Practice 5. Introduction to using MPLAB X IDE

Objectives

The student will become familiar with the MPLAB integrated development environment tools. The student will create his (her) first program using the MPLAB software and will put it into the PIC microcontroller making use of the Curiosity development/evaluation board.

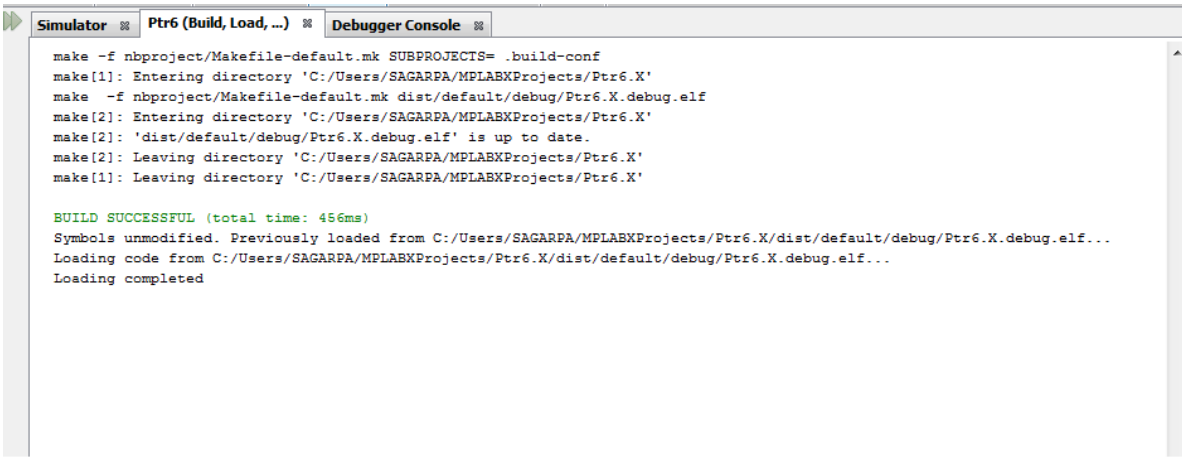
Introduction

Microcontrollers are all around he world. Each day, Microcontrollers, are more present in the many aspects of our lives: in our work, inside our houses, and in more. We can find them controlling small devices like cellphones, microwaves, washing machines, and televisions.

A microcontroller is one device or chip that is used to govern one or more processes. For example, the controller that regulates the room temperature of an air conditioner; it has a sensor that continuously measures the internal temperature and, when the preset limits are exceeded, it generates the necessary signals to adjust the temperature

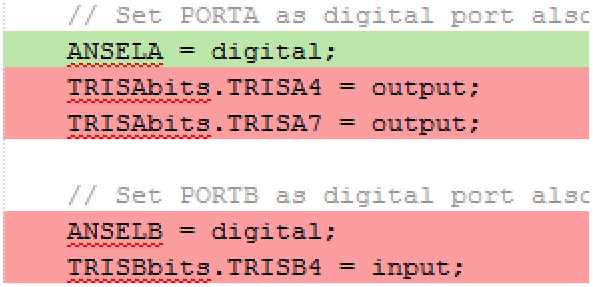
Results

Image of the successful build:

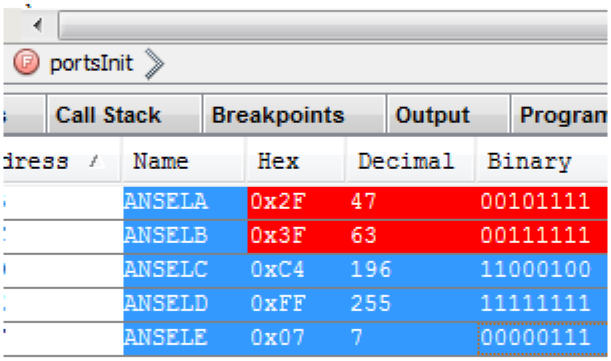


```
make -f nbproject/Makefile-default.mk SUBPROJECTS= .build-conf
make[1]: Entering directory 'C:/Users/SAGARPA/MPLABXProjects/Ptr6.X'
make  -f nbproject/Makefile-default.mk dist/default/debug/Ptr6.X.debug.elf
make[2]: Entering directory 'C:/Users/SAGARPA/MPLABXProjects/Ptr6.X'
make[2]: 'dist/default/debug/Ptr6.X.debug.elf' is up to date.
make[2]: Leaving directory 'C:/Users/SAGARPA/MPLABXProjects/Ptr6.X'
make[1]: Leaving directory 'C:/Users/SAGARPA/MPLABXProjects/Ptr6.X'

BUILD SUCCESSFUL (total time: 456ms)
Symbols unmodified. Previously loaded from C:/Users/SAGARPA/MPLABXProjects/Ptr6.X/dist/default/debug/Ptr6.X.debug.elf...
Loading code from C:/Users/SAGARPA/MPLABXProjects/Ptr6.X/dist/default/debug/Ptr6.X.debug.elf...
Loading completed
```

