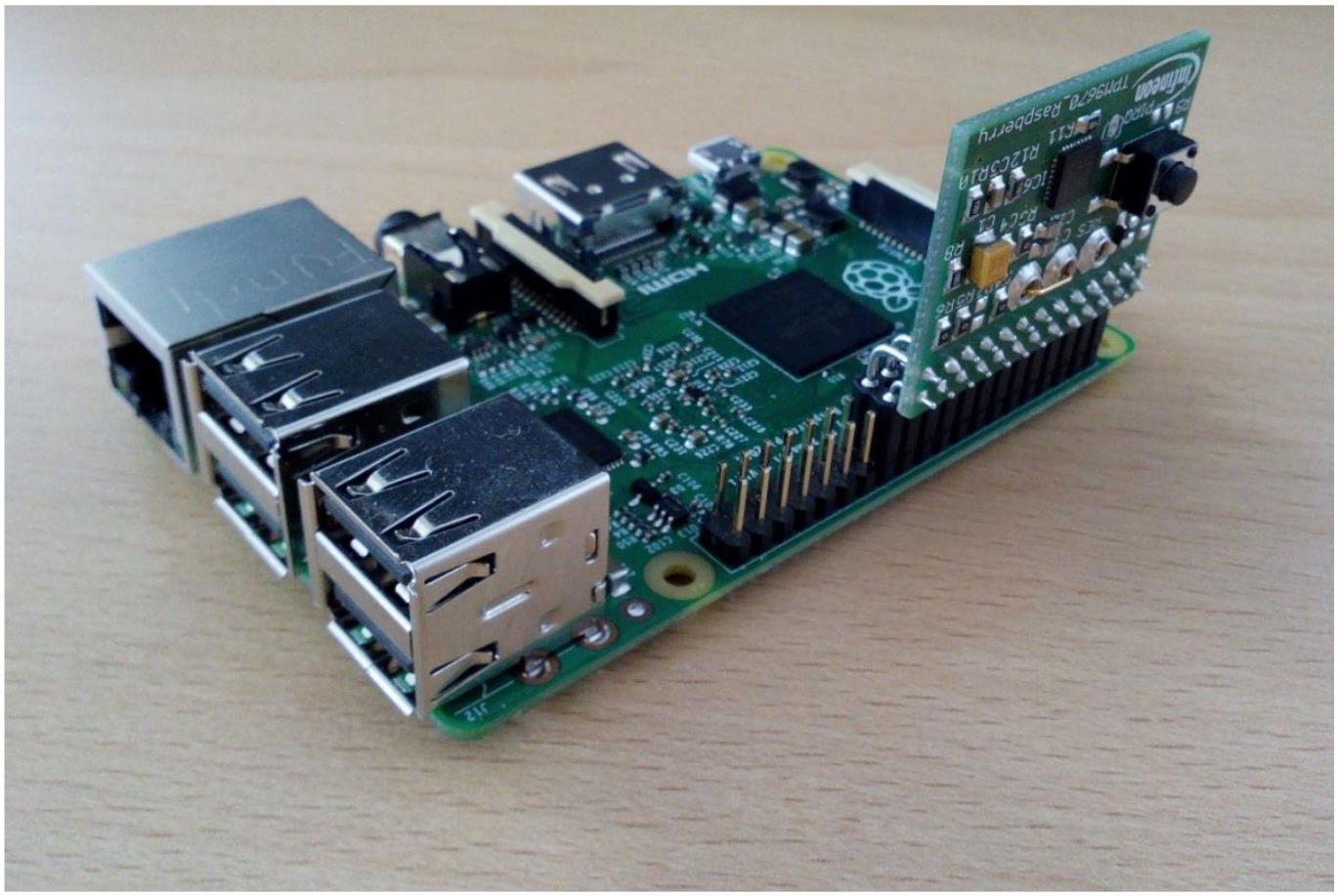
OPTIGA™ TPM Application Note

Integration of an OPTIGA™ TPM SLx 9670 TPM2.0 with SPI Interface in a Raspberry Pi® 3 Linux environment with integrated TPM Driver

Devices

- OPTIGA™ TPM SLB 9670 TPM2.0
- OPTIGA™ TPM SLI 9670 TPM2.0
- OPTIGA™ TPM SLM 9670 TPM2.0



About this document

Scope and purpose

This document explains how the OPTIGA™ TPM SLx TPM2.0 can be used on a Raspberry Pi® 3. The used Linux kernel does have driver support integrated for TPM 2.0. Thus the Linux kernel version which is described in this document is version 4.14.

