CC32xx AES Demo Application

Overview

The Advance Encryption Standard (AES) security modules provide hardware-accelerated data encryption and decryption operations based on a binary key. The AES is a symmetric cipher module that supports a 128-, 192-, or 256-bit key in hardware for both encryption and decryption. The AES module is based on a symmetric algorithm, meaning that the encryption and decryption keys are identical. To encrypt data means to convert it from plain text to an unintelligible form called cipher text. Decrypting cipher text converts previously encrypted data back to its original plain text form.

Application details

The application is a reference to usage of AES DriverLib functions on CC3200. Developer/User can refer to this simple application and re-use the functions in their applications. This application can be used with or without "Uart Terminal".

If the user wishes to use "Uart Terminal" to give some inputs and follow the execution path prints, then they might do so by defining "USER-INPUT" in the aes_main.c file.

- **aesdemo**: This command allows the user to excercise the AES funcitonality on CC3200. The command needs two parameters aes-mode and key-len.

    - aes-mode is the AES algorithm that user can choose, the value can be ECB or CBC or CTR or ICM or CFB.

    - key-len is the length of the key that user needs to define, the value can be 128 or 192 or 256.

Further, user will be prompted for more inputs for "key" and "plain text input"

Not defining or un-defining the USER-INPUT will allow the user to follow the execution path on the IAR or CCS IDE, in the "debugging" mode and no input is needed to be given by the user.

Source Files briefly explained

- **main.c** - The main file that contains the core-logic for encryption and decryption. The functions in the file uses DriverLib calls to perform encryption and decryption.

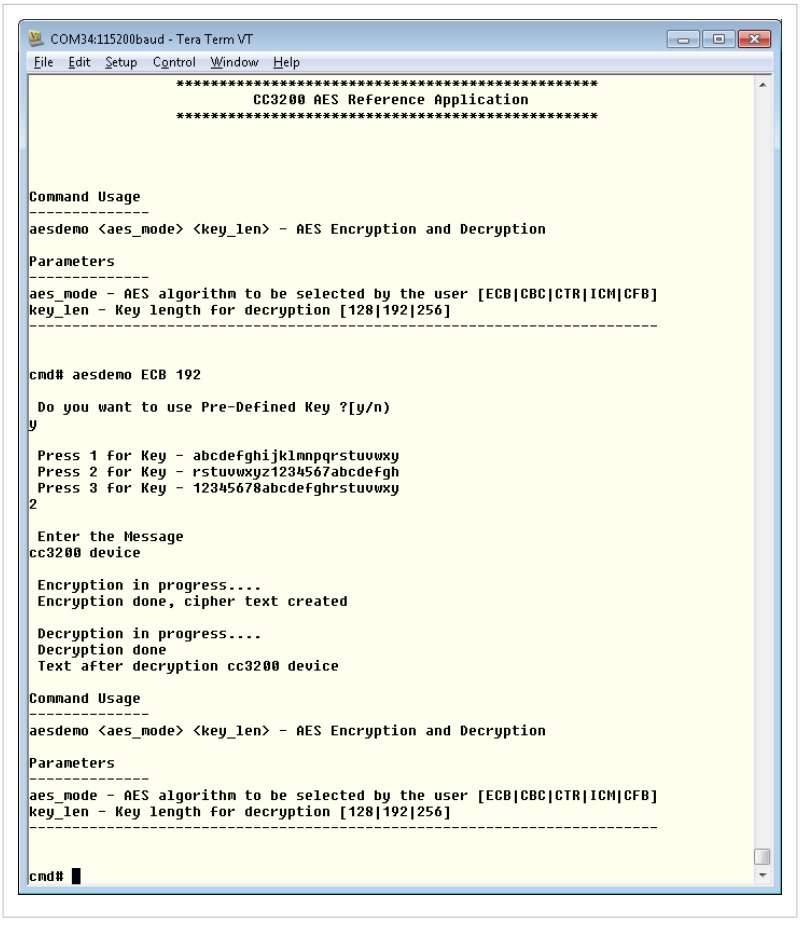
Supporting files

- **aes_userinput.c** - This file is used in the USER-INPUT mode. The function in the file reads the input from the user, parses the input string and feed the core-logic functions in the aes_main.c
- **pinmux.c** - Generated by the PinMUX utility. UART0 pins are brought out in this file.
- **startup_ccs.c** - CCS related functions
- **startup_ewarm.c -** IAR related functions
- **uart_if.c -** Functions to display information on UART

Usage

1. Setup a serial communication application (HyperTerminal/TeraTerm). For detail info visit Terminal setup.
   On the host PC, open a hyperterminal, with the following settings
   - **Port**: Enumerated COM port
   - **Baud rate**: 115200
   - **Data**: 8 bit
   - **Parity**: None
   - **Stop**: 1 bit
   - **Flow control**: None
2. Run the reference application (/IAR/CCS).
   - Flash the bin
   - Open the project in IAR/CCS. Build and download the application to the board
3. On the Hyperterminal, a prompt appears
   - The AES commands need to be issued and the results can be seen

     Terminal snapshot when user gives input message to encrypt and decrypt



Limitations/Known Issues

None.

Article Sources and Contributors

CC32xx AES Demo Application *Source*: http://processors.wiki.ti.com/index.php?oldid=180820 *Contributors*: Jitgupta

Image Sources, Licenses and Contributors

Image:CC3200 aes terminal runScreen.png *Source*: http://processors.wiki.ti.com/index.php?title=File:CC3200_aes_terminal_runScreen.png *License*: unknown *Contributors*: Codycooke