AUTONOMOUS LIGHTING AND SECURITY SYSTEMS

ENGKU MOHD AMRI BIN ENGKU ARIFF

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor Degree of Electrical Engineering (Electronics)

Faculty of Electrical & Electronics Engineering
University Malaysia Pahang

NOVEMBER, 2007

"All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author."

Signature : ______________________________

Author : ENGKU MOHD AMRI BIN ENGKU ARIFF

Date : 30 NOVEMBER 2007

ACKNOWLEDGMENT

Alhamdulillah, thank to God throughout all His Almighty kindness for letting me to finish this project.

First of all, I wish to express my sincere appreciation to my supervisor, Encik Bakri Bin Hassan for his valuable suggestions, guidance and consistent supports throughout finishing this project. Without his kindness and moral support this thesis would not have been the same as presented here.

Special thanks to all lecturers who directly contribute with my project. I also please my gratitude to technicians' staff that is willingly helping me for this project. And also a special thanks to my academic advisor, Encik Fadhil BinAbbas for his valuable support and guidance from the start of the project till the end.

I'm also indebted to all my colleagues for their supporting and ideas. Their views and tips are useful indeed. Last but not least, I am grateful to my mother for her support, understanding and encouragement.

ABSTRACT

There are no good scientific research studies that convincingly show the relationship between lighting and crime. However, lighting does deter crime and makes people feel more secure. This project is aimed to design a dual-functional system that acts as a low-cost home security system and garden lights controller. The main control system utilizes Motorola MC6811HCA1 as it offers good energy management and better efficiency. The system is equipped with light sensor to measure darkness level, infrared sensors to detect movement within the vicinity, a digital clock to manually control the main system and an interface circuit that enable the system to control the main power. The system program will be simulated using THRsim11 before being implemented into the hardware and tested for functionality. It is found that the system can effectively function as a home security system as well as a garden lighting controller.

.

ABSTRAK

Tiada kajian saintifik yang menunjukkan terdapat hubungan antara pengcahayaan dan jenayah. Namun pencahayaan yang baik menghindarkan jenayah dari berlaku dan juga sebab untuk penghuni berasa lebih selamat. Projek ini adalah untuk mereka bentuk suis lampu automatic yang akan menaktifkan bekalan kuasa utama apabila hari gelap. Fokus projek ini adalah untuk mereka bentuk satu sistem dua fungsi sebagai sistem keselamatan rumah yang berkos rendah dan juga sebagai pengawal lampu taman. Pengawal utama sistem ini menggunakan Motorola MC6811HCA1 kerana ia memberikan pengawalan tenaga yang baik dan efisyen. Sistem ini dilengkapi dengan sensor cahaya untuk mengukur tahap kegelapan, sensor infrared untuk mengesan pegerakan dalam kawasan taman, jam digital untuk mengawal sistem utama secara manual dan litar pengantaraan yang membolehkan sistem mengawal punca kuasa. Program bagi sistem ini akan disimulasikan dengan menggunakan THRsim11 sebelum diimplimentasi kedalam litar dan diuji fungsinya. Sebagai kesimpulannya, sistem in boleh berfungsi dengan baik sebagai sistem keselamatan rumah dan juga sebagai pengawal lampu taman.

TABLE OF CONTENTS

CHAPTER	TITLE		PAGE
	TITLE PAGE		i
	DECLARATION		ii
	ACKNOWLEDGEMENT		iii
	ABSTRACT		iv
	ABSTRAK		v
	TABLE OF CONTENT		vi
	LIST OF FIGURE		ix
	LIST OF TABLES		xi
	LIST OF APPENDICES		xii
1	**INTRODUCTION**		
	1.1.	Overview	1
	1.2.	Objectives	2
	1.3.	Thesis Structure	3
2	**LITERATURE REVIEW**		
	2.1	Introduction	4
	2.2	Light Dependent Resistor	5
	2.3	Infra-Red Sensor	7
	2.4	Microcontroller	7
	2.5	Wp11	8

	2.6	Digital Clock	8
	2.7	Opto-Coupler	9
	2.8	Summary	11
3	**METHODOLOGY**		
	3.1	Introduction	12
	3.2	System Block Diagram	13
		3.2.1 Main Power Supply	13
		3.2.2 Module System	14
		3.2.2.1 Microcontroller	14
		3.2.2.2 Digital Clock	14
		3.2.2.3 Interface Circuit	15
		3.2.3 Sensor	15
	3.3	Hardware Design	15
		3.3.1 Microcontroller	16
		3.3.2 Light Sensor	17
		3.3.3 Infra-Red Sensor	17
		3.3.4 Digital Clock	20
		3.3.5 Interface Circuit	21
	3.4	Interfacing Hardware To Microcontroller	22
		3.4.1 Interfacing Light Sensor	23
		3.4.2 Interfacing IR Sensor	23
		3.4.3 Interfacing Digital Clock	24
	3.5	Program Development	26
		3.5.1 Programmed Development For Light Sensor	27
		3.5.2 Programmed Development For IR Sensor	28
		3.5.3 Programmed Development For Light Sensor, IR Sensor & Digital Clock	30
	3.6	System Prototype	31
		3.6.1 Module System	33

3.7 Summary 34

4 Result And Analysis

4.1 Introduction 35
4.2 Test Set-up 35
4.2.1 Set-up For Microcontroller 36
4.2.2 Set-up For IR Sensor 36
4.2.3 Set-up For Light Sensor 36
4.3 Testing The Microcontroller 36
4.3.1 Power Supply & Reset Circuit 37
4.3.2 Phase 1 38
4.3.3 Phase 2 40
4.3.4 Phase 3 42
4.4 Testing The Range Of IR Sensor 42
4.5 Testing The Light Sensor 43
4.6 Summary 44

5 SUMMARY AND CONCLUSION

5.1 Summary 45
5.2 Conclusion 45
5.3 Limitation In This Project 46
5.4 Suggestion For Improvement 47
5.4.1 Cost & Commercialization 47

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
2.1	A common light dependent resistor	6
2.2	Light dependent resistor Symbol	6
2.3	Opto-Coupler	10
3.1	Block diagram	13
3.2	Microcontroller Circuit	16
3.3	Light Sensor	17
3.4	Infra-red transmitter	18
3.5	The relay condition	19
3.6	Infra-red receiver	19
3.7	Digital clock	20
3.8	Interface circuit	21
3.9	Interfacing light sensor and microcontroller	23
3.10	The Interface Circuit Between IR Sensor To The Microcontroller	24
3.11	Relay Interface The Digital Clock And the Microcontroller	26
3.12	Microcontroller & MAX233	27
3.13	Programmed Flowchart For The Light Sensor	28
3.14	Flowchart For The IR Sensor	29
3.15	Flowchart Programmed For The System	30
3.16	Prototype Module	32
3.17	Power Source For The Module	32
3.18	The Module System Using PVC Box	33
4.1	Voltage regulator	37

4.2	Reset circuit	37
4.3	Phase 1 Circuit	38
4.4	The E clock output	40
4.5	Microcontroller circuit Phase 2	42

LIST OF TABLE

TABLE	TITLE	PAGE
4.1	Result from the oscilloscope	40
4.2	Result from the test programmed	41
4.3	Result for the switching programmed	42
4.4	Test range for the IR sensor	43
4.5	The result of testing the light sensor	44
5.1	Project cost	48

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	List Of Components	50
B	Digital Clock	53
C	Interface Circuit Another Type	54
D	Interface Between Light Sensor To The Microcontroller	55
E	Interfacing The IR Sensor To The Microcontroller	56
F	Digital Clock And The Microcontroller	57
G	Microcontroller & MAX233	58
H	Wp11 Manual	59
I	Programmed For Light Sensor	76
J	Programmed For IR Sensor	77
K	Final Programmed For The System	78
L	Test Programmed	81
M	Test Programmed For Switching	85
N	References	86

CHAPTER I

INTRODUCTION

1.1 OVERVIEW

Security lights are one of the most practical and effective way to prevent crimes around the house. Even though a well-lighted property neither can prevent malicious activity nor guarantee personal safety, however it is a fact that crimes are less likely to occur because lighting is an excellent deterrent. Security lights expose the criminal thus, making them feel insecure and vulnerable. Even if the homeowners did not realize of an unwanted visitor, the presence of lights might alert neighbors, watch guard or police patrols of the situation.

Nowadays, the light-based security system can be buy from store have some weaknesses. One of the weakness is the bulb used is of a very high intensity light bulb. The problem is the light can produce a very large shadow. Especially, if the garden is full of trees and other objects. The large shadow can limit the vision within the garden. Moreover, the security lights need to be controlled manually.

Therefore, in this project, instead of using high intensity lights bulb, garden lights bulb will be used. Garden lights bulb is chosen because it gives better vision around the vicinity. The system is also design to be on and off automatically. To control the system, microcontroller MC68HC11A1 is use so that the system will be able to run automatically.

Other hardware that will be used is the light sensor, infra-red sensor, digital clock and also interfaces circuit.

The sensors that are use give the system the ability to detect the condition around the vicinity. Light sensors enable the system to differentiate between day and night. By the end of the day garden lights will on. The digital clock are used to replace timer. Its function is to enable the system to know the time. In this project, the garden lighting will activate by dusk. The activation of the garden lights will continue until the time set in the digital clock alarm system. When the digital clock alarm is on, it will deactivate the whole garden lights. However, after this deactivation, the infra-red sensor will take over. Infra-red sensor, are used to acts as a motion sensor. If the sensor detects any movement the garden lighting will be activated to scare off the intruder. Therefore, the system does act in a dual-function system, both as the security system and also as the garden lighting system.

1.2 OBJECTIVE

The overall aim of this project is to build an autonomous lighting and security system that saves energy and more efficient, the design system will have the dual-functionality on both the security and the garden lighting system in the vicinity. Moreover, the job scope is not limited to just turning 'on' and 'off' the main power supply but it extends to these works:

1. To build a light sensor circuit that is able to activate at a preset darkness. The light sensor is able to adjust its level of sensitivity.
2. To build an infra-red sensor circuit that act as a motion sensor.
3. Both circuits of the light sensors and infra-red sensors will be built independently before interfacing them with the microcontroller.

4. To build digital with an alarm function and able to energize a relay to the interfacing circuit. The clock use an AC power supply.
5. To build an interface circuit that is able to receive input from the microcontroller and able to both activate and de-activate the main power supply.
6. To build a microcontroller MC68HC11A1 circuit as the main controller of the system. The microcontroller is in the bootstrap mode.

1.3 THESIS STRUCTURE

The thesis consist of five chapters all together including this chapter. The contents of each chapter are outlined as follows:

Chapter 1 briefly explains about the overview of the project, its objective and also the scopes. Finally, the thesis structure is explained.

Chapter 2 discusses the detail of literature review, that are applied in the whole project. These literature review, are selected from books, journals and articles.

Chapter 3 explains the methodology and the system hardware. This chapter is divided into two parts. The first part explains the building the hardware and the second part interfacing the hardware with the microcontroller. This part also explain how the microcontroller is programmed in bootsrap mode.

Chapter 4 discuss the testing set-up and the result of interfacing the hardware with the microcontroller. The chapter focus, on the hardware testing.

Chapter 5 is the conclusion in finishing this project. This chapter also include suggestion for future development. Cost and commercialization also discuss in this chapter.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter is discussed about the literature review that related to the project. It is important to know further detail of the component that need to be used in this project. The focus of this literature review will be the microcontroller, light sensor, infra-red sensor and opto-coupler. These circuits must be well known and ensure that these circuit can interface with the microcontroller.

There are still other supported circuit such as Max233 and DB9. Which will be use to write program to the microcontroller. Max233 can only be used for bootstrap mode only. For expanded mode, the microcontroller require special burner to insert the program.

Other supported circuit that will be used are clock circuit and reset circuit. These support circuit are used to enable the microcontroller to operate. Both of the circuits are a must to a microcontroller even for an expended mode. There is another circuit that support the microcontroller which is the voltage regulator. It is use to supply 5Vdc to the microcontroller. However, once the project is in the final stage, the voltage regulator will not be used.

For further information on the components and concept that will be used is explain in this chapter of literature review. All the fact and information of designing the system comes mostly from books, and article. Which will be explain later on this chapter.

2.2 LIGHT DEPENDENT RESISTOR

A photoresistor or LDR is an electronic component whose resistance decreases with increasing incident light intensity. It can also be referred to as a light-dependent resistor (LDR), photoconductor, or photocell. A photoresistor is made of a high-resistance semiconductor. If light falling on the device is of high enough frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, thereby lowering resistance.

A photoelectric device can be either intrinsic or extrinsic. An intrinsic semiconductor has its own charge carriers and is not an efficient semiconductor, eg. silicon. In intrinsic devices, the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electron across the entire bandgap. Extrinsic devices have impurities added, which have a ground state energy closer to the conduction band since the electrons don't have as far to jump, lower energy photons (i.e. longer wavelengths and lower frequencies) are sufficient to trigger the device. If a sample of silicon has some of its atoms replaced by phosphorus atoms(impurities), there will be extra electrons available for conduction. This is an example of an extrinsic semiconductor.

Cadmium sulphide (CdS) cells rely on the material's ability to vary its resistance according to the amount of light striking the cell. The more light that strikes the cell, the lower the resistance. Although not accurate, even a simple CdS cell can have a wide range of resistance from less than 100 Ω in bright light to in excess of 10 MΩ in darkness. Many commercially available CdS cells have a peak sensitivity in the region of 500nm - 600nm. The cells are also capable of reacting to a broad range of frequencies, including infrared (IR), visible light, and ultraviolet (UV). They are often found on street lights as automatic on/off switches. They were once even used in heat-seeking missiles to sense for targets.

Photoresistors come in many different types. Inexpensive cadmium sulphide cells which shown in figure 2.1, can be found in many consumer items such as camera light meters, clock radios, security alarms, street lights and outdoor clocks. They are also used in some dynamic compressors together with a small incandescent lamp or light emitting diode to control gain reduction.

Lead sulphide- and indium antimonide-LDR are used for the mid infrared spectral region. At the other end of the scale, Ge:Cu photoconductors are among the best far-infrared detectors available, and are used for infrared astronomy and infrared spectroscopy. However, all light dependent resistor use the same symbol which is shown in figure 2.2.

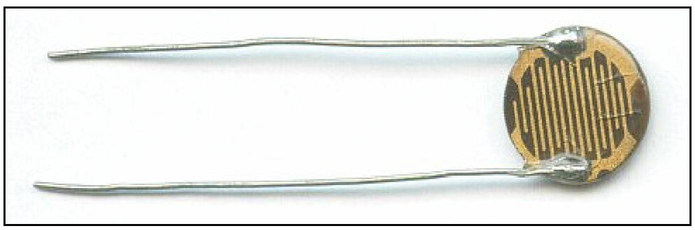


Figure 2.1: A common Light Dependent Resistor

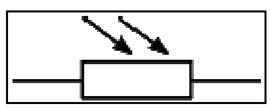


Figure 2.2: Light Dependent Resistor symbol.

2.3 INFRA-RED SENSOR

Is a device that is good at detecting object or movement within its range at a relatively low cost and low power consumption. Very wide angle sensing. The infrared sensor will be use in detecting objects/movement moving in the garden area. The range of the sensor is about 1.5 meter. It usually use as part of a robotic system. It is able to detect any object or movement within it range however it cannot predict the distance between the sensor and object. However the sensor weakness is not an issue. It is because we require it ability in detecting movement.

2.4 MICROCONTROLLER

The microcontroller board is the brain that control the whole system. Microcontroller MC68HC11 is chosen because it fulfill all the requirement that needs to run the system. MC68HC11 is chosen rather that PIC type because it has enough input to fulfill the project objective. The micro controller offers high speeds and low power consumption chips with multiplexed buses and a fully static design.

The MC68HC11A8 has up to 38 input/output lines, depending on the operating mode. Port A has three input-only pins, four output-only pins, and one bidirectional I/O pin. Port A shares functions with the timer system. Port B is an 8-bit output-only port in single-chip modes and is the high-order address in expanded modes. Port C is an 8-bit bidirectional port in single-chip modes and the multiplexed address and data bus in expanded modes. Port D is a 6-bit bidirectional port that shares functions with the serial systems. Port E is an 8-bit input-only port that shares functions with the A/D system.

For ensure the microcontroller is running normally, we must run the microcontroller through 3 phase. These phases show that the microcontroller is in good condition. For this project, the microcontroller is running in bootstrap mode. For bootstrap mode we can use Motorola programmable program. There are many such program and can be downloaded from the internet. For example Wp11, HC load and Jbug11. So therefore most of the time that applied for this project can be done in the room. The programming for the microcontroller can be done by using a PC that has DB9 port.

2.5 WP11

This manual provides information on our WP11.EXE program, Windows software used for programming 68HC11 family microcontrollers. WP11.EXE takes input from object code files created with 68HC11 assemblers or compilers. These object code files may be in Motorola S-record, Intel Hex or binary memory image formats. The software also allows the operator to perform many other useful programming functions such as erasing, blank checking & verifying devices, displaying, editing, exporting and changing the format of the object code files and filling unused memory locations with a user specified "fill" byte. The software was designed to work with our P11 Programming Board but will also work with any hardware that supports the 68HC11 special bootstrap mode of operation. *TECI* provides several other development tools for 68HC11 family microcontrollers such as our Cross Assemblers WASM11.EXE and TASM11.EXE and real time in-circuit emulator TECICEHC11.

2.6 DIGITAL CLOCK

A digital clock is a type of clock that uses digital electronic methods of keeping time. Digital clocks typically use the 50 or 60 hertz oscillation of AC power or a crystal oscillator as in a quartz movement to keep time. A digital clock typically displays a numerical hour range of 0–23, or 1–12 (with an indication of AM or PM), although digital versions of analog-style faces exist.

To represent the time, most digital clocks use a seven-segment LED, VFD, or LCD display for each of four digits. They generally also include other elements to indicate whether the time is AM or PM, whether or not an alarm is set, and so on. One drawback to digital clocks is the difficulty of setting the time in some designs. Most digital clocks flash 12:00 by default when first powered on and, since the clock is often not a critical function in many electronic devices, people often allow them to display this default.

Moreover, since they run on electricity and have no permanent memory, digital clocks must be reset every time they are moved or the power is cut off. This is a particular problem with alarm clocks, since a power outage during the night usually results in the clock failing to trigger the alarm in the morning. To reduce the problem, they often incorporate a battery backup to maintain the time during power outages. More recently, some devices incorporate a method for automatically setting the time, such as using a broadcast radio time signal from an atomic clock, getting the time from an existing satellite television or computer connection, or by being set at the factory and then maintaining the time from then on with a quartz movement powered by an internal rechargeable battery.

Digital clocks are very small, useful, and inexpensive. For these reasons, they are often incorporated into virtually every electronic device. Most commonly, digital clocks are incorporated into bedside alarm clock radios, but they are also found in televisions, microwave ovens, standard ovens, watches , computer and cell phones.
In the 1980s and 1990s, digital clocks were standard on most automobiles. However, the fashion is slowly leaning towards analog clocks, which was the first style of clock to be used in automobiles. Digital clocks were invented in 1956 and became more popular as microchips and LEDs became cheaply available.

2.7 OPTO-COUPLER

There are many situations where signals and data need to be transferred from one subsystem to another within a piece of electronics equipment, or from one piece of equipment to another, without making a direct .ohmic. electrical connection. Often this is because the source and destination are (or may be at times) at very different voltage levels, like a microprocessor which is operating from 5V DC but being used to control a triac which is switching 240V AC. In such situations the link between the two must be an isolated one, to protect the microprocessor from overvoltage damage.

Relays can of course provide this kind of isolation, but even small relays tend to be fairly bulky compared with ICs and many of today.s other miniature circuit components. Because, they are electro-mechanical, relays are also not as reliable . and only capable of relatively low speed operation. Where small size, higher speed and greater reliability are important, a much better alternative is to use an opto-coupler. These use a beam of light to transmit the signals or data across an electrical barrier, and achieve excellent isolation. Opto-couplers typically come in a small 6-pin or 8-pin IC package, but are essentially a combination of two distinct devices: an optical transmitter, typically a gallium arsenide LED (light-emitting diode) and an optical receiver such as a phototransistor or light-triggered diac. The two are separated by a transparent barrier which blocks any electrical current flow between the two, but does allow the passage of light. The basic idea is shown in Fig 2.3, along with the usual circuit symbol for an opto-coupler.

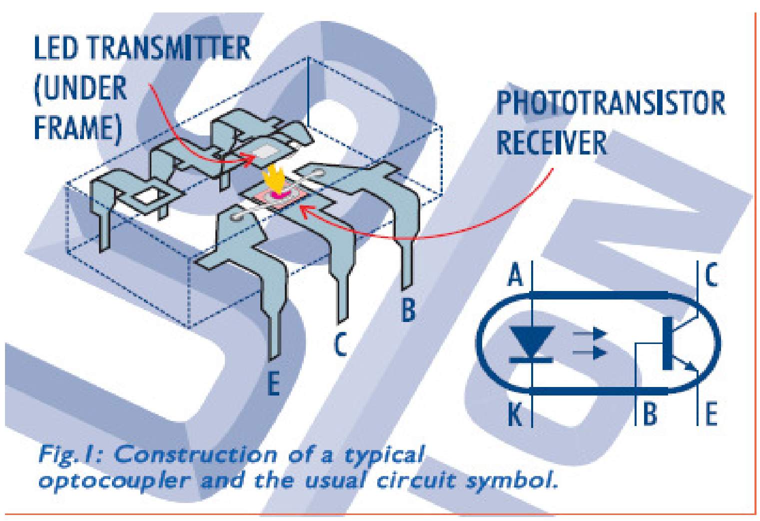


Figure 2.3 Opto-coupler

Usually the electrical connections to the LED section are brought out to the pins on one side of the package and those for the phototransistor or diac to the other side, to physically separate them as much as possible. This usually allows optocouplers to withstand voltages of anywhere between 500V and 7500V between input and output. Optocouplers are essentially digital or switching devices, so they.re best for transferring either on-off control signals or digital data. Analog signals can be transferred by means of frequency or pulse-width modulation.