The OPTIGA™ TPM SLx 9670 TPM2.0 uses a SPI interface to communicate with the Raspberry Pi®. The OPTIGA™ TPM SLx 9670 TPM2.0 product family with SPI interface consists of 3 different products:

- OPTIGA™ TPM SLB 9670 TPM2.0 standard security applications
- OPTIGA™ TPM SLI 9670 TPM2.0 automotive security applications
- OPTIGA™ TPM SLM 9670 TPM2.0 industrial security applications

We refer with “OPTIGA™ TPM SLx 9670 TPM2.0” to all of the above 3 members of the OPTIGA™ TPM 2.0 product family with SPI interface.

OPTIGA™ TPM SLx 9670 TPM2.0 products are fully TCG compliant TPM products with CC (EAL4+) and FIPS certification. The OPTIGA™ TPM SLx 9670 TPM2.0 products standard, automotive and industrial differ with regards to supported temperature range, lifetime, quality grades, test environment, qualification and reliability to fit the target applications requirements. For more details refer to Infineon’s website [1]. Since all 3 OPTIGA™ TPM SLx 9670 TPM2.0 products are fully TCG compliant with regards to the SPI and software interface, the described steps to integrate a TPM 2.0 in a Raspberry Pi® 3 Linux environment are the same and valid for all 3 variants. We are referring in this document also to the OPTIGA™ TPM SLx 9670 TPM2.0 by using simply “TPM” and to the Raspberry Pi® 3 by using “RPi”. An overview of all Infineon OPTIGA™ TPM products can be found on Infineon’s website [2]. More information about the TPM in general and how to integrate it into a platform can be found in the corresponding specifications of the Trusted Computing Group (TCG) in reference [3].

The described steps to integrate an OPTIGA™ TPM in a Raspberry Pi® 3 Linux environment can be performed with one of the Infineon Iridium SLx 9670 TPM2.0 SPI Boards, listed in the Table below.

Iridium Boards:

Supported TPM	Order type	Order number:
OPTIGA™ TPM SLB 9670 TPM2.0	IRIDIUM 9670 TPM2.0	SP001596592
OPTIGA™ TPM SLI 9670 TPM2.0	IRIDIUM SLI 9670 TPM2.0	SP004232000
OPTIGA™ TPM SLM 9670 TPM2.0	IRIDIUM SLM 9670 TPM2.0	SP004232004

The 3 Infineon Iridium Boards are referred in the following as “Infineon Iridium SLx 9670 TPM2.0 SPI Board”

Intended audience

This document is intended for customers who want to increase the security level of their embedded platforms using a TPM 2.0 and like to evaluate how to start with the TPM software integration for their target applications.

Table of contents

1 Introduction

1.1 Motivation

Two of the basic principles of information security when exchanging data are confidentiality and authenticity. While the first one is obvious to almost everyone, the second one often does not get the appropriate amount of attention. Authenticity is just as important as confidentiality, because for example the highest confidentiality of secret data is worth nothing in case the secret data come from or is shared with the wrong communication partner.

A TPM adds additional security features to a platform, which will not only help in gaining confidentiality but also authenticity of exchanged data or a communication partner.

One of the most important features to achieve these goals is that sensitive data such as cryptographic keys or secrets can be stored inside the TPM, where the data is protected by the TPM's hardware from unauthorized access or manipulation. This allows confidentiality of the data on a high level. But in addition to this protection, the data can also be bound to a single TPM and thus to the platform hosting this TPM. This in turn can be used to bring in another authenticity factor, since only the platform or user operating the correct TPM is able to provide the correct authentication secret required to access the protected sensitive data.

