CAN LINKED MONETARY VALIDATION SYSTEM

A PROJECT REPORT

Submitted by

ANKUSH D JAIN	**Reg.No:1110107009**
NAVEEN KUMAR M	**Reg.No:1110107059**
DINESH KUMAR V	**Reg.No:1110107303**
HARISH C S	**Reg.No:1110107307**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641049

(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2015

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641049

(An Autonomous Institution Affiliated to Anna University, Chennai)

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE-641049

(An Autonomous Institution Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report titled **"CAN LINKED VEHICLE MONETARY VALIDATION SYSTEM"** is the bonafied work of "**ANKUSH D JAIN, NAVEEN KUMAR. M, DINESH KUMAR. V, HARISH. CS"** who carried out the project work under my supervision.

SIGNATURE

Prof. K. Ram Prakash, M.E.,
Professor/E.C.E
Kumaraguru College of Technology
Coimbatore.

SIGNATURE

Dr. Rajeswari Mariappan M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Electronics & Communication Engineering
Kumaraguru College of Technology
Coimbatore.

The candidates with Register numbers 1110107009, 1110107059, 1110107303 and 1110107307 are examined by us in the project viva-voce examination held on …….

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First we would like to express our praise and gratitude to the Lord, who has showered his grace and blessing enabling us to complete this project in an excellent manner. He has made all things in beautiful in his time.

We express our sincere thanks to our beloved Joint Correspondent, **Shri. Shankar Vanavarayar** for his kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr.R.S.Kumar M.E., Ph.D.,** who encouraged us with his valuable thoughts.

We would like to express our sincere thanks and deep sense of gratitude to our HOD, **Dr. Rajeswari Mariappan M.E., Ph.D.,** for her valuable suggestions and encouragement which paved way for the successful completion of the project.

We are greatly privileged to express our deep sense of gratitude to the Project Coordinator **Ms.S.Nagarathinam M.E., (Ph.D)**, Assistant Professor (SRG), for her continuous support throughout the course.

In particular, We wish to thank and express our everlasting gratitude to the Supervisor **Mr.K.Ram Prakash M.E.,** Professor for her expert counselling in each and every steps of project work and we wish to convey our deep sense of gratitude to all teaching and non-teaching staff members of ECE Department for their help and cooperation.

Finally, we thank our parents and our family members for giving us the moral support in all of our activities and our dear friends who helped us to endure our difficult times with their unfailing support and warm wishes.

TABLE OF CONTENTS

ABSTRACT

Nowadays evasion from due payment of automobile loan, vehicle insurance is an ordinary affair. This paper deals with the design & development of an advanced vehicle locking system in the real time environment based on the loan / Insurance details. The proposed system consists of an PIC16F877A and PIC18F458 controller and GSM module that is embedded in the vehicle with an interfacing to Engine Control Module(ECM) through Control Area Network (CAN) Bus which is in turn, communicated to the ECM. The ECM will allow to start the vehicle only after receiving the fair signals from the PIC controller. If the user attempts to remove the PIC controller unit, the fair signal won't get received by the ECM and thereby the engine will not ignite. In this proposed work, Real Time Clock (RTC) and EEPROM data will automatically lock the vehicle on the every 30^{th} day of the month. The engine can be unlocked only by the banker / insurer by sending the due clearance confirmation message. It makes the way for the prompt payment of the dues by the customer. Inculcating hardware implementation, ensuring vehicle security, user friendliness are some of the valuable features enlisted in this project. PROTEUS software is used for simulation and MP LAB for compilation.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Indian automobile market has claimed global attention, being Indian automobile market has claimed global attention, being the second largest two wheeler market, fourth largest commercial vehicle market and eleventh largest passenger car market in the world, and poised to become the third largest automobile market next only to the United States and China. Research says 75% of vehicles purchased in the last decade were financed through loans. The automobile finance offered by banks and financial institutions at affordable rates of interest has paved the way for the growth of the automobile sector in India. Various schemes and features are available to consumers which can accommodate their every need, thus luring them into a financing option. Normally all the vehicle owners want to pay their regular loan/ insurance for their care. But many of the consumers doesn't pay their loan/insurance properly, they are preparing duplicate copy of insurance and in case of vehicle loan, few of them are cheating the bank or any private office by submitting duplicate nativity certificate.

Nowadays the Bankers & Insurers give the intimation to the customer at every month to pay their loan due amount through SMS to their registered mobile number. If the customer doesn't pay their due amount more than one month, immediately they will take the action to cease their vehicle. But it is very difficult to find their address because few of them are not submitting their proper nativity certificate and few of them will dismantle the vehicle after few due date gets over. So evasion from due payment of automobile loan, vehicle insurance is an ordinary affair. Currently there is no valid system to track the

defaulters, which affects the government exchequer hardly and also affects the prosperity of the individuals. Hence there is an urgent need to fill this vacuum by modernizing the existing technology in automobile industries. As a matter of initiation, an innovative PIC based control system exclusively for car has been designed and implemented in this project.

Today modern car already contain a multiplicity of controllers that are increasingly networked together by various bus communication systems with very different properties. Automotive communication networks have access to several crucial components of the vehicle, like breaks, airbags, and the engine control. Automation is the use of control systems and information technologies to reduce the need for human work in the production of goods and services. This project deals with the design and development of a CAN (Controller Area Network) linked automatic vehicle locking system for an automobile, which is being used to follow the proper loan due / Insurance of a vehicle. It makes the way for the prompt payment of the dues by the customer.

1.2 Objectives

- ❖ To design and develop an advanced vehicle locking system in real time environment based on vehicle monetary validation system.
- ❖ To make prompt payment of the dues by the customer.
- ❖ To track the defaulters with the help of advance automotive technology.
- ❖ To create a worthy work environment for Bankers & Insurers through a novel due collection system.

1.3 Outline of the Project

This project mainly consists of nine main chapters; Introduction, Literature review, , PIC Controller, EEPROM, RTC (Real Time Clock), GSM

(Global System for Mobiles) and Simulation using PROTEUS / MPLAB software, Simulation results analysis and Conclusion.

Chapter 1: This chapter explained the crucial aspect of the project work such as design, objectives and scope of the project as well the outline of the report will also be discussed finally.

Chapter 2: The Literature Review on Vehicle Locking And Tracking System.

Chapter 3: Overview of existing system and proposed system.

Chapter 4: Gives the overview about the PIC controllers.

Chapter 5: Gives discussion about EEPROM and different operations.

Chapter 6: This chapter discussed about the Real Time Clock and its operations.

Chapter 7: Overview of GSM and their working operations.

Chapter 8: This chapter gives the brief description about our Project work with block diagram.

Chapter 9: Simulation Results and Discussions.

Chapter 10: Hardware Description.

Chapter 11: Future Scope of the project.

Chapter 12: Conclusion.

Chapter 13: References.

Chapter 14. Appendix.

CHAPTER 2

LITERATURE REVIEW

Through this chapter, hopefully it will give some idea to the reader regarding the different technologies of vehicle tracking and monitoring system and its implementation for vehicle security as well. Besides that, basic concepts regarding vehicle tracking and monitoring will be reviewed as well as the embedded controllers.

In addition, related works regarding to this research project work also will be conferred in depth. It’s an honor to be the medium of knowledge manifestation from the universe. The concept of the reference papers and how it helped in designing this system is described in the following subsections.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

Nowadays evasion from due payment of automobile loan and vehicle insurance is an ordinary affair. Currently the vehicle tracking system using microcontroller, GSM and GPS are available plenty in the market. But the already developed system didn't address the problem of vehicle loan / Insurance defaulters.

Normally all the vehicle owners want to pay their regular loan/ insurance for their care. But many of the consumers doesn't pay their loan/insurance properly, they are preparing duplicate copy of insurance and in case of vehicle loan, few of them are cheating the bank or any private office by submitting duplicate nativity certificate.

Nowadays the Bankers & Insurers give the intimation to the customer at every month to pay their loan due amount through SMS to their registered mobile number. If the customer doesn't pay their due amount more than one month, immediately they will take the action to cease their vehicle. But it is very difficult to find their address because few of them are not submitting their proper nativity certificate and few of them will dismantle the vehicle after few due date gets over. So prevarication from due payment of automobile loan, vehicle insurance is an ordinary affair. Currently there is no valid system to track the defaulters, which affects the government exchequer hardly and also affects the prosperity of the individuals**.**

3.2 Proposed System

The proposed system attempts to make life more interesting by reducing unnecessary waste of man-power by employing microcontrollers. The system consists of an PIC (PIC16F877A & PIC18F458) controller and GSM module that is embedded in the vehicle with an interfacing to Engine Control Unit(ECU) through Control Area Network (CAN) Bus, which is in turn, communicated to the ECU. The ECU will allow to start vehicle only after receiving the fair signals from the PIC18F458 controller.

If there is any default in loan or insurance due payment, the vehicle will get locked by the vehicle distributor through GSM. The proposed system accepts the message and broadcast to the Vehicle Network through CAN Bus. The engine can be unlocked only by the vehicle distributor by sending the due clearance confirmation message. If the user attempts to remove the PIC controller unit, the fair signal won't get received by the ECU and thereby the engine will not ignite. Jammer usage can be overcome by Real Time Clock (RTC) which is inbuilt in the system that will send an interrupt signal to the control unit at the end of due date. It makes the way for the prompt payment of the dues. Additionally theft avoidance can also be addressed using the above said technique.

CHAPTER 4

PERIPHERAL INTERFACE CONTROLLER

4.1 Introduction

The Peripheral Interface Controller (PIC) contains an ALU, which does arithmetic and logic operations, the RAM , the EEPROM (Flash Memory), the data EEPROM, and the “W” register. The “W” register is not a part of the register-file but is a stand-alone, working register (also called an “accumulator”). The ALU, the RAM, the “W” register, and the data EEPROM each manipulate and hold 8-bit-wide data, which ranges in value from zero to 255 (or, in hexadecimal, from 0x00 to 0xFF).

The program EEPROM (Flash Memory) works with 14-bit-wide words and contains each of the user’s instructions. It is not uncommon for microcontrollers to have different sizes of data memory and program memory (in the PIC: 8-bits for data and 14-bits for program words). More than that, the key is that the data and program memories occupy separate spaces. This allows access to each at the same time.

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five three I/O ports.
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen interrupts.
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight A/D input channels.
- The Parallel Slave Port is implemented only on the 40/44-pin devices.

