MICROCONTROLLER and EMBEDDED SYSTEM DESIGNS
(EE-320)

LAB MANUAL

VI SEMESTER

RAO PAHALD SINGH GROUP OF INSTITUTIONS
BALANA(MOHINDER GARH)123029

Department Of Electronics and Communication Engg.
RPS CET, Balana(M/Garh)

LIST OF EXPERIMENTS

EXPERIMENT 1

AIM: To study development tools/environment for ATMEL/PIC microcontroller programs and architecture.

APPARATUS REQUIRED: µVision, Keil, ICPROG, AT89C52 Microcontroller, PIC16F877A Microcontroller, Proteus 7.8.

THEORITICAL CONCEPT:

Software environment and microcontroller description:

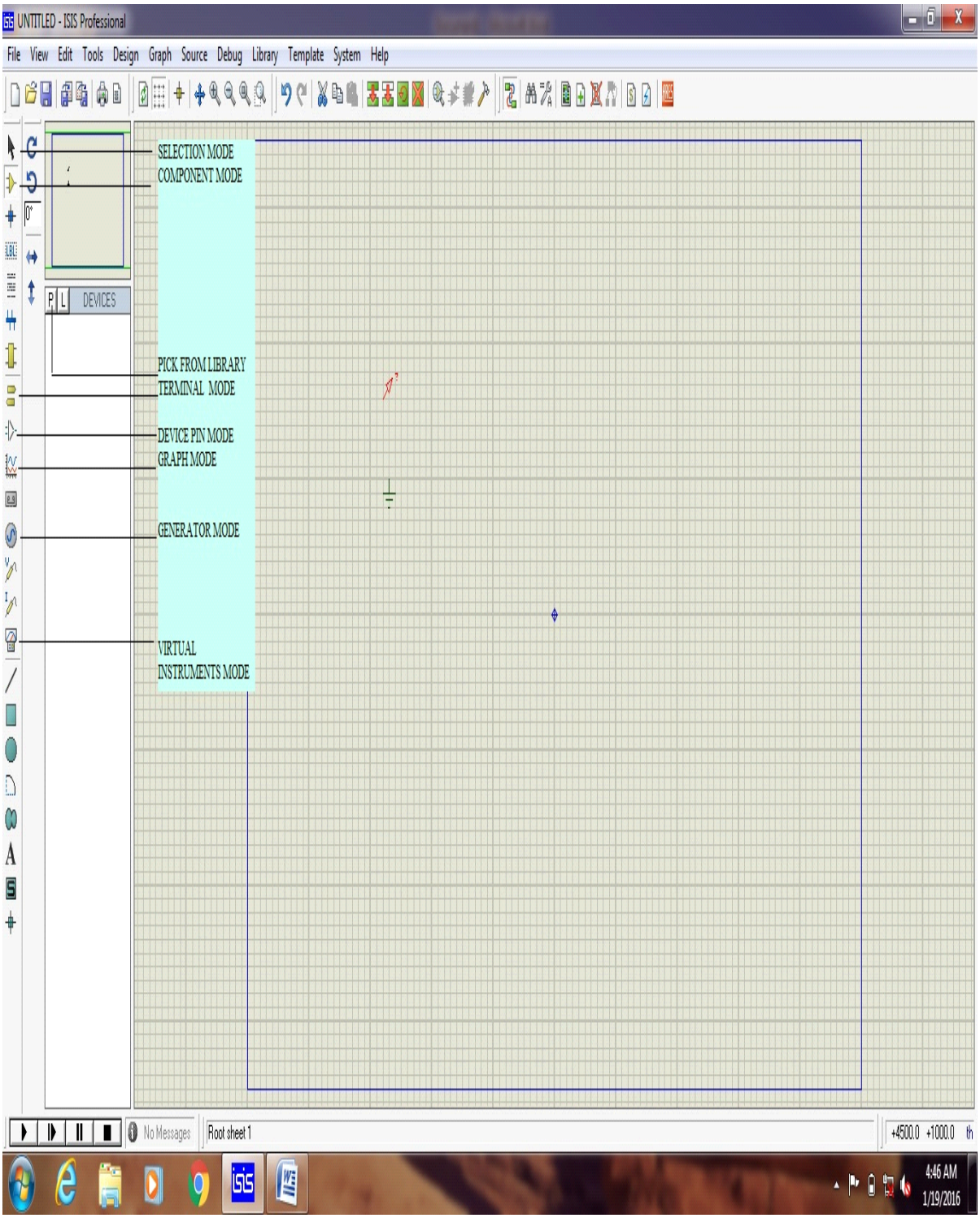
Procedure to write the program in µ Vision Keil:

- Create a **New folder** on the desktop for saving the contents of the program.
- Double click on the icon of **Keil**.
- Select the device for the target (**Select Atmel → Select 89C52 →Ok →No**)
- Go in the **project menu** and click on **µVision Project** after this an edit window will appear on desktop.
- Write the desired program in the editing window up to end.
- Right click on **source group** and select **remove start up** in **project workspace**.
- Go in the **file menu** and click on **save as** and save the program with the extinction **.asm** on desktop in the new folder.
- Right click on **source group** → Select add **file to group** → All file → Select file **.asm** → Select **Add.**
- Now go in the **project menu** and click on options for the target **"Target1".**
- Update the **frequency value (eg. 11.0592)** and click on **output** and enable the following.
    - ®Create Executable
    - √ or Ok –Debug info
    - Select Create Hex file
    - Select Browse info

  Now click on Ok
- Go in the **project menu** and click on **built target**.
- Go in the **project menu** and click on **Rebuild target**.

- Go in the **project menu** and click on **Run (or Ctrl +F5).**
- After this **Hex file** will be created in the **New Folder**

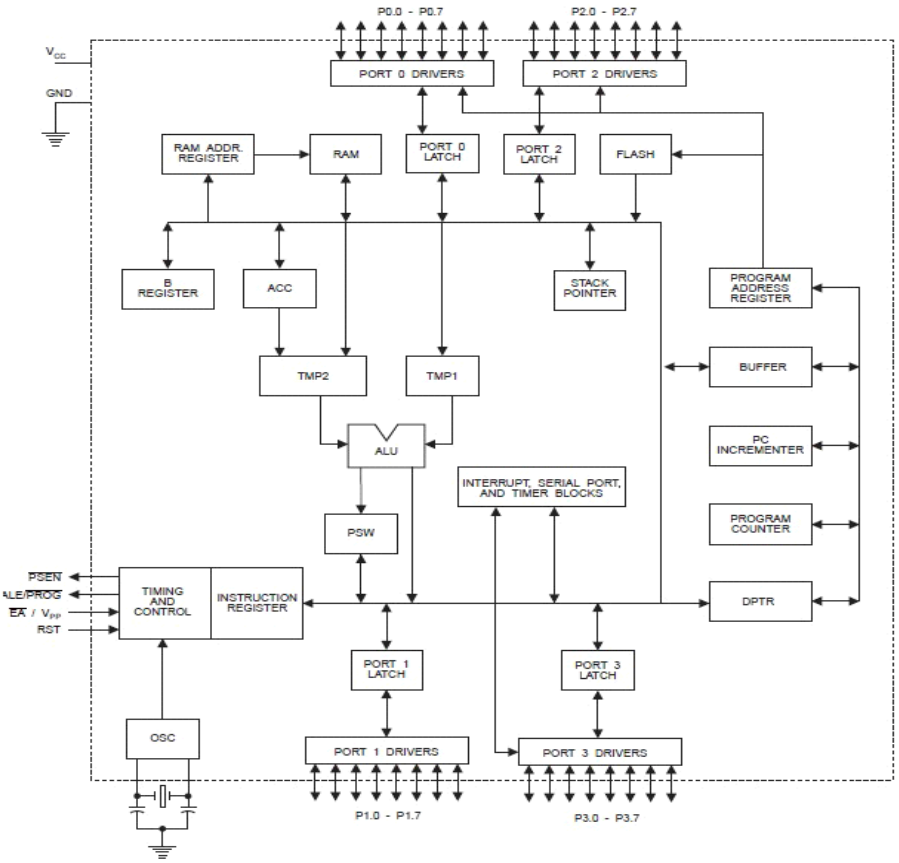
Proteus 7.8 simulation software:



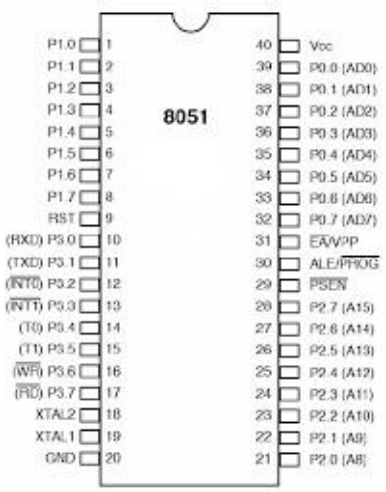
ATMEL INTRODUCTION 8051 ARCHITECTURE FAMILY

A microcontroller is a single chip microcomputer with on board program ROM and I/O that can be programmed for various control functions. Unlike a general purpose computer, which also includes all of these components, a microcontroller is designed for a very specific task to control a particular system.

