PLEASE NOTE: THE MATTER PRODUCED BELOW IS REFERENCE MATERIAL ONLY FOR PREPARING THE PROJECT REPORT. IT IS NOT THE COMPLETE PROJECT REPORT.

A PROJECT REPORT ON

[**Obstacle Avoider Robotic Vehicle**](http://nevonprojects.com/obstacle-avoider-robotic-vehicle-project/)

Submitted in partial fulfillment of the requirements

For the award of the degree

BACHELOR OF ENGINEERING

IN

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ENGINEERING

SUBMITTED BY

-------------------- (--------------)

--------------------- (---------------)

--------------------- (---------------)

DEPARTMENT OF \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ENGINEERING

\_\_\_\_\_\_\_\_\_\_COLLEGE OF ENGINEERING

AFFILIATED TO \_\_\_\_\_\_\_\_\_\_\_ UNIVERSITY

**CERTIFICATE**

This is to certify that the dissertation work entitled **Obstacle Avoider Robotic Vehicle** is the work done by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ submitted in partial fulfillment for the award of ‘BACHELOR OF ENGINEERING (B.E)’ in Electronics and Communication Engineering from\_\_\_\_\_\_\_ College of Engineering affiliated to \_\_\_\_\_\_\_\_\_ University , Hyderabad .

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_**

**(Head of the department, ECE) (Assistant Professor)**

**EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mentioning of the people whose constant guidance and encouragement made it possible. We take pleasure in presenting before you, our project, which is result of studied blend of both research and knowledge.

We express our earnest gratitude to our internal guide, Assistant Professor \_\_\_\_\_\_\_\_\_\_\_\_\_\_, Department of ECE, our project guide, for his constant support, encouragement and guidance. We are grateful for his cooperation and his valuable suggestions.

Finally, we express our gratitude to all other members who are involved either directly or indirectly for the completion of this project.

**DECLARATION**

We, the undersigned, declare that the project entitled **‘OBSTACLE AVOIDANCE ROBOT’**, being submitted in partial fulfillment for the award of Bachelor of Engineering Degree in Electronics and Communication Engineering, affiliated to \_\_\_\_\_\_\_\_\_ University, is the work carried out by us.

**ABSTRACT**

Our proposed project puts forward an obstacle avoider robotic vehicle that uses ultrasonic sensors for this purpose. The system uses an 8051 family microprocessor to achieve this functionality.

The robotic vehicle is designed to first track and avoid any kind of obstacles that comes it’s way. The vehicle achieves this smart functionality with the help of ultrasonic sensors coupled with an 8051 microprocessor and motors. The entire system combined gives the vehicle an intelligent object detection and obstacle avoidance scheme.

This system allows the vehicle to guide itself in case it encounters any obstacle. The obstacle detection is done using the ultrasonic sensor. This is detected and a signal is passed on to the 8051 microcontroller.  
On receiving the signal it guides the vehicle in another direction by actuating the motors through the motor driver IC.

**INTRODUCTION TO EMBEDDED SYSTEMS**

**What is embedded system?**

An Embedded System is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function. An embedded system is a microcontroller-based, software driven, reliable, real-time control system, autonomous, or human or network interactive, operating on diverse physical variables and in diverse environments and sold into a competitive and cost conscious market.

An embedded system is not a computer system that is used primarily for processing, not a software system on PC or UNIX, not a traditional business or scientific application. High-end embedded & lower end embedded systems. High-end embedded system - Generally 32, 64 Bit Controllers used with OS. Examples Personal Digital Assistant and Mobile phones etc .Lower end embedded systems - Generally 8,16 Bit Controllers used with an minimal operating systems and hardware layout designed for the specific purpose.

**SYSTEM DESIGN CALLS:**

Figure 2(a): Embedded system design calls

**EMBEDDED SYSTEM DESIGN CYCLE**

Figure:2(b) “V Diagram”

**Characteristics of Embedded System**

* An embedded system is any computer system hidden inside a product other than a computer.
* They will encounter a number of difficulties when writing embedded system software in addition to those we encounter when we write applications.
  + Throughput – Our system may need to handle a lot of data in a short period of time.
  + Response–Our system may need to react to events quickly
  + Testability–Setting up equipment to test embedded software can be difficult.
  + Debugability–Without a screen or a keyboard, finding out what the software is doing wrong (other than not working) is a troublesome problem.
  + Reliability – embedded systems must be able to handle any situation without human intervention.
  + Memory space – Memory is limited on embedded systems, and you must make the software and the data fit into whatever memory exists.
  + Program installation – you will need special tools to get your software into embedded systems.
  + Power consumption – Portable systems must run on battery power, and the software in these systems must conserve power.
  + Processor hogs – computing that requires large amounts of CPU time can complicate the response problem.
  + Cost – Reducing the cost of the hardware is a concern in many embedded system projects; software often operates on hardware that is barely adequate for the job.
* Embedded systems have a microprocessor/ microcontroller and a memory. Some have a serial port or a network connection. They usually do not have keyboards, screens or disk drives.

**APPLICATIONS**

1. Military and aerospace embedded software applications
2. Communication Applications
3. Industrial automation and process control software
4. Mastering the complexity of applications.
5. Reduction of product design time.
6. Real time processing of ever increasing amounts of data.
7. Intelligent, autonomous sensors.

**CLASSIFICATION**

* Real Time Systems.
* RTS is one which has to respond to events within a specified deadline.
* A right answer after the dead line is a wrong answer.

**RTS CLASSIFICATION**

* Hard Real Time Systems
* Soft Real Time System

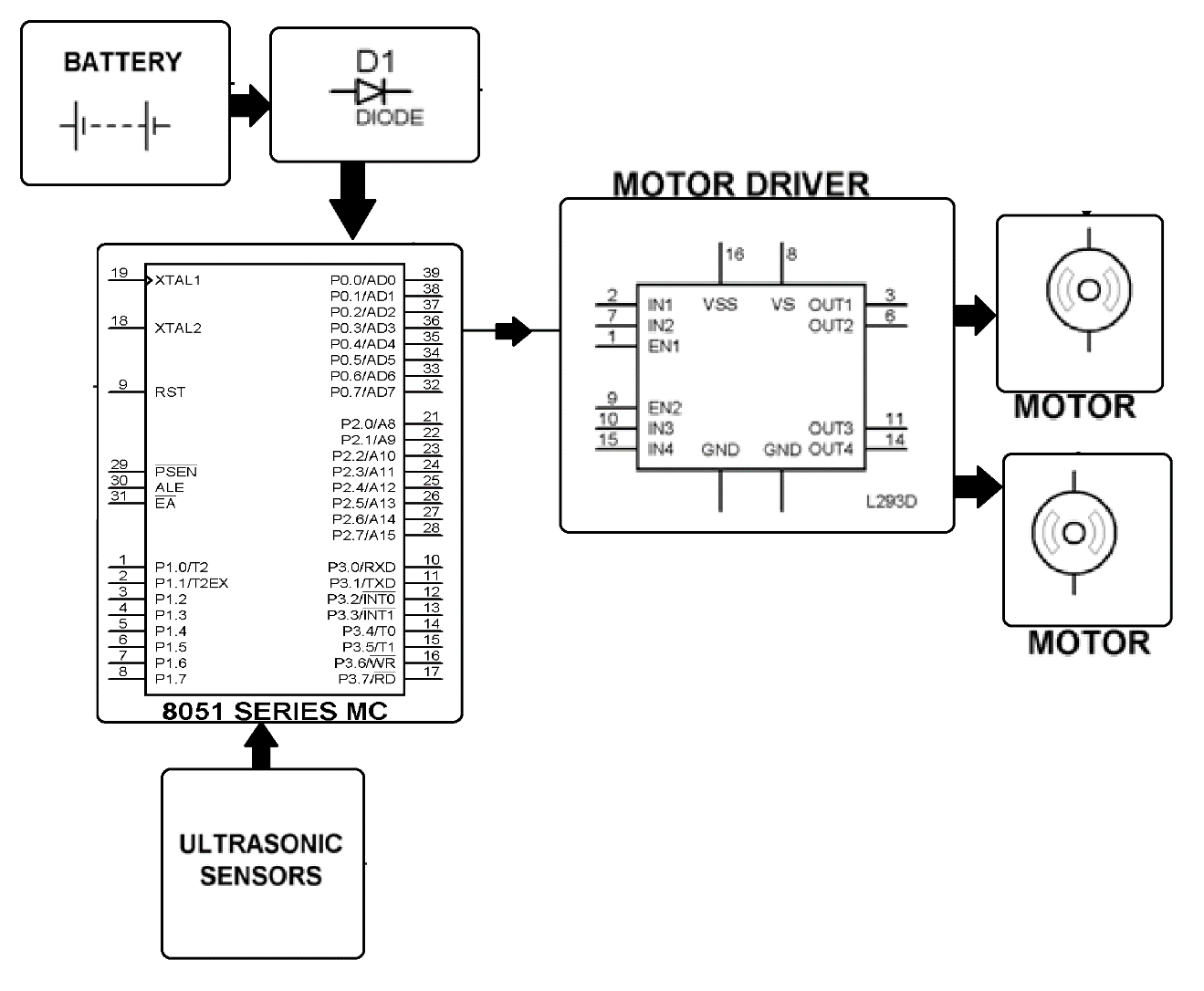
**HARD REAL TIME SYSTEM**

* "Hard" real-time systems have very narrow response time.
* Example: Nuclear power system, Cardiac pacemaker.

**SOFT REAL TIME SYSTEM**

* "Soft" real-time systems have reduced constrains on "lateness" but still must operate very quickly and repeatable.
* Example: Railway reservation system – takes a few extra seconds the data remains valid.

**3 PROJECT BLOCK DIAGRAM**



**4 HARDWARE REQUIREMENTS**

**HARDWARE COMPONENTS:**

1. BATTERY
2. MICROCONTROLLER (AT89S52/AT89C51)
3. PUSH BUTTONS
4. DC MOTOR
5. L293D
6. ULTRASONIC SENSOR PAIR
7. LED
8. 1N4007
9. RESISTORS
10. CAPACITORS

**4.1 BATTERY**

An electrical battery is a combination of one or more electrochemical cells, used to convert stored chemical energy into electrical energy. The battery has become a common power source for many household and industrial applications.

Batteries may be used once and discarded, or recharged for years as in standby power applications. Miniature cells are used to power devices such as hearing aids and wristwatches; larger batteries provide standby power for telephone exchanges or computer data centers.

**WORKING OF BATTERY:**

A battery is a device that converts chemical energy directly to electrical energy. It consists of a number of voltaic cells; each voltaic cell consists of two half cells connected in series by a conductive electrolyte containing anions and cat ions. One half-cell includes electrolyte and the electrode to which anions (negatively-charged ions) migrate, i.e. the anode or negative electrode; the other half-cell includes electrolyte and the electrode to which cat ions (positively-charged ions) migrate, i.e. the cathode or positive electrode. In the red ox reaction that powers the battery, reduction (addition of electrons) occurs to cat ions at the cathode, while oxidation (removal of electrons) occurs to anions at the anode. The electrodes do not touch each other but are electrically connected by the electrolyte. Many cells use two half-cells with different electrolytes. In that case each half-cell is enclosed in a container, and a separator that is porous to ions but not the bulk of the electrolytes prevents mixing.

Each half cell has an electromotive force (or emf), determined by its ability to drive electric current from the interior to the exterior of the cell. The net emf of the cell is the difference between the emfs of its half-cells. Therefore, if the electrodes have emfs and, in other words, the net emf is the difference between the reduction potentials of the half-reactions.

The electrical driving force or across the terminals of a cell is known as the terminal voltage (difference) and is measured in volts. The terminal voltage of a cell that is neither charging nor discharging is called the open-circuit voltage and equals the emf of the cell. Because of internal resistance, the terminal voltage of a cell that is discharging is smaller in magnitude than the open-circuit voltage and the terminal voltage of a cell that is charging exceeds the open-circuit voltage. An ideal cell has negligible internal resistance, so it would maintain a constant terminal voltage of until exhausted, then dropping to zero. If such a cell maintained 1.5 volts and stored a charge of one Coulomb then on complete discharge it would perform 1.5 Joule of work. In actual cells, the internal resistance increases under discharge, and the open circuit voltage also decreases under discharge. If the voltage and resistance are plotted against time, the resulting graphs typically are a curve; the shape of the curve varies according to the chemistry and internal arrangement employed.

An electrical **battery** is one or more [electrochemical cells](http://en.wikipedia.org/wiki/Electrochemical_cell) that convert stored chemical [energy](http://en.wikipedia.org/wiki/Energy) into electrical energy. Since the invention of the first battery (or "[voltaic pile](http://en.wikipedia.org/wiki/Voltaic_pile)") in 1800 by [Alessandro Volta](http://en.wikipedia.org/wiki/Alessandro_Volta), batteries have become a common power source for many household and industrial applications. According to a 2005 estimate, the worldwide battery industry generates [US$](http://en.wikipedia.org/wiki/United_States_dollar)48 [billion](http://en.wikipedia.org/wiki/1000000000_%28number%29) in sales each year, with 6% annual growth. There are two types of batteries: [primary batteries](http://en.wikipedia.org/wiki/Primary_battery) (disposable batteries), which are designed to be used once and discarded, and [secondary batteries](http://en.wikipedia.org/wiki/Secondary_battery) (rechargeable batteries), which are designed to be recharged and used multiple times. Miniature cells are used to power devices such as hearing aids and wristwatches; larger batteries provide standby power for telephone exchanges or computer data centers.

