CHAPTER 1

INTRODUCTION

1.1 General Introduction

Lighting is an essential element of any building. Proper lighting improves the aesthetics of indoor spaces and provides illumination for tasks and activities. An efficient lighting strategy, including natural daylighting, can provide proper levels of illumination and reduce energy costs. Achieving these benefits requires integration of natural and artificial lighting sources early in the building design process.

Lighting controls help to deal with energy conservation, give consumers more choice and make lighting systems more flexible. Most common lighting method today is the on/off switch. There are few other available technologies including manual dimming, photo sensor, occupancy sensor, timer and centralized control. Occupancy sensor can reduce waste lighting in unoccupied space.

Artificial lighting represents a major component of energy consumption, accounting for a major component of all energy consumed worldwide. Proper lighting can enhance task performance, while poorly designed lighting may result in energy waste and adverse health effects from over-illumination.

Efficient use of natural lighting may help in energy saving as well provides consumers with proper illumination.

With proper control system, both artificial lightings and natural illumination can be used to reduce the energy consumption.
1.2 Problem Statements

Electricity is becoming more expensive while its consumption is getting higher. In order to reduce electric consumption in commercial building, we need effective lighting system. Artificial lighting contributes major of electrical energy consumption worldwide. People often forget to turn off the light in their house when they go out so the room is lit even when there is no need for it. Sometimes the lights continue to be on until the room is already illuminated by natural light. Consumer often overlooked the energy that is wasted by letting this happen. Besides, this act also leads to increase in electrical bill.

There is also a problem of the use of excessive amounts of light (also known as over-illumination). Over-illumination can be defined as presence of light intensity beyond the need for a specified activity. Providing light to an already bright room may leads to over-illumination.

Most of the building such as office in our country is equipped with many windows. Level of natural light is quite high during working hours on most days but the lamps are switch on and finally this has brought electricity wasting.

Hence we need an intelligent lighting system, that use the day lighting of the room and an automatic occupancy system to shut-off when nobody in.
1.3 Objectives

This project main task it to design a smart light control system with room occupancy indicator. Various methods can be used to implement an effective and efficient lighting control system. The objectives of this project are:

- To automatically turn on the lights when a person has entered the room and automatically turn off the lights when the room is empty.
- To control the room’s brightness (control the intensity of brightness, window blind).
- To count the number of people in the room.
- To enable the system operates in certain modes.

1.4 Scope of Project

The scope of this project would involve in designing, implementing and testing of the smart light control system with room occupancy indicator. This project will be divided into two parts which are hardware and software development. Thus, the scopes of this project are:

- Write the program of the system using (Code Vision AVR) software and burn the program into the microcontroller, AT mega 16.
- Simulate the program using (Proteus 7.7) simulation software.
- Interface between microcontroller and sensors (LDR photo sensor, IR sensor).
- Connect the LED’s to the microcontroller as output and control the brightness intensity of the LED’s (adjusting the duty cycle of PWM).
- Connect DC motor to the microcontroller (through L293D chip) as output.
- The room model consists of 1 door and 1 window.
- The system will operate in certain modes.
- The modes can be selected through push buttons.
- Testing the overall system.
1.5 Thesis Outline

This thesis is organized in five chapters. Chapter one gives an overview of the project and the introduction of the project. Chapter two describes the literature review on SMART LIGHT CONTROL SYSTEM. Chapter three elaborates the design and implementation in this project. Chapter four discusses about the result this project. Lastly, chapter five includes the conclusions of the Final Year Project.
CHAPTER 2

LITERATURE REVIEW

2.1 Daylight Harvesting

Daylight harvesting is the term used for a control system that reduces the energy consumption by reducing the use of artificial lighting like electric lamps when natural daylight is available in the building controls and active daylight industries. The most important way to do this is to use natural light as far as possible by designing the form of the building so as to maximize the use of daylight as the functional building lighting during daylight hours [2]. The systems are typically designed to maintain a minimum recommended light level that varies according to the needs and use of the space.

Light sensors were used to detect current light level or brightness and integrate the daylight system with the lighting system in all daylight harvesting systems. The artificial lighting will operate only when the day lighting is insufficient.

The signal from the photo sensor will be interpreted by a lighting control system letting the lights to shut off or dim as necessary in order to reduce the use of artificial lighting. By using daylight harvesting system, energy consumption from artificial lighting can be reduced but it depends on where the system is deployed and its usage.

