SOUND OPTIMISER

A PROJECT REPORT

Submitted by

NIRMAL KUMAR A	**Reg. No.:1110107063**
NOEL SHAJU	**Reg. No.:1110107064**
RAJAMANIKANDAN V.M	**Reg. No.:1110107071**
SATHISH KUMAR K	**Reg. No.:1110107082**

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)

BONAFIDE CERTIFICATE

Certified that this project report titled **“SOUND OPTIMISER**” is the bonafide work of **“SATHISH KUMAR K, NIRMAL KUMAR A,RAJAMAIKANDAN V M ,NOEL SHAJU”**who carried out the project work under my supervision.

SIGNATURE
Ms.A.Amsaveni M.E.,(Ph.D.)
Assosiate Professor
Department of ECE,
Kumaraguru College of technology .
Coimbatore

SIGNATURE
Dr. Rajeswari Mariappan, Ph.D
Head of the Department
Department of ECE,
Kumaraguru College of Technology
Coimbatore.

The candidates with Register numbers 1110107063,1110107064, 1110107071 and 1110107082 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.A.Amsaveni, M. E.(Ph.D)** Assosiate Professor, for her continuous support throughout the course.

In particular, We wish to thank and express our everlasting gratitude to the Supervisor **Ms. A. Kalaiselvi, M.E., (Ph. D)** Assistant 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

The main objective of this project is to design a kit to analyze sound from external noise. In order to develop this application two sound sources are needed namely Internal Sound and External Noise. Our design will work according to the external noise. Whenever external noise gets increased internal sound will get adjusted accordingly. Internal and external sound will be shown in LCD Displays. PIC microcontroller is used as a controller device.An external mic will be added with the controller for analyzing the external noise. The external noise will be get amplified and passed to the band pass filter(BPF) and the filtered output will be given as input to the controller device. The internal sound will be displayed in the LCD. The Driver output from the controller will be connected with the Audio controller, which will adjust the volume of the internal sound. In case of increase in the external noise, internal sound will be adjusted according to the external noise.

LIST OF ABBREVATIONS

PIC	Peripheral Interface controller
EEPROM	Electrically Erasable Programmable Read Only Memory
SRAM	Static Random Access Memory
LCD	Liquid crystal display
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
ADC	Analog to Digital Converter
Txr PIN	Transmitter PIN
Rxr PIN	Reciever PIN
MIC	Microphone
BPF	Band Pass filter
BORN	Brown out reset enable
WDTDIS	Watchdog timer disable

CHAPTER 1

Introduction

Nowadays people are unable to enjoy the music or speeches or any pleasent audio from their electronic gadgets or the home theatres or fromthe speakers.Every day thousands of the noise sources created from the surroundings whic might be a more stressful disturbance that makes the unpleasant hearing experience of he pleasent music.

The main objective of the project is to make the people experience a pleasent noise that makes them happy without any obstacle.Whatever be the noise sources causing disturbances they can be eliminated by the use of the sound optimiser.This project which will automatically adjust the internal or sound level according to the noise level.

The project involves the use of the mic,an amplifier with filter,PIC microcontroller, Driver circuit.To do this project we need the microphone for reading the noise level and we also use an amplifier to amplify the external noise and then read the noise level and check with a preset threshold value of the volume of the sound level by use of microcontroller and if the input sound level exceed the threshold volume then automatically the volume will be adjusted by the system.This project can be used in the theatre Auditoriums,Music systems in Home etc.,

This project is helpful interms of avoiding the use of remote control which provides manual adjustment for the sound level.The repeated use of the remote control to control the volume can be eliminated by the use of the sound optimiser circuit.

CHAPTER 2

HARDWARE DESCRIPTION

2.1 BLOCK DIAGRAM

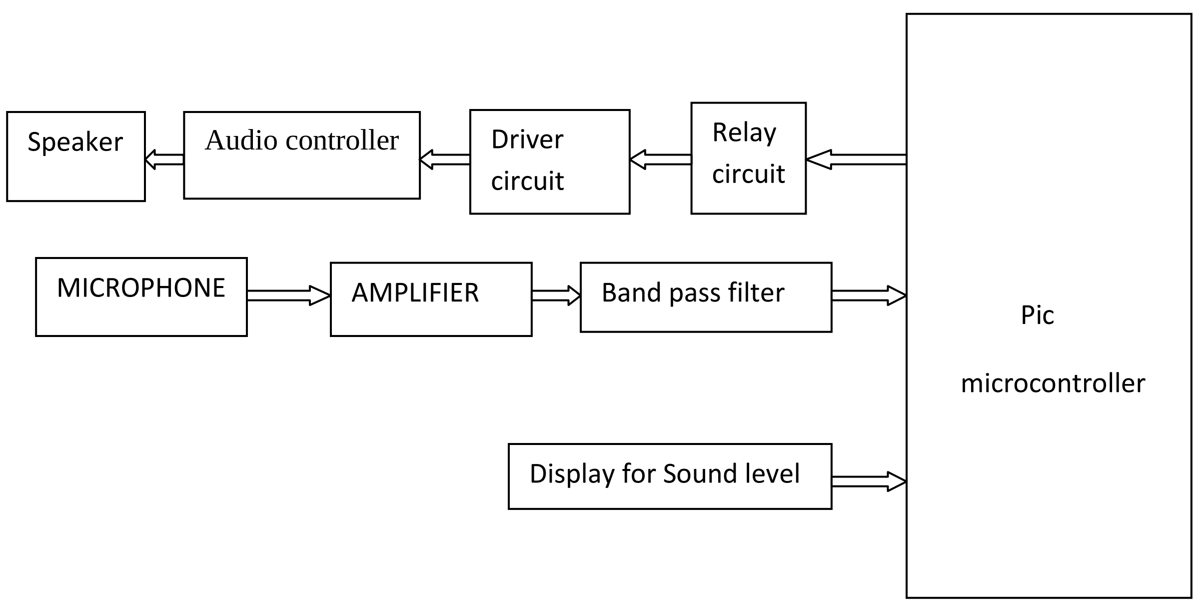


Fig 2.1 Block Diagram

2.2 POWER SUPPLY

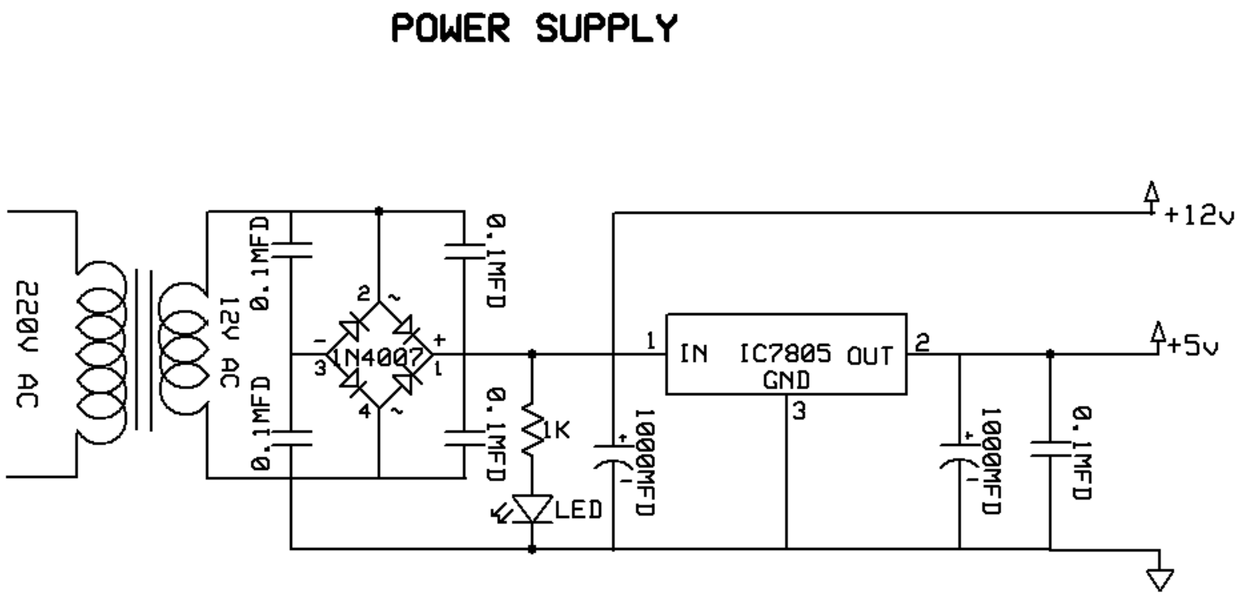


Fig 2.2 Power supply

Power supply is a reference to a source of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

Power supplies for electronic devices can be broadly divided into linear and switching power supplies. The linear supply is a relatively simple design that becomes increasingly bulky and heavy for high current devices; voltage regulation in a linear supply can result in low efficiency. A switched-mode supply of the same rating as a linear supply will be smaller, is usually more efficient, but will be more complex.

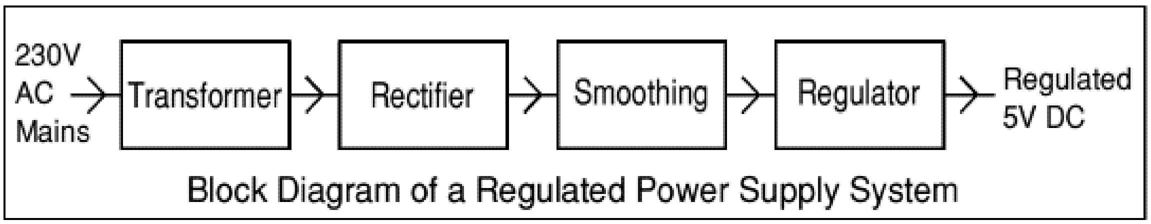
Linear Power supply:

An AC powered linear power supply usually uses a transformer to convert the voltage from the wall outlet (mains) to a different, usually a lower voltage. If it is used to produce DC, a rectifier is used. A capacitor is used to smooth the pulsating current from the rectifier. Some small periodic deviations from smooth direct current will remain, which is known as ripple. These pulsations occur at a frequency related to the AC power frequency (for example, a multiple of 50 or 60 Hz).