The AT89C52 is a low power, high performance CMOS 8 bit microcomputer with 8K bytes of Flash Programmable and Erasable Read Only Memory. The on chip flash allows the program memory to be reprogrammed in system or by a conventional non-volatile memory programmer. The AT89C52 provides 256 Bytes of RAM, 32 I/O lines, three 16 bit timer/counters, and six vector two levels interrupt.



The architecture of the 8051 family of Microcontrollers



Pin diagram of 8051

How to use icprog software for pic microcontroller:

- Click ICPROG software.
- Select device PIC 16F877 A.
- Click <settings> Select <Hardware> Select JDM programmer and Windows API in place of Direct I/O and click OK.
- Click <Setting> then <options> then <programming> then select the option <verify after programming>.
- Set configuration in ICPROG software as mentioned below:
- Oscillator----- XT
- Write Enable----OOOO-OFFFH
- Deselect all Fuses.
- Set swl in Un pressed condition for IAP mode. Jumper in 1 2 position.
- Select <command> then Program all.

Pic introduction and architecture:

PIC is the name for the Microchip microcontroller (MCU) family, consisting of a microprocessor, I/O ports, timer(s) and other internal, integrated hardware. The main advantages of using the PIC are low external part count, a wide range of chip sizes available, nice choice of compilers (assembly, C, Basic, etc.) good wealth of example/tutorial source code and easy programming. Once bought, the PIC's program memory is empty, and needs to be programmed

with code (usually HEX files) to be usable in a circuit. For the purpose, a wide range of simple programmer hardware docs and software is downloadable from the net.

PIC is a family of Harvard architecture microcontrollers made by Microchip Technology, derived from the PIC1650 originally developed by General Instrument's Microelectronics Division.

The PIC architecture is distinctively minimalist. It is characterized by the following features:

- Separate code and data spaces (Harvard architecture)
- A small number of fixed length instructions
- Most instructions are single cycle execution (4 clock cycles), with single delay cycles upon branches and skips A single accumulator (W), the use of which (as source operand) is implied (i.e is not encoded in the opcode)
- All RAM locations function as registers as both source and/or destination of math and other functions.
- A hardware stack for storing return addresses
- A fairly small amount of addressable data space (typically 256 bytes), extended through banking
- Data space mapped CPU, port, and peripheral registers
- The program counter is also mapped into the data space and writable (this is used to synthesize indirect jumps).

Unlike most other CPUs, there is no distinction between "memory" and "register" space because the RAM serves the job of both memory and registers, and the RAM is usually just referred to as the register file or simply as the registers.

PIC16F877A Specifications and Architecture:

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program Branches, which are two-cycle
- Operating speed : DC–20 MHz clock input DC–200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pin out compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers.

Peripheral Features:

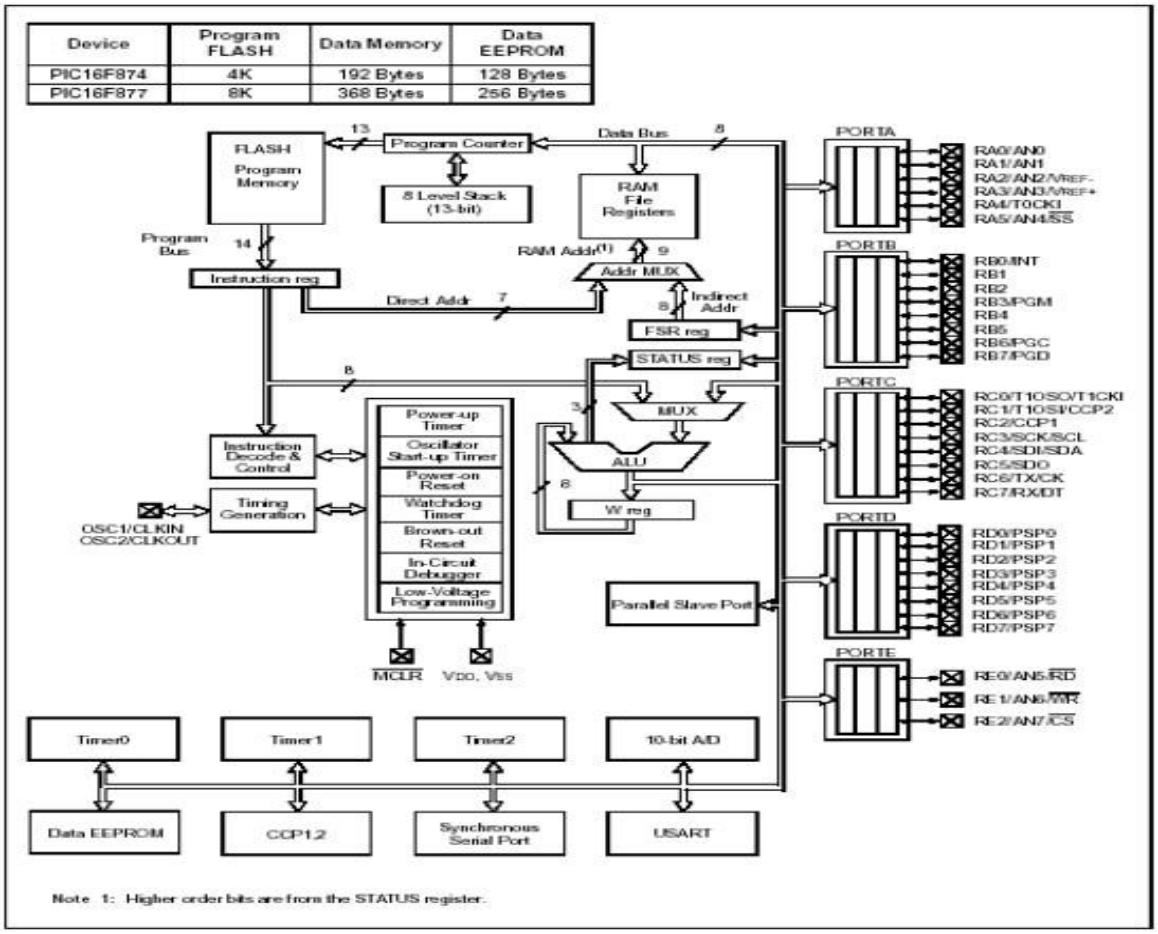
- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
    - Capture is 16-bit, max. Resolution is 12.5 ns
    - Compare is 16-bit, max. Resolution is 200 ns
    - PWM max. Resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP)–8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
    - Two analog comparators
    - Programmable on-chip voltage reference (VREF) module
    - Programmable input multiplexing from device inputs and internal voltage reference
    - Comparator outputs are externally accessible

Block Diagram of PIC 16F877A Controller

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



EXPERIMENT 2

AIM: Write an ALP to generate square of 10 Khz using Timer 0

THEORITICAL CONCEPT:

```
; we are displaying output at Port 1.1
; crystal frequency =12Mhz ;calculation for 10Mhz , 0.1ms time period
;0.05ms for half cycle ;so count will be 50
;SO TOTAL COUNT =65536-50=65486=FFCE H,TH0=FF,TL0=CE
A51 MACRO ASSEMBLER  1
MACRO ASSEMBLER A51 V8.02b  OBJECT MODULE PLACED IN 1.OBJ
ASSEMBLER INVOKED BY: C:\Keil\C51\BIN\A51.EXE 1.asm SET(SMALL) DEBUG EP
LOC  OBJ          LINE     SOURCE
0000                 2     ORG 0
0000 758A18          3     A1:MOV TL0,#18H
0003 758CFF          4     MOV TH0,#0FFH
0006 7589CE          5     MOV TMOD,#0CEH
0009 D28C            6     AGAIN:SETB TR0
000B B291            7     CPL P1.1
000D 308DF9          8     JNB TF0,AGAIN
0010 C28D            9     CLR TF0
0012 C28C            10    CLR TR0
0014 80EA            11    SJMP A1
                     12    END
A51 MACRO ASSEMBLER  1
SYMBOL TABLE LISTING------ ----- -------
N A M E          T Y P E  V A L U E   ATTRIBUTES
A1 . . . . . .  C ADDR   0000H   A
AGAIN. . . . . .  C ADDR   0009H   A
P1 . . . . . .  D ADDR   0090H   A
TF0. . . . . .  B ADDR   0088H.5 A
```

