CHAPTER 1

ABSTRACT

In this paper we have shown the concept of Microcontroller Based Automatic car parking system. As we see in the modern world everything is going automatic we have built a system which will automatically sense the entry and exit of cars with the help of microcontroller through the gate and then display the number of cars in the parking lot. We have deployed a microcontroller 89S52 which is used to sense the movement of cars and depending upon whether there is a capacity of cars to enter, it either opens the gate or not

The goal of this paper is to automatically park the car for allowing the cars into the parking area. Two IR sensors TX RX pairs are used in this paper to identify the entry or exit of the cars into/out of park. These two IR sensor TX RX pairs are arranged either side of the gateThis project uses regulated 5V, 500mA power supply.. 7805 three terminal voltage regulator is used for voltage regulation. Bridge type full wave rectifier is used to rectify the ac out put of secondary of 230/12V step down transformer.

<u>Required components:</u>

1. AT89s52 uc
2. LDR
3. Opto coupler
4. Triac

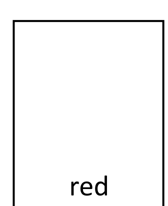
EXISTING SYSTEM

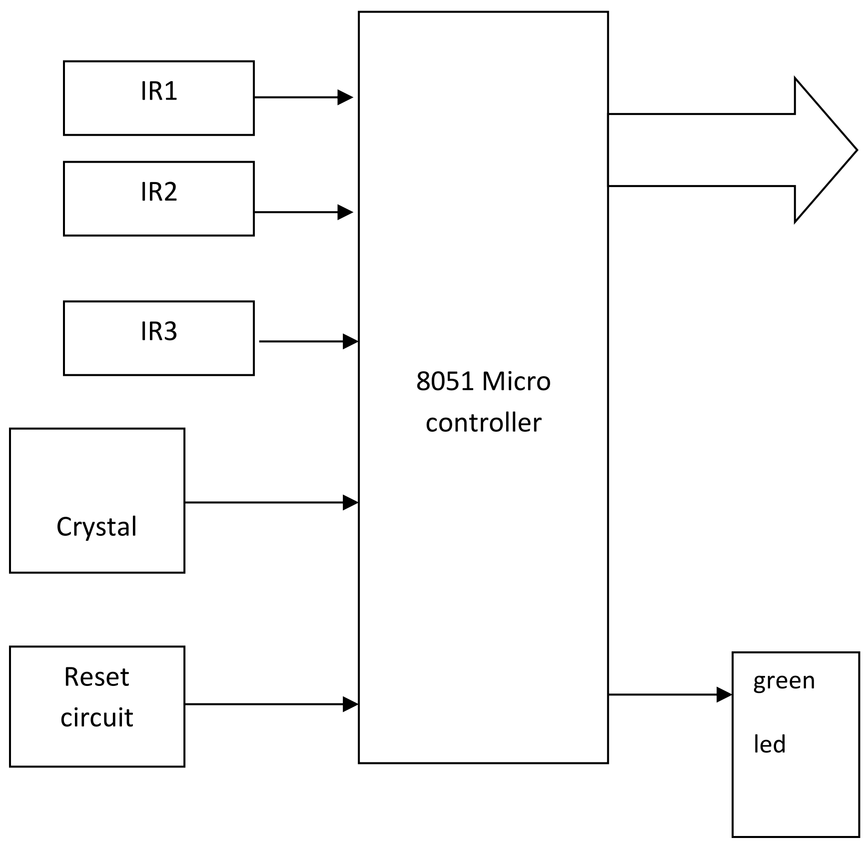
Past days a person needs to guide drivers to park the car but nowadays we are trying to automate this process. Which means without any guide of person the driver himself knows where the empty slot is there for parking. We are facing many problems for car parking system. As we need to park our car manually and there is no discipline in this process it creates a huge hurdle. People can park their cars anywhere they want to, which creates a mess as people don't follow a particular cue most of the time. As a result of this a huge traffic jam takes place in that place. While parking in and retrieving car due mismanagement cars can get dent by bumping with each other as there is lack of sufficient space. This23 leads to arguments, fights among people which sometimes create traffic jam. This is also an economical lose as we need to repair our damaged car. This chaos also leads to cars consuming extra fuel.

IV. PROPOSED SYSTEM

We proposed a method to automate the car parking system using IR transceiver which will guide the drivers for hassle free parking. By introducing the automated car parking systems we can handle the mismanagement of parking space save time and recover losses caused by the existing system and also earn money by charging money for car parking. V. WORKING PRINCIPLE First when a car enters the gate leds empty parking slot that empty parking slot was sensed by the IR sensor we are using object sensor to sense the car in the parking slot. The TX and RX-was kept in the parking slot. If any car parks at one slot the TX-transmits the signal and the RX can't receive any signal therefore,.

Block Diagram





Power Supply:

CHAPTER2

INTRODUCTION TO EMBEDDED SYSTEMS

An embedded system can be defined as a computing device that does a specific focused job. Appliances such as the air-conditioner, VCD player, DVD player, printer, fax machine, mobile phone etc. are examples of embedded systems. Each of these appliances will have a processor and special hardware to meet the specific requirement of the application along with the embedded software that is executed by the processor for meeting that specific requirement. The embedded software is also called "firm ware". The desktop/laptop computer is a general purpose computer. You can use it for a variety of applications such as playing games, *word* processing, accounting, software development and so on. In contrast, the software in the embedded systems is always fixed listed below:

· Embedded systems do a very specific task, they cannot be programmed to do different things. . Embedded systems have very limited resources, particularly the memory. Generally, they do not have secondary storage devices such as the CDROM or the floppy disk. Embedded systems have to work against some deadlines. A specific job has to be completed within a specific time. In some embedded systems, called real-time systems, the deadlines are stringent. Missing a deadline may cause a catastrophe-loss of life or damage to property. Embedded systems are constrained for power. As many embedded systems operate through a battery, the power consumption has to be very low.

· Some embedded systems have to operate in extreme environmental conditions such as very high temperatures and humidity.

Application Areas

Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication,

data communication, telecommunications, transportation, military and so on.

Consumer appliances: At home we use a number of embedded systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air-conditioner, VCO player, video game consoles, video recorders etc. Today’s high-tech car has about 20 embedded systems for transmission control, engine spark control, air-conditioning, navigation etc. Even wristwatches are now

becoming embedded systems. The palmtops are powerful embedded systems using which we can carry out many general-purpose tasks such as playing games and word processing.

Office automation: The office automation products using em embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.

Industrial automation: Today a lot of industries use embedded systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.

Medical electronics: Almost every medical equipment in the hospital is an embedded system. These equipments include diagnostic aids such as ECG, EEG, blood pressure measuring devices, X-ray scanners; equipment used in blood analysis, radiation, colonscopy, endoscopy etc. Developments in medical electronics have paved way for more accurate diagnosis of diseases.

Computer networking: Computer networking products such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM), X.25 and frame relay switches are embedded systems which implement the necessary data communication protocols. For example, a router interconnects two networks. The two networks may be running different protocol stacks. The router's function is to obtain the data packets from incoming pores, analyze the packets and send them towards the destination after doing necessary protocol conversion. Most networking equipments, other than the end systems (desktop computers) we use to access the networks, are embedded systems

.

Telecommunications: In the field of telecommunications, the embedded systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems. The network equipment includes multiplexers, multiple access systems, Packet Assemblers Dissemblers (PADs), sate11ite modems etc. IP phone, IP gateway, IP gatekeeper etc. are the latest embedded systems that provide very low-cost voice communication over the Internet.

Wireless technologies: Advances in mobile communications are paving way for many interesting applications using embedded systems. The mobile phone is one of the marvels of the last decade of the 20'h century. It is a very powerful embedded system that provides voice communication while we are on the move. The Personal Digital Assistants and the palmtops can now be used to access multimedia services over the Internet. Mobile communication infrastructure such as base station controllers, mobile switching centers are also powerful embedded systems.

Insemination: Testing and measurement are the fundamental requirements in all scientific and engineering activities. The measuring equipment we use in laboratories to measure parameters such as weight, temperature, pressure, humidity, voltage, current etc. are all embedded systems. Test equipment such as oscilloscope, spectrum analyzer, logic analyzer, protocol analyzer, radio communication test set etc. are embedded systems built around powerful processors. Thank to miniaturization, the test and measuring equipment are now becoming portable facilitating easy testing and measurement in the field by field-personnel.