Another basic feature of the TPM is that it can function as a starting point for establishing a root of trust, which allows the detection of unauthorized modifications to a platform's hardware or software.

Additionally, the TPM can be used to perform several cryptographic operations on data in hardware, decreasing the vulnerability to several kinds of attacks, such as reading out unencrypted sensitive data from the platform's memory.

1.2 Scope

This document is intended for users to help them getting familiar how to use a subset of the TPM's functionalities on embedded platforms running Linux. The final goal in this document is to interact with the TPM 2.0 to be able to run the test scripts provided in the GitHub project [4].

The topics described in this document are:

- The Evaluation Board (Figure 2) with its expansion header for the Raspberry Pi® 3.
- Usage of a TPM 2.0 on a Raspberry Pi® 3.

This document has been generated using kernel version 4.14.

In this application note we use a Raspberry Pi® 3 Linux distribution based on Debian® to describe the system setup and the application of the TPM 2.0. If not explicitly said otherwise, the application note refers to specific versions of the required Linux packages to prevent compatibility problems. The mentioned versions were publicly available and downloaded at the time of release from the specified links and repositories.

1.3 Command conventions

In this document, the operating system Linux will be used. Since some commands in this document are directly interacting with hardware or accessing protected system files, the commands need to be executed with root privileges (also known as administrative privileges). The convention in this document will be that whenever a command requires root privileges, the actual command is preceded by `"#"`, whereas commands requiring only normal user privileges are preceded by `"$"`:

```
# [command to be executed as root]
```

And

```
$ [command to be executed as user]
```

1.4 Acronyms and Abbreviations

Acronym	Explanation
BIOS	Basic Input / Output System
EK	Endorsement Key
FTD	Flattened Device Tree
LTS	Long Term Support
PCR	Platform Configuration Register
RNG	Random Number Generator
RPi	Raspberry Pi ®
SoC	System on a Chip
SPI	Serial Peripheral Interface
SRK	Storage Root Key
SSH	Secure Shell
SSL	Secure Sockets Layer
TCG	Trusted Computing Group
TCS	TCG Core Services
TCSD	TCS Daemon
TLS	Transport Layer Security
TPM	Trusted Platform Module
TSS	TCG Software Stack
UEFI	Unified Extensible Firmware Interface
VPN	Virtual Private Network

2 Hardware Setup

The required hardware to perform the steps described in this document consists of:

- Developer PC: This platform is used for patching the Kernel, maintaining and interacting with the Raspberry Pi® 3 in a more convenient and faster way compared to doing all actions directly on the Raspberry Pi® 3. The hardware requirements for the developer PC are:
    - Desktop computer or laptop with x86 architecture and USB 2.0 (or higher)
    - Capable of running Linux, for example Ubuntu® 18.04
    - Internet connection
- Infineon Iridium SLx 9670 TPM2.0 SPI Board. This board contains the Infineon OPTIGA™ TPM SLx 9670 TPM2.0 mounted on an easy-to-use hardware board, which can be attached to the Raspberry Pi® 3.
- Raspberry Pi® 3:
    - Raspberry Pi® 3 Model B
    - Micro SD Card[1] with at least 8 GB
    - Micro-B USB cable for power supply
- MicroSD-Card Reader

2.1 Raspberry Pi® 3

The Raspberry Pi® 3 Model B has a Broadcom® BCM2837 SoC with a 1.2 GHz quad-core ARM™ Cortex-A53 CPU with ARMv7 architecture and 1 GB RAM. An HDMI connector can be used for graphical output. It also has an Ethernet controller for network connectivity and a micro USB port for power supply. A cold reboot of the platform can only be triggered if the Raspberry Pi® 3 is disconnected from power. Figure 1 shows the Raspberry Pi® 3 with all expansions. For more information visit the official website [5].

[1] Some card models may not work.