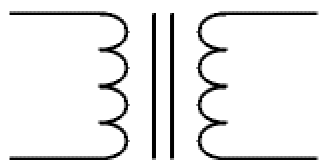
The voltage produced by an unregulated power supply will vary depending on the load and on variations in the AC supply voltage. For critical electronics applications a linear regulator will be used to stabilize and adjust the voltage. This regulator will also greatly reduce the ripple and noise in the output direct current. Linear regulators often provide current limiting, protecting the power supply and attached circuit from over current.

Adjustable linear power supplies are common laboratory and service shop test equipment, allowing the output voltage to be set over a wide range. For example, a bench power supply used by circuit designers may be adjustable up to

30 volts and up to 5 amperes output. Some can be driven by an external signal, for example, for applications requiring a pulsed output.



Transformer:



Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC.

Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in UK) to a safer low voltage.

The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead they are linked by an alternating magnetic field created in the soft-iron core of the transformer. The two lines in the middle of the circuit symbol represent the core.

Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up.

The ratio of the number of turns on each coil, called the turn's ratio, determines the ratio of the voltages. A step-down transformer has a large number of turns on its primary (input) coil which is connected to the high voltage mains supply, and a small number of turns on its secondary (output) coil to give a low output voltage.

Turns ratio=Vp/Vs=Nn/Ns and Power out=Power in

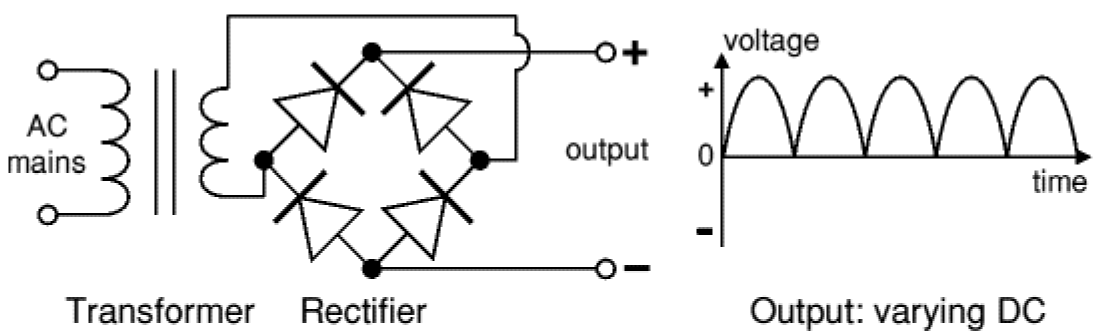
Vs*Is=Vp * Ip

Vp = primary (input) voltage	Vs = secondary (output) voltage
Np = number of turns on primary coil	Ns = number of turns on secondary coil
Ip = primary (input) current	Is = secondary (output) current

The low voltage AC output is suitable for lamps, heaters and special AC motors. It is not suitable for electronic circuits unless they include a rectifier and a smoothing capacitor.

Rectifier:

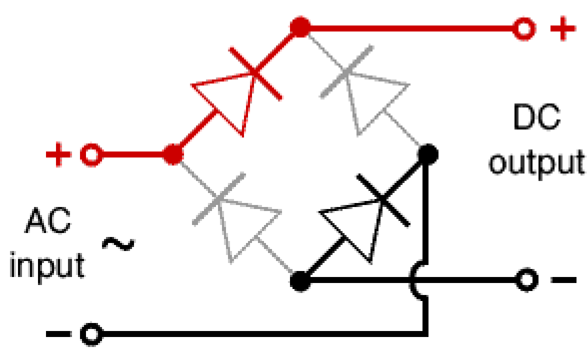
There are several ways of connecting diodes to make a rectifier to convert AC to DC. The bridge rectifier is the most important and it produces full-wave varying DC. A full-wave rectifier can also be made from just two diodes if a centre-tap transformer is used, but this method is rarely used now that diodes are cheaper. A single diode can be used as a rectifier but it only uses the positive (+) parts of the AC wave to produce half-wave varying DC.



The varying DC output is suitable for lamps, heaters and standard motors. It is not suitable for electronic circuits unless they include a smoothing capacitor.

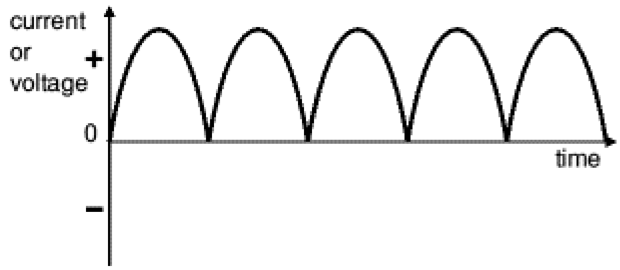
Bridge rectifier:

A bridge rectifier can be made using four individual diodes, but it is also available in special packages containing the four diodes required. It is called a full-wave rectifier because it uses the entire AC wave (both positive and negative sections). 1.4V is used up in the bridge rectifier because each diode uses 0.7V when conducting and there are always two diodes conducting, as shown in the diagram below. Bridge rectifiers are rated by the maximum current they can pass and the maximum reverse voltage they can withstand (this must be at least three times the supply RMS voltage so the rectifier can withstand the peak voltages). Please see the Diodes page for more details, including pictures of ridge rectifiers.



Alternate pairs of diodes conduct, changing over the connections so the alternating directions of AC are converted to the one direction of DC.

Output: full-wave varying DC: (using the entire AC wave):

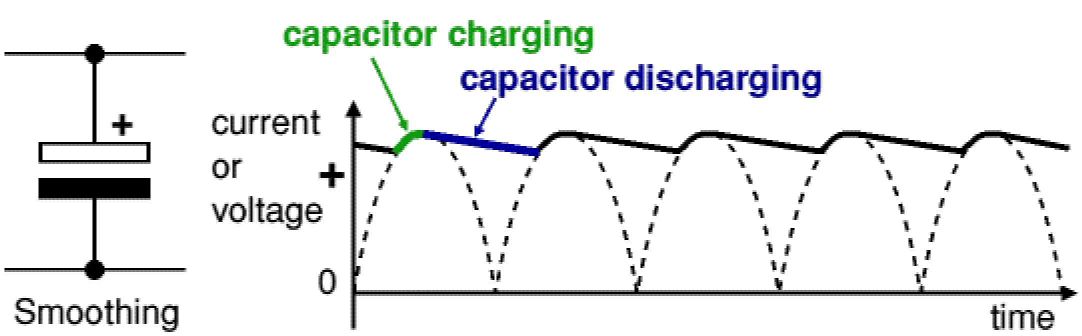


Single diode rectifier:

A single diode can be used as a rectifier but this produces **half-wave** varying DC which has gaps when the AC is negative. It is hard to smooth this sufficiently well to supply electronic circuits unless they require a very small current so the smoothing capacitor does not significantly discharge during the gaps. Please see the Diodes page for some examples of rectifier diodes

Smoothing:

Smoothing is performed by a large value electrolytic capacitor connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The diagram shows the unsmoothed varying DC (dotted line) and the smoothed DC (solid line). The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output.



Note that smoothing significantly increases the average DC voltage to almost the peak value (1.4 × RMS value). For example 6V RMS AC is rectified to full wave DC of about 4.6V RMS (1.4V is lost in the bridge rectifier), with smoothing this increases to almost the peak value giving 1.4 × 4.6 = 6.4V smooth DC.

Smoothing is not perfect due to the capacitor voltage falling a little as it discharges, giving a small ripple voltage. For many circuits a ripple which is 10% of the supply voltage is satisfactory and the equation below gives the required value for the smoothing capacitor. A larger capacitor will give fewer ripples. The capacitor value must be doubled when smoothing half-wave DC.

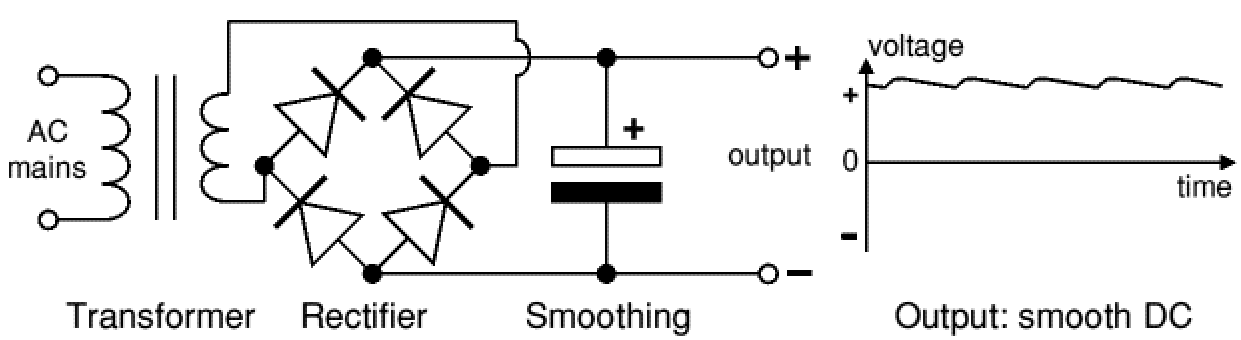
Smoothing Capacitor for 10% ripple, C=5*10/vs.*f

C = smoothing capacitance in farads (F)

Io = output current from the supply in amps (A)

Vs = supply voltage in volts (V), this is the peak value of the unsmoothed DC

f = frequency of the AC supply in hertz (Hz), 50Hz in the UK.



The smooth DC output has a small ripple. It is suitable for most electronic circuits.

