpoint is reached. On the other hand, if light intensity is greater than the predetermined setpoint, the lamp will be dimmed until the setpoint is reached (in this case the lamp might be completely OFF). If the light intensity is still greater than the setpoint and the lamp is OFF, then the stepper motor will close the blind. Next, in the presentation mode the operating way is quite similar. The difference is only on the setpoint for the first lamp (the lamp that will be placed in front side of the room) while the rest are exactly the same.

4.2 Testing

4.2.1 Zero Crossing Detector Testing

Before the zero crossing detector circuit is connected with the microcontroller, testing is done to ensure the interrupt signal that will be given to the interrupt pin. Using oscilloscope, the output signal looks like a repeated positive pulse with a fix time interval. It is showed by the Figure 4.2 below.

Figure 4.2 Zero crossing detector output
Theoretically, zero crossing point must occur every 10 ms as explained in the previous section. However this is impossible to be achieved in practical. The infrared LED of NPN optocoupler will goes off a moment before and after the AC sinusoidal wave crosses zero because the power is too small at this time interval. Because of this, the NPN optocoupler’s output signal will consume a small amount of time at every zero crossing point. Base on estimation, it is around ±0.5 ms on each edge of the zero crossing point. By knowing it, then the firing pulse must not be generated 0.5 ms before and after the zero crossing point.

4.2.2 Brightness Controller Circuit

Brightness level of the lamp can be determined by chopping the AC sinusoidal wave that is fed into it. In order to know how much voltage is exactly delivered into the lamp, several testing are performed by varying the firing delay of the triac. The results are shown in the figures below.

1. Firing delay: 0 ms

![Figure 4.3 Output of 0ms firing delay](image-url)
2. **Firing delay: 1 ms**

![Image of waveforms and lamp](image1.png)

*Figure 4.4 Output of 1ms firing delay*

3. **Firing delay: 2 ms**

![Image of waveforms and lamp](image2.png)

*Figure 4.5 Output of 2ms firing delay*
4. *Firing delay: 3 ms*

![Image](image1.png)

Figure 4.6 Output of 3ms firing delay

5. *Firing delay: 4 ms*

![Image](image2.png)

Figure 4.7 Output of 4ms firing delay
6. **Firing delay: 5 ms**

![Image of firing delay output]

Figure 4.8 Output of 5ms firing delay

7. **Firing delay: 6 ms**

![Image of firing delay output]

Figure 4.9 Output of 6ms firing delay
8. *Firing delay: 7 ms*

![Figure 4.10 Output of 7ms firing delay](image)

9. *Firing delay: 8 ms*

![Figure 4.11 Output of 8ms firing delay](image)
10. Firing delay: 9 ms

Figure 4.12 Output of 9ms firing delay

The comparison between calculated load voltage and measured load voltage are shown by the Table 4.1 below. The calculation is done by the help of Equation 2.4 and Equation 2.5 given in chapter 2. Note that the load voltage measurements are done ten times for each conduction time. Therefore the measured values listed below are the average of the ten times measurements.

Table 4.1 Calculated vs. measured load voltage

<table>
<thead>
<tr>
<th>Conduction Time (ms)</th>
<th>Calculated Load Voltage (V)</th>
<th>Measured Load Voltage (V)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.38</td>
<td>4.83</td>
<td>10.22</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>20.11</td>
<td>4.24</td>
</tr>
<tr>
<td>3</td>
<td>45.34</td>
<td>42.63</td>
<td>5.98</td>
</tr>
<tr>
<td>4</td>
<td>76</td>
<td>73.48</td>
<td>3.32</td>
</tr>
<tr>
<td>5</td>
<td>110</td>
<td>106.63</td>
<td>3.06</td>
</tr>
<tr>
<td>6</td>
<td>143.99</td>
<td>138.62</td>
<td>3.73</td>
</tr>
<tr>
<td>7</td>
<td>174.66</td>
<td>167.21</td>
<td>4.27</td>
</tr>
<tr>
<td>8</td>
<td>198.99</td>
<td>189.85</td>
<td>4.59</td>
</tr>
<tr>
<td>9</td>
<td>214.62</td>
<td>205.1</td>
<td>4.44</td>
</tr>
<tr>
<td>10</td>
<td>220</td>
<td>209.47</td>
<td>4.79</td>
</tr>
</tbody>
</table>
Unfortunately there are some errors in practical result. Several analysis have been done and found that the width of zero crossing pulses significantly affect the error of phase firing angle. As explained previously, the zero crossing pulses take about ± 0.5ms before and after zero crossing point. However this condition cannot be eliminated since it is belong to one of the zero crossing detector properties. By the existence of these pulses will prevent the triac firing to start exactly at the zero crossing points, but a moment after zero crossing points. Beside zero crossing pulses, there are also other factors that contribute in the error and they are:

1. Unstable or fluctuating source voltage
2. Misreading of the measurement tools
3. Precision of the measurement tools
4. Component tolerance

Figure 4.13 Measured voltage vs. calculated voltage
CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In conclusion, this final project entitled “Blind and Lighting Control to Maintain Comfort Light Intensity of The Classroom Utilizing Microcontroller ATmega8535 has already successfully done its function as follows.

