

EMBEDDED SYSTEMS DESIGN LAB MANUAL

Class: B. Tech II-I Semister

Prepared by

T.Tanuja

D.Mahesh

DEPARTMENT OF ELECTRONICS AND COMMUNICATION AND ENGINEERING

**CERTIFICATE**

This is to certify that this manual is a bonafide record of practical work in the **Embedded Systems Design** in **First Semester of IV yearB.Tech (ECE) programme** during the academic year **2018-19**.This book is prepared by **Ms.T.Tanuja (Asst.Professor), Mr.D.Mahesh(Asst.Professor),** Department of Electronics and Communication Engineering.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Content** | **Page No:** |
|  | Preface |  |
|  | Acknowledgement |  |
|  | General Instructions |  |
|  | Safety Precautions |  |
|  | Institute Vision and Mission |  |
|  | Department Vision,Mission, Programme Educational Objectives and Specific  Outcomes |  |
|  | Programme Outcomes |  |
|  | Course Structure, Objectives &Outcomes |  |
|  | Experiments Learning Outcomes |  |
| 1 | Write a program to blink an LED using Keil and Proteus software |  |
| 2 | Write a program to blink an LED continuously with a delay using Keil and Proteus software |  |
| 3 | Write a generate a square wave at pin p1.0 of 8051 using timer |  |
| 4 | Write a program to Generate a Stepper motor using Keil and Proteus software |  |
| 5 | Write a program showing LCD displays using Keil and Proteus software |  |
| 6 | Write a program for DC motor using Keil and Proteus software |  |
| 7 | Write a program Switch ON the LED by pressing the key ‘1’ on the keypad |  |
| 8 | Write a program to take parallel input from port p1 convert it into serial and send it via p0.0  . |  |
| 9 | Write a program for Traffic Light signal |  |
| 10 | Write a program to simultaneously transmit and receive data |  |

***PREFACE***

This book “Embedded Systems Design” lab manual is intended to teach the design and analysis of basics of Embedded Systems and their implementation in designing in Embedded Systems . Readers of this book need only be familiar with the basics of keil and peoteus software. The “Embedded systems ” became an inevitable part in todays world Therefore, for proper development of “Embedded Systems” skills among the students this practical manual has been prepared. The manual contains the exercise programs and their solution for easy & quick understanding of the students. We hope that this practical lab manual will be helpful for students of Electronics & Communication Engineering for understanding the subject from the point of view of applied aspects. There is always scope for improvement in the manual. We would appreciate to receive valuable suggestions from readers and users for future use.

By,

T.Tanuja

D.Mahesh

**ACKNOWLEDGEMENT**

It was really a good experience, working with ***Embedded Sytems*** lab. First we would like to thank Mr.K.Naga Bhushan, Assoc.Professor, HOD of Department of Electronics and Communication Engineering, Marri Laxman Reddy Institute of technology & Management for his concern and giving the technical support in preparing the document.

We are deeply indebted and gratefully acknowledge the constant support and valuable patronage of Dr.R.Kotaiah, Director, Marri Laxman Reddy Institute of technology & Management for giving us this wonderful opportunity for preparing the ***Embedded Systems***  laboratory manual.

We express our hearty thanks to Dr.K.Venkateswara Reddy, Principal, Marri Laxman Reddy Institute of technology & Management, for timely corrections and scholarly guidance.

At last, but not the least I would like to thanks the entire ECE Department faculties those who had inspired and helped us to achieve our goal.

**By,**

**T.Tanuja,**

**D.Mahesh,**

**GENERAL INSTRUCTIONS**

1. Students are instructed to come to Embedded Systems laboratory on time. Late comers are not entertained in the lab.

2. Students should be punctual to the lab. If not, the conducted experiments will not be repeated.

3. Students are expected to come prepared at home with the experiments which are going to be performed.

4. Students are instructed to display their identity cards before entering into the lab.

5. Students are instructed not to bring mobile phones to the lab.

6. Any damage/loss of system parts like keyboard, mouse during the lab session, it is student’s responsibility and penalty or fine will be collected from the student.

7. Students should update the records and lab observation books session wise. Before leaving the lab the student should get his lab observation book signed by the faculty.

8. Students should submit the lab records by the next lab to the concerned faculty members in the staffroom for their correction and return.

9. Students should not move around the lab during the lab session.

10. If any emergency arises, the student should take the permission from faculty member concerned in written format.

11. The faculty members may suspend any student from the lab session on disciplinary grounds.

12. Never copy the output from other students. Write down your own outputs.

**INSTITUTION VISION AND MISSION**

**VISION**

To be as an ideal academic institution by graduating talented engineers to be ethically strong, competent with qualityresearch and technologies

**MISSION**

To fulfill the promised vision through the following strategic characteristics and aspirations:

* Utilize rigorous educational experiences to produce talented   engineers
* Create an atmosphere that facilitates the success of students
* Programs that integrate global awareness, communication skills and Leadership qualities
* Education and Research partnership with institutions and industries to prepare the students for interdisciplinary research

**DEPARTMENT VISION, MISSION , PROGRAMME EDUCATIONAL OBJECTIVES AND SPECIFIC OUTCOMES**

**DEPARTMENT VISION:**

Imparting quality technical education through research, innovation and team work for a lasting technology development in the area of Electronics and Communication Engineering.

**MISSION:**

To develop a strong centre of excellence for education and research with excellent infrastructure and well qualified faculties to instill in them a passion for knowledge.

**To achieve the Mission the department will**

1. Establish a unique learning environment to enable the students to face the challenges of the Electronics and Communication Engineering field.
2. Promote the establishment of centre of excellence in niche technology areas to nurture the spirit of innovation and creativity among faculty and students.
3. Provide ethical and value based education by promoting activities addressing the societal needs.
4. Enable students to develop skills to solve complex technological problems of current times and also provide a framework for promoting collaborative and multidisciplinary activities.

**PROGRAMME EDUCATIONAL OBJECTIVES**

1. **PEO 1:** have successful **careers in Industry.**
2. **PEO 2:** show excellence in **higher studies/ Research.**
3. **PEO 3:** Show good competency towards **Entrepreneurship.**

**COURSE OUTCOMES**

**CO1:** The student will learn the internal organization of popular 8086/8051 microprocessors/microcontrollers.

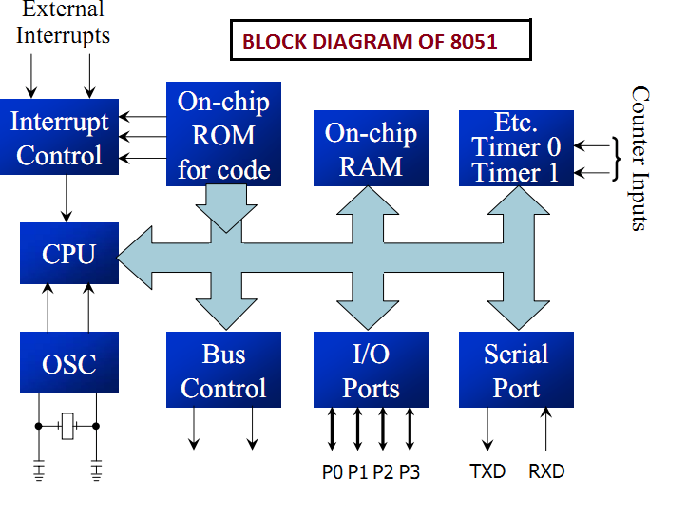
**CO2:** The student will learn hardware and software interaction and integration.

**CO3:** The students will learn the design of microprocessors microcontrollers-based systems.

**CO4 :** write and execute simple embedded c language programs.

**PROGRAM OUTCOMES**

|  |  |
| --- | --- |
| **a** | An ability to apply knowledge of Science, Mathematics, Engineering & Computing fundamentals for the solutions of Complex Engineering problems |
| **b** | An ability to identify, formulates, research literature and analyze complex engineering problems using first principles of mathematics and engineering sciences. |
| **c** | An ability to design solutions to complex process or program to meet desired needs. |
| **d** | Ability to use research-based knowledge and research methods including design of experiments to provide valid conclusions. |
| **e** | An ability to use appropriate techniques, skills and tools necessary for computing practice. |
| **f** | Ability to apply reasoning informed by the contextual knowledge to assess social issues, consequences & responsibilities relevant to the professional engineering practice. |
| **g** | Ability to understand the impact of engineering solutions in a global, economic, environmental, and societal context with sustainability. |
| **h** | An understanding of professional, ethical, Social issues and responsibilities. |
| **i** | An ability to function as an individual, and as a member or leader in diverse teams and in multidisciplinary settings. |
| **j** | An ability to communicate effectively on complex engineering activities within the engineering community. |
| **k** | Ability to demonstrate and understanding of the engineering and management principles as a member and leader in a team. |
| **l** | Ability to engage in independent and lifelong learning in the context of technological change. |

INTRODUCTION

# The 8051 PIN DIAGRAM

**Port 0 bit 0(Address/data 0)**

**Port 0 bit 1(Address/data 1)**

**Port 0 bit 2(Address/data 2)**

**Port 0 bit 3(Address/data 3)**

**Port 0 bit 4(Address/data 4)**

**Port 0 bit 5(Address/data 5)**

**Port 0 bit 6(Address/data 6)**

**Port 0 bit 7(Address/data 7)**

# 

**Crystal Input 2**

**CrystalInput 1**

**Port 2 bit 7 (Address 15)**

**Port 2 bit 6 (Address 14)**

**Port 2 bit 5 (Address 13)**

**Port 2 bit 4 (Address 12) Port 2 bit 3 (Address 11)**

**Port 2 bit 2 (Address 10)**

**Port 2 bit 1 (Address 9)**

**Port 2 bit 0 (Address 8)**

**EA:-External Enable (EPROM Programming voltage)**

**ALE:-Address latch Enable**

**PSEN:-Program store Enable**

**Port 3 bit 0 (Receive data)**

**Port 3 bit 1 (XMIT Data)**

**Port 3 bit 2 (Interrupt 0)**

**Port 3 Bit 3(Interrupt 1)**

**Port 3 bit 4(Timer 0 Input)**

**Port 3 bit 5(Timer 1 input)**

**Port 3 bit 6(Write Strobe)**

**Port 3 Bit 7(Read Strobe)**

**P 1.0 port 1 Bit 0**

**P1.1 Port 1 Bit 1**

**P1.2 port 1 bit 2**

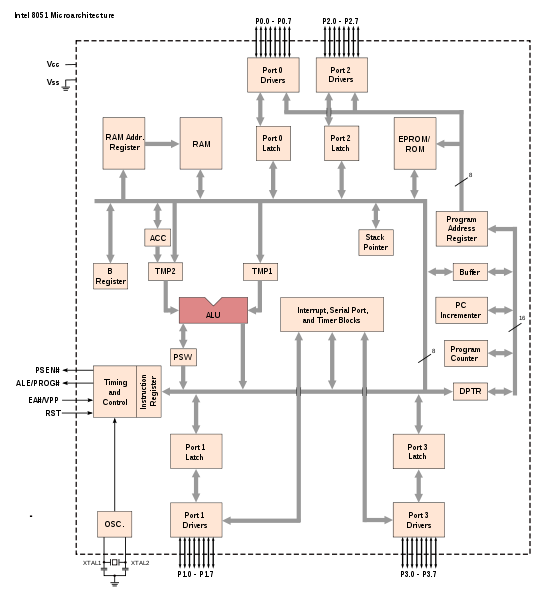
**P1.3 port 1 bit 3**

**P1.4 Port 1 bit 4**

**P1.5 Port 1 bit 5**

**P1.6 port 1 bit 6  
P1.7 port 1 bit 7**

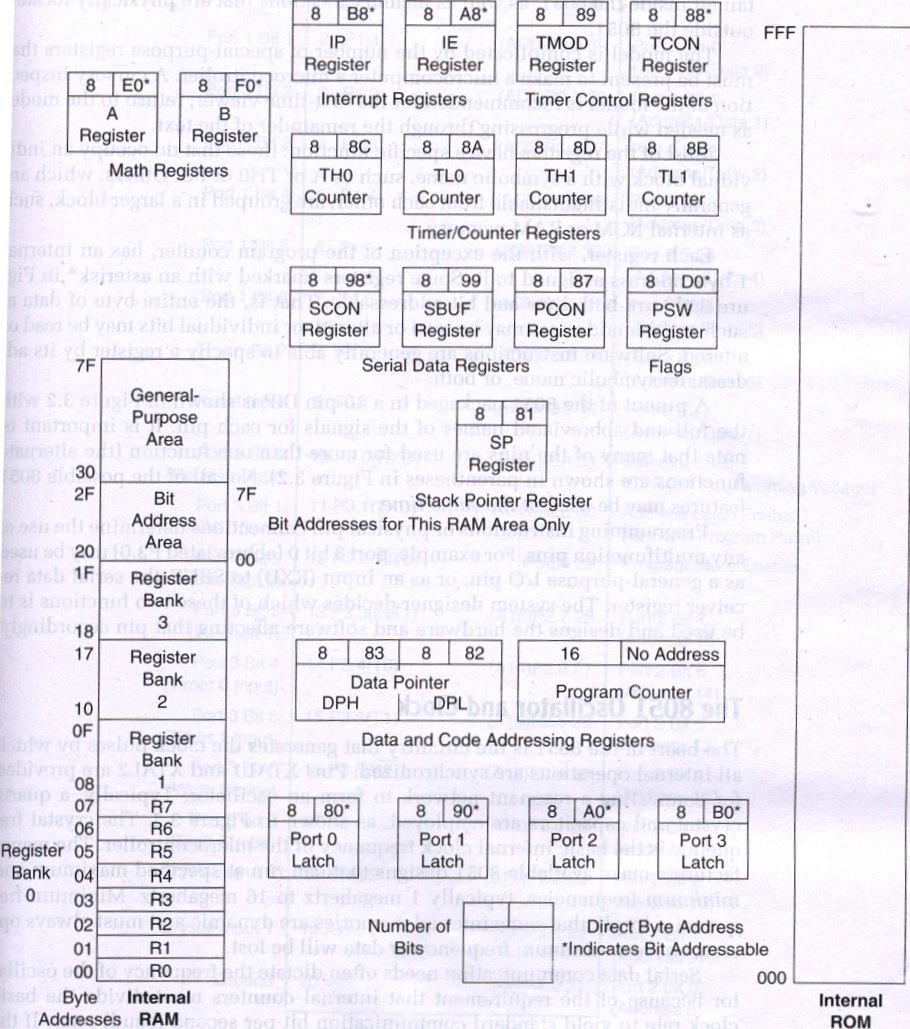
# The 8051 Architecture

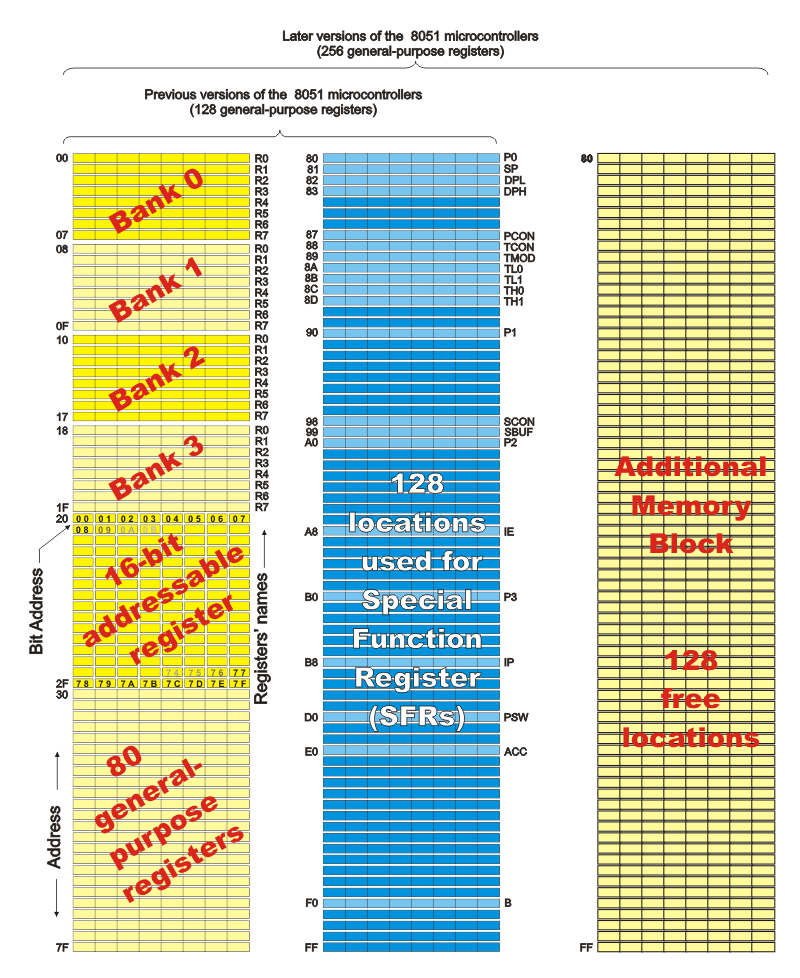


### **8051 Specific Features**

* The 8051 architecture provides many functions ([CPU](http://en.wikipedia.org/wiki/Central_processing_unit), [RAM](http://en.wikipedia.org/wiki/Random_access_memory), [ROM](http://en.wikipedia.org/wiki/Read-only_memory), [I/O](http://en.wikipedia.org/wiki/Input/output), [interrupt](http://en.wikipedia.org/wiki/Interrupt) logic, [timer](http://en.wikipedia.org/wiki/Timer), etc.) in a single [package](http://en.wikipedia.org/wiki/Integrated_circuit_packaging)
* 8-bit [ALU](http://en.wikipedia.org/wiki/Arithmetic_logic_unit), Accumulator and 8-bit Registers; hence it is an [8-bit](http://en.wikipedia.org/wiki/8-bit)[microcontroller](http://en.wikipedia.org/wiki/Microcontroller)
* 8-bit [data bus](http://en.wikipedia.org/wiki/Data_bus) – It can access 8 bits of data in one operation
* 16-bit [address bus](http://en.wikipedia.org/wiki/Address_bus) – It can access 216 memory locations – 64 [KB](http://en.wikipedia.org/wiki/Kilobyte) (65536 locations) each of RAM and ROM
* On-chip RAM – 128 [bytes](http://en.wikipedia.org/wiki/Bytes) (data memory)
* On-chip ROM – 4 Kbyte (program memory)
* Four [byte](http://en.wikipedia.org/wiki/Byte) bi-directional [input/output](http://en.wikipedia.org/wiki/Input/output) port
* UART ([serial port](http://en.wikipedia.org/wiki/Serial_port))
* Two 16-bit Counter/[timers](http://en.wikipedia.org/wiki/Timer)
* Two-level [interrupt](http://en.wikipedia.org/wiki/Interrupt) priority and [Power saving](http://en.wikipedia.org/wiki/Power_management) mode (on some derivatives)

# The 8051 Programming Model





**Keil** was founded in 1982 by Günter and Reinhard Keil, initially as a German [GbR](https://en.wikipedia.org/w/index.php?title=GbR&action=edit&redlink=1). In April 1985 the company was converted to *Keil Elektronik* [*GmbH*](https://en.wikipedia.org/wiki/GmbH) to market add-on products for the development tools provided by many of the [silicon vendors](https://en.wikipedia.org/wiki/Semiconductor_sales_leaders_by_year). Keil implemented the first [C](https://en.wikipedia.org/wiki/C_%28programming_language%29) [compiler](https://en.wikipedia.org/wiki/Compiler) designed from the ground-up specifically for the [8051](https://en.wikipedia.org/wiki/Intel_8051)[microcontroller](https://en.wikipedia.org/wiki/Microcontroller).