## Principle of operation

A battery is a device that converts chemical energy directly to electrical energy. It consists of a number of voltaic cells; each voltaic cell consists of two [half cells](http://en.wikipedia.org/wiki/Half_cell) connected in series by a conductive electrolyte containing anions and cations. One half-cell includes electrolyte and the electrode to which [anions](http://en.wikipedia.org/wiki/Ion#Ions) (negatively charged ions) migrate, i.e., the [anode](http://en.wikipedia.org/wiki/Anode) or negative electrode; the other half-cell includes electrolyte and the electrode to which [cations](http://en.wikipedia.org/wiki/Ion#Ions) (positively charged ions) migrate, i.e., the [cathode](http://en.wikipedia.org/wiki/Cathode) or positive electrode. In the [redox](http://en.wikipedia.org/wiki/Redox) reaction that powers the battery, cations are reduced (electrons are added) at the cathode, while anions are oxidized (electrons are removed) at the anode. The electrodes do not touch each other but are electrically connected by the [electrolyte](http://en.wikipedia.org/wiki/Electrolyte). Some cells use two half-cells with different electrolytes. A separator between half cells allows ions to flow, but prevents mixing of the electrolytes.

Each half cell has an electromotive force (or emf), determined by its ability to drive electric current from the interior to the exterior of the cell. The net emf of the cell is the difference between the emfs of its half-cells, as first recognized by Volta. Therefore, if the electrodes have emfs \mathcal{E}_1and \mathcal{E}_2, then the net emf is \mathcal{E}_{2}-\mathcal{E}_{1}; in other words, the net emf is the difference between the [reduction potentials](http://en.wikipedia.org/wiki/Reduction_potential) of the [half-reactions](http://en.wikipedia.org/wiki/Half-reaction). The electrical driving force or \displaystyle{\Delta V_{bat}}across the terminals of a cell is known as the *terminal voltage (difference)* and is measured in [volts](http://en.wikipedia.org/wiki/Volt). The terminal voltage of a cell that is neither charging nor discharging is called the [open-circuit voltage](http://en.wikipedia.org/wiki/Open-circuit_voltage) and equals the emf of the cell. Because of internal resistance, the terminal voltage of a cell that is discharging is smaller in magnitude than the open-circuit voltage and the terminal voltage of a cell that is charging exceeds the open-circuit voltage. An ideal cell has negligible internal resistance, so it would maintain a constant terminal voltage of \mathcal{E}until exhausted, then dropping to zero. If such a cell maintained 1.5 volts and stored a charge of one [coulomb](http://en.wikipedia.org/wiki/Coulomb) then on complete discharge it would perform 1.5 [joule](http://en.wikipedia.org/wiki/Joule) of work. In actual cells, the internal resistance increases under discharge, and the open circuit voltage also decreases under discharge. If the voltage and resistance are plotted against time, the resulting graphs typically are a curve; the shape of the curve varies according to the chemistry and internal arrangement employed.

As stated above, the voltage developed across a cell's terminals depends on the energy release of the chemical reactions of its electrodes and electrolyte. Alkaline and [carbon-zinc](http://en.wikipedia.org/wiki/Zinc-carbon_battery) cells have different chemistries but approximately the same emf of 1.5 volts; likewise [NiCd](http://en.wikipedia.org/wiki/Nickel-cadmium_battery) and [NiMH](http://en.wikipedia.org/wiki/Nickel-metal_hydride_battery) cells have different chemistries, but approximately the same emf of 1.2 volts. On the other hand the high electrochemical potential changes in the reactions of [lithium](http://en.wikipedia.org/wiki/Lithium) compounds give lithium cells emfs of 3 volts or more.

**4.2 MICROCONTROLLER AT89S52**

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel’s high-density non volatile memory technology and is compatible with the industry standard 80C51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional non volatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications. The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

**Features:**

• Compatible with MCS®-51 Products

• 8K Bytes of In-System Programmable (ISP) Flash Memory

– Endurance: 10,000 Write/Erase Cycles

• 4.0V to 5.5V Operating Range

• Fully Static Operation: 0 Hz to 33 MHz

• Three-level Program Memory Lock

• 256 x 8-bit Internal RAM

• 32 Programmable I/O Lines

• Three 16-bit Timer/Counters

• Eight Interrupt Sources

• Full Duplex UART Serial Channel

• Low-power Idle and Power-down Modes

• Interrupt Recovery from Power-down Mode

• Watchdog Timer

• Dual Data Pointer

• Power-off Flag

• Fast Programming Time

• Flexible ISP Programming (Byte and Page Mode)

• Green (Pb/Halide-free) Packaging Option

**Block Diagram of AT89S52:**

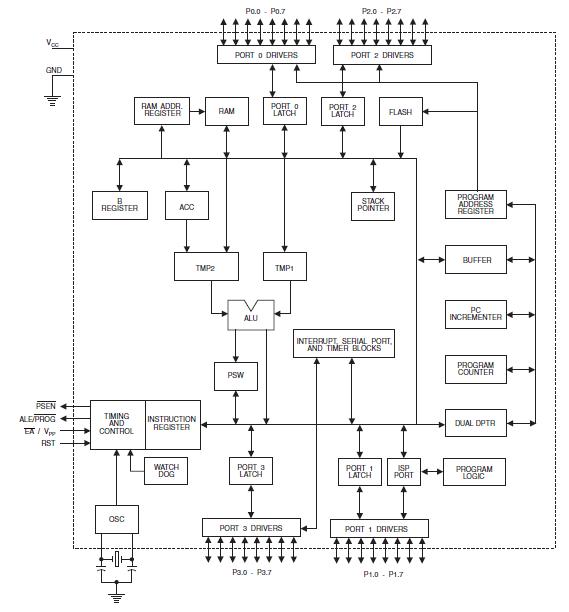
****

Fig 4.5(a): Block Diagram Of AT89S52

**Pin Configurations of AT89S52**

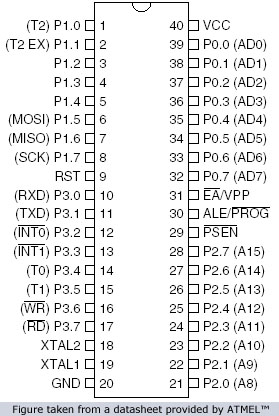


FIG 4.5(b): PIN DIAGRAM OF AT89S52

**Pin Description:**

**VCC:**

Supply voltage.

**GND:**

Ground

**Port 0:**

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.

**Port 1:**

Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX).

**Port 2:**

Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that uses 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

**Port 3:**

Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull-ups.

**RST:**

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives high for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

**ALE/PROG:**

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

**PSEN:**

Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S52 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

**EA/VPP:**

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming.

**XTAL1:**

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

**XTAL2:**

Output from the inverting oscillator amplifier

**Oscillator Characteristics:**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 6.2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

****

FIG 4.5(c): Oscillator Connections



FIG 4.5(d): External Clock Drive Configuration

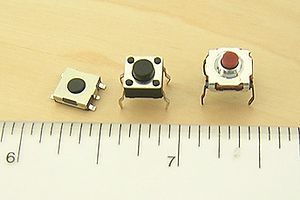
**Idle Mode**

In idle mode, the CPU puts itself to sleep while all the on chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

**Power down Mode**

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before VCC is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

**4.3 PUSH BUTTONS**

[](http://en.wikipedia.org/wiki/File:Tactile_switches.jpg)

A push-button (also spelled pushbutton) or simply button is a simple switch mechanism for controlling some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often biased switches, though even many un-biased buttons (due to their physical nature) require a spring to return to their un-pushed state. Different people use different terms for the "pushing" of the button, such as press, depress, mash, and punch.

**Uses:**

In industrial and commercial applications push buttons can be linked together by a mechanical linkage so that the act of pushing one button causes the other button to be released. In this way, a stop button can "force" a start button to be released. This method of linkage is used in simple manual operations in which the machine or process have no electrical circuits for control.

Pushbuttons are often color-coded to associate them with their function so that the operator will not push the wrong button in error. Commonly used colors are red for stopping the machine or process and green for starting the machine or process.

Red pushbuttons can also have large heads (mushroom shaped) for easy operation and to facilitate the stopping of a machine. These pushbuttons are called emergency stop buttons and are mandated by the electrical code in many jurisdictions for increased safety. This large mushroom shape can also be found in buttons for use with operators who need to wear gloves for their work and could not actuate a regular flush-mounted push button. As an aid for operators and users in industrial or commercial applications, a pilot light is commonly added to draw the attention of the user and to provide feedback if the button is pushed. Typically this light is included into the center of the pushbutton and a lens replaces the pushbutton hard center disk.

The source of the energy to illuminate the light is not directly tied to the contacts on the back of the pushbutton but to the action the pushbutton controls. In this way a start button when pushed will cause the process or machine operation to be started and a secondary contact designed into the operation or process will close to turn on the pilot light and signify the action of pushing the button caused the resultant process or action to start.

In popular culture, the phrase "the button" refers to a (usually fictional) button that a military or government leader could press to launch nuclear weapons.

## Push to ON button:

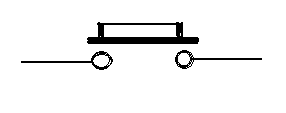
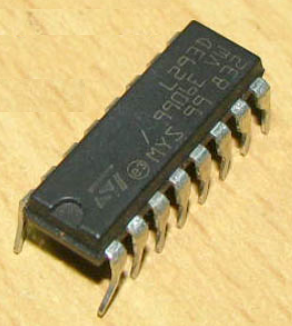


Fig 4.3(a): push on button

Initially the two contacts of the button are open. When the button is pressed they become connected. This makes the switching operation using the push button.

**4.4 MOTOR DRIVER (L293D)**

**Features:**

* Wide supply-voltage range: 4.5V to 36V
* Separate input- logic supply
* Internal ESD protection
* Thermal shutdown
* High-Noise-Immunity input
* Functional Replacements for SGS L293 and SGS L293D
* Output current 1A per channel (600 mA for L293D)
* Peak output current 2 A per channel (1.2 A for L293D)
* Output clamp diodes for Inductive Transient Suppression(L293D)

**DESCRIPTION:**

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.

**Block diagram:**

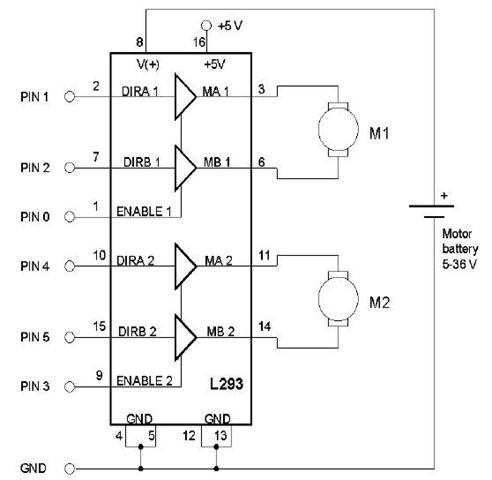
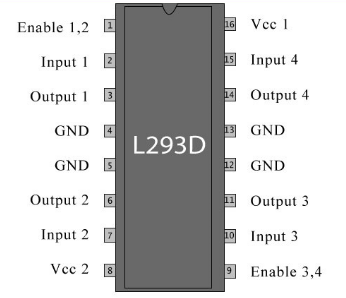
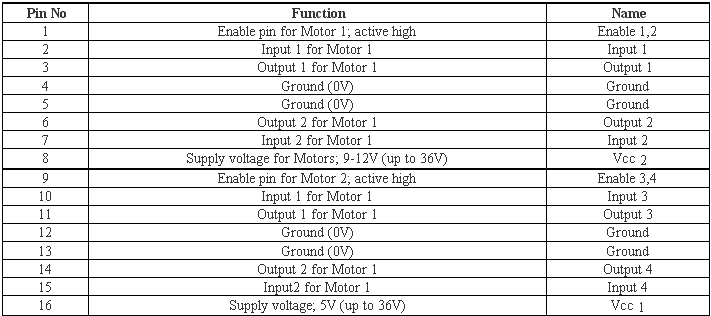
****

FIG: BLOCK DIAGRAM OF L293D

**Pin Diagram:**



** Pin description:**

**4.5 DC MOTOR**

A DC motor is an electric motor that runs on direct current (DC) electricity. In any electric motor, operation is based on simple electromagnetism. A [current](http://encyclobeamia.solarbotics.net/articles/current.html)-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the [current](http://encyclobeamia.solarbotics.net/articles/current.html) in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a [DC](http://encyclobeamia.solarbotics.net/articles/dc.html) motor is designed to harness the magnetic interaction between a [current](http://encyclobeamia.solarbotics.net/articles/current.html)-carrying conductor and an external magnetic field to generate rotational motion.

Let's start by looking at a simple 2-pole [DC](http://encyclobeamia.solarbotics.net/articles/dc.html) electric motor (here red represents a magnet or winding with a "North" polarization, while green represents a magnet or winding with a "South" polarization).

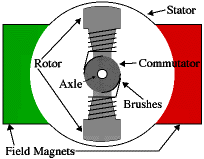
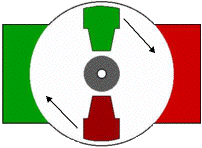


Fig.4.5(c) DC motor

Every [DC](http://encyclobeamia.solarbotics.net/articles/dc.html) motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors, the external magnetic field is produced by high-strength permanent magnets1. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor rotates with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

The geometry of the brushes, commutator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commutator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of [current](http://encyclobeamia.solarbotics.net/articles/current.html) through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.