Security: Security of persons and information has always been a major issue. We need to protect our homes and offices; and also the information we transmit and store. Developing embedded systems for security applications is one of the most lucrative businesses nowadays. Security devices at homes, offices, airports etc. for authentication and verification are embedded systems. Encryption devices are nearly 99 per cent of

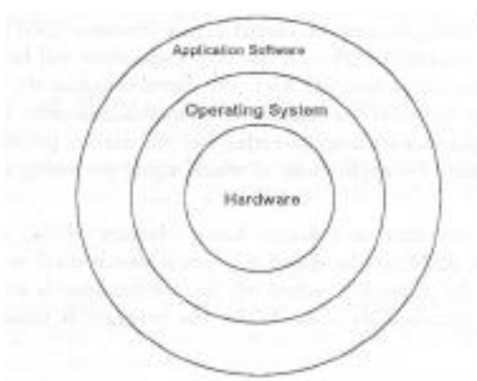
the processors that are manufactured end up in~ embedded systems. Embedded systems find applications in . every industrial segment- consumer electronics, transportation, avionics, biomedical engineering, manufacturing, process control and industrial automation, data communication, telecommunication, defense, security etc. Used to encrypt the data/voice being transmitted on communication links such as telephone lines. Biometric systems using fingerprint and face recognition are now being extensively used for user authentication in banking applications as well as for access control in high security buildings.

Finance: Financial dealing through cash and cheques are now slowly paving way for transactions using smart cards and ATM (Automatic Teller Machine, also expanded as Any Time Money) machines. Smart card, of the size of a credit card, has a small micro-controller and memory; and it interacts with the smart card reader! ATM machine and acts as an electronic wallet. Smart card technology has the capability of ushering in a cashless society. Well, the list goes on. It is no exaggeration to say that eyes wherever you go, you can see, or at least feel, the work of an embedded system!

Overview of Embedded System Architecture

Every embedded system consists of custom-built hardware built around a Central Processing Unit (CPU). This hardware also contains memory chips onto which the software is loaded. The software residing on the memory chip is also called the 'firmware'. The embedded system architecture can be represented as a layered architecture as shown in Fig.

The operating system runs above the hardware, and the application software runs above



Layered Architecture of an Embedded Syste

the operating system. The same architecture is applicable to any computer including a desktop computer. However, there are significant differences. It is not compulsory to have an operating system in every embedded system. For small appliances such as remote control units, air conditioners, toys etc., there is no need *for* an operating system and you can write only the software specific to that application. For applications involving complex processing, it is advisable to have an operating system. In such a case, you need to integrate the application software with the operating system and then transfer the entire software on to the memory chip. Once the software is transferred to the memory chip, the software will continue to run *for* a long time you don't need to reload new software.

Now, let us see the details of the various building blocks of the hardware of an embedded system. As shown in Fig. the building blocks are;

· Central Processing Unit (CPU)

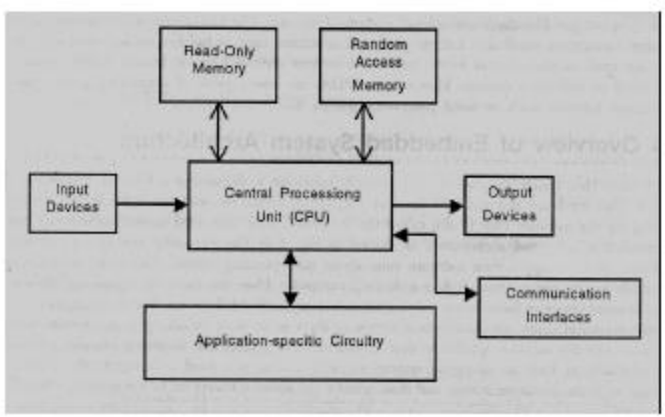
· Memory (Read-only Memory and Random Access Memory)

· Input Devices

· Output devices

· Communication interfaces

· Application-specific circuitry



Central Processing Unit (CPU):

The Central Processing Unit (processor, in short) can be any of the following: microcontroller, microprocessor or Digital Signal Processor (DSP). A micro-controller is a low-cost processor. Its main attraction is that on the chip itself, there will be many other components such as memory, serial communication interface, analog-to digital converter etc. So, for small applications, a micro-controller is the best choice as the number of external components required will be very less. On the other hand, microprocessors are more powerful, but you need to use many external components with them. D5P is used mainly for applications in which signal processing is involved such as audio and video processing.

Memory:

The memory is categorized as Random Access 11emory (RAM) and Read Only Memory (ROM). The contents of the RAM will be erased if power is switched off to the chip, whereas ROM retains the contents even if the power is switched off. So, the firmware is stored in the ROM. When power is switched on, the processor reads the ROM; the program is program is executed.

Input devices:

Unlike the desktops, the input devices to an embedded system have very limited capability. There will be no keyboard or a mouse, and hence interacting with the embedded system is no easy task. Many embedded systems will have a small keypad-you press one key to give a specific command. A keypad may be used to input only the digits. Many embedded systems used in process control do not have any input device *for* user interaction; they take inputs *from* sensors or transducers 1'fnd produce electrical signals that are in turn fed to other systems.

Output devices:

The output devices of the embedded systems also have very limited capability. Some embedded systems will have a *few* Light Emitting Diodes (LEDs) *to* indicate the health status of the system modules, or *for* visual indication of alarms. A small Liquid Crystal Display (LCD) may also be used to display *some* important parameters.

Communication interfaces:

The embedded systems may need to, interact with other embedded systems at they may have to transmit data to a desktop. To facilitate this, the embedded systems are provided with one or a *few* communication interfaces such as RS232, RS422, RS485, Universal Serial Bus (USB), IEEE 1394, Ethernet etc.

Application-specific circuitry:

Sensors, transducers, special processing and control circuitry may be required fat an embedded system, depending on its application. This circuitry interacts with the processor to carry out the necessary work. The entire hardware has to be given power supply either through the 230 volts main supply or through a battery. The hardware has to design in such a way that the power consumption is minimized.

CHAPTER 3

Hardware Implementation of the Project

This chapter briefly explains about the Hardware Implementation of the project. It discusses the design and working of the design with the help of block diagram in detail. It explains the features, timer programming, serial communication, interrupts of ARM7 microcontroller. It also explains the various modules used in this project.

3.1 Project Design

The implementation of the project design can be divided in two parts.

- Hardware implementation
- Firmware implementation

Hardware implementation deals in drawing the schematic on the plane paper according to the application, testing the schematic design over the breadboard using the various IC's to find if the design meets the objective, carrying out the PCB layout of the schematic tested on breadboard, finally preparing the board and testing the designed hardware.

The firmware part deals in programming the microcontroller so that it can control the operation of the IC's used in the implementation. In the present work, we have used the Orcad design software for PCB circuit design, the Keil µv4 software development tool to write and compile the source code, which has been written in the C language. The Proload programmer has been used to write this compile code into the microcontroller. The firmware implementation is explained in the next chapter.

The project design and principle are explained in this chapter using the block diagram and circuit diagram. The block diagram discusses about the required components of the design and working condition is explained using circuit diagram and system wiring diagram.

INTRODUCTION TO MICROCONTROLLER

Based on the Processor side Embedded Systems is mainly divided into 3 types

1. Micro Processor : - are for general purpose eg: our personal computer

2. Micro Controller:- are for specific applications, because of cheaper cost we will go for these

3. DSP (Digital Signal Processor):- are for high and sensitive application purpose

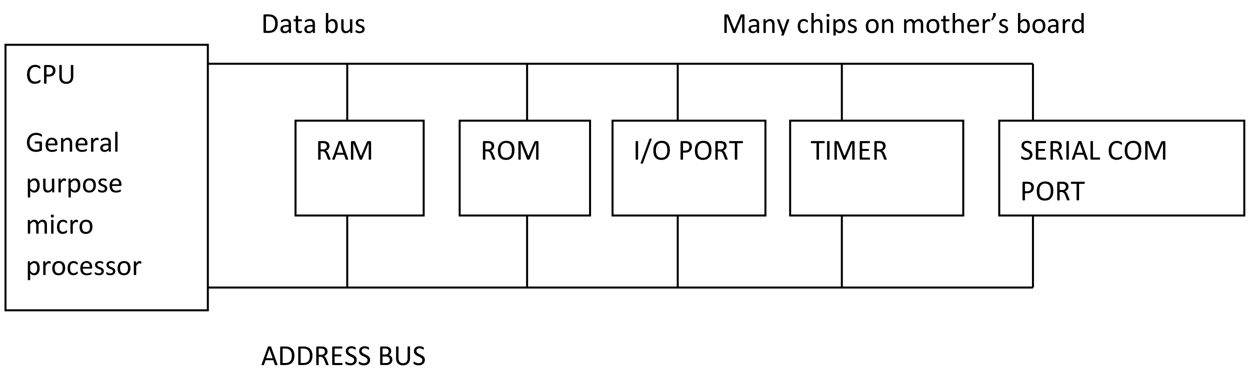
MICROCONTROLLER VERSUS MICROPROCESSOR

A system designer using a general-purpose microprocessor such as the Pentium or the 68040 must add RAM, ROM, I/O ports, and timers externally to make them functional. Although the addition of external RAM, ROM, and I/O ports makes these systems bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/O ports needed to fit the task at hand.

