Vidarbha Youth Welfare Society's
Prof. Ram Meghe Institute of Technology & Research Badnera, Amravati (M.S.) 444701

Practical Record
Semester-VIII

Subject: - Embedded System Lab (8KS06)

Sant Gadge Baba Amravati University Amravati

Department of
Computer Science & Engineering

Phone: (0721) - 2580402/2681246
Fax No. 0721 – 2681337
Website: www.mitra.ac.in
2019-2020

Vidarbha Youth Welfare Society's
Prof Ram Meghe Institute of Technology & Research
Badnera, Amravati (M.S.) 444701

Vidarbha Youth Welfare Society's
Prof Ram Meghe Institute of Technology & Research
Badnera, Amravati (M.S.) 444701

CERTIFICATE

This is to certify that Mr/Miss _____

EnrolmentNo._____Roll No. _____ Section_____of B.E.

Fourth year Semester VIII,Department of Computer Science & Engineering has satisfactorily completed the practical work of the subject **Embedded System Lab** prescribed by Sant Gadge Baba Amravati University, Amravatiduring the academic term 2019-2020.

Signature of the faculty **Head of Department**

Date:

INDEX

1. Mission & Vision statement of the Institute:

VISION

To become a pace-setting
centre of excellence believing in three
universal values namely
Synergy, Trust and Passion,
with zeal to serve the Nation
in the global scenario

MISSION

To dedicate ourselves
to the highest standard of technical education
& research in core & emerging engineering
disciplines and strive for the overall personality
development of students so as to nurture
not only quintessential technocrats
but also responsible citizens

Mission and Vision of the Department

Vision

To ensure that the world saves time and other depletable resources and free it from complexity by providing efficient computing services.

Mission

To dedicate ourselves to the highest standard by providing knowledge, skills and wisdom to the incumbent by imparting value based education to enable them to solve complex system by simple algorithms and to make them innovative, research oriented to serve the global society, while imbibing highest ethical values.

2.Programme Educational Objectives (PEOs)

PE01. Preparation: To prepare students for successful careers in software industry that meet the needs of Indian and multinational companies or to excel in Higher studies.

PEO2. Core competence: To develop the ability among students to synthesize data and technical concepts for software design and development.

PEO3. Breadth: To inculcate in students professional and ethical attitude, effective communication skills, teamwork skills, multidisciplinary approach and an ability to relate engineering issues to broader social context.

PEO4. Professionalism: To provide students with a sound foundation in the mathematical, scientific and computer engineering fundamentals required to solve engineering problems and also pursue higher studies.

PEO5. Learning Environment: To promote student with an academic environment aware of excellence, leadership, written ethical codes and guidelines and the life-long learning needed for a successful professional career.

- **PROGRAM OUTCOMES (POs)**

Engineering Graduate will be able to:

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

- **PROGRAM SPECIFIC OUTCOMES (PSO's)**

PS01: Foundation of Computer System Development: Ability to use knowledge of computer systems and design principles in building the hardware and software components, products in the domain of embedded system, artificial intelligence, databases, networking, web technology and mobile computing.

PS02: Problem Solving Ability: Ability to apply knowledge in various problem domains and implement innovative and suitable solutions to cater to needs of industry, business and e-governance by imbibing highest ethical and economical values.

3.Syllabus, Course outcomes and Mapping of CO's with PO's

Subject Name and Code:	**Embedded Systems (8KS02)**
Faculty Name:	P. K. Agrawal (pkagrawal@mitra.ac.in) A. U. Chaudhari (auchaudhari@mitra.ac.in) N. A. Deshmukh (nadeshmukh@mitra.ac.in)
Course Type:	Theory + Lab
Compulsory / Elective:	Compulsory
Teaching Methods:	**Lecture:** 04 Hrs/ week. **Laboratory:** 02 Hrs/ week (03 Batches)
Subject Credits:	04 Th + 01 Lab (as per given in syllabus). **Discussion:** Office hour discussion, Quiz.
Course Assessment:	**Homework:** Consisting of assignments. **Exams:** 2 Class Tests + 1 Improvement Test Semester end examination by SGBAU
Grading Policy:	**20%** Assignments, class test and viva-voce **80%** Semester end examinations
Prerequisite	Assembly Language Programming Computer Architecture Programming with C
Unit wise Course Content:	**Unit 1:** Introduction to Embedded System: Embedded Systems Vs General Computing Systems. History, classification, major application areas and purpose of Embedded Systems. Components of Embedded system: General Purpose and Domain Specific Processors, Memories for embedded systems. **Unit 2:** Components of Embedded system: Sensors & Actuators, Communication Interface, Embedded Firmware and other components. Characteristics of Embedded System, Quality Attributes of Embedded System. Embedded Systems Examples: Washing machine.Automotive application. **Unit 3:** Introduction to 8051 Microcontroller: 8051 Architecture, 8051 Memory Organization, Registers, Oscillator Unit, Ports, 8051 Interrupt System, Timer units, the Serial Port, 8051 Power Saving Modes.

Unit 4: Programming the 8051 Microcontroller: Addressing modes. 8051 Instruction Set: Data transfer instructions, Arithmetic instructions, Logical instructions, Boolean instructions, and Program Control Transfer instructions. Assembly Language based Embedded Firmware development.

Unit 5: Programming in Embedded C: Review of various constructs in C. Constant declarations, 'volatile' type qualifier, Delay generation and Infinite loops in Embedded C. Coding Interrupt Service Routines, Recursive and Re-entrant Functions, Dynamic memory allocation.
Unit 6: Vx Works Real Time Operating System (RTOS): , Real Time Kernel, Hard/Soft Real time. VxWorks Task Creation, Management and Task Scheduling, Kernel Services, Inter Task Communication, VxWorks Task Synchronization and Mutual Exclusion, Interrupt Handling, Watchdog for task Execution monitoring, Timing and Reference in Vx Works.

Text Book: Shibu K V "Introduction to Embedded Systems" McGraw-Hill.

Reference Books:

1. Rajkamal, "Embedded Systems, Architecture, Programming & Design" TMH.
2. Tammy Noergaard "Embedded Systems Architecture" Elsevier Newness Publication.
3. Vahid and Givargis "Embedded System Design" John Wiley & Sons P Ltd.
4. Peter Marwedel "Embedded Systems Design" Springer, Netherland.

Units covered in the course and contact hours per unit: **Unit 1:** 8 hrs **Unit 2:** 8 hrs **Unit 3:** 8 Hrs **Unit 4:** 8 hrs **Unit 5:** 8 hrs **Unit 6:** 8 Hrs

Course Learning Objective:

K802.1. In this course, Student will learn basics of embedded system hardware with required programming language to make the electronic device to work according to the order.

K802.2. Learn about various hardware which is essential for any general purpose system and its components with having various attributes such as single 7-seg LED and multi-seg LED and development of embedded code for various system with different architecture.

K802.3. Understanding the architecture of embedded system, with its attributes. Assigning the required hardware for the existing system.Understanding the working of Timer, PORTs and other hardware with embedded system.

K802.4. Enabling the hardware to communicate with each other using inter process communication and various methods. Understanding the data transfer between two electronic devices.

K802.5. Introduce the student with programming in embedded 'C', with its advanced features. Enabling the embedded system with its connected devices using programming languages. Student will learn to develop the software for the upliftment of the standalone electronic device.

K802.6. Student will learn the Real Time system which works with the continuous connectivity and the continuous data generation. By monitoring such system such will understand the scheduling of the process and allotment of memory space available.

K806.1 Subject specific skills with ability to interface various components of Embedded kit with 8051 microcontroller.

