KOLEJ UNIVERSITI KEJURUTERAAN & TEKNOLOGI MALAYSIA

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **CAR LOCK INDICATOR**

SESI PENGAJIAN: **2005/2006**

Saya **HAIRULAZAM BIN HAIRULDIN (830523-08-5369)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Kolej Universiti Kejuruteraan & Teknologi Malaysia.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

☐	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
√	**TIDAK TERHAD**	

Disahkan oleh:

_____ (TANDATANGAN PENULIS)

_____ (TANDATANGAN PENYELIA)

Alamat Tetap:

H-79, JLN SIAKAP ,
TMN SERI KURAU, 34350 KUALA KURAU
PERAK

BADRUDDIN BIN MUHAMMAD
(Nama Penyelia)

Tarikh: **……..**

Tarikh: : ……

CATATAN:
- * Potong yang tidak berkenaan.
- ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
- ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

Signature : …….

Author : HAIRULAZAM BIN HAIRULDIN

Date : …….

CAR LOCK INDICATOR

HAIRULAZAM BIN HAIRULDIN

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

Faculty of Electrical & Electronics Engineering
Kolej Universiti Kejuruteraan & Teknologi Malaysia

MAY, 2006

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : …….

Name : BADARUDDIN BIN MUHAMMAD

Date : …….

ACKNOWLEDGMENT

I would like to thank my supervisor En. Badaruddin Bin Muhammad for his advice and support throughout the project.

I would like to express my gratitude to Hairul Azwani Bin Aziman, Mohd Khairi Bin Sulaiman and Ali Munawir Bin Borhanudin for the co-operation during doing this project.

I also wish acknowledgement to the people who give support direct or indirectly to the project and during the thesis writing. With all my hearts, thank you very much.

ABSTRAK

Pada hari ini, kereta telah menjadi salah satu mod pengangkutan yang popular kerana kebanyakan rakyat Malaysia mampu untuk memilikinya. Terdapat banyak pilihan kereta di pasaran sekarang, daripada kereta bajet sampailah kepada kereta mewah, dan ianya cuma bergantung kepada kemampuan. Satu daripada aspek penting sesebuah kereta adalah keselamatan kerana kecurian kereta adalah jenayah yang berlaku di seluruh dunia, termasuklah Malaysia. Buat masa sekarang, pembuat kereta tempatan di Malaysia memberikan sistem keselamatan piawai untuk setiap kereta yang dihasilkan, iaitu sistem kunci berpusat dan penggera kereta piawai. Fokus sekarang adalah kepada sistem penguncian kereta, di mana remot konvensional untuk sistem penguncian kereta biasanya terdiri daripada dua butang, yang mana butang pertama adalah untuk fungsi kunci dan buka, dan butang yang kedua adalah untuk penggera kereta. Tiada penunjuk berfungsi menunjukkan status kunci kereta, yang berfungsi sebagai konformasi kedua kepada status kunci kereta. Di sinilah datangnya idea untuk memperbaharui sistem kunci kereta konvensional dengan meletakkan penunjuk yang berfungsi sebagai konformasi kedua untuk status kunci kereta. Projek ini dinamakan sebagai Car Lock Indicator dan menggunakan pengawal-mikro Motorola MC68HC11A1 sebagai pusat kepada sistem. Untuk penunjuk, Diod Pemancar Cahaya (LED) adalah paling sesuai kerana ianya adalah murah, boleh dipercayai, penggunaan voltan yang rendah dan seterusnya memanjangkan jangka hayat bateri.

ABSTRACT

Nowadays, car has become one of the most popular modes of transportation because many of Malaysians could afford to have a car. There are many choices of cars in the market, from budget cars to the luxurious cars, only a matter now is affordability. One of the important aspects of a car is security because car theft is a crime that occurs all around the world, including Malaysia. Currently, local car manufacturers in Malaysia provides a standard security system for each car manufactured, that is central locking system and standard car alarm. The focus is on the car locking system, where the conventional remote for the car locking system is commonly consists of two buttons, which is the first, is for lock and unlock function and the second button is for the car alarm. There is no indicator that indicates the lock status of the car, acting as the double confirmation for the lock status. Here comes the idea to improve the conventional car lock system by adding an indicator to work as the second confirmation for the car lock status. The project is named as Car Lock Indicator and using microcontroller Motorola MC68HC11A1 as the heart of the system. For the indicator, Light Emitting Diode (LED) is the most suitable because it is cheap, reliable, low voltage consumption thus makes the battery could last longer.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF ABBREVIATION

LED	-	Light Emitting Diode
LCD	-	Liquid Crystal Display
uC	-	Microcontroller
PC	-	Personal Computer
Tx	-	Transmitter
Rx	-	Receiver
RAM	-	Random Access Memory
ROM	-	Read Only Memory

LIST OF APPENDIX

APPENDIX TITLE **PAGE**

CHAPTER I

INTRODUCTION

1.1 Background

This chapter explains the background of modern cars, the security of the cars, the car lock system and the introduction to the car lock indicator. Currently the car lock system is the main preventive measures taken to avoid car theft and the indicator is the device that alerts the car owner about the status of the car. This chapter also explains the overview of the Car Lock Indicator project, the objective of the project, the project scope and finally the research methodology to conduct the project.

1.1.1 The Modern Cars

There are many prestigious car manufacturers in the world manufacturing prestigious car for the automobile market in the world such as Mercedes-Benz, Honda, Bavaria Motor Works (BMW), Ferrari and many other car manufacturers. They have gone through many years of experience and so that their product is well known all over the world.

In Malaysia, we also have our very own car manufacturers like Proton and Perodua. They also have produced their own car model, like Proton come with it first Proton Saga then followed by Iswara, Satria, Wira, Perdana, Waja, and many other

improvement models and meanwhile Perodua comes with it first Kancil then followed by Kenari, Kelisa, Kembara, and finally MyVi. The technology in the automotive industry becomes more advance and competitive because the demands for performance, designs, safety, and price have been the concern among Malaysians.

1.1.2 The Security

Today, the security of the car has been an important matter. According to the statistics made by Polis DiRaja Malaysia (PDRM), the total numbers for the car theft that have been reported from year 2000 to year 2005 is 51,214. Below is the graph showing the figure of the cases divided into years from 2000 to 2005.

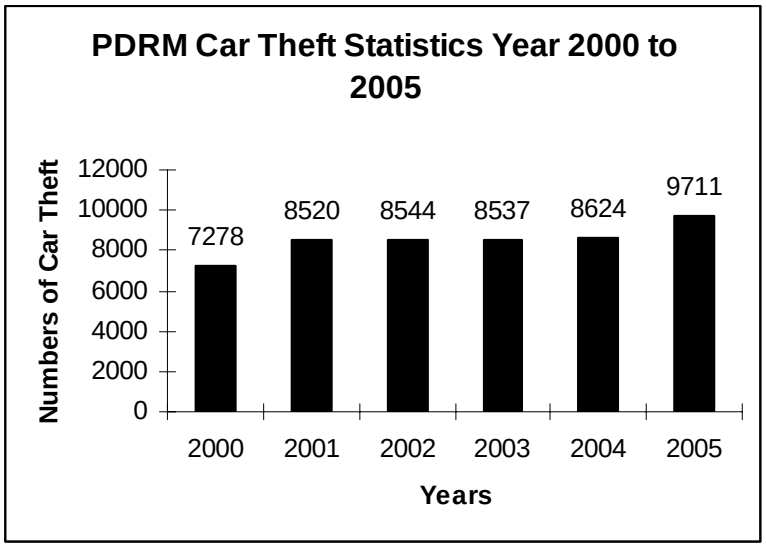


Figure 1.1: The increasing numbers of car theft in Malaysia.

There are many causes which can contribute for the car theft and one of them is the attitude of the car owner. The modern lifestyles have made people become rushing in doing everyday activities and sometimes they become careless and irresponsible to their own property, such as a car. Sometimes when people is rushing to do their daily chores they easily forget to lock their car and then uncertain about

their car have been locked or not. This could lead to the increasing figures of the car theft statistics.

1.1.3 Car Lock System

Car lock consists of the process of locking and unlocking process. There are several ways to lock and unlock the car such as:

i) By using the Car Key.
ii) By pressing lock or unlock button inside the car.
iii) By using the combination lock on outside of the door.
iv) By pulling up or pushing down the knob on the inside of the door.
v) By using keyless-entry remote control
vi) And by the signal from the control centre.

Meanwhile, the car lock system consists of the lock and unlocks mechanism, then the central lock mechanism and finally the car alarm mechanism. The conventional car lock system only provides minimal security features for the car. In order to improve the reliability of the car lock system, the Car Lock Indicator comes into the picture to act as a double-confirmation about the car status.

1.1.4 The Car Lock Indicator

As mentioned previously, the car lock indicator is a feature to improve the reliability of the car lock system by acting as the double confirmation about the car status. There are many electronic devices that could be used as an indicator and in this case LED is used as the indicator. LED is selected because this component is cheap, easy to find, reliable, need less maintenance, and small voltage consumption makes the battery life span longer.

1.2 Overview of the Car Lock Indicator

Indicator is a part of the device that informs the user in many ways, but commonly used is visual indicator such as LED, LCD, and etc. The purpose of the indicator is to indicate whether that device is working properly or not, functioning or failing so that the user will easily understand the status of the device and the proper action can be taken.

For the Car Lock Indicator, LED is used to indicate the locking condition of the car. If the check button is pushed and the LED is glowing, that's mean the car is properly locked but if it does not glowing, that's mean the car is not properly locked. So, by looking at the indicator and it does not glowing, the user could take a proper action that is to go back to the car and lock it.

The Car Lock Indicator consists of two units, which is the Remote unit and the Car unit. The Remote unit has two functions, which is to lock or unlock the car and to act as an indicator to the car lock status. Then the car unit function is to control the lock and unlock process of the car, and controlling other safety mechanism such as central lock and the car alarm.

The development of the Car Lock Indicator is divided into four parts. They are Part 1, Part 2, Part 3 and Part 4. Part 1 of the project concentrates on the development board of the Microcontroller Motorola MC68HC11A1 so that the microcontroller (uC) can communicate with personal computer (PC) and then programmed using JBug and HCLoad software. Part 2 of the project concentrates on the development of the Transmitter (Tx) and Receiver (Rx) unit using Tx and Rx module together with Integrated Circuit (IC) PT 2262 and PT 2272. Part 3 of the project concentrates on how to combine together the working uC development board with the Tx and Rx circuits. Finally, on Part 4 of the project concentrates on the programming of the microcontroller to make it work together with the Tx and Rx circuits so it can activate the indicator.

1.3 Project Objective

The aim of this project is to build an indicator that indicates the status of the car whether it is locked or not. The main Objectives are:

1) Improve the car lock system with the lock indicator on the remote.
2) The car owner can self-inform that his/her car was securely locked anytime.
3) The user does not have to come back to his/her car from a distance only to check whether his/her car was securely locked.

1.4 Project Scope

The project is expected to have a prototype set complete with indicator button and a LED indicator. The Indicator button is to check whether the car was securely locked or not just by pressing it. Then, the additional LED will react by glowing if the car was securely locked and will not glow if the car is not locked. The MC68HC11 will be programmed to control the sequence of the transmitter and receiver and also the component that store the information about the lock status of the car.

1.5 Research Methodology

This part will picture the general idea on how the Car Lock Indicator project will be conducted according to the systematical procedures. They are:

(i) Generates idea and the Theoretical design
(ii) Buying related components for the project
(iii) Assembling Tx and Rx circuits and their modules
(iv) Testing the finished Tx and Rx circuits

(v) Assembling the uC MC68HC11A1 circuit

(vi) Testing the finished uC MC68HC11A1 circuit

(vii) Combining the Remote unit (Input + uC + Tx + Rx + Indicator)

(viii) Combining the Car unit (Rx + uC + Tx)

(ix) Programming and Testing the Remote unit

(x) Programming and Testing the Car unit

(xi) Modeling the Remote unit and the Car unit

(xii) Complete prototype of the Car Lock Indicator

CHAPTER II

THE REMOTE KEYLESS ENTRY

2.1 Background

This chapter focuses on the remote keyless entry system for the car lock system which has become the standard security features for cars in Malaysia. This chapter also discusses and made comparisons between the car lock system products available in today's market.

2.2 The Remote Keyless Entry System

Figure 2.1: A standard Remote Keyless Entry unit.

A remote keyless system is a system designed to remotely lock, or unlock, access to automobile, and in this case the automobile is referred to the car. In the case of cars, it duplicates all of the features of a standard car key with the added

convenience of operating the power door locks, eliminating the need to physically manipulate a key into a lock as well as being able to do so at a distance. A remote keyless system can include both a Remote Keyless Entry system (RKE) and a Remote Keyless Ignition system (RKI). It works on the same principle as some television remote controls, by sending pulses of radio frequency energy on a particular frequency. These pulses are interpreted by the receiver in the automobile, which, in turn, performs the appropriate function. Newer systems implement encryption to prevent car thieves from intercepting and spoofing the signal.

For the door locks, the system works in two ways by either providing confirmation of successfully (un-)locking the car through a light or a horn signal. Usually the system offers an option to easily switch between these two variants. While both versions provide almost the same functionality, the light signals are more discrete while horn signals might create a nuisance in residential neighborhoods, in front of office buildings, and other busy parking areas.

The functions of a remote keyless entry system are contained on a small, hand-held fob attached to the keychain or built into the ignition key handle itself. Buttons are set aside to lock or unlock the doors, plus open the trunk. Some cars will also close any open windows and roof when remotely locking the car. On many newer cars, the remote keyless fob also contains a button to activate the car alarm. Cars with remote keyless ignition systems can be started by the push of a button on the handheld fob.

2.3 Product Comparisons

No.	Products	Company	Remote	Indicator	Range	Frequency	Price
1.	2 Door Power Locks and Keyless Entry	A1 ELECTRIC	4 buttons	LED	-	-	USD 99.95
2.	4 Door Power Locks and Keyless Entry	A1 ELECTRIC	4 buttons	LED	-	-	USD 119.95
3.	RF-420LCD	AUTOPAGE	5 buttons	LED & LCD	-	AM/AM	-
4.	Boomerang Alpha 2000	INSPECTOR	3 buttons	LCD	-	FM, 433.92 MHz	USD 394.99
5.	DEI Keyless Entry	A1 ELECTRIC	3 buttons	LED	-	-	USD 49.95
6.	Champion Mark 3	AVITAL	3 button / 9 Channel	LED	100 Feet	-	-

Figure 2.2: Product Comparisons

CHAPTER III

PROJECT METHODOLOGY

3.1 Background

Chapter 3 concentrates on how the project is carry on step by step from the start to the finish of this project. This chapter also explains the hardware and software used in this project.

3.2 Methodology

Methodology is the process on how the project is carried out systematically and the figure below show the methodology for the Car Lock Indicator project. Each box presents the components that be used in the project and there are 7 boxes. The transmitter circuit is build based on the transmitter module and IC PT 2262 and the receiver circuit is build based on the receiver module and IC PT 2272. The Microcontroller used is Motorola MC68HC11A1. The input for the car lock indicator is using the push buttons and LED is used as the indicating unit. To communicate with PC and programming, IC MAX 233 and DB9 connector was used. Finally, ASM 11 is used to assemble the program and JBug 11 is used to write, erase and then run the program.

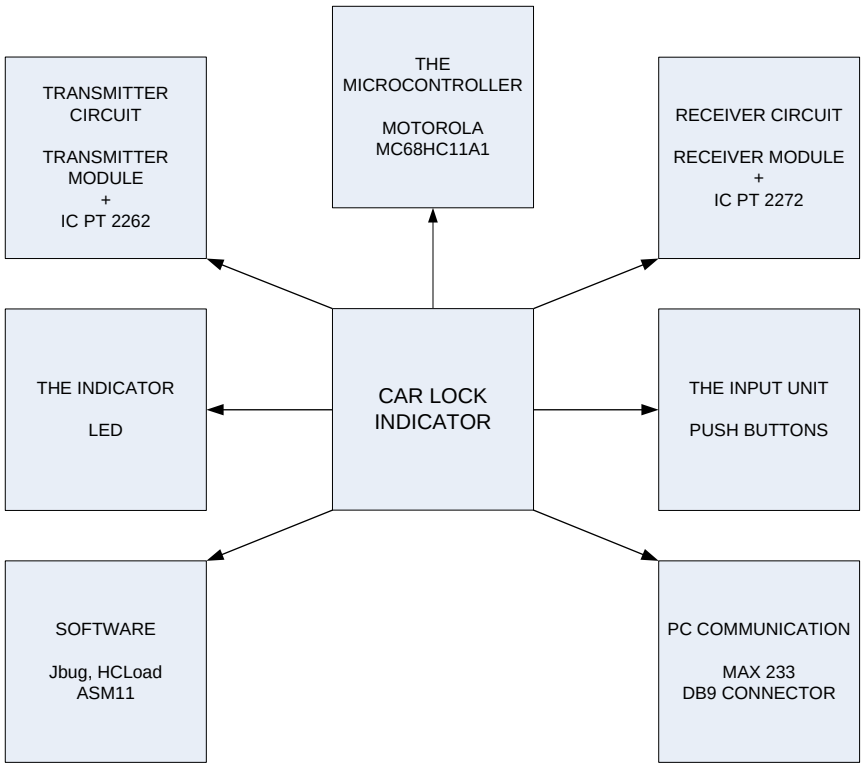


Figure 3.1: The Methodology for the Car Lock Indicator project.

