**A**

**Major Project Report**

**Entitled**

**COGNITIVE AUTOMATION OF ELECTRIC APPLIANCES**

*Submitted in partial fulfillment of the requirements*

*for the Award of the degree*

**BACHELOR OF TECHNOLOGY**

In

**ELECTRONICS AND COMMUICATION ENGINEERING**

***Submitted By***

**R. AKHIL REDDY (13TD1A0401)**

Under the guidance of

**Mr. S. RAGHU**

Assistant Professor

****

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**NIGAMA ENGINEERING COLLEGE**

(Approved by AICTE New Delhi, Affiliated to JNTU, Hyderabad)

Sambaiahpally, Peddapalli-505187

2013-2017

**NIGAMA ENGINEERING COLLEGE**

(Approved by AICTE New Delhi, Affiliated to JNTU, Hyderabad)

Sambaiahpally, Peddapalli -505187.

**Department Of Electronics and Communication Engineering**

### C:\Users\mpmc lab\Desktop\logo.jpg

**CERTIFICATE**

This is certify that the project report entitled “**COGNITIVE AUTOMATION OF ELECTRIC APPLIANCES”**  is being submitted by **R. AKHIL REDDY (13TD1A0401)** in partial fulfillment of the requirement for the award of the degree of BACHELOR OF TECHNOLOGY in **Electronics and Communication Engineering** from Nigama Engineering College, sambaiahpally, Peddapalli.

**Internal Guide Head of the Department**

Mr. S. RAGHU Mr. G.GOUTHAM KUMAR

Asst. Prof.

**Principal External Examiner**

Dr. G. SUBHASH CHANDAR

**DECLARATION**

I declare that the project entitled **“COGNITIVE AUTOMATION OF ELECTRIC APPLIANCES”** is a bonafied work successfully completed by us and submitted to the faculty of **Electronics and Communication Engineering** in partial fulfillment of the requirements for award of degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUICATION ENGINEERING**.

**R. AKHIL REDDY (13TD1A0401)**

**ACKNOWLEDGEMENTS**

At every outset I express my gratitude to almighty lord for showering his grace and blessing upon us to complete this major project.

Although my name appears on the cover of this book, many people had contributed in some form or the other from to this project work. I could not done this work without the assistance or support of each of the following we thank you all.

I express my gratitude to our principal **Dr. G. SUBHASH CHANDAR** for his support and encouragement in completing our project.

I wish to place on my record my deep sense of gratitude to **Mr. G. GOUTHAM KUMAR, HOD of ECE** for his suggestions and advice throughout the project courses.

I also extend gratitude to our beloved project guide **Mr. S.RAGHU, Asst. Prof,** for this instant motivation and valuable through the project work. I also extend my thanks to other faculties for their cooperation during my course.

Finally I would like to thank my friends for their cooperation to complete this project work.

**R. AKHIL REDDY (13TD1A0401)**

**CHAPTER-1**

**INTRODUCTION**

**1.1 EXISTING SYSTEM**

In the recent years, the Home Automation systems has seen a rapid changes due to introduction of various wireless technologies The explosion in the wireless technology has seen the emergence of many standards, especially in the industrial, scientific and medical (ISM) radio band. ZigBee is an IEEE 802.15.4 standard for data communications with business and consumer devices. Zigbee is targeted at applications that require low data rate, long battery life, and secure networking. Zigbee has a defined rate of 250 Kbits/s, best suited for periodic or intermittent data or a single signal transmission from a sensor or input device.

The wireless home Automation systems are supposed to be implemented in existing home environments, without any changes in the infrastructure. The automation centers on recognition of voice commands and uses low-power ZigBee wireless communication modules along with microcontroller. This system is most suitable for the elderly and the disabled persons especially those who live alone and since recognize voice so it is secure. The home automation system is intended to control all lights and electrical appliances in a home or office using voice commands. So in this paperwork our aim is to designed a voice recognition wireless ZigBee based home automation system.

**1.2 PROPOSED SYSTEM**

To make your home, office or industries smart using this artificial intelligence project through the Bluetooth module. An application should be used to automate the appliances. Here, we can use two instruction modes, one hand you can go through voice controlling, another hand using numerical numbers.

we came up with an outstanding technology at present the world is being like an instant work and instant output and comes to our project is voice controlling home automation system using Bluetooth technology through this application we should sound off commands like TV is on or TV is off likewise all the appliances using 8051family microcontroller. Bluetooth modules are more reliable, secure and low power modules and these modules do not require line of sight also. We can use mobile Bluetooth by developing some applications or we can use normal USB Bluetooth dongles by connecting to PC.

**1.2.1 Block diagram of proposed system**

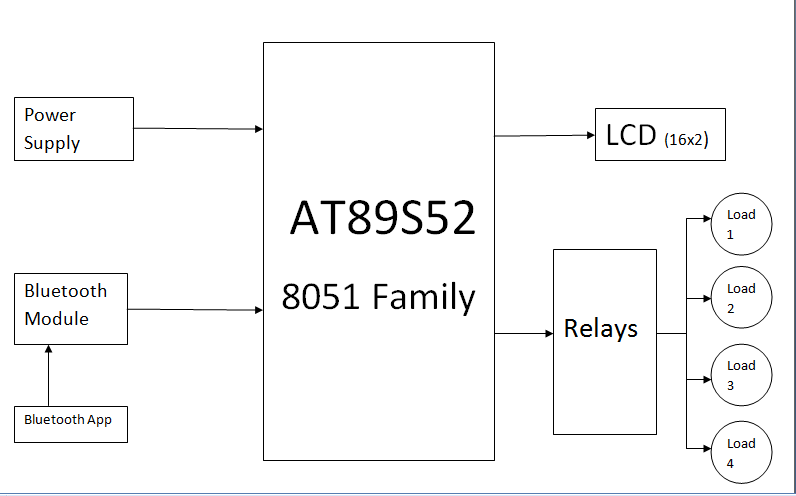
****

Figure 1.1 Block diagram of proposed system

**CHAPTER-2**

**EMBEDDED SYSTEMS**

# 2.1 INTRODUCTION TO EMBEDDED SYSTEMS

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious. All embedded systems are including computers or microprocessors.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. As the embedded system is the combination of both software and hardware.

**Software**

**Hardware**

* **ALP**
* **C**
* **VB**

**Etc.,**

* **Processor**
* **Peripherals**
* **memory**

**Embedded**

**System**

Figure 2.1 Block diagram of Embedded System

Software deals with the languages like ALP, C, and VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or address.

Peripherals: These are the external devices connected

Processor: It is an IC which is used to perform some task

Applications of embedded systems

* Manufacturing and process control
* Construction industry
* Transport
* Buildings and premises
* Domestic service
* Communications
* Office systems and mobile equipment
* Banking, finance and commercial
* Medical diagnostics, monitoring and life support
* Testing, monitoring and diagnostic systems

Processors are classified into four types like

* Micro Processor (µp)
* Micro controller (µc)
* Digital Signal Processor (DSP)
* Application Specific Integrated Circuits (ASIC)

Micro Processor (µp)

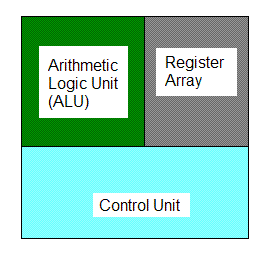
A silicon chip that contains a [CPU](http://www.webopedia.com/TERM/M/CPU.html). In the world of [personal computers](http://www.webopedia.com/TERM/M/personal_computer.html), the terms microprocessor and [CPU](http://www.webopedia.com/TERM/M/microprocessor.html) are used interchangeably. At the heart of all personal computers and most [workstations](http://www.webopedia.com/TERM/M/workstation.html) sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection [systems](http://www.webopedia.com/TERM/M/system.html) for automobiles.

**Three basic characteristics differentiate microprocessors**

* [**Instruction set**](http://www.webopedia.com/TERM/M/instruction.html) The set of instructions that the microprocessor can execute.
* [**Bandwidth**](http://www.webopedia.com/TERM/M/bandwidth.html) The number of [bits](http://www.webopedia.com/TERM/M/bit.html) processed in a single instruction.
* [**Clock speed**](http://www.webopedia.com/TERM/M/clock_speed.html) Given in megahertz ([MHz](http://www.webopedia.com/TERM/M/MHz.html)), the [clock speed](http://www.webopedia.com/TERM/M/microprocessor.html) determines how many instructions per second the [processor](http://www.webopedia.com/TERM/M/processor.html) can [execute](http://www.webopedia.com/TERM/M/execute.html).

In both cases, the higher the value, the more powerful the CPU. For example, a [32-bit](http://www.webopedia.com/TERM/M/32_bit.html) microprocessor that [runs](http://www.webopedia.com/TERM/M/run.html) at 50MHz is more powerful than a 16-bit microprocessor that runs at 25MHz. In addition to bandwidth and clock speed, microprocessors are classified as being either [RISC](http://www.webopedia.com/TERM/M/RISC.html) (reduced instruction set [computer](http://www.webopedia.com/TERM/M/computer.html)) or [CISC](http://www.webopedia.com/TERM/M/CISC.html) (complex instruction set computer).

A microprocessor has three basic elements, as shown above. The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array. The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. Some registers have specific functions - we will deal with these later. The Control Unit controls the entire process. It provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).

  
Figure 2.2 Three Basic Elements of a Microprocessor

2.2 Micro Controller (µc)

A microcontroller is a small computer on a single [integrated circuit](http://en.wikipedia.org/wiki/Integrated_circuit) containing a processor core, memory, and programmable [input/output](http://en.wikipedia.org/wiki/Input/output) peripherals. Program memory in the form of [NOR flash](http://en.wikipedia.org/wiki/NOR_flash) or [OTP ROM](http://en.wikipedia.org/wiki/Programmable_read-only_memory) is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the [microprocessors](http://en.wikipedia.org/wiki/Microprocessor) used in [personal computers](http://en.wikipedia.org/wiki/Personal_computer) or other general purpose applications.

**ALU**

**CU**

**Memory**

Figure 2.3 Block Diagram of Microcontroller (µc)

## Digital Signal Processors (DSPs)

Digital Signal Processors is one which performs scientific and mathematical operation. Digital Signal Processor chips - specialized microprocessors with architectures designed specifically for the types of operations required in digital signal processing. Like a general-purpose microprocessor, a DSP is a programmable device, with its own native instruction code. DSP chips are capable of carrying out millions of floating point operations per second, and like their better-known general-purpose cousins, faster and more powerful versions are continually being introduced. DSPs can also be embedded within complex "system-on-chip" devices, often containing both analog and digital circuitry.

**Application Specific Integrated Circuit (ASIC)**

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function

**ASIC typically contains**

* CPU cores for computation and control
* Peripherals to control timing critical functions
* Memories to store data and program
* Analog circuits to provide clocks and interface to the real world which is analog in nature
* I/Os to connect to external components like LEDs, memories, monitors etc.

**Computer Instruction Set**

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and

2. CISC (Complex Instruction Set computer)

**Reduced Instruction Set Computer (RISC)**

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation. Besides performance improvement, some advantages of RISC and related design improvements are:

* A new microprocessor can be developed and tested more quickly if one of its aims is to be less complicated.
* Operating system and application programmers who use the microprocessor's instructions will find it easier to develop code with a smaller instruction set.
* The simplicity of RISC allows more freedom to choose how to use the space on a microprocessor.
* Higher-level language compilers produce more efficient code than formerly because they have always tended to use the smaller set of instructions to be found in a RISC computer.

**RISC characteristics**

* **Simple instruction set**  
  In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.
* **Same length instructions**  
  Each instruction is the same length, so that it may be fetched in a single operation.

**Complex Instruction Set Computer (CISC)**

CISC, which stands for **Complex Instruction Set Computer,** is a philosophy for designing chips that are easy to program and which make efficient use of memory.

Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

* Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.
* The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.
* As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory.

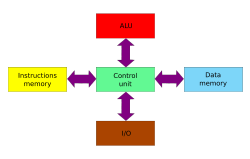
**Memory Architecture**

There two different type’s memory architectures there are:

* Harvard Architecture
* Von-Neumann Architecture

Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

****Figure 2.4 Harvard Architecture

## Modern uses of the Harvard architecture

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern [CPU cache](http://en.wikipedia.org/wiki/CPU_cache) systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

[Digital signal processors](http://en.wikipedia.org/wiki/Digital_signal_processors) (DSPs) generally execute small, highly-optimized audio or video processing algorithms. They avoid caches because their behavior must be extremely reproducible. The difficulties of coping with multiple address spaces are of secondary concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate [SIMD](http://en.wikipedia.org/wiki/SIMD) and [VLIW](http://en.wikipedia.org/wiki/VLIW) processing. [Texas Instruments TMS320](http://en.wikipedia.org/wiki/Texas_Instruments_TMS320) C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

[Microcontrollers](http://en.wikipedia.org/wiki/Microcontrollers) are characterized by having small amounts of program ([flash memory](http://en.wikipedia.org/wiki/Flash_memory)) and data ([SRAM](http://en.wikipedia.org/wiki/SRAM)) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access. The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that [instruction pre-fetch](http://en.wikipedia.org/wiki/Instruction_prefetch) can be performed in parallel with other activities. Examples include, the [AVR](http://en.wikipedia.org/wiki/Atmel_AVR) by [Atmel Corp](http://en.wikipedia.org/wiki/Atmel), the [PIC](http://en.wikipedia.org/wiki/PIC_microcontroller) by [Microchip Technology, Inc.](http://en.wikipedia.org/wiki/Microchip_Technology) and the [ARM](http://en.wikipedia.org/wiki/ARM_architecture) Cortex-M3 processor (not all ARM chips have Harvard architecture).

**Von-Neumann Architecture**

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. They cannot be performed at the same time

The von Neumann architecture is a design model for a stored-program [digital computer](http://en.wikipedia.org/wiki/Computer) that uses a [central processing unit](http://en.wikipedia.org/wiki/Central_processing_unit) (CPU) and a single separate [storage structure](http://en.wikipedia.org/wiki/Computer_data_storage) ("memory") to hold both instructions and [data](http://en.wikipedia.org/wiki/Data_%28computing%29). It is named after the [mathematician](http://en.wikipedia.org/wiki/Mathematician) and early [computer scientist](http://en.wikipedia.org/wiki/Computer_scientist) [John von Neumann](http://en.wikipedia.org/wiki/John_von_Neumann). Such computers implement a [universal Turing machine](http://en.wikipedia.org/wiki/Universal_Turing_machine) and have a [sequential architecture](http://en.wikipedia.org/wiki/SISD).

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

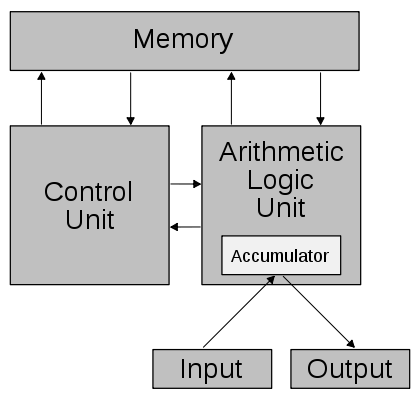


Figure 2.5 Schematic of the Von-Neumann Architecture.

**Basic Difference between Harvard and Von-Neumann Architecture**

* The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.
* Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.
* In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.
* In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.
* Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing, operating one memory system as opposed to two. Is much more flexible and allows for many concepts unavailable to Harvard architecture.

