X-10 Based Outdoor Light Automation System

by

Sapar Annayev

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information and Communication Technology)

November 2007

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

X-10 Based Outdoor Light Automation System

by

Sapar Annayev

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information Communication Technology)

November 2007

Approved By,

Mr. Low Tan Jung

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specifies in the references and acknowledgments, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

November 10, 2007

Sapar Annayev

ABSTRACT

For "X-10 based Outdoor Light Automation" system the powerline is used as a medium of signal transmitting. There are two parts of the system: Receiver and Sender. Both parts should be plugged in to the sockets anywhere in the house, and each of them has a PIC16F877 to send, generate, receive signals and execute the appropriate instruction. It is already being widely practiced and used in the North America where they use 110V (60Hz), while the powerline standard in South East Asia is 240V (50Hz). The modular methodology is used to develop the system. This paper describes the process of developing "X-10 based Outdoor Light Automation" system using the 240V (60Hz) powerline as a medium of signal transmission and 5bits of data will be tested to switch on/off the device rather than 11bits that are used by existing X-10 communication protocol. Fewer bits in a message means that the transmission will be faster, fewer appliances will be controlled and less control functions would be used.

ACKNOWLEDGEMENT

First of all author would like to express deepest gratitude to Allah (c.c) for giving opportunity to live, for the mercifulness, support and guidance throughout Final Year Project.

The accomplishment of the Final Year Project "X-10 Based Home Automation System" is truly due to the support, help, cooperation and commitment of the people around, thanks for any commitment and any contribution they have done to this project.

Author would also like to express the highest gratitude to the project supervisor, Mr. Low Tan Jung for his great support and help throughout the whole project research and development period. The author also would like to express sincere thanks to the course mates and friends, who contributed by advising, motivating and supporting during both of the semesters.

The special appreciation goes to the author's family and relatives for their love, support and encouragement and just for being beside!

Thank You.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Home Automation was all the time in the center of attention of the people, as everyone strives to live as easy as possible, to make life as simple as possible. With the latest technology advancements humans have a variety of ways to control equipment or any appliance remotely. One of the ways is to transmit the signal using the X-10 communication protocol.

X-10 based Home Automation systems were first used in 1970's, and became one of the most popular and widely used protocols in the home automation industry of North America. Its identity is the use of the power line as a transmission medium where the signal is sent during the extremely short period of zero-crossing of the power line's signal. On the other side, the receiver decides on the action based on the combination of the bits received.

1.2 Problem Statement

Home automation depends on communication protocols which are created to control the various electrical and electronic systems in your house. There are four main home automation players out there, each a little different, and they're not all compatible with each other.

X-10 is a well-established home automation technology which is more than 30 years old. With X-10, there's no need for new wiring because it transmits signals over existing power line. However, this can make it susceptible to interference.

The "X-10 based Outdoor Light Automation" system that is to be developed is briefly described here. As was mentioned above, there will be Sender and Receiver parts, and both of them will have a 16F877 PIC built in. On the Sender part, the light sensor will serve as an input, and send the bits to Receiver to either switch ON or OFF the lights.

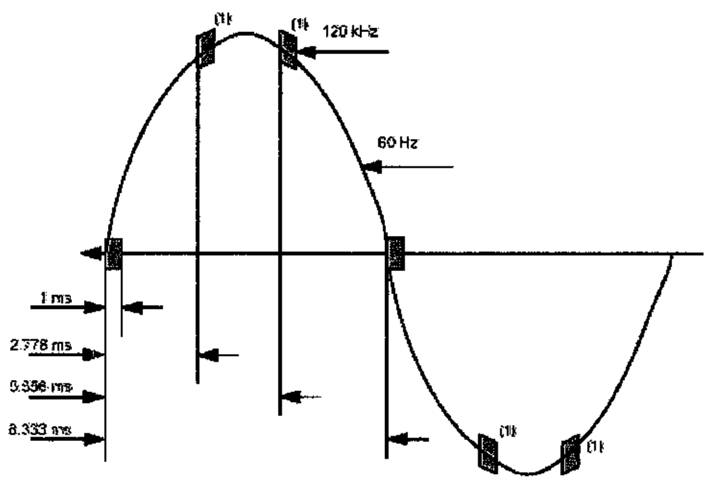


Figure 1.1: 120 KHz bursts on 60 Hz signal

The bits are transmitted in a batch of 11 bits (X-10 Communication Protocol), two times (for error checking). The PIC on the "Sender" side will generate a signal of 120 KHz, and transmit it during the Zero-Crossing interval.

1.3 Objectives and Scope of Study

1.2.1 Objectives

The objectives of the project are as follows:

a) To build the X-10 based product applicable to Malaysian Standard.
b) To make 5 bit data sufficient to switch ON/OFF the device.
c) To build the flexible X-10 based System.

1.2.2 Scope of Study

The project scope includes the following:

1) To carry out study on the Home Automation and X-10 communication protocol.
2) To observe the pros and cons of the X-10 based Automation
3) To develop and build both the Sender and Receiver for X-10 based Automation.

1.2.3 Feasibility of the Project within the Scope and Time frame

Due to lack of experience and hazard in developing and implementing the mains coupler, the System Integration and Testing task is left for the future enhancements. Nevertheless the theoretical research, development and testing of the rest of the modules is performed and is explained in this report.

The scope and Gantt chart (Please refer to Appendix J) of this project has been deliberately arranged in order to fulfill the requirements throughout two semesters of the Final Year Project. In case any problem faced, the supervisor was seeked for advices.

CHAPTER 2

LITERATURE REVIEW

2.1 Powerlines in home control systems

In this chapter, I will focus on suitability of the X-10 for Home Automation and Control. Home Control System (HCS) is an integration of the following technologies: home networking, smart appliances, the internet, and mobile. Home networking is the collection of elements that process, manage transporting and store information, enabling the connection and integration of multiple computing, control, monitoring and communication devices in the home. Home networking, in turn, has been enabled by the emergence of new trends such as broadband access, telecommuting, multi-PC households, remote home security services, remote home energy services, and even remote assistive solutions for disabled people. *(Chunduru & Subramanian, 2007)*

Smart appliances is a relatively newer development and several major appliance manufacturers (Toshiba, Samsung, LG, and Carrier) are developing internet-ready appliances such as stoves, refrigerators, washers, dryers, and the microwave, so that these smart appliances may be directly plugged-into the home network.*(Chunduru & Subramanian, 2007)* Once these smart appliances are plugged-in, they become another element in the home network and may be controlled via the controller, either from outside or the inside of the home. Internet has really helped propel the ability of remote control facility of the HCS.

Therefore, HCS provides unprecedented level of control to the home owner and as a result may increase the quality of his or her life. The distinguishing features of the data sent over the HCS system are the following:

1. Short bursts of control commands from the controller
2. Short bursts of response commands from the appliance or equipment
3. Typically several nodes connected to the system, where the node refers to a controller, appliance, or equipment
4. Typically long average distance usually measured in tens of feet
5. Occasionally large data transmissions

2.2 X-10 for Home Automation and Control

One of the technologies widely used for HCS is X10 protocol which is used for data transmission. The X10 protocol is perhaps the oldest standard for home networking. It was introduced in 1978 for the Sears home control system and the Radio Shack plug'n'power system. *(Chunduru & Subramanian, 2007)*

X10 communicates between transmitter and receiver by sending and receiving signals over the power line wiring. These signals involve short RF bursts which represent digital information.

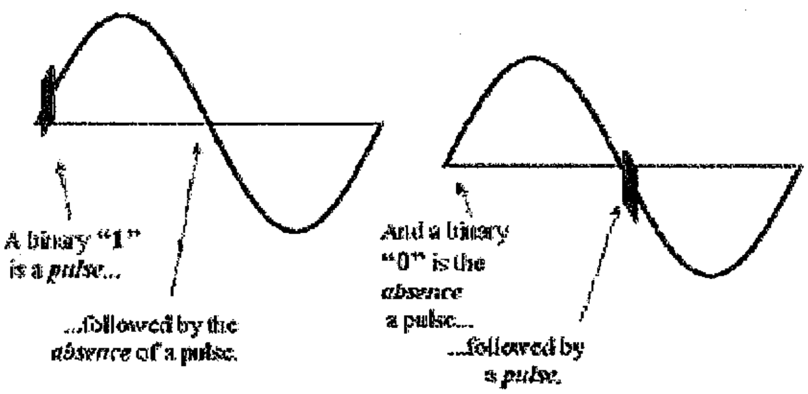


Figure 2.1: Bursts during the zero-crossing.

This protocol has many advantages including being inexpensive, no new wiring required, simple to install, compatible with many products, controls up to 256 devices. Household electrical wiring is used to send digital data between X10 devices. This digital data is encoded onto a 120 kHz carrier which is transmitted as bursts during the relatively quiet zero crossings of the 50 or 60 Hz AC alternating current waveform. One bit is transmitted at each zero crossing. The digital data consists of an address and a command sent from a controller to a controlled device. Controllers query equally advanced devices to respond with their status. This status may be as simple as "off" or "on", or the current dimming level, or even the temperature or other sensor reading. Devices usually plug into the wall where a lamp, television, or other household appliance plugs in; however some built-in controllers are also available for wall switches and ceiling fixtures. The relatively high-frequency carrier frequency carrying the signal cannot pass through a power transformer or across the phases of a multiphase system. *(Cole and Tran, 2002)*

In addition, because the signals are timed to coincide with the zero crossings of the voltage waveform, they would not be timed correctly to be coupled from phase-to-phase in a three-phase power system. Transmissions synchronized to zero crossing. For split phase systems, the signal can be passively coupled from phase-to-phase using a passive capacitor, but for three phase systems or where the capacitor provides insufficient coupling, an active X10 repeater is sometimes used. It may also be desirable to block X10 signals from leaving the local area so, for example, the X10 controls in one house don't interfere with the X10 controls in a neighboring house. In this situation, inductive filters can be used to attenuate the X10 signals coming into or going out of the local area.

2.3 Analyses of X-10 in HCS

Based on some literature review X10 may have negative impacts on the performance of HCS for the following reasons: (*Chunduru & Subramanian, 2007*):

1. Time to respond to the user is fast for short distances however, over the typical average long distances between the controller and the appliance encountered at home, the time to respond increases rapidly; the signal is sometimes so weak at the receiver that the receiver is not able to detect the signal at all.
2. Cost is usually low since no connectors need be installed between the controller and the appliance – however, amplifiers and noise filters are needed to send signals properly to distant nodes on the network and this quickly drives up the cost.
3. Ease of use may initially appear to be a strong point of the X10 technology – however, if the signal attenuation due to distance and line quality is significant then the system may well become unusable.
4. This technology is only accessible from home and not from outside – some form of converter will need to be used between outside connections and internal wiring; for example, if one requires that an appliance respond to commands from laptop in the homeowner's office, then the commands will have to be received over DSL or cable modem at the home and converted into X10 signals for transmission over home wiring which not only adds to the cost but could also result in undesirable delays.

