A.I.S.S.M.S. COLLEGE OF ENGINEERING PUNE 411001

INDEX

Expt. No.	Title of the Experiment	Date on which		Page No.	Remarks
		Experiment Performed	Expt. & result Submitted		
1.	LED Interfacing	24/07/23	31/07/23	1-10	
2.	Multiplexed 7-Seg interfacing	31/07/23	7/08/23	11-16	
3.	Waveform Using DAC	7/08/23	11/09/23	17-22	
4.	Interfacing Switch, Relay, LED	11/09/23	9/10/23	23-26	
5.	Interfacing LCD ~~with~~	9/10/23	16/10/23	~~27-34~~	
6.	Square wave generation using timer	16/10/23	23/10/23	35-41	
7.	Interfacing Serial Port	23/10/23	30/10/23	42-46	
8.	Interfacing DC Motor	30/10/23	3/11/23	47-49	
9.					
10.					
11.					
12.					
13.					
14.					
15.					

This is certify that Mr. / Miss Lekhanish Ghyar of Class T.E ENTC Roll No. 21ET019 has completed all the practical work as listed above, satisfactorily in the subject of Microcontroller in the Department of Electronics & Telecommunication as prescribed by the Savitribai Phule Pune University. During the academic year 20 23 20 24

Date

Prof. in-charge

Head of the Department

Page No.
Date

Experiment No-1

Title: Interfacing of output (LED) devices to the ports.

Aim: Write an embedded C program of 8051 microcontroller for interfacing LED's as output device to Port 2 as follows.

(i) Flashing LED
(ii) As a Counter.
(iii) To display BCD data
(iv) To display HEX data
(v) To display Character.

Hardware: ST(8051, PC, RS-232, cable with USB

Software: KeilDE, Flashmagic

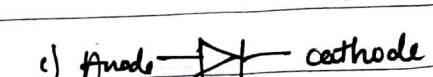
Theory: Current flows from anode to cathode A 330Ω register is connected between port line and LED Resistance of an LED is almost zero. Hence Current flowing through LED is $I = V/R$ which is approximately in mA.

In this experiment, the LEDs are connected to port of the microcontroller.

The LED's will glow alternatively with the time delay.

(a) 

Anode Cathode
Top View:

c) 

Symbol.

Page No.
Date

Procedure.

a) Compile the written code if errors are present then debug the code, modify it, save it and recompile the code. using build all option. After build succeed and HEX file generated

b) Download and run this program.

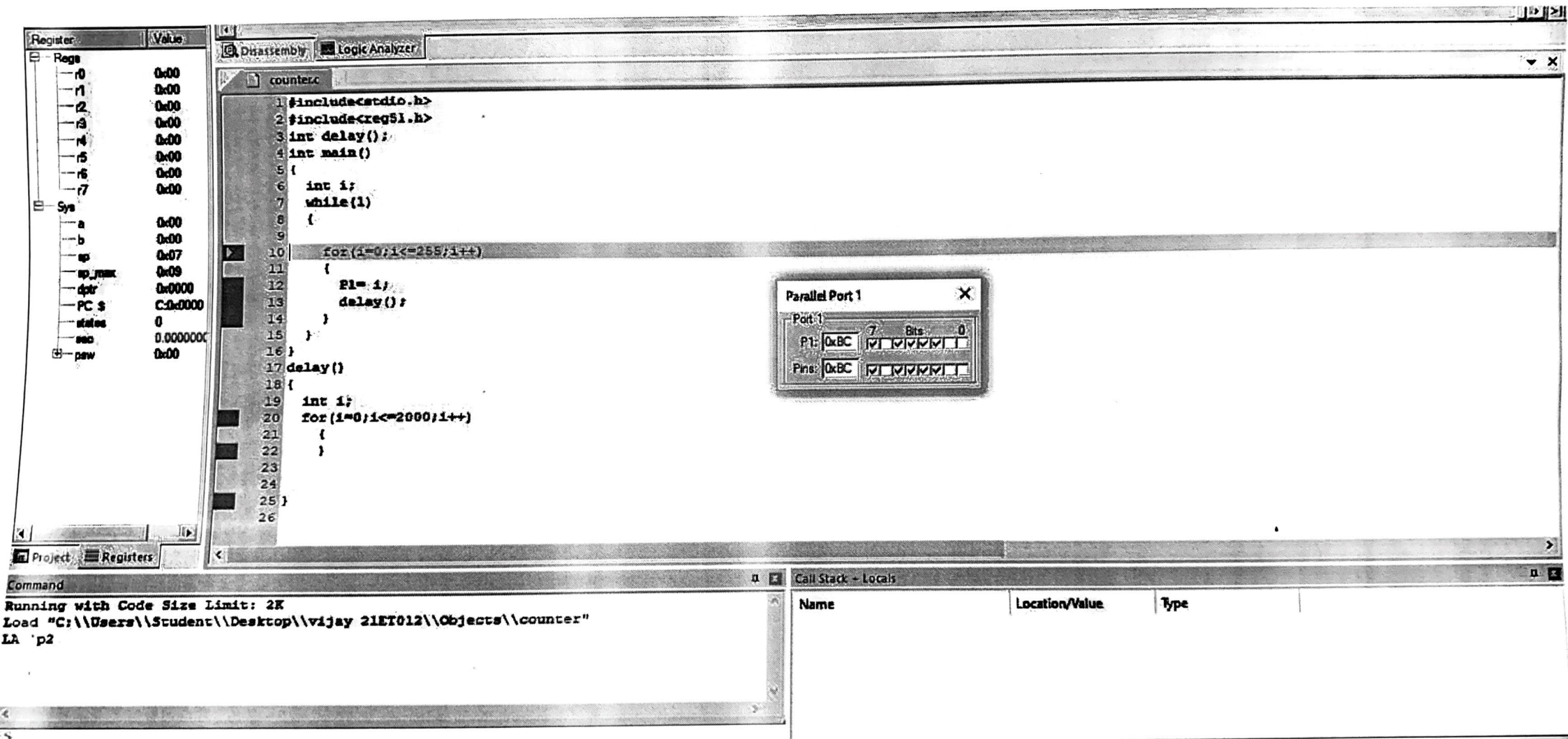
Conclusion: Thus we interfaced LED's with different outputs codes with microcontrollers.

hex.c*

```c
/*Name:Vijay Dhakane
  Roll No: 21ET012
  Batch:'C'
*/

#include<reg51.h>
#include<stdio.h>

void main()
{
  int i,j,k;
  while(1)
  {
    for(i=0;i<16;i++)
    {
      P3=i;
      for(j=0;j<1000;j++)
      {
        for(k=0;k<1000;k++)
        {
        }
      }
    }
  }
}

```

LED FL.C

```
/*Name: Vijay Dhakane
  Roll No:21ET012
  Batch:'C'
*/
#include<stdio.h>
#include<reg51.h>
main()
{
  int i;
  while(1)
  {
    P2=1;
    for(i=0;i<=100;i++)
    {
    }
    P2=0;
    for(i=0;i<=100;i++)
    {
    }
  }

}
```


Register
Value
Regs
r0 0x00
r1 0x00
r2 0x00
r3 0x00
r4 0x00
r5 0x00
r6 0x00
r7 0x00
Sys
a 0x00
b 0x00
sp 0x07
sp_max 0x09
dptr 0x0000
PC $ C:0x0000
states 0
sec 0.000000
psw 0x00
Project
Registers
Disassembly
Logic Analyzer
counter.c
1 #include<stdio.h>
2 #include<reg51.h>
3 int delay();
4 int main()
5 {
6 int i;
7 while(1)
8 {
9
10 for(i=0;i<=255;i++)
11 {
12 P1= i;
13 delay();
14 }
15 }
16 }
17 delay()
18 {
19 int i;
20 for(i=0;i<=2000;i++)
21 {
22 }
23
24
25 }
26
Parallel Port 1
Port 1
7 Bits 0
P1: 0xBC
Pins: 0xBC
Command
Running with Code Size Limit: 2K
Load "C:\\Users\\Student\\Desktop\\vijay 21ET012\\Objects\\counter"
LA `p2
Call Stack + Locals
Name
Location/Value
Type

triangle.c*

```
#include <reg51.h>

void delay();

void main()
{
  int i;
  while(1)
  {
    for(i=0; i<256;i++)
    {
      P1=i;
    }
    for(i=256; i>0; i--)
    {
      P1=i;
    }
  }

}
void delay()
{

  TMOD=0x10;
  TH0=0xFC;
  TL0=0x62;
  TR0=1;
  while(TF0==0);
  TR0=0;
  TF0=0;
}

