INTELLIGENT CAR PARK SYSTEM

Oleh
Saw Qher Shynn

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

ABSTRAK

Objektif utama projek ini adalah untuk mereka satu sistem pintar tempat letak kereta berdasarkan komputer dan Mikro Pengawal. Prototaip ini dibina mengunakan satu model untuk stimulasi tempat letak kereta yang sebenar. Bahasa peringkat tinggi seperti Visual Basic 6 boleh digunakan untuk membina satu system dengan menyenangkan.

Konsep asas projek ini adalah serupa dengan tempat letak kereta yang normal, tetapi sistem pintar tempat letak kereta ini telah tertingkat dengan beberapa sifat yang berguna seperti mengesan bilangan kereta yang memasuki tempat letak kereta, bilangan kereta yang meletak kereta secara haram, dan bilangan tempat letek kereta yang masih sedia ada. Semua signal akan diproses terlebih dahulu oleh PIC16F877A mikro pengawal dan kemudian dihantar ke komputer. Perkakasan utama projek ini terdiri daripada satu PIC16F877A mikro pengawal daripada syarikat Microchip Technology Inc, penderia LDR dan IC Penyongsang 7404 Hex. Penderia LDR diletakkan pada tempat masuk dan setiap tempat letak kereta untuk mengesan kereta. Kod punca PIC mikro pengawal dibangunkan dengan menggunakan MPLAB Integrated Development Environments manakala antara muka sistem tempat letak kereta ini dibina dengan menggunakan Microsoft Visual Basic 6.

Kebaikan projek ini adalah semua maklumat disegar-semula dari semasa ke semasa untuk memberi pandangan yang jelas tentang tempat letek kosong kepada pengguna. Ini akan menjimatkan masa pengguna. Selain itu, kos pembinaan dan pengurusan adalah rendah. Projek ini juga direka untuk menyenangkan pengawalan sebarabg aktiviti dalam tempat letek kereta.

ABSTRACT

The main objective of this project is to design an intelligent car park system based on computer and Micro controller. This prototype is build using a model to stimulate the actual car park system. High-level language such as Visual Basic 6 is chosen because it can be used to build a convenient system. By using Visual Basic 6, a user-friendly Graphic User Interface can be created.

The basic concept of this project is similar to that of a normal car park, but it has been enhanced with various useful features such as number of cars enters, number of cars illegally parked and number of available lots. All signals will be first processed by the PIC16F877A micro controller and then sent by it to the computer. The main hardware consists of a PIC16F877A micro controller by Microchip Technology Inc, LDR sensors and 7404 Hex Inverter ICs. LDR sensors are placed at entrance and every parking lot to detect cars. The source code of the PIC micro controller is developed using MPLAB Integrated Development Environments while the interface of the car park system is created using Microsoft Visual Basic 6.

The main advantages of this project are information is carried out from time to time to give a clear view of empty lots available for drivers. Time consuming can be reduced to the lowest level. Besides that, the cost required to build and manage this intelligent car park system is low so that any company or shopping centre able to possess. Any inconvenience during finding parking lot can be solved easily. This project also designs for capable of guardhouse monitoring activities inside the car park.

ACKNOWLEDGEMANT

First of all, I would like to convey my heartiest appreciation and express my sincere thanks particularly to my supervisor Pn. Chean Kooi Chyi for her precious advice, guidance, suggestion and co-operation throughout this project. Under her guidance, I have developed, improved and achieved the completion of the project.

Besides that, I would also like to thank Dr. Mohd Yusoff Manshor, Electronic Programme Chairman for giving me the opportunity to handle this project as well as his encouragement. I would also like to take this opportunity say thank to Encik Mohd. Naim Abdullah, Electronic Lab Research Officer for his allowance to use the lab to test me circuit as well as his co-operation and help. Furthermore, my thanks go to my friend who used to be my senior, Mr. Lee Peng Chee for his valuable knowledge of PIC micro controller and understanding of using parallel port with me. I would also like to thank my friends and course mates for their co-operation, support and suggestion to complete this project successfully.

Finally yet importantly, I would like to express my gratitude to both of my parents for their affectionate encouragement, moral support and financial support that they have given to me all the while in doing this project.

SAW QHER SHYNN

(2004 / 2005)

TABLE OF CONTENTS

LIST OF FIGURES

Page

CHAPTER 1:

CHAPTER 2:

LIST OF TABLES

Page

Chapter 1

INTRODUCTION

1.1 What Is Car Park?

Car park is an area of which allow vehicle park inside a specific building or location with automatic system. Normally car parks have at least two passageways: one is entrance and another is exit. For the barrier arms of the car park, it protects the boundary to restrict cars access to the entry and exit points dictate.

An automatic system allow vehicle enter and exit without human control. A security system can be added into the car park system to ensure all vehicle users are safe and sound all the time.

1.2 Overview Of The Present Car Park System

Traffic problem is a big problem in Malaysia. Many cost and effort had been put into to solve the problem. The cause of traffic problem occurs is due to searching parking lot. Many drivers know that it is difficult to find parking lot during peak hours.

Nowadays, there are many different type of car park system used and only having one purpose is to give the facilities of time and labour saving. Many of the car park system in Malaysia are fully automated such as Lift Car Park in hotels used due to the limited space. By using the new technology, more advance car park system can be implemented.

Although using new technology follows with many advantages but the major problem is the cost of implementing is very high and may need technology transfer from other country. Thus, we came out with a new way of implementing a convenience and user friendly has been introduced that is ‘Intelligent Car Park System’.

1.3 Objective Of The Project

In Malaysia, as the number of cars increases on the road, the probability of getting a parking space in a car park is getting lower. It is difficult to find an empty parking lot especially during the weekends in a shopping centre or during lunch hour. Driver may go around the car park finding for an empty space. It is time consuming and inconvenience especially in rushing hour. Besides that, traffic jam will occur while driver is searching for empty space. At times from far away the drivers think they see an empty parking lot, but when they get near to it, it is not really empty as there is a small car parked there.