2.8 SUMMARY

From all the fact that is collected, it can be summaries that interfacing the hardware to the microcontroller is possible. Using relay and transistor that will be use as the interfacing circuit. Controlling the main power using an IC like the microcontroller is not impossible with the use of an opto-coupler. The opto-coupler also protect the microcontroller from any damage.

The main reason microcontroller is use because the specs fulfill the requirement for this project. Microcontroller is able to control all the hardware that will be use in this project.

CHAPTER III

METHODOLOGY

3.1 INTRODUCTION

This chapter explains about the methodology used during the process of completing the project. The topic covered in this chapter is the block diagram, circuit design, programming the microcontroller, interfacing the circuit with microcontroller and programmed development.

The circuit is done step by step. Each of the hardware is design and built to run independently. So therefore the hardware will be tested for functionality before continuing to the next step. The next step after finishing the hardware is interfacing it with microcontroller. Each of the hardware are required another circuit before interfacing. Not all of the hardware are DC source. The digital clock and the main are AC source. Which mean it required another circuit to be interface with. The interface done with AC source will explain later on.

This chapter contents represent the whole project design. Start from basic step until the project is finish. Some of the circuit design are too large and will be represent in the appendix.

3.2 SYSTEM BLOCK DIAGRAM

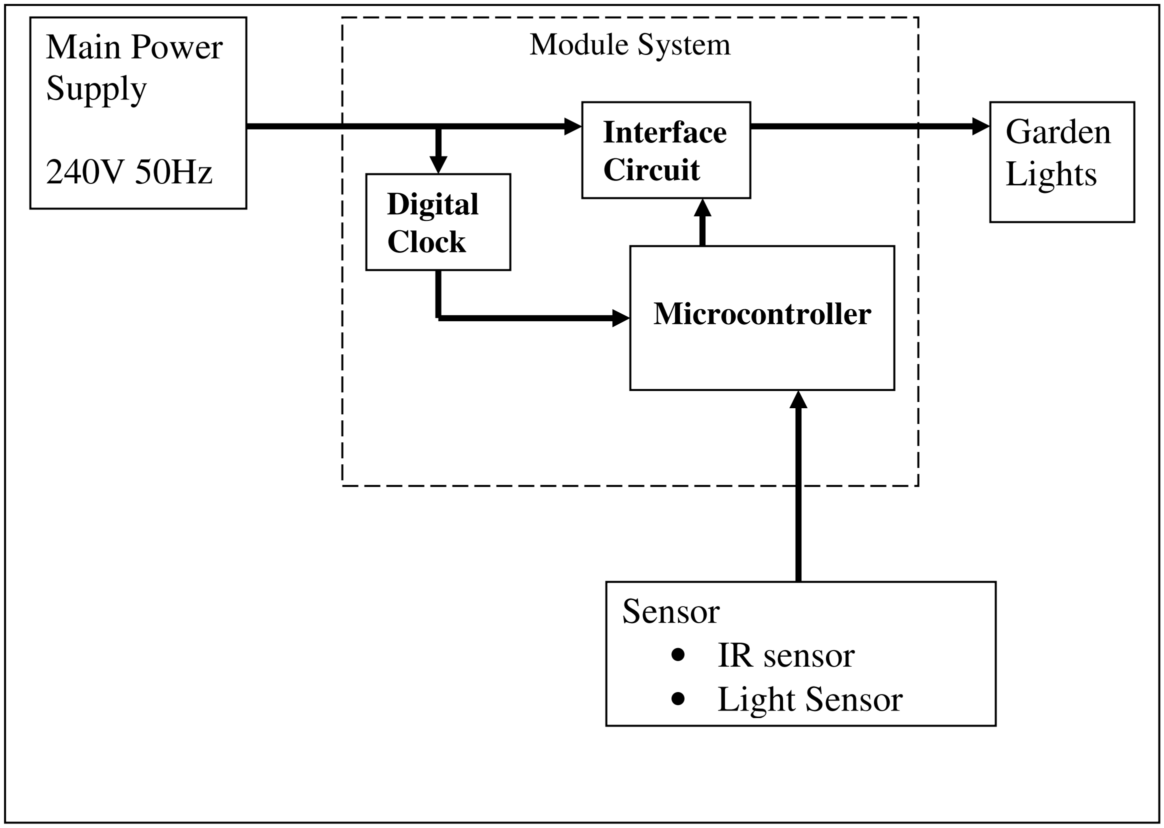


Figure 3.1: Block diagram

The design for the whole system are shown in a form of block diagram. Shown in figure 3.1 is how the system is function. The microcontroller is the controller of the system. Light sensor enable the microcontroller to differentiate day and night. Infra-red sensor enable the microcontroller to detect movement within the area of the garden. Timer enable the microcontroller to recognize the time and assign a certain function and command.

3.2.1 Main Power Supply

The supply of the main power can be obtained from any three pin socket which is supplied from TNB (Tenaga Nasional Berhad). Main power is supplied using a three pin plug. The source can be easily obtained whether at the hostel or at the lab. However for safety reason, in the early stage of the development, the testing is done at the lab.

3.2.2 Module System

Module system controlled the main power. The process of activate and deactivate are done in this part. Module system contain three main hardware, which is microcontroller, interface circuit and digital clock. Output of the module is a three pin socket. Therefore this module is compatible to any kind of lighting.

3.2.2.1 Microcontroller

Microcontroller is the brain of the system. All of the inputs and outputs are processed to the microcontroller. Inputs of the system are consists of sensors, digital clock and manual switch. Manual switch are fixed together with the microcontroller. Please refer to appendix F, which show a circuit of manual switch and microcontroller. Manual switch function are use to test run the microcontroller in the testing stage. Testing stage will be explained in chapter 4.

3.2.2.2 Digital Clock

Digital clock required an AC source, which is the only hardware that required this type of source. To simplify the system, digital clock are fix together inside the module. Clock function is to trigger the microcontroller when to deactivate the garden lights. Once garden light are deactivate, IR sensor will take over. Further detail will be explain later in this chapter.

3.2.2.3 Interface Circuit

Interface circuit function is to activate and deactivate the main power. The process of activating and deactivating are triggered by the microcontroller. Another function of the opto-coupler is to protect the microcontroller from over voltage. The over voltage is already explain in chapter 2.

3.2.3 Sensor

There are two types of sensors that are used in this project. Light sensor and infra-red sensor. Sensor output triggered the microcontroller when to activate and deactivate the main power. The process will later be explained in subchapter 3.5, which is programmed development.

3.3 HARDWARE DESIGN

The circuits are built step by step along with the programmed development. Each of the circuit are shown in the figure. However certain circuit which is consider to large will be place under appendix. The component use are clearly shown in the figure. For the list of component please refer appendix A

3.3.1 Microcontroller

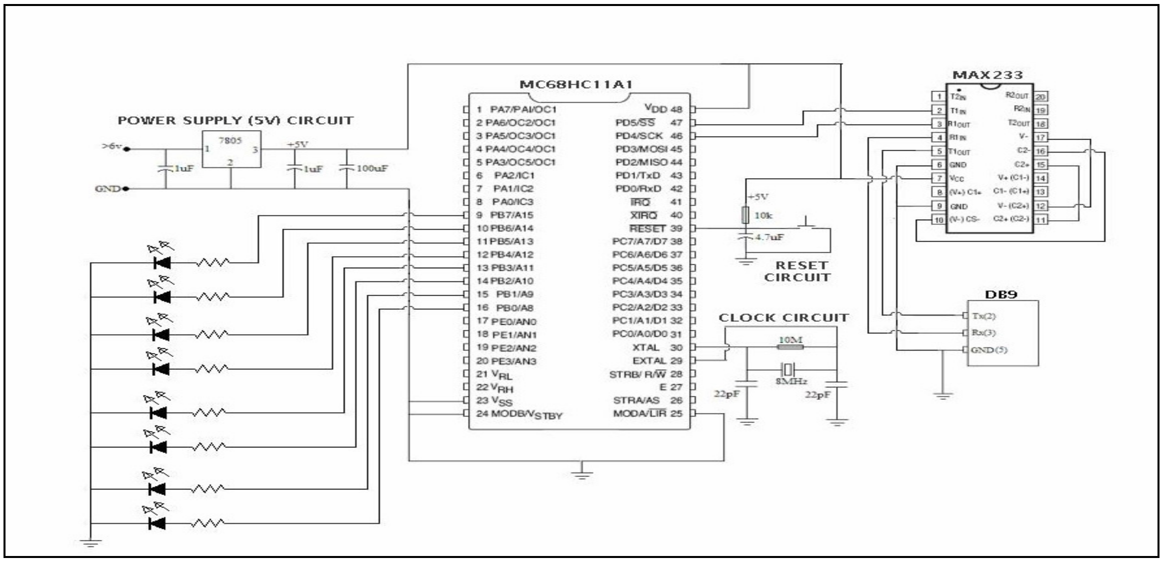


Figure 3.2 Microcontroller circuit

Microcontroller MC68HC11A1 is one of the 68 microcontroller family. The MC68HC11A1 is chose because it features fulfill the requirement that is required in this project. In this project MC68HC11A1 will be running in bootstrap mode. This mode will be using internal eeprom. If certain project required external eeprom, expanded mode will be use.

Figure 3.2 show the circuit for the microcontroller. The circuit microcontroller is divided by three parts. Which is voltage regulator, clock circuit and reset circuit. Voltage regulator supply 5V fix to the microcontroller. Since the microcontroller is a very sensitive circuit, it required a very low voltage. The clock circuit are use to supply pulse to the microcontroller. Without the clock pulse, the microcontroller will be unable to read and write data. So therefore further testing of the microcontroller will be later explain in chapter 4.

Once the microcontroller is done with testing for its functionality, than we shall proceed to the next hardware. The next hardware will be light sensor. Light sensor will give the system the ability to differentiate between day and night.

3.3.2 Light Sensor

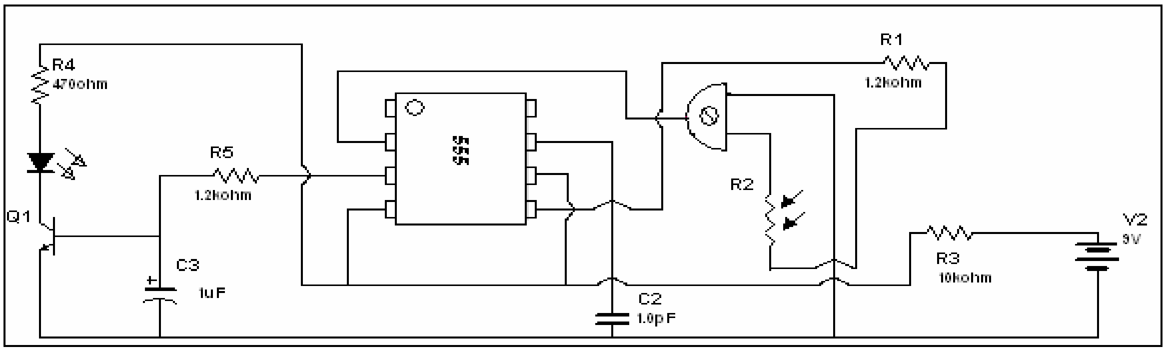


Figure 3.3 Light sensor

Light sensor is one of the most important part in this project. The sensor enable the system to differentiate between day and night. One of the system functions is to “on” the main power for garden lighting at night.

In figure 3.3 show the light sensor that is use in this project. If the sensor detects dark, the LED will on. If the circuit run as expected, than the circuit design will go to the next step. Which is circuit design for infra-red sensor. The Infra-red sensor will act as a motion sensor. With infra-red sensor, the system will not alone just for garden lighting but also for security reason.

3.3.3 Infra-red Sensor

Infra-red sensor is one of the many type of sensor that are use as a motion detector. For any IR sensor, it usually consists of receiver and transmitter. IR sensor usually use in a low cost security system is because the cost is very low. The IR sensor is consider the cheapest from the other sensor. However the IR length can be consider quite long. The range however depends on the quality of the hardware. In this project, the IR sensor is a self made circuit. So therefore the quality of the sensor is quite low, but the sensor is able to show an example of how a motion detector works.

The IR transmitter is shown in figure 3.4 in the next page. IR transmitter resemble of an LED. Shown in the circuit, the LED is in series with the IR transmitter. The LED function is to indicate the IR transmitter is transmitting. If the LED is 'on', the IR transmitter should be also transmitting. So therefore moving to the next step, began the design for IR receiver.

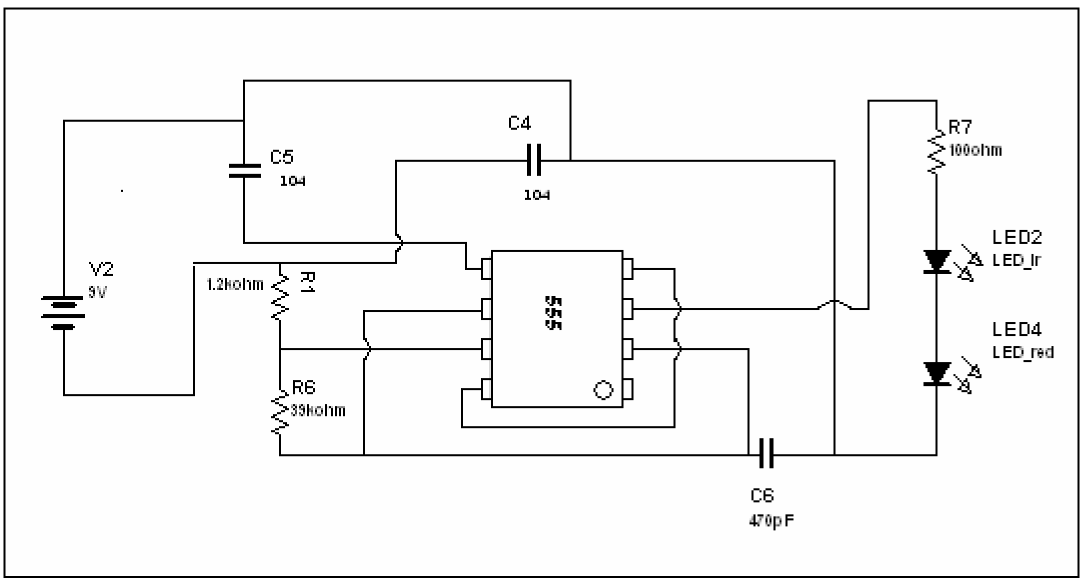


Figure 3.4 Infra red transmitter

Once the transmitter is finish, we began on the IR receiver. Both of these circuit are important. So therefore, both of the circuit are done at the same time. The circuit for the IR receiver are shown in figure 3.6 in the next page. IR receiver circuit are also use LED as an indicator. If the IR receiver received the IR transmitter, the LED will 'ON'. If the LED is 'ON', the relay also should switch to normally open condition. If the IR detects movement, the switch will change to normally close condition.

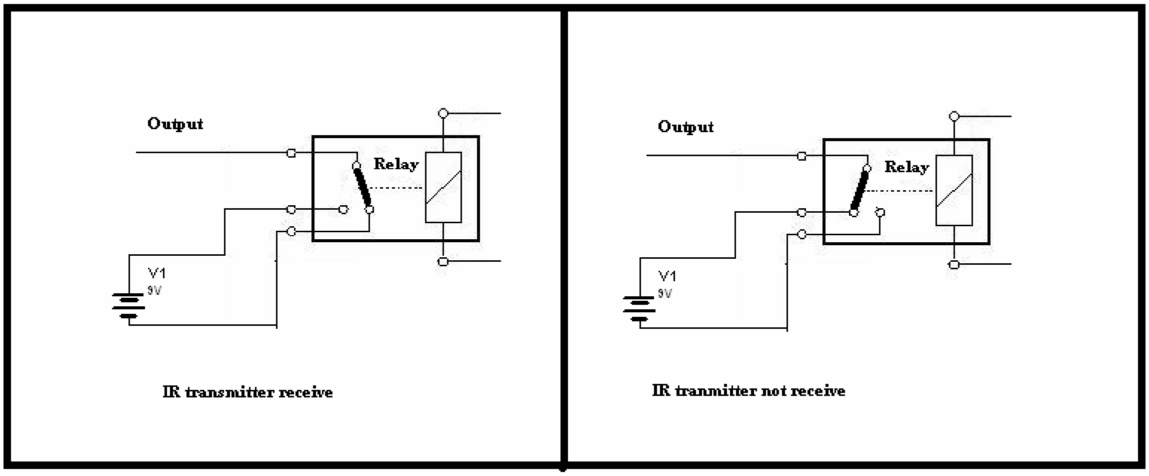


Figure 3.5 The relay condition

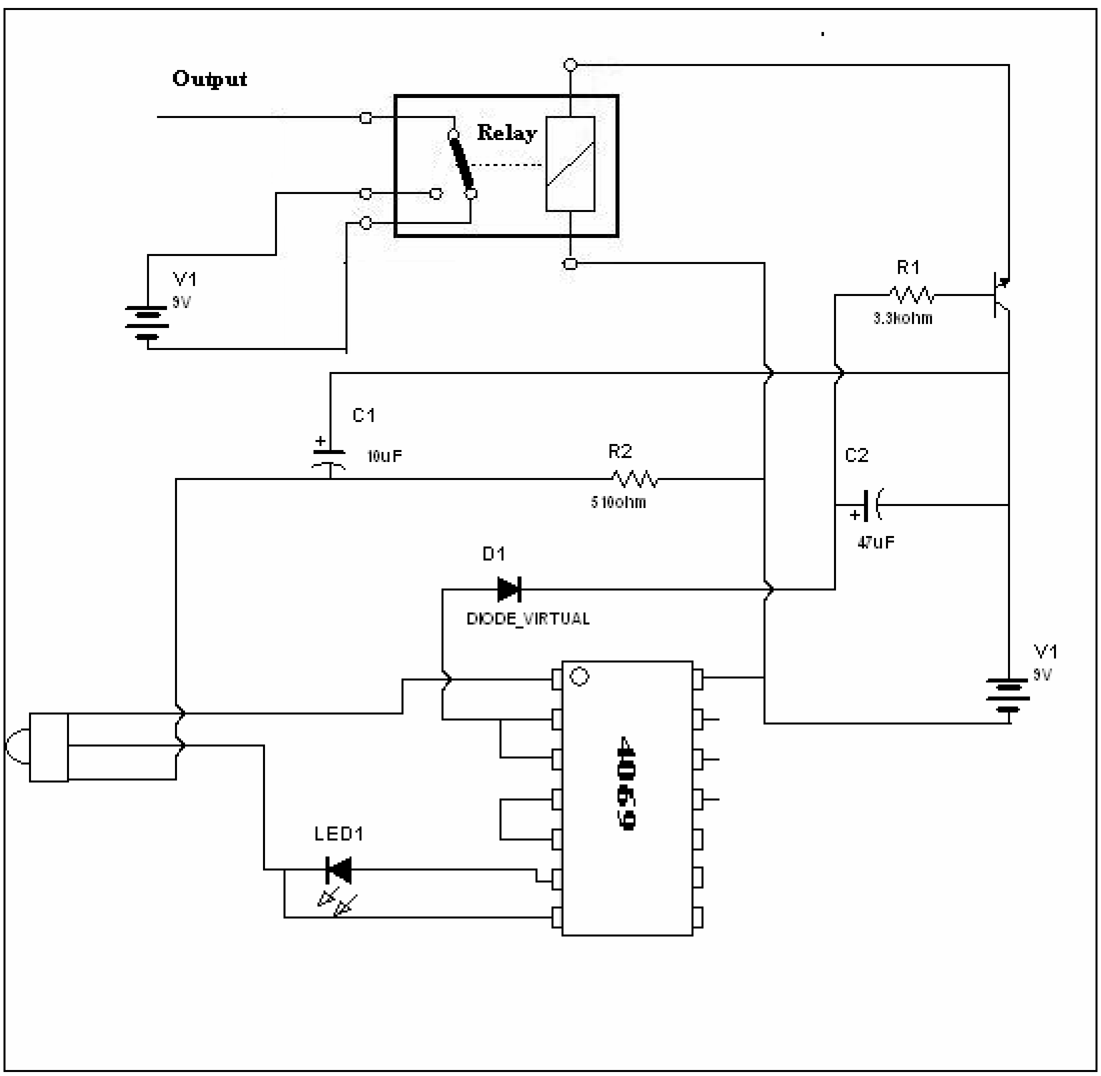


Figure 3.6 Infra-red receiver

Once the IR sensor is finish, the project will move to the next step. The next hardware to be design is the digital clock. Digital clock is function as timer for the system. The clock function is to enable the system to differentiate the time. IR sensor for an example will only work at a certain time only.

3.3.4 Digital Clock

Digital clock are use as a timer for the system. As stated earlier the system will activate the garden lighting after dark. The garden lights will remain on until a certain of time. Time will be set by the digital clock. That is the function for the digital clock, which to tell the time to the microcontroller. For example setting the time at 12 o'clock, the garden lighting will off but the system will remain on. If movement is detected the garden will reactivate for a certain of time. With the help of digital clock, it enable the microcontroller when to run a certain programmed.