2.2 LED

A light-emitting diode (LED) is a semiconductor light source. LEDs are used as indicator lamps in many devices and are increasingly used for other lighting.

Light-emitting diodes are used in applications as diverse as aviation lighting, digital microscopes, automotive lighting, advertising, general lighting, and traffic signals.
The wavelength and efficiency of the LED’s light is varies depend on the colour of LED.

<table>
<thead>
<tr>
<th>Color</th>
<th>Wavelength range (nm)</th>
<th>Typical efficacy (lm/W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>$620 &lt; \lambda &lt; 645$</td>
<td>72</td>
</tr>
<tr>
<td>Red-orange</td>
<td>$610 &lt; \lambda &lt; 620$</td>
<td>98</td>
</tr>
<tr>
<td>Green</td>
<td>$520 &lt; \lambda &lt; 550$</td>
<td>93</td>
</tr>
<tr>
<td>Cyan</td>
<td>$490 &lt; \lambda &lt; 520$</td>
<td>75</td>
</tr>
<tr>
<td>Blue</td>
<td>$460 &lt; \lambda &lt; 490$</td>
<td>37</td>
</tr>
</tbody>
</table>

Table 2.1: wavelength and efficiency of coloured LED

Green LED was selected and used, because is more efficient and easy to noticed by human eyes.

2.3 Light Dependent Resistor (LDR)

LDR is one of many kind of photo sensor that have been widely used in many applications in our everyday life. LDR is a low cost light sensor that detects the presence of light or the light level. LDR converts the light signal into electrical energy. Initially, the LDR
resistance is high but when it’s exposed to light, its resistance drops. Thus, LDR resistance depends on the light level. In other words, the resistance drops when the light level increases and vice versa.

![Figure 2.3: LDR](image)

2.4 Push button

A push-button or simply button is a simple switch mechanism for controlling some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often biased switches, though even many un-biased buttons (due to their physical nature) require a spring to return to their un-pushed state. The "push-button" has been utilized in calculators, push-button telephones, kitchen appliances, and various other mechanical and electronic devices, home and commercial.

![Figure 2.4: push button](image)
2.5 Microcontroller

A microcontroller is a single-chip computer. Micro suggests that the device is small, and controller suggests that it is used in control applications. It is designed for embedded application usually for such automatically controlled products and devices. Most microcontrollers have common features such as listed below.

- Central Processing Unit (CPU) or processor.
- Programmable Input/output Peripherals.
- Memory such as ROM, RAM, EPROM and EEPROM.
- Clock generator.
- Analog-to-Digital converter.
- In-circuit programming/debugging support.

 Although, by integrating all of that features onto a single chip may increase the cost of that chip, it may decrease the cost of the whole embedded system because the size is reduce compared to the design that uses separate components for each features. Originally, microcontroller is programmed by using assembly language but nowadays, there’s also can be programmed by high-level programming language such as C programming language.

2.5.1 ATMEGA 16

It is high-performance, low-power Atmel 8-bit AVR RISC-based microcontroller.
It has the following features:

– 32 Programmable I/O Lines

– 40-pin PDIP

- 16Kbytes of programmable flash memory,

- 1Kbytes internal SRAM,

- 512 Bytes EEPROM,

- 8-channel 10-bit A/D converter,

– Four PWM Channels

– Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes

– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

– Byte-oriented Two-wire Serial Interface

– Programmable Serial USART

– Master/Slave SPI Serial Interface

- JTAG interface for on-chip debugging.

- The device supports throughput of 16 MIPS at 16 MHz

- Operates between 4.5-5.5 volts.

By executing instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.
2.6 DC Motor

The basic principle of an electrical motor is it converts electrical energy into mechanical energy. It is based on simple electromagnetism, which is by interacting two magnetic fields part, the static magnetic field part with the rotating magnetic field part. Every DC motor has six basic parts which is axle, rotor or armature, stator, commutator, field magnets and brushes.

When the motor is supplied with voltage, the rotor will rotate because the polarities of the energized winding and the stator magnets are misaligned. It will rotate until it is almost aligned with the stator's field magnets. The brushes move to the next commutator contacts, and energize the next winding as the rotor reaches alignment. The rotor will continue rotating as the rotation reverses the direction of current through the rotor winding which lead to the change of the rotor's magnetic field.
2.7 Motor Driver

There are a wide range of motor driver designs available to meet most any robotic need. They all vary in the requirements they try to meet.

