OPTIGA™ Trust M2 ID2

Product Version: V2

About this document

Scope and purpose

The purpose of this document is to guide a beginner to demonstrate OPTIGA shell application software package with the OPTIGA™ Trust M2 ID2 ESP32-DevKitC V4. The scope is limited to OPTIGA™ Trust M2 ID2 ESP32-DevKitC V4 and its hardware and software components.

Intended audience

This document addresses: customers, solution providers, porting guide and system integrators.

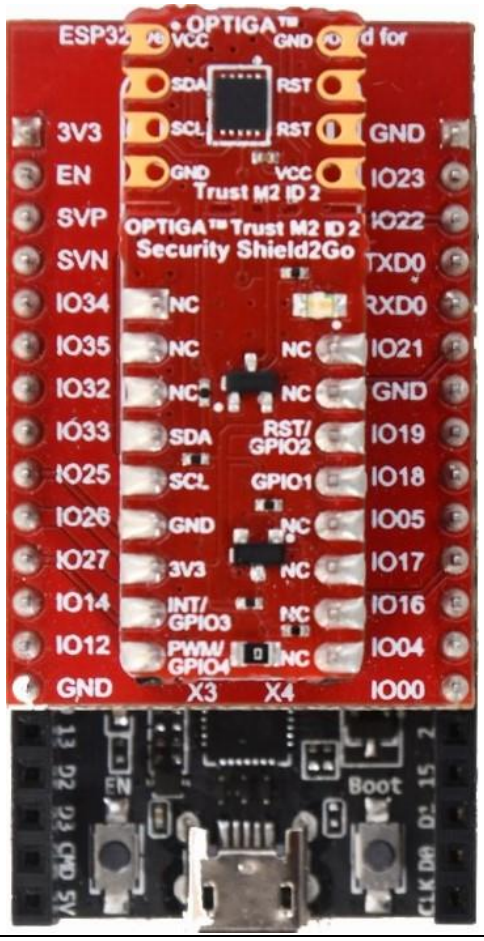

Table of Contents

1 Introduction

This document describes how to setup the environment to demonstrate OPTIGA shell application software package with the OPTIGA™ Trust M2 ID2 ESP32-DevKitC V4.

1.1 References

Table 1 References

Definition	Source
[1] ESP32-DevKitC V4_usermanual	espressif
[2] Infineon_I2C_Protocol	Infineon

1.2 Abbreviations

Table 2 Abbreviations

Abbreviation	Definition
API	Application Programming Interface
ESP32	ESP32-DevKitC V4
HW	Hardware
I2C	Inter Integrated Circuit
IoT	Internet of Things
OS	Operating System
PAL	Platform Abstraction Layer
RSA	Rivest-Shamir-Adleman
PC	Personal Computer
RST	Reset
SCL	Serial Clock
SDA	Serial Data
SW	Software
TTL	Transistor Transistor Logic
USB	Universal Serial Bus

2 OPTIGA™ Trust M2 ID2

OPTIGA™ Trust M2 ID2 is a security solution with a pre-programmed security controller with wide range of security features.

It supports secure data, key and metadata object update, hibernate and cryptographic toolbox functionalities, secure communication, platform integrity, data store protection and lifecycle management for connected device security.

2.1 OPTIGA™ Trust M2 ID2 with ESP32-DevKitC V4

OPTIGA™ Trust M2 ID2 with ESP32-DevKitC V4 is designed to provide all the components required to setup the environment to demonstrate the features of the OPTIGA™ Trust M2 ID2.

2.1.1 Evaluation Kit Components

Table 3 Evaluation Kit contents

No.	Item	Description
1	ESP32-DevKitC V4	Hardware Evaluation board for ESP32 microcontroller.
2	ESP32 DevKitC Adapter for Shield2Go	ESP32-DevKitC V4 compatible connector to add Shield2Go board on ESP32-DevKitC V4.
3	OPTIGA™ Trust M2 ID2 Security Shield2Go	ESP32 DevKitC V4 Adapter compatible Shield2Go board contains OPTIGA™ Trust M2 ID2 chip.
4	Micro USB to USB cable	The cable provides DC supply to ESP32-DevKitC V4 and to flash software.

2.2 Installed Software Components

The installed directory structure of OPTIGA™ Trust M2 ID2 setup software is shown below:

