CALYPSO REFERENCE MANUAL

AMB5201 / 261001102500X

VERSION 1.2

MARCH 6, 2019

Revision history

Manual version	FW version	HW version	Notes	Date
1.0	1.0.0	2.0	• Initial release of the manual	January 2019
1.1	1.0.0	2.0	• Added chapter `Reference design` • Added chapter `Information for Ex Protection`	February 2019
1.2	1.0.0	2.0	• Added known issues in chapter `Firmware history`	March 2019

Abbreviations

Abbreviation	Name	Description
NWP	Network processor unit	802.11 network processor unit
AP	Access point	WLAN (IEEE 802.11) infrastructure node offering stations to connect to
DC	Duty cycle	Transmission time in relation of one hour. 1% means, channel is occupied for 36 seconds per hour.
FSE	Field Sales Engineer	Support and sales contact person responsible for limited sales area
0xhh [HEX]	Hexadecimal	All numbers beginning with 0x are stated as hexadecimal numbers. All other numbers are decimal.
HIGH	High signal level	
LOW	Low signal level	
LSB	Least significant bit	
MSB	Most significant bit	
PL	Payload	The real, non-redundant information in a frame/packet.
RF	Radio frequency	Describes everything relating to the wireless transmission.
STA	Station	WLAN (802.11) node in station role, can connect to an AP
UART		Universal Asynchronous Receiver Transmitter allows communicating with the module of a specific interface.
US	UserSettings	Any relation to a specific entry in the UserSettings is marked in a special font and can be found in the respective chapter.
VDD	Supply voltage	
Wi-Fi		Is a Registered Trademark of the Wi-Fi Alliance for interoperability tested WLAN (IEEE 802.11) based products.

Abbreviation	Name	Description
WLAN	Wireless Local Area Network	
P2P	Peer to Peer	WLAN configuration
MAC	Medium access control	
IEEE	Institute of Electrical and Electronics Engineers	
IP	Internet Protocol	Network layer protocol
TCP	Transmission Control Protocol	Transport layer protocol
UDP	Use Datagram Protocol	Transport layer protocol
SSL	Secure Sockets Layer	Transport layer protocol
TLS	Transport Layer Security	Transport layer protocol
HTTP(s)	Hypertext transfer protocol (secure)	Application layer protocol
MQTT	Message Queuing Telemetry Transport	Application layer protocol
OTA	Over The Air	Update mechanism
DHCP	Dynamic Host Configuration Protocol	Application layer protocol
WEP	Wired Equivalent Privacy	802.11 security algorithm
WPA	Wi-Fi Protected Access	Wi-Fi security algorithm
WPS	Wi-Fi Protected Setup	Wi-Fi security algorithm
DNS	Domain Name System	Application layer protocol
mDNS	multicast-DNS	Application layer protocol
LLA	Link-local addressing	IPv4/IPv6 local addressing mechanism
DAD	Duplicate address detection	IPv4/IPv6 addressing mechanism

Contents

1. Introduction

The Calypso WLAN module is a compact WLAN radio module based on IEEE 802.11 b/g/n with a fully featured TCP/IP stack. The edge castellated connections, smart antenna configuration and an easy-to-use AT-style command interface enables easy integration of Calypso into any embedded application.
The module supports IPv4 as well as IPv6 and implements several commonly used network applications like SNTP, DHPv4, DHCPv6, mDNS, HTTP(S), MQTT out-of the box. Advanced security features like up to 6 simultaneous secure sockets, secure boot, secure storage and secure OTA update provide a good basis for a secure end product.
Whether a serial cable replacement or low power IoT application with cloud connectivity, the Calypso WLAN modules offers a robust and standard compliant wireless connectivity solution for low-power and low-medium throughput applications.
WLAN will be used as a synonym for IEEE 802.11 standard compliant radio communication throughout this manual.
Calypso is Wi-Fi Certified. The Certification ID is WFA81685.

1.1. Operational description

The Calypso WLAN module is intended to be used as a radio sub-system in order to provide WLAN communication capabilities to the system.
The UART acts as the primary interface between the module and a host micro-controller. The module can be fully configured and operated using a set of AT-commands over UART. Once configured, the module independently manages WLAN connectivity allowing the host controller to utilize its resources elsewhere.
Therefore, when using the standard firmware, a host MCU is required in the end product to control and access the radio module. Stand alone applications, without host, can be realized with a custom firmware development.

1.2. Block diagram

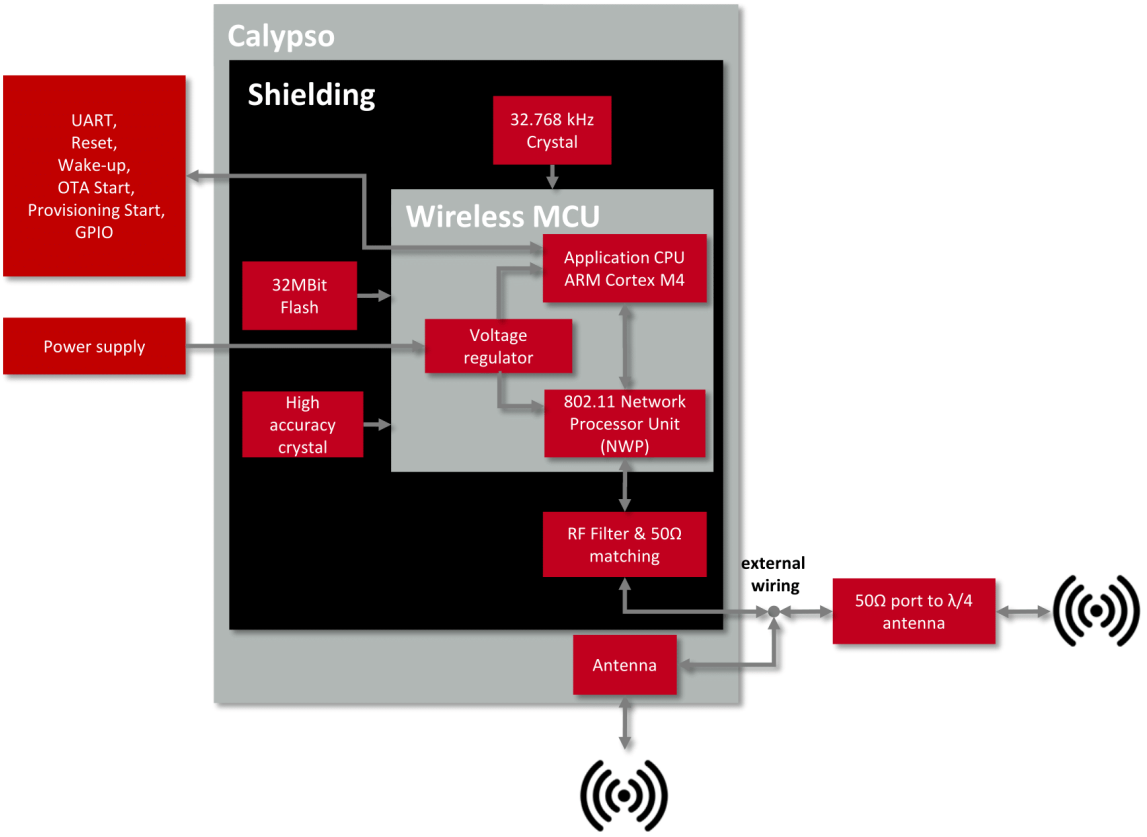


Figure 1: Block diagram

1.3. Ordering information

WE order code	Former order code	Description
2610011025000	AMB5201-TR	WLAN module in T&R packaging
2610011025009	AMB5201-DEV	3 pcs WLAN module
2610019225001	AMB5201-EV	EV kit for WLAN module

Table 1: Ordering information

2. Electrical specifications

Unless otherwise stated, all the values given here were measured on the Calypso evaluation board under the following conditions:T=25℃, VDDS=3.6V, internal DC-DC converter active and 50 Ω conducted.

2.1. Recommended operating conditions

Description	Min.	Typ.	Max.	Unit
VCC	2.1	3.3	3.6	V
Temperature range	-40	25	85	℃
Ambient thermal slew rate	-20		20	℃ / min

Table 2: Recommended operating conditions

When operating at an ambient temperature of over 75℃, the transmit duty cycle must remain below 50feature of the power amplifier. If the auto-protect feature triggers, the device takes a maximum of 60 seconds to restart the transmission.

2.2. Absolute maximum ratings

Description	Min.	Typ.	Max.	Unit
VCC	-0.5		3.8	V

Table 3: Absolute maximum ratings

2.3. Power consumption

2.3.1. Static

Description	Min	Typ.	Max	Unit
TX current consumption at max output power		230		mA
RX current consumption		76		mA
Low power mode		10		µA
Peak calibration current, *VCC*=2.1V		670		mA
Peak calibration current, *VCC*=3.3V		450		mA

Table 4: Power consumption

2.4. Radio characteristics

Description	Min	Typ.	Max	Unit
Max output power		16	18	dBm
Input sensitivity (1 Mbit)	-94	-92		dBm
Max input level, 802.11b		-4		dBm
Max input level, 802.11g		-10		dBm
Frequencies	2412		2472	MHz

Table 5: Radio characteristics

Standard	Modulation and coding	Peak Data rate
802.11b	DBPSK(DSSS)	1 Mbps
	DQPSK(DSSS)	2 Mbps
	DQPSK(CCK)	5.5 Mbps
	DQPSK(CCK)	11 Mbps
802.11g	BPSK(OFDM) coding rate 1/2	6 Mbps
	BPSK(OFDM) coding rate 3/4	9 Mbps
	QPSK(OFDM) coding rate 1/2	12 Mbps
	QPSK(OFDM) coding rate 3/4	18 Mbps
	16-QAM(OFDM) coding rate 1/2	24 Mbps
	16-QAM(OFDM) coding rate 3/4	36 Mbps
	64-QAM(OFDM) coding rate 2/3	48 Mbps
	64-QAM(OFDM) coding rate 3/4	54 Mbps
802.11n	BPSK(OFDM) coding rate 1/2	7.2 Mbps
	QPSK(OFDM) coding rate 1/2	14.4 Mbps
	QPSK(OFDM) coding rate 3/4	21.7 Mbps
	16-QAM(OFDM) coding rate 1/2	28.9 Mbps
	16-QAM(OFDM) coding rate 3/4	43.3 Mbps
	64-QAM(OFDM) coding rate 2/3	57.8 Mbps
	64-QAM(OFDM) coding rate 3/4	65Mbps
	64-QAM(OFDM) coding rate 5/6	72.2 Mbps

Table 6: Modulation schemes and peak data rate.

2.5. Pin characteristics

Property	Min	Typ.	Max	Unit
RF Pin input voltage			2.1	V
GPIO Voltage Input high	0.65 × *VCC*		*VCC*	V
GPIO Voltage Input low	-0.5		0.35 × *VCC*	V
GPIO Voltage Output high	0.8 × *VCC*		VCC	V
GPIO Voltage Output low	0		0.2 × *VCC*	V
/RESET Voltage Input low		0.6		V
Pin output current sunk by any I/O and control pin, drive mode dependant		2		mA
Pin output current sourced by any I/O and control pin, drive mode dependant		2		mA

Table 7: Pin characteristics

2.6. TX power vs current consumption

The following tables contains the typical TX power values and the corresponding typical average current for 3.6V supply voltage and 25℃ ambient temperature. Cable loses of the conducted measurement are about 2dB.