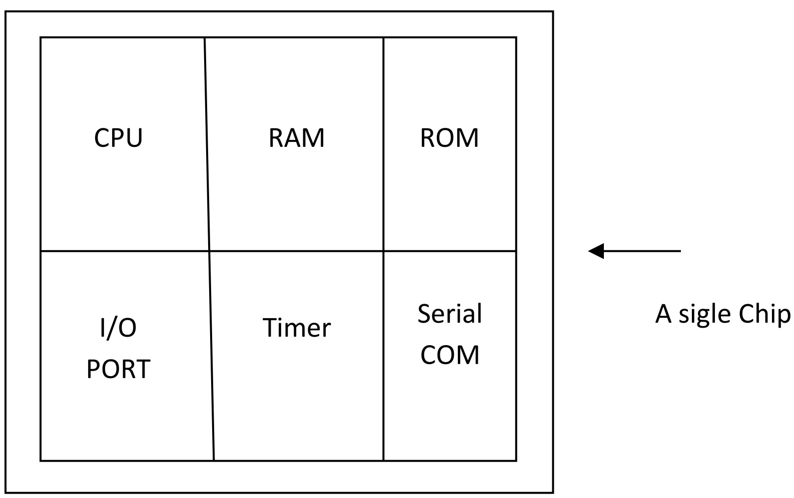
A Microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. In other words, the processor, the RAM, ROM, I/O ports and the timer are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O ports, or timer to it. The fixed amount of on-chip ROM, RAM, and number of I/O ports in Microcontrollers makes them ideal for many applications in which cost and space are critical.

General Micro Processor

1. cpu for computers
2. No RAM, ROM, I/O on CPU chip itself
3. Example : Intel's x86



Micro Controller



A smaller Computer

On chip RAM, ROM, I/O PORTS……..

Example: Intel 8052 etc….

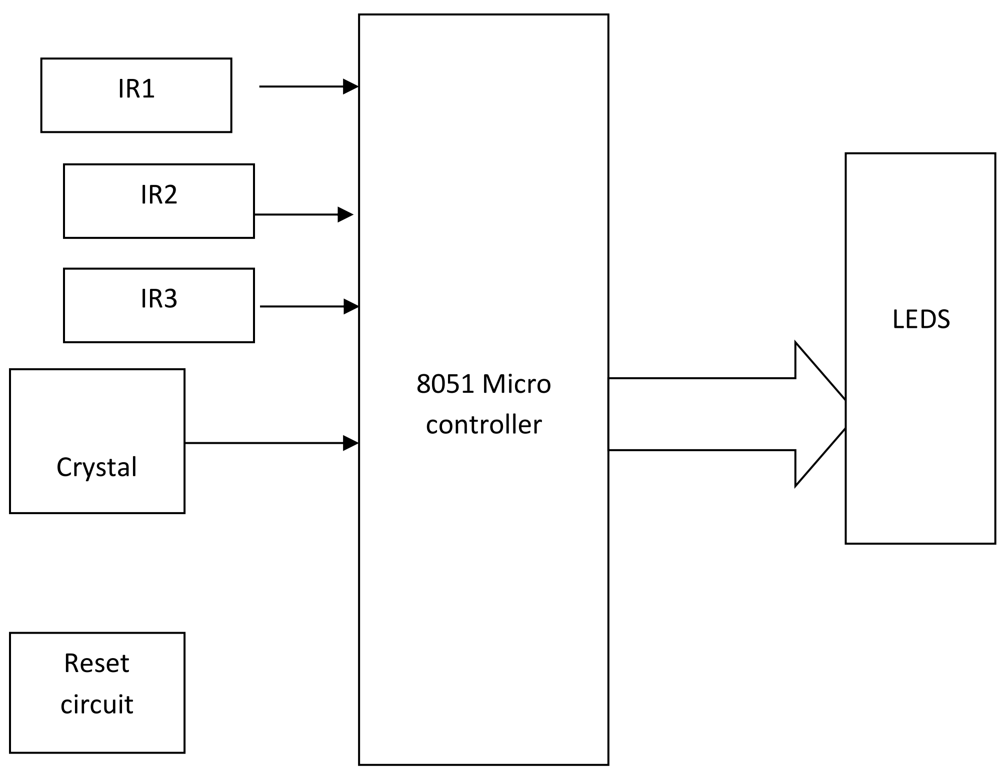
Microprocessor vs. Microcontroller	
Microprocessor	**Microcontroller**

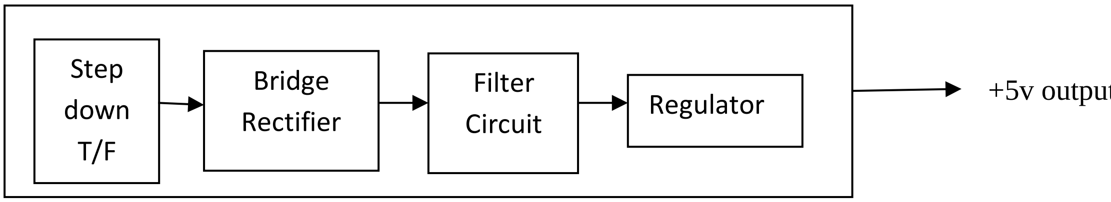
CPU is stand alone RAM, RAM, I/O, timer are separate	CPU,RAM,ROM,I/O and timer are all on a single chip
Designer can decide on the amount of ROM, RAM and I/O ports.	Fix amount of on chip ROM,RAM, I/O Ports.
Expansive, Versatility	For applications in which cost, power and space are critical
General purpose	Single purpose

3.1.1 Block Diagram of the Project and its Description

The block diagram of the design is as shown in Fig 3.1. It consists of power supply unit, microcontroller, sensor module, LCD. The brief description of each unit is explained as follows.

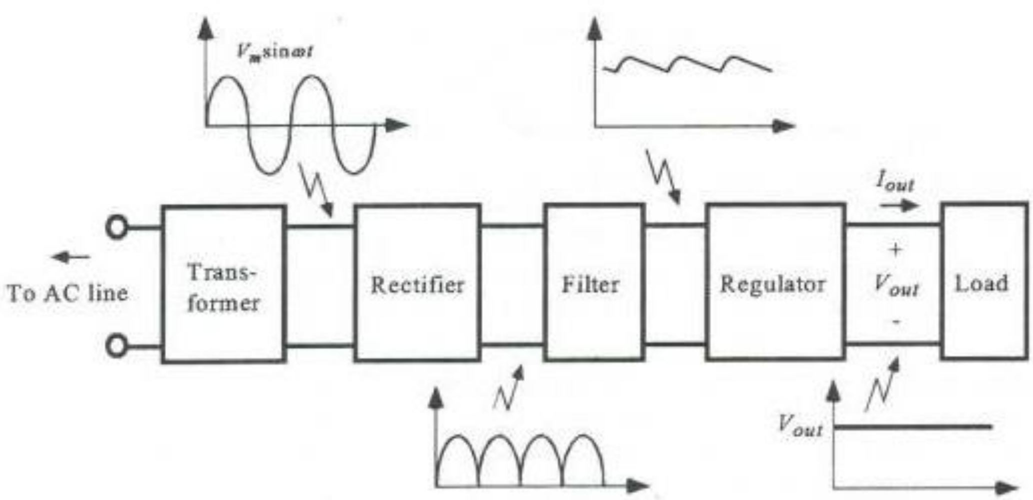
Block Diagram





3.2 Power Supply:

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.



Components of a regulated power supply

Transformer:

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus the a.c input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step down transformer is employed to decrease the voltage to a required level.

Rectifier:

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

Filter:

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore a regulator is applied at the output stage.

Voltage regulator:

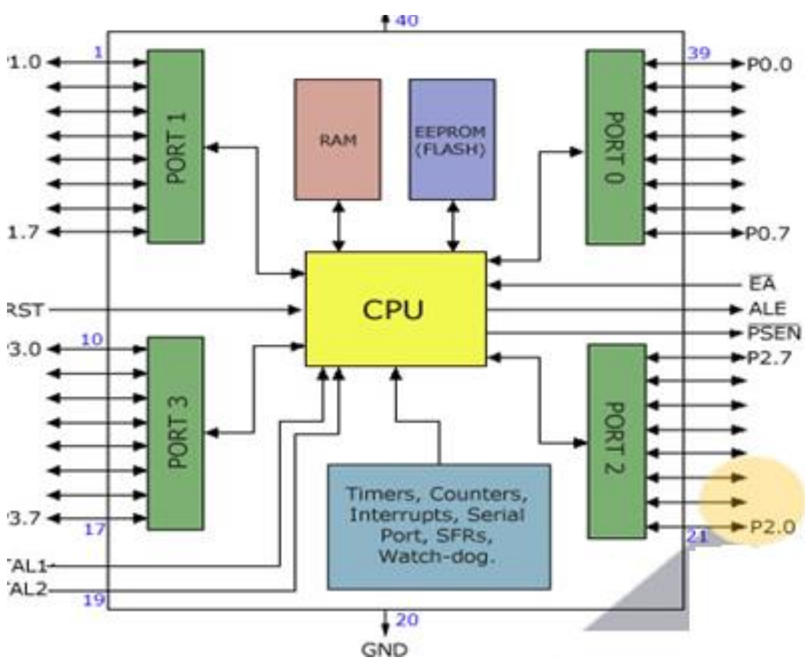
As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels.

