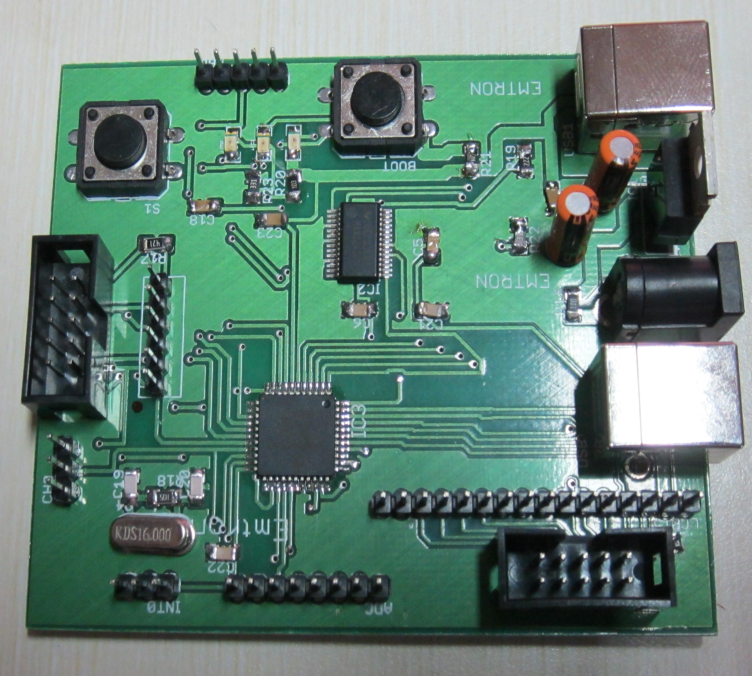


**Emtron Technologies Pvt. Ltd.   
Flat No-101, B3 Wing, 1st Floor, Divyam Hights, Gilbert Hill, Shreenath Nagar, Andheri –West, Mumbai-58   
+91-8080181911 E-mail:** [**emtron.tech@gmail.com**](mailto:emtron.tech@gmail.com)**, www.emtrontech.in**

***Microchip 18F4550 Interface, Signal conditioning, USB, USB-RS-232, 16x2 LCD Interface***

This development board centred on PIC18F4550 M 8-bit RISC micro controllers from Microchip. This controller has 32K flash and 2K RAM. In built 10-bit, up to 13-channel Analog-to-Digital Converter , Four Timer modules (Timer0 to Timer3), Three External Interrupts, Four Crystal modes, including High Precision PLL, Supports up to 32 Endpoints (16 bidirectional), Interface for Off-Chip USB Transceiver.