Tx power index	TX power [dBm]	Average current [mA]
0	13.97	260.15
1	12.59	255.95
2	11.62	249.5
3	11.53	251.17
4	10.57	189.35
5	9.47	184.4
6	8.93	182.3
7	8.96	182.3
8	8.89	182.27
9	8.88	182.22
10	8.81	182.29
11	8.86	182.2
12	8.88	182.17
13	8.89	182.18
14	8.92	182.2
15	8.93	182.11

Table 8: TX power vs current consumption, conducted measurement of continuous data transmission, rate 1Mbps (DSSS)

Tx power index	TX power [dBm]	Average current [mA]
0	11.74	119.74
1	10.48	118.95
2	9.46	118.36
3	8.36	117.91
4	8.87	103.10
5	8	102.29
6	6.80	101.73
7	5.83	101.29
8	4.93	100.84
9	3.93	100.59
10	2.88	100.30
11	1.98	100.18
12	1.09	100.02
13	0.75	100
14	0.73	100
15	0.64	100

Table 9: TX power vs current consumption, conducted measurement of continuous data transmission, rate 54Mbps (OFDM)

3. Pinout

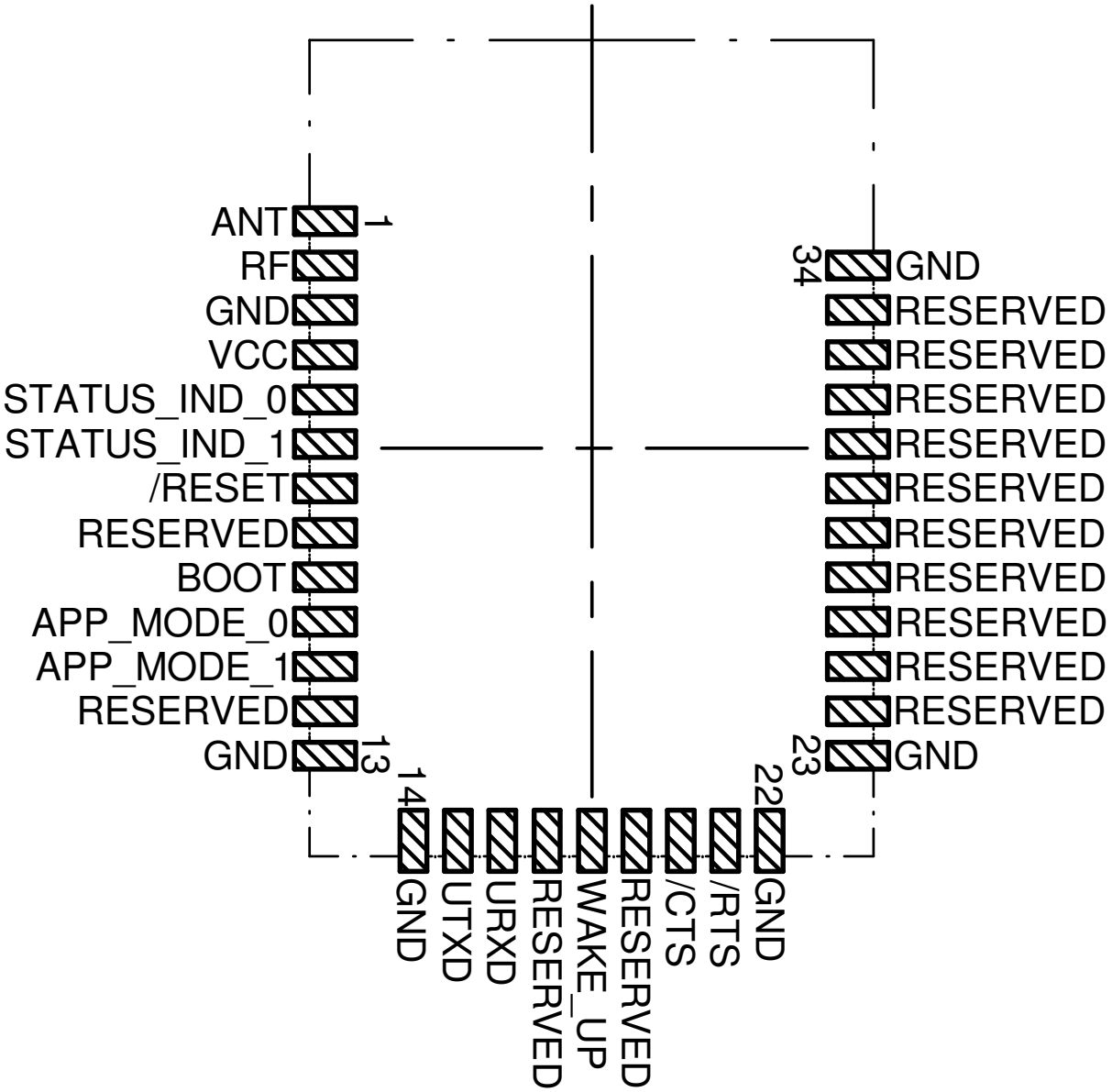


Figure 2: Pinout (top view)

No	Function	Description
1	*ANT*	RF connection to PCB antenna (see section 4.1)
2	*RF*	50Ω RF connection to external antenna or onboard antenna via ANT (see section 4.1)
3	*GND*	Negative supply voltage
4	*VCC*	Positive supply voltage
5	*STATUS_IND_0*	Status indication LED 0, do not connect if not needed
6	*STATUS_IND_1*	Status indication LED 1, do not connect if not needed
7	*/RESET*	Reset (active low), internal pull-up (100 kΩ)
8	*GPIO12*	Unused, output LOW, do not connect if not needed
9	*BOOT*	Input with internal pull-down (2.7 kΩ), pull low during start-up to boot the standard application, do not connect if not needed
10	*APP_MODE_0*	Input, internal weak pull-down (see section 5.2), do not connect if not needed
11	*APP_MODE_1*	Input, internal weak pull-down (see section 5.2), do not connect if not needed
12	*GPIO30*	Unused, output LOW, do not connect if not needed
13	*GND*	Negative supply voltage

No	Function	Description
14	*GND*	Negative supply voltage
15	*UTXD*	Module UART TX
16	*URXD*	Module UART RX, internal weak pull-up
17	*GPIO3*	Unused, output LOW, do not connect if not needed
18	*WAKE_UP*	Wake-up on rising edge, internal pull-down, do not connect if not needed
19	*GPIO5*	Unused, output LOW, do not connect if not needed
20	*GPIO6/CTS*	Optionally UART CTS (see section 6.2), internal pull-down, do not connect if not needed
21	*GPIO7/RTS*	Optionally UART RTS (see section 6.2), output LOW, do not connect if not needed
22	*GND*	Negative supply voltage

No	Function	Description
23	*GND*	Negative supply voltage
24	*GPIO10*	Unused, output LOW, do not connect if not needed
25	*GPIO11*	Unused, output LOW, do not connect if not needed
26	*GPIO14*	Unused, output LOW, do not connect if not needed
27	*GPIO15*	Unused, output LOW, do not connect if not needed
28	*GPIO16*	Unused, output LOW, do not connect if not needed
29	*GPIO17*	Unused, output LOW, do not connect if not needed
30	*JTAG_TDI*	Debug line (locked), do not connect
31	*JTAG_TDO*	Debug line (locked), do not connect
32	*JTAG_TCK*	Debug line (locked), internal pull-down, do not connect
33	*JTAG_TMS*	Debug line (locked), do not connect
34	*GND*	Negative supply voltage

Table 10: Pinout

4. Quick start guide

The Calypso WLAN module comes pre-flashed, tested and ready-to-use out-of-the-box. This chapter describes steps to quickly build a prototype system and test the capabilities of the module.

4.1. Antenna connection

Calypso 's smart antenna configuration enables the user to choose between two antenna options:

4.1.1. On-board PCB antenna

The Calypso has an on-board PCB antenna optimzed for operation in the 2.4 GHz band. A simple short between the pins *RF* and *ANT* feeds the RF output of the module to the on-board antenna. In this configuration, the module does not require any additional RF circuitry.

4.1.2. External antenna

For applications that use an external antenna, the Calypso provides a 50Ω RF signal on pin *RF* of the module. In this configuration, pin *ANT* of the module has to be connected to ground and pin *RF* to the external antenna via 50Ω feed line. Refer to chapter 16 for further information.

4.2. Minimal pin configuration

The following pins must be connected as described in table 11 for correct operation. The remaining can be left unconnceted.

Pin Number	Pin Function	Pin connection
1	*ANT*	Connect to pin 2 or *GND* (see 4.1)
2	*RF*	Connect to pin 1 or external antenna (see 4.1)
3,13,14,22,23,34	*GND*	*GND*
4	*VCC*	*VCC*
7	*/RESET*	Host GPIO and/or reset button
5,6	*STATUS_IND_x*	Optionally to Host GPIO for status indication
9	*BOOT*	Boot pin to host GPIO or GND
10,11	*APP_MODE_x*	Host GPIO for mode selection
15	*UTXD*	Host UART RX
16	*URXD*	Host UART TX
18	*WAKE_UP*	Host GPIO for wake up trigger

Table 11: Minimal pin configuration

4.3. Power up

Set and hold the */RESET* pin to LOW. After the supply voltage to the module has stabilized, the */RESET* pin shall be held LOW level for another Δt of at least 200 ms to ensure a safe start-up. Before releasing the */RESET* pin, make sure that the appropriate voltage levels are applied on pins *App_Mode_0* and *App_Mode_1* according to the desired application mode (see 5.2). Also make sure that the host's UART TX line to the module is configured as a logic HIGH level during module boot-up. The module will send a Start-up UART message once it has booted and started the application.
For further timing information refer chapter 11. If the module is used on a battery-powered system, using a suitable reset-IC (or a discrete RC block for a delay) is highly recommended to ensure a correct power up and stable behavior independent of battery status.
The typical time for Δt is in the order of 1 second. An additional 1.1 seconds (typical) delay is introduced on first boot due to extended calibration (e.g. after a firmware update). Upon hard-reset the firmware integrity check will add further latency into t_boot.

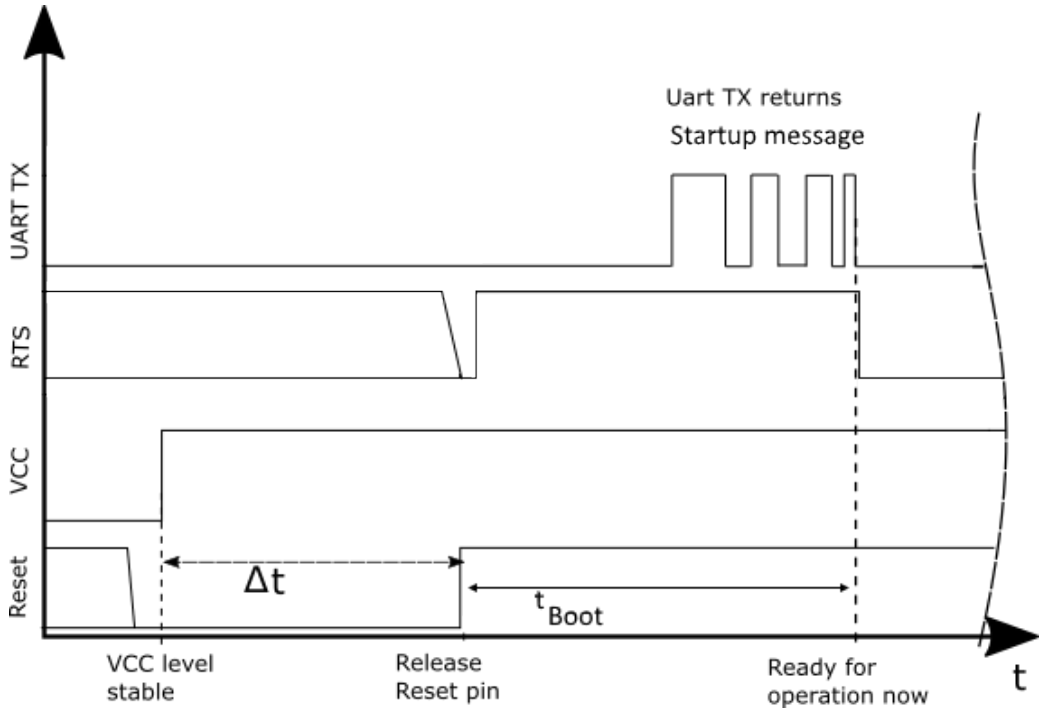


Figure 3: Power up

4.4. Region specific WLAN settings

Despite the world-wide availability of the 2.4 GHz frequency band, there are region specific restrictions on availability of certain channels. In order to be complaint with local regulations, the country code on the module has to be set-up before deployment. By default the country code is set to "US". Country code can be changed by sending the following command to the module. Refer to section 8.2.5 for more details. On request, the modules can be produced with the application-specific country code (see chapter 14).

```
AT+wlanset=general,country_code,EU
OK
```

4.5. Quick start example

This section is intended to help the user set-up a quick WLAN network consisting of an access point and two Calypso modules and exchange data between the two modules. Minimal pin and antenna connections have to be done on both the modules as described in sections 4.2 and section 4.1. It is recommended to use the Calypso evaluation kit for quick tests.

4.5.1. Prerequisites

The following hardware is required to go through the quick start example.

1. Two Calypso evaluation boards.
2. An IEEE 802.11b/g/n compatible access point working in the 2.4 GHz band.
3. Computer with a serial terminal emulator like Tera Term.

4.5.2. Hardware configuration

Make sure that the following jumpers are populated in the corresponding positions on the EV board. Refer to the Calypso evaluation board specific manual for a complete hardware description.

1. JP4 to select the USB bus as power supply.
2. JP3 current bridge is set.
3. Jumpers are set across pins 1-2 (*URXD*), 3-4 (*UTXD*), 9-10 (*STATUS_IND_0*), 11-12 (*STATUS_IND_1*), 13-14 (*WAKE_UP*) and 15-16 (*BOOT*) of the connector JP1.
4. For this example, the AT command terminal mode is used and hence the two *AP-P_MODE_x* pins shall be connected to *VCC*. On the Calypso EV board, this can be done by connecting pin 12 and 13 on Connector, CON8 to pin 6 or pin 10 of the same connector.

