# I13400

Generated by Doxygen 1.10.0

# Chapter 1

# ssl13400 Stack

## 1.1 Introduction

ssl13400 is high performance real-time ISO 13400 protocol stack written in ANSI C. ssl13400 adheres to both the ISO 13400 specification and to the software development best practices described in the MISRA C guidelines.

ssl13400 uses a modularized design with an emphasis on software readability and performance. ssl13400 is easy to understand and platform independent allowing it to be used on any CPU or DSP with or without an RTOS.

## 1.2 Integration

## 1.3 Configuration

## 1.4 Examples

# Chapter 2

# File Index

## 2.1 File List

Here is a list of all documented files with brief descriptions:

# Chapter 3

# File Documentation

## 3.1 doc/ssl13400.dox File Reference

## 3.2 src/i13400.c File Reference

```
#include "i13400.h"
#include "i13400cfg.h"
#include "i13400s.h"
#include "main.h"
#include "net.h"
#include "uds.h"
#include "udsnet.h"
#include <stdint.h>
#include <stdio.h>
#include <string.h>
```

### Functions

- void i13400_tx_gen_nack (uint8_t p, uint8_t nack)

  *Transmit a generic NACK.*
- void i13400_tx_rte_rsp (uint8_t p, uint16_t sa, uint8_t rsp)

  *Transmit a route response message.*
- void i13400_tx_rte_req (uint8_t p, uint16_t sa, uint8_t type)

  *Transmit a route request message.*
- uint8_t i13400_tx_diag_rsp (uint8_t p, uint16_t sa, uint16_t ta, uint8_t rsp)

  *Transmit a diagnostic response.*
- uint8_t i13400_tx_alive_rsp (uint8_t p, uint16_t sa)

  *Transmit an alive message.*
- uint8_t i13400_valid_source (uint16_t sa)

  *Check if a source address is valid for network.*
- uint8_t i13400_valid_ta (uint16_t ta)

  *Check if a source address is valid for network.*
- uint8_t i13400_auth_needed (uint16_t sa)

  *Check if network level authentication is required to access this source address.*

- uint8_t i13400_supported_type (uint16_t type)

  *Check if a message type is supported by this stack/application.*
- uint8_t i13400_oom (uint32_t len)

  *Check if we have enough room to buffer this message.*
- void i13400_handle_nack (uint8_t p, uint8_t nack)

  *Handle incoming NACK messages.*
- void i13400_handle_status_rsp (uint8_t p, uint8_t ∗data)

  *Handle a status response message.*
- void i13400_handle_route_rsp (uint8_t p, uint8_t ∗buf)

  *Handle a routing response message.*
- void i13400_process (uint8_t p, const i13400_msg ∗const msg)

  *Handle messages after they've passed basic checks.*
- void i13400_post (uint8_t p, uint8_t ∗buf, uint32_t buf_len)

  *Handle packets from the network layer.*

## 3.2.1 Detailed Description

**Copyright**

Simma Software, Inc. - 2019 Use this software at your own risk. Simma Software, Inc does not promise, state, or guarantee this software to be defect free.

ssl13400-2 (Version 0.1)

This file contains software that relates to the implementation of the network and transport layers of ISO 13400.

## 3.2.2 Function Documentation

### 3.2.2.1 i13400_auth_needed()

```
uint8_t i13400_auth_needed (
            uint16_t sa )
```

Check if network level authentication is required to access this source address.

**Parameters**

| | |
|---|---|
| *sa* | Source Address |

**Returns**

0 if valid

### 3.2.2.2 i13400_handle_nack()

```
void i13400_handle_nack (
            uint8_t p,
            uint8_t nack )
```

Handle incoming NACK messages.

**Parameters**

| | |
|---|---|
| *p* | Source port |
| *nack* | NACK code |

### 3.2.2.3 i13400_handle_route_rsp()

```
void i13400_handle_route_rsp (
             uint8_t p,
             uint8_t * buf )
```

Handle a routing response message.

**Parameters**

| | |
|---|---|
| *p* | Port |
| *buf* | The message |

### 3.2.2.4 i13400_handle_status_rsp()

```
void i13400_handle_status_rsp (
             uint8_t p,
             uint8_t * data )
```

Handle a status response message.

**Parameters**

| | |
|---|---|
| *p* | Port |
| *data* | The message |

### 3.2.2.5 i13400_oom()

```
uint8_t i13400_oom (
             uint32_t len )
```

Check if we have enough room to buffer this message.