OPTIGA™ Trust M2 ID2

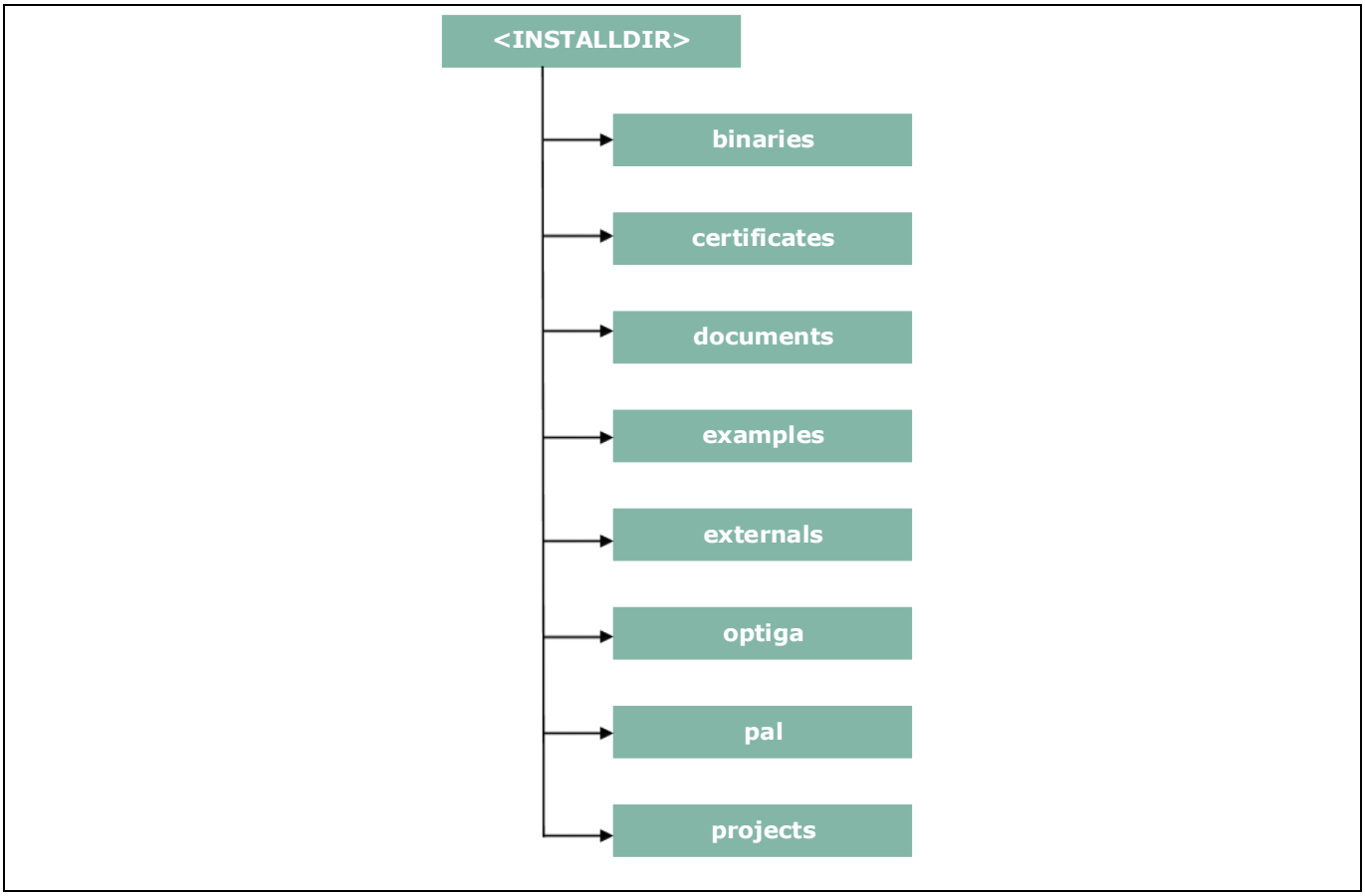

Figure 1 Installed directory structure

<INSTALLDIR> is the root directory to which the release package contents are extracted. The following section explains the contents of each subdirectory under installed directory:

1. **binaries** -- binaries for OPTIGA™ Trust M2 ID2 example application.
2. **certificates** – Place holder for OPTIGA™ Trust M2 ID2 certificates.
3. **documents** -- Relevant OPTIGA™ Trust M2 ID2 documentation.
4. **examples** – example use cases for Toolbox features and a tool for generation of manifest and fragment for protected update feature.
5. **externals** – alios and mbedtls software crypto library.
6. **optiga** -- OPTIGA™ Trust M2 ID2 libraries.
7. **pal** -- PAL for ESP32-DevKitC V4 device and PAL for mbedtls software crypto library.
8. **projects** -- ESP32-DevKitC V4 device example project.

Note: Package must be extracted in C:| drive to avoid any build errors.

3 System Setup

This section explains the basic components required for system setup.

3.1 System Overview

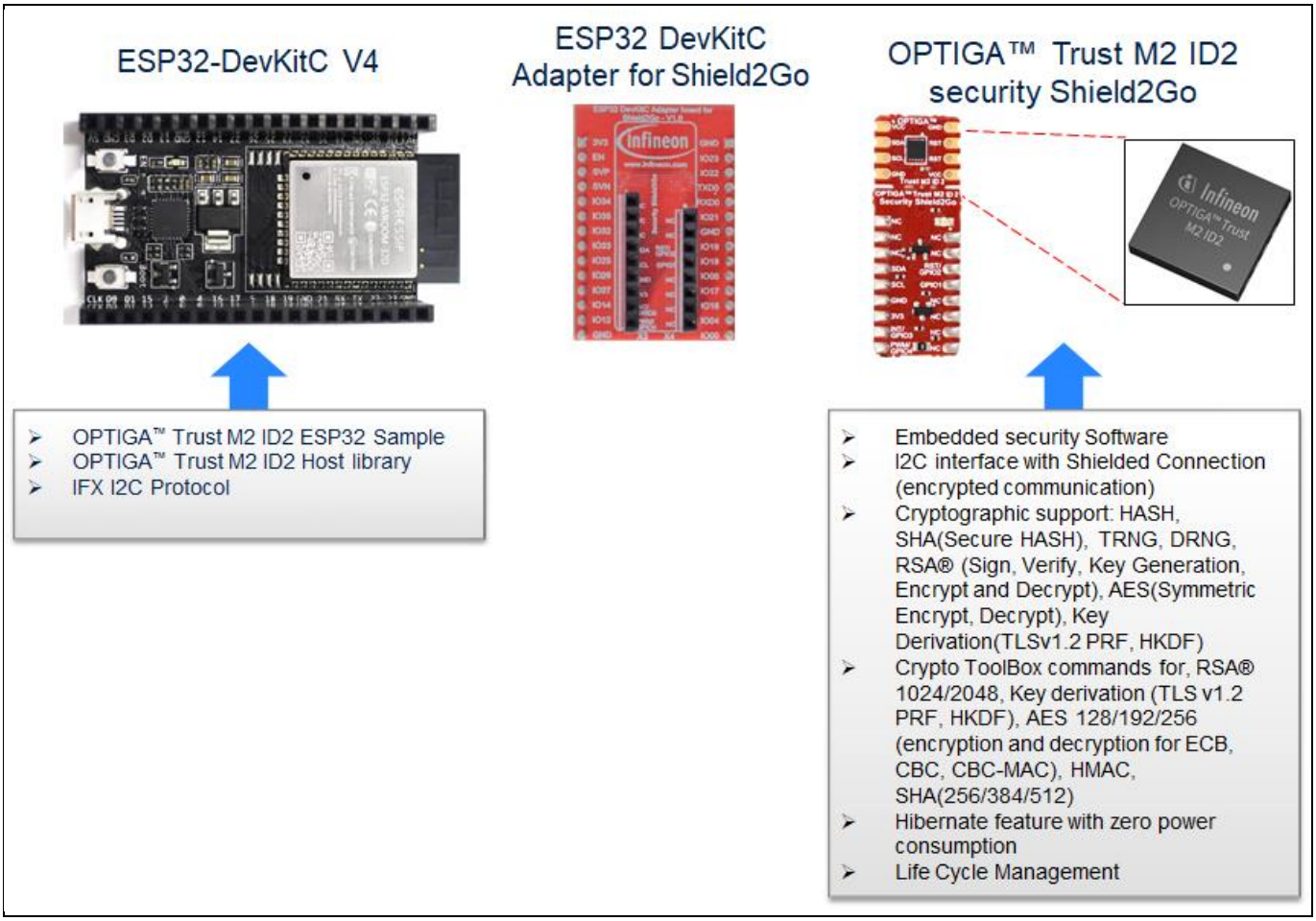

Figure 2 System Overview

This system consists of the following components:

1. ESP32-DevKitC V4
    - The ESP32-DevKitC V4 is an evaluation board with ESP32 Microcontroller from espressif. For more information refer document [1].
    - It is used as a reference platform to simulate the Host.
    - It interacts via I2C.
2. ESP32 DevKitC Adapter for Shield2Go
    - It acts as a gateway to add Shield2Go boards onto ESP32-DevKitC V4.
3. OPTIGA™ Trust M2 ID2 Security Shield2Go
    - Shield2Go board contains OPTIGA™ Trust M2 ID2 chip.