Image of Ports registers initialization:

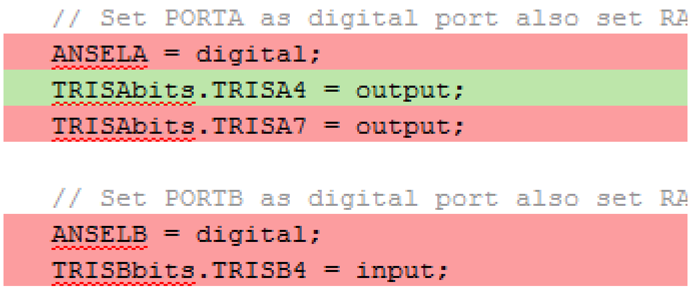


```
// Set PORTA as digital port alsc
ANSELA = digital;
TRISAbits.TRISA4 = output;
TRISAbits.TRISA7 = output;

// Set PORTB as digital port alsc
ANSELB = digital;
TRISBbits.TRISB4 = input;
```

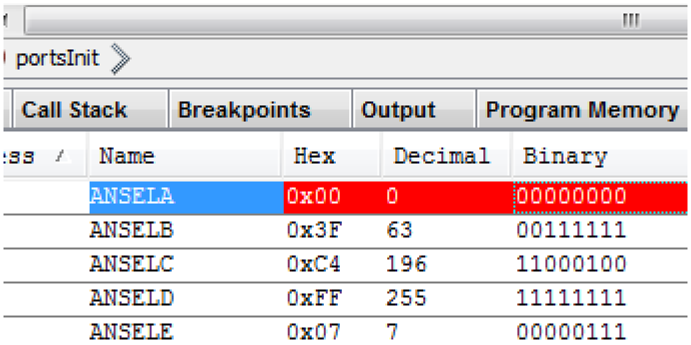


portsInit

dress /	Name	Hex	Decimal	Binary
	ANSELA	0x2F	47	00101111
	ANSELB	0x3F	63	00111111
	ANSELC	0xC4	196	11000100
	ANSELD	0xFF	255	11111111
	ANSELE	0x07	7	00000111

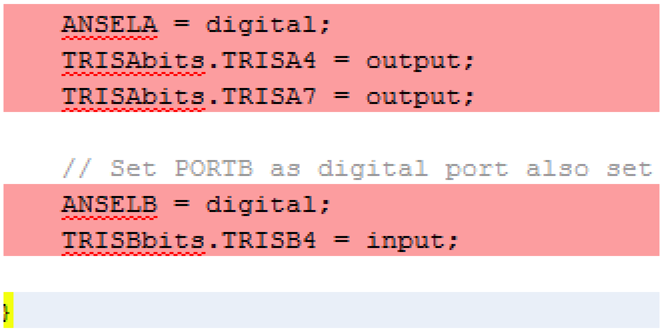


```
// Set PORTA as digital port also set RA
ANSELA = digital;
TRISAbits.TRISA4 = output;
TRISAbits.TRISA7 = output;

// Set PORTB as digital port also set RA
ANSELB = digital;
TRISBbits.TRISB4 = input;
```

