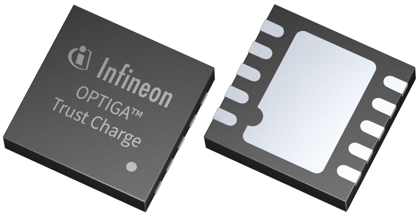
SLS 32AIA020U2/3

OPTIGA Trust Charge V1

Datasheet

Features



- High-end security controller
- Common Criteria Certified EAL6+ (high) hardware
- Turnkey solution
- Compliant with the USB Type-C™ Authentication standard
- Up to 10 KB user memory
- PG-USON-10-2,-4 package (3 mm x 3 mm)
- Standard and extended temperature ranges
- I2C interface with shielded connection (encrypted communication)
- Cryptographic support: ECC NIST P256/P384, SHA-256, TRNG, DRNG
- Crypto ToolBox commands for SHA-256, ECC NIST P256/P384 (sign, verify, and key generation)
- Device security monitor
- 4 Monotonic up counters
- Protected (integrity) update of data objects
- Hibernate for zero power consumption[1]
- Lifetime for industrial automation and Infrastructure is 20 years and 15 years for other application profiles
- Wireless Power Consortium (WPC) Qi authentication

Benefits

- Protection of consumers against fake charging devices
- Turnkey solution with full system integration support including embedded software, host software, development board, reference board, and documentation
- WPC-specific personalized keys and certificates preloaded at secured Infineon fabs
- Tiny package (3 mm x 3 mm) optimized for small devices
- Versions for standard and extended temperature available

Potential applications

Qi charging for mobile phones, tablets, laptops, and other accessories.

[1] Leakage current < 2.5 µA only

Please read the sections "Important notice" and "Warnings" at the end of this document

About this document

Scope and purpose

This document provides information to enable integration of a security device, includes package, connectivity, and technical data.

Intended audience

This document is intended for device integrators and board manufacturers.

Table of contents

List of tables

List of figures

1 Introduction

As embedded systems (for example, *Internet of Things (IoT)* devices) are increasingly gaining the attention of attackers, Infineon offers the OPTIGA™ Trust Charge as an authentication solution for wireless power charging applications. This high-end security controller comes with full system integration support for easy and cost-effective deployment of high-end security for your assets.

1.1 Broad range of benefits

Integrated into your device, the OPTIGA™ Trust Charge supports protection of your brand and business case, differentiates your product from your competitors, and adds value to your product, making it stronger against cyber attacks.

1.2 Enhanced security

The OPTIGA™ Trust Charge is based on an advanced security controller with built-in tamper proof *non-volatile memory (NVM)* for secure storage and symmetric/asymmetric crypto engines to support *elliptic curve cryptography (ECC)* 256/384 and *Secure Hash Algorithm (SHA)*-256. This new security technology greatly enhances your overall system security.

1.3 Fast and easy integration

The turnkey setup - with full system integration support, all keys, and certificate materials are pre-programmed to minimizes customer effort for design, integration, and deployment. The OPTIGA™ Trust Charge comes with pre-programmed locked *operating system (OS)*, locked application code, and host-side modules to integrate with host microcontroller software. Integration support includes a reference board and documentation for rapid design-in. The temperature range of -40°C to +105°C combined with a standardized *inter-integrated circuit (I2C)* interface and the small PG-USON-10-2,-4 (*ultrathin small outline no-lead (USON)*) footprints will facilitate on-boarding in your existing ecosystem.

1.4 Applications

Infineon's OPTIGA™ Trust Charge is a turnkey solution providing secured device authentication for inductive wireless charging according to the Qi2 wireless charging standard. Secured authentication with OPTIGA™ Trust Charge contributes to device and user safety by protecting against fake chargers.

1.5 Device features

The OPTIGA™ Trust Charge comes with up to 10 KB user memories that can be used to store *Wireless Power Consortium (WPC)* certificates and data. The OPTIGA™ Trust Charge is based on *Common Criteria for Information Technology Security Evaluation (CC)* certified *evaluation assurance level (EAL)*6+ (high) hardware enabling it to prevent physical attacks on the device itself and providing high assurance that the keys or arbitrary data stored cannot be accessed by an unauthorized entity. The CC certificate can be found at www.bsi.bund.de by searching for *Bundesamt für Sicherheit in der Informationstechnik (BSI)*-DSZ-CC-0961 (hardware identifier IFX_CCI_00000Bh) and referring to the latest CC certificate. OPTIGA™ Trust Charge supports a high-speed *I2C* communication interface of up to 1 MHz (FM+).

Table 1 Products

Type	Description	Temperature range	Package
OPTIGA™ Trust Charge SLS 32AIA020U2	Provisioned secure storage subsystem for Qi charging (Qi2 and higher)	-25°C to +85°C standard temperature range (STR)	PG-USON-10-2,-4
OPTIGA™ Trust Charge SLS 32AIA020U3	Provisioned secure storage subsystem for Qi charging (Qi2 and higher)	-40°C to +105°C extended temperature range (ETR)	PG-USON-10-2,-4
Evaluation kit	Provides all the components required to set up the environment to demonstrate the features of the OPTIGA™ Trust Charge	-	Board

Infineon and its distribution partners offer a wide range of customization options (for example, WPC certificate generation and key provisioning) for the security chip.

2 System block diagram

The following figure shows the system block diagram for OPTIGA™ Trust Charge.

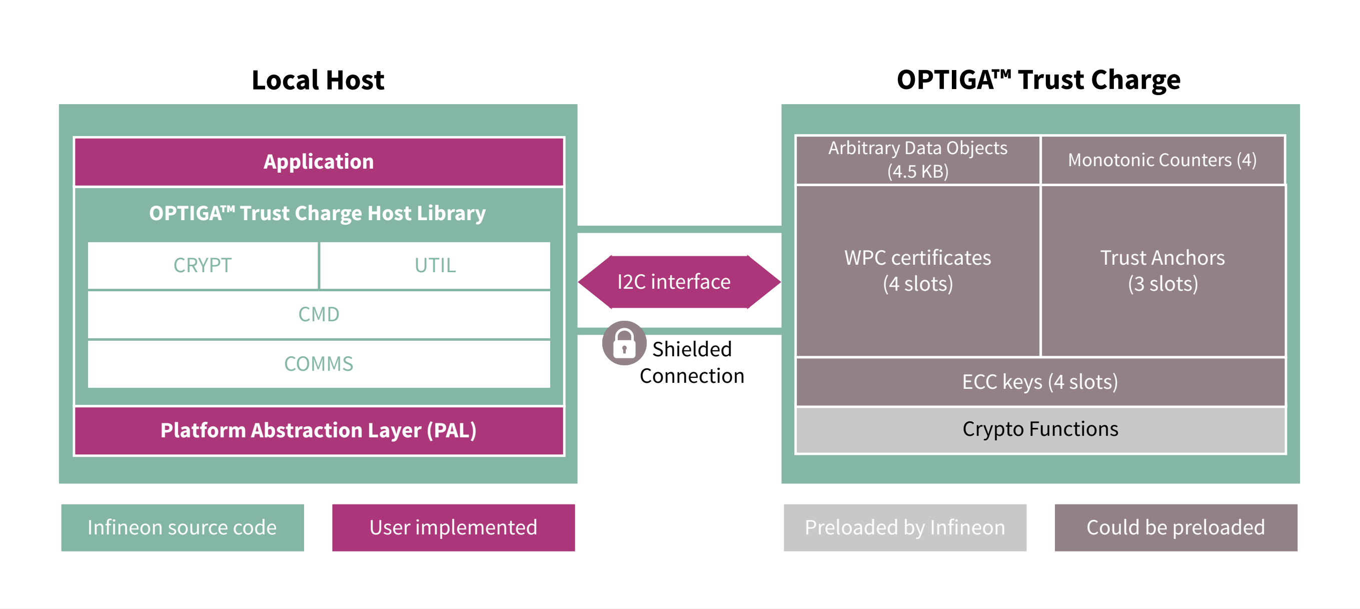


Figure 1 System block diagram

Each layer of the system block diagram is explained below:

1. **Local host:**
   - Local host application: This is the target application which utilizes OPTIGA™ Trust Charge for its security needs
   - OPTIGA™ Trust Charge host library:
     - CRYPT: Provides *application programming interface (API)* to perform cryptographic functionalities
     - UTIL: Provides APIs such as read/write, protected update of data objects and open/close application (for example, hibernate)
     - CMD: Provides APIs to send and receive commands (see Commands) to and from OPTIGA™ Trust Charge
     - COMMS: Provides wrapper APIs for communication (optional encrypted communication using Shielded Connection) with OPTIGA™ Trust Charge which internally uses Infineon *I2C* protocol (refer to *Infineon Technologies AG (IFX)* I2C protocol specification [2])
   - *platform abstraction layer (PAL)*: A layer that abstracts platform specific drivers (for example, I2C, timer, *general purpose input/output (GPIO)*, platform crypto library etc.,)
2. **OPTIGA™ Trust Charge:**
   - Arbitrary data objects: The target application can store up to 4.5 KB (~4600 bytes) of data into OPTIGA™ Trust Charge. The data could be additional trust anchors, certificates and shared secret
   - Monotonic counters: Provides 4 monotonic counting data objects (up counters). These can be used as general purpose counter or as linked counter to other objects

     For more information, refer to Solution Reference Manual document available as part of the package
   - *WPC* certificates: Up to 4 WPC certificate chains can be stored
   - Keys: Up to 4 *ECC* based keys can be stored

- Trust anchors: 3 slots
- Crypto functions: OPTIGA™ Trust Charge provides cryptographic functions that can be invoked via local host

***Note**:* *Unique* ECC *private keys and* WPC *certificates - during production at Infineon fab, unique asymmetric keys (private and public) are generated. The public key is signed by customer specific certificate authority (CA) and the resulting* WPC *certificate issued is securely stored in the OPTIGA™ Trust Charge. Special measures are taken to prevent the leakage and modification of private key material at the Common Criteria Certified production site.*

3 Interface and schematics

This section explains the schematics of the product and gives some recommendations as to how the controller should be externally connected.

3.1 System integration schematics

The following figure shows how to integrate OPTIGA™ Trust Charge with the local host.

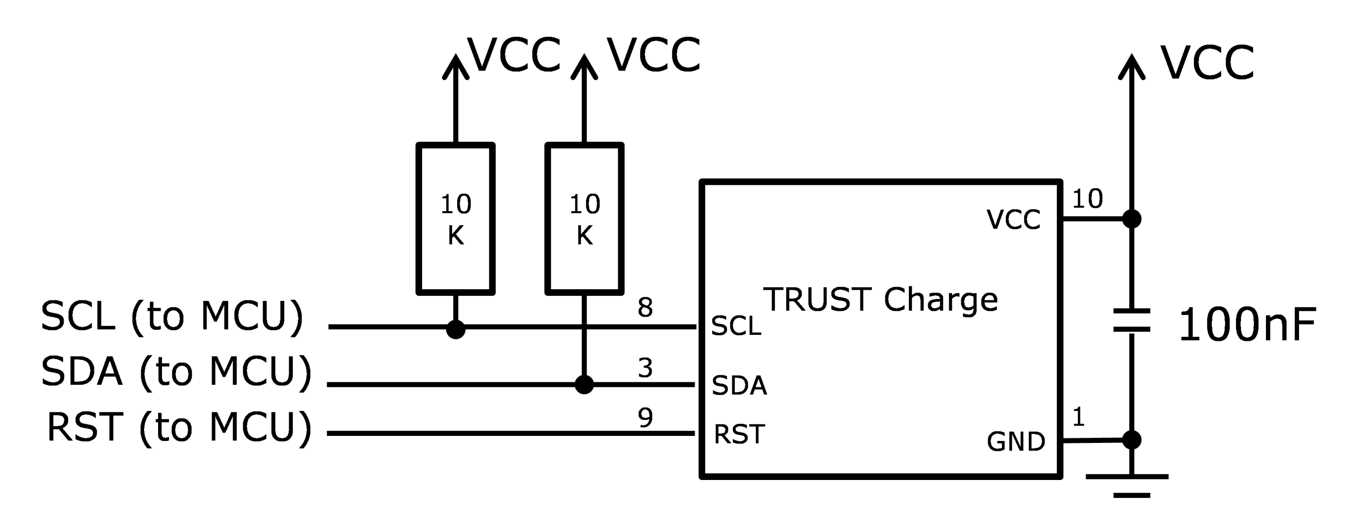


Figure 2 System integration schematic diagram

Note: *Value of the pullup resistors depend on the target application circuit and the targeted I2C frequency. V_{CC} can be driven by host microcontroller unit (MCU) GPIO pin or direct supply voltage (+3.3 V).*

3.2 System integration schematics with hibernation support

The following figure shows how to integrate OPTIGA™ Trust Charge with hibernation, with the local host.