The following interface/connection is done among the above components:

- Micro USB data cable (with Data line) from PC is connected to ESP32-DevKitC V4 to supply power.

3.2 Hardware Setup

The hardware required to run OPTIGA™ Trust M2 ID2 setup is described in this section.

3.2.1 ESP32-DevKitC V4

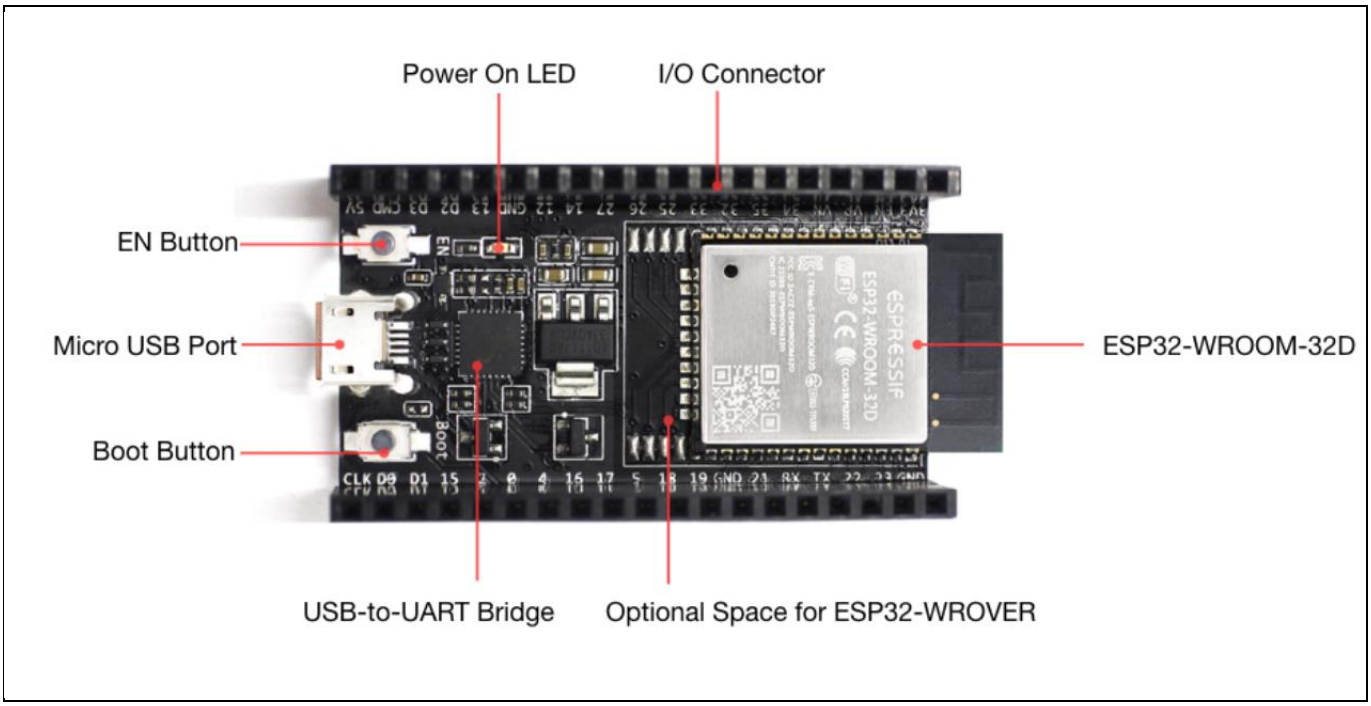

Figure 3 ESP32-DevKitC V4

Connector supports I2C, reset pin and power supply interfaces among others.

Table 4 ESP32-DevKitC V4 Pin Information

No.	Description	Pin
1	I2C SCL	IO 22
2	I2C SDA	IO 21
3	RST	IO 25
4	VCC	IO 26
5	GND	GND

For more information about the ESP32 Specification, Architecture and Design/Schematic, refer document [1]

3.2.2 ESP32 DevKitC Adapter for Shield2Go

The ESP32 DevKitC adapter is an evaluation board that allows users to easily combine different Shield2Go boards to ESP compliant ecosystem, for fast evaluation of IoT systems. With its solderless connectors, it allows users to easily stack Shield2Go boards instead of soldering it. The adapter design is derived from ESP32-DevKitC V4 evaluation board.

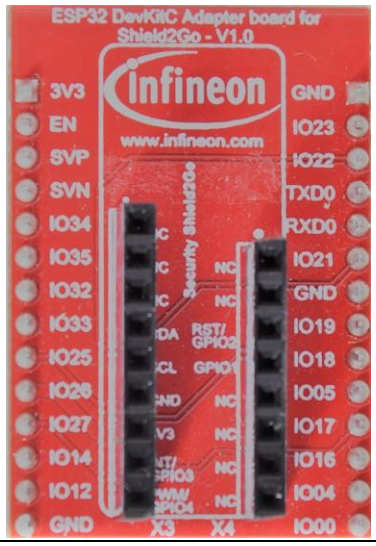

Figure 4 ESP32 DevKitC Adapter for Shield2Go

ESP32 DevKitC adapter features are as follows:

- Provide power supply and connectivity for Shield2Go boards.

More information is available at [Infineon website.]

3.2.3 Shield2Go Security OPTIGA™ Trust M2 ID2

Shield2Go boards are equipped with featured Infineon ICs and provide a standardized form factor and pin layout, allowing a 'plug and play' approach for easy prototyping.

Figure 5 OPTIGA™ Trust M2 ID2 Shield2Go

The OPTIGA™ Trust M2 ID2 Shield2Go is equipped with OPTIGA™ Trust M2 ID2 security chip. It allows users to develop system solutions by combining Shield2Go with ESP32 DevKitC Adapter for Shield2Go and ESP32.

Note: Ensure no voltage supplied to any of the pins exceeds the absolute maximum rating of V_{cc} + 0.3 V.