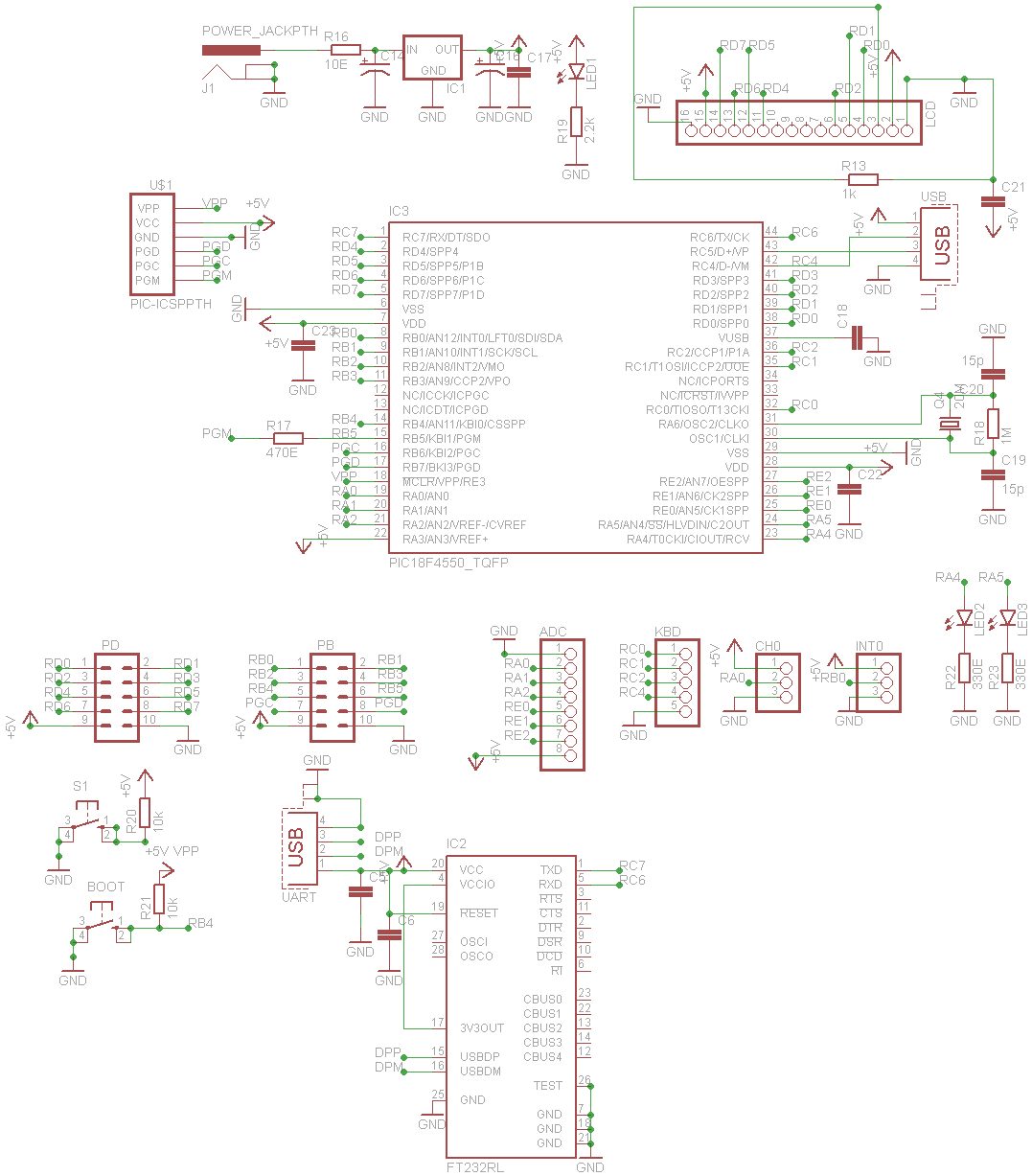
This module of micro controllers is extensively used for embedded and real-time applications in industry. This board is designed to be a general-purpose development board for Single Chip MCU applications that may be used as an instructional learning aid and also as a development tool in R&D labs in industries.

****

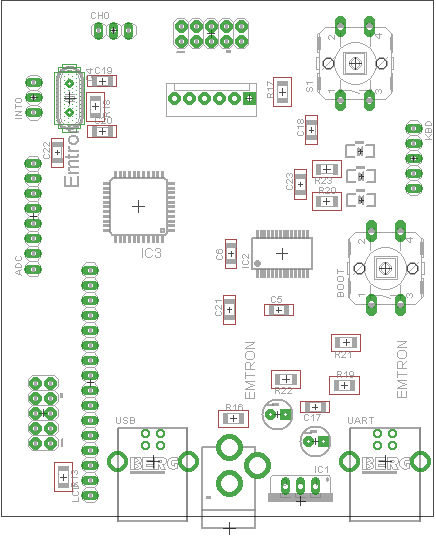
**Board Features:**

PIC18F4550-based board developed (TQFP-44 package) USB powered and programmable Interface. Standard +5v Regulated Power Supply. Serial Port Interface through FT232 UART-USB converter . Large number of on-board I/Os. Connectors are made available for the interfacing of standard I/O such as switches/LEDs and LCD. Compatible with standard Microchip Technology software tools (MPLAB IDE and HID Boot loader V2.6a). PIC Kit3-Debugger Interface. I/O Connectors are compatible to standard interface like Sensors, Opt coupler , 4x1 KBD, Motors etc.

