Subject: Re: Constructing integer variables from two bytes?
Posted by Lajos Foldy on Wed, 07 Dec 2011 12:52:35 GMT
View Forum Message <> Reply to Message

On Dec 7, 6:46 am, Jared Espley <jesp...@gmail.com> wrote:
> I have data in a byte array that I would like to use to construct
> arrays of unsigned integers and integers.
>
> In code:
> A = [1B, 2B]
> B = 0
> B = A[0]*256 + fix(A[1])
>
> Which will give B as an integer type with a value 258.
>
> I realize that I can construct functions to do this (multiple each bit
> by the appropriate value and checking which OS we're in so that we get
> endianness correct) but I was wondering if someone had already done
> this rigorously?
>
> Thanks,
> Jared

FIX can combine the bytes into integers (in host byte order, use
SWAP_ENDIAN if the source byte order is different):

```
IDL> b2=fix(a, 0, 1, type=2)
IDL> help, b2
B2              INT       = Array[1]
IDL> print, b2
     513
IDL> print, swap_endian(b2)
     258
```

regards,
Lajos

Subject: Re: Constructing integer variables from two bytes?
Posted by David Fanning on Wed, 07 Dec 2011 13:19:39 GMT
View Forum Message <> Reply to Message

Lajos Foldy writes:

> FIX can combine the bytes into integers (in host byte order, use
> SWAP_ENDIAN if the source byte order is different):
>
> IDL> b2=fix(a, 0, 1, type=2)

> IDL> help, b2
> B2   INT   = Array[1]
> IDL> print, b2
>   513
> IDL> print, swap_endian(b2)
>   258

You may have missed this in Lejos's explanation, but all of these "casting" functions (e.g., Fix, Long, Float, etc.) can extract data from byte arrays in this fashion. And not just one value at a time, but all the values at once. The trick is to use the offset parameter to tell the function how may byte values to "grab" when it does the conversion.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Constructing integer variables from two bytes?
Posted by greg.addr on Wed, 07 Dec 2011 16:32:17 GMT
View Forum Message <> Reply to Message

Yes - this is useful - I wish I'd known about it a long time ago!

IDL> a=[1b,2b,2b,3b,4b,5b]
IDL> b2=fix(a, 0, 3, type=2)
IDL> print,b2
    513   770   1284

Seems to me, though, that offset should be 0 (unless you want to skip bytes), and you have to figure out the final number of integers yourself. However, it would have made more sense if it worked the way David said... specifying to grab 2 bytes at a time.

While on the topic, is there a similarly direct way to convert bytes into an arbitrary structure? To convert the above array into two of these {b:0b,i:0} for example?

cheers,
Greg