3.3 Software Setup

This section describes the software used in ESP32 to run the AliOS-Things OPTIGA™ Trust M2 ID2 setup.

3.3.1 Software Components

All the software components required on AliOS-Things for ESP32 are explained in the following sections.

3.3.1.1 ESP32-DevKitC V4

1. OPTIGA™ Trust M2 ID2 Host Library consists of the following:
   - Service Layer
     The layers (Util and Crypt) provide APIs to interact with OPTIGA™ for various use-case functionalities.
   - Access Layer
     This layer manages the access to the command interface of OPTIGA™ security chip. It also provides the communication interface to the OPTIGA™.
   - Platform Abstraction Layer
     This layer provides platform agnostic interfaces for the underlying HW and SW platform functionalities used by OPTIGA™ libraries.
   - Platform Layer
     This layer provides the platform specific components and libraries for the supported platforms.
2. IFX I2C Protocol
   This is an implementation as per document [2].
3. ESP32 I2C Driver
   These are low level I2C device driver for I2C communication from ESP32 to OPTIGA™ Trust M2 ID2 Security chip.

3.3.2 PC Requirements and Configurations

3.3.2.1 PC Requirement

A 32-bit or 64-bit PC with Windows 7/10 Operating System with the below requirements need to be used for setting up ESP32 to run the AliOS-Things using OPTIGA™ Trust M2 ID2 setup:

1. One USB port.
2. Python 2.7.14 version to install AliOS-Thing dependency packages
   Link to download Python 2.7.14: Download link
3. Git for downloading source code.
   Link to download git: Download link
4. FTD driver to access ESP32 via COM port.
   Link to download FTD driver: Download link

Note: Add C:\Python27 and C:\Python27\Scripts path to environment variable in the beginning of the environment variable list.

4 Shell Application execution using OPTIGA™ Trust M2 ID2

4.1 Quick Setup

1. Navigate to *<INSTALLDIR>/projects/esp32_devkitc_alios* and execute the *alios-things-setup.bat* script.

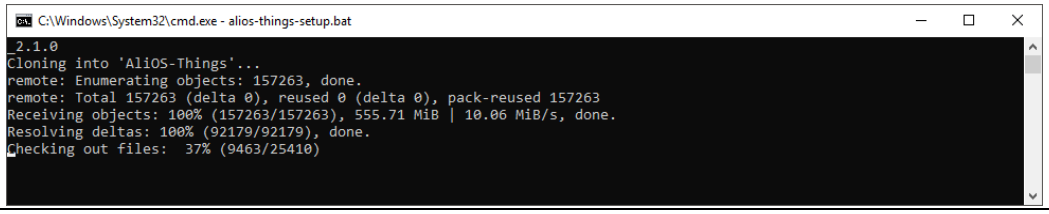

Figure 6 Download AliOS-Things source package ongoing

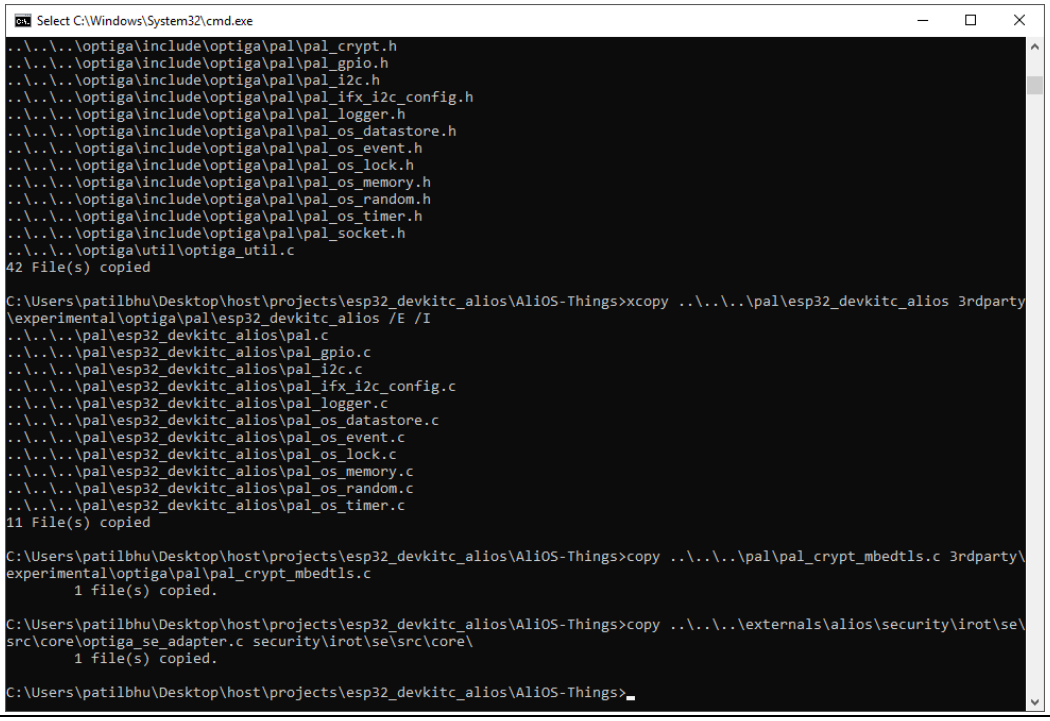

Figure 7 Download AliOS-Things source package complete

2. A folder with name **AliOS-Things** will appear under *<INSTALLDIR>/projects/esp32_devkitc_alios.*

   Note: Ignore warning: 1 line adds whitespace errors.

