Software DES

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Implement Data Encryption Standard by software.
BSP Version	NUC230/240 Series BSP v3.01.002
Hardware	NuTiny–EVB–NUC240–LQFP100 V1.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 Function Description

1.1 Introduction

This sample code demonstrates how to implement software DES (Data Encryption Standard) and how DES encryption/decryption works.

1.2 Principle

DES is a symmetric encryption system that uses 64-bit text blocks. It implements a 16-round encryption function and a key generator to generate 48-bit keys for the encrypt function, as shown in Figure 1-1.

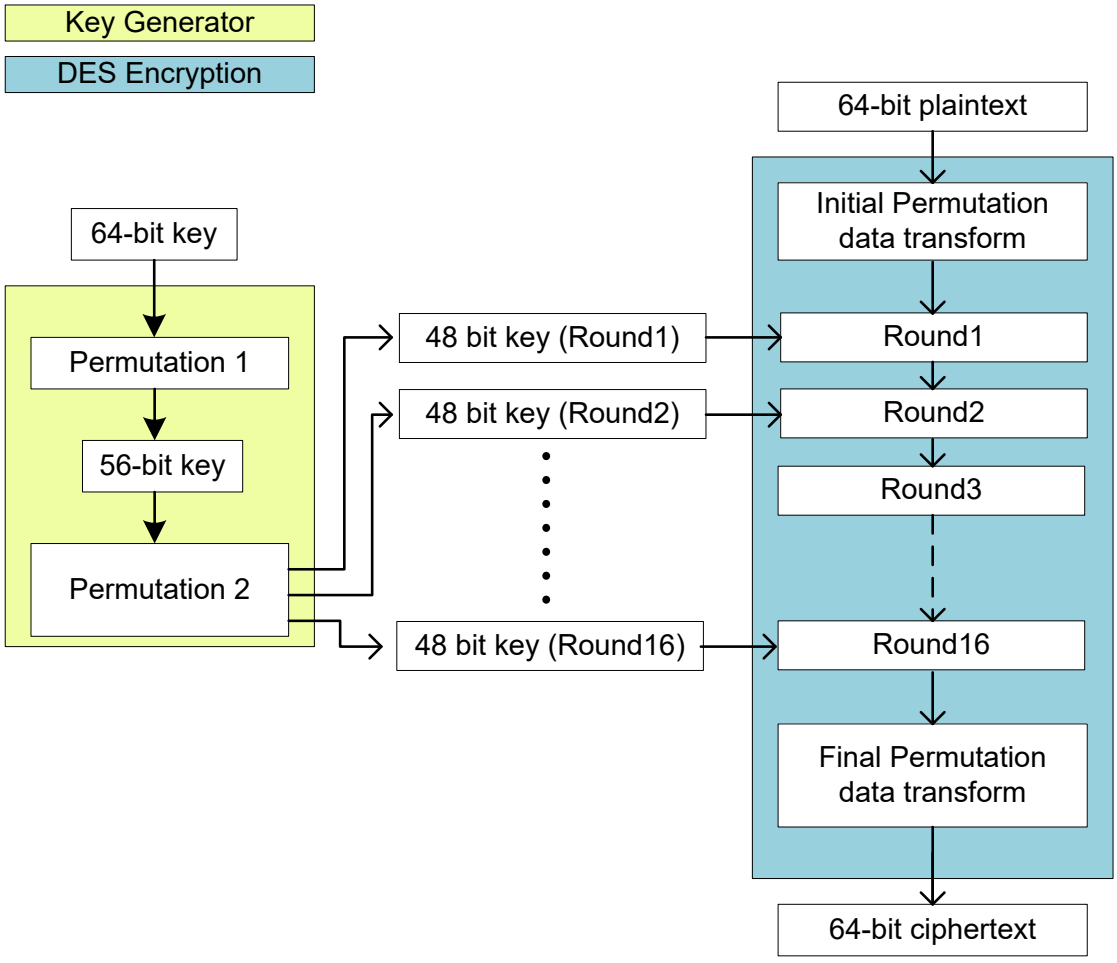

Figure 1-1 DES Encryption Module and Key Generator

Key generator takes a 64-bit key as input and the output are 16 48-bit keys for DES encryption. DES encryption function takes a 64-bit plaintext and 16 keys from key generator as inputs. The 64-bit plaintext is transformed to another 64-bit data by initial permutation, then enters a 16-round encryption process. Finally, the final permutation is executed and a 64-bit cipher text is the final output. The DES decryption function takes 64-bit ciphertext as input,

and also performs 16 rounds of decryption operations, and finally outputs 64-bit plaintext.