3.2.1 The Microcontroller

A microcontroller (uC) is a computer-on-a-chip used to control electronic devices. It is a type of microprocessor emphasizing self-sufficiency and cost-effectiveness, in contrast to a general-purpose microprocessor used in PC. A typical uC contains all the memory and interfaces needed for a simple application, whereas a general purpose microprocessor requires additional chips to provide these functions.

A microcontroller is a single integrated circuit with the following key features:

i) Central Processing Unit (CPU)

ii) Input and output interfaces such as serial ports

iii) Random Access Memory (RAM) for data storage

iv) Read Only Memory (ROM) for program storage, and

v) Clock Generator, often an oscillator for a quartz timing crystal.

This integration reduces the umbers of chips and the amount of wiring and space that would be needed to produce equivalent systems using separate chips.

There are many microcontrollers available in the market and the most common uC could be found was Intel and Motorola. The uC used for this project is uC Motorola MC68HC11A1 which is in the 6800 family. Below is the architecture of the MC68HC11A8, which is the same architecture with MC68HC11A1.

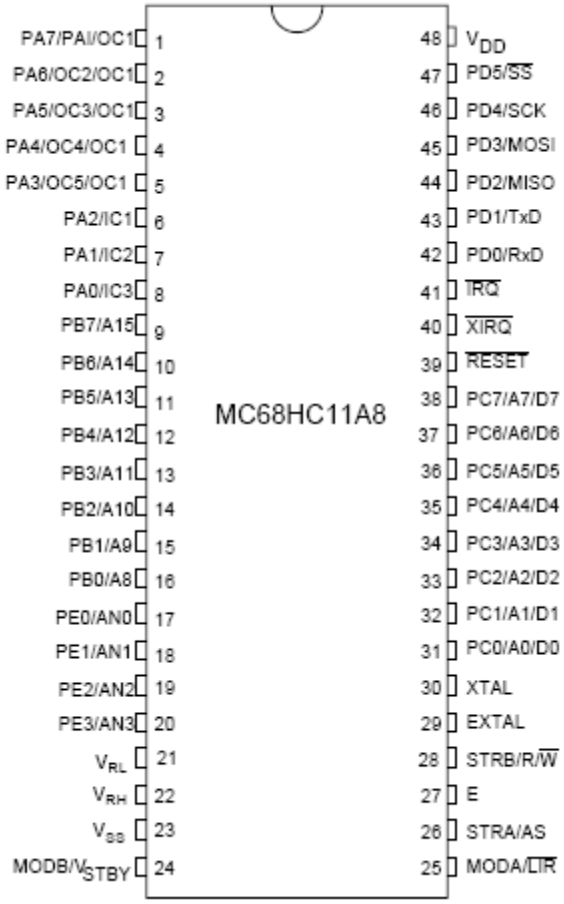


Figure 3.2: The Microcontroller MC68HC11A8 pin assignments

The microcontroller in the figure 3.2 is the dual-in-line package (DIP). It has 48 pins which each pins have its own function. The MC68HC11A1 is used because it capability working in bootstrap mode and there are five ports available on the uC. They are Port A, Port B, Port C, Port D and Port E. The MC68HC11A1 can work as a single chip in bootstrap mode because it has internal EPROM and RAM. The size of the internal EPROM is 512 bytes and the size of the RAM is 256 bytes.

3.2.2 The Transmitter circuit

The transmitter (Tx) circuit contains two important component. The first component is the Tx module 315 MHz and the second component is the remote control encoder, PT 2262. Combining these two components together produce a Tx circuit with four channel of data.

(i) The Transmitter Module.

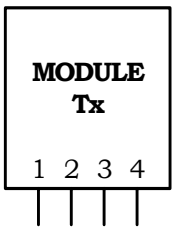


Figure 3.3: The Transmitter Module

The transmitter module is built on a small green board measures 17mm × 13mm × 1mm. The module has four pins and each pin has it own functions. Pin 1 is for the direct current (DC) input of +5 volts (V), Pin 2 is for the grounding of the module, then Pin 3 is for the data input to be transmitted and the final pin which is Pin 4 is to be connected to the antenna to transmit the data.

(ii) The Remote Control Encoder

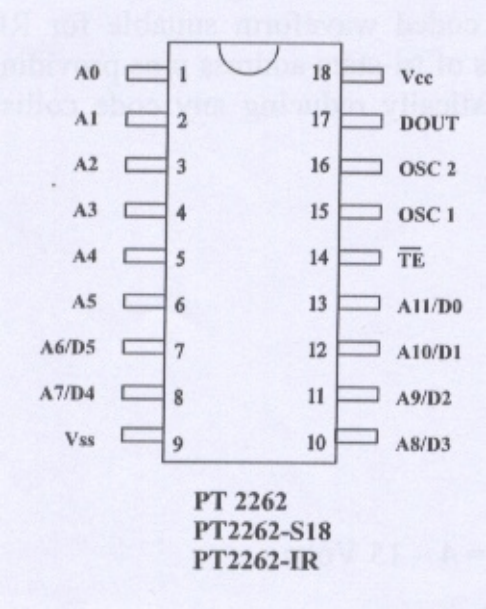


Figure 3.4: PT 2262 pin assignments

PT2262 is a remote control encoder paired with 2272 utilizing CMOS technology. It encodes data and address pins into a serial coded waveform suitable for RF or IR modulation. PT2262 has a maximum of 12-bits of tri-state address pin providing up to 531,441 (3^{12}) address codes; thereby, drastically reducing any code collision and unauthorized code scanning possibilities. Other features of PT2262 can be referred to the chip datasheet.

The chip can be used in various remote controlled applications such as the car security system, garage door controller, remote control fan, home security and automation system, remote control toys and also can be used as remote control for industrial use.

3.2.3 The Receiver Circuit

The receiver (Rx) circuit contains two important components. The first component is the Rx module also 315 MHz and the second component is the remote control decoder, PT 2272. Combining together these two components produce a Rx circuit with four channel of data.

(i) The Receiver Module

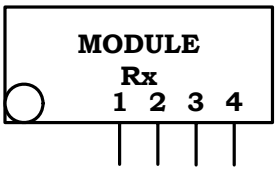


Figure 3.5: The Receiver Module

The Rx module is also built on a small green board measures 14mm× 30mm×1mm. The module has four pins and each pin has it own functions. Pin 1 is for the direct current (DC) input of +5 volts (V), Pin 2 and Pin3 is for the output of data and the final pin which is Pin 4 is the grounding for the module. The hole positioned in the left-bottom of the module is where the antenna should be placed.

(i) The Remote Control Decoder

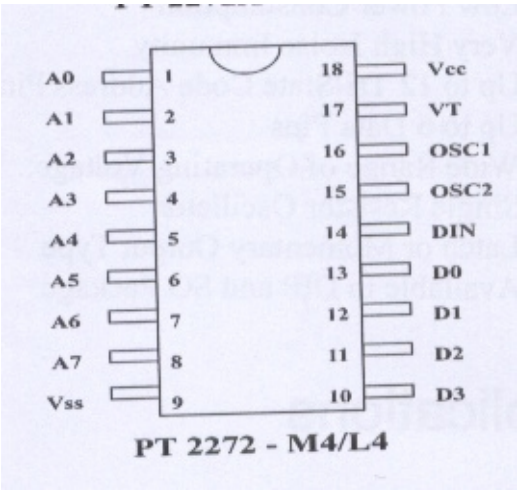


Figure 3.6: PT 2272 pin assignments

Not much different from PT2262, PT2272 is a remote control decoder paired with PT2262 also utilizing CMOS technology. PT2272 have the same tri-state address pins and the same address code. PT2272 is available in several options to suit every application need: variable number of data input pins, latch or momentary output type. Other features of PT2272 can be referred to the chip datasheet.

The chip can be used in various remote controlled applications such as the car security system, garage door controller, remote control fan, home security and automation system, remote control toys and also can be used as remote control for industrial use.

3.2.4 The Input and the Indicator Unit

(i) The Input Unit

The input unit consists of the four push buttons connected to the transmitter circuit. Each button has it own function, from one to the fourth button. The first button is the lock button, the second button is the unlock button, the third is for the car alarm button and finally the fourth button is for the indicator. The input signal from the push buttons is directed to pin 10 to pin 13 of the IC PT 2262. Then the inputted signal will be encoded by the IC PT 2262 and then directed to the Tx module. The module then transmitted the signal via antenna.

(ii) The Indicator Unit

The indicator unit for this project is the well known LED. This component is easy to find in any electronic shop. LED is used because it simplicity and durability to be used for a long time and can stand a rough usage. In this project, four LEDs are used to indicate the four condition of the car status. Firstly the lock condition, second the unlock condition, third the

car alarm, and finally the fourth is the indicator button. The entire four signals for the indicator are received from the remote control decoder, which is IC PT 2272.

3.2.5 PC Communication

The uC development board must be able to communicate with the PC. The components used to communicate the board with the PC are the IC MAX 233 with the DB9 connector via the port 1 on the PC. The circuit for the IC MAX 233 is shown below.

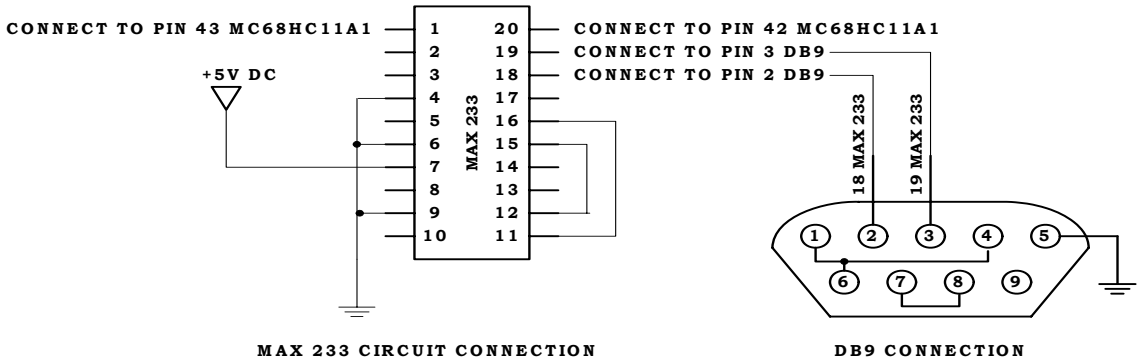


Figure 3.7: IC MAX 233 and DB9 Connection

3.2.6 The Software

Software is one of the elements used in this project. Without the software, it is impossible to program the hardware, especially the uC MC68HC11A1. The uC EPROM must be programmed in order to make the uC operates in the desired manner. Three
softwares were used in this project. They are:

(i) ASM11

(ii) JBug 11

(iii) HC Load

(i) ASM11

ASM11 software runs in MS-DOS application. The main purpose of this software is to convert the written instruction sets in the notepad into assembly language that has the file extension of .S19. Only the .S19 files can be transferred into the uC MC68HC11A1 and then run by the uC.

(ii) JBug 11

JBug 11 software can be run in the normal windows. This software must first be installed using its installer and only after that it can be used. The JBug 11 software must be used together with the MAX 233 circuit and DB9 connector so that it can communicate with the uC development board. The main purpose of this software is to burn, erase and run the program in the internal EPROM of the uC MC68HC11A1.

(iii) HC Load

HC Load software can be run in the normal windows. This software must first be installed using its installer and only after that it can be used. The HC Load software must be used together with the MAX 233 circuit and DB9 connector so that it can communicate with the uC development board. This software purpose is much as same as the JBug 11 software but it has a slight differences. Using HC Load, the interface is much easier than using JBug 11. It can burn, erase, and verify the data in the uC MC68HC11A1 but it cannot run the program.

3.3 The Functional Diagram.

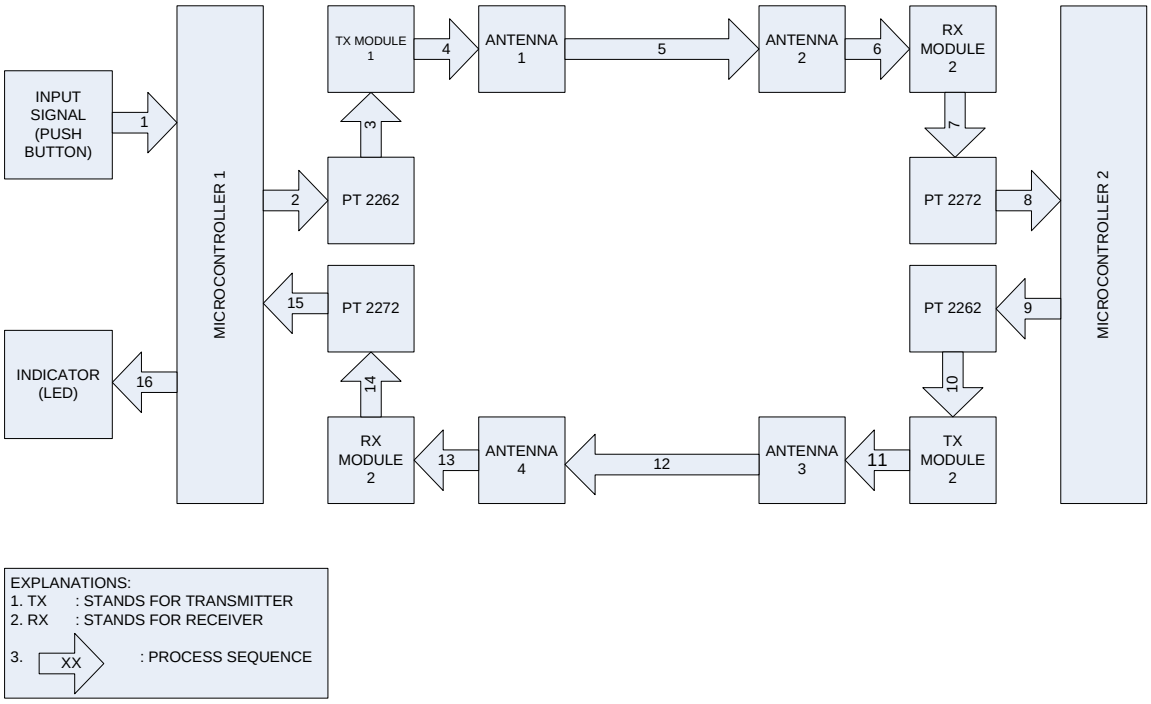


Figure 3.8: The Functional Diagram of the Car Lock Indicator

In the figure 3.8, it describes how the Car Lock Indicator should work. The arrows with their designated numbers show the steps of the system should work. There are 16 steps of the process. It starts from the input signal by the push buttons, going through processes of uC, Rx and Tx circuits, antenna, transmitting and receiving process and finally goes to the indicator.

3.4 The Design Process

The design process show the hierarchical flow of the Car Lock Indicator project. By referring to the design process, the project progress will be in their respective manners and it will be the guide through the project progress. Shown below is the Design process for this project.