4.5.3. Setup description

In this example, the two Calypso modules will be connected to the access point and exchange a hello (Figure 4).

1. Set-up the access point in IEEE 802.11b/g/n 2.4 GHz infrastructure mode.
2. The access point's SSID and WPA/WPA2 key(if enabled) will be necessary for module setup
3. Make sure that a DHCP server is running on the access point or in the same network.
4. Connect the two EV boards to a computer with the serial terminal installed via the USB interface on the evaluation board.

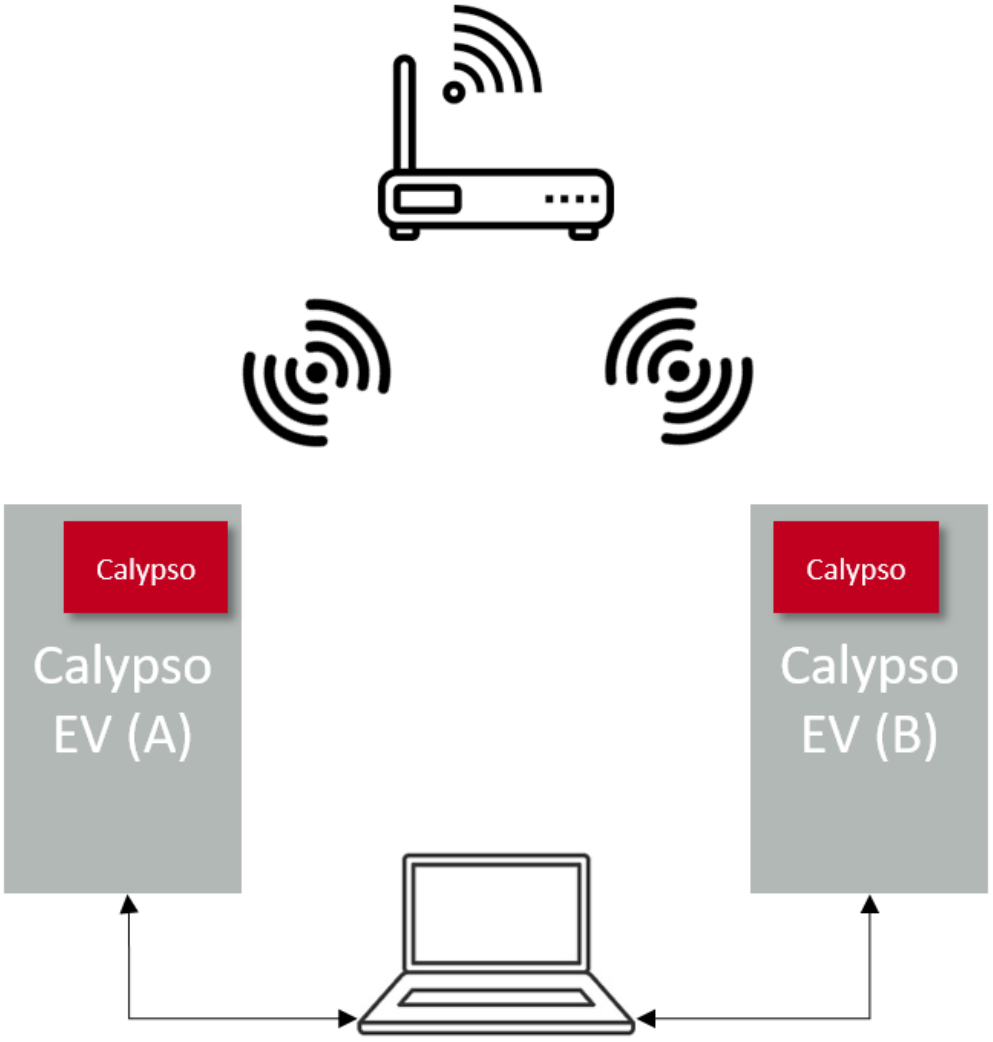


Figure 4: Quick start setup

4.5.4. Start-up

1. Connect the Calypso EV boards to the laptop/PC via USB.
2. The power LED indicates that supply voltage is active.

The FTDI driver for the converter IC on the evaluation board has to be installed and/or updated. On correct driver installation, the EV board appears as a virtual COM port.

3. Open an instance of the serial port emulator with COM port settings 921600 baud, 8e1 for each WLAN module connected to the PC via USB.
4. On pressing the Reset button, the start-up message appears on the terminal with the product article number, chipID, MAC address and the current software version.

```
+eventstartup:2610011025000,0x31000019,c8:fd:19:05:54:b4,1.0.0
```

4.5.5. Connect to an access point

1. In this example an access point with the following settings is used.

   SSID : WE_calypso

   Security method : WPA2_PSK

   Key : calypsowlan

2. Type in the following command into the terminal to connect to the access point.

```
AT+wlanconnect=WE_calypso,,WPA_WPA2,calypsowlan,,,
OK
+eventwlan:connect,WE_calypso,0x24:0xf5:0xa2:0x28:0x97:0x21
+eventnetapp:ipv4_acquired,192.168.1.168,192.168.1.1,192.168.1.1
```

3. The above log indicate a successful WLAN connect and subsequent IP acquisition. The WLAN connection process typically takes a few seconds to complete.

4. Repeat the process for module B and note the two different IP addresses assigned to the modules.

In the current example, the modules have the following addresses,

Module	MAC Address	IP Address	Role
A	0xc8:0xfd:0x19:0x05:0x54:0xb4	192.168.1.168	TCP server
B	0xc8:0xfd:0x19:0x05:0x74:0x98	192.168.1.140	TCP client

Table 12: Quick start addresses and roles

The MAC addresses are unique to every module and the IP address is as set by the DHCP server at the access point.

4.5.6. Creating a TCP server

The next step is to create a TCP server on module A.

1. Create a TCP socket using the following command. The modules returns a socket ID.

```
AT+socket=INET,STREAM,TCP
+socket:0
OK
```

2. Bind the TCP socket with the corresponding ID to the module IP and port:8888.

```
AT+bind=0,INET,8888,192.168.1.168
OK
```

3. Now the server listens for connection request on the specified port with the following command.

```
AT+listen=0,10
OK
```

4.5.7. Creating a TCP client

Module B should be configured as a TCP client

1. Create a TCP socket using the following command. The modules returns a socket ID.

```
AT+socket=INET,STREAM,TCP
+socket:0
OK
```

2. Initiate the connection to the server with a connect command with the correct server address and port.

```
AT+connect=0,INET,8888,192.168.1.168
OK
+connect:8888,192.168.1.168
OK
```

   On successful connection, a connect event is returned with the server address and port.

4.5.8. Data transfer

1. On the server side, the connection has to be accepted with a command,

```
AT+accept=0,INET
OK
+accept:1,inet,60108,192.168.1.140
OK
```

   The accept command returns the port and the IP address of the current client as well as the new socket ID generated for communication with this client.

2. At this stage, the modules are ready to exchange data. Here is an example of sending "hello" from A to B.

```
AT+send=1,0,5,hello
OK
```

3. At B the data is received as follows.

```
AT+recv=0,0,5
OK
+recv:0,0,5,hello
OK
```

4. Sending "hello" from B

```
AT+send=0,0,5,hello
OK
```

5. Receiving the message at A

```
AT+recv=1,0,5
OK
+recv:1,0,5,hello
OK
```

5. Functional description

The Calypso WLAN module is intended to be used as a radio sub-system in order to provide WLAN (IEEE 802.11) communication capabilities to system.
The UART acts as the primary interface between the module and a host micro-controller. The module can be fully configured and operated using a set of AT-commands over UART. Once configured, the module independently manages WLAN connectivity allowing the host controller to utilize its resources elsewhere.
As a standalone WLAN radio module running a fully featured TCP/IP stack, Calypso can be configured to operate in several modes at several layers of the protocol stack.

5.1. Key features

In this section, the features of the Calypso module is summarized in the form of a table. Calypso offers the user to configure and exploit its rich features through an easy-to-use command interface over UART.

Feature	Description
Radio standards	IEEE 802.11 b/g/n station IEEE 802.11 b/g Access point (for provisioning only) Wi-Fi Direct client and group owner
Channels	1-13
Security	WEP, WPA/WPA2PSK, WPA2 Enterprise (802.1x)
Provisioning	In AP mode using the on-board HTTPS server
Network layer	IPv4, IPv6
IP addresssing	Static, LLA, DHCPv4, DHCPv6 with DAD
Transport layer	TCP, UDP SSLv3.0/TLSv1.0/TLSv1.1/TLSv1.2
Network applications	SNTP client HTTTP(S) server mDNS, DNS-SD DHCP server Ping
Update	Secure OTA update with fall back mechanism
Security	Secure key storage Trusted root-certificate catalog Encrypted file system Secure OTA Software tamper detection Cloning protection
Power management	802.11 power save power modes Lower power sleep mode with timed or pin wake-up

Table 13: Key features

5.2. Modes of operation

When active, the Calypso can be in one of the following operation modes. The transition to/from the modes occurs due to one of the following reasons.

- Command from the host.
- Position of the *App_Mode_x* pins during boot up
- */Reset* signal
- *WAKE_UP* signal or time event

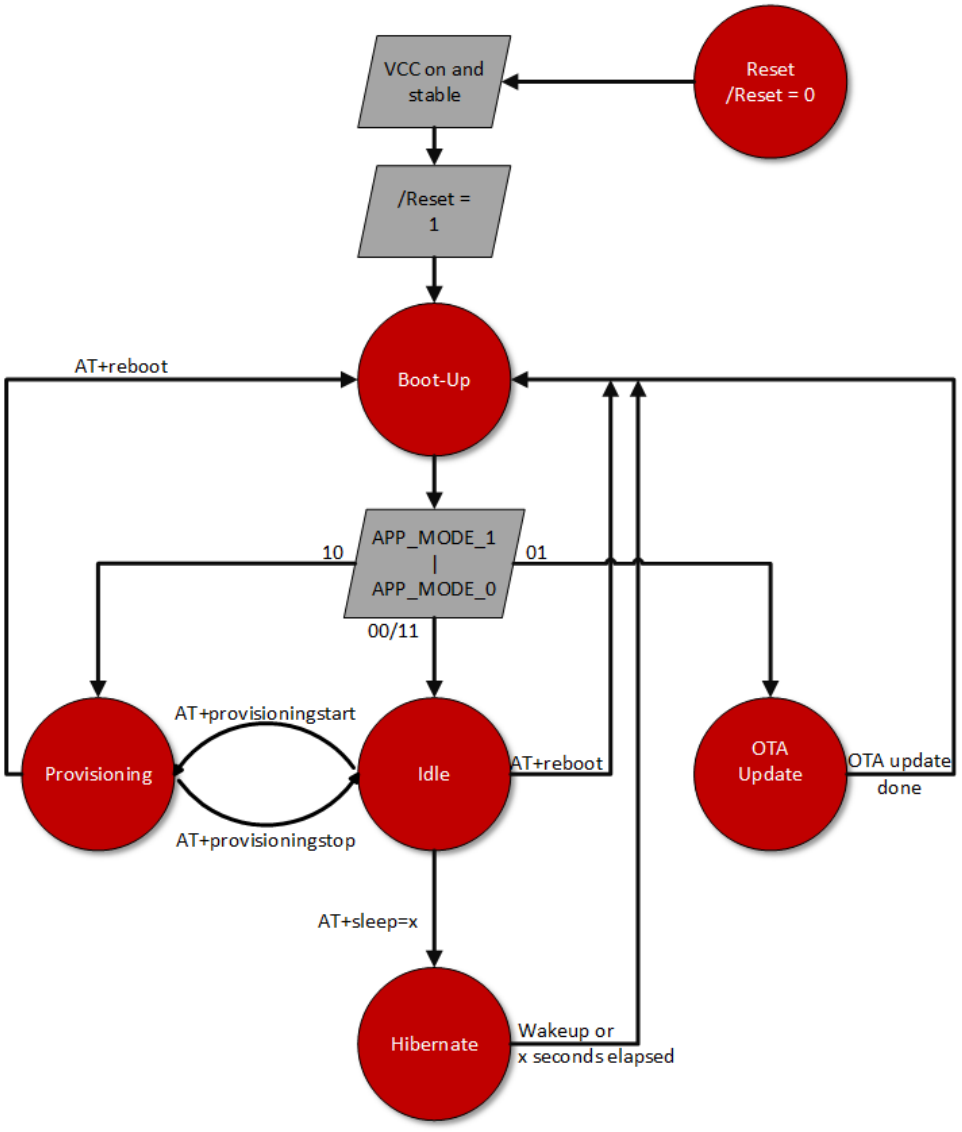


Figure 5: Modes of operation

5.2.1. BootUp

Based on the voltage level on the pins *App_Mode_0* and *App_Mode_1*, the module boots up in one of the following modes.