Upon completion of this course, students will have had an opportunity to learn about the following:

	Course Specific Practical Outcomes	**Course Outcomes**
1	Understand the basics of Embedded System and its different architecture like Von -Nueman, Harvard architecture.	**K802.1, K802.2, K806.1**
2	Learn basics of domain specific controller and General Purpose Processor.	**K802.1, K802.2, K802.3, K806.1**
3	Design & implement the embedded program for output devices such as Single 7 segment display using embedded C programming	**K802.1, K802.2, K802.4, K806.1**
4	Able to understand the differentiable attribute of General Purpose System and Embedded System.	**K802.1, K802.2, K802.4, K806.1**
5	Design & implement the embedded programs on the buzzer and the usage of buzzer using embedded C Programming	**K802.1, K802.2, K802.4, K806.1**

6	Design & implement the program for 16x2 display using embedded C programming.	**K802.1, K802.2, K802.5, K806.1**
7	Design & implement the program for stepper motor which will rotates in clock wise or anticlockwise direction or in a certain formation	**K802.1, K802.2, K802.5, K806.1**
8	Able to understand the working of switch for what purpose it is used and where	**K802.1, K802.2, K802.5, K806.1**
9	Able to understand the requirement of PSW and its functionality in embedded system	**K802.1, K802.2, K802.5, K806.1**
10	Understand the matrix key terminology using embedded C Programming.	**K802.1, K802.2, K802.5, K806.1**
11	Understand to demonstrate the working of IR Remote using embedded C programing.	**K802.1, K802.2, K802.5, K806.1**
12	Able to understand the Delay generation in embedded system using C Programming	**K802.1, K802.2, K802.5, K806.1**
13	Understand the Hard real time system for the betterment of understanding the real time operating system.	**K802.1, K802.2, K802.6, K806.1**

MappingofCourse Specific Practical Outcomes with Course Outcomes

Embedded Systems	**K802.1, K802.2, K802.3, K802.4, K802.5, K802.6, K806.1**
StrongCO'S(≥70%)	**K802.1, K802.2, K802.5, K806.1**
Average CO's(<70)	**K802.4**
SomewhatCO's	**K802.3**

Mapping of Course Outcomes with Program Outcomes

Enter correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Course	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
K802.1	2	2	2	2	3	1	-	2	-	2	1	3
K802.2	2	2	2	2	3	1	-	2	-	2	1	3
K802.3	2	2	3	3	2	3	-	3	-	-	-	3
K802.4	2	2	3	3	2	3	-	3	-	-	-	3
K802.5	2	2	3	3	2	3	-	3	-	-	-	3
K802.6	2	2	3	3	2	3	-	3	-	-	-	3

4.AssessmentFormati.e. ACIPV (Guidelines)

RegularAssessmentofExperimentduringthepractical session

(Sample Sheet)
RegularAssessment

Year:-
Branch:-

AssessmentSheet

Rollno.	A (5marks)	C (5marks)	I (5marks	P (5marks	V (5marks	Total (25Marks

(A-Attendance,C-Competency,I-Innovation,P-Presentation,V-Viva)

GuidelinesforAwardingInternalMarksForPractical

Each experiment/practical carries 25 marks.The student shall be evaluated for2 5marks as per the following guidelines. At the end of the semester, internal assessment marks for practical shall be the average of marks scored in all the experiments.

a. **Attendance(5marks):**These5markswillbegiventotheregularityofastudent.Ifthe studentispresent on thescheduled time,he/she willbeawarded5marks otherwise will begiven2.5marksforhis attendance.

b. Competency (5 marks): Here the basic aim is to check whether the student has developedtheskilltowritedownthecodeonhisownanddebug.Ifastudentexecutes thepracticalonthescheduleddaywithintheallocatedtime,hewillbeawardedfull marks.Otherwise,markswillbegivenaccording to thelevelofcompletion/execution of practical

c. Innovation(5marks): Herethebasicobjectiveistoexploretheinnovativeideasfrom thestudentswith respectto thecorrespondingpractical orhowinnovativelyheinterprets theaim ofthepractical.Itisexpectedthatthestudentsmustbeawareaboutthescopeof practical preciselysuch thatinfuture,they couldimplementthetaskinvarious applications.

d. Performance/Participation in-group activity(5 marks): These marks will be given on how a student is actively participating the group.If the student is performing the practical as the part of a group he must participate actively.

e. Viva-Voce(5marks): These 5 marks will be totally based on the performance in viva-voce.There shall be viva-voce on the completion of each practical in the same practical session.The student shall get the record checked before next practical.

MappingofGuidelines withPO's ofthe department:

Guidelines forAwardingInternalMarks ForPractical	**ProgramOutcomes**
Attendance	**PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO12**
Competency	**PO1, PO2, PO3, PO4, PO5, PO10, PO12**
Innovation	**PO1, PO2, PO3, PO4, PO5, PO11, PO12**
Performance/Participationin-groupactivity	**PO1, PO2, PO3, PO4, PO5,PO9, PO11, PO12**
Viva-Voce	**PO1, PO2, PO3, PO4, PO5, PO10, PO12**

5. About theLab manual:

This manual is designed for the Fourth year (VIII Semester) students of Computer Science & Engineering. The aim of the Embedded System Lab is to developed student's programming skills in embedded system for their applications.

This lab manual can be used as instructional book for students,staff and instructors to assist in performing and understanding the practical.In the first part of the manual,practical as per syllabus are described and in thes of the manual,practical that are beyond the syllabus but expected for university laboratory examination are displayed.

6. List ofPractical's

Department ofComputer Science & Engineering

Subject:-**Embedded System Lab**
(8KS06) Semester:-VIII

Sr. No	LISTOFPRACTICALS	Page
1.	To study 8051 microcontroller and its peripherals.	
2.	Write and Execute a program for flashing LED's	
3.	Write and execute a program to display numbers with the help of single 7-segment display	
4.	Write and execute a program to display numbers with the help of multiple 7-segment display	
5.	Write and execute a program to turn the buzzer ON and OFF	
6.	Write and execute a program for display the character using the 16 X 2 LCD display	
7.	Write and execute a program for rotating of stepper motor in clockwise and anticlockwise using Full Drive	
8.	Write and execute a program for rotating of stepper motor in clockwise and anticlockwise using Wave Drive	
9	Write and execute a program to demonstrate the Relay function	
10.	Write and execute a program to demonstrate the key technology on matrix	
11.	Write a Program to Demonstrate Switch.	
12.	Write a Program to Demonstrate PWM.	
13.	Write a Program to Demonstrate IR Remote.	
14.	To Study Vx Works Operating System	

Sr.No.	*LIST OF EXPERIMENTS* *(Actual ConductedFor the Session)*	Date	Page No.	Remark
1.	To study 8051 microcontroller and its peripherals.			
2.	Study and execute a program for control and looping statement and various functionality of Embedded C			
3.	Write and execute a program for flashing LED's			
4.	Write a Program to Demonstrate Single Seven Segment Display.			
5.	Write and execute a program for buzzer with ON and OFF flashing LED's			
6.	Interfacing 4 Switches and 4LEDs with 8051.			
7.	Write and execute a program for rotating of stepper motor in clockwise and anticlockwise Using Full Drive.			
8.	To Study Vx Works Operating System			

Practical No. 01

Aim: To study 8051 microcontroller and its peripherals.