**CHAPTER-3**

**MICRO CONTROLLERS**

**3.1 MICROPROCESSORS Vs MICROCONTROLLERS**

* Microprocessors are single-chip CPUs used in microcomputers.
* Microcontrollers and microprocessors are different in three main aspects: hardware architecture, applications, and instruction set features.
* Hardware architecture: A microprocessor is a single chip CPU while a microcontroller is a single IC contains a CPU and much of remaining circuitry of a complete computer (e.g., RAM, ROM, serial interface, parallel interface, timer, interrupt handling circuit).
* Applications: Microprocessors are commonly used as a CPU in computers while microcontrollers are found in small, minimum component designs performing control oriented activities.
* Microprocessor instruction sets are processing Intensive.
* Their instructions operate on nibbles, bytes, words, or even double words.
* Addressing modes provide access to large arrays of data using pointers and offsets.

**3.2 DIFFERENCE BETWEEN 8051 And 8052**

The 8052 microcontroller is the 8051's "big brother." It is a slightly more powerful microcontroller, sporting a number of additional features which the developer may make use of:

* 256 bytes of Internal RAM (compared to 128 in the standard 8051).
* A third 16-bit timer, capable of a number of new operation modes and 16-bit reloads.
* Additional SFRs to support the functionality offered by the third timer.

**FEATURES**

* Compatible with MCS-51 Products
* 8K Bytes of In-System Programmable (ISP) Flash Memory
* 4.0V to 5.5V Operating Range
* Fully Static Operation: 0 Hz to 33 MHz
* Three-level Program Memory Lock
* 256K Internal RAM
* 32 Programmable I/O Lines
* 3 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

# Description of microcontroller 89S52

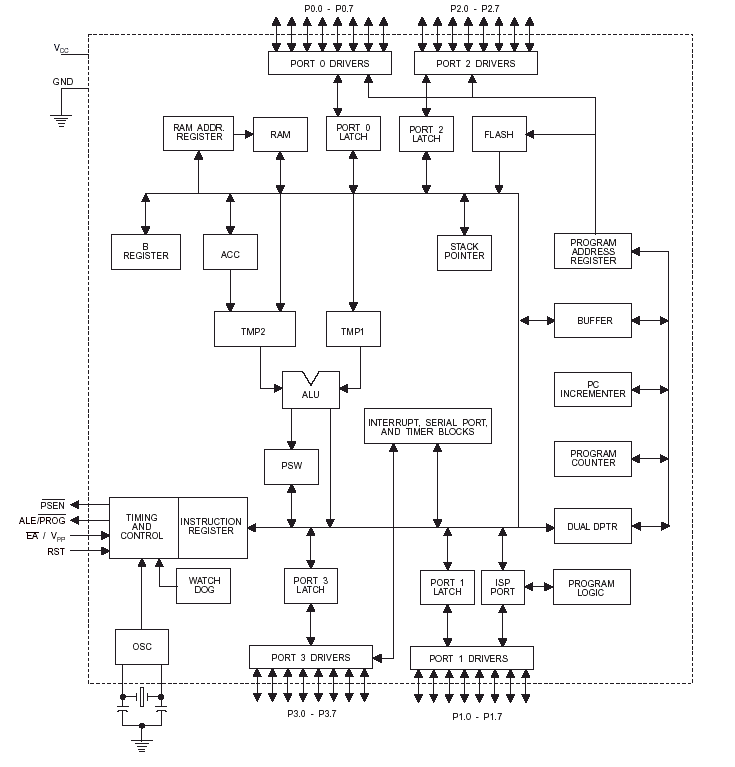
****

Figure 3.1 Description of microcontroller 89s52

**Pin Configuration**

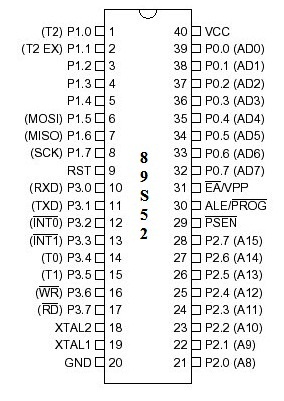


Figure 3.2 pin configuration

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, 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

**Pin Description of microcontroller 89s52**

**VCC**

Supply voltage.

**GND**

Ground

**Port 0**

Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1sare 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 bi-directional 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. In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input

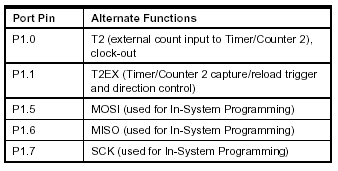


Table 3.1 Alternative functions of port pins

**Port 2**

Port 2 is an 8-bit bi-directional 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.

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 use 8-bit addresses (MOVX @ RI), Port 2emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

**Port 3**

Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are writt 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 also serves the functions of various special features of the AT89S52, as shown in the following table. Port 3 also receives some control signals for Flash programming.

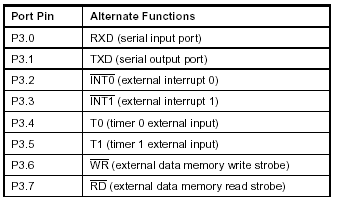


Table 3.2 Alternative functions of port pins

**RST**

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

**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 of1/6 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 be disabled by setting bit 0 of SFR location

**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. A should be strapped to VCC for internal program executions. This pin also receives the 12-voltProgramming enables 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 that 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 2.

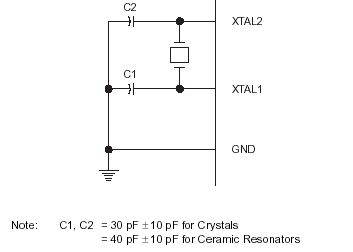


Figure 3.3 Oscillator Connections

**Special Function Register (SFR) Memory**

Special Function Registers (SFR s) are areas of memory that control specific functionality of the 8051 processor. For example, four SFRs permit access to the 8051’s 32 input/output lines. Another SFR allows the user to set the serial baud rate, control and access timers, and configure the 8051’s interrupt system.

**The Accumulator**

The Accumulator, as its name suggests is used as a general register to accumulate the results of a large number of instructions. It can hold 8-bit (1-byte) value and is the most versatile register.

**The “R” registers**

The “R” registers are a set of eight registers that are named R0, R1. Etc up to R7. These registers are used as auxiliary registers in many operations.

**The “B” registers**

The “B” register is very similar to the accumulator in the sense that it may hold an 8-bit (1-byte) value. Two only uses the “B” register 8051 instructions: MUL AB and DIV AB.

**The Data Pointer**

The Data pointer (DPTR) is the 8051’s only user accessible 16-bit (2Bytes) register. The accumulator, “R” registers are all 1-Byte values. DPTR, as the name suggests, is used to point to data. It is used by a number of commands, which allow the 8051 to access external memory.

**ADDRESSING MODES**

An “addressing mode” refers that you are addressing a given memory location. In summary, the addressing modes are as follows, with an example of each:

* Each of these addressing modes provides important flexibility.
* Immediate Addressing MOV A, #20 H
* Direct Addressing MOV A, 30 H
* Indirect Addressing MOV A, @R0
* Indexed Addressing
* External Direct MOVX A, @DPTR
* Code In direct MOVC A, @A+DPTR

**Immediate Addressing**

Immediate addressing is so named because the value to be stored in memory immediately follows the operation code in memory. That is to say, the instruction itself dictates what value will be stored in memory. For example, the instruction:

MOV A, #20H;

This instruction uses immediate Addressing because the accumulator will be loaded with the value that immediately follows in this case 20(hexadecimal). Immediate addressing is very fast since the value to be loaded is included in the instruction. However, since the value to be loaded is fixed at compile-time it is not very flexible.

**Direct Addressing**

Direct addressing is so named because the value to be stored in memory is obtained by directly retrieving it from another memory location. For example:

MOV A, 30h

This instruction will read the data out of internal RAM address 30(hexadecimal) and store it in the Accumulator. Direct addressing is generally fast since, although the value to be loaded isn’t included in the instruction, it is quickly accessible since it is stored in the 8051’s internal RAM. It is also much more flexible than Immediate Addressing since the value to be loaded is whatever is found at the given address which may variable. Also it is important to note that when using direct addressing any instruction that refers to an address between 00h and 7Fh is referring to the SFR control registers that control the 8051 micro controller itself.

**Indirect Addressing**

Indirect addressing is a very powerful addressing mode, which in many cases provides an exceptional level of flexibility. Indirect addressing is also the only way to access the extra 128 bytes of internal RAM found on the 8052. Indirect addressing appears as follows:

MOV A, @R0:

This instruction causes the 8051 to analyze Special Function Register (SFR) Memory: Special Function Registers (SFRs) are areas of memory that control specific functionality of the 8051 processor. For example, four SFRs permit access to the 8051’s 32 input/output lines. Another SFR allows the user to set the serial baud rate, control and access timers, and configure the 8051’s interrupt system.

**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 Pre scaler implemented with the lower five bits of the TL0 register. The upper three bits of TL0 register are indeterminate and should be ignored. Pre scaler overflow increments the TH0 register.

**Mode 1 ( 16-bit Timer )**

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits. Mode 1 configures timer 0 as a 16-bit timer with the TH0 and TL0 registers connected in cascade. The selected input increments the TL0 register.

**Mode 2 (8-bit Timer with Auto-Reload)**

Mode 2 configures timer 0 as an 8-bit timer (TL0 register) that automatically reloads from the TH0 register. TL0 overflow sets TF0 flag in the TCON register and reloads TL0 with the contents of TH0, which is preset by software.

**Mode 3 (Two 8-bit Timers)**

Mode 3 configures timer 0 so that registers TL0 and TH0 operate as separate 8-bit timers. This mode is provided for applications requiring an additional 8-bit timer or counter.

**Timer 1**

Timer 1 is identical to timer 0, except for mode 3, which is a hold-count mode.

**Mode 3 (Halt)**

Placing Timer 1 in mode 3 causes it to halt and hold its count. This can be used to halt Timer 1 when TR1 run control bit is not available i.e. , when Timer 0 is in mode 3

**TCON REGISTER Timer/counter Control Register**

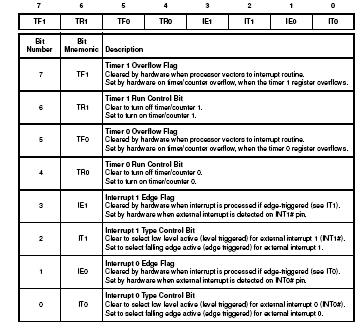


Table 3.3 TCON description of bit numbers

**TMOD REGISTER: Timer/Counter 0 and 1 Modes**

****

Table 3.4 TMOD description of bit numbers

**CHAPTER-4**

**POWER SUPPLY**

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

This power supply section is required to convert AC signal to DC signal and also to reduce the amplitude of the signal. The available voltage signal from the mains is 230V/50Hz which is an AC voltage, but the required is DC voltage(no frequency) with the amplitude of +5V and +12V for various applications.

In this section we have Transformer, Bridge rectifier, are connected serially and voltage regulators for +5V and +12V (7805 and 7812) via a capacitor (1000µF) in parallel are connected parallel as shown in the circuit diagram below. Each voltage regulator output is again is connected to the capacitors of values (100µF, 10µF, 1 µF, 0.1 µF) are connected parallel through which the corresponding output(+5V or +12V) are taken into consideration.



Figure 4.1 Circuit diagram of power supply

**4.1 CIRCUIT EXPLANATION**

**Transformer**

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled electrical conductors. A changing current in the first circuit (the primary) creates a changing magnetic field; in turn, this magnetic field induces a changing voltage in the second circuit (the secondary). By adding a load to the secondary circuit, one can make current flow in the transformer, thus transferring energy from one circuit to the other.

The secondary induced voltage VS, of an ideal transformer, is scaled from the primary VP by a factor equal to the ratio of the number of turns of wire in their respective windings:


\frac{V_{S}}{V_{P}} = \frac{N_{S}}{N_{P}}


## 4.2 BASIC PRINCIPLE

The transformer is based on two principles: firstly, that an electric current can produce a magnetic field (electromagnetism) and secondly that a changing magnetic field within a coil of wire induces a voltage across the ends of the coil (electromagnetic induction). By changing the current in the primary coil, it changes the strength of its magnetic field; since the changing magnetic field extends into the secondary coil, a voltage is induced across the secondary.

A simplified transformer design is shown below. A current passing through the primary coil creates a magnetic field. The primary and secondary coils are wrapped around a core of very high magnetic permeability, such as iron; this ensures that most of the magnetic field lines produced by the primary current are within the iron and pass through the secondary coil as well as the primary coil.

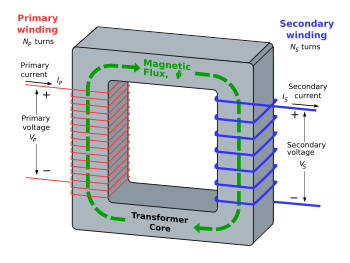
[](http://en.wikipedia.org/wiki/Image:Transformer3d_col3.svg)

Figure 4.2 An ideal step-down transformer showing magnetic flux in the core

### Induction law

The voltage induced across the secondary coil may be calculated from Faraday's law of induction, which states that:


V_{S} = N_{S} \frac{\mathrm{d}\Phi}{\mathrm{d}t}


Where VS is the instantaneous voltage, NS is the number of turns in the secondary coil and Φ equals the magnetic flux through one turn of the coil. If the turns of the coil are oriented perpendicular to the magnetic field lines, the flux is the product of the magnetic field strength B and the area A through which it cuts. The area is constant, being equal to the cross-sectional area of the transformer core, whereas the magnetic field varies with time according to the excitation of the primary. Since the same magnetic flux passes through both the primary and secondary coils in an ideal transformer, the instantaneous voltage across the primary winding equals


V_{P} = N_{P} \frac{\mathrm{d}\Phi}{\mathrm{d}t}


Taking the ratio of the two equations for VS and VP gives the basic equationfor stepping up or stepping down the voltage


\frac{V_{S}}{V_{P}} = \frac{N_{S}}{N_{P}}

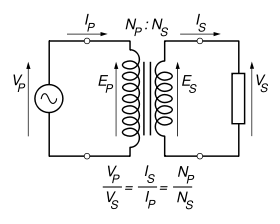

### Ideal power equation

If the secondary coil is attached to a load that allows current to flow, electrical power is transmitted from the primary circuit to the secondary circuit. Ideally, the transformer is perfectly efficient; all the incoming energy is transformed from the primary circuit to the magnetic field and into the secondary circuit. If this condition is met, the incoming electric power must equal the outgoing power.

Pincoming = IPVP = Poutgoing = ISVS

giving the ideal transformer equation


\frac{V_{S}}{V_{P}} = \frac{N_{S}}{N_{P}} = \frac{I_{P}}{I_{S}}


[](http://en.wikipedia.org/wiki/Image:Transformer_under_load.svg)

Pin-coming = IPVP = Pout-going = ISVS

giving the ideal transformer equation


\frac{V_{S}}{V_{P}} = \frac{N_{S}}{N_{P}} = \frac{I_{P}}{I_{S}}


If the voltage is increased (stepped up) (VS > VP), then the current is decreased (stepped down) (IS < IP) by the same factor. Transformers are efficient so this formula is a reasonable approximation.

If the voltage is increased (stepped up) (VS > VP), then the current is decreased (stepped down) (IS < IP) by the same factor. Transformers are efficient so this formula is a reasonable approximation.

The impedance in one circuit is transformed by the square of the turns ratio. For example, if an impedance ZS is attached across the terminals of the secondary coil, it appears to the primary circuit to have an impedance of

Z_S\!\left(\!\tfrac{N_P}{N_S}\!\right)^2\!\!