APP_MODE_1	*APP_MODE_0*	Description
0	0	AT command normal mode, see chapter 7
0	1	OTA mode, see chapter 12
1	0	Provisioning mode, see chapter 9
1	1	AT command terminal mode, see chapter 7

Table 14: Application modes

A 0 indicates logic LOW level, 1 indicates logical HIGH level.

5.2.2. Idle

In idle mode, Calypso allows the user to configure and use the module using the UART command interface. The AT-command interface is described in detail in chapter 7. A transition to provisioning or hibernate can be done using the appropriate commands.

5.2.3. OTA update

In this mode of operation, the module allows secure over the air firmware update to be carried out from a device (PC/Mobile) present in the same wireless network (local OTA update). Further details regarding the OTA update mechanism can be found in chapter 12.

5.2.4. Provisioning

To enable easy provisioning when integrated to an embedded system with limited HMI capabilities, the Calypso offers a provisioning mode. In this mode, the module acts as an AP and allows external devices with appropriate credentials to connect and access the on-board HTTPS server. The user can conveniently browse the settings web-page and configure the module using any web-browser. More details towards provisioning in chapter 9.

5.2.5. Hibernate

It is essential to have a low power sleep mode especially for a battery powered systems. Calypso offer a hibernate mode with a very low current consumption of less than 10 μA. Section 8.1.5 describes the commands used to put the module to hibernate and chapter 11 describes the timing characteristics. On wake-up, the module starts from the reset vector meaning that the RAM contents are lost. Based on the WiFi connection policy(see 8.2.6), the module can be set up to automatically connect to a saved access point profile and acquire IP address. However, the socket connections are lost on entering the sleep mode and have to be re-establish on wake-up.

6. Host connection

The Calypso is intended to be used as a radio module in a system, interfaced with a host micro-controller. The use of industry standard UART as the primary interface ensures a very minimal requirement set on the host MCU. As a result of this, the module can be designed in with most host controllers from a 8051 to the more advanced ARM core architecture.

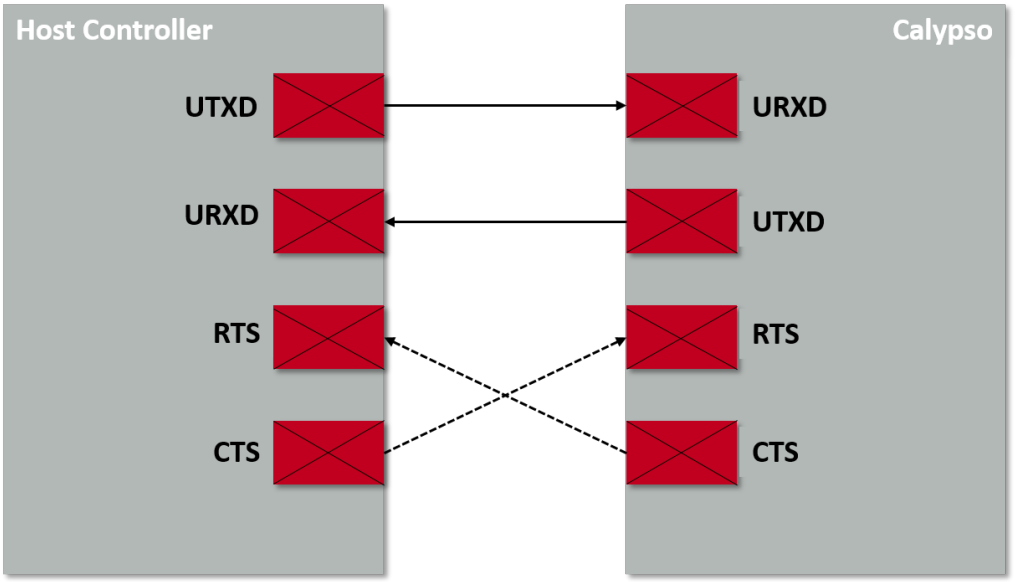


Figure 6: Host Interface

6.1. UART parameters

The Calypso implements the standard UART interface with the following parameters.

Parameter	Range	Standard
Baud	115200 to 3000000	921600
Data bits	8	8
Stop bits	1	1
Parity	none, odd, even	even
Flow control	none, RTS/CTS	none

Table 15: Application modes

The configuration of the UART in factory state is 921600 baud with data format of 8 data bits, even parity and 1 stop bit ("8e1"). The baud rate and the parity of the UART can be configured using corresponding commands (see section 8.1). The data format is fixed to 8 data bits and one stop bit. This results in a 11 UART symbols per 8 bit user data ratio.

6.2. Hardware flow control

The hardware flow control is disabled in the standard firmware which leads to a restriction on the UART data rate to 921600 Baud. The UART flow control can only be enabled during firmware creation (i.e. at compile time). A change of the flow control mode during runtime is currently not possible.
With the hardware flow control enabled on request (see chapter 14) through */RTS* and */CTS*, the UART baudrate can be increased up to 3 Mbps.

6.3. Timing and characteristics

The output of characters on the serial interface runs with secondary priority. For this reason, short interruptions may occur between the outputs of successive bytes. The host must not implement a strict timeout between two bytes to be able to receive packets that have interruptions in between. Up to four full byte durations (32 bit) delay between two successive bytes shall be accepted by the host.

For the direction "host to module" the host must respect byte-wise the line */RTS*, which will indicate that the next byte of the packet can be received by the module. This direction also accepts a pause of up to four full byte durations (32 bit) delay between two successive bytes before discarding received content (without user notification).

7. The command interface

The command interface on the Calypso enables full control over the module using ASCII based AT styled commands. In order to support easy integration with a wide range of micro-controllers, two different modes of the command interface are implemented. The user can choose one of the following modes by setting the two *APP_MODE_x* pins to the corresponding levels as described in section 5.2.

1. Terminal mode : In this mode the module behaves like a standard serial terminal. All characters received are looped back to the host and an action is triggered on receiving a "\r\n" (hex: 0x0D0A). The module supports backspace character "\b" in this mode. All user data has to be Base64 encoded as binary data transmission is not allowed in this mode of operation.

2. Normal mode : This is the standard mode without terminal characteristics. The received characters are not echoed. However, the host receives an explicit acknowledgement for every command it receives and triggers an action on receiving a "\r\n" (hex: 0x0D0A). In this mode, the user data can be binary as well as Base64.

7.1. Command types

There are three types of messages exchanged between the calypso and the host.

- Requests: The host requests the module to perform an action or start an operation. All Requests start with the "AT+" tag and end with "\r\n" (hex: 0x0D0A).
- Confirmations: On each request, the module answers with a confirmation message to give a feedback on the requested operation status.
  All Confirms contain the Request itself and either a "OK" or an error code. Appendix B gives a brief description of all the error codes. All confirmations end with "\r\n" (hex: 0x0D0A).
- Events: The module indicates spontaneously when a special event has occurred. All events start with the "+" tag and contain further data, error codes (see Appendix B) or status information. All events end with "\r\n" (hex: 0x0D0A).

7.2. AT command characteristics

This section describes the syntax and detailed characteristics of the aforementioned three command types.

7.2.1. Request

The generic syntax of an AT command request is as shown below :

```
AT+<command name> = <param1>, <param2>, ..., <paramX>
```

- All commands start with the prefix "AT". The delimiter "+" indicates the beginning of the command name.

- Commands can have parameters in which case the delimiter "=" separates the command name from the list of parameters.
- AT commands can be entered in upper or lower case with optional whitespace between the arguments.
- Further, each parameter is separated from the next with a "," delimiter. A comma shall not be followed by a whitespace.
- In cases where a parameter is optional or ignored, it could be left empty. Nevertheless the "," delimiter has to be present. An empty parameter looks like „,".
- String parameters containing spaces must be enclosed with quotation marks ("").
- All hexadecimal parameters must have a 0x prefix.
- MAC and network addresses must be entered as follows
  - MAC address - Six hexadecimal values of 8 bit each, represented as X:X:X:X:X:X (X can range from 0x00 up to 0xFF), the ":" is used as delimiter
  - IPv4 address - Four decimal values of 8 bit each, represented as X.X.X.X (X can range from 0 up to 255), the "." is used as delimiter per 8 bit
  - IPv6 address - Four hexadecimal numeric values of 32 bit each, represented as X:X:X:X (X can range from 0x00000000 up to 0xFFFFFFFF), the ":" is used as delimiter per 32 bit
- Bitmask parameters are represented using "|" delimiter - e.g. x|y
- Data should be either binary or Base64 format (binary to text encoding).

7.2.2. Confirmations

The commands confirms have the following syntax,

```
<command name>:<value1>, <value2>, ..., <valueX>
```

On success, the confirms contain a positive acknowledgement.

```
OK
```

In case of error, the corresponding error code is with optional description is returned.

```
ERROR:<error description>, <error code>
```

7.2.3. Events

Asynchronous events can arrive at any time and can be represented by

```
+<event name>:<value1>, <value2>, ..., <valueX>
```

8. AT commands

In this chapter, various commands used to configure and control the Calypso module are described.

8.1. Device commands

The commands under device category provide access to generic module properties like communication interface, time and date settings and version information. Additionally, basic device operations like start, stop, reboot and sleep are described in this section.

8.1.1. Start and stop commands

The start and stop commands control the state of the 802.11 network processor unit (NWP). On boot up the network processor is started by default. A stop command puts the network processor to hibernate effectively switching off the radio resulting in loss of all on-going transmissions and connections. A time-out can be specified to allow the network processor to gracefully disconnect before shutting down.

Request	Response
AT+start	OK or error code
Arguments: None	Arguments: None

Table 16: AT+start

Request	Response
AT+stop=[timeout]	OK or error code
Arguments: Timeout: in milliseconds • 0 - Hibernate immediately without waiting for a response from the NWP. • 0xFFFF - Wait indefinitely for a response from the NWP. • 0 < time-out < 0xFFFF - Wait for timeout before forcing to hibernate.	Arguments: None

Table 17: AT+stop

8.1.2. Test

This command provides a simple way of ensuring that the module is active and ready to receive further commands.

Request	Response
AT+test	OK or error code
Arguments: None	Arguments: None

Table 18: AT+test

8.1.3. Reboot

This command performs a software reset on the module. The module internally puts the NWP to hibernate before restarting from the reset vector.

Request	Response
AT+reboot	OK or error code
Arguments: None	Arguments: None

Table 19: AT+reboot

It is recommended to use this command whenever possible instead of hard reset (a falling edge on the */Reset* pin).

8.1.4. Factory reset

The factory reset command restores the module to factory state.

- All files stored in the file system will be reverted to factory state.
- New files that were added will be deleted.
- The network processor settings including MAC address will be restored to factory state.

Request	Response
AT+factoryreset	OK or error code
Arguments: None	Arguments: None

Table 20: AT+factoryreset

Factory reset operation can take up to 30 seconds to complete. The module responds with an "OK" only after this time period.

Resetting or powering off the module during this operation can result in permanent damage to the module.

A reset is performed automatically after the restore operation.

8.1.5. Sleep

The sleep command puts the module into the lowest possible power mode (hibernate) resulting in a current consumption of less than 10μA. In the hibernate mode, the network processor is in hibernate mode and the application processor is shut down. The module wakes up automatically after a time period specified in the sleep command. Alternatively, the module can be woken up manually with a rising edge on the *WAKE_UP* pin. On wake up, the module starts from the reset vector.

Request	Response
AT+sleep=[timeout]	OK or error code
Arguments: Timeout: in seconds, min 1, max 86400(24 hrs)	Arguments: None

Table 21: AT+sleep

8.1.6. Get

The generic get command can be used to read the device parameters including version, time, UDID and UART settings. A few points to be noted,

- The status bit mask is cleared once it is read. Additionally, status information is also available as events (see 7).
- The system persistent setting is enabled by default. This means that all the settings are retained after reset.

Request		Response
AT+get=[ID],[option]		+Get:[value1]...[valueX] (or error) OK
Arguments:		Arguments:
ID	**option**	**Return values**
status	device	value1 : error bitmask
	WLAN	value1 : bitmask - WLANASYNCONNECTEDRESPONSE - WLANASYNCDISCONNECTEDRESPONSE - STA_CONNECTED - STA_DISCONNECTED - P2P_DEV_FOUND - CONNECTION_FAILED - P2P_NEG_REQ_RECEIVED - RX_FILTERS - WLAN_STA_CONNECTED
	BSD	value1 : bitmask - TX_FAILED
	netapp	value1 : bitmask - IPACQUIRED - IPACQUIRED_V6 - IP_LEASED - IP_RELEASED - IPV4_LOST - IP_COLLISION - IPV6_LOST
general fprov	version	chip ID, FW version (X.X.X.X), PHY version (X.X.X.X), NWP Version (X.X.X.X), ROM version
	time	hh,mm,ss,dd,mm,yyyy
	persistent	1=enable, 0=disable
IOT	UDID	16 byte UDID
UART	baudrate	baudrate
	parity	0=none, 1=even, 2=odd

Table 22: AT+get

8.1.7. Set

The generic get command can be used to set the device parameters like time, persistence and UART settings.