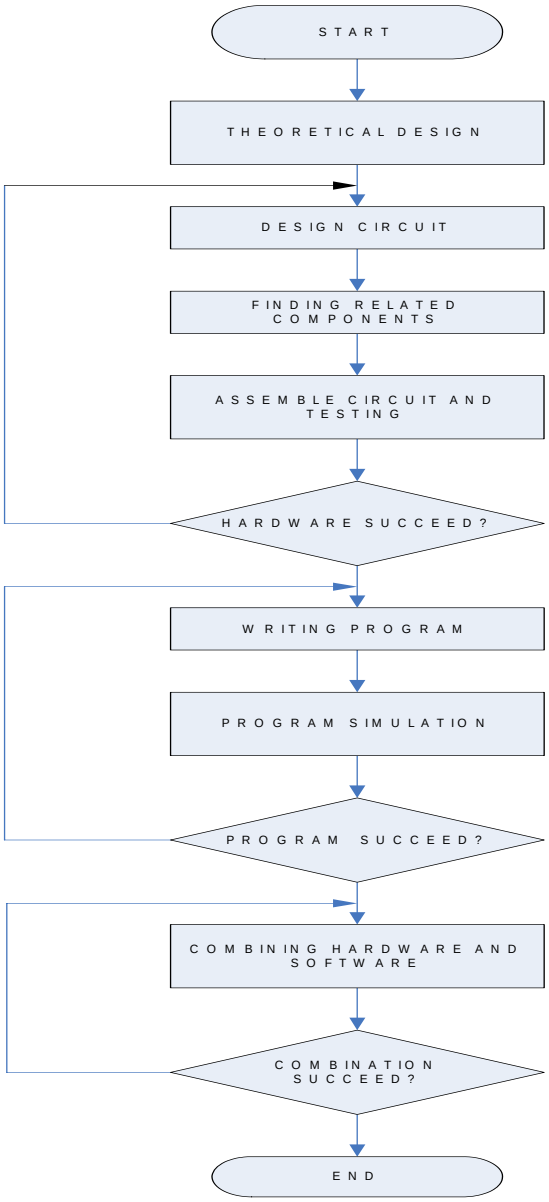


Figure 3.9: The Design Process

3.5 Development of the Remote Unit

As mentioned before, the remote unit is where the user sends the input signal to be processed, transmitted, and then received again by the uC. Then the designated signal will be indicated by the LED indicators.

3.5.1 Circuit Design

The remote unit circuit design starts with the assembling of the Tx1 circuit and the the Rx2 circuit. When it finished, the process go to the assembling of the uC Motorola MC68HC11A1 in bootstrap mode. When successful, all of the circuits is combined together and produce the complete circuit for the remote unit of the Car Lock Indicator.

3.5.2 Program Design

The program is basically on how to transmit the signal to the uC, then to the remote control encoder PT 2262, then to the module and finally to the antenna. After that, it focuses on the process of receiving back the signal, then to the decoder PT 2272, then to the uC and then sent to the LED indicators.

3.6 Development of the Car Unit

The car unit is the unit of the Car Lock Indicator that stationed in the car. This unit communicate with the remote unit and then processed the signal and then send it back to the remote by its Tx and antenna. This unit will process the signal according the data received.

3.6.1 Circuit Design

The remote unit circuit design starts with the assembling of the Rx1 circuit and the the Tx2 circuit. When it finished, the process go to the assembling of the uC Motorola MC68HC11A1 in bootstrap mode. When successful, all of the circuits is

combined together and produce the complete circuit for the car unit of the Car Lock Indicator.

3.6.2 Program Design

The program is basically on how to receive the signal from the antenna, and then to the remote control decoder PT 2272, then to the uC. The uC process the signal, sent it to the encoder PT 2262 and then to the antenna.

CHAPTER IV

PROJECT RESULTS

4.1 Background

This chapter presents the results of the Car Lock Indicator project. The results are divided by the Remote unit and the Car unit. The result includes the circuit of the uC MC68HC11A1 bootstrap mode, Tx circuit and Rx circuits.

4.2 MC68HC11A1 Bootstrap Mode

The values of the mode select inputs MODB and MODA during reset determine the operating mode. Bootstrap mode will be used for this project as it has a simple circuit, rather than expanded mode which require a complicated circuit. Furthermore, as this project need not to save a large memory in EEPROM, it is better to use this bootstrap mode. Expanded mode will be used only when we want to save a large data/memory in EEPROM.

To operate in this mode, both MODA and MODB pin of microcontroller need to be connected to the ground. In this mode, microcontroller doesn’t need an outsider data bus/outsider address to be implementing.

When the MCU is reset in special bootstrap mode, a small on-chip read-only memory (ROM) is enabled at address $BF00–$BFFF. The ROM contains a bootloader program and a special set of interrupt and reset vectors. The MCU fetches the reset vector, and then executes the bootloader.

Bootstrap mode is a special variation of the single-chip mode. Bootstrap mode allows special-purpose programs to be entered into internal random-access memory (RAM). When bootstrap mode is selected at reset, a small bootstrap ROM becomes present in the memory map. Reset and interrupt vectors are located in this ROM at $BFC0–$BFFF.

The bootstrap ROM contains a small program which initializes the serial communications interface (SCI) and allows the user to download a program into on-chip RAM. The size of the downloaded program can be as large as the size of the on-chip RAM. After a 4-character delay, or after receiving the character for the highest address in RAM, control passes to the loaded program at $0000.

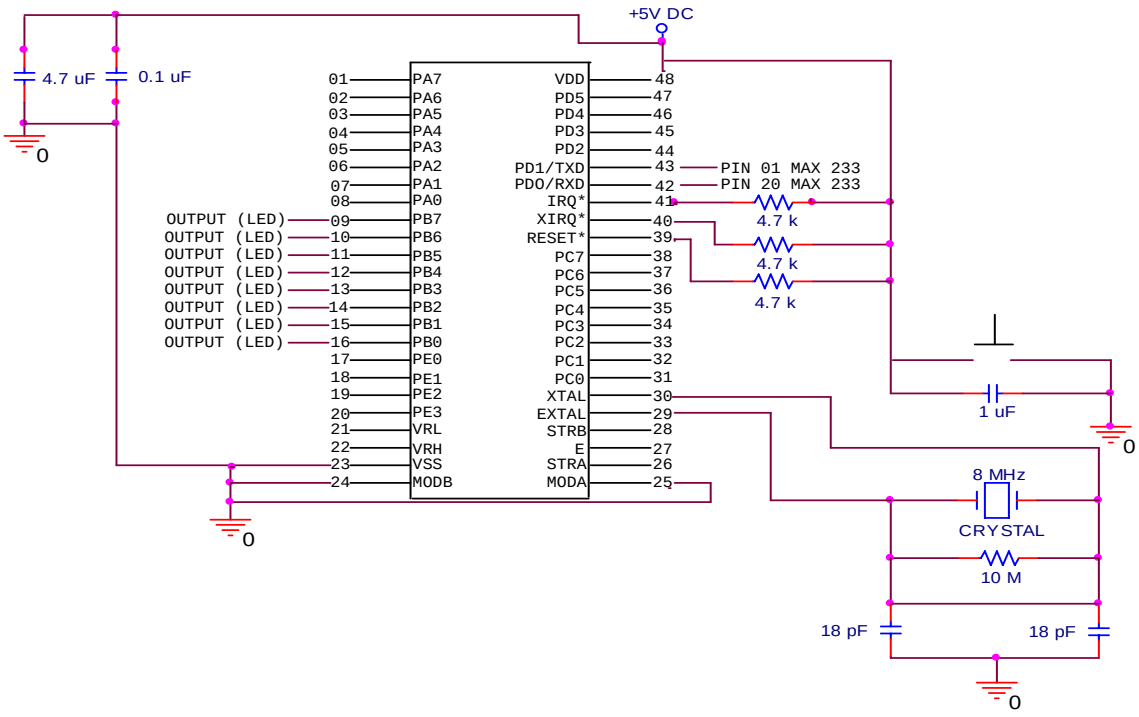


Figure 4.1: MC68HC11A1 in Bootstrap Mode

4.2.1 Results MC68HC11A1 in Bootstrap Mode

In determining the bootstrap mode working or not, we must first check output from pin 27 that is the pin for E-Clock. The frequency at that pin should be ¼ of the

frequency of the crystal used that is 2 MHz. Using the oscilloscope, the output frequency at pin 27 is confirmed at 2MHz.

4.3 The Transmitter circuit

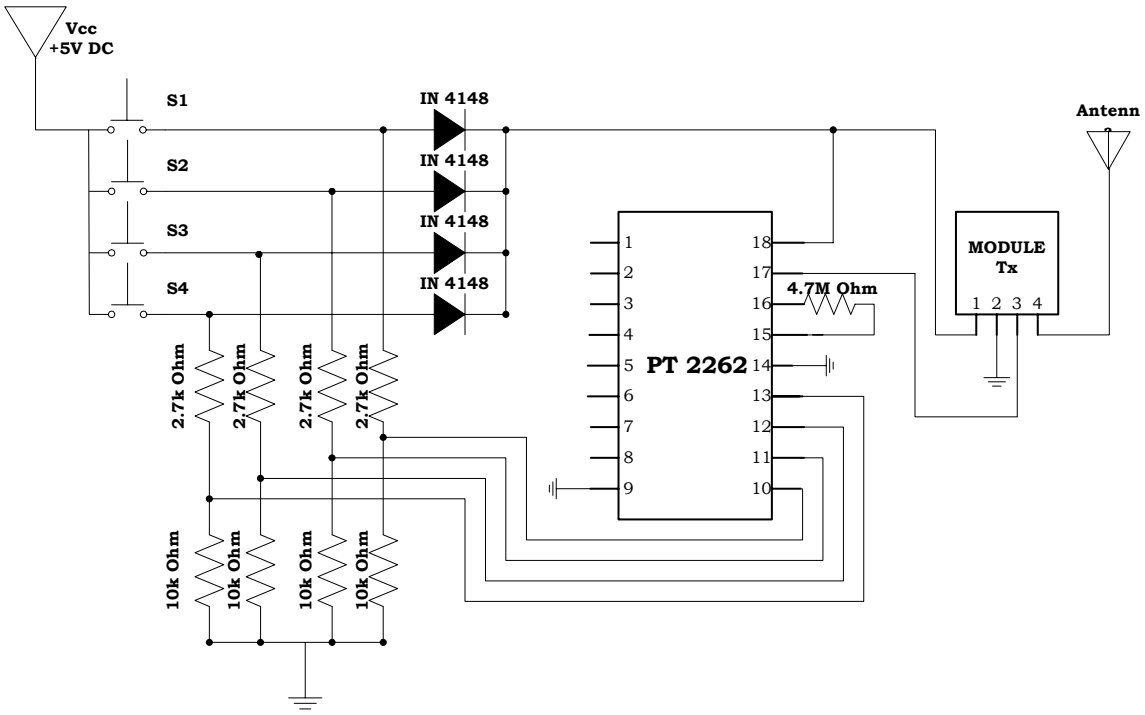


Figure 4.2: The Transmitter Circuit

4.3.1 Testing method

The Transmitter circuit is tested by measuring the voltage at the pin 10, 11, 12 and 13 of the PT 2262 chip by pressing switch 1, 2, 3, and 4 respectively and the result can be seen when the indicator LEDs blinking according to their positions.

4.3.2 Results of the Transmitter Circuit

Input	*PT 2262 Pin*	*Voltage*
S1	10 (D3)	3.97 V
S2	11 (D2)	3.94 V
S3	12 (D1)	3.98 V
S4	13 (D0)	4.05 V

Figure 4.3: Table of Transmitter circuit results

4.4 The Receiver Circuit

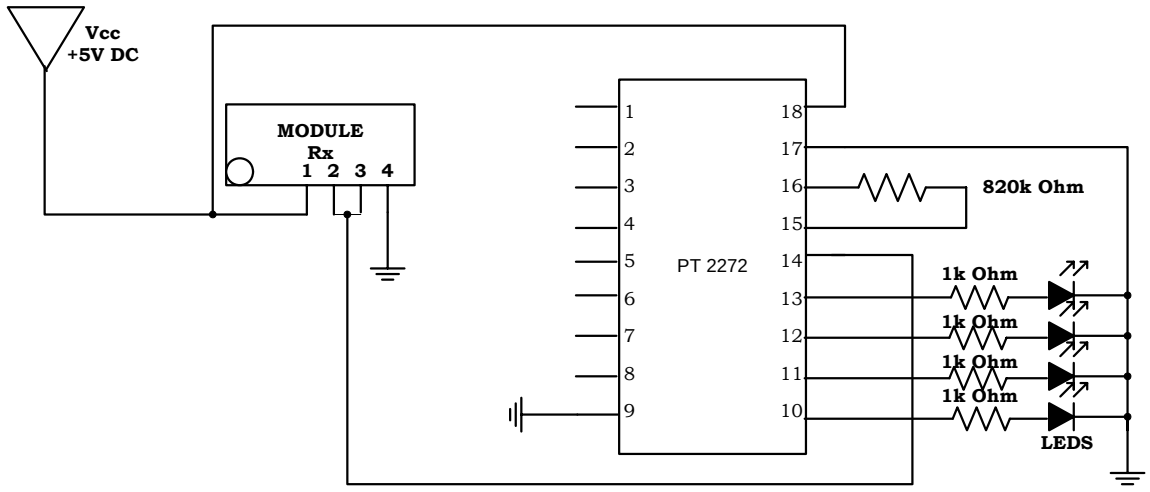


Figure 4.4: The Receiver Circuit

4.4.1 Testing method

The receiver circuit is tested by measuring the voltage at the pin 10, 11, 12 and 13 of the PT 2272 chip by pressing switch 1, 2, 3, and 4 respectively and the result can be seen when the indicator LEDs blinking according to their positions.

4.4.2 Results of the Receiver Circuit

Input	*PT 2272 Pin*	*Voltage*
Pin 2 and 3 Rx	10 (D3)	4.35 V
Pin 2 and 3 Rx	11 (D2)	4.35 V
Pin 2 and 3 Rx	12 (D1)	4.35 V
Pin 2 and 3 Rx	13 (D0)	4.35 V

Figure 4.5: Table of Receiver circuit results

4.5 The Remote Unit

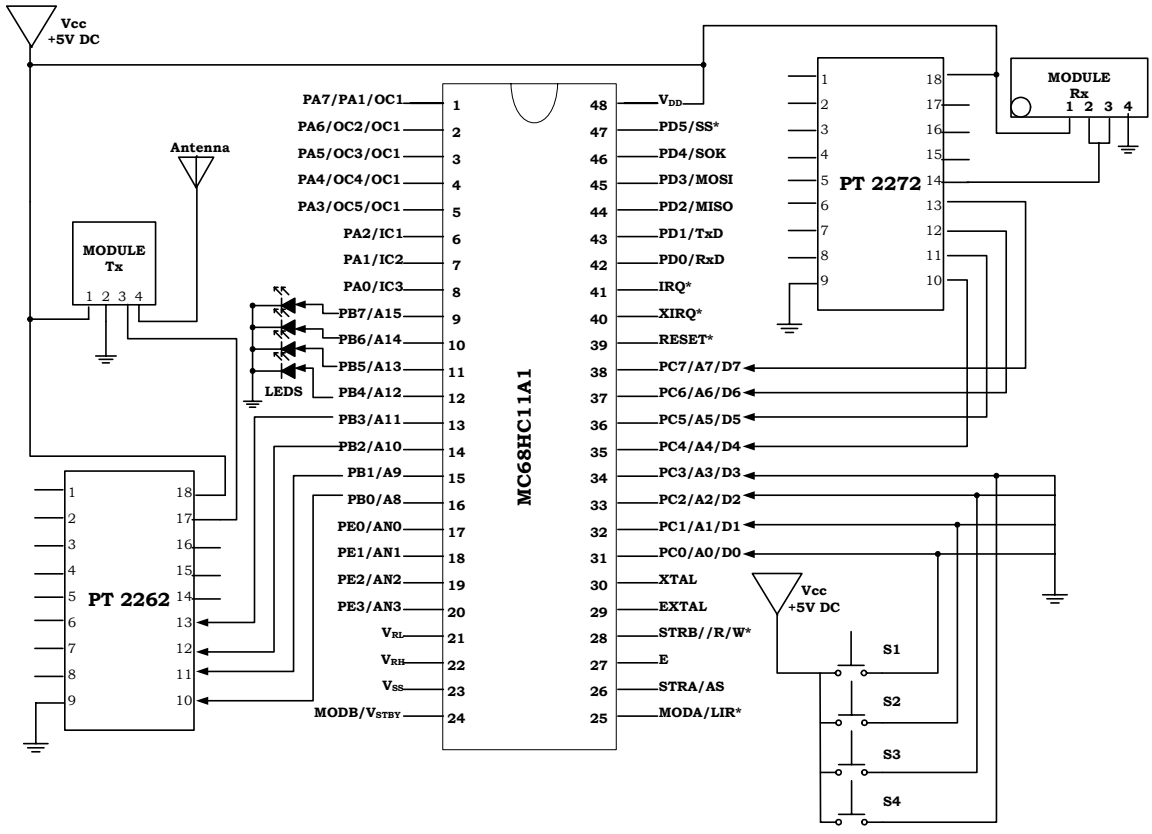


Figure 4.6: The Remote Unit Circuit

Figure 4.6 above show the connection of the remote unit. Take note that the uC was already in the bootstrap mode. This circuit already assembled and the connection is according to the figure 4.6. Now it depends on the programming of the uC to determine the function of the remote unit. The programming sets of the remote unit can be found in appendix F.