TH0. . . . . D ADDR 008CH A

TL0. . . . . D ADDR 008AH A

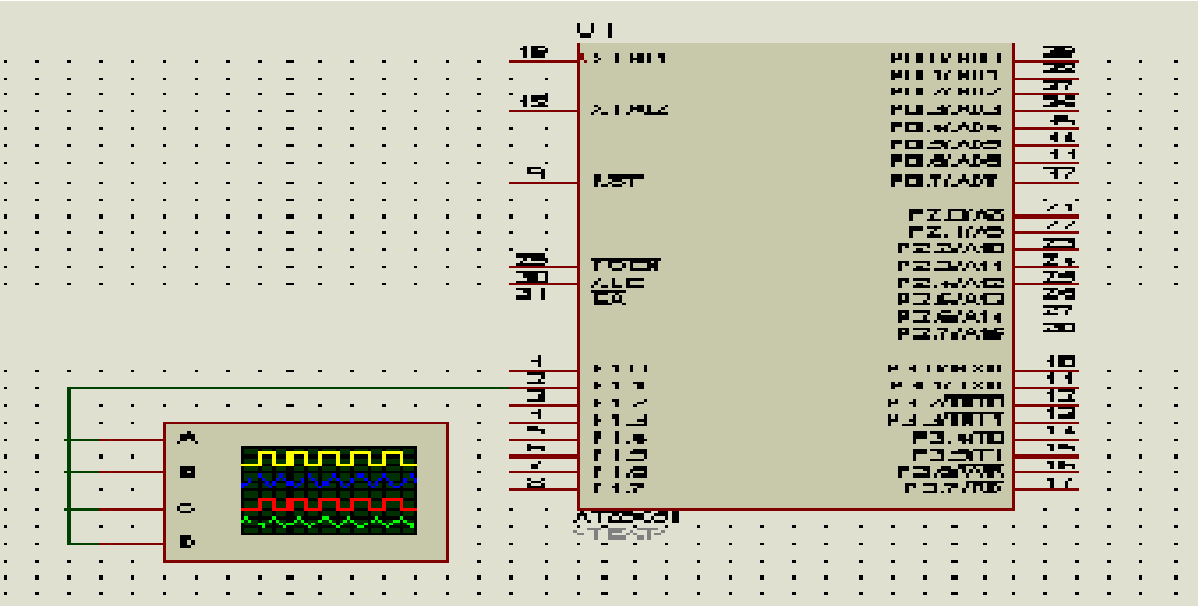
TMOD . . . . . D ADDR 0089H A

TR0. . . B ADDR 0088H.4 A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE. 0 WARNING(S), 0 ERROR(S)

EXPERIMENTAL SETUP



SPECIFICATION OF APPARATUS USED:- PC, Keil software, Proteus Software, Power Supply.

PROCEDURE:

- Create a **New folder** on the desktop for saving the contents of the program.
- Double click on the icon of **Keil**.
- Select the device for the target (**Select Atmel → Select 89C52 →Ok →No**)
- Go in the **project menu** and click on **µVision Project** after this an edit window will appear on desktop.
- Write the desired program in the editing window up to end.
- Right click on **source group** and select **remove start up** in **project workspace**.

- Go in the **file menu** and click on **save as** and save the program with the extinction **.asm** on desktop in the new folder.
- Right click on **source group** → Select add **file to group** → All file → Select file **.asm** → Select **Add.**
- Now go in the **project menu** and click on options for the target **"Target1".**
- Update the **frequency value (eg. 11.0592)** and click on **output** and enable the following.
    - ®Create Executable
    - √ or Ok –Debug info
    - Select Create Hex file
    - Select Browse info
- Now click on Ok
- Go in the **project menu** and click on **built target**.
- Go in the **project menu** and click on **Rebuild target**.
- Go in the **project menu** and click on **Run (or Ctrl +F5).**
- After this **Hex file** will be created in the **New Folder**

START PROTEUS SIMULATION:

- Place your component from the library , connect them accordingly.
- Click the "pick from library(P)" button .
- select item from the list (**AT89C51,Square generator**).
- Click ok .
- After selecting component ,click anywhere in the design area to select it and then click again to place it.
- Place all the required components.
- connect the desired node by clicking at starting and ending point.
- Double click the 8051 component to open its properties.
- Browse for the HEX file.
- The controls at the left-bottom corner will help you simulate the circuit in real time.

PRECAUTIONS:

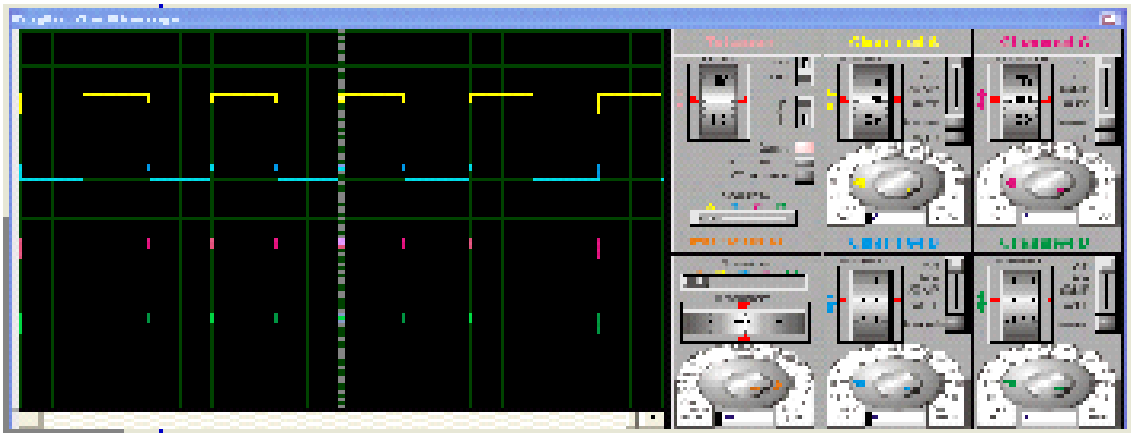
Make sure correct power supply is given to the kit/Equipment. Wrong power supplies may cause damage to your equipments.

RESULT and COMMENT:

This program can be used to generate 10Khz square wave and to use this wave externally on any device requiring square wave.

HEX FIILE:

:10000000758A18758CFF7589CED28CB291308DF9B6 :06001000C28DC28C80EAE3
:00000001FF



EXPERIMENT 3

AIM: To display a string on LCD

THEORITICAL CONCEPT:

Crystal frequency=12Mhz

D0-D7 pins of LCD connected to Port 1 of 89c51

RS= P2.0

R/W=P2.1

E=P2.2

*TEXT TO DISPLAY: "***RPS WELCOMES YOU***"*

A51 MACRO ASSEMBLER 1

MACRO ASSEMBLER A51 V8.02b

OBJECT MODULE PLACED IN 2.OBJ

ASSEMBLER INVOKED BY: C:\Keil\C51\BIN\A51.EXE 2.asm SET(SMALL) DEBUG EP

```
LOC  OBJ          LINE     SOURCE
0000                 1     ORG 0
0000 7438            2     A1:MOV A,#38H
0002 1128            3     ACALL CMNDWRT
0004 1142            4     ACALL DELAY
0006 7480            5     MOV A,#080H
0008 1128            6     ACALL CMNDWRT
000A 1142            7     ACALL DELAY
000C 740C            8     MOV A,#0CH
000E 1128            9     ACALL CMNDWRT
0010 1142           10     ACALL DELAY
0012 7401           12     MOV A,#01H
0014 1128           13     ACALL CMNDWRT
0016 1142           14     ACALL DELAY
0018 781E           15     MOV R0,#30
001A 900300         16     MOV DPTR, #300H
```

```
001D E4          17    AGAIN:CLR A
001E 93          18    MOVC A,@A+DPTR
001F 1135        19    ACALL DATAWRT
0021 1142        20    ACALL DELAY
0023 A3          21    INC DPTR
0024 D8F7        22    DJNZ R0,AGAIN
0026 80D8        23    SJMP A1
0028 F590        24    CMNDWRT:MOV P1,A
002A C2A0        25    CLR P2.0
002C C2A1        26    CLR P2.1
002E D2A2        27    SETB P2.2
0030 1142        28    ACALL DELAY
0032 C2A2        29    CLR P2.2
0034 22          30    RET
0035 F590        31    DATAWRT:MOV P1,A
0037 D2A0        32    SETB P2.0
0039 C2A1        33    CLR P2.1
003B D2A2        34    SETB P2.2
003D 1142        35    ACALL DELAY
003F C2A2        36    CLR P2.2
0041 22          37    RET
0042 79FF        38    DELAY: MOV R1,#255
0044 7A64        39    RE1:MOV R2,#100
0046 DAFE        40    RE2:DJNZ R2,RE2
0048 D9FA        41    DJNZ R1,RE1
004A 22          42    RET
0300             43    ORG 300H
0300 52505320    44    COUNT: DB 'RPS WELCOMES YOU'
0304 57454C43
0308 4F4D4553
```

