Computer Systems Engineering

Computer Organization and Microprocessor

ENCS2380

Assembly Project

~~~~~~

Student's Name: **Tariq Odeh**

Student's No: **1190699**

Sec: **1**

Instructor: Dr. Abualseoud Hanani

Date: 9-6-2021

~~~~~~

Table of content

Summary

In project, I used Keil Version 4 to implement a encryption and decryption algorithm in ARM assembly. The encryption method that I have used in project is Rotate left ASCII code of each character two bits, then its key is rotate right two bits (decryption), based on [699 mod 3] +1 = 1.

The text message that I used **"Don't let your fear destroy you!"**, I stored it in memory address (0x40000000) to (0x40000083), then I applied the encryption method on every single character and then stored the encrypted message in memory address (0x400000EC) to (0x4000016F), then I take the encrypted message from address (0x400000EC) and applied the reverse of the encryption process (decryption) and then stored the encrypted message in memory address (0x400001D8) to (0x4000025B).

In code I use two procedures: the first one to applied encryption and the second to applied reverse of the encryption (decryption).

Code

```
        AREA Tariq_Odeh,CODE,READONLY
        ENTRY

        LDR R0,=0x40000000
STRING  DCB     "Don't let your fear destroy you!",0
        ADR R1,STRING

STORE
        LDRB R2,[R1]
        CMP R2,#0
        BEQ FINISH
        STR R2,[R0]
        ADD R1,R1,#1
        ADD R0,R0,#4
        B STORE
FINISH
        MOV R7,#0
        STR R7,[R0]

        BL encryption
        BL decryption
HERE BAL HERE


encryption
        LDR R8,=0x40000000
        LDR R9,=0x400000EC
READ
        LDR R2,[R8]
        CMP R2,#0
        BEQ DONE
        MOV R3,R2, ROR #30
```

```
        STR R3,[R9]
        ADD R8,R8,#4
        ADD R9,R9,#4
        B READ
DONE
        MOV R7,#0
        STR R7,[R9]
        BX LR


decryption
        LDR R5,=0x400000EC
        LDR R6,=0x400001D8
AGAIN
        LDR R2,[R5]
        CMP R2,#0
        BEQ EXIT
        MOV R3,R2, ROR #2
        STR R3,[R6]
        ADD R5,R5,#4
        ADD R6,R6,#4
        B AGAIN

EXIT

        MOV R7,#0

        STR R7,[R6]

        BX LR


        END
```

ASCII Code

In the table below there is ASCII code for each character that were used in the code (text), an operation was performed encryption and decryption for each character.

Char	Hex	Hex after Rotate left two bits (encryption)	Hex after Rotate right two bits (decryption)
D	44	110	44
o	6F	1BC	6F
n	6E	1B8	6E
'	27	9C	27
t	74	1D0	74
space	20	80	20
l	6C	1B0	6C
e	65	194	65
t	74	1D0	74
space	20	80	20
y	79	1E4	79
o	6F	1BC	6F
u	75	1D4	75
r	72	1C8	72
space	20	80	20
f	66	198	66
e	65	194	65
a	61	184	61
r	72	1C8	72
space	20	80	20
d	64	190	64
e	65	194	65
s	73	1CC	73
t	74	1D0	74
r	72	1C8	72
o	6F	1BC	6F
y	79	1E4	79
space	20	80	20
y	79	1E4	79
o	6F	1BC	6F
u	75	1D4	75
!	21	84	21
zero	00	00	00

Table 1: ASCII code for each character that were used in the code

Test & Output

To check, I do the following:
1- Translate the file.
2- Build the target files.
3- Rebuild all target files.
4- Start Debug Session.
5- Show the output (result).

