H-DA
Electrical Engineering and Information Technology

Embedded Security Showcase on PSoC64(ESSOP)

Getting started PSoC64

Revision	Date	Editor	Reason
1.0	*05.24.2020*	*Vaishnavi Sankaranarayanan*	*Steps for Getting started with PSoC64*
1.1	*08.03.2020*	*Aadarsh Kumar Singh*	*Added pre-requisites and Missing installation and setup hints.*

Table of Contents

1. Software Installation

The tools required for PSOC6 are:

- ModusToolBox: https://www.cypress.com/products/modustoolbox-software-environment
- Mbed OS: https://os.mbed.com/docs/mbed-os/v5.13/tools/installation-and-setup.html
- Python 3.8.1: https://www.python.org/downloads/release/python-381/

Important steps for installation:

- During Installation of Python change the path to C:/Python37
- Edit Environment variable "path" and add the Python.exe location and Python "home folder"/Scripts

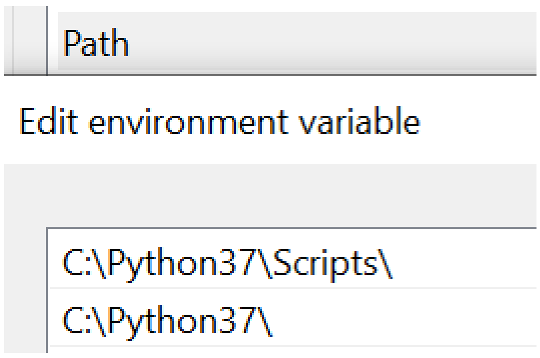
Path
Edit environment variable
C:\Python37\Scripts\
C:\Python37\

Figure 1 .1: Environment Variable for python script 1

- After successful installation of Python in Windows PowerShell run command to update mbed cli.

  pip install -U mbed-cli

  Note: Its necessary to make sure all mbed cli support packages are installed,if not manually run the below commands

  pip install hidapi
  pip install pywin32
  pip install mbed_cloud_sdk
  pip install jinja2
  pip install mbed_ls
  pip install mbed_host_tests
  pip install mbed_greentea
  pip install manifest_tool
  pip install icetea
  pip install pycryptodome
  pip install hidap
  pip install pywin32

pip install wmi
pip install psutil

- Install Cysecuretools in python using the cmd,
  pip install cysecuretools

Prerequisites for cysecuretools:

- Python 3.6 or later – Done if followed the above procedure
- **Installed the libusb driver** – Important that libusb has to be installed.
  (**Steps on Windows**:
  - Download and unzip libusb-1.0.21.7z from https://github.com/libusb/libusb/releases/tag/v1.0.21
  - Run the following command to determine if a Python shell is executing in 32-bit or 64-bit mode on the OS: python -c "import struct; print(struct.calcsize('P') * 8)"
  - Copy *libusb-1.0.dll* file into the Python root folder (in same folder with *python.exe*). Use the 64-bit version of DLL for the 64-bit Python (MinGW64 directory) and the 32-bit version of DLL for the 32-bit Python (MinGW32 directory).
  - Ensure the Python path is located at the beginning of the Path environment variable.)

2. Board Information

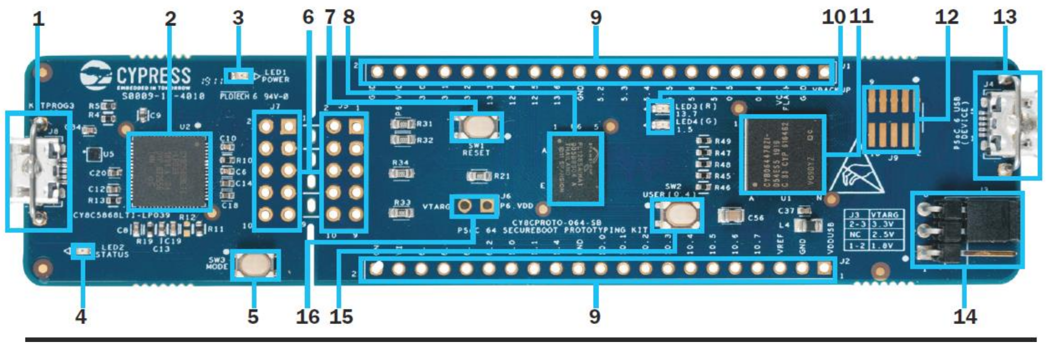

Fig 2.1: PSOC 64 Board Information

The PSoC 64 Secure Boot Kit Board has the following peripherals:

1. **KitProg3 USB connector (J8):**

   The USB cable provided along with the board connects between this USB connector and the PC. It is used to power the entire board as well as use the onboard KitProg3 programmer/debugger.

2. **KitProg3 (PSoC 5LP) programmer and debugger (CY8C5868LTI-LP039, U2):**

   The PSoC 5LP device (CY8C5868LTI-LP039) serving as KitProg3, is a multi-functional system, which includes a SWD programmer, debugger, USB-I2C bridge, and USB-UART bridge. KitProg3 also supports custom applications. For more details, see the KitProg3 User Guide.