030C 20594F55 45 END---- -------

N A M E T Y P E V A L U E ATTRIBUTES

A1 . . . . . C ADDR 0000H A

AGAIN. . . . . C ADDR 001DH A

CMNDWRT. . . . . C ADDR 0028H A

COUNT. . . . . C ADDR 0300H A

DATAWRT. . . . . C ADDR 0035H A

DELAY. . . . . C ADDR 0042H A

P1 . . . . . D ADDR 0090H A

P2 . . . . . D ADDR 00A0H A

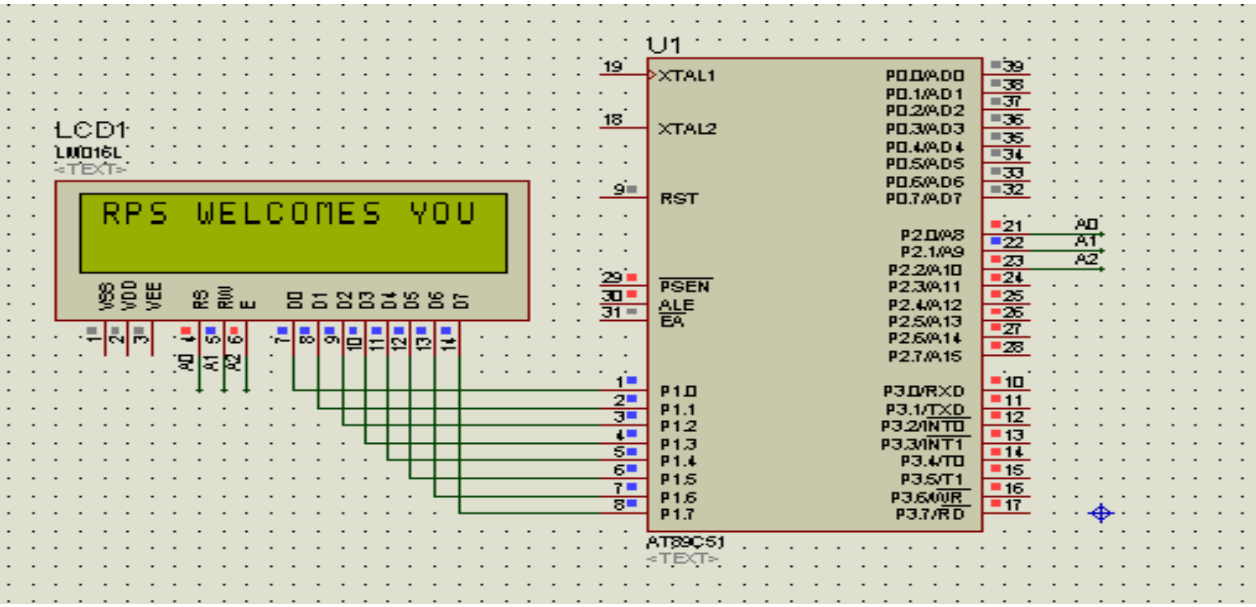
RE1. . . . . C ADDR 0044H A

RE2. . . . . C ADDR 0046H A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE. 0 WARNING(S), 0 ERROR(S)

EXPERIMENTAL SETUP:-



SPECIFICATION OF APPARATUS USED:- PC, Keil software, Proteus Software, Power Supply.

PROCEDURE:

- Create a **New folder** on the desktop for saving the contents of the program.
- Double click on the icon of **Keil**.
- Select the device for the target (**Select Atmel → Select 89C52 →Ok →No**)
- Go in the **project menu** and click on **µVision Project** after this an edit window will appear on desktop.
- Write the desired program in the editing window up to end.
- Right click on **source group** and select **remove start up** in **project workspace**.
- Go in the **file menu** and click on **save as** and save the program with the extinction **.asm** on desktop in the new folder.
- Right click on **source group** → Select add **file to group** → All file → Select file **.asm** → Select **Add.**
- Now go in the **project menu** and click on options for the target **"Target1".**
- Update the **frequency value (eg. 11.0592)** and click on **output** and enable the following.
    - ®Create Executable
    - √ or Ok –Debug info
    - Select Create Hex file
    - Select Browse info

  Now click on Ok
- Go in the **project menu** and click on **built target**.
- Go in the **project menu** and click on **Rebuild target**.
- Go in the **project menu** and click on **Run (or Ctrl +F5).**
- After this **Hex file** will be created in the **New Folder.**

START PROTEUS SIMULATION:

- Place your component from the library , connect them accordingly.
- Click the "pick from library(P)" button .
- select item from the list (**at89c51,16x2 LCD**).
- Click ok .

- After selecting component ,click anywhere in the design area to select it and then click again to place it.
- Place all the required components.
- connect the desired node by clicking at starting and ending point.
- Double click the 8051 component to open its properties.
- Browse for the HEX file.
- The controls at the left-bottom corner will help you simulate the circuit in real time.

PRECAUTIONS:

Make sure correct power supply is given to the kit/Equipment. Wrong power supplies may cause damage to your equipments

RESULT and COMMENT:

This program can be used for displaying any data on lcd .

HEX FILE:

:**1000000074381128114274801128114274OC11287F:10001000114274011128114278 1E900300E49311DB**

:10002000351142A3D8F780D8F590C2A0C2A1D2A2C0:100030001142C2A222F590D2A0C2A1D2A21142C204

:0B004000A22279FF7A64DAFED9FA22CE:1003000052505320574 54C434F4D455320594F555C:00000001FF

EXPERIMENT 4

AIM: An ALP to interface seven segment with 8051 and display 0-9 on it

THEORITICAL CONCEPT:

Crystal Frequency =12Mhz

7 segment used = Common Cathode Type

A51 MACRO ASSEMBLER 1

MACRO ASSEMBLER A51 V8.02b

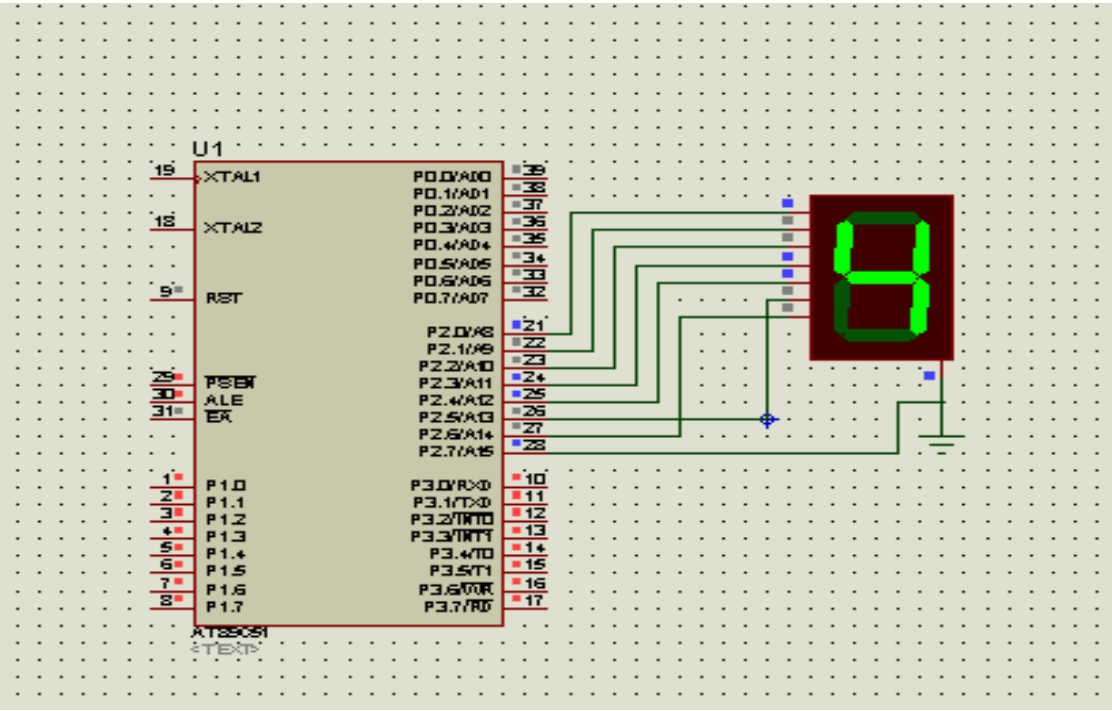
OBJECT MODULE PLACED IN 3.OBJ

ASSEMBLER INVOKED BY: C:\Keil\C51\BIN\A51.EXE 3.asm SET(SMALL) DEBUG EP

```
LOC  OBJ          LINE     SOURCE
0000                 1     ORG 0
0000 780A            2     A2:MOV R0,#10
0002 900300          3     MOV DPTR,#COUNT
0005 E4              4     AGAIN:CLR A
0006 93              5     MOVC A,@A+DPTR
0007 F5A0            6     MOV P2,A
0009 1110            7     ACALL DELAY
000B A3              8     INC DPTR
000C D8F7            9     DJNZ R0,AGAIN
000E 80F0           10     SJMP A2
0010 7B05           11     DELAY:MOV R3,#5
0012 7AFF           12     H1:MOV R2,#255
0014 79FF           13     H2:MOV R1,#255
0016 D9FE           14     H3:DJNZ R1,H3
0018 DAFA           15     DJNZ R2,H2
001A DBF6           16     DJNZ R3,H1
001C 22             17     RET
0300                18     ORG 300H
0300 3F065B4F       19     COUNT: DB 3FH,06H,5BH,4FH,66H,6DH,7CH,07H,7FH,67H
0304 666D7C07
```

```
0308 7F67

                20     END

A51 MACRO ASSEMBLER

SYMBOL TABLE LISTING------ ----- -------

N A M E          T Y P E  V A L U E   ATTRIBUTES

A2 . . . . . .  C ADDR   0000H   A

AGAIN. . . . . .  C ADDR   0005H   A

COUNT. . . . . .  C ADDR   0300H   A

DELAY. . . . . .  C ADDR   0010H   A

H1 . . . . . .  C ADDR   0012H   A

H2 . . . . . .  C ADDR   0014H   A

H3 . . . . . .  C ADDR   0016H   A

P2 . . . . . .  D ADDR   00A0H   A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE.  0 WARNING(S), 0 ERROR(S)
```

EXPERIMENTAL SETUP:-



SPECIFICATION OF APPARATUS USED:- PC, Keil software, Proteus Software, Power Supply.