Theory:

1. Introduction

- The first microprocessor 4004 was invented by Intel Corporation. 8085 and 8086 microprocessors were also invented by Intel. In 1981, Intel introduced an 8-bit microcontroller called the 8051.
- A highly integrated silicon chip containing a CPU, scratch pad RAM, Special and General purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports
- Microcontrollers can be considered as a super set of Microprocessors
- Microcontroller can be general purpose (like Intel 8051, designed for generic applications and domains) or application specific (Like Automotive AVR from Atmel Corporation. Designed specifically for automotive applications)
- Since a microcontroller contains all the necessary functional blocks for independent working, they found greater place in the embedded domain in place of microprocessors.
- Microcontrollers are cheap, cost effective and are readily available in the market

2. Block diagram 8051 MC

The following illustration shows the block diagram of an 8051 microcontroller −

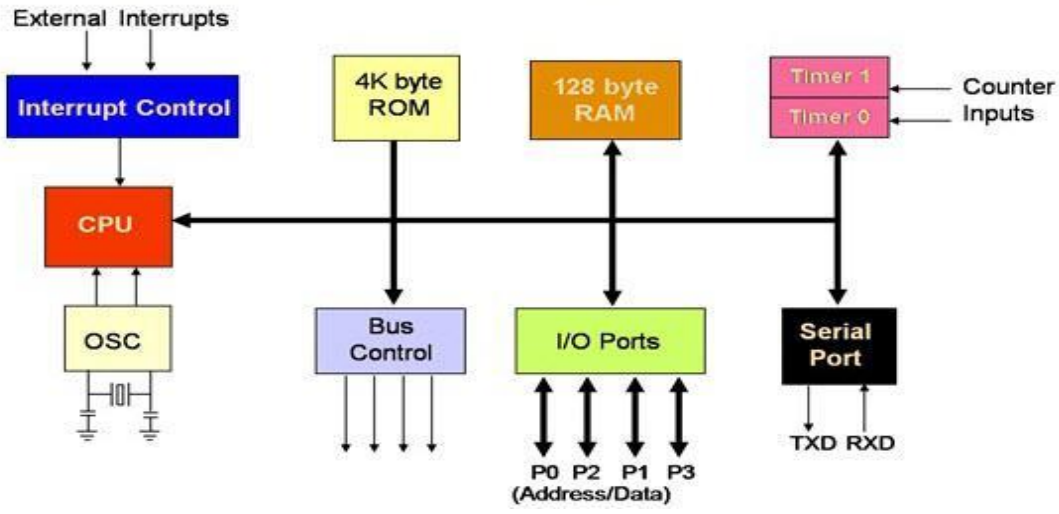


Figure1.1: Block Diagram of 8051 Microcontroller

- **CPU (Central Processor Unit):**

It scrutinizes and manages all processes that are carried out in the Microcontroller. User has no power over the functioning of CPU. It interprets program printed in storage space (ROM) and carries out all of them and do the projected duty. CPU manages different types of registers in 8051 microcontroller.

- **Interrupts:**

Interrupts are the events that temporarily suspend the main program, pass the control to the external sources and execute their task. It then passes the control to the main program where it had left off.

- **Memory:**

Micro-controller needs a program which is a set of commands. This program enlightens Microcontroller to perform precise tasks. These programs need a storage space on which they can be accumulated and interpret by Microcontroller to act upon any specific process. The memory which is brought into play to accumulate the program of Microcontroller is recognized as Program memory or code memory. In common language it's also known as Read Only Memory or ROM.

- **Bus:**

Fundamentally Bus is a group of wires which functions as a communication canal or mean for the transfer Data. These buses comprise of 8, 16 or more cables. As a result, a bus can bear 8 bits, 16 bits all together. There are two types of buses:

Address Bus: Microcontroller 8051 consists of 16 bit address bus. It is brought into play to address memory positions. It is also utilized to transmit the address from Central Processing Unit to Memory.

Data Bus: Microcontroller 8051 comprise of 8 bits data bus. It is employed to cart data.

- **Oscillator:**

Microcontroller 8051 consists of an on-chip oscillator which toils as a time source for CPU (Central Processing Unit). As the productivity thumps of oscillator are steady as a result, it facilitates harmonized employment of all pieces of 8051 Microcontroller. Input/output Port: As we are acquainted with that Microcontroller is employed in embedded systems to manage the functions of devices.For this function

Micro-controller 8051 consists of 4 input/output ports to unite it to other peripherals. Timers/Counters: Micro-controller 8051 is incorporated with two 16 bit counters & timers. The counters are separated into 8 bit registers. The timers are utilized for measuring the intervals, to find out pulse width etc.

3. Architecture Diagram of 8051MC

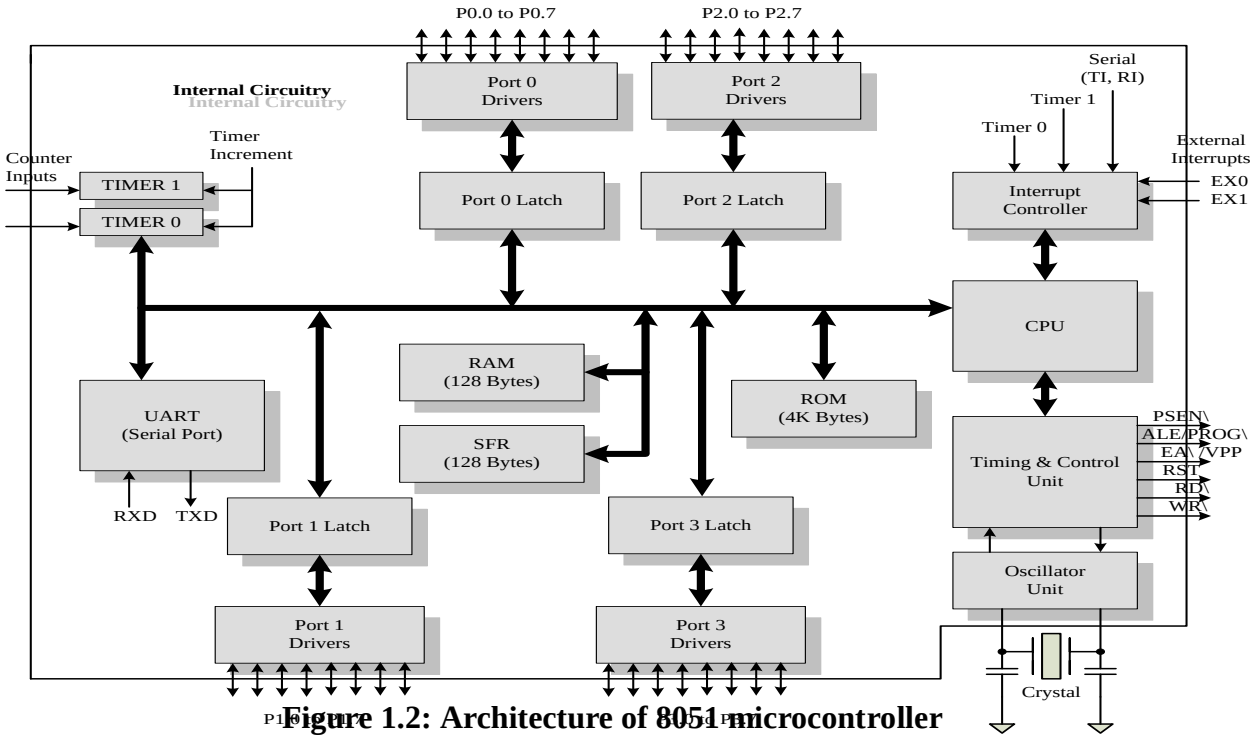


Figure 1.2: Architecture of 8051 microcontroller

4. Feature of 8051 MC

- Built around an 8bit CPU with Boolean processing capability
- Built-in oscillator driver unit
- 4K bytes of On-chip Program memory
- 128 bytes of internal Data memory
- 128 bytes of Special Function Register memory area
- 32 general purpose I/O lines organized into four 8bit bi-directional ports
- Two 16 bit timer units
- A full duplex programmable UART for serial data transmission with configurable baud rate
- Built around the '*Harvard*' processor architecture

- The program and data memory of 8051 is logically separated and they physically reside separately
- Separate address spaces are assigned for data memory and program memory
- 16 bit wide address bus which can address code memory up to 64KB (2^{16})

Conclusion:

Viva Question:

1) What is Embedded System?
2) What is task oriented system?
3) How many I/O ports are placed in microcontroller 8051?
4) Is it required that, embedded system must enabled with operating system.
5) What is use of EA pin?
6) What is interrupts signal?