3. **Power LED (LED1):**

   LED1 is an amber LED that indicates the status of power supplied to the board. It is only applicable when powered using KitProg3.

4. **KitProg3 status LED (LED2):**

   Amber LED (LED2) indicates the status of KitProg3. For details on the KitProg3 status, see the KitProg3 User Guide.

5. **KitProg3 programming mode selection button (SW3):**

   This button can be used to switch between various modes of operation of KitProg3 (CMSIS-DAP/Bulk, CMSIS-DAP/HID mode and DAPLink mode). For more details, see the KitProg3 User Guide.

6. **KitProg3 interface header (J7, J5):**

   This header brings out the SWD, USB-UART and USB-I2C interface of the KitProg3. This is used to program and debug the PSoC 64 MCU. If KitProg3 section is broken away, it can be used to program any device over the 5-pin SWD interface. Note that VTARG is an input to KitProg3, and therefore the target must be powered externally. The 10-pin connector includes a pin for VBUS used to provide 5 V to the on-board regulator in the PSoC 64 MCU section of the board. For more details on the KitProg3, see the KitProg3 User Guide.

7. **PSoC 64 MCU reset button (SW1):**

   This button is used to reset the PSoC 64 MCU. This button connects the PSoC 64 MCU reset (XRES) pin to ground.

8. **Cypress 128-Mbit serial NOR flash memory (S25FL128, U6):**

   The S25FL128SAGBHIA10 NOR flash of 128-Mbit capacity is connected to the serial memory interface (SMIF) of the PSoC 64 MCU. The NOR device can be used for both data and code memory with execute-in-place (XIP) support and encryption.

9. **PSoC 64 MCU I/O header (J1, J2):**

   These headers provide connectivity to PSoC 64 MCU GPIOs. Most of these I/Os are also connected to on-board peripherals.

10. **PSoC 64 MCU user LEDs (LED3, LED4):**

    The red LED (LED3) can operate at the entire operating voltage range of the PSoC 64 MCU, whereas the green LED (LED4) works only at 2.5 V and higher. Both LEDs are active LOW, so the pins must be driven to ground to turn ON the LED.

11. **PSoC 64 MCU (CYB06447BZI-D54, U1):**
    CYB06447BZI-D54 is a PSoC 64 Secure Boot MCU with 896 KB of flash and 184 KB of SRAM. The chip features out-of-the-box security functionality, providing an isolated root-of-trust with true attestation and provisioning services.

12. **PSoC 64 MCU program and debug header (J9):**
    This 10-pin header allows you to program and debug the PSoC 64 MCU using an external programmer such as MiniProg4. Note that this is not loaded by default.

13. **PSoC 6 USB device Connector (J4):**
    The USB cable provided with the Prototyping Kit can also be connected between this USB connector and the PC to use the PSoC 64 MCU USB device applications.

14. **System Power selection jumper (J3):**
    This switch is used to select the PSoC 64 MCU's supply voltage (P6.VDD) between three voltages: 1.8 V, 2.5 V, and 3.3 V.

15. **PSoC 64 MCU user button (SW2):**
    This button can be used to provide an input to the PSoC 64 MCU. Note that by default, the button connects the PSoC 64 MCU pin to ground when pressed, so you need to configure the PSoC 64 MCU pin as a digital input with resistive pull-up for detecting the button press. This button also provides a wake-up source from low-power modes of the device.

16. **PSoC 64 MCU Current measurement header (J6):**

This header can be used to measure the current consumption of PSoC 6 using an ammeter. It is not loaded by default and is bypassed using a zero-ohm resistor across the two pins.

3. Modes Of Operation

The PSOC6 feature an OnBoard Programmer/Debugger (KitProg3).KitProg3 supports CMSIS-DAP and DAPLink for programming the target MCU using SWD.KitProg3 uses industry-standard CMSIS-DAP V2.0.0 and V1.2.0 as the Bulk and HID endpoints transport mechanisms .KitProg3 programming mode selection button (SW3) is used to switch between different Selection modes (object 5 from figure 2.1)

4. Board Provisioning and project setup Eclipse.

1. Import MbedOs example ,In the Windows Powershell import the example blinky project using the command line :

mbed import mbed-os-example-blinky

Navigate to the source code folder, check version and upgrade if it’s lower than 5.14 using the below command line:

mbed ls
mbed update mbed-os-5.14

2. The Next step would be to prepare your local mbed workspace and to create new keys and provisional packets. To achieve this Run “**UnlockProcedure.bat “**
Note: UnlockProcedure.bat can be executed only through command prompt.