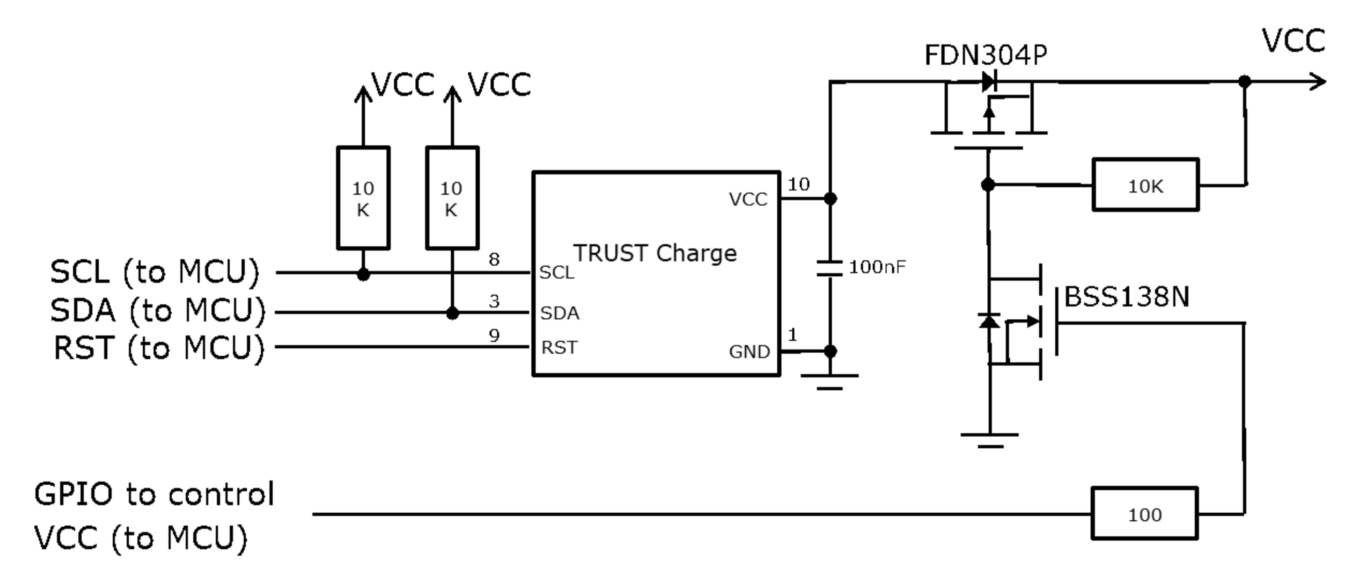


Figure 3 System integration schematic diagram with hibernation

***Note**: Value of the pullup resistors depend on the target application circuit and the targeted I2C frequency.*

4 Package description

This chapter provides information on the package types and how the interfaces of each product are assigned to the package pins. For further information on compliance of the packages with European Parliament Directives, see RoHS compliance.

For details and recommendations regarding the assembly of packages on *printed circuit board (PCB)*, refer to the following link: http://www.infineon.com/cms/en/product/technology/packages/.

4.1 PG-USON-10-2,-4

Available packages:

- PG-USON-10-2
- PG-USON-10-4

The figures in the sections below show the following aspects of the package:

- **Package outline:** It shows the package dimensions of the device in the individual packages
- **Package footprint:** It shows footprint recommendations
- Tape and reel packing
- **Sample marking pattern:** It describes the productive sample marking pattern on the package

Notes:

1. *The drawings are for information only and not drawn to scale. More detailed information about package characteristics and assembly instructions is available on request.*