4.2 Pin Diagram

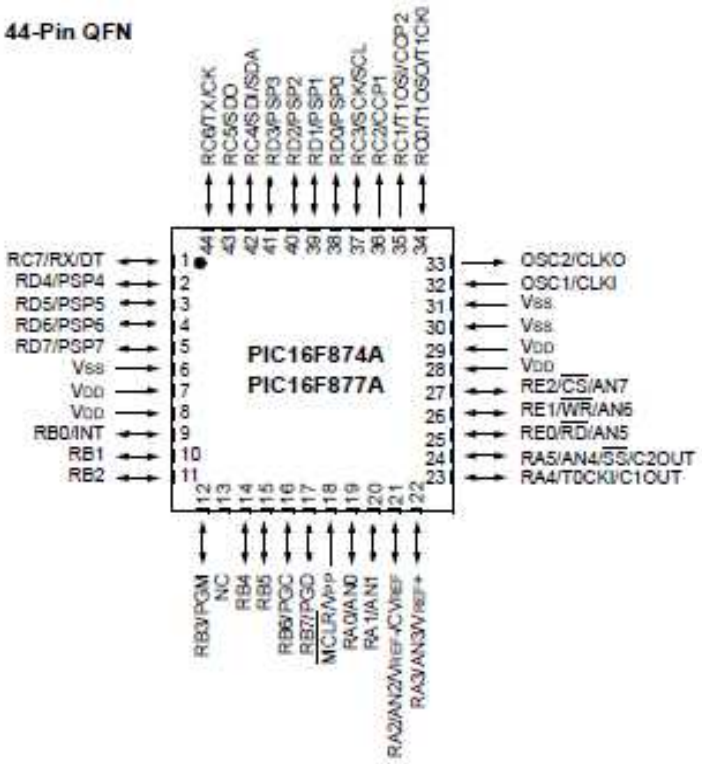
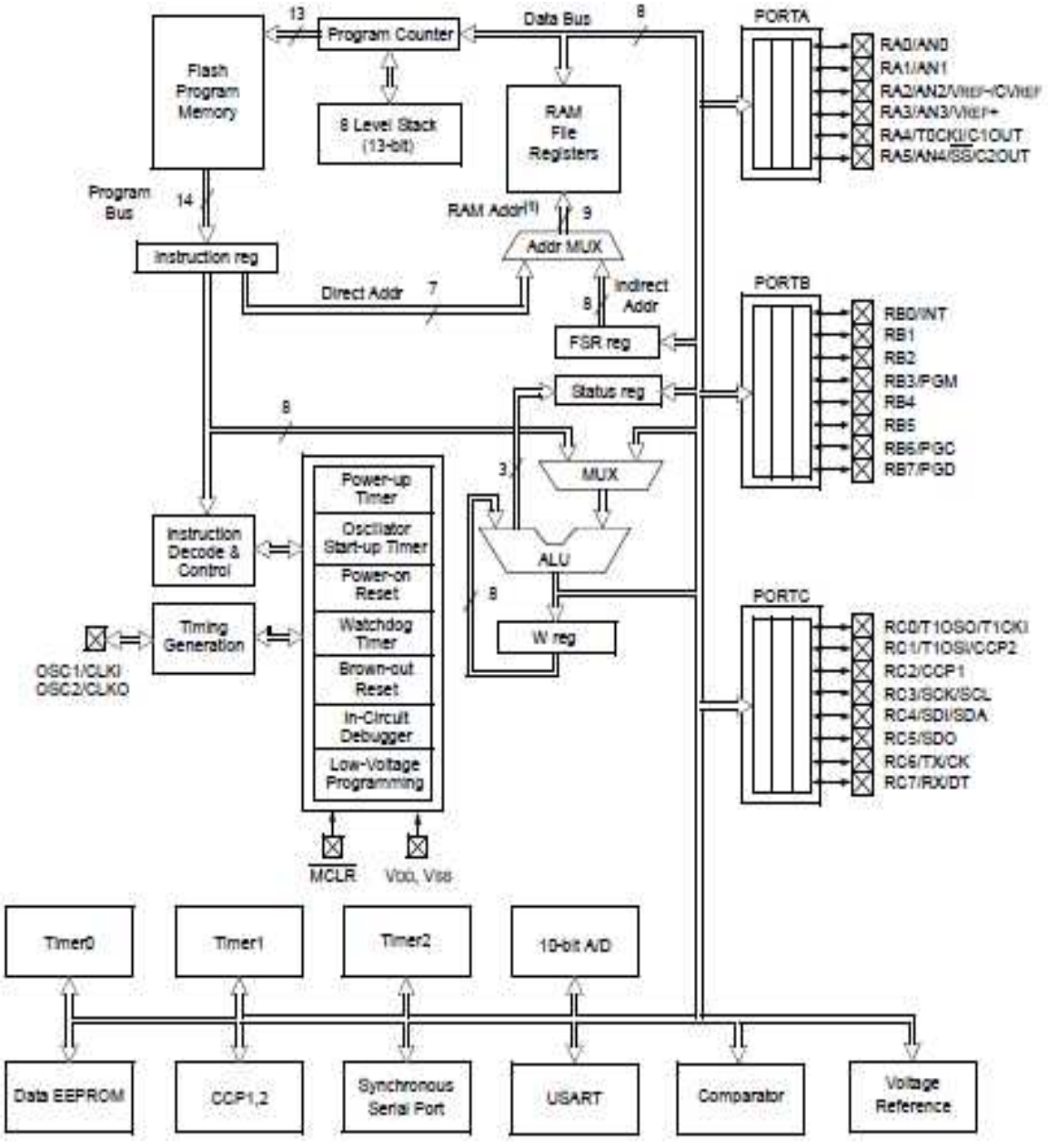


Fig 4.2 Pin Diagram for PIC 16F877A

Architecture PIC16F877A

The PIC(Peripheral Interfacing Controller) microcontrollers has been introduced by the microchip technology. while PIC16F874A/877A devices are available in 40-pin and 44-pin packages.



Devloe	Program Flash	Data Memory	Data EEPROM
PIC16F873A	4K words	192 Bytes	128 Bytes
PIC16F876A	8K words	368 Bytes	256 Bytes

Fig 4.3 Architecture of PIC 16F877A

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific write sequence to initiate the write for each byte. The write will not initiate if the write sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte.

We strongly recommend that interrupts be disabled during this code segment . Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set. At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

The steps to write to EEPROM data memory are:

1. If step 10 is not implemented, check the WR bit to see if a write is in progress.
2. Write the address to EEADR. Make sure that the address is not larger than the memory size of the device.
3. Write the 8-bit data value to be programmed in the EEDATA register.
4. Clear the EEPGD bit to point to EEPROM data memory.
5. Set the WREN bit to enable program operations.
6. Disable interrupts (if enabled).
7. Execute the special five instruction sequence:
8. Write 55h to EECON2 in two steps (first to W, then to EECON2)

9. Write AAh to EECON2 in two steps (first to W, then to EECON2)Set the WR bit
10. Enable interrupts (if using interrupts).
11. Clear the WREN bit to disable program operations.
12. At the completion of the write cycle, the WR bit is cleared and the EEIF interrupt flag bit is set. (EEIF must be cleared by firmware.) If step 1 is not implemented, then firmware should check for EEIF to be set, or WR to clear, to indicate the end of the program cycle.

```
    BSF   STATUS,RP1      ;
    BSF   STATUS,RP0
    BTFSC EECON1,WR       ;Wait for write
    GOTO  $-1             ;to complete
    BCF   STATUS, RP0     ;Bank 2
    MOVF  DATA_EE_ADDR,W  ;Data Memory
    MOVWF EEADR           ;Address to write
    MOVF  DATA_EE_DATA,W  ;Data Memory Value
    MOVWF EEDATA          ;to write
    BSF   STATUS,RP0      ;Bank 3
    BCF   EECON1,EEPGD    ;Point to DATA
                          ;memory
    BSF   EECON1,WREN     ;Enable writes

    BCF   INTCON,GIE      ;Disable INTs.
    MOVLW 55h             ;
    MOVWF EECON2          ;Write 55h
    MOVLW AAh             ;
    MOVWF EECON2          ;Write AAh
    BSF   EECON1,WR       ;Set WR bit to
                          ;begin write
    BSF   INTCON,GIE      ;Enable INTs.
    BCF   EECON1,WREN     ;Disable writes
```

Required Sequence

Fig 4.4 Coding to Write Data in EEPROM

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI	13	14	30	32		ST/CMOS(4)	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	15	31	33		—	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.
$\overline{MCLR}$/VPP	1	2	18	18		ST	Master Clear (input) or programming voltage (output).
$\overline{MCLR}$					I		Master Clear (Reset) input. This pin is an active low Reset to the device.
VPP					P		Programming voltage input.
							PORTA is a bidirectional I/O port.
RA0/AN0	2	3	19	19		TTL	
RA0					I/O		Digital I/O.
AN0					I		Analog input 0.
RA1/AN1	3	4	20	20		TTL	
RA1					I/O		Digital I/O.
AN1					I		Analog input 1.
RA2/AN2/VREF-/CVREF	4	5	21	21		TTL	
RA2					I/O		Digital I/O.
AN2					I		Analog input 2.
VREF-					I		A/D reference voltage (Low) input.
CVREF					O		Comparator VREF output.
RA3/AN3/VREF+	5	6	22	22		TTL	
RA3					I/O		Digital I/O.
AN3					I		Analog input 3.
VREF+					I		A/D reference voltage (High) input.
RA4/T0CKI/C1OUT	6	7	23	23		ST	
RA4					I/O		Digital I/O – Open-drain when configured as output.
T0CKI					I		Timer0 external clock input.
C1OUT					O		Comparator 1 output.
RA5/AN4/$\overline{SS}$/C2OUT	7	8	24	24		TTL	
RA5					I/O		Digital I/O.
AN4					I		Analog input 4.
$\overline{SS}$					I		SPI slave select input.
C2OUT					O		Comparator 2 output.

Legend: I = Input O = output I/O = Input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Table 4.3 PIC16F877A Pinout Descriptions

4.4 Memory Organization

A microcontroller is a complete computer system on a single chip. It is more than just a microprocessor: It also contains a Read-Only Memory (ROM), a Read-Write Memory (RAM), some input/output ports, and some peripherals, such as, counters/timers, analog-to-digital converters, digital-to-analog converters, and serial communication ports.

The internal view of a typical microprocessor and is composed of three things: an arithmetic/logic unit (ALU) which performs calculations on data. A set of registers which hold the user's data and the system's data; and a control unit which orchestrates everything and interprets and executes the user's instructions. As far as the microprocessor is concerned, it assumes that there are sets of data memories and program memories (RAM and ROM) in the system.

The only thing the microprocessor has to do is run a cycle of getting new instructions and executing them from the memories. Both the RAM and the ROM are organized as indexed sets of data words, where each "index" is the "address" of its corresponding data. Both the data and its address codes are numbers represented in binary or hexadecimal.

The RAM is a read-write memory which can rapidly read and write the data. It is a volatile memory which means that it loses its memory when power is removed (turned off). The ROM is for program memory and is "read-only" except in modern variants, such as Electrically Erasable Programmable Read Only Memory (EEPROM) and Flash Memory, which allow data words to be written as well as read.