4.6 The Car Unit

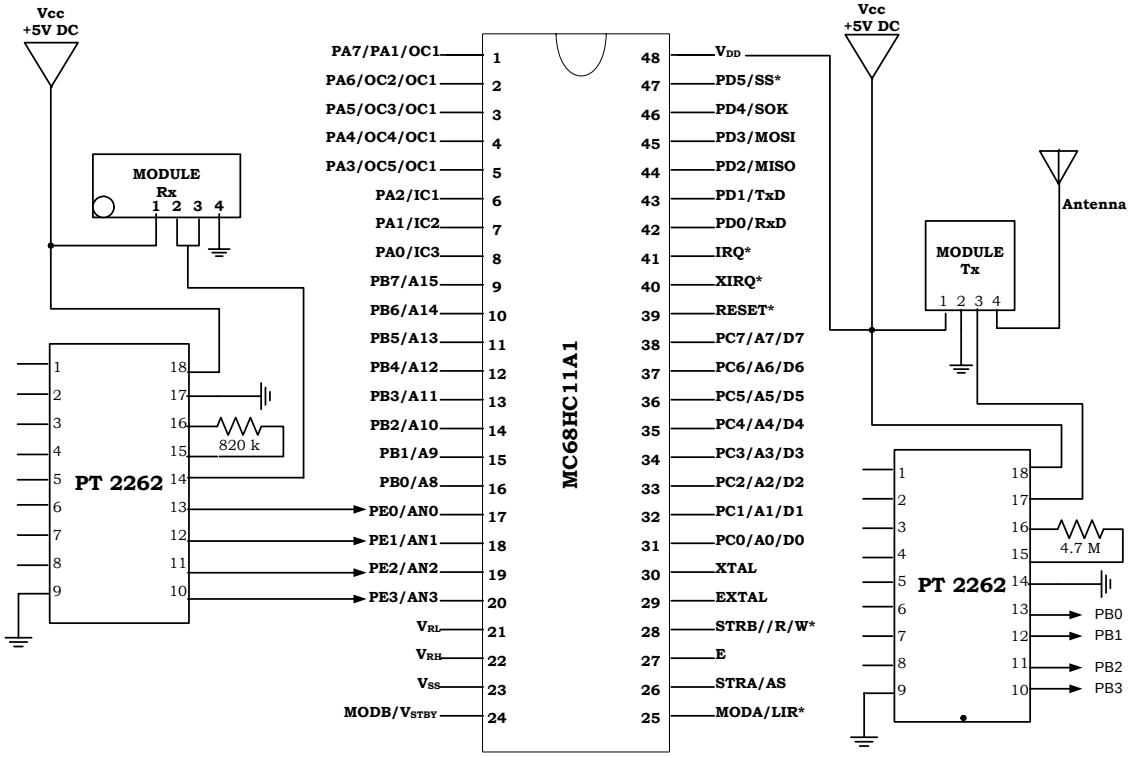


Figure 4.7: The Car Unit Circuit

Figure 4.7 above show the connection of the car unit. Take note that the uC was already in the bootstrap mode. This circuit already assembled and the connection is according to the figure 4.7. Now it depends on the programming of the uC to determine the function of the remote unit. The programming sets of the Car Unit can be found in appendix H.

4.7.1 The Obstacles

In order to make the remote unit and the car unit, it depends on the successful of the circuit design, programming and the most critical component of the system is the Tx and Rx module. During the testing and troubleshooting process, both modules have experienced failure and broken. The only problems to get a new modules is time and cost. The modules experienced failure maybe due to:

(i) Over voltage at the input voltage pin because it directly connected to the voltage source.

(ii) There is o protection from the source to the module, such as by putting a diode between them.

CHAPTER V

CONCLUSION AND RECOMMENDATION

5.1 Background

This chapter is the final chapter of this thesis. It concludes all about the project from the start to the end, and discusses about the project meets the objective or not. The recommendation also included in order to make this project become more advance and successful if it can be continued by other degree student.

5.2 Conclusion

The Car Lock Indicator project is not a new idea. Many products can be found in the internet and car accessories shop around, utilize more advance and expensive hardware and components. But the Car Lock Indicator project opens a new horizon in the local car manufacturers. Although the project did not fully functioning, but it meets the objective because the project have a prototype of the remote unit and the car unit.

5.3 Contribution

The first contribution of the project is to the local car manufacturers. They can further improve this product by doing research and development process and the

can implement it to the local cars as the standard features so it can make our local car industries become more competitive and unique.

The second contribution is to the car owner and the future car buyer. In this fast moving world today make the people always in a hurry and sometimes they become un-aware of their car. They could easily forget and start asking questions, “Did I lock the car?” and if they had the car lock indicator, the question will not arise in their head because they can check the lock status of their car anytime by just checking with the car lock indicator.

5.4 Recommendation for Future Works

In the future, whoever continues this project must think of the improvements so that the car lock indicator has more features. The development must take account about the cost, flexibility, functions, and user friendly. There are few recommendations for the future works and they are stated below:

(i) Using LCD as the Indicator

LCD is one of the common interface that is easy to understand. It can display black numbers and characters with the light background. The LCD usage can utilize the ability of the uC MC68HC11A1 as the powerful 8-bit microcontroller. The compromise here is the battery life span that could be shortened due to higher voltage usage by the LCD display unit.

(ii) Combining with the Remote Car Starter

The car lock indicator project can be enhanced by combining it with the remote car starter project so that this device has more functions and flexibility.

(iii) Smaller and cheaper

Do not limit the hardware by just using uC MC68HC11A1, because there is many other type of uC and controller could be used, such as PIC and Basic Stamp controller. They are smaller, cheaper and the project builder can learn a new software and hardware design.

REFERENCES

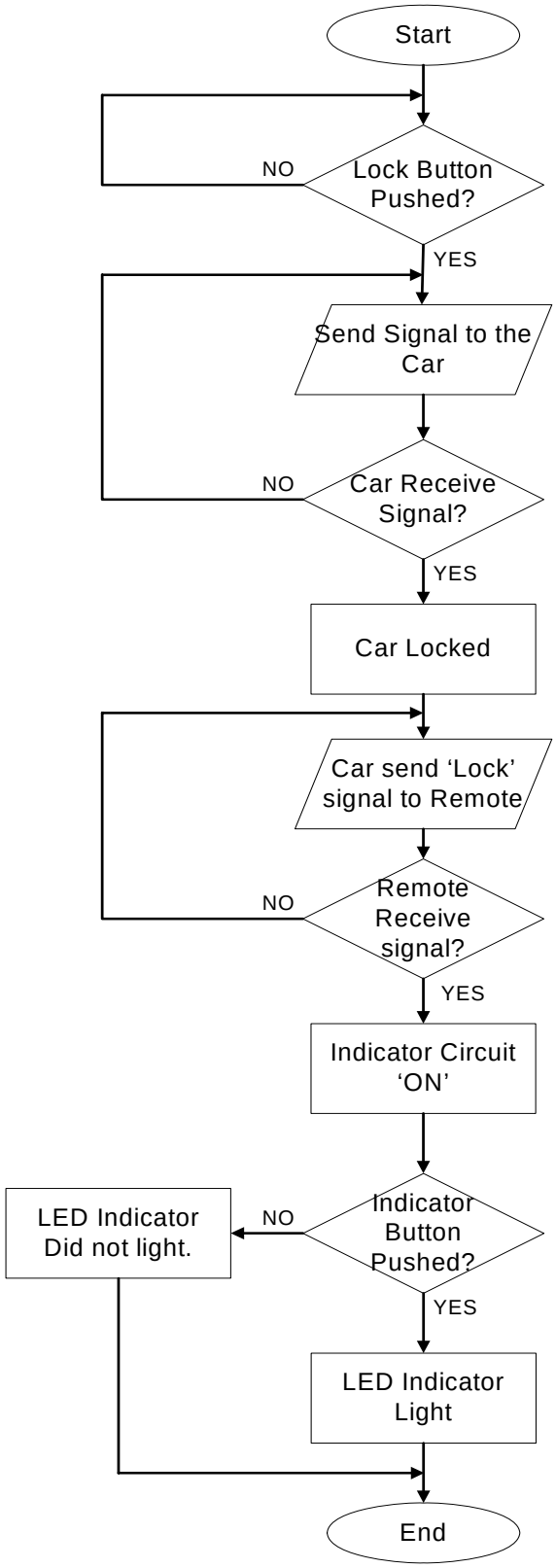
BOOKS

1. Microcontroller Technology, The 68HC11 and 68HC12, Fifth Edition, Peter Spasov, Prentice Hall, 2004.
2. Encyclopedia of Electric Circuits, Volume 6, Rudolf F. Graf & William Sheets, Mc Graw Hill, 1996.
3. Encyclopedia of Electric Circuits, Volume 7, Rudolf F. Graf & William, Mc Graw Hill, 1985.

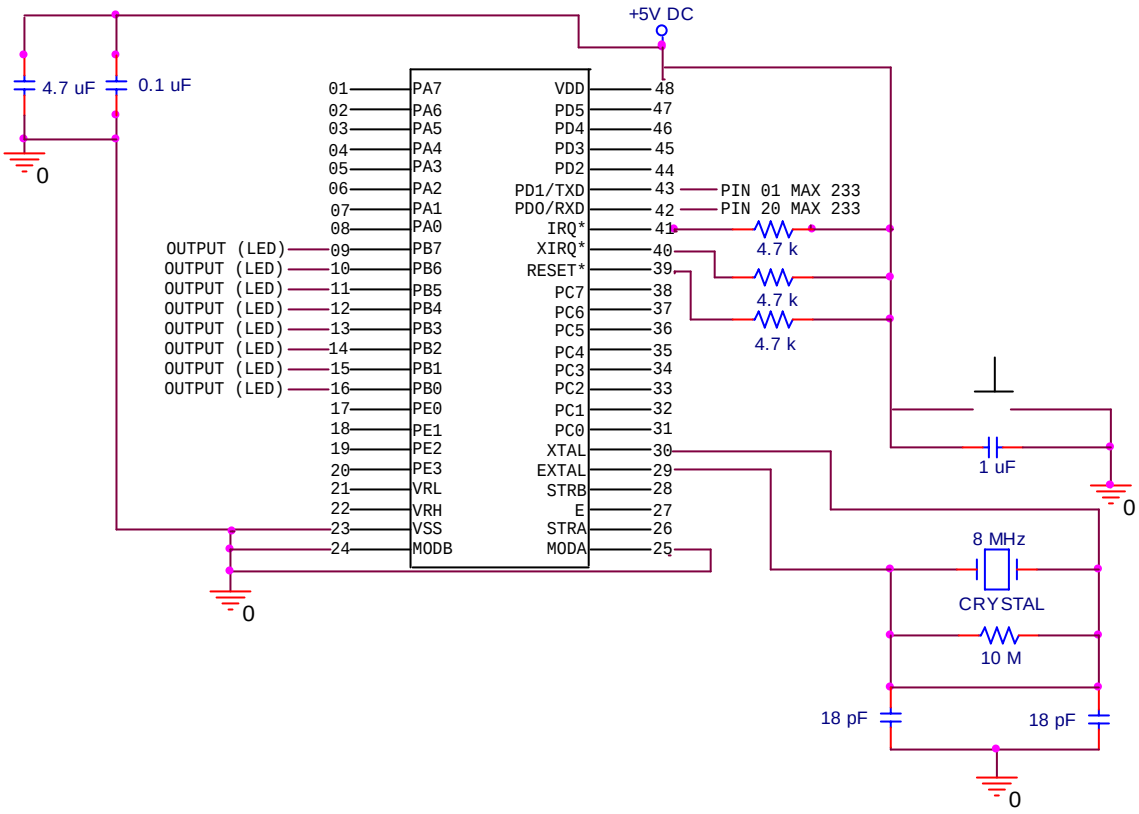
INTERNET

1. www.DesignNotes.com
2. www.howstuffworks.com
3. www.carmobilevideo.com
4. www.electronicslab.com
5. www.wikipedia.org
6. www.rmp.gov.my
7. www.jpj.gov.my

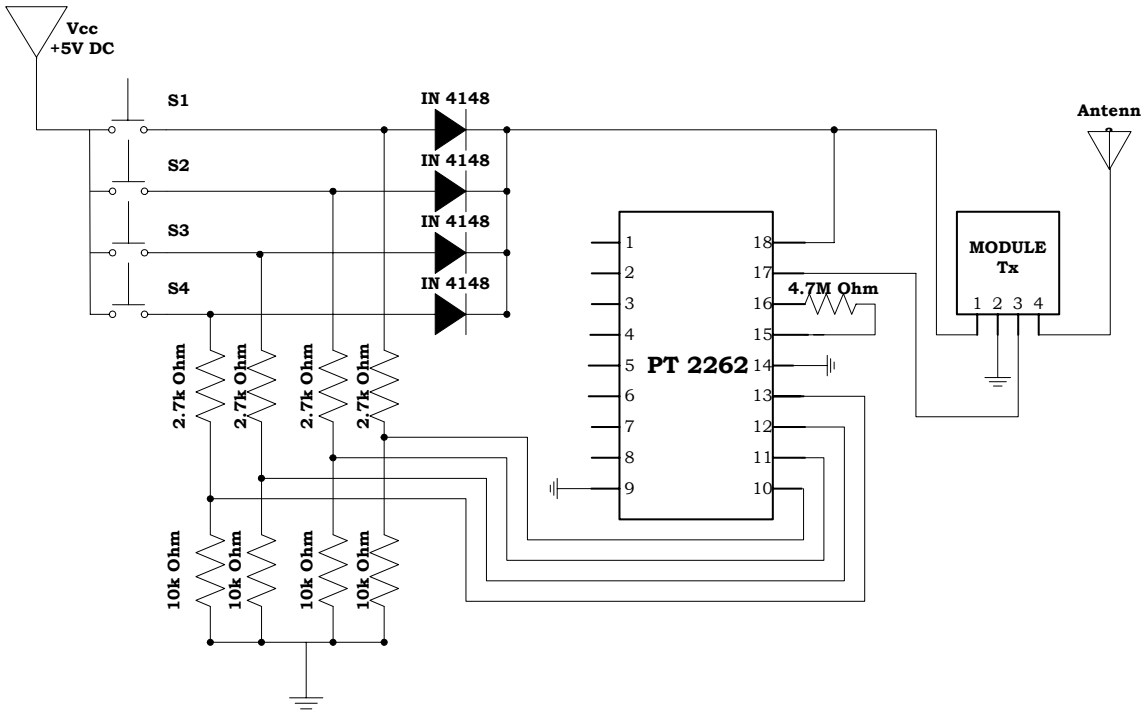
Appendix A
The Flow Chart: Car Lock Indicator Process



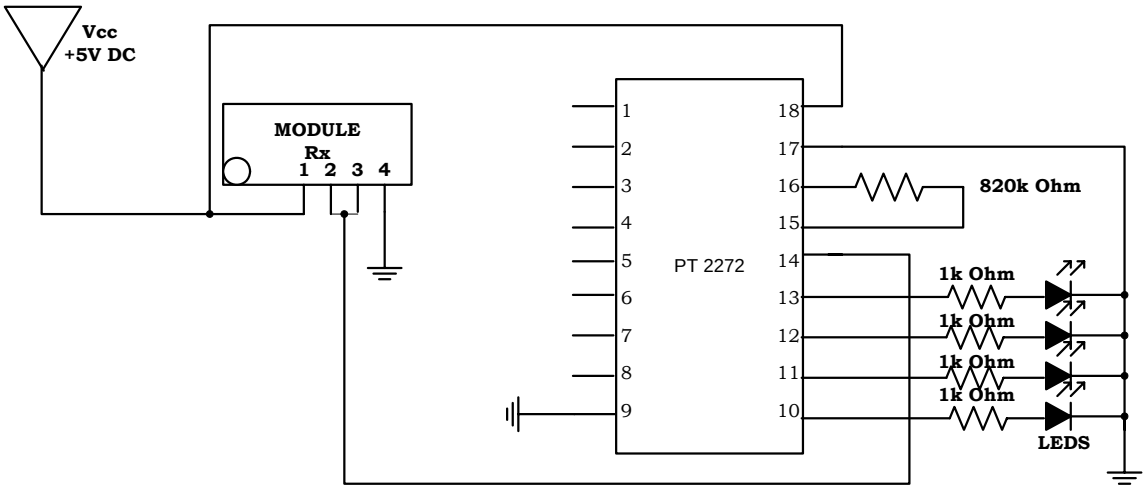
Appendix B
MC68HC11A1 Bootstrap Mode Circuit