3. Upgrade aos-cube by following the below steps in command line terminal at *<INSTALLDIR>/projects/esp32_devkitc_alios/AliOS-Things*
   - pip install --upgrade setuptools
   - pip install --upgrade wheel
   - pip install --upgrade aos-cube

```
C:\Windows\System32\cmd.exe
C:\Users\patilbhu\Desktop\host\projects\esp32_devkitc_alios\AliOS-Things>pip install --upgrade setuptools
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 wo
n't be maintained after that date. A future version of pip will drop support for Python 2.7.
Collecting setuptools
  Downloading https://files.pythonhosted.org/packages/e1/b7/182161210a13158cd3ccc41ee19aadef54496b74f2817cc147006ec932b4
/setuptools-44.1.1-py2.py3-none-any.whl (583kB)
     |UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU| 583kB 2.4MB/s
Installing collected packages: setuptools
  Found existing installation: setuptools 41.0.1
    Uninstalling setuptools-41.0.1:
      Successfully uninstalled setuptools-41.0.1
Successfully installed setuptools-44.1.1
WARNING: You are using pip version 19.1.1, however version 20.2.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\patilbhu\Desktop\host\projects\esp32_devkitc_alios\AliOS-Things>pip install --upgrade wheel
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 wo
n't be maintained after that date. A future version of pip will drop support for Python 2.7.
Collecting wheel
  Using cached https://files.pythonhosted.org/packages/a7/00/3df031b3ecd5444d572141321537080b40c1c25e1caa3d86cdd12e5e919
c/wheel-0.35.1-py2.py3-none-any.whl
Installing collected packages: wheel
  Found existing installation: wheel 0.33.4
    Uninstalling wheel-0.33.4:
      Successfully uninstalled wheel-0.33.4
Successfully installed wheel-0.35.1
WARNING: You are using pip version 19.1.1, however version 20.2.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\patilbhu\Desktop\host\projects\esp32_devkitc_alios\AliOS-Things>pip install --upgrade aos-cube
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 wo
n't be maintained after that date. A future version of pip will drop support for Python 2.7.
Collecting aos-cube
  Using cached https://files.pythonhosted.org/packages/76/a5/f4e3c52a205c9c5345de7cadf5e1712a330bc83b5f1110f9c9537be1c62
1/aos-cube-0.5.11.tar.gz
Requirement already satisfied, skipping upgrade: pyserial in c:\python27\lib\site-packages\pyserial-3.4-py2.7.egg (from
```

Figure 8 aos-cube upgrade

4.1.1 Configure and build for ESP32-DevKitC V4

This section describes how to configure and build optiga_shell_app in AliOS-Things source code for ESP32.

4.1.1.1 Configuration

1. Execute below command to configure the optiga_shell_app example
   - aos make optiga_shell_app@esp32devkitc -c config
     Note: git clone config in case of error, shown in Figure 9
   - Execute below command in the terminal
   - git clone https://gitee.com/alios-things/kconfig-frontends-win32.git ./build/kconfig/Win32/

```
C:\Windows\System32\cmd.exe

C:\Users\patilbhu\Desktop\host\projects\esp32_devkitc_alios\AliOS-Things>aos make optiga_shell_app@esp32devkitc -c confi
g
aos-cube version: 0.5.11
INFO: Could not find files for the given pattern(s).
INFO: Could not find files for the given pattern(s).

Can't reach url: https://gitee.com/alios-things/kconfig-frontends-win32.git
Please check your network and download it manually:

  $ git clone https://gitee.com/alios-things/kconfig-frontends-win32.git ./build/kconfig/Win32/

make: *** [build/build_rules/aos_kconfig.mk:94: .\build\kconfig\Win32\kconfig-conf.bat] Error 1

C:\Users\patilbhu\Desktop\host\projects\esp32_devkitc_alios\AliOS-Things>git clone https://gitee.com/alios-things/kconfi
g-frontends-win32.git ./build/kconfig/Win32/
Cloning into './build/kconfig/Win32'...
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 19 (delta 7), reused 4 (delta 2), pack-reused 0
Unpacking objects: 100% (19/19), done.

C:\Users\patilbhu\Desktop\host\projects\esp32_devkitc_alios\AliOS-Things>_
```

Figure 9 Error while configuring AliOS-Things

- Rerun aos make optiga_shell_app@esp32devkitc -c config

```
C:\Windows\System32\cmd.exe
C:\Users\patilbhu\Desktop\host\projects\esp32_devkitc_alios\AliOS-Things>aos make optiga_shell_app@esp32devkitc -c confi
g
aos-cube version: 0.5.11
Creating .defconfig ...
Creating ./.config ...
#
# configuration written to ./.config
#
Checking config output: ./.config ...

C:\Users\patilbhu\Desktop\host\projects\esp32_devkitc_alios\AliOS-Things>
```

Figure 10 AliOS-Things config complete

4.1.1.2 Build source code

1. To build source code execute below command
   - aos make

```
C:\Windows\System32\cmd.exe
Making out/optiga_shell_app@esp32devkitc/libraries/vfs.a
Making out/optiga_shell_app@esp32devkitc/libraries/optiga.a
Making optiga_shell_app@esp32devkitc.elf

Making optiga_shell_app@esp32devkitc.bin
Making optiga_shell_app@esp32devkitc.hex

                        AOS MEMORY MAP
|===============================================================|
| MODULE                                  | ROM       | RAM       |
|===============================================================|
| arch_xtensa_lx6                         | 4664      | 2088      |
| board_esp32devkitc                      | 311       | 0         |
| debug                                   | 5315      | 16        |
| kernel_init                             | 114       | 0         |
| kv                                      | 3399      | 24        |
| libbootloader_support                   | 1698      | 0         |
| libc                                    | 59263     | 0         |
| libcoexist                              | 3716      | 1471      |
| libcontainer                            | 2634      | 0         |
| libcore                                 | 1213      | 5         |
| libdriver                               | 13145     | 141       |
| libesp32                                | 31523     | 4420      |
| libgcc                                  | 5211      | 0         |
| libhal                                  | 547       | 0         |
| liblog                                  | 884       | 276       |
| libm                                    | 92        | 0         |
| libmesh                                 | 55        | 2         |
| libnet80211                             | 125138    | 9366      |
| libnewlib                               | 3137      | 424       |
| libnvs_flash                            | 11806     | 32        |
| libphy                                  | 34793     | 2194      |
| libpp                                   | 62940     | 6510      |
| librtc                                  | 879       | 4         |
| libsoc                                  | 7026      | 8         |
| libspi_flash                            | 6138      | 349       |
| libstdcc\+\+-cache-workaround           | 96        | 4         |
| libvfs                                  | 3602      | 335       |
| libwpa                                  | 23277     | 682       |
| libwpa2                                 | 0         | 1         |
| libwpa_supplicant                       | 15202     | 0         |
| libwps                                  | 0         | 1         |
| lwip                                    | 52251     | 1164      |
| mcu_esp32                               | 20577     | 574       |
| netmgr                                  | 285       | 8         |
| optiga                                  | 108968    | 12450     |
| optiga_shell_app                        | 36        | 0         |
| osal_aos                                | 1376      | 0         |
| rhino                                   | 24452     | 9888      |
| ulog                                    | 325       | 14        |
| vfs                                     | 1514      | 1105      |
| yloop                                   | 2118      | 32        |
| *fill*                                  | 4737      | 244       |
|===============================================================|
| TOTAL (bytes)                           | 644457    | 53832     |
|===============================================================|
esptool.py v2.6
653888
e98c8260c5d60f988d95e46286d390d4
Build complete: optiga_shell_app@esp32devkitc

C:\Users\patilbhu\Desktop\host\projects\esp32_devkitc_alios\AliOS-Things>
```