3.3 Microcontrollers:

Microprocessors and microcontrollers are widely used in embedded systems products. Microcontroller is a programmable device. A microcontroller has a CPU in addition to a fixed amount of RAM, ROM, I/O ports and a timer embedded all on a single chip. The fixed amount of on-chip ROM, RAM and number of I/O ports in microcontrollers makes them ideal for many applications in which cost and space are critical.

The Intel 8052 is Harvard architecture, single chip microcontroller (µC) which was developed by Intel in 1980 for use in embedded systems. It was popular in the 1980s and early 1990s, but today it has largely been superseded by a vast range of enhanced devices with 8051-compatible processor cores that are manufactured by more than 20 independent manufacturers including Atmel, Infineon Technologies and Maxim Integrated Products.

8052 is an 8-bit processor, meaning that the CPU can work on only 8 bits of data at a time. Data larger than 8 bits has to be broken into 8-bit pieces to be processed by the CPU. 8052 is available in different memory types such as UV-EPROM, Flash and NV-RAM.



Features of AT89S52:

- 8K Bytes of Re-programmable Flash Memory.
- RAM is 256 bytes.
- 4.0V to 5.5V Operating Range.
- Fully Static Operation: 0 Hz to 33 MHz’s
- 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).

Description:

The AT89s52 is a low-voltage, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable memory. The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set. 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 Flash on a monolithic chip, the Atmel AT89s52 is a powerful microcomputer, which provides a highly flexible and cost-effective solution to many embedded control applications.

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 hardware reset.

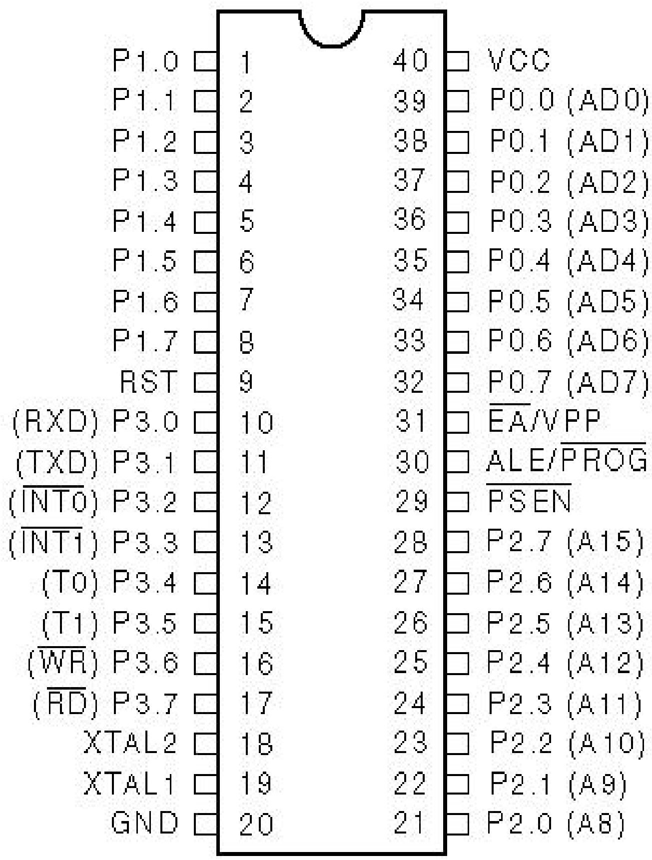


Fig: Pin diagram

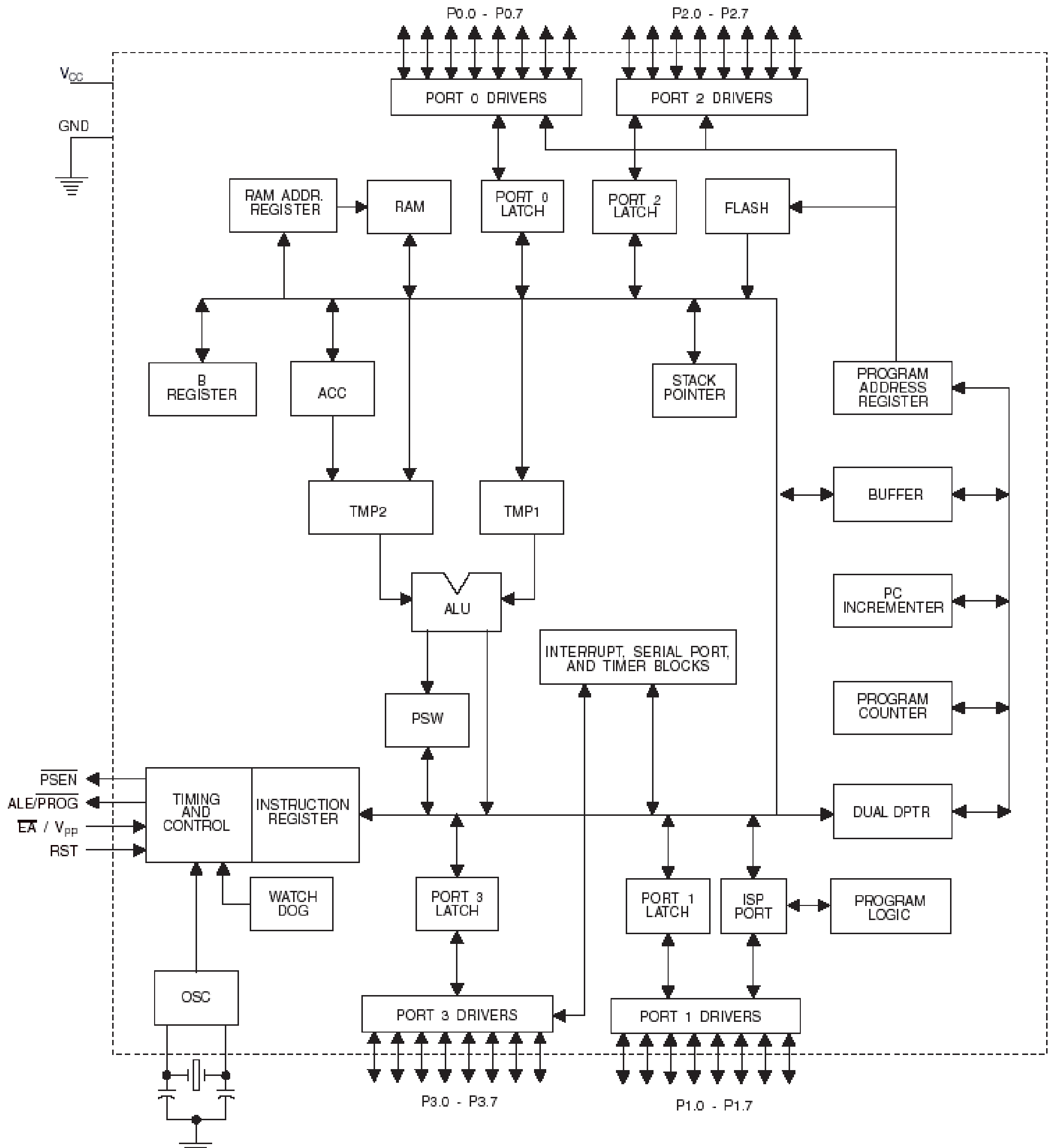


Fig: Block diagram

Pin description:

Vcc Pin 40 provides supply voltage to the chip. The voltage source is +5V.

GND Pin 20 is the 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), respectively, as shown in the following table.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

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 uses 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. The port also receives the high-order address bits and some control signals during Flash programming and verification.

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. Port 3 receives some control signals for Flash programming and verification.

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

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.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

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.

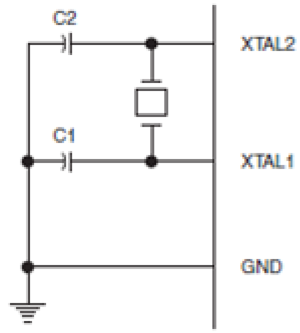
XTAL1

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

XTAL2

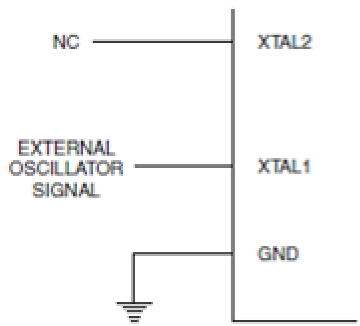
Output from the inverting oscillator amplifier.

Oscillator Connections



C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

External Clock Drive Configuration



XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator. 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. 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

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in the following table.