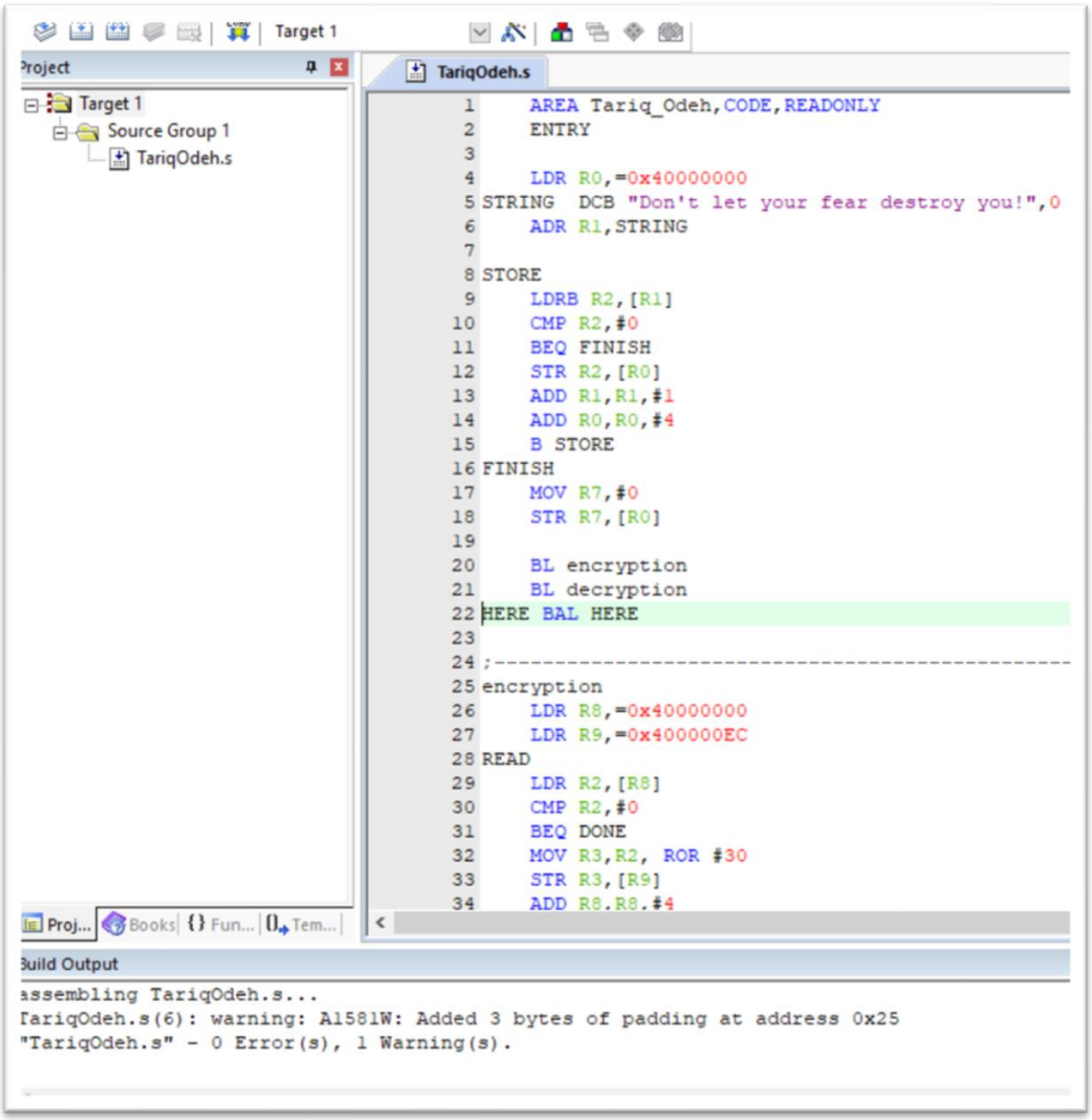

Figure 1: Translate the file.