```

Page No.
Date

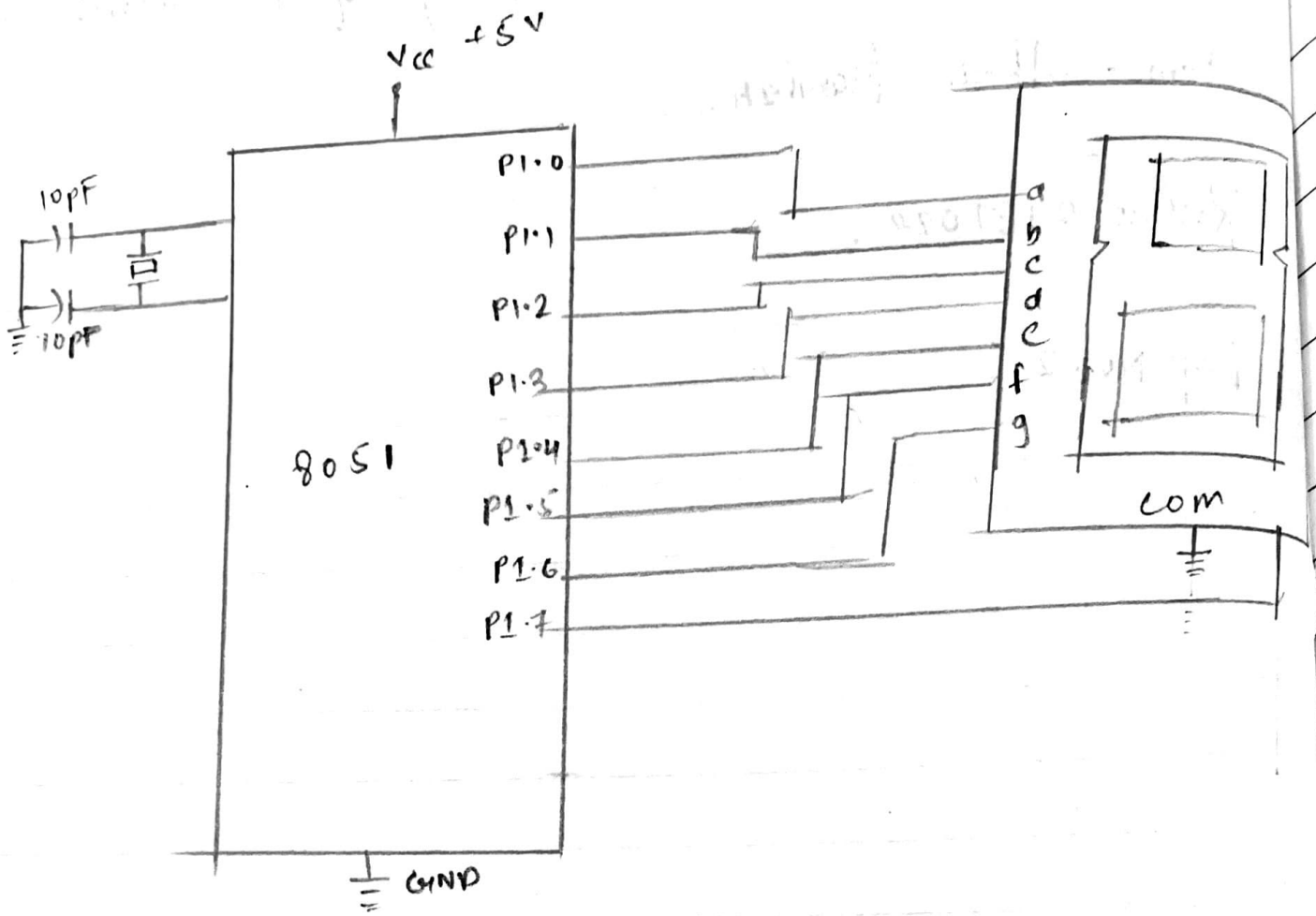
Title: Interfacing of Multiplexed 7-Segment Display.

Hardware: STK 8051, kit, PC, RS-232, cable with USB.

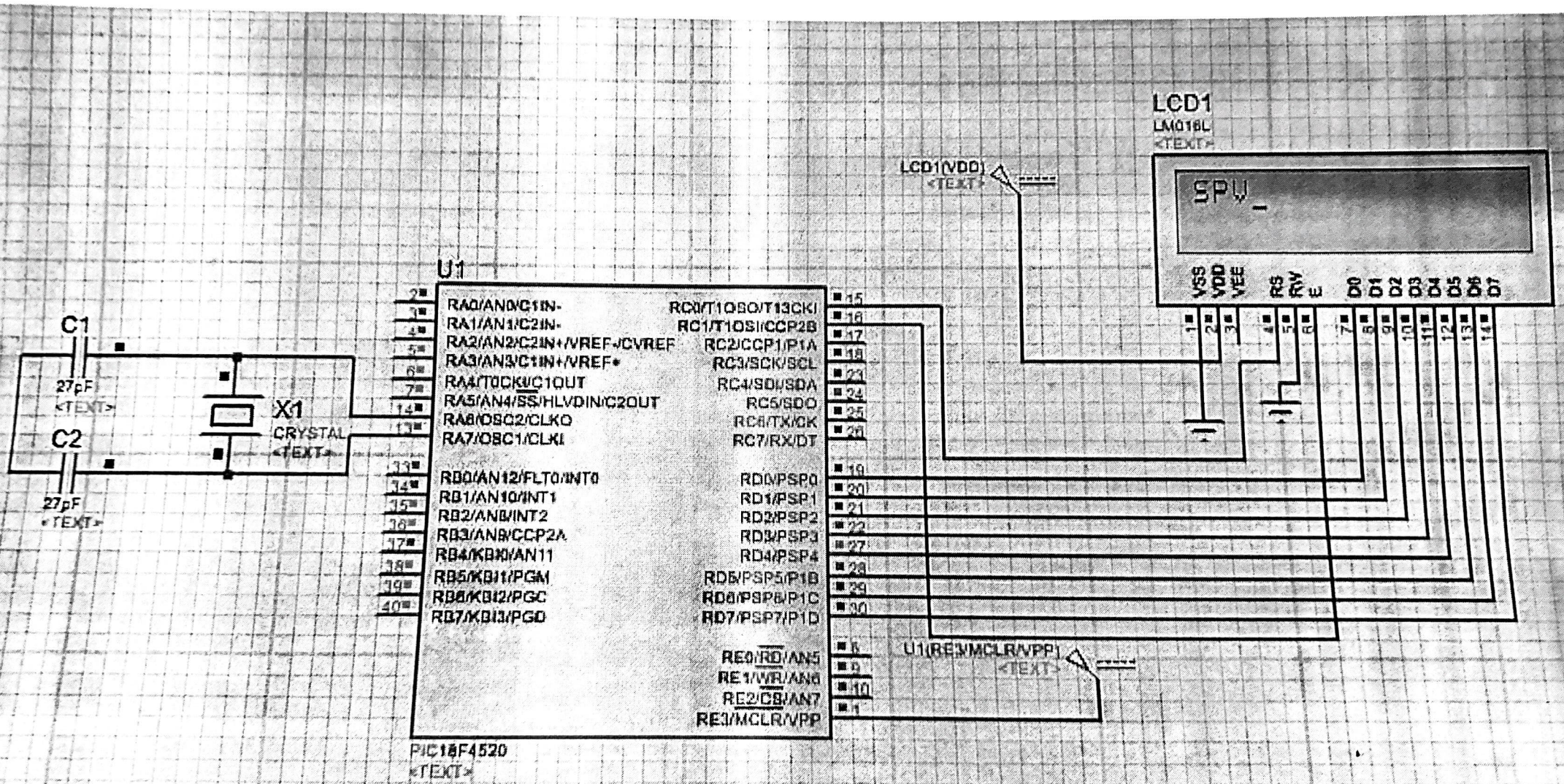
Software: KeilDE, Flashmagic.