ACHIEVED COURSEOUTCOMES:

At the end of this experiment the student will be able to:-

Sr. No	**CourseOutcomes**	**MappingwithPos**
1	Understanding the concepts of 8051 microcontroller and the its peripheral.	**PO1, PO2, PO3, PO4, PO5, PO10, PO12**
2	Understand the requirement of the task oriented devices	**PO1, PO2, PO3, PO4, PO5, PO10, PO12**

1	ProgramCourseoutcomes	**PO1, PO2, PO3, PO4, PO5, PO10,**
2	StrongPOs(≥50%)	**PO1, PO2, PO3, PO4, PO5, PO10,**
3	SomewhatPos	

Signature

Department of Computer Science & Engineering

Practical No. 02

Aim: Study and execute a program for control and looping statement and various functionality of Embedded C

Software Required: RIDE

Theory:

Embedded C is most popular programming language in software field for developing electronic gadgets. Each processor used in electronic system is associated with embedded software.

Embedded C programming plays a key role in performing specific function by the processor. In day-to-day life we used many electronic devices such as mobile phone, washing machine, digital camera, etc. These all device working is based on microcontroller that are programmed by embedded C.

In embedded system programming C code is preferred over other language. Due to the following reasons:

- Easy to understand
- High Reliability
- Portability
- Scalability

Let's have a glance of various functionalities of Embedded C

Function()

We can divide a large program into the basic building blocks known as function. The function contains the set of programming statements enclosed by {}. A function can be called multiple times to provide reusability and modularity to the C program.

The syntax of creating function in c language is given below

return_type function_name(data_type parameter,...)

```
{
        //code to be executed
}
```

A function may or may not accept any argument. It may or may not return any value. Based on these facts, There are four different aspects of function calls.

- function without arguments and without return value
- function without arguments and with return value
- function with arguments and without return value
- function with arguments and with return value

Control Statements

There are the following variants of if statement in C language.

- If statement
- If-else statement
- If else-if ladder
- Nested if

If Statement

The if statement is used to check some given condition and perform some operations depending upon the correctness of that condition. It is mostly used in the scenario where we need to perform the different operations for the different conditions. The syntax of the if statement is given below.

```
if(expression){
        //code to be executed
}
```

If-else statement

The if-else statement is used to perform two operations for a single condition. The if-else statement is an extension to the if statement using which, we can perform two different operations. Using if-else statement is always preferable since it always invokes an otherwise case with every if condition. The syntax of the if-else statement is given below.

```
if(expression){
        //code to be executed if condition is true
}else{
        //code to be executed if condition is false
}
```

If else-if ladder Statement:

The if-else-if ladder statement is an extension to the if-else statement. It is used in the scenario where there are multiple cases to be performed for different conditions. In if-else-if ladder statement, if a condition is true then the statements defined in the if block will be executed, otherwise if some other condition is true then the statements defined in the else-if block will be executed, at the last if none of the condition is true then the statements defined in the else block will be executed.

```
if(condition1){
        //code to be executed if condition1 is true
}else if(condition2){
        //code to be executed if condition2 is true
}
else if(condition3){
        //code to be executed if condition3 is true
}
        ...
else{
        //code to be executed if all the conditions are false
}
```

Switch Statement:

The switch statement is an alternate to if-else-if ladder statement which allows us to execute multiple operations for the different possible values of a single variable called switch variable. Here, We can define various statements in the multiple cases for the different values of a single variable.

The syntax of switch statement is given below:

```
switch(expression)
        {
        case value1:
        //code to be executed;
        break;  //optional
        case value2:
        //code to be executed;
        break;  //optional
                .....
        default:
        code to be executed if all cases are not matched;
}
```

Rules for switch statement

1) The *switch expression* must be of an integer or character type.

2) The *case value* must be an integer or character constant.

3) The *case value* can be used only inside the switch statement.

4) The *break statement* in switch case is not must. It is optional. If there is no break statement found in the case, all the cases will be executed present after the matched case. It is known as *fall through* the state of C switch statement.

Looping Statement

The looping can be defined as repeating the same process multiple times until a specific condition satisfies. There are three types of loops used. The looping simplifies the complex problems into the easy ones. It enables us to alter the flow of the program so that instead of writing the same code again and again, we can repeat the same code for a finite number of times.

Advantage of loops

1) It provides code reusability.

2) Using loops, we do not need to write the same code again and again.

3) Using loops, we can traverse over the elements of data structures (array or linked lists).

There are three types of loops:

1. do while
2. while
3. for

1. do-while

The do-while loop continues until a given condition satisfies. It is also called post tested loop. It is used when it is necessary to execute the loop at least once (mostly menu driven programs).

The syntax of do-while loop is given below:

```
do{

        //code to be executed
}while(condition);
```

2. While

The while loop is to be used in the scenario where we don't know the number of iterations in advance. The block of statements is executed in the while loop until the condition specified in the while loop is satisfied. It is also called a pre-tested loop.

The syntax of while loop is given below:

```
while(condition){
        //code to be executed
}
```

3. For

The for loop is used in the case where we need to execute some part of the code until the given condition is satisfied. The for loop is also called as a per-tested loop. It is better to use for loop if the number of iteration is known in advance.

The syntax of for loop is given below:

```
for(initialization;condition;incr/decr)
        {
        //code to be executed
        }
```

Write a program with the help of control and looping statements using Embedded C

```
#include <stdio.h>
int main()
{
  int n, first = 0, second = 1, next, c;

  printf("Enter the number of terms\n");
  scanf("%d", &n);
```

```
  printf("First %d terms of Fibonacci series are:\n", n);

  for (c = 0; c < n; c++)
  {
    if (c <= 1)
      next = c;
    else
    {
      next = first + second;
      first = second;
      second = next;
    }
    printf("%d\n", next);
  }

  return 0;
}
```

CONCLUSION:

_____

_____

_____

_____

VIVA QUESTION:

1) What is procedural language?
2) What is Object orientation?
3) How many data types are available in embedded C?

4) What is call by reference and call by value?

5) With the help of what logic the system will act as embedded system?

ACHIEVED COURSEOUTCOMES:

At theend of this experiment the studentwillbeable to:-

Sr. No	CourseOutcomes	MappingwithPos
1	Understanding the concepts of variables and passing the values	**PO1, PO2, PO3, PO4, PO5, PO10, PO12**
2	Understand the requirement of the control statements and looping statements	**PO1, PO2, PO3, PO4, PO5, PO10, PO12**

1	ProgramCourseoutcomes	**PO1, PO2, PO3, PO4, PO5, PO10,**
2	StrongPOs(≥50%)	**PO1, PO2, PO3, PO4, PO5, PO10,**
3	SomewhatPos	

Signature

Department of Computer Science & Engineering

Practical No. 03

Aim: Write and execute a program for flashing LED's.