**Circuit Diagram:**

****

**PCB Design:**

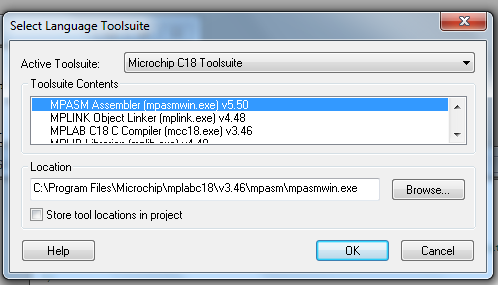
****

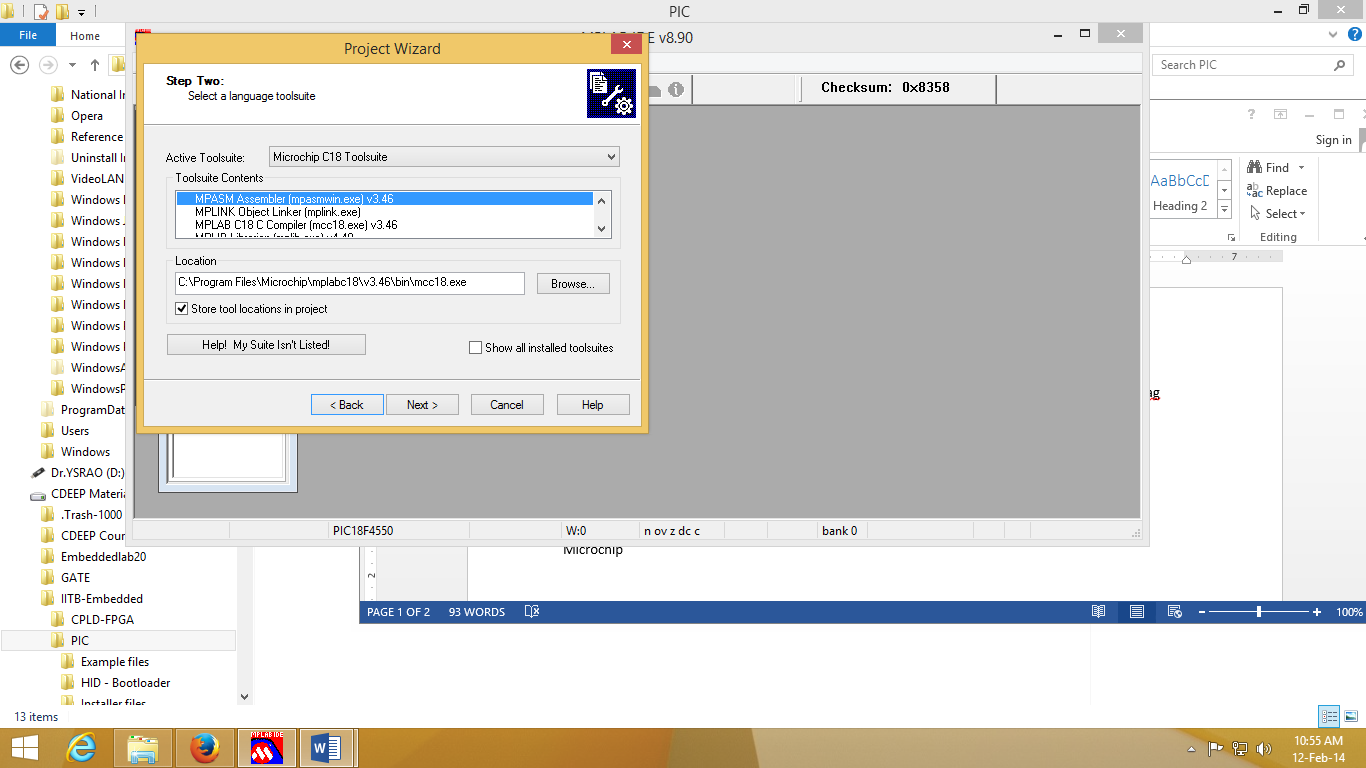
**PIC 18F4550 Installation Guide**

Bootloader is required to make USB enable (programming it using PICKIT3 Programmer through Jtag Port)