This relationship is reciprocal, so that the impedance ZP of the primary circuit appears to the secondary to be

Z_P\!\left(\!\tfrac{N_S}{N_P}\!\right)^2\!\!

### Detailed operation

The simplified description above neglects several practical factors, in particular the primary current required to establish a magnetic field in the core, and the contribution to the field due to current in the secondary circuit.

Models of an ideal transformer typically assume a core of negligible reluctance with two windings of zero resistance. When a voltage is applied to the primary winding, a small current flows, driving flux around the magnetic circuit of the core. The current required to create the flux is termed the magnetizing current; since the ideal core has been assumed to have near-zero reluctance, the magnetizing current is negligible, although still required to create the magnetic field.

**Bridge Rectifier**

A diode bridge or bridge rectifier is an arrangement of four diodes in a bridge configuration that provides the same polarity of output voltage for any polarity of input voltage. When used in its most common application, for conversion of alternating current (AC) input into direct current (DC) output, it is known as a bridge rectifier. A bridge rectifier provides full-wave rectification from a two-wire AC input, resulting in lower cost and weight as compared to a center-tapped transformer design, but has two diode drops rather than one, thus exhibiting reduced efficiency over a center-tapped design for the same output voltage.

**Basic Operation**

When the input connected at the left corner of the diamond is positive with respect to the one connected at the right hand corner, current flows to the right along the upper colored path to the output, and returns to the input supply via the lower one.

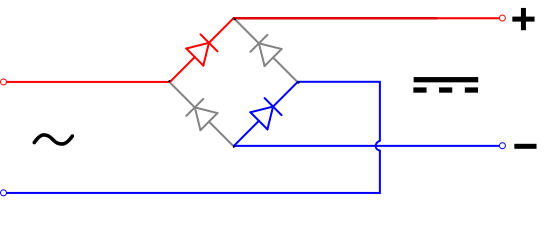
[](http://en.wikipedia.org/wiki/Image:Diode_bridge_alt_1.svg)

Figure 4.3 Operation of bridge rectifier during positive cycle

When the right hand corner is positive relative to the left hand corner, current flows along the upper colored path and returns to the supply via the lower colored path.

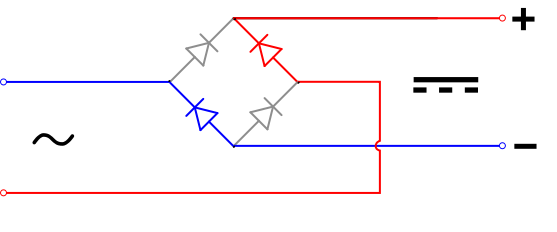
[](http://en.wikipedia.org/wiki/Image:Diode_bridge_alt_2.svg)

Figure 4.4 Operation of bridge rectifier during negative cycle

In each case, the upper right output remains positive with respect to the lower right one. Since this is true whether the input is AC or DC, this circuit not only produces DC power when supplied with AC power: it also can provide what is sometimes called "reverse polarity protection". That is, it permits normal functioning when batteries are installed backwards or DC input-power supply wiring "has its wires crossed" (and protects the circuitry it powers against damage that might occur without this circuit in place).

Prior to availability of integrated electronics, such a bridge rectifier was always constructed from discrete components. Since about 1950, a single four-terminal component containing the four diodes connected in the bridge configuration became a standard commercial component and is now available with various voltage and current ratings.

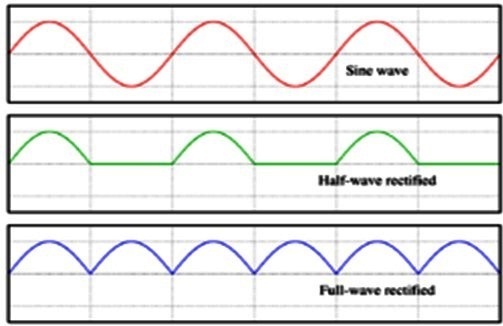


Figure 4.5 Output smoothing

**Output smoothing (Using Capacitor)**

For many applications, especially with single phase AC where the full-wave bridge serves to convert an AC input into a DC output, the addition of a capacitor may be important because the bridge alone supplies an output voltage of fixed polarity but pulsating magnitude (see diagram above).

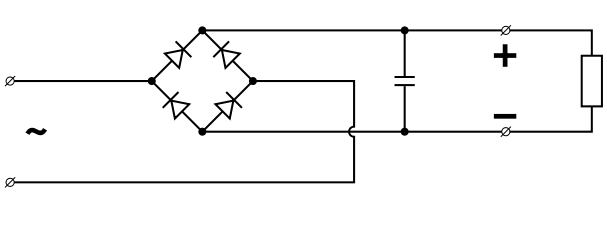
[](http://en.wikipedia.org/wiki/Image:Diode_bridge_smoothing.svg)

Figure 4.6 Bridge rectifier circuit with smoothing capacitor

The function of this capacitor, known as a reservoir capacitor (aka smoothing capacitor) is to lessen the variation in (or 'smooth') the rectified AC output voltage waveform from the bridge. One explanation of 'smoothing' is that the capacitor provides a low impedance path to the AC component of the output, reducing the AC voltage across, and AC current through, the resistive load. In less technical terms, any drop in the output voltage and current of the bridge tends to be cancelled by loss of charge in the capacitor.

This charge flows out as additional current through the load. Thus the change of load current and voltage is reduced relative to what would occur without the capacitor. Increases of voltage correspondingly store excess charge in the capacitor, thus moderating the change in output voltage / current. Also see rectifier output smoothing.

The idealized waveforms shown above are seen for both voltage and current when the load on the bridge is resistive. When the load includes a smoothing capacitor, both the voltage and the current waveforms will be greatly changed. While the voltage is smoothed, as described above, current will flow through the bridge only during the time when the input voltage is greater than the capacitor voltage. For example, if the load draws an average current of n Amps, and the diodes conduct for 10% of the time, the average diode current during conduction must be 10n Amps. This non-sinusoidal current leads to harmonic distortion and a poor power factor in the AC supply.

In a practical circuit, when a capacitor is directly connected to the output of a bridge, the bridge diodes must be sized to withstand the current surge that occurs when the power is turned on at the peak of the AC voltage and the capacitor is fully discharged. Sometimes a small series resistor is included before the capacitor to limit this current, though in most applications the power supply transformer's resistance is already sufficient.

Output can also be smoothed using a choke and second capacitor. The choke tends to keep the current (rather than the voltage) more constant. Due to the relatively high cost of an effective choke compared to a resistor and capacitor this is not employed in modern equipment.

Some early console radios created the speaker's constant field with the current from the high voltage ("B +") power supply, which was then routed to the consuming circuits, (permanent magnets were considered too weak for good performance) to create the speaker's constant magnetic field.

**Voltage Regulator**

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. The 78xx (also sometimes known as LM78xx) series of devices is a family of self-contained fixed linear voltage regulator integrated circuits. The 78xx family is a very popular choice for many electronic circuits which require a regulated power supply, due to their ease of use and relative cheapness. The 78xx line is positive voltage regulators, meaning that they are designed to produce a voltage that is positive relative to a common ground. There is a related line of 79xx devices which are complementary negative voltage regulators. 78xx and 79xx ICs can be used in combination to provide both positive and negative supply voltages in the same circuit, if necessary.

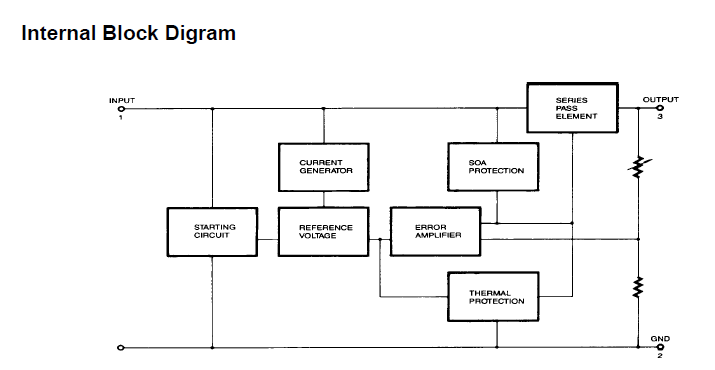


Figure 4.7 Voltage regulator

78xx ICs have three terminals and are most commonly found in the TO220 form factor, although smaller surface-mount and larger TrO3 packages are also available from some manufacturers. These devices typically support an input voltage which can be anywhere from a couple of volts over the intended output voltage, up to a maximum of 35 or 40 volts, and can typically provide up to around 1 or 1.5 amps of current (though smaller or larger packages may have a lower or higher current rating).

**CHAPTER-5**

**BLUETOOTH**

**5.1 INTRODUCTION**

Bluetoot**h** is a wireless protocol utilizing short-range communications technology facilitating data transmission over short distances from fixed and/or mobile devices, creating wireless personal area networks (PANs). The intent behind the development of Bluetooth was the creation of a single digital wireless protocol, capable of connecting multiple devices and overcoming issues arising from synchronization of these devices. Bluetooth uses a very robust radio technology called frequency hopping spread spectrum. It chops up the data being sent and transmits chunks of it on up to 75 different frequencies. In its basic mode, the modulation is Gaussian frequency shift keying (GFSK). It can achieve a gross data rate of 1 Mb/s. Bluetooth provides a way to connect and exchange information between devices such as mobile phones, telephones, laptops, personal computers, printers, GPS receivers, digital cameras, and video game consoles over a secure, globally unlicensed Industrial, Scientific, and Medical (ISM) 2.4 GHz short-range radio frequency bandwidth. The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group (SIG). The Bluetooth SIG consists of companies in the areas of telecommunication, computing, networking, and consumer electronics.

Bluetooth is a standard and communications protocol primarily designed for low power consumption, with a short range (power-class-dependent: 1 meter, 10 meters, 100 meters) based on low-cost transceiver microchips in each device. Bluetooth enables these devices to communicate with each other when they are in range. The devices use a radio communications system, so they do not have to be in line of sight of each other, and can even be in other rooms, as long as the received transmission is powerful enough. Bluetooth device class indicates the type of device and the supported services of which the information is transmitted during the discovery process.

|  |  |  |
| --- | --- | --- |
| **Class** | **Maximum Permitted Power mW(dBm)** | **Range (approximate)** |
| **Class 1** | 100 mW (20 dBm) | ~100 meters |
| **Class 2** | 2.5 mW (4 dBm) | ~10 meters |
| **Class 3** | 1 mW (0 dBm) | ~1 meter |

|  |  |
| --- | --- |
| **Version** | **Data Rate** |
| **Version 1.2** | 1 Mbit/s |
| **Version 2.0 + EDR** | 3 Mbit/s |
| **WiMedia Alliance (proposed)** | 53 - 480 Mbit/s |

In most cases the effective range of class 2 devices is extended if they connect to a class 1 transceiver, compared to pure class 2 network. This is accomplished by the higher sensitivity and transmission power of Class 1 devices.

### 5.2 BLUETOOTH PROFILES

In order to use Bluetooth, a device must be compatible with certain Bluetooth profiles. These define the possible applications and uses of the technology.

### Bluetooth vs Wi-Fi in networking

Bluetooth and Wi-Fi have different applications in today's offices, homes, and on the move: setting up networks, printing, or transferring presentations and files from PDAs to computers. Both are versions of unlicensed wireless technology. Wi-Fi differs from Bluetooth in that it provides higher throughput and covers greater distances, but requires more expensive hardware and may present higher power consumption. They use the same frequency range, but employ different modulation techniques. While Bluetooth is a replacement for cabling in a variety of small-scale applications, Wi-Fi is a replacement for cabling for general local area network access. Bluetooth can be taken as replacement for USB or any other serial cable link, whereas Wi-Fi is wireless Ethernet communications according to the protocol architectures of IEEE 802.3 with TCP/IP. Both standards are operating at a specified bandwidth not identical with that of other networking standards; the mechanical plug compatibility problem known with cables is replaced by the compatibility requirement for an air interface and a protocol stack.

#### Bluetooth devices

Bluetooth exists in many products, such as telephones, printers, modems and headsets. The technology is useful when transferring information between two or more devices that are near each other in low-bandwidth situations. Bluetooth is commonly used to transfer sound data with telephones (i.e. with a Bluetooth headset) or byte data with hand-held computers (transferring files). Bluetooth protocols simplify the discovery and setup of services between devices. Bluetooth devices can advertise all of the services they provide. This makes using services easier because more of the security, network address and permission configuration can be automated than with many other network types.

### Operating system support

Apple has supported Bluetooth since Mac OS X v10.2 which was released in 2002. For Microsoft platforms, Windows XP Service Pack 2 and later releases have native support for Bluetooth. Linux has two popular Bluetooth stacks, BlueZ and Affix.

## Mobile Phone requirements

A mobile phone that is Bluetooth enabled is able to pair with many devices. To ensure the broadest support of feature functionality together with legacy device support. The OMTP forum has recently published a recommendations paper, entitled "Bluetooth Local Connectivity", see external links below to download this paper.

This publication recommends two classes, Basic and Advanced, with requirements that cover imaging, printing, stereo audio and in car useage.

### Bluetooth 1.0 and 1.0B

Versions 1.0 and 1.0B had many problems, and manufacturers had difficulty making their products interoperable. Versions 1.0 and 1.0B also included mandatory Bluetooth hardware device address (BD\_ADDR) transmission in the Connecting process (rendering anonymity impossible at the protocol level), which was a major setback for certain services planned for use in Bluetooth environments.

### Bluetooth 1.1

* Ratified as IEEE Standard 802.15.1-2002.
* Many errors found in the 1.0B specifications were fixed.
* Added support for non-encrypted channels.
* Received Signal Strength Indicator (RSSI).

### Bluetooth 1.2

This version is backward-compatible with 1.1 and the major enhancements include the following:

* Faster Connection and Discovery
* Adaptive frequency-hopping spread spectrum (AFH), which improves resistance to radio frequency interference by avoiding the use of crowded frequencies in the hopping sequence.
* Higher transmission speeds in practice, up to 721 kbit/s, as in 1.1.
* Extended Synchronous Connections (eSCO), which improve voice quality of audio links by allowing retransmissions of corrupted packets, and may optionally increase audio latency to provide better support for concurrent data transfer.
* Host Controller Interface (HCI) support for three-wire UART.
* Ratified as IEEE Standard 802.15.1-2005.

### Bluetooth 2.0

This version of the Bluetooth specification was released on November 10, 2004. It is backward-compatible with the previous version 1.1. The main difference is the introduction of an Enhanced Data Rate (EDR) for faster data transfer. The nominal rate of EDR is about 3 megabits per second, although the practical data transfer rate is 2.1 megabits per second. The additional throughput is obtained by using a different radio technology for transmission of the data. Standard, or Basic Rate, transmission uses Gaussian Frequency Shift Keying (GFSK) modulation of the radio signal; EDR uses a combination of GFSK and Phase Shift Keying (PSK) modulation.

According to the 2.0 specification, EDR provides the following benefits:

* Three times faster transmission speed — up to 10 times (2.1 Mbit/s) in some cases.
* Reduced complexity of multiple simultaneous connections due to additional bandwidth.
* Lower power consumption through a reduced duty cycle.

The Bluetooth Special Interest Group (SIG) published the specification as "Bluetooth 2.0 + EDR" which implies that EDR is an optional feature. Aside from EDR, there are other minor improvements to the 2.0 specification, and products may claim compliance to "Bluetooth 2.0" without supporting the higher data rate. At least one commercial device, the HTC TyTN pocket PC phone, states "Bluetooth 2.0 without EDR" on its data sheet.