5. Presence of noise and other disturbances on the power lines significantly impact the performance of HCS negatively: X10 devices such as lights are triggered randomly without any control command being sent to them. Heating pads and fluorescent lights also seem to affect X10 devices. An experiment that used a USB connection between the computer and the power line also did not alleviate the problems due to power line disturbances.

X-10 based transmitters send a specially coded low-voltage signal that is superimposed over the 240 volts on the home's electrical wires. A transmitter is usually capable of sending up to 256 different addresses on the AC line. Multiple transmitters can send signals to the same module.

Devices with specific symbol receive the special signals sent by the transmitters. Once a matching signal comes in, the device responds and turns ON or OFF or dims or brightens. Receivers generally have "code dials" that are adjusted by the user to set the address. Multiple devices with the same address can co-exist in the same home. These devices both send and receive X10 signals. Like regular receivers and transmitters, they can communicate on all 256 addresses

2.4 Performance of X-10 protocol.

The performance of X10 depends on several factors apart from its advantages including being inexpensive, no new wiring required, simple to install, compatible with many products, controls up to 256 devices it has many drawbacks as well. The drawbacks of X10 are signals from a transmitter in one live conductor may not propagate through the high impedance of the distributed transformer winding to the other live conductor. Often, there's simply no reliable path to allow the X10 signals to propagate from one phase wire to the other; this failure may come and go as large 240 volt devices such as stoves or dryers are turned on and off. (When turned on, such devices provide a low-impedance bridge for the X10 signals between the two phase wires.) This problem can be permanently overcome by installing a capacitor between the phase wires as a path for the X10 signals; the manufacturers commonly sell signal couplers that plug into 240 volt sockets that perform this function. More sophisticated installations install an active repeater device between the phases, while others combine signal amplifiers with a coupling device. A repeater is also needed for inter-phase communication in homes with three-phase electric power.

Some X10 controllers may not work well or at all with low power devices (below 50 watts) or devices like fluorescent bulbs that do not present resistive loads. Use of an appliance module rather than a lamp module may resolve this problem.X10 signals can only be transmitted one command at a time. If two X10 signals are transmitted at the same time, they will collide and the receivers will not be able to decode the signal commands.

The X10 protocol is also slow. It takes roughly three quarters of a second to transmit a device address and a command. *(Chunduru & Subramanian, 2007)* While generally not noticeable when using a tabletop controller, it becomes a noticeable problem when using 2-way switches or when utilizing some sort of computerized controller. The apparent delay can be lessened somewhat through the use of scenes and by using slower device dim rates.

CHAPTER 3

METHODOLOGY

3.1 Stages in Methodology:

For this Final Year Project, the modular methodology is used. Total, there are 5 modules to design and build: Transformerless Power Supply (5V), Signal Generator, Signal Extractor, Zero-Crossing Detector and TRIAC. For each of the modules, stages like Information Gathering, Design, Implementation and Testing were carried out separately. And the last stage was be the most critical - Module Testing and Integration Stage.

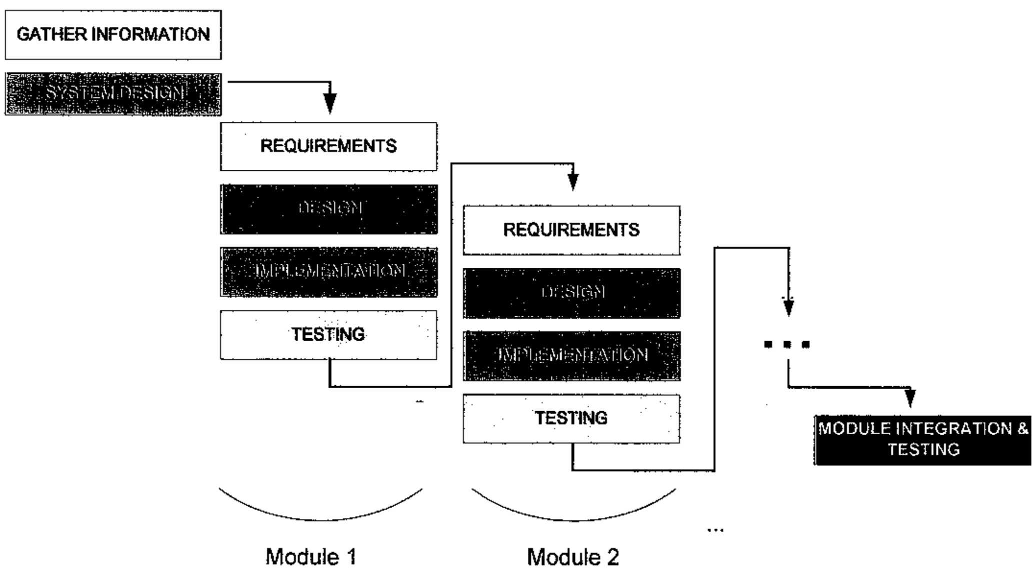


Figure 3.1: Design Phases

3.1.1 Gather Information

During this stage the information about the X-10 was collected, and reviewed in detail. The pros and cons of implementing the X-10 communication protocol for the Home Control System were analyzed. The X-10 and Home Automation research works and projects were sieve through, which were mentioned in the Literature and Review chapter.

3.1.2 System Design

The overall system's block diagram was prepared. The system Sender and Receiver are made up of 5 circuits: Signal Generator, Signal Extractor, TRIAC, Zero-Crossing Detector and Transformerless Power Supply. The estimated deadlines for each of the modules were calculated.

3.1.3 Requirements Gathering

Based on the study, the list of the required components is prepared and similar function circuits were found to adapt for the project.

3.1.4 Design

Some similar function circuits were used to make them compatible with the local powerlines standards.

3.1.5 Implementation

The circuits were built on the breadboard, and tested using relevant test gears (oscilloscope, multimeter…etc) for the respective functionality.

3.1.6 Testing

The circuits were tested using the oscilloscope, multimeter, logic probe etc. After all the successful testings of the modules, circuits will be soldered.

3.1.7 Module Integration & Testing

The final testing, when all the modules passed the functional test, this is the last testing to identify the problems with any of the modules. Final touch-up to the whole system will commence right after this stage.

It also requires adding the mains coupling circuit to transmit the 120 KHz (5V) signal over the powerline.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Module Development

In this section the development of Signal Generator, Signal Extractor, Transformerless Power Supply and Zero Crossing Detector modules will be discussed. There is a TRIAC module which was supposed to be built the last. On the Figure 4.1 you can see the Block Diagram of the whole System, which gives a better picture of how, what and where the modules are connected.

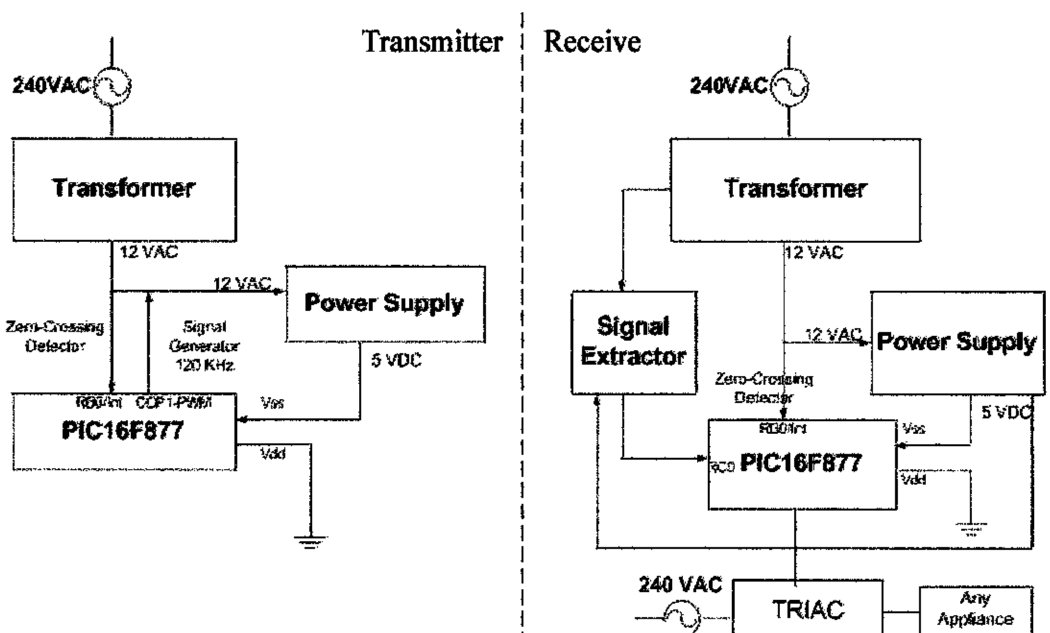


Figure 4.1: System Block Diagram

4.1.1 Dual-polarity Power Supply

This module is required to provide circuits with the +/-5V. Most of the electronic components require just +5V and Gnd, but for the operational amplifiers there should be -5 V instead of grounding.

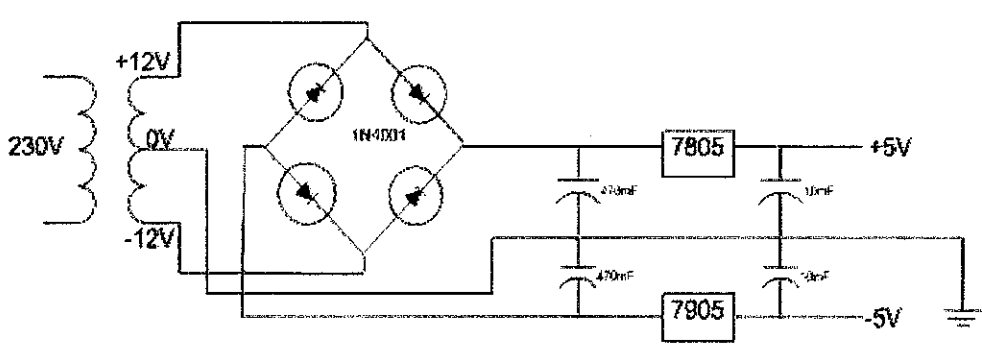


Figure 4.2: Power Supply Circuit

In the circuit, it can be observed that on the left there is a transformer converting from 240V to 12V, and to convert from 12VAC to +/-5VDC the full wave rectifier is used, with two voltage regulators. The heat sink is used to avoid any overheating of the voltage regulators.

