

---

Subject: Assembly - Convert 16 bit integer to ascii help please

Posted by [JB](#) on Sat, 12 Jan 2013 15:56:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

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  
* PRINTS PRECEDING 0'S
```

```
DECIMAL STX BINARY
```

```
  STA BINARY+1
```

```
  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
```

```
  STA DECCHR,Y
```

```
  JSR $FFD2
```

```
  INY
```

```
  CPY #5
```

```
  BNE DEC1
```

```
  RTS
```

```
DECTBL1 .BYTE <10000,<1000,<100
```

```
.BYTE <10,<1
DECTBL2 .BYTE >10000,>1000,>100
.BYTE >10,>1
```

```
DECCHR .WORD 0,0
BINARY .WORD 0
```

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please  
Posted by [Anton Treuenfels](#) on Sat, 12 Jan 2013 21:41:03 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

```
"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
> ;* PRINTS PRECEDING 0'S
>
> DECIMAL STX BINARY
> STA BINARY+1
> 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
> STA DECCHR,Y
> JSR $FFD2
> INY
> CPY #5
> BNE DEC1
>     RTS
>
>
> 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

Subject: Re: Assembly - Convert 16 bit integer to ascii help please

Posted by [rusure](#) on Sun, 13 Jan 2013 02:14:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Saturday, January 12, 2013 8:56:32 AM UTC-7, JB wrote:

> 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.

>

> 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. :)

A C64 systems routine at \$bdcd prints the decimal equivalent of a two byte integer. Here is how to use that code:

```
lda #$0d
jsr $ffd2
```

```
lda #$c0
ldx #$00
jsr $bdcd
rts
```

for the C128, substitute \$8e32 at bank 15 for \$bdcd. I've never had to do it but I'm sure that buried in the routine at \$bdcd is a routine for converting 4 nibble hexadecimal to decimal text digits. I think the routines suppresses leading zeroes.

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please

Posted by [Groepaz](#) on Sun, 13 Jan 2013 07:12:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

rusure wrote:

> On Saturday, January 12, 2013 8:56:32 AM UTC-7, JB wrote:

>> 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.

>>

>> 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.

>> :)

>

> A C64 systems routine at \$bdcd prints the decimal equivalent of a two byte  
> integer. Here is how to use that code:

its not a very good idea to use direct calls into basic rom though =P

--

<http://www.hitmen-console.org> <http://magicdisk.untergrund.net>

<http://www.pokefinder.org> <http://ftp.pokefinder.org>

Every program has at least one bug and can be shortened by at least one instruction --- from which, by induction, one can deduce that every program can be reduced to one instruction which doesn't work.

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please

Posted by [Anonymous](#) on Sun, 13 Jan 2013 14:38:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Originally posted by: <fr>

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
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
>

```

---

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

Originally posted by: <fr>

Oops, adding the special 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
>>
>

```



>  
>  
>

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please

Posted by [JB](#) on Sun, 13 Jan 2013 15:26:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Yeah I know about \$BDCD but that will only print the number not store it in a memory location. I did break that routine down and can make it work, but basic rom has to be switched in for it to work. I'd like to have a solution that will run regardless if basic is in or out.

On Saturday, January 12, 2013 9:14:39 PM UTC-5, rusure wrote:

> On Saturday, January 12, 2013 8:56:32 AM UTC-7, JB wrote:

>

>> 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.

>

>>

>

>> 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. :)

>

>

>

> A C64 systems routine at \$bdcd prints the decimal equivalent of a two byte integer. Here is how to use that code:

>

>

>

> lda \$#0d

>

> jsr \$ffd2

>

>

>

> lda #\$c0

>

> idx #\$00

>  
> jsr \$bdcd  
>  
> rts  
>  
>  
>  
> for the C128, substitute \$8e32 at bank 15 for \$bdcd. I've never had to do it but I'm sure that buried in the routine at \$bdcd is a routine for converting 4 nibble hexadecimal to decimal text digits. I think the routines suppresses leading zeroes.

