

```

1
2          AREA          main, READONLY, CODE
3          THUMB
4          EXTERN        OutChar ; Reference external subroutine
5          EXTERN        OutStr
6          EXPORT        __main ; Make available
7
8  __main    PROC
9  start    LDR           R0,=0x0 ;Temp value 1
10         LDR           R1,=0x0 ;Temp value 2
11         LDR           R2,=0x0 ;Temp value 3
12         LDR           R3,=0xA ;Since we are converting hex to decimal. It's based is 10 ( Hexa
[A]= Deci [10])
13         LDR           R4,=0x0 ;Value that will be converted
14         LDR           R5,=0x20000480 ;Address value that will be written ASCII Value
15         PUSH         {R5} ; Pushing adress value
16         MOV          R6,R5
17         BL           CONVRT ;Starter for subroutine
18         LDR           R1,=0x04 ;null char for outstr subroutine
19         STRB          R1,[R6]
20         POP           {R5} ; Popping adress value
21         BL           OutStr
22  forever  B           forever
23         ENDP
24
25  CONVRT    PROC
26  loop     CMP          R4,#0
27         BEQ          finish
28         UDIV         R0,R4,R3 ; R0=(R4//0xA)
29         MUL          R1,R0,R3 ; R1=(R0*10) That will be our current digit, starting from unit digit
30         SUB          R2,R4,R1 ; R2= R4-R1 (that will be data for the current digit,starting from
unit digit)
31         STRB          R2,[R5] ; Writing Data
32         ADD          R5,R5,#1 ; Increasing Data Address
33         MOV          R4,R0 ; Updating number so that we can go to next digit
34         CMP          R4,#10 ; If it finishes, the number will be less than 10 otherwise it should
go to label "loop"
35         BMI          finish
36         B           loop
37  finish    STRB          R4,[R5]; Writing converted data is finished here. It is time to rearrange
numbers and converting ASCII values
38         MOV          R7,R5
39         ADD          R5,R5,#1
40         MOV          R8,R5
41  loop1     LDRB          R1,[R7] ; This loop is writing the same table at the end of it. However, it
is in reversed order
42         STRB          R1,[R5]
43         ADD          R5,R5,#1
44         SUB          R7,R7,#1
45         CMP          R7,R6
46         BPL          loop1
47  loop2     LDRB          R1,[R8] ; This loop is writing ASCII values in the reversed table at the
desired location
48         ADD          R1,R1,#48
49         STRB          R1,[R6]
50         ADD          R6,R6,#1
51         ADD          R8,R8,#1
52         CMP          R8,R5
53         BMI          loop2
54         MOV          R5,R6
55         BX           LR
56         ENDP
57         END
58
59
60
61

```