2. *Unless specified otherwise, all figure dimensions are given in mm.*

Package outline

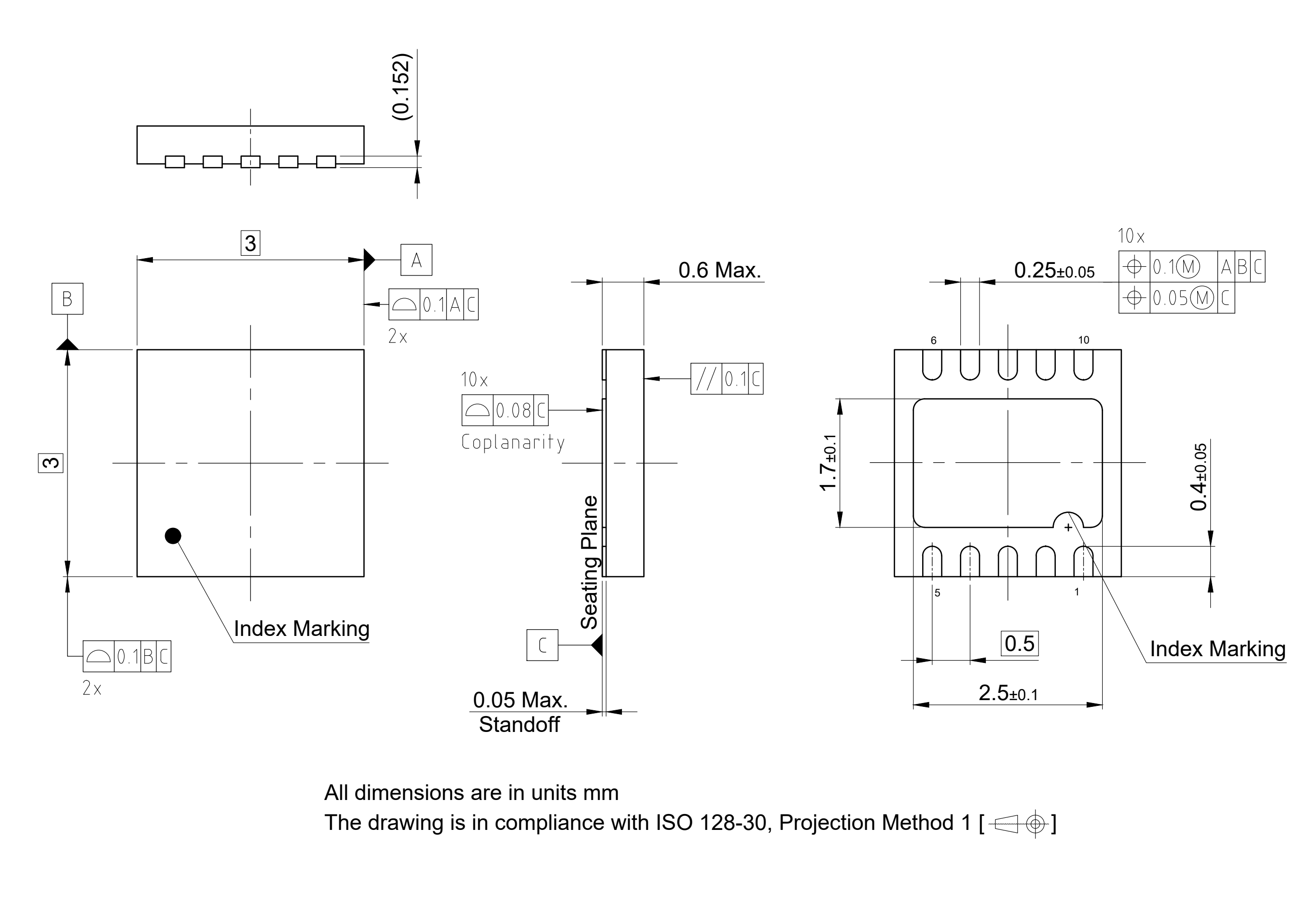


Figure 4 **PG-USON-10-2,-4 package outline**

Package footprint

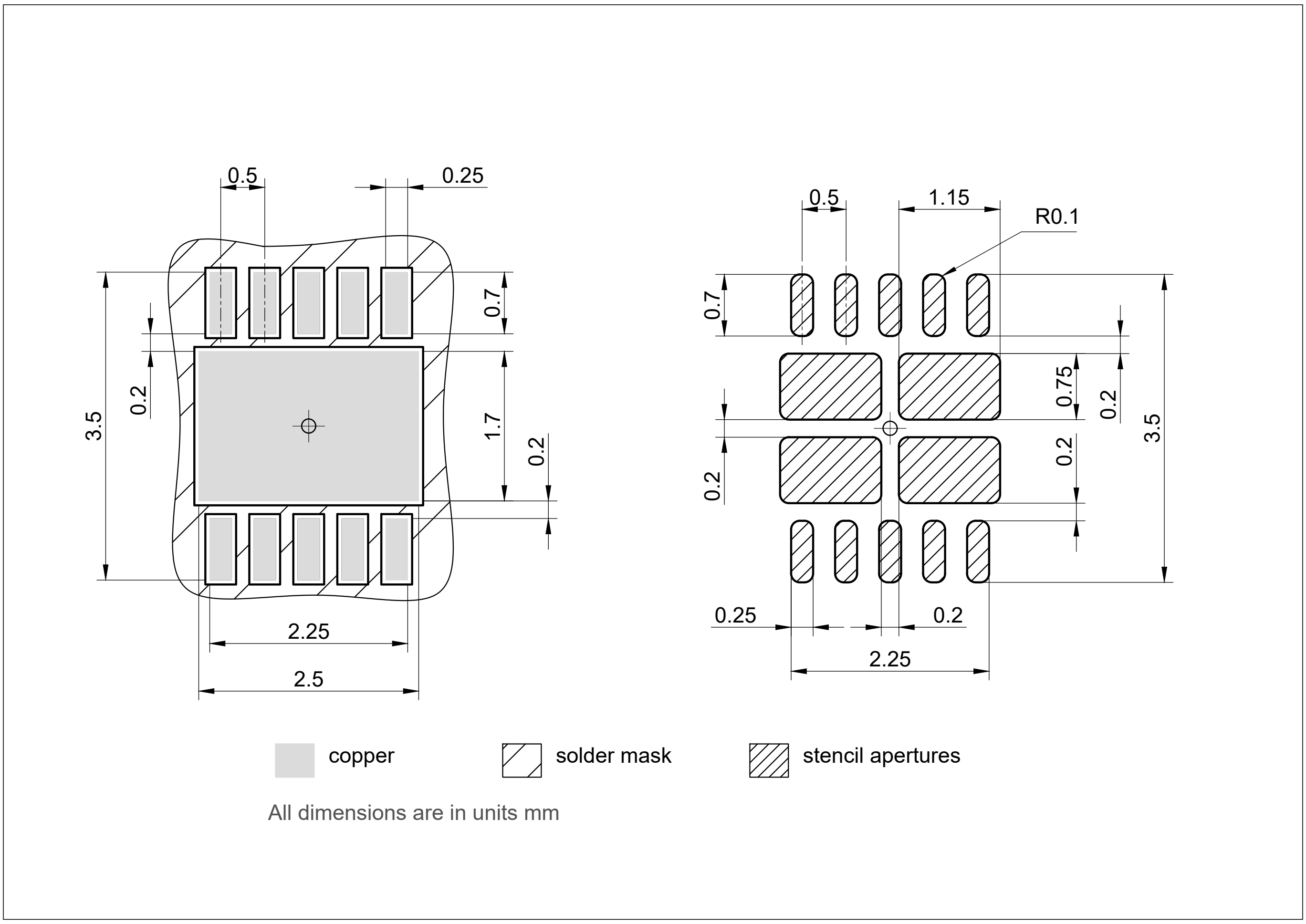


Figure 5 **PG-USON-10-2,-4 package footprint**

Tape and reel packing

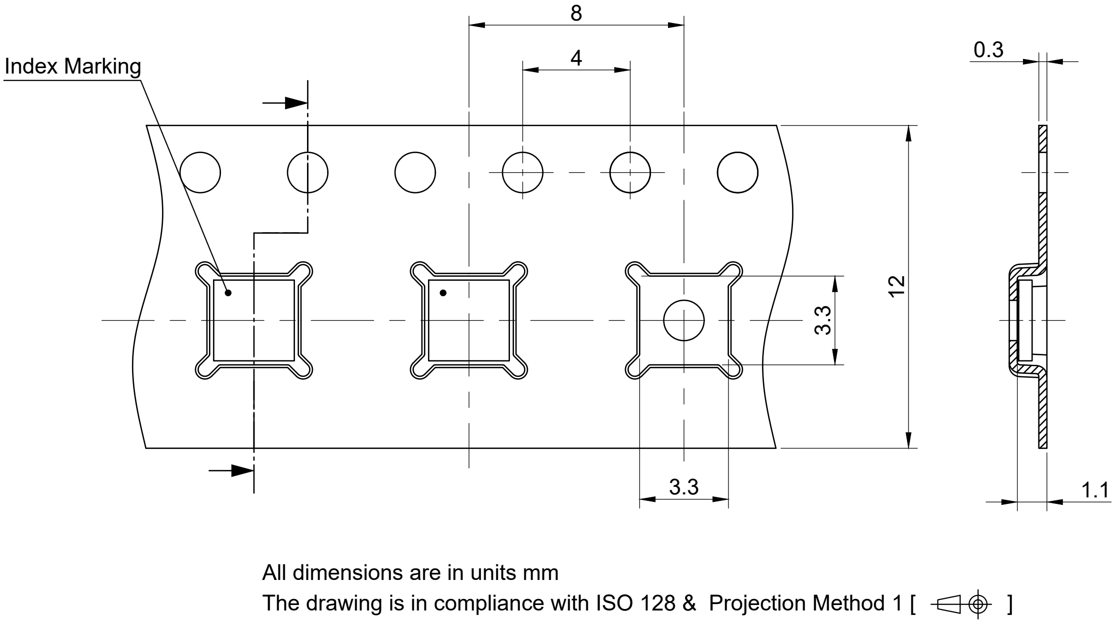

8
4
0.3
Index Marking
12
3.3
3.3
1.1
All dimensions are in units mm
The drawing is in compliance with ISO 128 & Projection Method 1 [ ]

Figure 6 **PG-USON-10-2,-4 tape and reel packing**

Production sample marking pattern

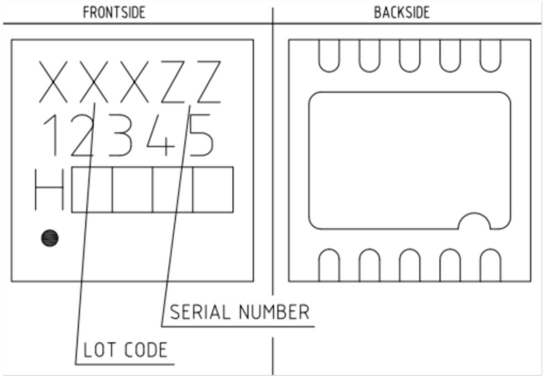


Figure 7 **PG-USON-10-2,-4 sample marking pattern**

The black dot indicates pin 01 for the chip. The following Table 2 describes the sample marking pattern:

Table 2 Marking table for PG-USON-10-2,-4 packages

Indicator	Description
XXX (Lot code)	Defined and inserted during fabrication, issued by the packaging site
ZZ	Indicates the certifying authority serial number/SKU#, for example "00" would mean "SKU#00"
12345	Convention: T&#$@ where: • The letter "T" indicates the OPTIGA™ Trust family • & indicates the product is a Trust Charge controller • # indicates the controller is a STR (S) variant • $ specifies the OPTIGA™ Trust Charge release version number • @ specifies the software version For example: "TCS10" means 'OPTIGA™ Trust Charge', 'STR variant', 'release version 1', 'software version 0'
H/E	H = "Halogen-free", E = "Engineering samples" This indicator is followed by "YYWW", where YY is the "Year" and WW is the "Work Week" of the production. This is inserted during fabrication. Engineering samples have "E YYWW" and productive samples have "H YYWW"

Pin layout

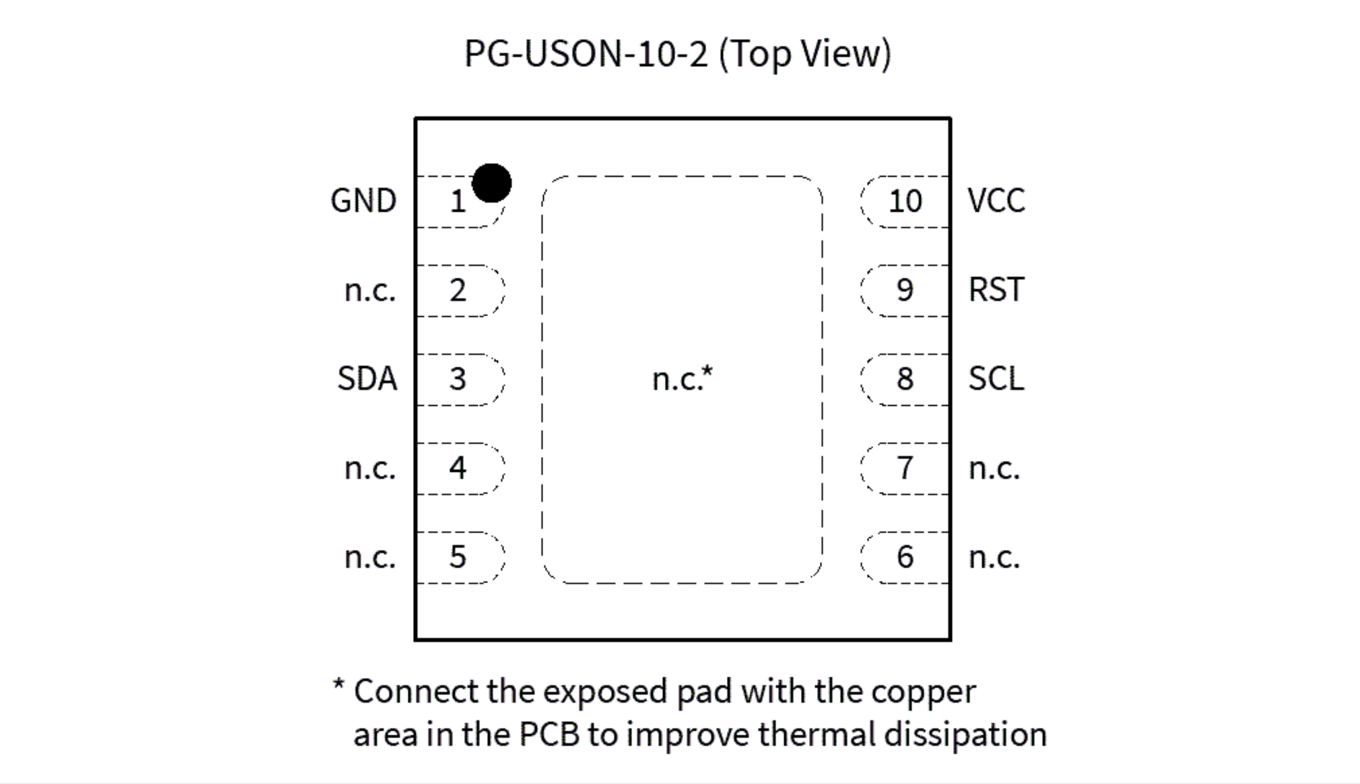

PG-USON-10-2 (Top View)
GND
1
n.c.
2
SDA
3
n.c.
4
n.c.
5
n.c.*
10
VCC
9
RST
8
SCL
7
n.c.
6
n.c.
* Connect the exposed pad with the copper area in the PCB to improve thermal dissipation

Figure 8 PG-USON-10-2,-4 top view

***Note**: It is recommended to connect the exposed die pad to the common ground reference (GND) for heat distribution.*

Pad-to-signal reference

Table 3 Pinout for PG-USON-10-2,-4 packages

Pin	Type	Function
01	GND	Supply voltage (ground)
02	NC	Not connected/do not connect externally
03	I/O	Serial data line (SDA)
04	NC	Not connected/do not connect externally
05	NC	Not connected/do not connect externally
06	NC	Not connected/do not connect externally
07	NC	Not connected/do not connect externally
08	I/O	Serial clock line (SCL)
09	IN	Active low reset (RST)
10	PWR	Supply voltage (V_{CC})

5 Technical data

This section summarizes the technical data of the product. It provides the operational characteristics as well as the electrical *direct current (DC)* and *alternating current (AC)* characteristics.

5.1 I2C interface characteristics

Table 4 I2C operation supply and input voltages

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
Supply voltage	V_{CC_I2C}	1.62	-	5.5	V	
serial data line (SDA), *serial clock line (SCL)* input voltage	V_{IN_I2C}	-0.3	-	V_{CC_I2C} + 0.5 or 5.5[2]	V	V_{CC_I2C} is in the operational supply range
		-0.3	-	5.5	V	V_{CC_I2C} is switched off

5.1.1 I2C standard/fast mode interface characteristics

For operation of the *I2C* interface, the electrical characteristics are compliant with the I2C bus specification Rev. 4 for “standard-mode” (f_{SCL} up to 100 kHz) and “fast-mode” (f_{SCL} up to 400 kHz), with certain deviations as stated in the below table.