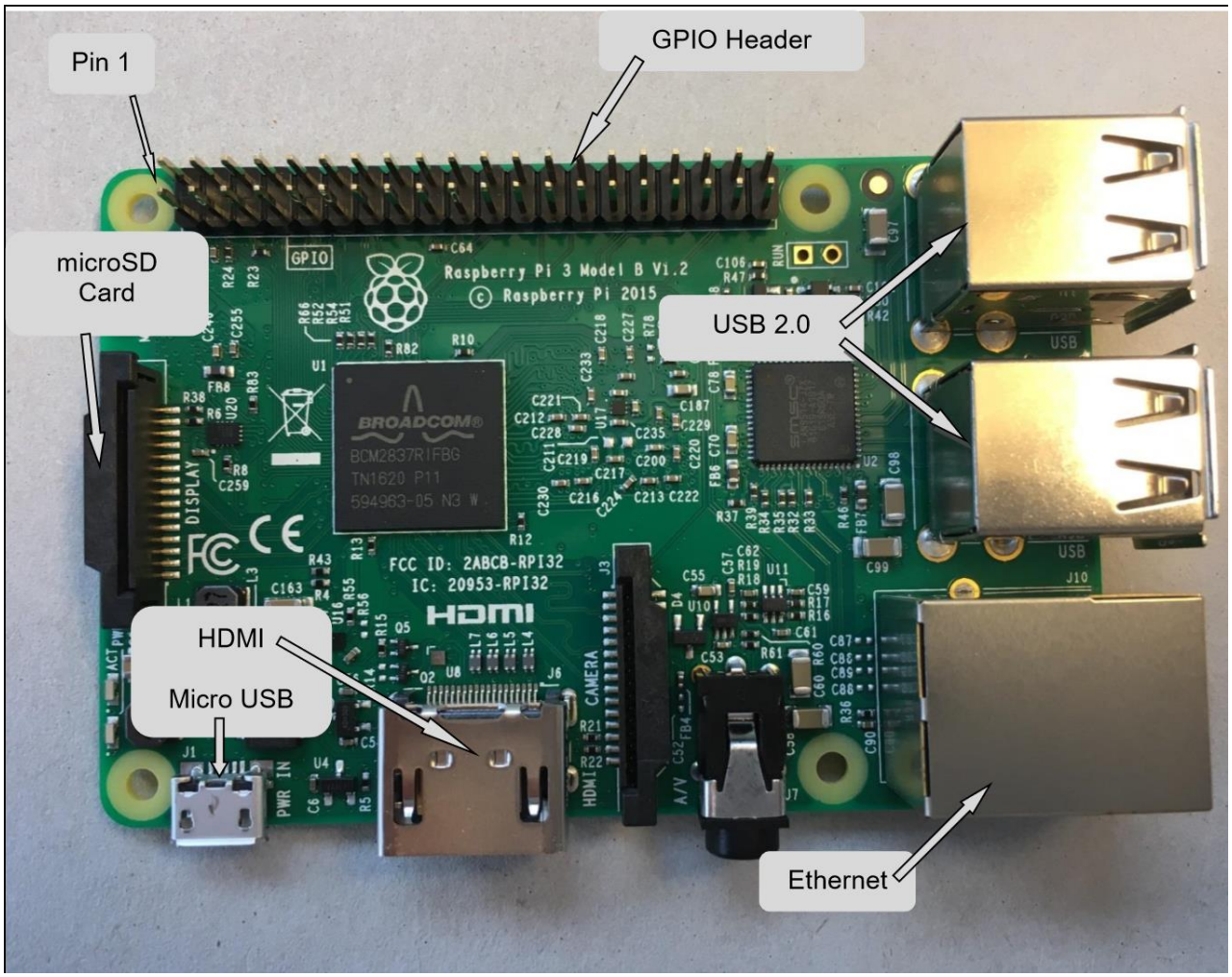


Figure 1 Raspberry Pi® 3 Expansions

2.2 Infineon Iridium SLx 9670 TPM2.0 SPI Board

The Infineon Iridium SLx 9670 TPM2.0 SPI Board can be connected to a Raspberry Pi® 3 via its extension header. For data transfer the board uses the SPI bus. The board is shown in Figure 2.