Keil provides a broad range of development tools like [ANSI](https://en.wikipedia.org/wiki/ANSI) “[C](https://en.wikipedia.org/wiki/C_%28programming_language%29)” [compiler](https://en.wikipedia.org/wiki/Compiler), [macro assemblers](https://en.wikipedia.org/wiki/Assembly_language#Assembler), [debuggers](https://en.wikipedia.org/wiki/Debuggers) and simulators, [linkers](https://en.wikipedia.org/wiki/Linker_%28computing%29), [IDE](https://en.wikipedia.org/wiki/Integrated_development_environment), library managers, [real-time operating systems](https://en.wikipedia.org/wiki/Real-time_operating_systems) and [evaluation boards](https://en.wikipedia.org/wiki/Evaluation_boards) for Intel 8051, [Intel MCS-251](https://en.wikipedia.org/wiki/Intel_MCS-251), ARM, and [XC16x](https://en.wikipedia.org/w/index.php?title=XC16x&action=edit&redlink=1)[[1]](https://en.wikipedia.org/wiki/Keil_%28company%29#cite_note-1)/[C16x](https://en.wikipedia.org/wiki/C16x)[[2]](https://en.wikipedia.org/wiki/Keil_%28company%29#cite_note-2)/[ST10](https://en.wikipedia.org/wiki/ST10)[[3]](https://en.wikipedia.org/wiki/Keil_%28company%29#cite_note-3) families.

In October 2005, Keil (**Keil Elektronik GmbH** in [Munich](https://en.wikipedia.org/wiki/Munich), [Germany](https://en.wikipedia.org/wiki/Germany), and **Keil Software, Inc**. in [Plano](https://en.wikipedia.org/wiki/Plano,_Texas), [Texas](https://en.wikipedia.org/wiki/Texas)) were acquired by ARM.

The **Keil Software development tools** listed below are programs you use to compile your C code, assemble your assembly source files, link and locate object modules and libraries, create **HEX** files, and debug your target program. **μVision** for Windows™ is an Integrated Development Environment that combines project management, source code editing, and program debugging in one single, powerful environment.

The **ARM7ANSI** Optimizing C Compiler creates re locatable object modules from your C source code. The **ARM Macro Assembler** creates re locatable object modules from your LPC21XX assembly source code. The Linker/Locator combines re-locatable object modules created by the Compiler and the Assembler into absolute object modules. The Library Manager combines object modules into libraries that may be used by the linker. The **Object-HEX Converter** creates Intel **HEX files** from absolute object modules.

The **Keil development tools for ARM** offer numerous features and advantages that help you quickly and successfully develop embedded applications. They are easy to use and are guaranteed to help you achieve your design goals. The **μVision IDE and Debugger** is the central part of the **Keil ARM development tools. µVision** offers a Build Mode and a Debug Mode.

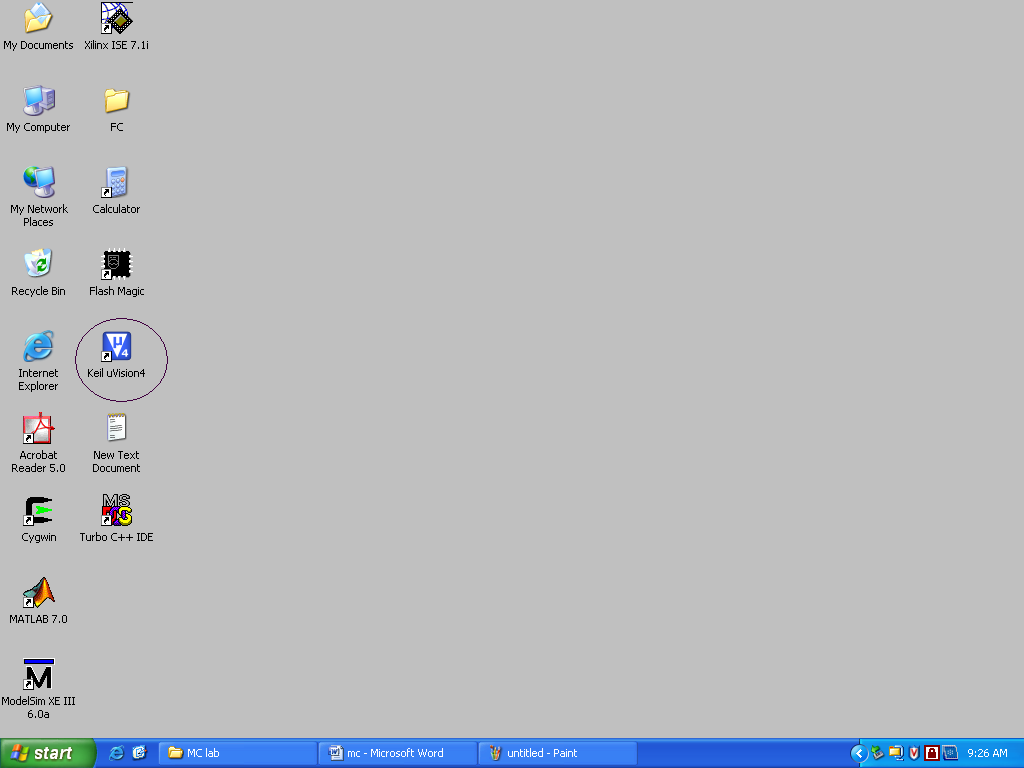
In the μVision **Build Mode** you maintain the project files and generate the application. μVision uses either the GNU or ARM ADS/Real View™ development tools. In the μVision **Debug Mode** you verify your program either with a powerful CPU and peripheral simulator that connects the debugger to the target system. The ULINK allows you also to download your application into Flash ROM of your target system. The **μVision IDE from Keil** combines project management, make facilities, source code editing, program debugging, and complete simulation in one powerful environment. The **μVision development** platform is easy-to-use and helping you quickly create embedded programs that work. The μVision editor and debugger are integrated in a single application that provides a seamless embedded project development environment.

The **μVision IDE** (Integrated Development Environment) is the easiest way for most developers to create embedded applications using the **Keil development tools.** To launch µVision, click on the icon on your desktop or select **Keil μVision** which version you are using from the Start Menu.

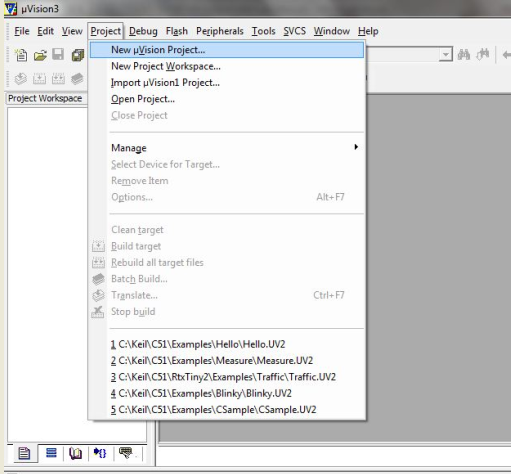
STEPS FOR EXECUTING THE SOFTWARE PROGRAM

## STEPS FOR EXECUTING THE SOFTWARE PROGRAM:

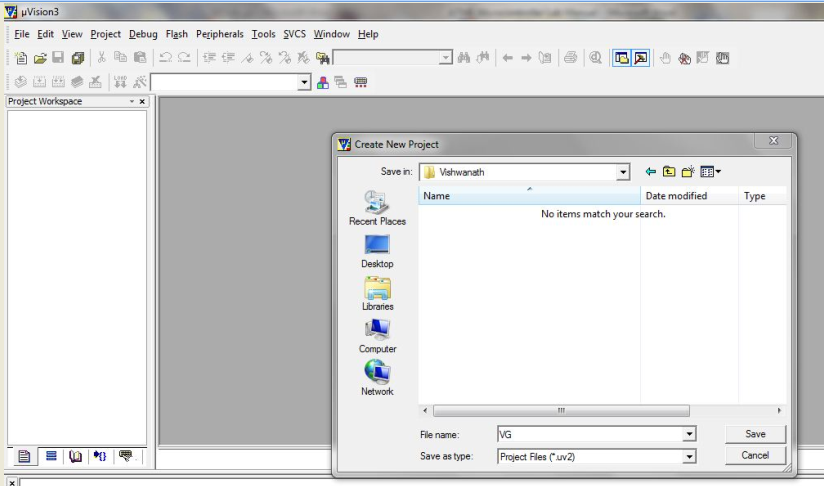
##### **STEP 1: Select the Kiel µVision software.**



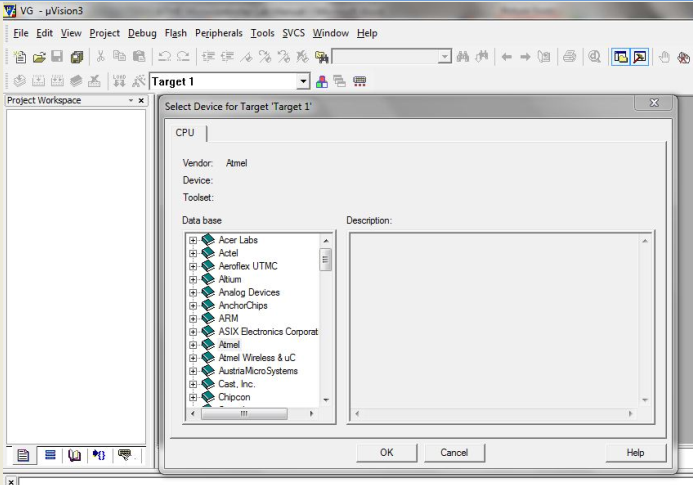
##### STEP 2: Select “Project” -> “New µVision Project”.



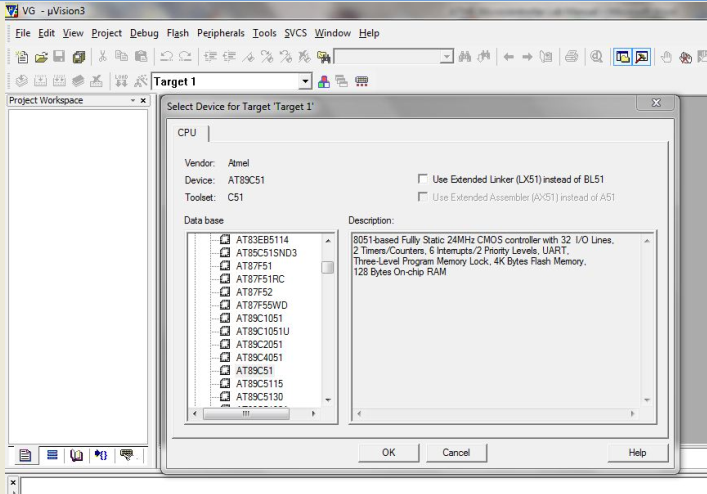
##### STEP 3: Create new project by entering your “File name” and then “Save” your file

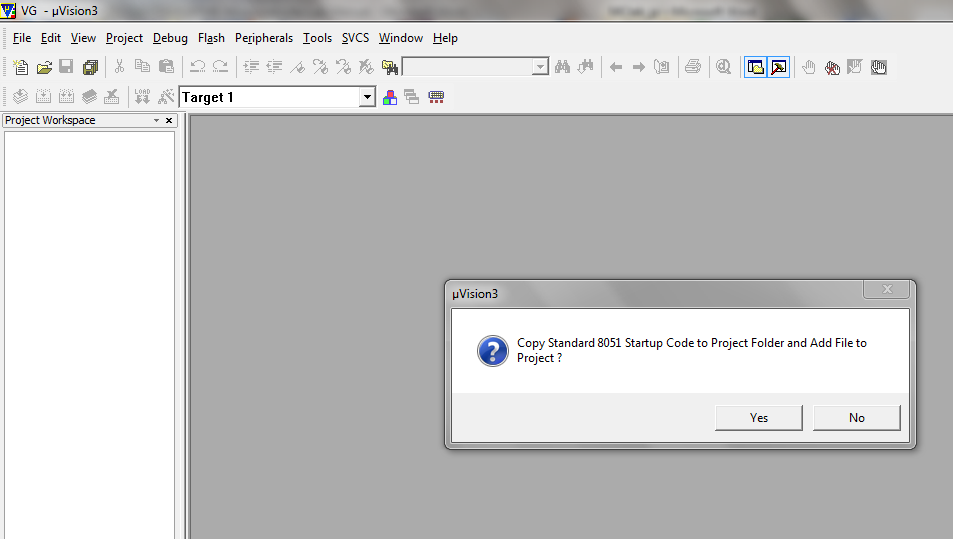


##### STEP 4: Choose “Atmel” microcontroller from the database

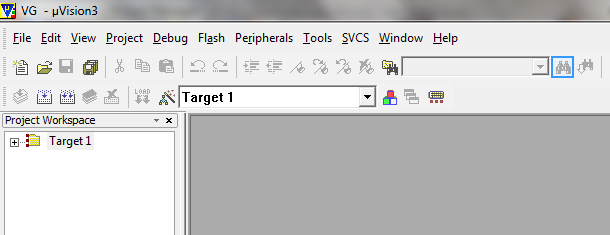


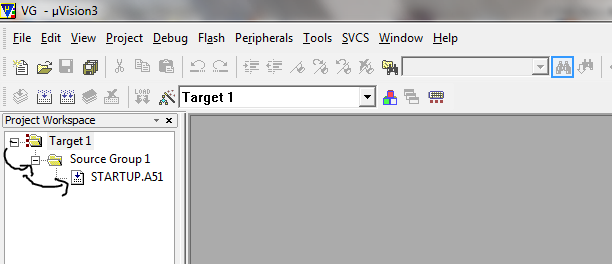
##### STEP 5: Select “AT89C51” µC and click “OK” and then “YES” and “YES”



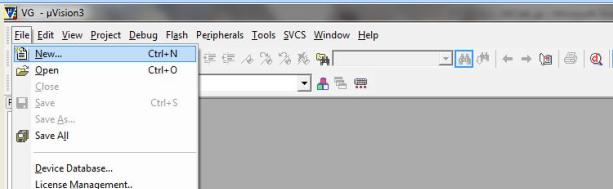


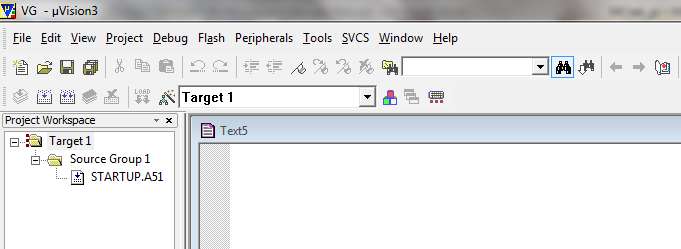
##### STEP 6: Make sure that “STARTUP.A51” file is added to the target.