Theory: A Seven Segment Display is the most basic electronic display device that can display the digits from (0-F). The Seven Segment pins (a, b, c, d, e, f, g) plus the decimal point of common anode display are connected to port pins of 8051. The program is developed using software display hexadecimal numbers of an display.

Number	h	g	f	e	d	c	b	a
0	0	0	1	1	1	1	1	1
1	0	0	0	0	0	1	1	0
2	0	1	0	1	1	0	1	1
3	0	1	0	0	1	1	1	1
4	0	0	1	1	0	1	1	0
5	0	1	1	0	1	1	0	1
6	0	1	1	1	1	1	0	1
7	0	0	0	0	0	1	1	1
8	0	1	1	1	1	1	1	1
9	0	1	1	0	1	1	1	1



Interfacing 8051 with Seven Segment Display


LCD1
LM016L
<TEXT>
LCD1(VDD)
<TEXT>
SPY_
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
U1
RA0/AN0/C1IN-
RA1/AN1/C2IN-
RA2/AN2/C2IN+/VREF-/CVREF
RA3/AN3/C1IN+/VREF+
RA4/T0CKI/C1OUT
RA5/AN4/SS/HLVDIN/C2OUT
RA6/OSC2/CLKO
RA7/OSC1/CLKI
RB0/AN12/FLT0/INT0
RB1/AN10/INT1
RB2/AN8/INT2
RB3/AN9/CCP2A
RB4/KBI0/AN11
RB5/KBI1/PGM
RB6/KBI2/PGC
RB7/KBI3/PGD
RC0/T1OSO/T13CKI
RC1/T1OSI/CCP2B
RC2/CCP1/P1A
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5/P1B
RD6/PSP6/P1C
RD7/PSP7/P1D
RE0/RD/AN5
RE1/WR/AN6
RE2/CS/AN7
RE3/MCLR/VPP
PIC18F4520
<TEXT>
U1(RE3/MCLR/VPP)
<TEXT>
C1
27pF
<TEXT>
C2
27pF
<TEXT>
X1
CRYSTAL
<TEXT>

Page No.
Date

Title: Waveform generation using DAC.

Aim: Write a program to generate sinusoidal wave.

Equipments: P89V51RD2 DAC board, CRO, RS232 Cable, FRC Cables, Power adapter etc.

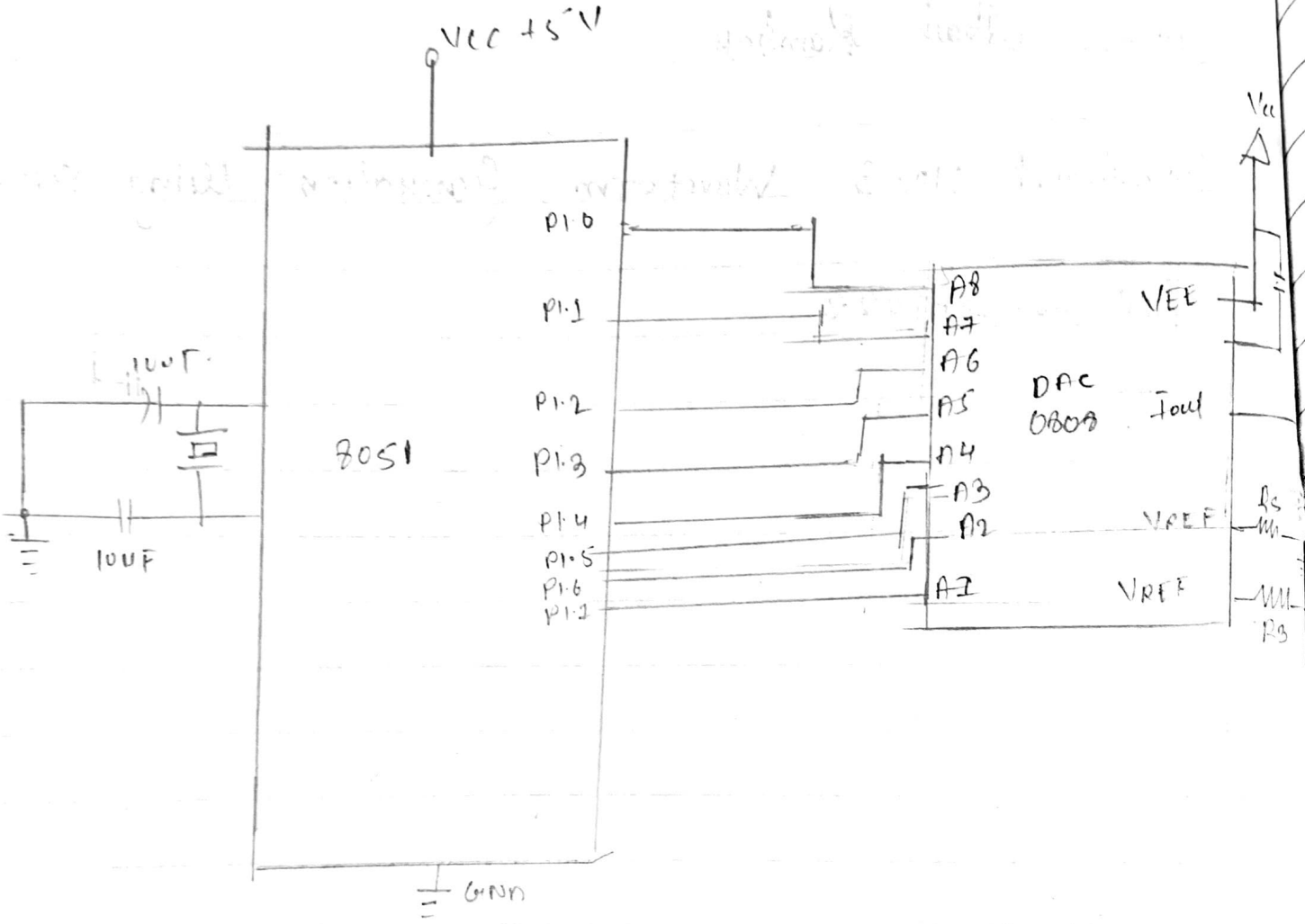
Software: Keil Version 4, Flash Magic

Theory: The digital to analog Converter is a device widely use to convert digital Pulses to analog signals. The two methods of creating DAC are binary weighted and R2R ladder. DAC 0808 uses the R-2R method. since it can achieve a high degree of precision. The number of data bit input decide the resolution of DAC since the number of data inputs. The total current provided by Iout pin is a function of binary number at D0-D7 pin inputs to DAC 0808 and reference current Iref.

