REMOTE CONTROL TIME DELAY SWITCH WITH SENSOR

Oleh

KHOO WELIX

REMOTE CONTROL TIME DELAY
SWITCH WITH SENSOR

Oleh

KHOO WELIX

Disertasi ini dikemukakan kepada
UNIVERSITI SAINS MALAYSIA

Sebagai memenuhi sebahagian daripada syarat keperluan
untuk ijazah dengan kepujian

SARJANA MUDA KEJURUTERAAN (KEJURUTERAAN ELEKTRONIK)

Pusat Pengajian Kejuruteraan
Elektrik Dan Elektronik
Universiti Sains Malaysia
Mac 2005

ABSTRACT

The infrared remote control is commonly seen around the house nowadays. Its usage can be seen ranging from TV, video player, CD player and so on. Recently, even the fans have employed the usage of infrared remote control. However, all of these infrared remote controls can only control one and only one electrical appliances at once. Thus, it will be more useful to design an infrared remote control learning switch that has the ability to learn the code from different infrared remote control sets and perform switching operations when data are sent. This project is dedicated to implementing switching using any infrared remote control in the market. It has a learning ability which makes it user friendly and eliminates the restriction whereby the user needs to use a dedicated infrared remote control. The user can use any infrared remote control to control four of the switches. Apart from that, by using a light sensor, implemented using a light dependant resistor (LDR) it can also scan the light intensity from the environment and turn on or off a light as needed. The amount of light intensity that is needed to turn off the lights can also be adjusted from the light sensor via a variable resistor. This gives the user freedom to adjust the light sensor to suit their needs. The design also comes with a door sensor where when activated, will sense the number of people going into a room. Once the first person enters the room, it will turn on the light. And when the last person exits the room, the lights will turn off automatically. Although this project is discussed mainly using lights, in practice, it can be used with any loads and have been tested with both inductive and resistive loads. The design is controlled using the Microchip PIC 16F84A microcontroller. The programming is done using the PIC microcontroller assembly language and the hardware used to flash the hex code into the microcontroller is the F84 programmer. The program used to flash the hex code into the microcontroller is Nigel Goodwin's WinPICProg version 1.91. As for the switching part, triac switch and relay switch are also discussed here.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I have been extremely fortunate in having the help of many throughout the course of pursuing the Bachelor Degree in Electronics Engineering at University Science Malaysia. I really have to emphasize that without their support and encouragement, the preparation of this dissertation would be impossible.

First of all, I would like to repeat my deepest gratitude to my advisor, Puan Chean Kooi Chyi for her thoughtful comments, valuable suggestions, indepth discussions, and continuing guidance, which allowed me to produce this dissertation in its current version. To be her student is my pleasure and blessing.

I am also particularly grateful to my very helpful course mate, Mr. Ng Jiun How who have helped me throughout my year in University Science Malaysia.

Finally, I would like to thank my mother and sister. Without their encouragement, support, patience and understanding, I would never have completed this dissertation.

Khoo Welix
University Science Malaysia
Engineering Campus
Feb 2005

CHAPTER 1
INTRODUCTION

The infrared remote control is commonly seen around the house nowadays. Its usage can be seen ranging from TV, video player, CD player and so on. Recently, even the fans have employed the usage of infrared remote control. However, all of these infrared remote controls can only control one and only one electrical appliances at once. Thus, it will be more useful to design an infrared remote control learning switch that has the ability to learn the code from different infrared remote control sets and perform switching operations when data are sent.

1.1 Project Objective

The objective of this project is to design a remote control time delay switch. As the title suggest, the design will be able to control the conventional switch. The time delay feature is to enable the switch to automatically switch of in case the user forgets to switch it off.

1.2 Project Research

Infrared remote control data format varies from type to type. Among the commonly used format are REC80, NEC, DENON, SIRCS, RC5, MOTOROLA, JAPAN, SAMSUNG and DAEWOO. Because there are so many types of format out there, it will be quite impossible to incorporate all the format types into a single infrared remote control. So, in order to make a universal infrared remote control switch, I have decided to enable the design to learn up the code instead.

The core unit of this project will be the Microchip PIC 16F84A microcontroller. This microcontroller will be programmed to learn up the code from different infrared remote control. And when the buttons for the learned codes are pressed again, the microcontroller will toggle a switch. The switch can be connected to any load (resistive or inductive) so the user will now convert the usual manual operation of flipping a switch into wireless infrared switching.

As for the lights sensor, we will be using a light dependant resistor to implement it. Finally, for the door sensor, an infrared transmitter and receiver pair will be used.

1.3 Project Overview

The infrared remote control learning switch is design to turn our house's normal lights switches into wireless infrared switch. Though we will be concentrating more on controlling lights here, its application is not limited to only these. The infrared remote control learning switch can be used to replace all the manual switches in the house. The designed has 3 different functions. Those are infrared data learning function, light sensor function and door sensor function. This project is done using the MicroChip PIC 16F84A microcontroller and has the ability to control 4 switches.

The infrared data learning function enable the user to use any type of infrared remote control to control the switching. Firstly, user will initiate the learning process by pressing a dedicated learning button. Then using any infrared remote control, the user will press the infrared remote control button and the code will be saved into the microcontroller. After which the learning process is done, the dedicated switch can be controlled through the infrared remote control button that has been learned.

As for the light sensor, a circuit that consists of light dependant resistor and comparator will be used. The sensitivity of the light sensor can also be adjusted to suit the users' need. The function of the light sensor here is to check the intensity of the light in the surrounding and if the light sensor switch is turned on, then it will respond by turning on that switch when the light density is low.

Lastly, the door sensor is used to enable the microcontroller to detect movement from the door. The door sensor is realized using two infrared transmitter and receiver pair. The door sensor is direction sensitive. This is very important because the microcontroller needs to know whether the user is walking into or out of a room so that it will be able to turn on or off a respective switch.