***Note**: T_A as given for the operating temperature range of the controller unless otherwise stated.*

Table 5 I2C standard mode interface characteristics

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
SCL clock frequency	f_{SCL}	0	-	100	kHz	-
Input low-level	V_{IL}	-0.3	-	0.3 * V_{CC_I2C}	V	-
Low-level output voltage	V_{OL1}	0	-	0.4	V	Sink current 3 mA; $V_{CC\_I2C} \geq 2.7$ V Sink current 2 mA; $V_{CC\_I2C} < 2.7$ V
Low-level output current	I_{OL}	3	-	-	mA	$V_{OL} = 0.4$ V; $V_{CC\_I2C} \geq 2.7$ V
		2				$V_{OL} = 0.4$ V; $V_{CC\_I2C} < 2.7$ V
Output fall time from V_{IHmin} to V_{ILmax} (at device pin)	t_{OF}	-	-	250	ns	$C_b \leq 400$ pF; $V_{CC\_I2C} \geq 2.7$ V $C_b \leq 200$ pF; $V_{CC\_I2C} < 2.7$ V
Capacitive load for each bus line	C_b	-	-	400	pF	$V_{CC\_I2C} \geq 2.7$ V
				200		$V_{CC\_I2C} < 2.7$ V

[2] Whichever is lower

Table 6 I2C fast mode interface characteristics

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
SCL clock frequency	f_{SCL}	0	-	400	kHz	-
Input low-level	V_{IL}	-0.3		0.3 * V_{CC_I2C}	V	-
Low-level output voltage	V_{OL1}	0	-	0.4	V	Sink current 3 mA; $V_{CC\_I2C} \geq 2.7$ V Sink current 2 mA; $V_{CC\_I2C} < 2.7$ V
Low-level output current	I_{OL}	3	-	-	mA	$V_{OL} = 0.4$ V; $V_{CC\_I2C} \geq 2.7$ V
		2				$V_{OL} = 0.4$ V; $V_{CC\_I2C} < 2.7$ V
Output fall time from V_{IHmin} to V_{ILmax} (at device pin)	t_{OF}	20 * V_{CC_I2C}/5.5 V[3]	-	250	ns	$C_b \leq 400$ pF; $V_{CC\_I2C} \geq 2.7$ V $C_b \leq 200$ pF; $V_{CC\_I2C} < 2.7$ V
Capacitive load for each bus line	C_b	15[4]	-	400	pF	$V_{CC\_I2C} \geq 2.7$ V
				200		$V_{CC\_I2C} < 2.7$ V

5.1.2 I2C fast Mode plus interface characteristics

For operation of the *I2C* interface, the electrical characteristics are compliant with the I2C bus specification Rev. 4 for “fast mode plus” (f_{SCL} up to 1 MHz), with certain deviations as stated in the below table.

***Note**: T_A as given for the operating temperature range of the controller unless otherwise stated.*

Table 7 I2C fast mode plus interface characteristics

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
SCL clock frequency	f_{SCL}	0	-	1000	kHz	-
Input low-level	V_{IL}	-0.3	-	0.3 * V_{CC_I2C}	V	-
Low-level output voltage	V_{OL1}	0	-	0.4	V	Sink current 3 mA; $V_{CC\_I2C} \geq 2.7$ V Sink current 2 mA; $V_{CC\_I2C} < 2.7$ V
Low-level output current	I_{OL}	3	-	-	mA	$V_{OL} = 0.4$ V; $V_{CC\_I2C} \geq 2.7$ V
		2				$V_{OL} = 0.4$ V; $V_{CC\_I2C} < 2.7$ V
Output fall time from V_{IHmin} to V_{ILmax} (at device pin)	t_{OF}	20 * V_{CC_I2C}/5.5 V[5]	-	120	ns	$C_b \leq 150$ pF
Capacitive load for each bus line	C_b	15[6]	-	150	pF	-

[3] A minimum capacitive load is necessary to reach t_{OF}

[4] A minimum capacitive load is necessary to reach t_{fmin}

[5] A minimum capacitive load is necessary to reach t_{OF}

[6] A minimum capacitive load is necessary to reach t_{fmin}

5.1.3 Electrical characteristics

This section summarizes certain electrical characteristics of the controller. It provides operational characteristics as well as electrical DC and AC characteristics and particular interface characteristics.

5.1.3.1 DC electrical characteristics

Note: *T_A as given for the controller's operating ambient temperature range unless otherwise stated. All currents flowing into the controller are considered positive.*

Table 8 DC electrical characteristics

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
Supply voltage	V_{CC}	1.62	-	5.5	V	Overall functional range
	V_{CC_I2C}	1.62	-	5.5	V	Supply voltage range for operation of I2C
Supply current[7)]	I_{CCAVG}	-	14.0	-	mA	While running a typical authentication profile T_A = 25°C; V_{CC} = 5.0 V
Supply current, in sleep mode	I_{CCS3}	-	70	100	µA	T_A = 25°C; V_{CC_I2C} = 3.3 V; I2C ready for operation (no bus activity), all other inputs at V_{CC}, no other interface activity
reset (RST) input low voltage	V_{IL}	-0.3	-	0.3 * V_{CC}	V	I_{IL} = -50 µA to +20 µA
RST input high voltage	V_{IH}	0.7 * V_{CC}	-	V_{CC} + 0.3	V	I_{IL} = -50 µA to +20 µA
Hibernate current	-	-	< 2.5	-	µA	V_{cc} = 0 V, GND = 0 V, RST = 0 V, SCL= 3.3 V and SCL = 3.3 V

5.1.3.2 AC electrical characteristics

Note: *T_A as given for the controller's operating ambient temperature range unless otherwise stated. All currents flowing into the controller are considered positive.*

Table 9 AC electrical characteristics

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
V_{CC} ram-pup time	t_{VCCR}	1	-	1000	µs	400 mV to 90% of V_{CC} target voltage ramp

The V_{CC} ramp is shown in Figure 9. 90% of the target supply voltage must be reached within t_{VCCR} after it has exceeded 400 mV. Moreover, its variation must be kept within a ±10% range.

[7] Supply current can be limited from 6 mA to 15 mA by software commands

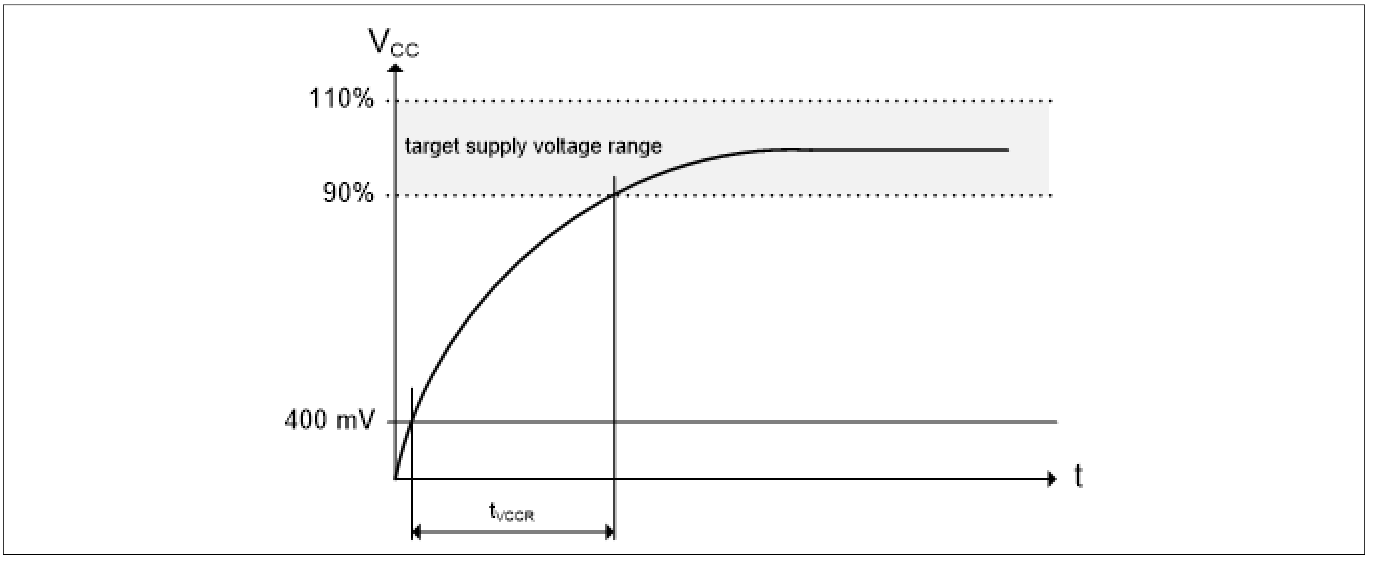


Figure 9 V_{cc} ram-pup

5.1.4 Start-up of I2C interface

There are 2 variants possible for performing the start-up procedure:

- Start-up after power-on
- Start-up for warm resets

5.1.4.1 Start-up after power-on

The activation of the *I2C* interface after power-on needs the following reset procedure:

- VCC is powered up and the state of the *SDA* and *SCL* line are set to high level during power-up
- The first transmission may start at the earliest $t_{STARTUP}$ after power-up of the device

The following figure shows the start-up timing of the I2C interface for this case.

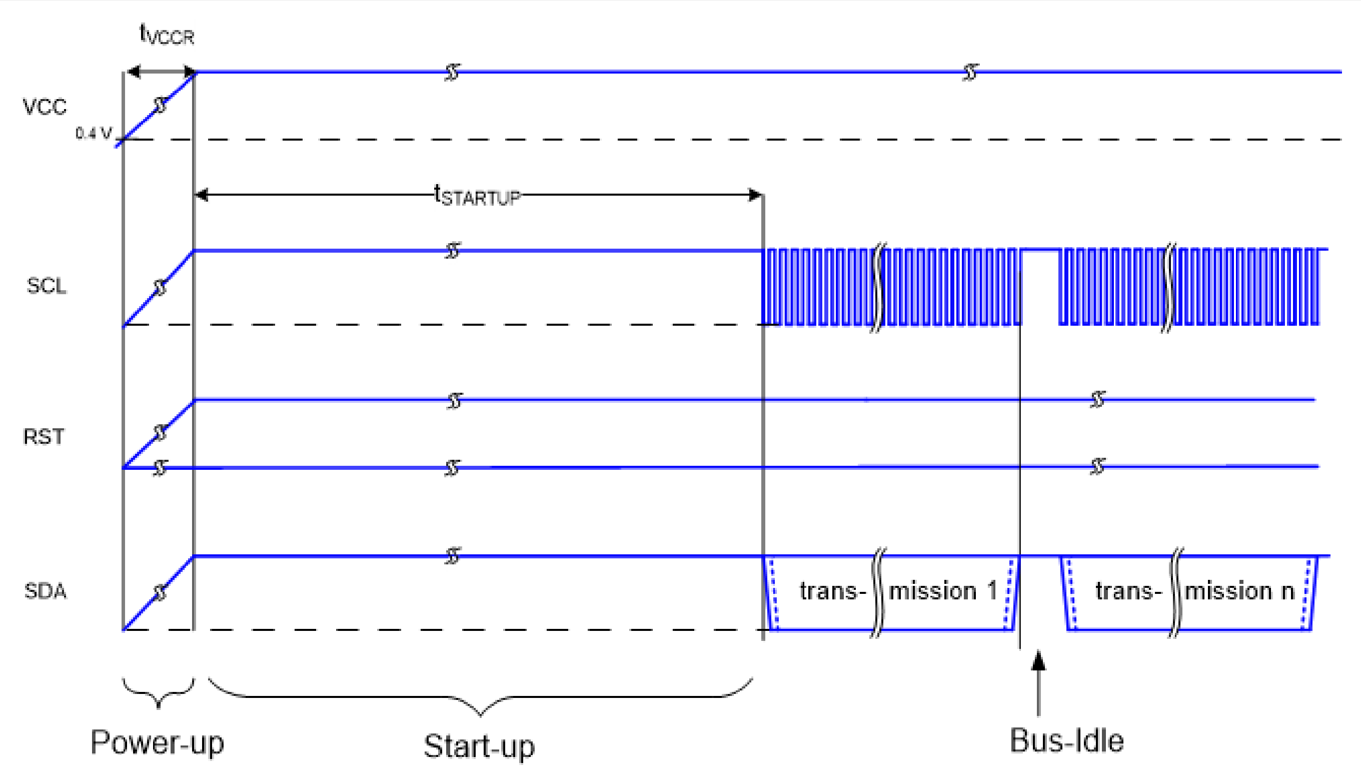


Figure 10 Start-up of I2C interface after power-on

Table 10 start-up of I2C interface after power-on

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
Startup time	$t_{STARTUP}$	15	-	-	ms	-

5.1.4.2 Start-up for warm resets

When using the reset signal for triggering a warm reset after power-on, the activation of the *I2C* interface needs the following reset procedure:

- VCC remains powered up
- The terminal stops I2C communication. *SDA* and *SCL* lines are set to high level before *RST* is set to low level
- After its falling edge, RST has to be kept at low level for at least t_1. At the latest t_2 after the falling edge of RST, the terminal must set RST to high level
- The first transmission may start at the earliest $t_{STARTUP}$ after the rising edge of RST

The following figure shows the timing for this start-up case.