The writing of an EEPROM is not the same as a RAM since the data-writing time of the EEPROM is about ten thousand times as long as the data-writing time of the RAM. The ROM and its variants are non-volatile memories that preserve their memories when the power is removed (turned off).

4.5 Special Features

- Timer0: 8-bit timer/counter with 8-bit pre scaler
- Timer1: 16-bit timer/counter with pre scaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, pre scaler and post scaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Key Features	PIC16F877A
Operating Frequency	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	8K
Data Memory (bytes)	368
EEPROM Data Memory (bytes)	256
Interrupts	15
I/O Ports	Ports A, B, C, D, E
Timers	3
Capture/Compare/PWM modules	2
Serial Communications	MSSP, USART
Parallel Communications	PSP
10-bit Analog-to-Digital Module	8 input channels
Analog Comparators	2
Instruction Set	35 Instructions
Packages	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

Table 4.4 PIC 16F877A Features

4.5.1 High-Performance RISC CPU

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16F877A.

4.6 SPECIAL REGISTERS

4.6.1 Status Register

The Status register contains the arithmetic status of the ALU, the Reset status and the bank select bits for data memory. The Status register can be the destination for any instruction, as with any other register. If the Status register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the TO and PD bits are not writable, therefore, the result of an instruction with the Status register as destination may be different than intended. For example, CLRF STATUS, will clear the upper three bits and set the Z bit. This leaves the Status register as 000u u1uu (where u = unchanged). It is recommended, therefore, that only BCF, BSF, SWAPF and MOVWF instructions are used to alter the Status register because these instructions do not affect the Z, C or DC bits from the Status register.

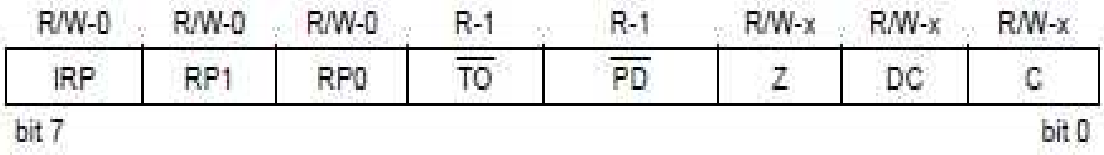


Fig 4.5 Status Register (ADDRESS 03h, 83h, 103h, 183h)

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)

1 = Bank 2, 3 (100h-1FFh)

0 = Bank 0, 1 (00h-FFh)

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)

11 = Bank 3 (180h-1FFh)

10 = Bank 2 (100h-17Fh)

01 = Bank 1 (80h-FFh)

00 = Bank 0 (00h-7Fh)

Each bank is 128 bytes.

bit 4 **TO**: Time-out bit

1 = After power-up, CLRWDT instruction or SLEEP instruction

0 = A WDT time-out occurred

bit 3 **PD**: Power-down bit

1 = After power-up or by the CLRWDT instruction

0 = By execution of the SLEEP instruction

bit 2 **Z**: Zero bit

1 = The result of an arithmetic or logic operation is zero

0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/borrow bit

1 = A carry-out from the 4th low order bit of the result occurred

0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/borrow bit (ADDWF, ADDLW,SUBLW,SUBWF instructions)

1 = A carry-out from the Most Significant bit of the result occurred

0 = No carry-out from the Most Significant bit of the result occurred

4.7 I^2C BUS (Inter Integrated Circuit)

The I^2C module provides an interface between the TCI648x/C6472 device and other devices compliant with the I^2C-bus specification and connected by way of an I^2C-bus. External components attached to this 2-wire serial bus can transmit and receive up to 8-bit wide data to and from the device through the I^2C module.

The I^2C allows connection of up to 128 individually addressable devices using only two bi-directional lines: clock (SCL) and data (SDA). The only additional hardware required is a pull-up resistor for each of the lines. Each of the connected devices can be either a master or slave device. Only master devices are allowed to drive the clock line. The I^2C protocol and the EFM32 I2C module feature several mechanisms for handling bus conflicts and contention

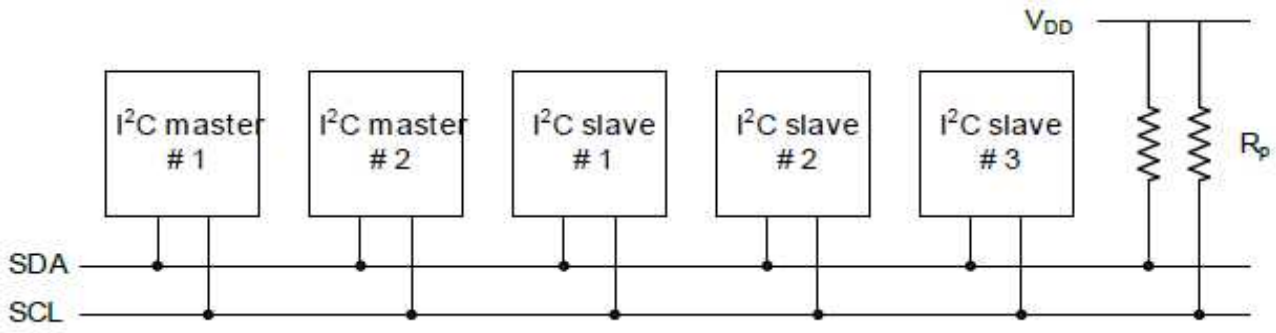


Fig 4.6 I^2C Connection Scheme

At the physical layer both SCL and SCA lines are in open-drain, hence the pull-up resistors. Increasing the number of devices on the I2C bus will also increase the line capacitance and thus reduce the slewrate. The slew-rate can be controlled by changing the drive strength in the GPIO module for the I2C pins. The size of the pull-up resistors can be calculated as a function of the maximum rise time allowed for the given bus speed and the estimated bus capacitance.

4.7.1 Functional Block diagram

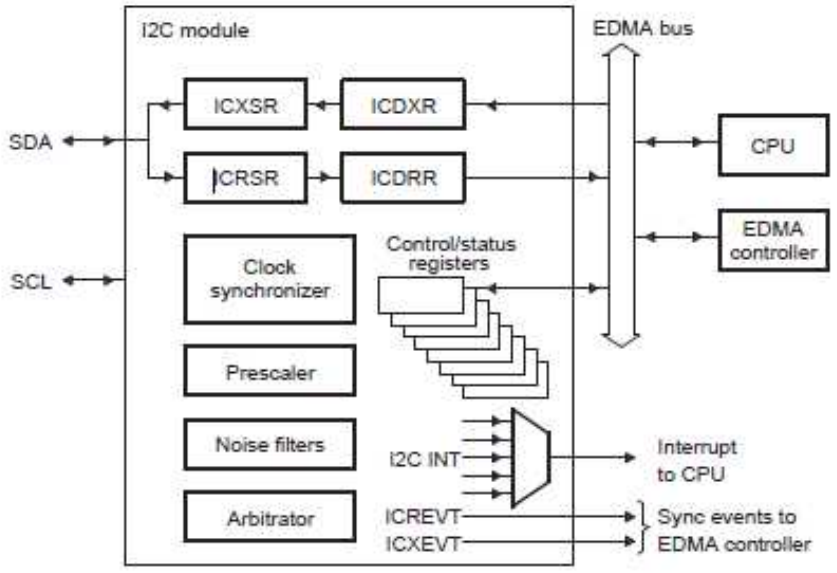


Fig 4.7.1 Block diagram of I^2C

4.7.2 Data Transfer – SCL (Serial Clock)

- Master sets SCL = 0 and generates pulse for each data bit
- 8 pulses for data bits are followed by one pulse for ack. bit
- After ack.
  - Master tries to generate next byte's first pulse
  - Slave can hold SCL low _ master switches to wait state.

4.7.3 Data Transfer -SDA (Serial Data)

- Data bits are generated by transmitter as SCL pulses
- 9^{th} pulse:
  - Transmitter releases SDA
  - Receiver must hold SDA low in order to ack. received data
  - Slave must release SDA after ack. bit (allows master to end frame)

4.8 PIC 18F458 -Architecture

The PIC18 microcontroller family provides PIC micro® devices in 18- to 80-pin packages, that are both socket and software upwardly compatible to the PIC16 family. The PIC18 family includes all the popular peripherals, such as MSSP, ESCI, CCP, flexible 8- and 16-bit timers, PSP, 10-bit ADC, WDT, and POR and CAN 2.0B Active for the maximum flexible solution.

Most PIC18 devices will provide FLASH program memory in sizes from 8 to 128 Kbytes and data RAM from 256 to 4 Kbytes; operating from 2.0 to 5.5 volts, at speeds from DC to 40 MHz Optimized for high-level languages like ANSI C, the PIC18 family offers a highly flexible solution for complex embedded applications.