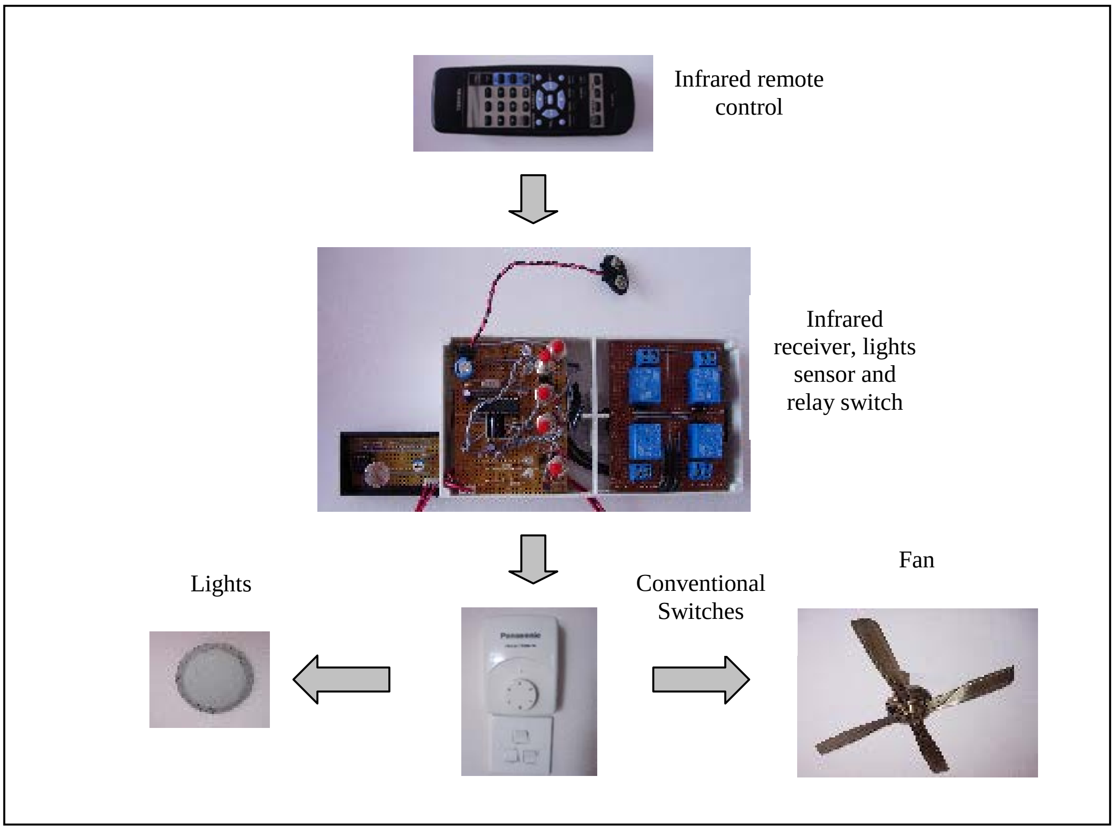

Figure 1.1 Overall view of the project

1.4 Project Commence

In this project, I will be using the PIC assembly language to write the program for the PIC16F84A microcontroller. The whole project can be divided into a few major parts such as:

- Bridge rectifier and voltage regulators to provide DC voltage needed throughout the project
- Nigel's F84 PIC programmer to be used for programming the PIC16F84A microcontroller
- Vishay 38kHz infrared receiver and demodulator module used for receiving data from infrared remote control
- Light sensor using light dependant resistor (LDR) and a comparator

- Door sensor using two infrared transmitter and receiver pair
- Optoisolator triac driver for driving the triac
- Triac switch for switching resistive loads
- Relay switch for switching both inductive and resistive loads

1.5 Report Guideline

The following shows the summary for each chapter contained in the report:

- **Chapter 1**

  The objective of the project, scope, and how the project is carried out will be explained in this chapter. A guideline to this report is also shown here.
- **Chapter 2**

  The microcontroller PIC16F84A will be discussed here along with the Nigel F84 programmer. All information on PIC16F84A ranging from its data memory organization and I/O ports will be explained there.
- **Chapter 3**

  This chapter will discuss briefly about the operation of infrared remote control. A brief explanation of how the learning process of this design is also shown here.
- **Chapter 4**

  The hardware design of the project is discussed here. All the circuits used are explained in details.
- **Chapter 5**

  We look in detail the programming used for the PIC microcontroller here. All the important subroutines used to control the hardware are explained here.
- **Chapter 6**

  Installation and operation of the device is shown here. Details on how to connect the device is also shown.
- **Chapter 7**

  Conclusion of the project is discussed here. Further expansion of this project is also discussed.

CHAPTER 2
MICROCHIP PIC16F84A MICROCONTROLLER

The PIC16F84A is an 18 pins enhanced FLASH/EEPROM 8-bit microcontroller. It only has 35 single word instructions and all instructions are single-cycle except for program branches which are two-cycle. The operating speed of the microcontroller depends on the crystal oscillator used. Each instruction takes four clock cycles. For a crystal oscillator with frequency, *f* Hz, the time taken to execute each instruction will be calculated as below:

$$\text{Time taken to execute one instruction} = \frac{1}{f} \times 4$$

So for a 4MHz crystal oscillator, the time taken to execute one instruction will be 1µs. However, as stated earlier, all instructions are single-cycle except for program branches which are two-cycle. Thus, the time take to execute a branch instruction will be 2µs.

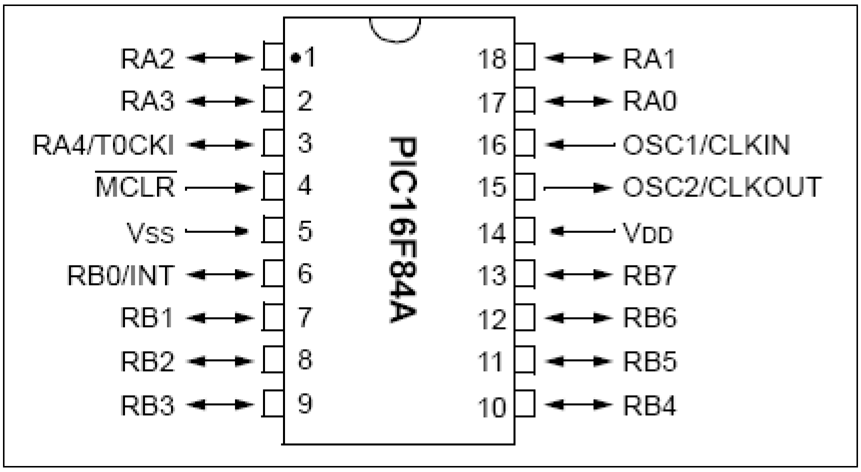

Figure 2.1 PIC16F84A pinout

2.1 Pin Description

PIC16F84A is an 18 pin microcontroller. There are 13 I/O pins that are user-configured on a pin-to-pin basis. Some pins are multiplexed with other device functions. These functions include:

• External interrupt

• Change on PORTB interrupts

• Timer0 clock input

The table below details the pinout of the device with descriptions.

Table 2.1 PIC16F84A pinout descriptions

Pin Name	Pin No	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	I	ST/CMOS(3)	Oscillator crystal input/external clock source input
OSC2/CLKOUT	15	O	-	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{MCLR}$	4	I/P	ST	Master Clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
RA0	17	I/O	TTL	PORTA is a bi-directional I/O port.
RA1	18	I/O	TTL	
RA2	1	I/O	TTL	
RA3	2	I/O	TTL	
RA4/T0CKI	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.

				PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	I/O	TTL/ST(1)	RB0/INT can also be selected as an external interrupt pin.
RB1	7	I/O	TTL	
RB2	8	I/O	TTL	
RB3	9	I/O	TTL	
RB4	10	I/O	TTL	Interrupt-on-change pin.
RB5	11	I/O	TTL	Interrupt-on-change pin.
RB6	12	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming clock.
RB7	13	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming data.
V_{SS}	5	P	-	Ground reference for logic and I/O pins.
V_{DD}	14	P	-	Positive supply for logic and I/O pins.

