1.Study of Embedded C programming language (Overview, syntax, One simple program like addition of two numbers).

#include <xc.h>

void main(void)

{ unsigned int num1, num2, sum;

TRISB=0;

LATB=0;

num1=0x04;

num2 =0x05;

sum=num1+num2;

PORTB=sum;

PORTC=num1;

PORTD=num2;

return;

}

2. Write an Embedded C program to add array of n numbers

#include <stdio.h>

#include <stdlib.h>

#include <PIC18F4550.h>

void main(void) {

int i,sum,n;

int number[]={1,2,3,4,5,6,7,8,9,10};

sum=0;

for(i=0;i<=9;i++)

{sum=sum+number[i];

}

TRISD=0;

PORTD=sum;

}

3. Write an Embedded C menu driven program for: i) Multiply 8-bit number by 8-bit number ii) Divide 8-bit number by 8-bit number.

#include<xc.h>

void main(void)

{ unsigned int num1, num2, div, mul ;

TRISB=0;

num1=0x02;

num2=0x02;

mul=num1/num2;

div=num1/num2;

PORTB=mul;

PORTC=div;

PORTD=num1;

PORTE=num2;

return;

}

4. Write an Embedded C program to interface PIC 18FXXX with LED & blinking it using specified delay

#include <pic18f4520.h>

#pragma config OSC=HS

#pragma config WDT=OFF

#pragma config LVP=OFF

#pragma PBADEN=OFF

// Function Prototype

void msdelay(unsigned int time);

void main() {

INTCON2bits.RBPU = 0;

ADCON1 = 0x0F;

TRISD = 0x00;

while(1) {

PORTD = 0xFF;

msdelay(250);

PORTD = 0x00; // Added a semicolon here

msdelay(250);

}

}

void msdelay(unsigned int time) {

unsigned int i, j;

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

for(j = 0; j < 710; j++); // Added a semicolon here

}

5. Buzzer & Relay.

//Interfacing LEDs, Switches, Buzzer and Relay

//Includes

#include <xc.h>         //Include Controller specific .h

//Configuration bit settings

#pragma config OSC = HS //Oscillator Selection

#pragma config WDT = OFF //Disable Watchdog timer

#pragma config LVP = OFF //Disable Low Voltage Programming

#pragma config PBADEN = OFF //Disable PORTB Analog inputs

//Declarations

#define lrbit   PORTBbits.RB0   //SW1 interfaced to RB0

#define rlbit   PORTBbits.RB1   //SW2 interfaced to RB1

#define relay   PORTBbits.RB2   //Relay interfaced to RB2

#define buzzer  PORTBbits.RB3   //Buzzer interfaced to RB3

//Function Prototypes

void msdelay (unsigned int time);//Function for delay

//Start of Program Code

void main()                     //Main Program

{

    unsigned char val=0;        //Variable to latch the switch condition

    INTCON2bits.RBPU=0;         //To Activate the internal pull on PORTB

    ADCON1 = 0x0F;              //To disable the all analog inputs

    TRISBbits.TRISB0=1;         //To configure RB4 as input for sensing SW0

    TRISBbits.TRISB1=1;         //To configure RB5 as input for sensing SW1

    TRISBbits.TRISB2=0;         //To configure RC1 (relay) as output

    TRISBbits.TRISB3=0;         //To configure RC2 (buzzer) as output

    TRISD = 0x00;               //To configure PORTD (LED) as output

    PORTD = 0x00;               //Initial Value for LED

    buzzer = 0;                 //Initial Value for Buzzer

    relay = 0;                  //Initial Value for Relay

while (1)                       //While loop for repeated operation

    {

    if (!(lrbit))               //To check whether SW0 is pressed

        val = 1;                // Latch the status of switch SW0

    if (!(rlbit))               //To check whether SW1 is pressed

        val = 2;                // Latch the status of switch SW1

    if (val == 1)

       {

        buzzer = 1;

        relay = 1;

        PORTD = PORTD >>1;      //Shift left by 1 bit

        if (PORTD == 0x00)

            PORTD = 0x80;       // Make the MSB bit equal to 1

        msdelay(250);

       }

    if (val == 2)

       {

        buzzer = 0;

        relay = 0;

        PORTD = PORTD<<1;       //Shift right by 1 bit

        if (PORTD == 0x00)

            PORTD = 0x01;       // Make the LSB bit eqaul to 1

        msdelay(250);

       }

    }

}                               //End of the Program

//Function Definitions

void msdelay (unsigned int time)//Function for delay

{

unsigned int i, j;

 for (i = 0; i < time; i++)

     for (j = 0; j < 275; j++); //Calibrated for a 1 ms delay in MPLAB

}

6. Write an Embedded C program for LCD interfacing with PIC 18FXXX.

#include <pic18f4520.h>

#pragma config OSC = HS

#pragma config WDT = OFF

#pragma config LVP = OFF

#pragma config PBADEN = OFF

#define LCD\_DATA PORTD

#define ctrl PORTE

#define rs PORTEbits.RE0

#define rw PORTEbits.RE1

#define en PORTEbits.RE2

void init\_LCD(void);

void LCD\_command(unsigned char cmd);

void LCD\_data(unsigned char data);

void LCD\_write\_string(char \*str);

void msdelay (unsigned int time);

void main(void)

{

char var1[] = "MAYUR";

char var2[] = " DXFG";

ADCON1 = 0x0F;

TRISD = 0x00;

TRISE = 0x00;

init\_LCD();

msdelay(50);