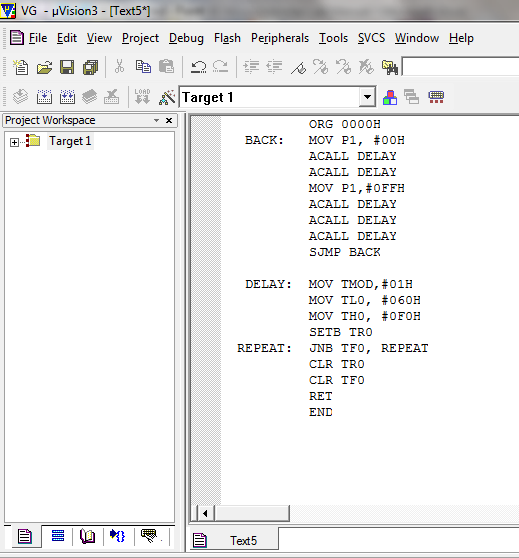


##### STEP 7: Go to “File” and select “New” for text (program) editing window.

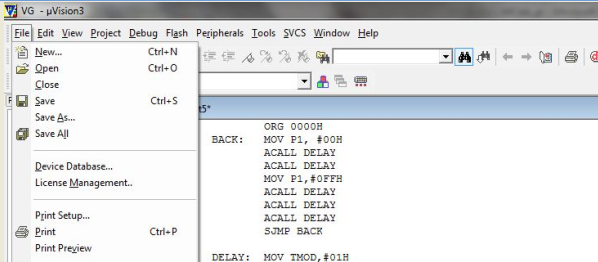




##### STEP 8: Type your program in the editing window.

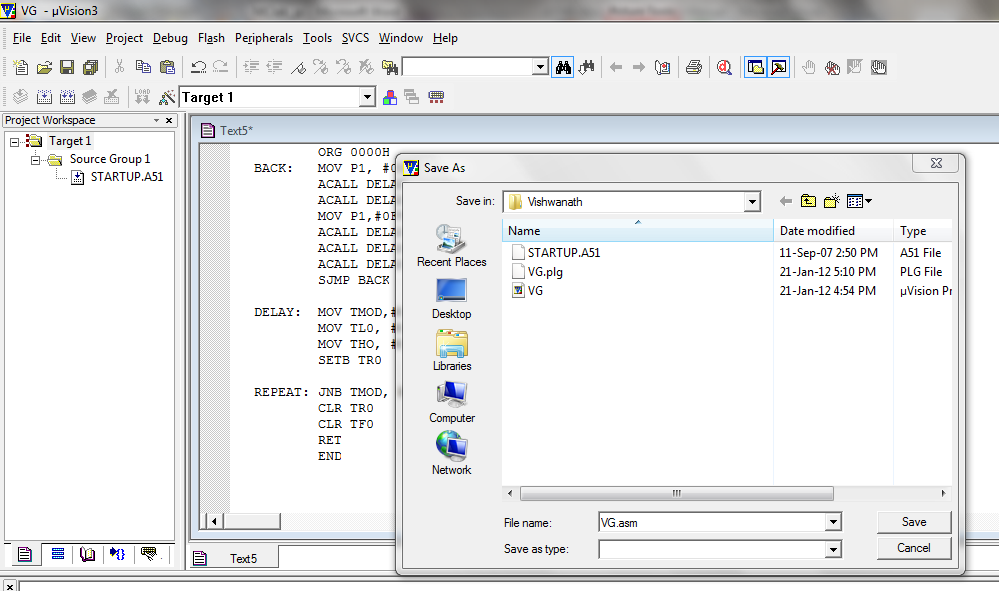


##### STEP 9: Save your program by going to “File” -> “Save” option



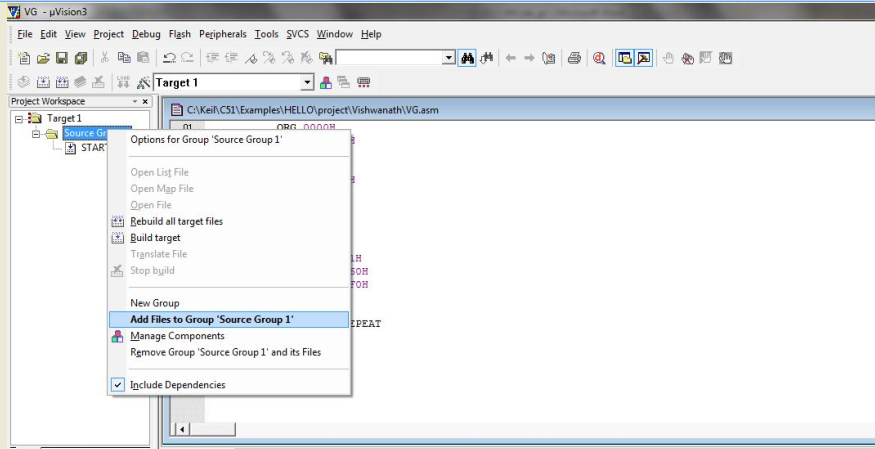
##### STEP 10:

* “Save in” your project folder.
* Give file name with “.asm“extension.
* And then click on “Save” option



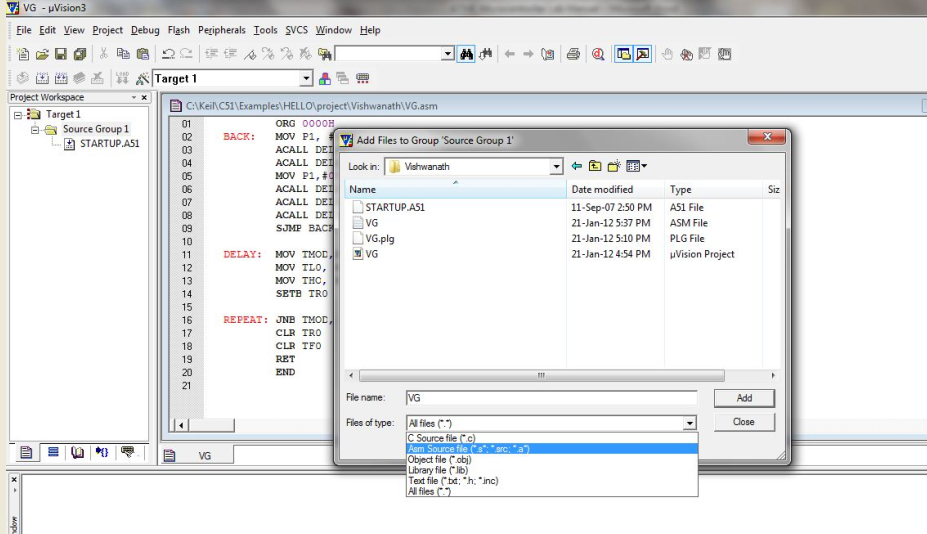
##### STEP 11:

* Right click on “Source Group1”
* Select “Add Files to ‘Group Source Group 1’”.



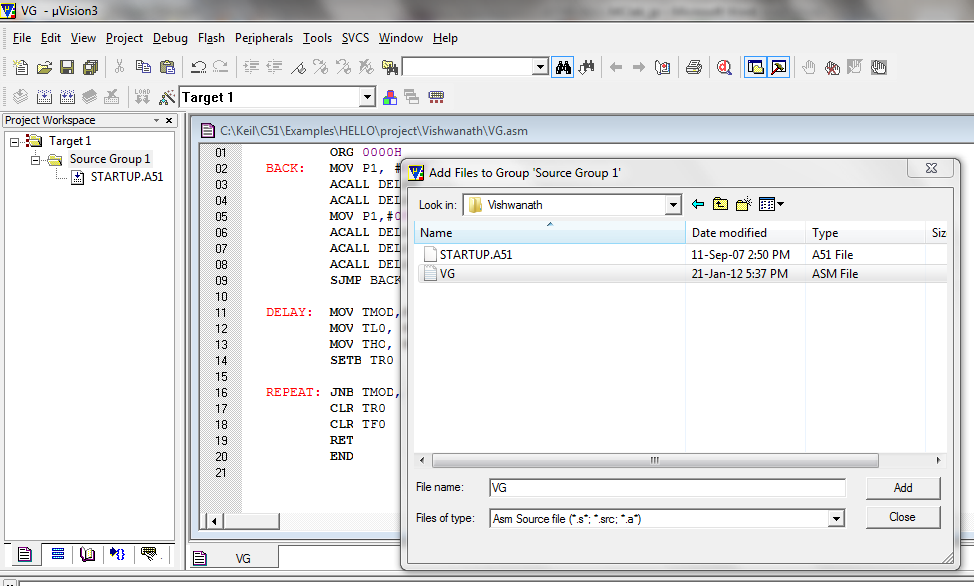
STEP 12:

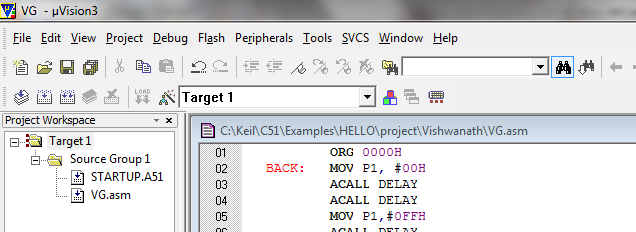
* Select to your Project folder
* Select “Files of type” as “Asm source file” if your program is written in assembly level language or else select “C file” if your program is in C language



##### STEP 13:

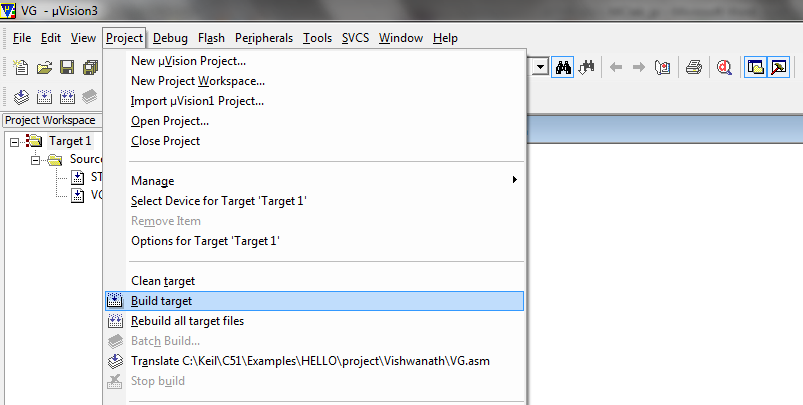
* Select your program file.
* And then click on “Add” to add the file to your source group.
* Notice that your file is added to the Source group





##### STEP 14: Build the target.

* Go to “Project”.
* Select “Build Target” or press”F7” key.

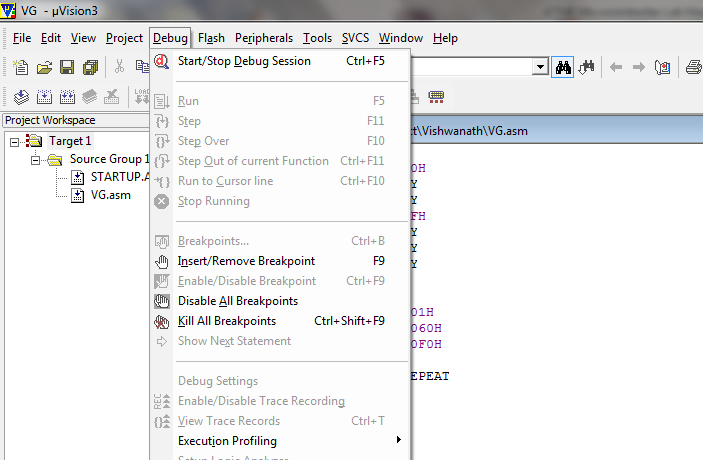


Important**:** After building the target check for the error(s). If there is any error(s) go back to your program, correct the error(s). The output window shows the line where error is found. After correcting the error go back to Step 14 and repeat the processes until there is zero error

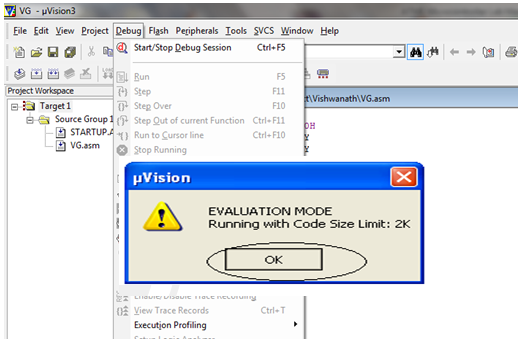


STEP 15: Debugging.

* Go to “Debug”.
* Select “Start/ Stop Debug Session” or press”Ctrl+F5” key.



Select “OK”.

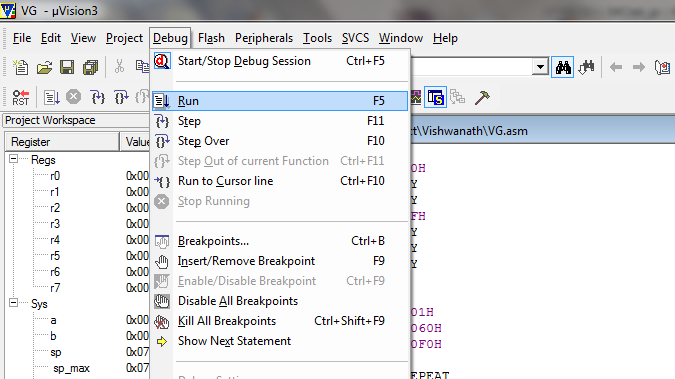


STEP 15: Selecting Output Window.

* Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output.
* Type in the input parameters (memory address/ port address/ timer) according to your program.

STEP 16: Execution.

* Go to “Debug”, Select “Run” or press” F5” key for one time execution.
* For single step execution Press11.



EXPERIMENT -1

**Write a program to blink an LED**

**Aim:** To write an Embedded C program to blink an LED by using Keil or Proteus software.

**Software used:**Computer, Keil Version4/5,Proteus 8.0.

**Procedure:**

**Keil Software**

* Select the Keil µVision software
* Select “Project” and “**New µVision Project**”.
* Create new project by entering your “**File name**” and then “**Save**” your file

##### Choose “**Atmel**” microcontroller from the database

* Select “**AT89C51**” µC and click “OK” and then “YES” and “YES”
* Make sure that “STARTUP.A51” file is added to the target.

##### Go to “File” and select “**New**” for text (program) editing window.

* Type your program in the editing window.
* Save your program by going to “File” -> “**Save**” option
* “Save in” your project folder.
* Give file name with “**.C**“ extension.
* And then click on “**Save**” option
* Right click on “Source Group1”
* Select “**Add Files** to ‘Group Source Group 1’”.
* Select to your Project folder
* Select “Files of type” as **“C source file”**
* Select your program file.
* And then click on “**Add**” to add the file to your source group.
* Notice that your file is added to the Source group
* Go to “**Debug**”.
* Select “**Start/ Stop Debug Session**” or press”Ctrl+F5” key
* **Selecting Output Window**.:- Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output. Type in the input parameters (memory address/ port address/ timer) according to your program.
* **Execution**:- Go to “Debug”, Select “**Run**” or press” F5” key for one time execution. For single step execution Press11.

# [Proteus Software](http://www.circuitstoday.com/proteus-software-introduction)

* Open **Software** and select New design in  File menu
* An untitled design sheet will be opened, save it according to your wish,it is better to create a new folder for every layout as it generates other files supporting your design.
* To Select **components**, Click on the component mode button.
* Click On **Pick** from Libraries. It shows the categories of components available and a search option to enter the part name.
* Select the components from **categories** or type the part name in Keywords text box.
* The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.
* Place all the required components and **route the wires** i.e, make connections.
* Either selection mode above the **component mode** or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.
* Double click on the component to edit the properties of the components and click on Ok.
* After connecting the circuit,click on the play button to run the simulation
* Simulation can be stepped, paused or stopped at any time.