Legend: I = input O = Output I/O = Input/Output P = Power
- = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

Note 3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

2.2 Data Memory Organization

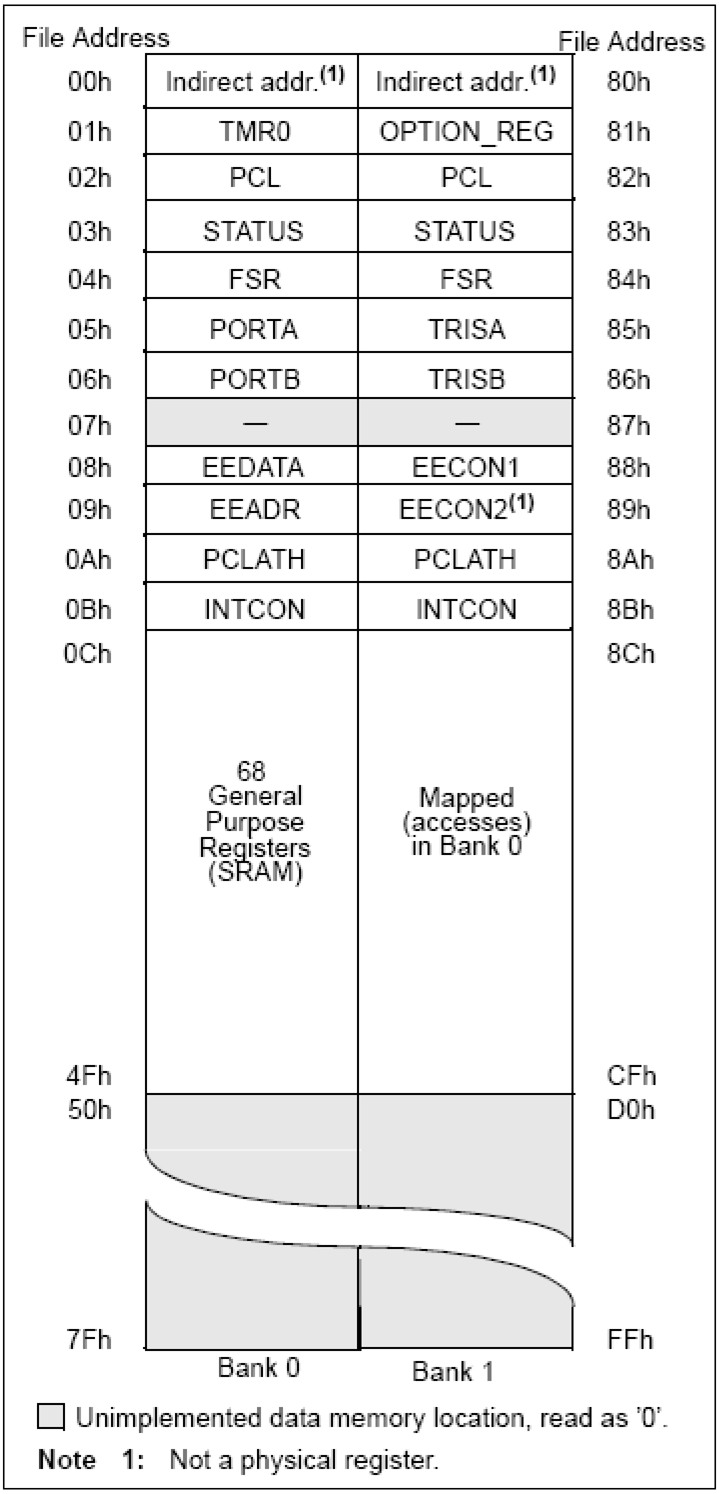

Figure 2.2 Register file map for PIC16F84A

The data memory is partitioned into two areas. The first is the Special Function Registers (SFR) area (Bank 1), while the second is the General Purpose

Registers (GPR) area (Bank 0). The SFR control the operation of the device. Portions of data memory are banked. This is for both the SFR area and the GPR area. The GPR area is banked to allow greater than 116 bytes of general purpose RAM.

The banked areas of the SFR are for the registers that control the peripheral functions. Banking requires the use of control bits for bank selection. These control bits are located in the STATUS Register. **Figure 2.2** shows the data memory map organization. The entire data memory can be accessed either directly using the absolute address of each register file or indirectly through the File Select Register (FSR). Indirect addressing uses the present value of the RP0 bit for access into the banked areas of data memory.

Data memory is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected by clearing the RP0 bit (STATUS<5>). Setting the RP0 bit selects Bank 1. Each Bank extends up to 7Fh (128 bytes). The first twelve locations of each Bank are reserved for the Special Function Registers (SFR). The remainders are General Purpose Registers (GPR), implemented as static RAM.

2.3 PIC16F84A Programmer

To flash the codes written into the PIC, I have used the F84 parallel programmer. This programmer only consists of a minimum amount of component count so it can be assembled quickly. The F84 programmer is compatible with Nigel Goodwin's WPIC16 software that is used to flash the codes in to the PIC chip.

In **Figure 2.3**, the circled portion is the modified part from the original F84 programmer. The crystal oscillator is included into the programmer so that we can run the programmed PIC directly this modified programmer. In addition, two more switches are also added for the user to select whether it is the programming mode or the testing mode. When using either mode, only one switch can be turned on at a time. For example, if we are using the testing mode, then we will turn on the testing switch and off the programming switch.

The switch shown above is implemented using the header and is shorted using a jumper. In either mode, the jumper is inserted so that one of the switches is shorted. The idea is to cut off the +5V to the buffer IC (74LS07) so that it is turned off and that there is no communication between the parallel port to the PIC during testing mode. Besides that, another switch that is turned on during testing enables the PIC to power up and also the $\overline{\text{MCLR}}$ pin to be set into logic '1'. While in programming mode, the buffer IC is powered up and the pin 14 and 4 is controlled by the programmer software.

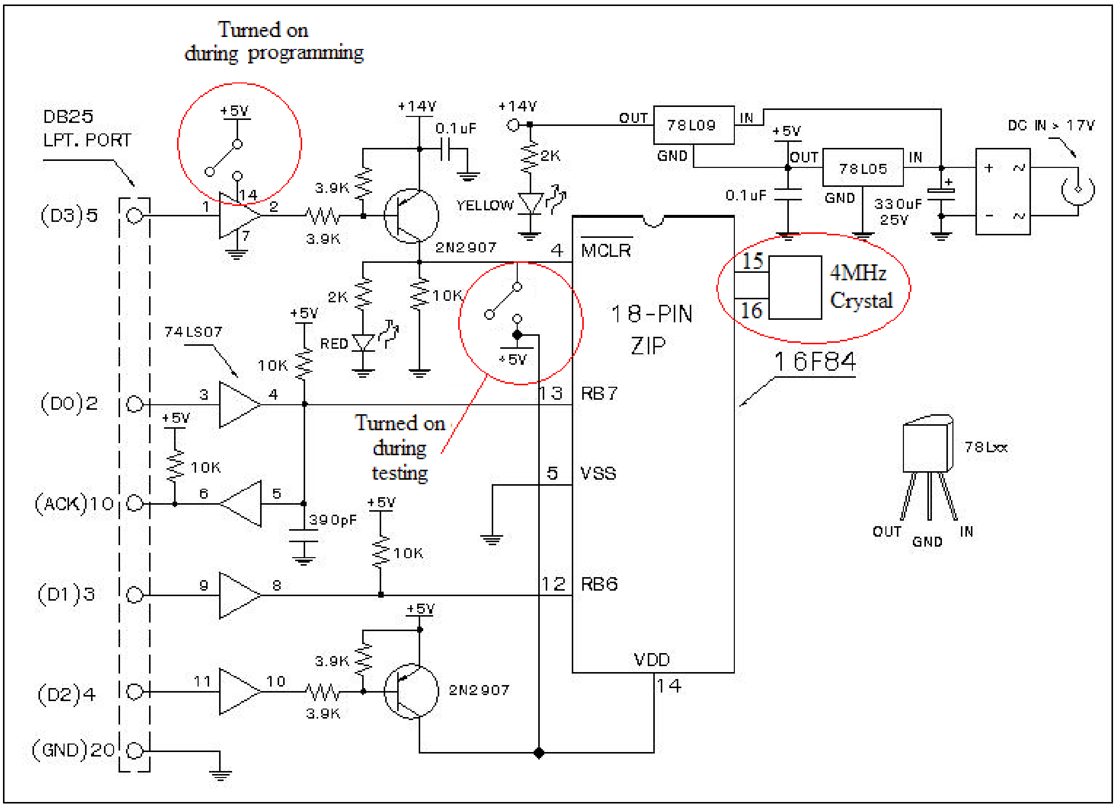

Figure 2.3 Modified F84 programmer

2.4 Nigel Goodwin's WPIC16 Programmer

Compiling of the source code into hex file can be done using the Microchip's Universal Assemble version 3.80. This assemble can be used for all of the Microchip PIC microcontroller type.