START PROTEUS SIMULATION:

- Place your component from the library , connect them accordingly.
- Click the "pick from library(P)" button .
- select item from the list (**at89c51,Seven segment display**).
- Click ok .
- After selecting component ,click anywhere in the design area to select it and then click again to place it.
- Place all the required components.
- connect the desired node by clicking at starting and ending point.
- Double click the 8051 component to open its properties.
- Browse for the HEX file.
- The controls at the left-bottom corner will help you simulate the circuit in real time.

PRECAUTIONS:

Make sure correct power supply is given to the kit/Equipment. Wrong power supplies may cause damage to your equipments

RESULT and COMMENT:

This program can be used to interface seven segment display with 8051.

Hex File:

:10000000780A900300E493F5A01110A3D8F780F0CC:0D0010007B057AFF79FFD9FEDA FADBF622D4: 0A0300003F065B4F666D7C077F67C8 :00000001FF

EXPERIMENT 5

AIM: An ALP to interface DC Motor with 8051

THEORITICAL CONCEPT:

Crystal Frequency=12MHz

Symbols Used: ***BFWD=BOTH FWD, BREV= BOTH REVERSE, MOT1= MOTOR 1 ONLY, MOT2= MOTOR2 ONLY***

A51 MACRO ASSEMBLER

MACRO ASSEMBLER A51 V8.02b

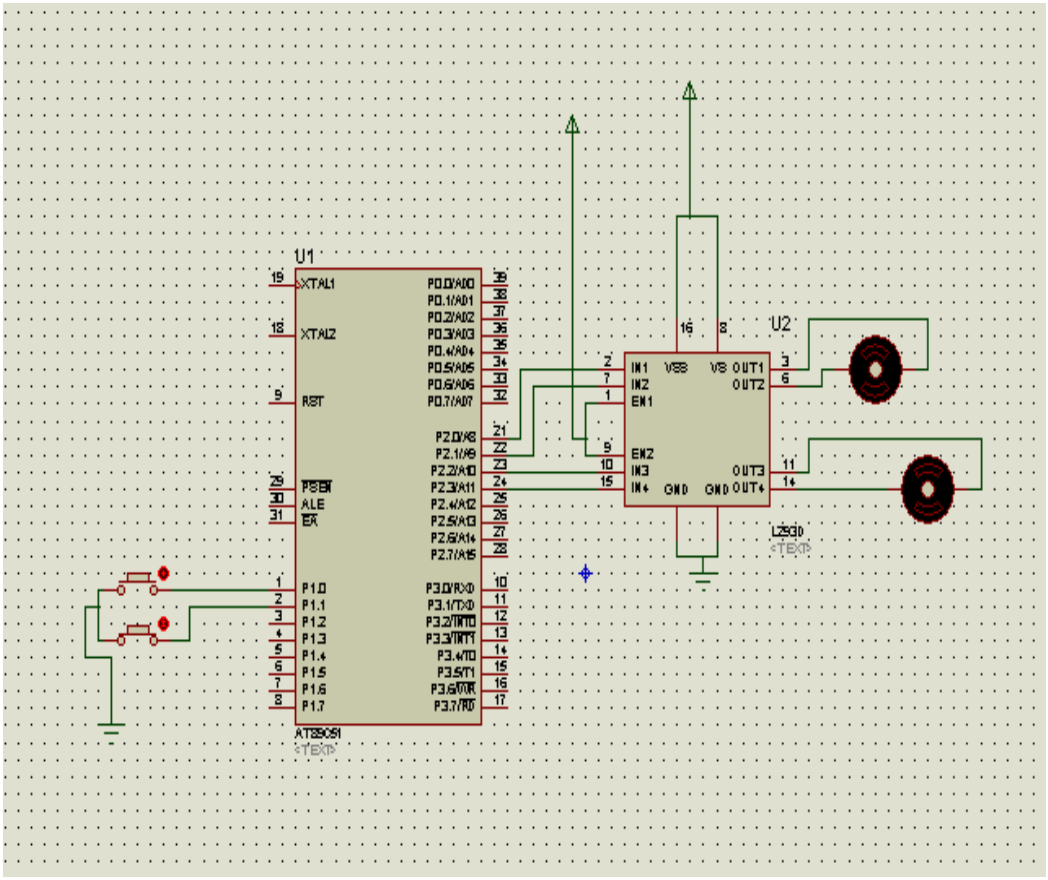
OBJECT MODULE PLACED IN 4.OBJ

ASSEMBLER INVOKED BY: C:\Keil\C51\BIN\A51.EXE 4.asm SET(SMALL) DEBUG EP

```
LOC  OBJ          LINE     SOURCE

0000                 1     ORG 0

0000 D290            2     SETB P1.0

0002 D291            3     SETB P1.1

0004 309017          4     AGAIN:JNB P1.0,MOT2

0007 30910A          5     JNB P1.1,MOT1

000A D2A0            6     BREV:SETB P2.0

000C C2A1            7     CLR P2.1

000E D2A2            8     SETB P2.2

0010 C2A3            9     CLR P2.3

0012 80F0           10     SJMP AGAIN

0014 C2A0           11     MOT1: CLR P2.0

0016 D2A1           12     SETB P2.1

0018 C2A2           13     CLR P2.2

001A C2A3           14     CLR P2.3

001C 80E6           15     SJMP AGAIN

001E 30910A         16     MOT2: JNB P1.1,BFWD

0021 C2A0           17     CLR P2.0
```

```
0023 C2A1          18     CLR P2.1
0025 C2A2          19     CLR P2.2
0027 D2A3          20     SETB P2.3
0029 80D9          21     SJMP AGAIN
002B C2A0          22     BFWD: CLR P2.0
002D D2A1          23     SETB P2.1
002F C2A2          24     CLR P2.2
0031 D2A3          25     SETB P2.3
0033 80CF          26     SJMP AGAIN
                   27     END
ASSEMBLY COMPLETE.  0 WARNING(S), 0 ERRORS
```

EXPERIMENTAL SETUP:



SPECIFICATION OF APPARATUS USED:- PC, Keil software, Proteus Software, Power Supply.

START PROTEUS SIMULATION:

- Place your component from the library , connect them accordingly.
- Click the "pick from library(P)" button .
- select item from the list (**at89c51,IC L293D,DC motor**).
- Click ok .
- After selecting component ,click anywhere in the design area to select it and then click again to place it.
- Place all the required components.
- connect the desired node by clicking at starting and ending point.
- Double click the 8051 component to open its properties.
- Browse for the HEX file.
- The controls at the left-bottom corner will help you simulate the circuit in real time.

PRECAUTIONS:

Make sure correct power supply is given to the kit/Equipment. Wrong power supplies may cause damage to your equipments

RESULT AND COMMENTS:

EXPERIMENT 6

AIM: Write an ALP to transmit the data using P1 of 8051.

THEORITICAL CONCEPT:

```
; we are transmitting data at Port 1 using the max232 .
; crystal frequency =12Mhz ;
A51 MACRO ASSEMBLER  SERIAL
MACRO ASSEMBLER A51 V8.02b
OBJECT MODULE PLACED IN serial.OBJ
ASSEMBLER INVOKED BY: C:\Keil\C51\BIN\A51.EXE serial.asm SET(SMALL) DEBUG EP
LOC  OBJ          LINE     SOURCE
0000                1     ORG 00
0000 758920         2     MOV TMOD,#020H
0003 758DFD         3      MOV TH1,#-3
0006 759850         4     MOV SCON,#050H
0009 D28E           5     SETB TR1
000B 759959         6      AGAIN: MOV SBUF,#'Y'
000E 3099FD         7      HERE:JNB TI,HERE
0011 C299           8     CLR TI
0013 75804E         9      MOV P0,#'N'
0016 80F3          10     SJMP AGAIN
0018 7B0A          11      DELAY: MOV R3,#10
001A 7CFF          12                HERE1:   MOV R4,#255
001C DCFE          13                 HERE2:  DJNZ R4,HERE2
001E DBFA          14                      DJNZ R3,HERE1
                   15     END
A51 MACRO ASSEMBLER  SERIAL
SYMBOL TABLE LISTING
N A M E           T Y P E  V A L U E   ATTRIBUTES
AGAIN. . . . . .  C ADDR   000BH   A
DELAY. . . . . .  C ADDR   0018H   A
```