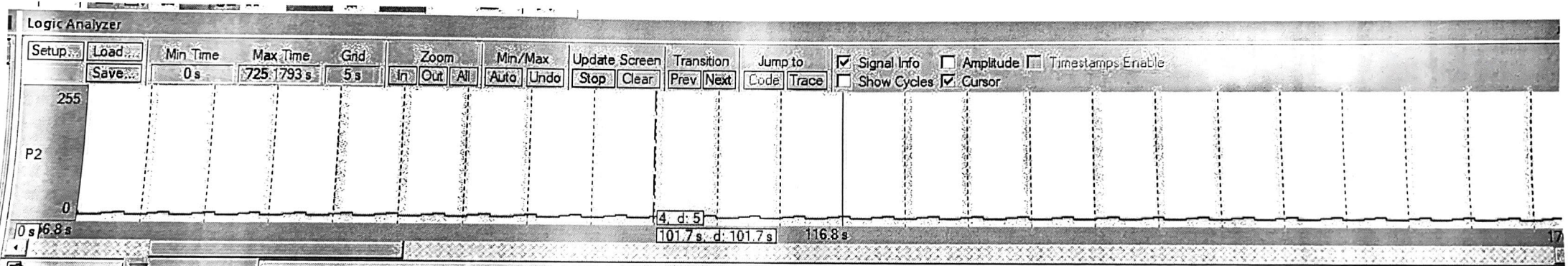
Generating a Sinewave.

To generate a sinewave, we first need a table whose values represent the magnitude of sine of angles between 0 and 360°. These table values are integer numbers representing the voltage. To generate table, we assumed full scale voltage of 10V. For DAC Full Scale output of the DAC is achieved when all data inputs are high.

INTERFACING DIAGRAM.

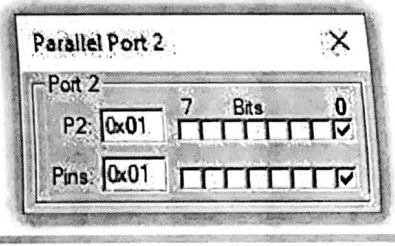


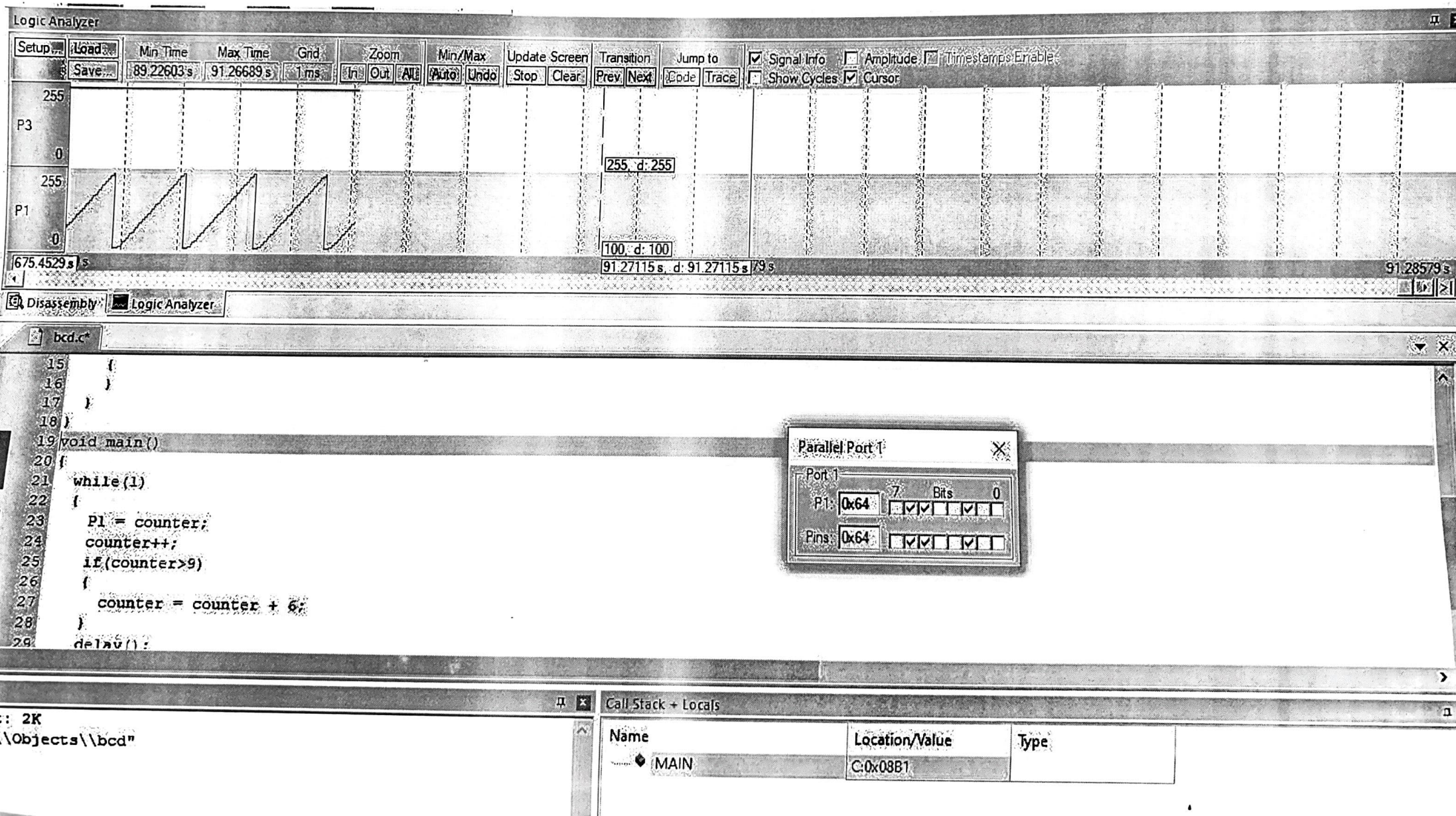
Interfacing 8051 with DAC to generate waveform



steppermotor.c*

```
3   Batch:'c'
4 */
5 #include<reg51.h>
6 #include<stdio.h>
7 void delay(int);
8 void main()
9 {
10   do
11   {
12     P2=0x01;
13     delay(1000);
14     P2=0x02;
15     delay(1000);
16     P2=0x04;
17     delay(1000);
18     P2=0x08;
```




Logic Analyzer
Setup...
Load...
Save...
Min Time
89.22603 s
Max Time
91.26689 s
Grid
1 ms
Zoom
In
Out
All
Min/Max
Auto
Undo
Update Screen
Stop
Clear
Transition
Prev
Next
Jump to
Code
Trace
Signal Info
Amplitude
Timestamps Enable
Show Cycles
Cursor
255
P3
0
255
P1
0
255, d: 255
100, d: 100
675.4529 s
91.27115 s, d: 91.27115 s
91.28579 s
Disassembly
Logic Analyzer
bcd.c*
15 {
16 }
17 }
18 }
19 void main()
20 {
21 while(1)
22 {
23 P1 = counter;
24 counter++;
25 if(counter>9)
26 {
27 counter = counter + 6;
28 }
29 delay();
Parallel Port 1
Port 1
P1: 0x64
7 Bits 0
Pins: 0x64
imit: 2K
BCD\\Objects\\bcd"
Call Stack + Locals
Name
Location/Value
Type
MAIN
C:0x08B1

Page No.
Date

Experiment No-5

Title: Interfacing of LCD (16x2) with PIC18F458

Aim: Write an assembly language program to interface an LCD to 8051. Write down a program to display "HELLO" on first line and "ENTC" on second line.

Hardware: STK PIC KIT, PC, PIC, kit 2

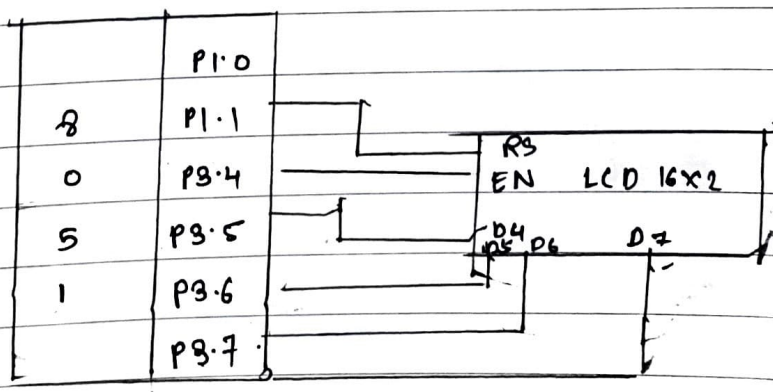
Software: MIQE IOE, Flash Magic

LCD 4-Bit ~~Comparation~~ Operation.