The hex codes can then be flashed into the PIC using the F84 hardware through Nigel Goodwin's WinPICProg version 1.91. One very good feature about

Nigel's WinPICProg is that it enables the user to check the programmer's circuit by manually turning on and off certain input and output. This is very good for debugging purposes. The programmer software is shown in the diagram below.

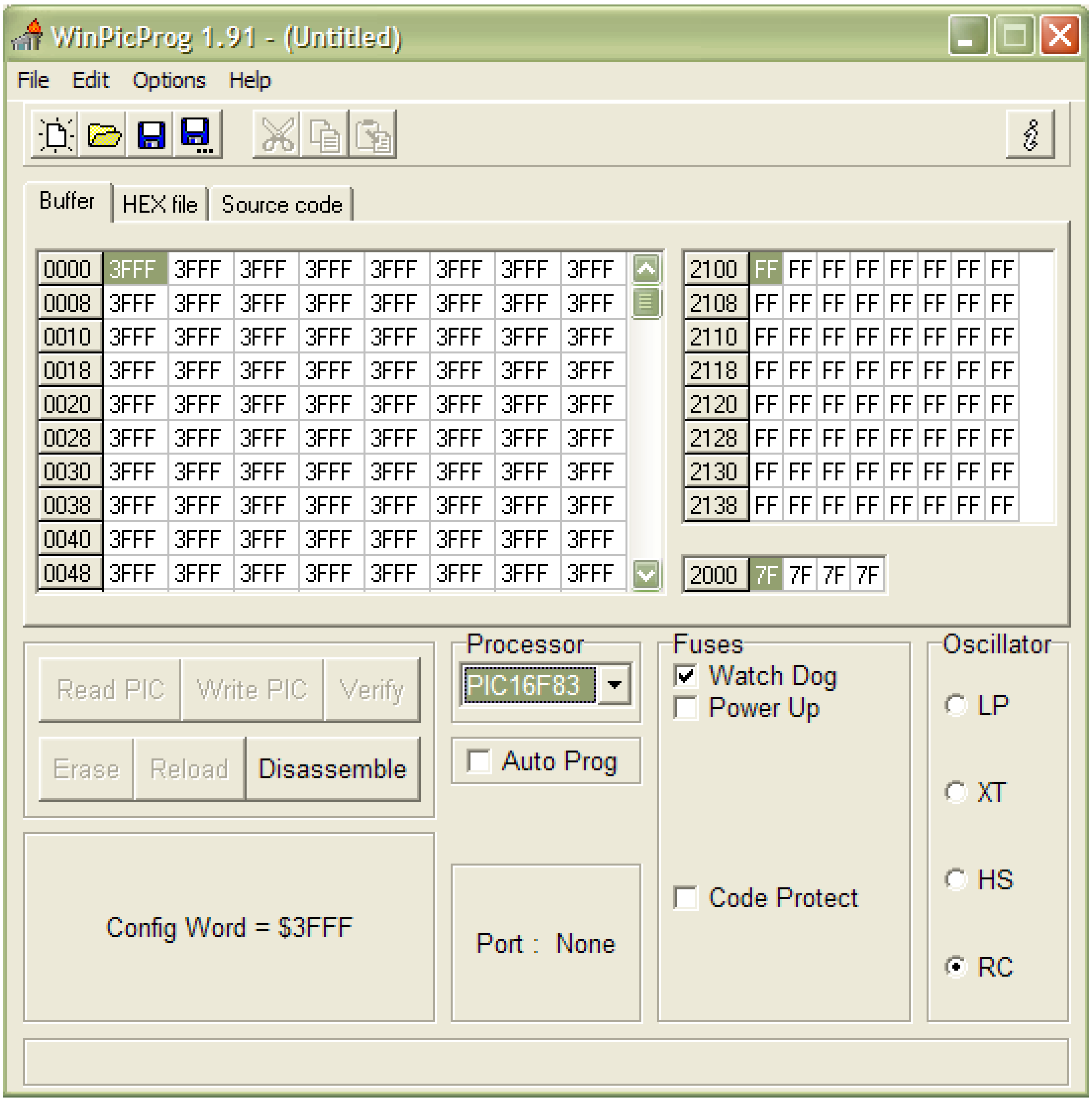

Figure 2.4 WinPICProg's user interface

As can be seen in **Figure 2.4**, all the hex code can be viewed from the buffer area. Here, the hex code is shown with respect to the memory location. Apart from that, this programmer is able to disassemble a given hex code into its equivalent source code.

2.5 Connecting to PIC

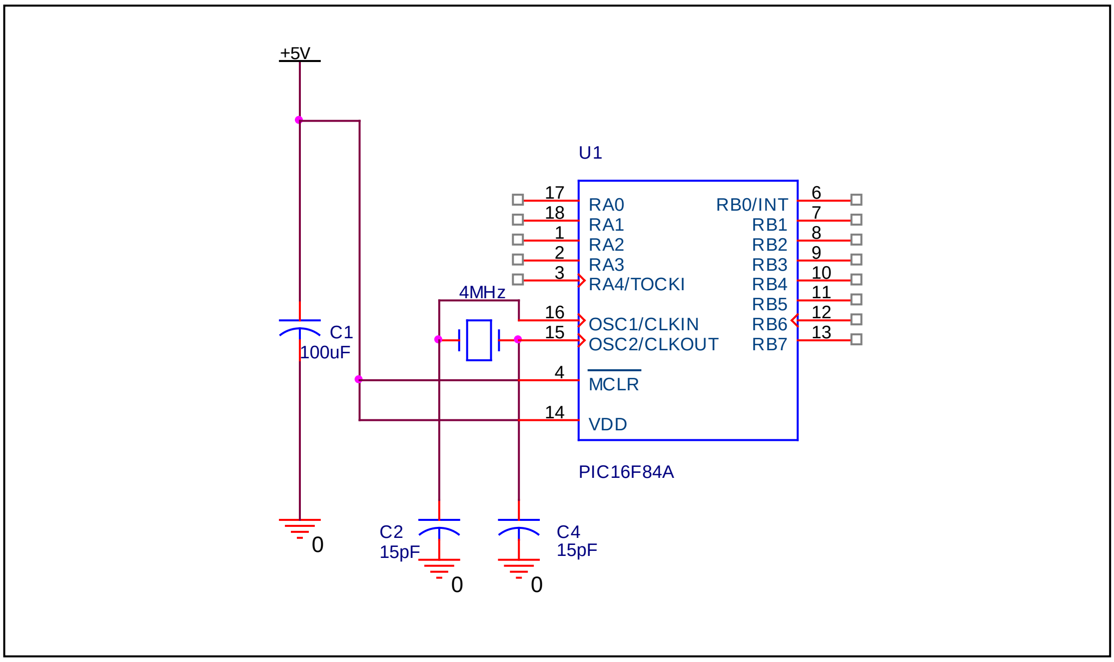

Figure 2.5 PIC microcontroller setup

Figure 2.5 above shows the configuration needed for a working PIC. The 100µF capacitor is known as a bypass capacitor. It is used to reduce any noise supply railing while capacitor C2 and C4 helps to reduce any stay oscillations across the crystal, and get rid of any unwanted noise before the signal goes into the PIC. In this project, we will be using a 4MHz crystal oscillator. A parallel resistor and capacitor circuit can also be used, but the cost of the crystal is negligible, and it is more stable. The advantages of using a 4MHz crystal is that each clock cycle will now be 1µs thus simplifying the timing calculation.