1. The light intensity sensing circuits has successfully detected the light intensity level and the microcontroller display it on the LCD in the unit of Lux.
2. The zero crossing detector circuit has successfully sent output signal that indicate the points when alternating current sinusoidal wave crosses zero (0) volt into the microcontroller.
3. The brightness controller circuit has successfully control the brightness level of the lamps.
4. All of the bipolar stepper motors have operated smoothly to open and close the blind and they are integrated well with the other circuit.

Implementing this device will automatically maintain a comfortable light intensity level inside the classroom both in normal mode and presentation mode. In addition, the student can study conveniently without need to care of the outside light intensity level.

5.2 Recommendations

However, future improvements are very important in order to produce high quality technology and there is still a space for that purpose in this project. There are two recommended improvements which are:

1. The light intensity conversion into Lux is done by normalizing the ADC value with respect to the LDR resistance. It will be better if the conversion is done by comparing the ADC value with a conventional Lux-meter and find the formula for it.
2. Use another type and more accurate sensors to detect the light intensity level to improve the device’s reliability.
REFERENCES


APPENDIX

/********************
Project: Blind and lighting control to maintain comfort light intensity of the classroom utilizing microcontroller ATmega8535
Version: 01
Date: 3/3/2013
Author: I Gusti L. Wahyudi Indrawan
Company: President University
Comments: Final Project

Chip type: ATmega8535
Program type: Application
AVR Core Clock frequency: 12.000000 MHz

#include <mega8535.h>
#include <stdio.h>
#include <delay.h>
#include <stdint.h>
define ALEFT 0
define ARIGHT 1
define BLEFT 0
define BRIGHT 1
define CLEFT 0
define CRIGHT 1

unsigned char buf[10];
volatile unsigned long int ldr1;
volatile unsigned long int ldr2;
volatile unsigned long int ldr3;
unsigned int count = 0;
volatile unsigned int x = 280;
volatile unsigned int y = 280;
volatile unsigned int z = 280;

unsigned char motor_stepA[4] = {0b00000101, 0b00000110, 0b00001010, 0b00001001};
unsigned char motor_stepB[4] = {0b01010000, 0b01100000, 0b10100000, 0b10010000};
unsigned char motor_stepC[4] = {0b00101000, 0b00110000, 0b01010000, 0b01001000};

// Alphanumeric LCD functions
#include <lcd.h>
#asm
 .equ __lcd_port=0x15 ;PORTC
#endasm
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // Reset Timer
    TCNT0 = 0;
    // Start Timer 0
    TCCR0 = 0x02;
}

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    // Local variables initialization
    static unsigned int pulse_flag = 0;
    // Reinitialize Timer 0 value
    TCNT0 = 0xD3;
    // Increment count every 30 us
    count++;

    // Front lamp
    if(count == x)
    {
        PORTD.4 = 0;
    }
    else
    if(count == x + 1)
    {
        PORTD.4 = 1;
        pulse_flag++;
    }

    // Middle lamp
    if(count == y)
    {
        PORTD.5 = 0;
    }
    else
    if(count == y + 1)
    {
        PORTD.5 = 1;
        pulse_flag++;
    }

    // Back lamp
    if(count == z)
    {
        PORTD.6 = 0;
    }
    else
if(count == z + 1)
{
    PORTD.6 = 1;
    pulse_flag++;
}

// Stop timer when all pulses are completed
if(pulse_flag >= 3)
{
    // Clear flag
    pulse_flag = 0;
    // Clear count
    count = 0;
    // Stop the timer
    TCCR0 = 0;
    // Reset the timer
    TCNT0 = 0;
    // Check ADC
}

#define ADC_VREF_TYPE 0x40
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

void turn_A(unsigned char directionA, int stepA)
{
    static unsigned char i = 0;
    unsigned int j = 0;

    while(j <= stepA)
    {
        j++;

        if(directionA)
        {
            i++;
        }
    }
else
{
    i--; 
}

if(i == 255)
{
    i = 3;
}
else
    if(i == 4)
        { 
            i = 0; 
        }

    PORTB = (PORTB & 0b11110000) | (motor_stepA[i] & 0b00001111);
    delay_ms(50);
}

void turn_B(unsigned char directionB, int stepB)
{
    static unsigned char k = 0;
    unsigned int l = 0;

    while(l <= stepB)
    {
        l++;

        if(directionB)
        { 
            k++; 
        } 
        else
        
            { 
                k--; 
            } 

        if(k == 255)
        { 
            k = 3;
        }
        else
            if(k == 4)
            { 
                k = 0; 
            } 