There are some things to look for in a driver design, based on your requirements:

- Output power capability.
- Number of circuit connections.
- Cost and availability of parts.
- Size.
- Reversibility (most - but not all - motor driver designs allow your motor to run in both forward and reverse).
- Output voltage.

H-bridge or sometimes referred as full bridge is an electronic circuit that provides the capability to apply voltage across a load in either direction. It usually used in a motor driver.
A basic H-Bridge has 4 switches, relays, transistors, or other means of completing a circuit to drive a motor. In the above diagram, the switches are labeled A1, A2, B1, B2. Since each of the four switches can be either open or closed, there are $2^4 = 16$ combinations of switch settings. Many are not useful and in fact, several should be avoided since they short out the supply current (e.g., A1 and B2 both closed at the same time). There are four combinations that are useful:

<table>
<thead>
<tr>
<th>Closed switches</th>
<th>Polarity</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 &amp; A2</td>
<td>forward</td>
<td>motor spins forward</td>
</tr>
<tr>
<td>B1 &amp; B2</td>
<td>reverse</td>
<td>motor spins backward</td>
</tr>
<tr>
<td>A1 &amp; B1</td>
<td>brake</td>
<td>motor acts as a brake</td>
</tr>
<tr>
<td>None</td>
<td>free</td>
<td>motor floats freely</td>
</tr>
</tbody>
</table>

Table 2.2: useful combination of switches

2.7.1 L293D

L293D is a 16-pin integrated chip that provides motor speed and direction control. It has Wide Supply-Voltage Range (4.5 V to 36 V). It has up to 600mA of output current and high immunity to noise because its logic ‘0’ input voltage is up to 1.5v. To control the motor using this chip, ‘Vss’ pin must be supplied with 5V and ‘Vs’ pin must be supplied with external power supply input to power up the motor up to 36V.
This device has 2 h-bridge which means it can control 2 motors. To adjust the speed of the motor, we can apply pulse-width modulation (PWM) through the enable pin. In addition, the enable pin also can be used to select which motor operates.

2.8 Similar Previous Projects

2.8.1 Automatic Light Control System

This project was completed in 2009 by Mohd Amir Hamzah. His project is based on comparing the surrounding light and the light intensity in the room to determine whether to switch on the light or not. The project encounter some problems such as the system operates for limited time because of the component heat up pretty fast when the system is turn on but overall, the project met all its objectives and was successfully implemented.

2.8.2 Intelligent Lighting System for Commercial Building

The project was done by Sukor Bin Muhammad in 2009. The intelligent lighting system designed by him only focused on using occupancy sensor and photosensor to automatically turn on and off the lights. Thus, he further suggests using dimming circuit to complete the function of an intelligent lighting system.

2.8.3 Home Automation Lighting System

This project was done by Goh Lin Chian in 2009. Based on the project title Itself, this project used automation concept to control the lights in the house. The system used motion detector to detect passerby in the area to automatically turn on and off the lights. He also designed a graphic user interface (GUI) to control the lights from personal computers and also used local area network (LAN) to interconnect other computer in any part of the house.
CHAPTER 3

METHODOLOGY

3.1 Block Diagram

As shown in the figure below, this project contains several essential parts needed to make the system works. Rather than implement and test the whole system, the project was divided into few smaller subprojects so that it is easier to troubleshoot when any problems come up. Each subproject will be discussed in detail throughout this chapter.

![System’s block diagram](image)

3.2 Hardware Implementation

3.2.1 Door Sensor

LDR sensors will be placed at the door as it is for interpreting the initial input where a person might enter or exit the room. The Atmega microcontroller will receive the sensors output and determine the occupancy of the room. Two LDR sensors will be used instead of one so that the system can determine which direction the person is going. Since range of detection is neglected, the LDR sensors will be only used for direction detection only.

Two lazars beam were forced at two LDRs surface, which were put on door. The first LDR then the second LDR in order to detect the occupancy of the room.
In the first situation, the lazar beam which forced at the first LDR was cut, then the lazar beam forced at the second LDR was also cut, that means that person entered the room.

In opposite situation (cut the second then the first lazar beam forced at LDRs), that means the room was left.

![Figure 3.2: door sensor](image)

The door sensor works as counter (counting number of persons in room), and detect the room occupancy.

The system is active when the room is occupied, but when the room is empty the system is idle.