In real life, though, [DC](http://encyclobeamia.solarbotics.net/articles/dc.html) motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of [torque](http://encyclobeamia.solarbotics.net/articles/torque.html) "ripple". So since most small [DC](http://encyclobeamia.solarbotics.net/articles/dc.html) motors are of a three-pole design, let's tinker with the workings of one via an interactive animation.

|  |  |
| --- | --- |
| Image |  |

You'll notice a few things from this -- namely, one pole is fully energized at a time (but two others are "partially" energized). As each brush transitions from one commutator contact to the next, one coil's field will rapidly collapse, as the next coil's field will rapidly charge up (this occurs within a few microsecond). We'll see more about the effects of this later, but in the meantime you can see that this is a direct result of the coil windings' series wiring.

**4.6 ULTRASONIC SENSOR PAIR**

****

Here is a more easy use serial ultrasonic module. It will auto output the distance information via serial port after power on, you don't need to do any trigger and calculated, just need to read the serial pin and get the distance information.

Ultrasonic sensor provides a very low-cost and easy method of distance measurement. This sensor is perfect for any number of applications that require you to perform measurements between moving or stationary objects. Naturally, robotics applications are very popular but you'll also find this product to be useful in security systems or as an infrared replacement if so desired. You will definitely appreciate the activity status LED and the economic use of just one I/O pin.

The ultrasonic sensor measures distance using sonar; an ultrasonic (well above human hearing) pulse is transmitted from the unit and distance-to-target is determined by measuring the time required for the echo return. Output from the ultrasonic sensor is a variable-width pulse that corresponds to the distance to the target.

**Features:**

* Provides precise, non-contact distance measurements within a 2 cm to 3 m range
* Simple pulse in/pulse out communication
* Burst indicator LED shows measurement in progress
* 20 mA power consumption
* Narrow acceptance angle
* 3-pin header makes it easy to connect using a servo extension cable, no soldering required

**Key Specifications:**

* power supply :5V DC
* quiescent current : <15mA
* effectual angle: <15°
* ranging distance : 2cm – 350 cm
* resolution : 0.3 cm
* Output cycle : 50ms
* Baud Rate : 9600

**Output frame format** （**4Bytes**）

* 1st Byte : 0xFF (Start bit, Fixed value)
* 2nd Byte: H\_Data (High 8 bit of the distance)
* 3rd Byte: L\_Date (Low 8 bit of the distance)
* 4th Byte: Check Sum (0XFF+H\_DATA+L\_DATA=SUM, Check SUM is the low 8 bit of the SUM)

**Uses:**

Ultrasonic sensors are used to detect the presence of targets and to measure the distance to targets in many automated factories and process plants. Sensors with an on or off digital output are available for detecting the presence of objects, and sensors with an analog output which varies proportionally to the sensor to target separation distance are commercially available.

Because ultrasonic sensors use sound rather than light for detection, they work in applications where photoelectric sensors may not. Ultrasonic are a great solution for clear object detection and for liquid level measurement, applications that photo electrics struggle with because of target translucence. Target colour and/or reflectivity don't affect ultrasonic sensors which can operate reliably in high-glare environments.

Other types of transducers are used in commercially available ultrasonic cleaning devices. An ultrasonic transducer is affixed to a stainless steel pan which is filled with a solvent (frequently water or isopropanol) and a square wave is applied to it, imparting vibrational energy on the liquid.

**4.7 LED**

A light-emitting diode (LED) is a [semiconductor](http://en.wikipedia.org/wiki/Semiconductor) light source. LEDs are used as indicator lamps in many devices, and are increasingly used for [lighting](http://en.wikipedia.org/wiki/Lighting). When a light-emitting [diode](http://en.wikipedia.org/wiki/Semiconductor_diode) is forward biased (switched on), [electrons](http://en.wikipedia.org/wiki/Electrons) are able to [recombine](http://en.wikipedia.org/wiki/Carrier_generation_and_recombination) with [holes](http://en.wikipedia.org/wiki/Electron_hole) within the device, releasing energy in the form of [photons](http://en.wikipedia.org/wiki/Photon).

This effect is called [electroluminescence](http://en.wikipedia.org/wiki/Electroluminescence) and the [color](http://en.wikipedia.org/wiki/Color) of the light (corresponding to the energy of the photon) is determined by the [energy gap](http://en.wikipedia.org/wiki/Energy_gap) of the semiconductor. An LED is often small in area (less than 1 mm2), and integrated optical components may be used to shape its radiation pattern. LEDs present many [advantages](http://en.wikipedia.org/wiki/Led#Advantages) over incandescent light sources including [lower energy consumption](http://en.wikipedia.org/wiki/Energy_conservation), longer [lifetime](http://en.wikipedia.org/wiki/Service_life), improved robustness, smaller size, faster switching, and greater durability and reliability.

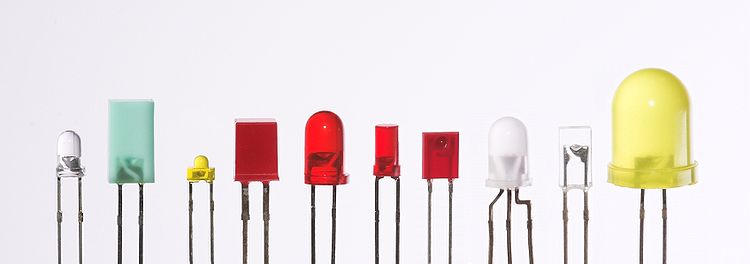
**Types of LED’S** [](http://en.wikipedia.org/wiki/File:Verschiedene_LEDs.jpg)

Fig 4.11(a): Types of LED

Light-emitting diodes are used in applications as diverse as replacements for [aviation lighting](http://en.wikipedia.org/wiki/Navigation_light#Aviation_navigation_lights), [automotive lighting](http://en.wikipedia.org/wiki/Automotive_lighting#Light_emitting_diodes_.28LED.29) as well as in [traffic signals](http://en.wikipedia.org/wiki/Traffic_signal). The compact size, the possibility of narrow bandwidth, switching speed, and extreme reliability of LEDs has allowed new text and video displays and sensors to be developed, while their high switching rates are also useful in advanced communications technology.

**Electronic Symbol:**

[LED symbol.svg](http://en.wikipedia.org/wiki/File:LED_symbol.svg)

Fig 4.11(b): symbol of LED

**4.8 1N4007**

Diodes are used to convert AC into DC these are used as half wave rectifier or full wave rectifier. Three points must he kept in mind while using any type of diode.

1. Maximum forward current capacity
2. Maximum reverse voltage capacity
3. Maximum forward voltage capacity

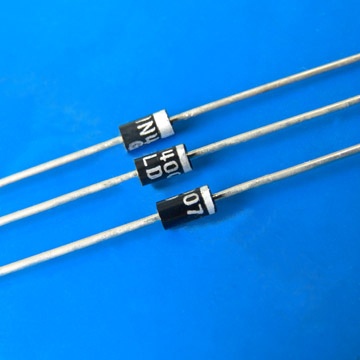


Fig: 1N4007 diodes

The number and voltage capacity of some of the important diodes available in the market are as follows:

* Diodes of number IN4001, IN4002, IN4003, IN4004, IN4005, IN4006 and IN4007 have maximum reverse bias voltage capacity of 50V and maximum forward current capacity of 1 Amp.
* Diode of same capacities can be used in place of one another. Besides this diode of more capacity can be used in place of diode of low capacity but diode of low capacity cannot be used in place of diode of high capacity. For example, in place of IN4002; IN4001 or IN4007 can be used but IN4001 or IN4002 cannot be used in place of IN4007.The diode BY125made by company BEL is equivalent of diode from IN4001 to IN4003. BY 126 is equivalent to diodes IN4004 to 4006 and BY 127 is equivalent to diode IN4007.

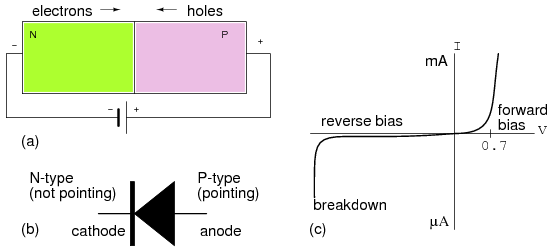


Fig:PN Junction diode

PN JUNCTION OPERATION

Now that you are familiar with P- and N-type materials, how these materials are joined together to form a diode, and the function of the diode, let us continue our discussion with the operation of the PN junction. But before we can understand how the PN junction works, we must first consider current flow in the materials that make up the junction and what happens initially within the junction when these two materials are joined together.

Current Flow in the N-Type Material

Conduction in the N-type semiconductor, or crystal, is similar to conduction in a copper wire. That is, with voltage applied across the material, electrons will move through the crystal just as current would flow in a copper wire. This is shown in figure 1-15. The positive potential of the battery will attract the free electrons in the crystal. These electrons will leave the crystal and flow into the positive terminal of the battery. As an electron leaves the crystal, an electron from the negative terminal of the battery will enter the crystal, thus completing the current path. Therefore, the majority current carriers in the N-type material (electrons) are repelled by the negative side of the battery and move through the crystal toward the positive side of the battery.

Current Flow in the P-Type Material

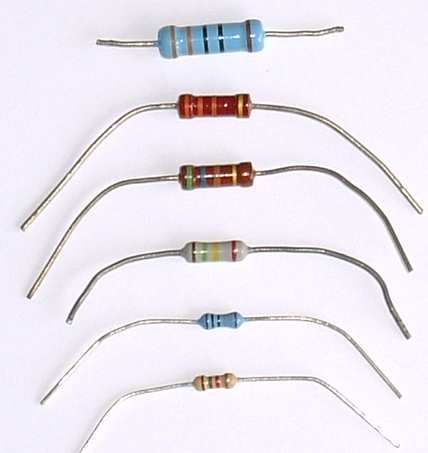
Current flow through the P-type material is illustrated. Conduction in the P material is by positive holes, instead of negative electrons. A hole moves from the positive terminal of the P material to the negative terminal. Electrons from the external circuit enter the negative terminal of the material and fill holes in the vicinity of this terminal. At the positive terminal, electrons are removed from the covalent bonds, thus creating new holes. This process continues as the steady stream of holes (hole current) moves toward the negative terminal

**4.9 RESISTORS**

A resistor is a two-terminal electronic component designed to oppose an electric current by producing a voltage drop between its terminals in proportion to the current, that is, in accordance with Ohm's law:

*V* = *IR*

Resistors are used as part of electrical networks and electronic circuits. They are extremely commonplace in most electronic equipment. Practical resistors can be made of various compounds and films, as well as resistance wire (wire made of a high-resistivity alloy, such as nickel/chrome).



The primary characteristics of resistors are their resistance and the power they can dissipate. Other characteristics include temperature coefficient, noise, and inductance. Less well-known is critical resistance, the value below which power dissipation limits the maximum permitted current flow, and above which the limit is applied voltage. Critical resistance depends upon the materials constituting the resistor as well as its physical dimensions; it's determined by design.

Resistors can be integrated into hybrid and printed circuits, as well as integrated circuits. Size, and position of leads (or terminals) are relevant to equipment designers; resistors must be physically large enough not to overheat when dissipating their power.

A resistor is a two-[terminal](http://en.wikipedia.org/wiki/Terminal_%28electronics%29) [passive](http://en.wikipedia.org/wiki/Passivity_%28engineering%29) [electronic component](http://en.wikipedia.org/wiki/Electronic_component) which implements [electrical resistance](http://en.wikipedia.org/wiki/Electrical_resistance) as a circuit element. When a voltage V is applied across the terminals of a resistor, a current I will flow through the resistor in [direct proportion](http://en.wikipedia.org/wiki/Direct_proportion) to that voltage. The reciprocal of the constant of proportionality is known as the [resistance](http://en.wikipedia.org/wiki/Resistance) R, since, with a given voltage V, a larger value of R further "resists" the flow of current I as given by [Ohm's law](http://en.wikipedia.org/wiki/Ohm%27s_law):

I = {V \over R}

Resistors are common elements of [electrical networks](http://en.wikipedia.org/wiki/Electrical_networks) and electronic circuits and are ubiquitous in most electronic equipment. Practical resistors can be made of various compounds and films, as well as [resistance wire](http://en.wikipedia.org/wiki/Resistance_wire) (wire made of a high-resistivity alloy, such as nickel-chrome). Resistors are also implemented within [integrated circuits](http://en.wikipedia.org/wiki/Integrated_circuits), particularly analog devices, and can also be integrated into [hybrid](http://en.wikipedia.org/wiki/Hybrid_circuit) and [printed circuits](http://en.wikipedia.org/wiki/Printed_circuit_board).

The electrical functionality of a resistor is specified by its resistance: common commercial resistors are manufactured over a range of more than 9 [orders of magnitude](http://en.wikipedia.org/wiki/Orders_of_magnitude). When specifying that resistance in an electronic design, the required precision of the resistance may require attention to the [manufacturing tolerance](http://en.wikipedia.org/wiki/Engineering_tolerance#Electrical_component_tolerance) of the chosen resistor, according to its specific application. The [temperature coefficient](http://en.wikipedia.org/wiki/Temperature_coefficient) of the resistance may also be of concern in some precision applications. Practical resistors are also specified as having a maximum [power](http://en.wikipedia.org/wiki/Power_%28physics%29) rating which must exceed the anticipated power dissipation of that resistor in a particular circuit: this is mainly of concern in power electronics applications. Resistors with higher power ratings are physically larger and may require [heat sinking](http://en.wikipedia.org/wiki/Heat_sink). In a high voltage circuit, attention must sometimes be paid to the rated maximum working voltage of the resistor.