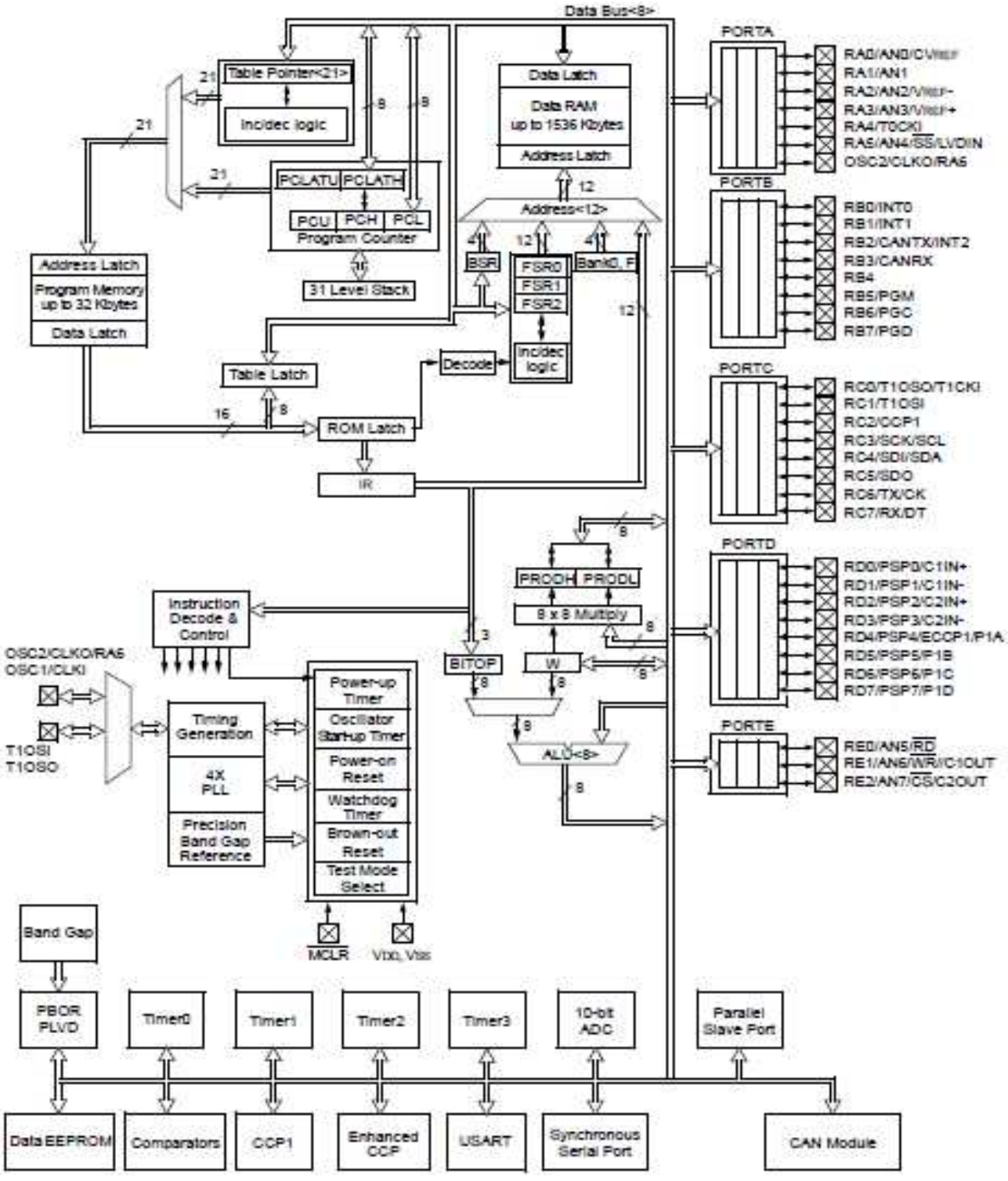


Fig 4.8 PIC 18F458 Architecture

Features		PIC18F458
Operating Frequency		DC – 40 MHz
Internal Program Memory	Bytes	32K
	# of Single-Word Instructions	16384
Data Memory (Bytes)		1536
Data EEPROM Memory (Bytes)		256
Interrupt Sources		21
I/O Ports		Ports A, B, C, D, E
Timers		4
Capture/Compare/PWM Modules		1
Enhanced Capture/Compare/ PWM Modules		1
Serial Communications		MSSP, CAN, Addressable USART
Parallel Communications (PSP)		Yes
10-bit Analog-to-Digital Converter		8 input channels
Analog Comparators		2
Analog Comparators VREF Output		Yes
Resets (and Delays)		POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low-Voltage Detect		Yes
Programmable Brown-out Reset		Yes
CAN Module		Yes
In-Circuit Serial Programming™ (ICSP™)		Yes
Instruction Set		75 Instructions
Packages		40-pin PDIP 44-pin PLCC 44-pin TQFP

Table 4.8.1 PIC 18F458 Features

4.9 Pin Diagram

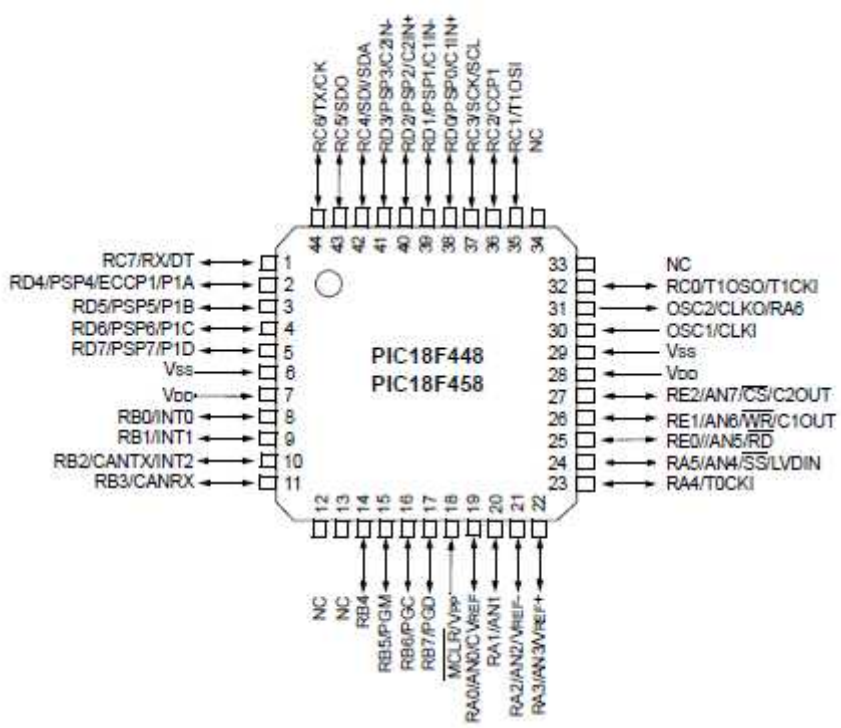


Fig 4.2 Pin Diagram for PIC 18F458

4.10 Memory Organization

There are three memory blocks in Enhanced MCU devices. These memory blocks are:

- Enhanced Flash Program Memory
- Data Memory
- EEPROM Data Memory

Data and program memory use separate busses, which allows concurrent access of these blocks. Additional detailed information on data EEPROM and Flash program memory.

4.10.1 Program Memory Organization

The PIC18F458 devices have a 21-bit program counter that is capable of addressing a 2-Mbyte program memory space. The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

4.10.2 Internal Program Memory Operation

The PIC18F458 have 32 Kbytes of internal Enhanced Flash program memory. This means that the PIC18F258 and the PIC18F458 can store up to 16K of single-word instructions. The PIC18F458 have 16 Kbytes of Enhanced Flash program memory. This translates into 8192 single-word instructions, which can be stored in the program memory. Accessing a location between the physically implemented memory and the 2-Mbyte address will cause a read of all ‘0’s (a NOP instruction).

4.11 Special Features

4.11.1 Advanced Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter module (A/D) with:
  - Conversion available during Sleep
  - Up to 8 channels available
- Analog Comparator module:
  - Programmable input and output multiplexing
- Comparator Voltage Reference module
- Programmable Low-Voltage Detection (LVD) module:
  - Supports interrupt-on-Low-Voltage Detection
- Programmable Brown-out Reset (BOR)

4.11.2 CAN bus Module Features:

- Complies with ISO CAN Conformance Test
- Message bit rates up to 1 Mbps
- Conforms to CAN 2.0B Active Spec with:

- 29-bit Identifier Fields
- 8-byte message length
- 3 Transmit Message Buffers with prioritization
- 2 Receive Message Buffers
- 6 full, 29-bit Acceptance Filters
- Prioritization of Acceptance Filters
- Advanced Error Management Features

4.11.3 Special Microcontroller Features:

• Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)

• Watchdog Timer (WDT) with its own on-chip RC oscillator

• Programmable code protection

• Power-saving Sleep mode

• Selectable oscillator options, including:

- 4x Phase Lock Loop (PLL) of primary oscillator
- Secondary Oscillator (32 kHz) clock input

• In-Circuit Serial Programming TM (ICSPTM) via two pins

4.11.4 Flash Technology:

• Low-power, high-speed Enhanced Flash technology

• Fully static design

• Wide operating voltage range (2.0V to 5.5V)

• Industrial and Extended temperature ranges

4.12 CONTROL AREA NETWORK (CAN)

4.12.1 Introduction

The CAN bus was developed by BOSCH as a multi-master, message broadcast system that specifies a maximum signaling rate of 1M bit per second (bps). Unlike a traditional network such as USB or Ethernet, CAN does not send large blocks of data point-to-point from node A to node B under the supervision of a central bus master.

In a CAN network many short messages like temperature or RPM are broadcast to the entire network, which allows for data consistency in every node of the system. This application report explains the CAN message format, message identifiers, and bit-wise arbitration—a major benefit of the CAN signaling scheme. CAN bus implementation is examined and typical waveforms are presented.

The MCP2551 is a high-speed CAN, fault-tolerant device that serves as the interface between a CAN protocol controller and the physical bus. The MCP2551 provides differential transmit and receive capability for the CAN protocol controller and is fully compatible with the ISO-11898 standard, including 24V requirements. It will operate at speeds of up to 1 Mb/s. Typically, each node in a CAN system must have a device to convert the digital signals generated by a CAN controller to signals suitable for transmission over the bus cabling (differential output). It also provides a buffer between the CAN controller and the high-voltage spikes that can be generated on the CAN bus by outside sources (EMI, ESD, electrical transients.

4.12.2 Arbitration

A fundamental CAN characteristic shown in Figure 4 is the opposite logic state between the bus, and the driver input and receiver output. Normally a

logic high is associated with a one, and a logic low is associated with a zero—but not so on a CAN bus. This is why it is desirable to have the driver input and receiver output pins of a CAN transceiver passively pulled high internally, as in the SN65HVD230. In the absence of any input, the device automatically defaults to a recessive bus state on all input and output pins.

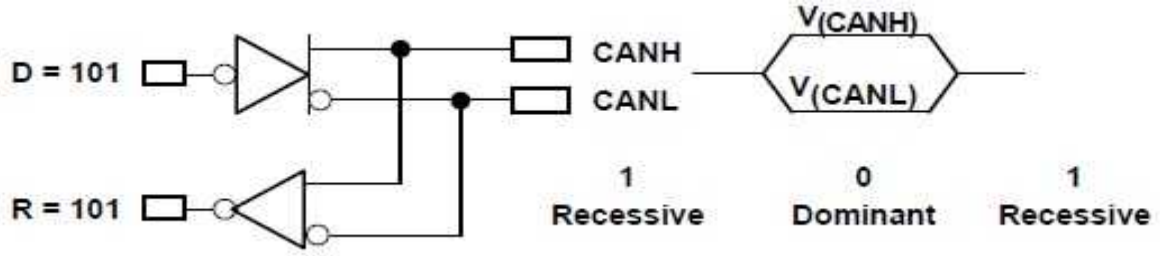


Fig.4.13.2 The Inverted Logic of a CAN Bus

Bus access is event-driven and takes place randomly. If two nodes try to occupy the bus simultaneously, access is implemented with a nondestructive, bit-wise arbitration. Nondestructive means that the node winning arbitration just continues on with the message, without the message being destroyed or corrupted by another node. The allocation of priority to messages in the identifier is a feature of CAN that makes it particularly attractive for use within a real-time control environment. The lower the binary message identifier number, the higher its priority. An identifier consisting entirely of zeros is the highest priority message on a network since it holds the bus dominant the longest. Therefore, if two nodes begin to transmit simultaneously, the node that sends a zero (dominant) while the other nodes send a one (recessive) gets control of the CAN bus and goes on to complete its message. A dominant bit always overwrites a recessive bit on a CAN bus. Note that a node constantly monitors its own transmission. This is the reason for the transceiver

configuration of Figure 4 in which the CANH and CANL output pins of the driver are internally tied to the receiver's input.