HERE . . . . . C ADDR 000EH A

HERE1. . . . . C ADDR 001AH A

HERE2. . . . . C ADDR 001CH A

P0 . . . . . D ADDR 0080H A

SBUF . . . . . D ADDR 0099H A

SCON . . . . . D ADDR 0098H A

TH1. . . . . D ADDR 008DH A

TI . . . . . B ADDR 0098H.1 A

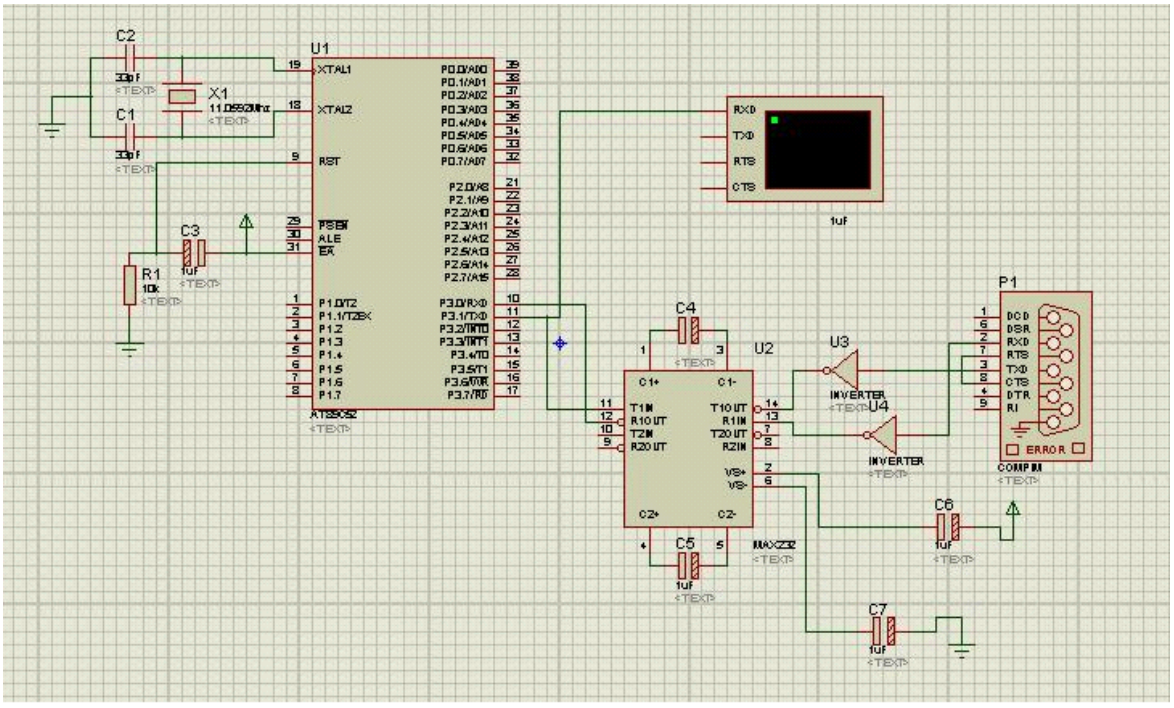
TMOD . . . . . D ADDR 0089H A

TR1. . . . . B ADDR 0088H.6 A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE. 0 WARNING(S), 0 ERROR(S)

EXPERIMENTAL SETUP:



START PROTEUS SIMULATION:

- Place your component from the library , connect them accordingly.
- Click the "pick from library(P)" button .

- select item from the list (**AT89C51,MAX232**).
- Click ok .
- After selecting component ,click anywhere in the design area to select it and then click again to place it.
- Place all the required components.
- connect the desired node by clicking at starting and ending point.
- Double click the 8051 component to open its properties.
- Browse for the HEX file.
- The controls at the left-bottom corner will help you simulate the circuit in real time.

PRECAUTIONS:

Make sure correct power supply is given to the kit/Equipment. Wrong power supplies may cause damage to your equipments

RESULT and COMMENT:

HEX FIILE:

:10000000758920758DFD759850D28E7599593099E6:10001000FDC29975804E80F37B0A7C
FFDCFEDBFA23

:00000001FF

EXPERIMENT 7

AIM: Write an ALP to interface 4x4 keyboard with 8051.

THEORITICAL CONCEPT:

```
; we are connecting rows of 4x4 keyboard  at Port 1(P1.0 to P1.3) and columns at Port 2(P2.0 to P2.3).
; crystal frequency =12Mhz ;
A51 MACRO ASSEMBLER  KeY
MACRO ASSEMBLER A51 V8.02b
OBJECT MODULE PLACED IN kay.OBJ
ASSEMBLER INVOKED BY: C:\Keil\C51\BIN\A51.EXE kay.asm SET(SMALL) DEBUG EP
LOC  OBJ          LINE     SOURCE
0000              1     ORG 00
0000 75A0FF       2     MOV P2,#0FFH
0003 759000       3     K1: MOV P1,#0
0006 E5A0         4          MOV A,P2
0008 540F         5          ANL A,#00FH
000A B40FF6       6          CJNE A,#00FH,K1
000D 116D         7     K2: ACALL DELAY
000F E5A0         8          MOV A,P2
0011 540F         9          ANL A,#00FH
0013 B40F02       10         CJNE A,#00FH,OVER
0016 80F5         11         SJMP K2
0018 116D         12    OVER: ACALL DELAY
001A E5A0         13          MOV A,P2
001C 540F         14          ANL A,#0FH
001E B40F02       15          CJNE A,#0FH,OVER1
0021 80EA         16          SJMP K2
0023 7590FE       17    OVER1: MOV P1,#0FEH
0026 E5A0         18               MOV A,P2
0028 540F         19               ANL A,#0FH
002A B40F21       20               CJNE A,#0FH,ROW_0
```

```
002D 7590FD          21                MOV P1,#0FDH
0030 E5A0            22                MOV A,P2
0032 540F            23                ANL A,#00FH
0034 B50F1C          24                CJNE A,00FH,ROW_1
0037 7590FC          25                MOV P1,#0FCH
003A E5A0            26                MOV A,P2
003C 540F            27                ANL A,#00FH
003E B50F17          28                CJNE A,00FH,ROW_2
0041 7590F7          29                MOV P1,#0F7H
0044 E5A0            30                MOV A,P2
0046 540F            31                ANL A,#00FH
0048 B40F12          32                CJNE A,#00FH,ROW_3
004B 02000D          33                LJMP K2
004E 900300          34     ROW_0: MOV DPTR,#KCODE0
0051 800D            35                SJMP FIND
0053 900304          36     ROW_1: MOV DPTR,#KCODE1
0056 8008            37                SJMP FIND
0058 900308          38     ROW_2: MOV DPTR,#KCODE2
005B 8003            39                SJMP FIND
005D 90030C          40     ROW_3: MOV DPTR,#KCODE3
                     41
0060 13              42     FIND: RRC A
0061 5003            43           JNC MATCH
0063 A3              44           INC DPTR
0064 80FA            45           SJMP FIND
0066 E4              46     MATCH: CLR A
0067 93              47                MOVC A,@A+DPTR
0068 F580            48                MOV P0,A
006A 020003          49                LJMP K1
006D 7B0A            50     DELAY: MOV R3,#10
```

```
006F 7CFF          51              HERE1:   MOV R4,#255
0071 DCFE          52              HERE:   DJNZ R4,HERE
0073 DBFA          53                      DJNZ R3,HERE1
0300               54    ORG 300H
0300 30313233      55    KCODE0: DB '0','1','2','3'
0304 34353637      56    KCODE1: DB '4','5','6','7'
0308 38394142      57    KCODE2: DB '8','9','A','B'
030C 43444546      58    KCODE3: DB 'C','D','E','F'
A51 MACRO ASSEMBLER  KeY
 59    END
A51 MACRO ASSEMBLER  KAY
SYMBOL TABLE LISTING
------ ----- -------
N A M E          T Y P E  V A L U E   ATTRIBUTES
DELAY. . . . . .  C ADDR   006DH   A
FIND . . . . . .  C ADDR   0060H   A
HERE . . . . . .  C ADDR   0071H   A
HERE1. . . . . .  C ADDR   006FH   A
K1 . . . . . .  C ADDR   0003H   A
K2 . . . . . .  C ADDR   000DH   A
KCODE0 . . . . . .  C ADDR   0300H   A
KCODE1 . . . . . .  C ADDR   0304H   A
KCODE2 . . . . . .  C ADDR   0308H   A
KCODE3 . . . . . .  C ADDR   030CH   A
MATCH. . . . . .  C ADDR   0066H   A
OVER . . . . . .  C ADDR   0018H   A
OVER1. . . . . .  C ADDR   0023H   A
P0 . . . . . .  D ADDR   0080H   A
P1 . . . . . .  D ADDR   0090H   A
P2 . . . . . .  D ADDR   00A0H   A
```