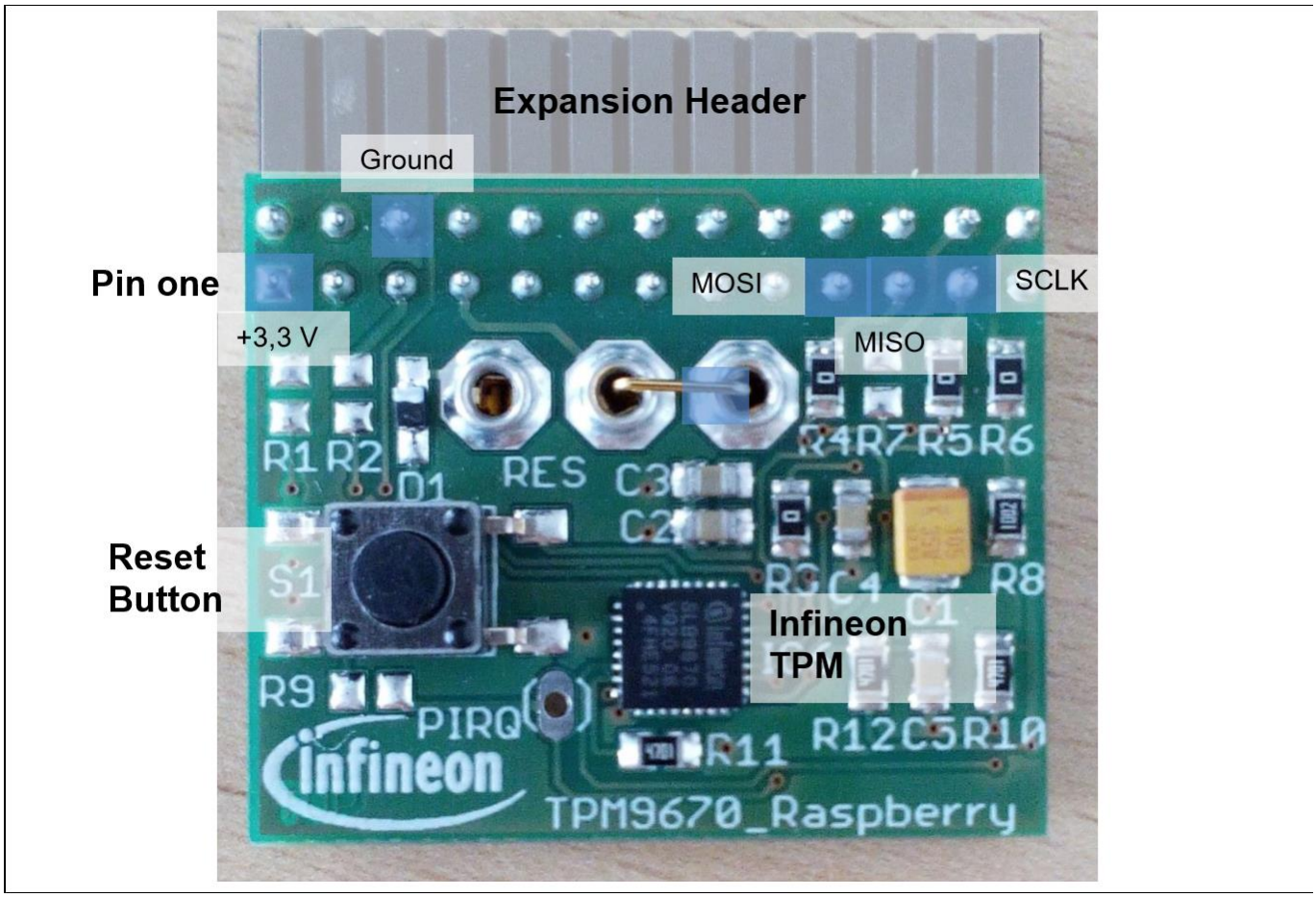


Figure 2 Infineon Iridium SLx 9670 TPM2.0 SPI Board

On the top is the Raspberry Pi® Header with 26 pins. The board contains a reset circuit on board, which pulls the reset line of the TPM to GND for the right amount of time after VCC becomes available. The reset of the module is active low and can also be set by connecting reset and ground via the reset button.