Software Required:

Theory:

A **light-emitting diode** (**LED**) is a two-lead semiconductor light source. It is a p–n junction diode that emits light when activated. When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor. LEDs are typically small (less than 1 mm2) and integrated optical components may be used to shape the radiation pattern.

Light Emitting Diodes are the semiconductor light sources. Commonly used LEDs will have a cut-off voltage of 1.7V and current of 10mA. When an LED is applied with its required voltage and current it glows with full intensity. The Light Emitting Diode is similar to the normal PN diode but it emits energy in the form of light. The color of light depends on the band gap of the semiconductor. The following figure shows "how an LED glows?"

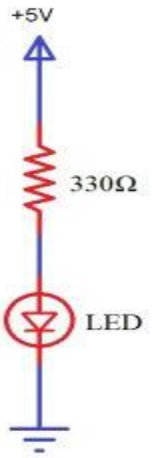


Figure 2.1: Circuit diagram of flashing LED

Thus, LED is connected to the AT89C51 microcontroller with the help of a current limiting resistor. The value of this resistor is calculated using the following formula.

R= (V-1.7)/10mA, where V is the input voltage.

LEDs have many advantages over incandescent light sources, including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. Light-emitting diodes are used in applications as diverse as aviation lighting, automotive headlamps, advertising, general lighting, traffic signals, camera flashes, and lighted wallpaper. They are also significantly more energy efficient and, arguably, have fewer environmental concerns linked to their disposal.

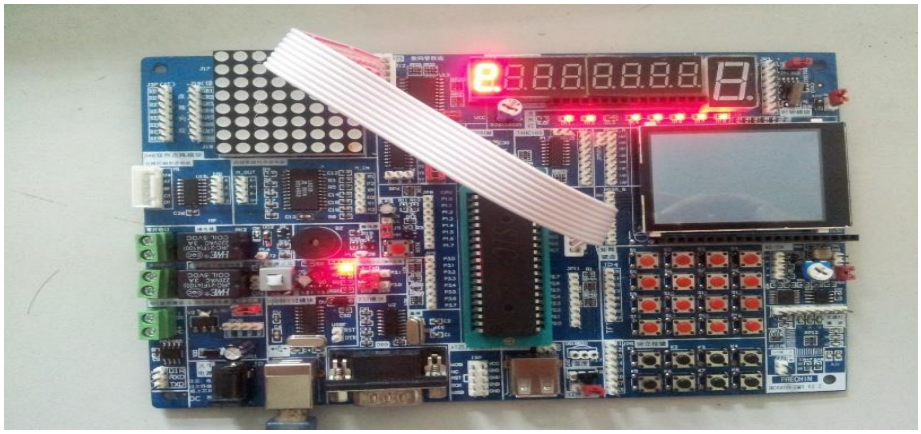
LED Conn Seq (cross)		**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Sr.No	**HEX DEC**	**P7**	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**
1	0X80	1	0	0	0	0	0	0	0
2	0X40	0	1	0	0	0	0	0	0
3	0X20	0	0	1	0	0	0	0	0
4	0X10	0	0	0	1	0	0	0	0
5	0X08	0	0	0	0	1	0	0	0
6	0X04	0	0	0	0	0	1	0	0
7	0X02	0	0	0	0	0	0	1	0
8	0X01	0	0	0	0	0	0	0	1

PROGRAM:

```
#include <reg51.h>
void Delay10ms (unsigned int c);
void main ()
{
while (1)
        {
                        {
                                P1 = 0x80;
                                Delay10ms(50);
                                P1 = 0x40;
                                Delay10ms(50);
```

```
                                        P1 = 0x20;
                                        Delay10ms(50);
                                        P1 = 0x10;
                                        Delay10ms(50);
                                        P1 = 0x08;
                                        Delay10ms(50);
                                        P1 = 0x04;
                                        Delay10ms(50);
                                        P1 = 0x02;
                                        Delay10ms(50);
                                        P1 = 0x01;
                                        Delay10ms(50);
                        }
        }

}

void Delay10ms (unsigned int c)
{
unsigned char a, b;
for (; c> 0; c--)
        {
                {
                                for (b = 38; b> 0; b--)
                        {
                                for (a = 130; a> 0; a--);
                        }
                }

        }

}
```

OUTPUT:



Conclusion:

_____

_____

_____

_____

Viva Question:

1. Explain the Hexadecimal conversion of the binary number format.
2. Explain the sequence of generation of series in LED.
3. What is/are the consequences of driving the LED in the form of an output function?

ACHIEVED COURSEOUTCOMES:

At theend of this experiment the studentwillbeable to:-

Sr. No	**CourseOutcomes**	**MappingwithPos**
1	Able to interface the programming with the hardware	**PO1, PO2, PO3, PO4, PO5,**
2	Able to learn the hexadecimal code conversion.	**PO1, PO2, PO3, PO4, PO5,**
3	Able to generate the series of led sequence.	**PO1, PO2, PO3, PO4, PO5,**
1	ProgramCourseoutcomes	**PO1, PO2, PO3, PO4, PO5,**
2	StrongPOs (≥50%)	**PO1, PO2, PO3, PO4, PO5,**
3	SomewhatPos	

Signature

Department of Computer Science & Engineering

Practical No. 04

Aim: Write a Program to Demonstrate Single Seven Segment Display.

Software Required: Kiel Version: 2 or 3, STC ISP

Theory:

The seven segment display is the most common display device used in many gadgets, and electronic appliances like digital meters, digital clocks, microwave oven and electric stove, etc. These displays consist of seven segments of light emitting diodes(LEDs) and that is assembled into a structure like numeral 8. Actually seven segment displays contain about 8-segments wherein an extra 8th segment is used to display dot. This segment is useful while displaying non integer number. Seven segments are indicated as A-G and the eighth segment is indicated as H.

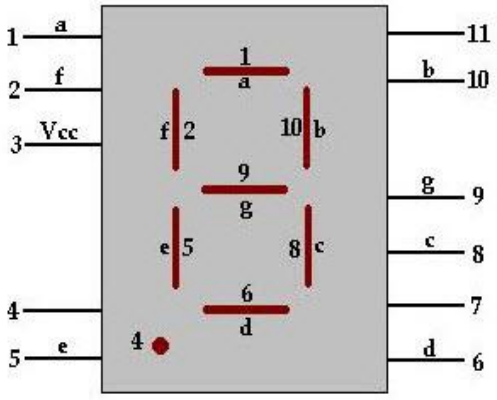


Figure 3.1: Seven Segment Display pin Diagram

A seven segment displays are generally available in ten pin package. In that 8 pins relate to the 8 LEDs, the remaining pins at middle are internally shorted. These segments come in two outlines they are common cathode and common anode. In common cathode configuration, the negative terminals are connected to the common pins and the common is connected to the ground. When the corresponding pin is given high, then particular LED glows. In a common anode arrangement, the common pin is given to a logic high and the pins of the LED are given low to display a number.

Working of Seven Segment Display

When the power is given to all the segments, then the number 8 will be displayed. If you disconnect the power for segment G (that means 7) then that will result number 0. The circuit of the seven segment display is designed in such a way that the voltage at different pins can be applied at the same time. In the same way, you can form the combinations to display numerals from 0 to 9. Practically, seven segment displays are available with two structures, both the type of displays consists of 10 pins.