LCD in 4 Bit means we are using 4 lines of databus. Instead of using 8 line data bus. In this method we are splitting byte of data in Nibbles. If you successfully interface Microcontrollers with LCD with 4 pins. Then we can save 4 lines of Microcontrollers which can be used for other purpose.

Connection.
8051 ports 3 and port 1 pins are used to interface LCD pins.

Port P1.0 RS
Port P1.1 EN
Port P3.4 D4
Port P3.5 D5
Port P3.6 D6



Page No.
Date

16x2 LCD

LCD display are available typically as 16x2, 20x1, 20x2 etc. along with LCD controllers. This type of LCD module is very common. and used widely in many types of display. application. It consists of 16 rows and 2 columns. of 5x7 dot matrices. The LCD display adjustment was 16 pin. pack with back light, contrast adjustment and 5x7 dot resolution. It consists of two built in registers known as data to be displayed and Common register is to place the ground. command. This pin description of 16x2 LCD was given below.

The datas as the commands. both the given to the LCD through the data placed D0 to D7 but the logic state in the pin decides. the data or command was given to the LCD.

Alphanumeric Liquid Crystal Display allow battery user interface with text messages. to enter the instruction and get the. respectively in the form of the text and known in a. better manner.

Advantages. of LCD over LED

The decline of LCD over LED.
The ability to display numbers, characters graphics.
Ease of programming for Characters.

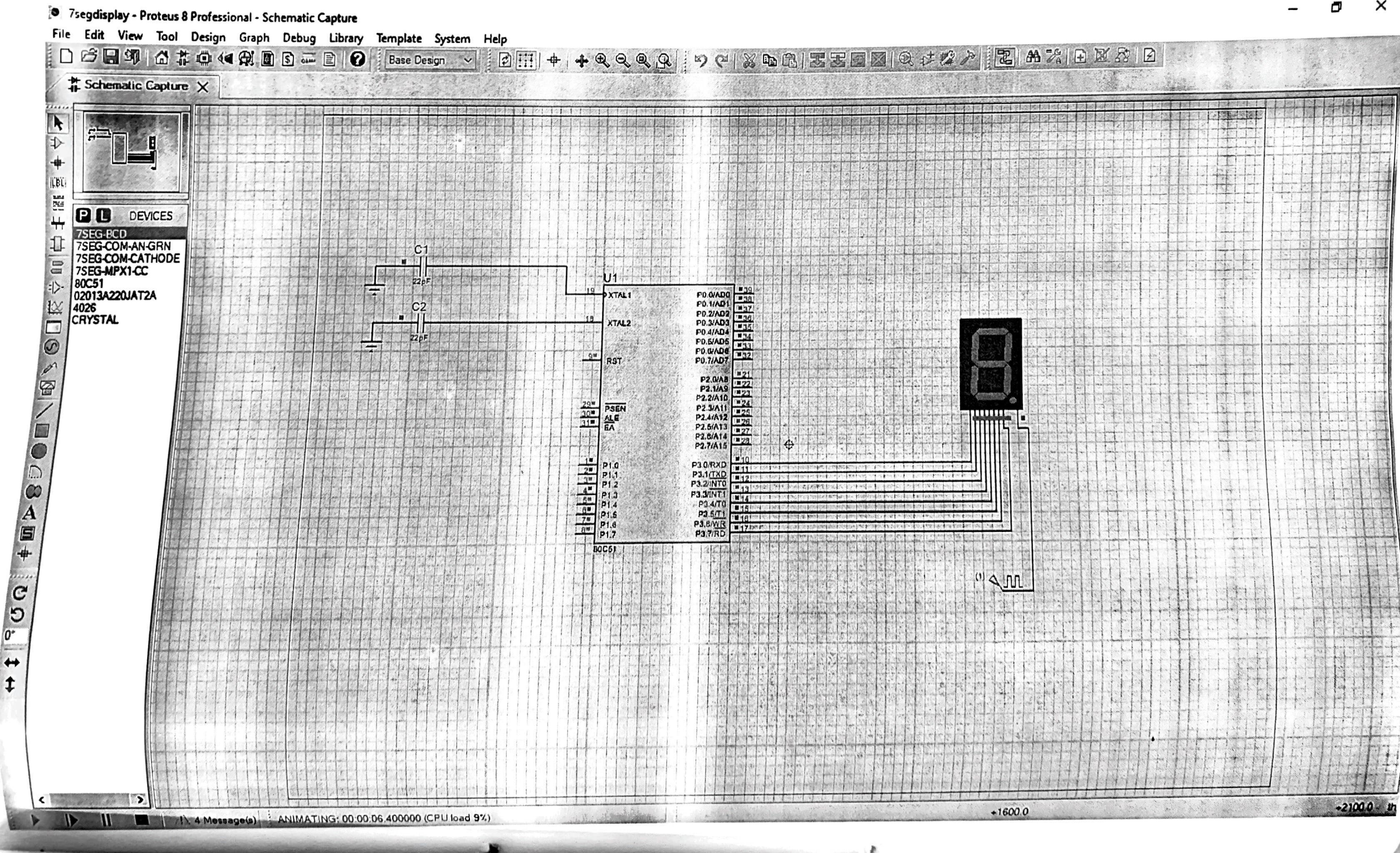
Page No.
Date

Pin Description

PIN	SYMBOL	Description
1	Vss	GROUND
2	Vcc	+5V or +6V
3	VEE	Power Supply
4	Rs	Rs=0 Command Rs=1 Data
5	R/W	RW=0 write RW=1 Read
6	E	ENABLE
7	DB0	Data bit 0
8	DB1	Data bit 1
9	DB2	Data bit 2
10	DB3	Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6
14	DB7	Data bit 7

Command Routine

1. Microcontroller will send command on port 3
2) Microcontroller makes P1.0 low which means RS = 0.
3) It is write operation
4) The command which is available is latched into command Regior.
5) After receiving the command, LCD will perform operation.
6) LCD will take some time to perform operation.


7segdisplay - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Base Design
Schematic Capture
DEVICES
7SEG-BCD
7SEG-COM-AN-GRN
7SEG-COM-CATHODE
7SEG-MPX1-CC
80C51
02013A220JAT2A
4026
CRYSTAL
C1
22pF
C2
22pF
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
80C51
4 Message(s)
ANIMATING: 00:00:06.400000 (CPU load 9%)
+1600.0
+2100.0

Page No.
Date

Experiment No-6.

Generate Square wave using timer with interrupt.

Aim: Generate Square wave using timer ~~and~~ with interrupt

Software: MPLABX, XC8, Compilers.

Theory: The Timer 0 module incorporates following features

- Software selectable operation as a timer, or Counter.
- Readable and writeable Registers.
- Dedicated 8 bit, software programmable prescale
- Selectable clock source.
- Edge Select for external Clock.
- Interrupt on overflow.

T0CON: Timer 0 control Register

bit 7: TMR0ON: Timer 0 ON/OFF

Control bit 1= Enable Timer 0.

bit 6: T0CS: Timer 0 Clock Source Select bit

1= Translation.

0= Internal instruction cycle.

bit5: T0CS: Timer 0 clock source select bit

1= Transition

0= Internal Instruction cycle.

bit 4 : 1 = Increament on high to low.
0 : Increment on low to high.

bit 2-0 : TOPS2 : TOPS0 : Timer0.
Prescaler Select bits 111 = 1 : 256.
110 = 1:128
101 = 1:64
100 = 1:32
011 = 1:8.
001 = 1:4
000 = 1:2

The PIC18F4550 device have multiple source and an interrupt feature that allows an interrupt source to be assigned a high or low level.