The Raspberry Pi® 3 Header has VDD, GND, MOSI, MISO and SCLK at the following pins:

Pin	1	6	19	21	23
Signal	VDD, 3.3 V	GND	MOSI	MISO	SCLK

Figure 3 shows how the Evaluation Board should be mounted on the Raspberry Pi® 3.

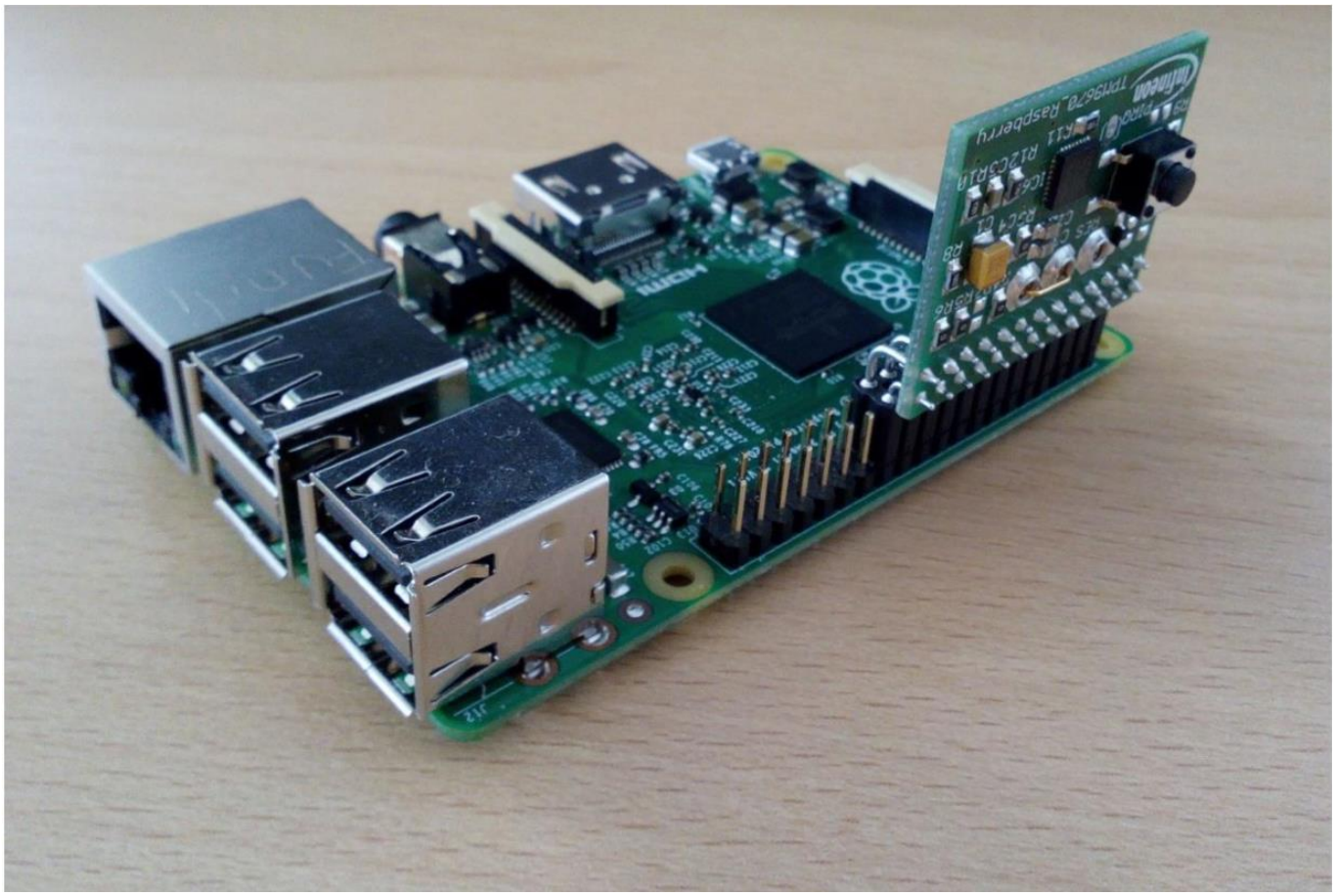


Figure 3 Infineon Iridium SLx 9670 TPM2.0 SPI Board on Raspberry Pi® 3

3 Software Setup

This section describes all necessary steps needed to use the TPM 2.0 with a Raspberry Pi® 3.

3.1 Developer PC installation

The developer PC can be either a native Linux PC or a virtual machine running Linux on a non-Linux machine (e.g. Windows). It will be used mainly for compiling and setting up the Linux Operating System for the Raspberry Pi® 3 on a microSD-Card. It can also be used to control the Raspberry Pi® 3 remotely via an SSH connection. It is recommended to use a Linux Distribution like Ubuntu®, Debian®, or similar on the developer PC. This document refers to the usage of a native machine running Ubuntu® 18.04, which can be downloaded at [6]. The basic installation of Ubuntu® comes by default with many packages installed. Beyond that, the following list shows all additional software packages required to be able to perform all steps in this application note.

- gddrescue
- git
- libncurses5-dev
- gcc-arm-linux-gnueabihf
- bc

First create a directory called "**raspberry-pi**"

Code Listing 1

```
001        $ cd ~/Documents
002        $ mkdir raspberry-pi
003        $ cd raspberry-pi
```

The package information should be updated:

Code Listing 2

```
001        # apt-get update
```

Now install all packages mentioned before with the following command in a terminal on the developer PC:

Code Listing 3

```
001        # apt-get install -y gddrescue git libncurses5-dev gcc-arm-
   linux-gnueabihf bc
```

3.2 Raspberry Pi® 3