LCD\_write\_string(var1);

msdelay(15);

LCD\_command(0xC0);

LCD\_write\_string(var2);

while (1);

}

//Function Definitions

void msdelay (unsigned int time)

{

unsigned int i, j;

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

for (j = 0; j < 275; j++);

}

void init\_LCD(void)

{

LCD\_command(0x38);

msdelay(15);

LCD\_command(0x01);

msdelay(15);

LCD\_command(0x0C);

msdelay(15);

LCD\_command(0x80);

}

void LCD\_command(unsigned char cmd)

{

LCD\_DATA = cmd;

rs = 0;

rw = 0;

en = 1;

msdelay(15);

en = 0;

}

void LCD\_data(unsigned char data)

{

LCD\_DATA = data;

rs = 1;

rw = 0;

en = 1;

en = 0;

}

void LCD\_write\_string( char \*str)

{

int i = 0;

while (str[i] != 0)

{

LCD\_data(str[i]);

msdelay(15);

i++;

}

}

7. DC Motor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Program for PWM Generation using PIC18F4520.

// PWM output : RC2

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <xc.h>

#pragma config OSC = HS //Oscillator Selection

#pragma config WDT = OFF //Disable Watchdog timer

#pragma config LVP = OFF //Disable Low Voltage Programming

#pragma config PBADEN = OFF //Disable PORTB Analog inputs

void myMsDelay (unsigned int time) // Definition of delay subroutine

{

unsigned int i, j;

for (i = 0; i < time; i++) // Loop for i time

for (j = 0; j < 275; j++); // Calibrated for a 1 ms delay in MPLAB

}

void main()

{

TRISCbits.TRISC0 = 0 ; // Set PORTC, RC6 as output (DCM IN1)

TRISCbits.TRISC1 = 0 ; // Set PORTC, RC6 as output (DCM IN2)

TRISCbits.TRISC2 = 0 ; // Set PORTC, RC2 as output (CCP1)

PR2 = 0x4E; // set PWM Frequency 4KHz

CCP1CON = 0x0C; // Configure CCP1CON as PWM mode.

T2CON = 0x07; //Start timer 2 with prescaler 1:16

PORTCbits.RC0 = 1; // Turn ON the Motor

PORTCbits.RC1 = 0;

while(1) // Endless Loop

{

// -----------------------------------

// ----------Duty Cycle 80%-----------

CCP1CONbits.DC1B0 = 0;

CCP1CONbits.DC1B1 = 1;

CCPR1L = 0x3E;

myMsDelay(2000);

// -----------------------------------

// -----------------------------------

// ----------Duty Cycle 60%-----------

CCP1CONbits.DC1B0 = 1;

CCP1CONbits.DC1B1 = 1;

CCPR1L = 0x2E;

myMsDelay(2000);

// -----------------------------------

// ----------Duty Cycle 40%-----------

CCP1CONbits.DC1B0 = 1;

CCP1CONbits.DC1B1 = 0;

CCPR1L = 0x1F;

myMsDelay(2000);

// ----------Duty Cycle 20%-----------

CCP1CONbits.DC1B0 = 0;

CCP1CONbits.DC1B1 = 1;

CCPR1L = 0x0F;

myMsDelay(2000);

}

}

8. Serial Communication

#include <pic18f4520.h>//Include Controller specific .h

#pragma config OSC = HS//Oscillator Selection

#pragma config WDT = OFF//Disable Watchdog timer

#pragma config LVP = OFF//Disable Low Voltage Programming

#pragma config PBADEN = OFF//Disable PORTB Analog inputs

//Variables

#pragma idata

unsigned char string1[]={"\n\rSmart Logic Technologies"};\

unsigned char string2[]={"\n\rUSART Test Code"};

unsigned char string3[]={"\n\rSend 10 character to uC\n\r"};

unsigned char string4[]={"\n\rTransmitted Characters are:"};

unsigned char string5[]={"\n\rRx Tx test complete\n\r"};

//Function Prorotypes

void TXbyte(unsigned char data); //To transmit single character

void TXstring (unsigned char \*string); //To transmit string

//Start of Main Program

void main()

{

unsigned char i=0;

unsigned char rx\_data [20]; // Buffer to store received data

TRISCbits.TRISC7=1;// RXD line as input

TRISCbits.TRISC6=0; // TXD line as output

SPBRG = 0x08;

SPBRGH = 0x02; // 0x0208 for 9600 baud

TXSTA = 0x24; // TX enable BRGH=1, SPEN=1

RCSTA = 0x90; // SPEN= 1, continuous RX = 1

BAUDCON = 0x08; // BRG16 = 1

TXstring (string1); // Transmit string 1

TXstring (string2); // Transmit string 2

TXstring (string3); // Transmit string 3

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

{

while (PIR1bits.RCIF==0); // Wait until data received

rx\_data [i]= RCREG;// Read the received data

}

rx\_data [10]= 0;//To make ASCIIZ string

TXstring (string4);// Transmit string 4

TXstring (rx\_data);// Transmit received data

TXstring (string5);// Transmit string 5

while(1); // loop forever

} // End of the program

void TXbyte(unsigned char data)

{

while(TXSTAbits.TRMT==0);//wait till transmit buffer is not empty

TXREG = data; // Transmit Data

}

void TXstring(unsigned char \*string)

{

unsigned char i=0;

for(i=0;string[i]!='\0';i++) //loop till end of the string

TXbyte(string[i]);//Send single character

}