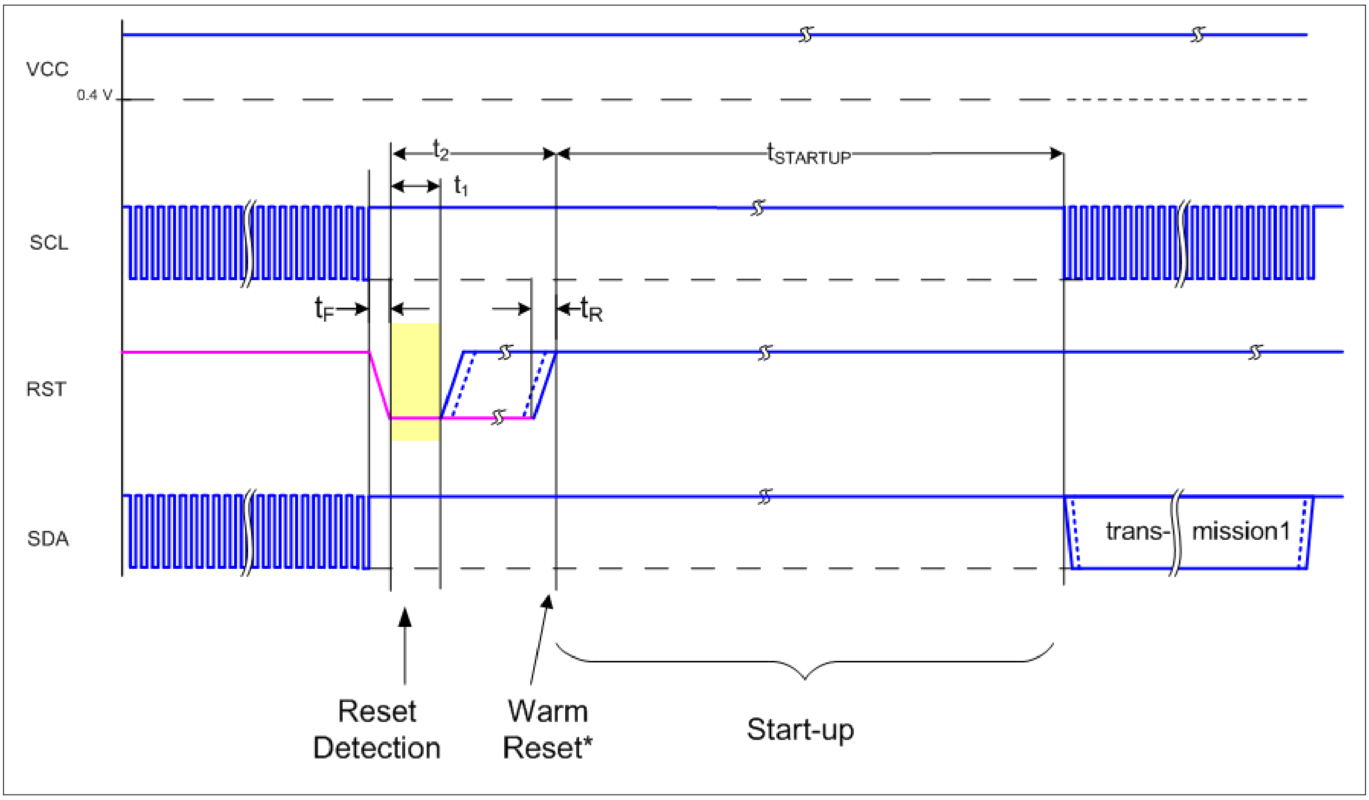


Figure 11 Start-up of I2C interface for warm resets

***Note**: If NVM programming was requested prior to the reset, $t_{STARTUP}$ will be extended from a typical value of 15 ms to a maximum of 20 ms.*

Table 11 Start-up of I2C interface for warm resets[8]

Parameter	Symbol	Values			Unit	Note or test condition
		Min.	Typ.	Max.		
Start-up time	$t_{STARTUP}$	15	-	-	ms	-
Rise time	t_R	-	-	1	µs	From 10% to 90% of signal amplitude
Fall time	t_F	-	-	1	µs	From 10% to 90% of signal amplitude
Reset detection	t_1	10	-	-	µs	-
Reset low	-	10	-	2500	µs	-

[8] Reset triggered by software (without power off/on cycle)

6 Connecting to host

6.1 OPTIGA™ Trust Charge host software architecture

The OPTIGA™ Trust Charge host library layers are explained in Figure 12. The following section explains, how to communicate host library with OPTIGA™ Trust Charge using *I2C*.

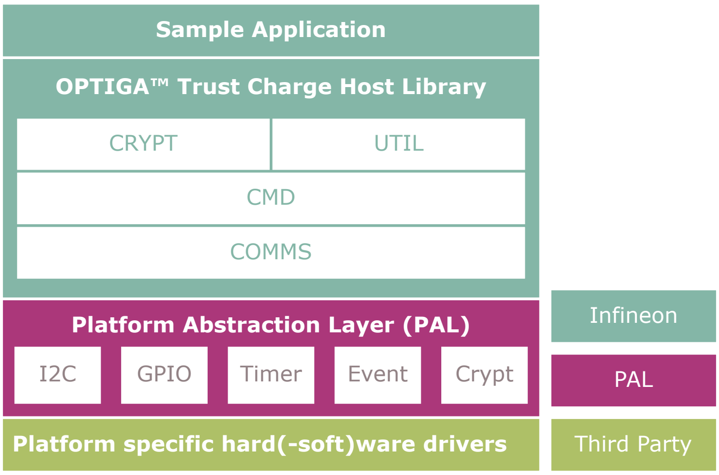

Sample Application
OPTIGA™ Trust Charge Host Library
CRYPT
UTIL
CMD
COMMS
Platform Abstraction Layer (PAL)
I2C
GPIO
Timer
Event
Crypt
Platform specific hard(-soft)ware drivers
Infineon
PAL
Third Party

Figure 12 **OPTIGA™ Trust Charge host software architecture**

6.2 Release package folder structure

The following figure shows the release package structure when OPTIGA™ Trust Charge is installed/extracted on personal computer (PC).

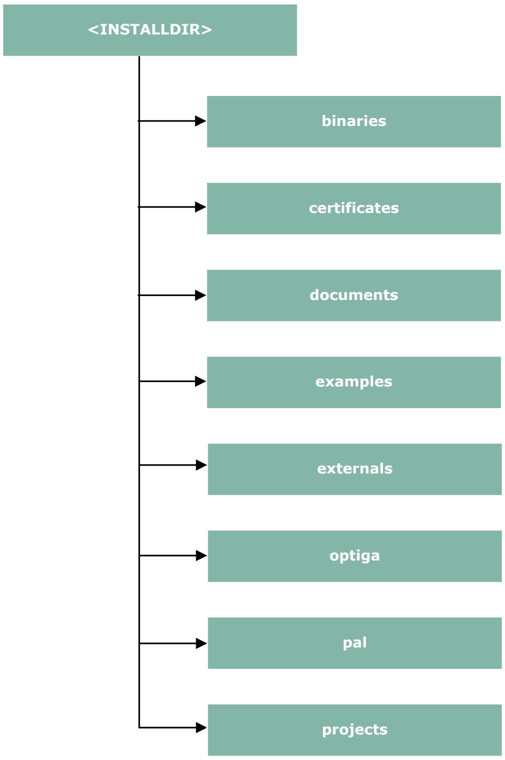


Figure 13 **Release package folder structure**

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

1. binaries: This directory contains binaries for OPTIGA™ Trust Charge sample application
2. certificates: This directory contains OPTIGA™ Trust Charge certificates
3. documents: This directory contains all relevant OPTIGA™ Trust Charge documentation
4. examples: This directory contains example use cases for Toolbox features and a tool for generation of manifest for secure data object feature
5. externals: This directory contains mbedtls software crypto libraries
6. optiga: This directory contains OPTIGA™ Trust Charge libraries
7. pal: This directory contains *PAL* for development kit and for mbedtls and wolfssl software crypto libraries
8. projects: This directory contains development kit sample project in DAVE™ workspace

The following figure shows the OPTIGA™ Trust Charge host software folder structure.

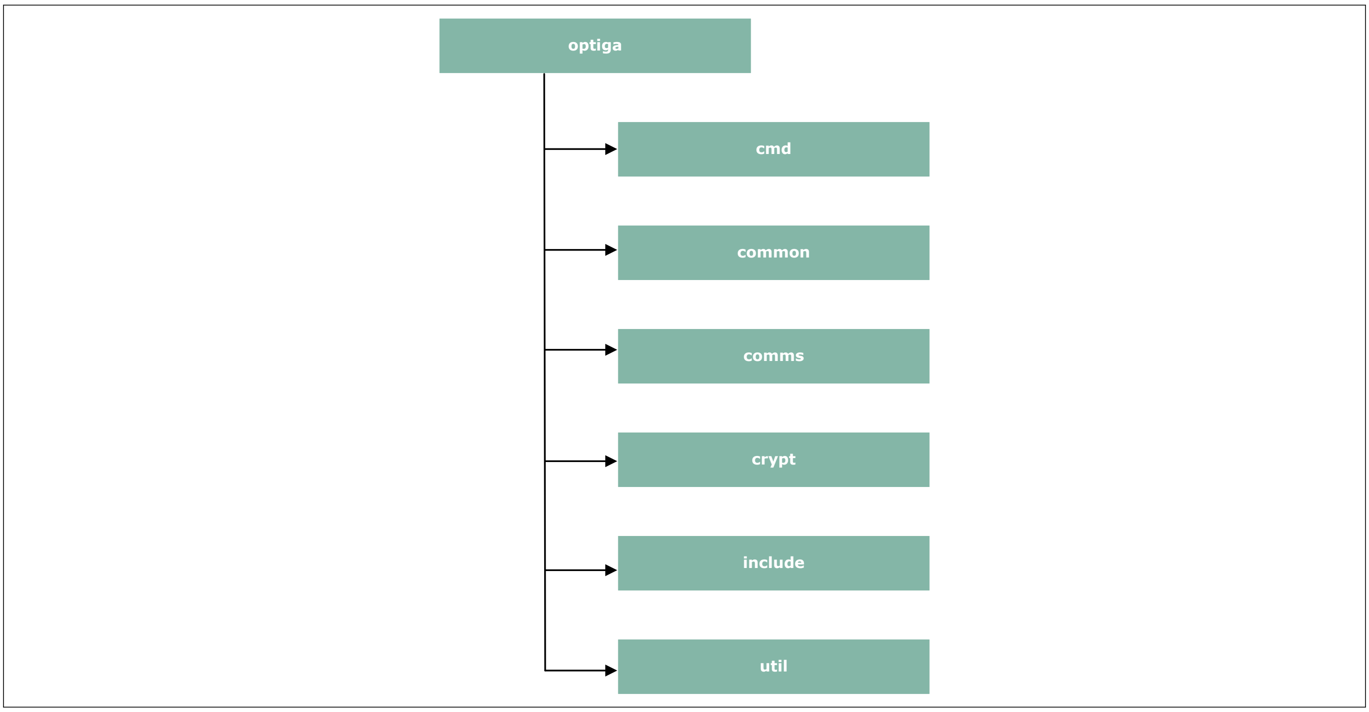


Figure 14 Host source folder structure

1. cmd: This folder contains sources for all OPTIGA™ Trust Charge commands
2. common: This folder contains the common functions used across all the modules
3. comms: This folder contains the driver to communicate with OPTIGA™ Trust Charge
4. crypt: This folder contains sources for cryptographic functionalities
5. include: This folder contains header files for all OPTIGA™ Trust Charge host software
6. util: This folder contains utility functions for example, read/write and open/close application

6.3 Porting notes

The implementation of *PAL* needs to be updated in order to migrate to a new target platform.