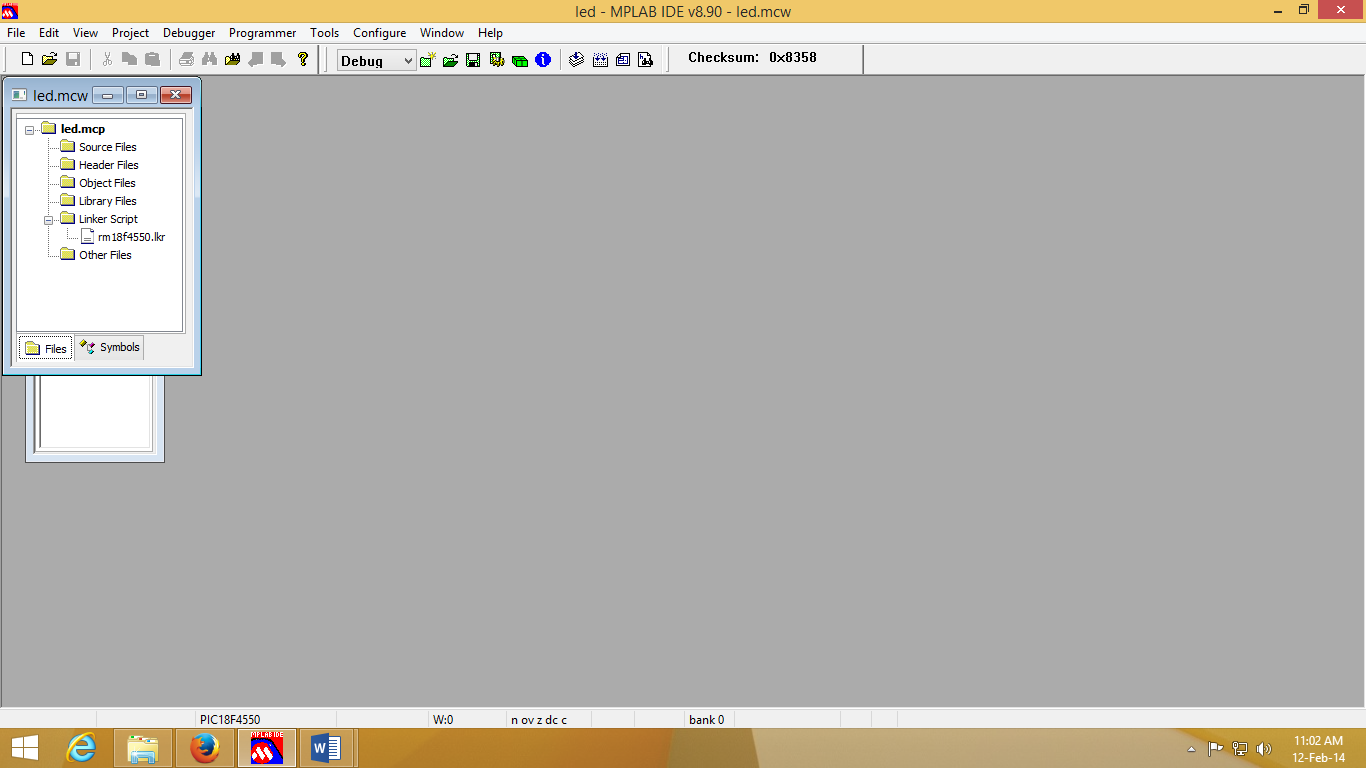
1. Install MPLAB
2. Install c18 compiler (mplabc18-v3.46-windows-lite-installer.exe)
3. Install from the folder NET/ NETCFS Driver
4. Start MPLAB
5. Project->Projectwizard->select device (PIC 18F4550)->select language tool set-> select mpasmwin, mplink,mccc18,mplib

provide path of all four like C:\Program Files\Microchip\mplabc18\v3.46\bin\mcc18.exe