Therefore, this project is implemented with the intention of facilitate the driver. Besides that, a low cost, user-friendly and easy maintained is the characteristic of this project. The computer’s monitor will show the whole plan view of the car park and empty lot can be easily found. It also can inform the drivers where the shortest path to that particular lot. Furthermore, there is a counter showing how many lots are available. Hence, time has been saved for finding empty lot.

This project not only benefits the drivers, but it also convenient the management of the car park to manage their car park. Management can monitor the incoming car to the car park from the control centre. If the car park is full, the system will show to the drivers that the car park is full and stop them from getting in. They will be requested to look for the other car park. With this system, the management will know how many cars are inside the car park premises, how many lots have been taken, how many parking lots available and also how many cars are illegally parked.

1.4 How Does This Project Function?

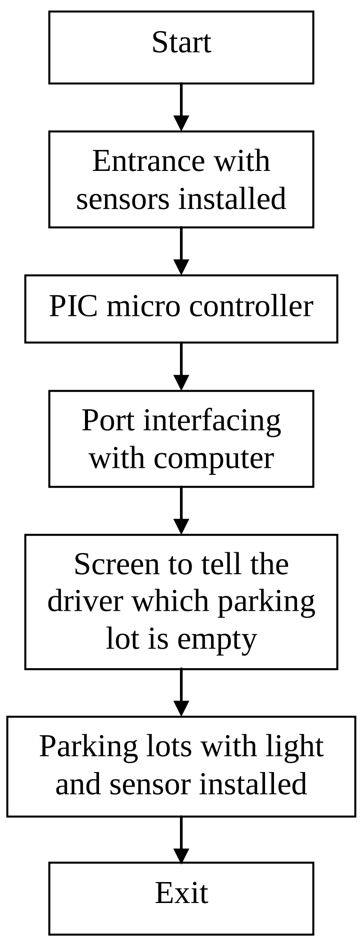
With the Intelligent Car Park System, car drivers will find that it is easier and convenient for them to look for an empty parking lot. When a driver drives into the Intelligent Car Park, there will be a screen showing the whole structure of the car park. From the screen, drivers can get to know which parking lot is empty and which is occupied.

This Intelligent Car Park System using PIC micro controller as the main processor to process the signal received from the sensors. In this project, LDR sensors have been used to replace weight-sensors in reality. These sensors will be placed in every parking lot and the entrance. Initially, the entrance's LDR sensor is in active low (0V) as no object is blocking it. When a car is approaching the entrance, the sensor will detect it and then it will send an output signal +5V to the micro controller (PIC16F877A). Then the PIC micro controller processes the taken data and sends the data to the computer to show a screen full of information about the car park to the driver and also the management.

For the parking lots, they are installed with LDR sensors too. Light will be always turned on, as the parking lots are empty. When there is a car parked on the particular parking lot, the LDR sensor is blocked and then it will send signal to the PIC micro controller. The PIC micro controller will know whether the parking lot is taken or not by reading the signal sent by the sensor. After the PIC micro controller processes the taken data, it will also send signal to the computer through the parallel port.

For the computer section, the Visual Basic 6.0 software will calculate how many cars enter into the premises, which and how many parking lots are available. The current date and time will also be shown in the screen that indicates to the drivers. The Graphic User Interface (GUI) of project is shown in Figure 1.1 and Figure 1.2.

A block diagram of this system is shown below.