4.1.2 Signal Extractor

Signal Extractor module will be placed on the Receiver side. It is the most complex circuit in the system. By referring to the existing circuits on the web and consulting some lecturer from EE department, I have adapted the circuit to fit to 240V. (For the circuit, please refer to Appendix F)

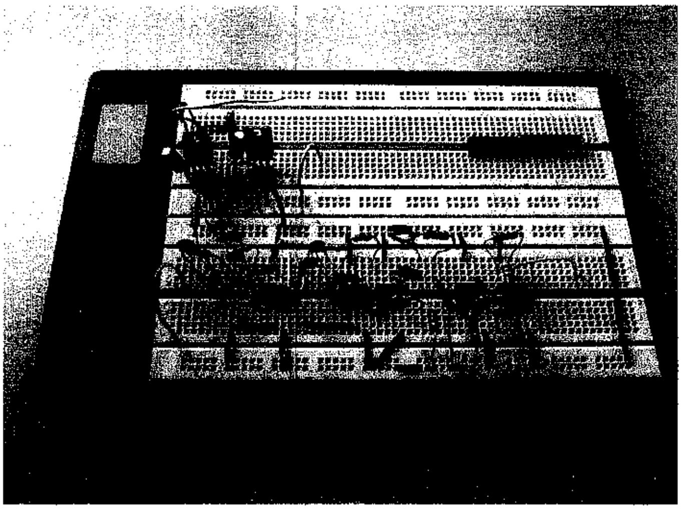


Figure 4.3: Signal Extractor Circuit.

Signal Extractor circuit has several functionalities such as:

- to cut the 50Hz frequency of the Powerlines
- to pass the 120 kHz signal

Testing was done using the Function Generator by Instec GFG, and Oscilloscope Tektronix. Below you can observe the results of the testing the circuit at different points.

As the output of the circuit goes straight to the PIC, it should produce either 0 or 1. As you can see below whenever there is 120 kHz signal appears on the input, there is output becomes '1'. Meaning the test was successful.

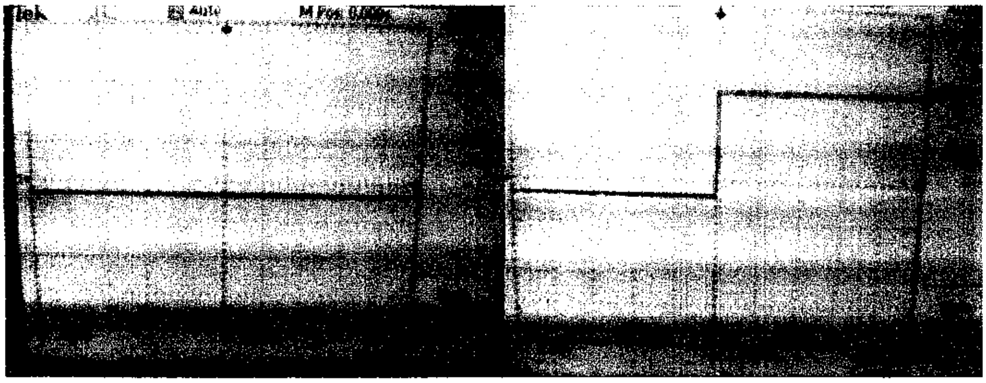


Figure 4.4: Before (left) and after (right) 120 kHz signal was sent through circuit.

And when there is 50 Hz signal on the input of the circuit, it is seen below that it does not affect the output of the circuit.

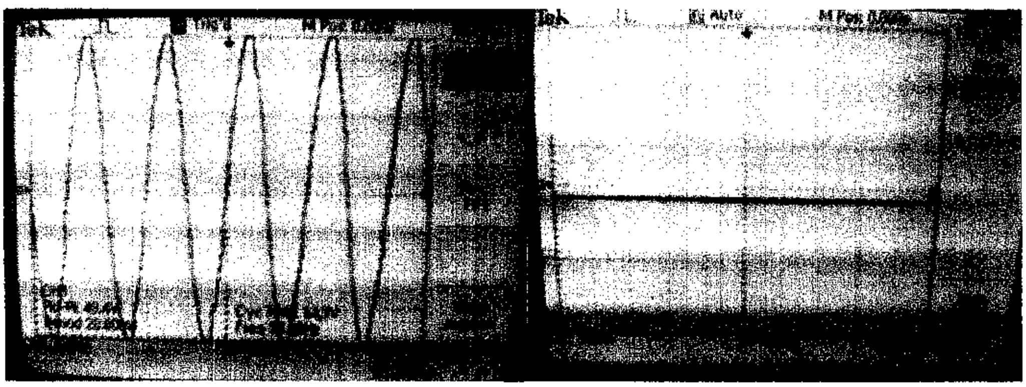


Figure 4.5: 50Hz input of the circuit (left), Empty output of the circuit (right)

4.1.3 Signal Generator

The PIC16F877 has a CCP1 and CCP2 pins, where the last 'P' stands for Pulse Width Modulation (PWM). Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output. There are several registers that affect the result of the PWM, they are PR2, CCPR1L, CCP1CON<5:4> bits and T2CON. To calculate the values of the above mentioned registers, the online available PWM calculator was used. [3. PWM Calculator]

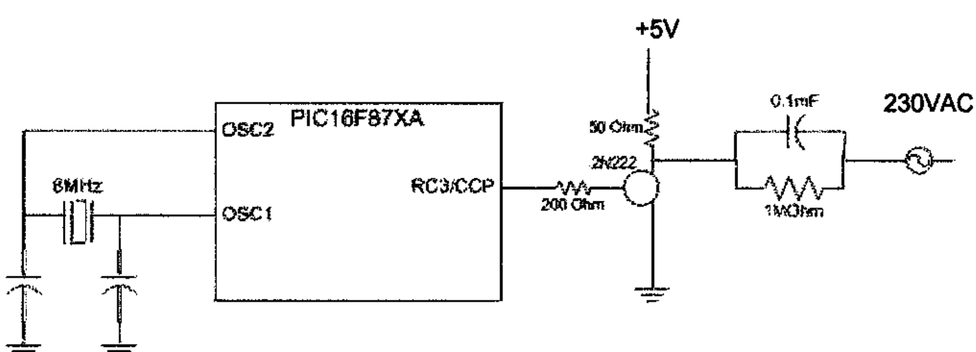


Figure 4.6: Signal Generator Circuit

The code to have PIC generating the 120 kHz signal is really simple, all we need to do is just set several registers to the particular value. The registers required are mentioned above, and here is the coding in Assembly:

```
PROCESSOR  16f877
        #INCLUDE      "P16F877.INC"
        __CONFIG _CP_OFF & _WDT_OFF & _HS_OSC
        org      0x00
        goto     Main
        org      0x04
        goto     Main

Main    bcf      STATUS,RP1
        bsf      STATUS,RP0  ;Select Bank1
        movlw  0x00          ;Port C as output-CCP1 and CCP2 are on Port C
        movwf  TRISC
        movlw  0x00
        movwf  OPTION_REG
        movlw  0x10    ;set the upperlimit for the timer2
        movwf  PR2
        bcf      STATUS,RP1
        bcf      STATUS,RP0  ;select bank 0
        movlw  0x04          ;set the prescaler and enable the timer2
        movwf  T2CON
        movlw  0x08          ;set the dutycycle
        movwf  CCPR1L
        movlw  0x1C          ;set the dutycycle
        movwf  CCP1CON

        END
```

Figure 4.7: Code for Signal Generator

The square wave 120 kHz signal was tested using the oscilloscope, and you can observe the signal from the Figure 4.8.

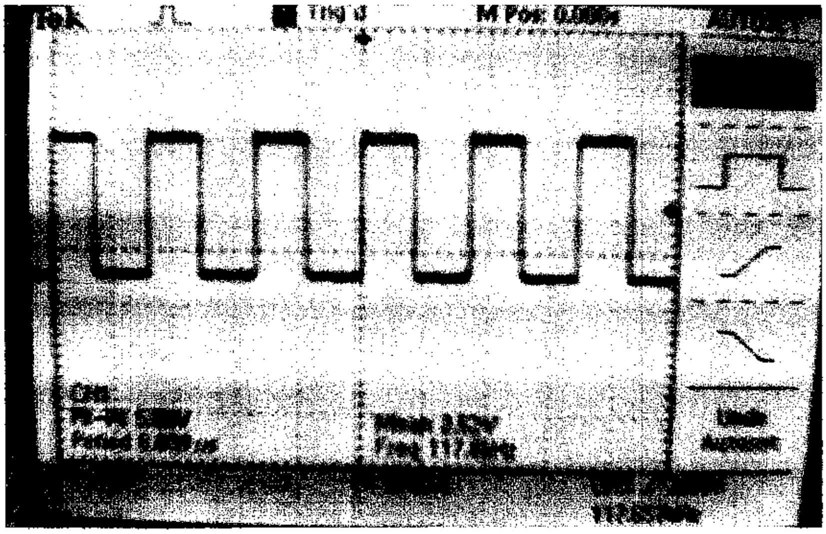


Figure 4.8: The 120 kHz PWM output (16F877)

4.1.4 Zero Crossing Detector

INTCON and OPTION_REG registers are playing the main role in setting the external interrupt. External interrupt on the RB0/INT pin is edge triggered, either rising, if bit INTEDG (OPTION_REG<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>).

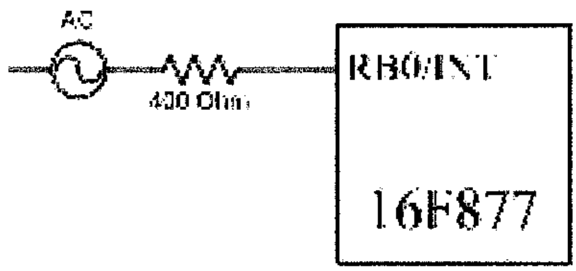


Figure 4.9: Zero Crossing Detector Circuit

The input will be 12VAC (the output of transformer), if we select a resistor of 0.4MΩ, 12/0.4=30mA, which is well within the current capacity of a PICmicro MCU I/O pin.

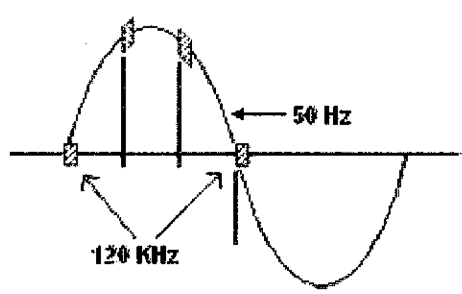


Figure 4.10: 120 kHz bursts on 50Hz signal

In order to send the bit every zero-crossing of the powerline signal, the INTEDG register is toggled on the falling and rising edge of the alternating powerline signal. During first few cycles interrupt will synchronize and later the pin C0 was changing with the rising and falling edges of the signal input at B0/Int. To test the Zero-Crossing Detector program the MPLAB IDE and PIC Simulator softwares were used. The coding is written in Assembly language, and did not exceed the 50 lines, you can have a look at Appendix H.