Regulator:

Voltage regulator ICs are available with fixed (typically 5, 12 and 15V) or variable output voltages. They are also rated by the maximum current they can pass. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current ('overload protection') and overheating ('thermal protection').

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and current.

Many of the fixed voltage regulator ICs has 3 leads and look like power transistors, such as the 7805 +5V 1A regulator shown on the right. They include a hole for attaching a heat sink if necessary.

1. Positive regulator
    1. input pin
    2. ground pin
    3. output pin

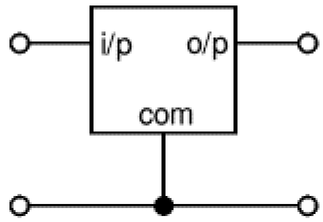
It regulates the positive voltage

2. Negative regulator has

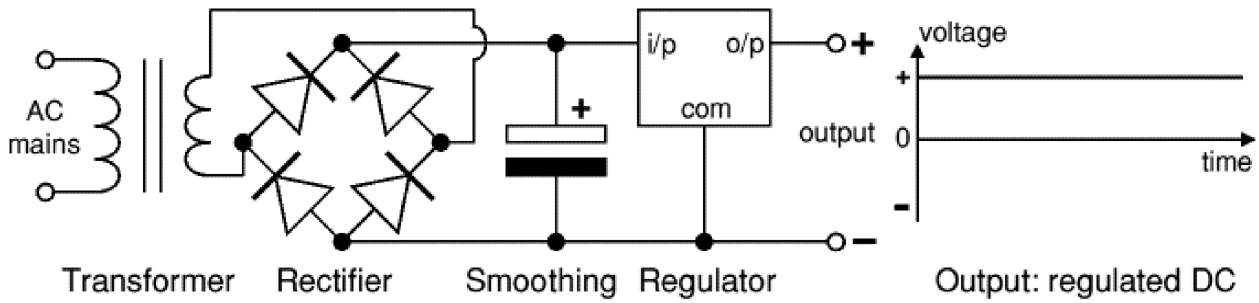
    4. ground pin
    5. input pin

6. output pin

It regulate the negative voltage



Regulator



The regulated DC output is very smooth with no ripple. It is suitable for all electronic circuits.

2.2 PIC MICROCONTROLLER

1. PIC 16F877 Architecture

PIC 16F877 is a 40-pin 8-Bit CMOS FLASH Microcontroller from Microchip. The core architecture is high-performance RISC CPU with only 35 single word[1] instructions. Since it follows the RISC architecture, all single cycle instructions take only one instruction cycle except for program branches which take two cycles. 16F877 comes with 3 operating speeds with 4, 8, or 20 MHz clock input. Since each instruction cycle takes four operating clock cycles, each instruction takes 0.2 s when 20MHz oscillator is used.

It has two types of internal memories: program memory and data memory. Program memory is provided by 8K words (or 8K*14 bits) of FLASH Memory, and data memory has two sources. One type of data memory is a 368-byte RAM (random access memory) and the other is 256-byte EEPROM (Electrically erasable programmable ROM).

The core feature includes interrupt capability up to 14 sources, power saving SLEEP mode, and single 5V In-Circuit Serial Programming (ICSP) capability. The sink/source current, which indicates a driving power from I/O port, is high with 25mA. Power consumption is less than 2 mA in 5V operating condition.

The peripheral features include:

3 time blocks: Timer0 for 8-bit timer/counter; Timer1 for 16-bit timer/counter; and Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler.

Two Capture, Compare, PWM modules for capturing, comparing 16-bit, and PWM generation with 10-bit resolution.

10-bit multi-channel (max 8)Analog-to-Digital converter module.

Synchronous Serial Port (SSP) with SPI (Master Mode) and I^2C^2 (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

I/O ports.

The key feature of 16F877 is summarized below:

FLASH Program Memory (14-bit word)	8K Words
Data Memory (RAM)	368 Bytes
Data Memory (EEPROM)	256 Bytes
Interrupts	14
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 channels
Instruction Set	35 Instructions

3

2. Pin and Package

There are three package types are available: DIP, PLCC, and QFP. This book assumes that we all use the DIP because of its best fit to breadboard or proto-board.

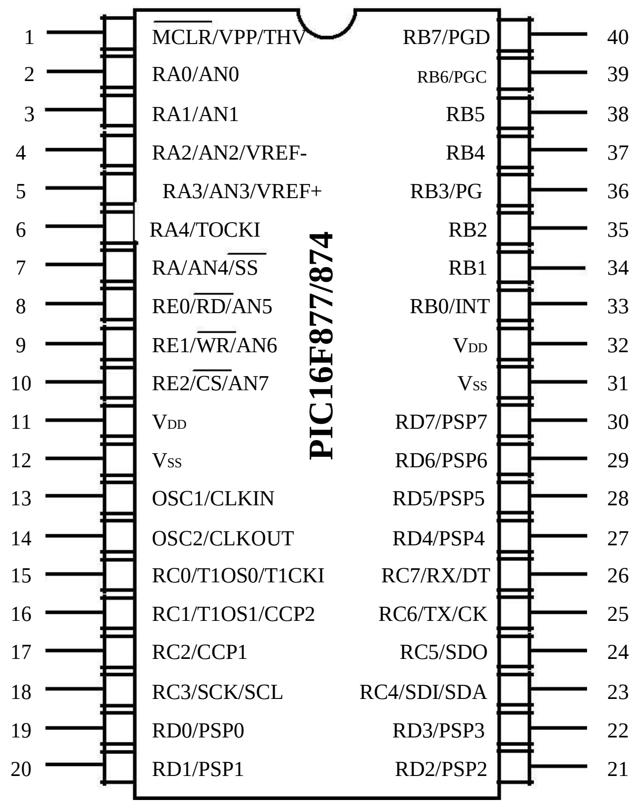


Fig 2.3 Pic 16f877 pin diagram

PIC microcontroller

PIC is a family of Harvard architecture microcontrollers made by Microchip Technology, derived from the PIC1640. Originally developed by General Instrument's Microelectronics Division. The name PIC initially referred to "**Programmable Interface Controller**".

PICs are popular with both industrial developers and hobbyists alike due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, and serial programming (and re-programming with flash memory) capability.

Microchip announced on February 2008 the shipment of its six billionth PIC processor.

Core architecture

The PIC architecture is characterized by the following features:

- Separate code and data spaces (Harvard architecture) for devices other than PIC32, which has a Von Neumann architecture.
- A small number of fixed length instructions
- Most instructions are single cycle execution (2 clock cycles), with one delay cycle on branches and skips
- One accumulator (W0), the use of which (as source operand) is implied (i.e. is not encoded in the opcode)
- All RAM locations function as registers as both source and/or destination of math and other functions.
- A hardware stack for storing return addresses
- A fairly small amount of addressable data space (typically 256 bytes), extended through banking
- Data space mapped CPU, port, and peripheral registers
- The program counter is also mapped into the data space and writable (this is used to implement indirect jumps).

There is no distinction between memory space and register space because the RAM serves the job of both memory and registers, and the RAM is usually just referred to as the register file or simply as the registers.

Data space (RAM)

PICs have a set of registers that function as general purpose RAM. Special purpose control registers for on-chip hardware resources are also mapped into the data space. The addressability of memory varies depending on device series, and all PIC devices have some banking mechanism to extend addressing to additional memory. Later series of devices feature move instructions which can cover the whole addressable space, independent of the selected bank. In earlier devices, any register move had to be achieved via the accumulator.

To implement indirect addressing, a "file select register" (FSR) and "indirect register" (INDF) are used. A register number is written to the FSR, after which reads from or writes to INDF will actually be to or from the register pointed to by FSR. Later devices extended this concept with post- and pre- increment/decrement for greater efficiency in accessing sequentially stored data. This also allows FSR to be treated almost like a stack pointer (SP).

External data memory is not directly addressable except in some high pin count PIC18 devices.

Code space

The code space is generally implemented as ROM, EPROM or flash ROM. In general, external code memory is not directly addressable due to the lack of an external memory interface. The exceptions are PIC17 and select high pin count PIC18 devices.

Word size

The word size of PICs can be a source of confusion. All PICs handle (and address) data in 8-bit chunks, so they should be called 8-bit microcontrollers. However, the unit of addressability of the code space is not generally the same as the data space.

For example, PICs in the baseline and mid-range families have program memory addressable in the same wordsize as the instruction width, i.e. 12 or 14 bits respectively. In contrast, in the PIC18 series, the program memory is addressed in 8-bit increments (bytes), which differs from the instruction width of 16 bits.

In order to be clear, the program memory capacity is usually stated in number of (single word) instructions, rather than in bytes.

Stacks

PICs have a hardware call stack, which is used to save return addresses. The hardware stack is not software accessible on earlier devices, but this changed with the 18 series devices.

Hardware support for a general purpose parameter stack was lacking in early series, but this greatly improved in the 18 series, making the 18 series architecture more friendly to high level language compilers.

Limits

The PIC architectures have several limits:

- Only one accumulator
- A small instruction set
- Operations and registers are not orthogonal; some instructions can address RAM and/or immediate constants, while others can only use the accumulator
- Memory must be directly referenced in arithmetic and logic operations, although indirect addressing is available via 2 additional registers
- Register-bank switching is required to access the entire RAM of many devices

The following limitations have been addressed in the PIC18, but still apply to earlier cores:

- Conditional skip instructions are used instead of conditional jump instructions used by most other architectures
- Indexed addressing mode is very rudimentary
- Stack:
  - The hardware call stack is so small that program structure must often be flattened
  - The hardware call stack is not addressable, so pre-emptive task switching cannot be implemented
  - Software-implemented stacks are not efficient, so it is difficult to generate reentrant code and support local variables
- Program memory is not directly addressable, and thus space-inefficient and/or time-consuming to access. (This is true of most Harvard architecture microcontrollers.)