### Bluetooth 2.1

Bluetooth Core Specification Version 2.1 is fully backward-compatible with 1.1, and was adopted by the Bluetooth SIG on July 26, 2007. This specification includes the following features:

* **Extended inquiry response** provides more information during the inquiry procedure to allow better filtering of devices before connection. This information includes the name of the device, a list of services the device supports, as well as other information like the time of day, and pairing information.
* **Sniff sub rating**: reduces the power consumption when devices are in the sniff low-power mode, especially on links with asymmetric data flows. Human interface devices (HID) are expected to benefit the most, with mouse and keyboard devices increasing the battery life by a factor of 3 to 10. It lets devices decide how long they will wait before sending keep alive messages to one another. Previous Bluetooth implementations featured keep alive message frequencies of up to several times per second. In contrast, the 2.1 specification allows pairs of devices to negotiate this value between them to as infrequently as once every 5 or 10 seconds.
* **Encryption Pause Resume**: enables an encryption key to be refreshed, enabling much stronger encryption for connections that stay up for longer than 23.3 hours (one Bluetooth day).
* **Secure Simple Pairing**: radically improves the pairing experience for Bluetooth devices, while increasing the use and strength of security. It is expected that this feature will significantly increase the use of Bluetooth.

#### High-speed Bluetooth

On 28 March 2006, the Bluetooth Special Interest Group announced its selection of the WiMedia Alliance Multi-Band Orthogonal Frequency Division Multiplexing (MB-OFDM) version of UWB for integration with current Bluetooth wireless technology.

This new version of Bluetooth technology will meet the high-speed demands of synchronizing and transferring large amounts of data, as well as enabling high-quality video and audio applications for portable devices, multi-media projectors and television sets, and wireless VOIP.

At the same time, Bluetooth technology will continue catering to the needs of very low power applications such as mouse, keyboards, and mono headsets, enabling devices to select the most appropriate physical radio for the application requirements, thereby offering the best of both worlds.

#### Bluetooth 3.0

The next version of Bluetooth after v2.1, code-named Seattle (the version number of which is TBD) has many of the same features, but is most notable for plans to adopt ultra-wideband (UWB) radio technology. This will allow Bluetooth use over UWB radio, enabling very fast data transfers of up to 480 Mbit/s, while building on the very low-power idle modes of Bluetooth.

#### Core protocols

“Bluetooth is defined as a layer protocol architecture consisting of core protocols, cable replacement protocols, telephony control protocols, and adopted protocols”. Bluetooth’s core protocols form a five-layer stack, consisting of the following:

**Bluetooth Radio** – specifics details of the air interface, including frequency, frequency hopping, modulation scheme, and transmission power

**Baseband** – concerned with connection establishment within a piconet, addressing, packet format, timing, and power control.

**Link Manager Protocol (LMP)** – Establishes the link setup between Bluetooth devices and manages ongoing links, including security aspects (e.g. authentication and encryption), and control and negotiation of baseband packet size

**Logical Link Control and Adaptation Protocol (L2CAP)** – adapts the upper-layer protocols to the baseband layer, providing both connectionless and connection-oriented services.

**Service Discovery Protocol (SDP)** – handles device information, services, and queries for service characteristics between two or more Bluetooth devices.

#### Cable replacement protocol

Radio frequency communications (RFCOMM) is the cable replacement protocol used to create a virtual serial port used to make replacement of cable technologies transparent through minimal modification of existing devices. RFCOMM provides for binary data transport and emulates EIA-232 (formerly RS-232) control signals over the Bluetooth baseband layer.

#### Telephony control protocol

Telephony control protocol-binary (TCS BIN) is the bit-oriented protocol that defines the call control signaling for the establishment of voice and data calls between Bluetooth devices. Additionally, “TCS BIN defines mobility management procedures for handling groups of Bluetooth TCS devices”

**Point-to-Point Protocol (PPP)** – Internet standard protocol for transporting IP datagrams over a point-to-point link

**TCP/IP/UDP** – Foundation Protocols for TCP/IP protocol suite

**Object Exchange Protocol (OBEX)** – Session-layer protocol for the exchange of objects, providing a model for object and operation representation

**Wireless Application Environment / Wireless Application Protocol (WAE/WAP)** – WAE specifies an application framework for wireless devices and WAP is an open standard to provide mobile users access to telephony and information services.

### Communication and connection

A master Bluetooth device can communicate with up to seven devices. This network group of up to eight devices is called a piconet. A piconet is an ad-hoc computer network, using Bluetooth technology protocols to allow one master device to interconnect with up to seven active devices. Up to 255 further devices can be inactive, or parked, which the master device can bring into active status at any time.

At any given time, data can be transferred between the master and one other device, however, the devices can switch roles and the slave can become the master at any time. The master switches rapidly from one device to another in a round-robin fashion. (Simultaneous transmission from the master to multiple other devices is possible, but not used much.)

Many USB Bluetooth adapters are available, some of which also include an IrDA adapter. Older (pre-2003) Bluetooth adapters, however, have limited services, offering only the Bluetooth Enumerator and a less-powerful Bluetooth Radio incarnation. Such devices can link computers with Bluetooth, but they do not offer much in the way of services that modern adapters do.

### Setting up connections

Any Bluetooth device will transmit the following information on demand:

* Device name.
* Device class.
* List of services.
* Technical information, for example, device features, manufacturer, Bluetooth specification used, clock offset.

Any device may perform an inquiry to find other devices to connect to, and any device can be configured to respond to such inquiries. However, if the device trying to connect knows the address of the device, it always responds to direct connection requests and transmits the information shown in the list above if requested. Use of device services may require pairing or acceptance by its owner, but the connection itself can be initiated by any device and held until it goes out of range. Some devices can be connected to only one device at a time, and connecting to them prevents them from connecting to other devices and appearing in inquiries until they disconnect from the other device.

Every device has a unique 48-bit address. However these addresses are generally not shown in inquiries. Instead, friendly Bluetooth names are used, which can be set by the user. This name appears when another user scans for devices and in lists of paired devices.

### Pairing

Pairs of devices may establish a trusted relationship by learning (by user input) a shared secret known as a passkey. A device that wants to communicate only with a trusted device can cryptographically authenticate the identity of the other device. Trusted devices may also encrypt the data that they exchange over the airwaves so that no one can listen in. The encryption can, however, be turned off, and passkeys are stored on the device file system, not on the Bluetooth chip itself.

Since the Bluetooth address is permanent, a pairing is preserved, even if the Bluetooth name is changed. Pairs can be deleted at any time by either device. Devices generally require pairing or prompt the owner before they allow a remote device to use any or most of their services. Some devices, such as mobile phones, usually accept OBEX business cards and notes without any pairing or prompts. Certain printers and access points allow any device to use its services by default, much like unsecured Wi-Fi networks. Pairing algorithms are sometimes manufacturer-specific for transmitters and receivers used in applications such as music and entertainment. Bluetooth 2.1 has an optional "touch-to-pair" feature based on NFC. By simply bringing two devices into close range (around 10cm), pairing can securely take place without entering a passkey or manual configuration.

### Air interface

The protocol operates in the license-free ISM band at 2.4-2.4835 GHz. To avoid interfering with other protocols that use the 2.45 GHz band, the Bluetooth protocol divides the band into 79 channels (each 1 MHz wide) and changes channels up to 1600 times per second. Implementations with versions 1.1 and 1.2 reach speeds of 723.1 kbit/s. Version 2.0 implementations feature Bluetooth Enhanced Data Rate (EDR) and reach 2.1 Mbit/s. Technically, version 2.0 devices have a higher power consumption, but the three times faster rate reduces the transmission times, effectively reducing power consumption to half that of 1.x devices (assuming equal traffic load).

**Features**

1) Output Interface UART, Compliant Bluetooth stack v1.2-improved AFH (Adaptive Frequency Hoping), Fast connection

2) Transmit Power - ESD100/110: Max. +18dBm ESD200/210: Max. +4dBm

3) Receiving Sensitivity - ESD100/110: -88dBm (0.1%BER) ESD200/210: -80dBm (0.1%BER)

4) Antenna gain - Chip: 0dBi, Stub: +2dBi, Dipole: +3dBi, Patch: +9dBi

5) Compact size - ESD100:27.5 x 30 x 14(mm)

ESD110: 27.5 x 27.7 x 14(mm)

ESD200/210:18 x 20 x 11.7 (mm)

6) Provides transparent RS232 serial cable replacement.

7) Supports Bluetooth Serial Port Profile

8) Interoperability with PDA, laptops etc.

9) Built-in chip antenna included

10) Supports firmware upgrade via windows-based Software (Parani Updater)

11) Working distance ( In an open field ):

Parani-ESD100: Class 1, Nom. 100 meters

Parani-ESD110: Class 1, Nom. 100 meters, up to 1000m using patch antenna

Parani-ESD200: Class 2, Nom. 30meters

Parani-ESD210: Class 2, Nom. 30meters, up to 300m using patch antenna

12) Easy to use Windows configuration tool available.

**Parani-ESD Series** is OEM Bluetooth-Serial Module type product line based on Bluetooth technology. Parani-ESD Series is designed for integration into user devices by on-board installation. They are connected to the device via built-in UART interface and communicate with other Bluetooth device.

Parani-ESD Series enables RS232-based serial devices to communicate wirelessly throughout the range of 30m ~300m (Parani- ESD210-Class 2) or 100m ~ 1000m (Parani-ESD110-Class 1). Parani-ESD100/200 has a built-in on-board antenna and Parani- ESD110/210 has an antenna connector for connecting external antenna. Users can extend Bluetooth communication range by connecting the external antenna to antenna connector. Users may configure the Parani-ESD Series by using easy-to-use Windows-based utility software or by using standard AT command set.

**Typical application areas of the Parani-ESD Series include:**

* RS232 cable replacement
* Wireless Factory monitoring
* Truck/Bus monitoring system
* PLC programming
* Car Diagnostics
* Wireless Printing
* Wireless POS system
* Wireless logistics
* Wireless machine (healthcare/industrial) monitoring

 ****

Figure 5.1 1Parani –ESD100/110 Figure 5.2 1Parani –ESD200/210

### List of applications

* Wireless control of and communication between a mobile phone and a hands-free headset. This was one of the earliest applications to become popular.
* Wireless networking between PCs in a confined space and where little bandwidth is required.
* Wireless communications with PC input and output devices, the most common being the mouse, keyboard and printer.
* Transfer of files between devices with OBEX.
* Transfer of contact details, calendar appointments, and reminders between devices with OBEX.
* Replacement of traditional wired serial communications in test equipment, GPS receivers, medical equipment, bar code scanners, and traffic control devices.
* For controls where infrared was traditionally used.
* Sending small advertisements from Bluetooth enabled advertising hoardings to other, discoverable, Bluetooth devices.

**Specifications**



**CHAPTER-6**

**LIQUID CRISTAL DISPLAY**

**6.1 INTRODUCTION**

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

**Shapes and S**

available. Line lengths of 8, 16, 20, 24, 32 and 40 characters are all standard, in one, two

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

**Features**

* Interface with either 4-bit or 8-bit microprocessor.
* Display data RAM
* 80x8 bits (80 characters).
* Character generator ROM
* 160 different 5 7 dot-matrix character patterns.
* Character generator RAM
* different user programmed 5 7 dot-matrix patter
* Display data RAM and character generator RAM may be Accessed by the microprocessor.
  + Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

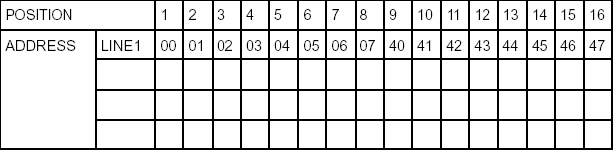


Figure 6.1 Address locations for a 1x16 line LCD

**Shapes and sizes**

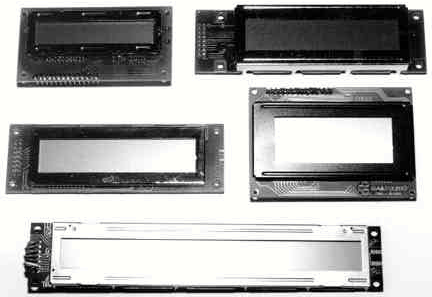


Figure 6.2 Shapes

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8, 16, 20, 24, 32 and 40 characters are all standard, in one, two and four line versions.

Several different LC technologies exist. “Supertwist” types, for example, offer improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

**Electrical block diagram**

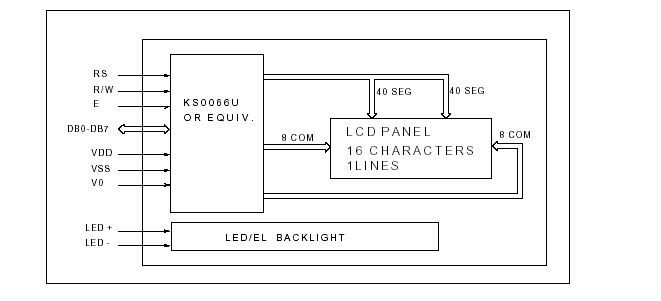


Figure 6.3 Electrical block diagram

**Power supply for LCD driving**

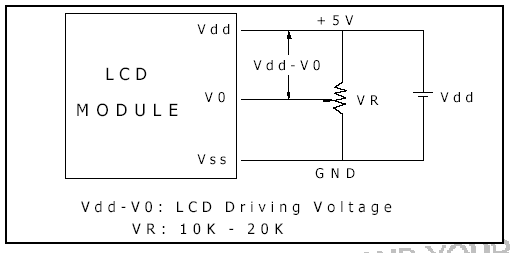
****

Figure 6.4 Power supply for lcd driving

#### Pin Description

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

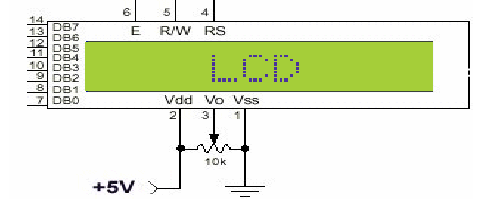


Figure 6.5 pin diagram of 1x16 lines LCD

**Control Lines**

**EN**

Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

**RS**

Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

**RW**

Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user.

**Logic status on control lines**

• E - 0 Access to LCD disabled

1 Access to LCD enabled

• R/W - 0 Writing data to LCD

1 Reading data from LCD

• RS - 0 Instructions

1 Character

**Writing data to the LCD**

* Set R/W bit to low
* Set RS bit to logic 0 or 1 (instruction or character)
* Set data to data lines (if it is writing)
* Set E line to high
* Set E line to low

**Read data from data lines (if it is reading) on LCD**

1) Set R/W bit to high

2) Set RS bit to logic 0 or 1 (instruction or character)

3) Set data to data lines (if it is writing)

4) Set E line to high and low

**Entering Text**

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d. i. l. switch pack with of re-wiring is necessary. The switches must be the type where on = 0, so that when they are turned to the zero position, all four outputs are shorted to the common pin, and in position “F”, all four outputs are open circuit.

All the available characters that are built into the module associated with the characters are quoted in binary and hexadecimal, most significant bits (“left-hand” four bits) across the top, and least significant bits (“right-hand” four bits) down the left.