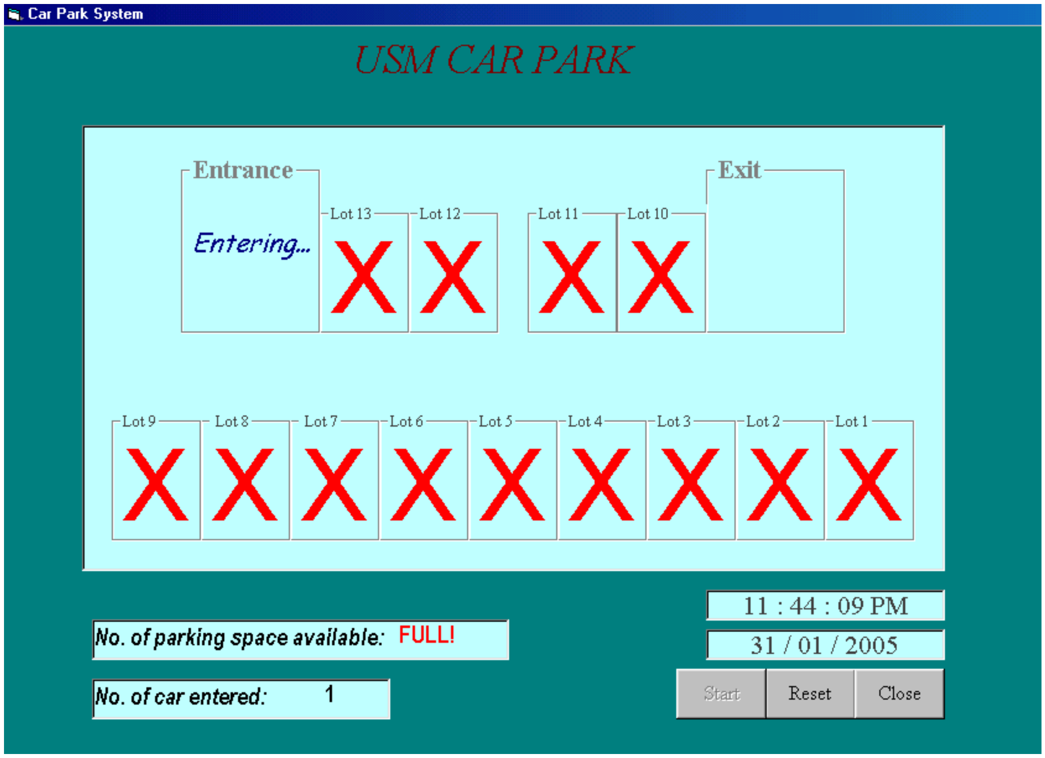


Figure 1.1: The interface showing the car park with all spaces occupied

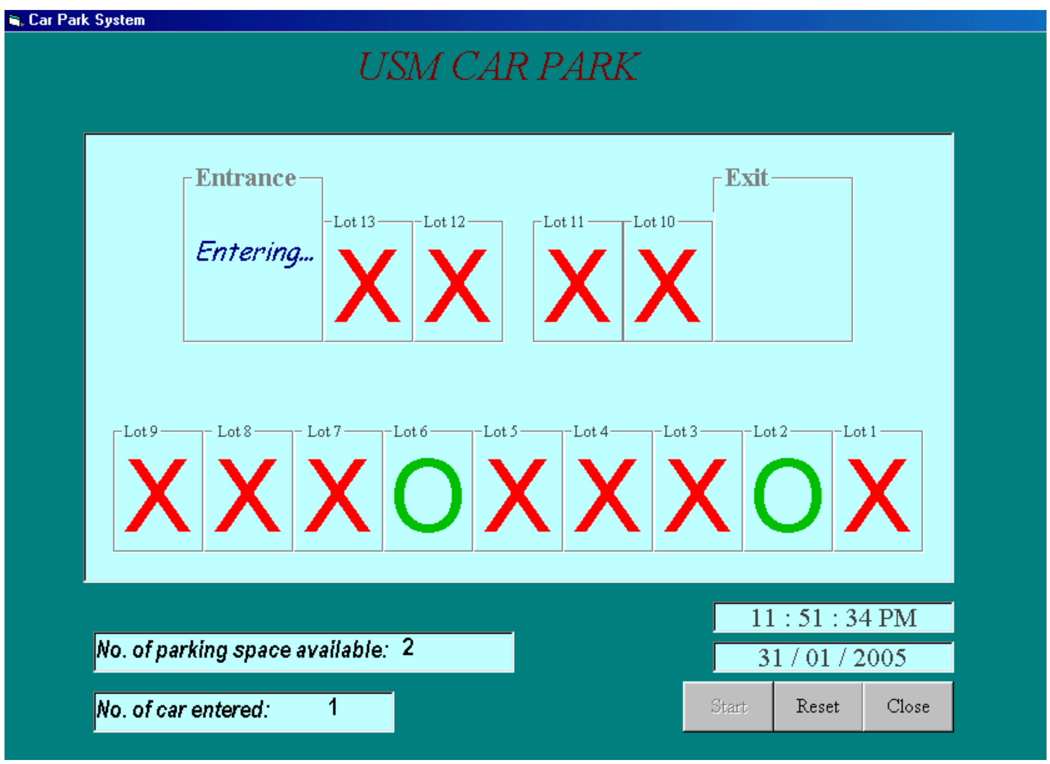


Figure 1.2: The interface showing two parking lots available

Chapter 2

HARDWARE

2.1 Main Board

Many kinds of tools and software used to build the main board of this project. Protel 99 SE is used to design my circuit. The main board is built to accommodate the PIC16F877A micro controller as the PIC16F877A is used to handle the whole process of this project. The LDR sensors are used as the input to PIC16F877A and from the main board, the signal will be sent to computer using parallel port. All information about the car park will be shown on the computer screen (Myke Predko, 1998).

Figure 2.1 shows a photo of the main board that has been built.

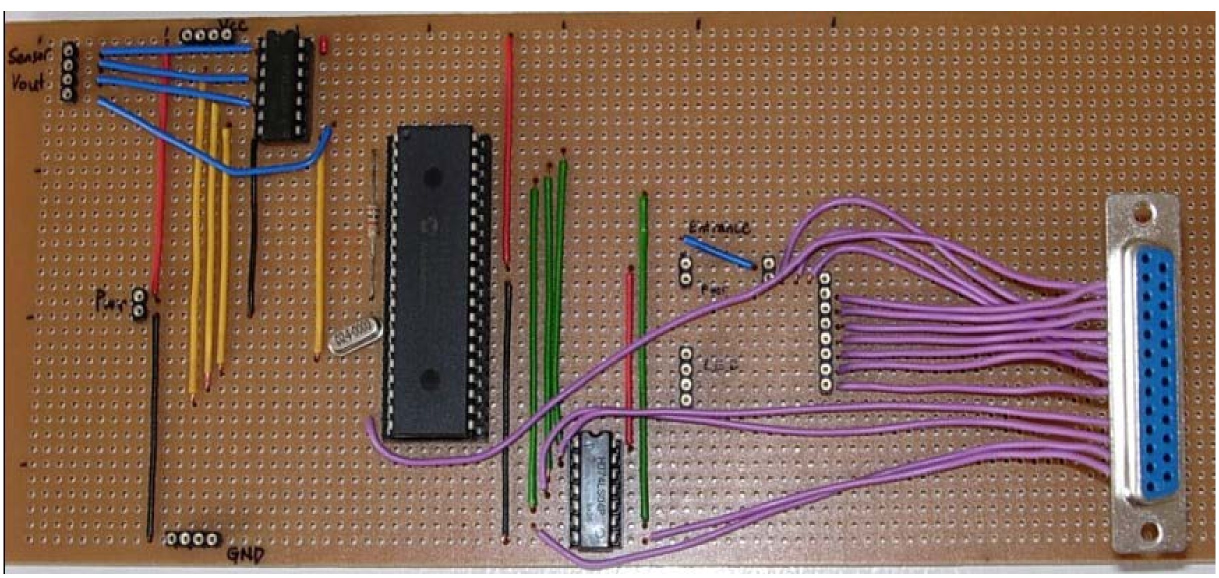


Figure 2.1: Main board of the project

2.1.1 Schematic Diagram Designing

The schematic diagram of the main board is designed using Protel 99 SE (http://vod.wipe.edu.cn/vod/soft). While designing the schematic in Protel 99 SE, a few problems have been encountered. One of the problem is the footprint of the PIC16F877A cannot be found in the library. Hence, to make things easier, a schematic library has been created using the footprint of DIP40 chip component by referring to the PIC16F877A's actual size. Multiple trials and errors of placing the components coordinate have been done. After all problems have been solved, the whole schematic diagram is ready. Please refer to Appendices A for the schematic diagram of the main board.