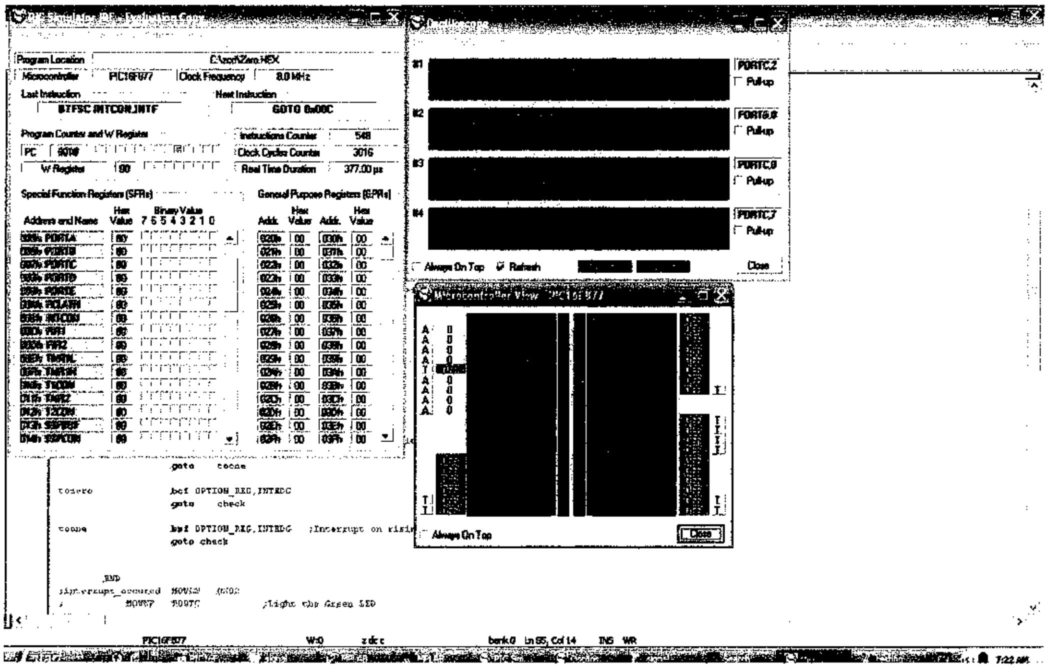


Figure 4.11: Testing the Zero Crossing Detector program using the PIC simulator.

4.1.5 Mains Coupling

The output is coupled to the mains via 1:1 isolating transformer T2. Coupling capacitors have a high reactance (318k) at 50Hz falling to only 160Ohms at 120 kHz. It is essential to use the high voltage capacitors because they must withstand the peak mains voltage. Transient suppressor TS1 protects the BJT Q1 from spikes of more than 10V.

The mains coupling has not been tested, but has been edited from the one used by the Andrew Holme in his Wireless Thermostat project in 2004. Due to the lack of experience and knowledge in electrical area, a main coupling is left for future enhancements.

For the full circuit of the Sender and Receiver please refer to the Appendix C and D.

4.1.6 Transmission of 5 bit data

One can ask, how both sender and receiver can be synchronized, the answer is that system uses bits to synchronize. The protocol is asynchronous as it is not necessary to send clocking information with the data that is sent.

The protocol is character oriented, and transmission is carried out without continuous character synchronization between the transmit and receive devices. Each group of data should be identified separately, and hence the beginning should be marked. Instead of having transmit and receive clocks connected, the receiver synchronized itself based on the start bit.

As it was mentioned above, one of the disadvantages of X-10 is slowness of transmitting data. Initially X-10 protocol was designed to have 11 bits, where the first 4 bits serve for the start code, the next 4 bits for address purpose (as each of the appliances in the house will have its own address) and the last 3 to control the unit. I have decided to test whether 5 bit will be sufficient to control devices. As some of the customers would like appliance to turn on/off only, and will have only one equipment to control. It should be enough to use the 5 bit, where the first 4 bit will serve as start and error-checking code, and the 5th bit to switch on/off the device.

1	1	0	1	1 or 0

Figure 4.12: 5-bit code

"1101" is set to be as "start code" and control bit is "1" to turn on, and "0" to turn off. All the coding was written in Assembly, please refer to the Appendix for more details and description of almost every line.

So, here is the testing that has been done for the coding of the Sender. To make it easier to understand the simulation part, reader should refer to the table of pins used.

Pins	Functionality
1. RBO	Serves as interrupt, and detects the zero-crossing. Has a built-in function to be triggered by raising and falling edge of the input signal.
2. RB5	The pin where the "triggering device" will be connected. To send "turn ON" command.
3. RB6	The pin where the "triggering device" will be connected. To send "turn OFF" command.
4. RC2	The output from the Sender part.
5. RC0	It is on the Receiver part. The input from the Signal Extractor.

Table 1: PIC pins used in simulation and coding

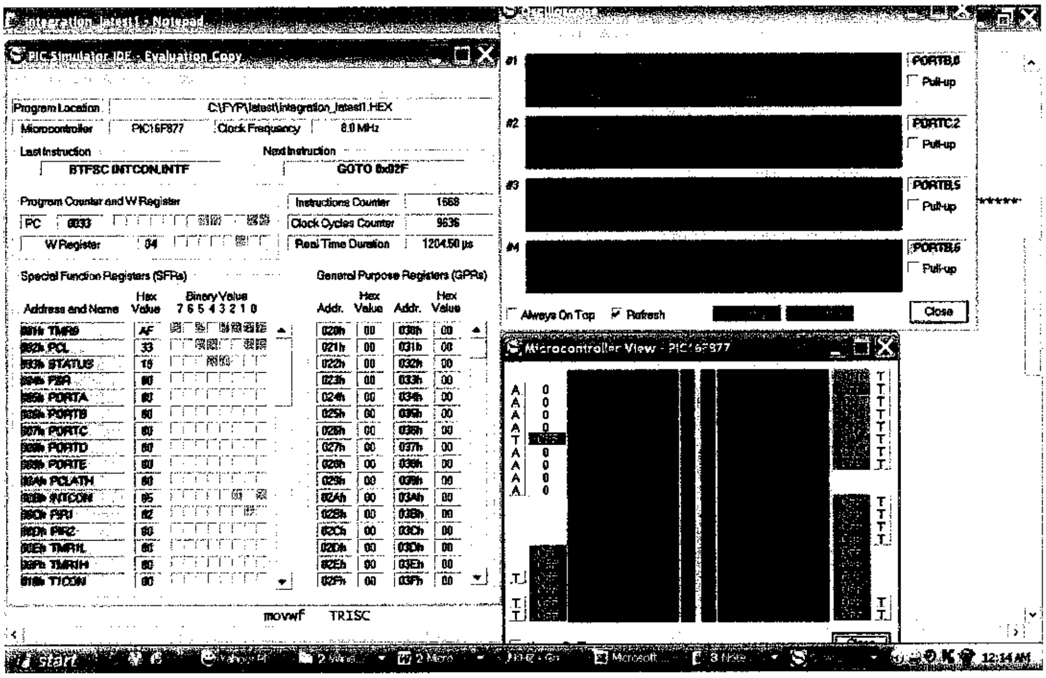


Figure 4.13: Sending "11011"-switch On command

As you can observe from the Figure 4.13, there is PB5 which is "On", meaning that "Switch On" command is sent. PB0 is turned on and off manually, as to simulate the zero-crossings of the powerline signal. And PC2 is output of the Sender, which is triggered by zero-crossing to send a "11011"-switch On command.

In the same way, but with PB5=Off and PB6=On, the switch Off ("11010") command is sent to the Receiver.

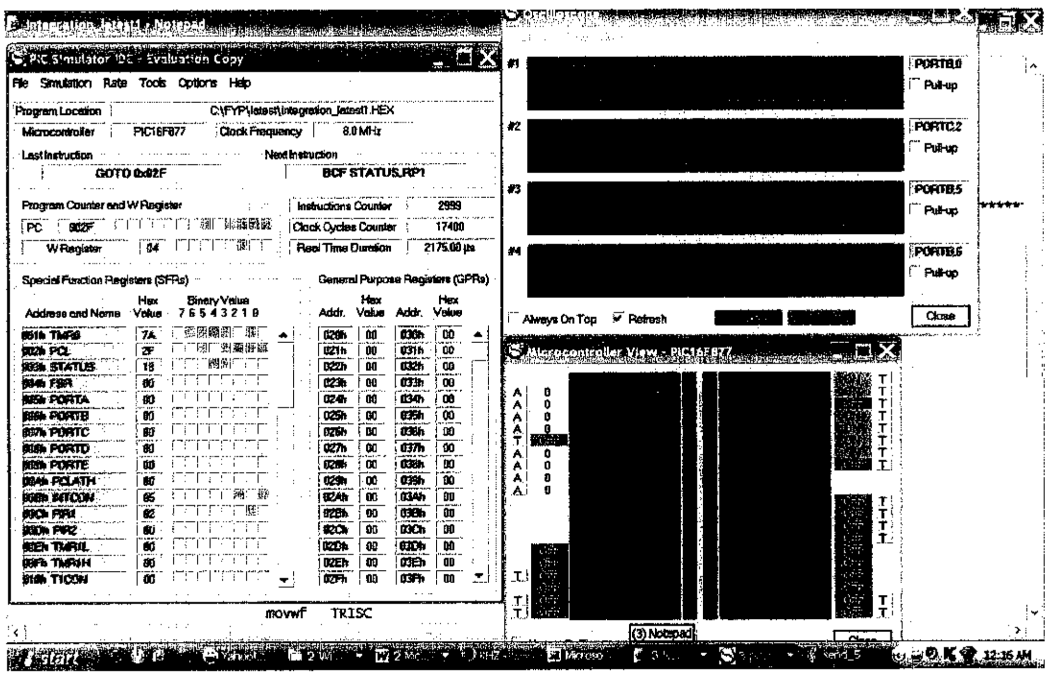


Figure 4.14: Sending "11010"-switch Off command

On the Receiver part, the same pins mentioned above are used. PB0 and PC0 are inputs from the powerline and Signal Extractor circuit respectively. Both PB0 and PC0 are turned on/off manually, but with respect to commands ("11011" and "11010"). And from below snapshots it can be observed that PB4 (PIC pin which supposed to control the TRIAC or Relay) is triggered by commands and functions properly.