Request		Response
AT+set=[ID],[option], [value1],...[valueX]		OK (or error)
Arguments:		
ID	**option**	**values**
general	persistent	1=enable, 0=disable
	time	hh,mm,ss,dd,mm,yyyy
UART	baudrate	baudrate (see 6.1)
	parity	0=none, 1=even, 2=odd

Table 23: AT+set

8.2. WLAN commands

In this section, all the commands necessary to configure the WLAN settings of the module are described.

8.2.1. Set mode

The Calypso can be operated as a WLAN station, access point or in P2P (Wi-Fi direct) mode. The mode can be selected using the following command and the configuration will take effect only after a stop/start of the NWP.

The AP mode is primarily intended for device provisioning and can support up to 4 stations.

Inherently the AP mode consumes higher currents and is therefore not suitable for battery powered applications.

Request	Response
AT+wlanSetMode=[mode]	OK or error code
Arguments: - STA : for station mode - AP : for access point mode - P2P : for P2P mode	

Table 24: AT+wlanSetMode

8.2.2. Scan

The scan function enables the user to perform a scan and discover devices on all the enabled channels. The module returns a list of up to 30 devices.

The first scan command initiates a scan and hence returns an error code S-L_ERROR_WLAN_GET_NETWORK_LIST_EAGAIN (-2073). A further scan command returns the list of available access points

Request	Response
AT+wlanScan=[index],[count]	+wlanscan:<Device[index]> ... or error code
Arguments: Index: starting index 0-29 count: number of device max 30	Arguments: Each device has the following parameters listed SSID, BSSID, Channel, Security type, hidden_ssid_enanled (0 or 1), cipher, key_management_method

Table 25: AT+wlanscan

8.2.3. Manual connection

In order to manually connect the Calypso to a known access point the following command has to be used. A manual connect has the highest priority over all the other connection types. A connect event confirms a successful connection.

Request	Response
AT+wlanConnect=[SSID], [BSSID], [SecurityType], [SecurityKey], [SecurityExtUser], [SecurityExtAnonUser], [SecurityExtEapMethod]	OK or error code
Arguments: - SSID : Name of the AP - BSSID : MAC address of the AP (optional) - SecurityType: OPEN, WEP, WEP_SHARED ,WPA_WPA2 , WPA_ENT ,WPS_PBC, WPS_PIN - SecurityKey : password (optional if not used) - SecurityExtUser: Enterprise user name parameters (Ignored in case WPA_ENT was not selected) - SecurityExtAnonUser: Enterprise anonymous user name parameters (Ignored in case WPA_ENT was not selected) - SecurityExtEapMethod: Extensible Authentication Protocol (Ignored in case WPA_ENT was not selected): TLS, TTLS_TLS, TTLS_MSCHAPv2, TTLS_PSK, PEAP0_TLS, PEAP0_MSCHAPv2, PEAP0_PSK, PEAP1_TLS, PEAP1_PSK	

Table 26: AT+wlanConnect

A manual disconnect from an existing connection is done using the following command.

Request	Response
AT+wlanDisconnect	OK or error code

Table 27: AT+wlanDisconnect

8.2.4. Profiles

Calypso allows the user to store up to seven preferred networks as profiles. Based on the connection policy (see section 8.2.6) the module automatically establishes connection with one of the saved profiles. Profile priority determines the order of connection. The profiles are saved in the non-volatile memory and can be added, read or deleted using the following commands.

Request	Response
AT+wlanProfileAdd=[SSID], [BSSID], [SecurityType], [SecurityKey], [SecurityExtUser], [SecurityExtAnonUser], [SecurityExtEapMethod],[priority]	+wlanProfileAdd: <Profile index> OK or error code
Arguments: - SSID : Name of the AP - BSSID : MAC address of the AP (optional) - SecurityType: OPEN, WEP, WEP_SHARED ,WPA_WPA2 , WPA_ENT ,WPS_PBC, WPS_PIN - SecurityKey : password (optional if not used) - SecurityExtUser: Enterprise user name parameters (Ignored in case WPA_ENT was not selected) - SecurityExtAnonUser: Enterprise anonymous user name parameters (Ignored in case WPA_ENT was not selected) - SecurityExtEapMethod: Extensible Authentication Protocol (Ignored in case WPA_ENT was not selected): TLS, TTLS_TLS, TTLS_MSCHAPv2, TTLS_PSK, PEAP0_TLS, PEAP0_MSCHAPv2, PEAP0_PSK, PEAP1_TLS, PEAP1_PSK - Profile priority (0 - 15(highest))	

Table 28: AT+wlanProfileAdd

Request	Response
AT+wlanProfileGet=[index]	+wlanProfileGet:<Profile[index]> ... or error code
Arguments: Index: profile index 0- 6	Arguments: SSID, BSSID, Channel, Security type, hidden_ssid_enanled (0 or 1), cipher, key_management_method, priority

Table 29: AT+wlanProfileGet

Request	Response
AT+wlanProfileDel=[index]	OK or error code
Arguments: Index: profile index 0- 6	

Table 30: AT+wlanProfileDel

8.2.5. WLAN settings

In this section commands to read and modify the WLAN modes in different modes are described. All the WLAN settings are non-volatile.

Request			Response
AT+wlanSet=[ID],[option],[value1],...[valueX]			OK (or error) OK
ID	**option**	**value**	
general	COUNTRY_CODE	US,EU or JP	
	STA_TX_POWER	0-15 (0 = Max transmit power)	
	AP_TX_POWER	0-15 (0 = Max transmit power)	
	SCAN_PARAMS	value1: channel mask, value2: RSSI threshold	
	SUSPEND_PROFILES	Suspend profile bit mask	
	DISABLE_ENT_SERVER_AUTH	0 or 1 (1 = disable server auth when manually connecting to an enterprise network)	
P2P	CHANNEL_N_REGS	value1:Listen channel (1/6/11), value2:Listen regulatory class (81), value3:operating channel (1/6/11), value4:operating regulatory class (81)	
AP	SSID	String up to 32 charecters	
	CHANNEL	Channels 1-11	
	HIDDEN_SSID	0:disabled, 1:send empty	
	SECURITY	open, WEP, WPA_WPA2	
	PASSWORD	WEP:8-63 charecters, WPA:5-13 charecters	
	MAX_STATIONS	1-4	

Table 31: AT+wlanSet

Request		Response
AT+wlanGet=[ID],[option]		+Get:[value1]...[valueX] (or error) OK
Arguments:		Arguments: see table 31
ID	**option**	
general	COUNTRY_CODE	
	STA_TX_POWER	
	SCAN_PARAMS	
P2P	CHANNEL_N_REGS	
Connection		Role, Status, security, SSID, BSSID, device name
AP	SSID	
	CHANNEL	
	HIDDEN_SSID	
	SECURITY	
	PASSWORD	
	MAX_STATIONS	
	MAX_STA_AGING	

Table 32: AT+wlanGet

8.2.6. WLAN policy

This set of commands allow changes in behaviour of the Calypso with respect to connection, power consumption, scan as well as P2P connections.

- **Connection**: This policy defines how the device initiates and maintains a specific connection after reset. The user can use one of the following options,

  Auto - The device automatically tries to connect to the stored profiles based on priority. In case of several profiles with same priority, the decision is made based on security type (WPA2>WEP>OPEN). In case of same security type, the one with the highest signal strength is chosen to be connected.

  Fast: The device tries to connect to the last connected AP without transmitting a probe request.

  AnyP2P - The device connect to the first available Wi-Fi direct device.

- **Scan**: In additional to the one-shot scan, calypso can be configured to perform periodic scan with a specific scan period.

- **Power management** : Based on the application, the power management policy of the WLAN NWP can be set to one of the following options: Normal, low latency, low power and long sleep.

- **P2P**: In P2P mode, the Calypso can be configured to either choose a specific role (GO or client) or negotiate with the peer. The connections initiation can be active or passive based on the policy set.

Request			Response
AT+wlanPolicySet=[ID],[option],[value]			OK (or error)
ID	**option**	**value**	
connection	Auto,Fast or P2P		
scan	Hidden_SSID	scan interval in seconds	
	Disable_Scan		
PM	Normal, low latency, low power or long sleep	Maximum sleep time in ms only for long sleep option	
P2P	CLIENT, GROUP_OWNER, NEGOTIATE	ACTIVE, PASSIVE, RAND_BACKOFF	

Table 33: AT+wlanPolicySet

Request	Response
AT+wlanPolicyGet= [Type]	+wlanPolicyGet:[option],[value] or error code
Arguments: connection, scan, PM or P2P	Arguments: (see table 33)

Table 34: AT+wlanPolicyGet

8.3. Network configuration commands

Configuration at the network level involves address management. The Calypso supports multiple address-acquisition methods for both IPv4 and IPv6 addressing. In Station and Wi-Fi direct client mode, the address acquisition process begins after a successful WLAN connection is established. AP and Wi-Fi direct modes start with a static address assigned to the module with a DHCP server available on-board.

- **IPv4 Stateful with Stateless fallback** : In this mode, the device waits for an IPv4 address from a DHCP server. On time-out, the LLA address is used. The LLA IP addressess are in the range 169.254.1.0 to169.254.254.255.
- **Stateful(DHCPv4) only** : Wait for DHCPv4 server to assign an IP address without time-out.
- **Static**: Addressed configured by the user.
- **IPv6 SLAAC**: The least significant 64 bits are filled with the device MAC address in EUI-64 format. In case of duplicate address (DAD failure), random 64 bits are used.
- **IPv6 Stateful (DHCPv6)** : IPv6 LLA is acquired from a DHCPv6 server. In case for DAD failure, Stateless configuration is used.
- **Static**: Preconfigured by the user. In case of DAD failure, a failure event is sent to the host.

- **Link-Global IPv6** : The IPv6 global address can be acquired similar to the LLA stateless(MSB 64 bits from RA messages),statefull or static.

IPv6 LLA must have a prefix - Fe80::/64

IPv6 global address have a prefix - 2000::/3

Due to its inherent properties, it is recommended not to enable IPv6 addressing in power critical applications.

	Wi-Fi Station	Wi-Fi AP	Wi-Fi Direct
IPv4	Always enabled	static	client - like station GO - like AP
	one address-DHCP,LLA,Static		
IPv6	disabled	not supported	not supported
	two addresses-Local,Stateless, Statefull,Static		
	Global - Stateless, Statefull, Static		

Table 35: IP addresses

Request			Response
AT+NetCfgSet=[ID],[option],[value1],...[valueX]			OK (or error)
ID	**option**	**value**	
IF	STATE (Enable/disable) bitmask	IPV6_STA_LOCAL IPV6_STA_GLOBAL DISABLE_IPV4_DHCP IPV6_LOCAL_STATIC IPV6_LOCAL_STATELESS IPV6_LOCAL_STATEFUL IPV6_GLOBAL_STATIC IPV6_GLOBAL_STATEFUL DISABLE_IPV4_LLA ENABLE_DHCP_RELEASE IPV6_GLOBAL_STATELESS DISABLE_FAST_RENEW	
SET_MAC_ADDR		MAC Address	
IPV4_STA_ADDR	STATIC DHCP_LLA RELEASE_IP_OFF RELEASE_IP_SET DHCP	For static only value1 : IP address value2 : subnet mask value3 : Default gateway value4 : DNS	
IPV6_ADDR_LOCAL	STATIC	IP address	
	STATELESS STATEFUL		
IPV6_ADDR_GLOBAL	STATIC STATELESS STATEFUL	value1: IP address value2 :DNS IP	
IPV4_DNS_CLIENT		Secondary DNS	

Table 36: AT+NetCfgSet

Request	Response
AT+netCfgGet= [configID]	+netCfgGet:[option],[value1],...[valueX]
Arguments:	Arguments:
GET_MAC_ADDR	Mac address
IPV4_STA_ADDR or IP_AP_ADDR	Method, IP Address, Subnet mask, Gateway, DNS
IPV6_ADDR_LOCAL or IPV6_ADDR_GLOBAL	Method,IP address
IP_DNS_CLIENT	Secondary DNS address

Table 37: AT+netCfgGet

8.4. Socket commands

Communication between peers in a network is done using sockets. Calypso complies with the industry standard BSD sockets which provides IP based connection interface for data transfer. In this section, all the commands necessary to utilize the socket features are described.

8.4.1. Sockets work flow

At the transport layer, connection between peers can be of two types :

- Connectionless socket : Also know as Datagram sockets, this allows data exchange between network entities without establishing a connection. This results in minimal connection latency but cannot ensure data integrity or packet order.
- Connection-oriented socket : Stream sockets establish a connection between two entities before data exchange there by ensuring data integrity and packet order.