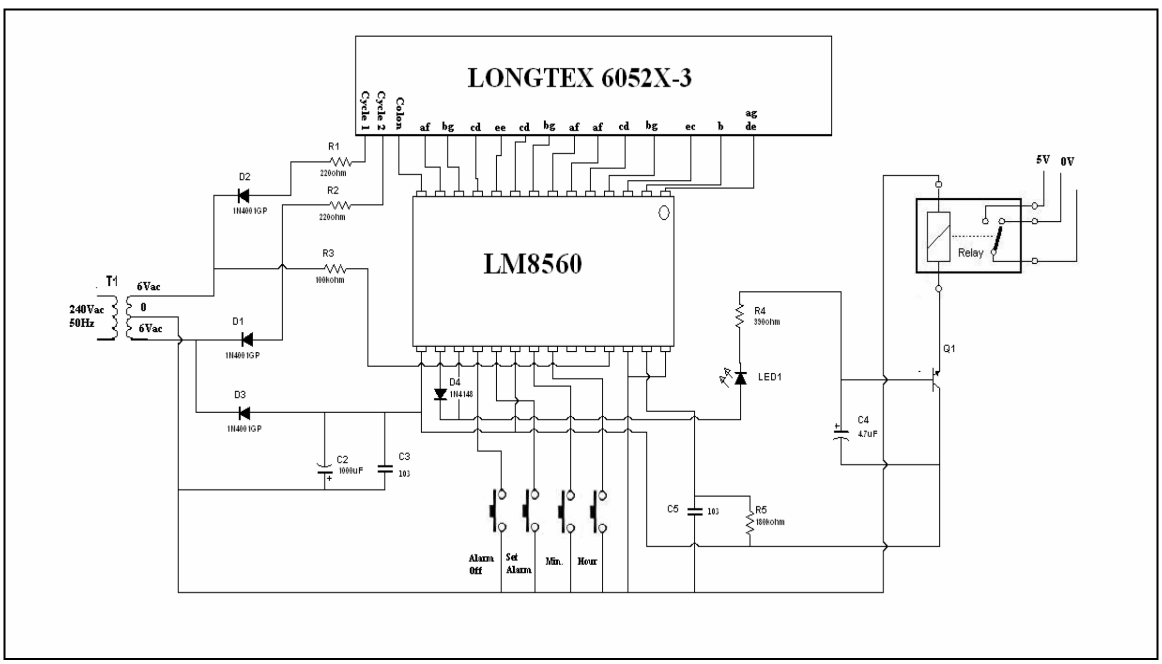


Figure 3.7 Digital clock. (Please refer to appendix B for larger image of the digital clock)

Figure 3.7 show the digital clock that are use as a timer in the project. Please refer to appendix B for a better image of the clock. The digital clock is equip with the ability to set the alarm, off the alarm, setting for minute and also setting for hour. Digital clock are running on AC source. So therefore to interface the digital clock with the microcontroller is using the relay.

3.3.5 Interface Circuit

Since the microcontroller produce output of 5V, it required another circuit so that it can control the main power. To control the main power a relay is used, however a microcontroller output is to low to energize the relay. So therefore another circuit was needed to energize and de-energize the relay.

The circuit that is able to do the task is the interface circuit. Shown in figure 3.7, is the circuit that will be use in this project. The circuit use a chip know as H11B1, also known as opto-coupler. There is another type of interface circuit. Please refer to appendix C for another type of interface circuit.

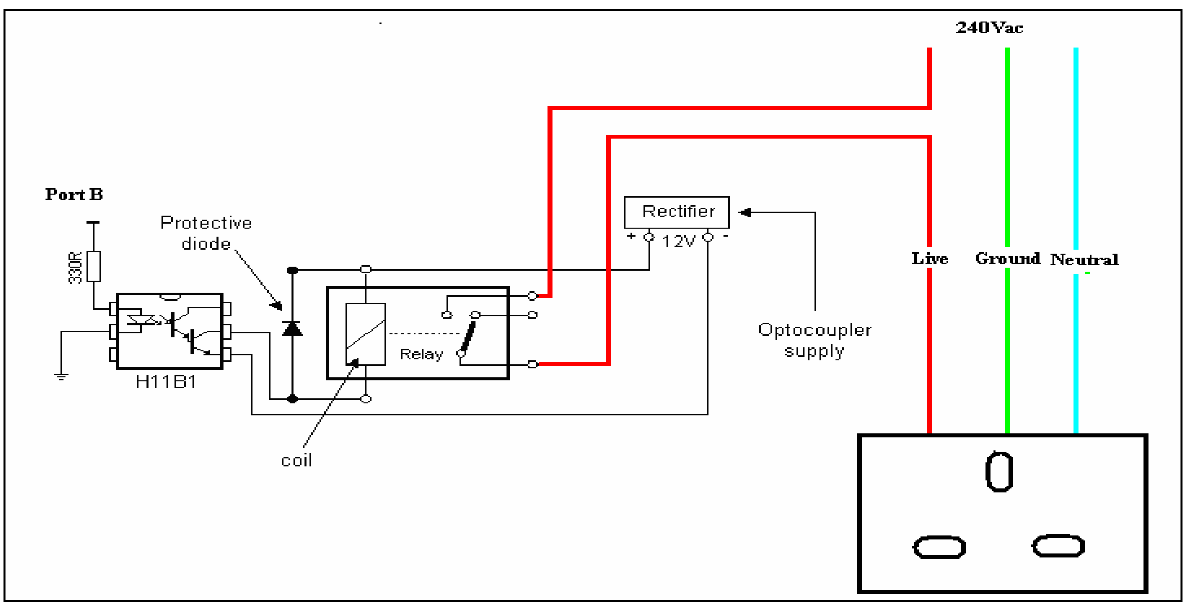


Figure 3.8 Interface circuit

Using opto-coupler give the best protection to the microcontroller. If overload occur, the opto-coupler will be damage however the microcontroller will be save. The another type of interface circuit shown in appendix c give the lowest protection. If overload occur, the microcontroller is most likely to be damage.

The interface circuit are tested at the lab. From the result, shown that the opto-coupler can receive voltage as low 2V is still able to energize the relay. So therefore microcontroller which produce 5V output can control the interface circuit.

3.4 INTERFACING HARDWARE TO MICROCONTROLLER

Once the hardware finish, the next step is to interface the hardware to the microcontroller. Interfacing the hardware is done by using transistor and also relay. Some of the hardware are unsuitable to use transistor. The only option is to use relay to interface. Digital clock for example is a circuit that used AC voltage. Microcontroller will not able to read an input from AC source.

So therefore interfacing is done so that the microcontroller is able to read the input and able to system as ordered. The first hardware that will be interface is the light sensor. Light sensor give the system ability to differentiate between day and night.

3.4.1 Interfacing Light Sensor

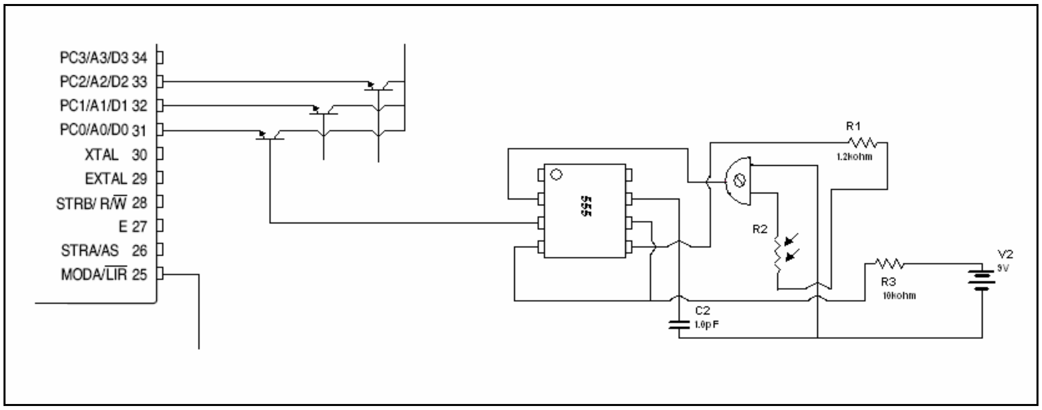


Figure 3.9 Interfacing light sensor and microcontroller

The interface between light sensor and microcontroller is done by using transistor as switching circuit. For the full circuit please refer to appendix D. The light sensor in this figure is different from the figure 3.3. During interfacing the circuit, the LED that are use as an indicator is remove from the circuit.

By this stage interfacing is done and so is the programmed develop for this stage. However the programmed development will be explain later on in programmed development. In this stage, when the sensor is detect that day has turn to night, the LED that are installed at the output of the microcontroller will on (Figure 3.2). If the sensor detect the condition is changing from night to day, then the LED will OFF.

If the system is running as expected, then the project will proceed to the next phase. The next phase will include the IR sensor. Interfacing the IR sensor will be using a relay as the interface circuit. The reason why a relay is used will be explain later on.

3.4.2 Interfacing IR Sensor

Interfacing the IR sensor is done with light sensor. The condition is that when IR sensor detects any movement the garden lighting will activate. However the main condition is that the garden lighting must activate at night only. So therefore if the IR sensor detects movement during the day, nothing will happen.

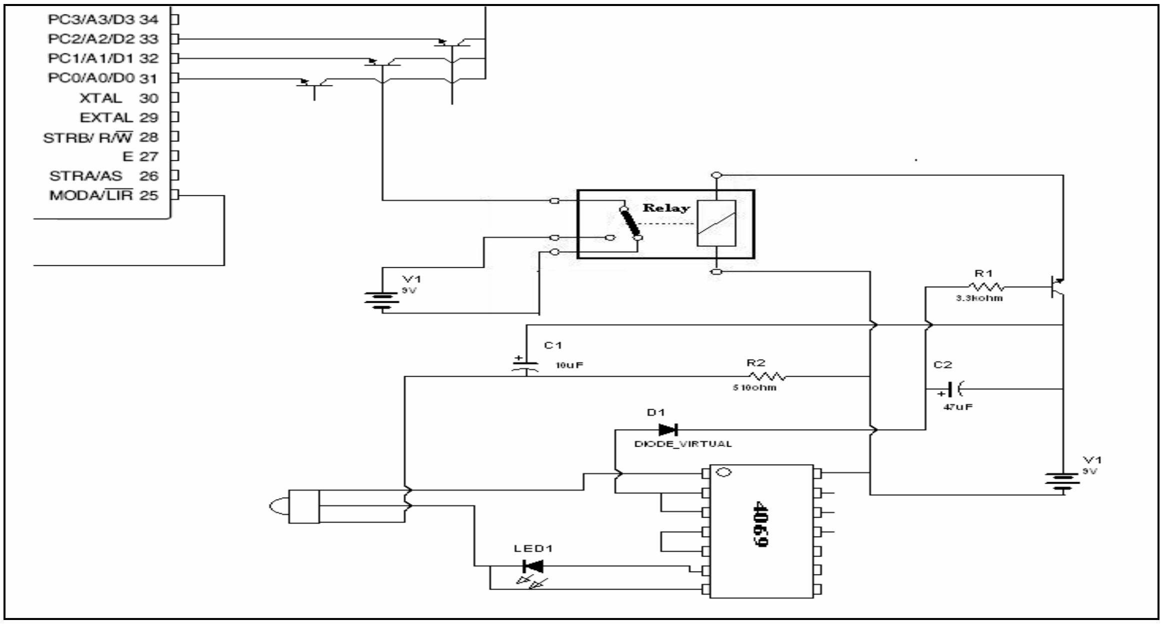


Figure 3.10 The interface circuit between IR sensor to the microcontroller

The interface circuit is shown in figure 3.9, the interface circuit is done using a relay and a transistor. For the full circuit of the interface please refer to appendix E. As stated earlier when the IR receive the transmit signal, the relay will be switch to normally open. The relay will switch back if the transmit signal is cut off.

The programmed develop will be explain later on in the programmed development. Once both sensor are running, the next step is to interface the digital clock. So that the system will be able to change mode from garden lighting to security lighting.

3.4.3 Interfacing Digital Clock

The digital clock that use in this project is using AC voltage source. Using a relay is the only option. The function of a digital clock is to tell the time for the system. This project is design to have two functions, which is as security system and also as garden lighting system. In the previous sub-chapter, the system should be able to function as security lights. Therefore in this chapter, the system should be able to function as garden lighting system and will change to security lighting system base on the setting of the time.

Since the circuit for the digital is to large please refer to appendix B for the circuit of the digital clock. Shown in figure 3.10, is the clock relay interface with the microcontroller. The full interface circuit is shown in appendix F. Since the figure is too large, only the connection of the relay is shown.

In appendix B, the digital clock has the function of setting the alarm. Once the alarm is set on, the LED which functions as an indicator will also on. If the LED is on, then the relay will also energize. Once energize, the input relay will be 5Vdc and 0Vdc. So therefore, the microcontroller will be able to read the correct input. The DC voltage source is connected from the microcontroller source. Please refer to appendix F.

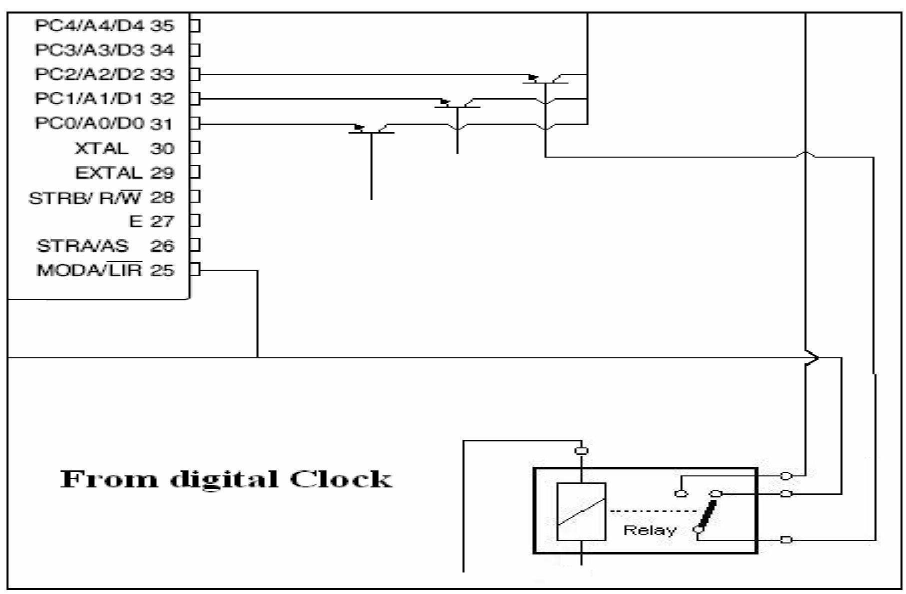


Figure 3.11 Relay interface the digital clock and the microcontroller

Once the digital clock is done, the project will enter the final phase. The final phase is building a power module for the system. Building the module will explain later on. The next subchapter will cover the programmed development. The programmed development is done along with the hardware.

3.5 PROGRAM DEVELOPMENT

Programmed development is done along with the development of the hardware. To programmed the microcontroller, certain chip and software must be installed. The chip are shown in figure 3.11, which is MAX233. For a complete circuit please refer to appendix G. The software that programmed the microcontroller also need to be installed. There are a number of such software, for example HCload, JBug11 and Wp11. In this project, Wp11 is chosen to be the software to be used. For the manual of using Wp11 please refer to appendix H.

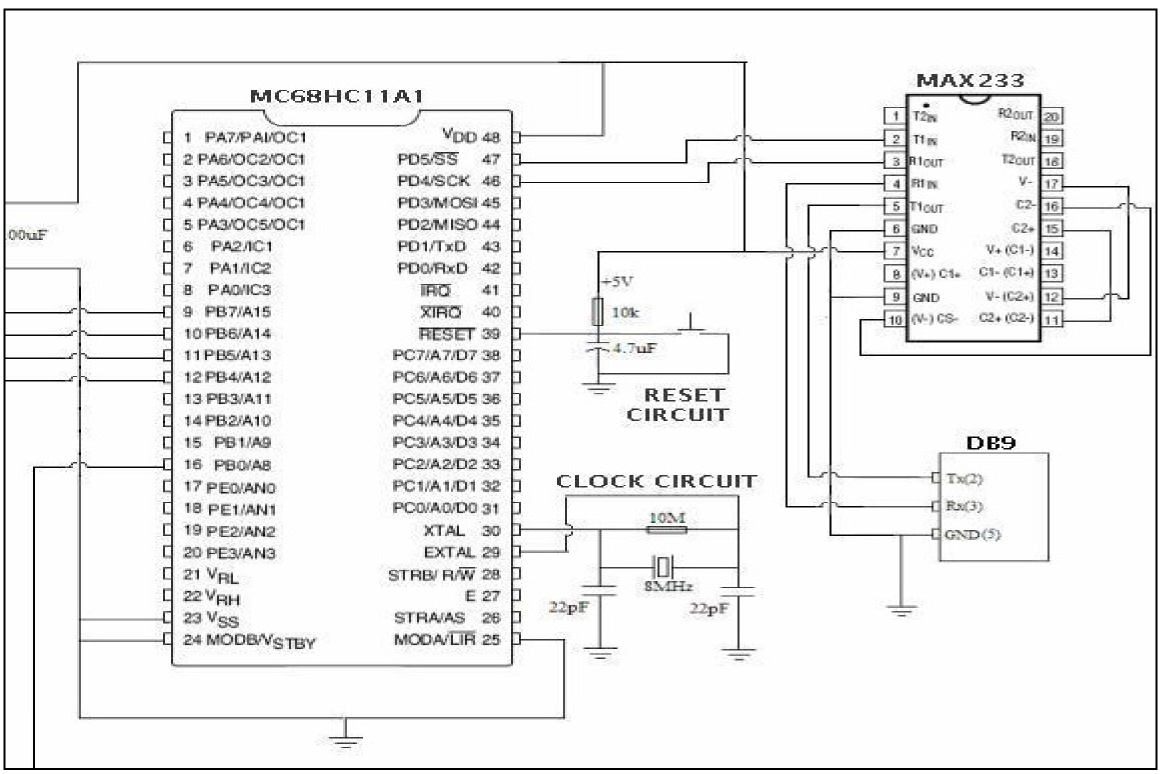


Figure 3.12 Microcontroller and MAX233.

3.5.1 Programmed Development For The Light Sensor

The programmed are done once the light sensor and microcontroller is functioning as expected. Interfacing the circuit are already explain in subchapter 3.4. So therefore for the programmed develop please refer to appendix I. In figure 3.12 show the flowchart of the program

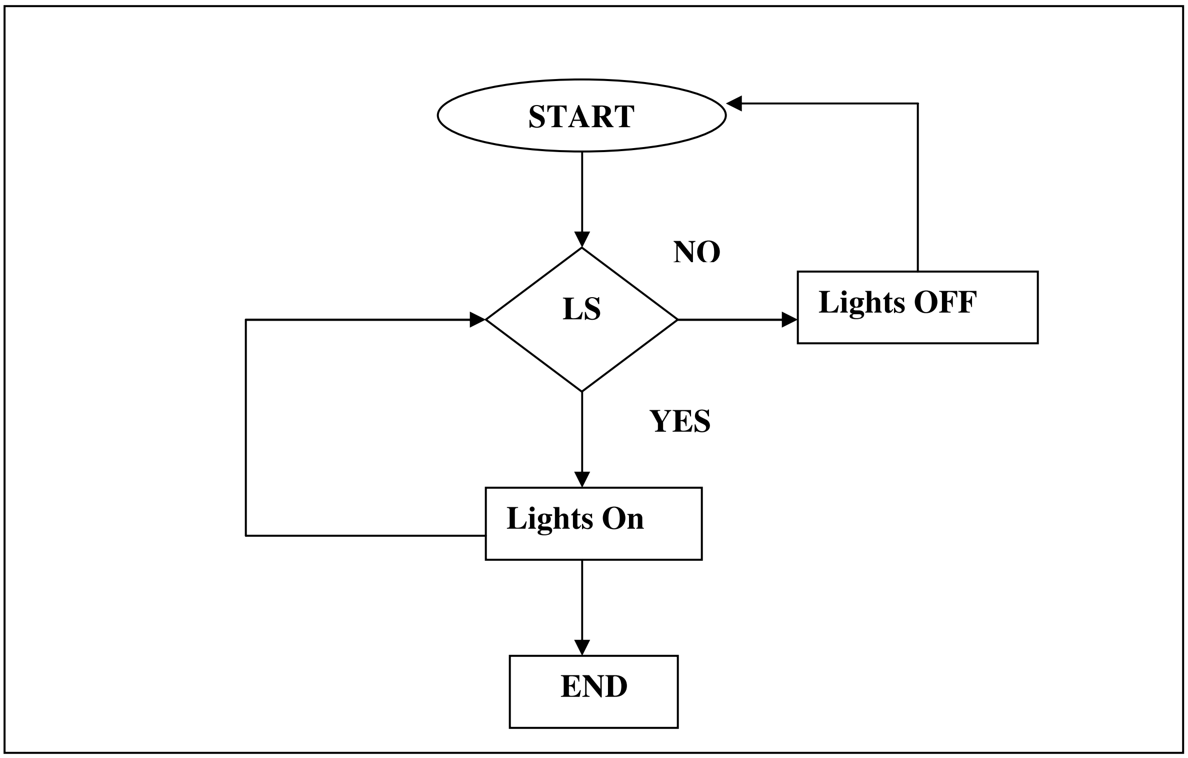


Figure 3.13 Programmed flowchart for the light sensor.

Base on the flowchart, if the day is turning to night the garden lighting will be activated. If the sensor detects day, the garden lighting will deactivate. The flowchart show that the process will be continuous. In this phase the system is already able to become automatic light switch for garden lighting. The next step, is to develop a programmed for light sensor and IR sensor. In the next phase, the system will be design for security lights instead of a garden lights.

3.5.2 Programmed Development For IR sensor

The development programmed are done by interfacing both the light sensor and infra-red sensor. By this stage, both of the light sensor and IR sensor are successfully interface to the microcontroller. For the programmed please refer to appendix J. In figure 3.12, is the flowchart of the programmed.