1.12.3 Message Types

There are four different message types, or frames that can be transmitted on a CAN bus: the data frame, the remote frame, the error frame, and the overload frame. A message is considered to be error free when the last bit of the ending EOF field of a message is received in the error–free recessive state. A dominant bit in the EOF field causes the transmitter to repeat a transmission.

4.12.4 CAN Bus

The data link and physical signaling layers of Figure 1, which are normally transparent to a system operator, are included in any controller that implements the CAN protocol, such as Texas Instruments' TMS320LF2407 3.3-V DSP with integrated CAN controller. Connection to the physical medium is then implemented through a line transceiver such as TI's SN65HVD230 3.3-V CAN transceiver to form what the ISO–11898 standard refers to as an electronic control unit (ECU) in Figure 4.13.4. Signaling is differential which is where CAN derives its robust noise immunity and fault tolerance.

Balanced differential signalling reduces noise coupling and allows for high signalling rates over twisted-pair cable. Balanced means that the current flowing in each signal line is equal but opposite in direction, resulting in a field-cancelling effect that is a key to low noise emissions. The use of balanced differential receivers and twisted-pair cabling enhance the common-mode rejection and high noise immunity of a CAN bus. The two signal lines of the bus, CANH and CANL, in the quiescent recessive state, are passively biased to ≈ 2.5 V.

4.12.5 CAN TRANSCEIVER OPERATION

A CAN bus has two states: dominant and recessive. A dominant state is present on the bus when the differential voltage between CANH and CANL is greater than 0.9 V. A recessive state is present on the bus when the differential voltage between CANH and CANL is less than 0.5 V. During a dominant bus state, the CANH pin is high and the CANL pin is low.

During a recessive bus state, both the CANH and CANL pins are in the high impedance state. The driver drives CANH high and CANL low (dominant state) if a logic low is present on TxD. If logic high is present on TxD, the driver output is placed in a high impedance state (recessive state).

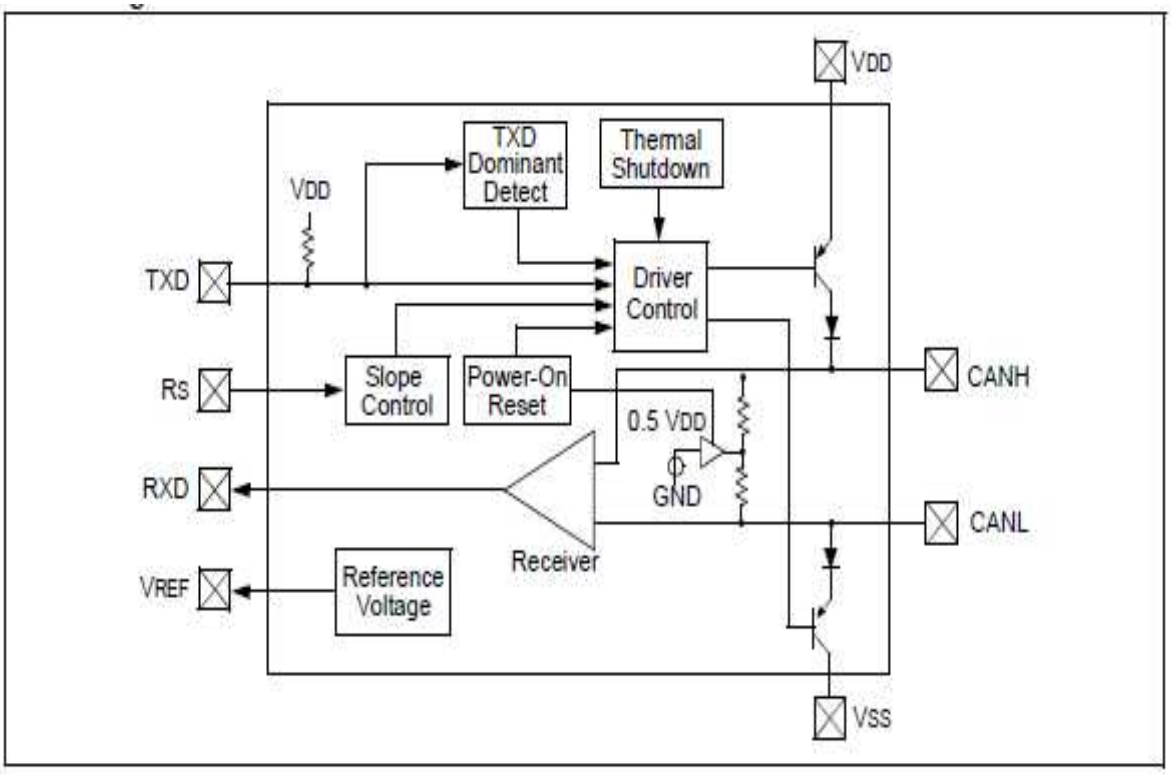


Fig 4.13.5 CAN Transceiver Diagram

CHAPTER 5

ELECTRICALLY ERASABLE PROGRAMMABLE READ ONLY MEMORY (EEPROM)

5.1 Introduction

EEPROM stands for **E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory and is a type of non-volatile memory used in computers and other electronic devices to store small amounts of data that must be saved when power is removed. An EPROM usually must be removed from the device for erasing and programming, whereas EEPROMs can be programmed and erased in-circuit, by applying special programming signals. Originally, EEPROMs were limited to single byte operations which made them slower, but modern EEPROMs allow multi-byte page operations. It also has a limited life - that is, the number of times it could be reprogrammed was limited to tens or hundreds of thousands of times. That limitation has been extended to a million write operations in modern EEPROMs. In an EEPROM that is frequently reprogrammed while the computer is in use, the life of the EEPROM can be an important design consideration. It is for this reason that EEPROMs were used for configuration information, rather than random access memory.

The 24C02 / 24C04 provides 2048 / 4096 bits of serial electrically erasable and programmable read-only memory (EEPROM) organized as 256 / 512 words of 8 bits each. The device is optimized for use in many industrial and commercial applications where low-power and low-voltage operation are essential. This specification covers a range of 4 Kbits I^2C bus EEPROM products, the ST24/25C04 and the ST24/25W04. In the text, products are referred to as ST24/25x04, where "x" is: "C" for Standard version and "W" for

hardware Write Control version. The ST24/25x04 are 4 Kbit electrically erasable programmable memories (EEPROM), organized as 2 blocks of 256 x8 bits. They are manufactured in STMicroelectronics's Hi Endurance Advanced CMOS technology which guarantees an endurance of one million erase/write cycles with a data retention of 40 years. Both Plastic Dual-in-Line and Plastic Small Outline packages are available.

The memories are compatible with the I^2C standard, two wire serial interface which uses a bi-directional data bus and serial clock. The memories carry a built-in 4 bit, unique device identification code (1010) corresponding to the I2C bus definition. This is used together with 2 chip enable inputs (E2, E1) so that up to 4 x 4K devices may be attached to the I2C bus and selected individually. The memories behave as a slave device in the I2C protocol with all memory operations synchronized by the serial clock. Read and write operations are initiated by a START condition generated by the bus master. The START condition is followed by a stream of 7 bits (identification code 1010), plus one read/write bit and terminated by an acknowledge bit.

5.2 PIN DIAGRAM

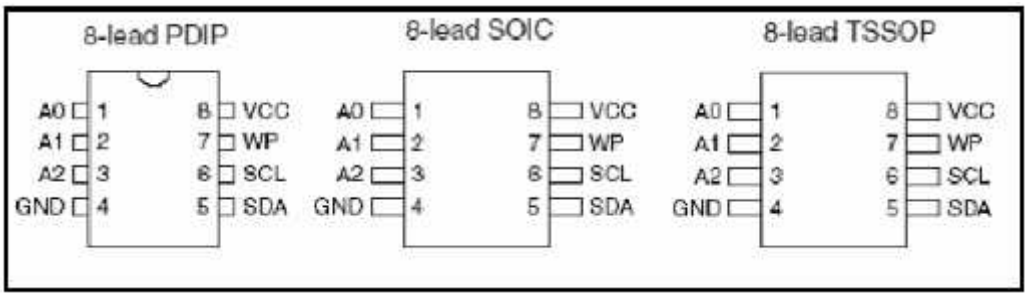


Fig.5.1 PIN Diagram of EEPROM

Pin Name	Function
A0 - A2	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect

GND	Ground
VCC	Power Supply

Table.5.1Pin Configuration

5.2.1 Serial Clock (SCL)

The SCL input pin is used to synchronize all data in and out of the memory. A resistor can be connected from the SCL line to VCC to act as a pull up.

5.2.2 Serial Data (SDA)

The SDA pin is bi-directional and is used to transfer data in or out of the memory. It is an open drain output that may be wire-OR’ed with other open drain or open collector signals on the bus. A resistor must be connected from the SDA bus line to VCC to act as pull up.

5.2.3 Chip Enable (E1 - E2)

These chip enable inputs are used to set the 2 least significant bits (b2, b3) of the 7 bit device select code. These inputs may be driven dynamically or tied to VCC or VSS to establish the device select code.

5.2.4 Protect Enable (PRE)

The PRE input pin, in addition to the status of the Block Address Pointer bit sets the PRE write protection active.

5.2.5 Write Control (WC)

A hardware Write Control feature (WC) is offered only for ST24W04 and ST25W04 versions on pin 7. This feature is useful to protect the contents of the memory from any erroneous erase/write cycle.

CHAPTER 6
REAL TIME CLOCK (RTC)

6.1 INTRODUCTION

As the name recommends are clock modules. The DS1307 real time clock (RTC) IC is an 8 pin device using an I^2C interface. The DS1307 is a low-power clock/calendar with 56 bytes of battery backup SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month and year qualified data. The end date of each month is automatically adjusted, especially for months with less than 31 days. They are available as integrated circuits (ICs) and supervise timing like a clock and also operate date like a calendar. The main advantage of RTC is that they have an arrangement of battery backup which keeps the clock/calendar running even if there is power failure. An exceptionally little current is required for keeping the RTC animated. We can find these RTCs in many applications like embedded systems and computer mother boards, etc. In this article we are going to see about one of the real time clock (RTC), i.e. DS1307.

