EMPTY PROGRAM – FOR MPLAB XC 8 / PIC16F887 / XT - osc – HS - 19,6608Mhz

```
#include <xc.h>
#define _XTAL_FREQ 19660800

#pragma config CONFIG1 = 0xEFF2
#pragma config CONFIG2 = 0xFFFF

void main(){
   //Write your code here!
}
```

#Include // #define

```
include File from lib #include <filename>
include user file     #include "filename"

replacement text  #define name text
Example:          #define LEDS  PORTD

replacement macro  #define name(var) text
#define clear_bit( reg, bitNumb )
          ((reg) &= ~(1 << (bitNumb)))
#define set_bit( reg, bitNumb )
          ((reg) |= (1 << (bitNumb)))
```

PROVIL
Provinciaal
Instituut Lommel
provinciaal onderwijs

SET BIT 3

```
PORTC=PORTC|0x08;  // OR
PORTC|=0x08; // shorter
```

CLEAR BIT 3

```
PORTC &=0xF7;  // OR
PORTC &= ~0x08;
```

FLIP BIT 3

```
PORTC=PORTC^0x08;  // OR
PORTC^=0x08;// shorter
```

TEST BIT 3

```
Is bit3 = 1 ?
       if (PORTB&0x08)
Is bit3 = 0 ?
       if (~PORTB&0x08)
```

IF / IF-ELSE / IF-ELSE IF - ELSE

```
if (i<10){
   som++;
} else if (i>23){
   som--;
} else{
   som =0;
}
```

WHILE

```
while (i < N){
   som = som + i;
   i++;
}
```

DO WHILE

```
do{
   som = som + i;
   i++;
} while (i < N);
```

FOR

```
for (i = 0; i < 10; i++){
   som = som + i;
}
```

GO TO

```
label_x:

goto label_x;
```

// try not to use goto!!

SWITCH

```
switch(PORTB){
 case 1:
    RC0=1;
    break;//jump to end
 case 2:
    RC1=1;
    break;
 default: //else….
    RC2=1;
    break;
}
```

void FUNCTION (void)

```
char x = 0;

void add_x (void){
  x = x + 1;
}


main(){
  TRISC = 0x00;
  while (1){
    add_x();
    PORTC = x;
    __delay_ms(30);
  }
}
```

void FUNCTION (int)

```
void del (unsigned int w){
  unsigned int i ;
  for ( i=0 ; i < w ; i++){

  }
}

main(){
  TRISC = 0x00;
  while (1){
    PORTC = 0x00;
    del(64000);
    PORTC = 0xFF;
    del(5000);
  }
}
```

int FUNCTION (int,int)

```
int macht(int x, int y){
  int i,m;
  int a = x;
  for (i = 1; i < y; i++){
    m = (a*x);
    a = m;
  }
  return m;
}

main(){
  int a = 3,b = 3,z = 0;
  z = macht (a,b);
  TRISC = 0x00;
  PORTC = z;
}
```

ARRAY / POINTER

```
char array[5] =
{2,4,3,1,5}; // array[0]=
2, array[1] = 4
char string[5] =
"Hello"; // string[0]=
"H", string[1] = "e"
```

```
char a = 5; // declare a
char a and fill it with
value 5
char *money; // declare a
pointer that can point to
a char
money = &a; // money
points to adress in RAM
where a is stored
*money = 8;      //
changes value of a to 8
```

STRING / POINTER

```
void SHOW(const char
*pString ){
  while ( *pString != 0){
       // no NULL char
     PORTB = *pString;//leds
     pString++;//next adress
   }
}
void main (void){
  char StringA[20] = "Hello
World";
  SHOW(StringA);
  //SHOW("Hello World");
  // also good code
}
```

OPERATORS

ARITHMETIC	+,-,*,/,%	X=5
		Y=8
			Z=Y/X (Z=1)
			Z=Y%X (Z=3)

EQUALITY	==, !=	X=5	If (X!=0) TRUE Read as: if X is not equal to 0

ORDER	<,<=,>,>=	X=5
		Y=8
			If (X>=Y) FALSE
			Read: if X is greater or equal to Y

BYTEWISE LOGIC	!,&&,||	X=5
		Y=8
		Z= 7
			If ((X<Z) &&(Z<Y)) TRUE : X<Z<Y
			((TRUE)&&(TRUE)) = TRUE
			If (!(Z<=Y)) FALSE
			read as: If Z is not <= Y

BITWISE LOGIC	~,&,|,^		~0b00001111 = 0b11110000 0b00111100^0b00001111 = 0b00110011 (bitwise exor)
BITWISE SHIFTS	<<,>>	X=1	X=X<<2 (X=4)(shift left 2 positions) PORTB |=(1<<3) read as: make bit 3 of PORTB = 1
ASSIGNMENT	=, +=, -=, *=, /=, %=, &= , |=, ^=, <<=, >>=		X +=2 (short for X = X + 2) X <<=4 (short for X = X << 4)
INCREMENT	++	X=5	X++ (X=6)
DECREMENT	--	X=5	X-- (X=4)

VARIABLES

TYPE	Size (bits)	RANGE
bit	1	0 to 1
signed char	8	-128 to 127
unsigned char	8	0 to 255
signed short	16	-32768 to 32767
unsigned short	16	0 to 65535
signed int	16	-32768 to 32767
unsigned int	16	0 to 65536
signed short long	24	-8388608 to 8388607
unsigned short long	24	0 to 16777215
signed long	32	-2147483648 to 2147483647
unsigned long	32	0 to 4294967295
float	24	Real (floating point)
double	24 / 32	(FP – double precicion)

Const : something is not modifyable during the run of the program

Volatile : It tells the compiler that the object is subject to sudden change.