With paged program memory, there are two page sizes to worry about: one for CALL and GOTO and another for computed GOTO (typically used for table lookups). For example, on PIC16, CALL and GOTO have 11 bits of addressing, so the page size is 2048 instruction words. For computed GOTOs, where you add to PCL, the page size is 256 instruction words. In both cases, the upper address bits are provided by the PCLATH register. This register must be changed every time control transfers between pages. PCLATH must also be preserved by any interrupt handler.

PIC24 and dsPIC 16-bit microcontrollers

In 2001, Microchip introduced the dsPIC series of chips, which entered mass production in late 2004. They are Microchip's first inherently 16-bit microcontrollers. PIC24 devices are designed as general purpose microcontrollers. dsPIC devices include digital signal processing capabilities in addition.

Architecturally, although they share the PIC moniker, they are very different from the 8-bit PICs. The most notable differences are:

- they feature a set of 16 working registers (W0-W15)
- they fully support a stack in RAM, and do not have a hardware stack
- bank switching is not required to access RAM or special function registers
- data stored in program memory can be accessed directly using a feature called Program Space Visibility
- interrupt sources may be assigned to distinct handlers using an interrupt vector table

Some features are:

- hardware MAC (multiply-accumulate)
- barrel shifting
- bit reversal
- (16×16)-bit single-cycle multiplication and other DSP operations
- hardware divide assist (19 cycles for 16/32-bit divide)
- hardware support for loop indexing
- Direct memory access

Variants

Within a series, there are still many device variants depending on what hardware resources the chip features.

- General purpose I/O pins.
- Internal clock oscillators.
- 8/16/32 Bit Timers.
- Internal EEPROM Memory.

- Synchronous/Asynchronous Serial Interface USART.
- MSSP Peripheral for I²C and SPI Communications.
- Capture/Compare and PWM modules.
- Analog-to-digital converters (up to ~1.0 MHz).
- USB, Ethernet, CAN interfacing support.
- External memory interface.
- Integrated analog RF front ends (PIC16F639, and rfPIC).
- KEELOQ Rolling code encryption peripheral (encode/decode)
- And many more.

Trends

The first generation of PICs with EPROM storage are almost completely replaced by chips with Flash memory. Likewise, the original 12-bit instruction set of the PIC1650 and its direct descendants has been superseded by 14-bit and 16-bit instruction sets. Microchip still sells OTP (one-time-programmable) and windowed (UV-erasable) versions of some of its EPROM based PICs for legacy support or volume orders. The Microchip website lists PICs that are not electrically erasable as OTP despite the fact that UV erasable windowed versions of these chips can be ordered.

Development tools

Commercially supported

Microchip provides a freeware IDE package called MPLAB, which includes an assembler, linker, software simulator, and debugger. They also sell C compilers for the PIC18 and dsPIC which integrate cleanly with MPLAB. Free student versions of the C compilers are also available with all features. But for the free versions, optimizations will be disabled after 60 days.[10]

Several third parties make C, BASIC and Pascal language compilers for PICs, many of which integrate to MPLAB and/or feature their own IDE. A fully featured compiler for the PICBASIC language to program PIC microcontrollers is available from meLabs, Inc.

A blockset for Matlab/Simulink allows one to generate C and binary files from a Simulink model. Most common peripherals have a blockset and you do not need to write the configuration code.

8/16/32-bit PIC microcontroller product families

These links take you to product selection matrices at the manufacturer's site.

8-bit microcontrollers

- PIC10
- PIC12
- PIC14
- PIC16
- PIC17
- PIC18

16-bit microcontrollers

- PIC24F
- PIC24H

32-bit microcontrollers

- PIC32

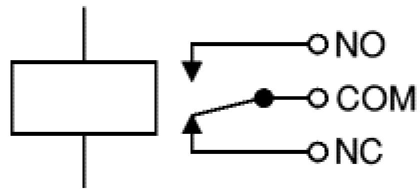
16-bit digital signal controllers

- dsPIC30
- dsPIC33F

The F in a name generally indicates the PICmicro uses flash memory and can be erased electronically. A C generally means it can only be erased by exposing the die to ultraviolet light (which is only possible if a windowed package style is used). An exception to this rule is the PIC16C84 which uses EEPROM and is therefore electrically erasable.

2.3 RELAY CIRCUIT

A relay is an **electrically operated switch**. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and most have **double throw** (**changeover**) switch contacts as shown in the diagram.



Circuit symbol for a relay

Fig 2.4 Relay Circuit

Relays allow one circuit to switch a second circuit which can be completely separate from the first. For example a low voltage battery circuit can use a relay to switch a 230V AC mains circuit. There is no electrical connection inside the relay between the two circuits, the link is magnetic and mechanical.

The coil of a relay passes a relatively large current, typically 30mA for a 12V relay, but it can be as much as 100mA for relays designed to operate from lower voltages. Most ICs (chips) cannot provide this current and a transistor is usually used to amplify the small IC current to the larger value required for the relay coil. The maximum output current for the popular 555 timer IC is 200mA so these devices can supply relay coils directly without amplification.

Relays are usuallly SPDT or DPDT but they can have many more sets of switch contacts, for example relays with 4 sets of changeover contacts are readily available.

The coil will be obvious and it may be connected either way round. Relay coils produce brief high voltage 'spikes' when they are switched off and this can destroy transistors and ICs in the circuit. To prevent damage you must connect a protection diode across the relay coil.

The relay's switch connections are usually labeled COM, NC and NO:

- **COM** = Common, always connect to this, it is the moving part of the switch.
- **NC** = Normally Closed, COM is connected to this when the relay coil is **off**.
- **NO** = Normally Open, COM is connected to this when the relay coil is **on**.
- Connect to COM and NO if you want the switched circuit to be **on when the relay coil is on**.
- Connect to COM and NC if you want the switched circuit to be **on when the relay coil is off**.

Choosing a relay

You need to consider several features when choosing a relay:

1. Physical size and pin arrangement

   If you are choosing a relay for an existing PCB you will need to ensure that its dimensions and pin arrangement are suitable. You should find this information in the supplier's catalogue.

2. Coil voltage

   The relay's coil voltage rating and resistance must suit the circuit powering the relay coil. Many relays have a coil rated for a 12V supply but

5V and 24V relays are also readily available. Some relays operate perfectly well with a supply voltage which is a little lower than their rated value.

3. Coil resistance

The circuit must be able to supply the current required by the relay coil. You can use Ohm's law to calculate the current:

$$\text{Relay coil current} = \frac{\text{supply voltage}}{\text{coil resistance}}$$

For example: A 12V supply relay with a coil resistance of 400 Ohm passes a current of 30mA. This is OK for a 555 timer IC (maximum output current 200mA), but it is too much for most ICs and they will require a transistor to amplify the current.

4. Switch ratings(voltage and current)

The relay's switch contacts must be suitable for the circuit they are to control. You will need to check the voltage and current ratings. Note that the voltage rating is usually higher for AC, for example: "5A at 24V DC or 125V AC".

5. Switch contact arrangement(SPDT,DPDT etc)

Most relays are SPDT or DPDT which are often described as "single pole changeover" (SPCO) or "double pole changeover" (DPCO).

Protection diodes for relays

Transistors and ICs must be protected from the brief high voltage produced when a relay coil is switched off. The diagram shows how a signal diode (eg 1N4148) is connected 'backwards' across the relay coil to provide this protection.

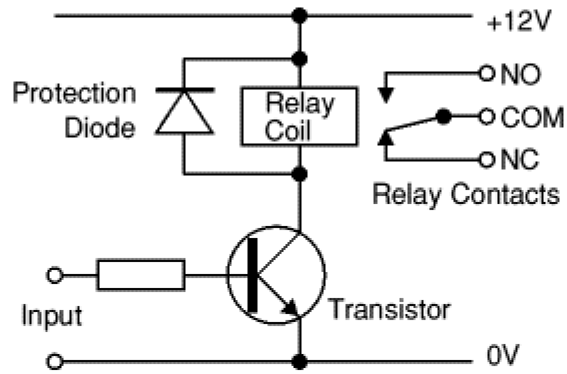


Fig 2.5 Protection diode of relays

Current flowing through a relay coil creates a magnetic field which collapses suddenly when the current is switched off. The sudden collapse of the magnetic field induces a brief high voltage across the relay coil which is very likely to damage transistors and ICs. The protection diode allows the induced voltage to drive a brief current through the coil (and diode) so the magnetic field dies away quickly rather than instantly. This prevents the induced voltage becoming high enough to cause damage to transistors and ICs.

Relays and transistors compared

Like relays, transistors can be used as an electrically operated switch. For switching small DC currents (< 1A) at low voltage they are usually a better choice than a relay. However, transistors cannot switch AC (such as mains electricity) and in simple circuits they are not usually a good choice for switching large currents (> 5A). In these cases a relay will be needed, but note that a low power transistor may still be needed to switch the current for the relay's coil! The main advantages and disadvantages of relays are listed below:

Advantages of relays:

- Relays can switch **AC and DC**, transistors can only switch DC.
- Relays can switch **higher voltages** than standard transistors.
- Relays are often a better choice for switching **large currents** (> 5A).
- Relays can switch **many contacts** at once.

Disadvantages of relays:

- Relays are **bulkier** than transistors for switching small currents.
- Relays **cannot switch rapidly** (except reed relays), transistors can switch many times per second.
- Relays **use more power** due to the current flowing through their coil.
- Relays **require more current than many ICs can provide**, so a low power transistor may be needed to switch the current for the relay's coil.

2.4 DC MOTOR

DESCRIPTION

DC MOTOR

DC motors are part of the electric motors using DC power as energy source. These devices transform electrical energy into mechanical energy. The basic principle of DC motors is same as electric motors in general, the magnetic interaction between the rotor and the stator that will generate spin.