**Program:**

#include <reg51.h>

void delay()

{

int i,j;

for(i=0; i<=10; i++)

for(j=0; j<=1024; j++);

}

void main()

{

while(1)

{

P1=0xCC;

delay();

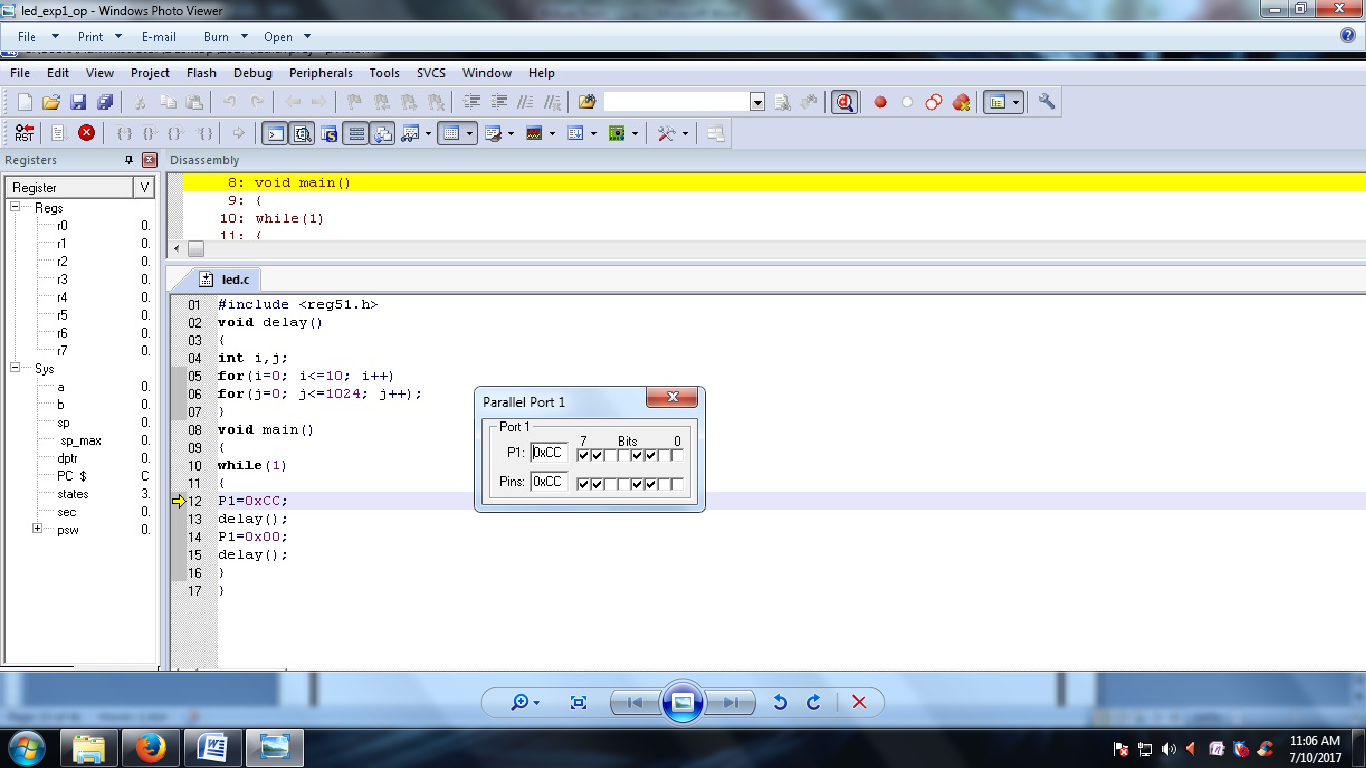
P1=0x00;

delay();

}

}

**Output:**



**VIVA QUESTIONS:**

1. What is watchdog timer?  
2. What is semaphore?  
3. What is mutex?  
4. Can structures be passed to the functions by value?  
5. Why cannot arrays be passed by values to functions?  
6. Advantages and disadvantages of using macro and inline functions?  
7. What happens when recursion functions are declared inline?  
8. Scope of static variables?  
9. What is the difference between a ‘thread’ and a ‘process’?  
10. Explain the working of Virtual Memory?  
11. What is Concurrency? Explain with example Deadlock and Starvation.  
12. What is the difference between fifo and the memory?  
13. Is it necessary to start the execution of a program from the main() in C?  
14. What is an anti aliasing filter? Why is it required?  
15. How to implement a fourth order Butterworth LP filter at 1kHz if sampling frequency is 8 kHz?  
16. IS 8085 an embedded system?  
17. What is the role of segment register?  
18. What type of registers contains an (INTEL) CPU?  
19. What is plc system?  
20. What is difference between micro processor & micro controller?  
21. Can we use semaphore or mutex or spin lock in interrupt context in linux kernel?  
22. DMA deals with which address (physical/virtual addresses)?  
23. What is dirac delta function and its Fourier transform and its importance?  
24. What is the difference between testing and verification of vlsi circuit?  
25. While writing interrupt handlers (ISR), which are points needed to be considered?  
26. Explain can microcontroller work independently?  
27. Explain What happens when recursion functions are declared inline?  
28. Explain Scope of static variables?  
29. What is interrupt latency?  
30. Explain Operations involving unsigned and signed? Unsigned will be converted to signed?  
31. Explain Order of constructor and destructor call in case of multiple inheritance?  
32. Explain Difference between object oriented and object based languages?  
33. What are the advantages and disadvantages of using macro and inline functions?  
34. Explain why cannot arrays be passed by values to functions?  
35. Explain what is interrupt latency? How can we reduce it?  
36. Explain what are the different qualifiers in C?  
37. Explain What are the 5 different types of inheritance relationship?  
38. Explain What will this return malloc(sizeof(-10))?  
39. Explain Can structures be passed to the functions by value?  
40. Explain can we have constant volatile variable?  
41. Explain what are the different storage classes in C?  
42. Explain what is forward reference w.r.t. pointers in c?  
43. How is function itoa() written in C?  
44. Explain what is the difference between embedded systems and the system in which RTOS is running?  
45. How to define a structure with bit field members?  
46. Explain what is interrupt latency?  
47. Explain Scope of static variables?  
48. What is pass by value and pass by reference? How are structure passed as arguments?  
49. What is difference between using a macro and a in line function?  
50. What is the volatile keyword used for?

REALTIME APPLICATION:

The [LCD display](https://www.elprocus.com/2-ways-display-messages-electronically/) is an electronic device, which is frequently used in many applications for displaying the information in a text or image format. The LCD is a display that can easily show characters on its screen. The LCD display have consists 8-data lines and 3-control lines which are used to interface to the microcontroller.

EXPERIMENT -2

**Write a program to blink an LED continuously with a delay**

**Aim:** To write an Embedded C program to blink an LED continuously with a delay by using Keil or Proteus software.

**Software used:**Computer,Keil Version4/5,Proteus 8.0.

**Procedure:**

**Keil Software**

* Select the Kiel µVision software
* Select “Project” and “**New µVision Project**”.
* Create new project by entering your “**File name**” and then “**Save**” your file

##### Choose “**Atmel**” microcontroller from the database

* Select “**AT89C51**” µC and click “OK” and then “YES” and “YES”
* Make sure that “STARTUP.A51” file is added to the target.

##### Go to “File” and select “**New**” for text (program) editing window.

* Type your program in the editing window.
* Save your program by going to “File” -> “**Save**” option
* “Save in” your project folder.
* Give file name with “**.C**“ extension.
* And then click on “**Save**” option
* Right click on “Source Group1”
* Select “**Add Files** to ‘Group Source Group 1’”.
* Select to your Project folder
* Select “Files of type” as **“C source file”**
* Select your program file.
* And then click on “**Add**” to add the file to your source group.
* Notice that your file is added to the Source group
* Go to “**Debug**”.
* Select “**Start/ Stop Debug Session**” or press”Ctrl+F5” key
* **Selecting Output Window**.:- Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output. Type in the input parameters (memory address/ port address/ timer) according to your program.
* **Execution**:- Go to “Debug”, Select “**Run**” or press” F5” key for one time execution. For single step execution Press11.

# [Proteus Software](http://www.circuitstoday.com/proteus-software-introduction)

* Open **Software** and select New design in  File menu
* An untitled design sheet will be opened, save it according to your wish,it is better to create a new folder for every layout as it generates other files supporting your design.
* To Select **components**, Click on the component mode button.
* Click On **Pick** from Libraries. It shows the categories of components available and a search option to enter the part name.
* Select the components from **categories** or type the part name in Keywords text box.
* The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.
* Place all the required components and **route the wires** i.e, make connections.
* Either selection mode above the **component mode** or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.
* Double click on the component to edit the properties of the components and click on Ok.
* After connecting the circuit,click on the play button to run the simulation
* Simulation can be stepped, paused or stopped at any time.

**Program:**

# include <reg51.h>

# include <stdio.h>

sbit LED=P1^1;

void delay(unsigned int t)

{

unsigned int i,j;

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

for (j=0; j<1024; j++);

}

void main()

{

while(1)

{

LED=1;

delay(100);

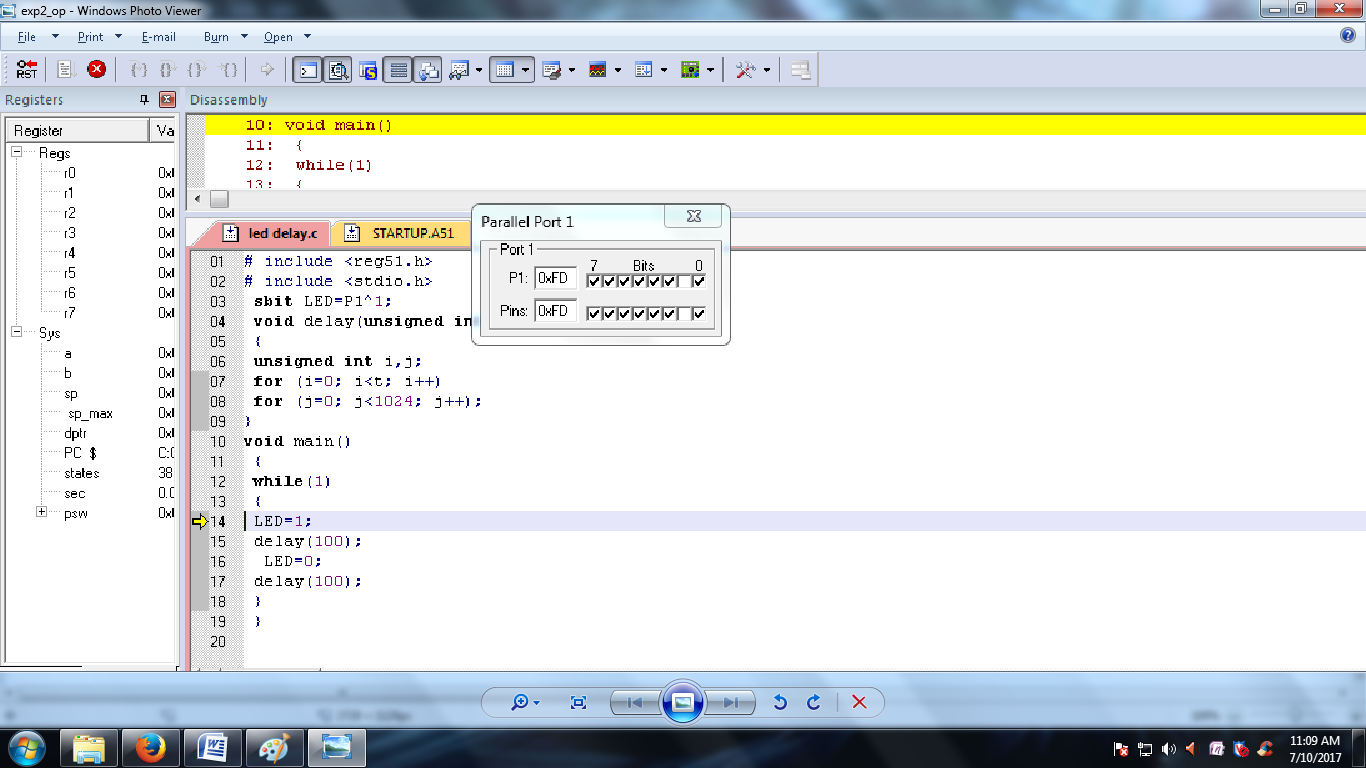
LED=0;

delay(100);

}

}

**Output:**



VIVA QUESTIONS:

1. What are hard and soft Real time systems?  
2. What is a semaphore? What are the different types of semaphore?  
3. Write a constant time consuming statement lot finding out If a given number Is a power of 2?  
4. What are recursive functions? Can we make them in line?  
5. What is the size of the int, char and float data types?  
6. What does malloc do? What will happen if we have a statement like malloc (sizeof(0));  
7. What is meant by a forward reference in C?  
8. What is the order of calling for the constructors and destructors in case of objects of inherited classes?  
9. Explain the properties of a Object oriented programming language.  
10. What do you mean by interrupt latency?  
11. What typecast is applied when we have a signed and an unsigned int in an expression?  
12. How are variables mapped across to the various memories by the C compiler?  
13. What is a memory leak? What is a segmentation fault?  
14. What is ISR? Can they be passed any parameter and can they return a value?  
15. a=7; b=8; x=a++-b; printf(“%d”, x ); What does this code give as output?  
16. What are little endian and big endian types of storage? How can you identify which type of allocation a system follows?  
17. What is the scope of a function that is declared as static?  
18. What is the use of having the const qualifier?  
19. Why do we need a infinite loop in embedded systems development? What are the different ways by which you can code in a infinite loop?  
20. What is the difference between embedded systems and the system in which rtos is running?

**21 .What Is The Difference Between Embedded Systems And The System In Which Rtos Is Running?**

**22 .What Is Pass By Value And Pass By Reference? How Are Structure Passed As Arguments?**

**23. What is the use of volatile keyword?**

**24. Can a variable be both const and volatile?**

24. What is the difference between testing and verification of vlsi circuit?  
25. While writing interrupt handlers (ISR), which are points needed to be considered?  
26. Explain can microcontroller work independently?  
27. Explain What happens when recursion functions are declared inline?  
28. Explain Scope of static variables?  
29. What is interrupt latency?  
30. Explain Operations involving unsigned and signed? Unsigned will be converted to signed?  
31. Explain Order of constructor and destructor call in case of multiple inheritance?  
32. Explain Difference between object oriented and object based languages?  
33. What are the advantages and disadvantages of using macro and inline functions?  
34. Explain why cannot arrays be passed by values to functions?  
35. Explain what is interrupt latency? How can we reduce it?  
36. Explain what are the different qualifiers in C?  
37. Explain What are the 5 different types of inheritance relationship?  
38. Explain What will this return malloc(sizeof(-10))?  
39. Explain Can structures be passed to the functions by value?  
40. Explain can we have constant volatile variable?  
41. Explain what are the different storage classes in C?  
42. Explain what is forward reference w.r.t. pointers in c?  
43. How is function itoa() written in C?  
44. Explain what is the difference between embedded systems and the system in which RTOS is running?  
45. How to define a structure with bit field members?  
46. Explain what is interrupt latency?  
47. Explain Scope of static variables?  
48. What is pass by value and pass by reference? How are structure passed as arguments?  
49. What is difference between using a macro and a in line function?  
50. What is the volatile keyword used for?

## .

REALTIME APPLICATION:

The matrix keypad is an analog switching device, which is used in many embedded applications to allow the user to perform the necessary tasks. A [matrix keypad](https://www.elprocus.com/matrix-keypad-interfacing-with-microcontroller/) consists of an arrangement of switches in matrix format in rows and columns. The rows and columns are connected to the microcontroller such that the row of switches are connected to one pin and switches in each column are connected to another pin, then perform the operations.

EXPERIMENT -3

**Generate a square wave of 10 hz at pin P1.0 of 8051 using timer**

**Aim:** To write an Embedded C program to generate a square wave of 10 hz at pin P1.0 of 8051 using timerby using Keil or Proteus software.

**Software used:**Computer,Keil Version4/5,Proteus 8.0.

**Procedure:**

**Keil Software**

* Select the Kiel µVision software
* Select “Project” and “**New µVision Project**”.
* Create new project by entering your “**File name**” and then “**Save**” your file

##### Choose “**Atmel**” microcontroller from the database

* Select “**AT89C51**” µC and click “OK” and then “YES” and “YES”
* Make sure that “STARTUP.A51” file is added to the target.

##### Go to “File” and select “**New**” for text (program) editing window.

* Type your program in the editing window.
* Save your program by going to “File” -> “**Save**” option
* “Save in” your project folder.
* Give file name with “**.C**“ extension.
* And then click on “**Save**” option
* Right click on “Source Group1”
* Select “**Add Files** to ‘Group Source Group 1’”.
* Select to your Project folder
* Select “Files of type” as **“C source file”**
* Select your program file.
* And then click on “**Add**” to add the file to your source group.
* Notice that your file is added to the Source group
* Go to “**Debug**”.
* Select “**Start/ Stop Debug Session**” or press”Ctrl+F5” key
* **Selecting Output Window**.:- Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output. Type in the input parameters (memory address/ port address/ timer) according to your program.

# [Proteus Software](http://www.circuitstoday.com/proteus-software-introduction)

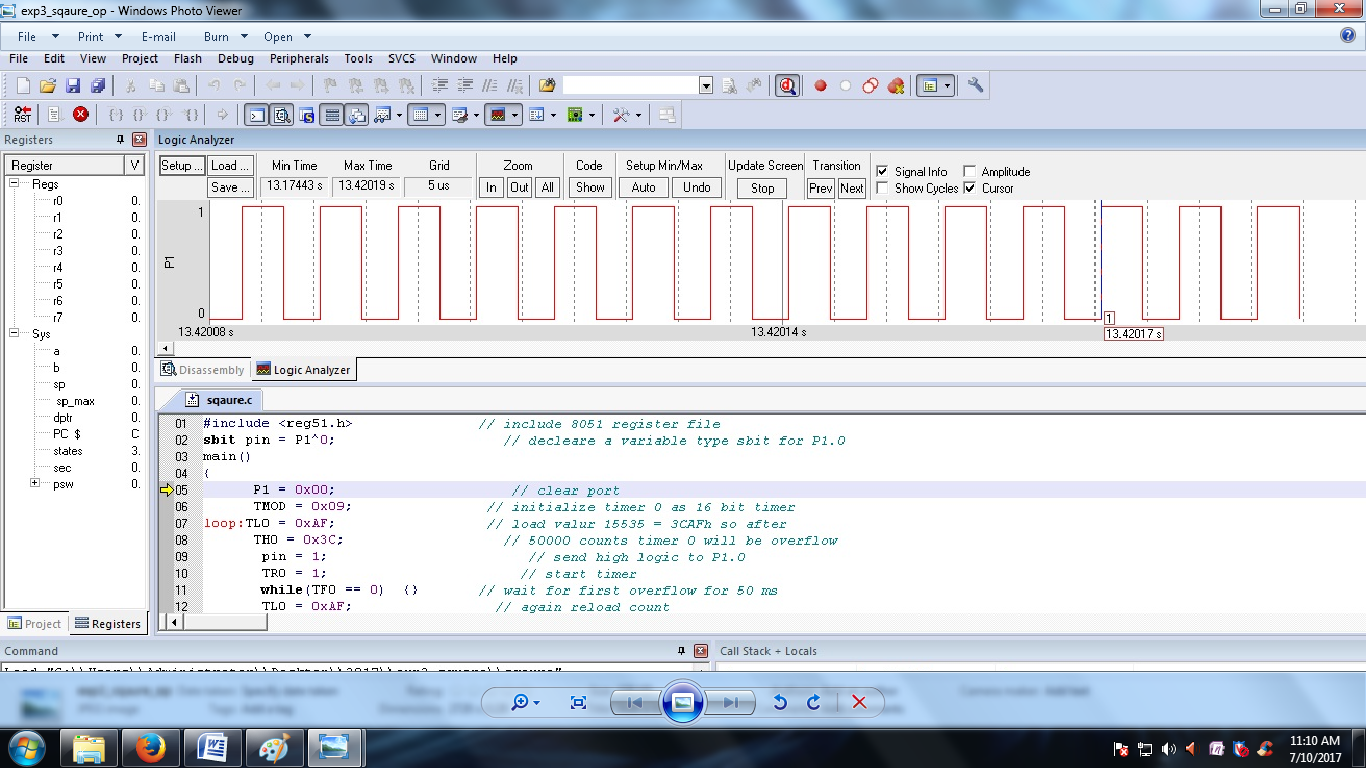
* Open **Software** and select New design in  File menu
* An untitled design sheet will be opened, save it according to your wish,it is better to create a new folder for every layout as it generates other files supporting your design.
* To Select **components**, Click on the component mode button.
* Click On **Pick** from Libraries. It shows the categories of components available and a search option to enter the part name.
* Select the components from **categories** or type the part name in Keywords text box.
* The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.
* Place all the required components and **route the wires** i.e, make connections.
* Either selection mode above the **component mode** or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.
* Double click on the component to edit the properties of the components and click on Ok.
* After connecting the circuit,click on the play button to run the simulation
* Simulation can be stepped, paused or stopped at any time.
* **Execution**:- Go to “Debug”, Select “**Run**” or press” F5” key for one time execution. For single step execution Press11.