1.3 Demo Result

The execution result can be output on UART, baud rate 115200, as shown in Figure 1-2.

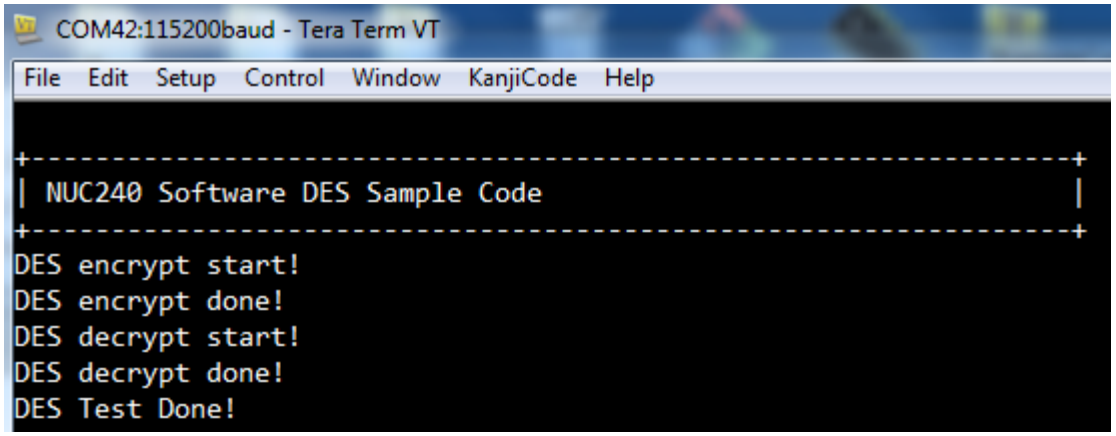

Figure 1-2 Execution Result on UART

2 Code Description

Below function shows how to generate 48-bit keys for DES：

```
/*---------------------------------------------------------------------------*/
/*  Set DES Private Key Function                                             */
/*---------------------------------------------------------------------------*/
void SetKey(char acMyKey[8])
{
    int i;
    static uint8_t au8Key_bit[64] = { 0 };
    static uint8_t *pu8Key_bit_L, *pucKey_bit_R;

    pu8Key_bit_L = &au8Key_bit[0];
    pucKey_bit_R = &au8Key_bit[28];
    ByteToBit(au8Key_bit, acMyKey, 64);

    /* Key transform: 64 bit --> 56 bit */
    TableReplace(au8Key_bit, au8Key_bit, s_acPC1Table, 56);

    /* Generate keys for 16 round DES */
    for (i = 0; i<16; i++)
    {
        /* DES rotation of left 28 bits */
        LoopBit(pu8Key_bit_L, s_acMoveTable[i], 28);

        /* DES rotation of right 28 bits */
        LoopBit(pucKey_bit_R, s_acMoveTable[i], 28);

        /* Key transform: 56 bit --> 48 bit */
        TableReplace(s_au8SubKey[i], au8Key_bit, s_acPC2Table, 48);
    }
}
```