It should be noted that not all of the addresses are occupied and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers:

Control and status bits are contained in registers T2CON and T2MOD for Timer 2. The register pair (RCAP2H, RCAP2L) is the Capture/Reload register for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Interrupt Registers:

The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

Dual Data Pointer Registers:

To facilitate accessing both internal and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H and 85H. Bit DPS = 0 in SFR AUXR1 selects DP0 and DPS = 1 selects DP1. The user should ALWAYS initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

Power off Flag:

The Power off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and rest under software control and is not affected by reset.

Memory Organization

MCS-51 devices have a separate address space for Program and Data Memory. Up to 64K bytes each of external Program and Data Memory can be addressed.

Program Memory

If the EA pin is connected to GND, all program fetches are directed to external memory. On the AT89S52, if EA is connected to VCC, program fetches to addresses 0000H through 1FFFH are directed to internal memory and fetches to addresses 2000H through FFFFH are to external memory.

Data Memory

The AT89S52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. This means that the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions which use direct addressing access the SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

MOV 0A0H, #data

The instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

MOV @R0, #data

It should be noted that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

Watchdog Timer (One-time Enabled with Reset-out)

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 14-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable from exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H).

When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

UART

The Atmel 8051 Microcontrollers implement three general purpose, 16-bit timers/ counters. They are identified as Timer 0, Timer 1 and Timer 2 and can be independently configured to operate in a variety of modes as a timer or as an event counter. When operating as a timer, the timer/counter runs for a programmed length of time and then issues an interrupt request. When operating as a counter, the timer/counter counts negative transitions on an external pin. After a preset number of counts, the counter issues an interrupt request. The various operating modes of each timer/counter are described in the following sections.

A basic operation consists of timer registers THx and TLx (x= 0, 1) connected in cascade to form a 16-bit timer. Setting the run control bit (TRx) in TCON register turns the timer on by allowing the selected input to increment TLx. When TLx overflows it increments THx; when THx overflows it sets the timer overflow flag (TFx) in TCON register. Setting the TRx does not clear the THx and TLx timer registers. Timer registers can be accessed to obtain the current count or to enter preset values. They can be read at any time but TRx bit must be cleared to preset their values, otherwise the behavior of the timer/counter is unpredictable.

The C/T control bit (in TCON register) selects timer operation or counter operation, by selecting the divided-down peripheral clock or external pin Tx as the source for the counted signal. TRx bit must be cleared when changing the mode of operation, otherwise the behavior of the timer/counter is unpredictable. For timer operation (C/Tx# = 0), the timer register counts the divided-down peripheral clock. The timer register is incremented once every peripheral cycle (6 peripheral clock periods). The timer clock rate is FPER / 6, i.e. FOSC / 12 in standard mode or FOSC / 6 in X2 mode. For counter operation (C/Tx# = 1), the timer register counts the negative transitions on the Tx external input pin. The external input is sampled every peripheral cycle. When the sample is high in one cycle and low in the next one, the counter is incremented.

Since it takes 2 cycles (12 peripheral clock periods) to recognize a negative transition, the maximum count rate is FPER / 12, i.e. FOSC / 24 in standard mode or FOSC / 12 in X2 mode. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full peripheral cycle. In addition to the “timer” or “counter” selection, Timer 0 and Timer 1 have four operating modes from which to select which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1and 2 are the same for both timer/counters. Mode 3 is different.

The four operating modes are described below. Timer 2, has three modes of operation: ‘capture’, ‘auto-reload’ and ‘baud rate generator’.

Timer 0

Timer 0 functions as either a timer or event counter in four modes of operation. Timer 0 is controlled by the four lower bits of the TMOD register and bits 0, 1, 4 and 5 of the TCON register. TMOD register selects the method of timer gating (GATE0), timer or counter operation (T/C0#) and mode of operation (M10 and M00). The TCON register provides timer 0 control functions: overflow flag (TF0), run control bit (TR0), interrupt flag (IE0) and interrupt type control bit (IT0).

For normal timer operation (GATE0= 0), setting TR0 allows TL0 to be incremented by the selected input. Setting GATE0 and TR0 allows external pin INT0# to control timer operation.

Timer 0 overflow (count rolls over from all 1s to all 0s) sets TF0 flag, generating an interrupt request. It is important to stop timer/counter before changing mode.

Mode 0 (13-bit Timer)

Mode 0 configures timer 0 as a 13-bit timer which is set up as an 8-bit timer (TH0 register) with a modulo-32 prescaler implemented with the lower five bits of the TL0 register. The upper three bits of TL0 register are indeterminate and should be ignored. Prescaler overflow increments the TH0 register.

As the count rolls over from all 1's to all 0's, it sets the timer interrupt flag TF0. The counted input is enabled to the Timer when TR0 = 1 and either GATE = 0 or INT0 = 1. (Setting GATE = 1 allows the Timer to be controlled by external input INT0, to facilitate pulse width measurements). TR0 is a control bit in the Special Function register TCON. GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH0 and the lower 5 bits of TL0. The upper 3 bits of TL0 are indeterminate and should be ignored. Setting the run flag (TR0) does not clear the registers.

Mode 0 operation is the same for Timer 0 as for Timer 1. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3)

Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON. Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/T2 = 0). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536\text{-}RCAP2H, RCAP2L)]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in the below figure. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus, when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

It should be noted that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Interrupts

The AT89S52 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2) and the serial port interrupt. These interrupts are all shown in the below figure.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once. The below table shows that bit position IE.6 is unimplemented. User software should not write a 1 to this bit position, since it may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

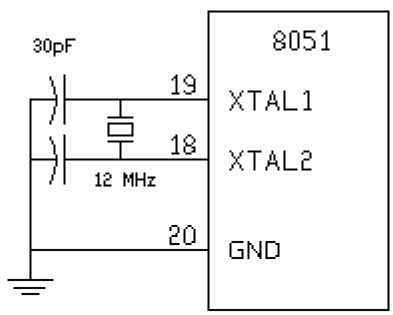
CRYSTAL OSCILLATOR

The 8051 uses the crystal for precisely that: to synchronize it’s operation. Effectively, the 8051 operates using what are called "machine cycles." A single machine cycle is the minimum amount of time in which a single 8051 instruction can be executed. Although many instructions take multiple cycles. 8051 has an on-chip oscillator. It needs an external crystal that decides the operating frequency of the 8051. The crystal is connected to pins 18 and 19 with stabilizing capacitors. 12 MHz (11.059MHz) crystal is often used and the capacitance ranges from 20pF to 40pF.

A cycle is, in reality, 12 pulses of the crystal. That is to say, if an instruction takes one machine cycle to execute, it will take 12 pulses of the crystal to execute. Since we know the we can calculate how many instruction cycles the 8051 can execute per second:

11,059,000 / 12 = 921,583

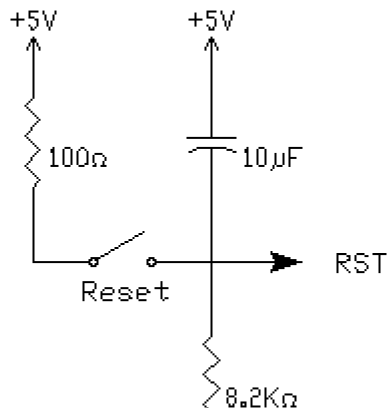
11.0592 MHz crystals are often used because it can be divided to give you exact clock rates for most of the common baud rates for the UART, especially for the higher speeds (9600, 19200).



Crystal

Reset

RESET is an active High input When RESET is set to High, 8051 goes back to the power on state.The 8051 is reset by holding the RST high for at least two machine cycles and then returning it low. Initially charging of capacitor makes RST High, When capacitor charges fully it blocks DC.



SIP Resistor

Sip Resistor is a single in pack Resistor (i.e.,) 8 resistors connected in series. Basically SIP resistor is a 9 pin connector first pin is for power supply to the entire 8 resistors in SIP.

Generally SIP Resistor is used to close the open drain connections of Port 0.

INFRARED SENSORS

Introduction

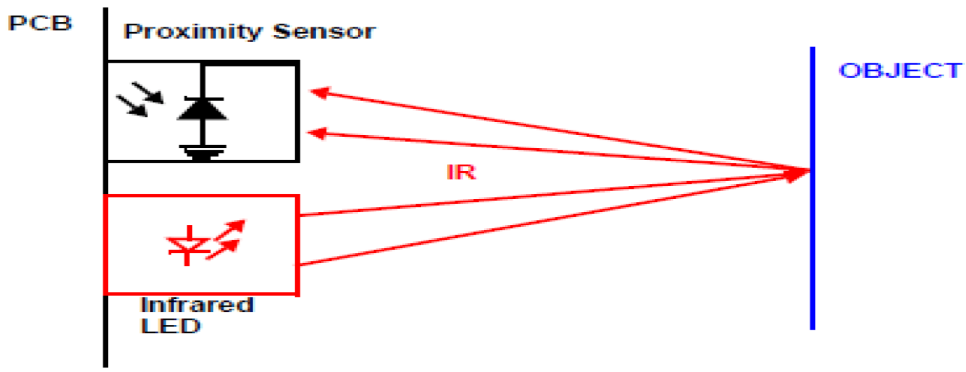
IR transceiver is used here for determining the arrival and departure of train. This is done by using IR Transceiver inwhich presence of train is detected as logical zero.