The series [inductance](http://en.wikipedia.org/wiki/Inductance) of a practical resistor causes its behaviour to depart from ohms law; this specification can be important in some high-frequency applications for smaller values of resistance. In a [low-noise amplifier](http://en.wikipedia.org/wiki/Low-noise_amplifier) or [pre-amp](http://en.wikipedia.org/wiki/Pre-amp) the noise characteristics of a resistor may be an issue. The unwanted inductance, excess noise, and temperature coefficient are mainly dependent on the technology used in manufacturing the resistor. They are not normally specified individually for a particular family of resistors manufactured using a particular technology.[[1]](http://en.wikipedia.org/wiki/Resistor#cite_note-0) A family of discrete resistors is also characterized according to its form factor, that is, the size of the device and position of its leads (or terminals) which is relevant in the practical manufacturing of circuits using them.

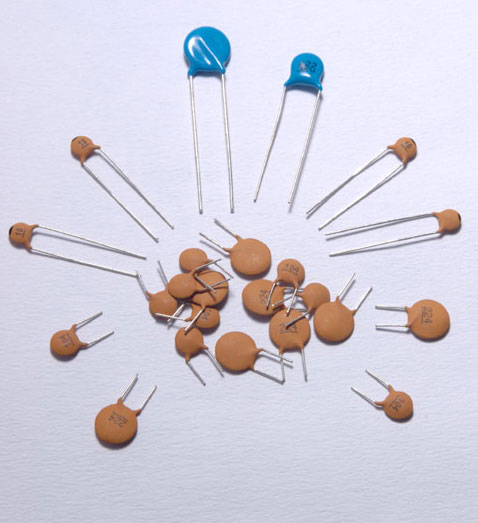
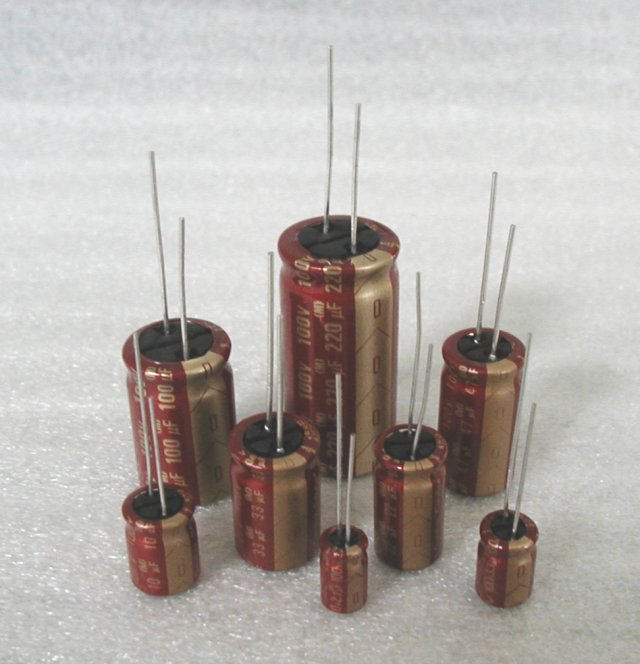
## Units

The [ohm](http://en.wikipedia.org/wiki/Ohm_%28unit%29) (symbol: [Ω](http://en.wikipedia.org/wiki/%CE%A9)) is the [SI](http://en.wikipedia.org/wiki/SI) unit of [electrical resistance](http://en.wikipedia.org/wiki/Electrical_resistance), named after [Georg Simon Ohm](http://en.wikipedia.org/wiki/Georg_Simon_Ohm). An ohm is equivalent to a [volt](http://en.wikipedia.org/wiki/Volt) per [ampere](http://en.wikipedia.org/wiki/Ampere). Since resistors are specified and manufactured over a very large range of values, the derived units of milliohm (1 mΩ = 10−3 Ω), kilohm (1 kΩ = 103 Ω), and megohm (1 MΩ = 106 Ω) are also in common usage.

The reciprocal of resistance R is called [conductance](http://en.wikipedia.org/wiki/Conductance) G = 1/R and is measured in [Siemens](http://en.wikipedia.org/wiki/Siemens_%28unit%29) ([SI](http://en.wikipedia.org/wiki/SI) unit), sometimes referred to as a [mho](http://en.wikipedia.org/wiki/Mho). Thus a Siemens is the reciprocal of an ohm: *S* = Ω − 1. Although the concept of conductance is often used in circuit analysis, practical resistors are always specified in terms of their resistance (ohms) rather than conductance.

**4.10 CAPACITORS**

A capacitor or condenser is a passive electronic component consisting of a pair of conductors separated by a dielectric. When a voltage potential difference exists between the conductors, an electric field is present in the dielectric. This field stores energy and produces a mechanical force between the plates. The effect is greatest between wide, flat, parallel, narrowly separated conductors.



An ideal capacitor is characterized by a single constant value, capacitance, which is measured in farads. This is the ratio of the electric charge on each conductor to the potential difference between them. In practice, the dielectric between the plates passes a small amount of leakage current. The conductors and leads introduce an equivalent series resistance and the dielectric has an electric field strength limit resulting in a breakdown voltage.

The properties of capacitors in a circuit may determine the resonant frequency and quality factor of a resonant circuit, power dissipation and operating frequency in a digital logic circuit, energy capacity in a high-power system, and many other important aspects.

A capacitor (formerly known as condenser) is a device for storing electric charge. The forms of practical capacitors vary widely, but all contain at least two conductors separated by a non-conductor. Capacitors used as parts of electrical systems, for example, consist of metal foils separated by a layer of insulating film.

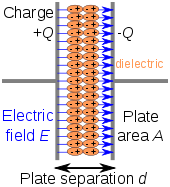
Capacitors are widely used in electronic circuits for blocking [direct current](http://en.wikipedia.org/wiki/Direct_current) while allowing [alternating current](http://en.wikipedia.org/wiki/Alternating_current) to pass, in filter networks, for smoothing the output of [power supplies](http://en.wikipedia.org/wiki/Power_supply), in the [resonant circuits](http://en.wikipedia.org/wiki/LC_circuit) that tune radios to particular [frequencies](http://en.wikipedia.org/wiki/Frequency) and for many other purposes.

A capacitor is a [passive](http://en.wikipedia.org/wiki/Passivity_%28engineering%29) [electronic component](http://en.wikipedia.org/wiki/Electronic_component) consisting of a pair of [conductors](http://en.wikipedia.org/wiki/Electrical_conductor) separated by a [dielectric](http://en.wikipedia.org/wiki/Dielectric) (insulator). When there is a [potential difference](http://en.wikipedia.org/wiki/Potential_difference) (voltage) across the conductors, a static [electric field](http://en.wikipedia.org/wiki/Electric_field) develops in the dielectric that stores [energy](http://en.wikipedia.org/wiki/Energy) and produces a mechanical force between the conductors. An ideal capacitor is characterized by a single constant value, [capacitance](http://en.wikipedia.org/wiki/Capacitance), measured in [farads](http://en.wikipedia.org/wiki/Farad). This is the ratio of the [electric charge](http://en.wikipedia.org/wiki/Electric_charge) on each conductor to the potential difference between them.

The capacitance is greatest when there is a narrow separation between large areas of conductor, hence capacitor conductors are often called "plates", referring to an early means of construction. In practice the dielectric between the plates passes a small amount of [leakage current](http://en.wikipedia.org/wiki/Leakage_%28electronics%29) and also has an electric field strength limit, resulting in a [breakdown voltage](http://en.wikipedia.org/wiki/Breakdown_voltage), while the conductors and [leads](http://en.wikipedia.org/wiki/Lead_%28electronics%29) introduce an undesired [inductance](http://en.wikipedia.org/wiki/Equivalent_series_inductance) and [resistance](http://en.wikipedia.org/wiki/Equivalent_series_resistance).

## Theory of operation

Main article: [Capacitance](http://en.wikipedia.org/wiki/Capacitance)

[](http://en.wikipedia.org/wiki/File:Capacitor_schematic_with_dielectric.svg)

Charge separation in a parallel-plate capacitor causes an internal electric field. A dielectric (orange) reduces the field and increases the capacitance.

[](http://en.wikipedia.org/wiki/File:Plattenkondensator_hg.jpg)

A simple demonstration of a parallel-plate capacitor

A capacitor consists of two [conductors](http://en.wikipedia.org/wiki/Electrical_conductor) separated by a non-conductive region[[8]](http://en.wikipedia.org/wiki/Capacitor" \l "cite_note-Ulaby_p168-7). The non-conductive region is called the dielectric or sometimes the [dielectric medium](http://en.wikipedia.org/wiki/Dielectric_medium). In simpler terms, the dielectric is just an [electrical insulator](http://en.wikipedia.org/wiki/Insulator_%28electrical%29). Examples of dielectric mediums are glass, air, paper, [vacuum](http://en.wikipedia.org/wiki/Vacuum), and even a [semiconductor](http://en.wikipedia.org/wiki/Semiconductor) [depletion region](http://en.wikipedia.org/wiki/Depletion_region) chemically identical to the conductors. A capacitor is assumed to be self-contained and isolated, with no net [electric charge](http://en.wikipedia.org/wiki/Electric_charge) and no influence from any external electric field. The conductors thus hold equal and opposite charges on their facing surfaces,[[9]](http://en.wikipedia.org/wiki/Capacitor" \l "cite_note-Ulaby_p157-8) and the dielectric develops an electric field. In [SI](http://en.wikipedia.org/wiki/SI) units, a capacitance of one [farad](http://en.wikipedia.org/wiki/Farad) means that one [coulomb](http://en.wikipedia.org/wiki/Coulomb) of charge on each conductor causes a voltage of one [volt](http://en.wikipedia.org/wiki/Volt) across the device.[[10]](http://en.wikipedia.org/wiki/Capacitor#cite_note-Ulaby_p169-9)

The capacitor is a reasonably general model for electric fields within electric circuits. An ideal capacitor is wholly characterized by a constant capacitance C, defined as the ratio of charge ±Q on each conductor to the voltage V between them:[[8]](http://en.wikipedia.org/wiki/Capacitor" \l "cite_note-Ulaby_p168-7)

C= \frac{Q}{V}

Sometimes charge build-up affects the capacitor mechanically, causing its capacitance to vary. In this case, capacitance is defined in terms of incremental changes:

C= \frac{\mathrm{d}q}{\mathrm{d}v}

### Energy storage

[Work](http://en.wikipedia.org/wiki/Work_%28thermodynamics%29) must be done by an external influence to "move" charge between the conductors in a capacitor. When the external influence is removed the charge separation persists in the electric field and energy is stored to be released when the charge is allowed to return to its [equilibrium](http://en.wikipedia.org/wiki/Equilibrium) position. The work done in establishing the electric field, and hence the amount of energy stored, is given by:[[11]](http://en.wikipedia.org/wiki/Capacitor" \l "cite_note-10)

W= \int_{q=0}^Q V \text{d}q = \int_{q=0}^Q \frac{q}{C} \text{d}q = {1 \over 2} {Q^2 \over C} = {1 \over 2}  C V^2 = {1 \over 2} VQ.

### Current-voltage relation

The current i(t) through any component in an electric circuit is defined as the rate of flow of a charge q(t) passing through it, but actual charges, electrons, cannot pass through the dielectric layer of a capacitor, rather an electron accumulates on the negative plate for each one that leaves the positive plate, resulting in an electron depletion and consequent positive charge on one electrode that is equal and opposite to the accumulated negative charge on the other. Thus the charge on the electrodes is equal to the [integral](http://en.wikipedia.org/wiki/Integral) of the current as well as proportional to the voltage as discussed above. As with any [antiderivative](http://en.wikipedia.org/wiki/Antiderivative), a [constant of integration](http://en.wikipedia.org/wiki/Constant_of_integration) is added to represent the initial voltage v (t0). This is the integral form of the capacitor equation,[[12]](http://en.wikipedia.org/wiki/Capacitor" \l "cite_note-Dorf_p263-11)

v(t)= \frac{q(t)}{C} = \frac{1}{C}\int_{t_0}^t i(\tau) \mathrm{d}\tau+v(t_0).

Taking the derivative of this, and multiplying by C, yields the derivative form,[[13]](http://en.wikipedia.org/wiki/Capacitor" \l "cite_note-Dorf_p260-12)

i(t)= \frac{\mathrm{d}q(t)}{\mathrm{d}t}=C\frac{\mathrm{d}v(t)}{\mathrm{d}t}.

The [dual](http://en.wikipedia.org/wiki/Duality_%28electrical_circuits%29) of the capacitor is the [inductor](http://en.wikipedia.org/wiki/Inductor), which stores energy in the [magnetic field](http://en.wikipedia.org/wiki/Magnetic_field) rather than the electric field. Its current-voltage relation is obtained by exchanging current and voltage in the capacitor equations and replacing C with the inductance L.

**5. SOFTWARE REQUIREMENTS**

**5.1 INTRODUCTION TO KEIL MICRO VISION (IDE)**

Keil an ARM Company makes C compilers, macro assemblers, real-time kernels, debuggers, simulators, integrated environments, evaluation boards, and emulators for ARM7/ARM9/Cortex-M3, XC16x/C16x/ST10, 251, and 8051 MCU families.