$$T = \frac{1}{4M} \times 4 = 1\mu s$$

Now that the basic PIC connection is made, the user can add in any other circuit at the I/O ports and configure the pins to function as an input or an output port respectively via the TRISA and TRISB register.

CHAPTER 3
INFRARED REMOTE CONTROL

In this project, the user will use any infrared remote control available in the market to control the switching operation. In many other infrared related application developed, there is always the restriction for the user to use back the same infrared remote to control the circuit function. This is because, the controllers programmed in those products are already fixed and will only respond to the data emitted from that particular infrared remote control. This trend limits the user to one specific remote control per application.

Nowadays, there are a lot of infrared control appliances at home, namely, television, VCD player, fan, cassette player, audio player and so on. Each of these appliances has its own remote control. However, not every button in those remote controls needs to be used all the time. So, in this project, the idea is to make use of the existing infrared remote to control the designed multipurpose switching circuit.

3.1 How It Works

All modern IR remote control devices produce a continuous coded stream of pulses at 37.9 kHz when any button on the remote is pressed. These IR pulses are detected and decoded by a receiver (your TV, VCR, etc.) and the appropriate function activated. Because there are a lot of protocols available for different brands, namely, REC80, NEC, DENON, SIRCS, RC5, MOTOROLA, JAPAN, SAMSUNG and DAEWOO, it will be impossible to enable to microcontroller to identify and decoded all of these protocols and retrieve the data.

So, in order to achieve a universal use of infrared remote control, the microcontroller will instead, receive the undecoded data emitted from the infrared remote control and save it into the data memory during its learning operations. And when the user presses the same button again next time, the data received will then be compared to the data saved in the microcontroller memory. If the data for the key pressed is the same as the ones learned, then the switch will toggle.

For all the infrared data protocols, there are certain similarities between all of them. Those similarities between the protocols are listed below:

- Starting bits
- Address bits
- Command bits
- Equipment bits

Note that not all the similarities listed above appear in all the available protocols. Some of the protocols may omit the equipment bits for example. However, all of the protocols have the starting bits and for a certain infrared remote control model, the starting bits for all the buttons will be the same. Thus, we can omit the starting bit in the learning process in order to save up data memory space in the PIC.

In this project, the starting bits to be omitted are fixed at 10ms. This startup delay of 10ms is just an estimation after having taking samples from a few different infrared remote control brands (Sony, Toshiba, Philips). This is because not all of the infrared remote control has the same starting bits length.

For the data to be learned, a total of 120 bits of data will be taken. The interval between each bit saved is 0.5ms. The microcontroller will start to save the data received once the user presses a button. This is because the infrared receiver and demodulator module (TSOP 1138) which the output is normally high will go low when a data is received. Thus, by using the interrupt (negative going transition) features in the PIC, we will be able to start sampling the data once the user presses a button.

CHAPTER 4
CIRCUIT DESIGN

For this project, the software development part is as crucial as the hardware design. Because the software will be controlling the functionality of the microcontroller, it is equally important that the hardware is able to execute the expected operations programmed. The hardware design for this project can be broken down into several parts listed below:

- Voltage regulator
- Infrared receiver
- Learning switch
- Light sensor
- Door sensor
- Relays switch
- Triac switch

4.1 Block Diagram Of The Design

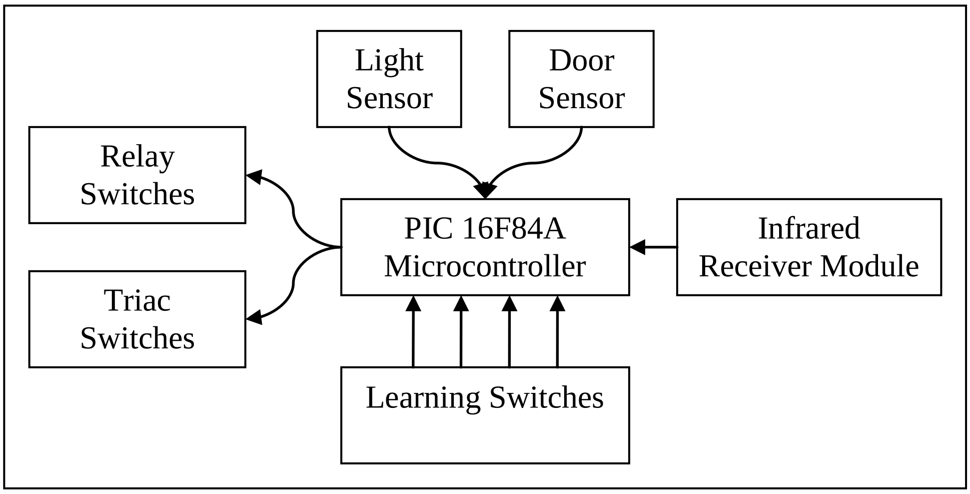

Figure 4.1 Block diagram of the hardware design

As can be seen in **Figure 4.1**, the PIC microcontroller is the core of the design. It will receive inputs from other circuits and sends out the proper output after processing the inputs.

The PIC will receive data emitted from the infrared remote control via the infrared receiver module. This circuit is used to detect infrared data that is modulated at 38 kHz and output the demodulated data.

The learning switch is used to enable to user to set the switch for the learning process. Apart from that it is also used to turn on the light sensor and also the door sensor.

The light sensor consists of a light dependant resistor (LDR) and a comparator. Its main function is to respond to a certain light intensity. For example, if the user set the light sensor to go high at 5 lux, it will output a high when the intensity is 5 lux and low when the intensity is less than 5lux. The door sensor is realized using a pair of infrared transmitter and receiver pair. Its main function is to enable to microcontroller to scan if there is people moving in or out of a room.

Lastly, the triac switch is used for switch resistive load while the relay switch can be used for both inductive and resistive loads. The microcontroller is used to trigger the triac and relay switches when ever data is received from the infrared receiver module, light sensor or the door sensor.