8.4.1.1. TCP socket

A TCP socket, a connection-oriented socket, creates a bi-directional connection between the two network peers, a client and a server. Calypso supports both client and server roles. Here is a general work flow of a TCP socket (see figure 7).

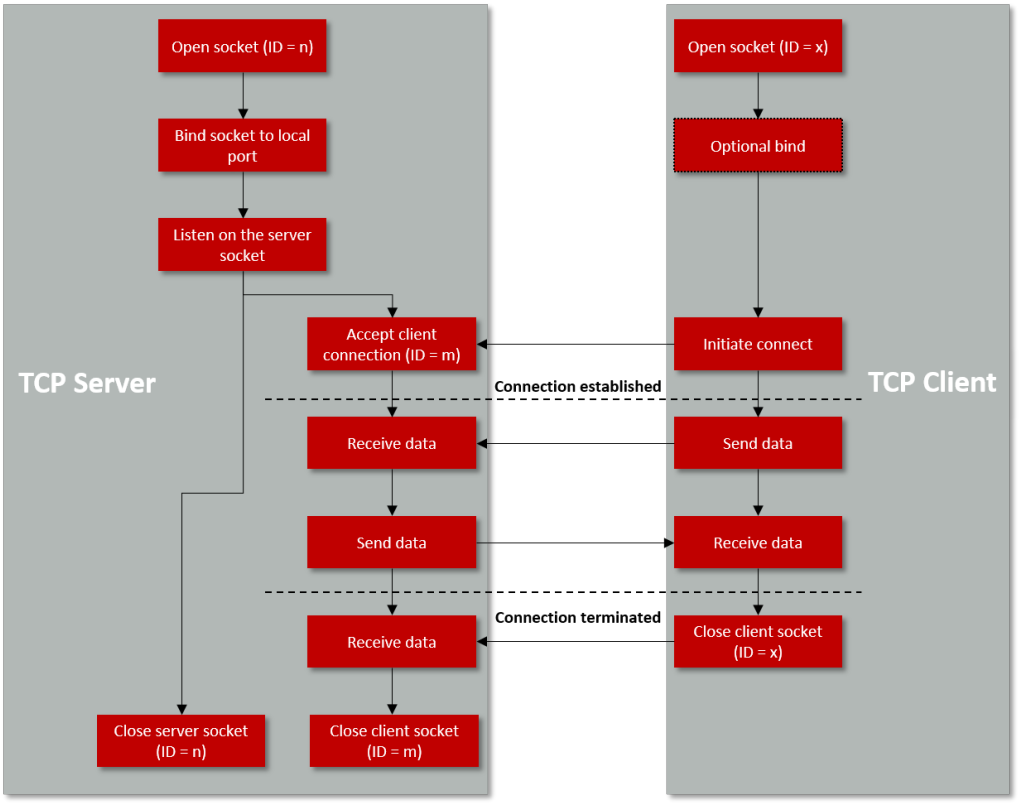


Figure 7: TCP socket work flow

8.4.1.2. UDP socket

UDP does not require a connection to exchange data among network peers. UDP does not have client and server as either can initiate communication by sending a packet with the corresponding destination address (see figure 8). Calypso supports a connection-oriented UDP mode where a client drops all the datagrams except the ones from the connected server. In this case the client work flow is similar to TCP (see figure 7).

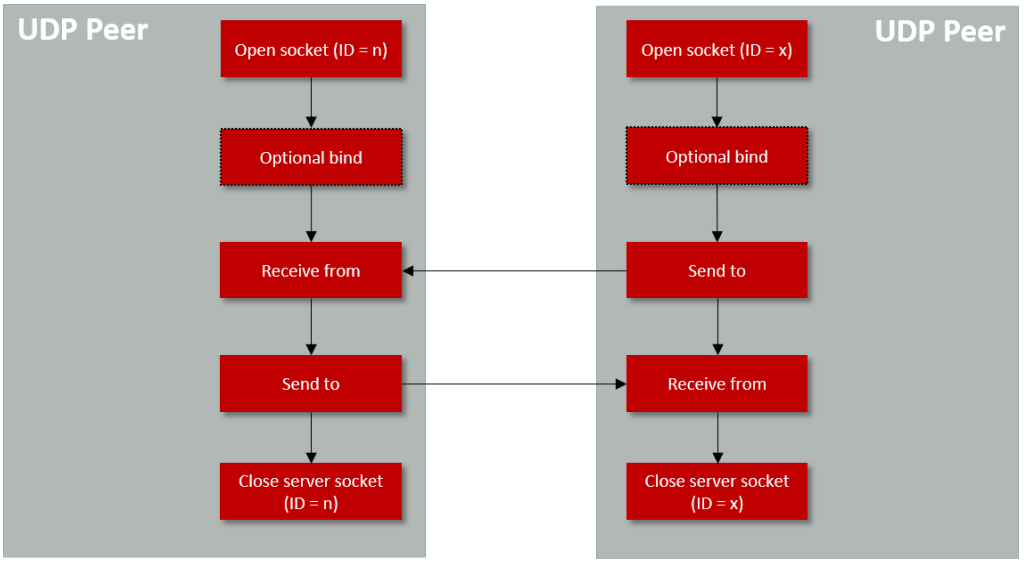


Figure 8: UCP socket work flow

8.4.1.3. Multicast

The Calypso also supports multicast (one-to-many) over the IP network. Ipv4 IGMPv2 and IPv6 MLDv1 protocols for joining or leaving a multicast group are supported

8.4.2. Secure sockets

Calypso supports secure socket communication using the SSL and TLS protocols. SSL/TLS protocols provides features like end-to-end encryption and authentication to ensure secure communication between network peers. A sequence of messages are exchanged between a TCP client and server leading to mutual authentication and encryption of data messages. The TLS/SSL handshake is summarized in the figure 9. The SSL/TLS processes are handled in an separate execution environment and hardware acceleration is used to speed up the cryptographic operations.

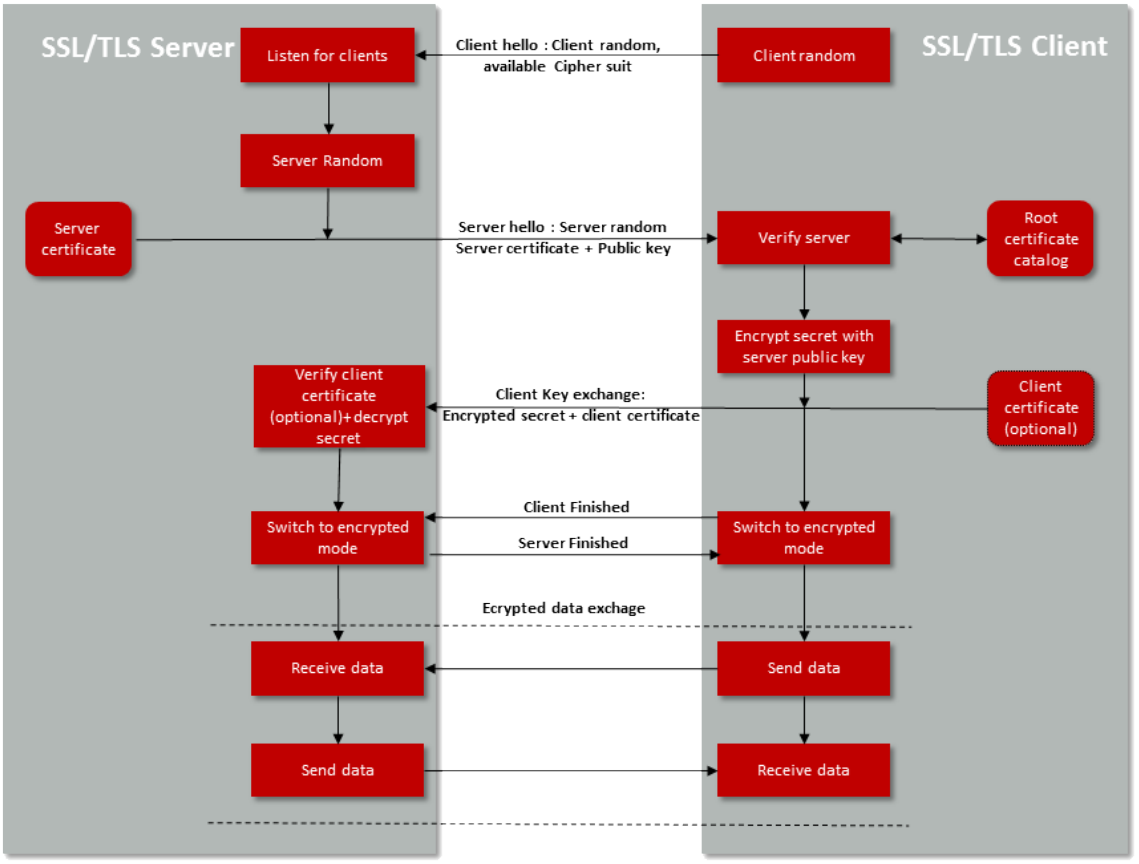


Figure 9: SSL/TLS handshake

The SSL/TLS protocol requires certificates for authentication and a trusted root certificate catalog to verify the certificates. The calypso provides secure key storage option through the encrypted file system (see section 8.5). A trusted root certificate catalog is present on board with a set of well known trusted root CAs (see appendix C).

8.4.3. Socket operations

In this section, the AT commands used to perform various operations on a socket is described. A socket can be created using the command `AT+socket` and the socket descriptor (socketID) returned by this command can be used to perform all the other socket operations. Socket select command allows monitoring multiple sockets and triggering on specific events.

Request	Response
AT+socket=[family],[type],[protocol]	+socket:[socketID] or error
Arguments: - family : INET or INET6 - Type : STREAM or DGRAM - protocol : TCP, UDP or SEC	

Table 38: AT+socket (create a socket)

Request	Response
AT+close=[socketID]	+close:[socketID] or error
Arguments: socketID :socket descriptor	

Table 39: AT+close (close a socket)

Request	Response
AT+bind=[socketID],[family],[localPort],[localAddress]	OK or error
Arguments: socketID: socket descriptor - family : INET or INET6 - localPort : Local port - localAddress: Local IP address	

Table 40: AT+bind

Request	Response
AT+listen=[socketID],[backlog]	OK or error
Arguments: socketID : socket descriptor backlog : max length of connect request queue	

Table 41: AT+listen

Request	Response
AT+connect=[socketID], [family], [remotePort], [remoteAddress]	+connect:[remotePort], [remoteAddress] or error
Arguments: socketID : socket descriptor family : INET or INET6 remotePort : Port of the peer to connect to remoteAddress : Address to connect to	

Table 42: AT+connect

Request	Response
AT+accept=[socketID],[family]	+accept:[clientSocketID],[family],[clientPort], [clientAddress] or error
Arguments: socketID:socket descriptor family : INET or INET6	

Table 43: AT+accept

Request	Response
AT+select=[nfds],[readsds],[timeout sec],[timeout usec]	+select:[readfs] or error
Arguments: nfds: The highest numbered file descriptor in any of the three sets (read, write or accept) readfs : socket descriptors as bitlist (0|2 to monitor 0 and 2) timeout sec : Time elapsed before select returns in sec timeout usec: Time in microseconds	

Table 44: AT+select

8.4.4. Socket settings

Once a socket is created, the descriptor can be used to modify its properties using the socket option commands described here.

Request			Response
AT+setSockOpt=[socketID],[levle],[option],[value1],...[valueX]			OK (or error)
level	**option**	**value**	
SOCKET	KEEPALIVE: enable/disable TCP keep active message	value1: 1=enable,0=disable	
	KEEPALIVETIME: keep alive timeout	value1: timeout in seconds	
	RX_NO_IP_BOUNDARY: enable/disable RX IP boundary	value1: 1=enable,0=disable	
	RX_NO_IP_BOUNDARY: enable/disable RX IP boundary	value1: 1=enable,0=disable	
	RCVTIMEO : timeout value that specifies maximum amount of time an input function waits until it completes	value1: seconds value2:microseconds	
	RCVBUF:TCP maximum receive window size	value1: size in bytes	
	NONBLOCKING: Set socket to non blocking	value1: 1=enable,0=disable	
	SECMETHOD: Sets security method to TCP socket	value1: SSLV3, TLSV1, TLSV1_1, TLSV1_2 SSLV3_TLSV1_2(highest possible)	
	SECURE_MASK:Set specific ciphers as bit mask (default= all ciphers)	value1: cipher type see table 47	
	SECURE_FILES_CA _FILE_NAME: Map secured socket to CA file by name	value1: absolute file path	
	SECURE_FILES_PRIVATE_ KEY_FILE_NAME: Map secured socket to private key by name	value1: absolute file path	
	SECURE_FILES_ CERTIFICATE_FILE_NAME: Map secured socket to certificate file by name	value1: absolute file path	
	SECURE_FILES_ DH_KEY_FILE_NAME: Map secured socket to Diffie Hellman file by name	value1: absolute file path	
	SECURE_DOMAIN_ NAME_VERIFICATION :Set a domain name, to check in SSL client connection	value1: Domain name	

Table 45: AT+setSockOpt

Request			Response
level	**option**	**value**	
IP	MULTICAST_TTL: Set the time-to-live value of outgoing multicast packets	value1: Number of hops	
	ADD_MEMBERSHIP:UDP socket, join a multicast group	Value1: IPv4 multicast address Value2: Multicast interface address	
	DROP_MEMBERSHIP:UDP socket, leave a multicast group	Value1: IPv4 multicast address Value2: Multicast interface address	

Table 46: AT+setSockOpt continued.