Below function shows the DES encryption process：

```
/*---------------------------------------------------------------------------*/
/*  DES Encrypt Function                                                     */
/*---------------------------------------------------------------------------*/
void RunDesEncrypt(char acg_acMyMessage[8], char acHexMssage[16])
{
```

```
    int i;
    static uint8_t au8Message_bit[64] = { 0 };
    static uint8_t *Message_bit_L = &au8Message_bit[0], *Message_bit_R =
&au8Message_bit[32];
    static uint8_t au8Temp[32] = { 0 };

    /* Convert byte message to bit */
    ByteToBit(au8Message_bit, acg_acMyMessage, 64);

    /* Data transform by IP Table*/
    TableReplace(au8Message_bit, au8Message_bit, g_acIPTable, 64);

    /* DES 16-round encryption */
    for (i = 0; i<16; i++)
    {
        BitCopy(au8Temp, Message_bit_R, 32);
        FChange(Message_bit_R, s_au8SubKey[i]);
        Xor(Message_bit_R, Message_bit_L, 32);
        BitCopy(Message_bit_L, au8Temp, 32);
    }

    /* Data transform by IPR Table*/
    TableReplace(au8Message_bit, au8Message_bit, g_acIPRTable, 64);

    /* Convert bit message to hex */
    BitToHex(acHexMssage, au8Message_bit, 64);
}
```

Below function shows the DES decryption process：

```
/*---------------------------------------------------------------------------*/
/*  DES Decrypt Function                                                     */
/*---------------------------------------------------------------------------*/
void RunDesDecrypt(char acg_acMyMessage[8], char acHexMessage[16])
{
    int i = 0;
    static uint8_t au8Message_bit[64] = { 0 };
    static uint8_t * Message_bit_L = &au8Message_bit[0], *Message_bit_R =
&au8Message_bit[32];
    static uint8_t au8Temp[32] = { 0 };

    /* Convert hex message to bit */
    HexToBit(au8Message_bit, acHexMessage, 64);
```

```
    /* Data transform by IP Table*/
    TableReplace(au8Message_bit, au8Message_bit, g_acIPTable, 64);

    /* DES 16-round decryption */
    for (i = 15; i >= 0; i--)
    {
        BitCopy(au8Temp, Message_bit_L, 32);
        FChange(Message_bit_L, s_au8SubKey[i]);
        Xor(Message_bit_L, Message_bit_R, 32);
        BitCopy(Message_bit_R, au8Temp, 32);
    }

    /* Data transform by IPR Table*/
    TableReplace(au8Message_bit, au8Message_bit, g_acIPRTable, 64);

    /* Convert bit message to byte */
    BitToByte(acg_acMyMessage, au8Message_bit, 64);

}
```

3 Software and Hardware Environment

Software Environment

- BSP version
  - NUC230/240 Series BSP CMSIS v3.01.002
- IDE version
  - Keil uVersion 5.28

Hardware Environment

- Circuit components
  - NuTiny–EVB–NUC240–LQFP100 V1.0
- Diagram
  - Connect UART TX (PB.1) pin to PC UART RX for display the execution result of example code on PC.

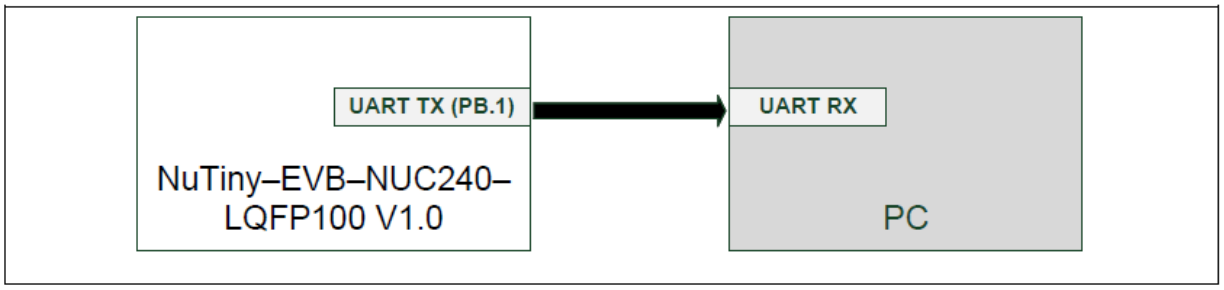

Figure 3-1 NUC240 UART TX and PC UART RX Connection

4 Directory Information

- EC_NUC240_Software_DES_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click Software_DES.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Sep. 9, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.