CE220511 – PSoC 6 MCU Cryptography: SHA Demonstration

Objective

This code example demonstrates generating a unique hash value or message digest for an arbitrary message using Secure Hash Algorithm (SHA) in PSoC® 6 MCU.

Overview

This code example shows how to generate a 20-byte hash value or message digest for an arbitrary user input message using the SHA2 algorithm using the Cryptographic hardware block in PSoC 6 MCU. The example further shows that any change in the message results in a unique hash value for the message. The hash value generated for the message is displayed on a UART terminal emulator.

Requirements

Tool: PSoC Creator™ 4.2

Programming Language: C (Arm® GCC 5.4)

Associated Parts: PSoC 6 MCU

Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Design

Secure Hash Algorithm is a function that takes a message of arbitrary length and reduces it to a fixed length residue or message digest after performing a series of mathematically defined operations that practically guarantee that any change in the message will change the hash value. A hash value is used for message authentication by transmitting a message with a hash value appended to it and recalculating the message hash value using the same algorithm at the recipient's end. If the hashes differ then it indicates that the message has been corrupted.

Cryptography in PSoC 6 MCU is based on a Client-Server model. The firmware initializes and starts the Crypto server. The server runs only on the CM0+ core, and works with the crypto hardware. Access to the server is through the Inter-Process Communication (IPC) driver.

The Crypto client can run on either core. In this example client runs on the CM4 core. The firmware initializes and starts the client. The firmware then provides the configuration data required for the SHA operation and requests the Crypto server to run the cryptographic operation. Secure Hash Algorithm is directly implemented in hardware in the Crypto block of PSoC 6 MCU.

In this example, the user input message is read from the UART terminal and a 20-byte long hash value is generated using the SHA2 algorithm. For any arbitrary message, a 20-byte long hash value is generated. The 20-byte hash value for the user input message is then displayed on the UART terminal emulator. Note that in this example, the maximum message size is restricted to 100 characters. If you need to increase the message size change the macro `MAX_MESSAGE_SIZE` defined in *CryptoSHA.h* file to the message size that you require. Figure 1 shows the firmware flowchart.