Supported Cipher methods
SSL_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_RC4_128_MD5
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_RC4_128_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256

Table 47: Supported cipher methods

Request	Response
AT+getSockOpt=[socketID],[level],[option]	+getSockOpt:[value1],...[valueX] or error
Arguments: socketID: socket descriptor level : SOCKET or IP option : see table 45 and 46 timeout usec: Time in microseconds	

Table 48: AT+getSockOpt

8.4.5. Socket data exchange

Once a socket is created and set up the data transfer can be done using send and receive commands using the commands described in this section.

Request	Response
AT+recv=[socketID], [format], [length]	+recv:[socketID], [length],[data] or error
Arguments: socketID : socket descriptor format : data format 0=binary, 1=base64 (binary to text encoding) length : Max number of bytes to receive	

Table 49: AT+recv

The module allocates memory for data reception, depending on the length field of the receive command. If not enough memory can be allocated an error is returned. We recommend to use a maximum length field of 1460.

Request	Response
AT+recvFrom=[socketID],[family],[remotePort], [remoteAddress], [format], [length]	+recvFrom:[socketID], [length],[data] or error
Arguments: socketID : socket descriptor family : INET or INET6 remotePort : Port of the peer to connect to remoteAddress : Address to connect to format : data format 0=binary, 1=base64 (binary to text encoding) length : Max number of bytes to receive	

Table 50: AT+recvFrom

The module allocates memory for data reception, depending on the length field of the receive command. If not enough memory can be allocated an error is returned. We recommend to use a maximum length field of 1460.

Request	Response
AT+send=[socketID], [format], [length],[data]	OK or error
Arguments: socketID : socket descriptor format : data format 0=binary, 1=base64 (binary to text encoding) length : number of bytes to send (max 1460) data: date to send	

Table 51: AT+send

Request	Response
AT+sendTo=[socketID],[family],[remotePort], [remoteAddress], [format], [length], [data]	OK or error
Arguments: socketID : socket descriptor family : INET or INET6 remotePort : Port of the peer to connect to remoteAddress : Address to connect to format : data format 0=binary, 1=base64 (binary to text encoding) length : number of bytes to send (max 1460) data: date to send	

Table 52: AT+sendTo

8.5. File system commands

Calypso creates and maintains an ecrypted file system on the serial flash present on-board. The file system provides secure storage for files like certificates, private keys and web pages. Here are some of the features of the file system.
The storage capacity for additional content in the radio module's file system is limited to the available capacity in the file system itself.

- The file system can only be accessed through AT commands.
- File system on one module cannot be read by another - this prevents cloning of sFlash.
- Built in tamper detection detects corrupt files and warns the user of unauthenticated file access.
- Each file has a minimum size of 4096 bytes (Fail-safe = 8192 bytes).
- Maximum number of files is 240 of which 100 is reserved for system files.
- File name can be a maximum 180 bytes.
- Files can created with one or more of the following flags: fail-safe, secure, public read, public write.
- Files cannot be enlarged once created hence the maximum size attribute has to be set appropriately during file creation.

File system does not handle fragmentation

Minimize the number of writes to flash to ensure data endurance

A file creation/deletion updates the FAT table. Rewrite/overwrite files when possible

Switch off the NWP when performing a file operation

Care needs to be taken to have a clean and stable supply voltage especially during flash writes in a battery powered applications. A drop in voltage during a erase cycle may lead to corruption of the file system

8.5.1. File system operations

Request	Response
AT+FileGetFileList	+FileGetFileList:[fileName], [maxFileSize], [properties], [fileBlocksAlloc] or error
Arguments: fileName : File name maxFileSize : Max file size properties: Bit mask - open_write - open_read - must_commit - bundle_file - pending_commit - pending_bundle_commit - not_failsafe - not_valid - sys_file - secure - nosignature - public_write - public_read fileBlocksAlloc : Allocated blocks	

Table 53: AT+FileGetFileList

8.5.2. File operations

In this section, the file operation commands are described.

Request	Response
AT+fileOpen=[fileName],[options],[fileSize]	+fileOpen:[fileID],[secureToke] or error
Arguments: fileName : full file path options: READ - Read a file (no bit mask) WRITE - Open for write (optionally bitmask with CREATE) CREATE - Create a new file (optionally bitmask with WRITE or OVERWRITE) CREATE_FAILSAFE CREATE_SECURE CREATE_NOSIGNATURE (for secure files only) CREATE_STATIC_TOKEN (for secure files only) CREATE_VENDOR_TOKEN (for secure files only) CREATE_PUBLIC_WRITE (for secure files only) CREATE_PUBLIC_READ (for secure files only) fileSize : Max file size in bytes (mandatory for CREATE option)	

Table 54: AT+fileOpen

Please note that the maximum file name length is 180 bytes.

Request	Response
AT+fileClose=[fileID],[certificateFileName],[signature]	OK or error
Arguments: fileID :ID assigend from `AT+FileOpen` certificateFileName: Full path to certificate (optional) signature : The signature is SHA1 (optional)	

Table 55: AT+fileClose

Request	Response
AT+fileDel=[fileName],[secureToke]	OK or error
Arguments: FileName: Full path to file secureToken : Token assigend from `AT+FileOpen` (optional)	

Table 56: AT+fileDel

Request	Response
AT+fileGetInfo=[fileName],[secureToke]	+FileGetInfo:[Flags],[FileSize], [Allocated-Size],[Tokens],[storageSize], [WriteCounter] or error
Arguments: FileName: Full path to file secureToken : Token assigend from `AT+FileOpen` (optional)	

Table 57: AT+fileGetInfo

Request	Response
AT+fileRead=[fileID],[offset],[format],[length]	+FileRead:[format], [numberOfReadBytes], [data] or error
Arguments: fileID :ID assigned from `AT+FileOpen` offset : Offset to specific read block Format : 0=binary, 1=Base64 Length : Number of bytes to read	

Table 58: AT+fileRead

Request	Response
AT+fileWrite=[fileID],[offset],[format],[length], [data]	+FileWrite:[numberOfReadBytes] or error
Arguments: fileID :ID assigned from `AT+FileOpen` offset : Offset to specific block Format : 0=binary, 1=Base64 Length : Number of bytes to write Data	

Table 59: AT+fileWrite

The module allocates memory for data read/write depending on the length field of the command. If not enough memory can be allocated an error is returned. We recommend to use a maximum length field of 1460.

8.6. Network application commands

8.6.1. mDNS

The mDNS/DNS-SD is a distributed device/service discovery protocol used for resolving IP addresses and ports on an IP network. In contrast to standard DNS, mDNS protocol is distributed where each device can join an IP multicast group and advertise its services. Both IPv4 and IPv6 are supported with addresses 224.0.0.251, FF02::FB and UDP port 5353 are reserved for mDNS messages. Each module can register to up to five services.

By default, the mDNS service is enabled and the host name and the internal HTTP server are advertised on enabled interfaces.

The mDNS server is not power optimized. It is recommended to disable mDNS in battery powered applications

Request	Response
AT+netAppStart=[AppBitMap]	OK or error
Arguments: Bitmap - HTTP_SERVER, DHCP_SERVER, MDNS, DNS_SERVER	

Table 60: AT+netAppStart

Request	Response
AT+netAppStop=[AppBitMap]	OK or error
Arguments: Bitmap - HTTP_SERVER, DHCP_SERVER, MDNS, DNS_SERVER	

Table 61: AT+netAppStop

8.6.2. SNTP client

Calypso implements an on-board SNTP client with configurable server addresses. A list of up to three SNTP servers can be stored in the non-volatile memory. The module tries to connect to the servers in order of the stored address index. The time zone has to be set manually. In order to avoid overload on the SNTP server, a configurable minimum update interval can be specified.

SNTP client is disabled by default.

Request	Response
AT+netAPPGet=[AppID],[option]	+netAPPGet:[value] or error
Arguments: AppID : sntp_client Options : - enable - update_interval - time_zone - server_address	 value: 0=disabled, 1=enabled value: minimum update interval in seconds value : UTC +/- minutes value : list of server addresses

Table 62: AT+NetAPPGet

Request	Response
AT+netAPPSet=[AppID],[option], [value1]...[valueX]	OK or error
Arguments: AppID : sntp_client Options : - enable, value: 0=disabled, 1=enabled - update_interval, value: minimum update interval in seconds - time_zone, value : UTC +/- minutes - server_address, value1:server index (0-2), value2:server address(IP addresses or URL)	

Table 63: AT+NetAPPSet

Request	Response
AT+netappUpdateTime	OK or error
Synchronize device time with SNTP server	

Table 64: AT+netappUpdateTime

8.6.3. HTTP client

Calypso offers creation of a HTTP client and execution of commonly used methods including get,post,connect and delete. This enables the user to connect to any HTTP(S) server and transmit and receive data with ease. In the following all the commands to create and control a HTTP client are described.

Request	Response
AT+HttpCreate	+HttpCreate:[index] or error
	Arguments: index - client handle for all further operations

Table 65: AT+HttpCreate

Request	Response
AT+HttpDestroy=[index]	OK or error
Arguments: index - client handle	

Table 66: AT+HttpDestroy

Request	Response
AT+HttpConnect=[index],[host],[flags],[private key], [cert], [ca]	OK or error
Arguments: index - client handle host - host name flags - bitmask (ignore_proxy, host_exist) private key - full path (optional) certificate - full path (optional) ca- full path (optional)	

Table 67: AT+HttpConnect

Request	Response
AT+HttpDisconnect=[index]	OK or error
Arguments: index - client handle	

Table 68: AT+HttpDisconnect

Request	Response
AT+HttpSetProxy=[family],[port],[address]	OK or error
Arguments: family - INET or INET6 port - proxy server port address - proxy server address	

Table 69: AT+HttpSetProxy

Request	Response
AT+HttpSendReq=[index],[method],[uri],[flags],[format], [length], [data]	OK or error
Arguments: index - client handle method - get, post, head, options, put, del, connect uri - request URI string flags - chunk_start (Sets the request into chunked body) - chunk_end (Sets the request out of chunked body) - drop_body (Flushes the response body) format - data format for post/put (0=Bin,1=Base64) lenght - length of payload for post/put data - request payload for post/put	

Table 70: AT+HttpSendReq

Request	Response
AT+HttpReadResBody=[index], [format], [length]	+HttpReadResBody:[index], [flag], [format], length], [body]
Arguments: index - client handle format - request format (0=Bin,1=Base64) length - request data length	Arguments: index - client handle flag (0=data end, 1=more data available) length - length of returned data body - received data

Table 71: AT+HttpReadResBody

Request	Response
AT+HttpSetHeader=[index],[option],[flags],[format], [length], [data]	OK or error
Arguments: index - client handle option - see table 74 flags- bitmask (not_persistent, persistent) format - data format(0=Bin,1=Base64) length - (optional) data - (optional)	

Table 72: AT+HttpSetHeader

Request	Response
AT+HttpGetHeader=[index],[option], [format], [length]	+HttpGetHeader:[index],[format], [length],[data]
Arguments: index - client handle option - see table 74 format - data format(0=Bin,1=Base64) length - max data legth	 index - client handle format - data format(0=Bin,1=Base64) length - actual data length data - value

Table 73: AT+HttpGetHeader

Header options
res_age, res_allow, res_cache_control, res_connection, res_content_encoding, res_content_language, res_content_length, res_content_location, res_content_range, res_content_type, res_date, res_etag, res_expires, res_last_modified, res_location, res_proxy_auth, res_retry_after, res_server, res_set_cookie, res_trailer, res_tx_encoding, res_upgrade, res_vary, res_via, res_www_auth, res_warning, req_accept, req_accept_charset, req_accept_encoding, req_accept_language, req_allow, req_auth, req_cache_control, req_connection, req_content_encoding, req_content_language, req_content_location, req_content_type, req_cookie, req_date, req_expect, req_forwarded, req_from, req_host, req_if_match, req_if_modified_since, req_if_none_match, req_if_range, req_if_unmodified_since, req_origin, req_proxy_auth, req_range, req_te, req_tx_encoding, req_upgrade, req_user_agent, req_via, req_warning

Table 74: HTTP header options

The module allocates memory for user data read depending on the length field specified in the above commands. If not enough memory can be allocated an error is returned. We recommend to use a maximum length field of 1460.