A. Transmitter

The Infrared Emitting Diode (IR333/H0/L10) is a high intensity diode, molded in a blue transparent package. Thedevice is spectrally matched with phototransistor, photodiode and IR receiver module. It finds applications in IR remotecontrol units, smoke detectors, free air transmission systems etc.

B. Receiver

The IR LED converts the incident IR radiations to an equivalent electric current which when passed through aresistor results in a certain amount of voltage drop. This value of voltage will depend upon the intensity of incident IRradiations or in other words, the distance between IR transmitter and receiver. The receiver is connected in reverse biasn the circuit. The IR rays emitted by the transmitter get reflected back after hitting the target. Receiver coverts thisreceived radiations to a corresponding electric current.



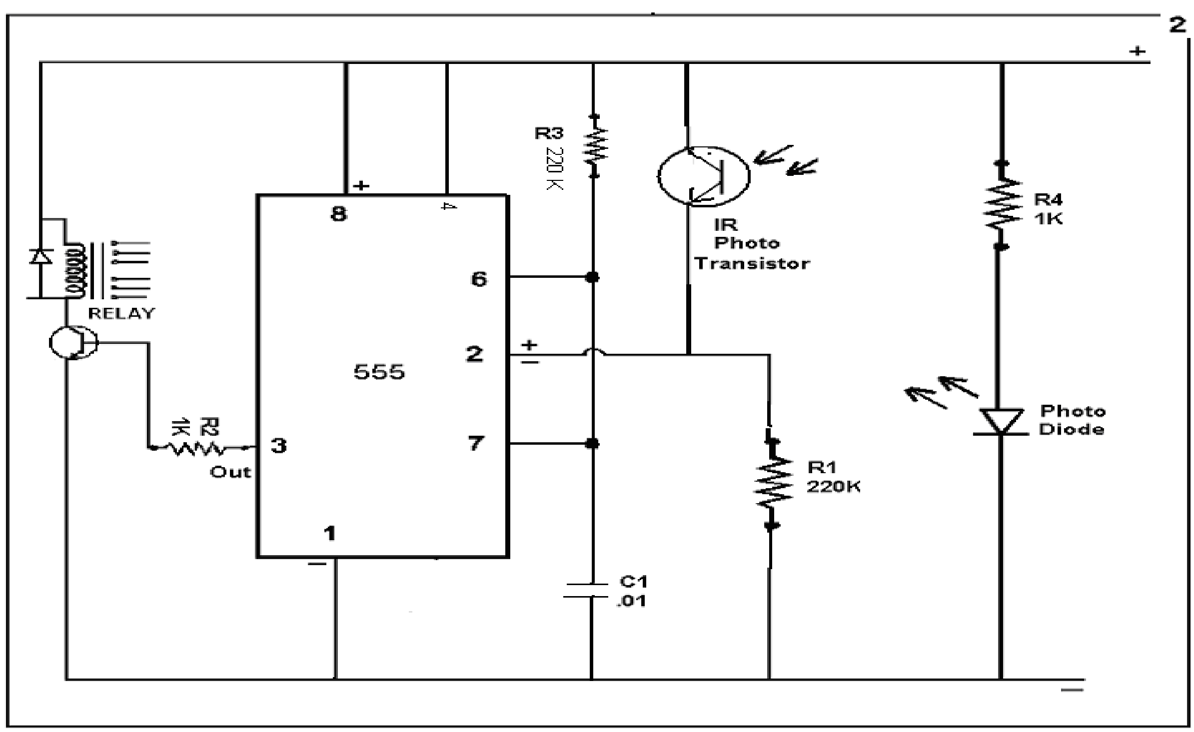


Fig 7.2.1 IR circuit

Working of the infrared sensor Circuit

In this IR detector and transmitter circuit the IC 555 is working under MONOSTABLE mode. The pin 2 i.e. trigger pin and when grounded via IR receiver, the pin 3 output is low. As soon as the IR light beam transmitted is obstructed, a momentary pulse actuates the relay output (or LED). The IR transmitter is simple series connected resistor network from battery. The timing capacitor connected to pin 6 and 7 to ground. The time can varied as per requirement by changing the R value.

1VIBRATION SENSOR

It uses piezoelectric effect to detect the vibrations in the rails due to the arrival or departure of train and the direction of vibration indicate the arrival or departure. This could sense the train's position at roughly at 800 to 900 m away. This input is fed to the microcontroller. This could help in avoiding accidents between trains in slopes because the arrival of one train found out using vibration sensor can be immediately sent to the Control Room and the power supply can be switched off within 3 minutes so trains could be stopped without colliding each other. Vibration or shock sensors are commonly used in

alarm systems to activate an alarm whenever the devices to which they are attached are touched, moved, or otherwise vibrated. Commercial vibration sensors use a piezoelectric ceramic strain transducer attached to a metallic proof mass in order to respond to an externally imposed acceleration. Piezoelectric vibration sensors used for detecting vibration from various vibration sources are generally classified into two large types, resonant type and no resonant type. Vibration sensors are several types. Before selecting the vibration sensor must consider five factors. 1)It's measuring range, 2)frequency range, 3)accuracy,4) transverse sensitivity and 5)ambient conditions. The most commonly used vibration sensor is minisense 100

MINISENSE 100

The Minisense 100 is a low-cost cantilever-type vibration sensor loaded by a mass to offer high sensitivity at low frequencies. The pins are designed for easy installation and are solderable. Horizontal and vertical mounting options are offered as well as a reduced height version. The active sensor area is

shielded for improved RFI/EMI rejection. Rugged, flexible PVDF sensing element withstands high shock overload. Sensor has excellent linearity and dynamic range, and may be used for detecting either continuous vibration or impacts. The mass may be modified to obtain alternative frequency response

and sensitivity selection. It can be classified into two 1)minisense 100 vertical,2)minisense 100 horizontal .The vibration sensor used here is minisense 100 vertical

REGULATED POWER SUPPLY:

A variable regulated power supply,also called a variable bench power supply,is one which you can continuously adjust the output voltage to your requirements. Varying the output of the power supply is recommended way to test a project after having double checked parts placement against circuit drawings and the parts placement

This type of regulation is ideal for having a simple variable bench power supply. Actually this is quite important because one of the first projects a hobbyist should undertake is the construction of a variable regulated power supply. While a dedicated supply is quite handy e.g 5V or 12V,it’s much handier to have a variable supply on hand, especially for testing.

Most digital logic circuits and processors need a 5 volt power supply. To use these parts we need to build a regulated 5 volt source. Usually you start with an unregulated power to make a 5 volt power supply, we use a LM7805 voltage regulator IC (Integrated Circuit).

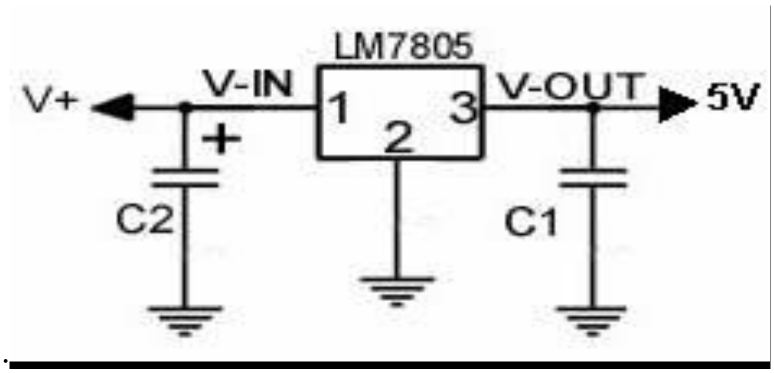


Fig: 8.1.1 LM 7805 block diagram

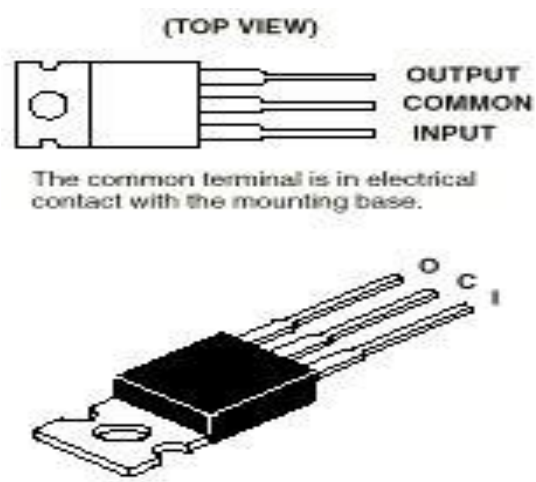


Fig: 8.1.2 Pin representation of LM 7805

The LM7805 is simple to use. You simply connect the positive lead of your unregulated DC power supply(anything from 9VDC to 24VDC) to the Input pin, connect the negative lead to the Common pin and then when you turn on the power, you get a 5 volt supply from the Output pin.

