Freescale Semiconductor
Application Note

Document Number: AN3528
Rev. 0, 09/2007

AES Library Usage

by: Ioseph Martinez
Applications Engineer
RTAC Americas

1 Introduction

This document describes the advanced encryption standard library for the HCS08 core microcontrollers. The library code is in C language, and can be migrated to other Freescale technologies. Performance is priority in the library architecture.

There are three public elements that can be used by the user:

- AES_USES_NEAR_VARIABLES switch
- AesCipher function
- AesInvCipher function

NOTE

For basic use, there is no need to touch any other elements in this library.

Below is an explanation of the three elements mentioned above.

Contents

freescale semiconductor

2 AES_USES_NEAR_VARIABLES

This compile time switch selects between using near and not near variables. The near variables are located in the zero RAM section. When the switch is activated, smaller and faster code is compiled. If this memory section is in use, or you do not want to allocate these variables in the zero RAM section, the switch must be commented. Then the linker will allocate used variables in the proper place. The only requirement for using this switch is to have a section called Z_RAM and placing AES_ZERO_MEMORY in this section. The following code is an example:

```
PLACEMENT
    DEFAULT_RAM                                          INTO  RAM;
    DEFAULT_ROM, ROM_VAR, STRINGS                        INTO  ROM;
    _DATA_ZEROPAGE, MY_ZEROPAGE, AES_ZERO_MEMORY         INTO  Z_RAM;
END
```

3 AesCipher

This function implements the cipher function of the engine. Below is the prototype:

```
void AesCipher
(const unsigned char *pData, const unsigned char *pKey, unsigned char *pReturnData);
```

The first argument:

*pData is a pointer to the array holding 16 bytes of plain text. This array is not modified by the function, unless it is used as a result array.

The next argument:

*pKey argument holds the 16 bytes key. This array is not modified by the function, unless it used as result pointer.

The last argument:

*pReturnData holds the 16 bytes result of the whole operation called the ciphertext.

4 AesInvCipher

The use of the AesInvCypher is similiar to the AesCipher. This function implements the inverse cipher function of the engine:

```
void AesInvCipher
(const unsigned char *pData, const unsigned char *pKey, unsigned char *pReturnData);
```

There is a difference between the AesInvCypher and the AesCipher:

*pData in this function holds the ciphertext

*pReturnData returns plaintext

This function follows the same rules as the AesCipher function

5 Conclusion

This document summarizes how to use main functions in the S08 AES Library.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN3528
Rev. 0
09/2007

freescale™
semiconductor