Keil development tools for the 8051 Microcontroller Architecture support every level of software developer from the professional applications engineer to the student just learning about embedded software development. When starting a new project, simply select the microcontroller you use from the Device Database and the µVision IDE sets all compiler, assembler, linker, and memory options for you.

  Keil is a cross compiler. So first we have to understand the concept of compilers and cross compilers. After then we shall learn how to work with keil.

**5.2 CONCEPT OF COMPILER**

Compilers are programs used to convert a High Level Language to object code. Desktop compilers produce an output object code for the underlying microprocessor, but not for other microprocessors. I.E the programs written in one of the HLL like ‘C’ will compile the code to run on the system for a particular processor like x86 (underlying microprocessor in the computer). For example compilers for Dos platform is different from the Compilers for Unix platform  So if one wants to define a compiler then compiler is a program that translates source code into object code.

The compiler derives its name from the way it works, looking at the entire piece of source code and collecting and reorganizing the instruction. See there is a bit little difference between compiler and an interpreter. Interpreter just interprets whole program at a time while compiler analyses and execute each line of source code in succession, without looking at the entire program.

The advantage of interpreters is that they can execute a program immediately. Secondly programs produced by compilers run much faster than the same programs executed by an interpreter. However compilers require some time before an executable program emerges. Now as compilers translate source code into object code, which is unique for each type of computer, many compilers are available for the same language.

**5.3 CONCEPT OF CROSS COMPILER**

  A cross compiler is similar to the compilers but we write a program for the target processor (like 8051 and its derivatives) on the host processors (like computer of x86). It means being in one environment you are writing a code for another environment is called cross development. And the compiler used for cross development is called cross compiler. So the definition of cross compiler is a compiler that runs on one computer but produces object code for a different type of computer.

**5.4 KEIL C CROSS COMPILER**

Keil is a German based Software development company. It provides several development tools like

•         IDE (Integrated Development environment)

•         Project Manager

•         Simulator

•         Debugger

•         C Cross Compiler, Cross Assembler, Locator/Linker

The Keil ARM tool kit includes three main tools, assembler, compiler and linker. An assembler is used to assemble the ARM assembly program. A compiler is used to compile the C source code into an object file. A linker is used to create an absolute object module suitable for our in-circuit emulator.

5.5 Building an Application in µVision2

To build (compile, assemble, and link) an application in µVision2, you must:

1. Select Project -(forexample,166\EXAMPLES\HELLO\HELLO.UV2).
2. Select Project - Rebuild all target files or Build target.µVision2 compiles, assembles, and links the files in your project.

5.6 Creating Your Own Application in µVision2

To create a new project in µVision2, you must:

1. Select Project - New Project.
2. Select a directory and enter the name of the project file.
3. Select Project - Select Device and select an 8051, 251, or C16x/ST10 device from the Device Database™.
4. Create source files to add to the project.
5. Select Project - Targets, Groups, Files. Add/Files, select Source Group1, and add the source files to the project.
6. Select Project - Options and set the tool options. Note when you select the target device from the Device Database™ all special options are set automatically. You typically only need to configure the memory map of your target hardware. Default memory model settings are optimal for most applications.
7. Select Project - Rebuild all target files or Build target.

5.7 Debugging an Application in µVision2

To debug an application created using µVision2, you must:

1. Select Debug - Start/Stop Debug Session.
2. Use the Step toolbar buttons to single-step through your program. You may enter G, main in the Output Window to execute to the main C function.
3. Open the Serial Window using the Serial #1 button on the toolbar.

Debug your program using standard options like Step, Go, Break, and so on.

**5.8 Starting µVision2 and Creating a Project**

µVision2 is a standard Windows application and started by clicking on the program icon. To create a new project file select from the µVision2 menu Project – New Project…. This opens a standard Windows dialog that asks you for the new project file name. We suggest that you use a separate folder for each project. You can simply use the icon Create New Folder in this dialog to get a new empty folder. Then select this folder and enter the file name for the new project, i.e. Project1. µVision2 creates a new project file with the name PROJECT1.UV2 which contains a default target and file group name. You can see these names in the Project.

**5.9 Window – Files.**

Now use from the menu Project – Select Device for Target and select a CPU for your project. The Select Device dialog box shows the µVision2 device data base. Just select the microcontroller you use. We are using for our examples the Philips 80C51RD+ CPU. This selection sets necessary tool Options for the 80C51RD+ device and simplifies in this way the tool Configuration.

**5.10 Building Projects and Creating a HEX Files**

Typical, the tool settings under Options – Target are all you need to start a new application. You may translate all source files and line the application with a click on the Build Target toolbar icon. When you build an application with syntax errors, µVision2 will display errors and warning messages in the Output Window – Build page. A double click on a message line opens the source file on the correct location in a µVision2 editor window. Once you have successfully generated your application you can start debugging.

After you have tested your application, it is required to create an Intel HEX file to download the software into an EPROM programmer or simulator. µVision2 creates HEX files with each build process when Create HEX files under Options for Target – Output is enabled. You may start your PROM programming utility after the make process when you specify the program under the option Run User Program #1.

**5.11 CPU Simulation**

µVision2 simulates up to 16 Mbytes of memory from which areas can be mapped for read, write, or code execution access. The µVision2 simulator traps

and reports illegal memory accesses. In addition to memory mapping, the simulator also provides support for the integrated peripherals of the various 8051 derivatives. The on-chip peripherals of the CPU you have selected are configured from the Device.

**5.12 Database selection**

You have made when you create your project target. Refer to page 58 for more Information about selecting a device. You may select and display the on-chip peripheral components using the Debug menu. You can also change the aspects of each peripheral using the controls in the dialog boxes.

**5.13 Start Debugging**

You start the debug mode of µVision2 with the Debug – Start/Stop Debug Session Command. Depending on the Options for Target – Debug Configuration, µVision2 will load the application program and run the startup code µVision2 saves the editor screen layout and restores the screen layout of the last debug session. If the program execution stops, µVision2 opens an editor window with the source text or shows CPU instructions in the disassembly window. The next executable statement is marked with a yellow arrow. During debugging, most editor features are still available.

For example, you can use the find command or correct program errors. Program source text of your application is shown in the same windows. The µVision2 debug mode differs from the edit mode in the following aspects:

\_ The “Debug Menu and Debug Commands” described on page 28 are available. The additional debug windows are discussed in the following.

\_ The project structure or tool parameters cannot be modified. All build commands are disabled.

**5.14 Disassembly Window**

The Disassembly window shows your target program as mixed source and assembly program or just assembly code. A trace history of previously executed instructions may be displayed with Debug – View Trace Records. To enable the trace history, set Debug – Enable/Disable Trace Recording.

If you select the Disassembly Window as the active window all program step commands work on CPU instruction level rather than program source lines. You can select a text line and set or modify code breakpoints using toolbar buttons or the context menu commands.

You may use the dialog Debug – Inline Assembly… to modify the CPU instructions. That allows you to correct mistakes or to make temporary changes to the target program you are debugging. Numerous example programs are included to help you get started with the most popular embedded 8051 devices.

The Keil µVision Debugger accurately simulates on-chip peripherals (I²C, CAN, UART, SPI, Interrupts, I/O Ports, A/D Converter, D/A Converter, and PWM Modules) of your 8051 device. Simulation helps you understand hardware configurations and avoids time wasted on setup problems. Additionally, with simulation, you can write and test applications before target hardware is available.

**5.15 EMBEDDED C**

Use of embedded processors in passenger cars, mobile phones, medical equipment, aerospace systems and defense systems is widespread, and even everyday domestic appliances such as dish washers, televisions, washing machines and video recorders now include at least one such device.

Because most embedded projects have severe cost constraints, they tend to use low-cost processors like the 8051 family of devices considered in this book. These popular chips have very limited resources available most such devices have around 256 bytes (not megabytes!) of RAM, and the available processor power is around 1000 times less than that of a desktop processor. As a result, developing embedded software presents significant new challenges, even for experienced desktop programmers. If you have some programming experience - in C, C++ or Java - then this book and its accompanying CD will help make your move to the embedded world as quick and painless as possible.

**6. SCHEMATIC DIAGRAM**

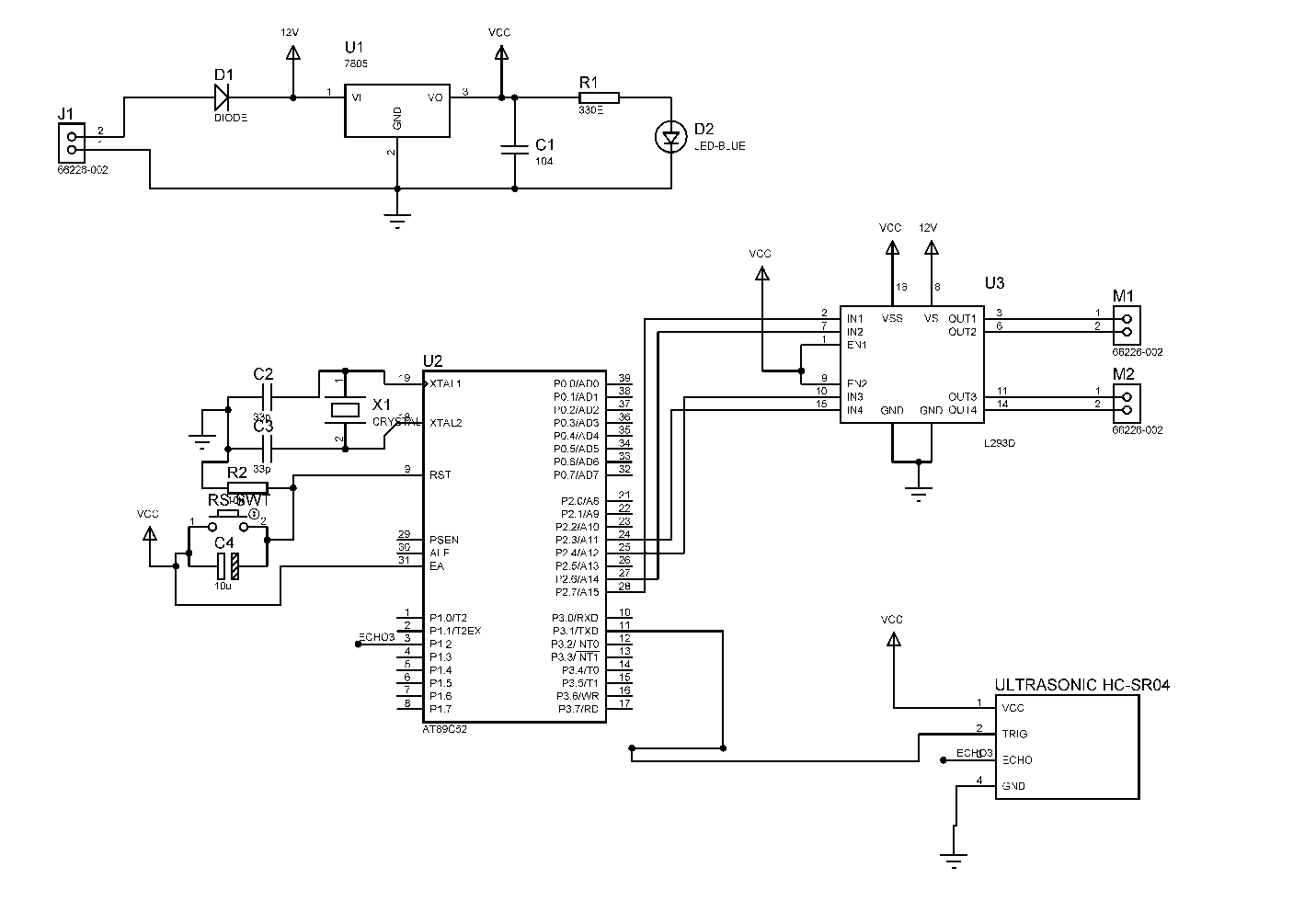
****

Fig 6: SCHEMATIC DIAGRAM

**6.1 DESCRIPTION**

**POWER SUPPLY**

This project uses a 6V battery for power supply. A diode is connected to get the required voltage after the drop across the diode. One LED is connected of this 5V point in series with a resistor of 330Ω to the ground i.e., negative voltage to indicate 5V power supply availability.

**STANDARD CONNECTIONS TO 8051 SERIES MICRO CONTROLLER**

ATMEL series of 8051 family of micro controllers need certain standard connections. The actual number of the Microcontroller could be “89C51” , “89C52”, “89S51”, “89S52”, and as regards to 20 pin configuration a number of “89C2051”. The 4 set of I/O ports are used based on the project requirement. Every microcontroller requires a timing reference for its internal program execution therefore an oscillator needs to be functional with a desired frequency to obtain the timing reference as t =1/f.

A crystal ranging from 2 to 20 MHz is required to be used at its pin number 18 and 19 for the internal oscillator. It may be noted here the crystal is not to be understood as crystal oscillator It is just a crystal, while connected to the appropriate pin of the microcontroller it results in oscillator function inside the microcontroller. Typically 11.0592 MHz crystal is used in general for most of the circuits using 8051 series microcontroller. Two small value ceramic capacitors of 33pF each is used as a standard connection for the crystal as shown in the circuit diagram.