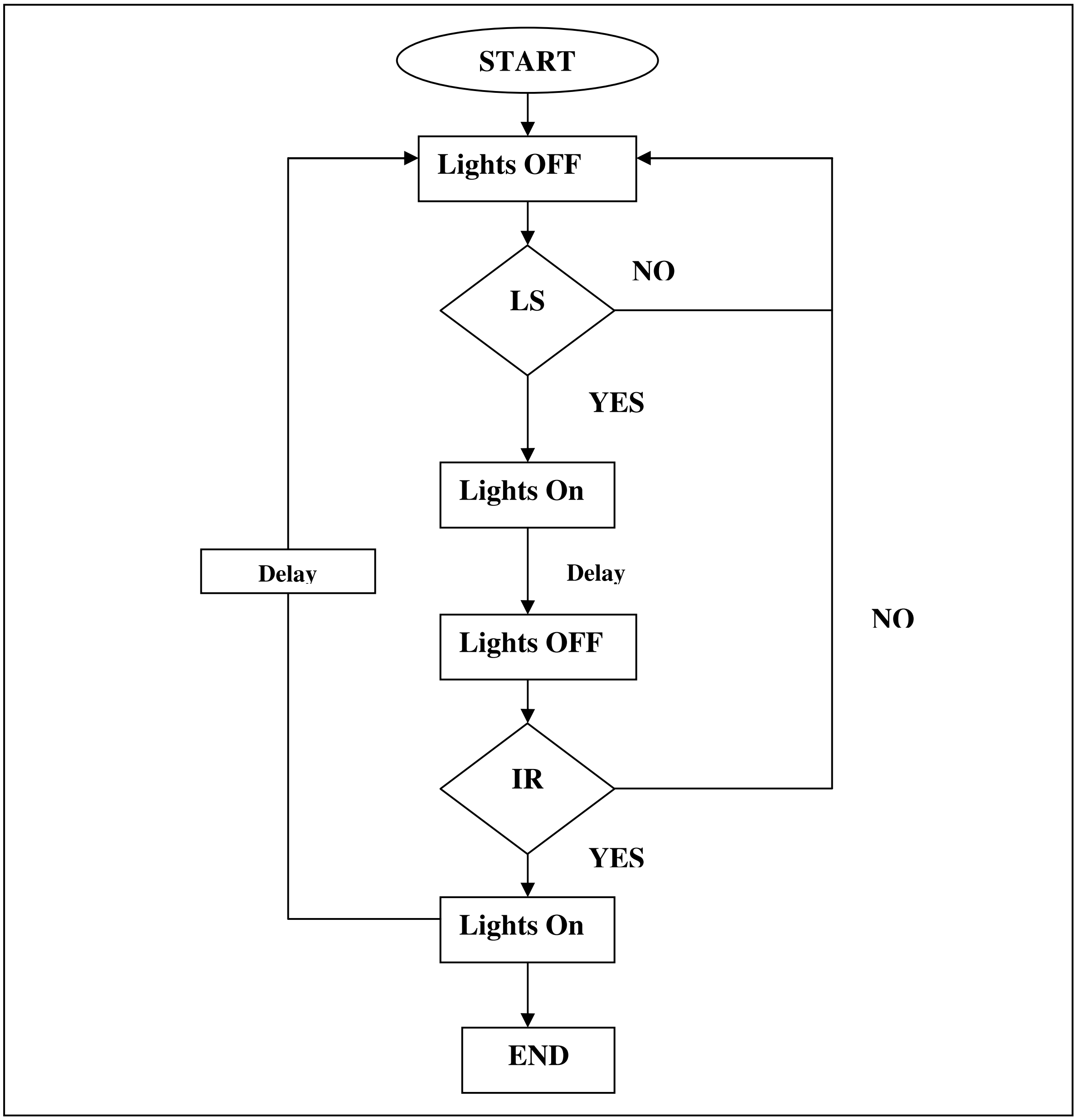


Figure 3.14 Flowchart for the IR sensor

The programmed is design for security lights purpose. Garden lighting will be activated once the IR sensor detects movement within its range. However the condition is, the time must be at night. By this stage, the project is more than half done. Proceeding to the next stage, which is programmed develop for the final stage. The programmed develop is for security and garden lighting purpose.

3.5.3 Programmed Develop For Light Sensor, IR Sensor & Digital Clock

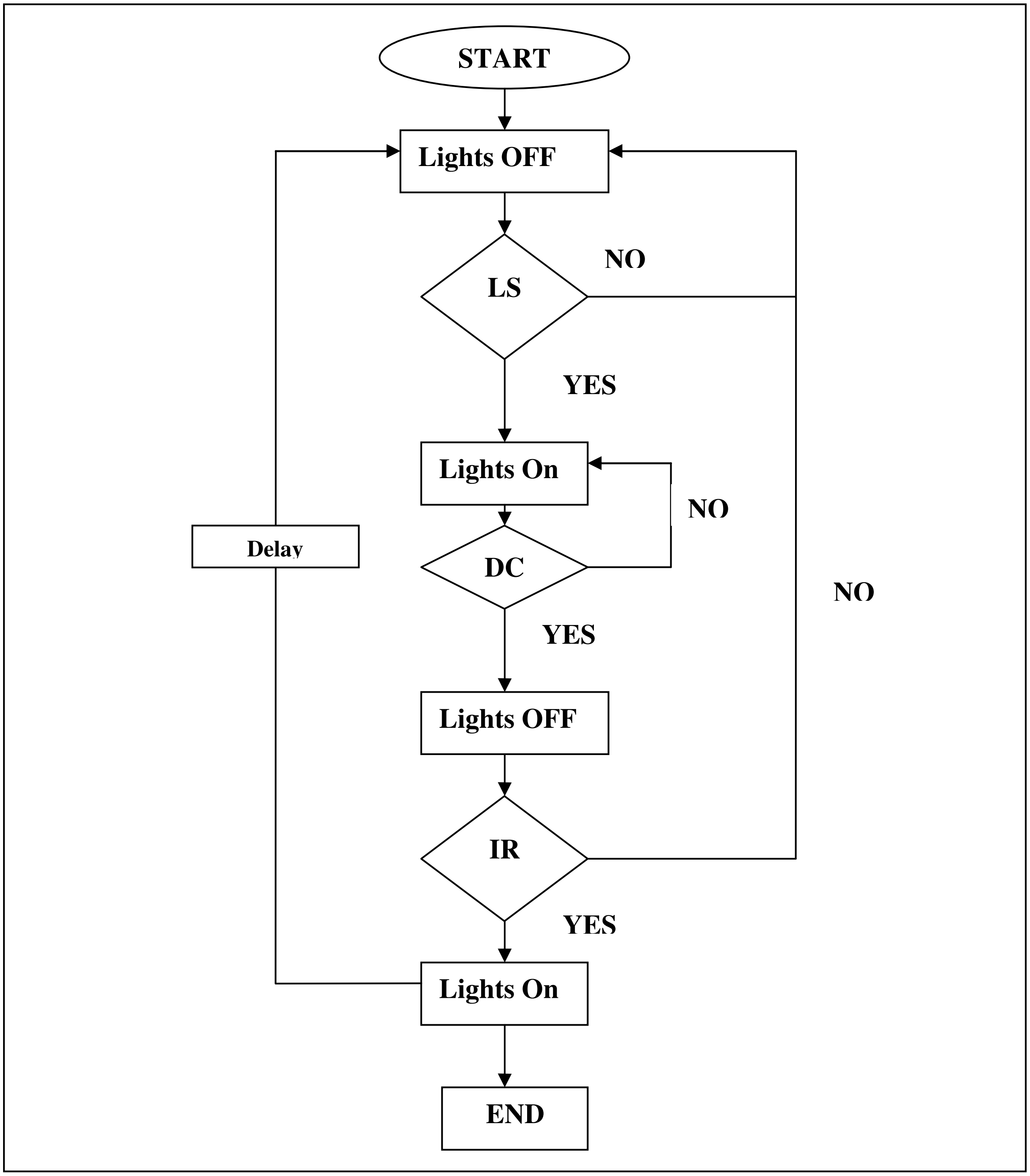


Figure 3.15 Flowchart programmed for system

This is the final part for the programmed development. For the programmed please refer to appendix K. The flowchart in figure 3.14 show the system is able to run as security lights and also garden lights. However the programmed in appendix K is also add with a few more functions. The main function is shown in the flowchart shown in figure 3.14. There are also other functions for the system. One of the functions is the system is able to be 'On' and 'Off' manually. The other functions are the system is able to run as garden lighting and also as security lights.

By the end of this chapter, the programmed develop has reach the final phase. Now the final phase is to design the system in one module. The design for the module will be explain in the next subchapter.

3.6 SYSTEM PROTOTYPE

Once all the hardware and programmed development is finish, the final phase is to built a module. The final result of this project is to show that this project is able to 'On' and 'OFF' the main power supply. To do so, certain safety features must be applied. One of them is to ensure main power is wired properly. And to avoid any accident, the project is built into a module. Figure 3.15 is the module for this project.

The system module is built using 2 module box. Each of the box size is 0.2m x 0.15m x 0.07m. The box are glued together to since one box is still to small. So therefore the real size of the module is 0.2m x 0.3m x 0.07m. The module has a 3 pin plug for the main power and also a socket for power regulator to supply 9Vdc for the microcontroller and other hardware. Figure 3.16 show the voltage regulator socket and the plug for the main power supply.

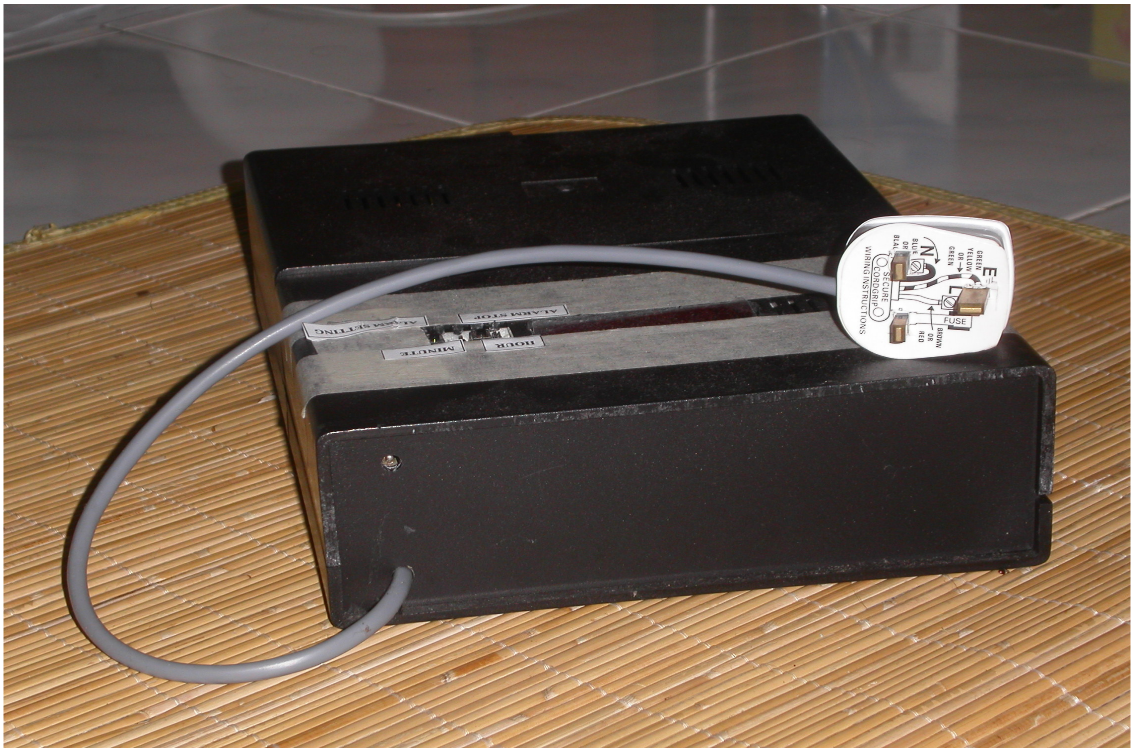


Figure 3.16 Prototype module

Figure 3.17 Power source for the module. 240Vac and 9Vdc for input source.

3.6.1 Module System

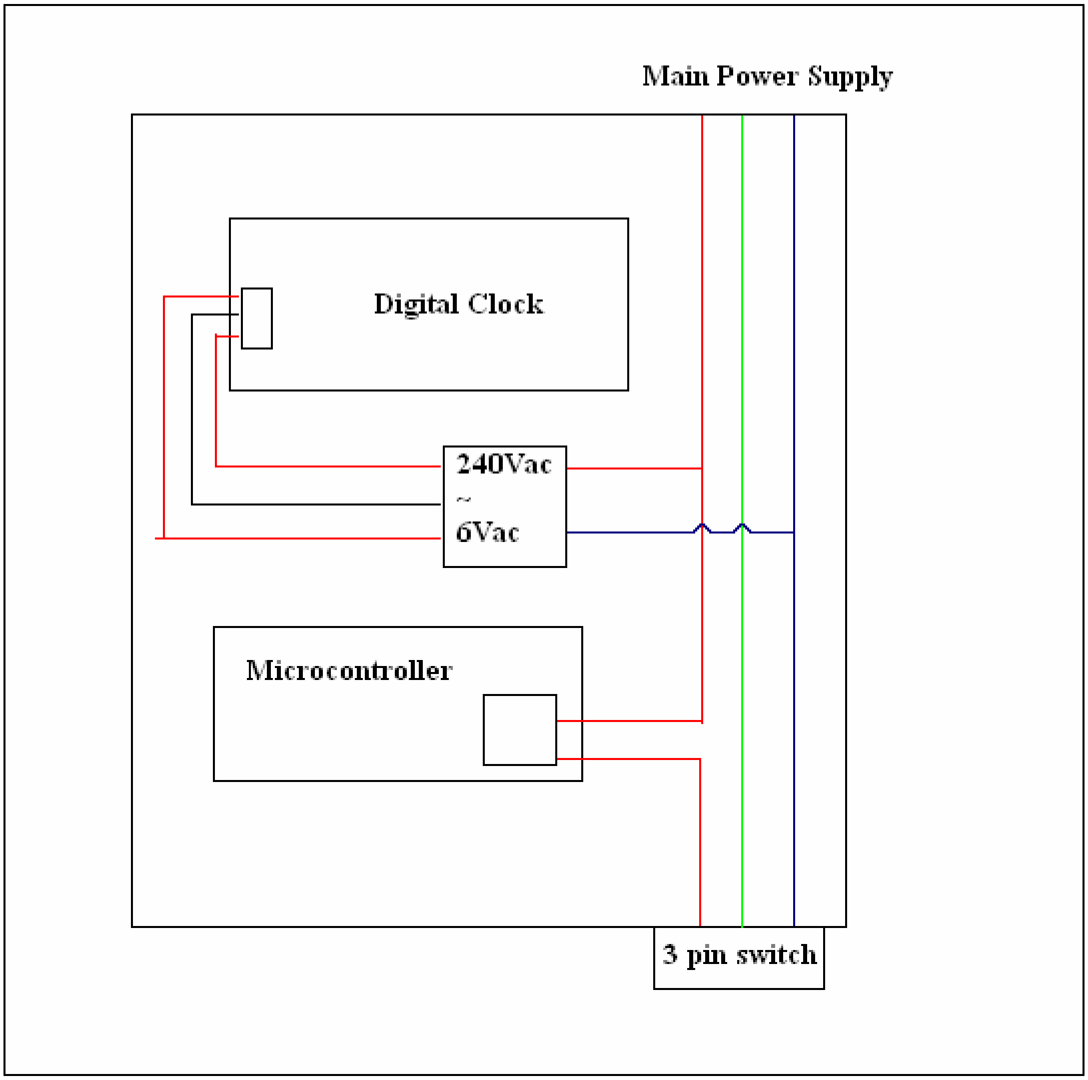


Figure 3.18 The module system using PVC box

Module system contains three hardware, which is the microcontroller, digital clock and interface circuit. Microcontroller is place inside the module since it needs to be interface with the interface circuit. Digital clock is also placed inside the module since it required AC source. More ever it can easily interface with the microcontroller. However, the most important thing is the hardware is built to be used indoors. Unlike the sensors, the role of the sensor is to be placed outdoors.

The hardware are insert into the module once all the hardware and software are finish. It is because once the microcontroller is inside the module, programming the microcontroller will become difficult.

3.7 SUMMARY

By the end of this chapter, all the process of completing this project are already being discuss and explain in detail. Which include how the hardware are interface with the microcontroller.

In the process of building the hardware design, programmed development must also be done. The successful of this project is depend on the success of interfacing the hardware and also the programmed develop.

The problem of interfacing the hardware is one the reason the project took a very long time to be finish. In chapter 4, the hardware will undergo some testing. For example the testing of the microcontroller. The testing of the microcontroller is divided to three phase. Which will be further explain the next chapter.

CHAPTER IV

RESULT AND ANALYSIS

4.1 INTRODUCTION

This chapter present the result and analysis done during the development of the project. The analysis is done to ensure the capability and the functionality of the hardware. Each of the hardware is set-up so that the hardware can perform optimally. However for the IR sensor, the set-up is set from optimal to the lowest condition. IR sensor is tested to measure its maximum and minimum range. This set-up and test will be explained later.

Since the project is using a microcontroller, it required to be test phase by phase. If the microcontroller does not past any of the phase than it will effected the outcome of the project. The testing done for the microcontroller will be further explain in the next subchapter. Other test will also include testing the range of the IR sensor. The objective of the test is to compare the input voltage is proportional to the range of the sensor. By the end of this chapter, the test result will be a guideline in the future development.

4.2 TEST SET-UP

Test set-up is about setting the hardware during the testing. Most of the hardware are set so that it will perform optimally. During the testing, the hardware will be using 9V battery. Since it can supply 9V fix. By the end of this project, the battery is replace with voltage regulator. Battery is only used for testing purpose only.

4.2.1 Set-up For Microcontroller

Microcontroller is supplied with 9Vdc during the test. 9Vdc is supplied to the voltage regulator. So therefore the voltage supplied to the microcontroller will be 5V. It also need to be supplied with 2MHz clock input. Clock input will be tested in phase one. If the clock value is other than 2MHz, then there is a problem with the microcontroller. The testing will stop until the problem is found. After phases one, the microcontroller set-up will be 5Vdc and 2Mhz clock input until the end of the test. This set-up will also be used until the end of the project.

4.2.2 Set-up For IR Sensor

Voltage regulator at the lab will be used for this testing. The voltage regulator need to supply 0Vdc, 1Vdc, 2Vdc, 3Vdc, 4Vdc, 5Vdc, 6Vdc, 7Vdc, 8Vdc and 9Vdc fix. IR receiver will be supplied with 9Vdc fix during the test. While the IR transmitter input supply will be varied from 0Vdc-9Vdc. The objective is to see whether the range of the IR sensor is proportional with the input voltage. If the range is proportional with the input voltage, then the maximum and minimum range is measured.

4.2.3 Set-up For Light Sensor

The light sensor will be supplied with 9Vdc during the test. Testing the light sensor will be tested under different condition. These condition are indoor, outdoor and direct light. Indoor condition will be tested in the room. The sensitivity of the light sensor will be set differently under different condition.

4.3 TESTING THE MICROCONTROLLER.

Testing the microcontroller is done phase by phase. Phase one required to be done at the lab. Since this phase required an oscilloscope. Phase two is to be done by programming a test program. The test program will be further explain. Since the microcontroller is in bootstrap mode, the testing can be done at the hostel. The third phase is switching. This phase is done using the light sensor as a switch.

Before going into the testing phase, power supply and reset circuit must also to be tested.

4.3.1 POWER SUPPLY AND RESET CIRCUIT

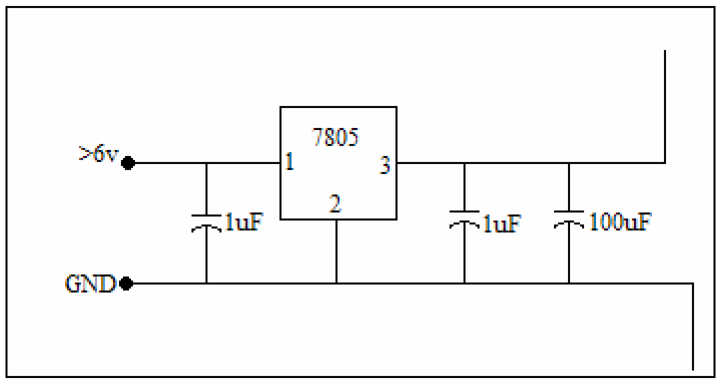


Figure 4.1 Voltage regulator

Figure 4.1 show the voltage regulator for the microcontroller. Since the microcontroller is a very sensitive chip, it required a supply voltage of 5V. The solution is using the voltage regulator shown in figure 4.1, the output voltage is suppose to be 5V-6V. If the voltage is higher than 6V than it is not recommended to be used.

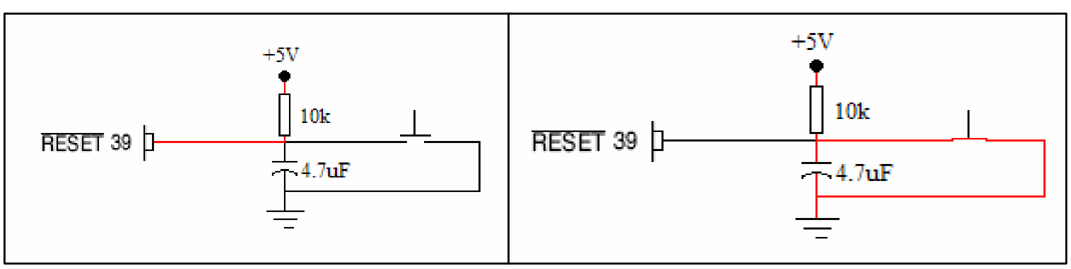


Figure 4.2 Reset circuit.

In figure 4.2 show the reset circuit. Reset pin in the microcontroller board is at pin 39. Shown in figure 4.2(left side), is the flow of voltage in the circuit when the switch is open. The pin function is to reset the microcontroller to its original state when the pin receive 0V input. The reset circuit function is to give 5V input continuously. When the switch is press, the capacitor will store all the voltage. Therefore the voltage at pin 39 will drop to 0V.