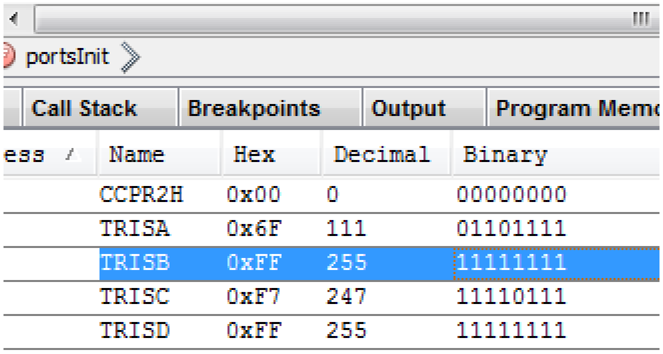


portsInit

ss /	Name	Hex	Decimal	Binary
	ANSELA	0x00	0	00000000
	ANSELB	0x3F	63	00111111
	ANSELC	0xC4	196	11000100
	ANSELD	0xFF	255	11111111
	ANSELE	0x07	7	00000111

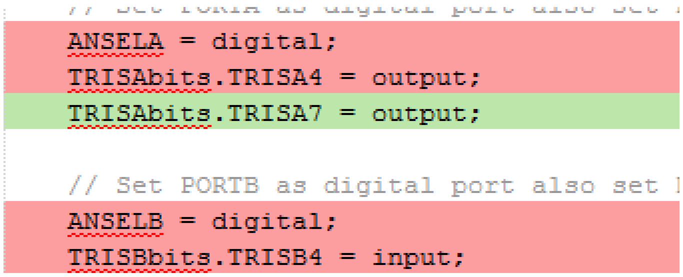


```
ANSELA = digital;
TRISAbits.TRISA4 = output;
TRISAbits.TRISA7 = output;

// Set PORTB as digital port also set
ANSELB = digital;
TRISBbits.TRISB4 = input;

}
```

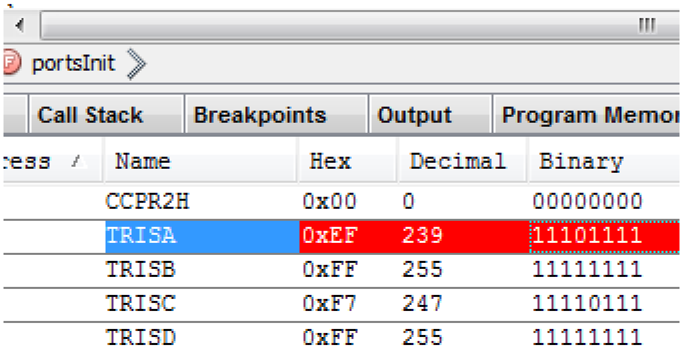


portsInit

ess /	Name	Hex	Decimal	Binary
	CCPR2H	0x00	0	00000000
	TRISA	0x6F	111	01101111
	TRISB	0xFF	255	11111111
	TRISC	0xF7	247	11110111
	TRISD	0xFF	255	11111111

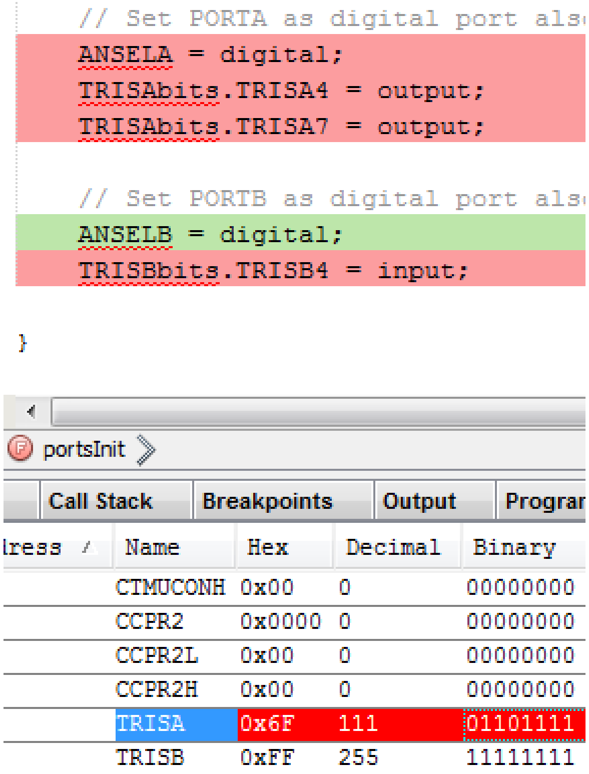


```
ANSELA = digital;
TRISAbits.TRISA4 = output;
TRISAbits.TRISA7 = output;

// Set PORTB as digital port also set I
ANSELB = digital;
TRISBbits.TRISB4 = input;
```