2.1.2 Building The Main Board

Due to insufficient time, the main board is built using Vero board instead of developing a Printed Circuit Board. Multiple trials and errors of placing the components on the Vero board have been done using the "Strip Board Planning Sheet" that can be downloaded from the website: www.kpsec.freeuk.com. After all the procedure has been done, the components are ready to be soldered on the Vero board.

2.2 PIC Micro Controller

PIC micro controller is a product of Microchips Technology Inc. The model PIC16F877A has been chosen as the main components of the board because it has more ports to use. Furthermore, it is an enhanced version of PIC16F877. The PIC micro controller is a very useful and low cost micro controller. Various types of electronic applications can be used in it, such as digital electronics, pulse-width modulation application, analogue-to-digital conversion, digital-to-analogue conversion and etc. It needed 5V to operate.

There are various types of PIC micro controller programmer. The most common and widely used programmer for PIC16F877A is the JDM programmer. In this project, the application of digital electronics is used in the PIC16F877A. This means it receive 0V as OFF or 0, and 5V as ON or 1 (Microchips Technology Inc., 2003).

Figure 2.2 shows the PIC16F877A micro controller chip.

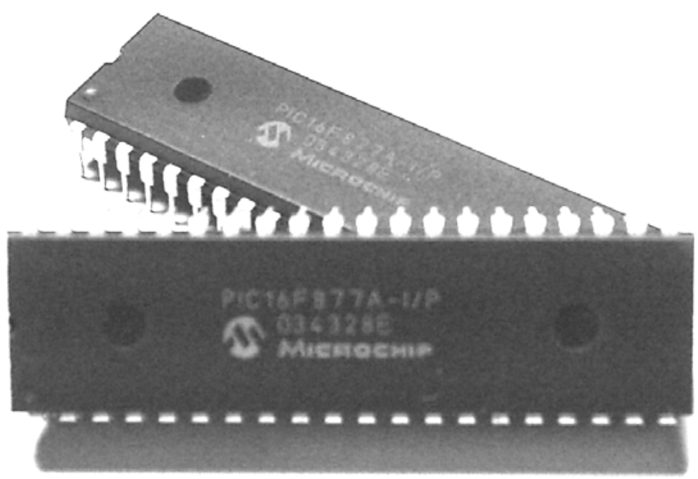


Figure 2.2: PIC micro controller chip – PIC16F877A

2.2.1 I/O Ports

PIC16F877A is a 40 pins micro controller. The PIC16F877A has a total of five ports in the chip that are Port A, Port B, Port C, Port D and Port E. For Port A, there are 6 I/O pins. For

Port B, Port C and Port D, there are 8 I/O pins individually. Finally, Port E has only 3 I/O pins. By summing up all the pins from all the ports, there are a total of 33 pins for input and output.

All these I/O pins can be defined to act as either input or output. Definitions are made while writing the assembly language for the chip. Some pins are multiplexed with an alternate function for peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin (Microchips Technology Inc., 2003). These functions include External Interrupts, Timer 0 External Clock Input, Timer 1 External Clock Input and Parallel Slave Port. The pin configuration of PIC16F877A is shown in Figure 2.3. Table 2.1 details the pin out of the device with description.

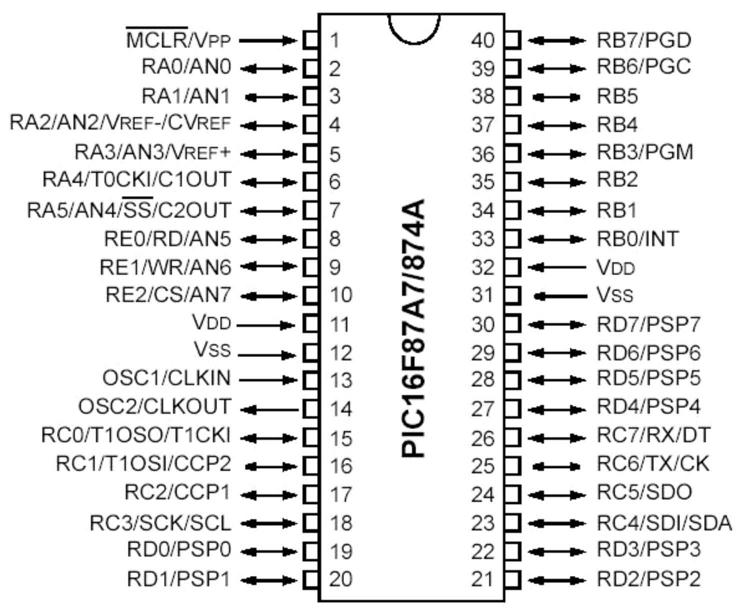


Figure 2.3: Pin configuration of PIC16F877A

(Adapted from Microchips Technology Inc.)