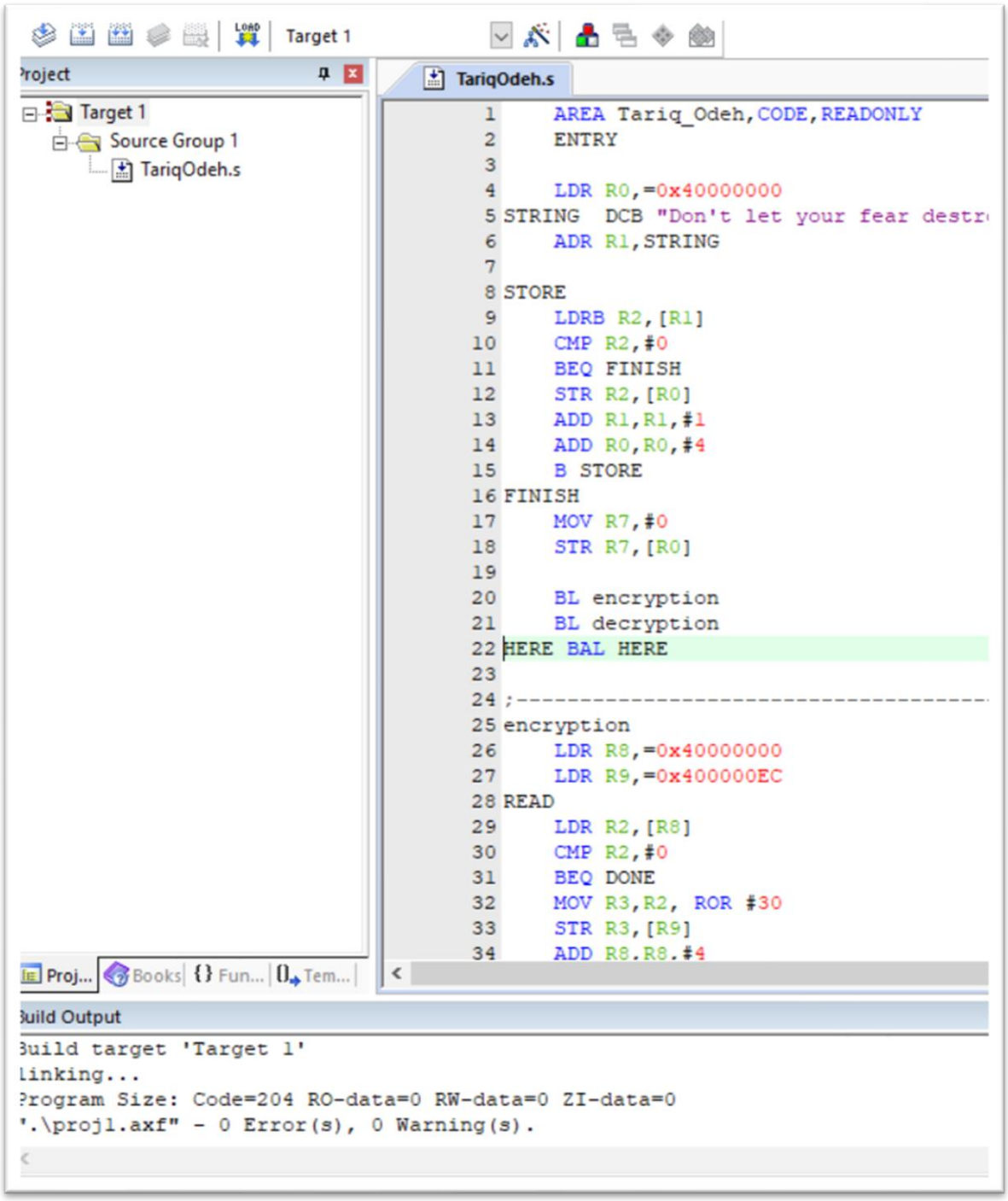

Figure 2: Build the target files.