Static :A variable declared static in a function retains its state between calls to that function.

ASCII TABLE

Dec	Hex	Description	Dec	Hex	Cha	Dec	Hex	Cha	Dec	Hex	Cha
0	0	null	33	21	!	64	40	@	95	5F	
1	1	start of heading	34	22	"	65	41	A	96	60	`
2	2	start of text	35	23	#	66	42	B	97	61	a
3	3	end of text	36	24	$	67	43	C	98	62	b
4	4	end of transmission	37	25	%	68	44	D	99	63	c
5	5	enquiry	38	26	&	69	45	E	100	64	d
6	6	acknowledge	39	27	'	70	46	F	101	65	e
7	7	bell	40	28	(	71	47	G	102	66	f
8	8	backspace	41	29	)	72	48	H	103	67	g
9	9	horizontal tab	42	2A	*	73	49	I	104	68	h
10	A	new line	43	2B	+	74	4A	J	105	69	i
11	B	vertical tab	44	2C	,	75	4B	K	106	6A	j
12	C	new page	45	2D	-	76	4C	L	107	6B	k
13	D	carriage return	46	2E	.	77	4D	M	108	6C	l
14	E	shift out	47	2F	/	78	4E	N	109	6D	m
15	F	shift in	48	30	0	79	4F	O	110	6E	n
16	10	data link escape	49	31	1	80	50	P	111	6F	o
17	11	device control 1	50	32	2	81	51	Q	112	70	p
18	12	device control 2	51	33	3	82	52	R	113	71	q
19	13	device control 3	52	34	4	83	53	S	114	72	r
20	14	device control 4	53	35	5	84	54	T	115	73	s
21	15	neg.acknowledge	54	36	6	85	55	U	116	74	t
22	16	synchronous idle	55	37	7	86	56	V	117	75	u
23	17	end of trans. block	56	38	8	87	57	W	118	76	v
24	18	cancel	57	39	9	88	58	X	119	77	w
25	19	end of medium	58	3A	:	89	59	Y	120	78	x
26	1A	substitute	59	3B	;	90	5A	Z	121	79	y
27	1B	escape	60	3C	<	91	5B	[	122	7A	z
28	1C	file separator	61	3D	=	92	5C	\	123	7B	{
29	1D	group separator	62	3E	>	93	5D	]	124	7C	|
30	1E	record separator	63	3F	?	94	5E	^	125	7D	}
31	1F	unit separator							126	7E	~
32	20	space							127	7F	DEL

INTERRUPT ROUTINE

```
unsigned char   sPORTC;      // GLOBAL VAR

void main(){
    TRISC = 0x00;            // conf. PORTC as an output
    OPTION = 0b00000111;     // config TMR0 -  presc = 256:
    TOIE = 1;                // enable Timer0 interrupt
    GIE = 1;                 // enable global interrupts

    while(1) PORTC = sPORTC;
}

void interrupt isr(void){    // INTERRUPT SERVICE ROUTINE
        sPORTC++             // increment interrupt count
        TOIF = 0;            // clear interrupt flag
}
```

LIBRARIES

```
#include <xc.h>
__delay_ms(x)  //x!= VAR && x<255
__delay_us(x)
__EEPROM_DATA(a,b,c,d,e,f,g,h)

#include <stdio.h>
#include <math.h>
fabs,floor,ceil,modf,sqrt,atof,sin
,cos,tan,asin,acos,atan,atan2,log,
log10,pow,exp,sinh,cosh,tanh,eval_
poly,frexp,ldexp

#include "HITECH_LCD.h"
LCD_Start
LCD_Clear
LCD_PrintASCII
LCD_Command
LCD_Cursor
LCD_PrintNumber
LCD_PrintString
LCD_ScrollDisplay
LCD_ClearLine
LCD_RAM_Write
```

MACRO's

```
#define GETAL 10000

#define LEDS PORTC


#define clear_bit( reg, bitNumb )
     ((reg) &= ~(1 << (bitNumb)))

#define set_bit( reg, bitNumb )
     ((reg) |= (1 << (bitNumb)))


#define maskbits_on(var,mask)
     var |= mask

#define maskbits_off(var,mask)
     var &= ~0 ^ mask


#define testbit_on(data,bitno)
     ((data>>bitno)&0x01)
```