**RESET**

Pin no 9 is provided with an re-set arrangement by a combination of an electrolytic capacitor and a register forming RC time constant. At the time of switch on, the capacitor gets charged, and it behaves as a full short circuit from the positive to the pin number 9. After the capacitor gets fully charged the current stops flowing and pin number 9 goes low which is pulled down by a 10k resistor to the ground. This arrangement of reset at pin 9 going high initially and then to logic 0 i.e., low helps the program execution to start from the beginning. In absence of this the program execution could have taken place arbitrarily anywhere from the program cycle. A pushbutton switch is connected across the capacitor so that at any given time as desired it can be pressed such that it discharges the capacitor and while released the capacitor starts charging again and then pin number 9 goes to high and then back to low, to enable the program execution from the beginning. This operation of high to low of the reset pin takes place in fraction of a second as decided by the time constant R and C.

For example: A 10µF capacitor and a 10kΩ resistor would render a 100ms time to pin number 9 from logic high to low, there after the pin number 9 remains low.

**External Access(EA):**

Pin no 31 of 40 pin 8051 microcontroller termed as EA¯ is required to be connected to 5V for accessing the program form the on-chip program memory. If it is connected to ground then the controller accesses the program from external memory. However as we are using the internal memory it is always connected to +5V.

**L293D MOTOR DRIVER**

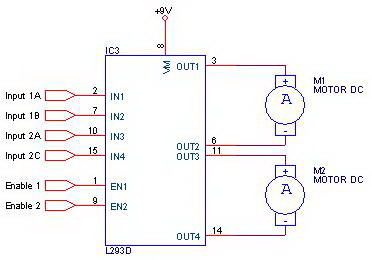
L293D has 2 set of arrangements where one set has input 1, input 2, output 1 and output 2 and other set has input 3, input 4, output 3 and output 4, according to block diagram if pin no 2 & 7 are high then pin no 3 & 6 are also high.

If enable 1 and pin number 2 are high leaving pin number 7 as low then the motor rotates in forward direction.

If enable 2 and pin number 10 are high leaving pin number 15 as low then the motor rotates in forward direction.

If enable 1 and pin number 2 are low leaving pin number 7 as high then the motor rotates in reverse direction.

If enable 2 and pin number 15 are high leaving pin number 10 as low then the motor rotates in forward direction.

****

**OPERATION EXPLANATION**

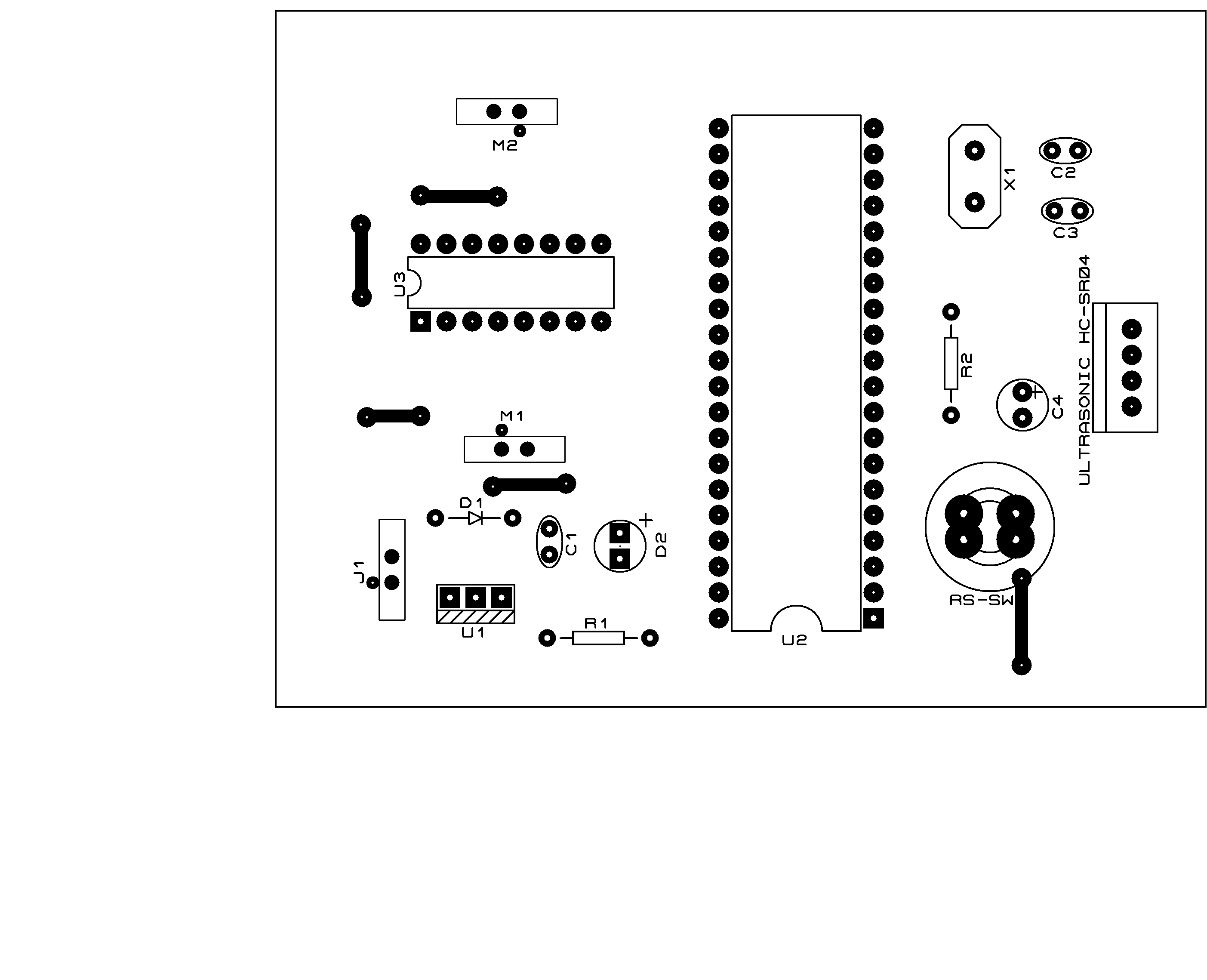
**Connections:**

The output of the power supply which is 5v is connected to 40 pin of MC and GND is connected to its 20th pin. Pin 10 of MC is connected to pin 2 of ultrasonic sensor. Pin 2.3 of port 2 of MC is connected to pin 9 i.e., EN2 of motor driver L293D. Pins 21, 22, 23 of MC are connected to pins 1,2,7 of L293D resp. Pins 24,25,26 of MC are connected to 9,10,15 of L293D resp. Pin 3 and 6 of L293D are connected to dc motor 1. Pin no 14 and 11 of L293D are connected to dc motor 2. Pin 8 & 16 of L293D is given +5v. Pin no 4, 5, 13, 12 of L293D are given to GND.

**Working:**

This circuit represents an obstacle avoidance robot. The motors are driven by the motor driver IC L293D as explained above. The ultrasonic distance sensor module output is connected to Rx pin of MC whenever any obstacle comes ahead of the sensor it detects at how much distance it is and sends the data to Rx pin of MC. The program is so written that whenever any obstacle comes less than 4cm ahead of it, MC gives signal to motor driver IC which drives the motor backward and turns by some angle and then moves forward in same direction. When input1 is logic high and input2 pin is logic low with enable1 pin of L293D high, the robot moves in forward direction .If input1 is logic low and input2 is logic high and logic high respectively. with en1 pin of L293D high, the robot moves in backward direction as explained above following the standard L293D functioning. Turning to right or left is achieved depending on the signal received from the MC one enable pin is driven low there by forcing the corresponding motor to stop while the other motor keeps on rotating giving the robot to take a turn.

**7. LAYOUT DIAGRAM**

****

**8. BILL OF MATERIALS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Quantity** | **References** | **Value** |
| Modules | 2 | M1-M2 | 66226-002 |
| Capacitors | 1 | C1 | 104 |
| Capacitors | 2 | C2-C3 | 33p |
| Capacitors | 1 | C4 | 10u |
| Resistors | 1 | R1 | 330E |
| Resistors | 1 | R2 | 10k |
| Integrated Circuits | 1 | U1 | 7805 |
| Integrated Circuits | 1 | U2 | AT89C52 |
| Integrated Circuits | 1 | U3 | L293D |
| Diodes | 1 | D1 | DIODE |
| Diodes | 1 | D2 | LED-BLUE |
| Miscellaneous | 1 | J1 | 66226-002 |
| Miscellaneous | 2 | RS-SWT,ULTRASONIC HC-SR04 |  |
| Miscellaneous | 1 | X1 | CRYSTAL |

**9.CODING**

# include <at89x52.h>

# include <string.h>

sbit IN1=P2^7;

sbit IN2=P2^6;

sbit IN3=P2^4;

sbit IN4=P2^3;

sfr16 DPTR =0x82;

sbit triger=P3^1;

sbit echo=P1^2;

unsigned int ch,high\_byte,low\_byte,high\_byte,distance;

unsigned int target\_range=0,d=0,left=0,right=0;

unsigned int range=0;

unsigned int s=0;

bit d1=0,d2=0,d3=0,sm1=0,sm2=0;

void init();

void delay(unsigned int delay\_ms);

void get\_range();

void process();

void main()

{

init();

triger=0;

echo=1;

triger=1;

delay(10);

triger=0;

delay(41);

d1=1;

delay(2);

P2=0x00;

delay(20);

TMOD=0x09;//timer0 in 16 bit mode with gate enable

TR0=1;//timer run enabled

TH0=0x00;

TL0=0x00;

P3=0x04;

IN1=0;

IN2=0;

IN3=0;

IN4=0;

left=0;

right=0;

while(1)

{

get\_range();

delay(500);

process();

}

}

void init()

{

TMOD=0X09;

TR0=1;

TH0=0X00;

TL0=0X00;

}

void delay(unsigned int delay\_ms)

{

unsigned int count,i;

for(count = 0;count<delay\_ms;count++)

{

for(i =0;i < 120;i++);

}

delay\_ms--;

}

void get\_range()

{

TH0=0X00;

TL0=0X00;

triger=1;

delay(10);

triger=0;

while(!echo);

while(echo);

DPH=TH0;

DPL=TL0;

TH0=0XFF;

TL0=0XFF;

if(DPTR<35000)

range=DPTR/78.66;

else

range=0;

}

void process()

{

if(range<=150)

{

if(sm2==1)

{

IN1=0;

IN2=0;

IN3=0;

IN4=0;

sm2=0;

delay(100);

}

IN1=1;

IN2=0;

IN3=0;

IN4=1;

sm1=1;

delay(1000);

left++;

if(left<=3)

{

IN1=0;

IN2=0;

IN3=1;

IN4=0;

}

else if(left==4)

{

IN1=0;

IN2=1;

IN3=0;

IN4=0;

left=0;

}

delay(2000);

IN1=0;

IN2=0;

IN3=0;

IN4=0;

}

else

// if(range>40 ||range==0)

{

if(sm1==1)

{

IN1=0;

IN2=0;

IN3=0;

IN4=0;

sm1=0;

delay(100);

}

IN1=0;

IN2=1;

IN3=1;

IN4=0;

sm2=1;

}

// if(range<2 )

// {

// IN1=0;

// IN2=0;

// IN3=0;

// IN4=0;

// sm1=1;

// sm2=1;

// }

// //else

// if(range>40 ||range==0)

// {

// if(sm1==1)

// {

// IN1=0;

// IN2=0;

// IN3=0;

// IN4=0;

// sm1=0;

// delay(100);

// }

// IN1=0;

// IN2=1;

// IN3=1;

// IN4=0;

// sm2=1;

// }

// //else

// if(range<20 && range>2)

// {

// if(sm2==1)

// {

// IN1=0;

// IN2=0;

// IN3=0;

// IN4=0;

// sm2=0;

// delay(100);

// }

// IN1=1;

// IN2=0;

// IN3=0;

// IN4=1;

// sm1=1;

// delay(1000);

// left++;

//

// if(left<=3)

// {

// IN1=0;

// IN2=0;

// IN3=1;

// IN4=0;

// }

//

// else if(left==4)

// {

// IN1=0;

// IN2=1;

// IN3=0;

// IN4=0;

// left=0;

// }

//

// delay(2000);

// IN1=0;

// IN2=0;

// IN3=0;

// IN4=0;