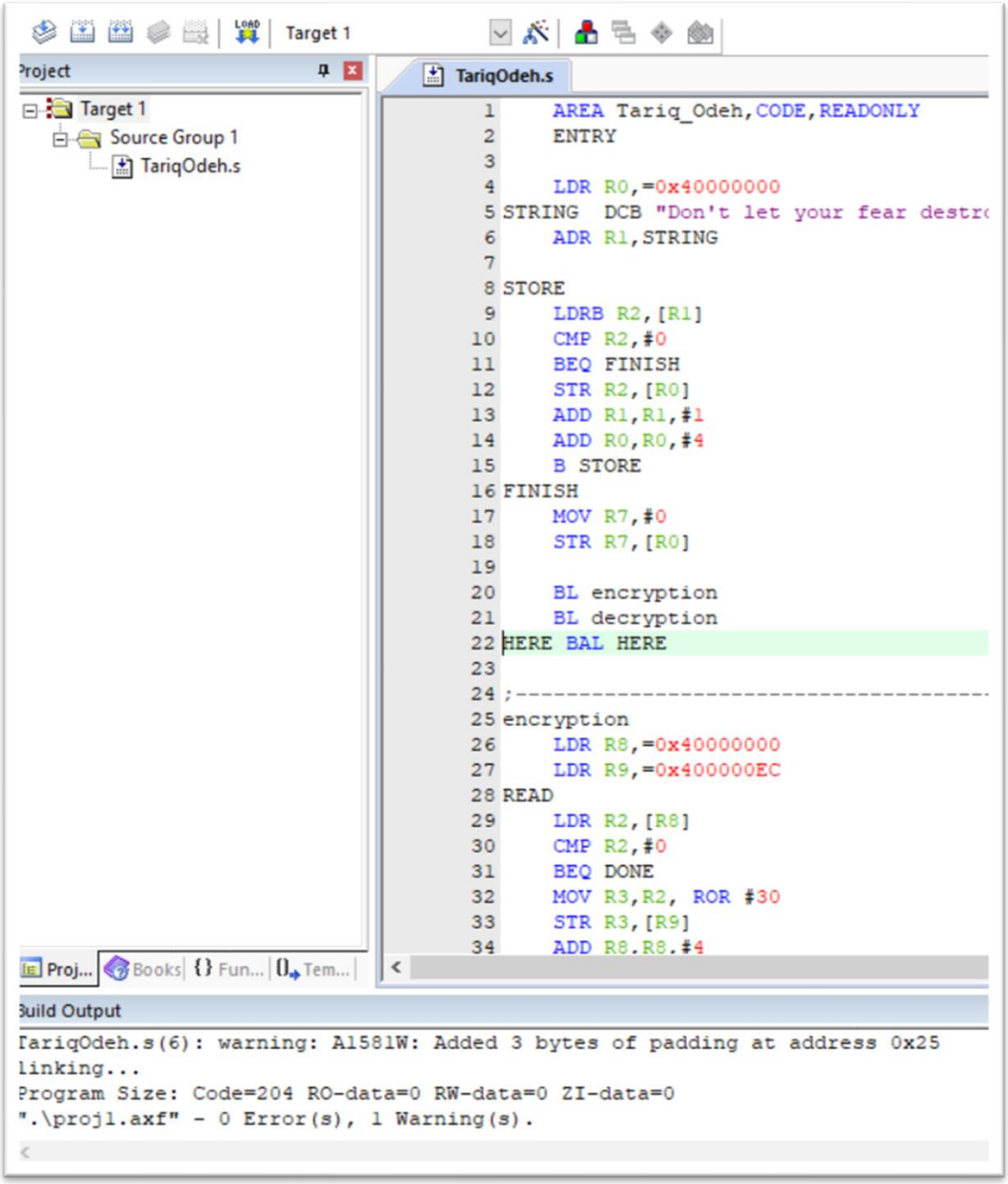

Figure 3: Rebuild the target files.