Initially the two lazars beams is fall on the surfaces of the two LDRs. the values of resistance of LDRs is low relative to the resistance of resistor which connected with it in serial, that makes the voltage across LDRs low relative to the resistor.

If lazars beams are crossed by someone, the resistance of the LDRs is become high relative to the resistors. Then the voltage across the LDRs is also become high relative to the voltage across resistors.

The voltage across LDRs was fed to the microcontroller. When this variation of LDRs voltage occurs, the microcontroller translates that as someone passed the door (entering or exiting).

If the beam of lazar 1 is crossed before the beam of lazar 2, that is means someone entered the room. And if the beam of lazar 2 is crossed before the beam of lazar 1 that is means someone leaved the room.
3.2.2 Light Sensor

The main component in light sensor circuit is LDR. As discussed in Chapter 2, LDR interacts with light level. LDR resistance drops as the intensity of the light increases and vice versa. The most suitable circuit for light sensor is voltage divider circuit.

The selection of resistor across the LDR is based on Table 3.1 below. First, the resistor value was randomly selected. Then, a simple calculation of voltage divider was done using equation below. To make the calculation easier, $V_{in}$ was set 5V and LDR resistance was measured. Under sun light it equal to 60Ω and 1MΩ in the dark. The resistor that gives high range of output voltage it was selected.
\[ V_o = \frac{R_{LDR}}{R_{LDR} + R} \times V_{in} \]

<table>
<thead>
<tr>
<th>Fixed R (Ω)</th>
<th>( V_o ) in the light (V)</th>
<th>( V_o ) in the dark (V)</th>
<th>( V_o ) range (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.875</td>
<td>4.999</td>
<td>3.124</td>
</tr>
<tr>
<td>1k</td>
<td>0.283</td>
<td>4.995</td>
<td>4.712</td>
</tr>
<tr>
<td>10k</td>
<td>0.029</td>
<td>4.950</td>
<td>4.921</td>
</tr>
<tr>
<td>100k</td>
<td>0.0029</td>
<td>4.545</td>
<td>4.5421</td>
</tr>
<tr>
<td>1M</td>
<td>0.00029</td>
<td>2.5</td>
<td>2.499</td>
</tr>
</tbody>
</table>

Table 3.1 LDR output voltage range

10kΩ resistor was selected since it gives the highest range of LDR output voltage. The output of voltage of LDR will be fed into analoge input port (PORT A) of Atmega16, then the atmega16 will convert it to digital (by analoge to digital converter) to deal with it. This LDR will be placed on the floor, near the window of the room model.

The range (4.921) of the LDR voltage is divided into five divisions. The microcontroller create certain PWM signal for each division it received from the LDR and fed it to LED to turn on by certain brightness.

The microcontroller receives the voltage of LDR as analoge, it convert it to digital (analoge to digital converter) by this equation \((V \times 1023/5)\), then it put the digital number in binary form in 10 bits. But 8bits A/D converter was used, for that the two least significant bits was ignored and the eight most significant bits of the 10 bits was taken and form the digital number in 8 bits.
### Table 3.2 conversion of analge volts to digital volts

<table>
<thead>
<tr>
<th>Analog volts</th>
<th>Digital volts in binary form</th>
<th>Digital volts in hexadecimal form</th>
<th>Digital volts in decimal form</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.029</td>
<td>00000001</td>
<td>0x01</td>
<td>1</td>
</tr>
<tr>
<td>1.0132</td>
<td>00110100</td>
<td>0x34</td>
<td>52</td>
</tr>
<tr>
<td>1.9974</td>
<td>01100110</td>
<td>0x66</td>
<td>102</td>
</tr>
<tr>
<td>2.9816</td>
<td>10011000</td>
<td>0x98</td>
<td>152</td>
</tr>
<tr>
<td>3.9658</td>
<td>11001011</td>
<td>0xCB</td>
<td>203</td>
</tr>
<tr>
<td>4.95</td>
<td>11111101</td>
<td>0xFD</td>
<td>253</td>
</tr>
</tbody>
</table>

