

This is a very simple routine that converts an 8-bit integer to the corresponding hexadecimal representation, which is output as a null-terminated string. Multiple calls to the routine can be chained, to convert longer integers. An example is provided to hexify your current IRQ vector (16-bit).

The null-termination is a bit awkward with the iny/dey that gets executed twice even for every call, but the routine must preserve the index, which has to be initialized to either zero or to the length of an existing string if you want to add this output, mimicing sprintf operation.

Alternatively, you could relocate the ldy #0 instruction to the beginning of the hexify routine, if you always only convert single bytes. If you should want to print a 3-digit hex number, increment the dst pointer before calling print, and this will omit the first digit.

Bring in the code for the print routine from [String manipulation routines](#). When you assemble and run the program (with SYS 8192) it will print the hexadecimal address to the IRQ routine that is currently running on your system and it will normally be EA31.

```
; Integer to Hex String conversion routine by FMan/Tropyx

    !to "inttostr.prg",cbm          ; use ACME

; example how to print a 16-bit value in hexadecimal

    dst = $64      ; output string pointer

    *=$2000

    lda #0
    sta dst
    lda #$7f
    sta dst+1
    ldy #0          ; initialize string index
    lda $315
    jsr hexify     ; convert the MSB first
    lda $314
    jsr hexify
    jsr print      ; see note above
    rts

; this routine converts the given value into a two-digit hex code
; inputs: A=byte, Y=index - outputs: to given buffer - preserves Y

hexify tax
    lsr
    lsr
    lsr
    lsr
    jsr hexc      ; convert upper nybble
    jsr output
    txa
    and #$f       ; convert lower nybble
    jsr hexc
```

```
output sta (dst),y ; output a byte using a zp-ptr and Y-index
incaddr iny ; increment the output address
  iny
  lda #0 ; null-terminate the string
  sta (dst),y
  dey
outpa rts
hexc cmp #$a ; subroutine converts 0-F to a character
  bcs hexa
  clc ; digit 0-9
  adc #48
  bne hexb ; unconditional jump coz Z=FALSE always
hexa clc
  adc #55 ; digit A-F
hexb rts
```

From:  
<http://codebase64.org/> - **Codebase 64 wiki**

Permanent link:  
[http://codebase64.org/doku.php?id=base:integer\\_to\\_hex\\_string](http://codebase64.org/doku.php?id=base:integer_to_hex_string)

Last update: **2015-04-17 04:32**