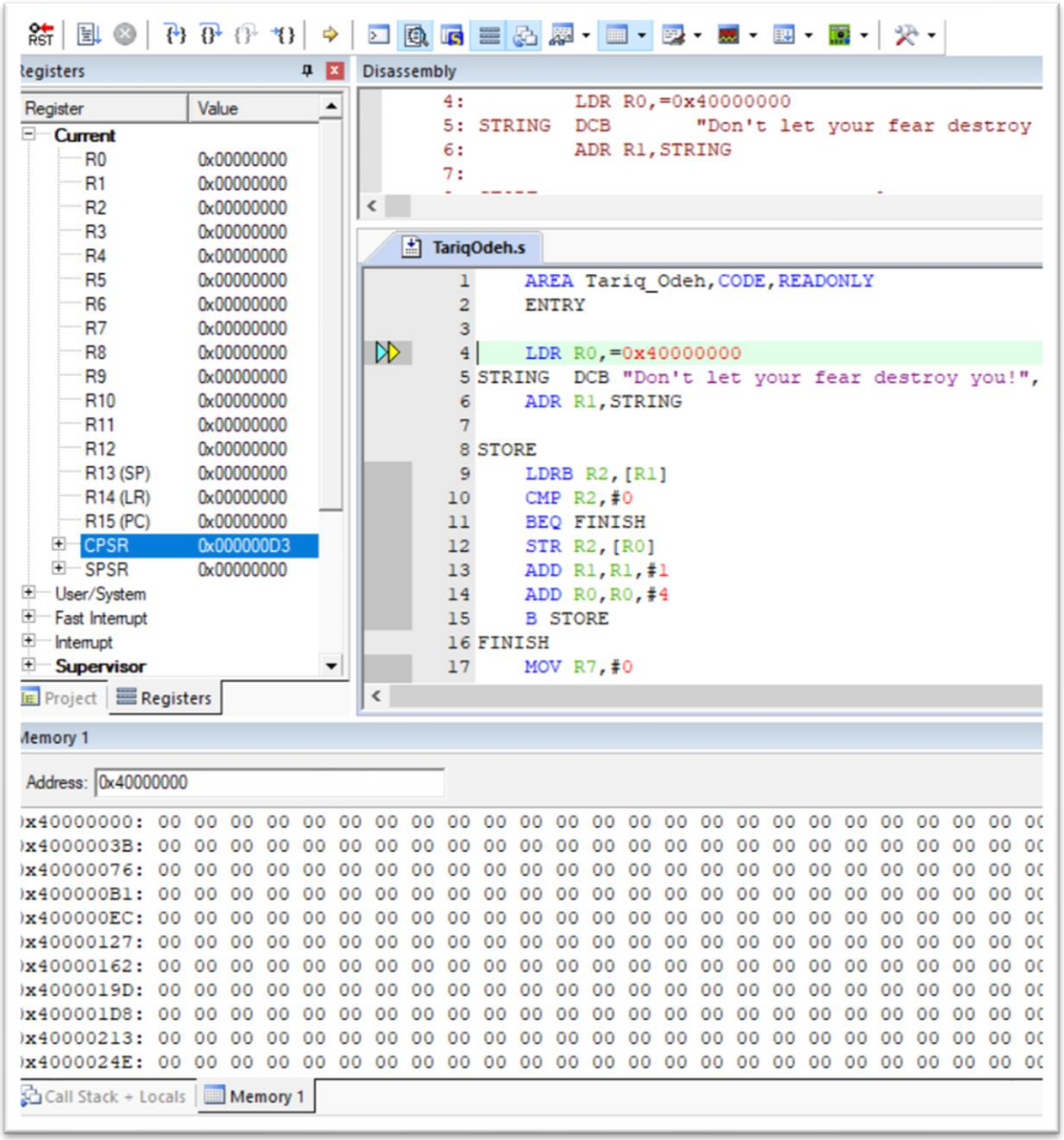

Figure 4: Start Debug Session.