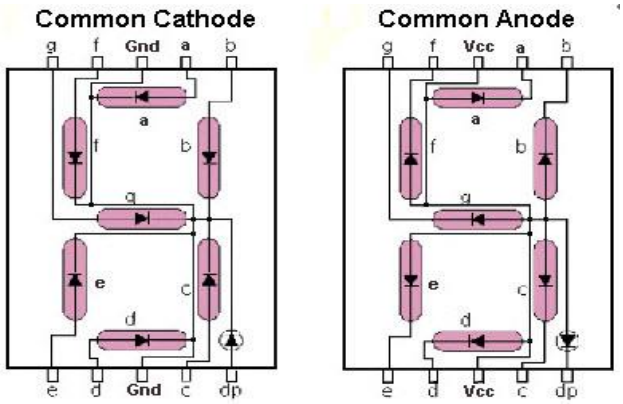


Figure 3.2: 7- Segment Display Configuration

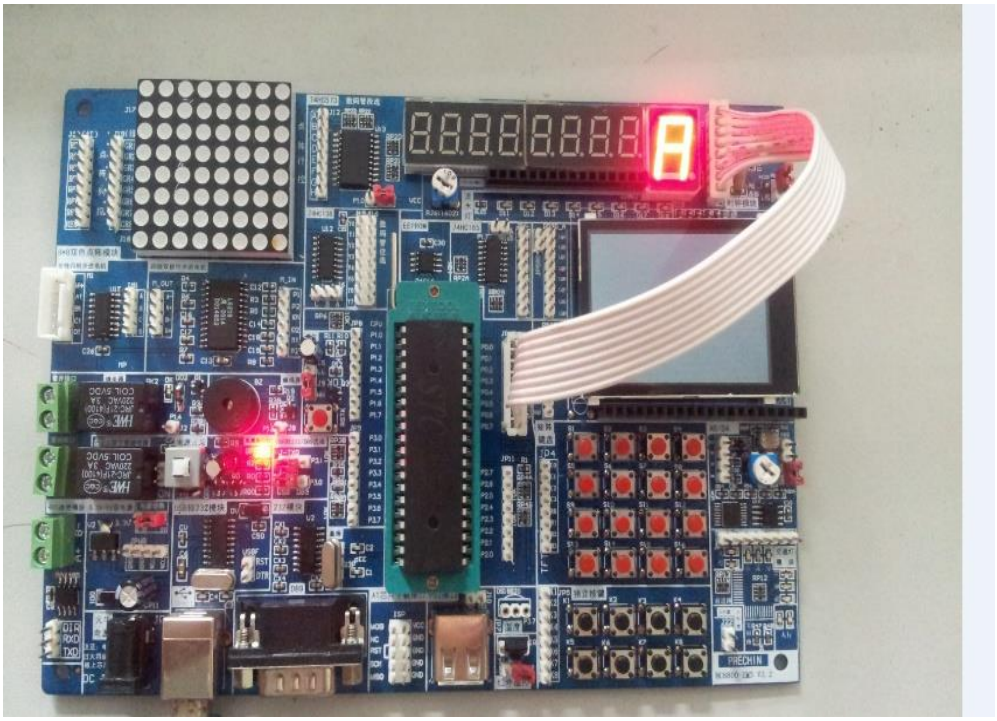
Logic:

LED Conn Seq			H	G	F	E	D	C	B	A
Sr. No	**Number**	**HEX DEC**	**P7**	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**
1	0	**0X40**	0	1	0	0	0	0	0	0
2	1	**0XF9**	1	1	1	1	1	0	0	1
3	2	**0X24**	0	0	1	0	0	1	0	0
4	3	**0X30**	0	0	1	1	0	0	0	0
5	4	**0X19**	1	0	0	1	0	0	0	1
6	5	**0X12**	0	0	0	1	0	0	1	0
7	6	**0X02**	0	0	0	0	0	0	1	0
8	7	**0XF8**	1	1	1	1	1	0	0	0
9	8	**0X00**	0	0	0	0	0	0	0	0
10	9	**0XF10**	0	0	0	1	0	0	0	0

Program:

```
#include <reg51.h>
void Delay10ms (unsigned int c);
void main ()
{
while (1)
        {
                        {

                                P1 = 0xF9;
                                Delay10ms(50);
                                P1 = 0x24;
                                Delay10ms(50);
                                P1 = 0x30;
                                Delay10ms(50);
                                P1 = 0x19;
                                Delay10ms(50);
                        }
        }

}
void Delay10ms (unsigned int c)
{
unsigned char a, b;
for (; c> 0; c--)
        {
                {
                                for (b = 38; b> 0; b--)
                        {
                                for (a = 130; a> 0; a--);
                        }
                }

        }

}
```

Output:



Conclusion:

_____

_____

_____

Viva Questions:

1) How many cathode and anodes are available in single 7-segment display?
2) How encoding are performed to display '3' in a single 7 segment display?

ACHIEVED COURSEOUTCOMES:

At theend of this experiment the studentwillbeable to:-

Sr.No	CourseOutcomes	MappingwithPOs
1	Understand the concept of 7-Segment display	**PO1, PO2, PO3, PO4, PO5, PO12**
2	Design and implement 7-Segment	**PO1, PO2, PO3, PO4, PO5, PO12**

1	ProgramCourseoutcomes	**PO1, PO2, PO3, PO4, PO5, PO12**
2	StrongPOs(≥50%)	**PO1, PO2, PO3, PO4, PO5, PO12**
3	SomewhatPos	

Signature

Department of Computer Science & Engineering

Practical No. 05

Aim: Write and execute a program for buzzer with ON and OFF flashing LED's

Software Required:

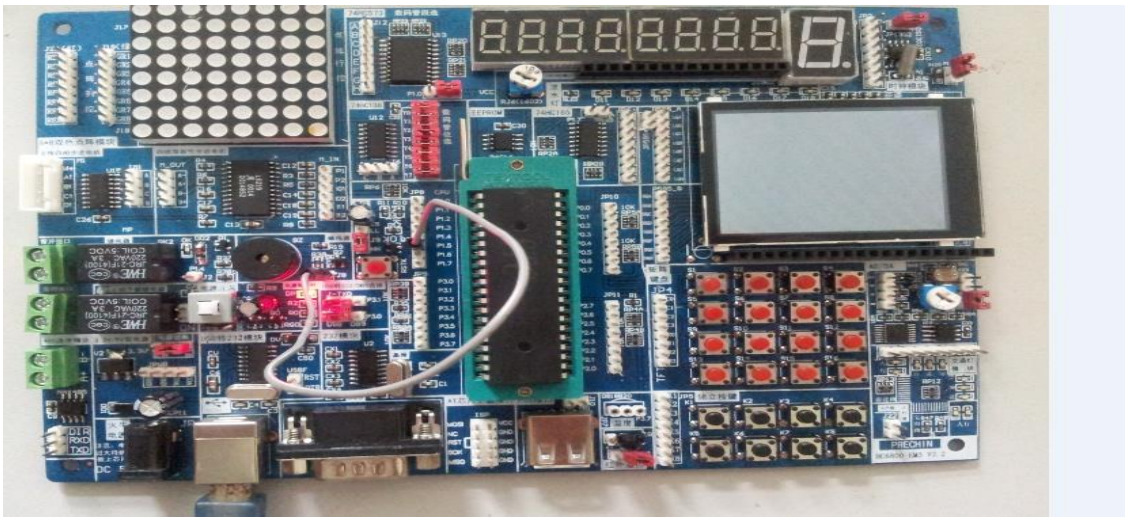
Theory:

Sound is generated by the shock and vibration of a certain frequency on a certain frequency of sound. The experiment was trumpet, issued A long one tick short of the alarm sounds, out of the port is p3.3 output 1khz, 2khz frequency alarm signal, each of the first exchange of seconds. Connection: 1PIN with a data line to insert one end of the CPU part of the JP53 (P3.I) Insert the other end of the P3.3 small part of the speaker's input JP16.