Simple motor has six parts:

1. Armature or rotor
2. Commutator
3. Brushes
4. Axle
5. Field magnet
6. DC power supply of some sort

Working Principle of DC motor

Graphically, the working principles of DC motors can be seen in Figure 1 below.

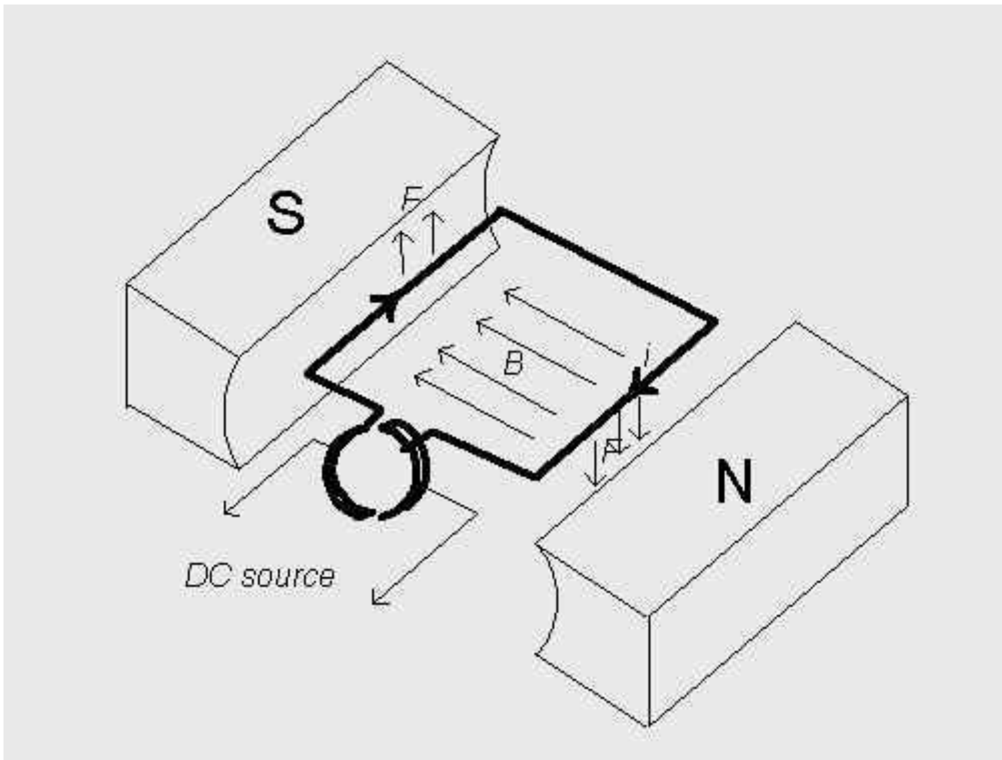


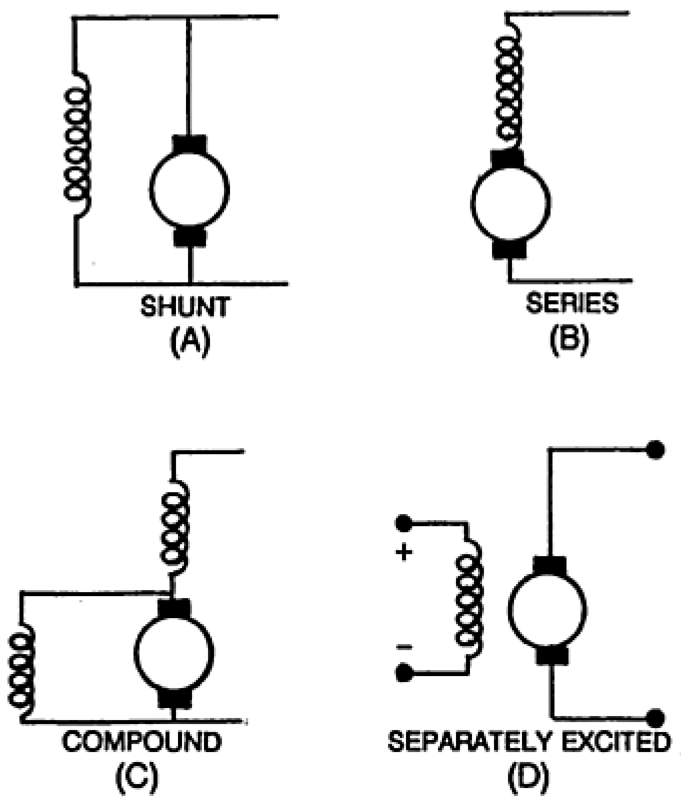
Figure 2.6. Principle of DC motor

When DC electric current flowing in the coil in accordance with the direction of the arrow, while the direction of the magnetic field B is from north to south pole, the coil will be driven by the force F in the direction as shown in Figure 1. This condition occurs continuously so will result in rotation on the axis of the coil. The direction of the electric current in the coil is fixed, because of the split ring on the end of the coil.

The major classes of DC motors are

- Shunt wound.
- Series wound.
- Compound wound.
- Separately excited.

These types of motors differ only in the connection of the field circuits The armatures, commutators, and so forth are nearly identical with each other and with those of the generators. All four major classes of motors are widely used. This is in contrast to the generators, in which the compound wound type is used for nearly all general power applications.



2.5 LCD DISPLAY

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs.

The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.

Pin Diagram:

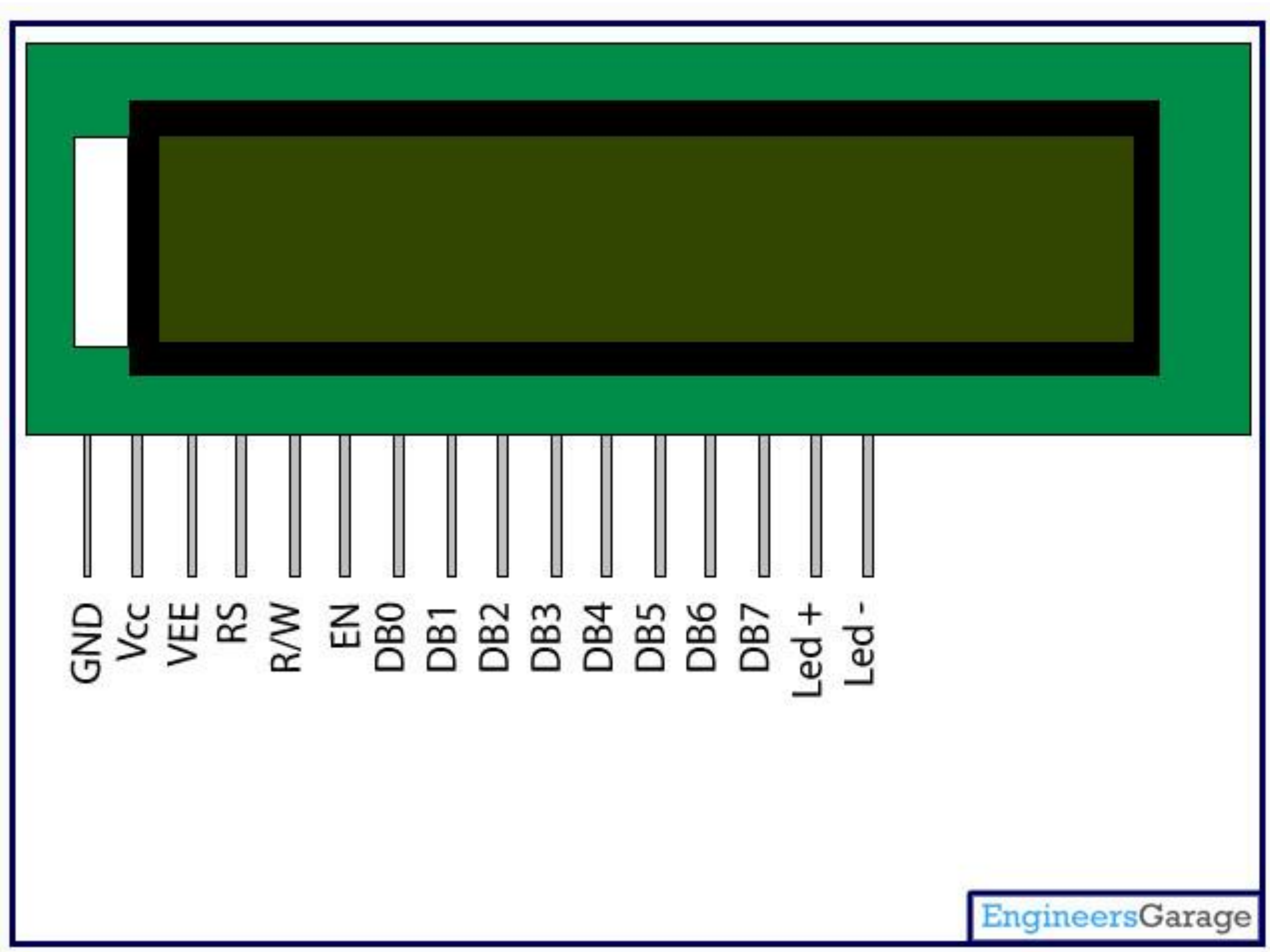


Fig 2.7 LCD display

pin configuration:

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

2.5.1 LCD INTERFACING WITH PIC 16F877

Programming PIC for Interfacing 16X2 LCD:

Only the pins, registers and architecture using for interfacing will be different. When we look at the program, functions like initialization, sending data to the LCD will be almost same.

In the pic programming also for initializing the LCD the R/W pin should be low for writing the data, Enable pins should be high and register select pin (RS) should be high for writing the data. For sending a command the RS should be low, R/W pin should be low and enable pin should be high.