There are ten registers which are used to control interrupt Operation.

- RCON.
- INTICON.
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2

When an interrupt is responded to, the global interrupt enable bit is cleared. to Sources can interrupt a low Priority interrupt.

INTCON : Interrupt Control Register 1 = ENABLE

Date

To generate a Square wave using timer.

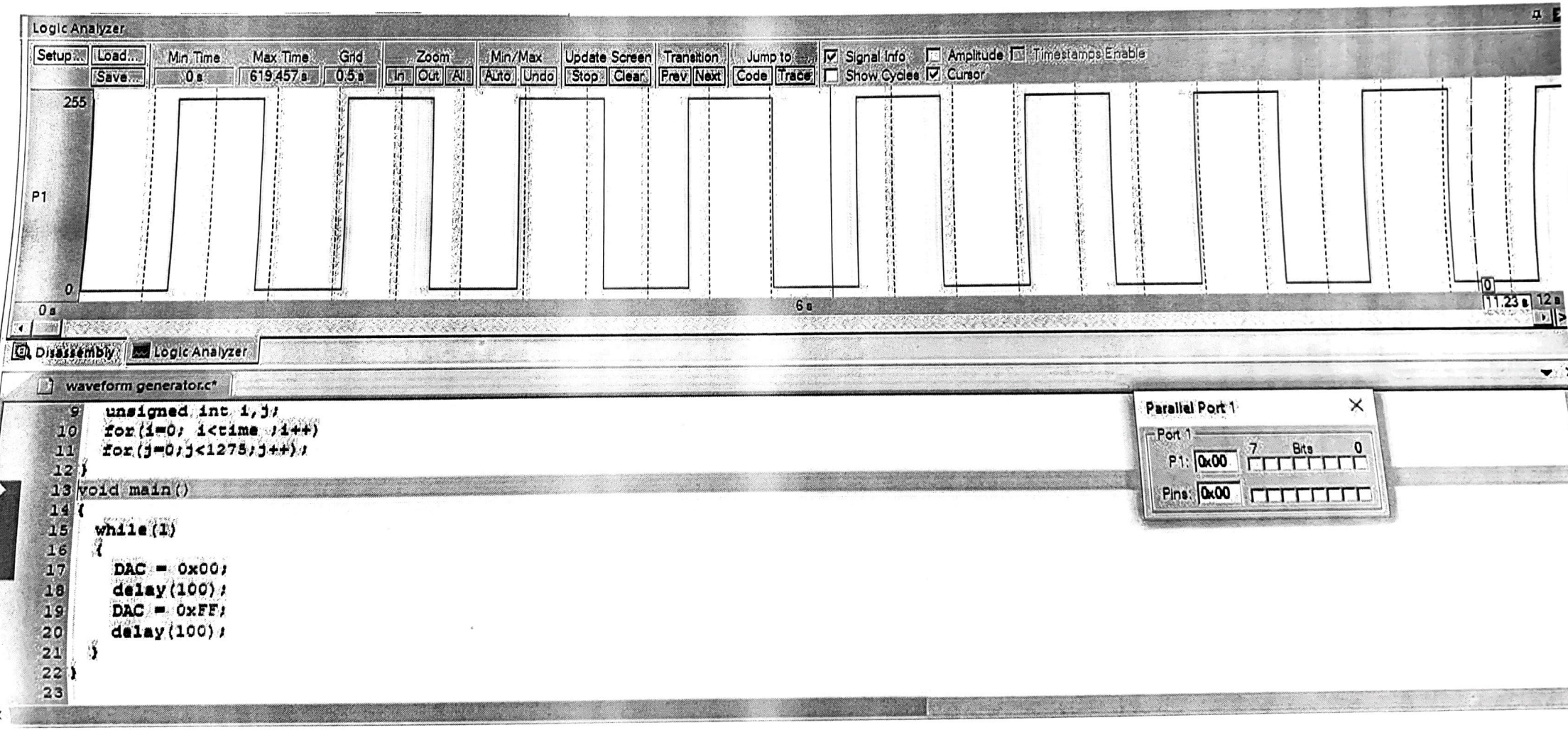
Configure port pin as output
Configure timer 0 in 16 bit mode.
Set timer 0 Interrupt Enable bit and Clear timer 0.
Enable Global Interrupt Register.
Load the Timer. 0 16bit in TMR0L and TMR0H
Enable Timer 0.

Interrupt Service Routine

Check if Timer 0 Interrupt flag is set
If the flag is set, clear the flag and Stop timer.
Toggle the port pin.
Reload the Timer 0 Values.
Return to the Main function.

Conclusion: Thus we Implemented and generated square wave using interrupt using PIC Microcontroller and learned how to Program timer and interrupts Experimentally.

```
#include <p18f458.h>

#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=ON, LVP=OFF

#define mybit PORTDbits.RD4

void main (void);
void ISR(void);

#pragma interrupt ISR      ///redirect it from address location 00008 to another rogram

void ISR(void)
{
mybit=~mybit;
INTCONbits.TMR0IF = 0;
T0CONbits.TMR0ON = 0;
}

#pragma code Interrupt = 0x08 //High Priority interrupt location
void Interrupt(void)
{
  _asm
   goto ISR //jump to interrupt routine
  _endasm
}
#pragma code    //End of code

void main ()
{
TRISDbits.TRISD4=0;
T0CON = 0x08;                    //set up timer0 - prescaler 1:128
TMR0H = 0XFF;                    //clear timer
TMR0L = 0XF0;                    //clear timer
INTCONbits.TMR0IF = 0;
INTCONbits.TMR0IE = 1;
T0CONbits.TMR0ON = 1;
INTCONbits.PEIE = 1;
INTCONbits.GIE = 1;              //enable interrupts
}
```


Logic Analyzer
Setup...
Load...
Save...
Min Time
0 s
Max Time
619.457 s
Grid
0.5 s
Zoom
In
Out
All
Min/Max
Auto
Undo
Update Screen
Stop
Clear
Transition
Prev
Next
Jump to ...
Code
Trace
Signal Info
Show Cycles
Amplitude
Cursor
Timestamps Enable
255
P1
0
0 s
6 s
11.23 s
12 s
Disassembly
Logic Analyzer
waveform generator.c*
unsigned int i,j;
for(i=0; i<time ;i++)
for(j=0;j<1275;j++);
}
void main()
{
while(1)
{
DAC = 0x00;
delay(100);
DAC = 0xFF;
delay(100);
}
}
Parallel Port 1
Port 1
P1:
0x00
7
Bits
0
Pins:
0x00

Page No.
Date

Experiment No - 7 Serial Communication

Aim: Interfacing Serial port with PC with both side Communication

Hardware: STK, PIC kit, PC, PIC kit 2

Software: MIDE, IDE, Flash Magic

Theory

Serial Communication is classified into three type of communication

Simplex: In Simplex, the line is dedicated for transmission.

Half Duplex: In half Duplex, the Communication link can be use for either transmission or reception.

Full Duplex: If data is ~~connected~~ transmitted in both ways at the same is full Duplex i.e both can be done simultaneously. Synchronous. Serial Data Communication.

In Synchronous Serial Data Communication transmitter and receiver are Synchronized. It use a common clock signal.