In order to get the TPM 2.0 working on the Raspberry Pi® 3, the kernel must have driver support for the TPM 2.0. The following sections give detailed instructions on how to enable the TPM 2.0 driver. Therefore a SD card with the Raspbian Stretch on it is required, which can be downloaded at [7]. This SD card will be plugged in the developer PC and is used for Code Listing 11.

3.3 Installation of Linux

Follow the next steps accordingly:

Download these repositories (via command line or web browser) in raspberry-pi directory

Code Listing 4

```
001       $ git clone  https://github.com/raspberrypi/tools
002       $ export PATH=$PATH:~/tools
```

Code Listing 5

```
001       $ git clone --depth=1 https://github.com/raspberrypi/linux
002       $ cd linux
```

Set the default kernel configuration:

Note: Double-check to take ‘kernel seven’ and not ‘kernel seventeen’.

Code Listing 6

```
001       $ KERNEL=kernel7
002       $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
   bcm2709_defconfig
```

Enter the menuconfig to enable the TPM 2.0 Support:

Code Listing 7

```
001       $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- menuconfig
```

In the menuconfig navigate with the arrow keys and follow the instruction to the TPM Hardware Support described in **Code Listing 8**. Press M to enable it and enter the sub menu with [Enter]. Within the submenu activate the TPM Interface Specification as stated in **Code Listing 8**. Last step is to save the setup in the .config.

Code Listing 8

```
Device Drivers [Enter] →
Character devices [Enter] →
TPM Hardware Support [M] & [Enter] →
TPM Interface Specification 1.3 Interface / TPM 2.0 FIFO Interface - native
SPI [M]
```

Add the Device Tree Overlay:

Code Listing 9

```
001       $ cd arch/arm/boot/dts/overlays/
002       $ nano infineon-tpm-overlay.dts
```

Copy the code from **Code Listing 10** and insert into *infineon-tpm-overlay.dts*.