**Program:**

#include <reg51.h>               // include 8051 register file  
sbit pin = P1^0;                    // decleare a variable type sbit for P1.0   
main()  
{  
      P1 = 0x00;                     // clear port  
      TMOD = 0x09;                // initialize timer 0 as 16 bit timer   
loop:TL0 = 0xAF;                  // load valur 15535 = 3CAFh so after   
      TH0 = 0x3C;                   // 50000 counts timer 0 will be overflow  
       pin = 1;                        // send high logic to P1.0  
       TR0 = 1;                       // start timer  
       while(TF0 == 0)  {}       // wait for first overflow for 50 ms

       TL0 = 0xAF;                 // again reload count  
       TH0 = 0x3C;  
       pin = 0;                       // now send 0 to P1.0  
      while(TF0 == 0)  {}       // wait for 50 ms again    
  goto loop;                         // continue with the loop    
}

**Output:**

****

VIVA QUESTIONS:

1. What are hard and soft Real time systems?  
2. What is a semaphore? What are the different types of semaphore?  
3. Write a constant time consuming statement lot finding out If a given number Is a power of 2?  
4. What are recursive functions? Can we make them in line?  
5. What is the size of the int, char and float data types?  
6. What does malloc do? What will happen if we have a statement like malloc (sizeof(0));  
7. What is meant by a forward reference in C?  
8. What is the order of calling for the constructors and destructors in case of objects of inherited classes?  
9. Explain the properties of a Object oriented programming language.  
10. What do you mean by interrupt latency?  
11. What typecast is applied when we have a signed and an unsigned int in an expression?  
12. How are variables mapped across to the various memories by the C compiler?  
13. What is a memory leak? What is a segmentation fault?  
14. What is ISR? Can they be passed any parameter and can they return a value?  
15. a=7; b=8; x=a++-b; printf(“%d”, x ); What does this code give as output?  
16. What are little endian and big endian types of storage? How can you identify which type of allocation a system follows?  
17. What is the scope of a function that is declared as static?  
18. What is the use of having the const qualifier?  
19. Why do we need a infinite loop in embedded systems development? What are the different ways by which you can code in a infinite loop?  
20. What is the difference between embedded systems and the system in which rtos is running?

**21 .What Is The Difference Between Embedded Systems And The System In Which Rtos Is Running?**

**22 .What Is Pass By Value And Pass By Reference? How Are Structure Passed As Arguments?**

**23. What is the use of volatile keyword?**

**24. Can a variable be both const and volatile?**

24. What is the difference between testing and verification of vlsi circuit?  
25. While writing interrupt handlers (ISR), which are points needed to be considered?  
26. Explain can microcontroller work independently?  
27. Explain What happens when recursion functions are declared inline?  
28. Explain Scope of static variables?  
29. What is interrupt latency?  
30. Explain Operations involving unsigned and signed? Unsigned will be converted to signed?  
31. Explain Order of constructor and destructor call in case of multiple inheritance?  
32. Explain Difference between object oriented and object based languages?  
33. What are the advantages and disadvantages of using macro and inline functions?  
34. Explain why cannot arrays be passed by values to functions?  
35. Explain what is interrupt latency? How can we reduce it?  
36. Explain what are the different qualifiers in C?  
37. Explain What are the 5 different types of inheritance relationship?  
38. Explain What will this return malloc(sizeof(-10))?  
39. Explain Can structures be passed to the functions by value?  
40. Explain can we have constant volatile variable?  
41. Explain what are the different storage classes in C?  
42. Explain what is forward reference w.r.t. pointers in c?  
43. How is function itoa() written in C?  
44. Explain what is the difference between embedded systems and the system in which RTOS is running?  
45. How to define a structure with bit field members?  
46. Explain what is interrupt latency?  
47. Explain Scope of static variables?  
48. What is pass by value and pass by reference? How are structure passed as arguments?  
49. What is difference between using a macro and a in line function?  
50. What is the volatile keyword used for?

REALTIME APPLICATION:

A timer can be generalized as a multi-bit counter which increments/decrements itself on receiving a clock signal and produces an interrupt signal up on roll over. When the counter is running on the processor’s clock , it is called a “Timer”, which counts a predefined number of processor clock pulses and  generates a programmable delay. When the counter is running on an external clock source (may be a periodic or aperiodic external signal) it is called a “Counter” itself and it can be used for counting external events.

EXPERIMENT -4

**Generate a Stepper motor**

**Aim:** To write an Embedded C program to design a Stepper Motor by using Keil or Proteus software.

**Software used:**Computer, Keil Version4/5,Proteus 8.0.

**Procedure:**

**Keil Software**

* Select the Kiel µVision software
* Select “Project” and “**New µVision Project**”.
* Create new project by entering your “**File name**” and then “**Save**” your file

##### Choose “**Atmel**” microcontroller from the database

* Select “**AT89C51**” µC and click “OK” and then “YES” and “YES”
* Make sure that “STARTUP.A51” file is added to the target.

##### Go to “File” and select “**New**” for text (program) editing window.

* Type your program in the editing window.
* Save your program by going to “File” -> “**Save**” option
* “Save in” your project folder.
* Give file name with “**.C**“ extension.
* And then click on “**Save**” option
* Right click on “Source Group1”
* Select “**Add Files** to ‘Group Source Group 1’”.
* Select to your Project folder
* Select “Files of type” as **“C source file”**
* Select your program file.
* And then click on “**Add**” to add the file to your source group.
* Notice that your file is added to the Source group
* Go to “**Debug**”.
* Select “**Start/ Stop Debug Session**” or press”Ctrl+F5” key
* **Selecting Output Window**.:- Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output. Type in the input parameters (memory address/ port address/ timer) according to your program.
* **Execution**:- Go to “Debug”, Select “**Run**” or press” F5” key for one time execution. For single step execution Press11.

# [Proteus Software](http://www.circuitstoday.com/proteus-software-introduction)

* Open **Software** and select New design in  File menu
* An untitled design sheet will be opened, save it according to your wish,it is better to create a new folder for every layout as it generates other files supporting your design.
* To Select **components**, Click on the component mode button.
* Click On **Pick** from Libraries. It shows the categories of components available and a search option to enter the part name.
* Select the components from **categories** or type the part name in Keywords text box.
* The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.
* Place all the required components and **route the wires** i.e, make connections.
* Either selection mode above the **component mode** or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.
* Double click on the component to edit the properties of the components and click on Ok.
* After connecting the circuit,click on the play button to run the simulation
* Simulation can be stepped, paused or stopped at any time.

**Program:**

#include<reg51.b>

sbit s1=P1^1;

sbit s2=P1^2;

sbit s3=P1^3;

sbit s4=P1^4;

void delay(unsigned int);

unsigned int i;

void main()

{

s1=s2=s3=s4=0;

while(1);

{

s1=1;s2=0;s3=0;s4=0;

delay(900);

s1=1;s2=1;s3=0;s4=0;

delay(900);

s1=0;s2=1;s3=0;s4=0;

delay(900);

s1=0;s2=1;s3=1;s4=0;

delay(900);

s1=0;s2=0;s3=1;s4=0;

delay(900);

s1=0;s2=0;s3=1;s4=1;

delay(900);

s1=0;s2=0;s3=0;s4=1;

delay(900);

s1=1;s2=0;s3=0;s4=1;

delay(900);

s1=1;s2=0;s3=0;s4=1;

delay(900);

s1=0;s2=0;s3=0;s4=0;

delay(900);

delay(900);delay(900);

}

}

void delay(unsigned int i);

{

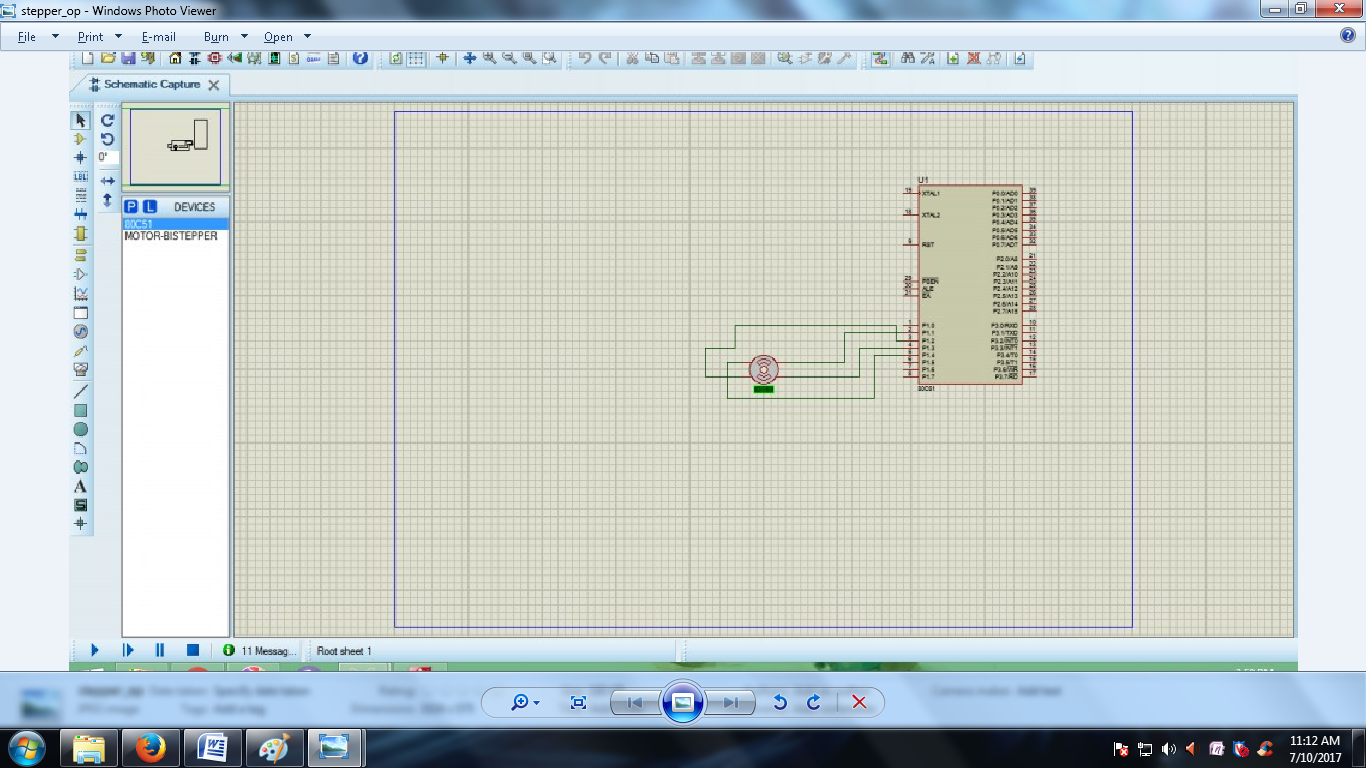
unsigned int p,j;

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

for(j=0;j<21;j++);

}

**Output:**

****

VIVA QUESTIONS:

1. What is watchdog timer?  
2. What is semaphore?  
3. What is mutex?  
4. Can structures be passed to the functions by value?  
5. Why cannot arrays be passed by values to functions?  
6. Advantages and disadvantages of using macro and inline functions?  
7. What happens when recursion functions are declared inline?  
8. Scope of static variables?  
9. What is the difference between a ‘thread’ and a ‘process’?  
10. Explain the working of Virtual Memory?  
11. What is Concurrency? Explain with example Deadlock and Starvation.  
12. What is the difference between fifo and the memory?  
13. Is it necessary to start the execution of a program from the main() in C?  
14. What is an anti aliasing filter? Why is it required?  
15. How to implement a fourth order Butterworth LP filter at 1kHz if sampling frequency is 8 kHz?  
16. IS 8085 an embedded system?  
17. What is the role of segment register?  
18. What type of registers contains an (INTEL) CPU?  
19. What is plc system?  
20. What is difference between micro processor & micro controller?  
21. Can we use semaphore or mutex or spin lock in interrupt context in linux kernel?  
22. DMA deals with which address (physical/virtual addresses)?  
23. What is dirac delta function and its Fourier transform and its importance?  
24. What is the difference between testing and verification of vlsi circuit?  
25. While writing interrupt handlers (ISR), which are points needed to be considered?  
26. Explain can microcontroller work independently?  
27. Explain What happens when recursion functions are declared inline?  
28. Explain Scope of static variables?  
29. What is interrupt latency?  
30. Explain Operations involving unsigned and signed? Unsigned will be converted to signed?  
31. Explain Order of constructor and destructor call in case of multiple inheritance?  
32. Explain Difference between object oriented and object based languages?  
33. What are the advantages and disadvantages of using macro and inline functions?  
34. Explain why cannot arrays be passed by values to functions?  
35. Explain what is interrupt latency? How can we reduce it?  
36. Explain what are the different qualifiers in C?  
37. Explain What are the 5 different types of inheritance relationship?  
38. Explain What will this return malloc(sizeof(-10))?  
39. Explain Can structures be passed to the functions by value?  
40. Explain can we have constant volatile variable?  
41. Explain what are the different storage classes in C?  
42. Explain what is forward reference w.r.t. pointers in c?  
43. How is function itoa() written in C?  
44. Explain what is the difference between embedded systems and the system in which RTOS is running?  
45. How to define a structure with bit field members?  
46. Explain what is interrupt latency?  
47. Explain Scope of static variables?  
48. What is pass by value and pass by reference? How are structure passed as arguments?  
49. What is difference between using a macro and a in line function?  
50. What is the volatile keyword used for?

REALTIME APPLICATION:

**A new algorithm for stepper-motor acceleration allows speed profiles to be parameterized and calculated in real time. This algorithm can run on a low-end microcontroller using only simple fixed-point arithmetic operations and no data tables. It develops an accurate approximation for the timing of a linear ramp with constant acceleration and deceleration.**

EXPERIMENT -5

**Write a program for LCD displays**

**Aim:** To write an Embedded C program for LCD displays by using Keil or Proteus software.

**Software used:**Computer,Keil Version4/5,Proteus 8.0.

**Procedure:**

**Keil Software**

* Select the Kiel µVision software
* Select “Project” and “**New µVision Project**”.
* Create new project by entering your “**File name**” and then “**Save**” your file

##### Choose “**Atmel**” microcontroller from the database

* Select “**AT89C51**” µC and click “OK” and then “YES” and “YES”
* Make sure that “STARTUP.A51” file is added to the target.

##### Go to “File” and select “**New**” for text (program) editing window.

* Type your program in the editing window.
* Save your program by going to “File” -> “**Save**” option
* “Save in” your project folder.
* Give file name with “**.C**“ extension.
* And then click on “**Save**” option
* Right click on “Source Group1”
* Select “**Add Files** to ‘Group Source Group 1’”.
* Select to your Project folder
* Select “Files of type” as **“C source file”**
* Select your program file.
* And then click on “**Add**” to add the file to your source group.
* Notice that your file is added to the Source group
* Go to “**Debug**”.
* Select “**Start/ Stop Debug Session**” or press”Ctrl+F5” key
* **Selecting Output Window**.:- Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output. Type in the input parameters (memory address/ port address/ timer) according to your program.
* **Execution**:- Go to “Debug”, Select “**Run**” or press” F5” key for one time execution. For single step execution Press11.

# [Proteus Software](http://www.circuitstoday.com/proteus-software-introduction)

* Open **Software** and select New design in  File menu
* An untitled design sheet will be opened, save it according to your wish,it is better to create a new folder for every layout as it generates other files supporting your design.
* To Select **components**, Click on the component mode button.
* Click On **Pick** from Libraries. It shows the categories of components available and a search option to enter the part name.
* Select the components from **categories** or type the part name in Keywords text box.
* The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.
* Place all the required components and **route the wires** i.e, make connections.
* Either selection mode above the **component mode** or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.
* Double click on the component to edit the properties of the components and click on Ok.
* After connecting the circuit,click on the play button to run the simulation
* Simulation can be stepped, paused or stopped at any time.