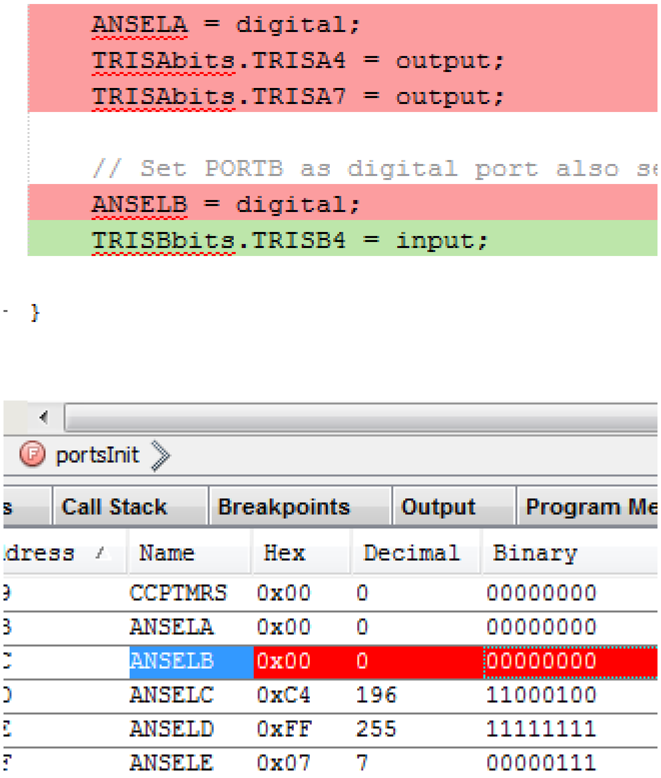
portsInit

ress /	Name	Hex	Decimal	Binary
	CCPR2H	0x00	0	00000000
	TRISA	0xEF	239	11101111
	TRISB	0xFF	255	11111111
	TRISC	0xF7	247	11110111
	TRISD	0xFF	255	11111111



```
    // Set PORTA as digital port als
    ANSELA = digital;
    TRISAbits.TRISA4 = output;
    TRISAbits.TRISA7 = output;

    // Set PORTB as digital port als
    ANSELB = digital;
    TRISBbits.TRISB4 = input;

}
```

Iress	Name	Hex	Decimal	Binary
	CTMUCONH	0x00	0	00000000
	CCPR2	0x0000	0	00000000
	CCPR2L	0x00	0	00000000
	CCPR2H	0x00	0	00000000
	TRISA	0x6F	111	01101111
	TRISB	0xFF	255	11111111



```
    ANSELA = digital;
    TRISAbits.TRISA4 = output;
    TRISAbits.TRISA7 = output;

    // Set PORTB as digital port also se
    ANSELB = digital;
    TRISBbits.TRISB4 = input;

}
```

dress	Name	Hex	Decimal	Binary
9	CCPTMRS	0x00	0	00000000
3	ANSELA	0x00	0	00000000
C	ANSELB	0x00	0	00000000
D	ANSELC	0xC4	196	11000100
E	ANSELD	0xFF	255	11111111
F	ANSELE	0x07	7	00000111

Link of demonstration: https://youtu.be/dra_E4VN1-w

Conclusion:

With this practice it is clarifies the way to use the debug and the SFR memory view to track how the program variables are changing and registers being modified, which despite of the simplicity of this exercise lead to the use of an indispensable tool for programming the micron roller, because it is easier to see directly how the code behaves than trying to find a bug or mistake by reading the code, compiling and executing it to remember the differences between each run after.