Figure 1. Firmware Flow

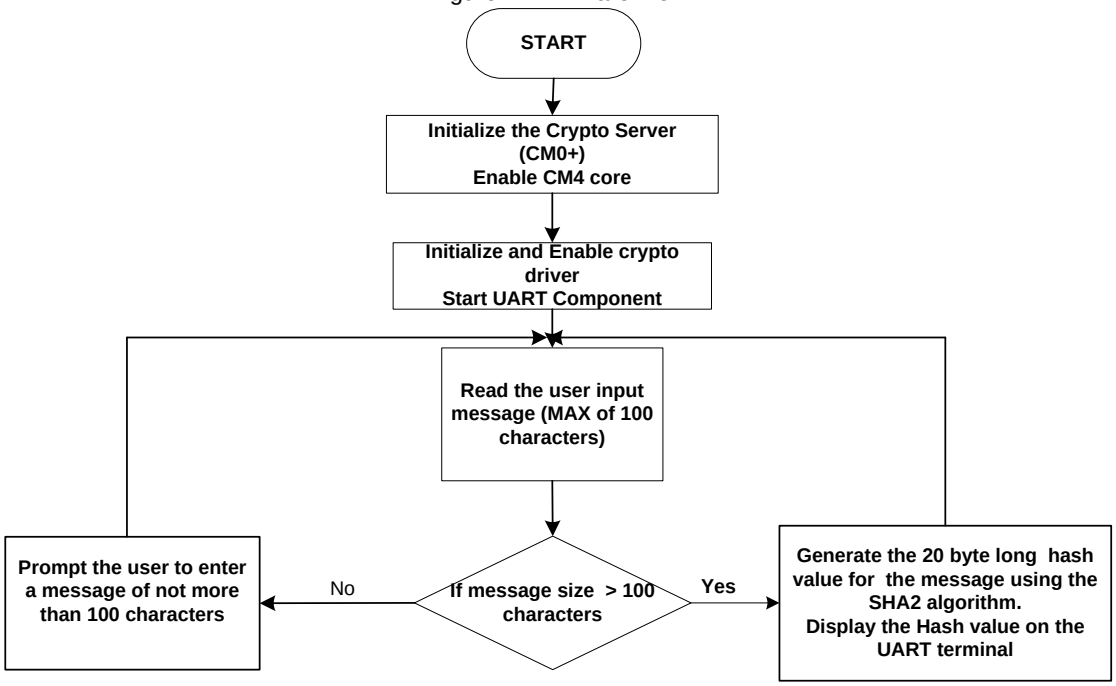
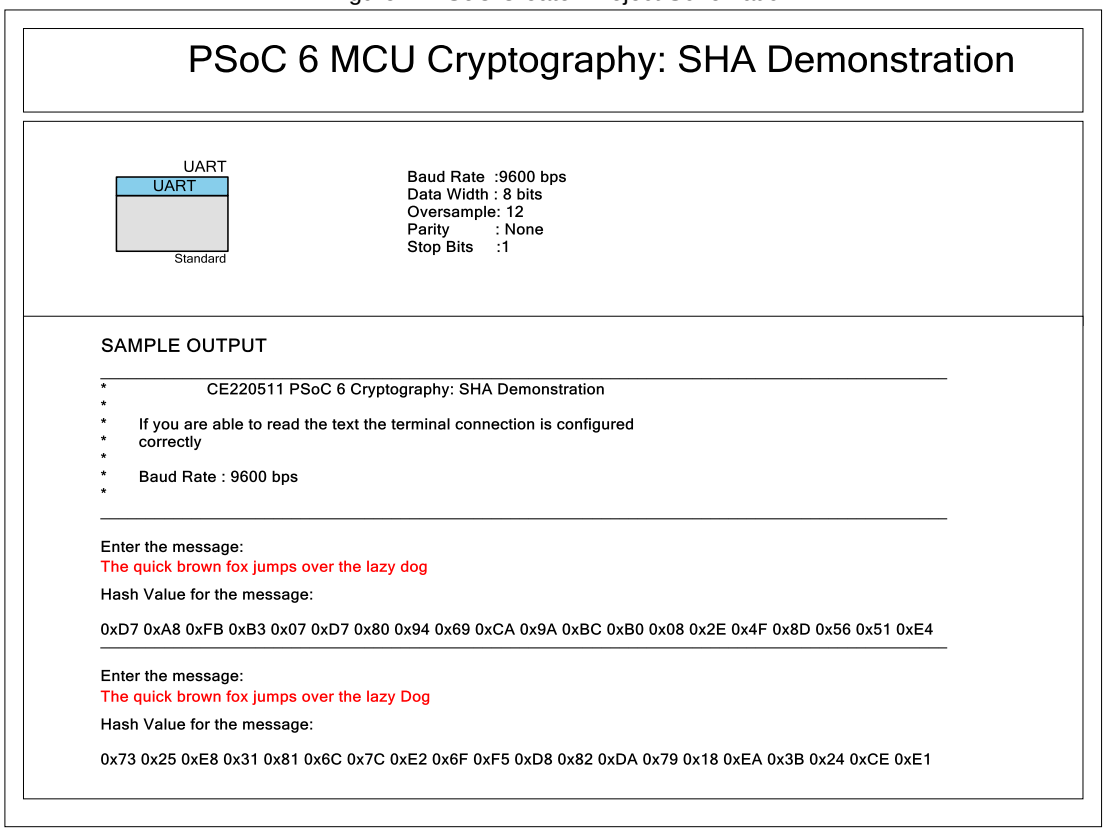


Figure 2 shows the PSoC Creator project schematic.

Figure 2. PSoC Creator Project Schematic



Hardware Setup

No special hardware setup is required for CY8CKIT-062-BLE. Connect the kit's USB port to your computer's USB port. The KitProg2 system on the kit acts as both a programmer for direct programming, and as a USB-UART bridge for displaying the generated hash value for the input message on a UART terminal.