ROW_0. . . . . C ADDR 004EH A

ROW_1. . . . . C ADDR 0053H A

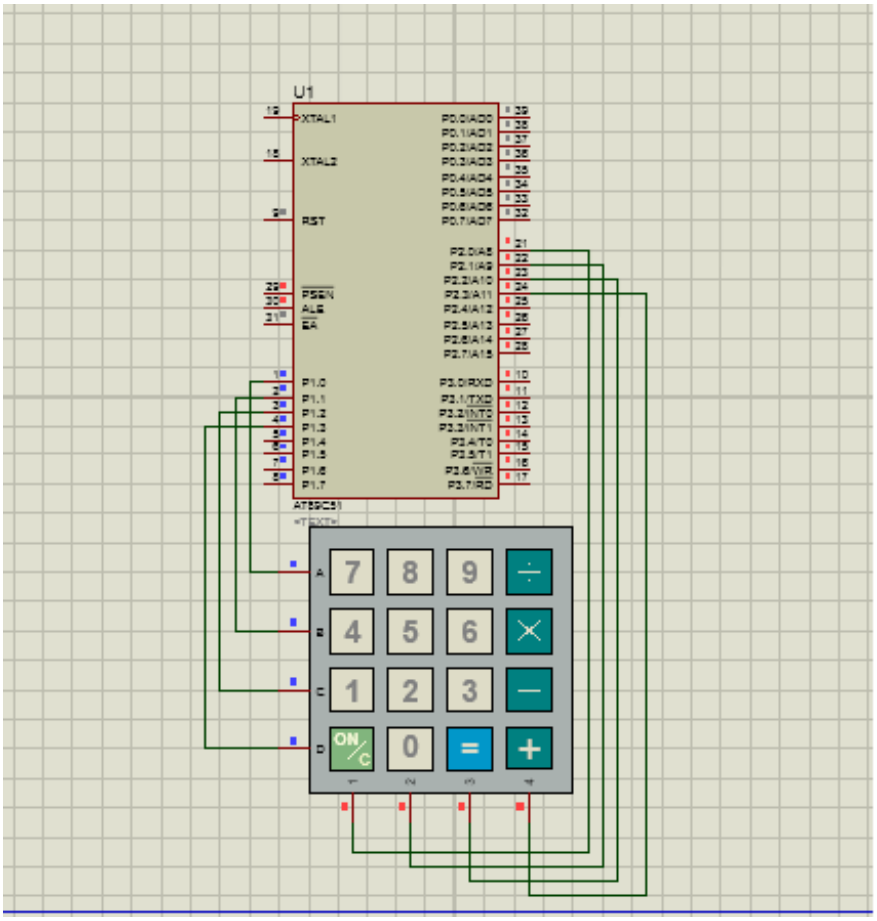
ROW_2. . . . . C ADDR 0058H A

ROW_3. . . . . C ADDR 005DH A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE. 0 WARNING(S), 0 ERROR(S)

EXPERIMENTAL SETUP:



START PROTEUS SIMULATION:

- Place your component from the library , connect them accordingly.
- Click the "pick from library(P)" button .
- select item from the list (**AT89C51 ,Keypad**).
- Click ok .

- After selecting component ,click anywhere in the design area to select it and then click again to place it.
- Place all the required components.
- connect the desired node by clicking at starting and ending point.
- Double click the 8051 component to open its properties.
- Browse for the HEX file.
- The controls at the left-bottom corner will help you simulate the circuit in real time.

PRECAUTIONS:

Make sure correct power supply is given to the kit/Equipment. Wrong power supplies may cause damage to your equipments

RESULT and COMMENTS:

HEX FIILE:

:1000000075A0FF759000E5A0540FB40FF6116DE5D3:10001000A0540FB40F0280F5116DE5
A0540FB40F7A

:100020000280EA7590FEE5A0540FB40F217590FD93:10003000E5A0540FB50F1C7590FCE
5A0540FB50F4B:10004000177590F7E5A0540FB40F1202000D90033E:1000500000800D9003
0480089003088003900 30C37:10006000135003A380FAE493F5800200037B0A7C1B:0500700
0FFDCFEDBFADD:1003000030313233343536373839414243444546 4B:00000001FF

EXPERIMENT 8

AIM: Write an ALP to interface the temperature sensor to 8051.

THEORITICAL CONCEPT:

```
; we are using ADC0848 to interface the temperature sensor.
;we use the Port 2 and Port 1 to connect the ADC0848.
; crystal frequency =12Mhz ;
A51 MACRO ASSEMBLER  TEMP
MACRO ASSEMBLER A51 V8.02b
OBJECT MODULE PLACED IN temp.OBJ
ASSEMBLER INVOKED BY: C:\Keil\C51\BIN\A51.EXE temp.asm SET(SMALL) DEBUG EP
LOC  OBJ          LINE     SOURCE
0000               1     ORG 00
 0090              2     MYDATA EQU P1
0000 7590FF        3     MOV P1,#0FFH
0003 D2A7          4     SETB P2.7
0005 C2A6          5     BACK: CLR P2.6
0007 D2A6          6                SETB P2.6
0009 20A7FD        7     HERE: JB P2.7,HERE
000C C2A5          8                CLR P2.5
000E E590          9                MOV A, MYDATA
0010 1124          10               ACALL DATA_DISPLAY
0012 D2A5          11               SETB P2.5
0014 80EF          12               SJMP BACK
0016               13     CONVERSION:
0016 75F00A        14               MOV B,#10
0019 84            15               DIV AB
001A AFF0          16               MOV R7,B
001C 75F00A        17               MOV B,#10
001F 84            18               DIV AB
0020 AEF0          19               MOV R6,B
```

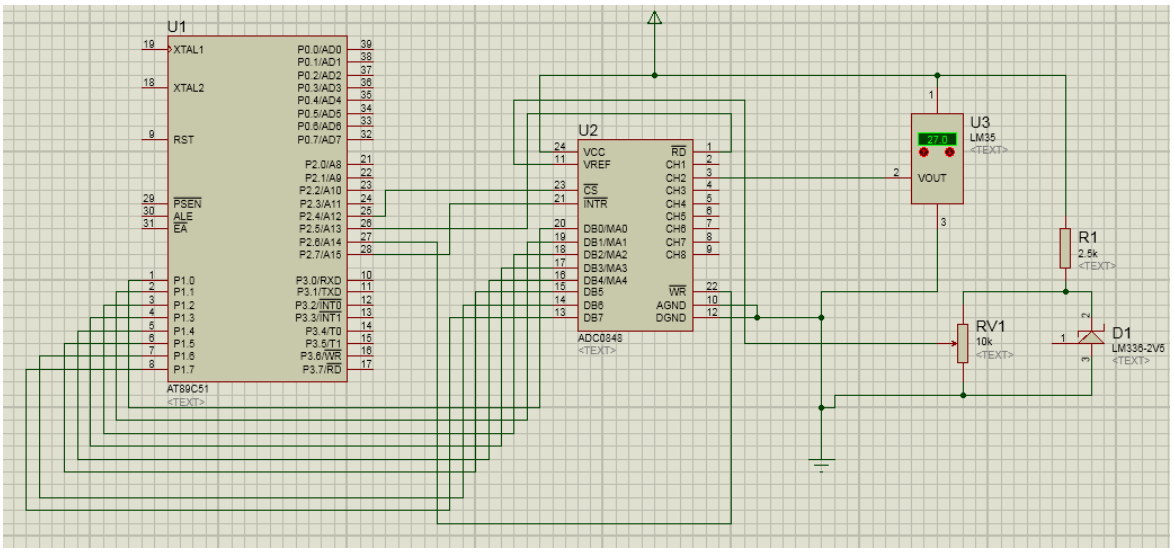
```
0022 FD            20                MOV R5,A
0023 22            21                RET
0024               22    DATA_DISPLAY:
0024 8F80          23                MOV P0,R7
0026 1131          24                ACALL DELAY
0028 8E80          25                MOV P0,R6
002A 1131          26                ACALL DELAY
002C 8D80          27                MOV P0,R5
002E 1131          28                ACALL DELAY
0030 22            29                RET
0031 7B0A          30    DELAY: MOV R3,#10
0033 7CFF          31                HERE1:   MOV R4,#255
0035 DCFE          32                HERE2:  DJNZ R4,HERE2
0037 DBFA          33                        DJNZ R3,HERE1
                   34    END
 A51 MACRO ASSEMBLER  TEMP
SYMBOL TABLE LISTING
------ ----- -------
N A M E          T Y P E  V A L U E   ATTRIBUTES
B. . . . . .  D ADDR   00F0H   A
BACK . . . . . .  C ADDR   0005H   A
CONVERSION . . . .  C ADDR   0016H   A
DATA_DISPLAY . . .  C ADDR   0024H   A
DELAY. . . . . .  C ADDR   0031H   A
HERE . . . . . .  C ADDR   0009H   A
HERE1. . . . . .  C ADDR   0033H   A
HERE2. . . . . .  C ADDR   0035H   A
MYDATA . . . . . .  D ADDR   0090H   A
P0 . . . . . .  D ADDR   0080H   A
P1 . . . . . .  D ADDR   0090H   A
```

P2 . . . . . D ADDR 00A0H A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE. 0 WARNING(S), 0 ERROR(S)

EXPERIMENTAL SETUP:



START PROTEUS SIMULATION:

- Place your component from the library , connect them accordingly.
- Click the "pick from library(P)" button .
- select item from the list (**at89c51,ADC0848**).
- Click ok .
- After selecting component ,click anywhere in the design area to select it and then click again to place it.
- Place all the required components.
- connect the desired node by clicking at starting and ending point.
- Double click the 8051 component to open its properties.
- Browse for the HEX file.
- The controls at the left-bottom corner will help you simulate the circuit in real time.