**Program:**

#include<reg51.h>

void delay\_ms(unsigned int k)

{

unsigned int i,j;

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

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

}

void LCD\_display(char \*str)

{

unsigned int j,k;

unsigned int sd[4]={0x3F,0x0E,0x01,0x80};

for(k=0; sd[k]!=0; k++)

{

P3=0x02;

P2=sd[k];

delay\_ms(1);

P3=0x00;

delay\_ms(100);

}

for(j=0; str[j]!=0; j++)

{

P3=0X03;

P2=str[j];

delay\_ms(1);

P3=0x01;

}

}

void main()

{

P2=0x00;

P3=0x00;

while(1)

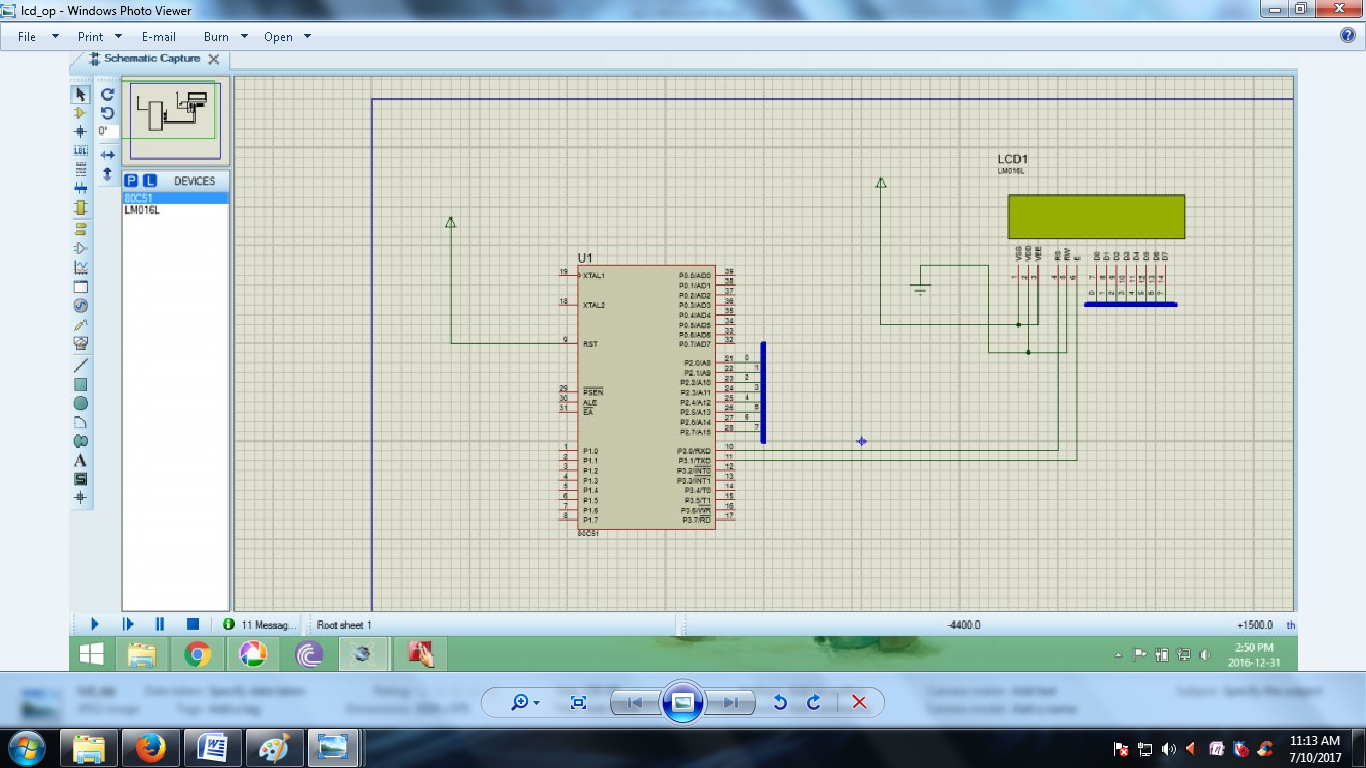
{

LCD\_display("Embedded Systems");

}

}

**Output:**

****

**VIVA QUESTIONS:**

1. What is watchdog timer?  
2. What is semaphore?  
3. What is mutex?  
4. Can structures be passed to the functions by value?  
5. Why cannot arrays be passed by values to functions?  
6. Advantages and disadvantages of using macro and inline functions?  
7. What happens when recursion functions are declared inline?  
8. Scope of static variables?  
9. What is the difference between a ‘thread’ and a ‘process’?  
10. Explain the working of Virtual Memory?  
11. What is Concurrency? Explain with example Deadlock and Starvation.  
12. What is the difference between fifo and the memory?  
13. Is it necessary to start the execution of a program from the main() in C?  
14. What is an anti aliasing filter? Why is it required?  
15. How to implement a fourth order Butterworth LP filter at 1kHz if sampling frequency is 8 kHz?  
16. IS 8085 an embedded system?  
17. What is the role of segment register?  
18. What type of registers contains an (INTEL) CPU?  
19. What is plc system?  
20. What is difference between micro processor & micro controller?  
21. Can we use semaphore or mutex or spin lock in interrupt context in linux kernel?  
22. DMA deals with which address (physical/virtual addresses)?  
23. What is dirac delta function and its Fourier transform and its importance?  
24. What is the difference between testing and verification of vlsi circuit?  
25. While writing interrupt handlers (ISR), which are points needed to be considered?  
26. Explain can microcontroller work independently?  
27. Explain What happens when recursion functions are declared inline?  
28. Explain Scope of static variables?  
29. What is interrupt latency?  
30. Explain Operations involving unsigned and signed? Unsigned will be converted to signed?  
31. Explain Order of constructor and destructor call in case of multiple inheritance?  
32. Explain Difference between object oriented and object based languages?  
33. What are the advantages and disadvantages of using macro and inline functions?  
34. Explain why cannot arrays be passed by values to functions?  
35. Explain what is interrupt latency? How can we reduce it?  
36. Explain what are the different qualifiers in C?  
37. Explain What are the 5 different types of inheritance relationship?  
38. Explain What will this return malloc(sizeof(-10))?  
39. Explain Can structures be passed to the functions by value?  
40. Explain can we have constant volatile variable?  
41. Explain what are the different storage classes in C?  
42. Explain what is forward reference w.r.t. pointers in c?  
43. How is function itoa() written in C?  
44. Explain what is the difference between embedded systems and the system in which RTOS is running?  
45. How to define a structure with bit field members?  
46. Explain what is interrupt latency?  
47. Explain Scope of static variables?  
48. What is pass by value and pass by reference? How are structure passed as arguments?  
49. What is difference between using a macro and a in line function?  
50. What is the volatile keyword used for?

**REALTIME APPLICATIONS:**

The [7-segment displays](https://www.elprocus.com/3-different-types-displays-available/) is the basic electronic displays, which are used in many systems to display the numeric information. It consists of eight LEDs which are connected in sequential manner so as to display digits from 0 to 9, when proper combinations of LEDs are switched on. They can display only one digit at a time.

EXPERIMENT -6

**Write a program for DC motor**

**Aim:** To write an Embedded C program to design a DC Motor using Keil or Proteus software.

**Software used:**Computer,Keil Version4/5,Proteus 8.0

**Procedure:**

**Keil Software**

* Select the Kiel µVision software
* Select “Project” and “**New µVision Project**”.
* Create new project by entering your “**File name**” and then “**Save**” your file

##### Choose “**Atmel**” microcontroller from the database

* Select “**AT89C51**” µC and click “OK” and then “YES” and “YES”
* Make sure that “STARTUP.A51” file is added to the target.

##### Go to “File” and select “**New**” for text (program) editing window.

* Type your program in the editing window.
* Save your program by going to “File” -> “**Save**” option
* “Save in” your project folder.
* Give file name with “**.C**“ extension.
* And then click on “**Save**” option
* Right click on “Source Group1”
* Select “**Add Files** to ‘Group Source Group 1’”.
* Select to your Project folder
* Select “Files of type” as **“C source file”**
* Select your program file.
* And then click on “**Add**” to add the file to your source group.
* Notice that your file is added to the Source group
* Go to “**Debug**”.
* Select “**Start/ Stop Debug Session**” or press”Ctrl+F5” key
* **Selecting Output Window**.:- Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output. Type in the input parameters (memory address/ port address/ timer) according to your program.
* **Execution**:- Go to “Debug”, Select “**Run**” or press” F5” key for one time execution. For single step execution Press11.

# [Proteus Software](http://www.circuitstoday.com/proteus-software-introduction)

* Open **Software** and select New design in  File menu
* An untitled design sheet will be opened, save it according to your wish,it is better to create a new folder for every layout as it generates other files supporting your design.
* To Select **components**, Click on the component mode button.
* Click On **Pick** from Libraries. It shows the categories of components available and a search option to enter the part name.
* Select the components from **categories** or type the part name in Keywords text box.
* The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.
* Place all the required components and **route the wires** i.e, make connections.
* Either selection mode above the **component mode** or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.
* Double click on the component to edit the properties of the components and click on Ok.
* After connecting the circuit,click on the play button to run the simulation
* Simulation can be stepped, paused or stopped at any time.

**Program:**

#include<reg52.h>

#include<stdio.h>

void delay(void);

sbit motor\_pin\_1 = P2^0;

sbit motor\_pin\_2 = P2^1;

void main()

{

do

{

motor\_pin\_1 = 1;

motor\_pin\_2 = 0; //Rotates Motor Anit Clockwise

delay();

motor\_pin\_1 = 1;

motor\_pin\_2 = 1; //Stops Motor

delay();

motor\_pin\_1 = 0;

motor\_pin\_2 = 1; //Rotates Motor Clockwise

delay();

motor\_pin\_1 = 0;

motor\_pin\_2 = 0; //Stops Motor

delay();

}while(1);

}

void delay()

{

int i,j;

for(i=0;i<1000;i++)

{

for(j=0;j<1000;j++)

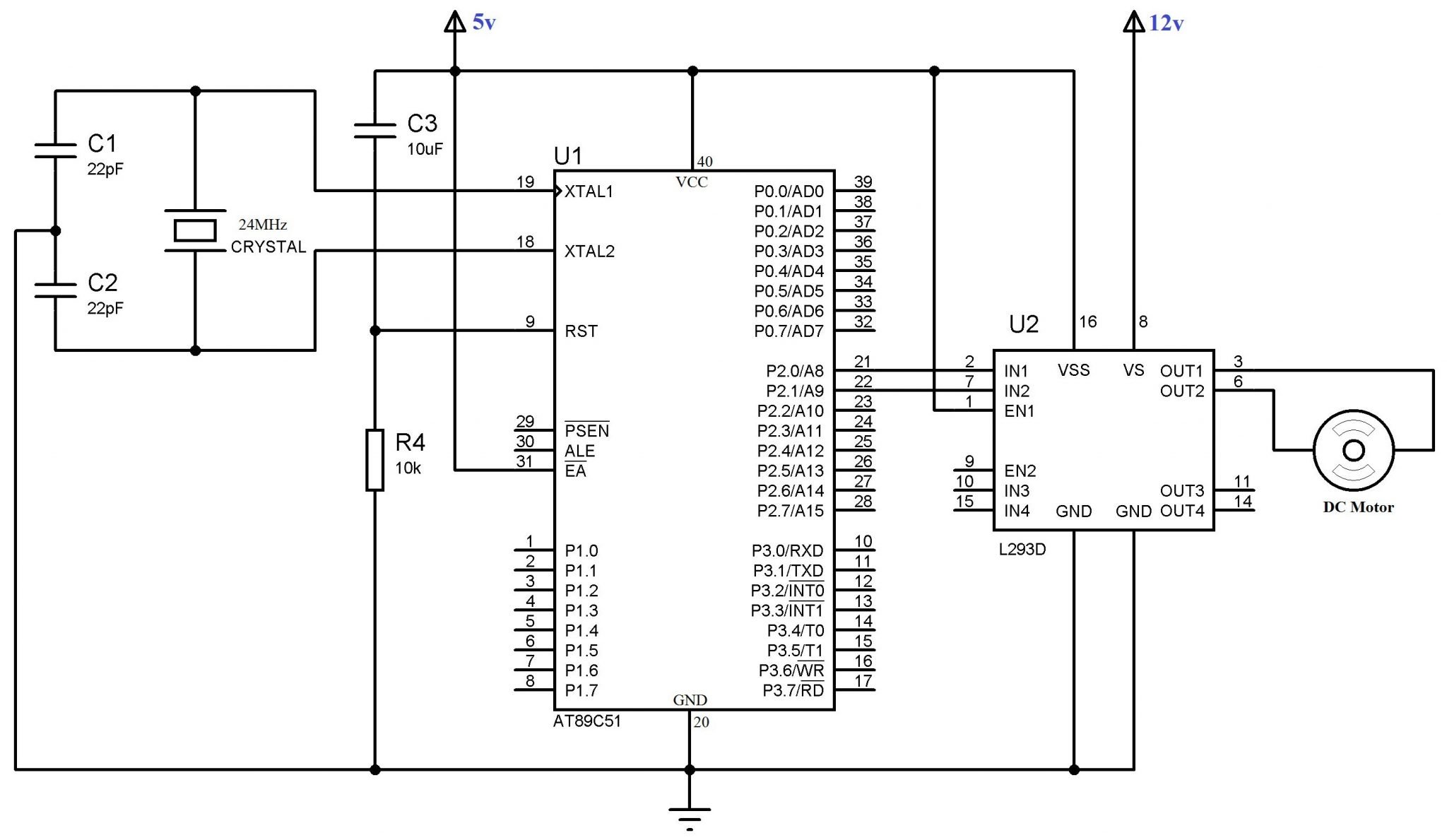
{

}

}

}

**Output:**



VIVA QUESTIONS:

1. What is watchdog timer?  
2. What is semaphore?  
3. What is mutex?  
4. Can structures be passed to the functions by value?  
5. Why cannot arrays be passed by values to functions?  
6. Advantages and disadvantages of using macro and inline functions?  
7. What happens when recursion functions are declared inline?  
8. Scope of static variables?  
9. What is the difference between a ‘thread’ and a ‘process’?  
10. Explain the working of Virtual Memory?  
11. What is Concurrency? Explain with example Deadlock and Starvation.  
12. What is the difference between fifo and the memory?  
13. Is it necessary to start the execution of a program from the main() in C?  
14. What is an anti aliasing filter? Why is it required?  
15. How to implement a fourth order Butterworth LP filter at 1kHz if sampling frequency is 8 kHz?  
16. IS 8085 an embedded system?  
17. What is the role of segment register?  
18. What type of registers contains an (INTEL) CPU?  
19. What is plc system?  
20. What is difference between micro processor & micro controller?  
21. Can we use semaphore or mutex or spin lock in interrupt context in linux kernel?  
22. DMA deals with which address (physical/virtual addresses)?  
23. What is dirac delta function and its Fourier transform and its importance?  
24. What is the difference between testing and verification of vlsi circuit?  
25. While writing interrupt handlers (ISR), which are points needed to be considered?  
26. Explain can microcontroller work independently?  
27. Explain What happens when recursion functions are declared inline?  
28. Explain Scope of static variables?  
29. What is interrupt latency?  
30. Explain Operations involving unsigned and signed? Unsigned will be converted to signed?  
31. Explain Order of constructor and destructor call in case of multiple inheritance?  
32. Explain Difference between object oriented and object based languages?  
33. What are the advantages and disadvantages of using macro and inline functions?  
34. Explain why cannot arrays be passed by values to functions?  
35. Explain what is interrupt latency? How can we reduce it?  
36. Explain what are the different qualifiers in C?  
37. Explain What are the 5 different types of inheritance relationship?  
38. Explain What will this return malloc(sizeof(-10))?  
39. Explain Can structures be passed to the functions by value?  
40. Explain can we have constant volatile variable?  
41. Explain what are the different storage classes in C?  
42. Explain what is forward reference w.r.t. pointers in c?  
43. How is function itoa() written in C?  
44. Explain what is the difference between embedded systems and the system in which RTOS is running?  
45. How to define a structure with bit field members?  
46. Explain what is interrupt latency?  
47. Explain Scope of static variables?  
48. What is pass by value and pass by reference? How are structure passed as arguments?  
49. What is difference between using a macro and a in line function?  
50. What is the volatile keyword used for?

REALTIME APPLICATIONS:

DC Motor plays a crucial role in research, industry and laboratory experiments because of their simplicity and low cost. The speed of the motor can be controlled by three methods namely terminal voltage control, armature rheostat control method and flux control method. Here in this paper terminal voltage control method is employed. A control system is an interconnection of components forming a system configuration that will provide a desired system response. A controlled DC-motor is developed allowing Arduino hardware which acts as the interface between the computer (LabVIEW) and the outside world. It primarily functions as a device that digitizes incoming analog signals so that the LabVIEW can interpret them. The user interface was developed in an Arduino environment. The aim is to control the speed of the dc motor using the Low Cost data acquisition board i.e. the Arduino board interfaced with PID Controller in LabVIEW.

EXPERIMENT -7

#### Write a program Switch ON the LED by pressing the key ‘1’ on the keypad?

**Aim:** To write an Embedded C program to Switch ON the LED by pressing the key ‘1’ on the keypadby using Keil or Proteus software.