### Table 3.3: the OCRs corresponding to range of volts

<table>
<thead>
<tr>
<th>OCR0</th>
<th>Range of Digital volts in decimal form</th>
</tr>
</thead>
<tbody>
<tr>
<td>0X00</td>
<td>V&lt;1</td>
</tr>
<tr>
<td>0X33</td>
<td>1&lt;V&lt;52</td>
</tr>
<tr>
<td>0X66</td>
<td>52&lt;V&lt;102</td>
</tr>
<tr>
<td>0X99</td>
<td>102&lt;V&lt;152</td>
</tr>
<tr>
<td>0XCC</td>
<td>152&lt;V&lt;203</td>
</tr>
<tr>
<td>0XFF</td>
<td>V&gt;203</td>
</tr>
</tbody>
</table>

Table 3.2 conversion of analge volts to digital volts

Table 3.3: the OCRs corresponding to range of volts
3.2.3 Light Output

In this project, LED was used as source of light for the room model. The LED will be mounted on ceiling of the model room. The LED brightness will be determined by the PWM created by ATmega 16. It is fully software which means no external circuit needed to create and control the PWM. Thus, the system has less circuit.

To control the room brightness, the LED was connected to PB3 pin of ATmega 16. Pin PB3 is the output of the PWM created in the programming. By Adjusting the duty cycle in the PWM, we can control the brightness of the LED. The maximum duty cycle is 100% and the LED were at its maximum brightness. Thus, at higher duty cycle, the LEDs will become brighter and vice versa.

![Figure 3.5: duty cycle of PWM signal](image)

3.2.4 Motorized Window Blind

There are many types of window blinds. Rollover type is suitable with DC motor. The blind will be controlled by a DC motor. The motor will control the blind either to open or close it depends on the mode was selected.

The speed of dc motor is high for open and closes the blind, so the speed of the DC motor was controlled (reduced) by PWM signal generated by AT mega 16.

Direction of the motor will be determined by the motor driver, L293D chip. The circuit for controlling the window blind is as Figure 3.14 below.

The PWM signal was generated from pin PD5 (OC1A) of the microcontroller and fed to the enable (EN2) of the L293D chip to control the speed of motor.
Pin PD5 (OC1A) is the output of the PWM signal created in the programming. By adjusting the duty cycle in the PWM, we can control the speed of the motor. The maximum duty cycle is 100% and the motor was at its maximum speed. Thus, at higher duty cycle, the motor will rotate with more speed and vice versa.

Duty cycle that gives suitable speed of motor which smoothly opens and closes the blind was selected.

![Figure 3.6: connections of L293D](image)

To control the motor, inputs given to L293D are as in Table 3.2 below. Since only one motor was used, only one side of L293D needed. Enable 2, input 3 and input 4 pins was selected. PD5 (OC1A) of the microcontroller was fed to the enable (EN2) of the L293D chip. And the pins PC0, PC1 were fed to the input3, input4 of the L293D chip in sequence.

![Figure 3.7: circuit of L293d in proteus](image)
3.3 Software Development

3.3.1 Programming ATmega16C

Code Vision AVR is a complete set of tools designed for rapid and efficient software development for the Atmel AVR microcontrollers. Code Vision AVR is the only IDE on the market that features an Automatic Program Generator (Code Wizard AVR) for the new ATxmega devices and the only C compiler that supports the reduced core (ATtiny4, ATtiny5, ATtiny9, ATtiny10, ATtiny20, ATtiny40) chips.

ATMEGA 16 microcontroller was programmed using (Code Vision AVR Evaluation V2.04.9a) version. ATMEGA 16 and C language fit together very well. Thus, it was used in writing the whole program for the system. The complete program for the system can be referred in Appendix A.

After the complete program were done written in that application, the program was compiled and burned into the Atmega16 using parallel AVR Programmer.

The parallel AVR programmer was connected to the atmega16 as shown in data sheet below.

<table>
<thead>
<tr>
<th>Enable 2</th>
<th>Input 3</th>
<th>Input 4</th>
<th>Output 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>Motor stop</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Motor stop</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Blind open</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Blind close</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Motor stop</td>
</tr>
</tbody>
</table>

Table 3.4: DC motor configuration
3.3.2 Automatic operation

The automatic operation of the system is based on flow chart below. Initially the room is empty when someone enters the room; the lighting will be turned on automatically. Then the light sensor will measure the intensity of brightness in the room and the lighting will be (more brightness or less brightness) based on the read of light sensor. And this is done by the pwm signal as discussed earlier.
Figure 3.10: Automatic operation flow chart

There are four modes were created in system. It is combination of (blind open/closed) and (LED on/off). The purpose of these modes to make the system more flexible and the user will have more options in the system.