Code Listing 10

```
/*
 * Device Tree overlay for Infineon SLx 9670
*/

/dts-v1/;
/plugin/;
/ {
	compatible = "brcm,bcm2835", "brcm,bcm2708", "brcm,bcm2709";
	fragment@0 {
		target = <&spi0>;
		__overlay__ {
			status = "okay";
		};
	};
	fragment@1 {
		target = <&spidev1>;
		__overlay__ {
			status = "disabled";
		};
	};
	fragment@2 {
		target = <&spi0>;
		__overlay__ {
			/* needed to avoid dtc warning */
			#address-cells = <1>;
			#size-cells = <0>;

			slb9670: slb9670@0{
				compatible = "infineon,slb9670";
				reg = <1>;	/* CE1 */
				#address-cells = <1>;
				#size-cells = <0>;
				spi-max-frequency = <32000000>;
				status = "okay";
			};

		};
	};
};
```

Next step is to go back to the /linux directory and build the kernel with the device tree.

Code Listing 11

```
001        $ cd -
002        $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- zImage
   modules dtbs overlays/infineon-tpm.dtbo -j8
```

Plug in the SD card with the Raspian Stretch on it. Run the following commands:
Note: Commands 004 and 005 in Code Listing 12 need to be adapted. Run: *lsblk* and check for your SD card and edit the */sdb1* and */sdb2* accordingly.

Code Listing 12

```
001        $ mkdir mnt
002        $ mkdir mnt/fat32
003        $ mkdir mnt/ext4
004        # mount /dev/sdb1 mnt/fat32
005        # mount /dev/sdb2 mnt/ext4
006        # make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
   INSTALL_MOD_PATH=mnt/ext4 modules_install
007        # cp mnt/fat32/$KERNEL.img mnt/fat32/$KERNEL-backup.img
008        # cp arch/arm/boot/zImage mnt/fat32/$KERNEL.img
009        # cp arch/arm/boot/dts/*.dtb mnt/fat32/
010        # cp arch/arm/boot/dts/overlays/*.dtb* mnt/fat32/overlays/
011        # cp arch/arm/boot/dts/overlays/README mnt/fat32/overlays/
```

Open the config.txt and change following values:

Code Listing 13

```
001        $ nano mnt/fat32/config.txt
```

Ensure that the *dtparam=spi=on* is enabled (no # before)
Add *dtoverlay=infineon-tpm* below it.

Unmount the SD card:

Code Listing 14

```
001        # umount mnt/fat32
002        # umount mnt/ext4
```

To use the Infineon Iridium SLx 9670 TPM2.0 SPI Board, run Embedded Linux TPM Toolbox 2 (ELTT2) on GitHub [8] and follow the steps provided in the ReadMe.

In order to integrate the OPTIGA™ TPM SLx 9670 TPM2.0 into your application, consider using open source TPM2 Software Stack (TSS) [4]. This software stack is available on GitHub.

In case you have questions or further specific interest in OPTIGA™ TPM, please get in touch with your local sales.

References

[1] https://www.infineon.com/cms/de/product/evaluation-boards/iridium9670-tpm2.0-linux/

[2] http://www.infineon.com/tpm

[3] https://trustedcomputinggroup.org/resource/tpm-main-specification/

[4] https://github.com/tpm2-software

[5] https://www.raspberrypi.org/

[6] https://www.ubuntu.com/download/desktop

[7] https://www.raspberrypi.org/downloads/raspbian/

[8] https://github.com/Infineon/eltt2

Revision history

Page or Reference	Description of change
Revision 1.3, 2019-03-14	
	Support of OPTIGA™ TPM SLM 9670
Revision 1.2, 2019-01-11	
	Support of OPTIGA™ TPM SLI 9670
Revision 1.1, 2018-09-03	
	Fixed typos in section 3.3.
Revision 1.0, 2018-08-10	
	Initial Release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2019-03-14

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email:
dsscustomerservice@infineon.com

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.