**Software used:**Computer, Keil Version4/5,Proteus 8.0

**Procedure:**

* Select the Kiel µVision software
* Select “Project” and “**New µVision Project**”.
* Create new project by entering your “**File name**” and then “**Save**” your file

##### Choose “**Atmel**” microcontroller from the database

* Select “**AT89C51**” µC and click “OK” and then “YES” and “YES”
* Make sure that “STARTUP.A51” file is added to the target.

##### Go to “File” and select “**New**” for text (program) editing window.

* Type your program in the editing window.
* Save your program by going to “File” -> “**Save**” option
* “Save in” your project folder.
* Give file name with “**.C**“ extension.
* And then click on “**Save**” option
* Right click on “Source Group1”
* Select “**Add Files** to ‘Group Source Group 1’”.
* Select to your Project folder
* Select “Files of type” as **“C source file”**
* Select your program file.
* And then click on “**Add**” to add the file to your source group.
* Notice that your file is added to the Source group
* Go to “**Debug**”.
* Select “**Start/ Stop Debug Session**” or press”Ctrl+F5” key
* **Selecting Output Window**.:- Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output. Type in the input parameters (memory address/ port address/ timer) according to your program.
* **Execution**:- Go to “Debug”, Select “**Run**” or press” F5” key for one time execution. For single step execution Press11.

# [Proteus Software](http://www.circuitstoday.com/proteus-software-introduction)

* Open **Software** and select New design in  File menu
* An untitled design sheet will be opened, save it according to your wish,it is better to create a new folder for every layout as it generates other files supporting your design.
* To Select **components**, Click on the component mode button.
* Click On **Pick** from Libraries. It shows the categories of components available and a search option to enter the part name.
* Select the components from **categories** or type the part name in Keywords text box.
* The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.
* Place all the required components and **route the wires** i.e, make connections.
* Either selection mode above the **component mode** or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.
* Double click on the component to edit the properties of the components and click on Ok.
* After connecting the circuit,click on the play button to run the simulation
* Simulation can be stepped, paused or stopped at any time.

**Program:**

#include<reg51.h>

sbit r1=P2^0;

sbit c1=P3^0;

sbit LED=P0^1;

void main()

{

r1=0;

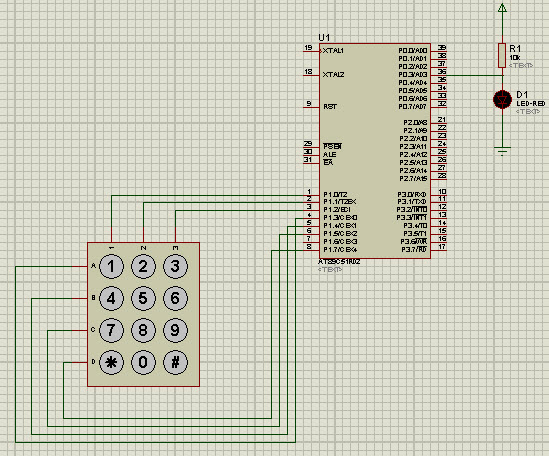
if(c1==0)

{

LED=0xff;

}

}

**Output:**

**VIVA QUESTIONS:**

1. What are hard and soft Real time systems?  
2. What is a semaphore? What are the different types of semaphore?  
3. Write a constant time consuming statement lot finding out If a given number Is a power of 2?  
4. What are recursive functions? Can we make them in line?  
5. What is the size of the int, char and float data types?  
6. What does malloc do? What will happen if we have a statement like malloc (sizeof(0));  
7. What is meant by a forward reference in C?  
8. What is the order of calling for the constructors and destructors in case of objects of inherited classes?  
9. Explain the properties of a Object oriented programming language.  
10. What do you mean by interrupt latency?  
11. What typecast is applied when we have a signed and an unsigned int in an expression?  
12. How are variables mapped across to the various memories by the C compiler?  
13. What is a memory leak? What is a segmentation fault?  
14. What is ISR? Can they be passed any parameter and can they return a value?  
15. a=7; b=8; x=a++-b; printf(“%d”, x ); What does this code give as output?  
16. What are little endian and big endian types of storage? How can you identify which type of allocation a system follows?  
17. What is the scope of a function that is declared as static?  
18. What is the use of having the const qualifier?  
19. Why do we need a infinite loop in embedded systems development? What are the different ways by which you can code in a infinite loop?  
20. What is the difference between embedded systems and the system in which rtos is running?

**21 .What Is The Difference Between Embedded Systems And The System In Which Rtos Is Running?**

**22 .What Is Pass By Value And Pass By Reference? How Are Structure Passed As Arguments?**

**23. What is the use of volatile keyword?**

**24. Can a variable be both const and volatile?**

24. What is the difference between testing and verification of vlsi circuit?  
25. While writing interrupt handlers (ISR), which are points needed to be considered?  
26. Explain can microcontroller work independently?  
27. Explain What happens when recursion functions are declared inline?  
28. Explain Scope of static variables?  
29. What is interrupt latency?  
30. Explain Operations involving unsigned and signed? Unsigned will be converted to signed?  
31. Explain Order of constructor and destructor call in case of multiple inheritance?  
32. Explain Difference between object oriented and object based languages?  
33. What are the advantages and disadvantages of using macro and inline functions?  
34. Explain why cannot arrays be passed by values to functions?  
35. Explain what is interrupt latency? How can we reduce it?  
36. Explain what are the different qualifiers in C?  
37. Explain What are the 5 different types of inheritance relationship?  
38. Explain What will this return malloc(sizeof(-10))?  
39. Explain Can structures be passed to the functions by value?  
40. Explain can we have constant volatile variable?  
41. Explain what are the different storage classes in C?  
42. Explain what is forward reference w.r.t. pointers in c?  
43. How is function itoa() written in C?  
44. Explain what is the difference between embedded systems and the system in which RTOS is running?  
45. How to define a structure with bit field members?  
46. Explain what is interrupt latency?  
47. Explain Scope of static variables?  
48. What is pass by value and pass by reference? How are structure passed as arguments?  
49. What is difference between using a macro and a in line function?  
50. What is the volatile keyword used for?

**REALTIME APPLICATIONS:**

The matrix keypad is an analog switching device, which is used in many embedded applications to allow the user to perform the necessary tasks. A [matrix keypad](https://www.elprocus.com/matrix-keypad-interfacing-with-microcontroller/) consists of an arrangement of switches in matrix format in rows and columns. The rows and columns are connected to the microcontroller such that the row of switches are connected to one pin and switches in each column are connected to another pin, then perform the operations.

EXPERIMENT -8

## **Using Push Button Switch with 8051**

**Aim:** To write an Embedded C program to design a push button switch with 8051by using Keil or Proteus software.

**Software used:**Computer, Keil Version4/5,Proteus 8.0

**Procedure:**

**Keil Software**

* Select the Kiel µVision software
* Select “Project” and “**New µVision Project**”.
* Create new project by entering your “**File name**” and then “**Save**” your file

##### Choose “**Atmel**” microcontroller from the database

* Select “**AT89C51**” µC and click “OK” and then “YES” and “YES”
* Make sure that “STARTUP.A51” file is added to the target.

##### Go to “File” and select “**New**” for text (program) editing window.

* Type your program in the editing window.
* Save your program by going to “File” -> “**Save**” option
* “Save in” your project folder.
* Give file name with “**.C**“ extension.
* And then click on “**Save**” option
* Right click on “Source Group1”
* Select “**Add Files** to ‘Group Source Group 1’”.
* Select to your Project folder
* Select “Files of type” as **“C source file”**
* Select your program file.
* And then click on “**Add**” to add the file to your source group.
* Notice that your file is added to the Source group
* Go to “**Debug**”.
* Select “**Start/ Stop Debug Session**” or press”Ctrl+F5” key
* **Selecting Output Window**.:- Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output. Type in the input parameters (memory address/ port address/ timer) according to your program.
* **Execution**:- Go to “Debug”, Select “**Run**” or press” F5” key for one time execution. For single step execution Press11.

# [Proteus Software](http://www.circuitstoday.com/proteus-software-introduction)

* Open **Software** and select New design in  File menu
* An untitled design sheet will be opened, save it according to your wish,it is better to create a new folder for every layout as it generates other files supporting your design.
* To Select **components**, Click on the component mode button.
* Click On **Pick** from Libraries. It shows the categories of components available and a search option to enter the part name.
* Select the components from **categories** or type the part name in Keywords text box.
* The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.
* Place all the required components and **route the wires** i.e, make connections.
* Either selection mode above the **component mode** or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.
* Double click on the component to edit the properties of the components and click on Ok.
* After connecting the circuit,click on the play button to run the simulation
* Simulation can be stepped, paused or stopped at any time.

**Program:**

#include<reg52.h> /\* special function register declarations \*/

sbit LED\_pin = P2^0; //Defining LED PIN

sbit switch\_pin = P0^0; //Defining Switch PIN

void Delay(int); //Function prototype declaration

void main (void)

{

switch\_pin = 1; // Making Switch PIN input

LED\_pin=1; //LED off initially

while(1) //infinite loop

{

if(switch\_pin == 0 ) //If switch pressed

{

LED\_pin = 0; //LED ON

Delay(2000); //Delay

LED\_pin = 1; //LED OFF

}

}

}

void Delay(int k)

{

int j;

int i;

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

{

for(j=0;j<100;j++)

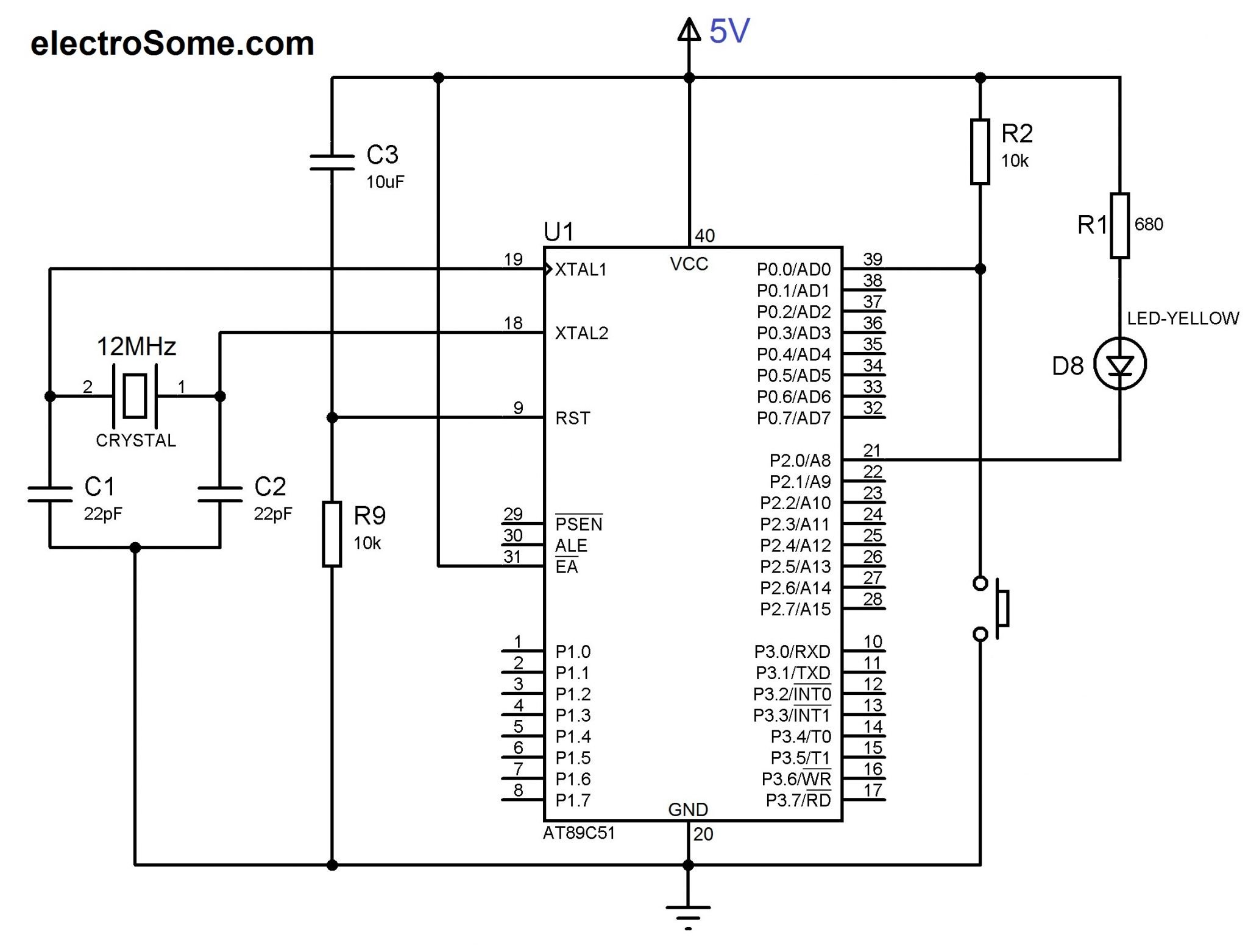
{

}

}

}

**Output:**

****

**VIVA QUESTIONS:**

1. What are hard and soft Real time systems?  
2. What is a semaphore? What are the different types of semaphore?  
3. Write a constant time consuming statement lot finding out If a given number Is a power of 2?  
4. What are recursive functions? Can we make them in line?  
5. What is the size of the int, char and float data types?  
6. What does malloc do? What will happen if we have a statement like malloc (sizeof(0));  
7. What is meant by a forward reference in C?  
8. What is the order of calling for the constructors and destructors in case of objects of inherited classes?  
9. Explain the properties of a Object oriented programming language.  
10. What do you mean by interrupt latency?  
11. What typecast is applied when we have a signed and an unsigned int in an expression?  
12. How are variables mapped across to the various memories by the C compiler?  
13. What is a memory leak? What is a segmentation fault?  
14. What is ISR? Can they be passed any parameter and can they return a value?  
15. a=7; b=8; x=a++-b; printf(“%d”, x ); What does this code give as output?  
16. What are little endian and big endian types of storage? How can you identify which type of allocation a system follows?  
17. What is the scope of a function that is declared as static?  
18. What is the use of having the const qualifier?  
19. Why do we need a infinite loop in embedded systems development? What are the different ways by which you can code in a infinite loop?  
20. What is the difference between embedded systems and the system in which rtos is running?

**21 .What Is The Difference Between Embedded Systems And The System In Which Rtos Is Running?**

**22 .What Is Pass By Value And Pass By Reference? How Are Structure Passed As Arguments?**

**23. What is the use of volatile keyword?**

**24. Can a variable be both const and volatile?**

24. What is the difference between testing and verification of vlsi circuit?  
25. While writing interrupt handlers (ISR), which are points needed to be considered?  
26. Explain can microcontroller work independently?  
27. Explain What happens when recursion functions are declared inline?  
28. Explain Scope of static variables?  
29. What is interrupt latency?  
30. Explain Operations involving unsigned and signed? Unsigned will be converted to signed?  
31. Explain Order of constructor and destructor call in case of multiple inheritance?  
32. Explain Difference between object oriented and object based languages?  
33. What are the advantages and disadvantages of using macro and inline functions?  
34. Explain why cannot arrays be passed by values to functions?  
35. Explain what is interrupt latency? How can we reduce it?  
36. Explain what are the different qualifiers in C?  
37. Explain What are the 5 different types of inheritance relationship?  
38. Explain What will this return malloc(sizeof(-10))?  
39. Explain Can structures be passed to the functions by value?  
40. Explain can we have constant volatile variable?  
41. Explain what are the different storage classes in C?  
42. Explain what is forward reference w.r.t. pointers in c?  
43. How is function itoa() written in C?  
44. Explain what is the difference between embedded systems and the system in which RTOS is running?  
45. How to define a structure with bit field members?  
46. Explain what is interrupt latency?  
47. Explain Scope of static variables?  
48. What is pass by value and pass by reference? How are structure passed as arguments?  
49. What is difference between using a macro and a in line function?  
50. What is the volatile keyword used for?

REALTIME APPLICATION:

the I/O pin configurations for the TM4C123 microcontrollers. The regular function of a pin is to perform parallel I/O. Most pins, however, have an alternative function. For example, port pins PA1 and PA0 can be either regular parallel port pins or an asynchronous serial port called universal asynchronous receiver/transmitter (UART). The ability to manage time, as an input measurement and an output parameter, has made a significant impact on the market share growth of microcontrollers. Joint Test Action Group (**JTAG**), standardized as the IEEE 1149.1, is a standard test access port used to program and debug the microcontroller board. Each microcontroller uses five port pins for the JTAG interface.

EXPERIMENT -9

**Write a programming for Traffic light signal**

**Aim:** To write an Embedded C program to design a Traffic light signal by using Keil or Proteus software.

**Software used:**Computer,Keil Version4/5,Proteus 8.0

**Procedure:**

* Select the Kiel µVision software
* Select “Project” and “**New µVision Project**”.
* Create new project by entering your “**File name**” and then “**Save**” your file

##### Choose “**Atmel**” microcontroller from the database

* Select “**AT89C51**” µC and click “OK” and then “YES” and “YES”
* Make sure that “STARTUP.A51” file is added to the target.