The PAL reference code for the XMC4700 Relax Kit is provided as part of package which can be used. The implementation can be found in "`<INSTALLDIR>/pal/xmc4700`" and the header files are available in "`<INSTALLDIR>/optiga/include`" with the required *API* used by upper layers. The header files are platform agnostic and would not require any changes. The low level drivers used by PAL for XMC4700 are configured and generated using DAVE™.

6.4 Communication with OPTIGA™ Trust Charge

The hardware/platform resource configuration with respect to *I2C* master and *GPIO* (Vdd and reset) are to be updated in pal_ifx_i2c_config.c. These configurations are used by the *IFX* I2C implementation to communicate with OPTIGA™ Trust Charge.

1. Update I2C master platform specific context [for example, (void*)&i2c_master_0]

```
/**
 * \brief PAL I2C configuration for OPTIGA
*/
pal_i2c_t optiga_pal_i2c_context_0 =
{
    /// Pointer to I2C master platform specific context
    (void*)&i2c_master_0,
    /// Slave address
    0x30,
    /// Upper layer context
    NULL,
    /// Callback event handler
    NULL
};
```

2. Update platform specific context for GPIO (Vdd and reset)

```
/**
 * \brief Vdd pin configuration for OPTIGA
*/
pal_gpio_t optiga_vdd_0 =
{
    // Platform specific GPIO context for the pin used to toggle Vdd
    (void*)&vdd_pin
};

/**
 * \brief Reset pin configuration for OPTIGA
*/
pal_gpio_t optiga_reset_0 =
{
    // Platform specific GPIO context for the pin used to toggle Reset
    (void*)&reset_pin
};
```

3. Update *PAL* I2C *API* [pal_i2c.c] to communicate with OPTIGA™ Trust Charge

The pal_i2c is expected to provide the API for I2C driver initialization, de-initialization, read, write and set bit rate kind of operations.

a. pal_i2c_init
b. pal_i2c_deinit
c. pal_i2c_read
d. pal_i2c_write
e. pal_i2c_set_bitrate

A few target platforms, the I2C master driver initialization (pal_i2c_init) is done during the platform start up. In such an environment, there is no need to implement pal_i2c_init and pal_i2c_deinit functions. Otherwise, these (pal_i2c_init and pal_i2c_deinit) functions must be implemented as per the upper layer expectations based on the need. The details of these expectations are available in the host library API documentation (chm).

The reference implementation of PAL I2C based on development kit does not need to have the platform I2C driver initialization explicitly done as part of pal_i2c_init as it is taken care by the DAVE™ library initialization. Therefore, pal_i2c_init and pal_i2c_deinit are not implemented.

In addition to the above specified API, the PAL I2C must handle the events from the low level I2C driver and invoke the upper layer handlers registered with PAL I2C context for the respective transaction as shown in the below example.

```
//I2C driver callback function when the transmit is completed successfully
void i2c_master_end_of_transmit_callback(void)
{
    invoke_upper_layer_callback(gp_pal_i2c_current_ctx,
                                (uint8_t)PAL_I2C_EVENT_TX_SUCCESS)
 }
```

In above example the I2C driver callback, when transmission is successful invokes the handler to inform the result.

4. Update PAL GPIO [pal_gpio.c] to power on and reset the OPTIGA™ Trust Charge

- **a.** pal_gpio_set_high
- **b.** pal_gpio_set_low

5. Update PAL timer [pal_os_timer.c] to enable timer

- **a.** pal_os_timer_get_time_in_milliseconds
- **b.** pal_os_timer_delay_in_milliseconds

6. Update event management for the asynchronous interactions for IFX I2C [pal_os_event.c]

- **a.** pal_os_event_register_callback_oneshot
- **b.** pal_os_event_trigger_registered_callback

The pal_os_event_register_callback_oneshot function is expected to register the handler and context provided as part of input parameters and triggers the timer for the requested time. The p_pal_os_event is an event instance created using pal_os_event_create.

```
void pal_os_event_register_callback_oneshot(
                              pal_os_event_t * p_pal_os_event,
                              register_callback callback,
                              void* callback_args,
                              uint32_t time_us)
{
    p_pal_os_event->callback_registered = callback;
    p_pal_os_event->callback_ctx = callback_args;

    //lint --e{534} suppress "Return value is not required to be checked"
    TIMER_SetTimeInterval(&scheduler_timer, (time_us*100));
    TIMER_Start(&scheduler_timer);
}
```

The handler registered must be invoked once the timer has elapsed as shown in pal_os_event_trigger_registered_callback. The pal_os_event_trigger_registered_callback is to be registered with event timer interrupt to get trigerred when the timer expires. The pal_os_event_0 is the instance in the pal_os_event used store the registered callback and context.

```
void pal_os_event_trigger_registered_callback(void)
{
    register_callback callback;

    TIMER_ClearEvent(&scheduler_timer);
    //lint --e{534} suppress "Return value is not required to be checked"
    TIMER_Stop(&scheduler_timer);
    TIMER_Clear(&scheduler_timer);

    if (pal_os_event_0.callback_registered)
    {
        callback = pal_os_event_0.callback_registered;
        callback((void * )pal_os_event_0.callback_ctx);
    }
}
```

6.5 Reference code for communicating with OPTIGA™ Trust Charge

```
static volatile uint32_t optiga_pal_event_status;
static void optiga_pal_i2c_event_handler(void* upper_layer_ctx,
uint8_t event);

pal_i2c_t optiga_pal_i2c_context_0 =
{
    /// Pointer to I2C master platform specific context
    (void*)&i2c_master_0,
    /// Slave address
    0x30,
    /// Upper layer context
    NULL,
    /// Callback event handler
    NULL,
};

// OPTIGA pal i2c event handler
static void optiga_pal_i2c_event_handler(void* upper_layer_ctx,
                                          uint8_t event)
{
    optiga_pal_event_status = event;
}
```

```
/* Function to verify I2C communication with OPTIGA */
pal_status_t test_optiga_communication(void)
{
    pal_status_t pal_return_status;
    uint8_t data_buffer[10] = {0x82};

    // set callback handler for pal i2c
    optiga_pal_i2c_context_0.upper_layer_event_handler =
                          optiga_pal_i2c_event_handler;

    // Send 0x82 to read I2C_STATE from optiga
    do
    {
         optiga_pal_event_status = PAL_I2C_EVENT_BUSY;
         pal_return_status =
                    pal_i2c_write(&optiga_pal_i2c_context_0,
                                  data_buffer,
                                  1);
        if (PAL_STATUS_FAILURE == pal_return_status)
        {
        // Pal I2C write failed due to I2C busy is in busy
           // state or low level driver failures
            break;
        }

    // Wait until writing to optiga is completed
    }  while (PAL_I2C_EVENT_SUCCESS != optiga_pal_event_status);


    // Read the I2C_STATE from OPTIGA
    do
    {
        optiga_pal_event_status = PAL_I2C_EVENT_BUSY;
        pal_return_status =
                      pal_i2c_read(&optiga_pal_i2c_context_0 ,
                                   data_buffer ,
                                   4);
    // Pal I2C read failed due to I2C busy is in busy
      // state or low level driver failures
        if (PAL_STATUS_FAILURE == pal_return_status)
        {
            break;
        }
    // Wait until reading from optiga is completed
    } while (PAL_I2C_EVENT_SUCCESS != optiga_pal_event_status);

    return pal_return_status;
}

/* Main Function */
int32_t main(void)
{
    DAVE_STATUS_t status;
    pal_status_t pal_return_status;

    // Initialisation of DAVE Apps
    status = DAVE_Init();

    // Stop if DAVE init fails
    if (DAVE_STATUS_FAILURE == status)
    {
        while (1U)
        {;}
    }
    pal_return_status = test_optiga_communication();

    return (int32_t)pal_return_status;068
}
```

7 OPTIGA™ Trust Charge external interface

7.1 Commands

This section provides short description of the commands exposed by the OPTIGA™ Trust Charge security chip and mapping of these commands with respect to use cases.

Table 12 Command table

Command name	Description
OpenApplication	Command to launch an application
CloseApplication	Command to close/hibernate an application
GetDataObject	Command to get (read) a data object
SetDataObject	Command to set (write) a data object
SetObjectProtected	Command to set (write) data objects protected (integrity protection)
GetRandom	Command to generate a random stream
CalcHash	Command to calculate a Hash
CalcSign	Command to calculate a signature
VerifySign	Command to verify a signature
GenKeyPair	Command to generate public/private key pairs

Table 13 Mapping of commands with use cases

Use case	OPTIGA™ Trust Charge commands used
Charge authentication request	GetRandom, CalcHash, VerifySign
Charge authentication response	CalcHash, CalcSign
Datastore (user memory ~10 KB)	GetDataObject and SetDataObject
Secure firmware update	VerifySign
Secure update of trust anchors on security chip	SetObjectProtected command

7.2 Crypto performance

The performance metrics for various schemes are provided in Table 14. If not particularly mentioned, the performance is measured at OPTIGA™ Trust Charge *input/output (I/O)* interface with:

- *I2C* FM (400 kHz)
- Without power limitation
- At 25°C
- V_{CC} = 3.3 V
- *elliptic curve digital signature algorithm (ECDSA)* signature scheme: ECDSA *Federal Information Processing Standards (FIPS)* 186-3 without hashing
- Hash scheme: *SHA* 256
- *ECC* key size: 256 bits (*National Institute of Standards and Technology (NIST)* P-256)

Table 14 Crypto performance

Scheme	Algorithm	Performance in ms[9]	Performance with shielded connection in ms[9]	Notes
Calculate signature	ECDSA	~60	~65	Does not include message hashing before calling a toolbox function
Verify signature	ECDSA	~85	~90	
Key pair generation	ECC	~75	~80	Generate 256-bit ECC key pair
Hash calculation	SHA 256	~5 KB/s	~4.5 KB/s	In blocks of 500 bytes

[9] Minimum execution of the entire sequence in ms, except the external world timings

8 Security monitor

The security monitor is a central component which enforces the security policy of the OPTIGA™ Trust Charge. It consumes security events sent by security aware parts of the OPTIGA™ Trust Charge embedded software and takes actions accordingly as specified in Security monitor policy.

8.1 Security events

The events below actively influence the security monitor.