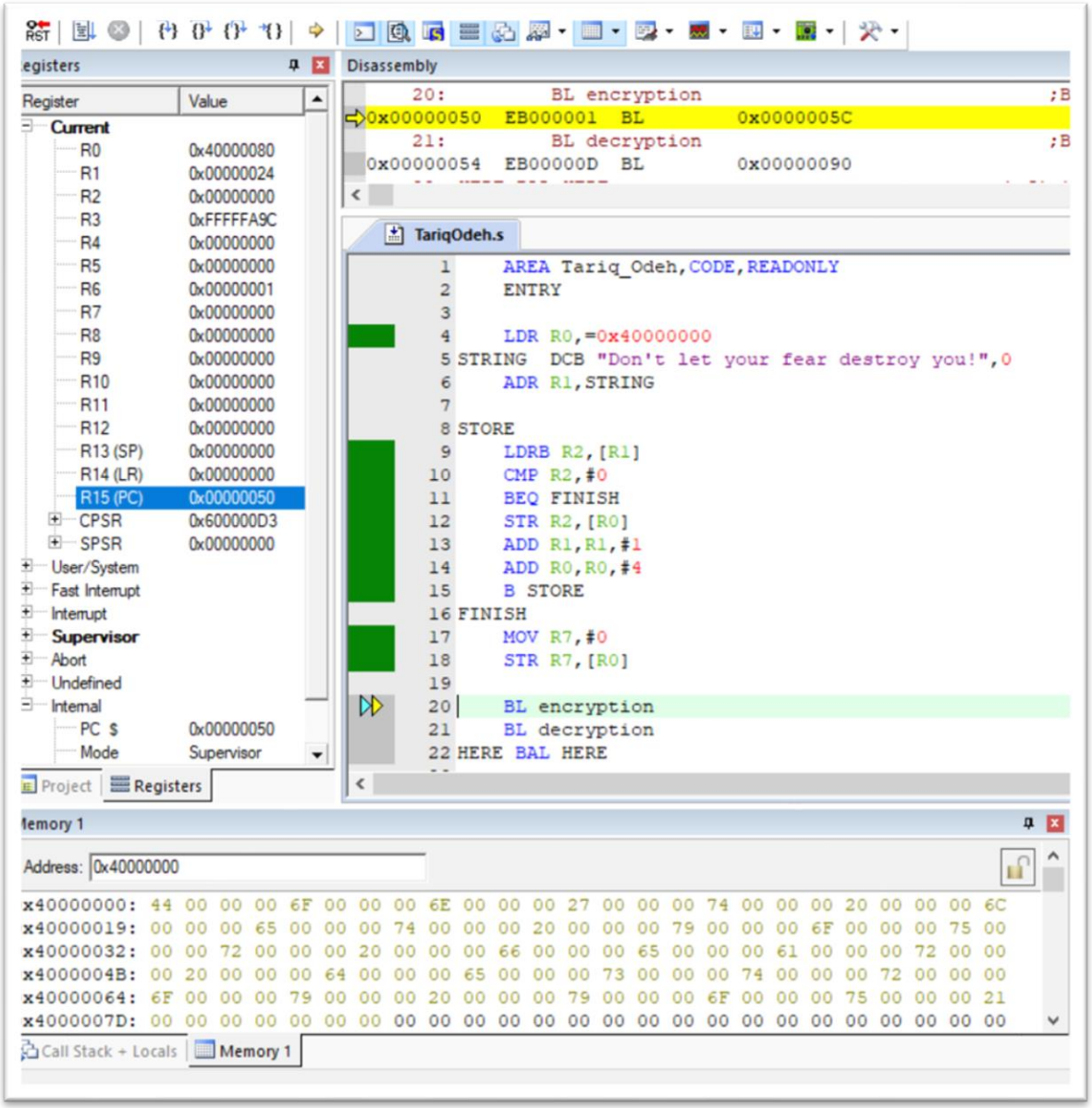

Figure 5: Store String in memory address (0x40000000) to (0x40000083).