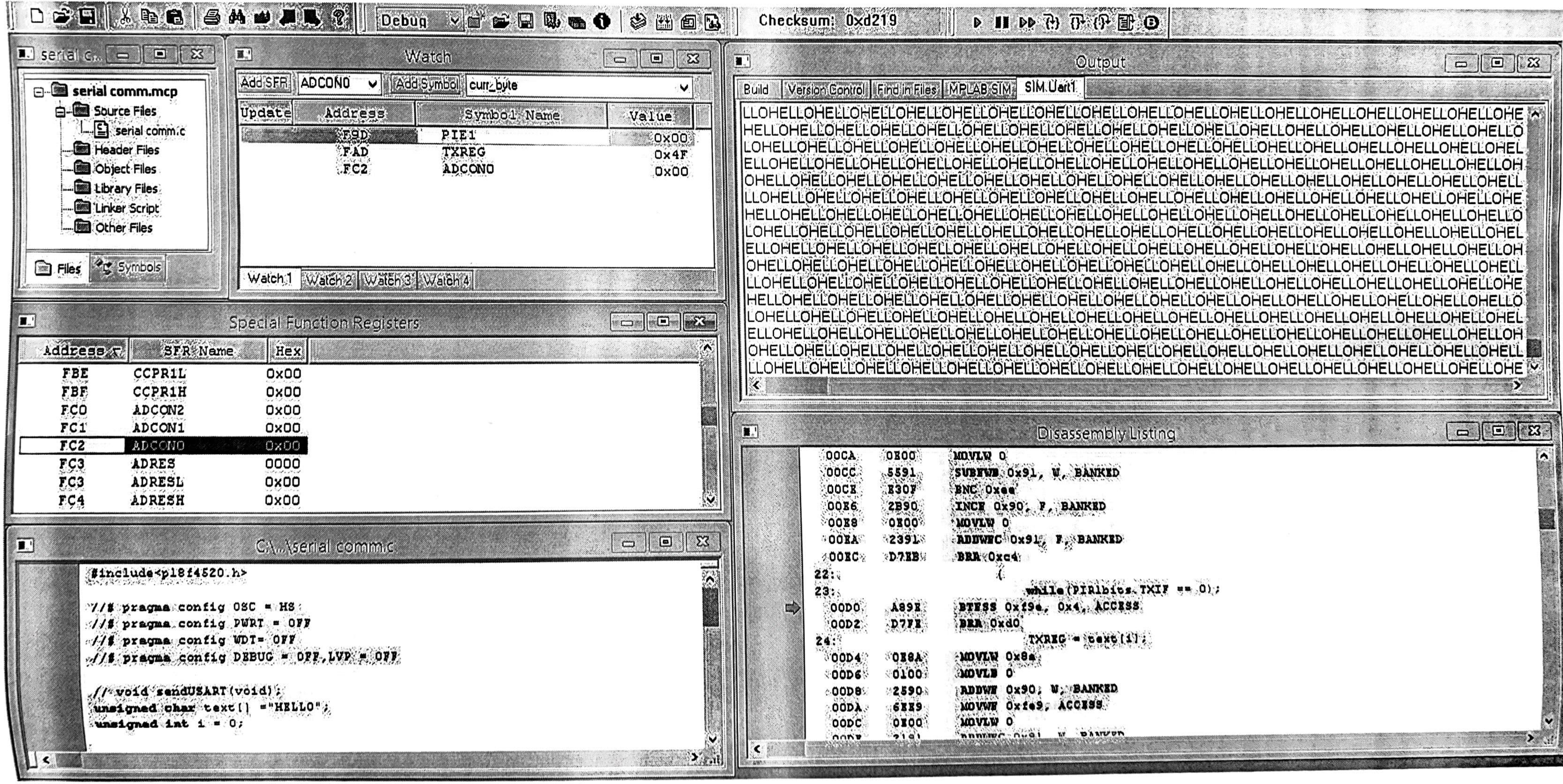
Asynchronous. Serial Data Communication.
In asynchronous. Serial data Communication different clock source are used for transmitter and receiver. comparison.

Asynchronous Transfer.	Synchronous transfer
1) Used to transfer one character at a time	1) Used to transfer a block of characters at a time.
2) Used for transfer rates L=20k. BPS.	2) Used for high data rate
3) Short bit and Stop bit for Character is present.	3) No Stop and Stop bit is used
4) Sync Characters are not transmitted.	4) Sync characters are transmitted
5) Speed is slow.	5) Speed is high.

The Operation of Enhanced USART is performed using following registers.

1) Serial port Baud rate Generator (SPBRG).
2) Transfer Buffer Register (TXREG) and Receiver Buffer Register (RCREG)
3) Transmit Status and Control (TXSTA)
4) Receive Status and Control (RCSTA).
5) Baud Rate Control (BAUDCON).

```
# include<p18f4520.h>

#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF ,LVP = OFF


void main(void)
{
    TRISC=0x00;
    PR2=249;
    CCPR1L=189;
    T2CON=0x01;
    CCP1CON=0x1C;
    TMR2=0x00;
    T2CONbits.TMR2ON=1;
    while(1)
    {
        PIR1bits.TMR2IF=0;
        while(PIR1bits.TMR2IF==0);
    }
}
```


Debug
Checksum: 0xd219
serial comm.mcp
Source Files
serial comm.c
Header Files
Object Files
Library Files
Linker Script
Other Files
Files
Symbols
Watch
Add SFR
ADCON0
Add Symbol
curr_byte
Update
Address
Symbol Name
Value
F9D PIE1 0x00
FAD TXREG 0x4F
FC2 ADCON0 0x00
Watch 1
Watch 2
Watch 3
Watch 4
Output
Build
Version Control
Find in Files
MPLAB SIM
SIM Uart1
LLOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHELLOHE
Special Function Registers
Address
SFR Name
Hex
FBE CCPR1L 0x00
FBF CCPR1H 0x00
FC0 ADCON2 0x00
FC1 ADCON1 0x00
FC2 ADCON0 0x00
FC3 ADRES 0000
FC3 ADRESL 0x00
FC4 ADRESH 0x00
C:\...\serial comm.c
#include<p18f4520.h>
//# pragma config OSC = HS
//# pragma config PWRT = OFF
//# pragma config WDT= OFF
//# pragma config DEBUG = OFF,LVP = OFF
// void sendUSART(void);
unsigned char text[] ="HELLO";
unsigned int i = 0;
Disassembly Listing
00CA 0E00 MOVLW 0
00CC 5591 SUBFWB 0x91, W, BANKED
00CE E30F BNC 0xee
00E6 2B90 INCF 0x90, F, BANKED
00E8 0E00 MOVLW 0
00EA 2391 ADDWFC 0x91, F, BANKED
00EC D7EB BRA 0xc4
22: {
23: while(PIR1bits.TXIF == 0);
00D0 A89E BTFSS 0xf9e, 0x4, ACCESS
00D2 D7FE BRA 0xd0
24: TXREG = text[i];
00D4 0E8A MOVLW 0x8a
00D6 0100 MOVLB 0
00D8 2590 ADDWF 0x90, W, BANKED
00DA 6EE9 MOVWF 0xfe9, ACCESS
00DC 0E00 MOVLW 0

Experiment No-8

Title: Interfacing of DC Motor using pwm.

Aim: Write an embedded C program for generation of PWM signal for DC Motor.

Hardware: STK PIC kit, PC, PIC, kit 2

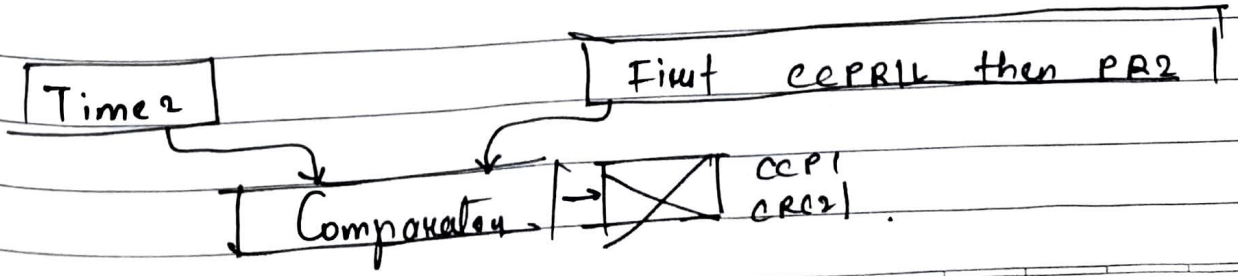
Software: MPLAB IDE.