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please  
Posted by [JB](#) on Sun, 13 Jan 2013 15:28:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

FR:

where does the CPY #4 and BEQ DEC4 go in the code?

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please  
Posted by [Anonymous](#) on Sun, 13 Jan 2013 16:42:22 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Originally posted by: <fr>

after BNE DEC4... let me remove the ">" :

```
;* DECIMAL TO ASCII ROUTINE *  
;* LOW-BYTE IN .X HI-BYTE IN .A  
;* STORES ASCII STRING IN MEMORY  
;* Y=0 REMOVES 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
```

f8a4f3fb-66d5-4192-a36b-df6d4d881b95@googlegroups.com...

> FR:

>

> where does the CPY #4 and BEQ DEC4 go in the code?

>

>

>

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please

Posted by [rusure](#) on Mon, 14 Jan 2013 02:20:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Sunday, January 13, 2013 12:12:30 AM UTC-7, Groepaz wrote:

> rusure wrote:

>> On Saturday, January 12, 2013 8:56:32 AM UTC-7, JB wrote:

>>> 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.  
>>> 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.  
>>> :)  
>  
>> A C64 systems routine at \$bdcd prints the decimal equivalent of a two byte  
>> integer. Here is how to use that code:  
>  
> its not a very good idea to use direct calls into basic rom though =P

Jim Butterfield agrees with you, but I can't figure out why. I first saw \$bdcd used in a reverse assembly of a program in COMPUTE. It took me less than an hour to figure out what the call was doing. I probably coded a call to it or \$8e32 on a C128 about 10 times without any difficulty. The program overwrites the contents of the accumulator and index registers.

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please  
Posted by [rusure](#) on Mon, 14 Jan 2013 03:01:59 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Sunday, January 13, 2013 8:26:56 AM UTC-7, JB wrote:  
> Yeah I know about \$BDCD but that will only print the number not store it in a memory location. I did break that routine down and can make it work, but basic rom has to be switched in for it to work. I'd like to have a solution that will run regardless if basic is in or out.

You decided that it would be easier to write your own routine from scratch rather than fuss with \$BDCD using a C64 machine language monitor. Right now, you don't know where the routine converts the binary to text decimal as well as where it stores the text. Because the C128 has a hard wired monitor in its system, I found the decimal text in the stack at \$0100. That would be the first place to look in the C64.

My programs are simple and short enough to peacefully coexist with BASIC.

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please  
Posted by [JB](#) on Mon, 14 Jan 2013 03:35:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I have been writing a BBS, so 99% of my ML is under basic. Not only does my routine need to print the #, but it needs to store it and print through the modem as well. Pretty tough to do with the basic routine.

On Sunday, January 13, 2013 10:01:59 PM UTC-5, rusure wrote:  
> On Sunday, January 13, 2013 8:26:56 AM UTC-7, JB wrote:  
>

>> Yeah I know about \$BDCD but that will only print the number not store it in a memory location. I did break that routine down and can make it work, but basic rom has to be switched in for it to work. I'd like to have a solution that will run regardless if basic is in or out.

>  
>  
>

> You decided that it would be easier to write your own routine from scratch rather than fuss with \$BDCD using a C64 machine language monitor. Right now, you don't know where the routine converts the binary to text decimal as well as where it stores the text. Because the C128 has a hard wired monitor in its system, I found the decimal text in the stack at \$0100. That would be the first place to look in the C64.

>  
>  
>

> My programs are simple and short enough to peacefully coexist with BASIC.

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please

Posted by [Groepaz](#) on Mon, 14 Jan 2013 10:20:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

rusure wrote:

> On Sunday, January 13, 2013 12:12:30 AM UTC-7, Groepaz wrote:

>> rusure wrote:

>>> On Saturday, January 12, 2013 8:56:32 AM UTC-7, JB wrote:

>>>> 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.

