Swapping between Hex and Integers

by Martin Whitlock, May 2011

Abstract

In this article Martin describes his program which converts hexadecimal values into decimal ones, as well as explaining the program and the reasoning behind some of his decisions. The program itself accompanies the article.

Requirements

Software:

- Any licence of Flowcode v3 or v4 for any chip variant.

Hardware:

- No hardware required.

Introduction:

This article will describe the methodology used to create a Flowcode program which converts hexa-decimal values into decimal ones, and visa versa. The program has been created in Flowcode 3, so it will work in both versions and uses common variables to keep the ROM usage down. Both these conversions are done in their own separate independent macros, so using the Flowcode macro tools you can **export** and **import** these macros into other programs.
One potential application is receiving hex via RS232/USB.

Here is how the conversion works:

From Integer to Hex:

e.g 1234

- Integer Value is converted to a string using: `ToString$`
(*This is a built in function).*

- *Int_Var* uses MOD 16 to store the remainder of 1234/16 = 2
(1234/16=77 r 2)

- *Int_Conversion* stores 1234

The remainder is the least significant hex value.
="2"
Hex_Val1="2"

- Next the 77 is divided by 16:
77/16 = 4 r 13
Using decision branches 13 = "D"
Hex_Val1="2D"

- Next the 4 is divided by 16:
4/16 = 0 r 4
="4"
Hex_Val1="2D4"

This process stops as soon as whole number after division = 0 which is now!

Finally a string reversing routine changes "2D4" to "4D2" = *Hex_Val2*

Conversion of Hexadecimal to Integer:

This is a smaller routine. Each value from the least significant to the most significant is used one at a time. Taking a hex value of **4D2** for example. When you use a string with a character within square brackets within a calculation box, then the ASCII value of the character is assigned to the string variable.

Using calculation for 4D2:

```
Int_Var = Hex_Val1[Index]
```

Index is the x position of each Char. E.g the index for 4 = 0, index for D = 1 and index for 2 = 2

We are working backwards so using Hex value of 2 first.

ASCII of 2 = 50

There is a discussion branch which states if this ASCII value < 64 then

```
Int_Var = Int_Var - 48
```

This is because a number from 0 to 9(ASCII 48 to 57) is used. So 48 must be subtracted from the ASCII value.

50 - 48 = 2

ASCII of D = 68
if this ASCII value >64 then

```
Int_Var = Int_Var - 55
```

This is because an upper case letter is used (A to F = ASCII 65 to 70) so 55 must be subtracted from the ASCII value.

68 - 55 = 13
ASCII of 4 = 52

Again using decision if this ASCII value < 64

52 - 48 = 4

Using the formula: Int value = 2*(16^0) + 13*(16^1) + 4*(16^2)

= 2 + 208 + 1024
= 1234

This is easy to achieve though bit shifting i.e <<1 = bit shift all by left by 1 place this is the same as multiplying by 2^1 = 2. <<2 = 2^2 = 4. <<3 = 2^3 = 8 ect.

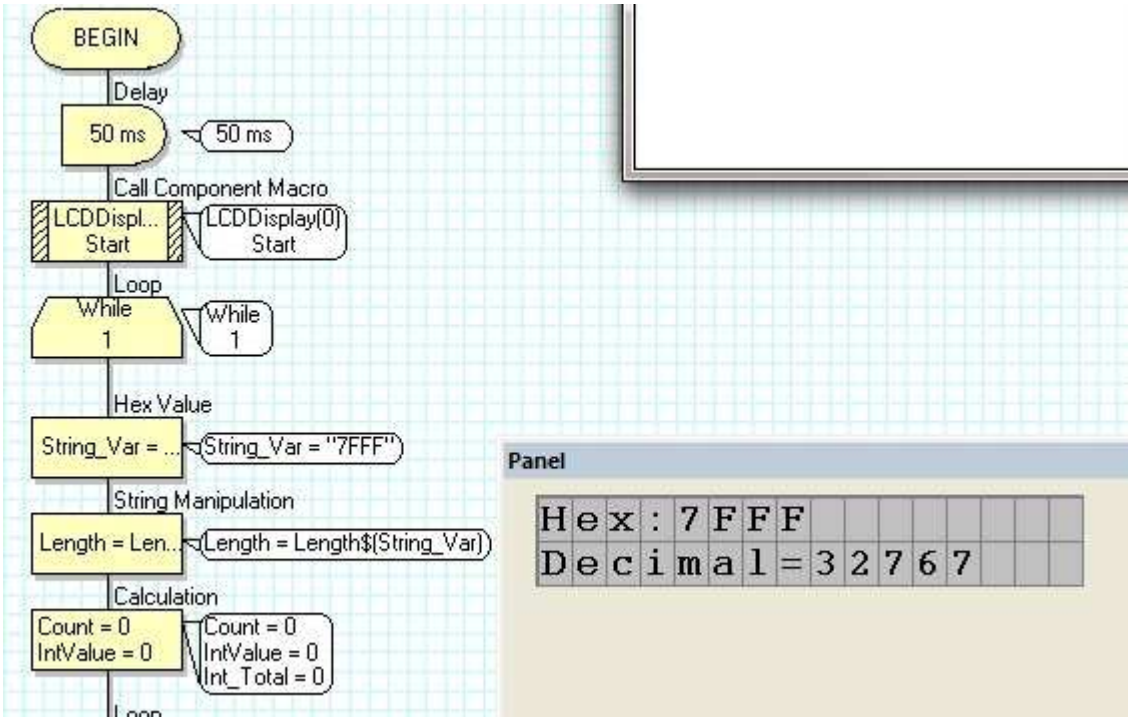
Applying this in Flowcode I used:

```
Int_Value = Int_Var << Count * 4
```

The count starts at 0 and increments by 1 until all digits are added up.
For the first digit (4) count = 0, for the second digit (13) Count = 1 and for the third digit (2) count = 2
To understand why I used << Count * 4
Basically <<4 = 16

When *int_Var* = 2 and count = 0, 2<< 0*4 = 2<<0 = 2^0 = 2
When *int_Var* = 13 and count = 1, 13<< 1*4 = 13<<4 = 13*2^4 = 13*16 = 208
When *int_Var* = 4 and count = 2, 4<< 2*4 = 2<<8 = 4*2^8 = 4*256 = 1024

I hope this article helps you to understand the processes involved when converting Integer to Hexadecimal and Hexadecimal to Integer .



Further reading

Below are some links to other resources and articles on related subjects, and technical documentation relating to the hardware used for this project...

Flowcode:	http://www.matrixmultimedia.com/flowcode.php
Learning Centre:	http://www.matrixmultimedia.com/lc_index.php
User Forums:	http://www.matrixmultimedia.com/mmforums
Product Support:	http://www.matrixmultimedia.com/sup_menu.php