Initializing the LCD function:

lcdcmd(0x38);//Configure the LCD in 8-bit mode,2 line and 5×7 font
lcdcmd(0x0C);// Display On and Cursor Off
lcdcmd(0x01);// Clear display screen
lcdcmd(0x06);// Increment cursor
lcdcmd(0x80);// Set cursor position to 1st line,1st column

Sending command to the LC:

- rs=0; Register select pin is low.
- rw=0; Read/write Pin is also for writing the command to the LCD.
- en=1;enable pin is high.

Sending data to the LCD:

- rs=1; Register select pin is high.
- rw=0; Read/write Pin is also for writing the command to the LCD.
- en=1; enable pin is high.

Steps for Programming:

- Install MPLAB in your system and create a new project, in selecting device and family select PIC18F family and add PIC18F4550 controller to your project.
- Select the compiler which you have installed and add the file to your project. After adding the file paste the code which is given below and run it. As it is a precompiled and tested program you will not find any errors.
- After compiling the program with no errors dump the program into your development board using PICKIT2 or PICKIT3 programmer/ debugger.
- If you are not using PICKIT then just compile the code and make the HEX file use this HEX file for programming the PIC microcontroller.

Program for Interfacing LCD to PIC16F77:

```
#define rs LATA.F0
#define rw LATA.F1
#define en LATA.F2
//LCD Data pins
#define lcdport LATB

void lcd_init();
void lcdcmd(unsigned char);
void lcddata(unsigned char);
unsigned char data[20]=”hello world”;
unsigned int i=0;

void main(void)
{
TRISA=0;                // Configure Port A as output port
LATA=0;
TRISB=0;                // Configure Port B as output port
LATB=0;
lcd_init();             // LCD initialization
while(data[i]!=’\0′)
{
```

```
lcddata(data[i]);     // Call lcddata function to send characters
// one by one from “data” array
i++;
Delay_ms(300);
}
}

void lcd_init()
{
lcdcmd(0x38);
lcdcmd (0x0C);
lcdcmd(0x01);
lcdcmd(0x06);
lcdcmd(0x80);

}

void lcdcmd(unsigned char cmdout)
{
lcdport=cmdout;
rs=0;
rw=0;
en=1;
Delay_ms(10);
en=0;
}

void lcddata(unsigned char dataout)
{
lcdport=dataout;
rs=1;
rw=0;
en=1;
Delay_ms(10);
en=0;
}
```

2.6 DRIVER CIRCUIT

Relay Driver:

A relay is an electro-magnetic switch which is useful if you want to use a low voltage circuit to switch on and off a light bulb (or anything else) connected to the 220v mains supply.

The diagram below shows a typical relay (with "normally-open" contacts).

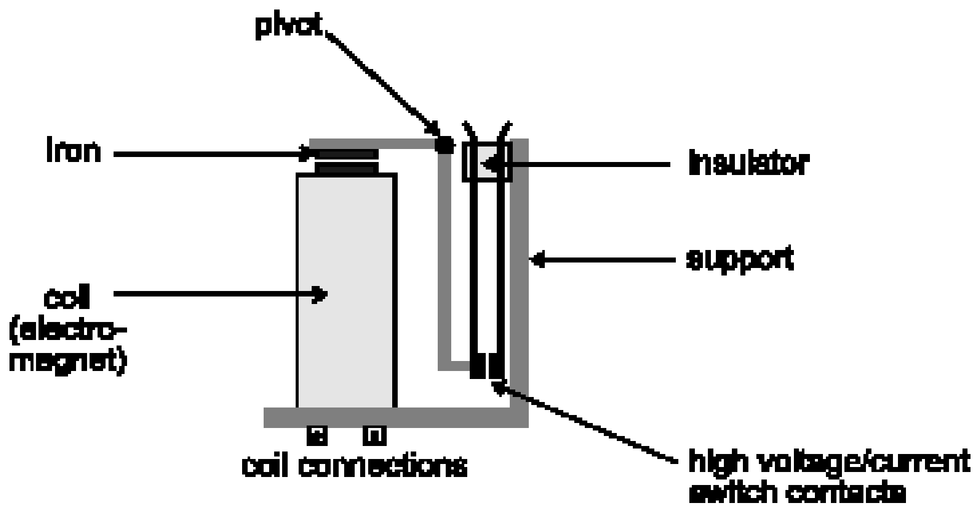


Fig 2.8 Relay circuit diagram

The current needed to operate the relay coil is more than can be supplied by most chips (op. amps etc), so a transistor is usually needed.

Driver using IC ULN 2803:

A ULN2803 is an Integrated Circuit (IC) chip with a High Voltage/High Current Darlington Transistor Array. It allows you to interface TTL signals with higher voltage/current loads. In English, the chip takes low level signals (TLL, CMOS, PMOS, NMOS - which operate at low voltages and low currents) and acts as a relay of sorts itself, switching on or off a higher level signal on the opposite side.

A TTL signal operates from 0-5V, with everything between 0.0 and 0.8V considered "low" or off, and 2.2 to 5.0V being considered "high" or on. The maximum power available on a TTL signal depends on the type, but generally does not exceed 25mW (~5mA @ 5V), so it is not useful for providing power to something like a relay coil. Computers and other electronic devices frequently generate TTL signals. On the output side the ULN2803 is generally rated at 50V/500mA, so if can operate small loads directly. Alternatively, it is frequently used to power the coil of one or more relays, which in turn allow even higher voltages/currents to be controlled by the low level signal. In electrical terms, the ULN2803 uses the low level (TTL) signal to switch on/turn off the higher voltage/current signal on the output side.

The ULN2803 comes in an 18-pin IC configuration and includes eight (8) transistors. Pins 1-8 receive the low level signals; pin 9 is grounded (for the low level signal reference). Pin 10 is the common on the high side and would generally be connected to the positive of the voltage you are applying to the relay coil. Pins 11-18 are the outputs (Pin 1 drives Pin 18, Pin 2 drives 17, etc.).

The ULN2803 is a small integrated circuit that contains 8 transistor driver channels. Each channel has an input to a resistor connected to the base of a transistor and a 1 amp open collector output capable of handling up to about 30volts .Each of the collectors has a reverse biased diode connected to a common Vcc pin that provides inductive spike protection.

Typical uses are for micro-processor interfaces to relays, lamps, solenoids and small motors. A 2803 with a set of relays is a simple and effective way of switching mains voltages for example. They are used less commonly today but were once an almost universal means of interfacing processors to power devices.

Driver Features

1. TTL, DTL, PMOS, or CMOS Compatible Inputs
2. Output Current to 500 mA
3. Output Voltage to 95 V
4. Transient-Protected Outputs
5. Dual In-Line Package or Wide-Body Small-Outline Package

Logical Diagram:

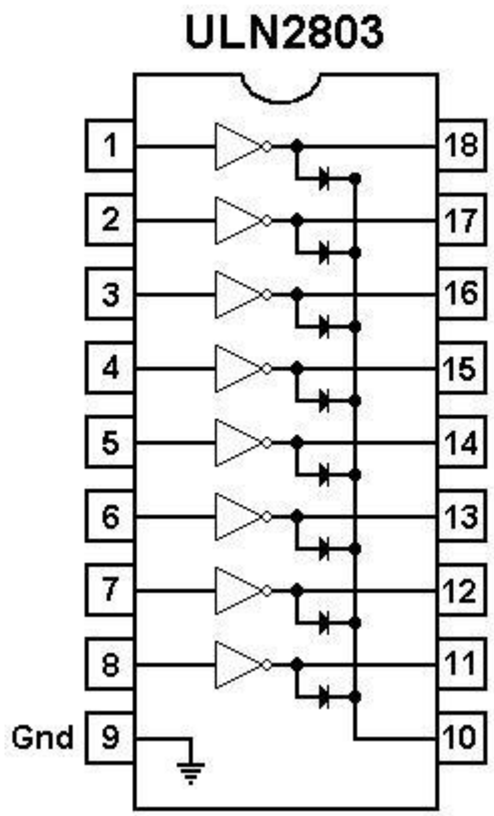


Fig 2.9 Relay Logic diagram

Specification:

Output Voltage, VCE .... **50 V**

Input Voltage, VIN ....**30 V**

Continuous Output Current, IC .... **500 mA**

Continuous Input Current, IIN .... **25 mA**

Power Dissipation, PD

(One Darlington pair) .... **1.0 W**

(Total package) .... **> 2.0 W**

CHAPTER 3

SOFTWARE DESCRIPTION

3.1 MPLAB IDE

DESCRIPTON

Integrated Development Environment (IDE) is an application that has multiple functions for software development. MPLAB IDE an executable program that integrates a compiler, an assembler, a project manager, an editor, a debugger, simulator, and an assortment of other tools within one Windows application. A user developing an application should be able to write code, compile, debug and test and application without leaving the MPLAB IDE desktop.

