

```

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

```