//

// }

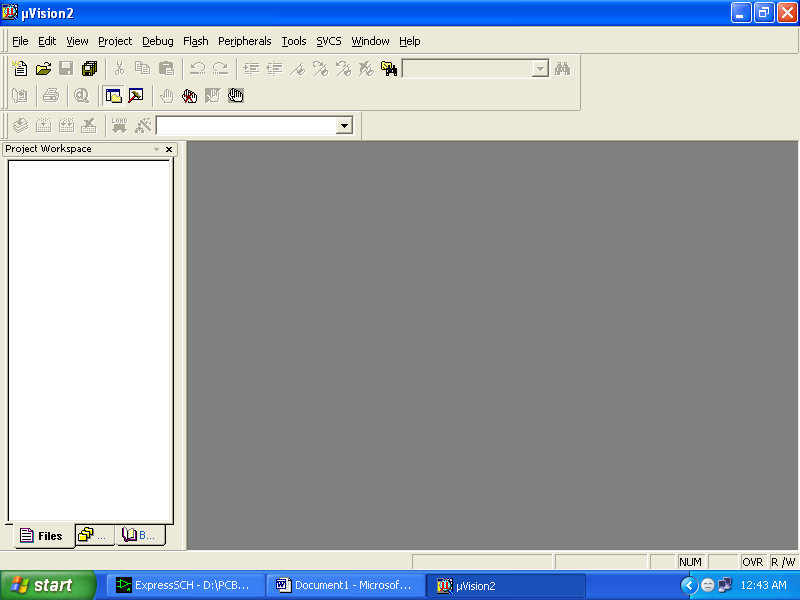
//

}

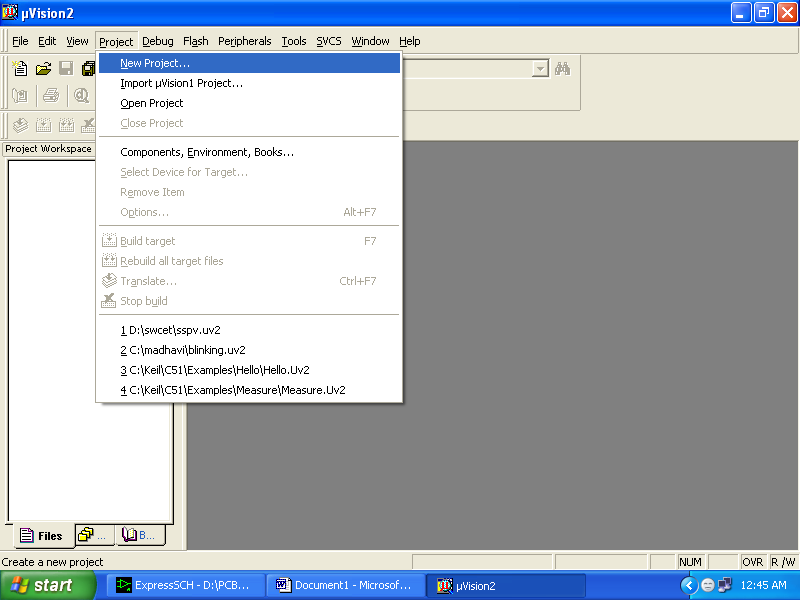
# 

# 9.1 PROGRAM CODE

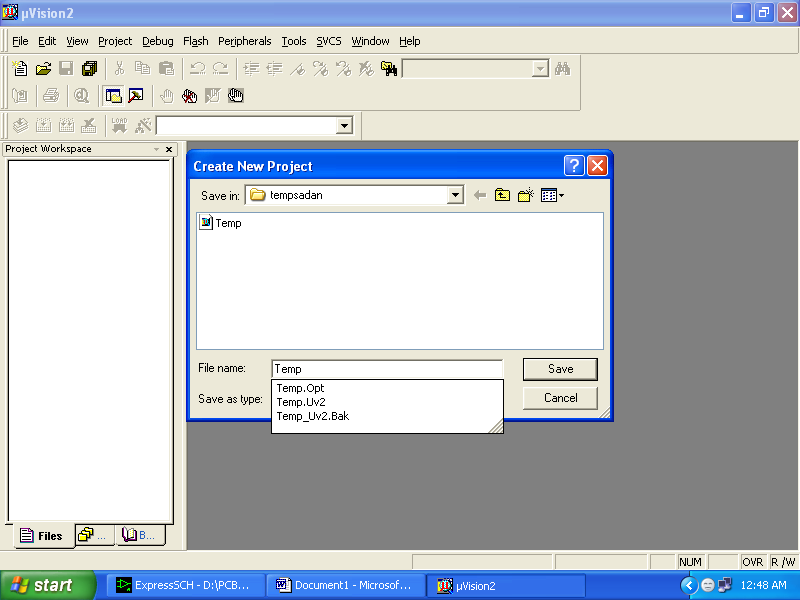
1. Click on the Keil Vision Icon on Desktop
2. The following fig will appear



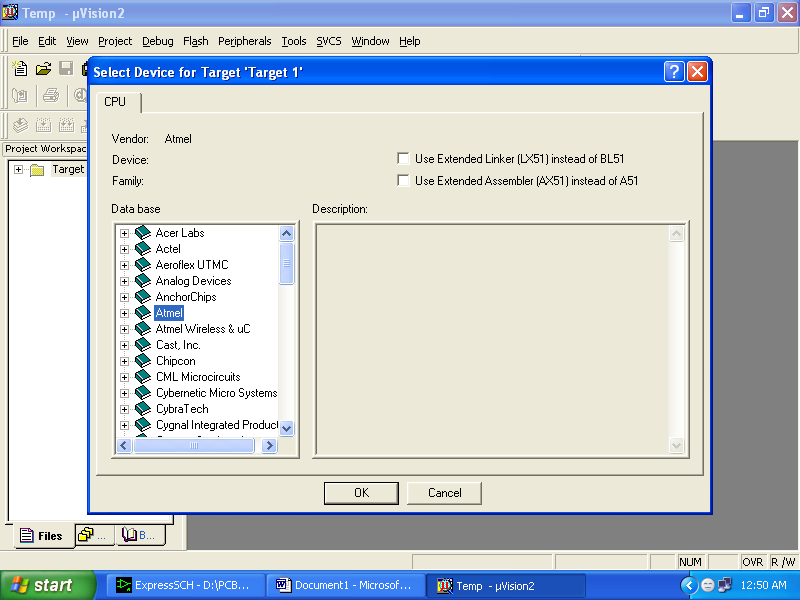
1. Click on the Project menu from the title bar
2. Then Click on New Project



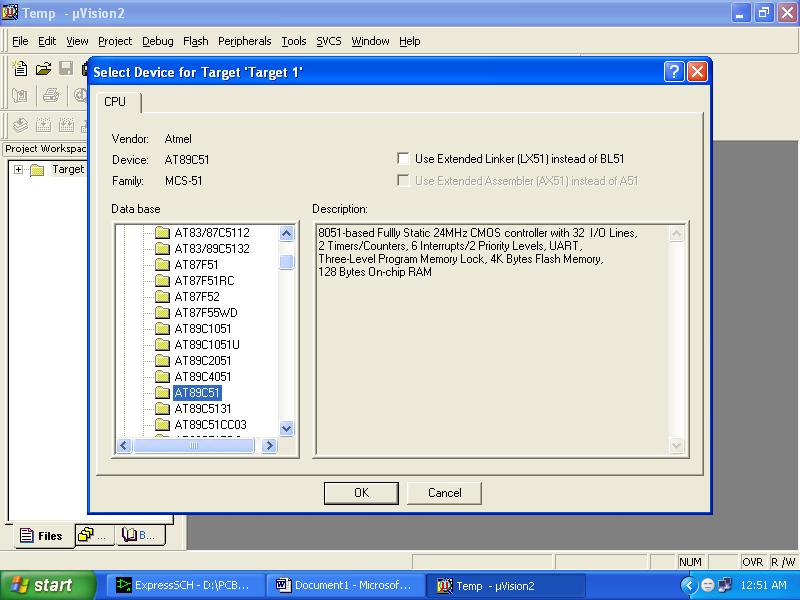
1. Save the Project by typing suitable project name with no extension in u r own folder sited in either C:\ or D:\



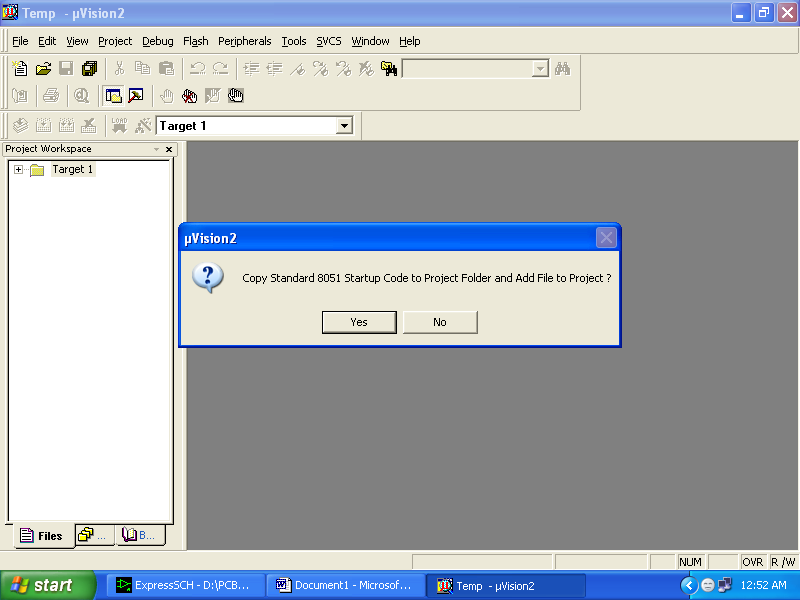
1. Then Click on Save button above.
2. Select the component for u r project. i.e. Atmel……
3. Click on the + Symbol beside of Atmel



1. Select AT89C51 as shown below



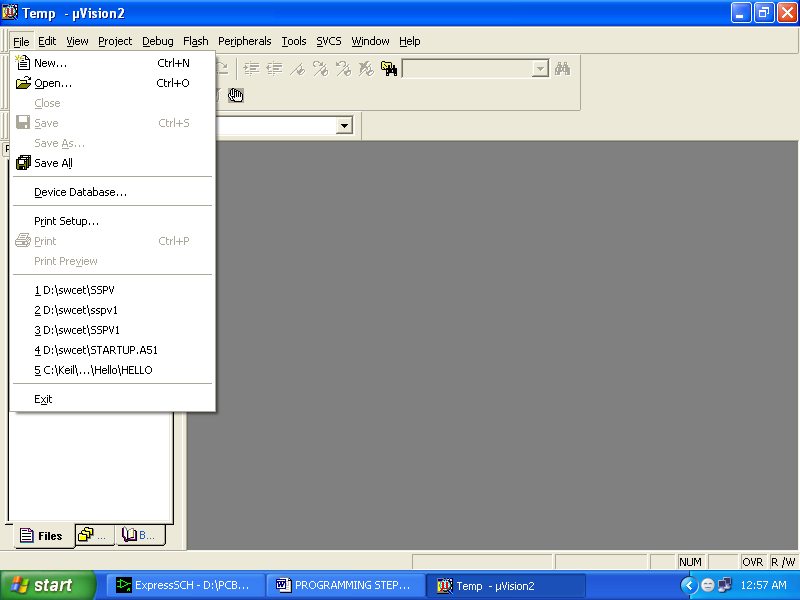
1. Then Click on “OK”
2. The Following fig will appear



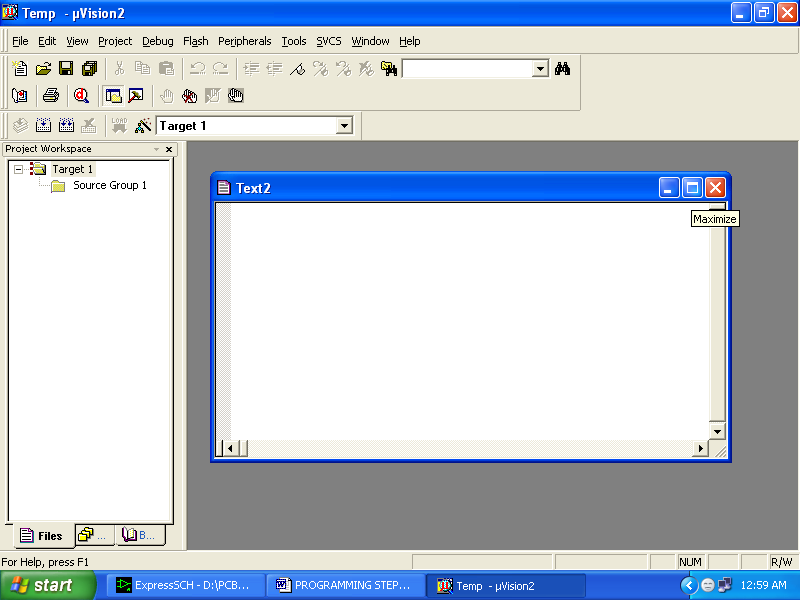
1. Then Click either YES or NO………mostly “NO”.
2. Now your project is ready to USE.
3. Now double click on the Target1, you would get another option “Source group 1” as shown in next page.



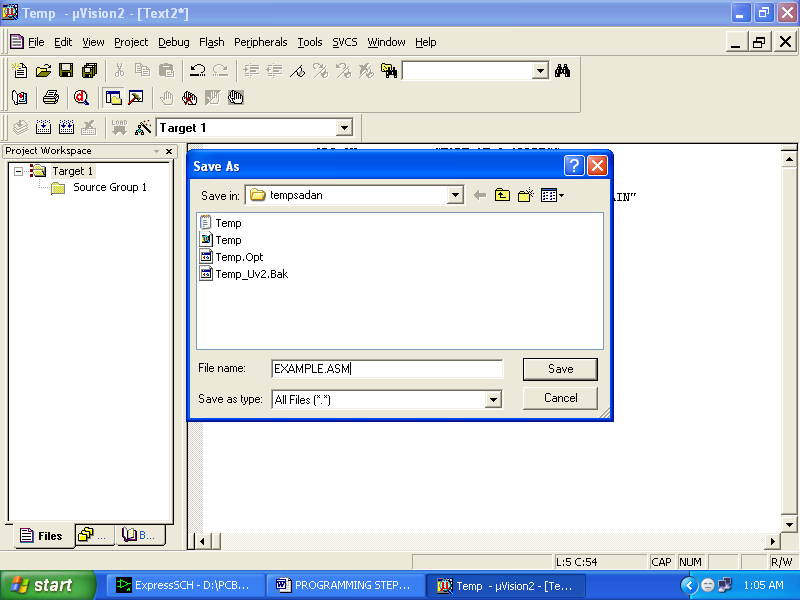
1. Click on the file option from menu bar and select “new”.