4.2 Voltage Regulator

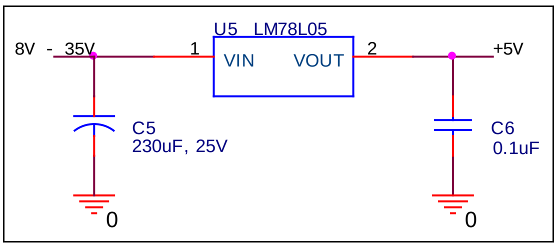

Figure 4.2 Schematics for the voltage regulator

In order to power up the circuit in this project, we will need a +5V. To get a regulated +5V, a voltage regulator is used. The voltage regulator is a three terminal device. That is input, ground and output. The input can range from a minimum of 8V up to a maximum of 35V. Both capacitor C5 and C6 is used to smoothen the input and output voltage.

4.3 Infrared Receiver

The VISHAY TSOP 1138 infrared receiver and demodulator module is used for the infrared receiver part. This module will only receive data modulated at 38 kHz. In the configuration shown in **Figure 4.3** below, the capacitor C3 and resistor R1 is used to suppress power supply disturbance. The output of the infrared receiver is connected directly to the input pin of the PIC.

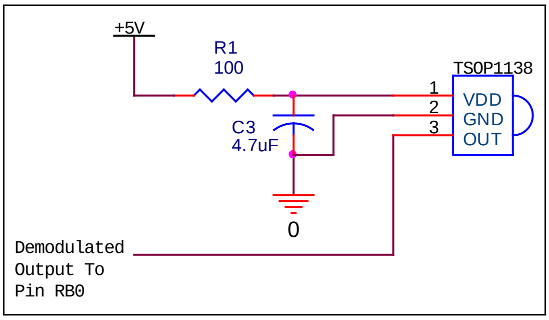

Figure 4.3 Configuration for the infrared receiver and demodulator module

The output of the receiver module is active-low. In the software, we make use of the PIC interrupt to detect the falling edge from the infrared receiver module. By using the interrupt feature, we do not have to keep on looping and scanning for a negative edge. The interrupt feature enables the microcontroller to execute an interrupt service routine whenever it receives a falling edge

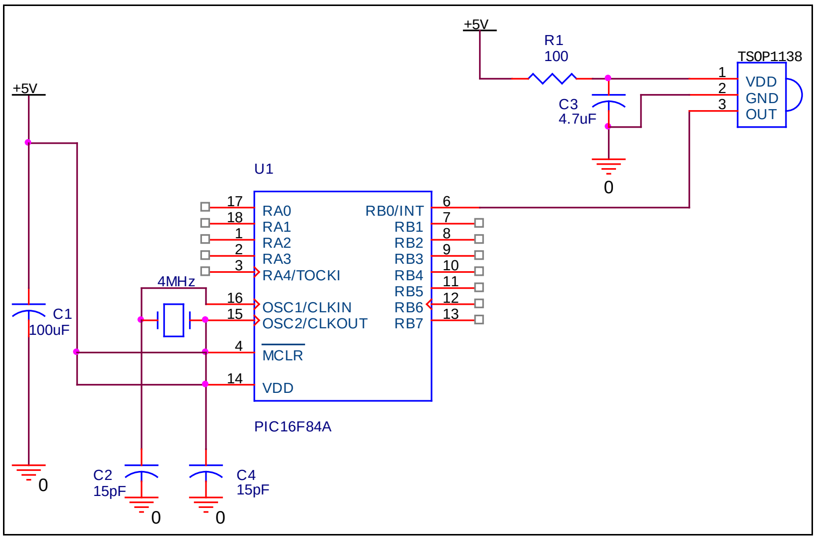

Figure 4.4 Connection between the infrared receiver module and the PIC microcontroller

4.4 Learning Switch

The purpose of the learning switch is to enable the user to activate the learning process for a particular switch. In this project, we will be triggering 4 switches (triac or relay). **Figure 4.5** below shows the learning switch (LS1 – LS4) and also the door switch DS1.

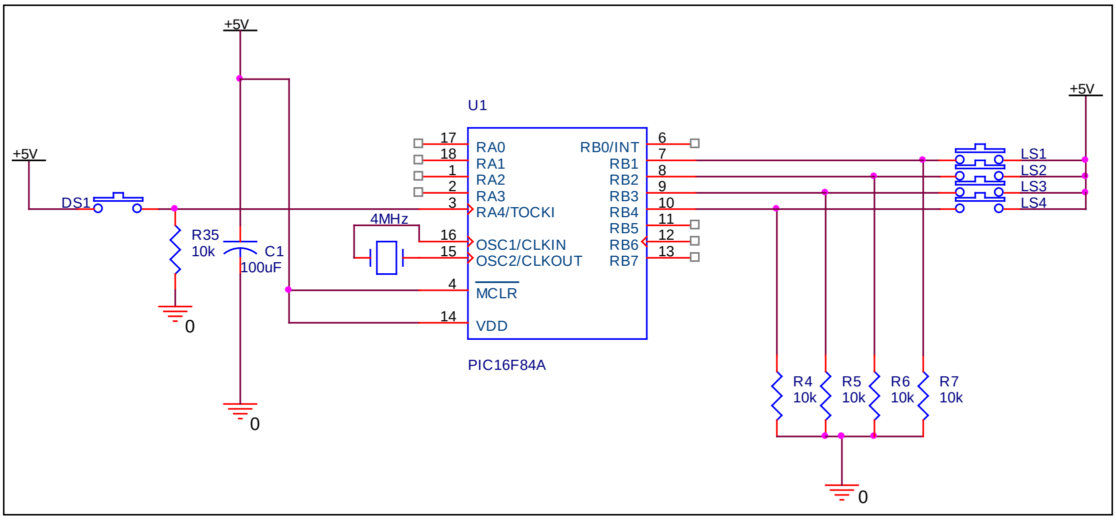

Figure 4.5 Connection between the learning switches and PIC microcontroller

The learning switches LS1 – LS4 are used when the user wants to activate the learning process for any one of the 4 switches. To activate the infrared learning process for a specific switch, the user needs to press the learning button for that particular switch. Next, use the infrared remote control and press the desire button. The infrared receiver module will receive the data and the PIC microcontroller will save the received data into its data memory.

To activate the light sensor, user will need to press the learning switch (LS1 – LS4) twice. For example, if we wish to activate the light sensor on switch 1, we just need to press the LS1 button twice. Lastly, the door sensor is used to activate a particular switch to respond to the door sensor. To activate the door sensor to a switch, the user needs to press the door sensor switch (DS1) and then press the correspondent learning switch (LS1 – LS4) in which he wishes to activate the door sensor.

In **Figure 4.5**, all the input ports are pulled down with a 10kΩ resistor. This is to avoid floating pins in the PIC.

4.5 Light Sensor

The light sensor is realized using the light dependant resistor and a comparator. In this project, the NORP12 light dependant resistor (LDR) is used. This device acts like a resistor in the circuit. However, its value of resistance will change depending on how much light is falling on it. The resistance of the NORP12 varies inversely proportional to the amount of light falling on it.