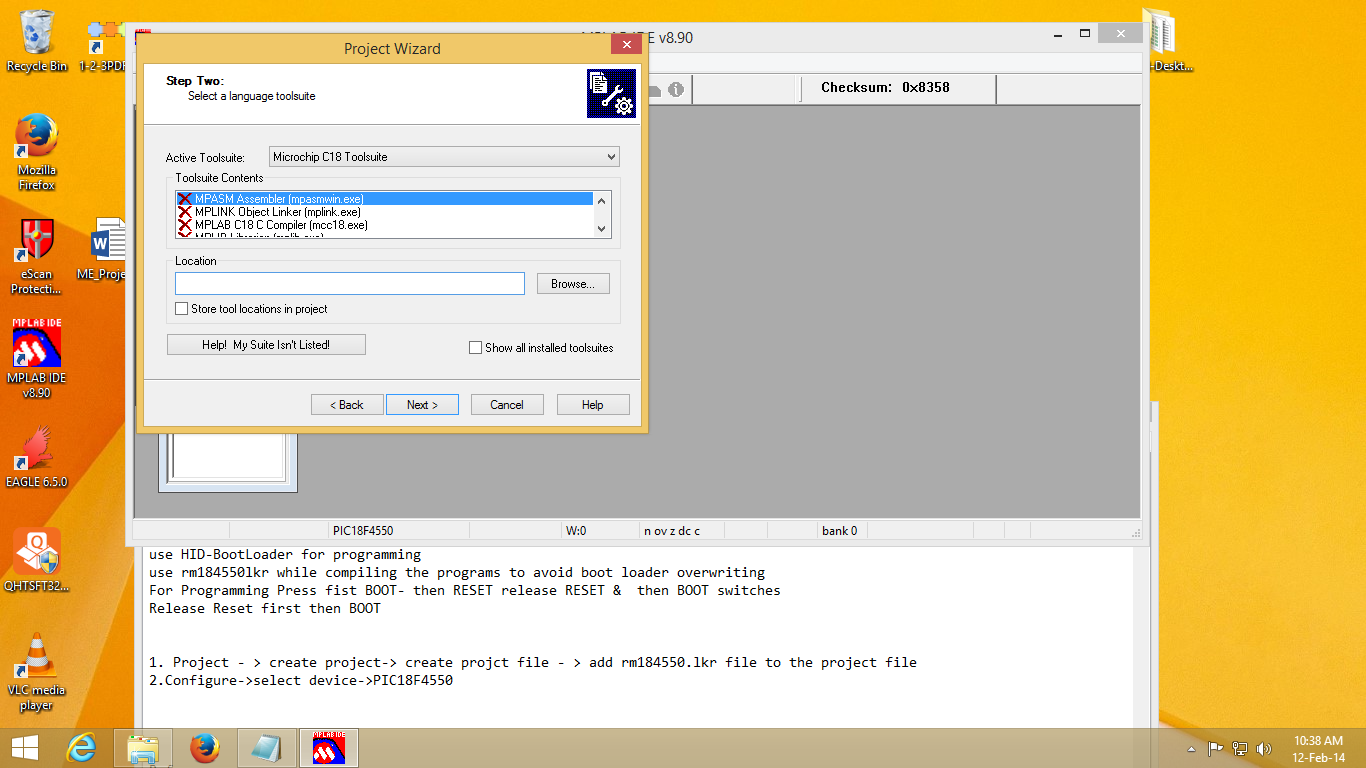




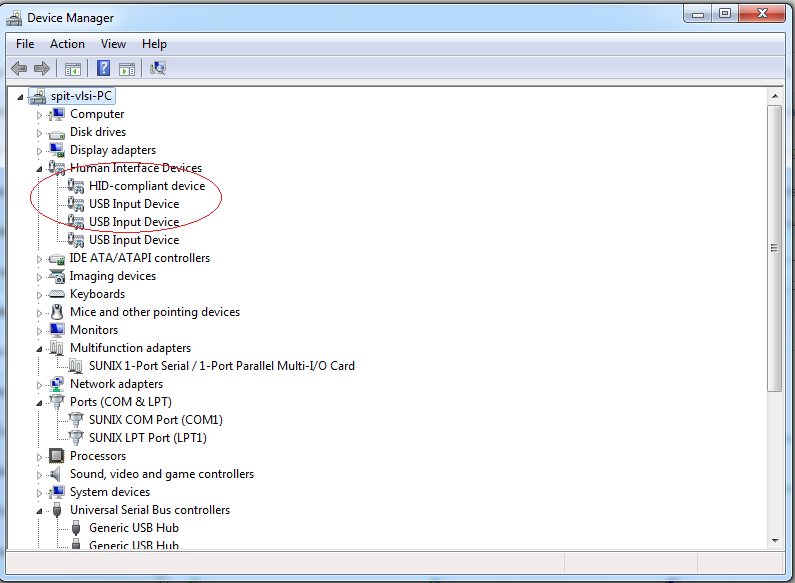
1. Create example folder
2. add rm184550.lkr file to the project file (use rm184550lkr while compiling the programs to avoid boot loader overwriting



1. add source file (.c)
2. locate header files: Project -> built options -> project->
3. show directories for -> include search path -> New : C:\Program Files\Microchip\mplabc18\v3.46\h
4. Library search path path -> New : C:\Program Files\Microchip\mplabc18\v3.46\lib
5. Linker search path-> C:\Program Files\Microchip\mplabc18\v3.46\bin\LKR
6. configure -> select device->PIC18F4550
7. Debugger -> select tool-> MPLABSIM (for simulations)
8. Build the file.
9. use HID-BootLoader for programming
10. For Programming Press fist BOOT- then RESET release RESET & then BOOT switches

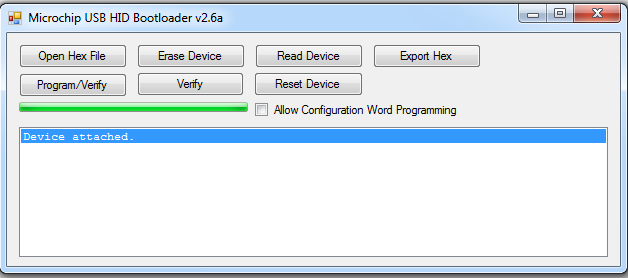


1. **In case the board is not detected verify : In Windows -> Control Panel -> Device Manager->**



16: Open Hex File -> select .HEX file -> Program/Verify-> Reset Device

Repeat this step every time.



Reference:

http://www.microchip.com/stellent/idcplg?IdcService=SS\_GET\_PAGE&nodeId=1406&dDocName=en010014

For Existing Project:

1. to modify device: Configure-> Select Device

PICKIT3: Start MPLAB IDE: Directly load Hex File

1. Programmer-> Select PICKIT3->Debugger-> Connect it
2. File-> import-> LED.Hex file
3. Programmer-> Program

PICKIT3: Start MPLAB IDE : HID bootloader

1. Programmer-> Select PICKIT3->Debugger-> Connect it
2. File-> import-> USB Device - HID - HID Bootloader - C18 - PIC18F4550.Hex file
3. Programmer-> Program

To automatically load Program from Pickit for mass Programming: repeat above steps except step 3