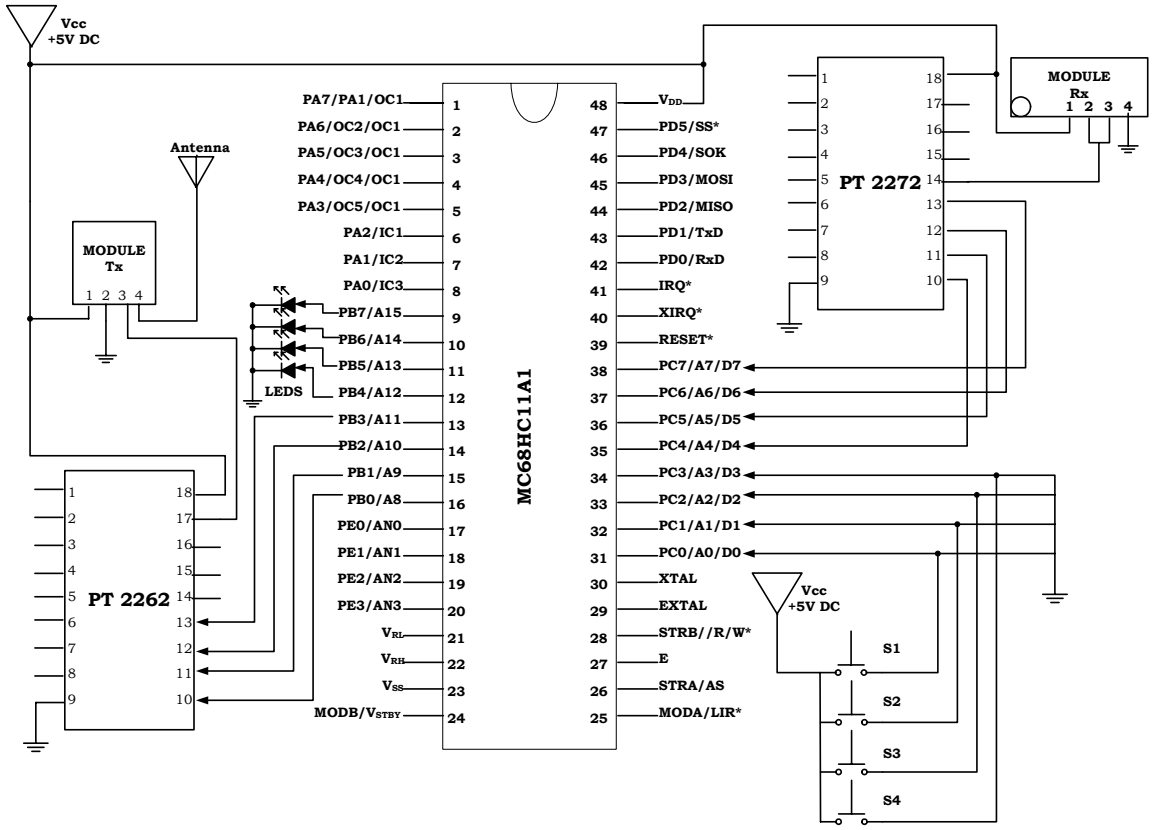
Appendix C
The Transmitter Circuit



Appendix D
The Receiver Circuit



Appendix E
The Circuit of Remote Unit



Appendix F
The Program of Remote Unit

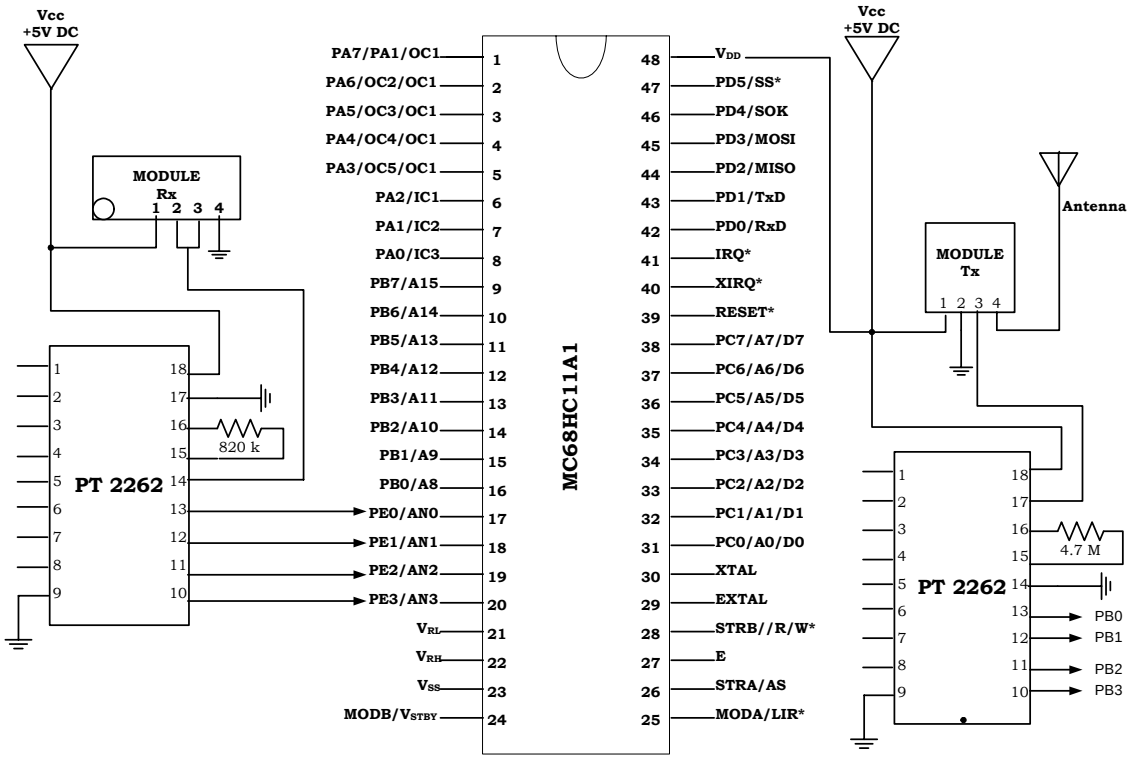
File REMOTE.ASM þ By ASM11 v1.84d [Thursday, April 13, 2006 12:43 am]

```
 1     B600                ORG         $B600
 2     1000                PORTA       EQU    $1000
 3     0004                PORTB       EQU    $4
 4     0003                PORTC       EQU    $3
 5     0007                DDRC        EQU    $7
 6     000A                PORTE       EQU    $A
 7
 8
 9 B600:CE10 00            [ 3]        LDX    #PORTA
10 B603:8600               [ 2]        LOOP   LDAA   #00
11 B605:A707               [ 4]        STAA   DDRC,X
12 B607:8600               [ 2]        LDAA   #00
13 B609:A704               [ 4]        STAA   PORTB,X
14 B60B:1E03 010E          [ 7] SCAN   BRSET  PORTC,X,$01,TX1
15 B60F:1E03 0219          [ 7]        BRSET  PORTC,X,$02,TX2
16 B613:1E03 0424          [ 7]        BRSET  PORTC,X,$04,TX3
17 B617:1E03 082F          [ 7]        BRSET  PORTC,X,$08,TX4
18 B61B:20E6 (B603)        [ 3]        BRA    LOOP
19
20 B61D:1C04 00            [ 7] TX1    BSET   PORTB,X,$00
21 B620:8601               [ 2] AGAIN1 LDAA   #01
22 B622:A704               [ 4]        STAA   PORTB,X
23 B624:BDB6 AE            [ 6]        JSR    DELAY
24 B627:26F7 (B620)        [ 3]        BNE    AGAIN1
25 B629:7EB6 0B            [ 3]        JMP    SCAN
26
27 B62C:1C04 00            [ 7] TX2    BSET   PORTB,X,$00
28 B62F:8602               [ 2] AGAIN2 LDAA   #02
29 B631:A704               [ 4]        STAA   PORTB,X
30 B633:BDB6 AE            [ 6]        JSR    DELAY
31 B636:26F7 (B62F)        [ 3]        BNE    AGAIN2
32 B638:7EB6 0B            [ 3]        JMP    SCAN
33
34 B63B:1C04 00            [ 7] TX3    BSET   PORTB,X,$00
35 B63E:8604               [ 2] AGAIN3 LDAA   #04
36 B640:A704               [ 4]        STAA   PORTB,X
37 B642:BDB6 AE            [ 6]        JSR    DELAY
38 B645:26F7 (B63E)        [ 3]        BNE    AGAIN3
39 B647:7EB6 0B            [ 3]        JMP    SCAN
40
41 B64A:1C04 00            [ 7] TX4    BSET   PORTB,X,$00
42 B64D:8608               [ 2] AGAIN4 LDAA   #08
43 B64F:A704               [ 4]        STAA   PORTB,X
44 B651:BDB6 AE            [ 6]        JSR    DELAY
45 B654:26F7 (B64D)        [ 3]        BNE    AGAIN4
```

```
46 B656:7EB6 0B          [ 3]             JMP    SCAN
47
48 B659:CE10 00          [ 3]             LDX    #PORTA
49 B65C:8600             [ 2] LOOP2       LDAA   #00
50 B65E:A704             [ 4]             STAA   PORTB,X
51 B660:1E03 100E        [ 7] SCAN2       BRSET  PORTC,X,$10,RX5
52 B664:1E03 2019        [ 7]             BRSET  PORTC,X,$20,RX6
53 B668:1E03 4024        [ 7]             BRSET  PORTC,X,$40,RX7
54 B66C:1E03 802F        [ 7]             BRSET  PORTC,X,$80,RX8
55 B670:20EA (B65C)      [ 3]             BRA    LOOP2
56
57 B672:1C04 00          [ 7] RX5         BSET   PORTB,X,$00
58 B675:860A             [ 2] AGAIN5      LDAA   #10
59 B677:A704             [ 4]             STAA   PORTB,X
60 B679:BDB6 AE          [ 6]             JSR    DELAY
61 B67C:26F7 (B675)      [ 3]             BNE    AGAIN5
62 B67E:7EB6 0B          [ 3]             JMP    SCAN
63
64 B681:1C04 00          [ 7] RX6         BSET   PORTB,X,$00
65 B684:8614             [ 2] AGAIN6      LDAA   #20
66 B686:A704             [ 4]             STAA   PORTB,X
67 B688:BDB6 AE          [ 6]             JSR    DELAY
68 B68B:26F7 (B684)      [ 3]             BNE    AGAIN6
69 B68D:7EB6 0B          [ 3]             JMP    SCAN
70
71 B690:1C04 00          [ 7] RX7         BSET   PORTB,X,$00
72 B693:8628             [ 2] AGAIN7      LDAA   #40
73 B695:A704             [ 4]             STAA   PORTB,X
74 B697:BDB6 AE          [ 6]             JSR    DELAY
75 B69A:26F7 (B693)      [ 3]             BNE    AGAIN7
76 B69C:7EB6 0B          [ 3]             JMP    SCAN
77
78 B69F:1C04 00          [ 7] RX8         BSET   PORTB,X,$00
79 B6A2:8650             [ 2] AGAIN8      LDAA   #80
80 B6A4:A704             [ 4]             STAA   PORTB,X
81 B6A6:BDB6 AE          [ 6]             JSR    DELAY
82 B6A9:26F7 (B6A2)      [ 3]             BNE    AGAIN8
83 B6AB:7EB6 0B          [ 3]             JMP    SCAN
84
85 B6AE:18CE 4444        [ 4] DELAY       LDY    #$4444
86 B6B2:1809             [ 4] SEMULA      DEY
87 B6B4:26FC (B6B2)      [ 3]             BNE    SEMULA
88 B6B6:39               [ 5]             RTS
89     0000                               END
90
91
92
93
94
95
```

```
  96
  97
  98

------------------------------------------------------------

        S E G M E N T   U S A G E   R E P O R T

Segment  Start   End    Size  CodObj  DatObj  TotObj  Lines
-------  -----  -----  -----  ------  ------  ------  -----
ROM      $B600  $B6B6  $00B7   $00B7   $0000   $00B7     98

Summary  $B600  $B6B6  $00B7   $00B7   $0000   $00B7     98

------------------------------------------------------------

        O V E R A L L   M E M O R Y   U S A G E

Total RAM   size:      0   $0000    0.00 KB
Total CODE  size:    183   $00B7    0.18 KB   (70 instructions)
Total DATA  size:      0   $0000    0.00 KB
Total IMAGE size:    183   $00B7    0.18 KB

Lowest address  :  46592   $B600
Highest address :  46774   $B6B6
Address Range   :    183   $00B7    0.18 KB   (Used Range: 100.00%)

Number of INCLUDE files: 0

Assembled 98 lines (No Errors, Warnings: 12)

                *** End of REMOTE.ASM listing ***
```

Appendix G
The Circuit of Car Unit



Appendix H
The Program of Car Unit

File CAR.ASM þ By ASM11 v1.84d [Thursday, April 13, 2006 2:07 pm]

```
 1     B600                ORG         $B600
 2     1000                PORTA       EQU    $1000
 3     0004                PORTB       EQU    $4
 4     0003                PORTC       EQU    $3
 5     0007                DDRC        EQU    $7
 6     000A                PORTE       EQU    $A
 7
 8 B600:CE10 00            [ 3]        LDX    #PORTA
 9 B603:8600               [ 2] LOOP   LDAA   #00
10 B605:A704               [ 4]        STAA   PORTB,X
11 B607:1E0A 010E          [ 7] SCAN   BRSET  PORTE,X,$01,RX1
12 B60B:1E0A 020A          [ 7]        BRSET  PORTE,X,$02,RX1
13 B60F:1E0A 0418          [ 7]        BRSET  PORTE,X,$04,RX3
14 B613:1E0A 081D          [ 7]        BRSET  PORTE,X,$08,RX4
15 B617:20EA (B603)        [ 3]        BRA    LOOP
16
17 B619:1C04 00            [ 7] RX1    BSET   PORTB,X,$00
18 B61C:8601               [ 2]        LDAA   #01
19 B61E:A704               [ 4]        STAA   PORTB,X
20 B620:20E5 (B607)        [ 3]        BRA    SCAN
21
22 B622:1C04 00            [ 7] RX2    BSET   PORTB,X,$00
23 B625:8602               [ 2]        LDAA   #02
24 B627:A704               [ 4]        STAA   PORTB,X
25 B629:20DC (B607)        [ 3]        BRA    SCAN
26
27 B62B:1C04 00            [ 7] RX3    BSET   PORTB,X,$00
28 B62E:8604               [ 2]        LDAA   #04
29 B630:A704               [ 4]        STAA   PORTB,X
30 B632:20D3 (B607)        [ 3]        BRA    SCAN
31
32 B634:1C04 00            [ 7] RX4    BSET   PORTB,X,$00
33 B637:8608               [ 2]        LDAA   #08
34 B639:A704               [ 4]        STAA   PORTB,X
35 B63B:20CA (B607)        [ 3]        BRA    SCAN
36
37 B63D:18CE 4444          [ 4] DELAY  LDY    #$4444
38 B641:1809               [ 4] SEMULA DEY
39 B643:26FC (B641)        [ 3]        BNE    SEMULA
40 B645:39                 [ 5]        RTS
41      0000                           END
42
43
44
45
```

```
  46
  47
  48
  49
  50

------------------------------------------------------------

      S E G M E N T   U S A G E   R E P O R T

Segment  Start   End    Size  CodObj  DatObj  TotObj  Lines
-------  -----  -----  -----  ------  ------  ------  -----
ROM      $B600  $B645  $0046   $0046   $0000   $0046     50

Summary  $B600  $B645  $0046   $0046   $0000   $0046     50

------------------------------------------------------------

      O V E R A L L   M E M O R Y   U S A G E

Total RAM   size:      0   $0000    0.00 KB
Total CODE  size:     70   $0046    0.07 KB   (28 instructions)
Total DATA  size:      0   $0000    0.00 KB
Total IMAGE size:     70   $0046    0.07 KB

Lowest address  :  46592   $B600
Highest address :  46661   $B645
Address Range   :     70   $0046    0.07 KB   (Used Range: 100.00%)

Number of INCLUDE files: 0

Assembled 50 lines (No Errors, Warnings: 0)

                *** End of CAR.ASM listing ***
```

Appendix I
MC68HC11A1

Figure 2-1 shows the pin assignments for the MC68HC11A8 in the 52-pin PLCC package and the 48-pin DIP package.