        PORTB = (PORTB & 0b00001111) | (motor_stepB[k] & 0b11110000);
        delay_ms(50);
void turn_C(unsigned char directionC, int stepC)
{
    static unsigned char m = 0;
    unsigned int n = 0;
    while(n <= stepC)
    {
        n++;
        if(directionC)
        {
            m++;
        }
        else
        {
            m--;
        }
        if(m == 255)
        {
            m = 3;
        }
        else
        if(m == 4)
        {
            m = 0;
        }
        PORTA = (PORTA & 0b10000111) | (motor_stepC[m] & 0b01111000);
        delay_ms(50);
    }
}

void display_adc()
{
    static unsigned long int adc1;
    static unsigned long int adc2;
    static unsigned long int adc3;
    adc1 = read_adc(0);
    ldr1 = adc1 + 40;
    ldr1 = (ldr1*1000)/1023;
    lcd_gotoxy(0,0);
    sprintf(buf,"L1:%d%d%d%d",ldr1/1000,(ldr1%1000)/100,(ldr1%1000%100)/10,
            ldr1%1000%100%10);
    lcd_puts(buf);
adc2 = read_adc(1);
ldr2 = (adc2*1000)/1023;
lcd_gotoxy(9,0);
sprintf(buf,"L2:%d%d%d%d",ldr2/1000,(ldr2%1000)/100,(ldr2%1000%100)/10,
ldr2%1000%100%10);
lcd_puts(buf);

adc3 = read_adc(2);
ldr3 = adc3 + 15;
ldr3 = (ldr3*1000)/1023;
lcd_gotoxy(0,1);
sprintf(buf,"L3:%d%d%d%d", ldr3/1000,( ldr3%1000)/100,( ldr3%1000%100)/10,
ldr3%1000%100%10);
lcd_puts(buf);

void change_intensity()
{
    static unsigned int motor_flagA = 0;
    static unsigned int motor_flagB = 0;
    static unsigned int motor_flagC = 0;
    volatile unsigned int lower_limit;
    volatile unsigned int upper_limit;

    if(PINA.7 == 0)
    {
        lower_limit = 100;
        upper_limit = 150;
    }
    else
    {
        lower_limit = 250;
        upper_limit = 300;
    }

    // If light intensity below the setpoint, open blinds, and shorten the firing delay
    if((ldr1 <= lower_limit) && (motor_flagA < 20))
    {
        turn_A(ARIGHT,1);
        motor_flagA++;
    }
    #asm("cli")
    if((PORTD.4 == 1) && (ldr1 <= lower_limit) && (motor_flagA == 20))
    {
        if(x > 1)
        {
            x = x - 1;
        }
// If light intensity greater than the setpoint, increase the firing delay, and close blinds
if((PORTD.4 == 1) && (ldr1 >= upper_limit))
{
    if(x < 280)
    {
        x = x + 1;
    }
}

#asm("sei")

if((ldr1 >= upper_limit) && (x == 280) && (motor_flagA > 0))
{
    turn_A(ALEFT,1);
    motor_flagA--;
}

// If light intensity below the setpoint, open blinds, and shorten the firing delay
if((ldr2 <= 250) && (motor_flagB < 20))
{
    turn_B(BRIGHT,1);
    motor_flagB++;
}

#asm("cli")

if((PORTD.5 == 1) && (ldr2 <= 250) && (motor_flagB == 20))
{
    if(y > 1)
    {
        y = y - 1;
    }
}

// If light intensity greater than the setpoint, increase the firing delay, and close blinds
if((PORTD.5 == 1) && (ldr2 >= 300))
{
    if(y < 280)
    {
        y = y + 1;
    }
}

#asm("sei")

if((ldr2 >= 300) && (y == 280) && (motor_flagB > 0))
{

turn_B(BLEFT,1);
motor_flagB--;
}

// If light intensity below the setpoint, open blinds, and shorten the firing delay
if((ldr3 <= 250) && (motor_flagC < 20))
{
    turn_C(CRIGHT,1);
    motor_flagC++;
}

#ifdef("cli")
if((PORTD.6 == 1) && (ldr3 <= 250) && (motor_flagC == 20))
{
    if(z > 1)
    {
        z = z - 1;
    }
}
#endif

// If light intensity greater than the setpoint, increase the firing delay, and close blinds
if((PORTD.6 == 1) && (ldr3 >= 300))
{
    if(z < 280)
    {
        z = z + 1;
    }
}
#ifdef("sei")
if((ldr3 >= 300) && (z == 280) && (motor_flagC > 0))
{
    turn_C(CLEFT,1);
    motor_flagC--;
}
#endif

void main(void)
{
    // Input/Output Ports initialization
    PORTA=0x80;
    DDRA=0x78;
    PORTB=0x00;
    DDRB=0xFF;
    PORTC=0x00;
    DDRC=0x00;
    PORTD=0xF0;
}
DDRD=0xF0;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 1500.000 kHz
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0 = 0x02;
TCNT0 = 0xD3;
OCR0 = 0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Falling Edge
GICR|=0x40;
MCUCR=0x02;
MCUCSR=0x00;
GIFR=0x40;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x01;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;
SFIOR&=0xEF;

// Alphanumeric LCD initialization
// Characters/line: 16
lcd_init(16);

// Global enable interrupts
#asm("sei")

while (1)
{
    display_adc();
    change_intensity();
}