Table 15 Security events

Event	Description
Private key use	This event occurs in case the internal services are going to use an OPTIGA™ Trust Charge hosted private key
Suspect system behavior	This event occurs in case the embedded software detects inconsistencies with the expected behavior of the system. Those inconsistencies might be redundant information which does not fit to their counterpart

8.2 Security monitor policy

Security monitor judges the notified security events regarding the number of occurrence over time and in case those violate the permitted usage profile of the system takes actions to throttle down the performance and so the possible frequency of attacks.

The permitted usage profile is defined as:

1. t_{max} is set to 5 seconds (±5%)
2. A suspect system behavior event is never permitted and will cause setting the *security event counter (SEC)* to its maximum (= 255)
3. One protected operation (refer to Table 15) events per t_{max} period

In other words it must not allow more than one out of the protected operations per t_{max} period (worst case, refer to bullet 3. above). This condition must be stable, at least after 500 uninterrupted executions of protected operations.

For more information, refer to Solution Reference Manual document available as part of the package.

A Infineon I2C protocol registry map

OPTIGA™ Trust Charge supports *IFX I2C* protocol specification (refer to [2]) and is implemented as I2C slave, which uses different address locations for status, control and data communication registers. These registers with description are outlined below in the following table.

Table 16 IFX I2C registry map table

Register address	**Name**	**Size in bytes**	**Description**	**Master access**
0x80	DATA	DATA_REG_LEN	This is the location where data shall be read from or written to the I2C slave	Read/Write
0x81	DATA_REG_LEN	2	This register holds the maximum data register (Addr 0x80) length. The allowed values are 0x0010 up to 0xFFFF. After writing the new data register length it becomes effective with the next I2C master access. However, in case the slave could not accept the new length it indicates its maximum possible length within this register. Therefore it is recommended to read the value back after writing it to be sure the I2C slave did accept the new value. ***Note**: The value of MAX_PACKET_SIZE is derived from this value or vice versa (MAX_PACKET_SIZE= DATA_REG_LEN-5)*	Read/Write
0x82	I2C_STATE	4	Bits 31:24 of this register provides the I2C state in regards to the supported features (for example, clock stretching …) and whether the device is busy executing a command and/or ready to return a response etc., Bits 15:0 defining the length of the response data block at the physical layer.	Read only
0x83	BASE_ADDR	2	This register holds the I2C base address as specified by Table 17. Default value is 0x30. After writing a different address the new address become effective with the next I2C master access. In case the bit 15 is set in addition to the new address (bit 6:0) it becomes the new default address at reset (persistent storage).	Write only

(table continues...)

Table 16 (continued) IFX I2C registry map table

Register address	Name	Size in bytes	Description	Master access
0x84	MAX_SCL_FREQU	4	This register holds the maximum clock frequency in kHz supported by the I2C slave. The value gets adjusted to the register I2C_Mode setting. Fast mode (FM): The allowed values are 50 up to 400. Fast mode (FM+): The allowed values are 50 up to 1000.	Read
0x85	GUARD_TIME[10]	4	For details refer to Table 20	Read only
0x86	TRANS_TIMEOUT[10]	4	For details refer to Table 20	Read only
0x88	SOFT_RESET	2	Writing to this register will cause a device reset. This feature is optional	Write only
0x89	I2C_MODE	2	This register holds the current I2C Mode as defined by Table 18. The default mode is SM and FM (011B)	Read/Write

Table 17 Definition of BASE_ADDR

Fields	Bits	Value	Description
DEF_ADDR	15	0	Volatile address setting by bit 6:0, lost after reset
		1	Persistent address setting by bit 6:0, becoming default after reset
BASE_ADDR	6:0	0x00-0x7F	I2C base address specified by Table 16

15	14	13	12	11	10	9	8
DEF_ADDR	*reserved for future use (RFU)*						
7	6	5	4	3	2	1	0
RFU	BASE_ADDR						

15	14	13	12	11	10	9	8
DEF_MODE	RFU						
7	6	5	4	3	2	1	0
RFU					Mode		

[10] In case the register returns 0xFFFFFFFF the register is not supported and the default values specified in Table 'list of protocol variations' shall be applied.

Table 18 Definition of I2C_MODE

Fields	Bits	Value	Description
-	15	0	Volatile mode setting by bit 2:0, lost after reset.
		1	Persistent mode setting by bit 2:0, becoming default after reset. This bit is always read as 0
MODE[11]	2:0	001 010 011 100 other values	SM FM SM and Fm (fab out default) FM+ Not valid; writing will be ignored

31	30	29	28	27	26	25	24
BUSY	RESP_RDY	RFU		SOFT_RESET	CONT_READ	REP_START	CLK_STRETCHING
23	22	21	20	19	18	17	16
PRESENT_LAYER	RFU						
15-0							
Length of data block to be read							

Table 19 Definition of I2C_STATE

Field	Bit(s)	Value	Description
BUSY	31	0	Device is not busy
		1	Device is busy executing a command
RESP_RDY	30	0	Device is not ready to return a response
		1	Device is ready to return a response
SOFT_RESET	27	0	SOFT_RESET not supported
		1	SOFT_RESET supported
CONT_READ	26	0	Continue Read not supported
		1	Continue Read supported
REP_START	25	0	Repeated start not supported
		1	Repeated start supported
CLK_STRETCHING	24	0	Clock stretching not supported
		1	Clock stretching supported

(table continues...)

[11] This mode defines the adherence of the bus signals to the electrical characteristics according standard I2C bus specification

Table 19 (continued) Definition of I2C_STATE

Field	Bit(s)	Value	Description
PRESENT_LAYER	23	0	Presentation layer not supported
		1	Presentation layer supported

A.1 Infineon I2C protocol variations

To fit best to application specific requirements the protocol might be tailored by specifying a couple of parameters which is described in the following table.

Table 20 List of protocol variations

Parameter	Default value	Description
MAX_PACKET_SIZE	0x110	Maximum packet size accepted by the receiver. The protocol limits this value to 0xFFFF, but there might be project specific requirements to reduce the transport buffers size for the sake of less *random access memory (RAM)* footprint in the communication stack. If shortened, it could be statically defined or negotiated at the physical layer
WIN_SIZE	1	Window size of the sliding windows algorithm. The value could be 1 up to 2
MAX_NET_CHAN	1	Maximum number of network channels. The value could be 1 up to 16. One indicates the OSI layer 3 is not used and the CHAN field of the PCTR must be set to 0000
CHAINING	TRUE	Chaining on the transport layer is supported (TRUE) or not (FALSE)
TRANS_TIMEOUT	10 ms	(Re) transmission timeout specifies the number of milliseconds to be elapsed until the transmitter considers a frame transmission is lost and retransmits the non-acknowledged frame. The timer gets started as soon as the complete frame is transmitted. The value could be 1 up to 1000. However, the higher the number, the longer it takes to recover from a frame transmission error. ***Note***: *The acknowledge timeout on the receiver side must be shorter than the retransmission timeout to avoid unnecessary frame repetitions.*
TRANS_REPEAT	3	Number of transmissions to be repeated until the transmitter considers the connection is lost and starts a re-synchronization with the receiver. The value could be 1 up to 4
BASE_ADDR	0x30	*I2C* (base) address. This address could be statically defined or dynamically negotiated by the physical layer
MAX_SCL_FREQU	1000 kHz	Maximum *SCL* clock frequency in kHz

(table continues...)

Table 20 (continued) List of protocol variations

Parameter	Default value	Description
GUARD_TIME	50 µs	Minimum time to be elapsed at the I2C master measured from read data (STOP condition) until the next write data (start condition) is allowed to happen. ***Note**: For two consecutive accesses on the same device GUARD_TIME re-specifies the value of t_{BUF} as specified by [I2C bus].* ***Note**: Even if another* I2C *address is accessed in between GUARD_TIME has to be respected for two consecutive accesses on the same device.*
SOFT_RESET	1	Any write attempt to the SOFT_RESET register will trigger a warm reset (reset w/o power cycle). This register is optional and its presence is indicated by the I2C_STATE register's "SOFT_RESET" flag
PRESENT_LAYER	1	This flag at the I2C_STATE register indicates the optional availability of the presentation layer, which is providing confidentiality and integrity protection of payloads (*application protocol data unit (APDU)*s) transferred across the I2C interface. The presentation layer is used as part of shielded connection

B OPTIGA™ Trust Charge command/response I2C sample logs

The default *I2C* slave address for the OPTIGA™ Trust Charge is 0x30 [I2C_ADDR]. All the values in this section are specified in decimal form unless stated otherwise.

B.1 Sequence of commands to read coprocessor UID from OPTIGA™ Trust Charge

Pre-requisites

1. Ensure that the security device is powered up
2. The OPTIGA™ Trust Charge will not acknowledge the slave address sent by a host if it is either busy or in idle state. Therefore, the host must retry or repeat the transaction until it is successful or timed out for 100 milliseconds (extreme case)
3. The specified guard time must be applied between each attempt of write/read operation by the host *I2C* driver
4. The log information for OPTIGA™ Trust Charge commands specified in below tables contains the [*IFX* I2C] protocol information which comprises sequence numbers and checksum of the transactions
   - a. A sequence of commands must be strict for the OPTIGA™ Trust Charge (for example, OpenApplication followed by GetDataObject to read a coprocessor *unique identifier (UID)*)
   - b. A checksum in the data depends on the data received or sent via write/read operations. So any data change in the transaction is reflected in the check sum. Otherwise the write data transaction will not be accepted/acknowledged by the OPTIGA™ Trust Charge
5. The logs specified below are without the presentation layer (used for the shielded connection) of [IFX I2C protocol specification [2]]

B.1.1 Check the status [I2C_STATE]

This is a very basic register read operation which ensures the behavior of the read/write operations of the local host *I2C* driver.

Table 21 **Check I2C_STATE register of OPTIGA™ Trust Charge**

I2C_ADDR	Transaction type	Data values
30	Write [01 bytes]	82_H
30	Read [04 bytes]	08 80 00 00_H

B.1.2 Issue OpenApplication command

Before issuing any application specific command, for example read coprocessor *UID* using GetDataObject, it is a must to send the OpenApplication command to initialize the application on the OPTIGA™ Trust Charge as shown in below table.

Table 22 **OpenApplication on OPTIGA™ Trust Charge**

I2C_ADDR	Transaction type	Data values
Step 1: Send OpenApplication command to initiate the application context on the OPTIGA™ Trust Charge		
30	Write [27 bytes]	80 03 00 15 00 70 00 00 10 D2 76 00 00 04 47 65 6E 41 75 74 68 41 70 70 6C 04 $1A_H$

(table continues...)

Table 22 (continued) OpenApplication on OPTIGA™ Trust Charge

I2C_ADDR	Transaction type	Data values
Step 2: Read the I2C_STATE register [Repeat this step until the read contains the data as specified below]		
30	Write [01 bytes]	82_H
30	Read [04 bytes]	C8 80 00 05_H
Step 3: Read the DATA register [Acknowledgment from OPTIGA™ Trust Charge for the last data transaction]		
30	Write [01 bytes]	80_H
30	Read [05 bytes]	80 00 00 0C EC_H
Step 4: Read the I2C_STATE register [Repeat this step until the read contains the data as specified below]		
30	Write [01 bytes]	82_H
30	Read [04 bytes]	48 80 00 $0A_H$
Step 5: Read the DATA register which contains the response for the command issued		
30	Write [01 bytes]	80_H
30	Read [10 bytes]	00 00 05 00 00 00 00 00 14 87_H
Step 6: Send an acknowledgment for the data read		
30	Write [06 bytes]	80 80 00 00 0C EC_H

B.1.3 Read coprocessor UID

The Coprocessor UID contains the OPTIGA™ Trust Charge unique ID and the build information details. The GetDataObject command is used to read the Coprocessor UID information.