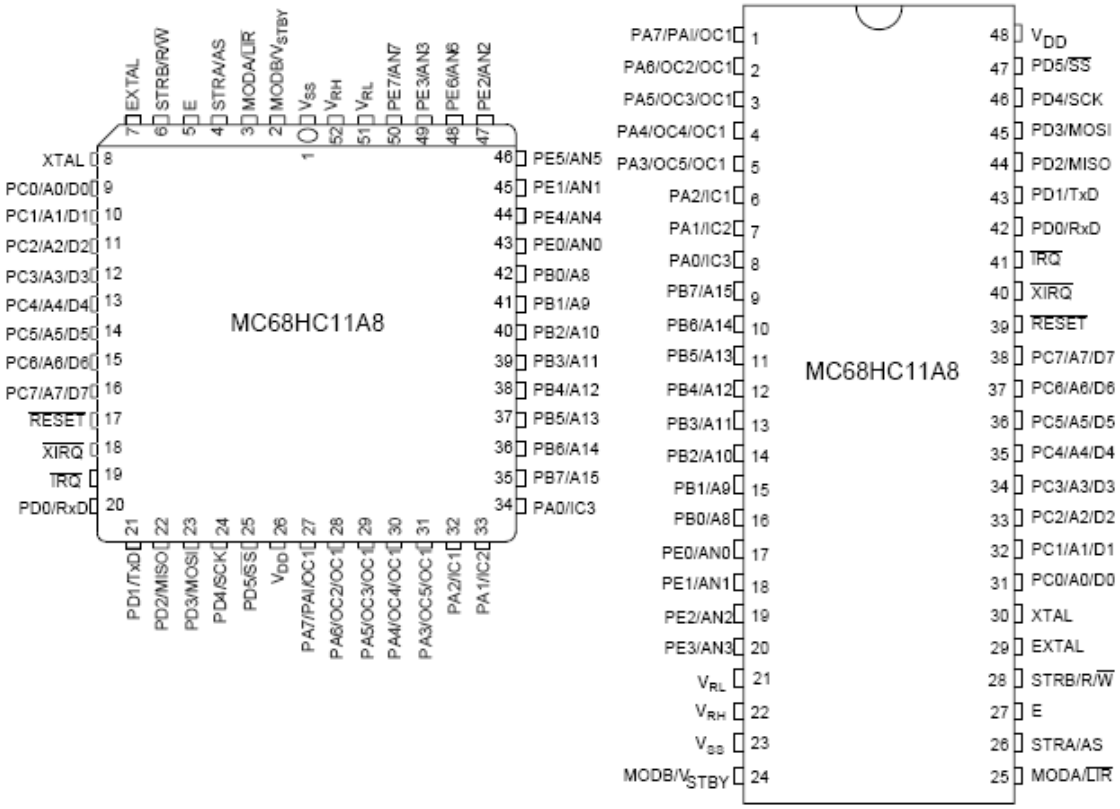
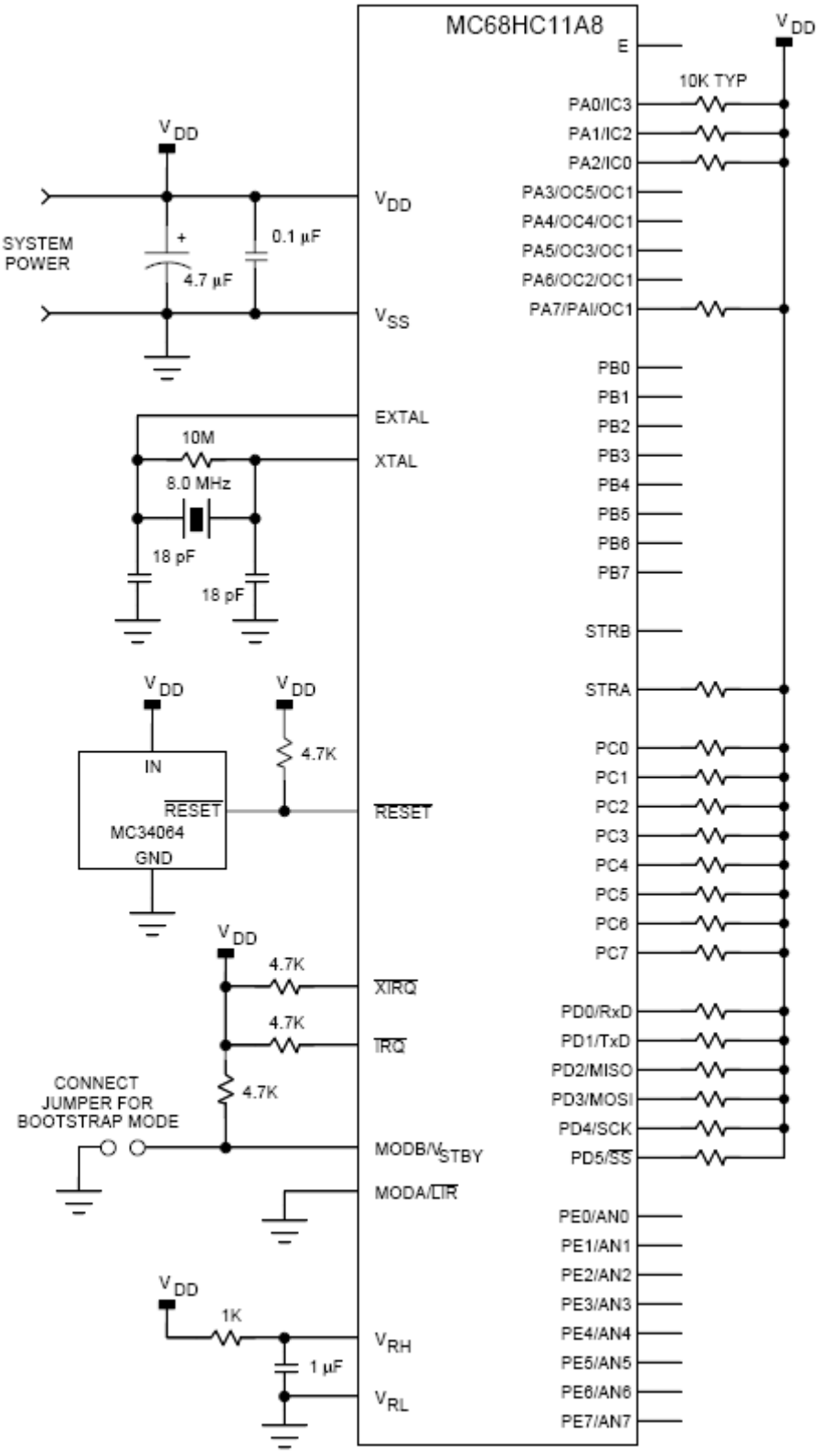


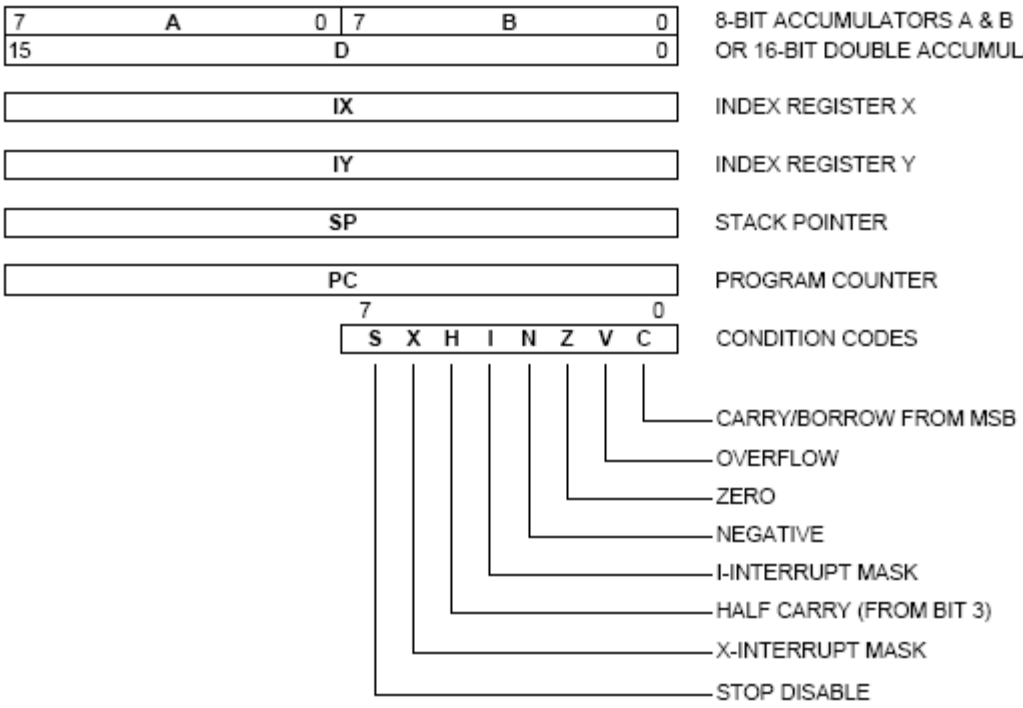
Figure 2-1 MC68HC11A8 Pin Assignments

Table 2-1 Hardware Mode Select Summary

Inputs		Mode Description	Control Bits in HPRIO (Latched at Reset)			
MODB	MODA		RBOOT	SMOD	MDA	IRV
1	0	Normal Single Chip	0	0	0	0
1	1	Normal Expanded	0	0	1	0
0	0	Special Bootstrap	1	1	0	1
0	1	Special Test	0	1	1	1



Basic Single-Chip-Mode Connections



M68HC11 Programmer's Model

Function	Mnemonic	IMM	DIR	EXT	INDX	INDY	INH
Clear Memory Byte	CLR			X	X	X	
Clear Accumulator A	CLRA						X
Clear Accumulator B	CLRB						X
Load Accumulator A	LDAA	X	X	X	X	X	
Load Accumulator B	LDAB	X	X	X	X	X	
Load Double Accumulator D	LDD	X	X	X	X	X	
Pull A from Stack	PULA						X
Pull B from Stack	PULB						X
Push A onto Stack	PSHA						X
Push B onto Stack	PSHB						X
Store Accumulator A	STAA	X	X	X	X	X	
Store Accumulator B	STAB	X	X	X	X	X	
Store Double Accumulator D	STD	X	X	X	X	X	
Transfer A to B	TAB						X
Transfer A to CCR	TAP						X
Transfer B to A	TBA						X
Transfer CCR to A	TPA						X
Exchange D with X	XGDX						X
Exchange D with Y	EGDY						X

Loads, Stores, And Transfers

Function	Mnemonic	IMM	DIR	EXT	INDX	INDY	INH
Add Accumulators	ABA						X
Add Accumulator B to X	ABX						X
Add Accumulator B to Y	ABY						X
Add with Carry to A	ADCA	X	X	X	X	X	
Add with Carry to B	ADCB	X	X	X	X	X	
Add Memory to A	ADDA	X	X	X	X	X	
Add Memory to B	ADDB	X	X	X	X	X	
Add Memory to D (16 Bit)	ADDD	X	X	X	X	X	
Compare A to B	CBA						X
Compare A to Memory	CMPA	X	X	X	X	X	
Compare B to Memory	CMPB	X	X	X	X	X	
Compare D to Memory (16 Bit)	CPD	X	X	X	X	X	
Decimal Adjust A (for BCD)	DAA						X
Decrement Memory Byte	DEC			X	X	X	
Decrement Accumulator A	DECA						X
Decrement Accumulator B	DECB						X
Increment Memory Byte	INC			X	X	X	
Increment Accumulator A	INCA						X
Increment Accumulator B	INCB						X
Two's Complement Memory Byte	NEG			X	X	X	
Two's Complement Accumulator A	NEGA						
Two's Complement Accumulator B	NEGB						
Subtract with Carry from A	SBCA	X	X	X	X	X	
Subtract with Carry from B	SBCB	X	X	X	X	X	
Subtract Memory from A	SUBA	X	X	X	X	X	
Subtract Memory from B	SUBB	X	X	X	X	X	
Subtract Memory from D (16 Bit)	SUBD	X	X	X	X	X	
Test for Zero or Minus	TST			X	X	X	
Test for Zero or Minus A	TSTA						X
Test for Zero or Minus B	TSTB						X

Arithmetic Operations

Function	Mnemonic	INH
Multiply (A X B ⇒ D)	MUL	X
Fractional Divide (D ÷ X ⇒ X; r ⇒ D)	FDIV	X
Integer Divide (D ÷ X ⇒ X; r ⇒ D)	IDIV	X

Multiply and Divide

Function	Mnemon-ic	IMM	DIR	EXT	INDX	INDY	INH
AND A with Memory	ANDA	X	X	X	X	X	
AND B with Memory	ANDB	X	X	X	X	X	
Bit(s) Test A with Memory	BITA	X	X	X	X	X	
Bit(s) Test B with Memory	BITB	X	X	X	X	X	
One's Complement Memory Byte	COM			X	X	X	
One's Complement A	COMA						X
One's Complement B	COMB						X
OR A with Memory (Exclusive)	EORA	X	X	X	X	X	
OR B with Memory (Exclusive)	EORB	X	X	X	X	X	
OR A with Memory (Inclusive)	ORAA	X	X	X	X	X	
OR B with Memory (Inclusive)	ORAB	X	X	X	X	X	

Logical Operations

Function	Mnemonic	IMM	DIR	EXT	INDX	INDY
Bit(s) Test A with Memory	BITA	X	X	X	X	X
Bit(s) Test B with Memory	BITB	X	X	X	X	X
Clear Bit(s) in Memory	BCLR		X		X	X
Set Bit(s) in Memory	BSET		X		X	X
Branch if Bit(s) Clear	BRCLR		X		X	X
Branch if Bit(s) Set	BRSET		X		X	X

Data Testing and Bit Manipulation

Function	Mnemonic	IMM	DM	EXT	INDX	INDY	INH
Arithmetic Shift Left Memory	ASL			X	X	X	
Arithmetic Shift Left A	ASLA						X
Arithmetic Shift Left B	ASLB						X
Arithmetic Shift Left Double	ASLD						X
Arithmetic Shift Right Memory	ASR			X	X	X	
Arithmetic Shift Right A	ASRA						X
Arithmetic Shift Right B	ASRB						X
(Logical Shift Left Memory)	(LSL)			X	X	X	
(Logical Shift Left A)	(LSLA)						X
(Logical Shift Left B)	(LSLB)						X
(Logical Shift Left Double)	(LSLD)						X
Logical Shift Right Memory	LSR			X	X	X	
Logical Shift Right A	LSRA						X
Logical Shift Right B	LSRB						X
Logical Shift Right D	LSRD						X
Rotate Left Memory	ROL			X	X	X	
Rotate Left A	ROLA						X
Rotate Left B	ROLB						X
Rotate Right Memory	ROR			X	X	X	
Rotate Right A	RORA						X
Rotate Right B	RORB						X

Shifts and Rotates

Function	Mnemonic	IMM	DIR	EXT	INDX	INDY	INH
Add Accumulator B to X	ABX						X
Add Accumulator B to Y	ABY						X
Compare X to Memory (16 Bit)	CPX	X	X	X	X	X	
Compare Y to Memory (16 Bit)	CPY	X	X	X	X	X	
Decrement Stack Pointer	DES						X
Decrement Index Register X	DEX						X
Decrement Index Register Y	DEY						X
Increment Stack Pointer	INS						X
Increment Index Register X	INX						X
Increment Index Register Y	INY						X
Load Index Register X	LDX	X	X	X	X	X	
Load Index Register Y	LDY	X	X	X	X	X	
Load Stack Pointer	LDS	X	X	X	X	X	
Pull X from Stack	PULX						X
Pull Y from Stack	PULY						X
Push X onto Stack	PSHX						X
Push Y onto Stack	PSHY						X
Store Index Register X	STX	X	X	X	X	X	
Store Index Register Y	STY	X	X	X	X	X	
Store Stack Pointer	STS	X	X	X	X	X	
Transfer SP to X	TSX						X
Transfer SP to Y	TSY						X
Transfer X to SP	TXS						X
Transfer Y to SP	TYS						X
Exchange D with X	XGDX						X
Exchange D with Y	XGDY						X

Stack and Index Register Instructions

Function	Mnemonic	INH
Clear Carry Bit	CLC	X
Clear Interrupt Mask Bit	CLI	X
Clear Overflow Bit	CLV	X
Set Carry Bit	SEC	X
Set Interrupt Mask Bit	SEI	X
Set Overflow Bit	SEV	X
Transfer A to CCR	TAP	X
Transfer CCR to A	TPA	X

Condition Code Register Instructions

Function	Mnemonic	REL	DIR	INDX	INDY	Comments
Branch if Carry Clear	BCC	X				C = 0 ?
Branch if Carry Set	BCS	X				C = 1 ?
Branch if Equal Zero	BEQ	X				Z = 1 ?
Branch if Greater Than or Equal	BGE	X				Signed ≥
Branch if Greater Than	BGT	X				Signed >
Branch if Higher	BHI	X				Unsigned >
Branch if Higher or Same (same as BCC)	BHS	X				Unsigned ≥
Branch if Less Than or Equal	BLE	X				Signed ≤
Branch if Lower (same as BCS)	BLO	X				Unsigned <
Branch if Lower or Same	BLS	X				Unsigned ≤
Branch if Less Than	BLT	X				Signed <
Branch if Minus	BMI	X				N= 1 ?
Branch if Not Equal	BNE	X				Z = 0 ?
Branch if Plus	BPL	X				N = 0 ?
Branch if Bit(s) Clear in Memory Byte	BRCLR		X	X	X	Bit Manipulation
Branch Never	BRN	X				3-cycle NOP
Branch if Bit(s) Set in Memory Byte	BRSET		X	X	X	Bit Manipulation
Branch if Overflow Clear	BVC	X				V = 0 ?
Branch if Overflow Set	BVS	X				V = 1 ?