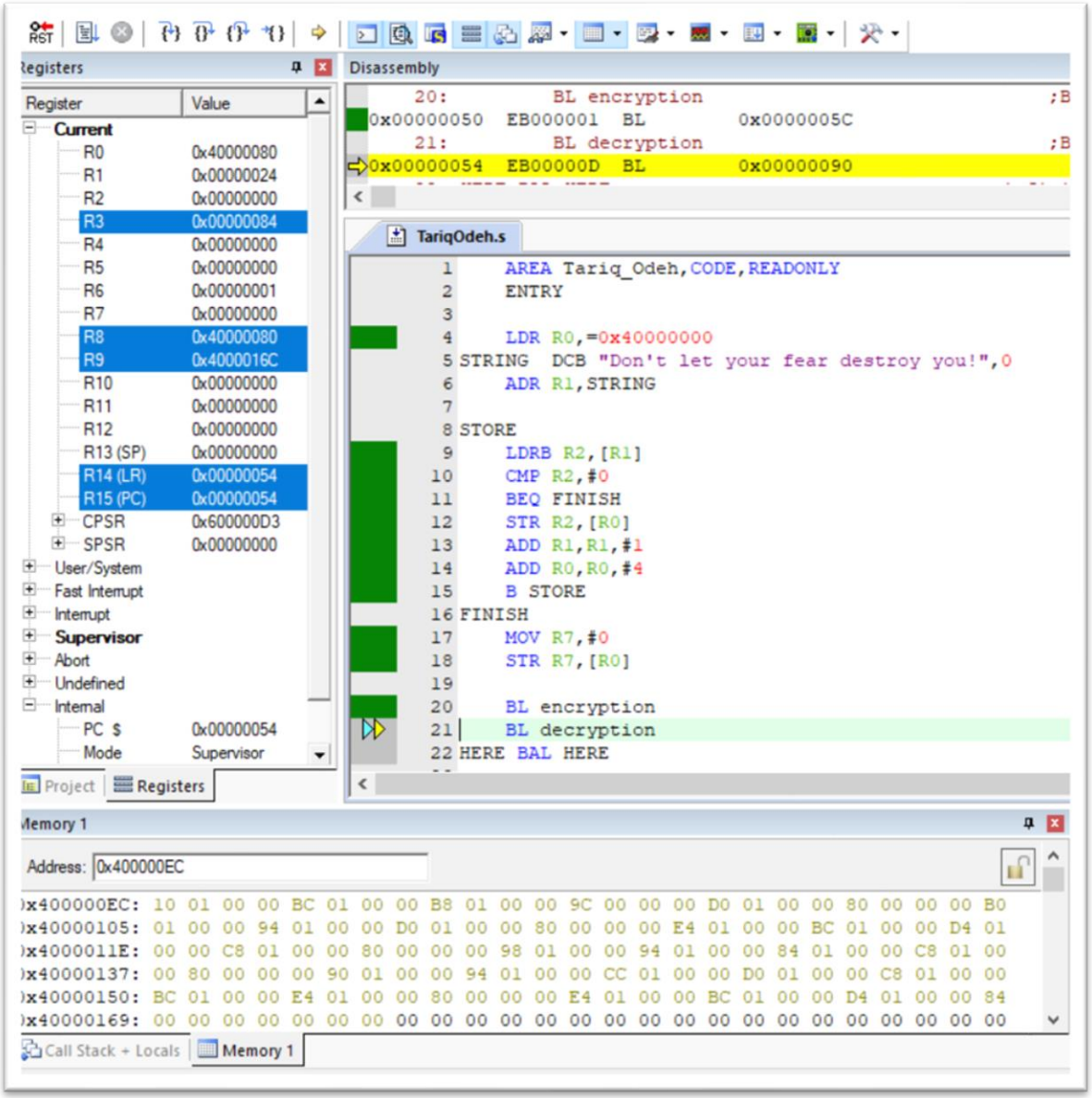

Figure 6: Store encryption result in memory address (0x400000EC) to (0x4000016F).

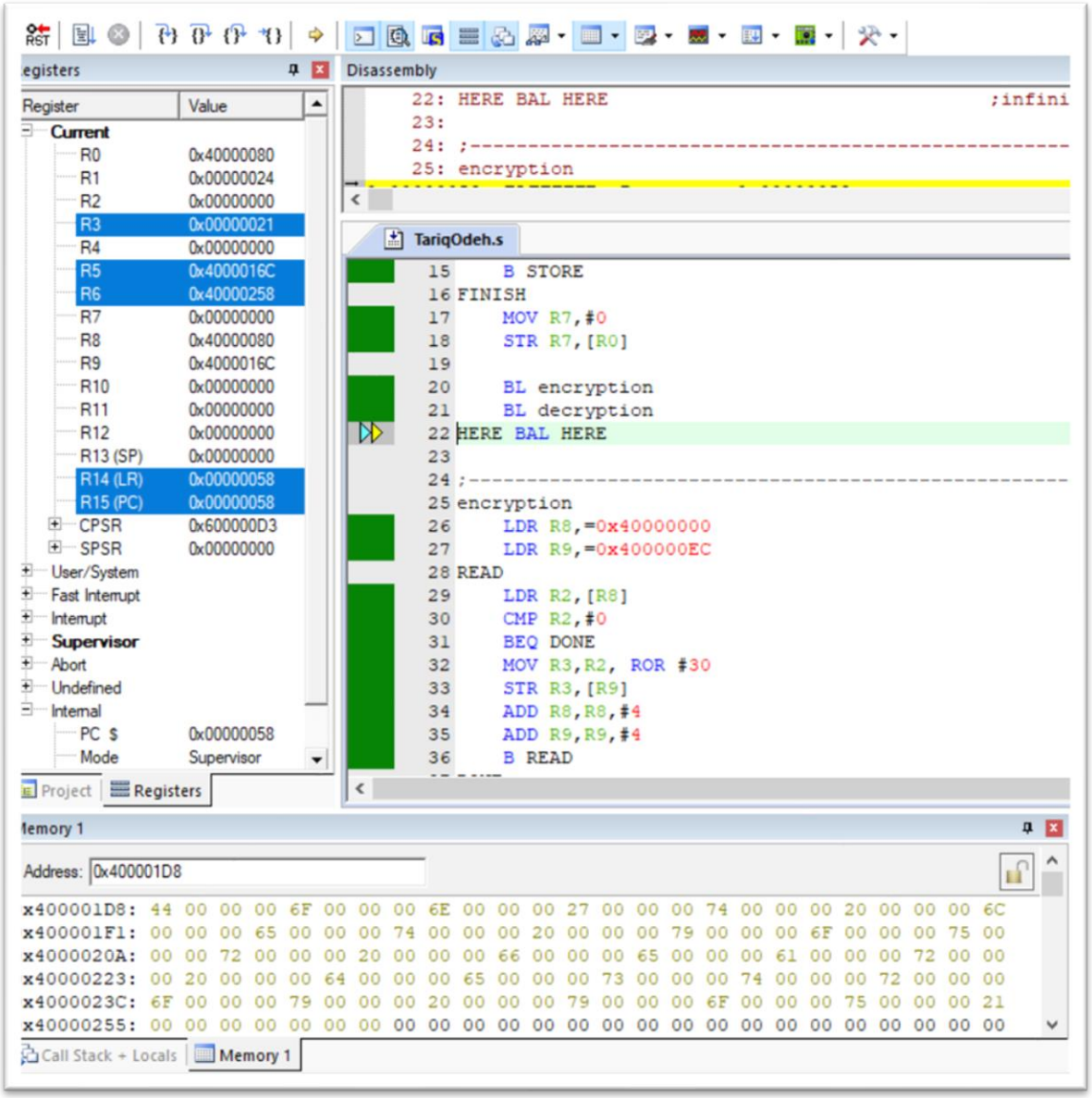

Figure 7: Store decryption result in memory address (0x400001D8) to (0x4000025B).