After the user has entered the room and automatic operation was activated, the user can choose each one of four modes by pressing four push buttons.

The following table shows the functions of each mode.
<table>
<thead>
<tr>
<th>Modes</th>
<th>lighting</th>
<th>blind</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode1</td>
<td>On</td>
<td>Opened</td>
</tr>
<tr>
<td>Mode2</td>
<td>On</td>
<td>Closed</td>
</tr>
<tr>
<td>Mode3</td>
<td>Off</td>
<td>Opened</td>
</tr>
<tr>
<td>Mode4</td>
<td>Off</td>
<td>Closed</td>
</tr>
</tbody>
</table>

Table 3.5: functions of modes

Notices:

- When the lighting is on that is means it was turned on in automatic operation (fed by PWM signal).
- If the window blind has already been opened, it cannot be opened again. That is means that if mode3 was selected after mode1 the window blind will not move, because it has already been opened. And so on. This combination of selecting mode after another mode is taken in consideration in the code which will be attached to appendix.
- Each of sub circuits was connected separately of the others, to easily test, detect and find troubleshoots and problems in connections.
- Then all sub circuits were combined together to form the completed circuit. The complete circuit was tested.
Figure 3.11: Complete circuit

Figure 3.12: Complete circuit in proteus
The PCB layout of the completed circuit was designed as shown in the figure below.

Figure 3.13: PCB layout

After the PCB was designed it was implemented and designed.

Figure 3.14: PCB board
Figure 3.15: PCB board from other side before fixing the components

Figure 3.16: Final PCB board after fixing components
CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Results

The system was divided into parts, each part was implemented separately.

4.1.1 The first part (door sensor)

The objective of this part is to count the number of people entered and left the room and then the microcontroller determines the occupancy of the room and uses it in automatic operation.

The hardware implementation of door sensor was shown in chapter three and software will be shown in the appendix (code).

The objective of this part was met successfully.

4.1.2 The second part (light sensor and light output)

The objective of this part is to measure the intensity of light in the room then microcontroller (depends on sensor read) change the lighting intensity of the lighting source (LED).

The hardware implementation of light sensor was shown in chapter three and software will be shown in the appendix (code).

The objective of this part was met successfully.

4.1.3 The third part (the motor /blind)

The objective of this part is to control the motion of window blind either to open or close it.

The objective of this part was met successfully.

The three parts were integrated to form complete circuit which met the requirements successfully.

Moreover, the system was enabled to operate in certain modes.

The objective of modes is to make the system more flexible and more optional to user.
Each mode objective was met as shown in the figures below.

Figure 4.1: The system is idle (there is no person inside the room)

Figure 4.2: Mode 1(LED is turned on, window blind is opened)

Figure 4.3: Mode 2(LED is turned on, window blind is closed)
Figure 4.4: Mode 3(LED is turned off, window blind is opened)

Figure 4.5: Mode 4(LED is turned off, window blind is closed)

The images cannot clarify the operations of system in all stages obviously. For this CD contains video will be attached with the thesis of the project.

4.2 Discussions

The system was verified and fully tested. Combination of hardware and software for the system works as expected. All the objectives for this project were met. Although, there are a few problems and difficulties encountered during development of the project, the system works as expected. Smart lighting control works as desired.

An attempt to replace the LED with lamp was done, because of obviousness of variation in lamp’s brightness by using transistor which will be switched on by PWM signal fed to it from ATmega16.

This idea to turn on the lamp instead of LED was not completed, because the submission dead time of project was come and time is over.
CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

As the conclusion, all objectives of this project were successfully met. The system was able to control light brightness according to its surrounding. The LEDs were able to dim as necessary. Lights were automatically on when the room is occupied and automatically off when the room is empty. The system operates in certain modes, can be selected by users.

The software part was the core or the heart of the system. Without the software part, the ATMEGA will not be able to interface and interact with all components. Also, the automatic operation will not be successfully implemented if there is no software part. These project concepts can be useful for taking a step to a more modern lifestyle. This project can be enhanced more to meet the users and market needs.