>>>> 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.

>>>> :)

>>

>>> A C64 systems routine at \$bdcd prints the decimal equivalent of a two

>>> byte

>>> integer. Here is how to use that code:

>>

>> its not a very good idea to use direct calls into basic rom though =P

>

> Jim Butterfield agrees with you, but I can't figure out why.

because you should generally see rom code as a "black box" and only ever use the official entry points (the kernal jumtable). that way your program will always work and doesnt need special case handling for when a modified basic

rom and/or kernal is present.

--

<http://www.hitmen-console.org> <http://magicdisk.untergrund.net>  
<http://www.pokefinder.org> <http://ftp.pokefinder.org>

Wenn eine vom Kopierräuber geraubte Kopie eine Raubkopie ist, dann ist ein vom Kopiererräuber geraubter Kopierer ein Raubkopierer.

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please

Posted by [rusure](#) on Mon, 14 Jan 2013 15:37:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Sunday, January 13, 2013 8:35:55 PM UTC-7, JB wrote:

> I have been writing a BBS, so 99% of my ML is under basic. Not only does my routine need to print the #, but it needs to store it and print through the modem as well. Pretty tough to do with the basic routine.

>

>

>

> On Sunday, January 13, 2013 10:01:59 PM UTC-5, rusure wrote:

>

>> On Sunday, January 13, 2013 8:26:56 AM UTC-7, JB wrote:

>

>>

>

>>> Yeah I know about \$BDCD but that will only print the number not store it in a memory location. I did break that routine down and can make it work, but basic rom has to be switched in for it to work. I'd like to have a solution that will run regardless if basic is in or out.

>

>>

>

>>

>

>>

>

>> You decided that it would be easier to write your own routine from scratch rather than fuss with \$BDCD using a C64 machine language monitor. Right now, you don't know where the routine converts the binary to text decimal as well as where it stores the text. Because the C128 has a hard wired monitor in its system, I found the decimal text in the stack at \$0100. That would be the first place to look in the C64.

>

>>

>

>>

>

>>

>

>> My programs are simple and short enough to peacefully coexist with BASIC. In addition, my modest programs are usually intended to replace sluggish sections of magazine type in BASIC programs. Usually, I retain much of the original BASIC code. For this application, I need BASIC ROM. About the only disadvantage to this programming practice is that my programs only work on one COMMODORE computer. If I can at all help it, I program for the C128. Programming for the C64 is like pulling teeth.

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please

Posted by [Anonymous](#) on Fri, 18 Jan 2013 19:30:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Originally posted by: <fr>

I finally found some time to test it with an emulator, here is an updated code.

Frederic

```
.ORG $C000
LDA #$04
LDX #$00
LDY #$01

;* DECIMAL TO ASCII ROUTINE *
;* LOW-BYTE IN .X HI-BYTE IN .A
;* STORES ASCII STRING IN MEMORY
;* Y=0 REMOVES 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
    STA DECCHR,Y
    BNE DEC4
    CPY #4
    BEQ DEC4
    LDX DIGITS
    BEQ DEC5
DEC4:
    INC DIGITS
    CLC
    ADC #$30 ;* ASCII CODE FOR 0
    JSR $FFD2
DEC5:
    INY
    CPY #5
    BNE DEC1
    RTS

DECTBL1:
    .BYTE <10000,<1000,<100,<10,<1
DECTBL2:
    .BYTE >10000,>1000,>100,>10,>1
BINARY:
    .WORD 0
DECCHR:
    .BYTE 0,0,0,0,0
DIGITS:
    .BYTE 0 ;* NUMBER OF DIGITS IF Y=0
```

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please  
Posted by [Anonymous](#) on Sat, 01 Jun 2013 02:24:57 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Originally posted by: The Doctor