Table 2.1: PIC16F877A Pin Out Description

Pin Name	PDIP Pin Number	I/O/P Type	Buffer Type	Description
OSC1/CLKI	13		ST/CMOS	Oscillator crystal or external clock input.
OSC1		I		Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; Otherwise CMOS.
CLKI		I		External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO	14		—	Oscillator crystal or clock output.
OSC2		O		Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO		O		In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP	1		ST	Master Clear (input) or programming voltage (output).
MCLR		I		Master Clear (Reset) input. This pin is an active low Reset to the device.
VPP		P		Programming voltage input.
				PORTA is a bi-directional I/O port.
RA0/AN0	2		TTL	
RA0		I/O		Digital I/O.
AN0		I		Analog input 0.
RA1/AN1	3		TTL	

RA1 AN1 RA2/AN2/VREF- /CVREF RA2 AN2 VREF- CVREF RA3/AN3/VREF+ RA3 AN3 VREF+ RA4/T0CKI/C1OUT RA4 T0CKI C1OUT RA5/AN4/SS/C2OUT RA5 AN4 SS C2OUT	 4 5 6 7	I/O I I/O I I O I/O I I I/O I O I/O I I O	 TTL TTL ST TTL	Digital I/O. Analog input 1. Digital I/O. Analog input 2. A/D reference voltage (Low) input. Comparator VREF output. Digital I/O. Analog input 3. A/D reference voltage (High) input. Digital I/O – Open-drain when configured as output. Timer0 external clock input. Comparator 1 output. Digital I/O. Analog input 4. SPI slave select input. Comparator 2 output.
RB0/INT RB0 INT RB1 RB2 RB3/PGM RB3 PGM RB4	33 34 35 36 37	 I/O I I/O I/O I/O I I/O	TTL/ST(1) TTL TTL TTL TTL	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. Digital I/O. External interrupt. Digital I/O. Digital I/O. Digital I/O. Low-voltage ICSP programming enable pin. Digital I/O.

RB5	38	I/O	TTL	Digital I/O.
RB6/PGC	39		TTL/ST(2)	
RB6		I/O		Digital I/O.
PGC		I		In-circuit debugger and ICSP programming clock.
RB7/PGD	40		TTL/ST(2)	
RB7		I/O		Digital I/O.
PGD		I/O		In-circuit debugger and ICSP programming data.
				PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	15		ST	
RC0		I/O		Digital I/O.
T1OSO		O		Timer1 oscillator output.
T1CKI		I		Timer1 external clock input.
RC1/T1OSI/CCP2	16		ST	
RC1		I/O		Digital I/O.
T1OSI		I		Timer1 oscillator input.
CCP2		I/O		Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1	17		ST	
RC2		I/O		Digital I/O.
CCP1		I/O		Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL	18		ST	
RC3		I/O		Digital I/O.
SCK		I/O		Synchronous serial clock input/output for SPI mode.
SCL		I/O		Synchronous serial clock input/output for I2C mode.
RC4/SDI/SDA	23		ST	
RC4		I/O		Digital I/O.
SDI		I		SPI data in.
SDA		I/O		I2C data I/O.
RC5/SDO	24		ST	
RC5		I/O		Digital I/O.
SDO		O		SPI data out.
RC6/TX/CK	25		ST	
RC6		I/O		Digital I/O.
TX		O		USART asynchronous transmit.
CK		I/O		USART1 synchronous clock.

RC7/RX/DT	26		ST	
RC7		I/O		Digital I/O.
RX		I		USART asynchronous receive.
DT		I/O		USART synchronous data.
				PORTD is a bi-directional I/O port or Parallel Slave Port when interfacing to a microprocessor bus.
RD0/PSP0	19		ST/TTL(3)	
RD0		I/O		Digital I/O.
PSP0		I/O		Parallel Slave Port data.
RD1/PSP1	20		ST/TTL(3)	
RD1		I/O		Digital I/O.
PSP1		I/O		Parallel Slave Port data.
RD2/PSP2	21		ST/TTL(3)	
RD2		I/O		Digital I/O.
PSP2		I/O		Parallel Slave Port data.
RD3/PSP3	22		ST/TTL(3)	
RD3		I/O		Digital I/O.
PSP3		I/O		Parallel Slave Port data.
RD4/PSP4	27		ST/TTL(3)	
RD4		I/O		Digital I/O.
PSP4		I/O		Parallel Slave Port data.
RD5/PSP5	28		ST/TTL(3)	
RD5		I/O		Digital I/O.
PSP5		I/O		Parallel Slave Port data.
RD6/PSP6	29		ST/TTL(3)	
RD6		I/O		Digital I/O.
PSP6		I/O		Parallel Slave Port data.
RD7/PSP7	30		ST/TTL(3)	
RD7		I/O		Digital I/O.
PSP7		I/O		Parallel Slave Port data.
				PORTE is a bi-directional I/O port.
RE0/RD/AN5	8		ST/TTL(3)	Digital I/O.
RE0		I/O		Read control for Parallel

RD		I		Slave Port.
AN5		I		Analog input 5.
RE1/WR/AN6	9		ST/TTL[3]	
RE1		I/O		Digital I/O.
WR		I		Write control for Parallel Slave Port.
AN6		I		Analog input 6.
RE2/CS/AN7	10		ST/TTL[3]	
RE2		I/O		Digital I/O.
CS		I		Chip select control for Parallel Slave Port.
AN7		I		Analog input 7.
VSS	12, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	P	—	Positive supply for logic and I/O pins.
NC	—	—	—	These pins are not internally connected. These pins should be left unconnected.

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.
2. This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3. This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

To send the output signal, the port has to be assigned as output port first. Once a pin of the 16F877A has been assigned as output, the program in the chip can send a signal out to other devices using this particular pin. In this project, the concept is digital electronic, therefore sending out signal is either a 5V as HIGH or 0V as LOW.

When the pin that is connected to a LED with the cathode pin connected to the VSS receives a signal HIGH, the pin will light the LED up. If the pin is cleared, that means it drops from 5V to 0V, therefore the LED will turn off immediately. This step can be used to test the level of the particular pin.