**Parameters**

| | |
|---|---|
| *len* | Length of incoming message in bytes |

**Returns**

0 if enough room

#### 3.2.2.6 i13400_post()

```
void i13400_post (
            uint8_t p,
            uint8_t * buf,
            uint32_t buf_len )
```

Handle packets from the network layer.

**Parameters**

| | |
|---|---|
| *p* | Port |
| *buf* | Packet |
| *buf_len* | Length of packet |

#### 3.2.2.7 i13400_process()

```
void i13400_process (
            uint8_t p,
            const i13400_msg *const msg )
```

Handle messages after they've passed basic checks.

**Parameters**

| | |
|---|---|
| *p* | Port |
| *msg* | The message |

#### 3.2.2.8 i13400_supported_type()

```
uint8_t i13400_supported_type (
            uint16_t type )
```

Check if a message type is supported by this stack/application.

**Parameters**

| | |
|---|---|
| *type* | Message Type |

**Returns**

1 if supported

#### 3.2.2.9 i13400_tx_alive_rsp()

```
uint8_t i13400_tx_alive_rsp (
            uint8_t p,
            uint16_t sa )
```

Transmit an alive message.

**Parameters**

| | |
|---|---|
| *p* | Port for data to go out |
| *sa* | Source Address |

#### 3.2.2.10 i13400_tx_diag_rsp()

```
uint8_t i13400_tx_diag_rsp (
             uint8_t p,
             uint16_t sa,
             uint16_t ta,
             uint8_t rsp )
```

Transmit a diagnostic response.

**Parameters**

| | |
|---|---|
| *p* | Port for data to go out |
| *sa* | Source Address |
| *ta* | Target Address of message |
| *type* | type of request to be transmitted |

#### 3.2.2.11 i13400_tx_gen_nack()

```
void i13400_tx_gen_nack (
             uint8_t p,
             uint8_t nack )
```

Transmit a generic NACK.

**Parameters**

| | |
|---|---|
| *p* | Port for data to go out |
| *nack* | NACK code to be transmitted |

#### 3.2.2.12 i13400_tx_rte_req()

```
void i13400_tx_rte_req (
             uint8_t p,
             uint16_t sa,
             uint8_t type )
```

Transmit a route request message.

**Parameters**

| | |
|---|---|
| *p* | Port for data to go out |
| *sa* | Source Address |
| *type* | type of request to be transmitted |

#### 3.2.2.13 i13400_tx_rte_rsp()

```
void i13400_tx_rte_rsp (
            uint8_t p,
            uint16_t sa,
            uint8_t rsp )
```

Transmit a route response message.

**Parameters**

| | |
|---|---|
| *p* | Port for data to go out |
| *sa* | Source Address |
| *rsp* | rsp code to be transmitted |

#### 3.2.2.14 i13400_valid_source()

```
uint8_t i13400_valid_source (
            uint16_t sa )
```

Check if a source address is valid for network.

**Parameters**

| | |
|---|---|
| *sa* | Source Address |

**Returns**

0 if valid

#### 3.2.2.15 i13400_valid_ta()

```
uint8_t i13400_valid_ta (
            uint16_t ta )
```

Check if a source address is valid for network.

**Parameters**

| | |
|---|---|
| *sa* | Source Address |

**Returns**

0 if valid

## 3.3 src/i13400s.c File Reference

```
#include "i13400s.h"
```

```
#include "i13400.h"
#include "i13400app.h"
#include "i13400cfg.h"
#include <stdint.h>
```

### Functions

- uint8_t i13400_tx_rte_rsp (uint8_t p, uint16_t sa, uint8_t rsp)

    *Transmit a route response message.*

## 3.3.1 Detailed Description

**Copyright**

Simma Software, Inc. - 2020

Use this software at your own risk. Simma Software, Inc does not promise, state, or guarantee this software to be defect free. Use of this software requires agrement to the Simma Software license agreement.

ssi13400-socket (version 0.1)

This file contains software that relates to the implementation of the socket layer of ISO 13400-2 (DoIP).

If modifications of this file are required, please contact your technical services representative at Simma Software or whitelje AT simmasoftware.com

## 3.3.2 Function Documentation

### 3.3.2.1 i13400_tx_rte_rsp()

```
uint8_t i13400_tx_rte_rsp (
            uint8_t p,
            uint16_t sa,
            uint8_t rsp )  [extern]
```

Transmit a route response message.

**Parameters**

| | |
|---|---|
| *p* | Port for data to go out |
| *sa* | Source Address |
| *rsp* | rsp code to be transmitted |

# Index