On Sat, 12 Jan 2013 07:56:32 -0800 (PST), JB <[jbrown1289@gmail.com](mailto:jbrown1289@gmail.com)>  
wrote:

```
> 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
> ;* PRINTS PRECEDING 0'S
>
> DECIMAL STX BINARY
> STA BINARY+1
> 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
> STA DECCHR,Y
> JSR $FFD2
> INY
> CPY #5
> BNE DEC1
>   RTS
>
>
> DECTBL1 .BYTE <10000,<1000,<100
> .BYTE <10,<1
> DECTBL2 .BYTE >10000,>1000,>100
> .BYTE >10,>1
>
> DECCHR .WORD 0,0
> BINARY .WORD 0

```

I used to do a lot of deep assembly on the 64/1541. I was always

looking for speed and the way to get that is by using tables wherever possible. Eliminate the looping and calculating.

It's been a long long time but I have no doubt there are routines to accomplish this sitting on some old 5 1/4 floppys in storage. The question is are the disks good anymore? They haven't been touched in at least 20 years and they have suffered extremes of temperature.

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please  
Posted by [Anonymous](#) on Mon, 03 Jun 2013 09:24:15 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Originally posted by: Johann Klasek - swap last two domainparts

The wrote:

> On Sat, 12 Jan 2013 07:56:32 -0800 (PST), JB <[jbrown1289@gmail.com](mailto:jbrown1289@gmail.com)>  
> wrote:  
>  
>> 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 used to do a lot of deep assembly on the 64/1541. I was always  
> looking for speed and the way to get that is by using tables wherever  
> possible. Eliminate the looping and calculating.  
>  
> It's been a long long time but I have no doubt there are routines to  
> accomplish this sitting on some old 5 1/4 floppys in storage. The  
> question is are the disks good anymore? They haven't been touched in  
> at least 20 years and they have suffered extremes of temperature.

It depends, but my experience so far is that my disks from 1985 to 1992 (very different qualities and types) could be read nearly without problems in 2012. Just disks which have already problems in these old days rendered unreadable in some cases. I'm not sure, but it may help to read the disks with the drive they were written in the past.

So I would encourage you to give them a try.

JeeK

---

---

Subject: Re: Assembly - Convert 16 bit integer to ascii help please  
Posted by [Anonymous](#) on Mon, 03 Jun 2013 09:50:44 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Originally posted by: Johann Klasek - swap last two domainparts

Groepaz <groepaz@gmx.net> wrote:

> rusure wrote:

>> On Sunday, January 13, 2013 12:12:30 AM UTC-7, Groepaz wrote:

>>> rusure wrote:

>>>> On Saturday, January 12, 2013 8:56:32 AM UTC-7, JB wrote:

>>>> > 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.

>>>> > 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.

>>>> > :)

>>>

>>>> A C64 systems routine at \$bdcd prints the decimal equivalent of a two

>>>> byte

>>>> integer. Here is how to use that code:

>>>

>>> its not a very good idea to use direct calls into basic rom though =P

>>

>> Jim Butterfield agrees with you, but I can't figure out why.

>

> because you should generally see rom code as a "black box" and only ever use

> the official entry points (the kernal jumtable). that way your program will

> always work and doesnt need special case handling for when a modified basic

> rom and/or kernal is present.

Yes, that's well known but portability has a complete other value in specific system environments like a C64. As you probably know the BASIC routines are not offered via a jump table or call interface, but it would be far to idealistic and wasteful to omit the valuable code in the ROM and reprogram or reload everything again and again just to hold the blackbox paradigm. Yes, you are right, but practical needs and common habit disobeyes blackboxing. In a C64 environment, even for several different hardware generations most - not to say all - well known and useful BASIC entries did not change through the time.

It's nice to mention the blackbox paradigm, but repeatedly replying on every post with a BASIC ROM routine reference becomes tiresome. It's not the todays programmer who needs this advice, but Microsoft and CBM had the obligation to provide such a interface. Alas, the missed it and the time passes away ...