PRECAUTIONS:

Make sure correct power supply is given to the kit/Equipment. Wrong power supplies may cause damage to your equipments

RESULT and COMMENT:

HEX FIILE:

:100000007590FFD2A7C2A6D2A620A7FDC2A5E590F3:100010001124D2A580EF75F00A84AFF075F00A8440

:10002000AEF0FD228F8011318E8011318D80113123:09003000227B0A7CFFDCFEDBFAF6:00000001FF

EXPERIMENT 9

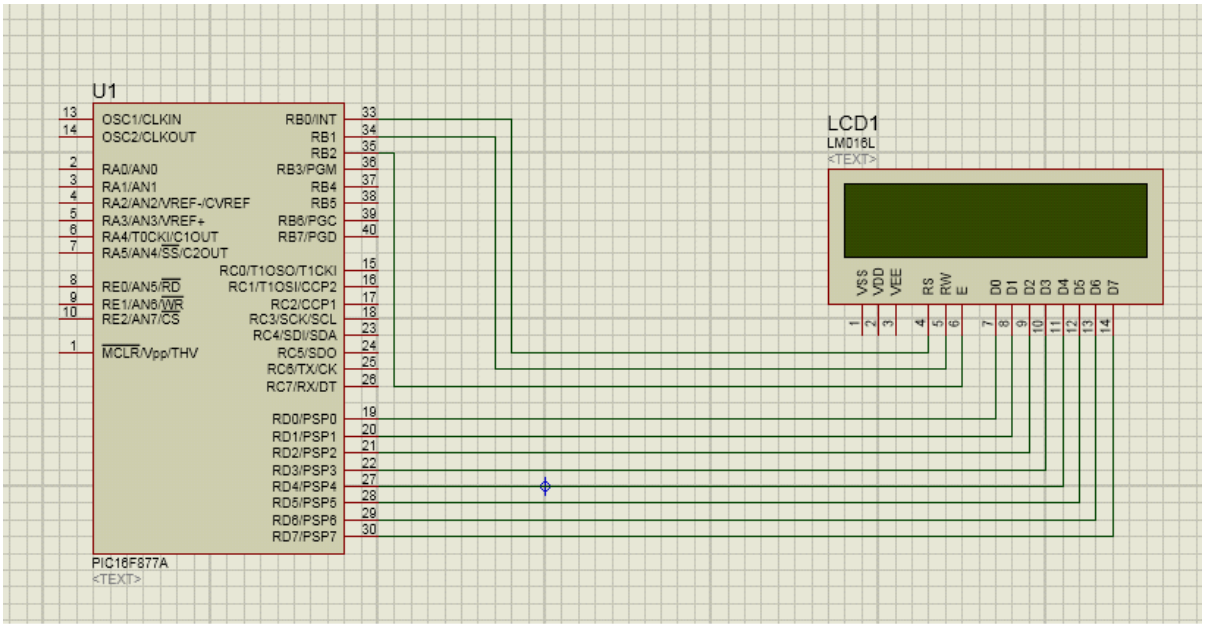
AIM: Write an ALP to interface the lcd 16x2 to P16f877A.

THEORITICAL CONCEPT:

```
; we are displaying output at lcd conneted to PORTB and PORTD.
; crystal frequency =12Mhz ;
LIST P=PIC16F877A,F=INHX32,N=0,ST=OFF,R=HEX
#include p16f877a.inc
CONFIG OSC =HS,OSCS=OFF
CONFIG WDT = OFF
CONFIG BORV = 45,PWRT = ON,BOR=ON
CONFIG DEBUG = OFF,LVP = OFF,STVR = OFF
ORG 00
LCD_DATA EQU PORTD
LCD_CTRL EQU PORTB
RS      EQU RB0
RW      EQU RB1
EN      EQU RB2
      CLRF TRISD
       CLRF TRISB
         BCF LCD_CTRL,EN
         CALL LDELAY
         MOVLW 0X38
         CALL COMNWRT
         CALL LDELAY
         MOVLW 0X0E
         CALL COMNWRT
         CALL LDELAY
         MOVLW 0X01
         CALL COMNWRT
         CALL LDELAY
```

```
        MOVLW 0X06
        CALL COMNWRT
        CALL LDELAY
        MOVLW 0X84
        CALL COMNWRT
        CALL LDELAY
        MOVLW A'N'
        CALL DATAWRT
        CALL DELAY
        MOVLW A'O'
        CALL DATAWRT
AGAIN:  BTG LCD_CTRL,0
        BRA AGAIN
COMNWRT:  MOVWF LCD_DATA
        BCF LCD_CTRL,RS
        BCF LCD_CTRL,RW
        BSF LCD_CTRL,EN
        CALL SDELAY
        BCF LCD_CTRL,EM
        RETURN
DATAWRT:    MOVWF LCD_DATA
        BSF LCD_CTRL,RS
        BCF LCD_CTRL,RW
        BSF LCD_CTRL,EN
          CALL SDELAY
         BCF LCD_CTRL,EN
         RETURN
END
```

EXPERIMENTAL SETUP



START PROTEUS SIMULATION:

- Place your component from the library , connect them accordingly.
- Click the "pick from library(P)" button .
- select item from the list (**PIC16f877A ,16x2 LCD**).
- Click ok .
- After selecting component ,click anywhere in the design area to select it and then click again to place it.
- Place all the required components.
- connect the desired node by clicking at starting and ending point.
- Double click the 8051 component to open its properties.
- Browse for the HEX file.
- The controls at the left-bottom corner will help you simulate the circuit in real time.

PRECAUTIONS:

Make sure correct power supply is given to the kit/Equipment. Wrong power supplies may cause damage to your equipments

RESULT AND COMMENTS: We have successfully studied the interfacing of PIC with LCD.

EXPERIMENT 10

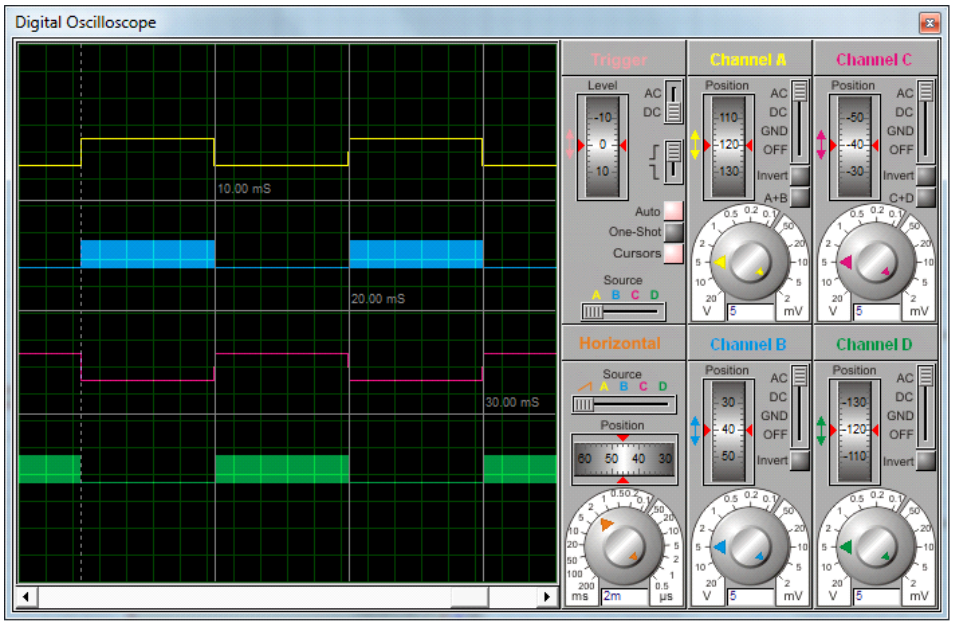
AIM: Write an ALP to generate square wave P16f877A.

THEORITICAL CONCEPT:

```
; we are generating square wave using PortB as output Port.
; crystal frequency =12Mhz ;
LIST P=PIC16F877A,F=INHX32,N=0,ST=OFF,R=HEX
#INCLUDE P16F877A.INC
CONFIG OSC =HS,OSCS=OFF
CONFIG WDT = OFF
CONFIG BORV = 45,PWRT = ON,BOR=ON
CONFIG DEBUG = OFF,LVP = OFF,STVR = OFF
R1 EQU 0X07
R2 EQU 0X08
R3 EQU 0X09
ORG 00
CLRF TRISB
MOVLW 0X55
MOVWF PORTB
L3 COMF PORTB,F
        CALL QDELAY
        BRA L3
QDELAY
        MOVLW D'2'
        MOVWF R1
D1      MOVLW D'250'
        MOVWF R2
D2      MOVLW D'250'
        MOVWF R3
```

```
D3      NOP
        NOP
        DECF R3,F
        BNZ D1
RETURN
END
```

EXPERIMENTAL SETUP:



Drive signals from top to bottom:

- Drive Signal A
- Drive Signal D
- Drive Signal C
- Drive Signal B

PROCEDURE:

START PROTEUS SIMULATION:

- Place your component from the library , connect them accordingly.
- Click the "pick from library(P)" button .
- select item from the list (**PIC16f877A ,16x2 LCD**).
- Click ok .
- After selecting component ,click anywhere in the design area to select it and then click again to place it.

- Place all the required components.
- connect the desired node by clicking at starting and ending point.
- Double click the 8051 component to open its properties.
- Browse for the HEX file.
- The controls at the left-bottom corner will help you simulate the circuit in real time

PRECAUTIONS:

Make sure correct power supply is given to the kit/Equipment. Wrong power supplies may cause damage to your equipments

RESULT AND COMMENTS:

We have successfully studied the generation of square wave using PIC microcontroller.