CE221295 – PSoC 6 MCU Cryptography: True Random Number Generation

Objective

This code example demonstrates generating a One-Time Password (OTP) using the True Random Number generation feature of PSoC® 6 MCU cryptography block.

Requirements

Tool: ModusToolbox™ IDE 1.1

Programming Language: C

Associated Parts: All PSoC 6 MCU parts

Related Hardware: PSoC 6 BLE Pioneer Kit, PSoC 6 WiFi-BT Pioneer Kit, PSoC6 WiFi-Prototyping Kit

Overview

This example demonstrates generating a One-Time-Password (OTP) of eight characters in length. Using the True Random Number generation feature of the PSoC 6 MCU crypto block, a random number corresponding to each character of the OTP is generated. The generated random number is such that it corresponds to alpha-numeric and special characters of the ASCII code. The generated OTP is then displayed on a UART terminal emulator.

Hardware Setup

This example uses the kit's default configuration. Refer to the kit guide to ensure that the kit is configured correctly.

Software Setup

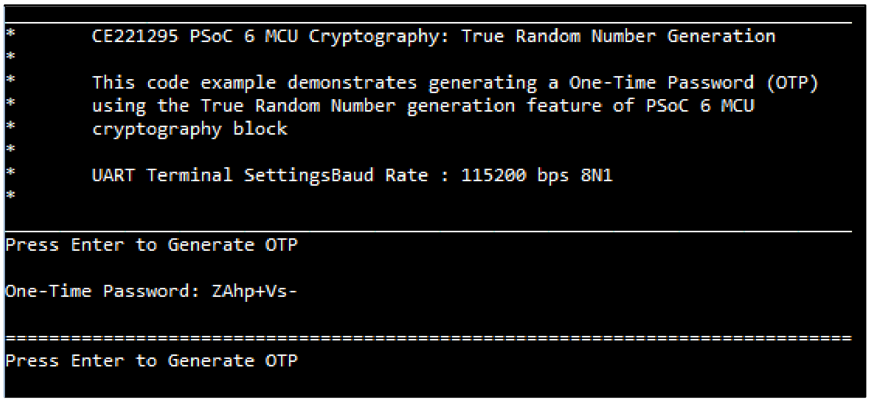
This example uses Tera Term as the UART terminal for displaying the generated OTP. If you don't have one, install one. This example uses Tera Term.

Operation

1. Connect the kit to your PC using the provided USB cable.
2. Import the code example into a new workspace. See KBA225201.
3. Program the PSoC 6 MCU device. In the project explorer, select the **mainapp** project. In the Quick Panel, scroll to the **Launches** section and click the **Program (KitProg3)** configuration. Program configurations also build the code.
4. Open Tera Term and connect to the USB-UART bridge COM port. Set the connection to 115200 bps, 8N1.
5. Press the **Enter** key to generate an OTP. The generated OTP will be displayed on the UART terminal. Note that you must press the **Enter** key every time when you need to generate an OTP.

Figure 1 shows a sample output as displayed on the Tera Term UART Terminal.

Figure 1. Sample Output as Displayed on Tera Term



```
*       CE221295 PSoC 6 MCU Cryptography: True Random Number Generation
*
*       This code example demonstrates generating a One-Time Password (OTP)
*       using the True Random Number generation feature of PSoC 6 MCU
*       cryptography block
*
*       UART Terminal SettingsBaud Rate : 115200 bps 8N1
*

Press Enter to Generate OTP

One-Time Password: ZAhp+Vs-

===========================================================================
Press Enter to Generate OTP
```