Branches

Function	Mnemonic	DIR	EXT	INDX	INDY	INH
Jump	JMP	X	X	X	X	

Jumps

Function	Mnemonic	REL	DIR	EXT	INDX	INDY	INH
Branch to Subroutine	BSR	X					
Jump to Subroutine	JSR		X	X	X	X	
Return from Subroutine	RTS						X

Subroutine Calls And Returns (BSR, JSR, RTS)

Appendix J
MAX 233

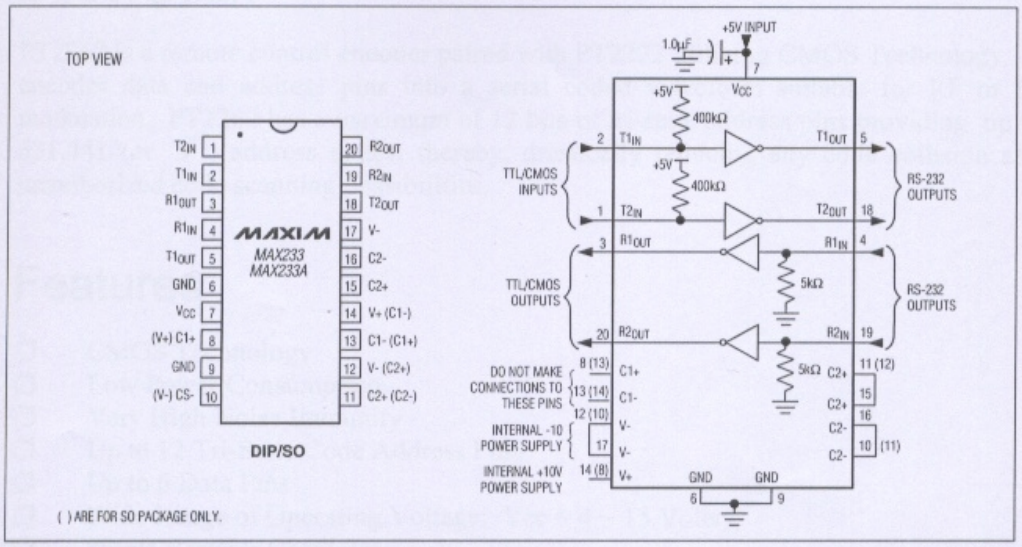


Figure 11. MAX233/MAX233A Pin Configuration and Typical Operating Circuit

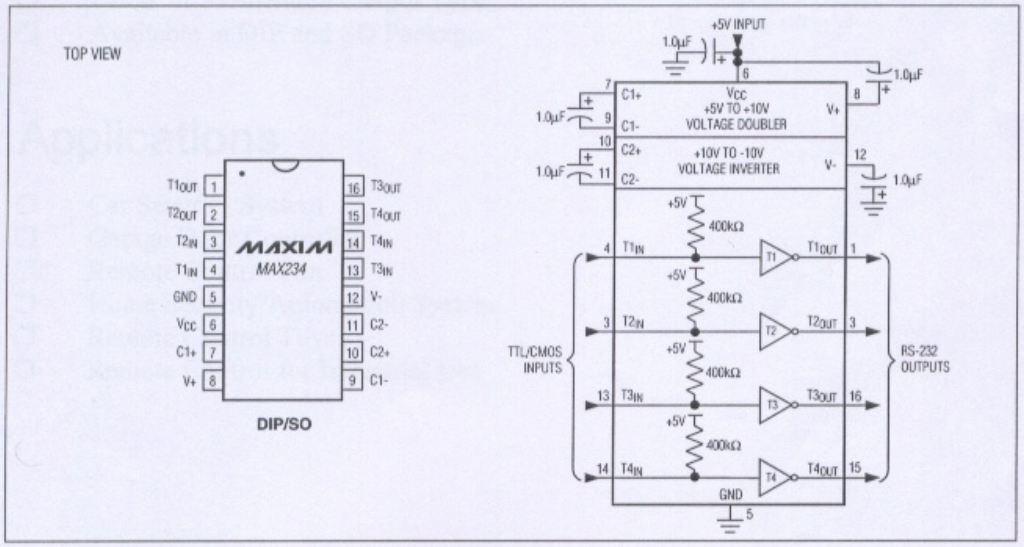


Figure 12. MAX234 Pin Configuration and Typical Operating Circuit

Appendix K
PT 2262

Princeton Technology Corp.
Tel : 886-2-29162151
Fax : 886-2-29174598
URL: *http://www.princeton.com.tw*

Remote Control Encoder PT2262

Description

PT2262 is a remote control encoder paired with PT2272 utilizing CMOS Technology. It encodes data and address pins into a serial coded waveform suitable for RF or IR modulation. PT2262 has a maximum of 12 bits of tri-state address pins providing up to 531,441 (or 3^{12}) address codes; thereby, drastically reducing any code collision and unauthorized code scanning possibilities.

Features

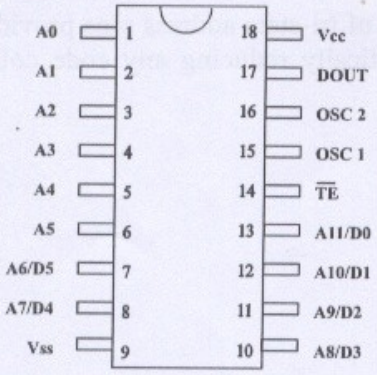
- CMOS Technology
- Low Power Consumption
- Very High Noise Immunity
- Up to 12 Tri-State Code Address Pins
- Up to 6 Data Pins
- Wide Range of Operating Voltage: Vcc = 4 ~ 15 Volts
- Single Resistor Oscillator
- Latch or Momentary Output Type
- Available in DIP and SO Package

Applications

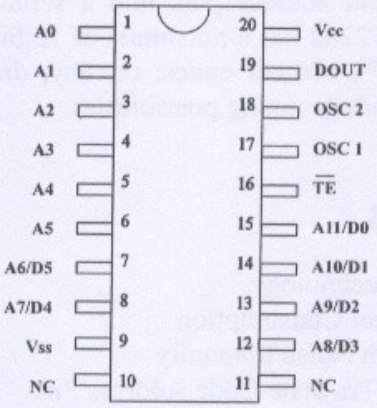
- Car Security System
- Garage Door Controller
- Remote Control Fan
- Home Security/Automation System
- Remote Control Toys
- Remote Control for Industrial Use

Princeton Technology Corp.
Tel : 886-2-29162151
Fax : 886-2-29174598
URL: *http://www.princeton.com.tw*

Pin Configuration



PT 2262
PT2262-S18
PT2262-IR



PT2262 S
PT2262 - IR - S

Princeton Technology Corp.
Tel : 886-2-29162151
Fax : 886-2-29174598
URL: *http://www.princeton.com.tw*

Ordering Information

Valid Product No.	Package
PT2262	18 Pins, DIP
PT2262-S18	18 Pins, SO
PT2262-IR	18 Pins, DIP
PT2262-S	20 Pins, SO
PT2262-IR-S	20 Pins, SO

PT 2262 - XX - X

Package Code :

P : Plastic DIP (this code can be omitted)
S : Small Outline, 20 Pins
S18 : Small Outline, 18 Pins

Output Type :

RF : RF Application (can be omitted)
IR : IR Application

Product Number

PTC Prefix

Appendix L
PT 2272

Princeton Technology Corp.
Tel: 886-2-29162151
Fax: 886-2-29174598
URL: *http://www.princeton.com.tw*

Remote Control Decoder PT 2272

Description

PT 2272 is a remote control decoder paired with PT 2262 utilizing CMOS Technology. It has 12 bits of tri-state address pins providing a maximum of 531,441 (or 3^{12}) address codes; thereby, drastically reducing any code collision and unauthorized code scanning possibilities. PT 2272 is available in several options to suit every application need : variable number of data output pins, latch or momentary output type.

Features

- ❒ CMOS Technology
- ❒ Low Power Consumption
- ❒ Very High Noise Immunity
- ❒ Up to 12 Tri-State Code Address Pins
- ❒ Up to 6 Data Pins
- ❒ Wide Range of Operating Voltage: Vcc = 4 ~ 15 Volts
- ❒ Single Resistor Oscillator
- ❒ Latch or Momentary Output Type
- ❒ Available in DIP and SO Package

Applications

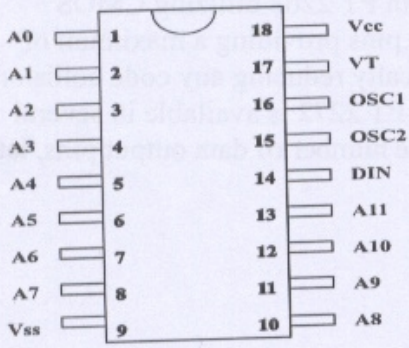
- ❒ Car Security System
- ❒ Garage Door Controller
- ❒ Remote Control Fan
- ❒ Home Security/Automation System
- ❒ Remote Control Toys
- ❒ Remote Control for Industrial Use

Princeton Technology Corp.
Tel: 886-2-29162151
Fax: 886-2-29174598
URL: *http://www.princeton.com.tw*

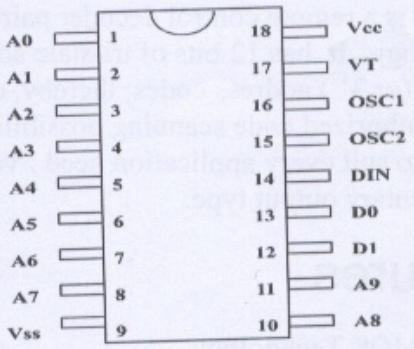
Remote Control Decoder

PT 2272

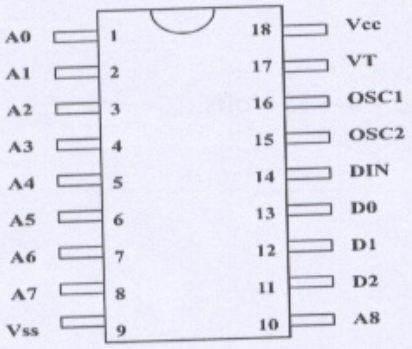
Pin Configuration



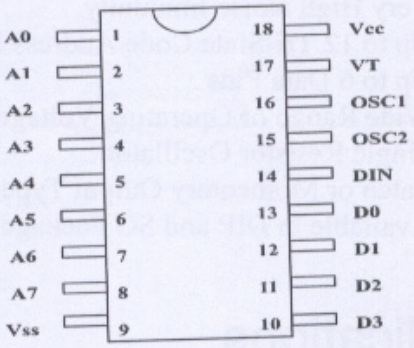
PT 2272



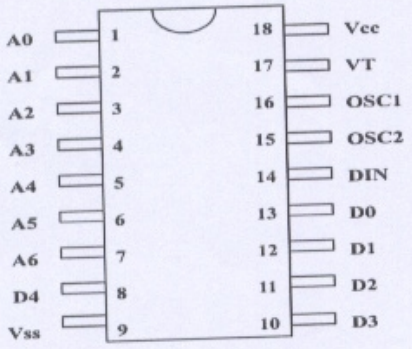
PT 2272A - M2/L2



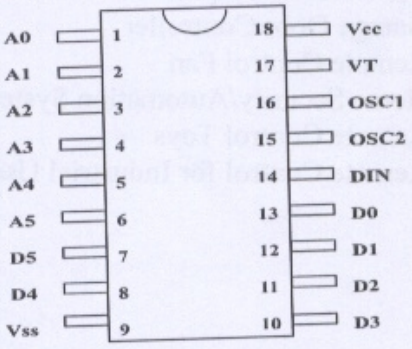
PT 2272 - M3/L3



PT 2272 - M4/L4



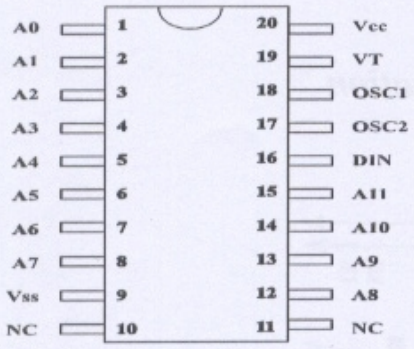
PT 2272 - M5/L5



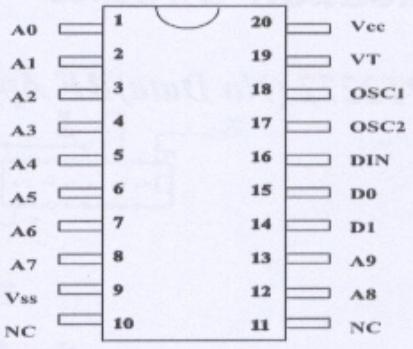
PT 2272 - M6/L6

Princeton Technology Corp.
Tel: 886-2-29162151
Fax: 886-2-29174598
URL: *http://www.princeton.com.tw*

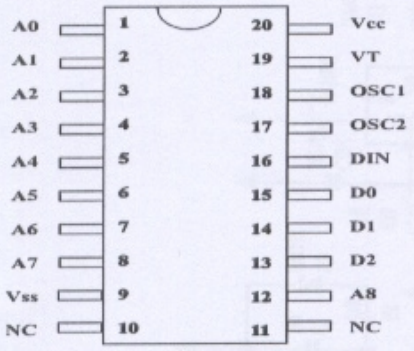
Remote Control Decoder PT 2272



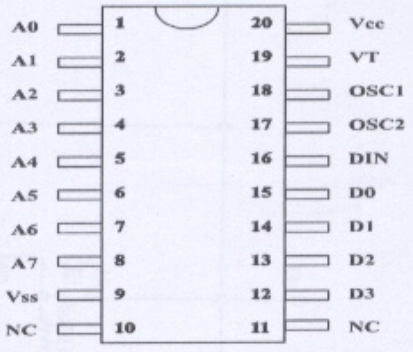
PT 2272 (20 Pins)



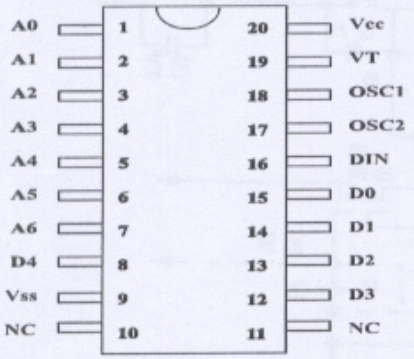
PT 2272A - M2/L2 (20 Pins)



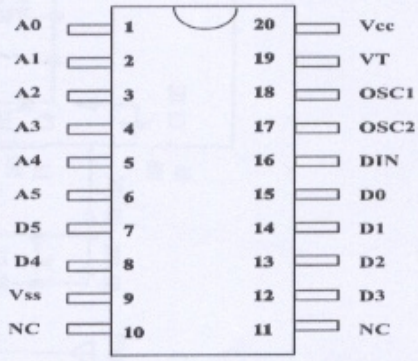
PT 2272 - M3/L3 (20 Pins)



PT 2272 - M4/L4 (20 Pins)



PT 2272 - M5/L5 (20 Pins)



PT 2272 - M6/L6 (20 Pins)

Princeton Technology Corp.
Tel: 886-2-29162151
Fax: 886-2-29174598
URL: *http://www.princeton.com.tw*

Ordering Information

A. 18- Pin Package:

Part No.	Data Bit(s)	Output Type	Package
PT2272 (-S18)	None	*	DIP (SO)
PT2272A-L2 (-S18)	2	Latch	DIP (SO)
PT2272A-M2 (-S18)	2	Momentary	DIP (SO)
PT2272-L3 (-S18)	3	Latch	DIP (SO)
PT2272-M3 (-S18)	3	Momentary	DIP (SO)
PT2272-L4 (-S18)	4	Latch	DIP (SO)
PT2272-M4 (-S18)	4	Momentary	DIP (SO)
PT2272-L5 (-S18)	5	Latch	DIP (SO)
PT2272-M5 (-S18)	5	Momentary	DIP (SO)
PT2272-L6 (-S18)	6	Latch	DIP (SO)
PT2272-M6 (-S18)	6	Momentary	DIP (SO)

B. 20-Pin Package

Part No.	Data Bit(s)	Output Type	Package
PT2272-S	None	*	SO
PT2272A-L2S	2	Latch	SO
PT2272A-M2S	2	Momentary	SO
PT2272-L3S	3	Latch	SO
PT2272-M3S	3	Momentary	SO
PT2272-L4S	4	Latch	SO
PT2272-M4S	4	Momentary	SO
PT2272-L5S	5	Latch	SO
PT2272-M5S	5	Momentary	SO
PT2272-L6S	6	Latch	SO
PT2272-M6S	6	Momentary	SO