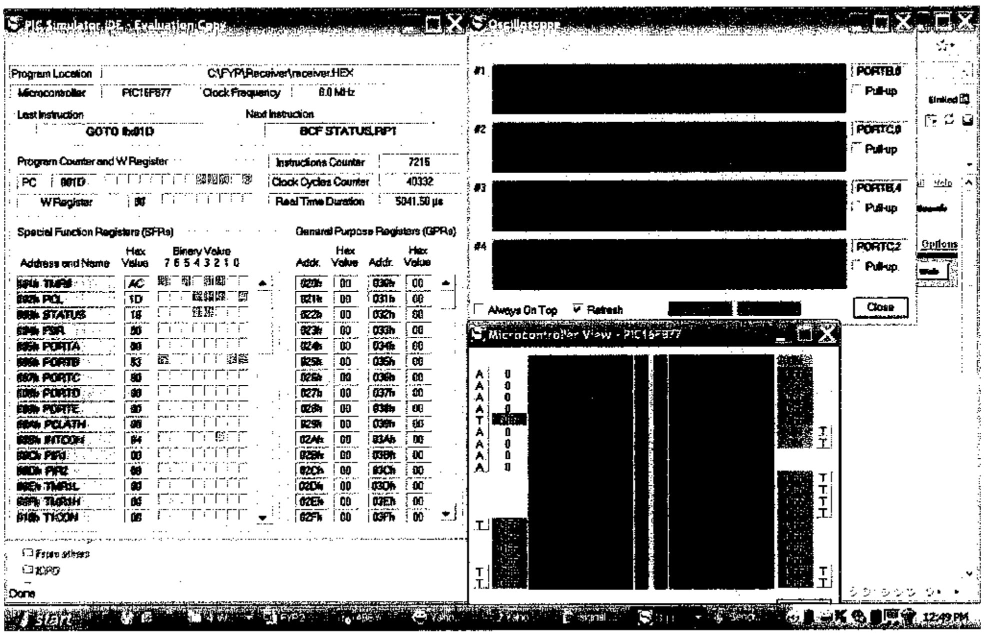


Figure 4.15: Sending "ON" signal. (11011)

And on the Figure 4.16, there are "11010" and "11011" bits sent to switch "OFF" and "ON" the device.

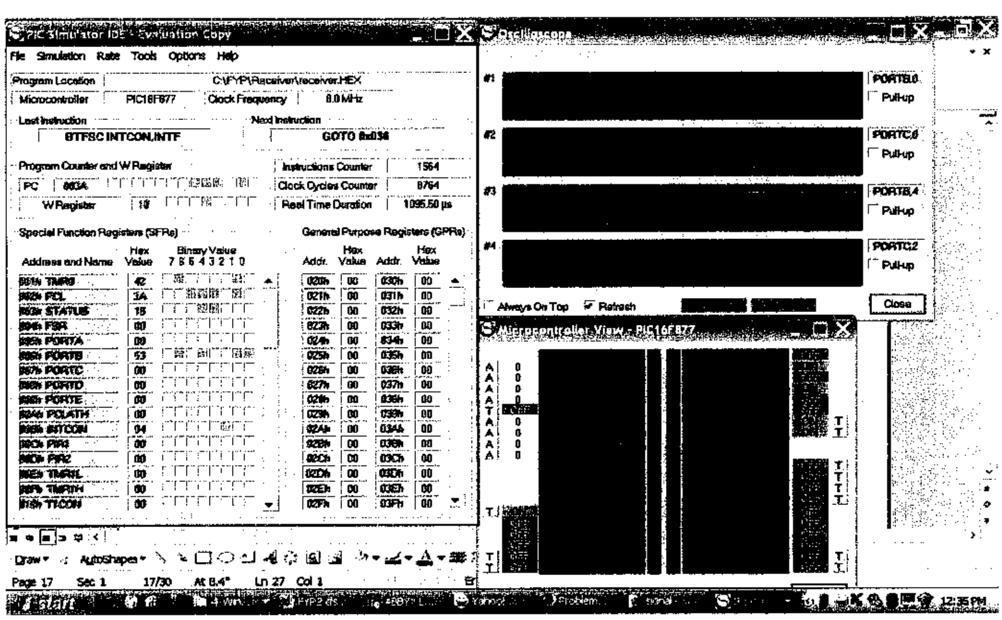


Figure 4.16: Sending "OFF" and "ON" signals. (11010 and 11011)

CHAPTER 5

CONCLUSION and RECOMMENDATION

5.1 Conclusion

Development of the X-10 based Outdoor Light Automation requires a deep knowledge and expertise in analog components, and circuit analyses. Due to the author's lack of knowledge in electrical area, the mains coupling could not be completed as it also required the electrically isolated lab to test the circuit. Thus the mains coupling is left for future enhancements of the system.

Nevertheless, the development and testing stages for all 4 out of 5 modules is completed. And the sending and receiving of 5 bit messages using 16F877 PIC is successfully simulated.

The X-10 based Outdoor Light Automation System will bring the following benefits:

- Fully-automated - the lights will be triggered by the relevant input device(for this project it is photodiode)
- Multi-purpose- we can plug into the system's socket any appliance in order to automate it.
- Portable-we can move and place the system's receiver and sender anywhere we want.
- Reprogrammable PIC- we can reprogram the PIC to change the functionality.

But I believe with an effort, any challenge can be passed, and any aim can be reached, as long as we are properly stuck to the true path.

5.2 Recommendation

I would like to recommend adding the human control to the sender module. It can be done using the infrared, or even on the module itself, by adding a LCD and some control buttons like switch on and off, dimming, numbers from 1-10 to input the address of the required object etc. So by adding human control, we can give addresses to the receiver modules, and will be able to control up to 256 appliances.

As was mentioned before, after we have built the sender and receiver modules, we can automatize many households items. It is based on our creativity and needs, to decide where and how to apply this powerful X10 communication protocol.

REFERENCES

[1] Dave Rye, 2004, "X-10 Technical Note", X-10 Inc., USA.

[2] Jon Burroughs, 2002, "AN236-X-10 Home Automation Using the PIC16F877A", Microchip Technology Inc., USA

[3] Shabsi Walfish, Omar Badoolah, 2004,"X10 Receiver Project", Cooper University, USA

[4] "TDS1000 Oscilloscope User Manual", 2004, Tektronix Inc., USA.

[5] Robert C. Elsenpeter, 2005, "Build Your Own Smart Home", Prentice Hall, USA

[6] Donal Scouten, 2004, <www.x10-mentor.com>.

[7] Mike Kuniavsky, 2007, <http://www.orangecone.com/archives/2007/01/we_need_an_appl.html>

[8] Smarthome Store, 2004, <http://www.smarthome.com/about_X10_Compatible.html>

[9] BrunoG, 2006, < http://www.micro-examples.com>

[10] V. Chunduru, *Student Member, IEEE,* and N. Subramanian, *Member, IEEE, 2007,* "Effects of Power Lines on Performance of Home Control System", University of Texas at Tyler, USA

[11] A.M.Cole, B.Q.Tran, 2002, "Home Automation to Promote Independent Living in elderly populations", HomeCare & Telerehabilitation Center, The Catholic University of America, Washington, DC.

[12] Microchip Technology Inc., 2001, "PIC16F87x Data sheet".

[13] Murray Gunn, 2000, "Communications Over Power Lines", University of New South Wales-IEEE, Australia.

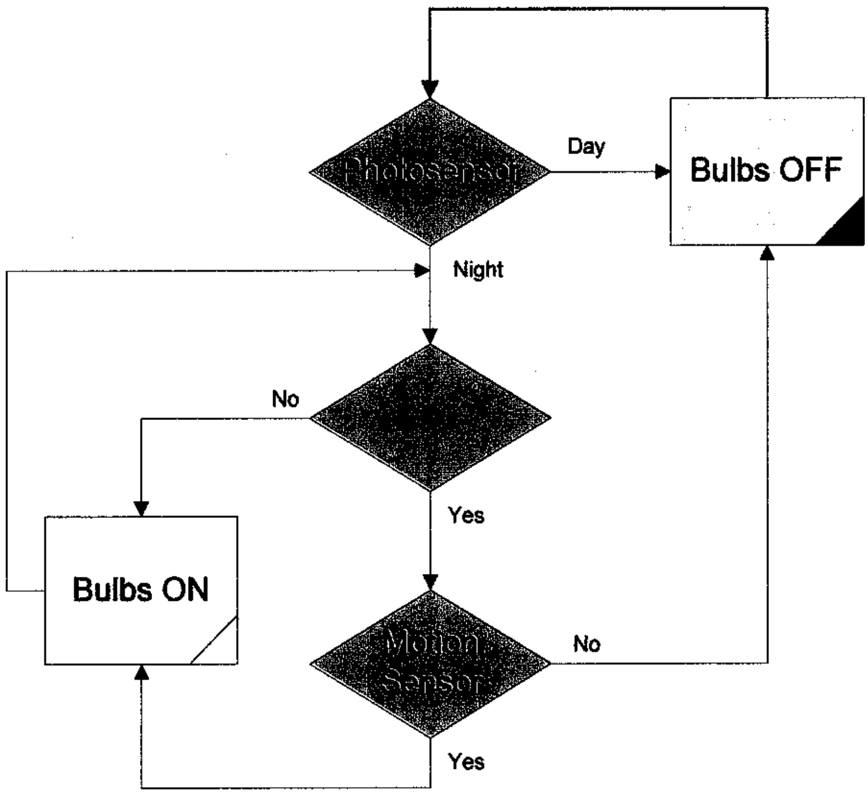
[14] "Microelectronics", McGraw-Hill, 1988.

[15] Kevin Wade Ackerman, 2005, "Timed Power Line Data Communication", Thesis, Department of Electrical Engineering, University of Saskatchewan, Canada.