Figure 11 Build source code

4.1.2 Download example hex file to ESP32-DevKitC V4

1. Execute below command to flash the generated HEX file
   - aos upload optiga_shell_app@esp32devkitc
     Note: Note the COM port asserted after connecting the ESP32-DevKitC board.

```
--- Available ports:
---  1: COM3                u'Intel(R) Active Management Technology - SOL (COM3)'
---  2: COM6                u'Silicon Labs CP210x USB to UART Bridge (COM6)'
---  3: COM25               u'Standard Serial over Bluetooth link (COM25)'
---  4: COM26               u'Standard Serial over Bluetooth link (COM26)'
--- Enter port index or full name 2
```

Figure 12 Selecting COM port

If the COM port asserted by ESP32-DevKitC board is open it does not allow to flash. A typical error message is shown in Figure 13. Close the COM port and flash the hex.

```
C:\Windows\System32\cmd.exe
    main()
  File "c:\python27\lib\site-packages\esptool.py", line 2652, in main
    esp = chip_class(each_port, initial_baud, args.trace)
  File "c:\python27\lib\site-packages\esptool.py", line 222, in __init__
    self._port = serial.serial_for_url(port)
  File "build\bdist.win32\egg\serial\__init__.py", line 90, in serial_for_url
  File "build\bdist.win32\egg\serial\serialwin32.py", line 64, in open
serial.serialutil.SerialException: could not open port 'COM30': WindowsError(5, 'Access is denie
d.')
---host_os:Win32
[ERROR]: Firmware upload failed!


C:\Users\patilbhu\Desktop\host\projects\esp32_devkitc_alios\AliOS-Things>
```