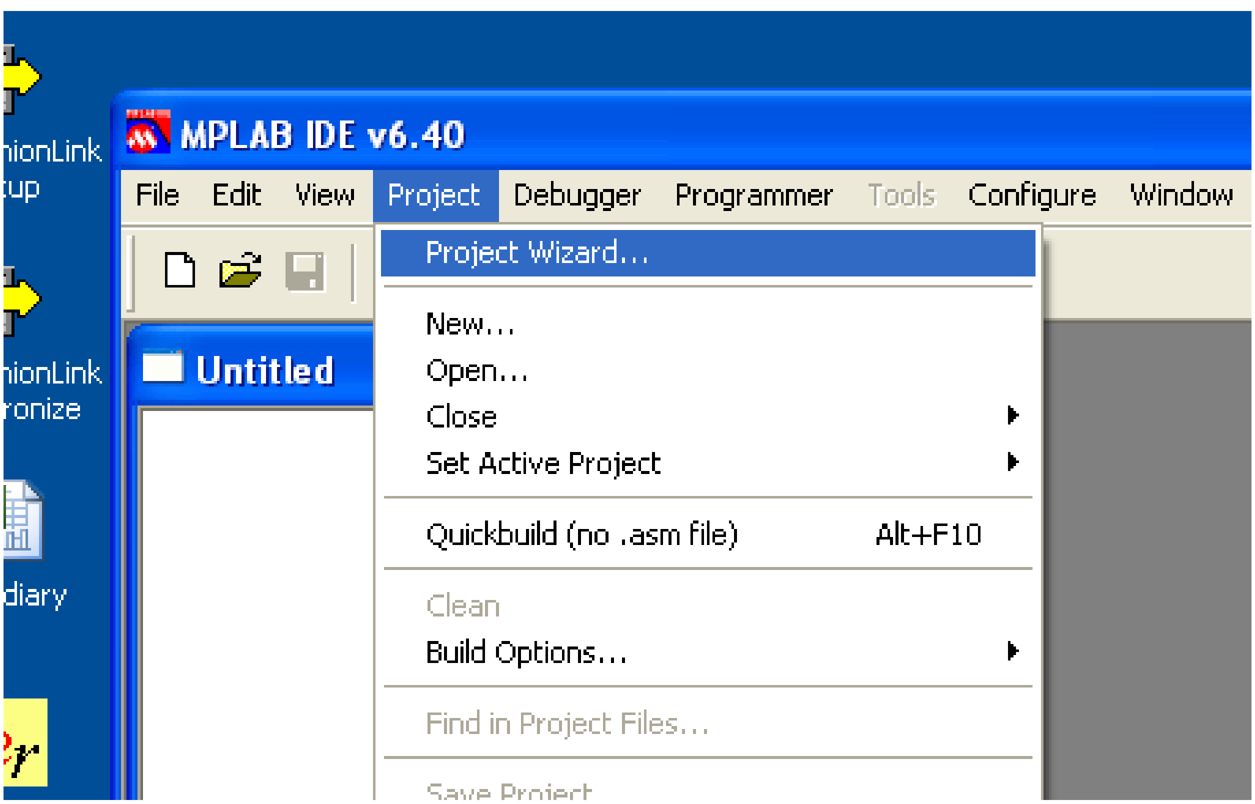
Bundled with MPLAB IDE software are several code generation tools. MPASM™ assembler is a full-featured universal macro assembler for all PICmicro MCUs. It can produce absolute code directly in the form of HEX files for device programmers, or it can generate relocatable objects for MPLINK™ linker. MPLINK™ linker links relocatable objects from assembly or C source files along with pre-compiled libraries using directives from a linker script, and supports MPASM™, MPLABC17, and MPLABC18. MPLIB™ librarian ia a librarian for pre-compiled code to be used with MPLINK™ linker.

When a routine from a library is called from another souce file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. The latest versions of MPASM™, MPLINK™, and MPLIB™ are bundled with MPLAB and can be downloaded separately by choosing the web install and selecting only the MPASM™, MPLINK™, and/or MPLIB™

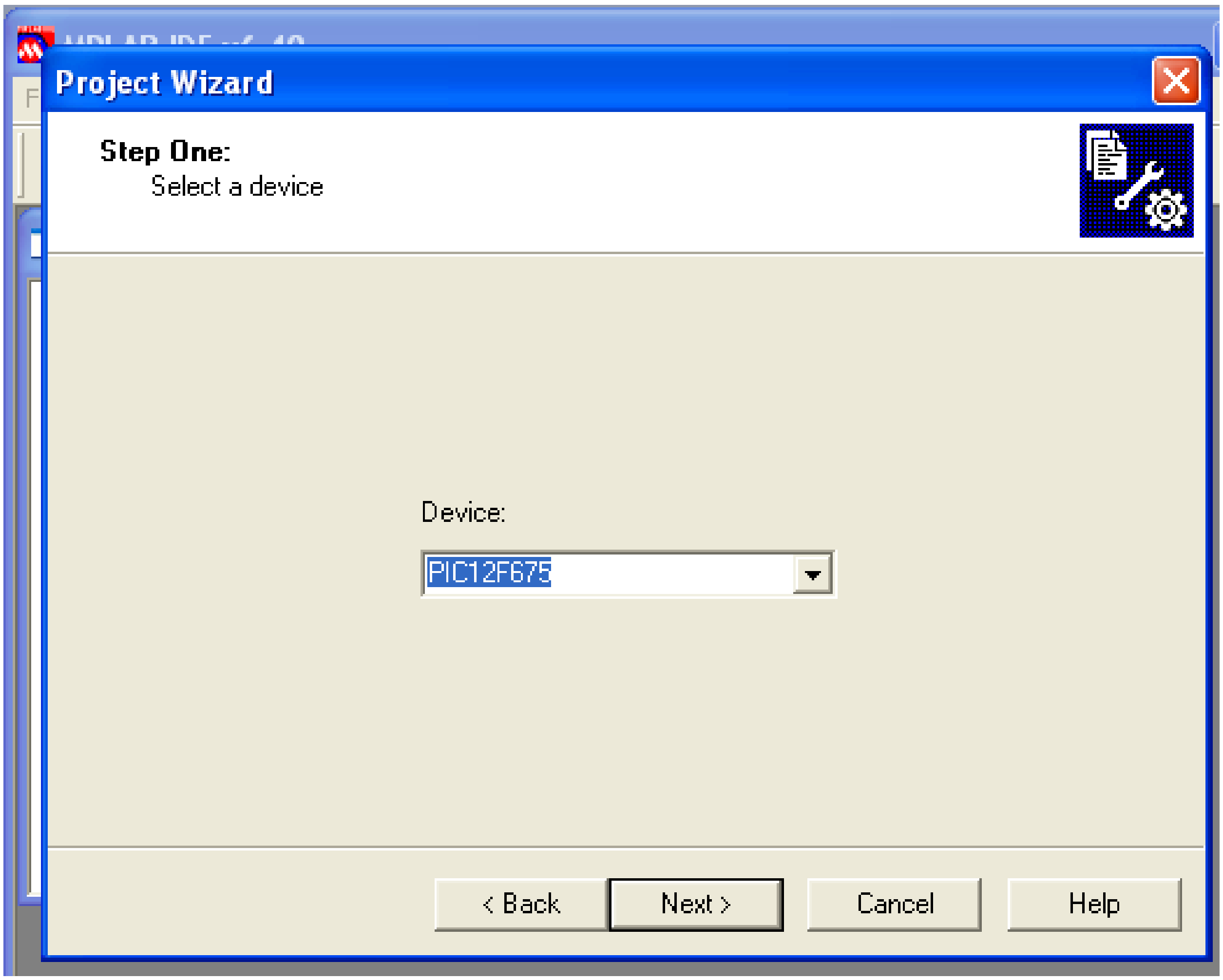
HOW TO USE

- create a project
- enter program
- assemble program
- down load program
- run program
- simulate

STEP 1: Create New project

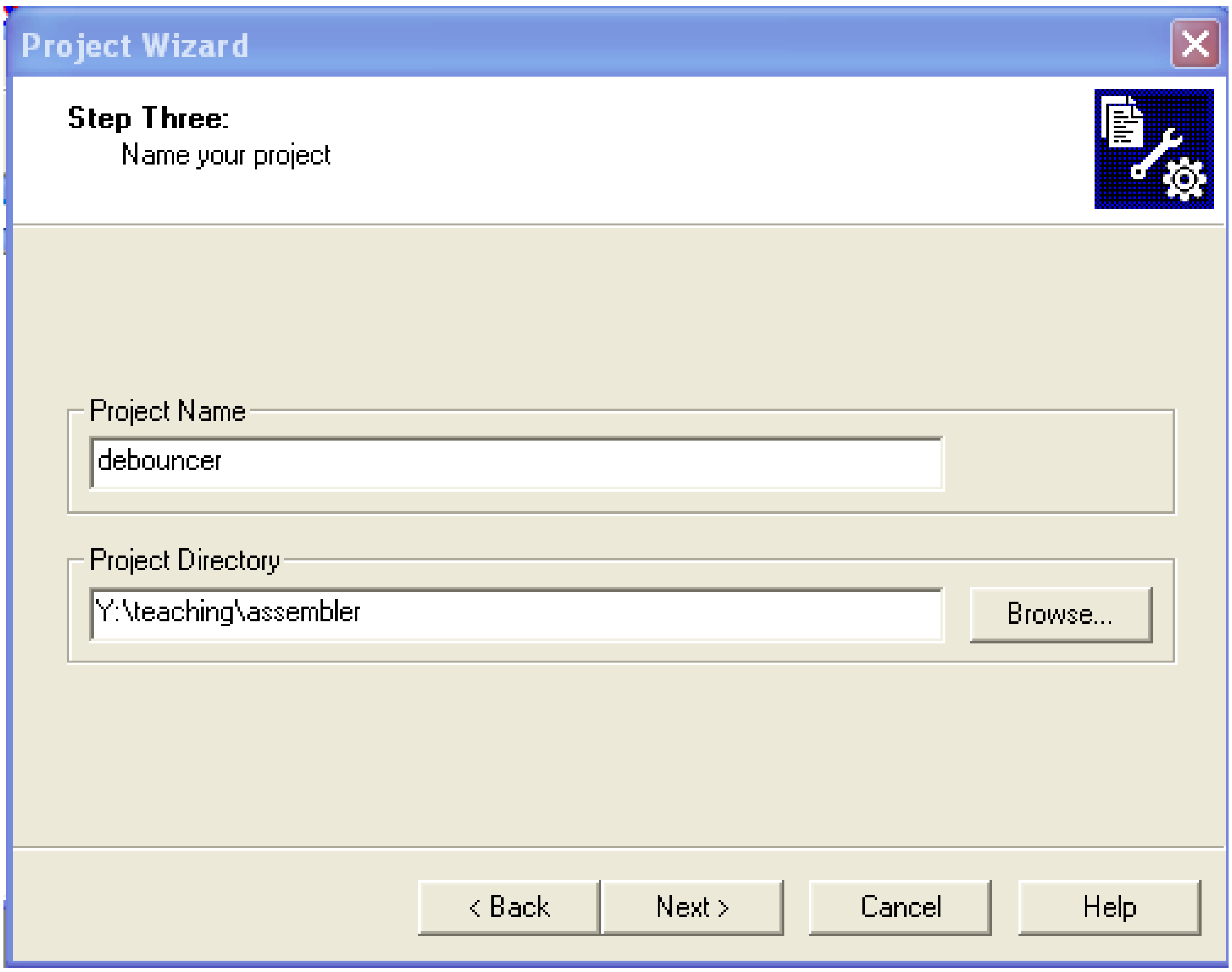


STEP 2 :SELECTING PIC TYPE



.

