
Subject: Re: Assembly - Convert 16 bit integer to ascii help please
Posted by [Anonymous](#) on Sun, 13 Jan 2013 14:47:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Originally posted by: <fr>

Oops, adding the spceial 0 case :-)

```
> Hi, here is an example for such a modification.
> Setting Y to 0 before calling the conversion will remove the leading
> zeros,
> setting it to 1 will keep them...
> Let me know if it works :-)
> Frederic.
>

> news:-6CdncKa27vIRWzNnZ2dnUVZ_h-dnZ2d@earthlink.com...
>>
>> "JB" <jbrown1289@gmail.com> wrote in message
>> news:7bcec11b-22cd-4d84-b19d-9fc272896e4a@googlegroups.com...
>>> I am looking for a routine to convert a 16 bit integer into ascii (text)
>>>
>>> ex. convert $C000 (49152) to the characters "49152" stored in a memory
>>> address.
>>>
>>> I found one that almost does what I want, but it adds leading 0's.
>>>
>>> Ideally I would like to have a routine that you can pass an option to
>>> either print leading 0's or strip them. But first things first I guess.
>>> :)
>>>
>>>
>>> Here is that code:
>>>
>>> .*
>>> ;
>>> ;* DECIMAL TO ASCII ROUTINE *
>>> ;* LOW-BYTE IN .X HI-BYTE IN .A
>>> ;* STORES ASCII STRING IN MEMORY
> ;* Y=0 REMOVE LEADING 0'S
> ;* Y=1 PRINTS LEADING 0'S
>>>
>>> DECIMAL STX BINARY
>>> STA BINARY+1
> STY DIGITS
>>> LDY #0
>>> DEC1 LDX #"0"
>>> DEC2 LDA BINARY
```

```

>>> CMP DECTBL1,Y
>>> LDA BINARY+1
>>> SBC DECTBL2,Y
>>> BCC DEC3
>>> STA BINARY+1
>>> LDA BINARY
>>> SBC DECTBL1,Y
>>> STA BINARY
>>> INX
>>> BNE DEC2
>>> DEC3 TXA
> BNE DEC4
CPY #4
BEQ DEC4
> LDX DIGITS
> BEQ DEC5
> DEC4 INC DIGITS
>>> STA DECCHR,Y
>>> JSR $FFD2
> DEC5 INY
>>> CPY #5
>>> BNE DEC1
>>>     RTS
>>>
> DIGITS .BYTE 0
>>>
>>> DECTBL1 .BYTE <10000,<1000,<100
>>> .BYTE <10,<1
>>> DECTBL2 .BYTE >10000,>1000,>100
>>> .BYTE >10,>1
>>>
>>> DECCHR .WORD 0,0
>>> BINARY .WORD 0
>>
>> Modify this routine so that it doesn't start converting until the number
>> being converted is larger than the value in the tables.
>>
>> For example, if the value being converted is, say, 677, the values of
>> 10000 and 1000 are larger. When the Y-register value is zero or one, all
>> that should happen is the Y-register is increased by one so the next
>> table
>> value can be checked. When the index value reaches two, the table value
>> of 100 is less than 677. Start converting at that point.
>>
>> This modification might also require a special check that the converted
>> value is not zero, because then all table values are larger. If it is
>> zero, just output "0" and exit.
>>

```

>> - Anton Treuenfels

>>

>

>

>

>

>