##### Go to “File” and select “**New**” for text (program) editing window.

* Type your program in the editing window.
* Save your program by going to “File” -> “**Save**” option
* “Save in” your project folder.
* Give file name with “**.C**“ extension.
* And then click on “**Save**” option
* Right click on “Source Group1”
* Select “**Add Files** to ‘Group Source Group 1’”.
* Select to your Project folder
* Select “Files of type” as **“C source file”**
* Select your program file.
* And then click on “**Add**” to add the file to your source group.
* Notice that your file is added to the Source group
* Go to “**Debug**”.
* Select “**Start/ Stop Debug Session**” or press”Ctrl+F5” key
* **Selecting Output Window**.:- Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output. Type in the input parameters (memory address/ port address/ timer) according to your program.
* **Execution**:- Go to “Debug”, Select “**Run**” or press” F5” key for one time execution. For single step execution Press11.

# [Proteus Software](http://www.circuitstoday.com/proteus-software-introduction)

* Open **Software** and select New design in  File menu
* An untitled design sheet will be opened, save it according to your wish,it is better to create a new folder for every layout as it generates other files supporting your design.
* To Select **components**, Click on the component mode button.
* Click On **Pick** from Libraries. It shows the categories of components available and a search option to enter the part name.
* Select the components from **categories** or type the part name in Keywords text box.
* The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.
* Place all the required components and **route the wires** i.e, make connections.
* Either selection mode above the **component mode** or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.
* Double click on the component to edit the properties of the components and click on Ok.
* After connecting the circuit,click on the play button to run the simulation
* Simulation can be stepped, paused or stopped at any time.

**Program:**

#include<reg51.h>

sbit ledG1=P0^1;

sbit ledR1=P0^2;

sbit ledY1=P0^3;

sbit ledG2=P0^4;

sbit ledR2=P0^5;

sbit ledY2=P0^6;

sbit ledG3=P2^1;

sbit ledR3=P2^2;

sbit ledY3=P2^3;

sbit ledG4=P2^4;

sbit ledR4=P2^5;

sbit ledY4=P2^6;

void delay(int x)

{

int i,j;

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

for(j=0;j<=1024;j++);

}

void main()

{

P0=0x00;

P2=0x00;

while(1)

{

ledG1=1;

ledR1=0;

ledR2=1;

ledR4=1;

ledR3=1;

delay(2000);

ledG1=0;

ledY1=1;

delay(800);

ledY1=0;

ledR1=1;

ledR2=0;

ledG2=1;

delay(2000);

ledG2=0;

ledY2=1;

delay(800);

ledY2=0;

ledR2=1;

ledR3=0;

ledG3=1;

delay(2000);

ledG3=0;

ledY3=1;

delay(800);

ledY3=0;

ledR3=1;

ledR4=0;

ledG4=1;

delay(2000);

ledG4=0;

ledY4=1;

delay(800);

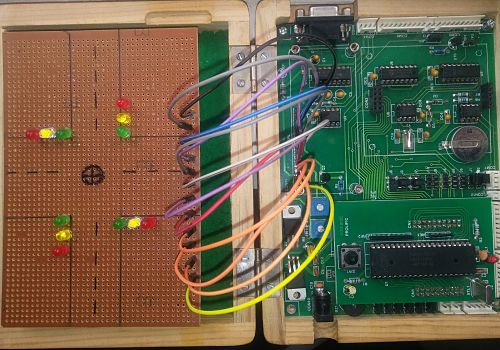
ledY4=0;

ledR4=1;

}

}

OUTPUT:



VIVAQUESTIONS:

1. What are hard and soft Real time systems?  
2. What is a semaphore? What are the different types of semaphore?  
3. Write a constant time consuming statement lot finding out If a given number Is a power of 2?  
4. What are recursive functions? Can we make them in line?  
5. What is the size of the int, char and float data types?  
6. What does malloc do? What will happen if we have a statement like malloc (sizeof(0));  
7. What is meant by a forward reference in C?  
8. What is the order of calling for the constructors and destructors in case of objects of inherited classes?  
9. Explain the properties of a Object oriented programming language.  
10. What do you mean by interrupt latency?  
11. What typecast is applied when we have a signed and an unsigned int in an expression?  
12. How are variables mapped across to the various memories by the C compiler?  
13. What is a memory leak? What is a segmentation fault?  
14. What is ISR? Can they be passed any parameter and can they return a value?  
15. a=7; b=8; x=a++-b; printf(“%d”, x ); What does this code give as output?  
16. What are little endian and big endian types of storage? How can you identify which type of allocation a system follows?  
17. What is the scope of a function that is declared as static?  
18. What is the use of having the const qualifier?  
19. Why do we need a infinite loop in embedded systems development? What are the different ways by which you can code in a infinite loop?  
20. What is the difference between embedded systems and the system in which rtos is running?

**21 .What Is The Difference Between Embedded Systems And The System In Which Rtos Is Running?**

**22 .What Is Pass By Value And Pass By Reference? How Are Structure Passed As Arguments?**

**23. What is the use of volatile keyword?**

**24. Can a variable be both const and volatile?**

24. What is the difference between testing and verification of vlsi circuit?  
25. While writing interrupt handlers (ISR), which are points needed to be considered?  
26. Explain can microcontroller work independently?  
27. Explain What happens when recursion functions are declared inline?  
28. Explain Scope of static variables?  
29. What is interrupt latency?  
30. Explain Operations involving unsigned and signed? Unsigned will be converted to signed?  
31. Explain Order of constructor and destructor call in case of multiple inheritance?  
32. Explain Difference between object oriented and object based languages?  
33. What are the advantages and disadvantages of using macro and inline functions?  
34. Explain why cannot arrays be passed by values to functions?  
35. Explain what is interrupt latency? How can we reduce it?  
36. Explain what are the different qualifiers in C?  
37. Explain What are the 5 different types of inheritance relationship?  
38. Explain What will this return malloc(sizeof(-10))?  
39. Explain Can structures be passed to the functions by value?  
40. Explain can we have constant volatile variable?  
41. Explain what are the different storage classes in C?  
42. Explain what is forward reference w.r.t. pointers in c?  
43. How is function itoa() written in C?  
44. Explain what is the difference between embedded systems and the system in which RTOS is running?  
45. How to define a structure with bit field members?  
46. Explain what is interrupt latency?  
47. Explain Scope of static variables?  
48. What is pass by value and pass by reference? How are structure passed as arguments?  
49. What is difference between using a macro and a in line function?  
50. What is the volatile keyword used for?

REALTIME APPLICATION:

Nowadays, controlling the traffic becomes major issue because of rapid increase in automobiles and also because of large time delays between traffic lights. So, in order to rectify this problem, we will go for density based traffic lights system. This article explains you how to control the traffic based on density.

EXPERIMENT -10

**Write a program to simultaneously Transmit and Receive data.**

**Aim:** To write an Embedded C program to simultaneously Transmit and Receive data by using Keil or Proteus software.

**Software used:**Computer,Keil Version4/5,Proteus 8.0

**Procedure:**

**Keil Software**

* Select the Kiel µVision software
* Select “Project” and “**New µVision Project**”.
* Create new project by entering your “**File name**” and then “**Save**” your file

##### Choose “**Atmel**” microcontroller from the database

* Select “**AT89C51**” µC and click “OK” and then “YES” and “YES”
* Make sure that “STARTUP.A51” file is added to the target.

##### Go to “File” and select “**New**” for text (program) editing window.

* Type your program in the editing window.
* Save your program by going to “File” -> “**Save**” option
* “Save in” your project folder.
* Give file name with “**.C**“ extension.
* And then click on “**Save**” option
* Right click on “Source Group1”
* Select “**Add Files** to ‘Group Source Group 1’”.
* Select to your Project folder
* Select “Files of type” as **“C source file”**
* Select your program file.
* And then click on “**Add**” to add the file to your source group.
* Notice that your file is added to the Source group
* Go to “**Debug**”.
* Select “**Start/ Stop Debug Session**” or press”Ctrl+F5” key
* **Selecting Output Window**.:- Choose appropriate Output window (Memory/serial/logic analyzer) according to your program output. Type in the input parameters (memory address/ port address/ timer) according to your program.
* **Execution**:- Go to “Debug”, Select “**Run**” or press” F5” key for one time execution. For single step execution Press11.

# [Proteus Software](http://www.circuitstoday.com/proteus-software-introduction)

* Open **Software** and select New design in  File menu
* An untitled design sheet will be opened, save it according to your wish,it is better to create a new folder for every layout as it generates other files supporting your design.
* To Select **components**, Click on the component mode button.
* Click On **Pick** from Libraries. It shows the categories of components available and a search option to enter the part name.
* Select the components from **categories** or type the part name in Keywords text box.
* The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.
* Place all the required components and **route the wires** i.e, make connections.
* Either selection mode above the **component mode** or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.
* Double click on the component to edit the properties of the components and click on Ok.
* After connecting the circuit,click on the play button to run the simulation
* Simulation can be stepped, paused or stopped at any time.

**Program**

#include <reg51.h>  
main()  
 {  
       TMOD=0x20;          // set timer1 in 16 bit timer mode  
       SCON=0x40;          // initialize serial communication  
       TL1=0xFD;             // load timer 1 to generate baud rate of 96KBps  
       TH1 = 0xFD;  
       TR1 = 1;               // start timer 1    
loop:REN = 1;               // enable reception  
      while(RI==0)

{

}        // wait until data is received  
      RI=0;                     // clear receive flag  
      ACC = SBUF;        // get data in to acc  
      REN = 0;                // now disable reception  
      SBUF = ACC;          // start transmission  
      while(TI==0){}       // wait until data transmitted  
      TI=0;                    // clear transmission flag     
      goto loop;              // continue this cycle

}

//main program

//

int main(void)

{

char key; //key char for keeping record of pressed key

int num1=0; //first number

char func=’+’;//function to be performed among two numbers

int num2=0; //second number

cct\_init(); //make input and output pins are required

1cdinit(); //intilize lcd

while(1)

{

//get function

key=get\_key();

writecmd(0x01); //clear display

writedata(key); // echo the key pressed to lcd

func=get\_func(key); // get int number from char value,it checks for rom input as well

if(func!=’e’); //if correct input then proceed,num1==error means wrong input

{

key=get\_key();

writedata(key); //echo the key press to lcd

num2=get\_num(key); //it checks for wrong func

if(num2!=error) //if correct input then proceed,func==’e’ means wrong input

{

key = get\_key();

writedata(key); //echo the key pressed to lcd

if(key==’=’) //if=is pressed then proceed

{

Switch(func) //switch on func

{

case ‘+’ : disp\_num(num1+num2); break;

case ‘-‘ : disp\_num(num1-num2); break;

case ‘x’ : disp\_num(num1\*num2); break;

case ‘/’ : disp\_num(num1/num2); break;

}

}

else //key other then=near means error wrong input

{

if(key == ‘c’) //if clear screen is pressed then clear screen and reset

writecmd(0x01); //clear screen

else

disperror(0); //display wrong input error

}

}

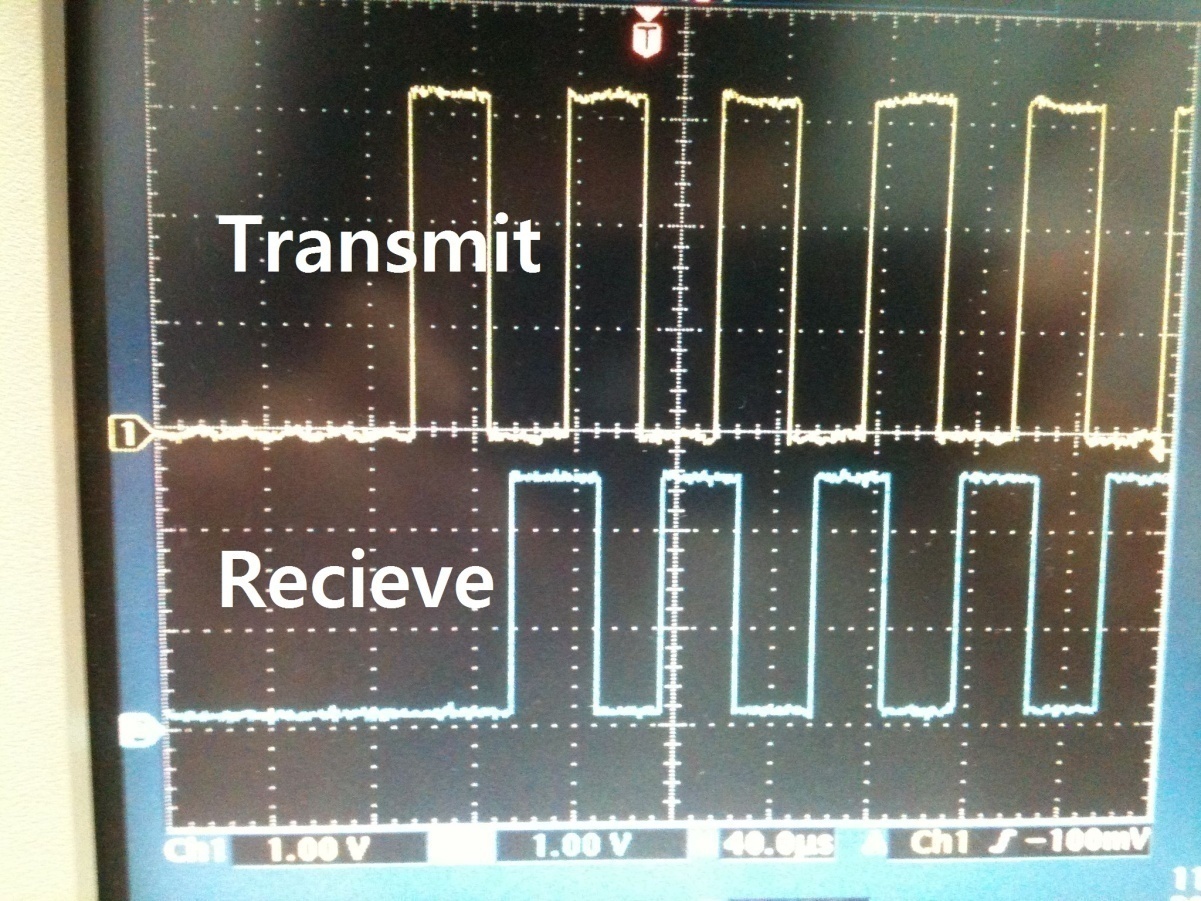
}

}

}

}

**Output:**



**VIVA QUESTIONS:**

1. What are hard and soft Real time systems?  
2. What is a semaphore? What are the different types of semaphore?  
3. Write a constant time consuming statement lot finding out If a given number Is a power of 2?  
4. What are recursive functions? Can we make them in line?  
5. What is the size of the int, char and float data types?  
6. What does malloc do? What will happen if we have a statement like malloc (sizeof(0));  
7. What is meant by a forward reference in C?  
8. What is the order of calling for the constructors and destructors in case of objects of inherited classes?  
9. Explain the properties of a Object oriented programming language.  
10. What do you mean by interrupt latency?  
11. What typecast is applied when we have a signed and an unsigned int in an expression?  
12. How are variables mapped across to the various memories by the C compiler?  
13. What is a memory leak? What is a segmentation fault?  
14. What is ISR? Can they be passed any parameter and can they return a value?  
15. a=7; b=8; x=a++-b; printf(“%d”, x ); What does this code give as output?  
16. What are little endian and big endian types of storage? How can you identify which type of allocation a system follows?  
17. What is the scope of a function that is declared as static?  
18. What is the use of having the const qualifier?  
19. Why do we need a infinite loop in embedded systems development? What are the different ways by which you can code in a infinite loop?  
20. What is the difference between embedded systems and the system in which rtos is running?

**21 .What Is The Difference Between Embedded Systems And The System In Which Rtos Is Running?**

**22 .What Is Pass By Value And Pass By Reference? How Are Structure Passed As Arguments?**

**23. What is the use of volatile keyword?**

**24. Can a variable be both const and volatile?**

24. What is the difference between testing and verification of vlsi circuit?  
25. While writing interrupt handlers (ISR), which are points needed to be considered?  
26. Explain can microcontroller work independently?  
27. Explain What happens when recursion functions are declared inline?  
28. Explain Scope of static variables?  
29. What is interrupt latency?  
30. Explain Operations involving unsigned and signed? Unsigned will be converted to signed?  
31. Explain Order of constructor and destructor call in case of multiple inheritance?  
32. Explain Difference between object oriented and object based languages?  
33. What are the advantages and disadvantages of using macro and inline functions?  
34. Explain why cannot arrays be passed by values to functions?  
35. Explain what is interrupt latency? How can we reduce it?  
36. Explain what are the different qualifiers in C?  
37. Explain What are the 5 different types of inheritance relationship?  
38. Explain What will this return malloc(sizeof(-10))?  
39. Explain Can structures be passed to the functions by value?  
40. Explain can we have constant volatile variable?  
41. Explain what are the different storage classes in C?  
42. Explain what is forward reference w.r.t. pointers in c?  
43. How is function itoa() written in C?  
44. Explain what is the difference between embedded systems and the system in which RTOS is running?  
45. How to define a structure with bit field members?  
46. Explain what is interrupt latency?  
47. Explain Scope of static variables?  
48. What is pass by value and pass by reference? How are structure passed as arguments?  
49. What is difference between using a macro and a in line function?  
50. What is the volatile keyword used for?

**REALTIME APPLICATION:**

Serial communication is commonly used for transmitting and receiving the signal. The 8051 microcontroller have consist [UART serial communication](https://www.elprocus.com/avr-microcontroller-serial-data-communication/) the signals transmitted and received by the Rx and Tx pins. The UART takes bytes of data and sends the individual bits in a sequential manner. The registers are a way to collect and store the data in the memory. UART is a half-duplex protocol. Half-duplex means transferring and receiving the data, but not at the same time.