Figure 13 Hex file flash issue due to COM port open

4.1.3 Steps to execute optiga_shell_app example

1. Configure COM port with 115200 8N1.

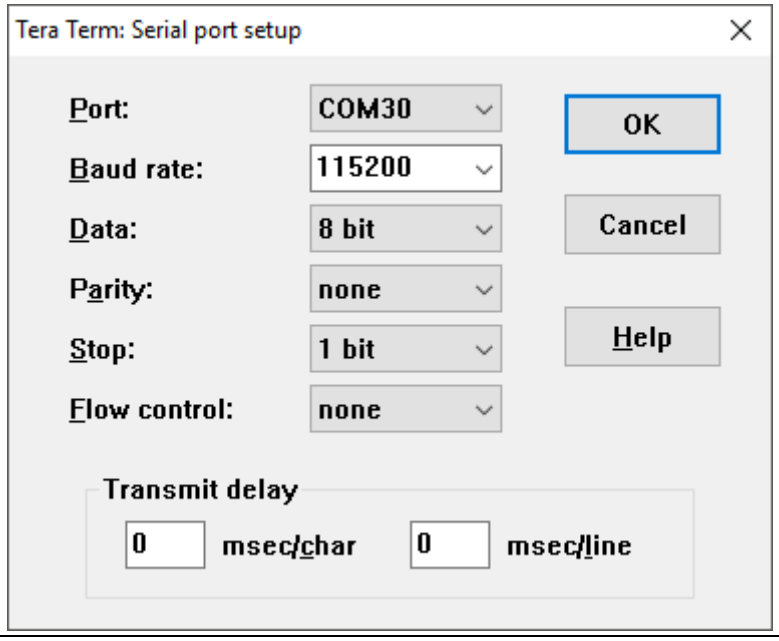

Figure 14 TeraTerm terminal serial configuration

2. After connecting the terminal it will start with boot sequence.

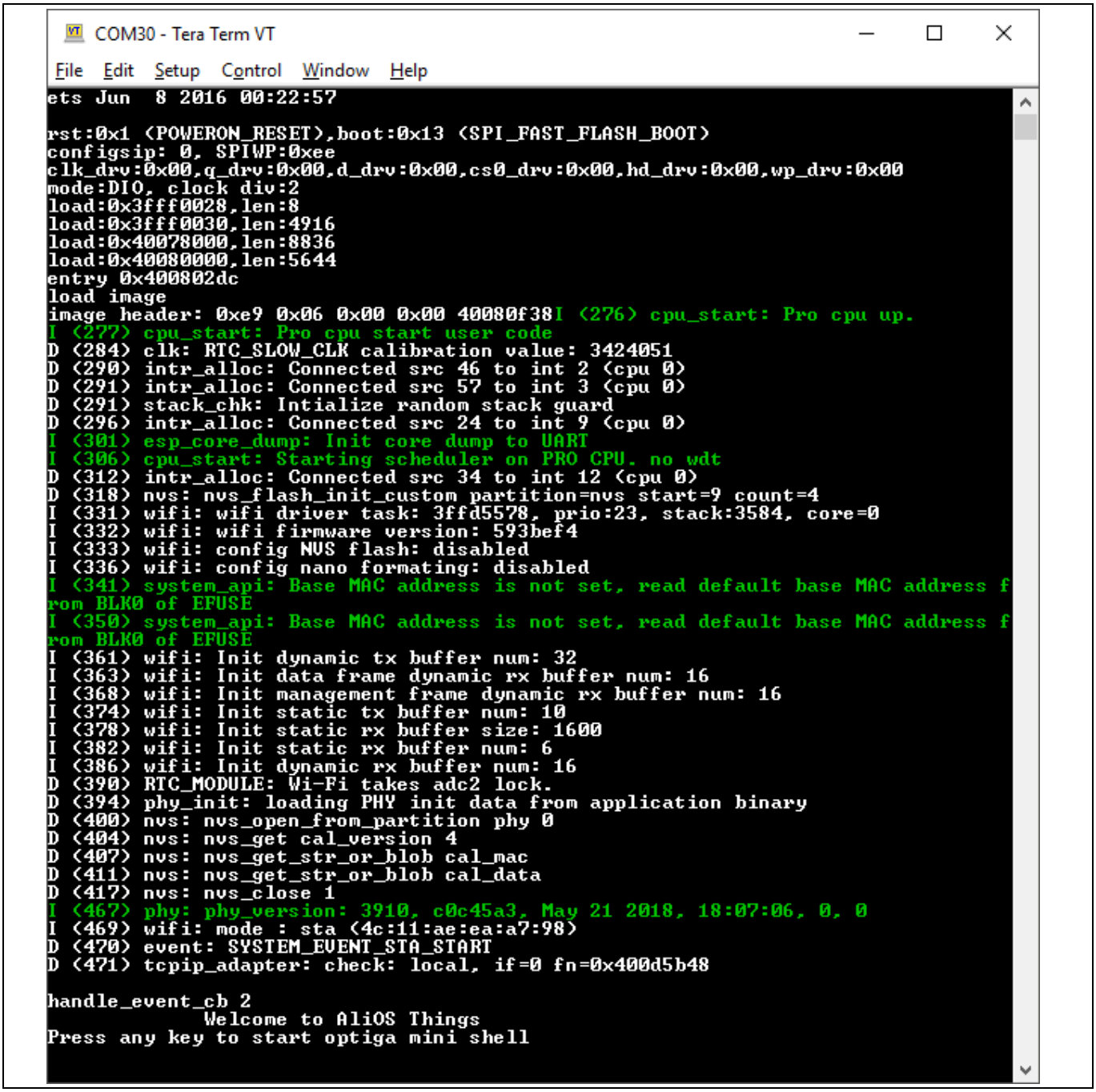

Figure 15 AliOS-Things initial boot sequence

Note: In some cases the boot sequence is not shown, in that case press enter on terminal.

1. Press any key to start the optiga mini shell.

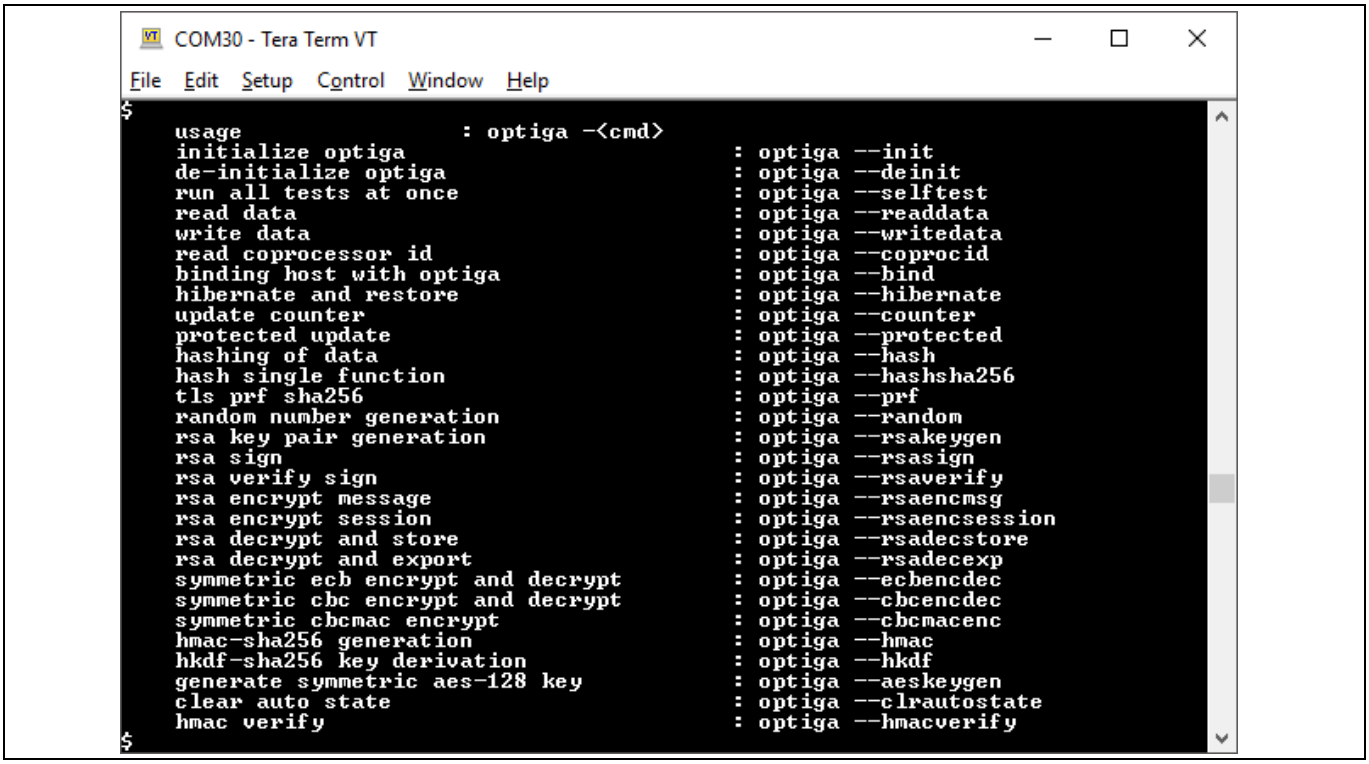

Figure 16 Optiga available commands for shell application

2. Enter optiga command in format of “optiga --<cmd>” for a self-contained example.

Note: By default self-contained concept of examples is enabled, where each example performs init & deinit of OPTIGA.

```
COM30 - Tera Term VT
File  Edit  Setup  Control  Window  Help
    usage                    : optiga -<cmd>
    initialize optiga                          : optiga --init
    de-initialize optiga                       : optiga --deinit
    run all tests at once                      : optiga --selftest
    read data                                  : optiga --readdata
    write data                                 : optiga --writedata
    read coprocessor id                        : optiga --coprocid
    binding host with optiga                   : optiga --bind
    hibernate and restore                      : optiga --hibernate
    update counter                             : optiga --counter
    protected update                           : optiga --protected
    hashing of data                            : optiga --hash
    hash single function                       : optiga --hashsha256
    tls prf sha256                             : optiga --prf
    random number generation                   : optiga --random
    rsa key pair generation                    : optiga --rsakeygen
    rsa sign                                   : optiga --rsasign
    rsa verify sign                            : optiga --rsaverify
    rsa encrypt message                        : optiga --rsaencmsg
    rsa encrypt session                        : optiga --rsaencsession
    rsa decrypt and store                      : optiga --rsadecstore
    rsa decrypt and export                     : optiga --rsadecexp
    symmetric ecb encrypt and decrypt          : optiga --ecbencdec
    symmetric cbc encrypt and decrypt          : optiga --cbcencdec
    symmetric cbcmac encrypt                   : optiga --cbcmacenc
    hmac-sha256 generation                     : optiga --hmac
    hkdf-sha256 key derivation                 : optiga --hkdf
    generate symmetric aes-128 key             : optiga --aeskeygen
    clear auto state                           : optiga --clrautostate
    hmac verify                                : optiga --hmacverify
$optiga --random
[optiga shell]  : Initializing OPTIGA for example demonstration...
[optiga shell]  : Starting Generate Random Example
[optiga shell]  : 1 Step: Generate 32 bytes random
[optiga shell]  : 2 Step: Close the application on OPTIGA
[optiga example]  : example_optiga_init
I (475602) gpio: GPIO[25]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pul
ldown: 0| Intr:0
I (475607) gpio: GPIO[26]| InputEn: 0| OutputEn: 1| OpenDrain: 0| Pullup: 0| Pul
ldown: 0| Intr:0
[475312]<V> 61A80
[optiga example]  : pair_host_and_optiga_using_pre_shared_secret
[optiga example]  : Passed
[optiga example]  : Passed
[optiga example]  : example_optiga_crypt_random
[optiga example]  : Passed
[optiga example]  : example_optiga_deinit
[optiga example]  : Passed
[optiga example]  : Example takes 38 msec
$
```