There are various ways to write the assembly language to send output signal from the chip. User may move a value into the register and then later move it to the port to send the signal out. For example, moving a value of 11101101 into the W register and then move the value in W, which are 11101101 to Port B. Now, refer back to Figure 3.2. Now the pins for Port B having the values shown below:

Table 2.2: Value of the Pins for Port B

Pin No.	Pin Name	Value
33	RB0	1
34	RB1	0
35	RB2	1
36	RB3	1
37	RB4	0
38	RB5	1
39	RB6	1
40	RB7	1

When using this method to send signals, the instructions used are ‘MOVLW’ and ‘MOVWF’. Another way of turning a particular pin on is by using the instruction ‘BSF’. For example, bit 3 of Port B is set as shows below,

```
BSF     portb, 3
```

To read the input signal, the port has to be assigned as input port first. The PIC micro controller will read from the particular pin that the user assigned it to read from. It must be noted that the input signal voltage should not more than 5 volts. The maximum tolerance of input voltage can go until 5.5 volts only. If any input signals that exceed the maximum voltage allowed, it will damage the PIC16F877A micro controller. It is advisable to check the voltage from other devices before connecting it to the PIC micro controller.

Even after the user had set the port as input port, but the written program inside the chip does not have the instruction to read the pin, therefore any input signal to that particular pin will be ignored. The PIC will only do what it is asked to do.

There are two instructions used in the PIC assembly language to read the input port. They are ‘BTFSS’ and ‘BTFSC’. These instructions will only read one pin at a time. For example,

```
BTFSC        portc, 6
```

With this instruction, the PIC micro controller will read only bit 6 of Port C (RC6 – pin 25). If the pin received a signal HIGH or 5V, the next instruction will be executed. But if the pin received a signal LOW or 0V, then the next instruction is discarded and will proceed immediately to the instruction after the next one. For the instruction ‘BTFSS’, it works in the opposite way. It will skip the next instruction if the pin has a signal HIGH or 5V and execute next instruction if the pin has a LOW or 0V.

The PIC micro controller has a total of 35 instructions only. With these 35 instructions, various type programs can be written. Please refer to Appendices D for the “Instruction Set of the Micro Controller” and how it can be used in the program.

2.2.2 Writing Assembly Language

When writing the PIC assembly language, try to keep it as simple as possible. In this project, MPLAB IDE has been chosen to write and compile the program, which is written in PIC assembly language.

For writing a PIC program, we must first define the input and output ports of the chip. Refer to the datasheet's register file map in Appendices D. TRISA, TRISB, TRISC, TRISD and TRISE are registers that define the character of a specific port. For example, setting 00000000 to TRISB will define all bits in PORTB as output bits. 0 means output, while 1 means input.

The registers of PIC are separated into 4 banks. Bank 1 is used to control the actual operation of the PIC. It will be used to set port as input or output. Bank 0 is used to manipulate the data.

Switching from Bank 0 to Bank 1 is necessary to set the port to become either input or output. STATUS is the register that is used to do the switch. Setting bit 5 of STATUS to HIGH will move the pointer to Bank 1. Therefore, port setting can be done. The instruction 'BSF' can be used to set a HIGH on bit 5 of STATUS. For Example,

```
BSF     STATUS, 5
```

Now, the pointer is pointing at Bank 1. Next is to move a value into the TRIS registers. For example:

```
MOVLW    00h
MOVWF    TRISB
```

Executing the instructions above will set all Port B pins as output bit.

After finish setting all the ports, switching back from Bank 1 to Bank 0 is necessary. In this case, the instruction 'BCF' will be used to clear bit 5 of STATUS register. For example,