3.Automatic Programmer-> Settings-> Programmer to go -> Send Image in Memory (Select)

For PIC & MPLAB videos:

-------------------------------------

https://www.youtube.com/watch?v=aMK8zkWn7I8&list=PL0F54805E3165D763

**Example Program: LED Blink**

#include <p18f4550.h>

/\*The following lines of code perform interrupt vector relocation to work with the USB bootloader. These must be

used with every application program to run as a USB application.\*/

extern void \_startup (void);

#pragma code \_RESET\_INTERRUPT\_VECTOR = 0x1000

void \_reset (void)

{

\_asm goto \_startup \_endasm

}

#pragma code

#pragma code \_HIGH\_INTERRUPT\_VECTOR = 0x1008

void high\_ISR (void)

{

}

#pragma code

#pragma code \_LOW\_INTERRUPT\_VECTOR = 0x1018

void low\_ISR (void)

{

}

#pragma code

/\*End of interrupt vector relocation\*/

/\*Start of main program\*/

void myMsDelay (unsigned int time)

{

unsigned int i, j;

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

for (j = 0; j < 710; j++);/\*Calibrated for a 1 ms delay in MPLAB\*/

}

void main()

{

TRISA = 0;

while(1)

{

LATA = 0x30;/\*Toggling Port A.4, PORTA.5 pins\*/

myMsDelay(1000);

LATA = 0x00;

myMsDelay(1000);

}

}

**Example Program: Mixed “C” and Assembly Program**

//MPLAB C18 Microchip Inline Assembly

// Page 18 (51288a.pdf)

#include <p18f4550.h>

/\*The following lines of code perform interrupt vector relocation to work with the USB bootloader. These must be

used with every application program to run as a USB application.\*/

extern void \_startup (void);

extern void asm\_delay (void);

#pragma code \_RESET\_INTERRUPT\_VECTOR = 0x1000

void \_reset (void)

{

\_asm goto \_startup \_endasm

}

#pragma code

#pragma code \_HIGH\_INTERRUPT\_VECTOR = 0x1008

void high\_ISR (void)

{

}

#pragma code

#pragma code \_LOW\_INTERRUPT\_VECTOR = 0x1018

void low\_ISR (void)

{

}

#pragma code

/\*End of interrupt vector relocation\*/

/\*Start of main program\*/

void main()

{

unsigned int count =0000;

\_asm

/\* User assembly code \*/

CLRF LATA, ACCESS

CLRF TRISA, ACCESS

again:

MOVLW 0xFF

MOVWF LATA, ACCESS

CALL delay, BANKED

MOVLW 0x00

MOVWF LATA, ACCESS

CALL delay, BANKED

\_endasm

\_asm

delay:

MOVWF count, 0

loop1:

DECFSZ count, 1, 0

GOTO loop1

RETURN BANKED

\_endasm

while(1)

{}

}

**Example Program: Assembly Program**

#include<p18f4450.inc>

CONFIG WDT=OFF; disable watchdog timer

CONFIG MCLRE = ON; MCLEAR Pin on

CONFIG DEBUG = ON; Enable Debug Mode

CONFIG LVP = OFF; Low-Voltage programming disabled (necessary for debugging)

CONFIG FOSC = INTOSCIO\_EC;Internal oscillator, port function on RA6

org 0; start code at 0

Delay1 res 2 ;reserve 2 byte for the variable Delay1

Start:

CLRF LATD

CLRF TRISD

CLRF Delay1

MainLoop:

movlw 0xff ;

movwf LATD

Delay11:

DECFSZ Delay1,1 ;Decrement Delay1 by 1, skip next instruction if Delay1 is 0

GOTO Delay11

movlw 0x00 ;

movwf LATD

Delay22:

DECFSZ Delay1,1 ;Decrement Delay1 by 1, skip next instruction if Delay1 is 0

GOTO Delay22

GOTO MainLoop

end