Program:

```
#include<REGX52.h>
#define buzzer P0_1
voidHmsDelay(unsigned char) ;
void main()
{
while(1)
{
buzzer = 1 ; // Make Buzzer ON
P0=0x08; // Make LED ON
HmsDelay(500) ; // 500 ms delay
buzzer = 0 ; Make Buzzer Off
P0=0x00; // Make LED Off
HmsDelay(500) ;
}
}
voidHmsDelay(unsigned char hmsDelay) { // 100 mSec @ 11.0592 MHz
unsignedint i ;
unsigned char j ;
if (hmsDelay == 0) hmsDelay = 1 ; // Non zero value
for (j=0; j<=hmsDelay; j++)
for(i=0;i<=500;i++);

}
```

Output:



Conclusion:

_____

_____

_____

_____

_____

Viva Questions:

1) How many pins are required to activate Buzzer.
2) State the Logic for the Buzzer activation.

ACHIEVED COURSEOUTCOMES:

At theend of this experiment the studentwillbeable to:-

Sr.No	CourseOutcomes	MappingwithPOs
1	Understand the concept of buzzer sequencing	**PO1, PO2, PO3, PO4, PO5, PO12**
2	Design and implement delay functionality in buzzer with LED	**PO1, PO2, PO3, PO4, PO5, PO12**
1	ProgramCourseoutcomes	**PO1, PO2, PO3, PO4, PO5, PO12**
2	StrongPOs(≥50%)	**PO1, PO2, PO3, PO4, PO5, PO12**
3	SomewhatPos	

Signature

Department of Computer Science & Engineering

Practical No. 06

Aim: Interfacing of 4 switches and 4 LEDs with 8051.

Theory: Input and output devices are the important components of embedded system, we cannot imagine any embedded device without the input and output device, switch and led are the basic example of input and output device, so it is very important to understand the interfacing of the switch and led.

LIGHT EMITTING DIODE (LED)

Led comes in various colour, its colour depends on its semiconductor material. Led have two leads one is the cathode and another one is the anode.We can easily identify the cathode and anode to see the length of leads, the length of cathode leads is lesser than the length of anode but sometimes they come in equal size.

When the length of both leads cathode and anode are equal in the size that time we can identify the anode and cathode to see their filament, the cathode has broader filament than the anode.

Connection of Led

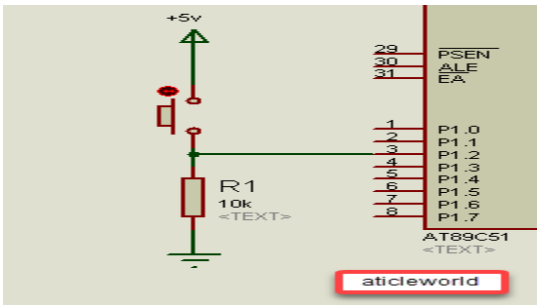
It is important to remember never connected the led directly with Vcc (output voltage which comes from directly 7805). If you connected the Led directly with the Vcc then maybe your led burnout.

So always connect the led using the resistance, if you need good brightness then you can select the value of resistance between 100 to 150-ohm either for medium brightness, you can select 300 ohms.

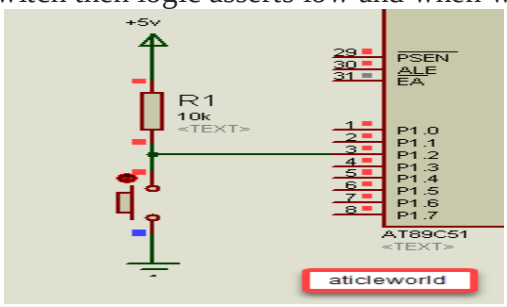
Electrical Switch

The switch is a basic input device, use to control the operation of any output device using the microcontroller or control unit. It basically breaks the electrical circuit and interrupts the flow of current.

Positive Logic: In this connection, we use a pull-down resistor connected to ground. When we pressed the switch then logic asserts high and when we disconnect the switch logic assert low.



Negative Logic: In this connection, we use a pull-up resistor connected to Vcc. When we pressed the switch then logic asserts low and when we disconnect the switch logic assert high.




```
#include<reg51.h>

sbit Led  = P2^1;    //pin connected to toggle Led
sbit Switch =P1^1;  //Pin connected to toggle led
int main()
{
   Led  = 0; //configuring as output pin
   Switch = 1; //Configuring as input pin
   while(1) //Continuous monitor the status of the switch.
   {

      if(Switch == 0)
      {
         Led =1; //Led On
      }
      else
      {
         Led =0; //Led Off
      }
   }
   return 0;
}
```

Output:

Conclusion:

viva question:

1.What are the characteristics f LED.

2.What is electrical switch.

3.What is positive logic and negative logic.

ACHIEVED COURSEOUTCOMES:

At theend of this experiment the studentwillbeable to:-

Sr. No	CourseOutcomes	MappingwithPos
1	Able to interface the programming with the hardware	**PO1, PO2, PO3, PO4, PO5, PO12**
2	Able to learn the hexadecimal code conversion.	**PO1, PO2, PO3, PO4, PO5, PO12**
3	Able to generate the series of led sequence.	**PO1, PO2, PO3, PO4, PO5, PO12**
1	ProgramCourseoutcomes	**PO1, PO2, PO3, PO4, PO5, PO12**
2	StrongPOs (≥50%)	**PO1, PO2, PO3, PO4, PO5, PO12**
3	SomewhatPos	

Signature

Department of Computer Science & Engineering

Practical No. 07

Aim: Write and execute a program for rotating of stepper motor in clockwise and anticlockwise Using Full Drive.

Software Required:

Theory:

Stepper motor is electric pulses into a line of angular displacement or displacement of the open-loop control components. In the non-overloading, the motor speed, and stop only depend on the location of the signal pulse frequency and pulse a few, and not subject to changes in the load.

Stepper motor pulse must be double-ring signal, power-driven control system components such as circuit can be used. Therefore, make good use of the stepper motor is not an easy task, which involves mechanical, electrical, electronic and computer expertise.

The main characteristics of stepper motor:

1 stepper motor can be driven to increase operation, must be driven model for the pulse, no pulse, still stepping motor, adding that if the appropriate pulse signals, it will be based on a specific point of view (known as the step angle) rotation. The rotation speed and frequency of the pulse is directly proportional. 3 step motor with an instant start and stop the rapid superior characteristics. Pulse 4 to change the order, you can easily change the direction of rotation. Embedded Development Kit package adopted by the 12v is a stepping motor, in order to demonstrate the convenience, we provide him with a 5v power supply, this time turning small moment, the reader can also be applied to him for the 12v. A stepper motor the power flow around 200ma using uln2003 drive, the drive for the port p1.0, p1.1, p1.2, p1.3. As the uln2003 it is a reverse, in practice we have In front of him, we designed a reverse- 74ls14. He was with the final results of the phase.