Design and Implementation

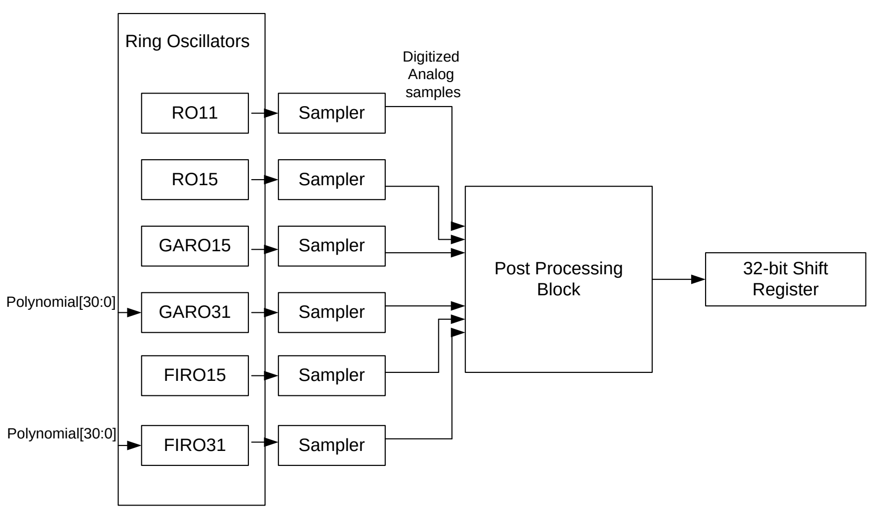
Random number generation is the generation of a sequence of numbers or symbols that cannot be predicted based on the previous knowledge of the generated sequence. Random number generators have applications in cryptography, statistical sampling, gambling, and other areas where producing an unpredictable result is desirable.

A true random number is generated using a hardware random number generator that generates random numbers from a physical process. The true random number generator (TRNG) in PSoC 6 MCU generates true random numbers of programmable bit size ranging from 0 to 32 bits. The TRNG relies on up to six ring oscillators to provide physical noise sources namely:

- Two fixed ring oscillators consisting of 11 and 15 inverters (RO11 and RO15).
- A fixed Galois-based ring oscillator (GARO15) and a fixed Fibonacci-based ring oscillator (FIRO15) each consisting of 15 inverters.
- A flexible Galois-based (GARO31) and a flexible Fibonacci-based oscillator (FIRO31) consisting of 31 inverters with a programmable polynomial of up to order 31.

A ring oscillator consists of a series of inverters connected in a feedback loop to form a ring. Due to (temperature) sensitivity of inverter delays, jitter is introduced on a ring's oscillating signal. The jittered oscillating signal is sampled to produce a digitized analog signal (DAS). This is done for all multiple ring oscillators. To increase entropy and to reduce the bias in DAS bits, the DAS bits are further post-processed. Post-processing produces bit samples that are considered true random bit samples. The true random bit samples are shifted into a register to provide random values of up to 32 bits. Figure 2 shows an overview of how generation of true random generation is implemented in PSoC 6 MCU.

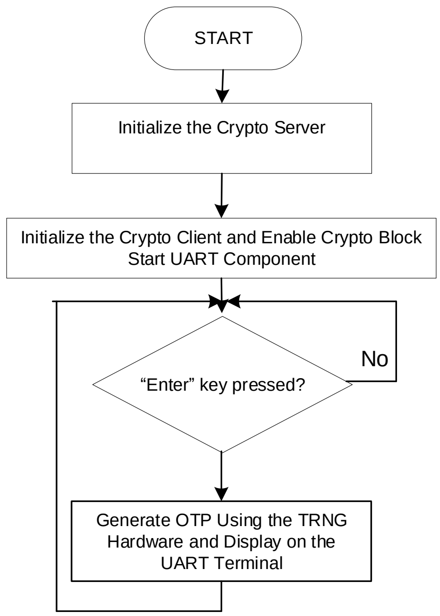
Figure 2. True Random Number Generation in PSoC 6 MCU



Cryptographic operation in this example is based on a Client-Server model. In this example, both the Crypto Server and the Crypto Client runs on the CM4 CPU. The firmware initializes and starts the Crypto Server. The firmware then provides the configuration data required for generation of true random numbers and requests the Crypto Server to run the cryptographic operation.

In this example, an OTP of eight characters in length is generated. Using the true random number generator, a random number is generated corresponding to each character of the OTP. The generated random number is such that it corresponds to alpha-numeric and special characters of the ASCII code. The generated OTP is then displayed on a UART terminal emulator. Each time, the firmware waits for the user to press the “Enter” key to generate a new OTP.