- In no light the resistance is ≈ 1MΩ
- With 10 lux falling on it the resistance is ≈ 9kΩ
- With 1000 lux falling on it the resistance is ≈ 400Ω

(Note: **lux** is the metric unit of light intensity)

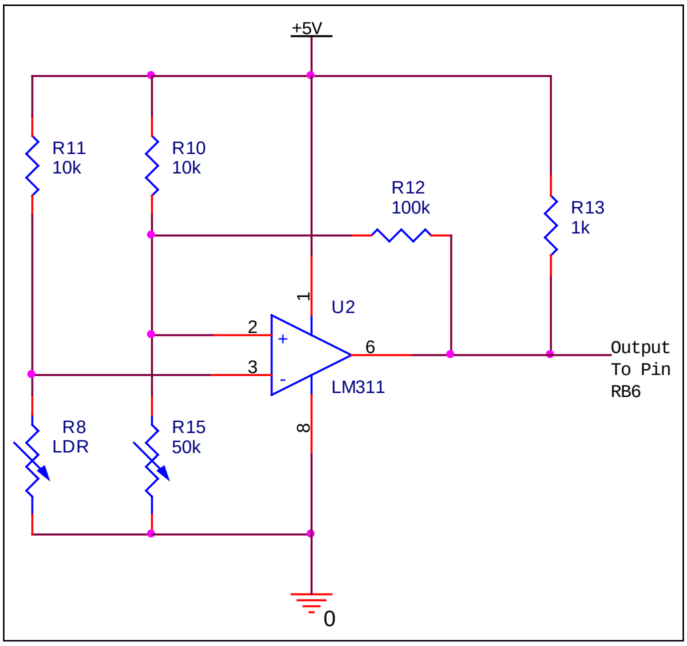

Figure 4.6 Realizing the light sensor using comparator and LDR

Figure 4.6 shows how the light sensor is realized using the comparator and the LDR. It that circuit we can see 2 voltage dividers. The first voltage divider consists of a 10kΩ resistor and a LDR while the second consist of a 10kΩ resistor and a 50kΩ variable resistor. The purpose of the 50kΩ variable resistor voltage divider is to be the reference voltage to the LDR voltage divider. The 50kΩ resistor can be adjusted so that the compared voltage differs with the resistance on the 50kΩ resistor.

To perform this comparison, the comparator is used. A comparator acts like an op-amp with infinite gain.

- If V+ > V- then V_{out} ≈ +ve supply voltage (in this case +5V)
- If V+ < V- then V_{out} ≈ -ve supply voltage (in this case 0V)

In **Figure 4.6**, the 1kΩ resistor R13 is actually a pull-up resistor. This is because the LM311 comparator has an open collector output. Assuming that the variable resistor is set at 10kΩ, the output of the circuit in **Figure 4.6** will be:

- ≈ 5V if the resistance of the LDR is <10kΩ (i.e bright light)
- ≈ 0V if the resistance of the LDR is >10kΩ (i.e low light)

Going back to **Figure 4.6** again, we will see the 100kΩ feedback resistor R12. The purpose of this feedback resistor is to incorporate hysteresis into the circuit. Assuming that the circuit is design so that the 50kΩ variable resistor is fixed at 10kΩ and that the feedback resistor is fixed at 150kΩ, the calculation for the upper threshold and the lower threshold is shown below.

$$\text{Upper threshold, } V_{UTH} = \frac{10k}{10k + 10k \| 100k} \times 5V = 2.62V$$

$$\text{Lower threshold, } V_{LTH} = \frac{10k \| 100k}{10k + 10k \| 100k} \times 5V = 2.38V$$

Thus, if we make the feedback resistance smaller, then the gap between V_{UTH} and V_{LTH} will increase.

Assuming that the comparator is in the high output state (i.e. the resistance of the LDR is << 10kΩ) hence V- < V+ and the upper threshold will apply. Now if the resistance of the LDR starts to increase (because less light is falling on it) the V- will also increase. Once V- is just > 2.62V, the output will change to a low output, and thus the threshold will shift to the lower value (2.38V), in effect locking in the changes as V- will have to drop to < 2.38 V to change the output back again. The reverse process applies when the illumination increases: the LDR resistance and V- will decrease, and once the comparator has shifted back to the high state the threshold will move to the high value, locking in that change. Thus, hysteresis acts to minimize the possibility of comparator output oscillation caused by minor variations in the input voltage.

Figure 4.7 below shows the connection between the light sensor and the PIC microcontroller.

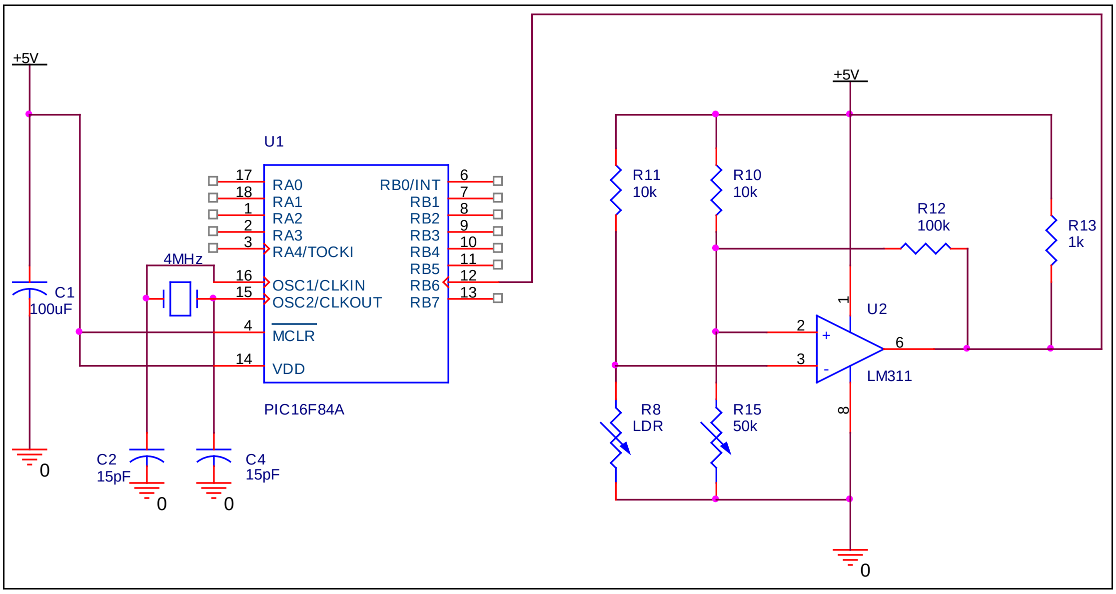

Figure 4.7 Connection between the light sensor and the PIC microcontroller

4.6 Door Sensor

Another feature of this design is the door sensor. In this project the door sensor is implemented using a pair of infrared transmitter and receiver. The block diagram of the idea is shown in **Figure 4.8** below.