Note: * = use VT (Valid Transmission)

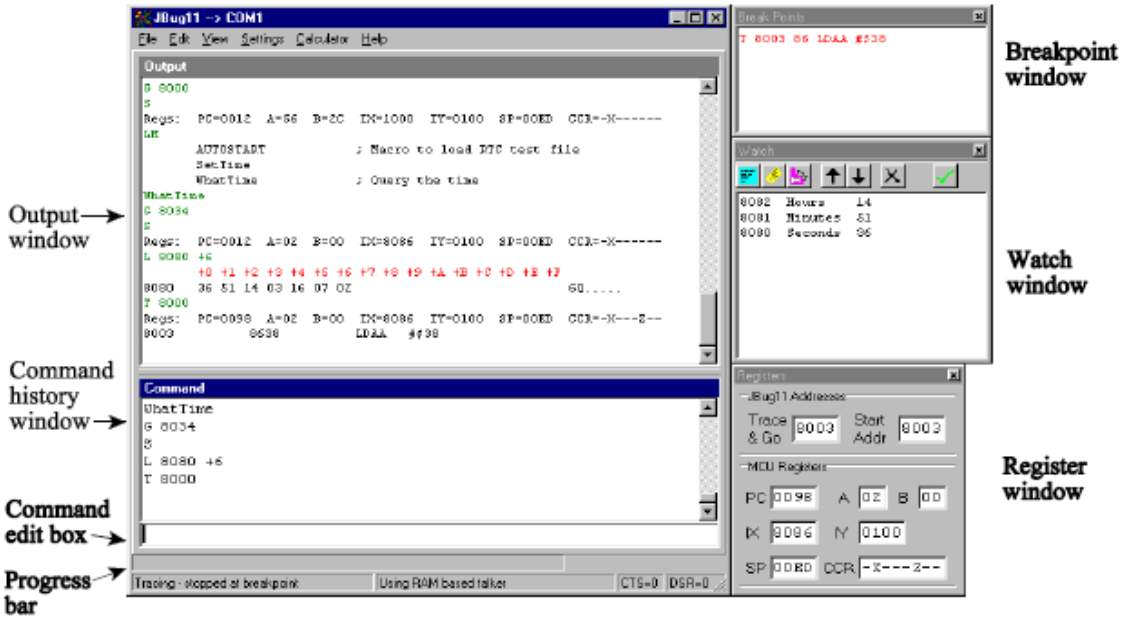
Princeton Technology Corp.
Tel: 886-2-29162151
Fax: 886-2-29174598
URL: *http://www.princeton.com.tw*

Remote Control Decoder PT 2272

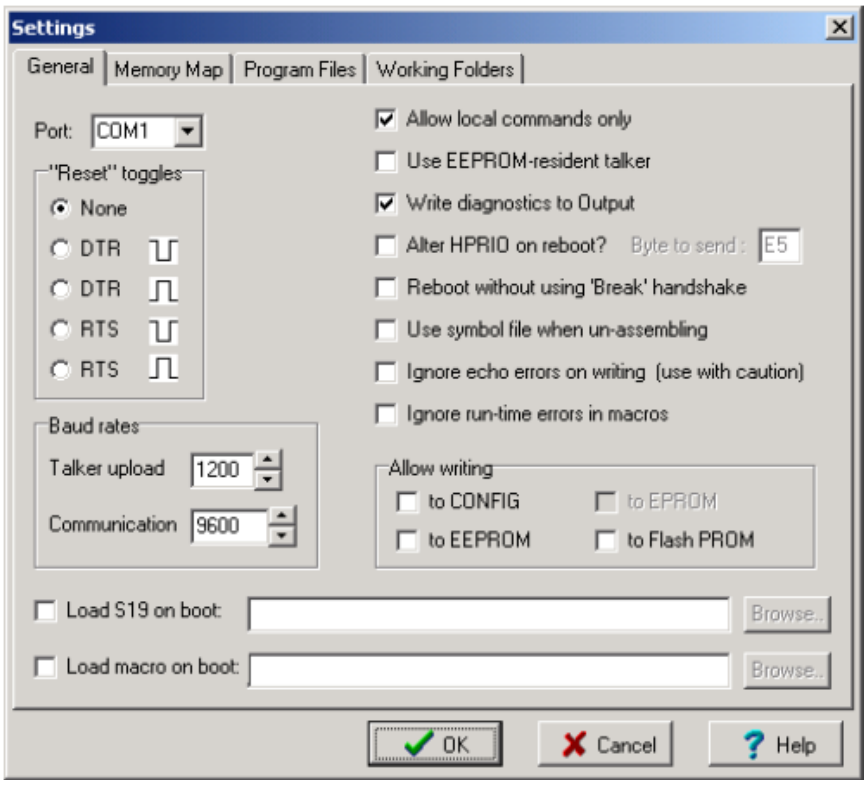
PT 2272 X - X X X

- Package: P = PDIP, 18 Pins (can be omitted)
  S18 = Small Outline, 18 Pins
  S = Small Outline, 20 Pins
- Data Bits: 0 = 0 Bit (can be omitted)
  2 = 2 Data Bits
  3 = 3 Data Bits
  4 = 4 Data Bits
  5 = 5 Data Bits
  6 = Data Bits
- Data Output Type: L = Latch Type
  M = Momentary Type
- Version: A =New Version (For 2 Data Bits Only)
- PTC Number
- PTC Prefix

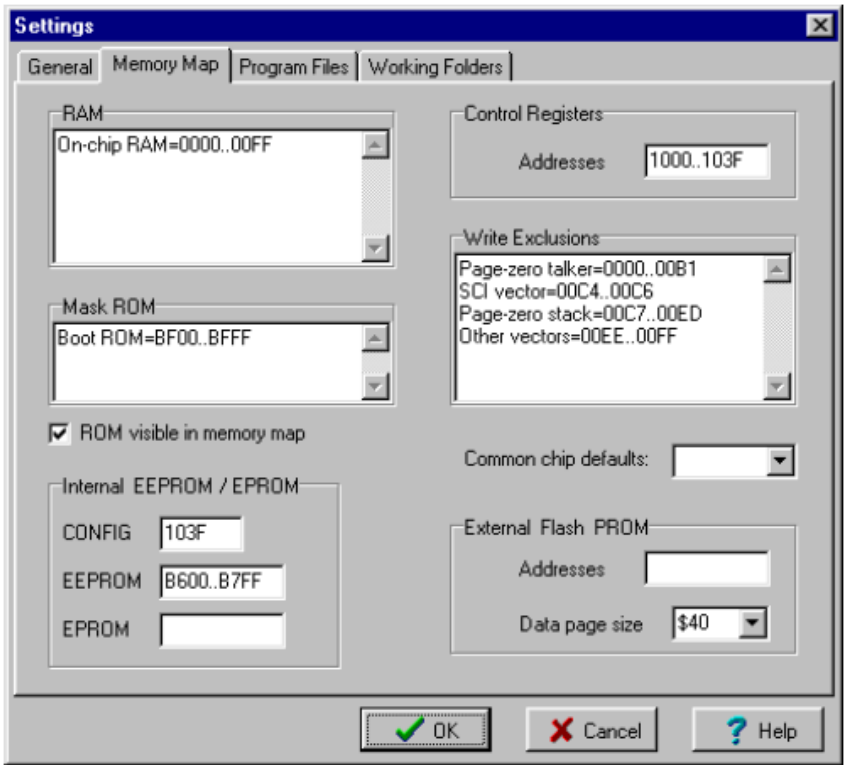
Appendix M
JBug 11 Manual



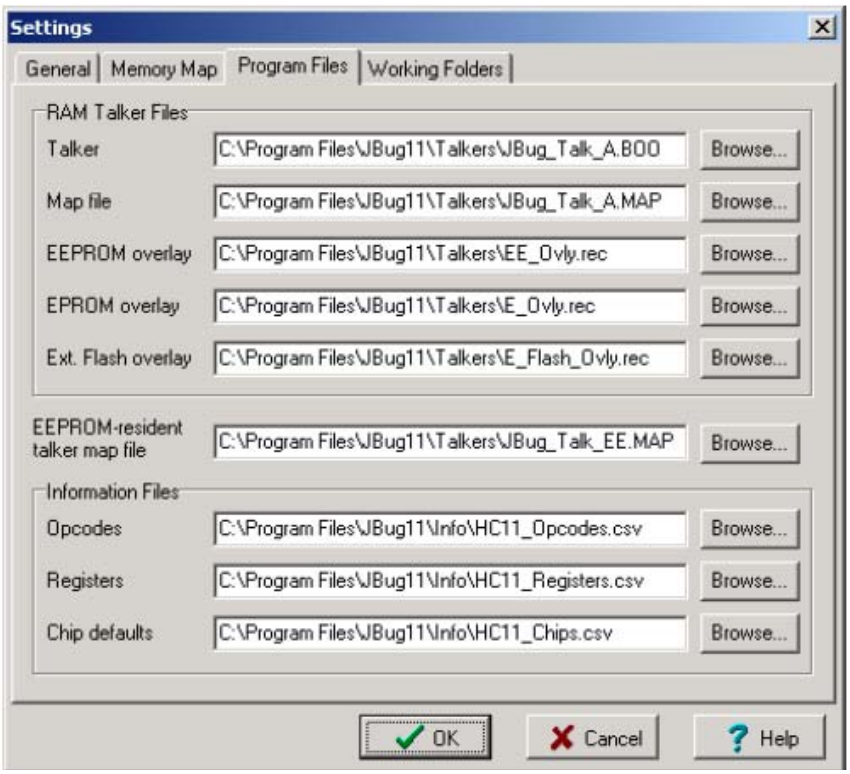
General Layout



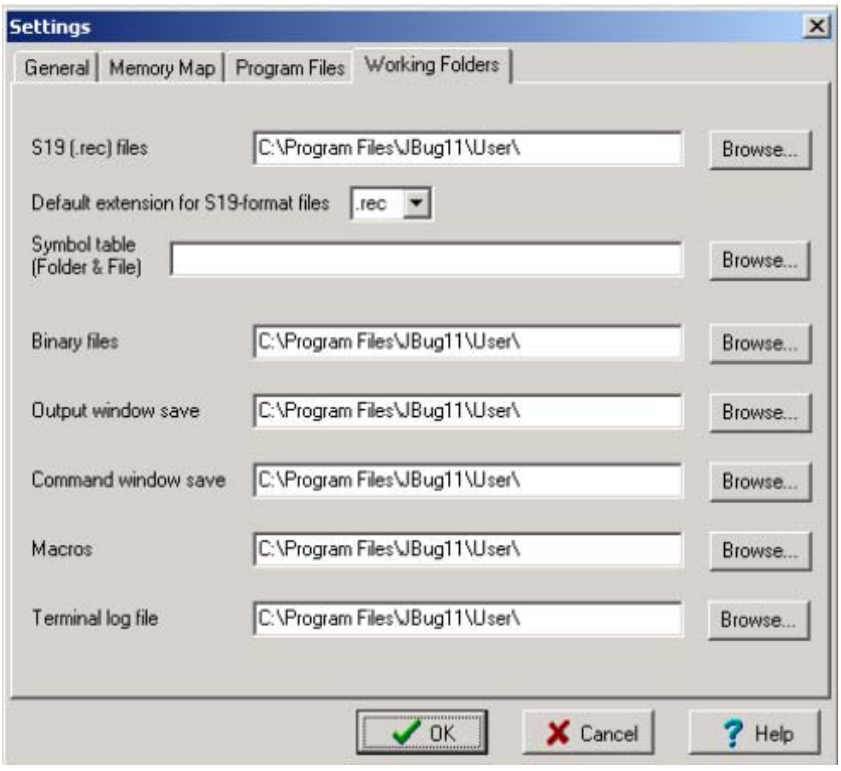
General Tab



Memory Map Tab



Program Files Tab



Working Folder Tabs

Command Summary

A StartAddr	Assemble HC11 instruction and write to MCU-controlled memory
AL StartAddr	Assemble HC11 instruction to local memory
BR Breakpoint1[X] [Breakpoint2[X] [Breakpoint3...]]	Set one or more transient or fixed breakpoints
BRP[PassCount] Breakpoint *or* PP[PassCount] Breakpoint	Set Pass type breakpoint
CLS	Clear output window
CLM	Clear the local, JBug11, copy of memory
CLRM	Clear loaded macro library
CMP StartAddr EndAddr|+Length CompAddr	Compare two blocks of MCU memory
CMPL StartAddr EndAddr|+Length CompAddr	Compare two blocks of local memory
CRC [Path\]Filename[.rec|.s19] *or* CRC StartAddr EndAddr|+Length	Compute a CRC-16 sum for a file or a block of MCU memory
D StartAddr EndAddr|+Length ToAddr	Duplicate a block of MCU memory
DL StartAddr EndAddr|+Length ToAddr	Duplicate a block of local memory
EBULK	Erase all of EEPROM
F StartAddr EndAddr|+Length ByteString|CharString	Fill MCU memory with bytes or a character string
FL StartAddr EndAddr|+Length ByteString|CharacterString	Fill local memory with bytes or a character string
FIND StartAddr EndAddr|+Length ByteString|CharacterString	Find a byte or character string in MCU controlled memory
FINDL StartAddr EndAddr|+Length ByteString|CharacterString	Find a byte or character string in local memory
G [StartAddr|* [Breakpoint1 [Breakpoint2 [Breakpoint3...]]]]	Go - run a program, stopping at breakpoints if requested.
LD [Path\]Filename[.rec|.s19] LD [Path\]Filename.obj|bin StartAddr	Load MCU memory with a Motorola S19 format file, or with a binary image file.

`LDL [Path\]Filename[.rec|.s19]` `LDL [Path\]Filename.obj|bin StartAddr`	Load JBug11 local memory with a Motorola S19 format file, or with a binary image file.
`L [StartAddr|* [EndAddr|+Length]]`	List MCU controlled memory
`LL [StartAddr|* [EndAddr|+Length]]`	List JBug11 local memory
`LM [MacroName]`	List macro names and commands
`M StartAddr` *or* `MM StartAddr`	Modify MCU controlled memory
`NEXT`	Search for further occurrences of string nominated in the FIND command
`NOBR [Breakpoint1 [Breakpoint2 [Breakpoint3...]]]`	Clear all or selected breakpoints
`O[Repeats] [StartAddr|*]`	Trace over subroutines
`PAUSE [milliseconds]`	Pause macro execution (same as WAIT)
`Q`	Quit JBug11
`R [InhReg=NewValue| CtrlReg|CtrlReg=NewValue]`	Display and modify registers
`S`	Stop a program running on the MCU
`SV StartAddr EndAddr|+Length [Path\]Filename[.Ext]`	Save a block of MCU memory to an S19 or binary image file
`SVL StartAddr EndAddr|+Length [Path\]Filename[.Ext]`	Save a block of local memory to an S19 or binary image file
`T[Repeats] [StartAddr|*]`	Trace a program in MCU controlled memory
`TERM`	Open the terminal window

`U[Repeats] [StartAddr|*]`
or
`U StartAddr|* EndAddr|+Length`
	Un-assemble (disassemble) a program in MCU controlled memory

`UL[Repeats] [StartAddr|*]`
or
`UL StartAddr|* EndAddr|+Length`
	Un-assemble a program in local memory

`V [Path\]Filename[.rec|.s19]`	Verify a program in MCU memory against a Motorola S19 format file
`VE StartAddr [EndAddr|+Length]`	Verify that memory is erased (all bytes $FF)
`WAIT [milliseconds]`	Pause macro execution (same as PAUSE)

Keyboard Shortcuts - Main Window Active

Shortcut	Description
Ctrl+B	Re-boot the target MCU. Useful if you have fitted the remote resetting hardware.
Ctrl+K	Open the calculator
Ctrl+M	Open the macro library file editor
Ctrl+Q	Quit the programme. **Careful!** no confirmation is requested
Ctrl+T	Open the terminal window
Esc	Clear the command line

Keyboard Shortcuts - Terminal Window Active

Shortcut	Description
Ctrl+I	Open a file to send as an image
Ctrl+Q	Close the Terminal Window
Esc	Stop writing an image file

Keyboard Shortcuts - Macro Editing Window Active

Shortcut	Description
Ctrl+C	Copy to Clipboard
Ctrl+V	Paste from Clipboard
Ctrl+X	Cut to Clipboard
Ctrl+O	Open a file to edit
Ctrl+S	Save a macro library file
Ctrl+L	Insert in the editor the macro library currently loaded in JBug11
Ctrl+Q	Close the macro editor