[16] Andrew Holme, 2004, "Wireless Thermostat.",
<http://www.holmea.demon.co.uk/Thermostat/Stat.htm>

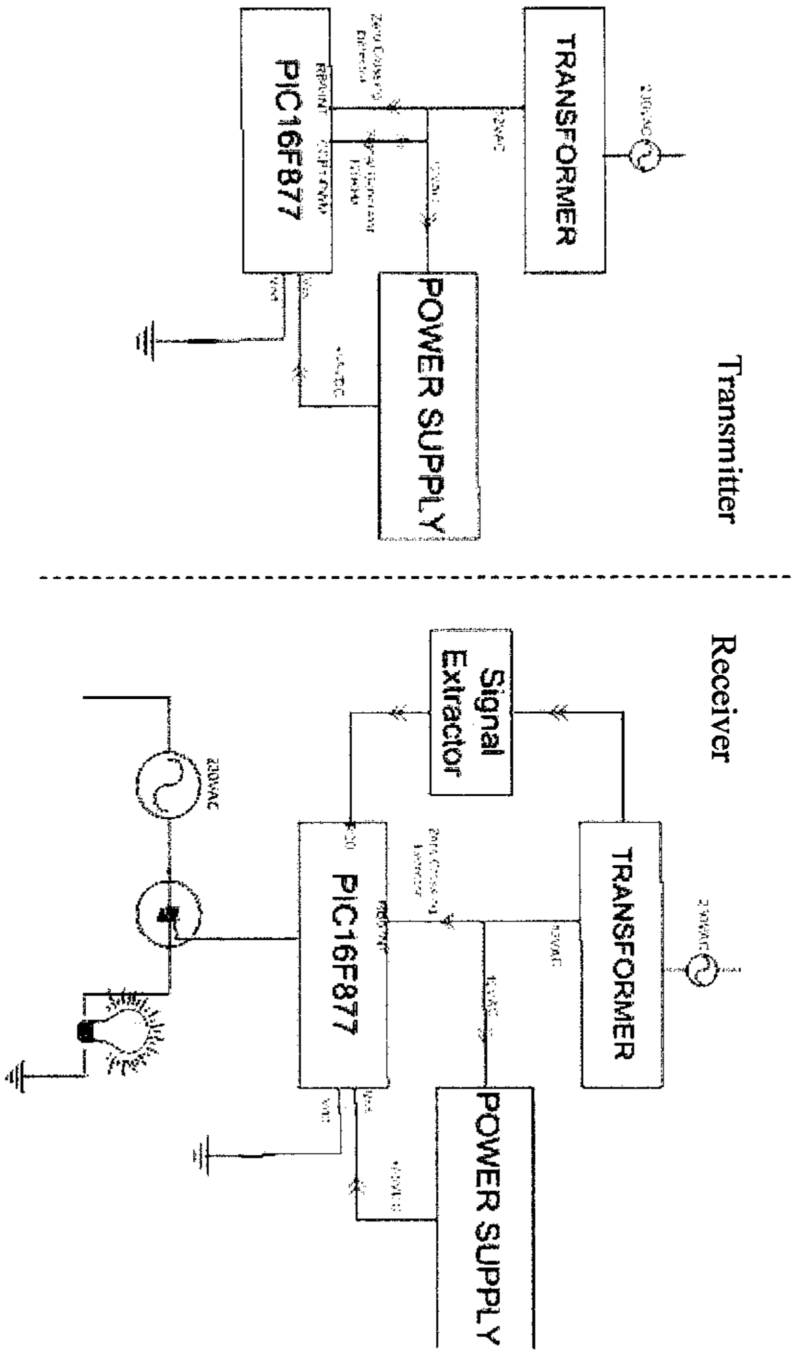
APPENDICES

Appendix A: Flow Chart of the System

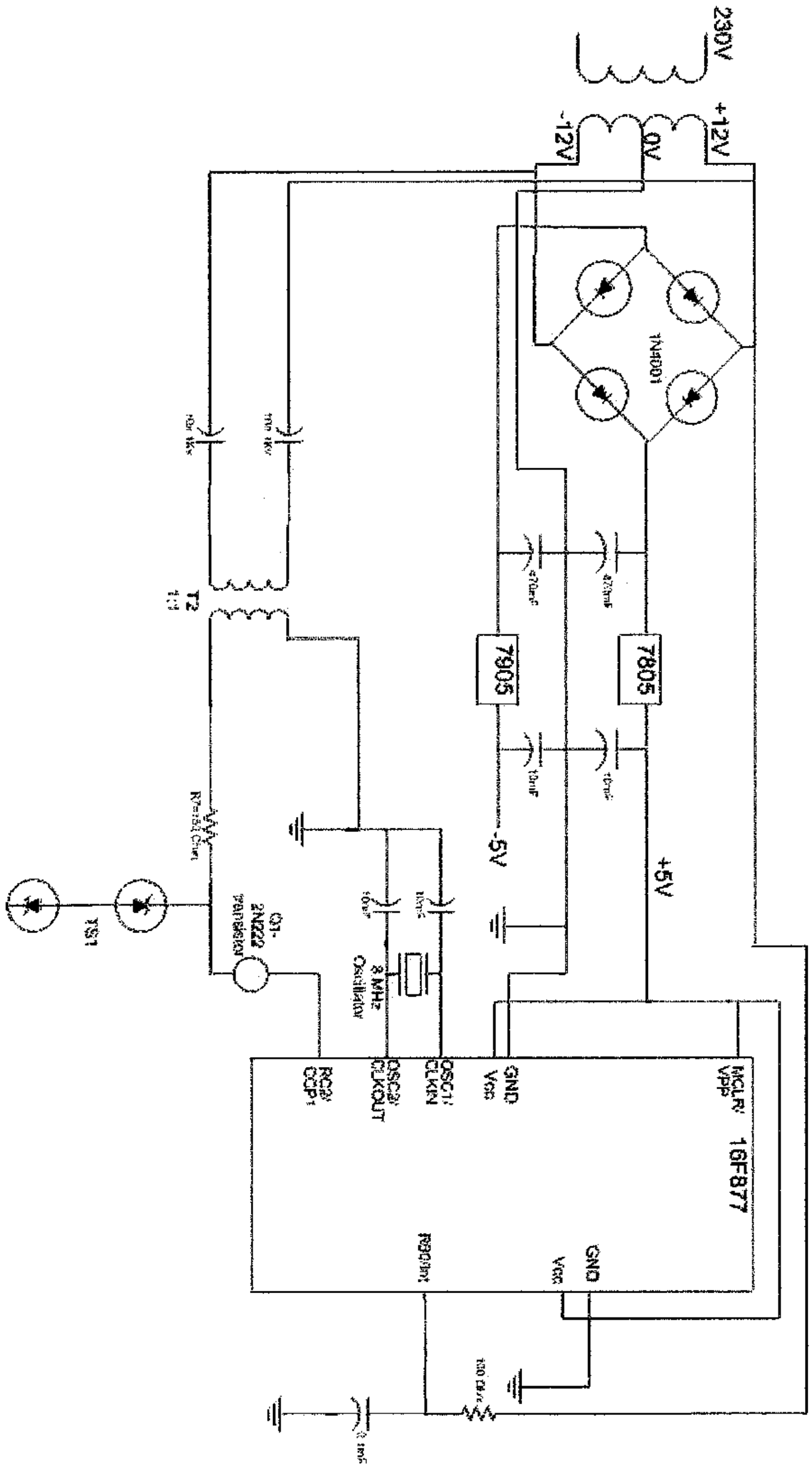


Note: Photo sensor will be implemented upon successful completion of the main modules.

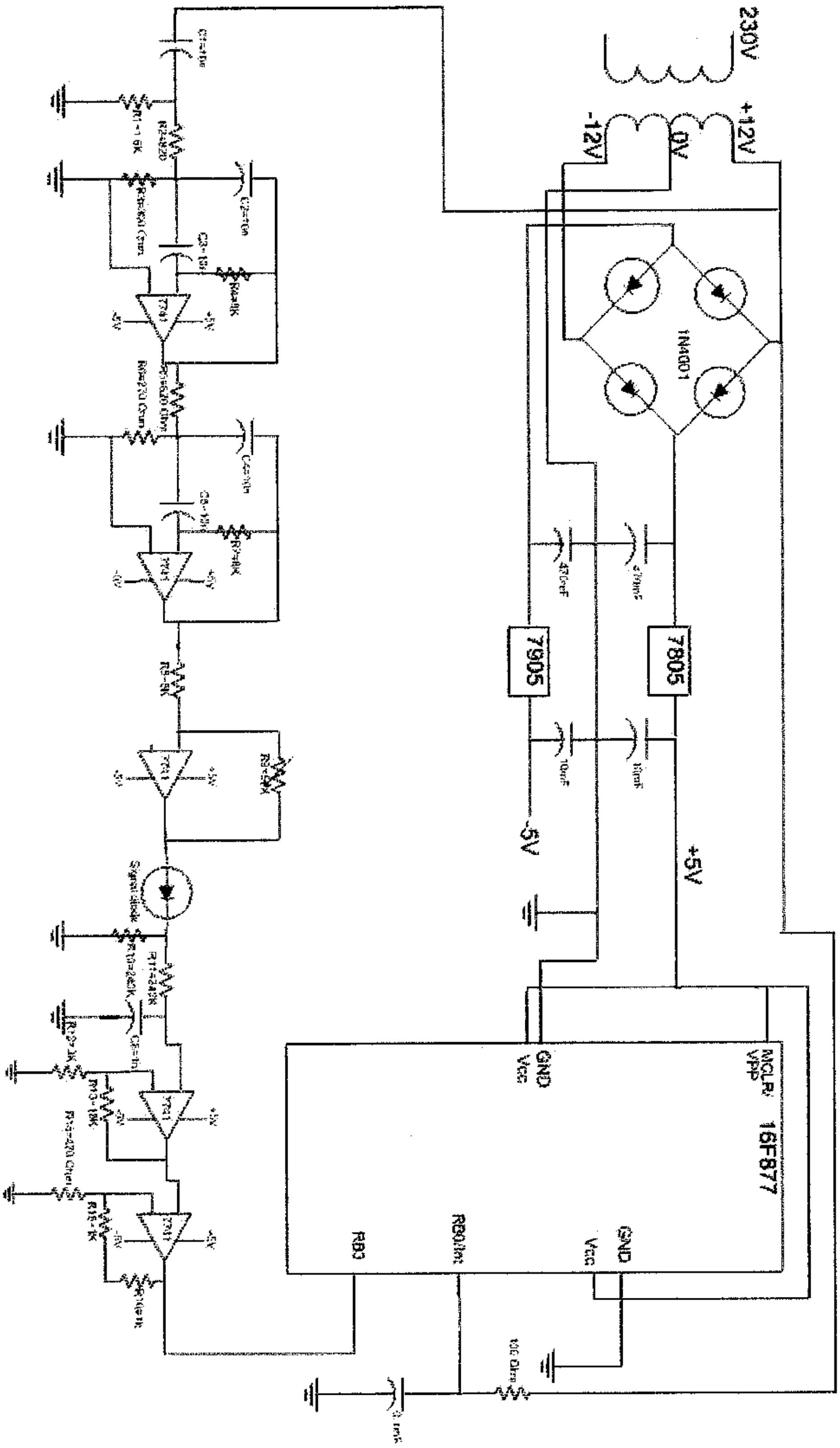
APPENDIX B: Overall System Block Diagram