Most of the characters conform to the ASCII standard, although the Japanese and Greek characters (and a few other things) are obvious exceptions. Since these intelligent modules were designed in the “Land of the Rising Sun,” it seems only fair that their Katakana phonetic symbols should also be incorporated. The more extensive Kanji character set, which the Japanese share with the Chinese, consisting of several thousand different characters, is not included!

****

**Initialization by Instructions**

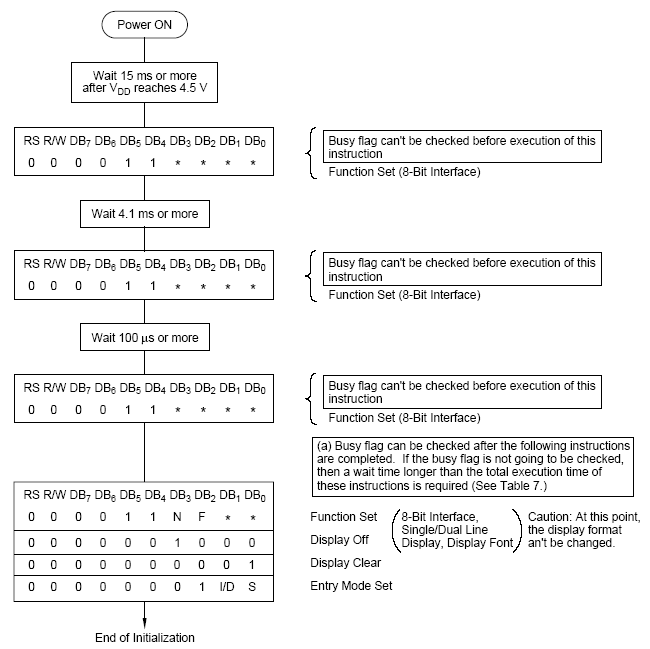


Figure 6.6 initialization by instruction

**CHAPTER-7**

**RELAYS**

**7.1 INTRODUCTION**

A relay is an electrical switch that opens and closes under the control of another electrical circuit. In the original form, the switch is operated by an electromagnet to open or close one or many sets of contacts. A relay is able to control an output circuit of higher power than the input circuit, it can be considered to be, in a broad sense, a form of an electrical amplifier.

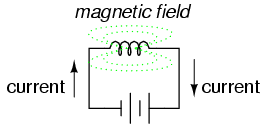
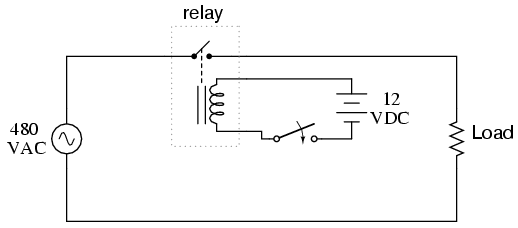
Relays are usually SPDT (single pole double through switch)or DPDT (double pole double through switch) but they can have many more sets of switch contacts, for example relays with 4 sets of changeover contacts are readily available.

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts,instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

Magnetic latching relays require one pulse of coil power to move their contacts in one direction, and another, redirected pulse to move them back. Repeated pulses from the same input have no effect. Magnetic latching relays are useful in applications where interrupted power should not be able to transition the contacts.

**7.2 Basic operation of a relay**

An electric current through a conductor will produce a magnetic field at right angles to the direction of electron flow. If that conductor is wrapped into a coil shape, the magnetic field produced will be oriented along the length of the coil. The greater then current, the greater the strength of the magnetic field, all other factors being equal.

Inductors react against changes in current because of the energy stored in this magnetic field. When we construct a transformer from two inductor coils around a common iron core, we use this field to transfer energy from one coil to the other. However, there are simpler and more direct uses for electromagnetic fields than the applications we've seen with inductors and transformers. The magnetic field produced by a coil of current-carrying wire can be used to exert a mechanical force on any magnetic object, just as we can use a permanent magnet to attract magnetic objects, except that this magnet (formed by the coil) can be turned on or off by switching the current on or off through the coil.

If we place a magnetic object near such a coil for the purpose of making that object move when we energize the coil with electric current, we have what is called a solenoid. The movable magnetic object is called an armature, and most armatures can be moved with either direct current (DC) or alternating current (AC) energizing the coil. The polarity of the magnetic field is irrelevant for the purpose of attracting an iron armature. Solenoids can be used to electrically open door latches, open or shut valves, move robotic limbs, and even actuate electric switch mechanisms and is used to actuate a set of switch contacts.

**Relays can be categorized according to the magnetic system and operation**

#### Neutral Relays

This is the most elementary type of relay. The neutral relays have a magnetic coil, which operates the relay at a specified current, regardless of the polarity of the voltage applied.

#### Biased Relays

Biased relays have a permanent magnet above the armature. The relay operates if the current through the coil winding establishes a magneto-motive force that opposes the flux by the permanent magnet. If the fluxes are in the same direction, the relay will not operate, even for a greater current through the coil.

#### Polarized Relays

Like the biased relays, the polarized relays operate only when the current through the coil in one direction. But there the principle is different. The relay coil has a diode connected in series with it. This blocks the current in the reverse direction. The major difference between biased relays and polarized relays is that the former allows the current to pass through in the reverse direction, but does the not operate the relay and the later blocks the current in reverse direction. You can imagine how critical these properties when relays are connected in series to form logic circuits.

#### Magnetic Stick Relays or Perm polarized Relays

These relays have a magnetic circuit with high permanence. Two coils, one to operate (pick up) and one to release (drop) are present. The relay is activated by a current in the operate coil. On the interruption of the current the armature remains in picked up position by the residual magnetism. The relay is released by a current through the release coil.

#### Slow Release Relays

These relays have a capacitor connected in parallel to their coil. When the operating current is interrupted the release of relay is delayed by the stored charge in the capacitor. The relay releases as the capacitor discharges through the coil.

#### Relays for AC

These are neutral relays and picked up for a.c. current through their coil. These are very fast in action and used on power circuits of the point motors, where high current flows through the contacts. A normal relay would be slow and make sparks which in turn may weld the contacts together. All relays have two operating values (voltages), one pick-up and the other drop away. The pick-up value is higher than the drop away value.

**Applications**

* To control a high-voltage circuit with a low-voltage signal, as in some types of modems or audio amplifiers,
* To control a high-current circuit with a low-current signal, as in the starter solenoid of an automobile,
* To detect and isolate faults on transmission and distribution lines by opening and closing circuit breakers (protection relays),
* To isolate the controlling circuit from the controlled circuit when the two are at different potentials, for example when controlling a mains-powered device from a low-voltage switch.
* To perform logic functions. For example, the boolean AND function is realised by connecting NO relay contacts in series, the OR function by connecting NO contacts in parallel. The change-over or Form C contacts perform the XOR (exclusive or) function. Similar functions for NAND and NOR are accomplished using NC contacts. The Ladder programming language is often used for designing relay logic networks.
* Early computing. Before vacuum tubes and transistors, relays were used as logical elements in digital computers. See ARRA (computer), Harvard Mark II, Zuse Z2, and Zuse Z3.
* Safety-critical logic. Because relays are much more resistant than semiconductors to nuclear radiation, they are widely used in safety-critical logic, such as the control panels of radioactive waste-handling machinery.
* To perform time delay functions. Relays can be modified to delay opening or delay closing a set of contacts. A very shorts (a fraction of a second) delay would use a copper disk between the armature and moving blade assembly. Current flowing in the disk maintains magnetic field for a short time, lengthening release time. For a slightly longer (up to a minute) delay, a dashpot is used. A dashpot is a piston filled with fluid that is allowed to escape slowly.

**CHAPTER-8**

**KEIL SOFTWARE**

**Installing the Keil software on a Windows PC**

* Insert the CD-ROM in your computer’s CD drive
* On most computers, the CD will “auto run”, and you will see the Keil installation menu. If the menu does not appear, manually double click on the Setup icon, in the root directory: you will then see the Keil menu.
* On the Keil menu, please select “Install Evaluation Software”. (You will not require a license number to install this software).
* Follow the installation instructions as they appear.

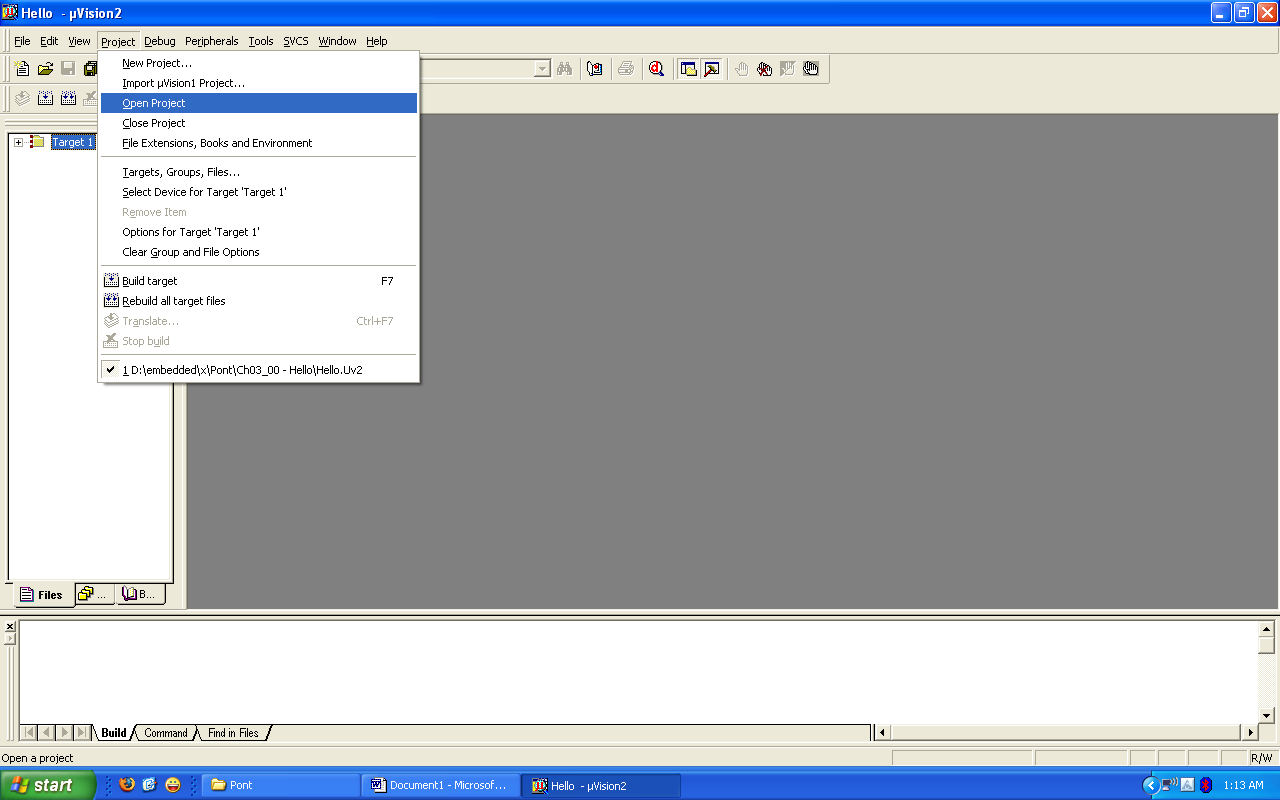
**Loading the Projects**

The example projects for this book are NOT loaded automatically when you install the Keil compiler. These files are stored on the CD in a directory “/Pont”. The files are arranged by chapter: for example, the project discussed in Chapter 3 is in the directory “/Pont/Ch03\_00-Hello”.Rather than using the projects on the CD (where changes cannot be saved), please copy the files from CD onto an appropriate directory on your hard disk.

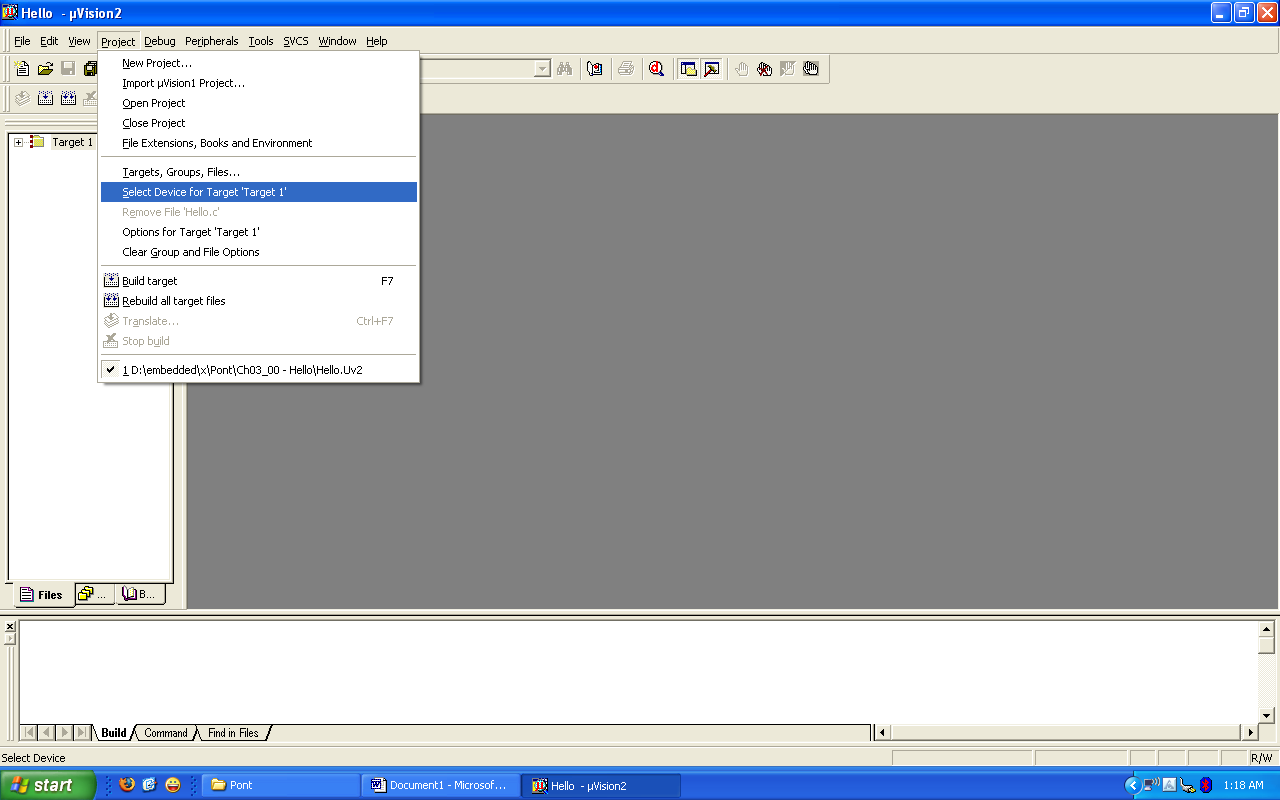
**Note:** You will need to change the file properties after copying: file transferred from the CD will be ‘read only’.