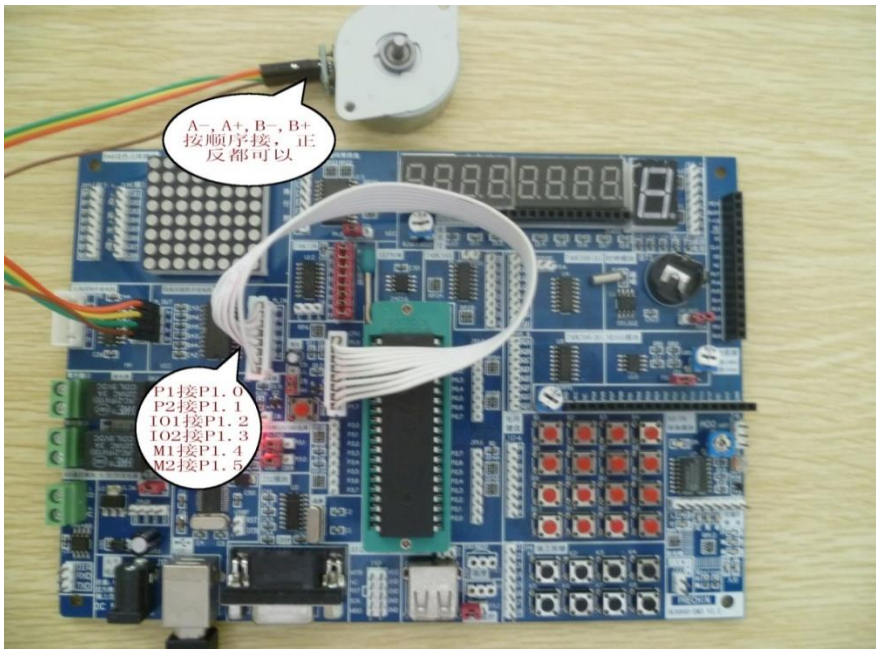
Program

Full Step

```
#include<reg52.h>

#include<stdio.h>

void delay(int);

void main()

{

do
```

```
  {
    P2 = 0x03; //0011
delay(1000);
    P2 = 0x06; //0110
delay(1000);
    P2 = 0x0C; //1100
delay(1000);
    P2 = 0x09; //1001
delay(1000);
  }
while(1);
}
void delay(int k)
{
inti,j;
for(i=0;i<k;i++)
  {
for(j=0;j<100;j++)
    {;}
  }
}
```

Output:



Conclusion:

_____

_____

Viva Questions:

1) How can the No of Steps per revolution of Stepper motor can be increased.

2) What happen if the direction of the current at the terminals of a series motor is reversed?

ACHIEVED COURSEOUTCOMES:

At theend of this experiment the studentwillbeable to:-

Sr.No	CourseOutcomes	MappingwithPOs
1	Understand the concept of Stepper Motor	**PO1, PO2, PO3, PO4, PO5, PO12**
2	Design and implement Stepper Motor	**PO1, PO2, PO3, PO4, PO5, PO12**
1	ProgramCourseoutcomes	**PO1, PO2, PO3, PO4, PO5, PO12**
2	StrongPOs(≥50%)	**PO1, PO2, PO3, PO4, PO5, PO12**
3	SomewhatPos	

Signature

Department of Computer Science & Engineering

Practical No. 8

Aim: To Study Vx Works Operating System

Theory:

VxWorks is a real-time operating system (RTOS) developed as proprietary software by Wind River Systems of Alameda, California, US. First released in 1987, VxWorks is designed for use in embedded systems requiring real-time, deterministic performance and, in many cases, safety and security certification, for industries, such as aerospace and defense, medical devices, industrial equipment, robotics, energy, transportation, network infrastructure, automotive, and consumer electronics.

VxWorks supports Intel (x86, including the new Intel Quark SoC, and x86-64), MIPS, PowerPC, SH-4, and ARM architectures. The RTOS can be used in multicore asymmetric multiprocessing (AMP), symmetric multiprocessing (SMP), and mixed modes and multi-OS (via Type 1 hypervisor)[5] designs on 32- and 64-bit processors.

VxWorks comes with the kernel, middle ware, board support packages, Wind River Workbench development suite and complementary third-party software and hardware technologies. In its latest release, VxWorks 7, the RTOS has been re-engineered for modularity and upgrade-ability so the OS kernel is separate from middle ware, applications and other packages. Scalability, security, safety, connectivity, and graphics have been improved to address Internet of Things (IoT) needs.

Platform Overview:

VxWorks supports Intel (x86, including the new Intel Quark SoC, and x86-64), MIPS, PowerPC, SH-4, and ARM architectures. The RTOS can be used in multicore asymmetric multiprocessing (AMP), symmetric multiprocessing (SMP), and mixed modes and multi-OS (via Type 1 hypervisor) designs on 32- and 64-bit processors.

The VxWorks Core Platform consists of a set of runtime components and development tools. The run time components are an operating system (UP and SMP; 32- and 64-bit), software for applications support (file system, core network stack, USB stack and inter-process communications) and hardware support (architecture adaptor, processor support library, device

driver library and board support packages). VxWorks core development tools are compilers such as Diab, GNU, and Intel C++ Compiler (ICC)) and its build and config tools. The system also includes productivity tools such as its Workbench development suite and Intel tools and development support tools for asset tracking and host support.

The platform is a modular, vendor-neutral, open system that supports a range of third-party software and hardware. The OS kernel is separate from middleware, applications and other packages, which enables easier bug fixes and testing of new features. An implementation of a layered source build system allows multiple versions of any stack to be installed at the same time so developers can select which version of any feature set should go into the VxWorks kernel libraries.

Optional Profiles for VxWorks add incremental functionality required for specific industries (such as medical, industrial, networking and consumer) or technology-related capabilities, such as a small footprint RTOS (Microkernel Profile) and a Type 1 real-time embedded hypervisor (Virtualization Profile).

Features:

Vx Works is designed for use in embedded systems.

A list of some of the features of the OS is:

- Multitasking kernel with preemptive and round-robin scheduling and fast interrupt response
- Native 64-bit operating system (only one 64-bit architecture supported: x86-64). Data model: LP64.
- User-mode applications ("Real-Time Processes", or RTP) isolated from other user-mode applications as well as the kernel via memory protection mechanisms.
- SMP, AMP and mixed mode multiprocessing support
- Error handling framework
- Bluetooth, USB, CAN protocols, Firewire IEEE 1394, BLE, L2CAP, Continua stack, health device profile
- Binary, counting, and mutual exclusion semaphores with priority inheritance
- Local and distributed message queues
- POSIX PSE52 certified conformity in user-mode execution environment

- File systems: High Reliability File System (HRFS), FAT-based file system (DOSFS), Network File System (NFS), and TFFS
- Dual-mode IPv6 networking stack with IPv6 Ready Logo certification
- Memory protection including real-time processes (RTPs), error detection and reporting, and IPC
- Multi-OS messaging using TIPC and Wind River multi-OS IPC
- Symbolic debugging

Conclusion:

_____

_____

_____

_____

Viva Questions:

1) What is Soft Real System?

2) Which scheduling algorithms are preferable used for hard real time system?

3) Is Ostrich Methodology used for Deadlock Prevention in Hard Real Time System?

ACHIEVED COURSEOUTCOMES:

At theend of this experiment the studentwillbeable to:-

Sr. No	CourseOutcomes	MappingwithPos
1	Understand the concept of Hard Real Time System	**PO1, PO2, PO3, PO4, PO5, PO12**
2	Understand the concepts of the operating system like threads, cores, semaphores, etc	**PO1, PO2, PO3, PO4, PO5, PO12**
1	ProgramCourseoutcomes	**PO1, PO2, PO3, PO4, PO5,**
2	StrongPOs(≥50%)	**PO1, PO2, PO3, PO4, PO5,**
3	SomewhatPos	

Signature

Department of Computer Science & Engineering