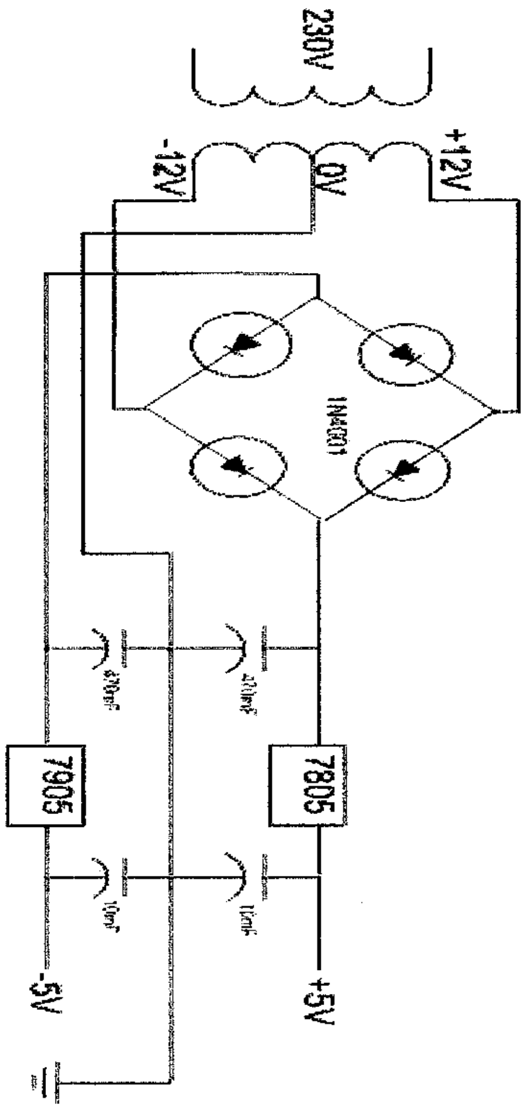
Appendix C: Full circuit for the Sender



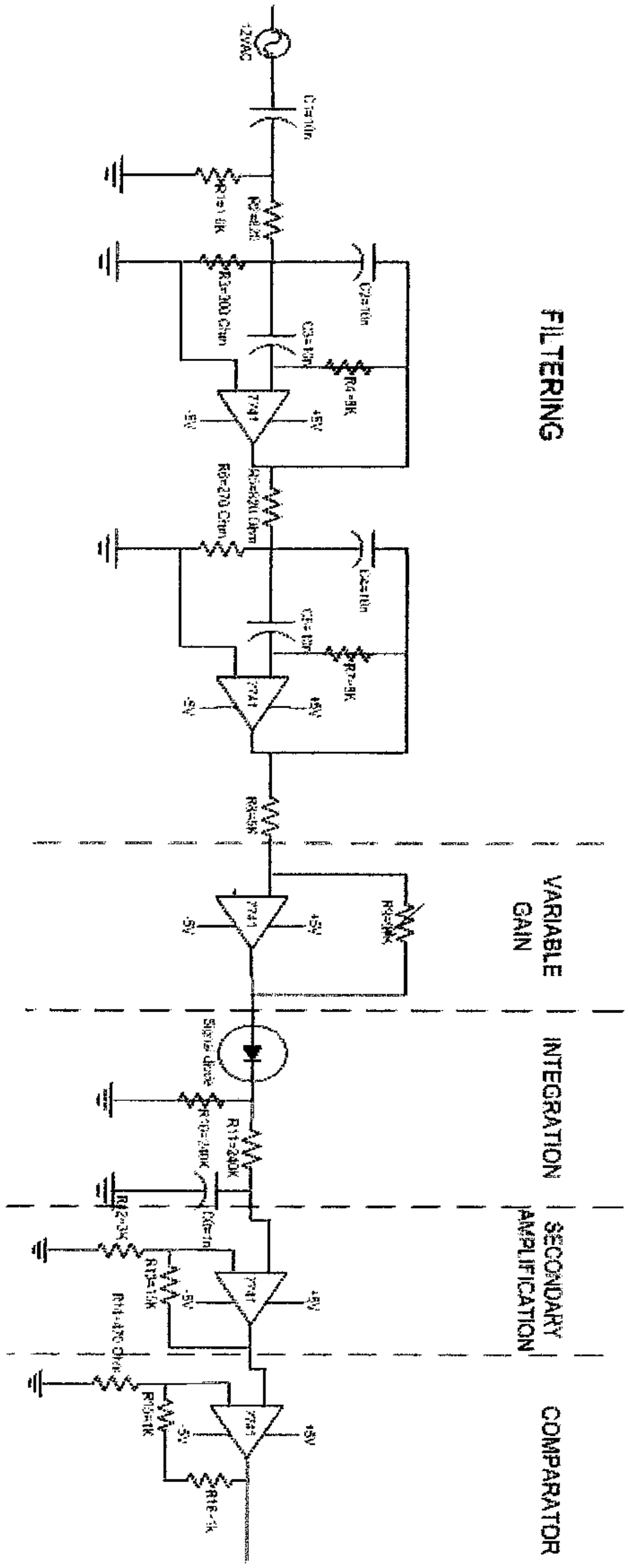
Appendix D: Full circuit of the Receiver



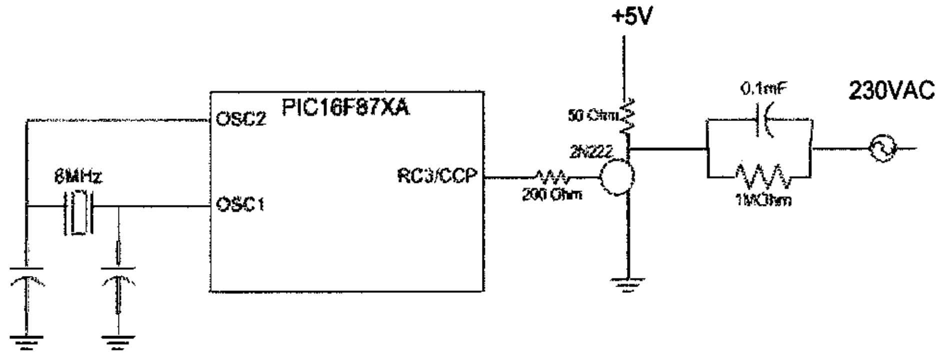
Appendix E: Circuit of the Dual-Polarity Power Supply



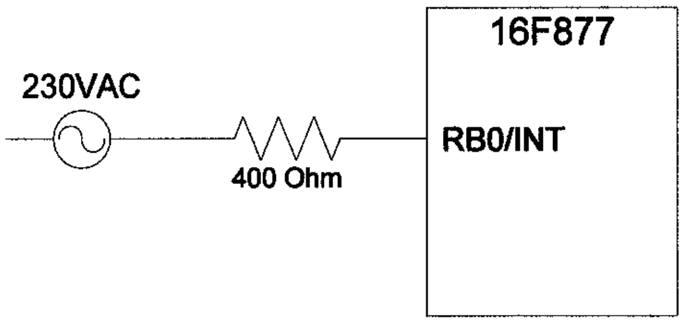
Appendix F: Signal Extractor Circuit



Appendix G: Signal Generator and Zero-Crossing Detector Circuits.



Signal Generator Circuit



Zero-Crossing Detector Circuit

Appendix H: Zero-Crossing Detector Coding

```
                PROCESSOR     16f877
                #INCLUDE      "P16F877.INC"

                __CONFIG _CP_OFF & _WDT_OFF & _HS_OSC

        ORG     0X00
        goto    Main
        org     0x04
        goto    interrupt_occured

        BCF             STATUS, RP1     ;Select bank 1
        BSF             STATUS, RP0
        bcf             OPTION_REG, INTEDG ;Interrupt on falling edge

        MOVLW           0X01
        MOVWF           TRISB           ;Set Port B0 as Input
        MOVLW           0X00
        MOVWF           TRISC           ;Set Port C as Output,for LED's

check   BCF     STATUS, RP1             ;Select bank 1
        BCF     STATUS, RP0
        btfsc   INTCON, INTF            ;check the RB0/INT flag
        goto    interrupt_occured
        goto    check


interrupt_occured       BCF     STATUS, RP1             ;Select bank0
                        BCF     STATUS, RP0
                        MOVLW           0X01
                        MOVWF           PORTC           ; Light the RED LED
                        goto default

default BCF     STATUS, RP1             ;Select bank 1
        BSF     STATUS, RP0
        bcf     INTCON,INTF             ;make RB0/INT flag=0->interrupt didn't occur
        BCF             STATUS, RP1     ; Select bank0
        BCF             STATUS, RP0
        MOVLW           0X00
        MOVWF           PORTC           ;all are OFF
        goto toggle

toggle  BCF     STATUS, RP1             ;Select bank 1
        BSF     STATUS, RP0
        btfsc   OPTION_REG,INTEDG       ;toggle bit, if falling, become rising and vice-verse.
        goto    tozero
        goto    tone

tozero  bcf     OPTION_REG,INTEDG
        goto    check

toone   bsf     OPTION_REG,INTEDG       ;Interrupt on rising edge
        goto check

        END
```

Appendix I: Code to Send 5bit On/Off commands.

```
        PROCESSOR       16f877
        #INCLUDE        "P16F877.INC"
        __CONFIG _CP_OFF & _WDT_OFF & _HS_OSC

        ORG     0X00
        goto    Main
        org     0x04
        goto    check_On_button

Main    BCF     STATUS, RP1             ;Select bank 1
        BSF     STATUS, RP0

        MOVLW           0X01
        MOVWF           TRISC                   ;Set Port C0 as Input
        MOVLW           0X03
        MOVWF           TRISB                   ;Set Port B as output,B0 and B1 as input

        movlw   0x00
        movwf   OPTION_REG              ;interrupt on falling edge

check_zero_crossing             BCF     STATUS, RP1             ;Select bank 0
                                BCF     STATUS, RP0

                                btfsc   INTCON, INTF            ;check the RB0/INT flag
                                goto    check__bit_1
                                goto    check_zero_crossing

check_bit_1                     BCF     STATUS, RP1             ;Select bank 0
                                BCF     STATUS, RP0

                                btfsc   PORTC,0                         ;check for signal
                                goto    check_bit_2
                                goto    check_zero_crossing
;*******************************************************************************
check_bit_2                     BCF     STATUS, RP1             ;Select bank 0
                                BCF     STATUS, RP0
                                btfsc   INTCON, INTF            ;check the RB0/INT flag
                                goto    check_bit_2a
                                goto    check_bit_2

check_bit_2a                    BCF     STATUS, RP1             ;Select bank 0
                                BCF     STATUS, RP0
                                btfsc   PORTC,0                 ;check for signal
                                goto    check_bit_3
                                goto    check_zero_crossing
;*******************************************************************************
check_bit_3                     BCF     STATUS, RP1             ;Select bank 0
                                BCF     STATUS, RP0
                                btfsc   INTCON, INTF            ;check the RB0/INT flag
                                goto    check_bit_3a
                                goto    check_bit_3

check_bit_3a                    BCF     STATUS, RP1             ;Select bank 0
                                BCF     STATUS, RP0
                                btfsc   PORTC,0                         ;check for signal
                                goto    check_zero_crossing
                                goto    check_bit_4             ;3rd bit supposed to be 0-
error checking bit!
```

```
check_bit_4         BCF     STATUS, RP1         ;Select bank 0
                    BCF     STATUS, RP0

                    btfsc   INTCON, INTF        ;check the RB0/INT flag
                    goto    check_bit_4a
                    goto    check_bit_4

check_bit_4a        BCF     STATUS, RP1         ;Select bank 0
                    BCF     STATUS, RP0

                    btfsc   PORTC,0                     ;check for signal
                    goto    check_bit_5
                    goto    check_zero_crossing

;*******************************************************************************

check_bit_5         BCF     STATUS, RP1         ;Select bank 0
                    BCF     STATUS, RP0

                    btfsc   INTCON, INTF        ;check the RB0/INT flag
                    goto    check_bit_5
                    goto    check_bit_5a

check_bit_5a        BCF     STATUS, RP1         ;Select bank 0
                    BCF     STATUS, RP0

                    btfsc   PORTC,0                     ;check for signal
                    goto    ON
                    goto    OFF

ON                  movlw   0x10
                    movwf   PORTB               ;switch On the Device
                    goto    check_zero_crossing

OFF                 movlw   0x00
                    movwf   PORTB
                    goto    check_zero_crossing


          END
```