1) Open the Keil μVision2

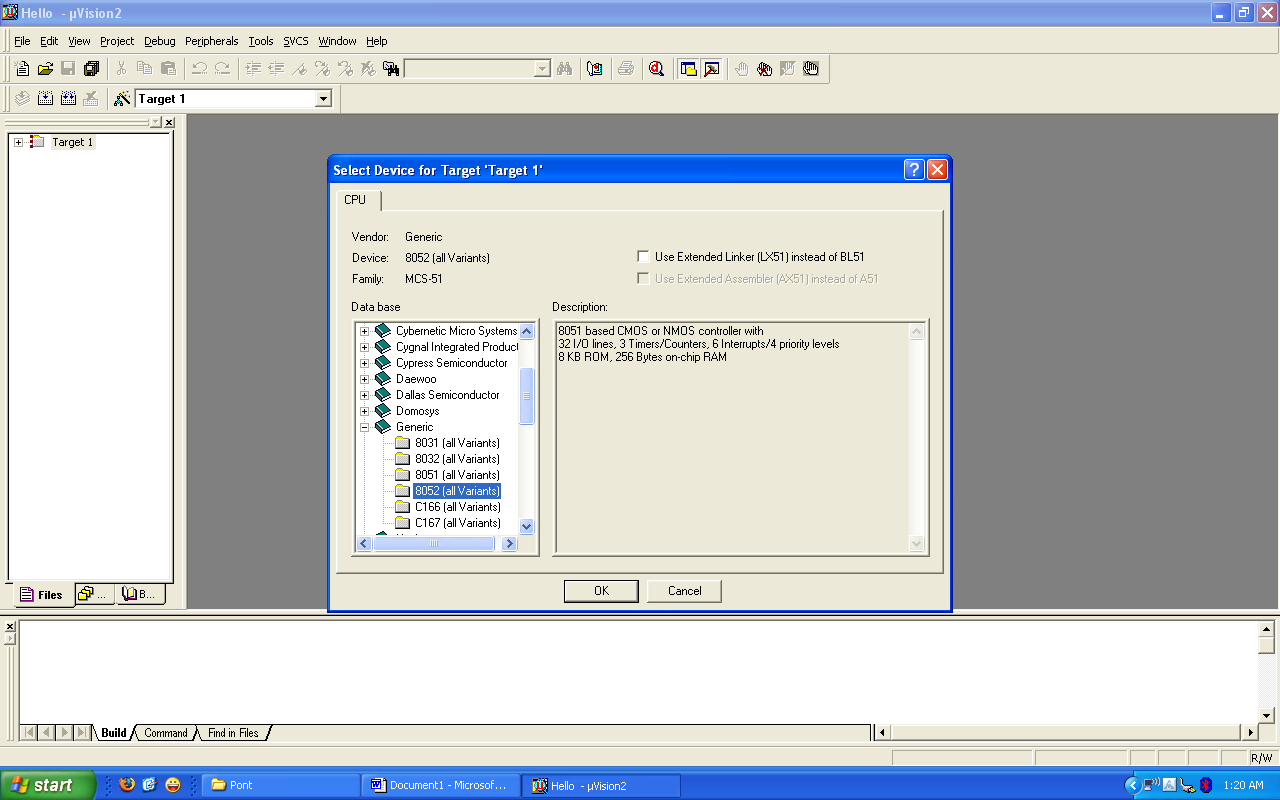
2) Go to Project – Open Project and browse for Hello in Ch03\_00 in Pont and open it.



3) Go to Project – Select Device for Target ‘Target1’

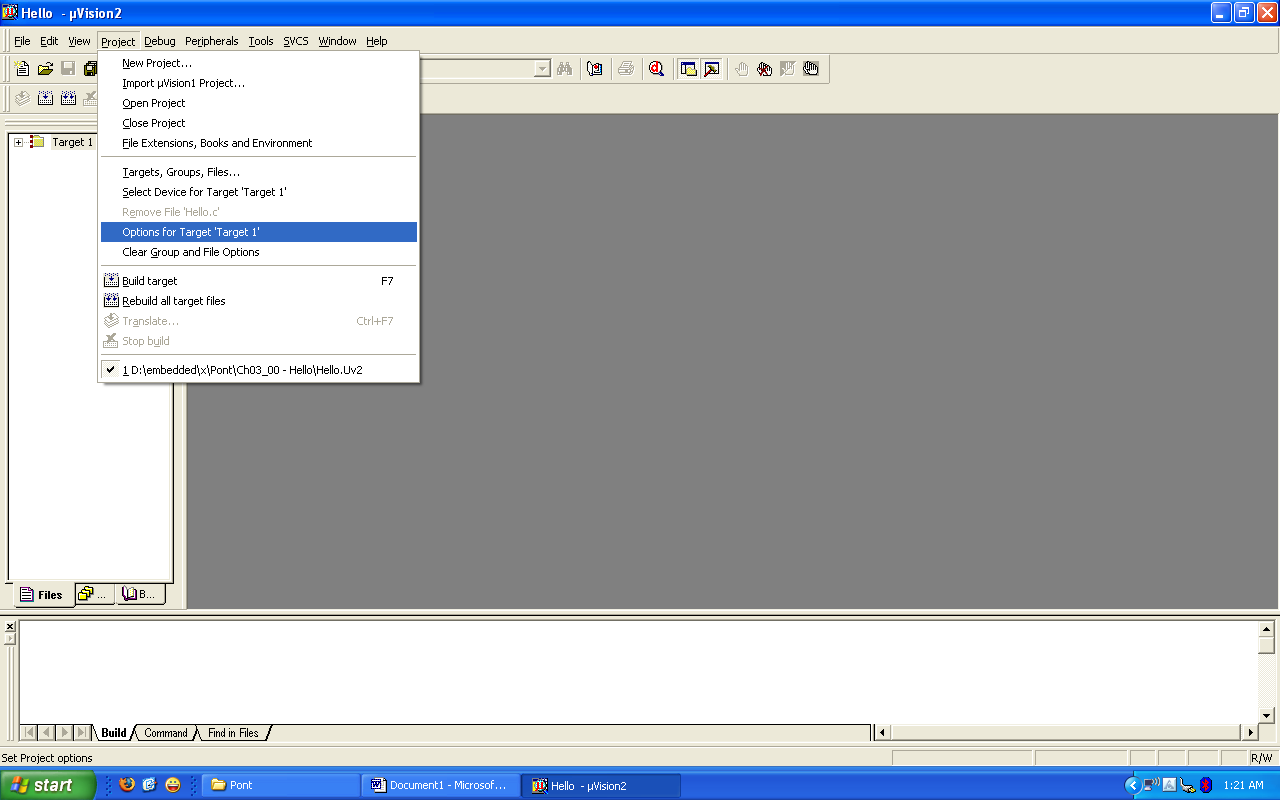


4) Select 8052(all variants) and click OK

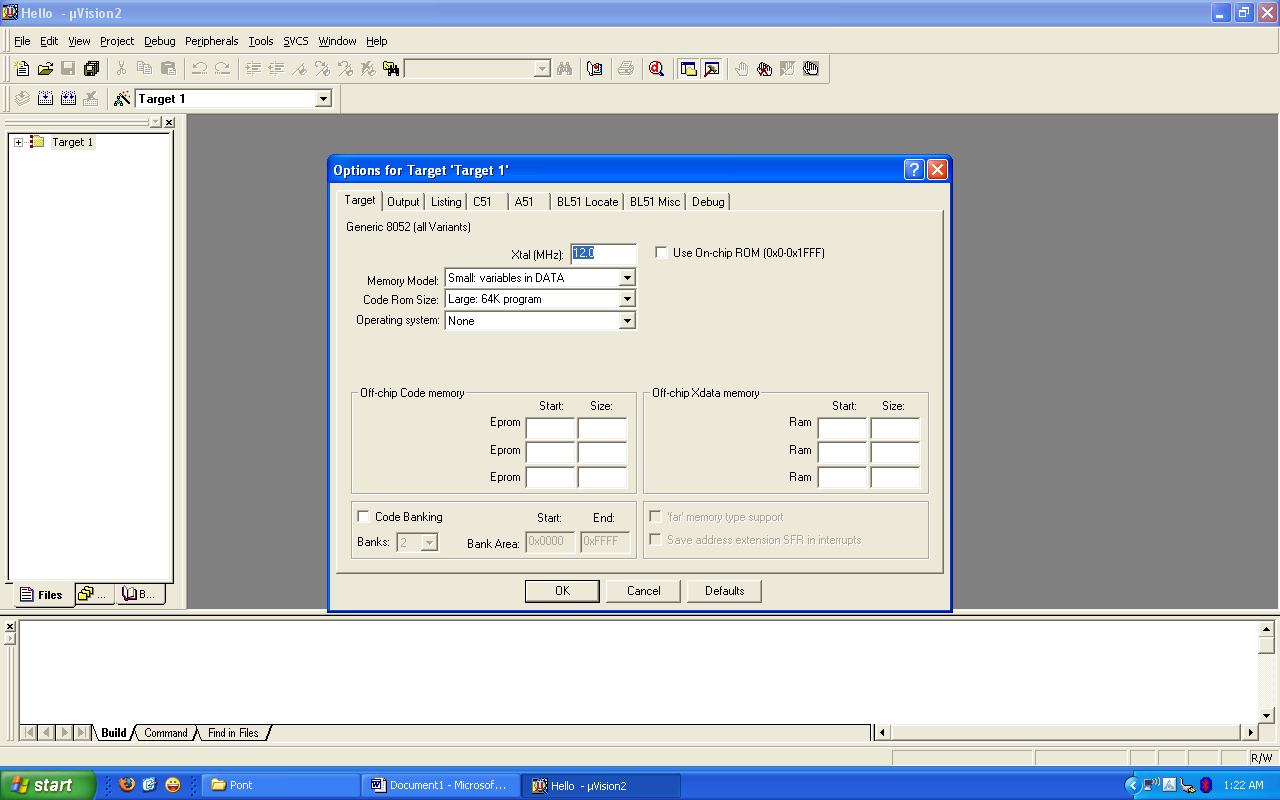


5) Now we need to check the oscillator frequency:

6) Go to project – Options for Target ‘Target1’

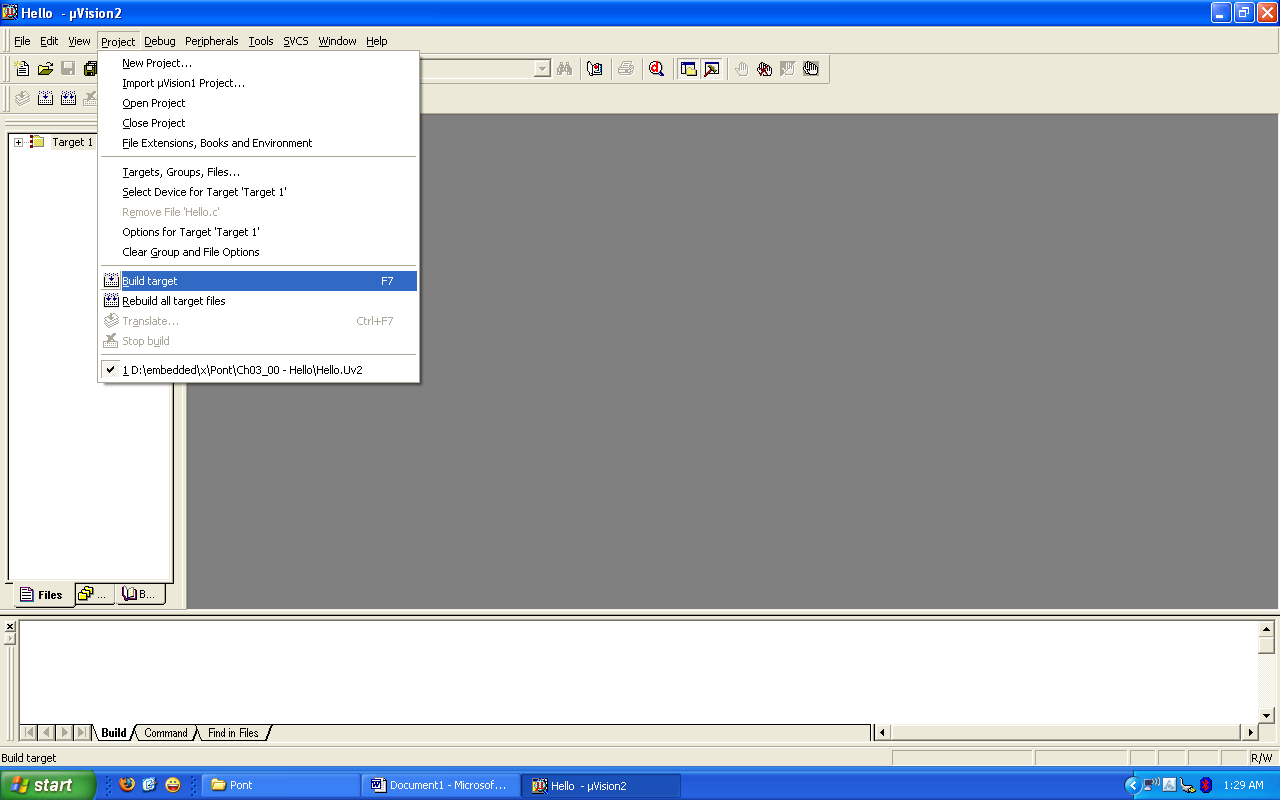


7) Make sure that the oscillator frequency is 12MHz.



8) Building the Target

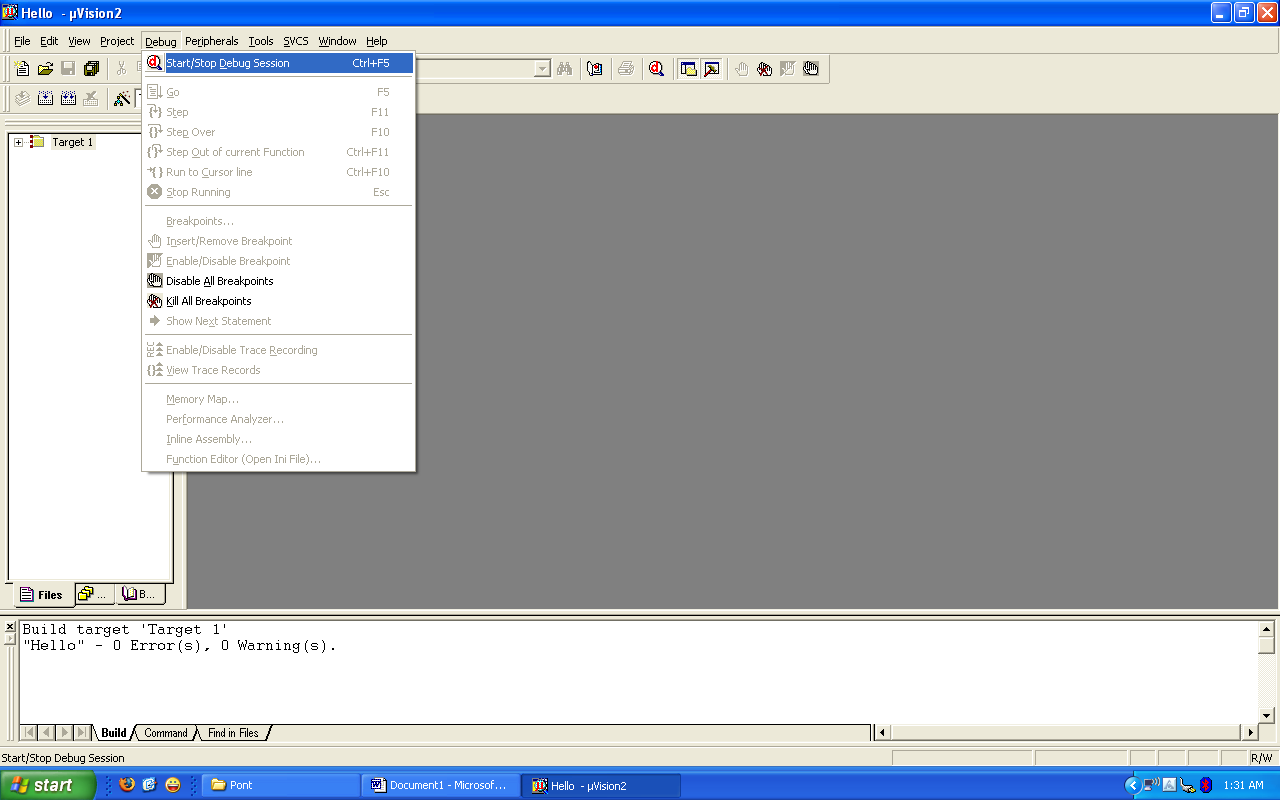
9) Build the target as illustrated in the figure below