Table 23 Read Coprocessor UID

I2C_ADDR	Transaction type	Data values
Step 1: Send the GetDataObject command to read the coprocessor UID		
30	Write [17 bytes]	80 04 00 0B 00 01 00 00 06 E0 C2 00 00 00 64 F0 $9F_H$
Step 2: Read the I2C_STATE register [Repeat this step until the read contains the data as specified below]		
30	Write [01 bytes]	82_H
30	Read [04 bytes]	48 80 00 25_H
Step 3: Read the DATA register which contains the response for the command issued		
30	Write [01 bytes]	80_H

(table continues...)

Table 23 (continued) Read Coprocessor UID

I2C_ADDR	Transaction type	Data values
30	Read [37 bytes]	05 00 20 00 00 00 00 1B CD XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX YY YY ZZ ZZ_H ***Notes**:* ***1.*** *XX is the unique ID part of the co-processor UID* ***2.*** *"YY YY" is the OPTIGA™ Trust Charge build number in binary coded decimal (BCD) format* ***3.*** *ZZ ZZ is the checksum of the transaction*
Step 4: Send an acknowledgment for the data read		
30	Write [06 bytes]	80 81 00 00 56 30_H

C Power management

When operating, the power consumption of OPTIGA™ Trust Charge is limited to meet the requirements regarding the power limitation set by the host. The power limitation is implemented by utilizing the current limitation feature of the underlying hardware device in steps of 1 mA from 6 mA to 15 mA with a precision of ±5%.

C.1 Hibernation

This maximizes power saving (zero power consumption[12]), while the *I2C* bus stays connected. In this case OPTIGA™ Trust Charge saves the application context before power-off (switching off V_{CC}) and restores it after power-up. After power-up the application continues seamlessly from the state before hibernate.

C.2 Low power sleep mode

The OPTIGA™ Trust Charge automatically enters a low-power mode after a configurable delay. Once it has entered Sleep mode, the OPTIGA™ Trust Charge resumes normal operation as soon as its address is detected on the *I2C* bus.

In case no command is sent to the OPTIGA™ Trust Charge it behaves as shown in Figure 15.

1. As soon as the OPTIGA™ Trust Charge is idle it starts to count down the “delay to sleep” time (t_{SDY})
2. In case this time elapses the device enters the “go to sleep” procedure
3. The “go to sleep” procedure waits until all idle tasks are finished (for example, counting down the *SEC*). In case all idle tasks are finished and no command is pending, the OPTIGA™ Trust Charge enters sleep mode

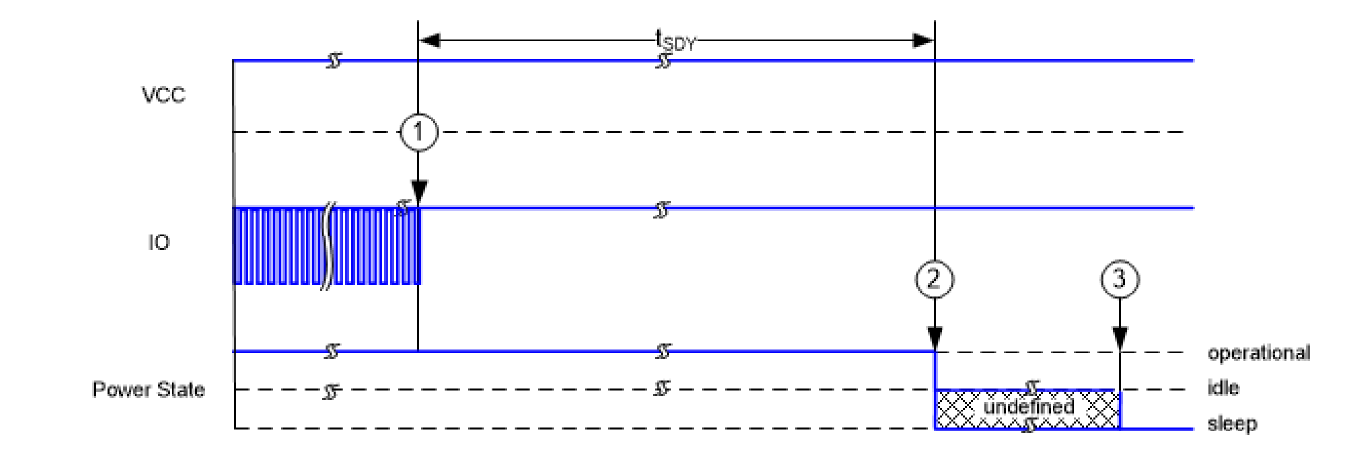


Figure 15 **Go-to-sleep procedure**

[12] Leakage current < 2.5 µA

RoHS compliance

On January 27, 2003 the European Parliament and the council adopted the directives:

- 2002/95/EC on the Restriction of the use of certain Hazardous Substances in electrical and electronic equipment ("RoHS")
- 2002/96/EC on Waste Electrical and Electrical and Electronic Equipment ("WEEE")

Some of these restricted (lead) or recycling-relevant (brominated flame retardants) substances are currently found in the terminations (e.g. lead finish, bumps, balls) and substrate materials or mold compounds.

The European Union has finalized the Directives. It is the member states' task to convert these Directives into national laws. Most national laws are available, some member states have extended timelines for implementation. The laws arising from these Directives have come into force in 2006 or 2007.

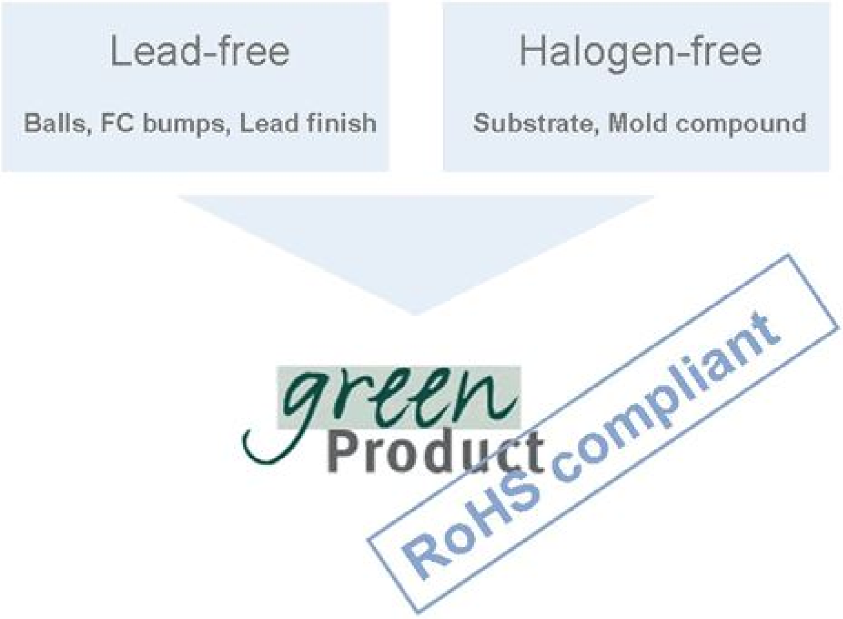
The electro and electronic industry has to eliminate lead and other hazardous materials from their products. In addition, discussions are on-going with regard to the separate recycling of certain materials, e.g. plastic containing brominated flame retardants.

Infineon is fully committed to giving its customers maximum support in their efforts to convert to lead-free and halogen-free[13] products. For this reason, Infineon's "Green Products" are ROHS-compliant.

Since all hazardous substances have been removed, Infineon calls its lead-free and halogen-free semiconductor packages "green." Details on Infineon's definition and upper limits for the restricted materials can be found here.

The assembly process of our high-technology semiconductor chips is an integral part of our quality strategy. Accordingly, we will accurately evaluate and test alternative materials in order to replace lead and halogen so that we end up with the same or higher quality standards for our products.

The use of lead-free solders for board assembly results in higher process temperatures and increased requirements for the heat resistivity of semiconductor packages. This issue is addressed by Infineon by a new classification of the Moisture Sensitivity Level (MSL). In a first step the existing products have been classified according to the new requirements.



[13] Any material used by Infineon is PBB and PBDE-free. Plastic containing brominated flame retardants, as mentioned in the WEEE directive, will be replaced if technically/economically beneficial.

References

[1] USB Organization: *USB Authentication Specification (Revision 1.0) with ECN and Errata*; 2019-01

[2] Infineon Technologies: *IFX I2C Protocol Specification (latest version)*

[3] Wireless Power Consortium: *Qi Specification – Authentication Protocol (latest version)*

Glossary

AC

alternating current (AC)

A synonym for dynamic parameters of an electronic circuit.

APDU

application protocol data unit (APDU)

The communication unit between a smart card reader and a smart card.

API

application programming interface (API)

A set of defined rules that enables various software components to communicate with each other.

BCD

binary coded decimal (BCD)

BSI

Bundesamt für Sicherheit in der Informationstechnik (BSI)

German Federal Office for Information Security.

CA

certificate authority (CA)

CC

Common Criteria for Information Technology Security Evaluation (CC)

An international standard (International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 15408) for computer security certification.

DC

direct current (DC)

A synonym for static parameters of an electronic circuit.

EAL

evaluation assurance level (EAL)

ECC

elliptic curve cryptography (ECC)

ECDSA

elliptic curve digital signature algorithm (ECDSA)

FIPS

Federal Information Processing Standards (FIPS)

Publicly announced standards developed by the United States federal government for use in computer systems by non-military government agencies and government contractors.

GPIO

general purpose input/output (GPIO)

I/O

input/output (I/O)

I2C

inter-integrated circuit (I2C)

A synchronous serial communication bus.

IFX

Infineon Technologies AG (IFX)

The stock market acronym for Infineon Technologies AG shares. It is sometimes used in diagrams or tables where the long term hinders readability.

IoT

Internet of Things (IoT)

MCU

microcontroller unit (MCU)

One or more processor cores along with memory and programmable input/output peripherals.

NIST

National Institute of Standards and Technology (NIST)

NVM

non-volatile memory (NVM)

OS

operating system (OS)

PAL

platform abstraction layer (PAL)

PCB

printed circuit board (PCB)

RAM

random access memory (RAM)

RFU

reserved for future use (RFU)

RST

reset (RST)

SCL

serial clock line (SCL)

SDA

serial data line (SDA)

SEC

security event counter (SEC)

SHA

Secure Hash Algorithm (SHA)

UID

unique identifier (UID)

USON

ultrathin small outline no-lead (USON)

An surface-mounted device (SMD) package delivery form.

WPC

Wireless Power Consortium (WPC)

Revision history

Reference	Description
Revision 1.5, 2024-06-07	
All	Migrated to latest template and updated editorial changes
Revision 1.4, 2020-09-22	
All	Productive release version
Revision 1.3, 2020-07-27	
All	Engineering sample release version
Revision 1.0, 2020-05-22	
All	Initial version
Revision 0.71, 2020-03-19	
All	Draft version Document security state updated and Iccavg value corrected.
Revision 0.7, 2020-01-27	
All	Product renamed and image added
Revision 0.6, 2019-12-06	
All	Incorporated review comments
Revision 0.5, 2019-12-05	
All	Internal release

Trademarks

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

Edition 2024-06-07

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email:
CSSCustomerService@infineon.com

Document reference
IFX-uzh1714036962159

Important notice

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, 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.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

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.

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.