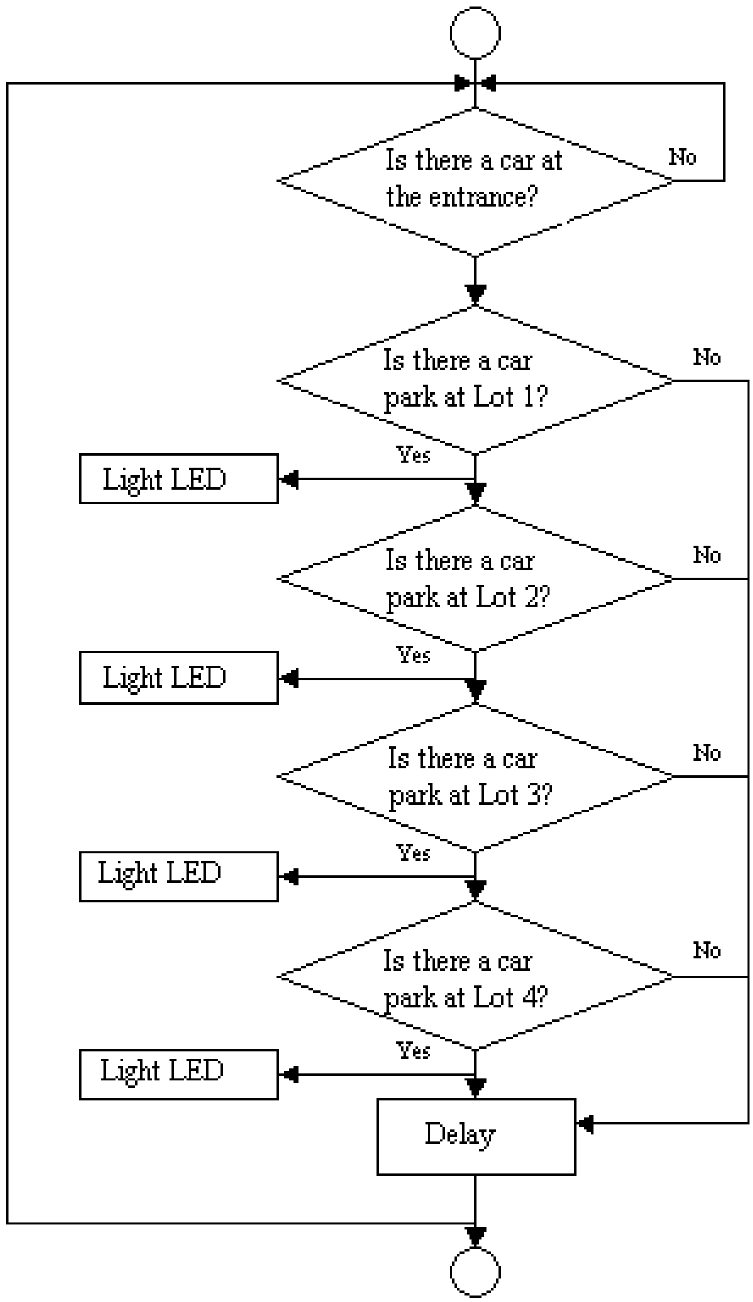
```
BCF    STATUS, 5
```

After all ports have been set, writing the main program will be the next step. In building this project, a few steps have been taken before the actual PIC program can be produced. First is to determine the number of ports to be used and what is it for. Table 2.3 shows the port setting used in this project.

Table 2.3: Port setting

Port	Pin Name	Pin No.	I/O	Purpose
Port A	Not in use	Not in use	Not in use	Not in use
Port B	RB0	33	Input	Reserved
	RB1	34	Input	Entrance
	RB2	35	Input	Exit
	RB4	37	Output	Signal to comp.
	RB5	38	Output	Signal to comp.
	RB6	39	Output	Signal to comp.
	RB7	40	Output	Signal to comp.
Port C	RC0	15	Input	Parking Lot
	RC1	16	Input	Parking Lot
	RC2	17	Input	Parking Lot
	RC3	18	Input	Parking Lot
	RC4	23	Output	LED
	RC5	24	Output	LED
	RC6	25	Output	LED
	RC7	26	Output	LED
Port D	RD0	19	Output	Signal to comp
Port E	Not in use	Not in use	Not in use	Not in use

Secondly, drawing out a flow chart of how the program should run is necessary as shown below.



The PIC program can be written according to the flow chart above. Notice that the program is always running in loop. This is because it keeps on refreshing to detect whether there is any car coming in at the entrance. When there is a car enters, the PIC will read the signal from the sensors at parking lots and light a LED if it is empty. Please refer Appendices B for the PIC micro controller source code.

2.2.3 Burning the PIC Micro Controller Chip

The software used to burn the PIC micro controller chip in this project is IC-Prog by Bonny Gijzen. It is compatible with multiple types of PIC programmer. A screenshot of the IC-Prog is shown in Figure 2.4 as follow:

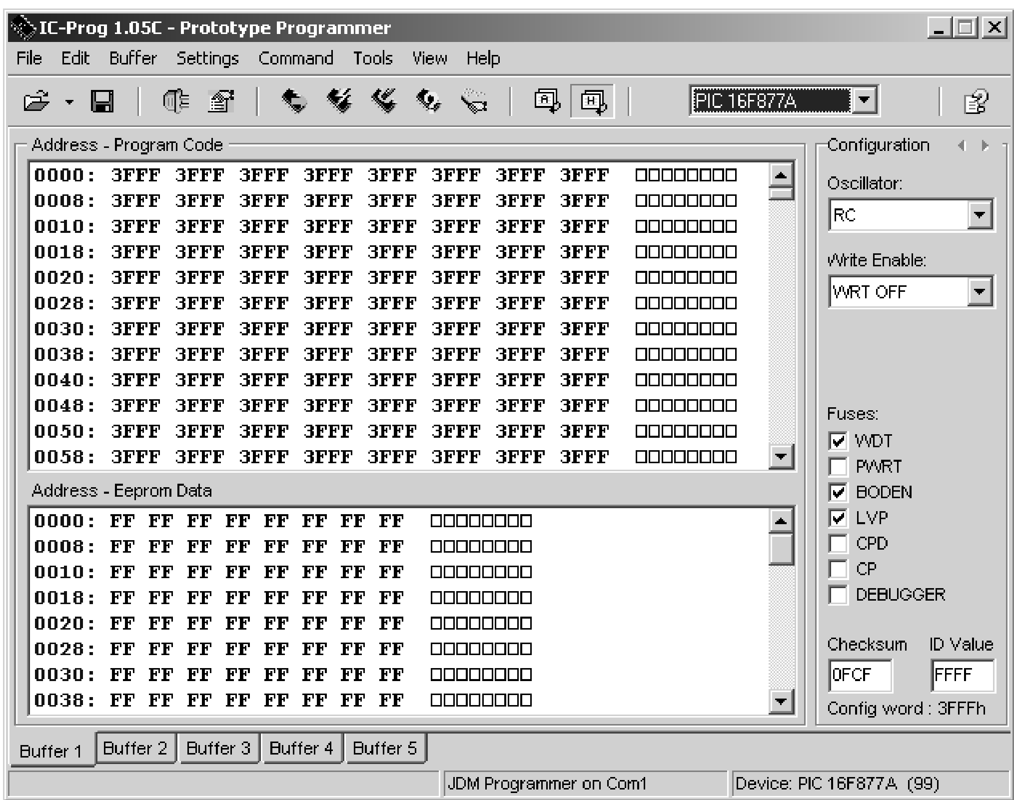


Figure 2.4: Screenshot of the IC-Prog

To burn a program into the PIC micro controller chip, a .hex file has to be generated. It can be done using the MPLAB IDE software by compiling the program that had written. Once the .hex is ready, go to ‘File > Open File’ to open the .hex file that was compiled earlier. Notice that the program codes will change. Next, change the oscillator type to XT. In XT mode, the PIC micro controller will be using a crystal oscillator o generate clock pulse. This can save cost and time to build a clock generator externally. The crystal oscillator is connected to the PIC at pin 13 and pin 14. Once everything is set, the burn process will start by pressing the  button.