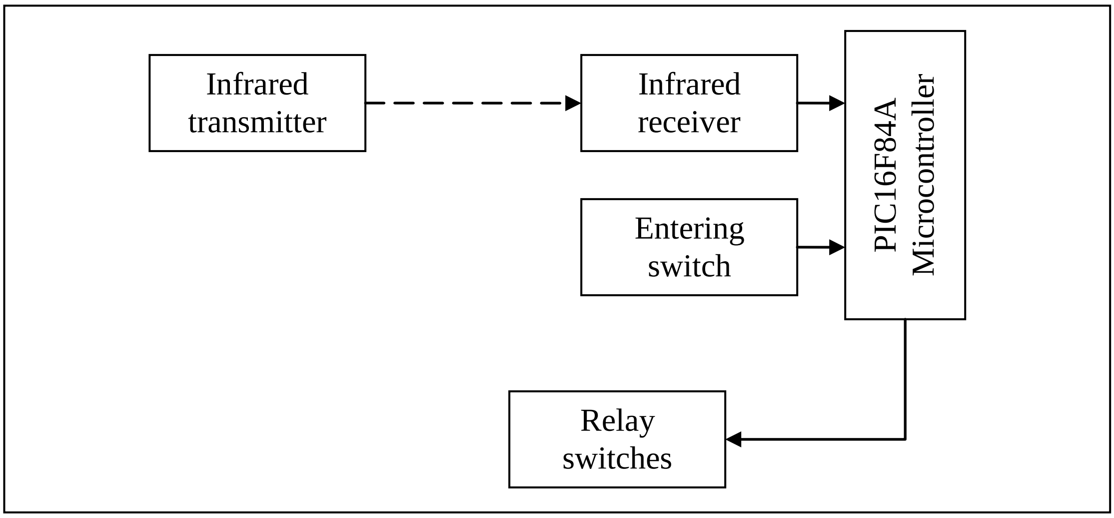

Figure 4.8 Block diagram for the door sensor

The infrared transmitter and receiver is placed facing each other on the door. The idea is for the user to cut of the infrared transmission when they walk pass the door (receivers). As we can see from **Figure 4.8**, there is an entering switch. The main purpose of the entering switch is for the user to press before entering the room. In this project, we will be using a normal switch, however in practical situation, the entering switch can varies according to need. For example, if we were to implement it in the lecture hall, the entering switch might be a card sensor. Before a student enters into a lecture hall, he will need to swipe the card sensor to take his attendance and also to turn on the lights in the hall. In other situation, the entering can also be a second infrared transmitter and receiver pair. This way, when the user walks into a room, it will trigger the lights on and when he walks out, he will trigger the lights off.

The infrared receiver and transmitter pair in **Figure 4.8** is used to scan for the number of people going out of the room. The microcontroller will count down by one if the infrared receiver senses that a person walks through the door. And the relay switch will turn off once the number of people in the room is zero.

Figure 4.9 shows the schematic used to realize the infrared transmitter. The timer 555 IC is used to get the required 38 kHz modulated output. The frequency output that we desire can be adjusted using the 5kΩ variable resistor R41. In the schematic, there are 2 pairs of infrared LED. Each pair is used to implement the infrared transmitter 1 and 2 shown in the block diagram to extend the range of the infrared data.

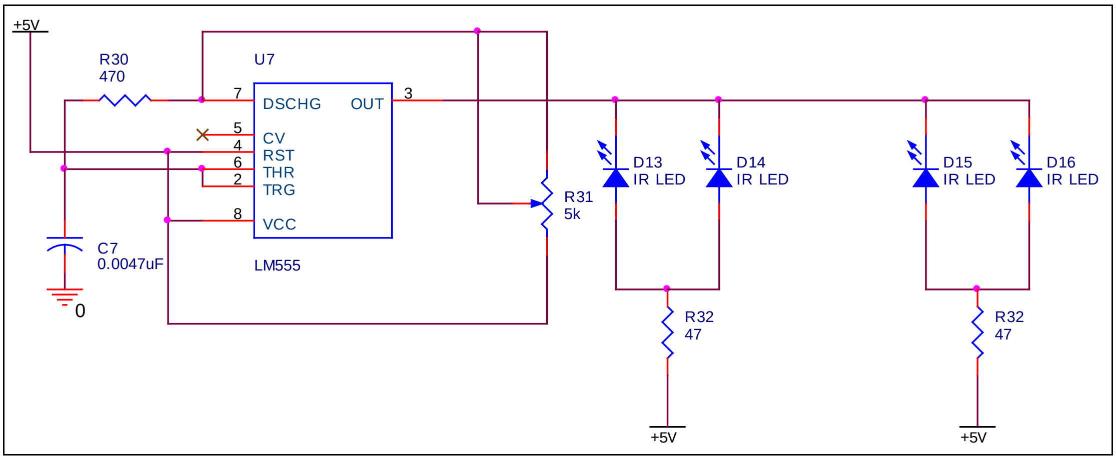

Figure 4.9 Infrared transmitter using IC timer 555

The infrared receiver is implemented using the IS1U60 infrared receiver and demodulator module. This SHARP receiver is used instead VISHAY TSOP 1138 because unlike the VISHAY receiver, IS1U60 is able to output a constant LOW while the VISHAY receiver module can only output a constant LOW for a maximum of 1.8ms.

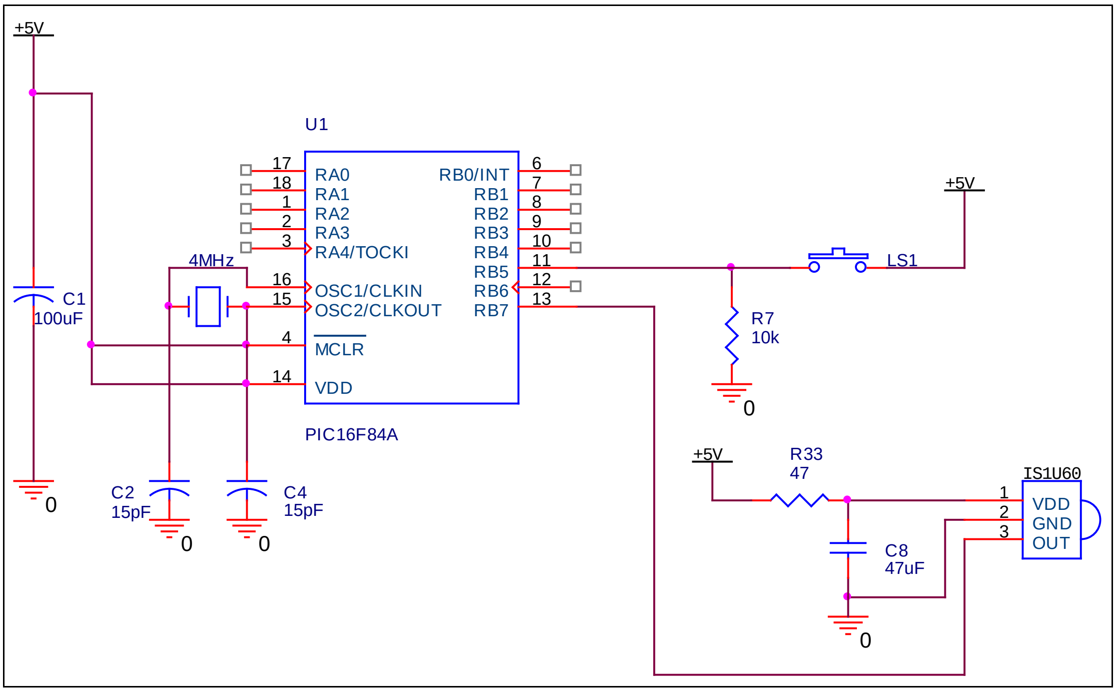

Figure 4.10 Connection between the entering switch, infrared receiver (SHARP IS1U60) and the PIC microcontroller

4.7 Relay Switch

The relay switch is used to replace conventional manual switches here. In this project, two types of switching methods are proposed, the triac switch and the relay switch. The advantage of the relay switch is the simplicity of the circuit. A relay is an electromagnetic switch. The electromagnetic relay consists of a multi-turn coil, wound on an iron core, to form an electromagnet. When the coil is energized, by passing current through it, the core becomes temporarily magnetized. The magnetized core attracts the iron armature. The armature is pivoted which causes it to operate one or more sets of contacts. When the coil is de-energized the armature and contacts are released.