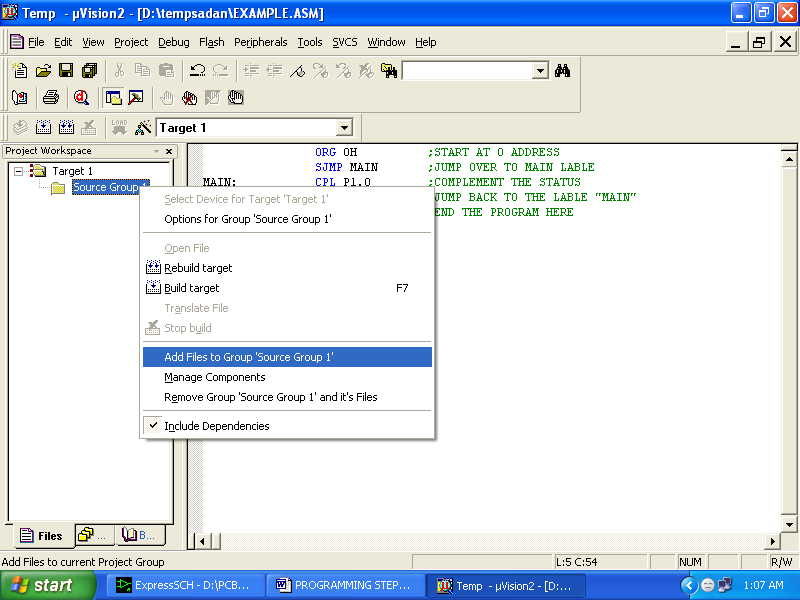
1. The next screen will be as shown in next page, and just maximize it by double clicking on its blue boarder.



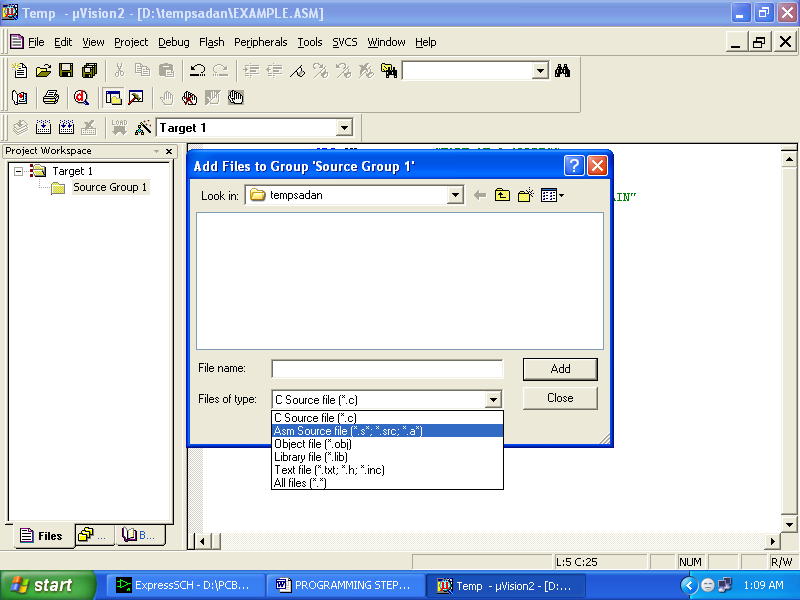
1. Now start writing program in either in “EMBEDDED C” or “ASM”.
2. For a program written in Assembly, then save it with extension “. asm” and for “EMBEDDED C” based program save it with extension “ .C”



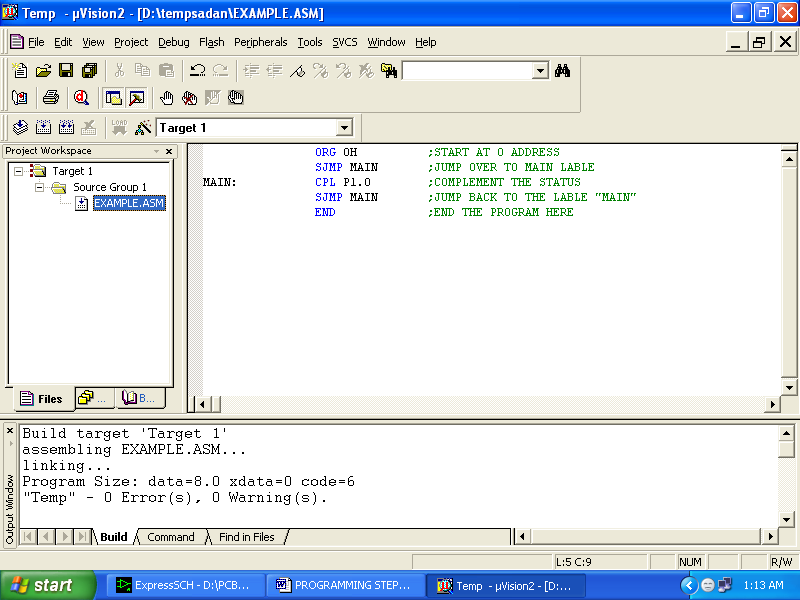
1. Now right click on Source group 1 and click on “Add files to Group Source”.



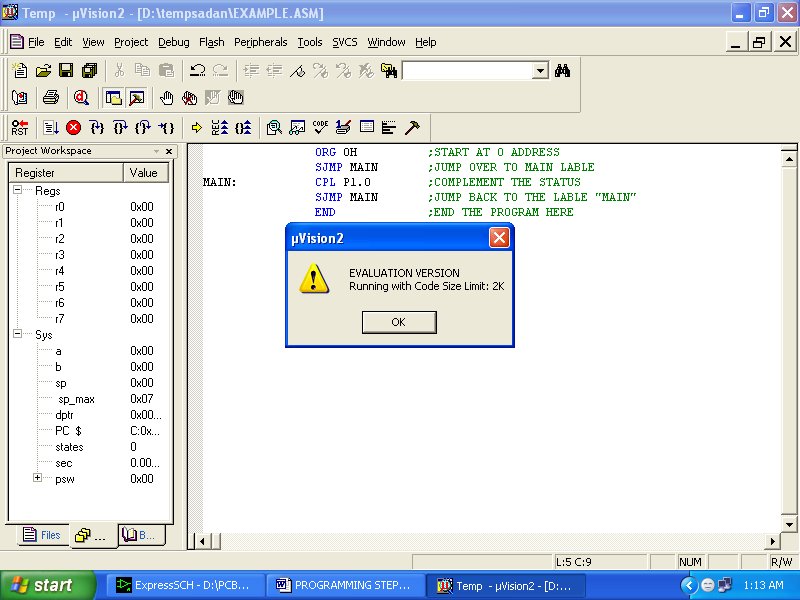
1. Now you will get another window, on which by default “EMBEDDED C” files will appear.



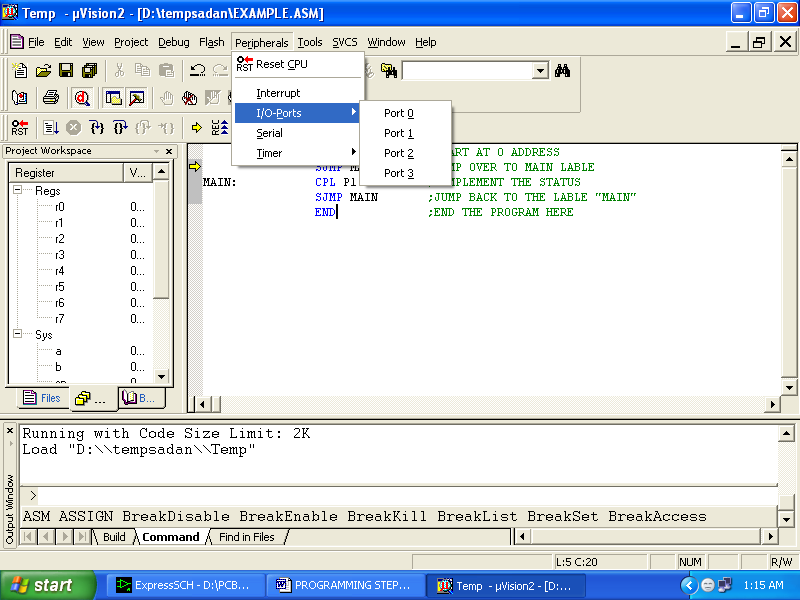
1. Now select as per your file extension given while saving the file
2. Click only one time on option “**ADD**”.
3. Now Press function key F7 to compile. Any error will appear if so happen.



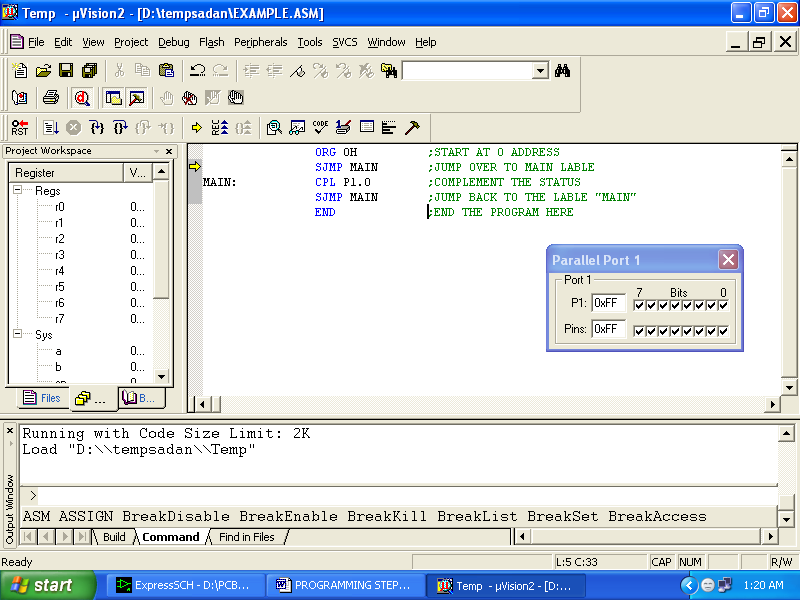
1. If the file contains no error, then press Control+F5 simultaneously.
2. The new window is as follows



1. Then Click “OK”.
2. Now click on the Peripherals from menu bar, and check your required port as shown in fig below.



1. Drag the port a side and click in the program file.



1. Now keep Pressing function key “F11” slowly and observe.
2. You are running your program successfully.

**10.HARDWARE TESTING**

**10.1 CONTINUITY TEST:**

In electronics, a continuity test is the checking of an electric circuit to see if current flows (that it is in fact a complete circuit). A continuity test is performed by placing a small voltage (wired in series with an LED or noise-producing component such as a piezoelectric speaker) across the chosen path. If electron flow is inhibited by broken conductors, damaged components, or excessive resistance, the circuit is "open".

Devices that can be used to perform continuity tests include multi meters which measure current and specialized continuity testers which are cheaper, more basic devices, generally with a simple light bulb that lights up when current flows.

An important application is the continuity test of a bundle of wires so as to find the two ends belonging to a particular one of these wires; there will be a negligible resistance between the "right" ends, and only between the "right" ends.

This test is the performed just after the hardware soldering and configuration has been completed. This test aims at finding any electrical open paths in the circuit after the soldering. Many a times, the electrical continuity in the circuit is lost due to improper soldering, wrong and rough handling of the PCB, improper usage of the soldering iron, component failures and presence of bugs in the circuit diagram. We use a multi meter to perform this test. We keep the multi meter in buzzer mode and connect the ground terminal of the multi meter to the ground. We connect both the terminals across the path that needs to be checked. If there is continuation then you will hear the beep sound.

**10.2 POWER ON TEST:**

This test is performed to check whether the voltage at different terminals is according to the requirement or not. We take a multi meter and put it in voltage mode. First of all check the voltage across the battery terminal whether it is fully charged or not, the battery used in this project is 6V, so touch the ‘red terminal’ of battery with ‘red probe’ of multi meter and touch ‘black terminal’ of battery with ‘black probe’ of multi meter, if 6V is being displayed on multi meter screen then we can proceed for next steps.

Now that the power supply is available, no IC should be inserted in the base, first apply power and check whether proper voltage is reaching at ‘vcc’ and ‘gnd’ pins of each IC base or not. If proper voltages appear at the supply pins of IC bases then insert IC and check the required output.

Now we have to check whether the LEDs are in working condition or not, Red LED or IR LED or Photo diode has got one ‘longer leg’ and one ‘shorter leg’. Longer leg is positive terminal of LED and shorter leg is negative terminal.

Now keep the multi meter in ‘buzzer mode or continuity mode’ and touch ‘red probe’ of multi meter to the ‘longer leg’ of LED and ‘black probe’ of multi meter to the ‘shorter leg’ of LED, if LED glows in such case that means it’s working.

Now solder Red LED into PCB, remember ‘longer leg’ of LED should be inserted into positive marking on PCB and ‘shorter leg’ should be inserted into other hole of LED marking on PCB. Now after soldering LED with a series resistor apply battery voltage to the board and see whether the LED is glowing or not.

The black LED is photodiode and white LED is IR diode even these components have got longer leg and shorter leg, insert longer leg into +ve marking hole on PCB and insert shorter leg into other hole of LED marking on PCB .

**11. RESULTS**

**12.CONCLUSION**

**13.BIBLIOGRAPHY**

**TEXT BOOKS REFERED:**

1. “The 8051 Microcontroller and Embedded systems” by Muhammad Ali Mazidi and Janice Gillispie Mazidi , Pearson Education.

2. ATMEL 89S52 Data Sheets.

**WEBSITES**

* www.atmel.com
* www.beyondlogic.org
* www.wikipedia.org
* www.howstuffworks.com
* www.alldatasheets.com