Theory

The PWM feature allows us to create pulses with variable. Although can program timers to create PWM, the CCP module Makes the programming of PWM much easier. It is widely used in- industrial Controls such as DC Motor Control. In Creating pulse with Variable width ~~for the~~ ~~sum~~. PWM. Variable width for the PWM.

Steps in Programming PWM.

1) Set PWM period by writing to PR2
2) Set duty cycle by writing to CCPR1L
3) Set CCP pin as an output.
4) Using the TCON register, set the prescale value.
5) Clear the TMR2 register.
6) Start Timer 2.

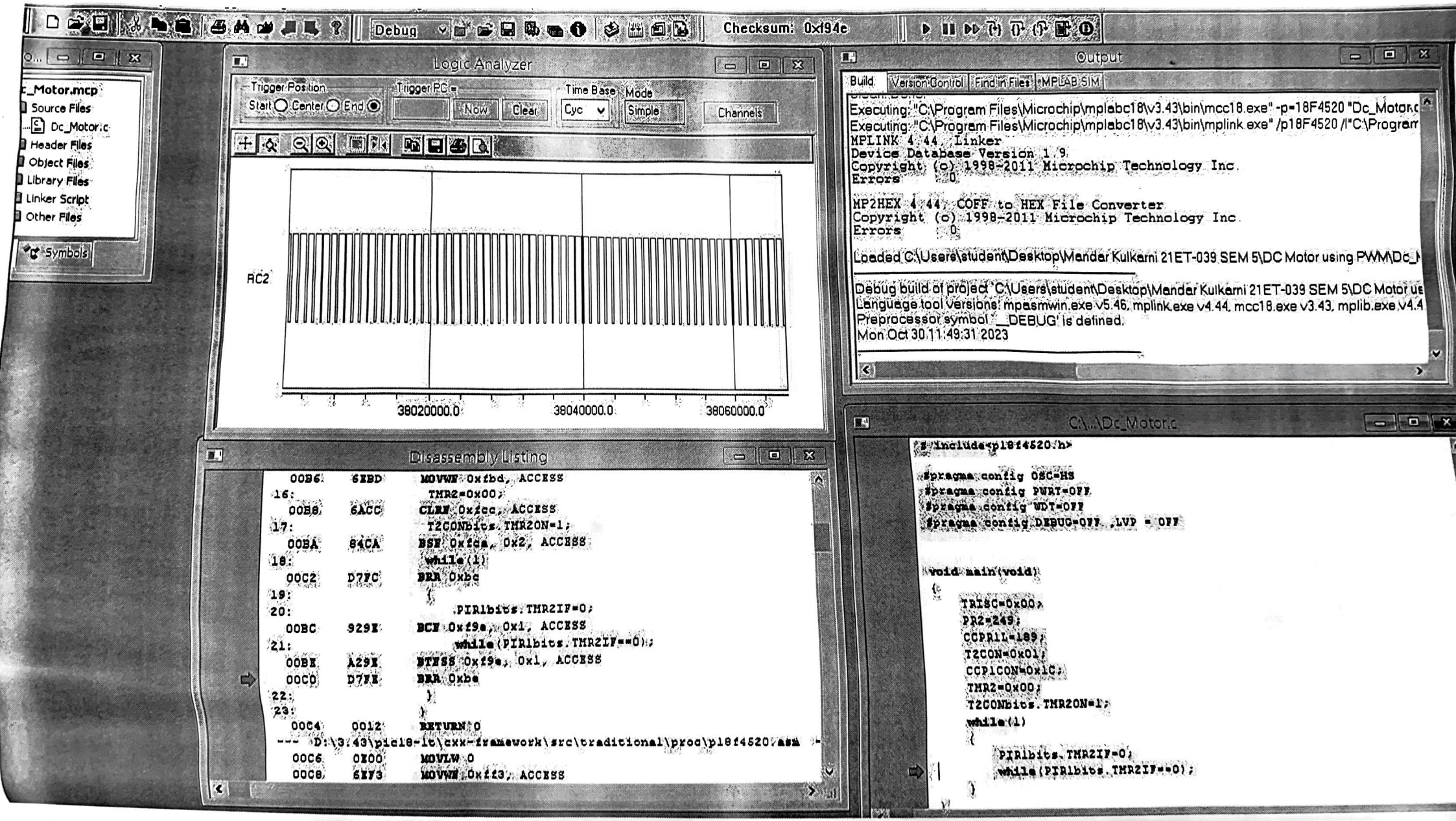


Procedure.

Compile the written code if errors are present then debug the code, modify it, save it and recompile the code using build all option.
Download and Run this program.

Conclusion

Thus we performed the experiment on MPLAB IDE Software and generated PWM using DC motor interfaced with PIC18 microcontroller.


Debug
Checksum: 0x194e
Logic Analyzer
Trigger Position
Start
Center
End
Trigger PC =
Now
Clear
Time Base
Cyc
Mode
Simple
Channels
RC2
38020000.0
38040000.0
38060000.0
Output
Build
Version Control
Find in Files
MPLAB SIM
Executing: "C:\Program Files\Microchip\mplabc18\v3.43\bin\mcc18.exe" -p=18F4520 "Dc_Motor.c
Executing: "C:\Program Files\Microchip\mplabc18\v3.43\bin\mplink.exe" /p18F4520 /l"C:\Program
MPLINK 4.44, Linker
Device Database Version 1.9
Copyright (c) 1998-2011 Microchip Technology Inc.
Errors : 0
MP2HEX 4.44, COFF to HEX File Converter
Copyright (c) 1998-2011 Microchip Technology Inc.
Errors : 0
Loaded C:\Users\student\Desktop\Mandar Kulkarni 21ET-039 SEM 5\DC Motor using PWM\Dc_M
Debug build of project 'C:\Users\student\Desktop\Mandar Kulkarni 21ET-039 SEM 5\DC Motor us
Language tool versions: mpasmwin.exe v5.46, mplink.exe v4.44, mcc18.exe v3.43, mplib.exe v4.4
Preprocessor symbol '__DEBUG' is defined.
Mon Oct 30 11:49:31 2023
Disassembly Listing
00B6 6EBD MOVWF 0xfbd, ACCESS
16: TMR2=0x00;
00B8 6ACC CLRF 0xfcc, ACCESS
17: T2CONbits.TMR2ON=1;
00BA 84CA BSF 0xfca, 0x2, ACCESS
18: while(1)
00C2 D7FC BRA 0xbc
19: {
20: PIR1bits.TMR2IF=0;
00BC 929E BCF 0xf9e, 0x1, ACCESS
21: while(PIR1bits.TMR2IF==0);
00BE A29E BTFSS 0xf9e, 0x1, ACCESS
00C0 D7FE BRA 0xbe
22: }
23: }
00C4 0012 RETURN 0
--- D:\3.43\pic18-lt\cxx-framework\src\traditional\proc\p18f4520.asm
00C6 0E00 MOVLW 0
00C8 6EF3 MOVWF 0xff3, ACCESS
C:\...\Dc_Motor.c
#include<p18f4520.h>
#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF ,LVP = OFF
void main(void)
{
TRISC=0x00;
PR2=249;
CCPR1L=189;
T2CON=0x01;
CCP1CON=0x1C;
TMR2=0x00;
T2CONbits.TMR2ON=1;
while(1)
{
PIR1bits.TMR2IF=0;
while(PIR1bits.TMR2IF==0);
}
Dc_Motor.mcp
Source Files
Dc_Motor.c
Header Files
Object Files
Library Files
Linker Script
Other Files
Symbols