4.3.2 PHASE 1

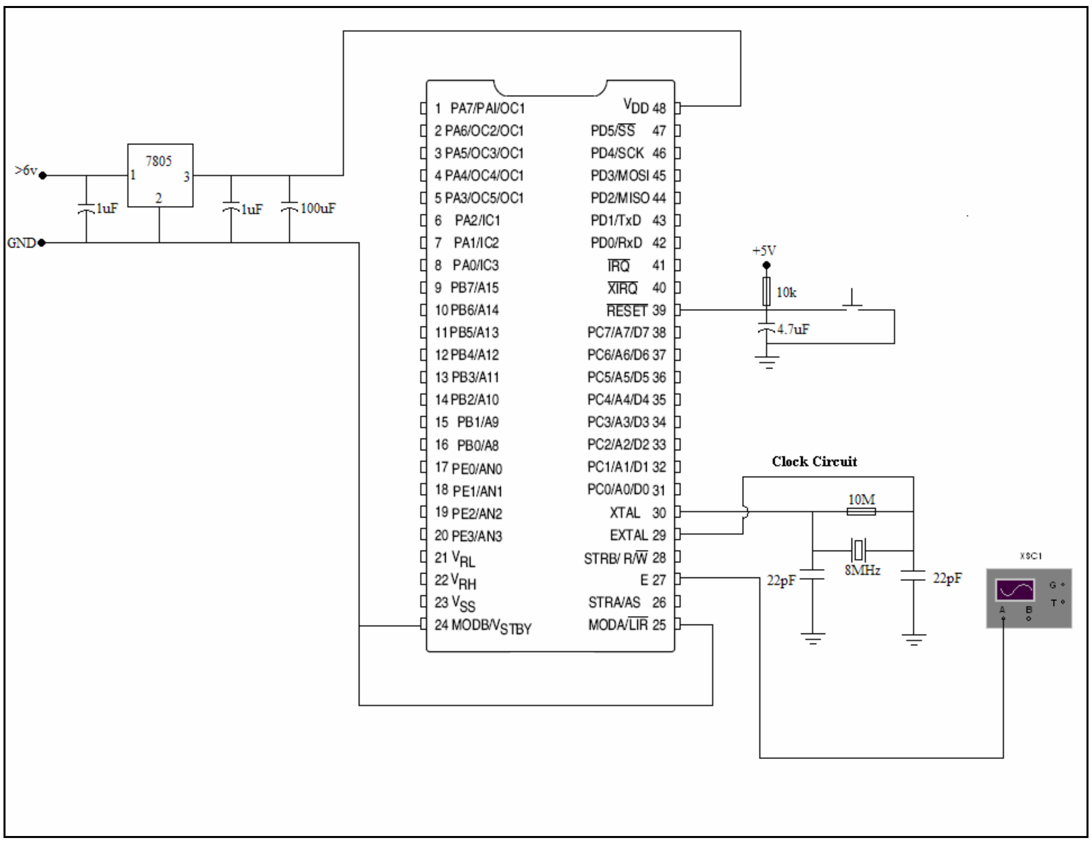


Figure 4.3 Phase 1 circuit

Figure 4.3 show the microcontroller circuit use in phase 1. The test must be done at the lab. Since this phase required an oscilloscope to test the clock. Clock function is to generate a periodic sequence of pulse. MC68HC11A1 is able to generate it own clock with the help of clock circuit shown in the figure 4.3. This clock is often referred to as the system clock or the E clock. When the E clock output is low, an internal process is taking place. When it is high, the microcontroller is writing or reading data. If the clock does not function, then the microcontroller will also fail to function.

For this phase, one of the oscilloscope channel is connect to E clock(pin 27). The value of E clock should be one fourth from the value of the crystal. So therefore to calculate the value:

$$\frac{1}{4} \times 8MHz = 2MHz \qquad \textbf{(4.1)}$$

So therefore, the value of E clock should be 2MHz. If other value is different than 2Mhz, then the microcontroller should be replace. The microcontroller might cause the project to fail..

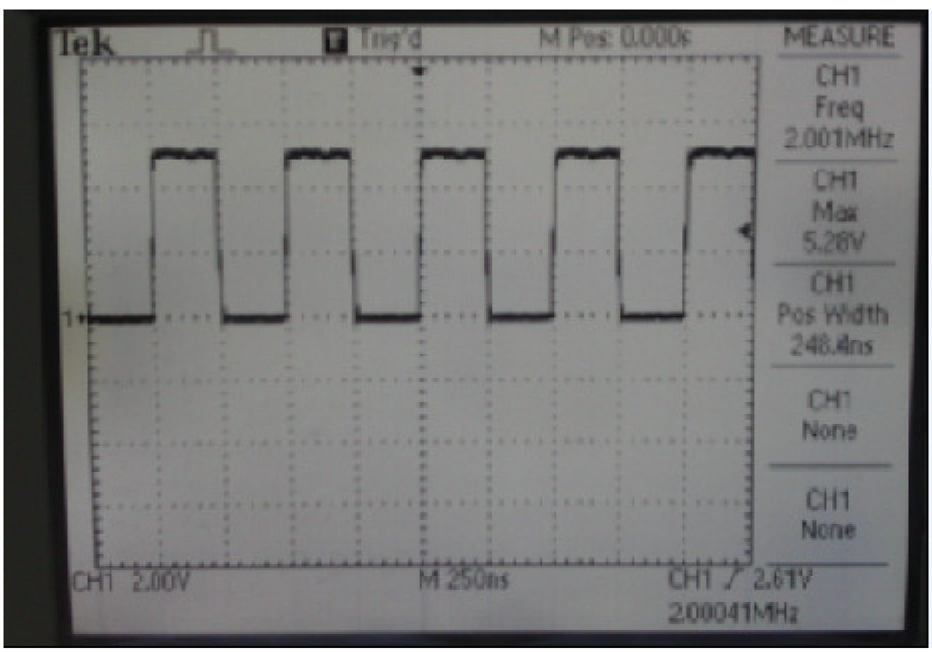


Figure 4.4 The E clock output

Figure 4.4 show the output from pin E. The graph show the clock is functioning well. The E clock frequency is exactly 2MHz. The result of phase 1 is shown in table 4.1. Other measurements are also stated in the table. This test proved that the microcontroller is in a good condition. So therefore moving to the next phase, which is programmed the microcontroller with a test programmed.

Frequency (Hz)	2.000MHz
Max Voltage (V)	5.28V
Minimum Voltage (V)	-239mV
Pos Width (second)	247.4ns
Period	500.0ns
Vmean	2.48V
V_{pk-pk} (V)	5.44V
Fall Time (s)	8.300ns
Rise Time (s)	10.00ns
Cyc Rms (V)	3.54V
Neg Width	251.2ns

Table 4.1 Result from the osciloscope

4.3.3 Phase 2

Once phase 1 is tested and the microcontroller is functioning as expected than the test can be continued to phase 2. Phase 2 objective is to programmed a simple programmed into the microcontroller. For this part, a set of LED play an important role.

Shown in figure 4.5, port B is used as the output. All of port B are connected to a LED. The result that we should see is the LED is running a simple form command. Please refer to appendix L for a set of test programmed. To programmed the microcontroller, ensure the microcontroller is able to initialize with the PC.

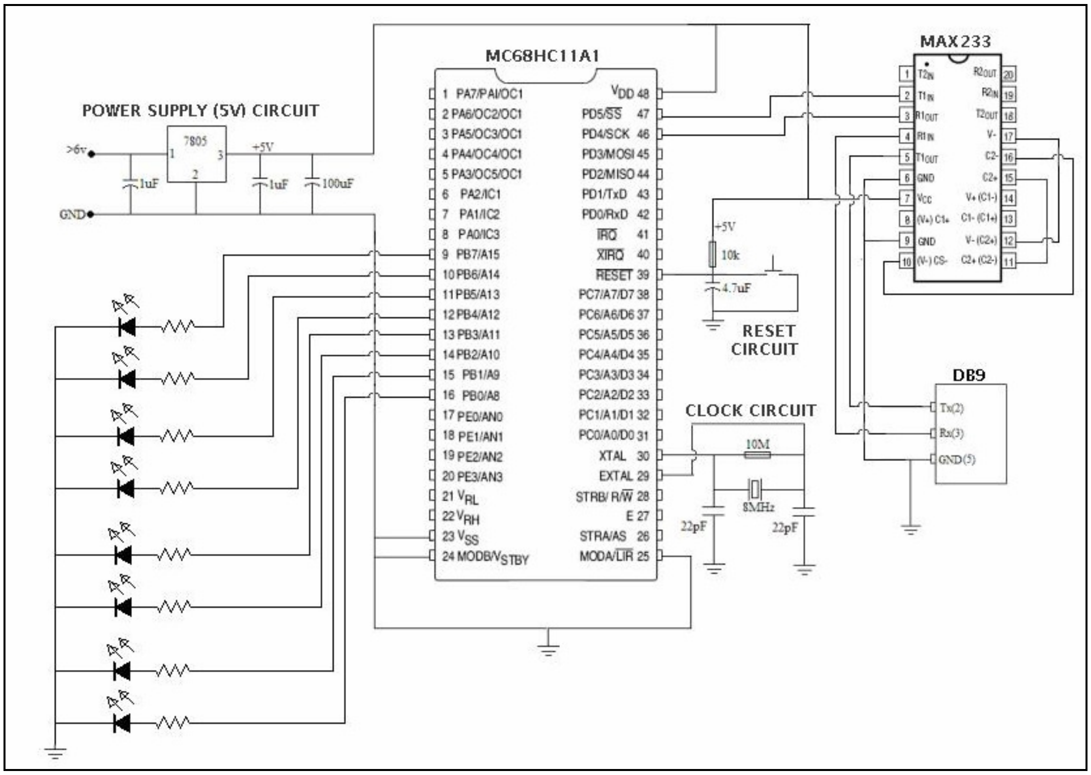


Figure 4.5 Microcontroller circuit phase 2.

Programmed	Result
Disco	LED at PB0, PB2, PB4 and PB6 will 'ON'. When these LED are 'OFF' PB1, PB3, PB5 and PB7 will 'ON". The process will repeat continuously, showing as if the lights are dancing.
Free Running	LED at PB0 will 'ON' while the other are off. Start from PB0 to PB7 the LED will 'ON' simultaneously showing as if the LED is running.
Flash	PB0-PB7 will 'ON' and 'OFF" simultaneously.
Knight	Starts at the middle of the LED. Which is PB4 and PB3. From the middle the LED will run to the side. This form is refer as knight rider lights.

Table 4.2 Result for the test programmed

Table 4.2 show the result that will obtain using the test programmed. Phase 2 is done if the LED is running as programmed. In this phase we can conclude that the microcontroller is running as expected. So therefore moving to the next and final phase which is switching.

4.3.4 Phase 3

Phase 3 is to test the function of switching. Please refer to appendix M for the programmed that will be use. Before continuing this phase, ensure that the other pin at Port C are to be grounded. If the pin are ungrounded that it might effect the result. The other pin that are not used will be grounded. If not the microcontroller might read the wrong data. . So therefore using the programmed in appendix M, the result for this phase should be:

Switch	**Result**
Switch 1 (PC0)	Pin PC0, PC2, PC4 and PC6 are on
Switch 2 (PC7)	Pin PC1, PC3, PC5 and PC7 are on

Table 4.3 Result for the switching programmed

By the end of this testing, it has been confirm that the microcontroller is working as expected. So therefore the project can proceed with program development in chapter 3.

4.4 TESTING THE RANGE OF INFRA-RED SENSOR

The objective of this test, is to show that the range of the IR sensor is proportional to the input voltage. If the voltage is low than the transmitter might not be able to transmit at all. The test is done by setting the supply of the IR receiver 9V. IR transmitter supply will varies from 9V~0V.

IR transmitter	**IR receiver**	**Range (m)**

(Voltage)	(Voltage)	
9V	**9V**	**0.20m**
8V	**9V**	**0.175m**
7V	**9V**	**0.15m**
6V	**9V**	**0.125m**
5V	**9V**	**0.10m**
4V	**9V**	**0.06m**
3V	**9V**	**0**
2V	**9V**	**0**
1V	**9V**	**0**

Table 4.4 Test range for the IR sensor.

The result from the test is stated in table 4.3. From the result, the minimum supply voltage for the IR transmitter is 4V. Lower than 4V, the IR transmitter will not able to transmit since the supply voltage is too low. So therefore for the IR transmitter to transmit the longest range, it is recommended to supply the transmitter with 9V.

4.5 TESTING THE LIGHT SENSOR

As stated in chapter 3, the light sensor should be able to change its sensitivity. In theory the lower the resistance, the more sensitive the sensor should be. In order to prove this, the light sensor will be change the sensitivity. The resistive of the trimmer will varies to prove the sensitivity. The test will be done in three condition, in a room condition, outside and direct light. This test is done during the day. So therefore the test is to see whether the light sensor indicator will on or not. ‘NO’ means the LED is off. ‘Yes’ mean the LED is on. Directs light mean that the test is done by using torch light directly to the LDR.

Trimmer (kOhm)	Room	Outside	Direct Light
22	No	No	Yes
20	No	No	Yes
18	No	No	Yes
16	No	No	Yes
14	No	No	Yes
12	No	No	Yes
10	No	No	Yes
8	No	Yes	Yes
6	No	Yes	Yes
5	No	Yes	Yes
4	Yes	Yes	Yes
2	Yes	Yes	Yes
0	Yes	Yes	Yes

Table 4.5 The result of testing the light sensor.

From the result given, it is appropriate to set the trimmer higher than10kOhm. So therefore the indicator will detect when the condition is at night. However the test is limited. Other condition such as when the day is cloudy. During this test the condition is sunny. So therefore further test are recommended to ensure the sensitivity of the light sensor.

4.6 SUMMARY

In conclusion, by the end of this project certain limitation has been recognized. The result alone will be use as a guideline for the future development. IR sensor is suitable to be replace with a better quality. The sensor use in this project are self-made. There are other IR sensors that can give better range. The sensors use in this project play an important part in the system, better sensor will provide better efficiency.

CHAPTER V

SUMMARY AND CONCLUSION

5.1 SUMMARY

The project of building an autonomous lighting and security system is a dual-system that can function both as security system and garden lighting system. This project which takes nearly 14 week for completion is still in the level of a prototype. Further improvement need to be done before commercialization.

Most of the hardware that has been applied has given the expected result. The IR sensor needed to be replaced with a better sensor. Since the IR sensor can only transmit for a very short range. This drawback is one of the limitations in this project. However, this project has proven that all the hardware can be interfaced to the microcontroller. Overall the hardware interfacing and programmed development have met the expected result.

However the most important results, is the successful of using the microcontroller to control the main power supply. Using the interface circuit, the microcontroller is able to activate and deactivate the main power. This is the result expected from this project. So therefore any garden lighting can used this project.

5.2 CONCLUSION

1. Based on the test done in chapter 4, the light sensor detects darkness at preset darkness. The light sensor also able to change the level of sensitivity.

2. The IR sensor is able to perform as a motion sensor. Even though the sensor range is quite small. Therefore it is suggested to use other type of sensor that can provide better range.

3. The IR sensor and light sensor are able to function independently before interfacing to the microcontroller.

4. A digital clock with an alarm function is build. The digital clock alarm is an important part to interface the digital clock to the microcontroller.

5. Build an interface circuit that is able to receive low voltage input to control the activation and de-activation of the main power supply.

6. Build a microcontroller MC68HC11A1 circuit as the main controller in the system. For this project, the microcontroller will run in bootstrap mode.

7. All of the hardware are successfully interfaced with the microcontroller.

5.3 LIMITATION IN THIS PROJECT

From the analysis conducted in chapter 4, the system design has many limitations. Most of the chips use required voltage to be not more than 9V. Especially the microcontroller, the input voltage for the microcontroller must no be more than 6V. The IR sensor has very small range. It is suggested for the sensor to be replace with a better one. Another weakness is the digital clock required manual setting on the alarm. Also setting off the alarm need to be done manually. These limitation has shown that the system is still far from perfect.

5.4 SUGGESTION FOR IMPROVEMENT

1. Due to budget constraint, improvement such as using a better IR sensor instead of using self-made, will give a better result. For suggestion it is better to replace the IR sensor for a better one.

2. For the second suggestion, it is probably better to replace the microcontroller with a PIC. The advantage of using PIC is because it is much smaller in size and also much cheaper. So therefore, the project will reach its goals, which is to build a low cost security system.

3. Replace the digital clock with a timer. Using timer, this project will be able to run fully automatic. The limitation of the clock is, digital clock will required the alarm setting to be manually. If timer is used, the system will be able to function fully automatically..

5.4.1 Cost & Commercialization

Overall cost for this project is shown in table 5.1. The cost consists the list of parts and components that obtain before starting this project. After finishing the project, the total cost is actually higher than the list shown table 5.1.

Bil	Bahan/Komponen	Spesifikasi	Anggaran Harga / unit (RM)	Kuantiti	Anggaran Harga (RM)

1	Strip board (10" x 4 ")		3.00	3	9.00
2	Single Phase rectifier bridge	1 amp bridge	4.00	1	4.00
3	MC68HC11A1		40.00	1	40.00
4	Capacitor 470uF		0.30	1	0.30
5	7805		2.00	1	2.00
6	Capacitor 10uF		0.30	1	0.30
7	Resistor 5.1k		0.30	1	0.30
8	Light Dependent Resistor MPY54C569	Low resistance when dark	1.00	1	1.00
9	Resistor 10ohm		0.30	1	0.30
10	Crystal 8Mhz		2.00	1	2.00
11	Capacitor 22pF		0.30	2	0.60
12	Resistor 470ohm		0.30	1	0.30
13	Potentiometer 10K		1.00	1	1.00
14	Optocoupler	H11B1	1.00	1	1.00
15	Relay	12V relay Able to interface with high voltage (240V)	10.00	1	10.00
16	Fuse 13amp		0.50	1	0.50
17	Reset switch	4 pin	0.50	1	0.50
18	DIL switch	8 pin	1.00	1	1.00
19	74121		3.00	1	3.00
20	7490		3.00	6	3.00
21	74196		3.00	1	3.00
22	7447		3.00	5	3.00
23	3 pin plug	240Vac source	5.00	1	5.00
24	Transformer	240Vac to 9V ac	30.00	1	30.00
25	Solder iron	1 roll	7.00	1	7.00
26	Wire wrap	1 roll	3.00	1	3.00
27	MAX233		5.00	1	5.00
28	IC 4069		3.00	1	3.00
29	Resistor 510 ohm		0.30	1	0.30
30	Resistor 3.3k Ohm		0.30	1	0.30
31	Capacitor 10uF		0.30	1	0.30
32	Capacitor 47uF		0.30	1	0.30
33	Transistor	9013 NPN	1.00	1	1.00

34	Diode	IN4001/4002	0.50	1	0.50
35	Infra-red receiver (IRR)		5.00	1	5.00
36	IC base 48 pin		1.00	1	1.00
37	IC base 14 pin		1.00	8	8.00
38	IC base 16 pin		1.00	5	5.00
	JUMLAH ANGGARAN HARGA				**165.80**

Table 5.1 Project cost

From the test and result that has done, overall this project can be used for commercial purpose. The system is perfect for a low cost security system. Moreover, this type of lights security system is still unavailable in the market. The advantage of this project is it gives better night vision in a garden that has full of trees and other objects. Therefore commercialization this project might replace the previous system. It is better to applied the suggestion before commercialize this system.

APPENDIX A

Part (Description)	Quantity
Microcontroller	
MC68HC11A1	1
Voltage regulato 7805	1
Capacitor 1uF	2
Capacitor 100uF	1
Capacitor 4.7uF	1
Capacitor 22pF	2
Reset switch	1
Resistor 10Mohm	1
Crystal 8Mhz	1
Light Sensor	
Resistor 10kOhm	1
Resistor 1.2kOhm	2
Resistor 470Ohm	1
Capacitor 0.1(104)	1
Capacitor 1uF	1
Trimmer 22kOhm	1
Light Dependent Resistor	1
555 timer (8 pin)	1
Transistor 9013	1
LED	1

Infra-Red Sensor	
Transmitter	
555 Timer(8 pin)	1
Resistor 1.2KOhm	1
Resistor 39kOhm	1
Resistor 100Ohm	1
Capacitor 470pF	1
Capacitor 0.1(104)	2
LED	1
Infra-red diode	1
Receiver	
IC(4069)	1
Infra-red receiver	1
Diode	1
Capacitor 47uF	1
Capacitor 10uF	1
Resistor 3.3kOhm	1
Resistor 510Ohm	1
LED	1
Transistor	1
Relay (6V)	1

Table 1: List of components

Digital Clock	
IC (LM8560)	1
Push button(Reset switch)	4
LED	1
Longtex 6052X-3	1
Resistor 100kOhm	1
Resistor 390Ohm	1
Resistor 180kOhm	1
Resistor 220Ohm	2
Capacitor 1000uF	1
Capacitor 4.7uF	1
Capacitor 0.01(103)	2
Diode 4001 (SI diode)	3
Diode 4148 (PN diode)	1
Transistor (9013)	1
Miscellaneous	
Power adapter (0Vdc-9Vdc)	1
Max233	1
Ribbon cable (0.25M)	1