Appendix J: Code to Receive 5bit On/Off commands.

```
        PROCESSOR   16f877
        #INCLUDE    "P16F877.INC"
        __CONFIG _CP_OFF & _WDT_OFF & _HS_OSC

        ORG     0X00
        goto    Main
        org     0x04
        goto    check_bit_1


Main    BCF     STATUS, RP1          ;Select bank 1
        BSF     STATUS, RP0

        MOVLW       0X01
        MOVWF       TRISC                ;Set Port C0 as Input
        MOVLW       0X03
        MOVWF       TRISB                ;Set Port B as output,B0 and B1 as input

        movlw       0x00
        movwf       OPTION_REG           ;interrupt on falling edge
        BCF         STATUS, RP1          ;Select bank 0
        BCF         STATUS, RP0
        MOVLW       0x03
        movwf   PORTB


check_zero_crossing          BCF     STATUS, RP1          ;Select bank 1
                             BSF     STATUS, RP0

                bcf     INTCON,INTF             ;make RB0/INT flag=0->interrupt didnt occur

toggle_1        BCF     STATUS, RP1             ;Select bank 1
                BSF     STATUS, RP0

                btfsc   OPTION_REG,INTEDG       ;toggle the bit, if was falling, become rising &
                                                ;viceverse.
                goto    tozero_1
                goto    toone_1

tozero_1        bcf     OPTION_REG,INTEDG
                goto    continue_1
toone_1         bsf     OPTION_REG,INTEDG       ;Interrupt on rising edge
                goto    continue_1
continue_1      BCF     STATUS, RP1             ;Select bank 0
                BCF     STATUS, RP0
                BSF     PORTB,7

                btfsc   INTCON, INTF            ;check the RB0/INT flag
                goto    check_bit_1
                goto    continue_1

check_bit_1     BCF     STATUS, RP1             ;Select bank 0
                BCF     STATUS, RP0

                bcf     PORTB,7
                btfsc   PORTC,0                         ;check for signal
                goto    check_bit_2
                goto    check_zero_crossing
```

```
check_bit_2     bsf     PORTB,6
                BCF     STATUS, RP1             ;Select bank 1
                BSF     STATUS, RP0
                bcf     INTCON,INTF             ;make RB0/INT flag=0->interrupt didnt occur

toggle_2        BCF     STATUS, RP1             ;Select bank 1
                BSF     STATUS, RP0

                btfsc   OPTION_REG,INTEDG       ;toggle the bit, if it was falling, become rising
                                                ;and vice-verse.
                goto    tozero_2
                goto    toone_2

tozero_2        bcf     OPTION_REG,INTEDG
                goto    continue_2

toone_2         bsf     OPTION_REG,INTEDG       ;Interrupt on rising edge
                goto    continue_2

continue_2      BCF     STATUS, RP1             ;Select bank 0
                BCF     STATUS, RP0

                btfsc   INTCON, INTF            ;check the RB0/INT flag
                goto    check_bit_2a
                goto    continue_2

check_bit_2a    BCF     STATUS, RP1             ;Select bank 0
                BCF     STATUS, RP0
                btfsc   PORTC,0                         ;check for signal
                goto    check_bit_3
                goto    check_zero_crossing

;*********************************************************************************
check_bit_3     bcf     PORTB,6
                BCF     STATUS, RP1             ;Select bank 1
                BSF     STATUS, RP0
                bcf     INTCON,INTF             ;make RB0/INT flag=0->interrupt didnt occur

toggle_3        BCF     STATUS, RP1             ;Select bank 1
                BSF     STATUS, RP0
                btfsc   OPTION_REG,INTEDG       ;toggle the bit, if it was falling, become rising
and vice-verse.
                goto    tozero_3
                goto    toone_3

tozero_3        bcf     OPTION_REG,INTEDG
                goto    continue_3

toone_3         bsf     OPTION_REG,INTEDG       ;Interrupt on rising edge
                goto    continue_3

continue_3      BCF     STATUS, RP1             ;Select bank 0
                BCF     STATUS, RP0
                btfsc   INTCON, INTF            ;check the RB0/INT flag
                goto    check_bit_3a
                goto    continue_3

check_bit_3a    BCF     STATUS, RP1             ;Select bank 0
                BCF     STATUS, RP0
                btfsc   PORTC,0                         ;check for signal
                goto    check_zero_crossing
                goto    check_bit_4             ;3rd bit supposed to be 0-error checking bit!
```

```
check_bit_4     bsf     PORTB,6
                BCF     STATUS, RP1             ;Select bank 1
                BSF     STATUS, RP0
                bcf     INTCON,INTF             ;make RB0/INT flag=0->interrupt didnt occur
toggle_4        BCF     STATUS, RP1             ;Select bank 1
                BSF     STATUS, RP0
                btfsc   OPTION_REG,INTEDG  ;toggle the bit, if it was falling, become rising
and vice-verse.
                goto    tozero_4
                goto    toone_4
tozero_4        bcf     OPTION_REG,INTEDG
                goto    continue_4
toone_4         bsf     OPTION_REG,INTEDG  ;Interrupt on rising edge
                goto    continue_4
continue_4      BCF     STATUS, RP1             ;Select bank 0
                BCF     STATUS, RP0
                btfsc   INTCON, INTF            ;check the RB0/INT flag
                goto    check_bit_4a
                goto    continue_4
check_bit_4a    BCF     STATUS, RP1             ;Select bank 0
                BCF     STATUS, RP0
                btfsc   PORTC,0                         ;check for signal
                goto    check_bit_5
                goto    check_zero_crossing
;*****************************************************************************************
check_bit_5     bcf     PORTB,6
                BCF     STATUS, RP1             ;Select bank 1
                BSF     STATUS, RP0
                bcf     INTCON,INTF             ;make RB0/INT flag=0->interrupt didnt occur
toggle_5        BCF     STATUS, RP1             ;Select bank 1
                BSF     STATUS, RP0
                btfsc   OPTION_REG,INTEDG  ;toggle the bit, if it was falling, become rising
and vice-verse.
                goto    tozero_5
                goto    toone_5

tozero_5        bcf     OPTION_REG,INTEDG
                goto    continue_5

toone_5         bsf     OPTION_REG,INTEDG  ;Interrupt on rising edge
                goto    continue_5

continue_5      BCF     STATUS, RP1             ;Select bank 0
                BCF     STATUS, RP0
                btfsc   INTCON, INTF            ;check the RB0/INT flag
                goto    check_bit_5a
                goto    continue_5

check_bit_5a    BCF     STATUS, RP1             ;Select bank 0
                BCF     STATUS, RP0
                bsf     PORTB,6
                btfsc   PORTC,0                         ;check for signal
                goto    ON
                goto    OFF

ON              movlw   0x10
                movwf   PORTB                   ;switch On the Device
                goto    check_zero_crossing

OFF             movlw   0x00
                movwf   PORTB
                goto    check_zero_crossing
                END
```

Appendix K: Gantt Chart

ID		Task Name	Duration	Start	Finish
1		**GATHERING INFORMATION**	**10 days**	**Mon 2/19/07**	**Fri 3/2/07**
2		Conduct the research	5 days	Mon 2/19/07	Fri 2/23/07
3		Literature review	10 days	Mon 2/19/07	Fri 3/2/07
4		**System design**	**7 days**	**Mon 3/5/07**	**Tue 3/13/07**
5		Finalize System Requirements	2 days	Mon 3/5/07	Tue 3/6/07
6		Create a Flow Chart	2 days	Wed 3/7/07	Thu 3/8/07
7		Draw Schematics	3 days	Fri 3/9/07	Tue 3/13/07
8		**Module 1: Transformerless Power**	**17 days**	**Wed 3/14/07**	**Thu 4/5/07**
9		Requirements	4 days	Wed 3/14/07	Mon 3/19/07
10		Design	4 days	Tue 3/20/07	Fri 3/23/07
11		Implementation	2 days	Mon 3/26/07	Tue 3/27/07
12		Testing	7 days	Wed 3/28/07	Thu 4/5/07
13		**Module 2: Signal Extractor**	**18 days**	**Fri 4/6/07**	**Tue 5/1/07**
14		Requirements	4 days	Fri 4/6/07	Wed 4/11/07
15		Design	4 days	Thu 4/12/07	Tue 4/17/07
16		Implementation	2 days	Wed 4/18/07	Thu 4/19/07
17		Testing	8 days	Fri 4/20/07	Tue 5/1/07
18		**Module 3:Signal Generator**	**20 days**	**Wed 5/2/07**	**Tue 5/29/07**
19		Requirements	5 days	Wed 5/2/07	Tue 5/8/07
20		Design	3 days	Wed 5/9/07	Fri 5/11/07
21		Implementation	2 days	Mon 5/14/07	Tue 5/15/07
22		Testing	10 days	Wed 5/16/07	Tue 5/29/07
23		**Module 4: Zero-Crossing Detector**	**10 days**	**Thu 7/19/07**	**Wed 8/1/07**
24		Requirements	5 days	Thu 7/19/07	Wed 7/25/07
25		Design	3 days	Thu 7/26/07	Mon 7/30/07
26		Implementation	2 days	Tue 7/31/07	Wed 8/1/07
27		Testing	10 days	Thu 7/19/07	Wed 8/1/07
28		**Module 5:TRIAC**	**12 days**	**Thu 8/2/07**	**Fri 8/17/07**
29		Requirements	3 days	Thu 8/2/07	Mon 8/6/07
30		Design	2 days	Tue 8/7/07	Wed 8/8/07
31		Implementation	2 days	Thu 8/9/07	Fri 8/10/07
32		Testing	5 days	Mon 8/13/07	Fri 8/17/07
33		**System Integration & Testing**	**15 days**	**Mon 8/20/07**	**Fri 9/7/07**
34		System Integration Testing	15 days	Mon 8/20/07	Fri 9/7/07
35		**Completing final report**	21 days	Mon 9/3/07	Mon 10/1/07