Mode Of Operation	**Mode Description**	**Mode Switch**	**Mode Detection**
CMSIS-DAP Bulk(default)	KitProg3 implements USB Bulk endpoints for faster communication and hence Programming is faster.	Press Button SW3 once to go into CMSIS-DAPBulk Mode. This is the default mode of operation.	The status LED (Amber) is always ON in Bulk mode.
CMSIS-DAP HID	KitProg3 also supports HID endpoints for use cases that require them, but communication is slower	When in CMSIS-DAPBulk Mode press the button SW3 again to enter into CMSIS-DAP HID mode	The status LED (Amber) is ramping ON/OFF at 1-Hz rate in HID mode (Slow Blinking).
DAPLink	This Mode is used to program the application onto the Board	When in CMSIS-DAPBulk Mode press the button SW3 twice to enter into DAP Link mode	The status LED (Amber) is ramping ON/OFF at 2-Hz rate in HID mode(Fast Blinking**).**

ATTENTION: Remove Jumper J3 to supply the chip with 2.5V and plug in the kit to the PC. The 2.5V supply is necessary for Step 6, where PSoC64 eFuses are blown. KitProg3 must be in CMSIS-DAP HID mode for provisioning. The Status LED (LED2) will be ramping ON/OFF slowly (~1Hz) in this mode. Press and release the Mode button (SW3) one or more times until the KIt¬Prog3 is in CMSIS-DAP mode.

3. Next step is to perform provisioning.
   ATTENTION: PSoC 6 supply voltage of 2.5V is required to perform provisioning. The 2.5V requirement is because this step involves blowing eFuses to change the device lifecycle to SECURE CLAIMED. If the chip is not at 2.5V, it may cause provisioning to fail and permanently lock the chip in a dead state.

   Run command in power shell:
   python -c "from cysecuretools import CySecureTools; tools = CySecureTools('CY8CPROTO-064S1-SB', 'policy/policy_single_stage_CM4.json'); tools.provision_device();"

4. Next step is to build and program the application
   - In order to build and program the project, move the kit to DAPlink mode by pressing and releasing the mode button (**SW3**) one or more times until the Status LED (LED2) is ramping ON/OFF at a fast (~2Hz) rate.
   - Navigate back to your *mbed-os-example-blinky* folder.
   - Build and program the application using the following command (Note: the -f option programs the device after the build completes. Omit that option if you want to build without programming.

   mbed compile -m CY8CPROTO_064_SB -t GCC_ARM -f

   Note :When using the target CY8CPROTO_064_SB, Mbed OS has a post-build signing script which uses the following key to sign the output image:<mbed-os>\targets\TARGET_Cypress\TARGET_PSOC6\sb-tools\keys\USERAPP_CM4_KEY_PRIV.pem

5. If the PSoC64 is successfully programmed, observe the Red LED blinking at 1 Hz. In addition, mbed-OS starts are printed every 10 seconds over the Kitprog3 UART at 9600 baud.

6. The security keys that has been flashed can be found in the location:

   C:\Users\Aadarshxp\mbed-os-example-blinky\mbed-os\targets\TARGET_Cypress\TARGET_PSOC6\sb-tools\keys

   Contents of the folder:
   1- Public key : dev_pub_key.json , dev_pub_key.pem
   2- Private Key : USERAPP_CM4_KEY.json , USERAPP_CM4_KEY_PRIV.pem

7. Copy the files dev_pub_key.pem and USERAPP_CM4_KEY_PRIV.pem in the python path:
   C:\Python37\Lib\site-packages\cysecuretools\targets\cy8cproto_064s1_sb\keys

This is the place from where the application running on eclipse will read the keys.

8. Clone the SVN project from the repository and open the project in the eclipse IDE.

9. Build the project and after the build has successfully finished run sign_hex build target.

10. Now copy the hex file from build folder and paste in the MSD of the PSoC64.

5. Dos/Donts and Hints to use PSOC 6

- The CY8CPROTO-064S1-SB PSoC 64S1 Secure Boot Kit is sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static-free surface.

- Make sure to connect to KitProg3 USB connector while connecting to PC instead of PSoC 6 USB device Connector.

- Prior to running any code on the PSoC 64 line of secure MCUs, they must first be provisioned with keys and device security policies so only signed code can be executed.

- While creating new keys If 'encrypt' is enabled, then an additional 128-bit AES key is generated which can be used to encrypt the application image. The 'encrypt' field is present in the policy_single_stage_CM4.json file.

- Before Running the EntranceExam make sure to remove Jumper J3 to supply the chip with 2.5V and plug in the kit to the PC. The 2.5V supply is necessary for provisioning the keys, where PSoC64 eFuses are blown. KitProg3 must be in CMSIS-DAP HID mode for provisioning. The Status LED (LED2) will be ramping ON/OFF slowly (~1Hz) in this mode. Press and release the Mode button (SW3) one or more times until the KItProg3 is in CMSIS-DAP mode.Failure to Do this step may permanently lock the Board.

- Ensure that you are using the same key that you used to provision the device; otherwise PSoC64 Secure Boot will fail. Success or Failure of the Secure Boot process can be observed by opening the Kitprog3 COM port with a terminal application of your choice.

- While Building and programming the code onto the board if you get If you, "Target board you compiled for is not connected to your system" run the following command to add the kit and then try programming again:
  mbedls --mock 1907:CY8CPROTO_064_SB