6.2 Pin Diagram (DS1307)

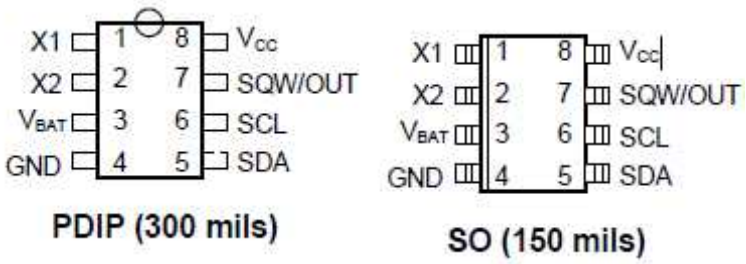


Fig 6.2 Pin Diagram of DS1307

Pin 1, 2: Connections for standard 32.768 kHz quartz crystal. The internal oscillator circuitry is intended for operation with a crystal having a specified load capacitance of 12.5pF. X1 is the input to the oscillator and can alternatively be connected to an external 32.768 kHz oscillator. The output of the internal oscillator, X2 is drifted if an external oscillator is connected to X1.

Pin 3: Battery input for any standard 3V lithium cell or other energy source. Battery voltage should be between 2V and 3.5V for suitable operation. The nominal write protect trip point voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x VBAT nominal. A lithium battery with 48mAhr or greater will backup the DS1307 for more than 10 years in the absence of power at 25ºC. UL recognized to ensure against reverse charging current when utilized as a part of conjunction with a lithium battery.

Pin 4: Ground.

Pin 5: Serial data input/output. The input/output for the I2C serial interface is the SDA, which is open drain and requires a pull up resistor, allowing a pull up voltage upto 5.5V. Regardless of the voltage on VCC.

Pin 6: Serial clock input. It is the I^2C interface clock input and is used in data synchronization.

Pin 7: Square wave/output driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4 kHz, 8 kHz, and 32 kHz). This is also open drain and requires an external pull-up resistor. It requires application of either Vcc or Vb at to operate SQW/OUT, with an allowable pull up voltage of 5.5V and can be left floating, if not used.

Pin 8: Primary power supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and Vcc is below VTP, read and writes are inhibited. However at low voltages, the timekeeping function still functions.

6.3 Working of DS1307

In the simple circuit the two inputs X1 and X2 are connected to a 32.768 kHz crystal oscillator as the source for the chip. VBAT is connected to positive culture of a 3V battery chip. Vcc power to the I2C interface is 5V and can be given using microcontrollers. If the power supply Vcc is not granted read and writes are inhibited. RTC DS1307 Circuit START and STOP conditions are required when a device wants to establish communication with a device in the I2C network. By providing a device identification code and a register address, we can implement the START condition to access the device. The registers can be accessed in serial order until a STOP condition is implemented The START condition and STOP condition when the DS1307 I2C communication with the microcontroller.

6.4 General Description

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I2C™, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24- hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the battery supply. The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The DS1307 operates as a slave device on the I2C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address.

Subsequent registers can be accessed sequentially until a STOP condition is executed. When VCC falls below 1.25 x VBAT, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of tolerance system. When VCC falls below VBAT, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to VCC when VCC is greater than VBAT +0.2V and recognizes inputs when VCC is greater than 1.25 x VBAT. The block diagram in Figure 1 shows the main elements of the serial RTC. DS1307 64 x 8, Serial, I2C Real-Time Clock 7 of 15

6.5 Oscillatory Circuit

The DS1307 uses an external 32.768 kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. If using a crystal with the specified characteristics, the startup time is usually less than one second.

6.6 Clock Accuracy

The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast.

CHAPTER 7

GLOBAL SYSTEM FOR MOBILES (GSM)

7.1 INTRODUCTION

GSM/GPRS module is used to establish communication between a computer and a GSM-GPRS system. Global System for Mobile communication (GSM) is an architecture used for mobile communication in most of the countries. Global Packet Radio Service (GPRS) is an extension of GSM that enables higher data transmission rate. GSM/GPRS module consists of a GSM/GPRS modem assembled together with power supply circuit and communication interfaces (like RS-232, USB, etc) for computer. The MODEM is the soul of such modules.

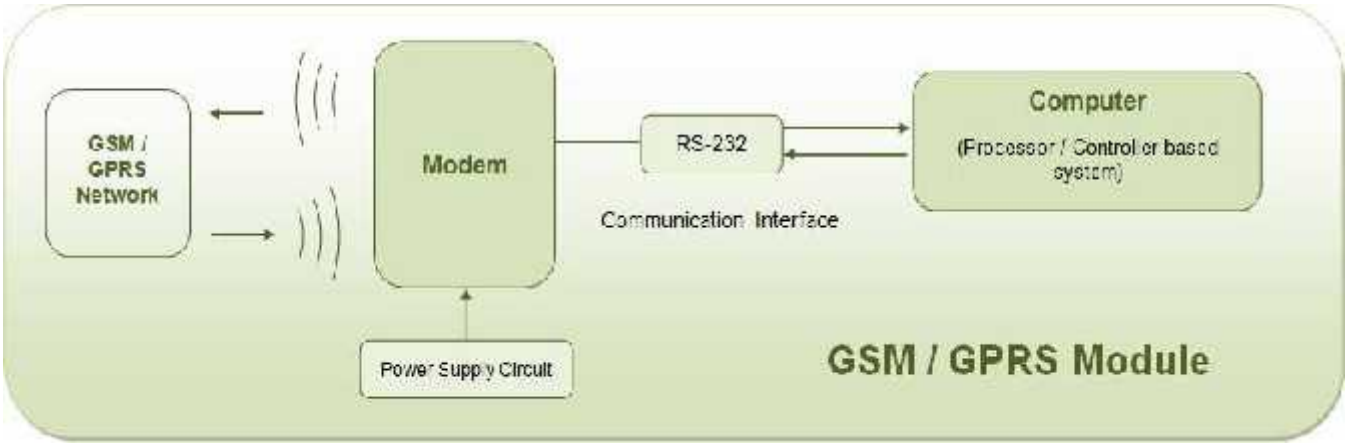


Fig 7.1 Basic Diagram of GSM (Global System for Mobile)

Wireless MODEMs are the MODEM devices that generate, transmit or decode data from a cellular network, for establishing communication between the cellular network and the computer.

These are manufactured for specific cellular network or specific cellular data standard (GSM/UMTS/GPRS/EDGE/HSDPA) or technology (GPS/SIM). Wireless MODEMs like other MODEM devices use serial communication to interface with and need Hayes compatible AT commands for communication with the computer (any microprocessor or microcontroller system).

7.2 GSM/GPRS MODEM

GSM/GPRS MODEM is a class of wireless MODEM devices that are designed for communication of a computer with the GSM and GPRS network. It requires a SIM (Subscriber Identity Module) card just like mobile phones to activate communication with the network. Also they have IMEI (International Mobile Equipment Identity) number similar to mobile phones for their identification.

A GSM/GPRS MODEM can perform the following operations:

1. Receive, send or delete SMS messages in a SIM.
2. Read, add, search phonebook entries of the SIM.
3. Make, Receive, or reject a voice call.

The MODEM needs AT commands, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the GSM and GPRS cellular network. A GSM/GPRS module assembles a GSM/GPRS modem with standard communication interfaces like RS-232 (Serial Port), USB etc., so that it can be easily interfaced with a computer or a microprocessor / microcontroller based system. The power supply circuit is also built in the module that can be activated by using a suitable adaptor.

7.3 Mobile Station (Cell phones and SIM)

A mobile phone and Subscriber Identity Module (SIM) together form a mobile station. It is the user equipment that communicates with the mobile network. A mobile phone comprises of Mobile Termination, Terminal Equipment and Terminal Adapter. Mobile Termination is interfaced with the GSM mobile network and is controlled by a baseband processor. It handles access to SIM, speech encoding and decoding, signaling and other network related tasks. The Terminal Equipment is an application processor that deals with handling operations related to keypad, screen, phone memory and other hardware and software services embedded into the handset. The Terminal Adapter establishes communication between the Terminal Equipment and the Mobile Termination using AT commands. The communication with the network in a GSM/GPRS mobile is carried out by the baseband processor.

AT commands with a GSM/GPRS MODEM or mobile phone can be used to access following information and services:

1. Information and configuration pertaining to mobile device or MODEM and SIM card.
2. SMS services.
3. MMS services.
4. Fax services.
5. Data and Voice link over mobile network.

The Hayes subset commands are called the basic commands and the commands specific to a GSM network are called extended AT commands.

CHAPTER 8

PROJECT DISCRIPTION

8.1 Introduction

Nowadays fudging from due payment of automobile loan, vehicle insurance is an ordinary affair. Currently there is no valid system to track the defaulters, which affects the government exchequer hardly and also affects the prosperity of the individuals. So there is an urgent need to fill this vacuum by modernizing the existing technology in automobile industries. As a matter of initiation, an innovative CAN protocol based control system exclusively for car has been designed and implemented in this project.

8.2 Block Diagram & Description

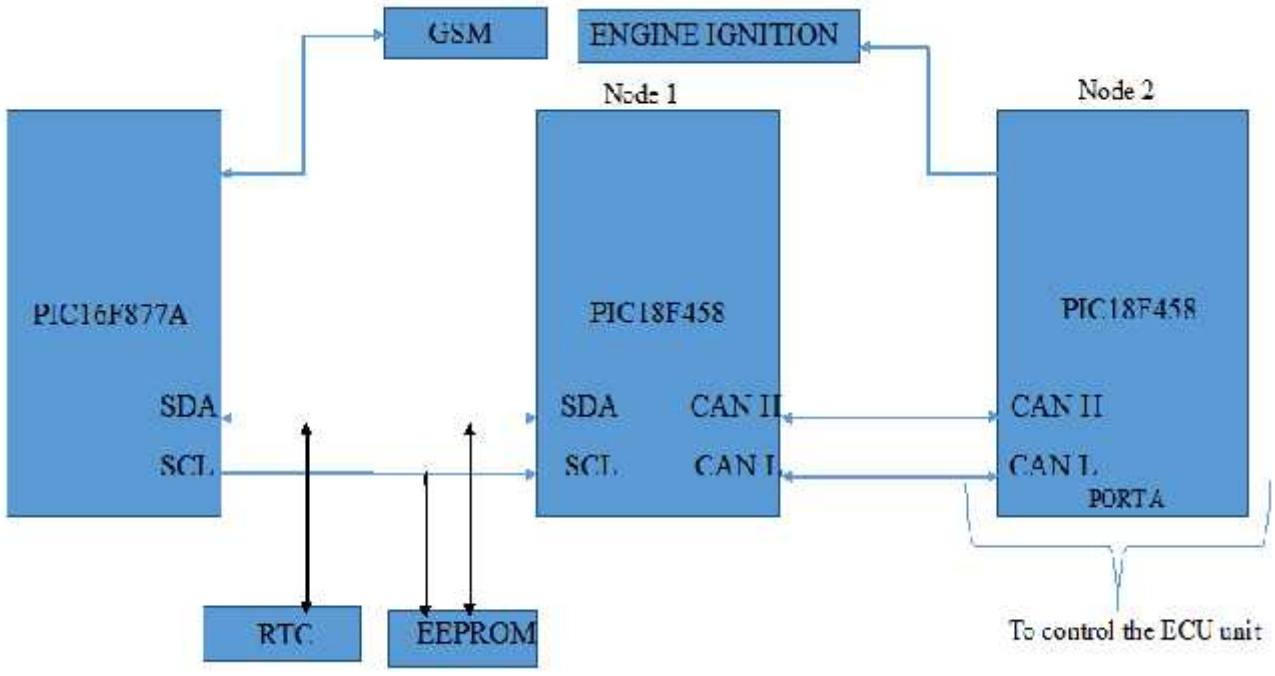


Fig 8.2.1 Block Diagram of Proposed system

The developed system makes use of an embedded system and GSM technology. The proposed system installed in the vehicle can be easily controlled from the loan/Insurance office by sending a message from their

mobile / internet to the vehicle engine by interfacing with CAN bus and GSM modem**.** Figure.8.2.1 shows the block diagram of the proposed work.