Table 1: List of components

APPENDIX B

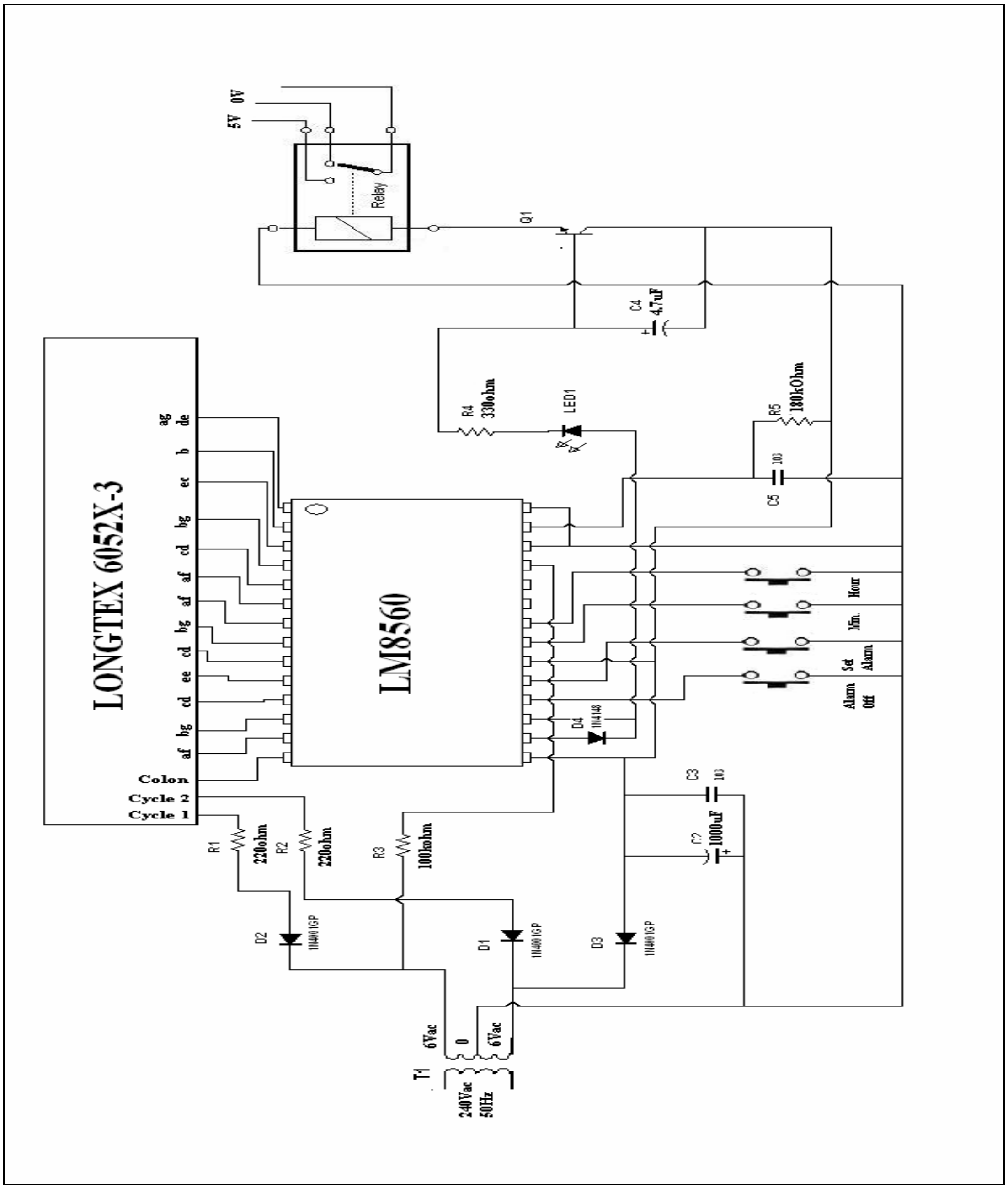


Figure 1 Digital clock

APPENDIX C

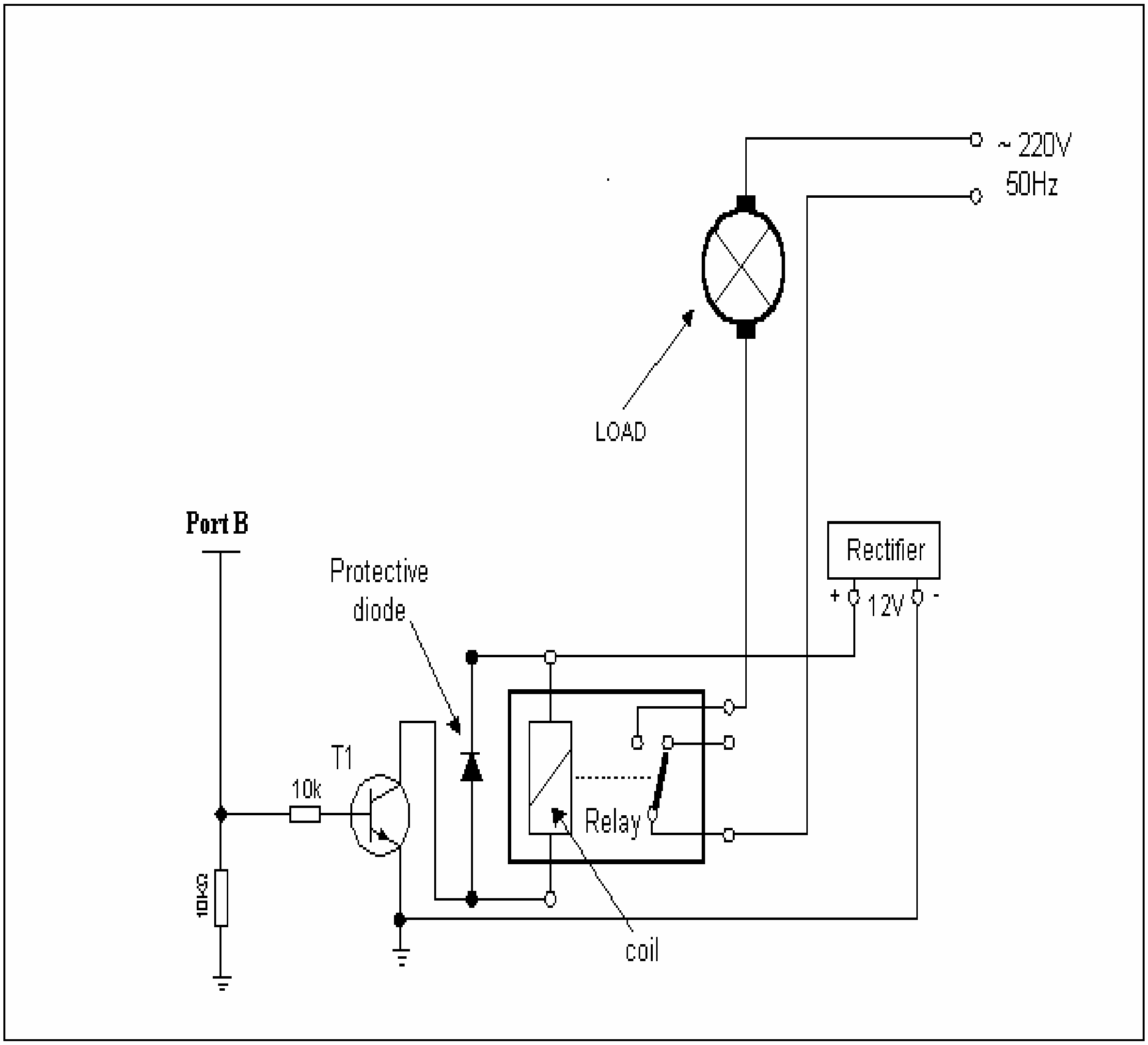


Figure 2 Interface circuit another type

APPENDIX D

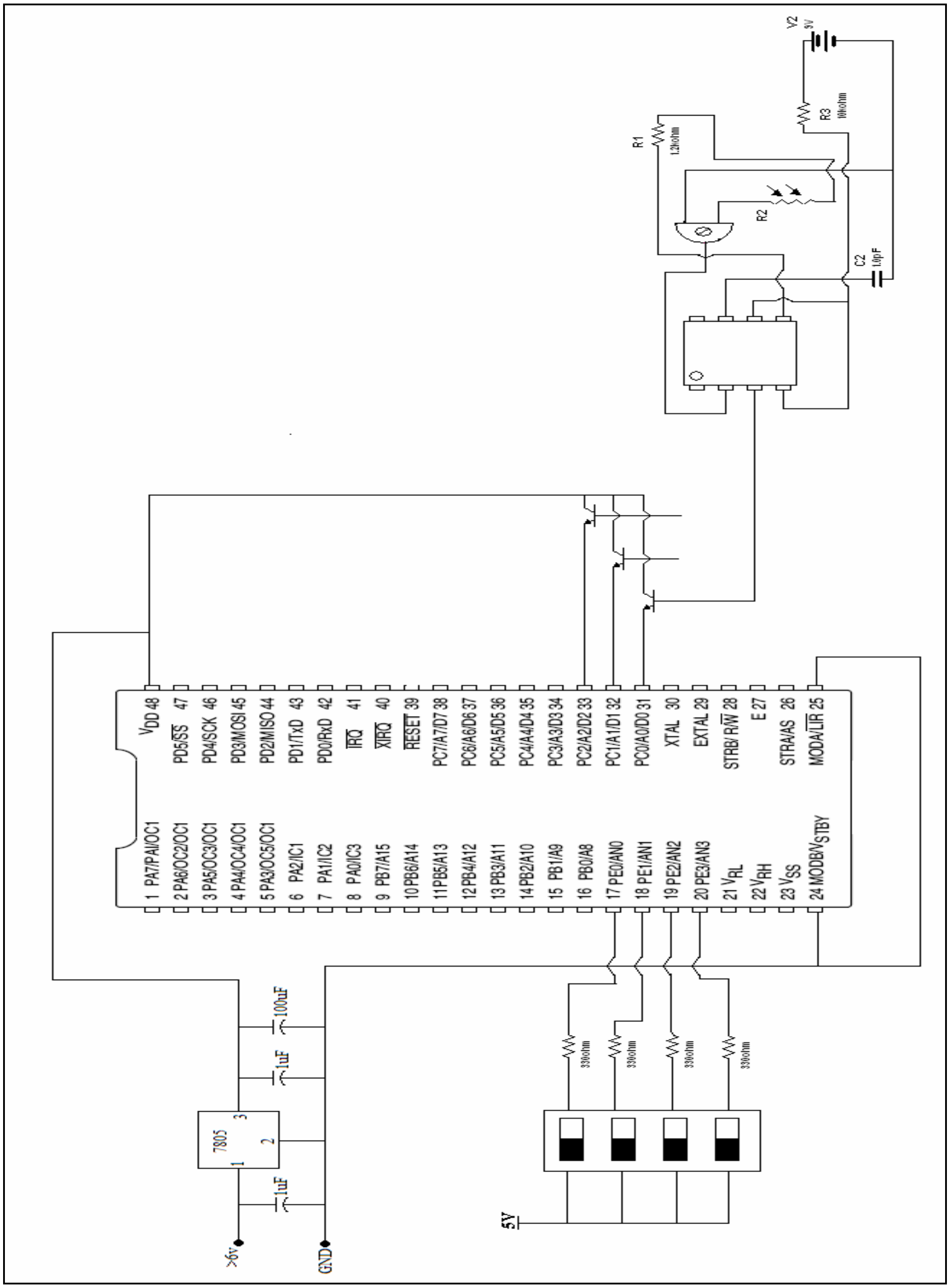


Figure 3 Interface between the light sensor to the microcontroller

APPENDIX E

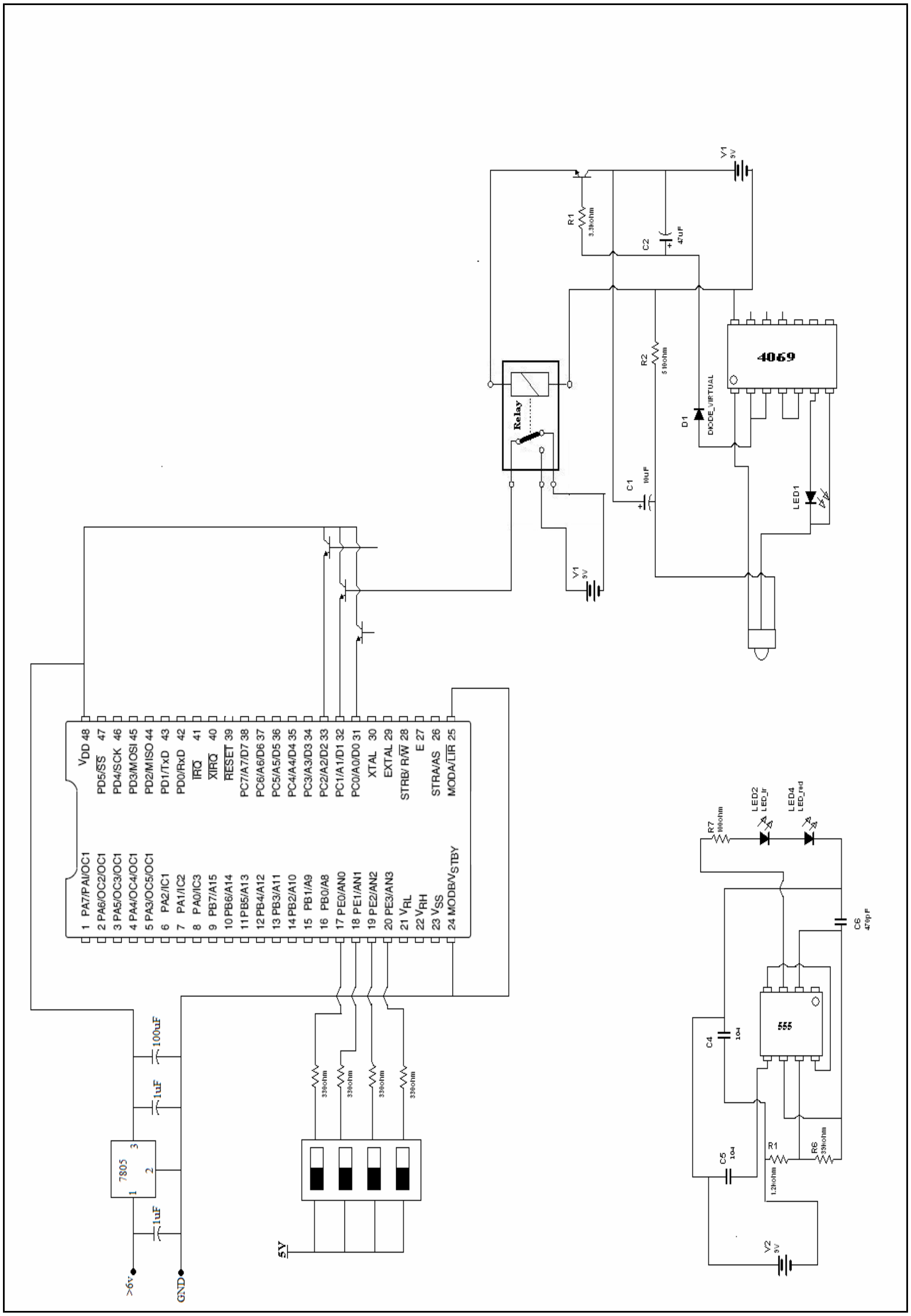


Figure 4 Interfacing the IR sensor to the microcontroller

APPENDIX F

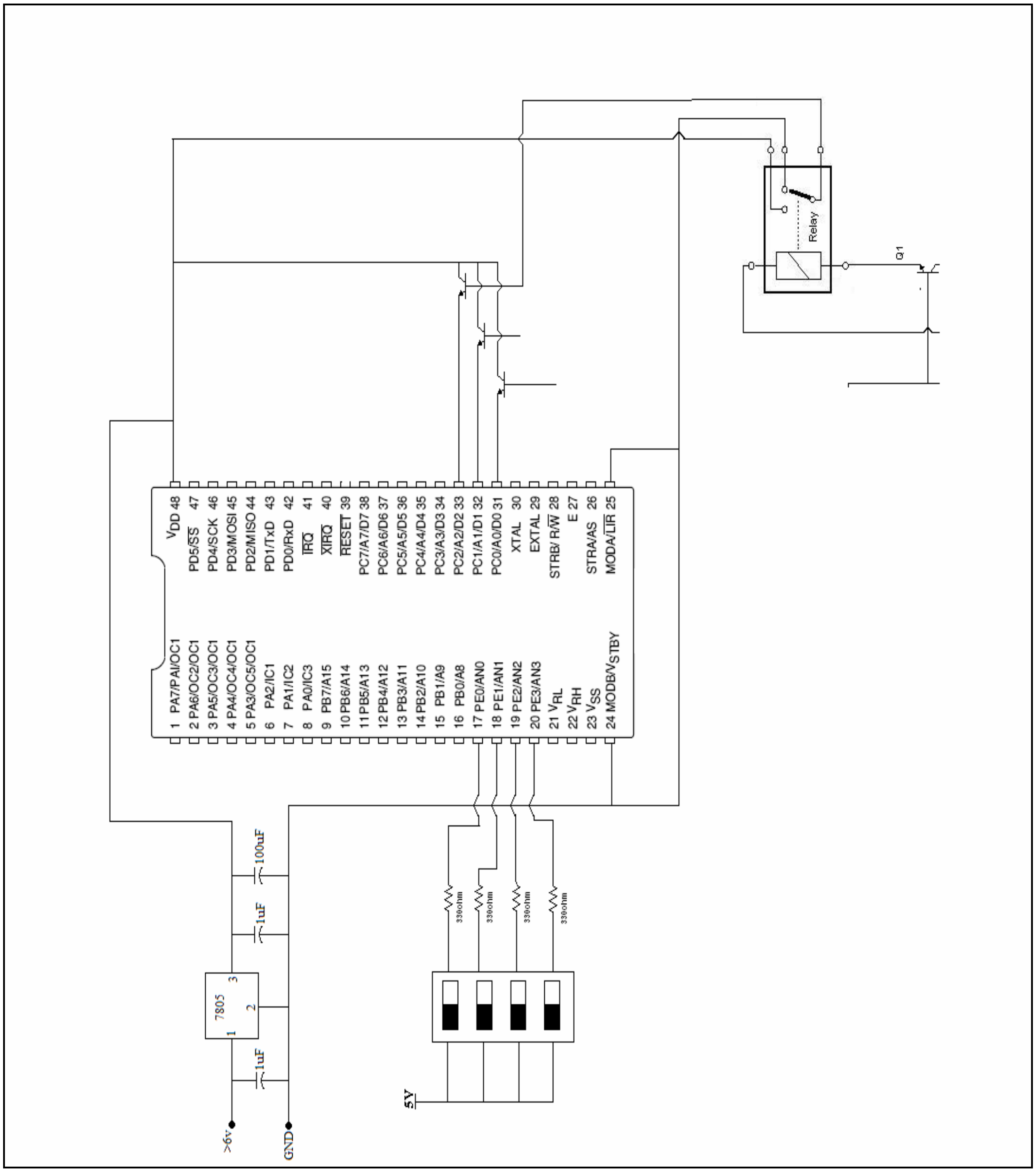


Figure 5 Digital clock and the microcontroller

APPENDIX G

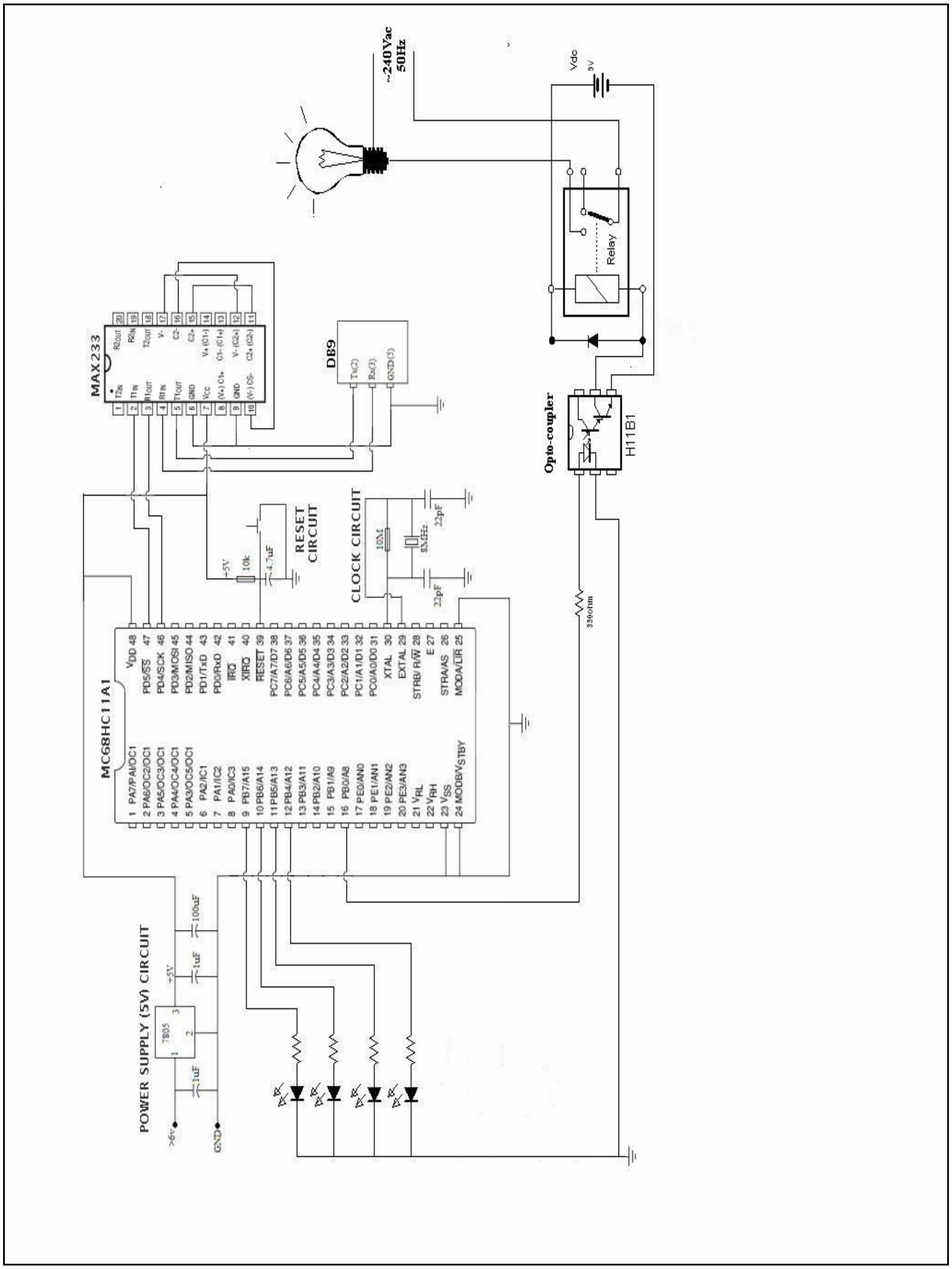


Figure 6 Microcontroller and MAX233

APPENDIX H

Products Described

WP11.EXE Windows Bootloader Program

68HC11 Devices Supported

MC68HC711E20
MC68HC711E9
MC68HC811E2
MC68HC711D3
MC68HC11A1/A8
MC68HC711K4

The Engineers Collaborative, Inc.
Web Site at http://www.tec-i.com
Email support@tec-i.com

TABLE OF CONTENTS

1.0 INTRODUCTION 4

2.0 SYSTEM REQUIREMENTS 4

3.0 DEVICES SUPPORTED 4

4.0 GETTING STARTED 4
- **4.1 Software Installation** 4

5.0 68HC11 SPECIAL BOOTSTRAP MODE 6

6.0 THE WP11.EXE PROGRAM 6
- **6.1 Program Overview** 6
- **6.2 WP11 Screen Shot** 7
  - **6.2.1 WP11 Status Panel** 7
  - **6.2.2 Changing the Config Register** 8
- **6.3 WP11 Menus** 8
  - **6.3.1 File Menu** 8
    - Load Buffer from File 8
    - Load Buffer from Chip 9
    - Save Buffer 9
    - Save Buffer As 9
    - Exit Program 9
  - **6.3.2 Edit Menu** 9
    - Clear Buffer 9
    - Fill Buffer 10
    - Verify Buffer Against Chip 10
    - Edit Buffer Contents 10
  - **6.3.3 Select Device Menu** 11
  - **6.3.4 Communications Menu** 12
    - Com Port Setup and Test 12
    - Com Port Troubleshooting Chart 13
  - **6.3.5 Options Menu** 14
    - File Format Options 14
    - Load Buffer On Startup 14
    - Turn Off Hint Boxes 14
    - Turn Off Status Line Hints 15
    - Turn Off Memory Map Checks 15
  - **6.3.6 Help Menu** 15
    - Contents 15
    - About 15
- **6.4 Chip Operation Buttons** 15
  - **Initialize Device** 15
  - **Communications Check** 16
  - **Blank Check** 16
  - **Erase EEPROM and Config Reg** 16
  - **Program Config Reg** 16
  - **Program EEPROM** 16
  - **Program EPROM** 17
  - **Program Entire Device** 17

7.0 TUTORIAL EXAMPLE SESSION 17

1.0 INTRODUCTION

This manual provides information on our WP11.EXE program, Windows software used for programming 68HC11 family microcontrollers.