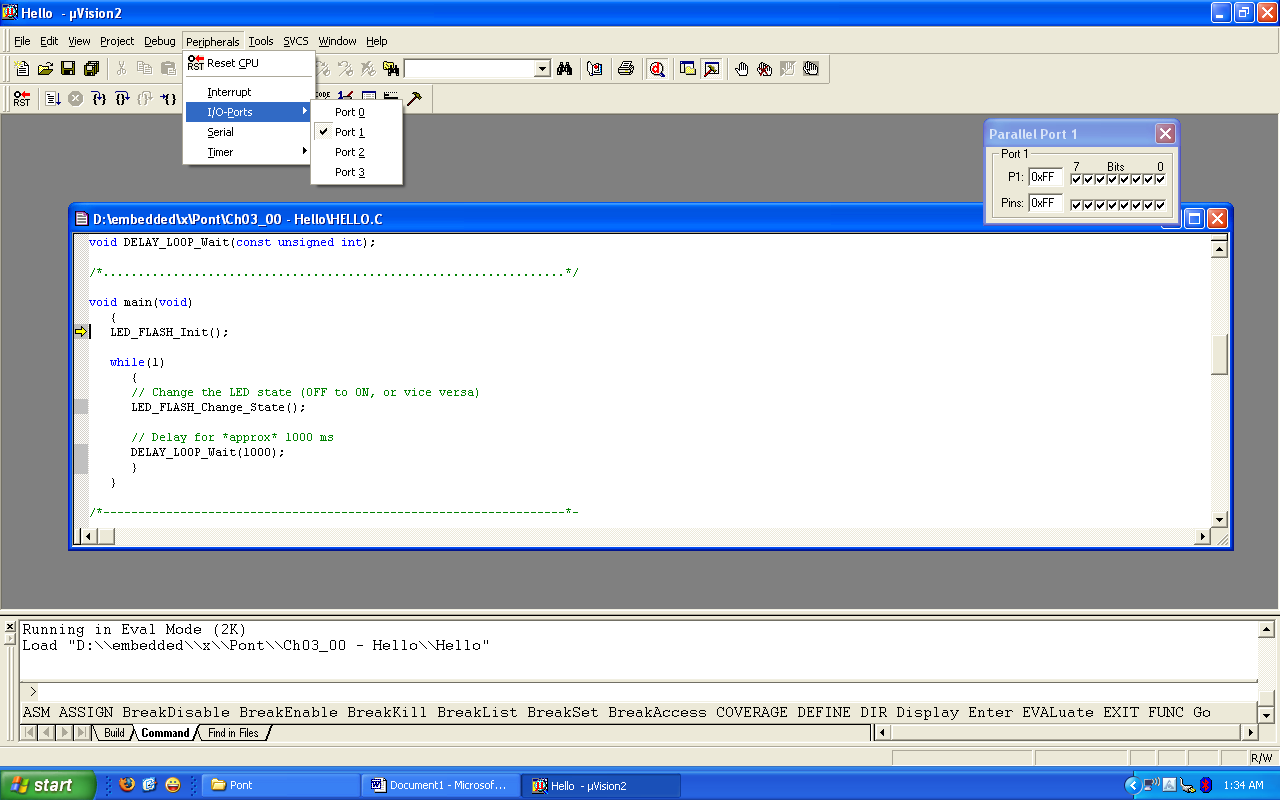
10) Running the Simulation

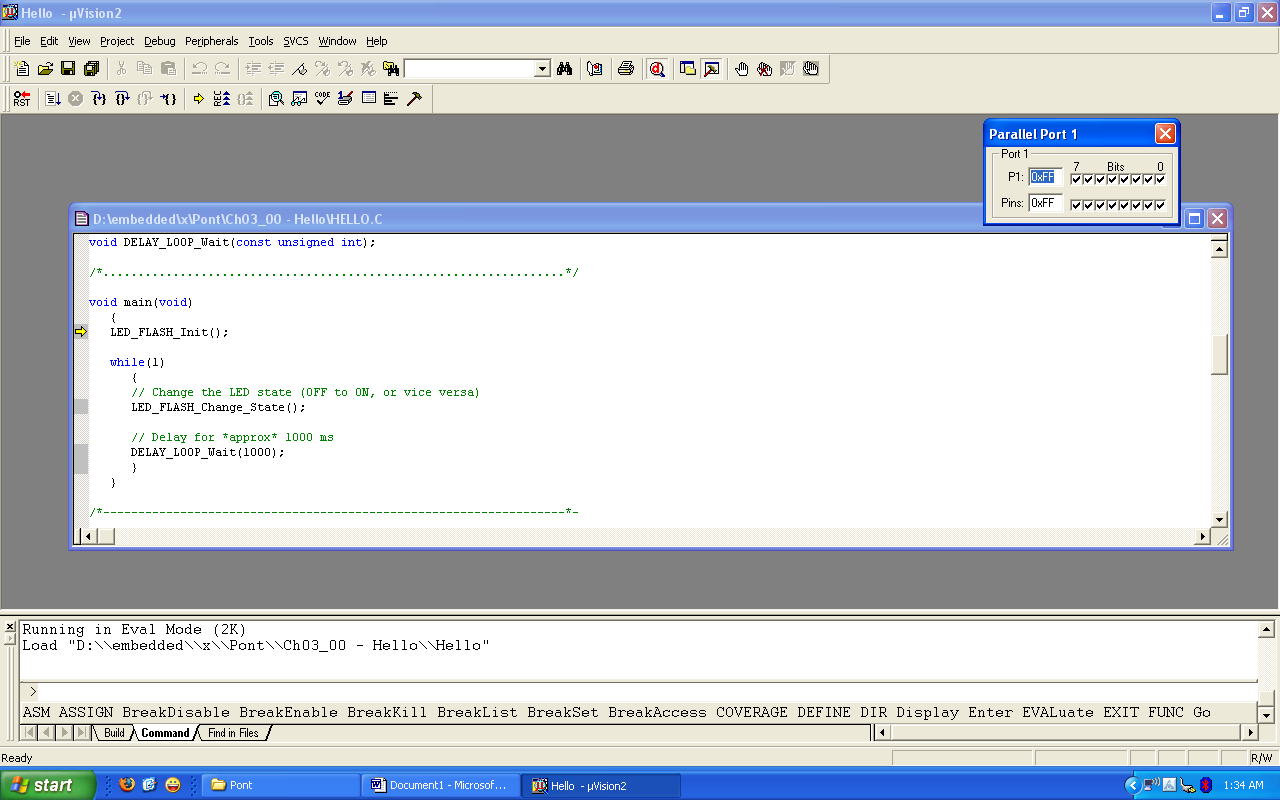
11) Having successfully built the target, we are now ready to start the debug session and run the simulator.

12) First start a debug session

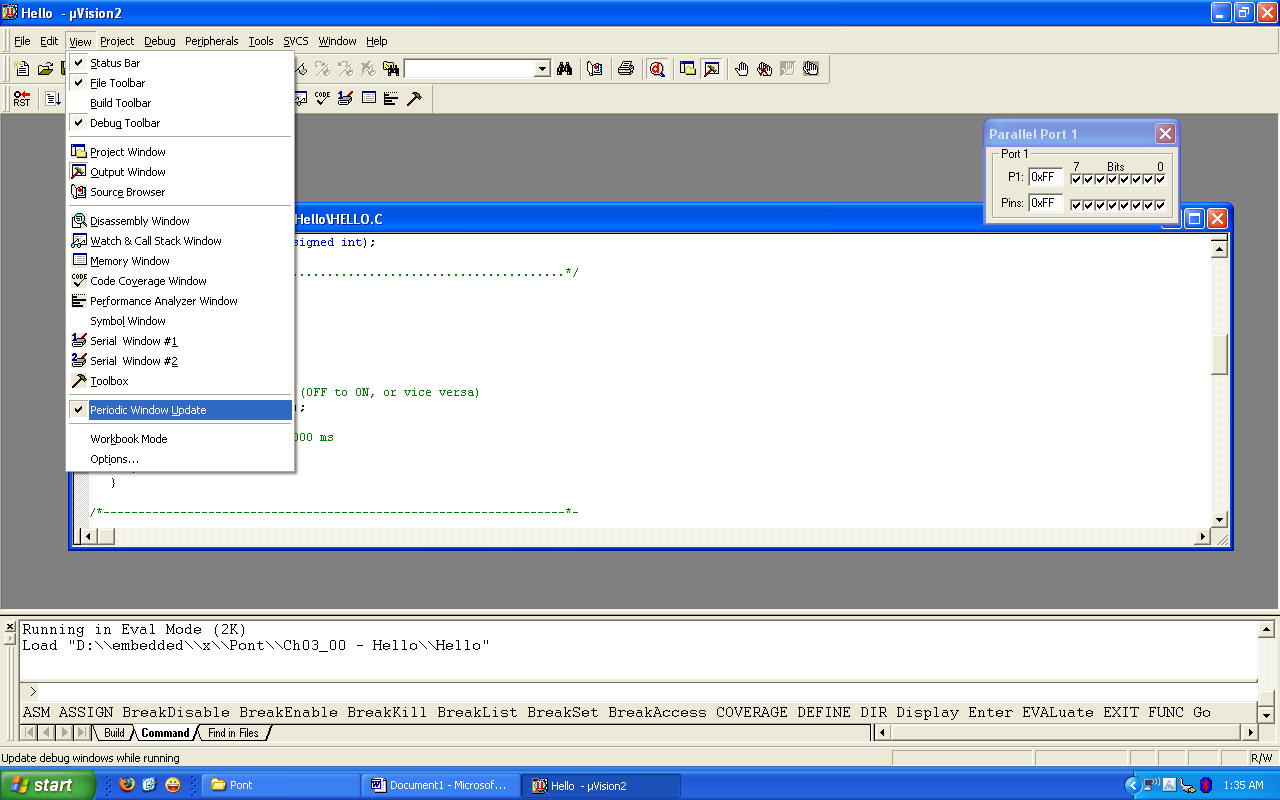


13) The flashing LED we will view will be connected to Port 1. We therefore want to observe the activity on this port

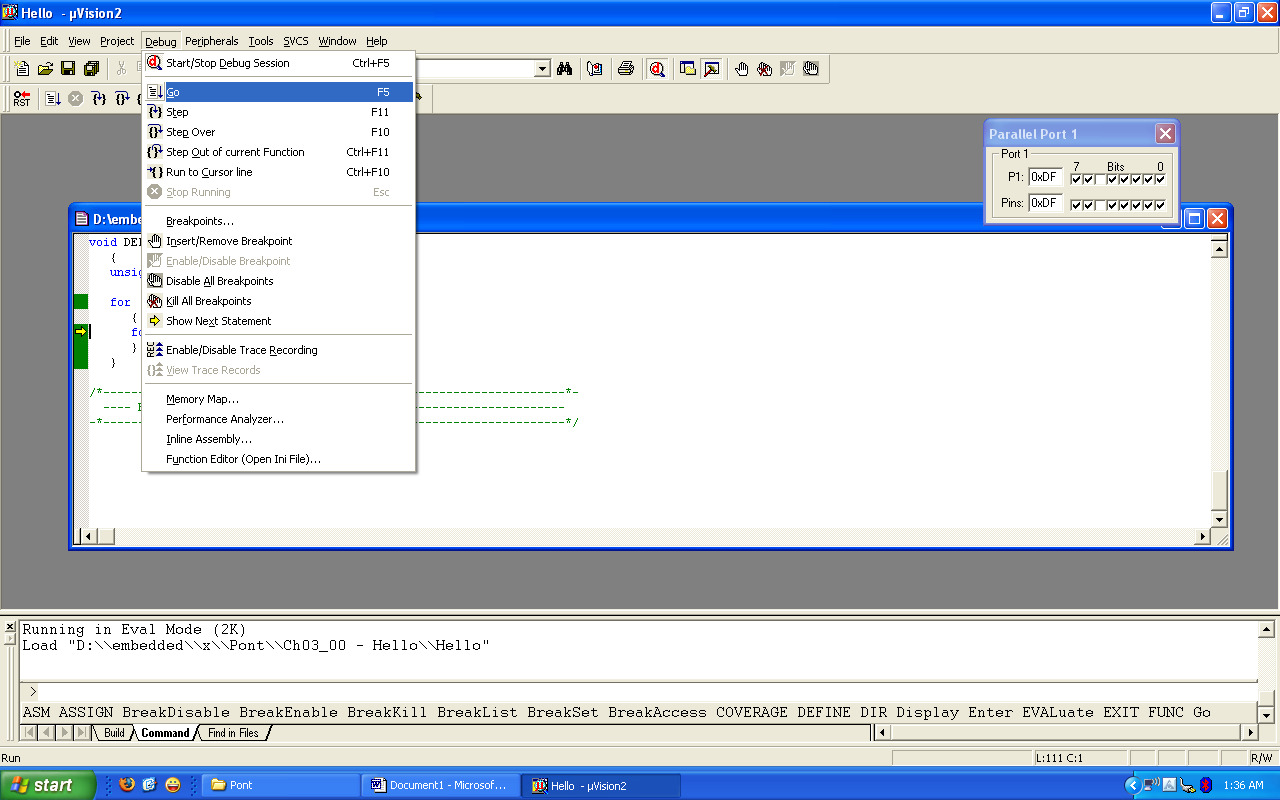




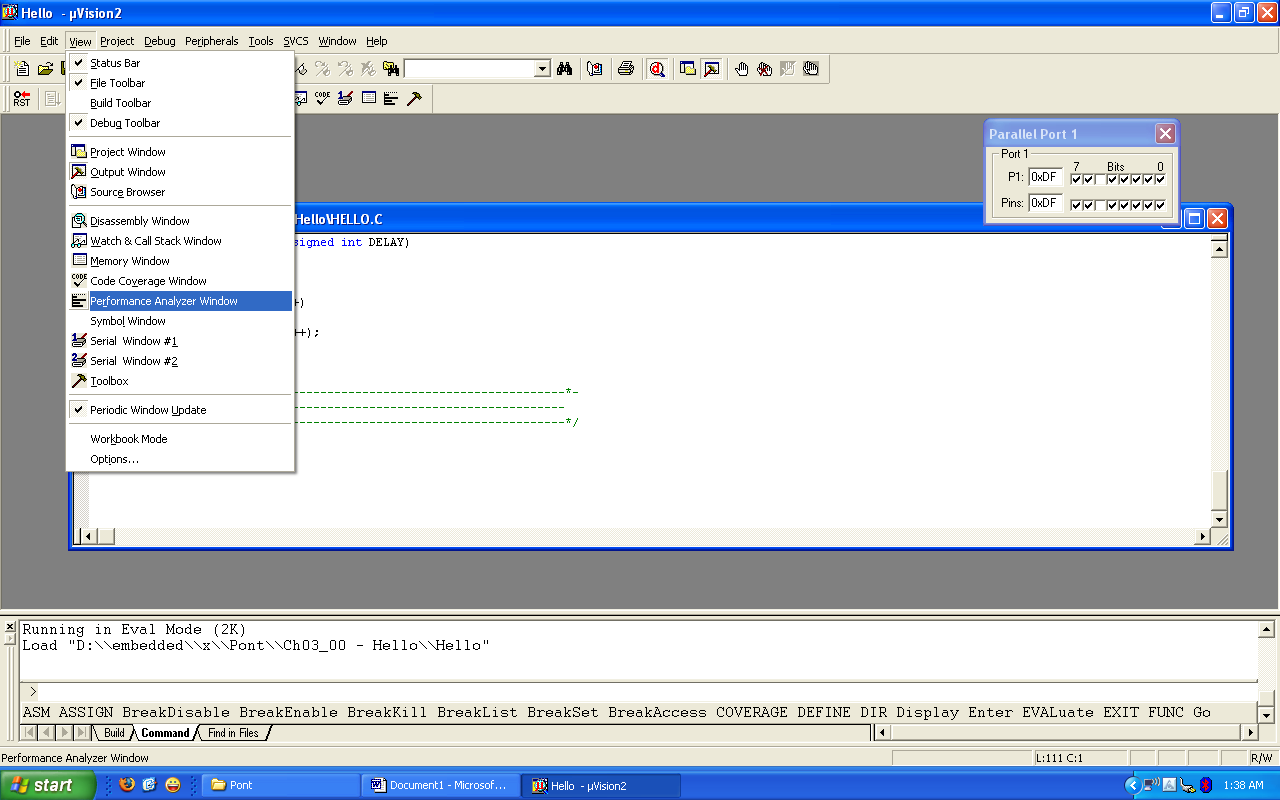
14) To ensure that the port activity is visible, we need to start the ‘periodic window update’ flag