Software Setup

This example uses Tera Term as the UART terminal for displaying the generated message digest. Set the UART configuration settings same as that used by the UART SCB on PSoC 6 MCU.

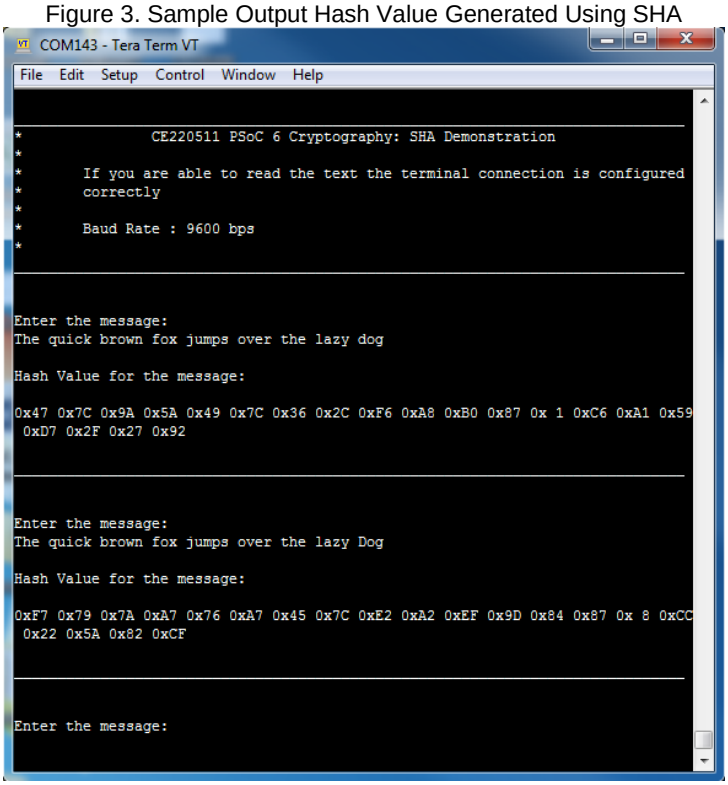
Operation

1. Plug the CY8CKIT-062 kit board into your computer's USB port.
2. Build the project and program it into the PSoC 6 MCU device. Choose **Debug** > **Program**. For more information on device programming, see PSoC Creator Help. Flash for both CPUs is programmed in a single program operation.
3. Open Tera Term and connect to the KitProg2 USB-UART bridge COM port. Set the baud rate as 9600 bps and enable the local echo option under **Setup** >**Terminal**.
4. Press the reset button on the kit and enter the message for which hash value or the message digest has to be generated. The generated hash value is printed on the UART terminal. Note that for every input message the SHA operation generates a unique hash value.

   For example, the message "The quick brown fox jumps over the lazy dog", which uses every letter in the English alphabets, has a message digest value completely different from the message digest value for the message "The quick brown fox jumps over the lazy Dog" even though they have a very small variation ('D' instead of 'd').

Figure 3 shows a sample output as displayed on Tera Term UART terminal.

Figure 3. Sample Output Hash Value Generated Using SHA



The sections that follow discuss the Components, parameter settings, and resources used to make the example.