WP11.EXE takes input from object code files created with 68HC11 assemblers or compilers. These object code files may be in Motorola S-record, Intel Hex or binary memory image formats. The software also allows the operator to perform many other useful programming functions such as erasing, blank checking & verifying devices, displaying, editing, exporting and changing the format of the object code files and filling unused memory locations with a user specified "fill" byte. The software was designed to work with our P11 Programming Board but will also work with any hardware that supports the 68HC11 special bootstrap mode of operation.

TECI provides several other development tools for 68HC11 family microcontrollers such as our Cross Assemblers WASM11.EXE and TASM11.EXE and real time in-circuit emulator TECICE-HC11.

2.0 SYSTEM REQUIREMENTS

WP11.EXE requires a minimally configured IBM PC or compatible with at least 64Meg RAM memory, one 3.5" disk drive, one free serial port designated as COM1-COM4 and Windows 3.1, 98, ME, NT, 2000 or XP.

3.0 DEVICES SUPPORTED

Presently WP11 supports the following 68HC11 family members:
MC68HC711E9
MC68HC711E20
MC68HC811E2
MC68HC711D3
MC68HC11A1/A8
MC68HC711K4

4.0 GETTING STARTED

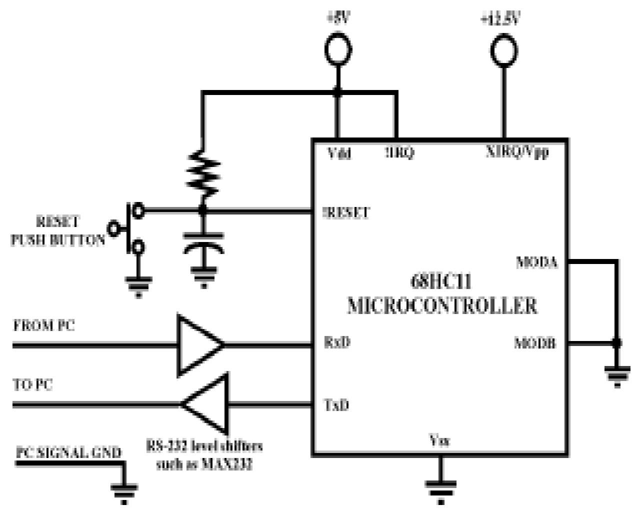
4.1 Software Installation
The software is contained in a single self-extracting .EXE file called wp11setup.exe. This is a standard windows install file and you need to be running windows for it to guide you through the install process. You may have downloaded the wp11setup.exe file from our web site at www.tec-i.com or received the file on a single 3.5" floppy diskette. To start the setup process you can either double click on the file name from windows explorer or use "RUN" from the windows start menu.
The files extracted from wp11setup.exe are:

`WP11.HLP`	This is the help file for WP11.EXE.
`WP11Man.PDF`	This manual in Adobe Acrobat .PDF format. The Adobe Acrobat reader software is available free of charge from a number of online sites such as: www.adobe.com/products/acrobat/readstep.html
`P11_A8.MIK`	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711A1/A8 devices. `WP11.EXE` downloads this code to the programmer when A1 or A8 devices are being programmed.
`P11_D3.MIK`	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711D3 devices. `WP11.EXE` downloads this code to the programmer when D3 devices are being programmed.
`P11_E2.MIK`	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC811E2 devices. `WP11.EXE` downloads this code to the programmer when E2 devices are being programmed.
`P11_E9.MIK`	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711E9 devices. `WP11.EXE` downloads this code to the programmer when E9 devices are being programmed.
`P11_E20.MIK`	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711E20 devices. `WP11.EXE` downloads this code to the programmer when E20 devices are being programmed.
`P11_K4.MIK`	This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711K4 devices. `WP11.EXE` downloads this code to the programmer when K4 devices are being programmed.
`E2_TEST.S19`	This is an object code file in Motorola S-Record format. It is used for test programming of MC68HC811E2 devices.
`K4EEPROM.S19`	This is an object code file in Motorola S-Record format. It is used for test programming of the EEPROM memory area of MC68HC711K4 devices.

`EEPROM.S19` This is an object code file in Motorola S-Record format. It is used for test programming of the EEPROM memory area of MC68HC711A1/A8, MC68HC711E9, and MC68HC711E20 devices. Memory locations from $B600 through $B7FF are programmed by this file.

5.0 68HC11 SPECIAL BOOTSTRAP MODE

WP11 uses the 68HC11 special bootstrap mode to program devices. You will need to build or buy the circuitry required to do this. The figure below shows the basics of this circuitry.



Special Bootstrap Circuit

Much more information on the special bootstrap mode can be found online by doing a Google search for "68HC11 special bootstrap mode".

6.0 THE WP11.EXE PROGRAM

WP11.EXE is a Windows program that is used to program 68HC11 microcontrollers by controlling hardware that supports the special bootstrap mode of these devices. WP11 uses a serial port to communicate with the programming hardware, which can be the P11 board, or any circuit that uses the HC11 special bootstrap mode.

WP11 is usually started from the "START" menu of your Windows PC or by double clicking on the WP11.EXE file name from within Windows Explorer.

6.1 Program Overview

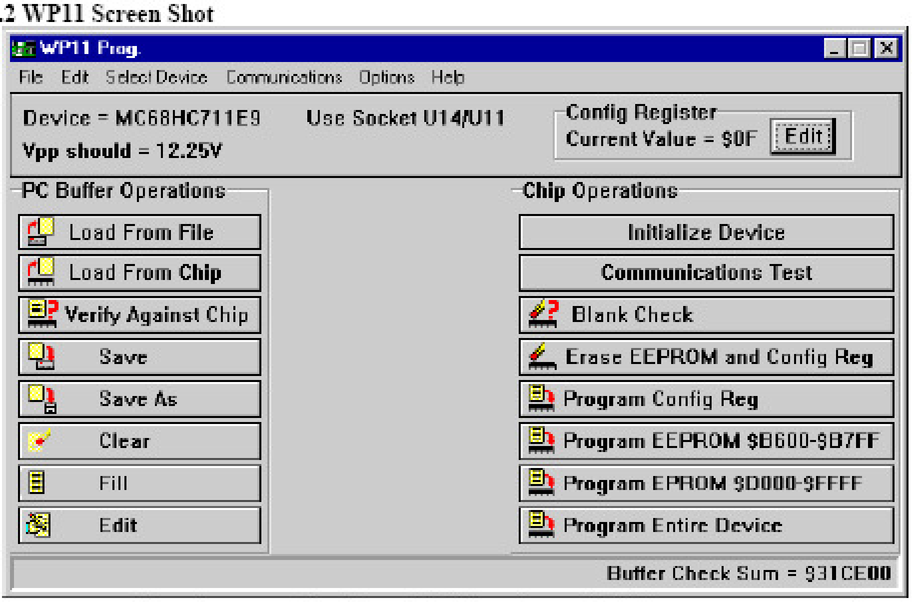
To program a device using WP11.EXE follow these steps:

1) Make sure that your PC serial port is properly set up to communicate with the hardware. Use the "Communications" menu item "Com Port Setup and Test" to verify proper operation of your serial port.
2) Select the desired HC11 family member using the "Select Device" menu.

3) Load the PC Buffer with the data that you want to place in the HC11 device. This data can come from a number of different sources, as we will see.
4) Make sure that the programming hardware is powered off and insert a device in the appropriate programming socket. Power up the programming hardware.
5) Click the "Initialize Device" button and follow the instructions.
6) Click on a programming button to perform the desired operation.

The rest of this section will explain the WP11 menu options.

6.2 WP11 Screen Shot



As you can see from the screen shot above WP11 has the usual Windows menu structure above a status panel. Below the status panel is a column of buttons on the left side, which are used to invoke various operations on the PC buffer. Remember, the PC buffer is memory inside your PC that contains the data that will be transferred to the HC11 chip during programming.
The column of buttons on the right side of the screen is used to invoke various chip programming operations.

6.2.1 WP11 Status Panel

The status panel is used to show information about the currently selected device. It gets updated when a different HC11 device is chosen from the Select Device menu.

6.2.2 Changing the CONFIG Register

The Status Panel contains a means of changing the CONFIG register value. When the 68HC11 family member is changed the CONFIG Register value (in the PC) automatically changes to the erased state of the new device. You may, however, wish to change the CONFIG Register to some other value. Pressing the "EDIT" button will enable you to specify a new value. You must be careful to specify values that are valid for the selected device. Changing the CONFIG Register will not affect the microcontroller until the device is programmed.

6.3 WP11 Menus

6.3.1 File Menu

Load Buffer from File is the same as ✍

Load Buffer from Chip is the same as ✍

Save Buffer is the same as ✍

Save Buffer As is the same as ✍

Load Buffer from File

Use this selection to load an object code file into the PC buffer.

When this option is selected, a standard Windows "Open" dialog box is displayed and the operator selects the desired object code file. File formats are automatically detected by WP11.EXE, and may be Motorola S-Record, Intel Hex or a binary memory image. Binary files must be 64K bytes in length. If the file name is entered correctly and can be opened and read by WP11.EXE, a memory image of the target chip is created in the buffer. If the file can't be opened, an error message is displayed and the operator is given another chance to enter the file name. This continues until the operator enters a correct file name or presses the "Cancel" button.

Load Buffer from Chip
Use this selection to load the contents of a programmed chip into the PC buffer.
A chip must be installed in the programming hardware and properly initialized for this menu option to be enabled. When this option is selected, a status Window with a Progress Gauge is displayed to keep the operator aware of the command progress. During the upload, the chip being read computes a checksum of the bytes sent to the PC. This checksum is sent to the PC where it is compared to a checksum of the received bytes computed by the PC. The data is placed in the buffer only if the correct number of bytes was received and the checksums match.

Save Buffer
This selection saves the contents of the buffer in a file on your PC. The file name used is the same as the currently opened file. If no file is currently opened a standard Windows "Save As" dialog box is opened to allow the operator to enter a file name. The file can be in one of three object code formats, Motorola S-Record, Intel HEX, or binary image format. Use the "Options" menu to select the format.

Save Buffer As
This selection is very nearly the same as the "Save Buffer" menu item described above except the "Save As" dialog box is opened immediately so that a new file name can be specified.

Exit Program
This selection ends WP11.EXE execution. If the buffer was changed since the last save the operator is given the option of saving it. A WP11.INI file is created and saved in the WP11.EXE directory and is used to restore program options and settings the next time the program is used. The WP11.INI file is a standard text file and may be edited to change program options if need be. If WP11.INI cannot be found, WP11.EXE starts up with default settings for the program.

6.3.2 Edit Menu

Clear Buffer is the same as

Fill Buffer is the same as

Verify Buffer Against Chip is the same as

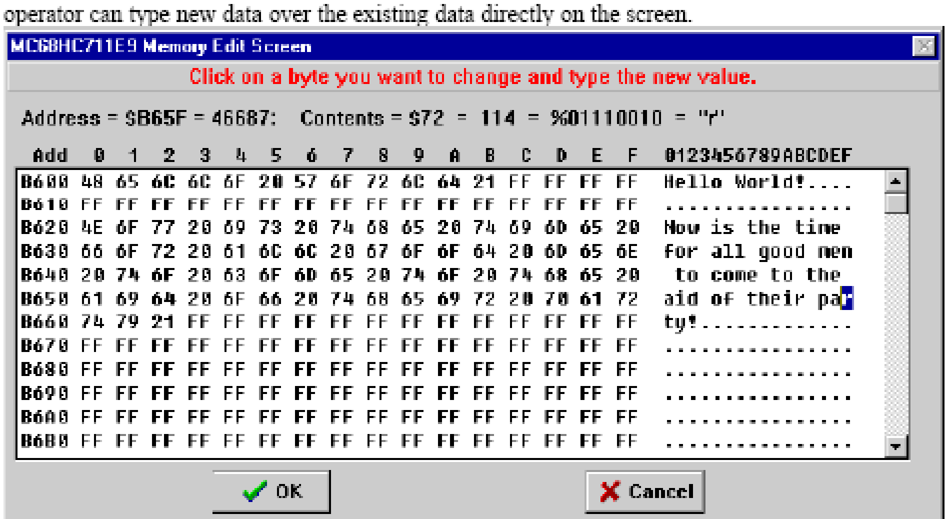
Edit Buffer Contents is the same as

Clear Buffer
This selection erases the data in the buffer in such a way as to leave each byte of the buffer the same as the corresponding byte in an erased chip. With presently supported devices each byte is returned to $FF. This function erases the current contents of the buffer.

Fill Buffer
This selection provides a means of filling the entire buffer with a byte of data that you specify. This function overwrites the current buffer contents. As an example of the use of this function consider this:
Suppose we want to have all the bytes in a device that are not taken up by our program set to a specific value. Say we want all unused memory locations filled with $3F, the op-code for the SWI instruction. To do this use the "Fill Buffer" function and specify $3F as the fill byte. Then use the "Load Buffer from File" function to load your program into the buffer. The buffer is not changed before the "Load Buffer from File" function is executed and the "Load Buffer from File" function does not change any bytes in the buffer that do not have data in the object code file. We started with the buffer completely filled with $3F and the "Load Buffer from File" function overwrote only those bytes that had data specified in the object code file so we end up with all of the bytes in the buffer that are not used by our program still set to $3F.

Verify Buffer Against Chip
This selection first uploads data from the chip similar to the "Load Buffer from Chip" function. The same dialog box and progress gauge is used during the upload. If uploading of the data is successful then each byte of the uploaded data is compared to the corresponding byte in the buffer. Data in the buffer is not changed. A pass/fail dialog box is displayed to end the function

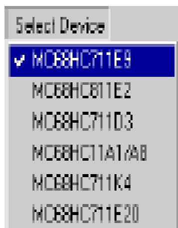
Edit Buffer Contents
When this function is selected data in the buffer is copied to a special edit area. The contents of this special edit area are then displayed on a Memory Edit Screen as shown below where the operator can type new data over the existing data directly on the screen.



Each line in the edit box above shows 16 bytes of the special edit area data in both hex and ASCII formats with the address of the first of the 16 bytes shown at the beginning of each line.

As an example, in the sixth line shown above, the 'r' in "party" is highlighted. We just clicked on the 'r' to highlight it. The status line at the top of the screen tells us that the 'r' is at address $B65F. It further tells us that $B65F = 46687 decimal and that the hex value of the 'r' is $72 which is 114 decimal which is 01110010 binary. If we wanted to change the 'r' we would simply type the new value, the screen would be updated, and the cursor would move to highlight the next character. We can also edit in the HEX display area by simply clicking on the character we want to change and typing the new value. It's all very easy. Try it!
Clicking the "OK" button copies the data in the special edit area back to the buffer thereby making the edits take effect. Clicking the "Cancel" button discards the data in the special edit area leaving the buffer unchanged.

6.3.3 Select Device Menu



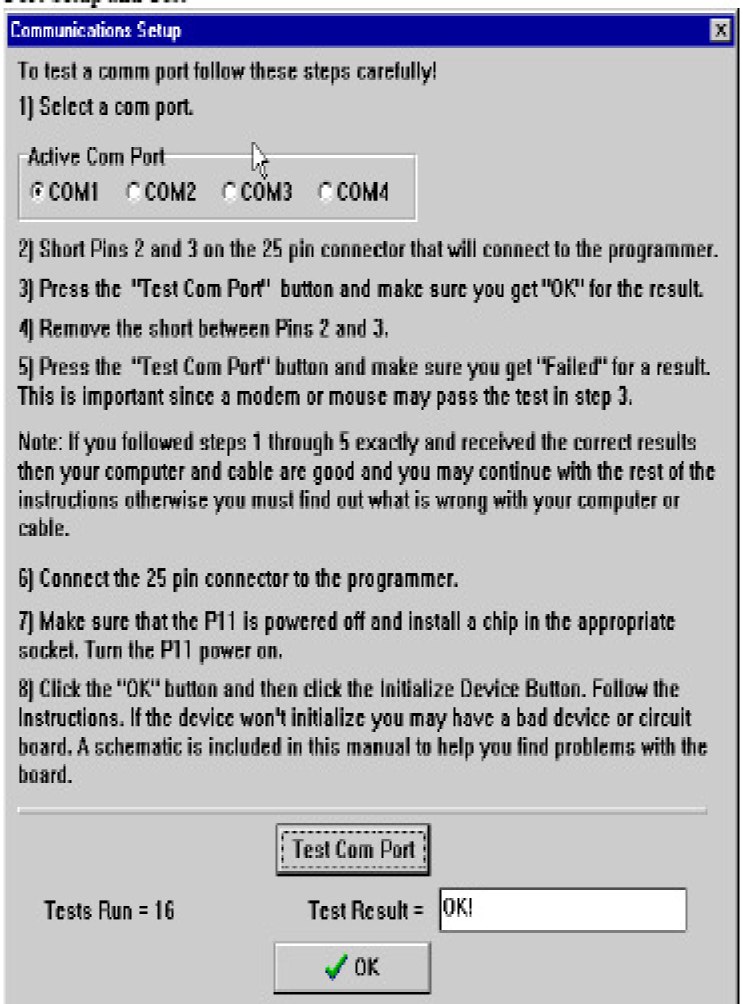
This menu is used to select the HC11 family member that you want to program. The part number is checked when it is the currently selected device. The part number is also displayed on the status panel with the "Device = XXXXXXXXXX" label.

6.3.4 Communications Menu

*** * * Important Note * * ***

The instructions on the Communications Setup screen below were written specifically for our P11 programmer that has a 25-pin serial port connector. Your hardware may have a different connector. If your hardware has a standard 9-pin serial port connector you would still short pins 2 and 3 where called for in the instructions. If your hardware has a non-standard serial port connector then you would short **RxD** to **TxD** where the instructions call for shorting pins 2 and 3. The WP11 help file has more information on PC serial port connections.

Com Port Setup and Test

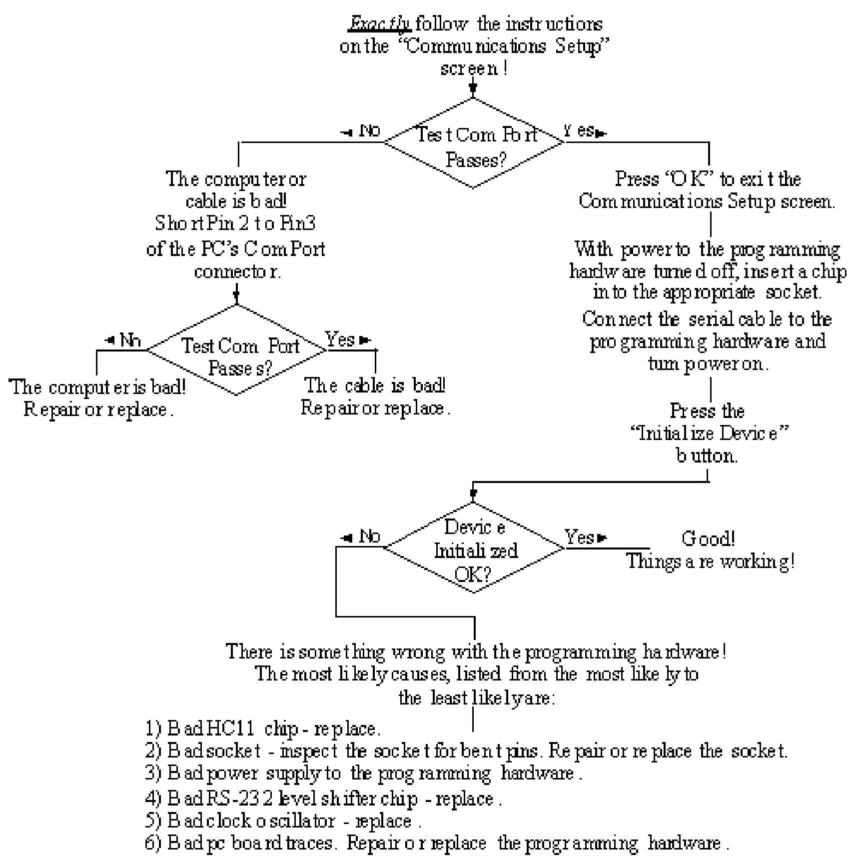


When the Com Port Setup and Test menu item is selected the Communications Setup screen shown above is displayed. The screen has detailed instructions and a test facility for determining whether or not a working serial port and cable are in use.

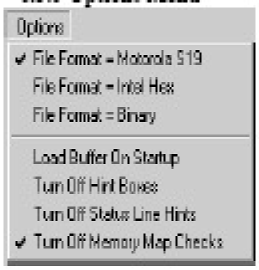
* * * Important Note * * *

In order to perform any of the chip operations the computer must be able to communicate with the programming hardware. Our experience indicates that the most difficult task faced by a first time user is to establish this communication. This has to do with the non-standard nature of the EIA RS-232 Standard and the many different RS-232 cables that exist. If communications aren't working, the ***very first*** thing to do is to determine if the cause lies with the computer and cable or with the programming hardware.
The test facility and instructions on this screen provide the tools required to determine if the problem is with the computer and cable or with the programming hardware.

Com Port Troubleshooting Chart



6.3.5 Options Menu



File Format Options

The first three Option Menu choices allow the user to specify the format to use when saving the buffer in an object code file. This format will be used when the "Save" or "Save As" buttons are pressed.