Circuit features:

- **Brief description of operation**: Gives out well regulated +5V output, output current capability of 100mA.

- **Circuit protection**: Built-in overheating protection shuts down output when regulator IC gets too hot.
- **Circuit complexity**: Very simple and easy to build.
- **Circuit performance**: Very stable +5V output voltage, reliable operation
- **Availability of components:** Easy to get, uses only very common basic components.
- **Design testing:** Based on datasheet example circuit, I have used this circuit successfully as part of many electronic projects.
- **Applications:** Part of electronics devices, small laboratory power supply

CHAPTER 4

Firmware Implementation of the project design

This chapter briefly explains about the firmware implementation of the project. The required software tools are discussed in section 4.2. Section 4.3 shows the flow diagram of the project design. Section 4.4 presents the firmware implementation of the project design.

4.1 Software Tools Required

Keil µv3, Proload are the two software tools used to program microcontroller. The working of each software tool is explained below in detail.

4.1.1 Programming Microcontroller

A compiler for a high level language helps to reduce production time. To program the AT89S52 microcontroller the Keil µv3 is used. The programming is done strictly in the embedded C language. Keil µv3 is a suite of executable, open source software development tools for the microcontrollers hosted on the Windows platform.

The compilation of the C program converts it into machine language file (.hex). This is the only language the microcontroller will understand, because it contains the original program code converted into a hexadecimal format. During this step there are some warnings about eventual errors in the program. This is shown in Fig 4.1. If there are no errors and warnings then run the program, the system performs all the required tasks and behaves as expected the software developed. If not, the whole procedure will have to be repeated again. Fig 4.2 shows expected outputs for given inputs when run compiled program.

One of the difficulties of programming microcontrollers is the limited amount of resources the programmer has to deal with. In personal computers resources such as RAM and processing speed are basically limitless when compared to microcontrollers. In contrast, the code on microcontrollers should be as low on resources as possible.

Keil Compiler:

Keil compiler is software used where the machine language code is written and compiled. After compilation, the machine source code is converted into hex code which is to be dumped into the microcontroller for further processing. Keil compiler also supports C language code.

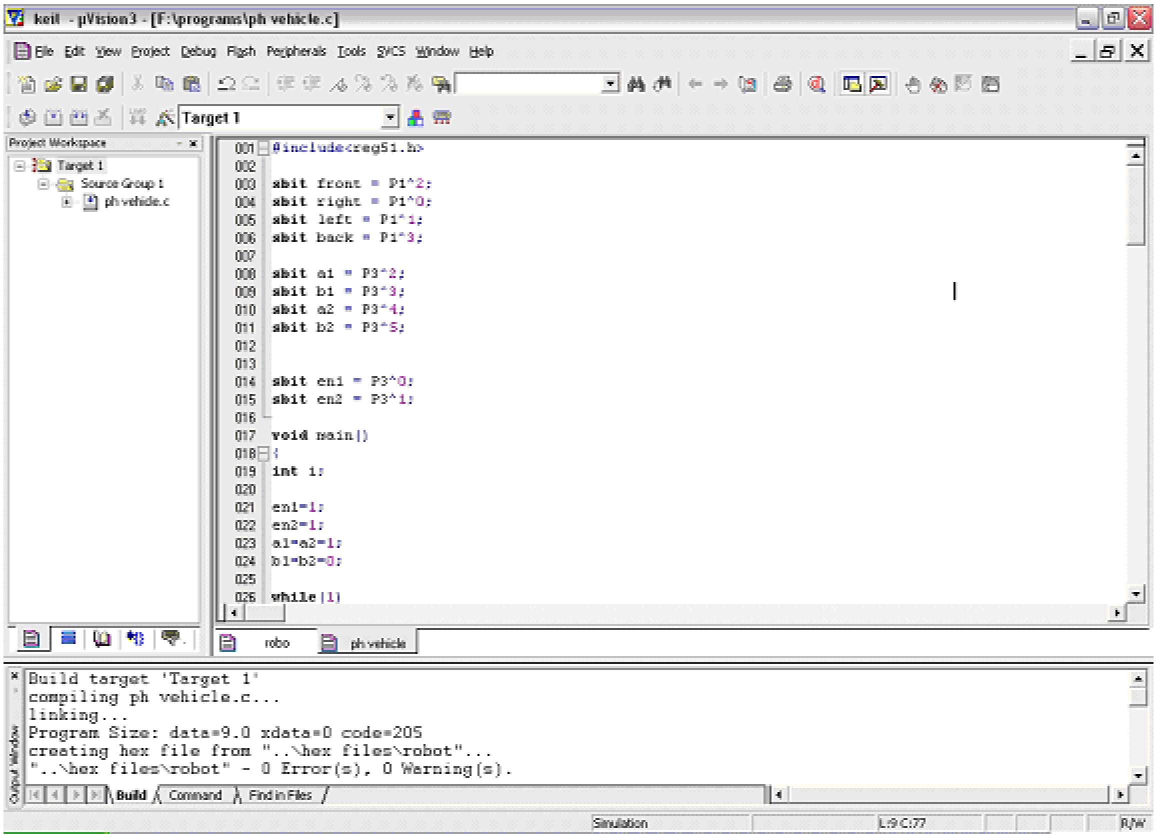


Fig 4.1: Compilation of source Code

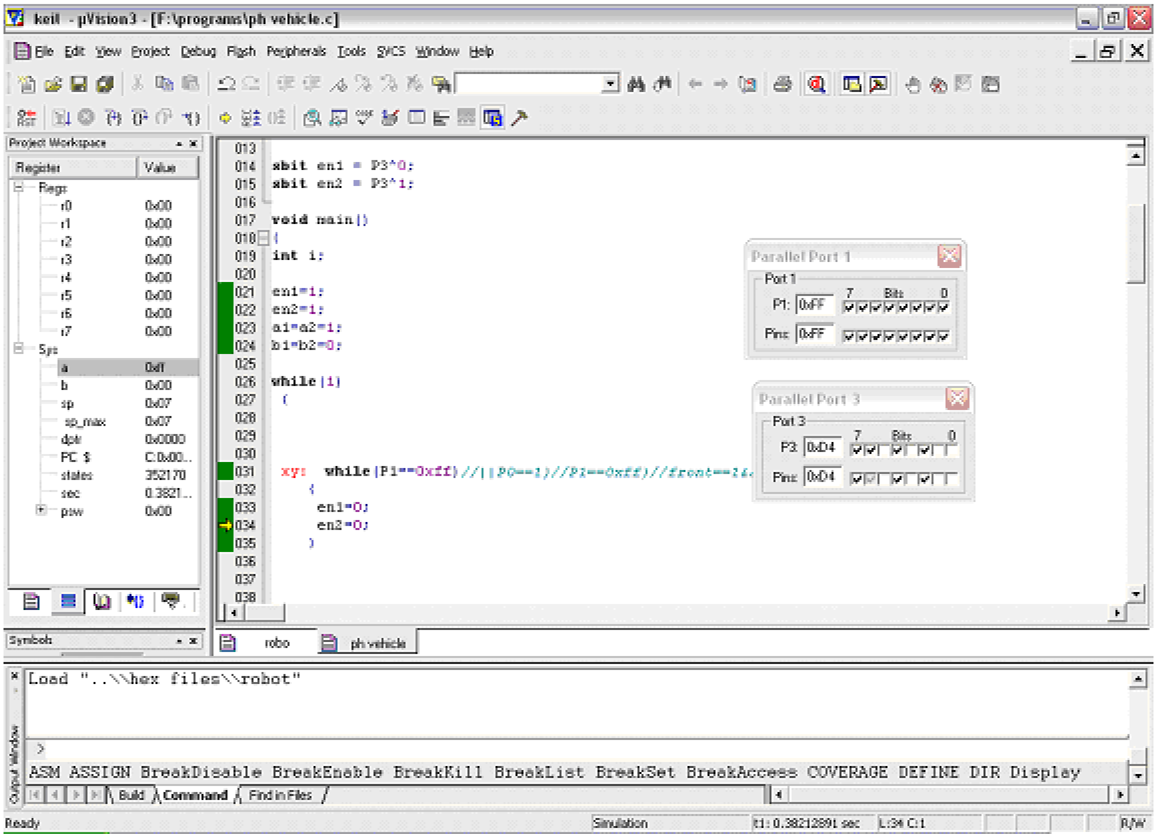


Fig 4.2: Run the compiled program

Proload:

Proload is software which accepts only hex files. Once the machine code is converted into hex code, that hex code has to be dumped into the microcontroller and this is done by the Proload. Proload is a programmer which itself contains a microcontroller in it other than the one which is to be programmed. This microcontroller has a program in it written in such a way that it accepts the hex file from the Keil compiler and dumps this hex file into the microcontroller which is to be programmed. As the Proload programmer kit requires power supply to be operated, this power supply is given from the power supply circuit designed above. It should be noted that this programmer kit contains a power supply section in the board itself but in order to switch on that power supply, a source is required. Thus this is accomplished from the power supply board with an output of 12volts.