Before the actual program has been written, many samples have been written and burnt into the chip. Multiple trials and errors have been done to test the program before the final program is produced. The simple program tested had shown that pin 36, RB3 cannot be use produce a HIGH or LOW signal. Therefore, this pin had been ignored in the project.

2.3 Light Dependent Resistor (LDR) Sensor

Light Dependent Resistor is proportional to the intensity of light. It means that when the intensity of light increases, the resistance of the LDR drops and vice versa. As we know, transistors can be function a switch or an amplifier. In this project, transistors are used as a switch (Joseph J. Carr, 2000). When used as an electronic switch, a transistor is normally operated alternately in cut-off and saturation. Detail of the switching characteristics of transistor can be found in reference (Thomas L. Floyd, 1999).

LDR sensor schematic circuit diagram is shown in Figure 2.5.

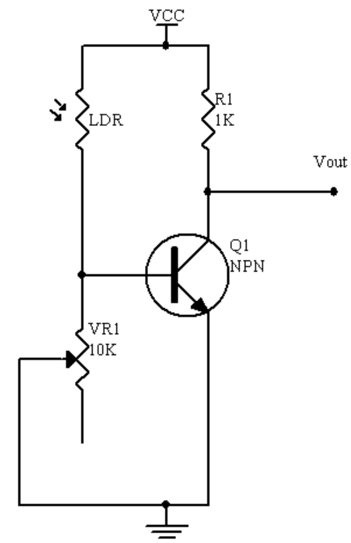


Figure 2.5: LDR sensor circuit

The light intensity of the environment is not stable enough and it is different at every place. Therefore, the VR1 is placed there for tuning purpose. In between the path of the LDR and VR1, the rules that apply there is the Voltage Divider Rule. The resistance of the LDR varies approximately from 30K ohms to 150K ohms. The VCC is connected to a +5V. When the transistor is turn on, the Vout is approximately 5V. This signal will be sent into the PIC16F877A to process. When the transistor is turned off, the Vout will be short circuit to ground, thus giving the output voltage a 0V (Robert G.Middleton, 1986).

Figure 2.6 shows the Light Dependent Resistor (LDR) and Figure 2.7 shows the LDR sensor constructed in a Vero board. Figure 2.8 shows the preset type variable resistor.

Figure 2.6: Light Dependent Resistor (LDR)

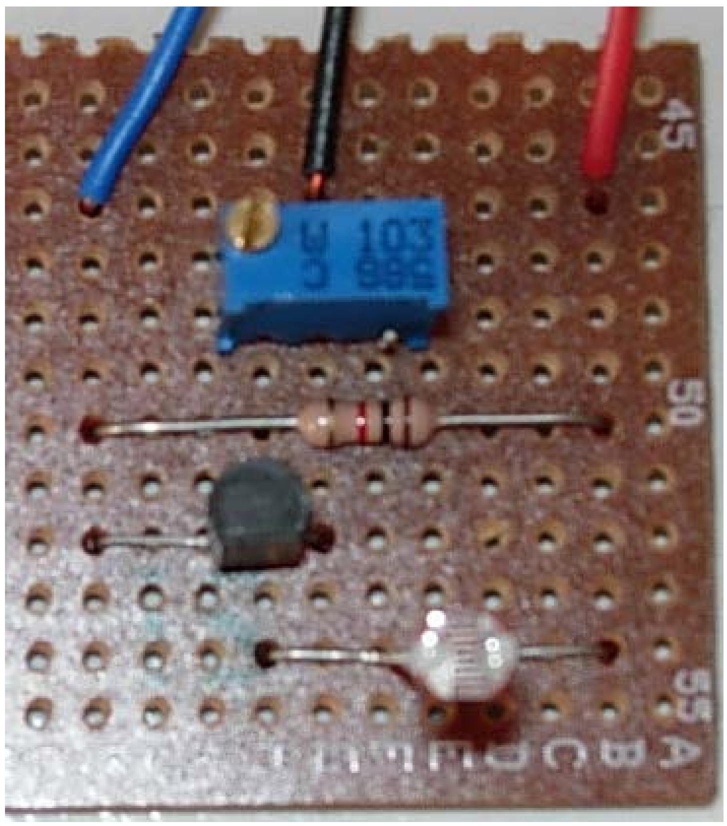


Figure 2.7: LDR sensor constructed in a Vero board

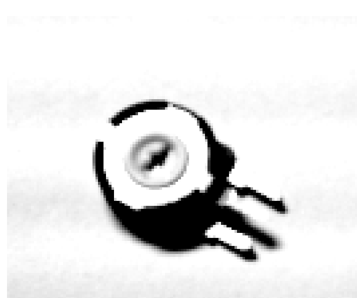


Figure 2.8: Variable Resistor (Preset type)

2.4 Parallel Port

The parallel port of a computer is a type of I/O port. It can be considered as a gateway for external devices to communicate with the computer. The best example of an external device that is connected to the parallel port is the printer. Some people even called the parallel port as printer port.

There are three types of parallel port setting. They are Standard Parallel Port (SPP), Enhanced Parallel Port (EPP) and Extended Capabilities Parallel Port (ECP).

In a parallel port, there are 25 connection pins. The parallel port is made up of four different sections. The four sections are Status Lines, Data Lines, Control Lines and Ground. There are 8 data lines, and they are the primary means of getting information out of the port. In simple projects, mostly will concentrating on the data lines. The control lines are another 4 outputs. They are meant to provide control signals to the printer (such as form feed or initialise). The status lines are a standard parallel port's only inputs. There are 5 of them. They were meant to allow the printer to communicate things such as error, paper out and busy to the PC. All the pins available can be used as either an input pin or an output pin except for those grounded pins (pin 18 – pin 25) (http://www.aaroncake.net/index.asp).

Figure 2.9 shows the connection of a parallel port.