The developed system here analyzed, makes use of a CAN based Global System for Mobile communication (GSM) technology. An interfacing GSM module is connected to the PIC16F877A controller which is in turn, linked to the main controller PIC18F458. In this proposed work, Real Time Clock (RTC) and EEPROM data will automatically lock the vehicle on the every 30^{th} day of the month. Once the date exceeds 20^{th} day, the LED glows in the vehicle continuously till 30^{th} of the month. When the RTC reflects the date 30, the PIC16F877A Controller sends the data '0' to the EEPROM using Inter-Integrated Circuit (I^2C). The message stored in the EEPROM will be read by the PIC18F458 controller whenever the car is turned on.

If the stored data reflects '0', the car is locked immediately through Control Area Network (CAN) Bus. The above mentioned system is designed and installed in the vehicle. If the user paid the loan / Insurance due amount on time, the information is being used by the loan/Insurance officer for further processing. The message '$1'which will be sent by the banker / insurer to the PIC16F877A controller, that reads the SMS and is stored in the EEPROM.

Whenever the car is turned on, the main controller PIC18F458 reads the SMS ($1) from the EEPROM that will be communicated to the Engine Control Unit (ECU) through Control Area Network (CAN) Bus, which leads to unlock the car immediately. The main concept in this design is introducing the mobile communication into a CAN bus system. Figure .1 shows the block diagram of the proposed system. Our research based on PIC16F877A & PIC18F458 microcontroller due to its low cost and the PIC is a famous, known, easy to programmed and build with another devices directly such as LCD, it's a reliable controller even its accessories is available such as its programmer, compiler (Micro C), and simulator such as PROTUS.

The proposed system has following features & advantages:

Features of new invention:

- Compact Size of the board.
- In built theft avoidance system.
- Removal of PIC controller unit leads to zero ignitions.
- Automatic vehicle loan and insurance dues collection.
- Jammer usage won't affect the control unit because of Real Time Clock (RTC).

8.3 Ignition Control of the Vehicle

The ignition system of an internal-combustion engine is an important part of the overall engine system that provides for the timely burning of the fuel mixture within the engine. All conventional petrol (gasoline) engines require an ignition system. The ignition system is usually switched on/off through a lock switch, operated with a key or code patch. The ignition system works in perfect concert with the rest of the engine of a vehicle. In this proposed work, the ignition control of the vehicle is based on the information stored in EEPROM.

Once, the vehicle is loan / Insurance due date has been exceeded, based on the output of the real time clock the PIC16F877A controller sends a data '0' to the PIC16F877A controller, and it is stored in the EEPROM. The message stored in the EEPROM will be read by the PIC18F458 controller whenever the car is turned on, and then the car is locked immediately. If the user paid the loan / Insurance due amount on time, the information is being used by the loan/Insurance officer for further processing. The message '$1'which will be sent by the banker / insurer to the PIC16F877A controller, that reads the SMS and is stored in the EEPROM. Whenever the car is turned on, the main controller PIC18F458 reads the SMS ($1) from the EEPROM that will be communicated to the Engine Control Unit (ECU) through Control Area Network (CAN) Bus, which leads to unlock the car immediately.

CHAPTER 9

SOFTWARE IMPLEMENTATION

9.1 Introduction

PROTEUS software is used for simulation and MP LAB for compilation. In this proposed work, Real Time Clock (RTC) and EEPROM data will automatically lock the vehicle on the every 30th day of the month. Once the date exceeds 20th day, the LED glows in the vehicle continuously till 30th of the month (Figure.9.1). When the RTC reflects the date 30, the PIC16F877A Controller sends the data '0' to the EEPROM using Inter-Integrated Circuit (I^2C). The message stored in the EEPROM will be read by the PIC18F458 controller whenever the car is turned on. If the stored data reflects '0', the car is locked immediately through Control Area Network (CAN) Bus. The above mentioned system is designed and installed in the vehicle.

9.2 Simulation Results

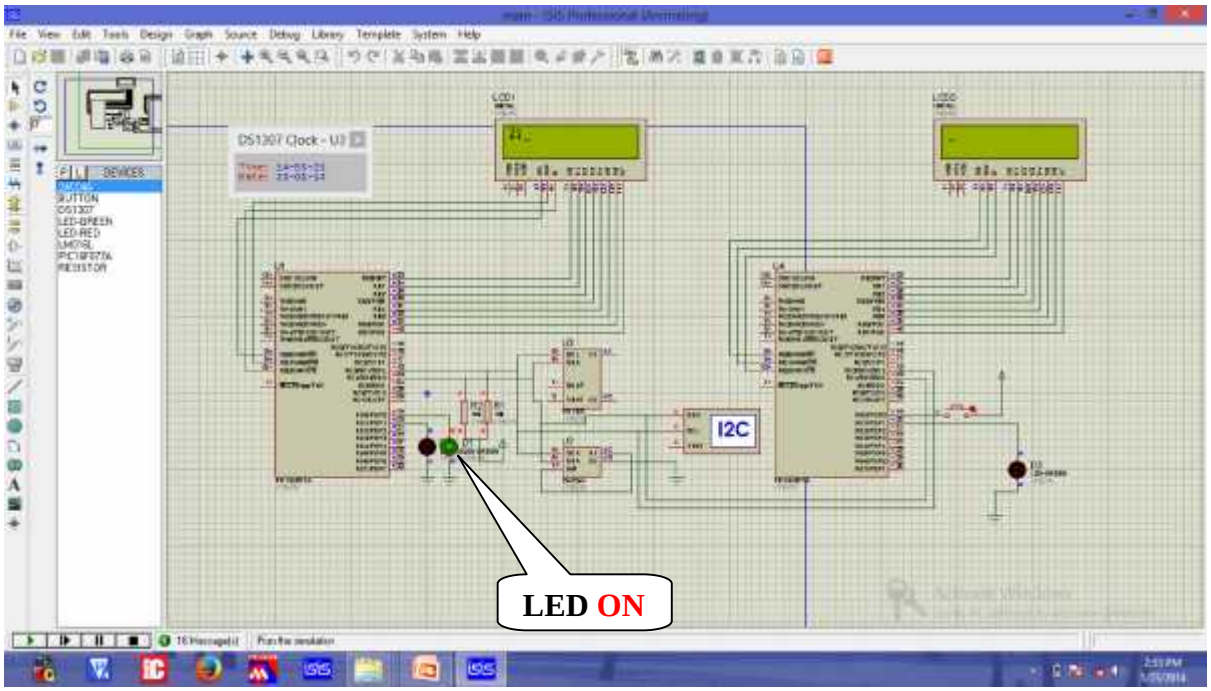
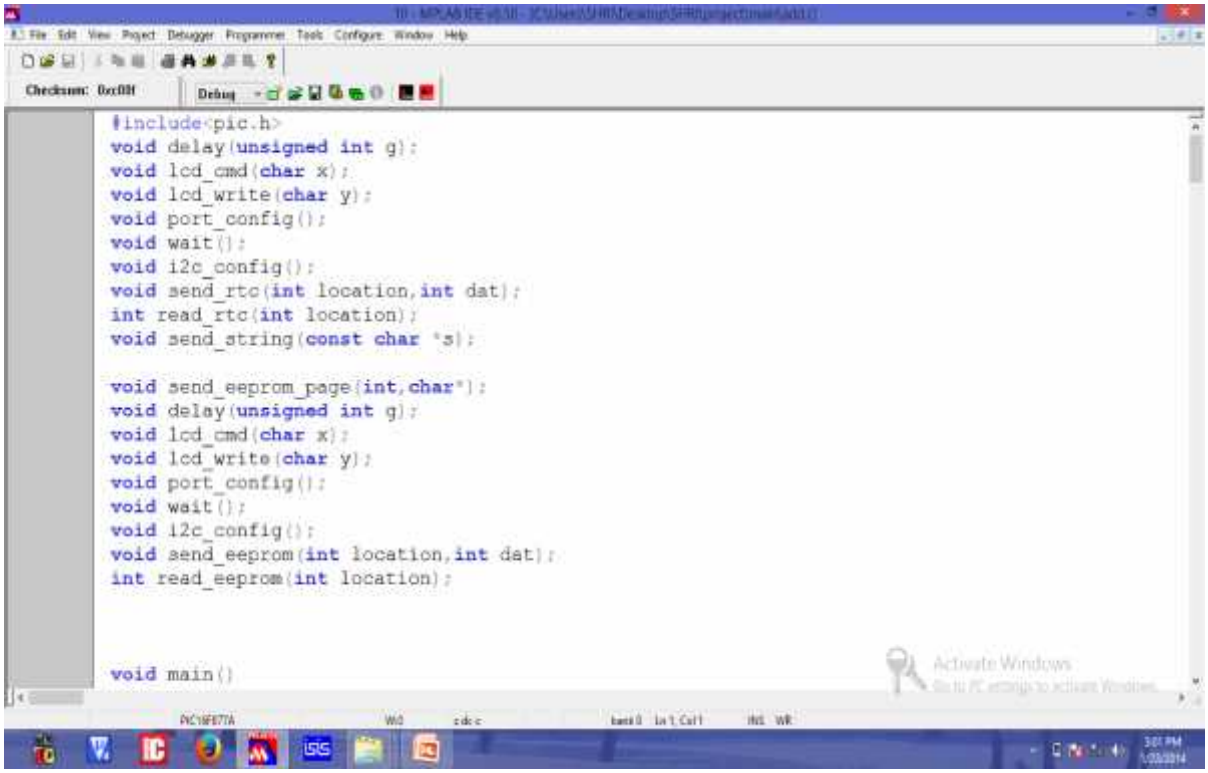


Fig. 9.1 Alert Intimation to the user



```
#include<pic.h>
void delay(unsigned int g);
void lcd_cmd(char x);
void lcd_write(char y);
void port_config();
void wait();
void i2c_config();
void send_rtc(int location,int dat);
int read_rtc(int location);
void send_string(const char *s);

void send_eeprom_page(int,char*);
void delay(unsigned int g);
void lcd_cmd(char x);
void lcd_write(char y);
void port_config();
void wait();
void i2c_config();
void send_eeprom(int location,int dat);
int read_eeprom(int location);


void main()
```