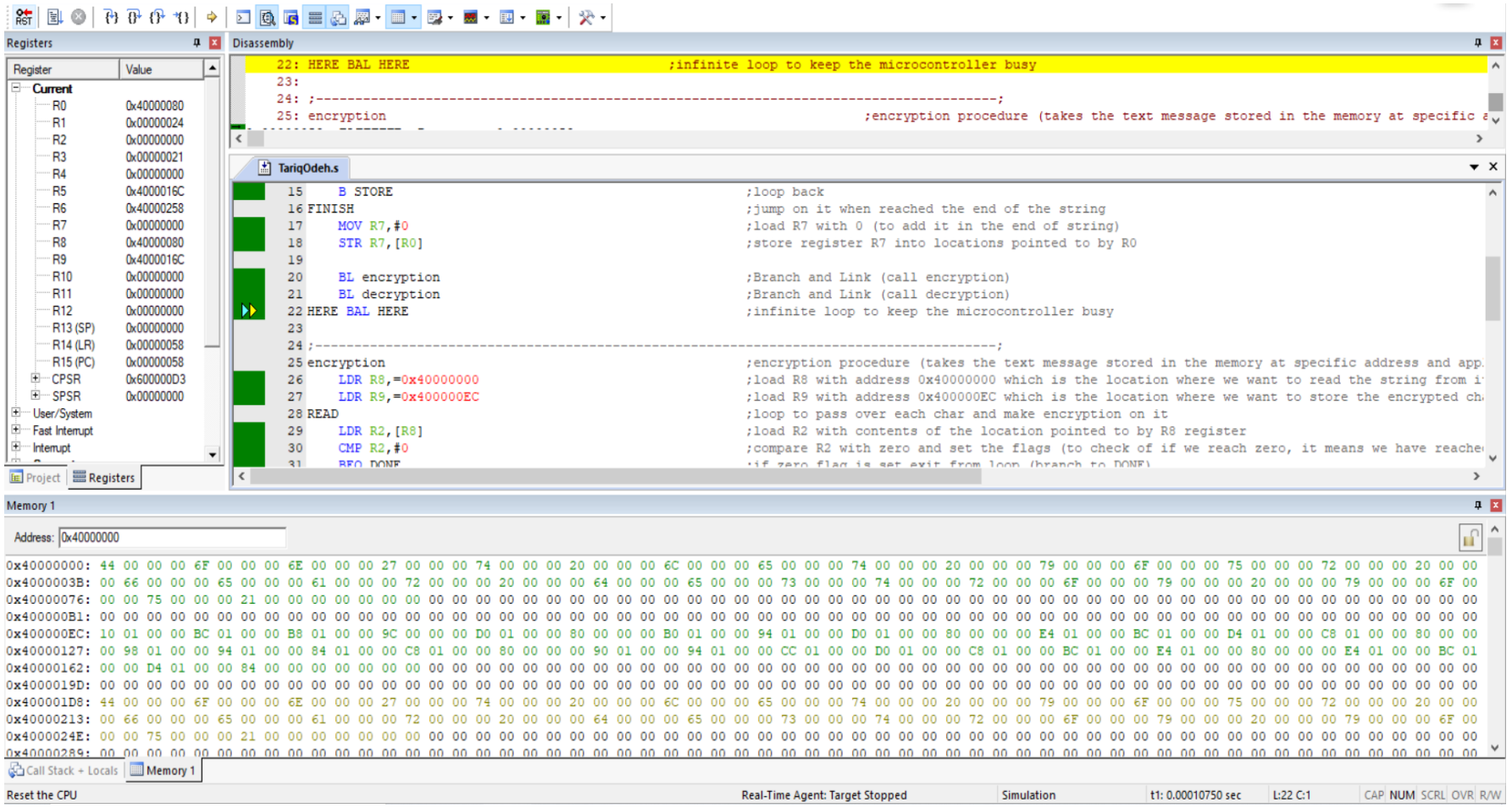

Figure 8: The final result.