Figure 3. Firmware Flowchart



Resources and Settings

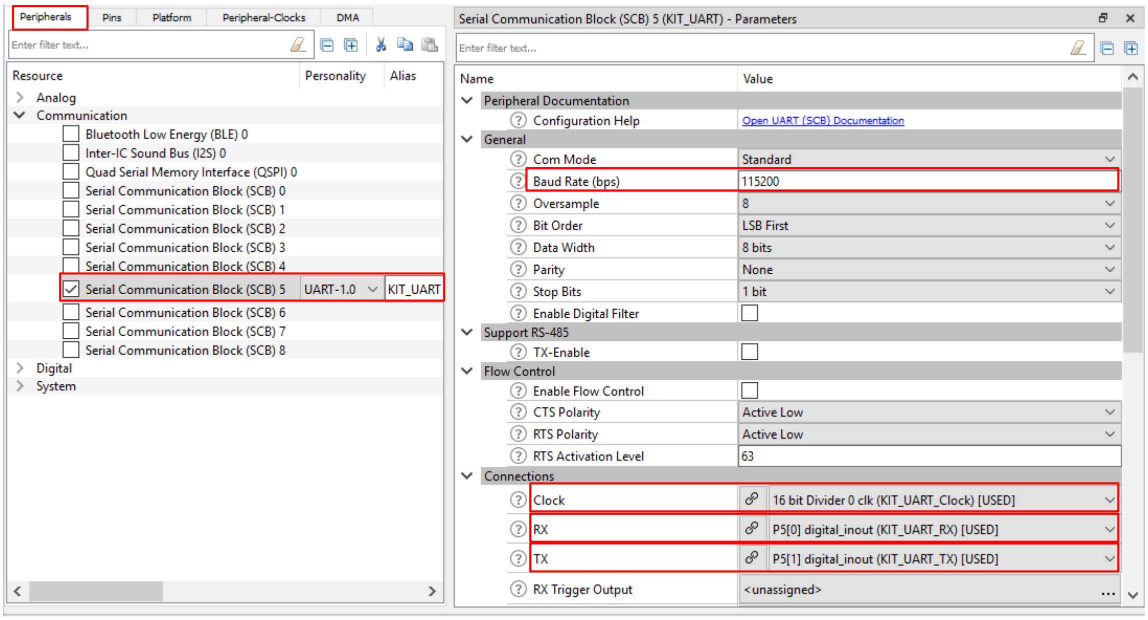
Table 1 lists the ModusToolbox resources used in this example, and how they are used in the design. For pin usage and configuration, open the **Pins** tab of the *design.modus* file.

Table 1. ModusToolbox Resources

Resource	Alias	Purpose
Serial Communication Block (SCB) – UART mode	KIT_UART	Send to and receive data from the UART Terminal.

Figure 4 highlights the non-default settings for the SCB (UART mode).

Figure 4. SCB (UART mode) Configuration



Reusing This Example

This example is designed for the supported kits. To port the design to a different PSoC 6 MCU device, right-click an application project and choose **Change Device**. If changing to a different kit, you may need to reassign pins.

Table 2. Device and Pin Mapping Table Across PSoC 6 MCU Kits

Kit Name	Device Used	KIT_UART_RX	KIT_UART_TX
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	P5[0]	P5[1]
CY8CKIT-062-WiFi-BT	CY8C6247BZI-D54	P5[0]	P5[1]
CY8CPROTO-062-4343W	CY8C624ABZI-D44	P5[0]	P5[1]

In some cases, a resource used by a code example (for example, an IP block) is not supported on another device. In that case, the example will not work. If you build the code targeted at such a device, you will get errors. See the device datasheet for information on what a particular device supports.

Related Documents

For a comprehensive list of PSoC 6 MCU resources, see KBA223067 in the Cypress community.

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices.
AN215656 – PSoC 6 MCU: Dual-CPU System Design	Describes the dual-CPU architecture in PSoC 6 MCU, and shows how to build a simple dual-CPU design
Code Examples	
Visit the Cypress GitHub site for a comprehensive collection of code examples using ModusToolbox IDE	
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual

Development Kits	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	
CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit	
CY8CPROTO-062-4343W PSoC 6 Wi-Fi BT Prototyping Kit	
Tool Documentation	
ModusToolbox IDE	The Cypress IDE for IoT designers

Cypress Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right device, and quickly and effectively integrate the device into your design.

For PSoC 6 MCU devices, see KBA223067 in the Cypress community for a comprehensive list of PSoC 6 MCU resources.

Document History

Document Title: CE221295 – PSoC 6 MCU Cryptography: True Random Number Generation

Document Number: 002-25946

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6482073	VKVK	02/21/2019	New code example

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 MCU

Cypress Developer Community

Community | Code Examples | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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