Fig. 9.2 Programming for Microcontroller

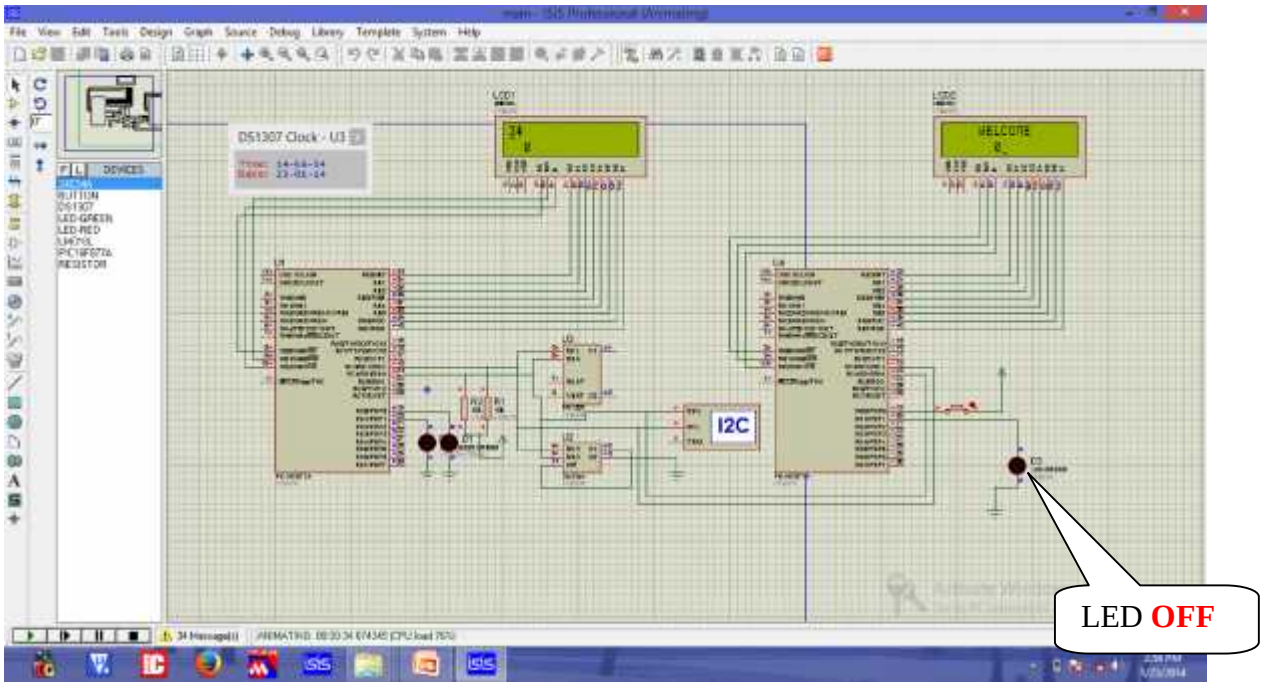


Fig. 9.3 Main Controller OFF

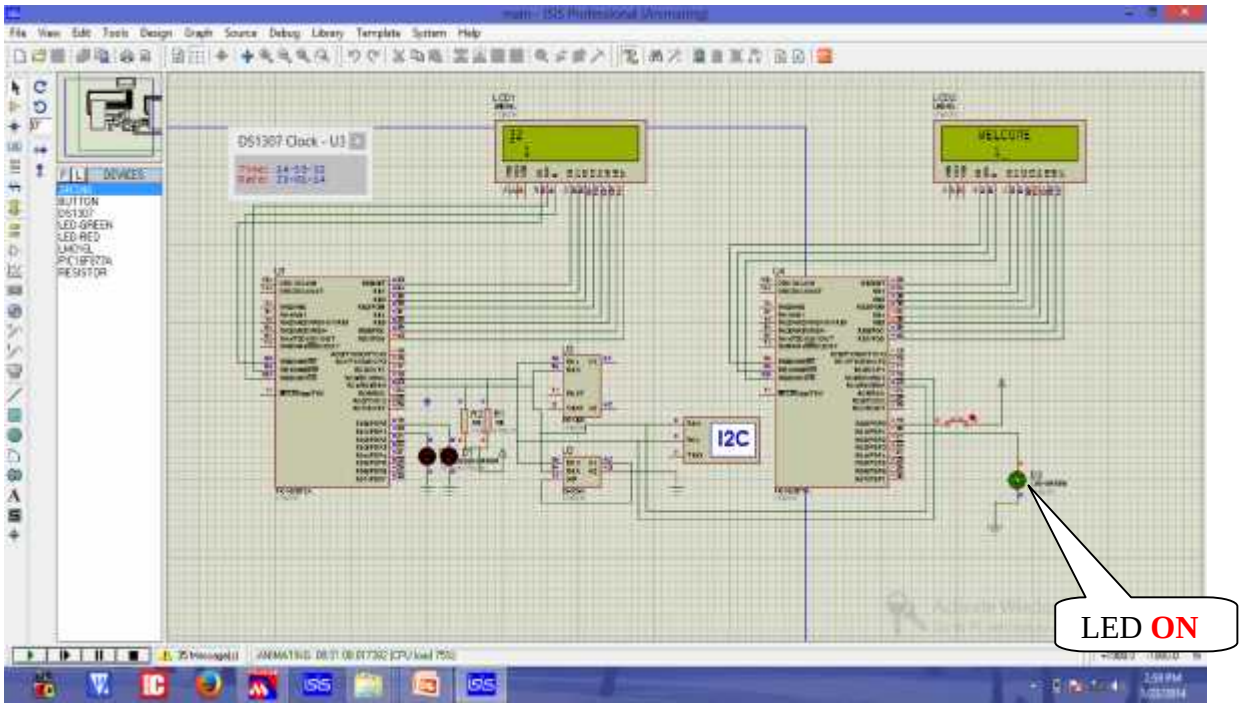


Fig. 9.4 Main Controller ON

Figure.9.2 shows the programming details of the microcontrollers. When the RTC shows the date 30, the PIC16F877A Controller sends the data '0' to the EEPROM using Inter-Integrated Circuit (I^2C) & PIC18F458 Controller will check the EEPROM whenever the vehicle gets started. If the EEPROM data is 0, the vehicle will not ignite. If it is 1, the vehicle will ignite the engine. (Fig.9.3 & 9.4).

- If the EEPROM Data is 0, the LED connected in the PIC18F458 controller will be in OFF condition.
- If the EEPROM Data is 1, the LED connected in the PIC18F458 controller will be in ON condition.

CHAPTER 10
HARDWARE IMPLEMENTATION

10.1 Hardware Description

The results are obtained after carrying out the experimentation by using the following hardware components. The component includes PIC 16F877A & PIC 18F458 Controller, RTC (DS1307), EEPROM (24C04), GSM Module (SIM300), and LCD Display. Fig. 10.1 shows PIC Controllers, RTC (DS1307), EEPROM (24C04), GSM Module (SIM300), and LCD Display are interfaced on a single board which is embedded to a vehicle as a control unit. The robustness of the system developed was easily demonstrated using remote control unit.

In this part, PIC16F877A controller is used to observe the Real Time Clock (RTC), receive the SMS from the loan / Insurance office and to store the data to the EEPROM through I^2C serial interface communication protocol. Though the controllers have an inbuilt EEPROM memory, external 128bytes EEPROM (24C04) attached in this system to hasten the process. Based on the EEPROM data, the PIC18F458 controller will lock / unlock the vehicle through Control Area Network (CAN) Bus. On 20^{th} day of every month, LED will blink inside the vehicle and 30^{th} day the vehicle will be locked automatically with the help of RTC (DS1307). Hence the RTC plays an important role in this project work.

If the user paid the loan / Insurance due amount on time, the information is being used by the loan/Insurance officer for further processing. The message '$1'which will be sent by the banker / insurer to the PIC16F877A controller, that reads the SMS and is stored in the EEPROM.

CHAPTER 11

SCOPE FOR FUTURE WORK

- Whenever the vehicle is turned on, an alert SMS to the customer & voice message inside the vehicle may also additionally be introduced.
- Introduction of theft avoidance system may also be added which could improve the vehicle security and accessibility.
- Vehicle tracking system might also be initiated with the help of GPS.

CHAPTER 12

CONCLUSION

This project introduces an embedded system with a combination of CAN bus systems. It makes the way for the prompt payment of the dues by the customer. The simulation of the circuit design and its implementation is carried out using PROTEUS software. Inculcating hardware implementation, ensuring vehicle security, user friendliness are some of the valuable features enlisted in this project. With the effective utilization of advance automotive communication technology, this project becomes an eye opener to student's fraternity, the researchers, innovators and those who are more interested for framing a healthy economy of the Nation.

CHAPTER 13

REFERENCES

[1] Ali Subhi Hiyawi Alsadi, " *The Development of Embedded GPS-GSM Based Real Time Vehicle Tracking System*," Eng. &Tech Journal, Vol.31, Part (A), No.10, 2013, pp.1982-1999.

[2] Pravada P. Wankhade1 and Prof. S.O. Dahad, " *Real Time Vehicle Locking and Tracking System using GSM and GPS Technology-An Anti-theft System* "International Journal of Technology And Engineering System(IJTES):Jan –March 2011- Vol.2.No.3,*pp.272-275.*

[3] Omidiora E. O., Fakolujo O. A., Arulogun O. T., Aborisade D. O. "*A Prototype of a Fingerprint Based Ignition Systems in Vehicles*" European Journal of Scientific Research, Vol.62 No.2 (2011), pp. 164-171.

[4] Mohammed Abuzalata, Muntaser Momani, Sayel Fayyad and Suleiman Abu-Ein "*A Practical Design of Anti-Theft Car Protection System Based on Microcontroller"* American Journal of Applied Sciences 9 (5): 709-716, 2012.

[5] Kunal Maurya , Mandeep Singh , Neelu Jain *"Real Time Vehicle Tracking System using GSM and GPS Technology- An Anti-theft Tracking System"* International Journal of Electronics and Computer Science Engineering, Volume1,Number 3, pp.1103-1107.

[6] O. Shoewu , Segun O. Olatinwo "*Design and Implementation of a Microcontroller Based Automatic Gate*" African Journal of Computing & ICT, Vol 6. No. 1, March 2013,pp.21-32.

[7] Philip A. Adewuyi, Muniru O. Okelola, Adewale O. Jemilehin "*PIC Based Model of an Intelligent Gate Controller"* International Journal of Engineering and Advanced Technology , Volume-2, Issue-3, February 2013,pp.601-605