STEP 3: Name your project



STEP 5: TYPE THE PROGRAM

- Now click on File->New to get a new blank file
- Type in your program.

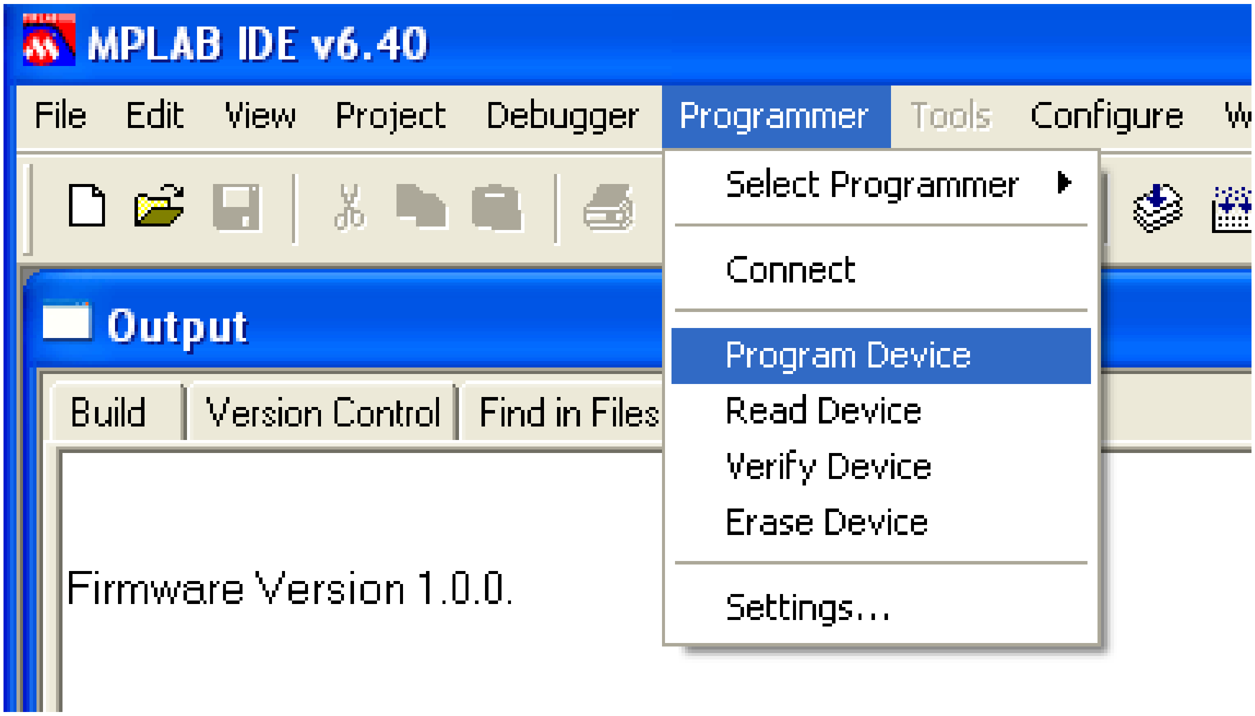
Step 6: Save the file:

- Once you have entered your program save it with the extension .asm
- At this point the syntax highlighting will switch on.

Step 7:Run the assembler

- Goto the project menu
- Select Build and correct the errors

Step 7: Program the chip



Step 8:Test on board

Test on the board by pressing the rest button

CHAPTER 4

4.1 ALGORITHM FOR VOLUME CONTROL:

Step 1: Start

Step2:Get The Input fromMIC

Step3:Display the Sound level in LCD in ASCII Digits

Step4:Compare the noise level and input threshold sound level.

Step5:Based on increasing level of the Noise, Delays Increased

Step6:Stop

4.1 PROGRAM FOR VOLUME CONTROL:

```
#include<pic.h>

__CONFIG(HS & WDTDIS & PWRTEN & BOREN & UNPROTECT &
LVPDIS);

#define rs RE0

#define rw RE1

#define en RE2

unsigned char Mic_Input1_3,Mic_Input1_2,Mic_Input1_1;

unsigned char Flag1=0,Flag2=0,Flag3=0;

unsigned int Mic_Input;

bit b;

void delay(unsigned int y)

{

while(y--);
```

```
}
void lcd_command(unsigned char com)
{
PORTD=com;
en=1;
rw=0;
rs=0;
delay(250);
en=0;
delay(250);
}
void lcd_data(unsigned char dat)
{
PORTD=dat;
en=1;
rw=0;
rs=1;
delay(250);
en=0;
delay(250);
}
void lcd_condis(const unsigned char *word, unsigned int n)
```

```
{
unsigned char i;
for(i=0;i<n;i++)
{
lcd_data(word[i]);
}
}
void lcd_init()
{
lcd_command(0x38);
lcd_command(0x06);
lcd_command(0x0c);
lcd_command(0x01);
lcd_command(0x80);
}
int hex_ascii1()
{
unsigned int a,b,c,d,e,temp;
temp =ADRESH<<8;
temp =(int)((temp+ADRESL)/2.044);
a=temp%10;
b=temp/10;
```

```
c=b%10;
d=b/10;
e=d%10;
lcd_data(e+0x30);
Mic_Input1_3=(e+0x30);
lcd_data(c+0x30);
Mic_Input1_2=(c+0x30);
lcd_data(a+0x30);
Mic_Input1_1=(a+0x30);
return temp;
}
void scan_5channels()
{
CHS2=0;
CHS1=0;
CHS0=0;
ADON=1;
delay(200);
ADCON0=(ADCON0|0x04);
delay(200);
lcd_command(0x0c0);
Mic_Input=hex_ascii1();
```

```
}
void main()
{
ADCON1=0x84;
ADCON0=0x00;
TRISA=0xff;
TRISB=0xff;
TRISC=0x00;
TRISD=0x00;
TRISE=0x00;
PORTA=0xff;
PORTB=0xff;
PORTC=0x00;
PORTD=0x00;
PORTE=0x00;
lcd_init();
delay(100);
RBPU=0;
lcd_command(0x80);
lcd_condis("Sound Optimizer ",16);
lcd_command(0x0c0);
lcd_condis("                ",16);
```

```
delay(60000);
lcd_command(0x80);
lcd_condis("Home position...",16);
lcd_command(0x0c0);
lcd_condis("                ",16);
delay(60000);
do
{
RC0=1;
RC1=0;
}while(RB0==1);
RC0=0;
RC1=0;
while(1)
{
scan_5channels();
scan_5channels();
if(Mic_Input>485)
{
do
{
RC0=1;
```

```
RC1=0;
}while(RB0==1);
RC0=0;
RC1=0;
Flag1=0;
Flag2=0;
Flag3=0;
}
else if((Mic_Input<391)&&(Mic_Input>340)&&(Flag1==0))
{
RC1=1;
delay(60000);
delay(30000);
RC1=0;
Flag1=1;
Flag2=0;
Flag3=0;
}
else if((Mic_Input<340)&&(Mic_Input>281)&&(Flag2==0))
{
RC1=1;
delay(60000);
```

```
delay(60000);
RC1=0;
Flag1=0;
Flag2=1;
Flag3=0;
}
else if((Mic_Input<281)&&(Mic_Input>180)&&(Flag3==0))
{
RC1=1;
delay(60000);
delay(60000);
delay(30000);
RC1=0;
Flag1=0;
Flag2=0;
Flag3=1;
}
}
}
```

CHAPTER 5

APPLICATIONS:

- Travel Bus:

  Our project can be used in the audio systems of the buses and vehicles to automatcally adjust the Sound level of systems and eliminates the noise disturbances and the device can be placed in the edge of the windows of the vehicles.

- Public Meetings:

  The public speech meetings usually addressed in a large public gathering where noise acts as a major disturbance but with use of our project the noise disturbances can be eliminated by the placing the sound optimizer in the bottom of the seats.

- Cinema Theaters:

  The Cinema thetre is the main application of the sound optimizer.The noises or applauds from the audiences may cause disturbance in the cinema.According to seats arrangement in theatres,sound optimizer placed at certain space from the screen near to the seats

- Houses in traffic areas

  The audio or tv systems in houses located in the highly populated cities may experience a noise disturbances from the outside environments such as the horns,Vehicles sounds etc..,that can be eliminated by the use of sound optimiser in the place of the edges of roofs ,windows and doors.

CONCLUSION:

Our project has been successfully completed to analyze the external noise and adjust the level of the internal sound volume.Eventhough there are certain limitations in the project such as placing the sound optimiser at some higher distance from the sound source ,But the project serves as a automated system to eliminate the use of the manual volume control.Through this project which can be served as a massive application in the areas of the sound systems and cinema theatres.

Though there are several sound adaptive systems in the home theatre systems for the automatic adaption of the sound level according to the noise disturbances.But this project will serve as cheap alternative to the sound optimisers .

BIBILOGRAPHY:

1.www.beyondlogic.com

2.pic 16f877 data sheet

3.www.wikipedia.org

4.www.howstuffworks.com

5.www.alldatasheets.com

6.www.electronicshub.org