Components

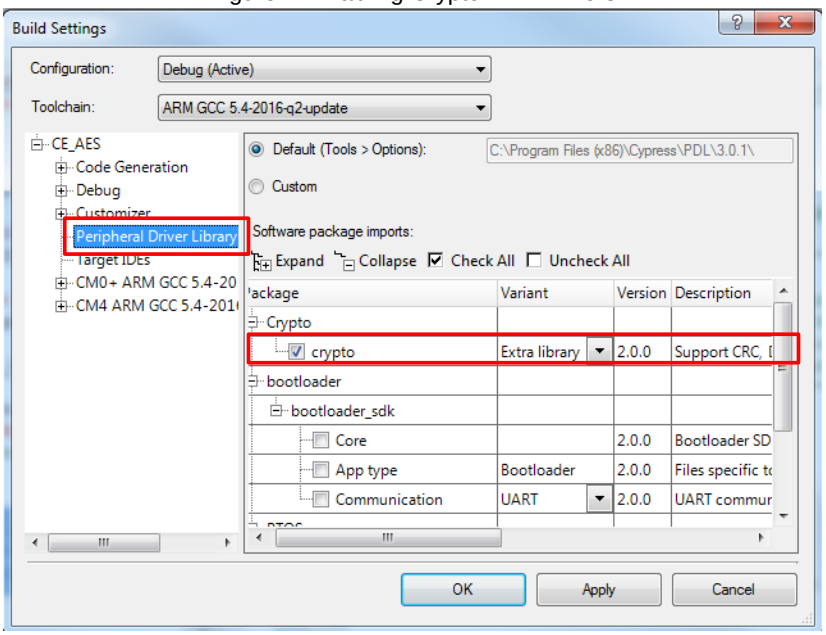
Table 1 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each.

Table 1. PSoC Creator Components

Component	Instance Name	Hardware Resources	Parameter Settings
UART	UART	1 SCB	Baud Rate set to 9600 bps

In order to use the Crypto block of PSoC 6 MCU in your design, the Crypto driver must be enabled. To enable the drivers, check the crypto option under **Project\Build Settings\ Peripheral Driver Library** as shown in Figure 4.

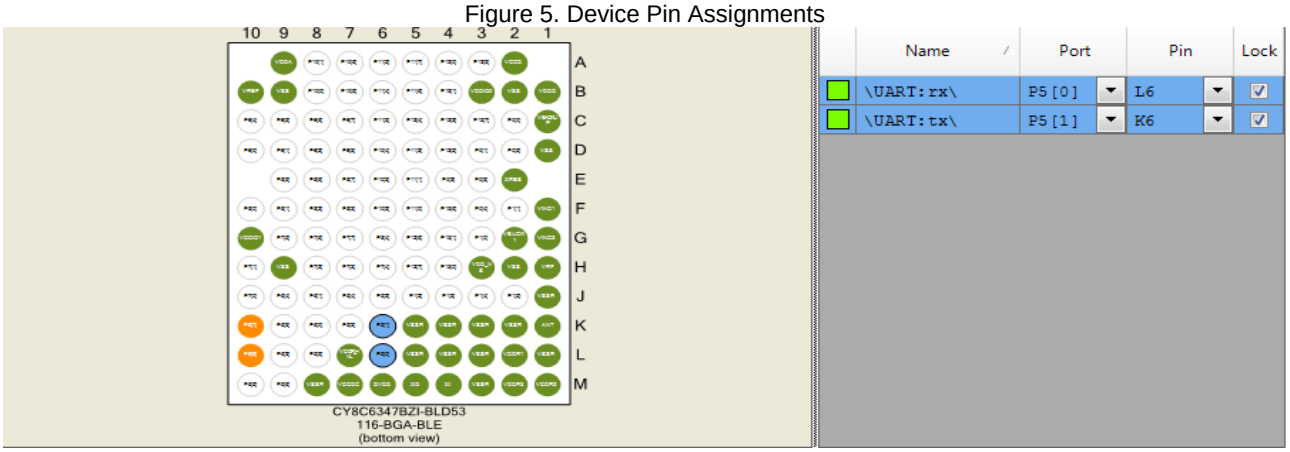
Figure 4. Enabling Crypto PDL Drivers



Design-Wide Resources

Figure 5 shows the pin assignments for the CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit required for the UART Component.

Figure 5. Device Pin Assignments



Related Documents

Table 2. Related Documents

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes the PSoC 6 MCU with BLE, and how to create your first PSoC Creator Project.
PSoC Creator Component Datasheets	
UART	Supports standard UART interface
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual

Development Kit (DVK) Documentation
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Document History

Document Title: CE220511 - PSoC 6 MCU Cryptography: SHA Demonstration

Document Number: 002-20511

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5846644	VKVK	08/07/2017	New code example
*A	6002370	VKVK	12/22/2017	Updated for PSoC Creator 4.2

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709