Figure 17 Self-contained optiga random command execution using shell application

3. To execute the example without shielded connection, disable the macro OPTIGA_COMMS_SHIELDED_CONNECTION in file optiga_lib_config.h at location *<INSTALLDIR>/projects/esp32_devkitc_alios/AliOS-Thing/3rdparty/experimental/optiga/optiga/include/optiga*

4.1.4 Logger control for shell application

By default only logging from example is enabled in the release package.

Further control for OPTIGA™ Trust M2 ID2 host code logging is available in optiga_lib_config.h.

The macro OPTIGA_LIB_ENABLE_LOGGING provides complete control to enable/disable logging at host code. In addition, logging at UTIL, CRYPT, CMD and COMMS layer can be controlled using the following macros,

- OPTIGA_LIB_ENABLE_UTIL_LOGGING

- OPTIGA_LIB_ENABLE_CRYPT_LOGGING
- OPTIGA_LIB_ENABLE_CMD_LOGGING
- OPTIGA_LIB_ENABLE_COMMS_LOGGING

For Example,

1. To enable logging for only COMMS layer, enable OPTIGA_LIB_ENABLE_COMMS_LOGGING and disable rest all layer macros.
2. Build the project as described in section 4.1.1.2 and execute the example as defined in 4.1.3.

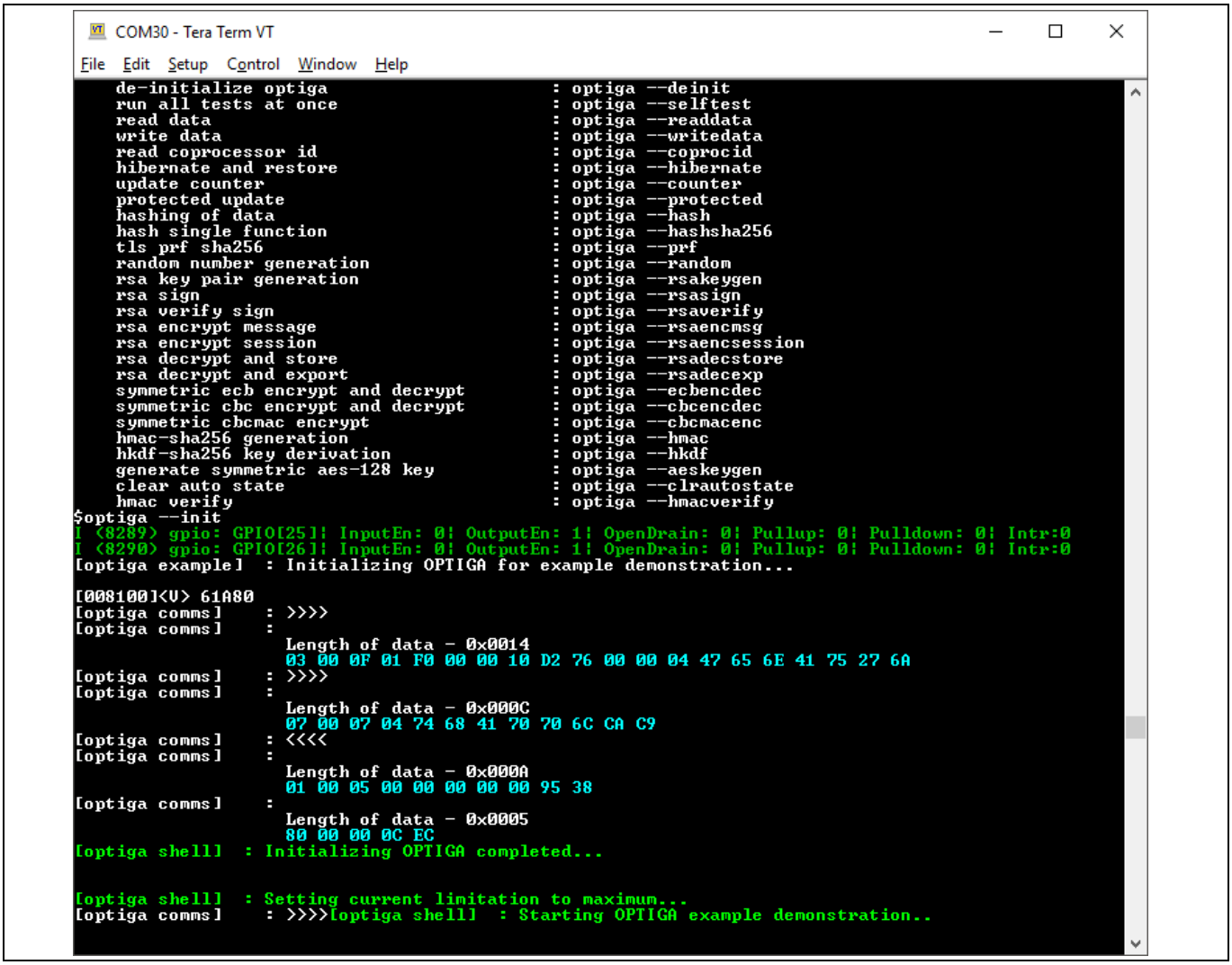

Figure 18 Logging data with only COMMS layer enabled

5 Troubleshooting

Table 5 Troubleshooting

No	Problem	Reason	Solution

Revision History

Table 6

Document version	Date of release	Description of changes
2.00	2020-09-09	Initial version
2.10	2020-09-21	Release to production release

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

Edition 2020-09-21

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email:
CSSCustomerService@infineon.com

Document reference