5.2 Recommendations

Several suggestions are recommended for future research works to further enhancing the project. There are consists of:

1. Implement the system on real environment instead of using a room model.
2. Use 7-segments display to show the current setting of the system.
3. Display the counted number of people (on computer’s screen or LCD), can be used in the evacuation in fire situation or in hotels can skip occupied for room services and every application that the number of people is useful.
4. Add more functions to the system such as controlling other electrical devices that is usually found in a room such as air conditioner or fan.
5. If we choose computer’s screen to display the number of people on it, We will Use wireless network connection in interfacing the ATMEGA 16 with the computer system instead of ordinary RS-232 cable.
REFERENCES


4- Wikipedia, the free encyclopedia (www.wikipedia.org).
Appendix A

Microcontroller Program Source Code

/********************************************
This program was produced by the
CodeWizardAVR V2.04.9a Evaluation
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date   : 6/2/2013
Author  : Freeware, for evaluation and non-commercial use only
Company :
Comments:

Chip type      : ATmega16
Program type   : Application
AVR Core Clock frequency: 4.000000 MHz
Memory model   : Small
External RAM size : 0
Data Stack size : 256
********************************************/
#include <mega16.h>

#include <delay.h>

#define ADC_VREF_TYPE 0x60

// Read the 8 most significant bits
// of the AD conversion result
unsigned char 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 ADCH;
}

// Declare your global variables here
int n,x,y,z,a=0,b=0,counter=0,c,d,e,f;

void ldr(void)
{
  z= read_adc(2);
  if (z < 1)     {OCR0=0;};
if (z > 1 && z < 52) {OCR0=51.2;};
if (z > 52 && z < 102) {OCR0=102.4;};
if (z > 102 && z < 152) {OCR0=153.6;};
if (z > 152 && z < 203) {OCR0=204.8;};
if (z > 203) {OCR0=255;};
}

void motor_blind_rise (void)
{
OCR1A=100;
PORTC.0=1;
PORTC.1=0;
delay_ms(500);
OCR1A=0;
}

void motor_blind_fall (void)
{
OCR1A=100;
PORTC.0=0;
PORTC.1=1;
delay_ms(500);
OCR1A=0;
}
void mode1(void)
{

    if (n!=1 && n!=3)
    {
        motor_blind_rise ();
    }

    ldr();
    n=1;
}

void mode2(void)
{
    if (n!=2 && n!=4)
    {
        motor_blind_fall ();
    }

    ldr();
    n=2;
}

void mode3(void)
{


if (n!=1 && n!=3)
{
    motor_blind_rise();
}
OCR0=0;
n=3;

}

void mode4()
{

if(n!=2 && n!=4)  
{
    motor_blind_fall();
}
OCR0=0;
n=4;

}

void main(void)
{

    // Declare your local variables here

    // Input/Output Ports initialization


// Port A initialization
// Func7=In Func6=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTB=0x00;
DDRB=0xFF;

// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
DDRD=0xFF;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 500.000 kHz
// Mode: Phase correct PWM top=0xFF
// OC0 output: Non-Inverted PWM
TCCR0=0x62;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 500.000 kHz
// Mode: Fast PWM top=0x00FF
// OC1A output: Non-Inv.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x81;
TCCR1B=0x0A;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

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

// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: AVCC pin

// ADC Auto Trigger Source: Free Running

// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0xA2;
SFIOR&=0x1F;

while (1)
{

  // Place your code here

  if(counter==0)
  {
    n=0;
    c=0;
    d=0;
    e=0;
    f=0;
    PORTC.6=1;
    PORTC.7=0;
    OCR0=0;
    OCR1A=0;
    PORTC.0=0;
    PORTC.1=0;
  }
}
x = read_adc(0);
y = read_adc(1);
z = read_adc(2);

if (x >= 102)
    {a=1;}

if (y >= 102)
    {b=1;}

if (a == 1 && y >= 102)
    {
    counter = counter + 1;
    a = 0;
    b = 0;
    delay_ms(500);
    }

if (b == 1 && x >= 102)
    {
    counter = counter - 1;
a=0;
b=0;
delay_ms(500);
}

if (counter>0)
{

PORTC.6=0;
PORTC.7=1;
if (n!=1&&n!=2&&n!=3&&n!=4)
{
ldr();
OCR1A=0;
}

if(PINA.3==1)
{
c=1;
d=e=f=0;
}
if (c==1)
{
mode1();
}

if(PINA.4==1)
{
d=1;
c=e=f=0;
}
if(d==1)
    {mode2();}

if(PINA.5==1)
    { e=1;
      c=d=f=0;
    }
if(e==1)
    {mode3();}

if(PINA.6==1)
    { f=1;
      c=d=e=0;
    }
if (f==1)
    {mode4();}