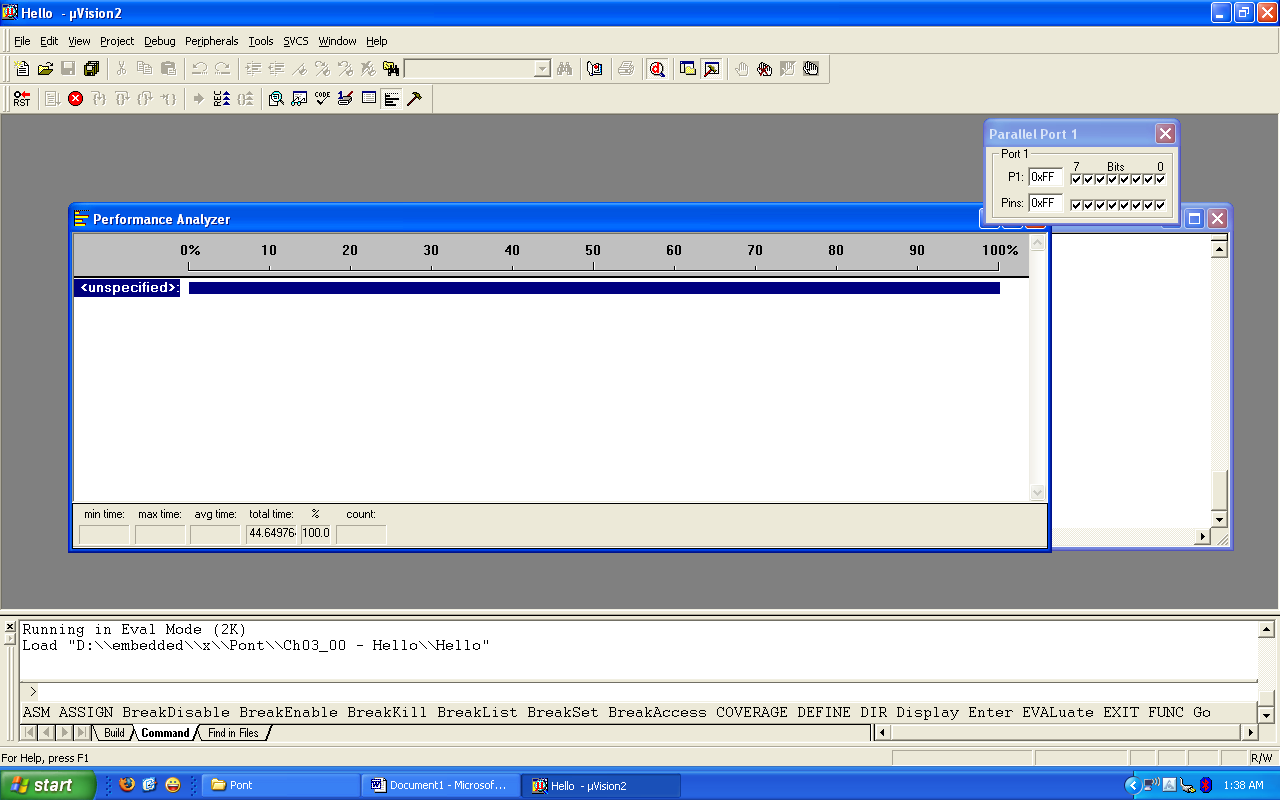


15) Go to Debug - Go

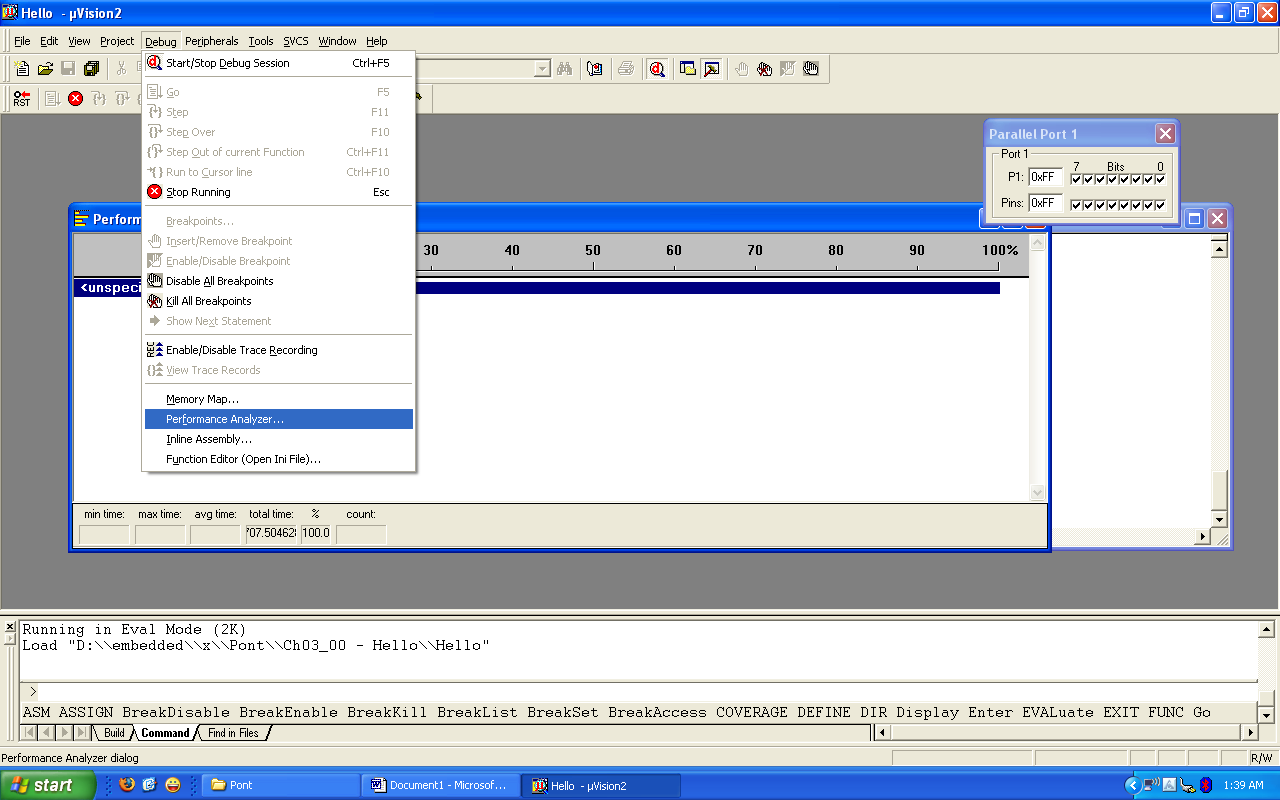


16) While the simulation is running, view the performance analyzer to check the delay durations.





17) Go to Debug – Performance Analyzer and click on it



18) Double click on DELAY\_LOOP\_WAIT in Function Symbols: and click Define button

**CHAPTER-9**

**SOURCE CODE**

#include<reg51.h>

#include<uart.h>

#include<lcddisplay.h>

#include<string.h>

sbit chr = P0^0;

sbit load1 = P0^1;

sbit fan = P0^2;

sbit tv = P0^3;

unsigned char mybyte,i,rcv,rcv1,rcv2,rcv3,rcv4,rcv5,i;

unsigned char str[5],flag1=0,flag2=0,flag3=0,flag4=0;

void main()

{

P0=0x00;

delay(50);

UART\_init();

lcd\_init();

lcdcmd(0x01);

msgdisplay("VOICE CNTRL HOME");

lcdcmd(0xc0);

msgdisplay(" AUTOMATION ");

delay(300);

start:

while(1)

{

do{

rcv=receive();

}while(rcv != 't');

i=0;

str[i]=receive();

if(strcmp(str,"TV on") == 0)

{

tv=1;

delay(150);

lcdcmd(0x01);

msgdisplay(" TV ON");

delay(100);

goto start;

}

if(strcmp(str,"TV of") == 0)

{

tv=0;

lcdcmd(0x01);

msgdisplay("TV OFF");

delay(100);

goto start;

}

if(strcmp(str,"light") == 0)

{

load1=~load1;

delay(100);

if(load1==1)

{

lcdcmd(0x01);

msgdisplay("LIGHT ON");

delay(100);

}

else

{

lcdcmd(0x01);

msgdisplay("LIGHT OFF");

delay(100);

}

goto start;

}

if(strcmp(str,"charg") == 0)

{

chr=~chr;

delay(100);

if(chr==1)

{

lcdcmd(0x01);

msgdisplay("CHARGER ON");

delay(100);

}

else

{

lcdcmd(0x01);

msgdisplay("CHARGER OFF");

delay(100);

}

goto start;

}

if(strcmp(str,"fan o") == 0)

{

fan=~fan;

delay(150);

if(fan==1)

{

lcdcmd(0x01);

msgdisplay("COOLING FAN ON");

delay(100);

}

else

{

lcdcmd(0x01);

msgdisplay("COOLING FAN OFF");

delay(100);

}

go to start;

}

}

}

#define lcd\_data P2

sbit lcd\_rs = P2^0; // Here we are using LCD in four bit mode that's why LCD's Data pins and control

sbit lcd\_en = P2^1;

void lcd\_init(void);

void lcdcmd(unsigned char value);

void lcddata(unsigned char value);

void lcd\_i2c(unsigned char value);

void msgdisplay(unsigned char b[]);

void delay(unsigned int value);

void convert(unsigned int temp1\_value);

void lcd\_init(void)

{

lcdcmd(0x02);

lcdcmd(0x02);

lcdcmd(0x28); //intialise the lcd in 4 bit mode\*/

lcdcmd(0x28); //intialise the lcd in 4 bit mode\*/

lcdcmd(0x0e); //cursor blinking

lcdcmd(0x06); //move the cursor to right side

lcdcmd(0x01); //clear the lcd

}

void lcdcmd(unsigned char value) // LCD COMMAND

{

lcd\_data=value&(0xf0); //send msb 4 bits

lcd\_rs=0; //select command register

lcd\_en=1; //enable the lcd to execute command

delay(3);

lcd\_en=0;

lcd\_data=((value<<4)&(0xf0)); //send lsb 4 bits

lcd\_rs=0; //select command register

lcd\_en=1; //enable the lcd to execute command

delay(3);

lcd\_en=0;

}

void lcddata(unsigned char value)

{

lcd\_data=value&(0xf0); //send msb 4 bits

lcd\_rs=1; //select data register

lcd\_en=1; //enable the lcd to execute data

delay(3);

lcd\_en=0;

lcd\_data=((value<<4)&(0xf0)); //send lsb 4 bits

lcd\_rs=1; //select data register

lcd\_en=1; //enable the lcd to execute data

delay(3);

lcd\_en=0;

delay(3);

}

void msgdisplay(unsigned char b[]) // send string to lcd

{

unsigned char s,count=0;

for(s=0;b[s]!='\0';s++)

{

count++;

if(s==16)

lcdcmd(0xc0);

if(s==32)

{

lcdcmd(1);

count=0;

}

lcddata(b[s]);

}

}

void delay(unsigned int value)

{

unsigned int x,y;

for(x=0;x<value;x++)

for(y=0;y<500;y++);

}

void convert(unsigned int temp1\_value)

{

unsigned int d1,d2,d3;

d3=temp1\_value%10;

temp1\_value=temp1\_value/10;

d2=temp1\_value%10;

d1=temp1\_value/10;

//lcddata(d1/10+48);

lcddata(d1+48);

lcddata(d2+48);

lcddata(d3+48);

}

**CHAPTER-10**

**ADVANTAGES & APPLICATIONS**

**ADVANTAGES**

* No need of separate remote or switch boards for controlling electrical appliances
* Protection from shock while operating
* Easy to use, children, old aged people, physically challenged people can also operate

**APPLICATIONS**

* Office automation
* Home automation
* Machines controlling system in industry
* Door access control system

**FUTURE SCOPE**

Of course, the every system has to be some upgrades then only we can do some research on the project on various aspects as well improve the technology. On this, we have to avoid the app to give commands and built more efficient and give commands in open environment, it should grab the voice. And one more thing is, obviously it’s essential to improve the security using this one, through the GPS system mainly in door access systems in homes, offices, industries etc..

**CONCLUSION**

Well, Automation of electric appliances system has successfully evaluated, This project met all the challenges using voice can automate appliances and sending the numerical numbers automate appliances too. The main aim of this project is to make your surroundings like a heaven. When work too hard, all the parts of the body were exhausted but Tongue, isn’t so it’s a better option.

**REFERENCES**

1. a b "Washing Machine Voice Control". Appliance Magazine.
2. Borzo, Jeanette (8 February 2007). "Now You're Talking". CNN Money. Retrieved 25 April 2012.
3. "Voice Control, the End of the TV Remote?". Business Week. Retrieved 1 May 2012.
4. "Windows Vista Built In Speech". Windows Vista. Retrieved 25 April 2012.
5. "Speech Operation On Vista". Microsoft.
6. "Speech Recognition Set Up". Microsoft.
7. a b "Physical and Motor Skills". Apple.
8. "DragonNaturallySpeaking PC". Nuance.
9. "DragonNaturallySpeaking Mac". Nuance.
10. a b "Voice Actions". Google.
11. "Google Voice Search For Android Can Now Be "Trained" To Your Voice". Retrieved 24 April 2012.
12. "Using Voice Command". Microsoft. Retrieved 24 April 2012.
13. a b "Using Voice Commands". Microsoft. Retrieved 27 April 2012.
14. "Siri, The iPhone 3GS & 4, iPod 3 & 4, have voice control like an express Siri, it plays music, pauses music, suffle, Facetime, and calling Features". Apple. Retrieved 27 April 2012.
15. "Siri FAQ". Apple.