Fig 4.3: Atmel 8051 Device programmer

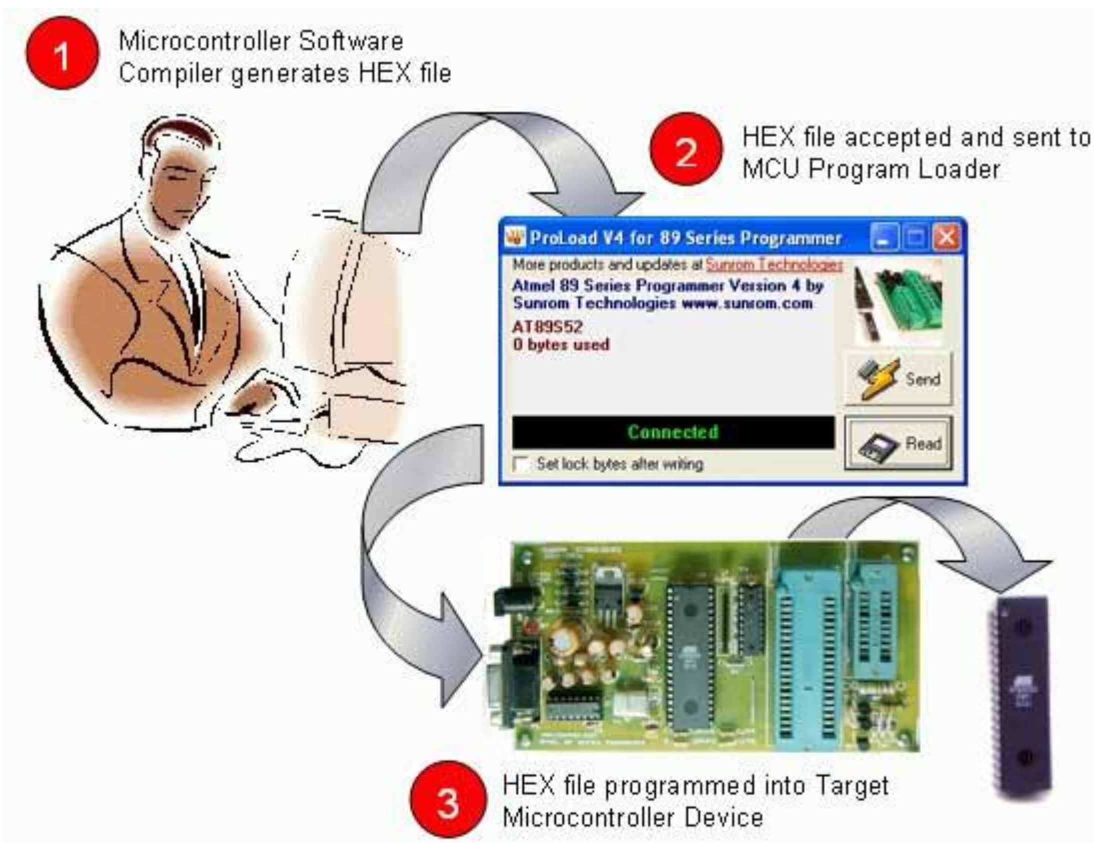
Features

- Supports major Atmel 89 series devices
- Auto Identify connected hardware and devices
- Error checking and verification in-built
- Lock of programs in chip supported to prevent program copying
- 20 and 40 pin ZIF socket on-board
- Auto Erase before writing and Auto Verify after writing
- Informative status bar and access to latest programmed file
- Simple and Easy to use
- Works on 57600 speed

Description

It is simple to use and low cost, yet powerful flash microcontroller programmer for the Atmel 89 series. It will Program, Read and Verify Code Data, Write Lock Bits, Erase and Blank Check. All fuse and lock bits are programmable. This programmer has intelligent onboard firmware and connects to the serial port. It can be used with any type of computer and requires no special hardware. All that is needed is a serial communication ports which all computers have.

All devices have signature bytes that the programmer reads to automatically identify the chip. No need to select the device type, just plug it in and go! All devices also have a number of lock bits to provide various levels of software and programming protection. These lock bits are fully programmable using this programmer. Lock bits are useful to protect the program to be read back from microcontroller only allowing erase to reprogram the microcontroller. The programmer connects to a host computer using a standard RS232 serial port. All the programming 'intelligence' is built into the programmer so you do not need any special hardware to run it. Programmer comes with window based software for easy programming of the devices.



Programming Software

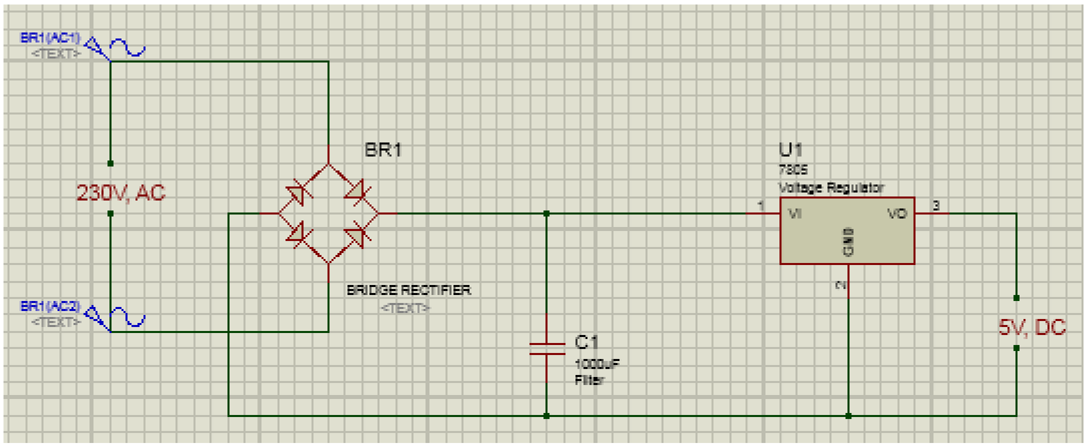
Computer side software called **'Proload V4.1'** is executed that accepts the Intel HEX format file generated from compiler to be sent to target microcontroller. It auto detects the hardware connected to the serial port. It also auto detects the chip inserted and bytes used. Software is developed in Delphi 7 and requires no overhead of any external DLL.

Fig 4.4: Writing the programs bytes onto the microcontroller

CHAPTER5

CIRCUIT DIAGRAM

Power supply schematic:



WORKING PROCEDURE:

In this project we have designed for vehicle parking and the main aim of this paper is to atomize the vehicle park for allowing the vehicles into the park. It can provide the exact location of the free space where the vehicles have to be parked. Here we use the microcontroller AT89C52 and the IR Sensors to identify the vehicles entering in to the park. led is provided to display the information about the total no of vehicles can be parked and the place free for parking by using Embedded C language. This can provide the exact location of the free space where the vehicles have to be parked. Whenever a car comes in front of the gate, the IR signal gets disturbed and the microcontroller will open the red led .

Chapter 6

Results and Discussions

6.1 Testing And Results

We started our project by making power supply. That is easy for me but when we turn toward the main circuit, there are many problems and issues related to it, which we faced, like component selection, which components is better than other and its feature and cost wise a We started our project by making power supply. That is easy for me but when I turn toward the main circuit, there are many problems and issues related to it, which are I faced, like component selection, which components is better than other and its feature and cost wise also, then refer the data books and other materials related to its. I had issues with better or correct result, which I desired. And also the software problem. I also had some soldering issues which were resolved using continuity checks performed on the hardware.

6.2 Conclusion

By implementing this system there are various benefits will be provided for parking. By this system the economic, social and safetybased aspects of the society can be helped and it also helps in preserving the environment, fuel and time. The efforts made in the proposed system is to improve the parking facilities of city and they're by aiming to enhance the quality of life of its people. Parking in populated areas has become challenged, so requirement of smart technologies in order to assist the user in finding parking solutions to shorten the time necessary for parking. In this way we can decrease the traffic congestion, and to improve the everyday life of parking solutions.

FUTURE EXPANSION

- The future the smart parking is to expected by the arrival of automatic vehicles (AVS). Using this system we can park up to 4 times as many cars in the same amount of space like traditional garage. Deduction of space in congested areas like metropolitan cities and populated areas can be improved by the same concept of parking system with horizontal alignment of car parking with the help of some mechanical instruments like motor conveyor sensors and weight cage machines by using the combination with IoT based parking.

Applications

For automatic street lighting system

For automatic room light control

Advantages

Low cost

Easy to use

Power saving

Implement in parking system

BIBILOGRAPHY:

Reference Books

Programming in ANSI C: E BALAGURUSAMY

The 8051microcontroller and embedded systems:MUHAMMAD ALI MAZIDI

JANICE GILLISPIEMAZIDI

The 8051 microcontroller: KENNETH J. AYALA

Website

www.datasheets4u.com

www.8051.com