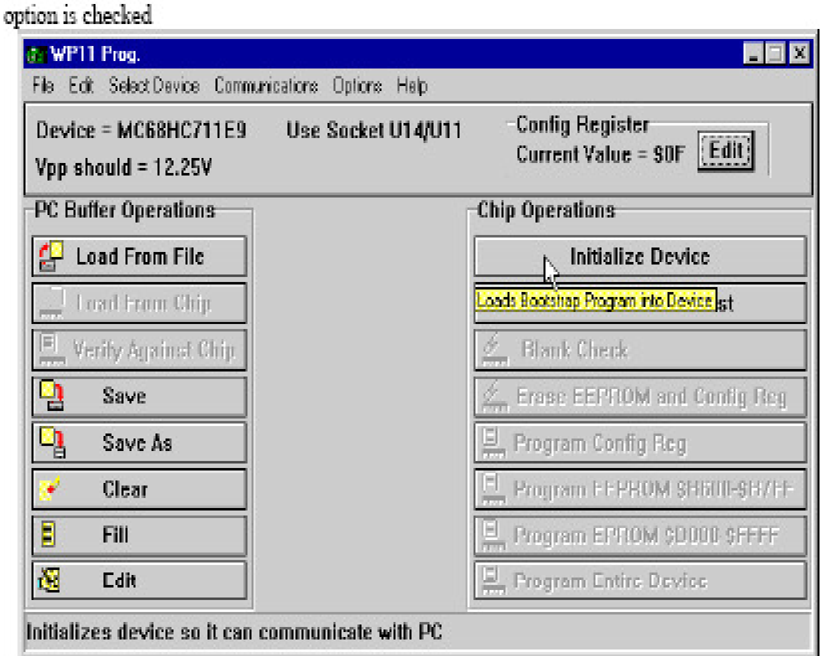
Note: When loading a file into the buffer the file format is automatically detected so these options come into play only when saving the buffer contents to a file.

Load Buffer On Startup

WP11 remembers the file name of the object code file loaded or saved in an ".INI" file. When the "Load Buffer On Startup" option is checked WP11 automatically loads this file back into the buffer.

Turn Off Hint Boxes

Referring to the screen shot below notice that the cursor is located over the "Initialize Device" button. Also notice that just below the cursor is a box containing the text "Loads Bootstrap Program into Device". This box is called a "hint box" and tells the operator what will happen when the button is pressed. These fly over hint boxes are probably not needed once a user is familiar with the program. The hint boxes will not be displayed when the "Turn off Hint Boxes" option is checked



Turn Off Status Line Hints

Referring to the screen shot above notice that the cursor is located over the "Initialize Device" button and that the panel at the bottom of the WP11 window has the text "Initializes device so it can communicate with PC". This text is called a "help hint" and gives the user more information about what will happen when the "Initialize Device" button is pressed. These fly over help hints are probably not needed once a user is familiar with the program. The help hints will not be displayed when the "Turn off Status Line Hints" option is checked.

Turn Off Memory Map Checks

WP11 knows the memory maps of supported family members. That is to say that it knows where in memory the EEPROM, EPROM, CONFIG Register and vectors are located. It knows which memory locations are valid for a particular family member and which ones are invalid. Normally, when WP11 is loading the buffer with data in an object code file it checks each byte to verify that it is going into a valid memory location for the currently selected device. If errors are detected they are reported to the operator. Generally, this is just what we want because if, by some mistake, the program doesn't fit in the device memory map the program won't run properly. But, what if all we want to do is load an object code file into the buffer, perhaps edit it, and then save it in a different file or file format? Having this normally nice feature turned on might prevent us from doing this simple thing. To solve this problem, check the "Turn Off Memory Map Checks" option.

6.3.6 Help Menu

Contents

Selecting "Help – Contents" opens the WP11 help file.

About

Selecting "Help – About" opens the WP11 About screen where information about the program version number and our company contact information can be found.

6.4 Chip Operation Buttons

Initialize Device

WP11 uses the HC11 special bootstrap mode to program devices. This requires that a small program be downloaded to the target device RAM for the purpose of controlling the programming process and communicating with the PC. The "Initialize Device" button performs this task. No programming operation on a chip can be done without initializing the chip. This is why all menu items and buttons that talk to the programming hardware are disabled until the chip is initialized.

The small programs that "Initialize Device" downloads to the target RAM ***must*** be in the same directory as the WP11.EXE program so that they can be found when needed. The programs are different for each family member and have names such as "P11_E9.MIK" for the

MC68HC711E9 chip and "P11_K4.MIK" for the MC68HC711K4 chip. If WP11 can't find these files an error message is generated.

When the "Initialize Device" button is pressed a dialog box is displayed that has a progress gauge to show the status of the operation. If the operation is successful the dialog box just goes away and the appropriate menu items and buttons are enabled. If the operation is not successful an error message is generated.

Normally this initialization process has to be performed only once when a target device is inserted into the programming hardware. It does *not* have to be repeated unless power is turned off to the target device or the target device is reset.

Communications Check

Communications Test

This button is used to determine if the target device is properly initialized. The PC sends $00 to the programming hardware and expects to receive $AA followed by $55. If the expected response is received then the appropriate menu items and buttons are enabled else they are disabled.

The "Communications Test" button can be used to enable the chip operations buttons when you know that the target device is properly initialized but the button are not enabled such as would be the case if you exited and re-entered the program. Of course, you could use the "Initialize Device" to do the same thing but it would take much longer.

Blank Check

Blank Check

Press this button to test the target device's CONFIG register, EPROM, and EEPROM memory to determine if every location is in the erased state.

Erase EEPROM and Config Reg

Erase EEPROM and Config Reg

This button returns the EEPROM and CONFIG register in the target device to its erased state.

Program CONFIG Reg

Program Config Reg

Pressing this button programs the value shown for the CONFIG register in the WP11 status panel into the CONFIG register of the target device then tests to determine if the operation was successful. This button will not be enabled if the currently selected device does not have a CONFIG register.

Program EEPROM

Program EEPROM $B600-$B7FF

Pressing this button programs the contents of the EEPROM portion of the WP11 buffer into the EEPROM memory of the target device then tests to determine if the operation was successful.

Program EPROM

Program EPROM $D000-$FFFF

Pressing this button programs the contents of the EPROM portion of the WP11 buffer into the EPROM memory of the target device then tests to determine if the operation was successful.

Program Entire Device

Program Entire Device

Pressing this button programs the contents of the EEPROM & EPROM portion of the WP11 buffer and the value shown for the CONFIG register on the WP11 status panel into the target device then tests to determine if the operation was successful.

7.0 TUTORIAL EXAMPLE SESSION

Working through this tutorial will take just a short time and will provide you with an overview of WP11 by actually using it.

It is assumed that you have read the rest of this manual, installed the software in accordance with the *Getting Started* section and are certain that your hardware and serial port connection are working properly. It is further assumed that you have a 68HC11 chip that can be used with this tutorial. We will only program the EEPROM memory so the chip can be erased and re-used for another purpose.

Three demonstration/test 68HC11 programs were included for use with this tutorial. If your chip is a MC68HC711K4 use K4EEPROM.S19 as the test file. If your chip is a MC68HC811E2 use E2_TEST.S19 as the test file. Use EEPROM.S19 as the test file for all other chips.

The steps required to program a chip are as follows:

1) Select the desired device type from the *Select Device* menu.
2) Insert the target device into the proper socket of your programming hardware. **NOTE:** You must insert and remove devices from the ZIF sockets with power removed from the programming hardware.
3) Load the *Buffer* with the data that you want to program into the target device. This can be done in a number of ways, from a file, from another chip, manually etc.
4) Initialize the target device for programming.
5) Program the target device using one of the available *chip operations* buttons.

Tutorial:

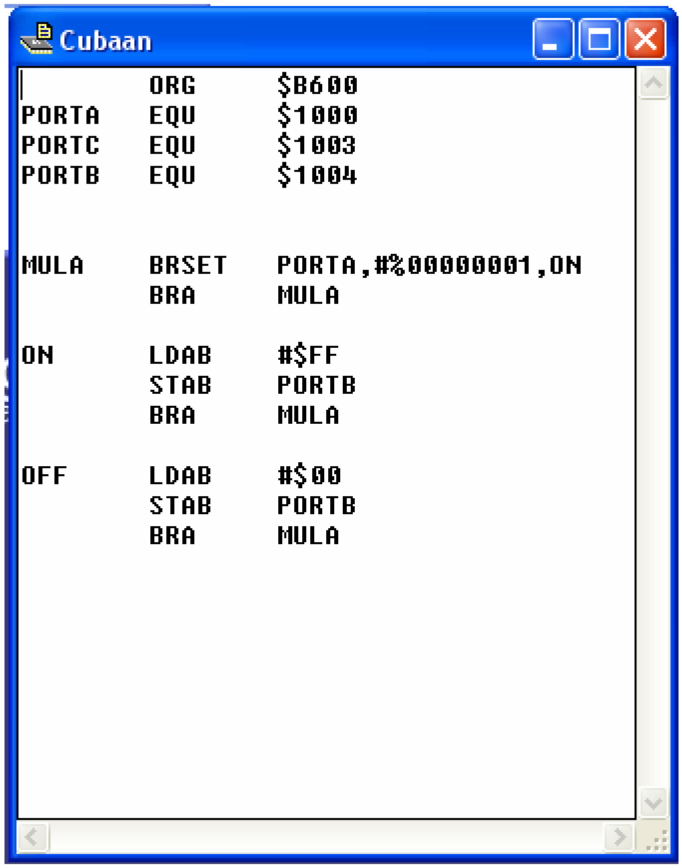
1) Load the Buffer with the desired data to be programmed into the target device by pressing the *Load From File* button on the WP11 main screen. From the Open dialog box open the test file for your chip as determined above.
2) Press the *Edit* button and use the scroll bars to view the data. Exit the screen by pressing the *Cancel* button.
3) Place the target device in the programming socket. **NOTE:** You must insert and remove devices from the programming socket with power removed from the programming board.

4) Press the *Initialize Device* button. This downloads a small program to the target device that controls the programming process.
5) Press the *Program & Verify Device* button if using an E2 device or *Program & Verify EEPROM* button if using any other device. This programs the contents of the *Buffer* into the target device.
6) To prove that we can read the contents of a programmed device, press the *Clear* button to clear the buffer, verify this with the edit button if you are the untrusting type, then press the *Load From Chip* button. Then press the *Edit* button. You should see the data that was originally programmed into the device.

That's it! You have just performed all of the necessary steps to program an HC11 device.

We sincerely hope that after using this tool, you will consider it to be a sound investment. We look forward to your comments and suggestions and to providing any assistance with their use that you may require.

APPENDIX I




```
        ORG     $B600
PORTA   EQU     $1000
PORTC   EQU     $1003
PORTB   EQU     $1004


MULA    BRSET   PORTA,#%00000001,ON
        BRA     MULA

ON      LDAB    #$FF
        STAB    PORTB
        BRA     MULA

OFF     LDAB    #$00
        STAB    PORTB
        BRA     MULA
```

Figure 7 Programmed for light sensor

APPENDIX J

IISEN2

```
DDRA     EQU      $1800
PORTA    EQU      $0
REGS     EQU      $1000
PORTC    EQU      $03
DDRC     EQU      $07
PORTB    EQU      $04

         ORG      $B600
         LDS      #$FF
         STAA     DDRA
         LDX      #REGS

CHECK    BRSET    PORTA,X,#%00000001,SWITCH1
         BCLR     PORTB,X,#%11111111
         BRA      CHECK

SWITCH1  BSET     PORTB,X,#$FF
         BSR      DELAY
SCAN     CLR      PORTB,X
         BRSET    PORTA,X,#%00000011,SWITCH2
         BRCLR    PORTA,X,#%00000001,CHECK
         BRA      SCAN

SWITCH2  BSET     PORTB,X,#$FF
         BSR      DELAY
         BRCLR    PORTA,X,#%00000001,CHECK
         BRA      SCAN

DELAY    PSHX
         LDAA     #5
RETURN   LDY      #$FFFF
AGAIN    DEY
         BNE      AGAIN
         DECA
         CMPA     #0
         BEQ      RETURN
         PULX
         RTS
```

Figure 8 Programmed for infra-red sensor.

APPENDIX K

Final3

```
PORTA   EQU     $0
REGS    EQU     $1000
PORTB   EQU     $04
PORTE   EQU     $0A
PORTC   EQU     $03
DDRC    EQU     $07

        ORG     $B600
        LDS     #$FF
        LDX     #REGS
        CLR     PORTB,X
        LDAA    #$00
        STAA    DDRC,X
        BRA     MAIN

SWITCH  BRSET   PORTC,X,%00000001,ON
        BRA     MAIN

ON      LDAA    #$FF
LOOP    STAA    PORTB,X
        BRSET   PORTC,X,%00000101,MID
        BRCLR   PORTC,X,%00000001,DAY
        BRA     MAIN

MID     LDAA    #$00
LOOP2   STAA    PORTB,X
        BRSET   PORTC,X,%00000111,IRED
        BRSET   PORTE,X,%00000001,MAIN
        BRSET   PORTE,X,%00000010,MAIN
        BRA     LOOP2

IRED    LDAA    #$FF
        STAA    PORTB,X
        BSR     DELAY
        BRA     MID

DAY     LDAA    #$00
        STAA    PORTB,X
```

Figure 9 Final programmed for the system

Final3

```
MAIN     LDAA    PORTE,X
         ANDA    #$0F
         BEQ     SWITCH
         CMPA    #1
         BEQ     SWITCH1
         CMPA    #2
         BEQ     SWITCH2
         CMPA    #4
         BEQ     SWITCH3
         BRA     MAIN

DELAY    PSHX
         LDAA    #5
RETURN   LDY     #$03FF
AGAIN    DEY
         BNE     AGAIN
         DECA
         CMPA    #0
         BEQ     RETURN
         PULX
         RTS

SWITCH1  LDAA    #$FF
LAGI     STAA    PORTB,X
         LDAB    PORTE,X
         ANDB    #$01
         CMPB    #$01
         BEQ     LAGI
         LDAA    #$00
         STAA    PORTB,X
         LDAB    PORTE,X
         CMPB    #$1
         BNE     MAIN
         BRA     LAGI

OFF      CLR     PORTB,X
         BRA     MAIN
```

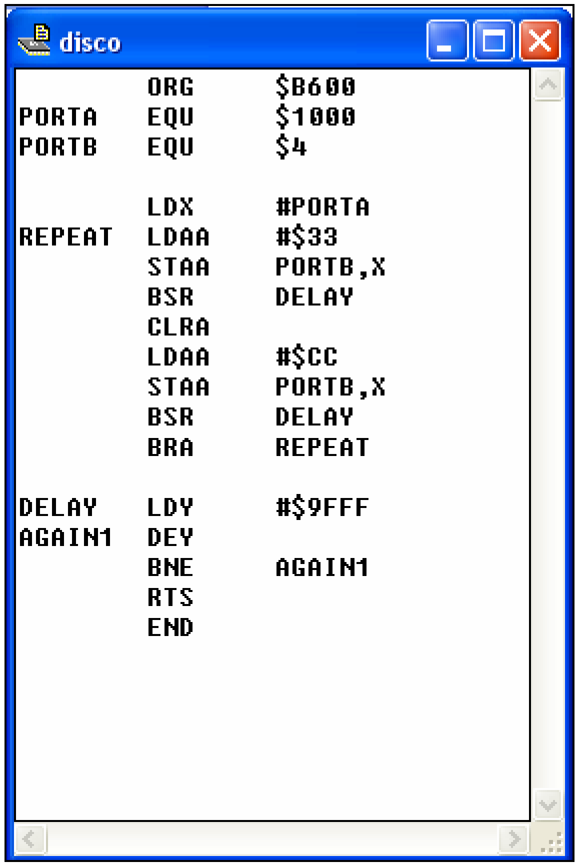
Figure 9 Final programmed for the system.(Continued)

Final3

```
        BNE     MAIN
        BRA     LAGI

OFF     CLR     PORTB,X
        BRA     MAIN

SWITCH2 LDAA    PORTC,X
        ANDA    #$01
        CMPA    #$01
        BEQ     ON2
        LDAB    PORTE,X
        ANDB    #$02
        CMPB    #$2
        BNE     MAIN
        BRA     SWITCH2

ON2     LDAA    #$FF
        STAA    PORTB,X
        BRCLR   PORTC,X,%00000001,OFF
        LDAB    PORTE,X
        ANDB    #$02
        CMPB    #$2
        BNE     MAIN
        BRA     ON2

SWITCH3 LDAA    #$00
        STAA    PORTB,X
        BRSET   PORTC,X,%00000011,IRED2
        BRCLR   PORTC,X,%00000001,OFF
        BRA     SWITCH3

IRED2   LDAA    #$FF
        STAA    PORTB,X
        BSR     DELAY
        LDAA    #$00
        STAA    PORTB,X
        BRCLR   PORTC,X,%00000001,OFF
        JMP     MAIN
```

Figure 9 Final programmed for the system (Continued)

APPENDIX L



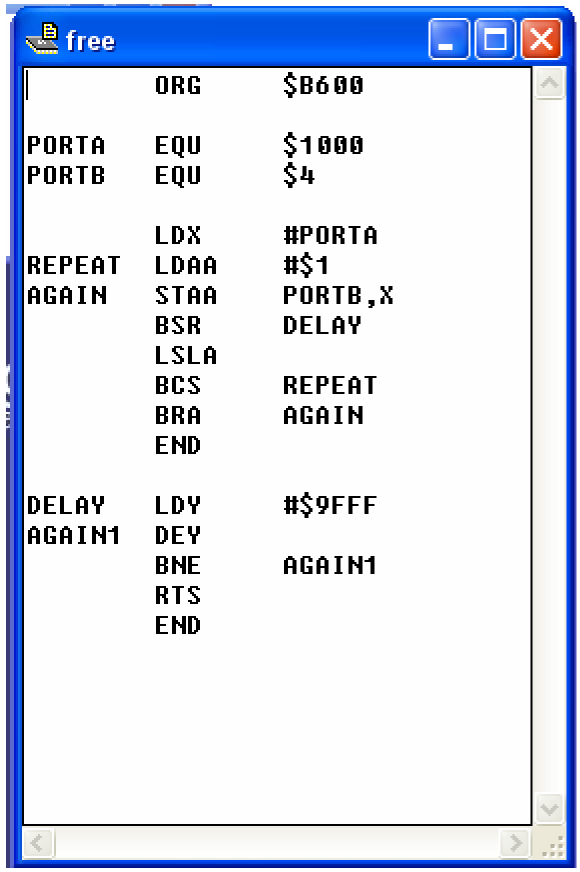

```
        ORG     $B600
PORTA   EQU     $1000
PORTB   EQU     $4

        LDX     #PORTA
REPEAT  LDAA    #$33
        STAA    PORTB,X
        BSR     DELAY
        CLRA
        LDAA    #$CC
        STAA    PORTB,X
        BSR     DELAY
        BRA     REPEAT

DELAY   LDY     #$9FFF
AGAIN1  DEY
        BNE     AGAIN1
        RTS
        END
```

Figure 10 Test programmed 1(disco lights)

flash

```
        ORG     $B600
PORTA   EQU     $1000
PORTB   EQU     $4

        LDX     #PORTA
REPEAT  LDAA    #$FF
        STAA    PORTB,X
        BSR     DELAY
        CLRA
        LDAA    #$00
        STAA    PORTB,X
        BSR     DELAY
        BRA     REPEAT

DELAY   LDY     #$9FFF
AGAIN1  DEY
        BNE     AGAIN1
        RTS
        END
```

Figure 11 Test programmed 2 (Flashing)



free

```
          ORG       $B600

PORTA     EQU       $1000
PORTB     EQU       $4

          LDX       #PORTA
REPEAT    LDAA      #$1
AGAIN     STAA      PORTB,X
          BSR       DELAY
          LSLA
          BCS       REPEAT
          BRA       AGAIN
          END

DELAY     LDY       #$9FFF
AGAIN1    DEY
          BNE       AGAIN1
          RTS
          END
```

Figure 12 Test programmed 3(free running)

knight

```
        ORG     $B600
PORTA   EQU     $1000
PORTB   EQU     $4

        LDX     #PORTA
REPEAT  LDAA    #$81
        STAA    PORTB,X
        BSR     DELAY
        CLRA
        LDAA    #$42
        STAA    PORTB,X
        BSR     DELAY
        LDAA    #$24
        STAA    PORTB,X
        BSR     DELAY
        CLRA
        LDAA    #$18
        STAA    PORTB,X
        BSR     DELAY
        CLRA
        BRA     REPEAT

DELAY   LDY     #$9FFF
AGAIN1  DEY
        BNE     AGAIN1
        RTS
        END
```

Figure 13 Test programmed 4(Knight rider)

APPENDIX M

sensor

```
         ORG      $B600
PORTA    EQU      $0
REGS     EQU      $1000
PORTC    EQU      $03
DDRC     EQU      $07
PORTB    EQU      $04


         LDX      #$1000

CHECK    BRCLR    PORTC,X,#%00000001,SWITCH1      ;CHECK BUTTON
         BRCLR    PORTC,X,#%10000000,SWITCH2      ;CHECK BUTTON
         BRA      CHECK

SWITCH1  BSET     PORTB,X,#%01010101              ;PB3 ON
                  BSR      DELAY
                  BRA      CHECK

SWITCH2  BSET     PORTB,X,#%10101010              ;PB4 ON
                  BSR      DELAY
                  BRA      CHECK

DELAY    PSHX
         PSHY
         LDY      #100
LOOPY    LDX      #3332
LOOP1    DEX
         BNE      LOOP1
         DEY
         BNE      LOOPY
         PULY
         PULX
         RTS
         END
```

Figure 14 Test programmed for switching

APPENDIX N

REFERENCES

1. Peter Spasov, Mircocontroller Technology 5th edition, Prentice Hall.

2. http://en.wikipedia .org/wiki/security_lightting.

3. http://www.residential-landscape-lighting-design.com/security_flood_light_fixtures.htm

4. http://www.aboutlightingcontrols.org/education/papers/introduction.shtml