8.6.4. MQTT client

MQTT (Message Queue Telemetry Transport) is a machine-to-machine (M2M) connectivity protocol based on publish/subscribe transport mechanism. Features like light-weight, low network bandwidth, scalability makes it ideal for low-power, low-bandwidth IoT applications. A MQTT network consists of a broker connected to multiple clients. Clients can each subscribe to several topics or publish on any topic. The broker on the other hand is responsible for receiving a published topic and pushing it to all the subscribed nodes.
Calypso offers AT commands to create an MQTT client, subscribe as well as publish topics. The following section describes these commands.

Request	Response
AT+mqttCreate=[clientID], [flags], [server address], [server port], [security method], [cipher] [private key], [CA], [DH key], [protocol], [blocking send], [data format]	+mqttCreate:[index] or error
Arguments: -client ID: MQTT client ID string -flags (bit mask): ip4 = IPv4 connection, ip6 = IPv6 connection, url = server address is an URL, sec = secure connection skip_domain_verify, skip_cert_verify, skip_date_verify -server address: IP or URL -server port: 0-65535 -security method: SSLV3, TLSV1, TLSV1_1, TLSV1_2, TLSV1_1, SSLV3_TLSV1_2 (mandatory if sec flag) -cipher: cipher type see table 47 (optional) -private key: Full path to key file (optional) -certificate: Full path to certificate (optional) -CA: Full path to CA (optional) -DH key: Full path to Diffie Hellman key (optional) -protocol: v3_1 = MQTT version 3.1, v3_1_1 = MQTT version 3.1.1 -blocking send: 0 = do not wait for server response, 1 = wait for server response -data format: set globally for all further commands, 0 = bin, 1 = Base64	index-client handle used for all other MQTT operations

Table 75: AT+MqttCreate

Request	Response
AT+mqttDelete=[index]	OK or error
Arguments: index: client handle	

Table 76: AT+MqttDelete

Request	Response
AT+mqttConnect=[index]	OK or error
Arguments: index: client handle	

Table 77: AT+MqttConnect

Request	Response
AT+mqttDisconnect=[index]	OK or error
Arguments: index: client handle	

Table 78: AT+MqttDisconnect

Request	Response
AT+mqttPublish=[index],[topic],[QOS],[retain], [messageLength],[message]	OK or error
Arguments: - index: client handle - topic: topic string - QOS: QOS0, QOS1, QOS2 - retain: 0 = do not retain, 1 = retain - message length: max 1460 - message: payload	

Table 79: AT+MqttPublish

Request	Response
AT+mqttSubscribe=[index], [number of Topics], [topic1],[QoSn],[reserved1] ... [topicX],[QoSn],[reservedX]	OK or error
Arguments: - index: client handle - number of topics: max 4 - topic: topic string - QOS: QOS0, QOS1, QOS2 - reserved: leave empty	

Table 80: AT+MqttSubscribe

Request	Response
AT+mqttUnsubscribe=[index], [number of Topics], [topic1],[reserved1] ... [topicX],[reservedX]	OK or error
Arguments: - index: client handle - number of topics: max 4 - topic: topic string - reserved: leave empty	

Table 81: AT+MqttUnsubscribe

Request		Response
AT+MqttSet=[index],[option], [value1],...[valueX]		OK (or error)
Arguments:		
index: client index		
option	**value**	
user	username string	
password	password string	
will	topic, QOS, retain, Message-Length, message	
keepalive	value in seconds	
clean	0 = persistent connection, 1 = clean connection	

Table 82: AT+MqttSet

8.6.5. Ping

Calypso supports ping network utility based on the standard ICMP protocol. Both IPv4 and IPv6 are supported. This utility can be used to test connectivity and round trip delay.

Request	Response
AT+netAPPPing=[family], [destination], [size], [delay], [timeout], [max], [flags]	+netAPPPing:[packetsSent], [packetReceived], [RoundTripTime] or error
Arguments: family : INET or INET6 destination : Destination IP address (0 to stop an ongoing ping) size : Size of ping in bytes delay : Delay between pings in milliseconds timeout : Timeout for each ping in milliseconds max : Number of pings to send (0 = forever) flag : 0 = report once all pings are done, 1 = report after every ping, 2= Stop after first successful ping	

Table 83: AT+netAPPPing

8.7. Events

The host can receive an indication of specific states through events or errors. Asynchronous events can be sent to the host at any given time with an indication of specific states and specific data for each event.

8.7.1. General events

The general event may be received in relation to general device operation.

Event:	
+eventgeneral=[ID],[value1],...[valueX]	
ID	**Values**
reset_request	value1=Code
	value2=Software module - other - wlan - netcfg - netapp - security
error	value1=Code
	value2=Software module - other - wlan - netcfg - netapp - security

Table 84: +eventgeneral event

8.7.2. WLAN events

The WLAN event may be received in relation to a WLAN connection.

Event:	
+eventwlan=[ID],[value1],...[valueX]	
ID	**Values**
connect	value1=SSID
	value2=BSSID
disconnect	value1=SSID
	value2=BSSID
	value3=Reason, see chapter B.1
sta_added	value1=MAC
sta_removed	value1=MAC
p2p_connect	value1=SSID
	value2=MAC
	value3=GO device name
p2p_disconnect	value1=SSID
	value2=MAC
	value3=Reason, see chapter B.1
	value4=GO device name
p2p_client_added	value1=MAC
	value2=GO device name
	value3=Own SSID
p2p_client_removed	value1=MAC
	value2=GO device name
	value3=Own SSID
p2p_devfound	value1=GO device name
	value2=MAC
	value3=WPS
p2p_request	value1=GO device name
	value2=MAC
	value3=WPS
p2p_connectfail	value1=Status - disconnected - scaning - connecting - connected

Table 85: +eventwlan event

Event:	
+eventwlan=[ID],[value1],...[valueX]	
ID	**Values**
provisioning_status	value1=Provisioning status
	value2=Role
	value3=Status - disconnected - scaning - connecting - connected
	value4=SSID
provisioning_profile_added	value1=Provisioning status
	value2=SSID

Table 86: +eventwlan event

8.7.3. Socket events

The socket event may be received in relation to socket operation.

Event:	
+eventsocket=[ID],[value1],...[valueX]	
ID	**Values**
tx_failed	value1=SD
	value2=Status
async_event	value1=SD
	value2=Type - ssl_accept - rx_frag_too_big - other_side_close_ssl - connected_secured - wrong_root_ca
	value4=SSID
	value3=Value

Table 87: +eventsocket event

8.7.4. NetApp events

The NetApp event may be received in relation to network processor operation.

Event:	
+eventnetapp=[ID],[value1],...[valueX]	
ID	**Values**
ipv4_acquired	value1=Address
	value2=Gateway
	value3=DNS
ipv6_acquired	value1=Address
	value2=DNS
ip_collision	value1=Address
	value2=DHCP MAC
	value3=Conflict MAC
dhcpv4_leased	value1=Address
	value2=Lease time
	value3=BSSID
dhcpv4_released	value1=Address
	value2=BSSID
	value3=Reason
ipv4_lost	value1=Status
dhcp_ipv4_acquire_timeout	value1=Status
ipv6_lost	value1=IP lost

Table 88: +eventnetapp event

8.7.5. MQTT events

The MQTT event may be received in relation to one of the MQTT operations performed by the module.

Event:	
+eventmqtt=[ID],[value1],...[valueX]	
ID	**Values**
operation	value1=Operation ID (connack, puback, suback, unsuback)
	Connack:value2= 8 bit MSB - ACK flags, 8 bit LSB - Return code 0=connection accepted, 1=identifier rejected, 2=server unavailable, 3=bad username/password, 4=not authorised
	Puback:value2= Packet ID
	Suback:value2= Packet ID, value3 to valueX=return code per topic. 0=Success(QOS0), 1=Success(QOS1), 2=Success(QOS2), 128= Failure
	Unsuback:value2= Packet ID
recv	value1=Topic
	value2=QoS
	value3=Retain(0=not retain; 1=retain)
	value4=Duplicate(0=new; 1=duplicate)
	value5=Data format(0=bin,1=Base64)
	value6=Data length
	value7=Data
disconnect	

Table 89: +eventmqtt event

8.7.6. Fatal error events

The fatal error event may be received in case of device malfunction.

Event:	
+eventfatalerror=[ID],[value1],...[valueX]	
ID	**Values**
device_abort	value1=Code
	value2=Value
driver_abort	
sync_loss	
no_cmd_ack	value1=Code
cmd_timeout	value1=Code

Table 90: +eventfatalerror event

9. Provisioning

To enable easy provisioning when integrated to an embedded system with limited HMI capabilities, the Calypso offers a provisioning mode. In this mode, the module acts as an AP and allows external devices with appropriate credentials to connect and access the on-board HTTPS server. The user can conveniently browse the settings web-page and configure the module using any web-browser.

The web pages for provisioning require JavaScript.

9.1. Start the provisioning mode

There are two ways to set the Calypso to provisioning mode.

1. When starting the module in AT command mode the command

   ```
   AT+provisioningstart
   ```

   starts the provisioning.

2. Alternatively the application mode pins *APP_MODE_0* and *APP_MODE_1* can be used to define the application mode, as described in chapter 5.2.1. To do so, apply a LOW signal to the *APP_MODE_0* pin, a HIGH signal to the *APP_MODE_1* pin and restart the module.

When the provisioning mode has been started successfully, the LED at *STATUS_IND_1* flashes with interval of 1s. The module has created an access point with a SSID "calypso_" followed by by the MAC of the module (example "calypso_CAFFEE123456"). Now any WiFi enabled device can connect to the access point using WPA2 security and the key "calypsowlan".

Please note that after a timeout of 5 minutes the radio module exits the provisioning mode.

9.2. Enter the credentials

When the provisioning mode has been started, connect to the provisioning device and open the provisioning website under "calypso.net".

The domain "calypso.net" is local and hence cannot be verified by known root CAs. Hence, the on-board HTTPS server uses a self-signed certificate. As a result of this, the browser on the configuring device may a report security risk. In this case the user has to trust the certificate and proceed to the website in order to perform the provisioning operation on the WLAN module.

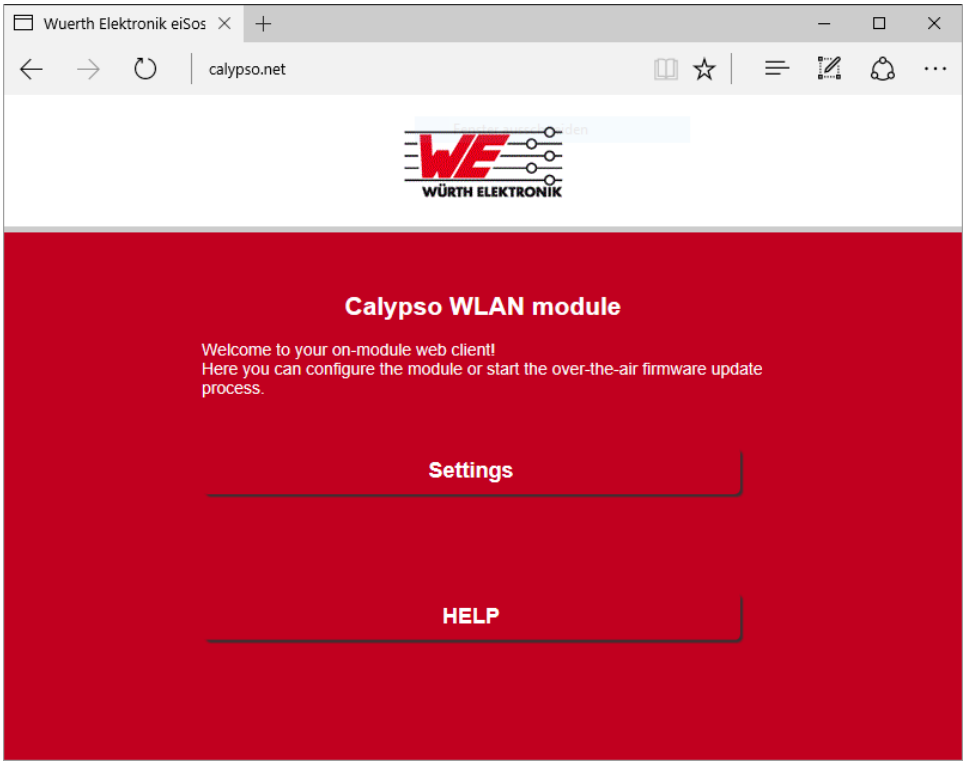


Figure 10: Provisioning main page

Click on the "Settings" button to open the settings menu. Several tabs are available to read the current status of the radio module and to configure it. To save a WLAN profile in the radio module go to the "Profiles" tab and enter the credentials of the access point the radio module is supposed to connect to.

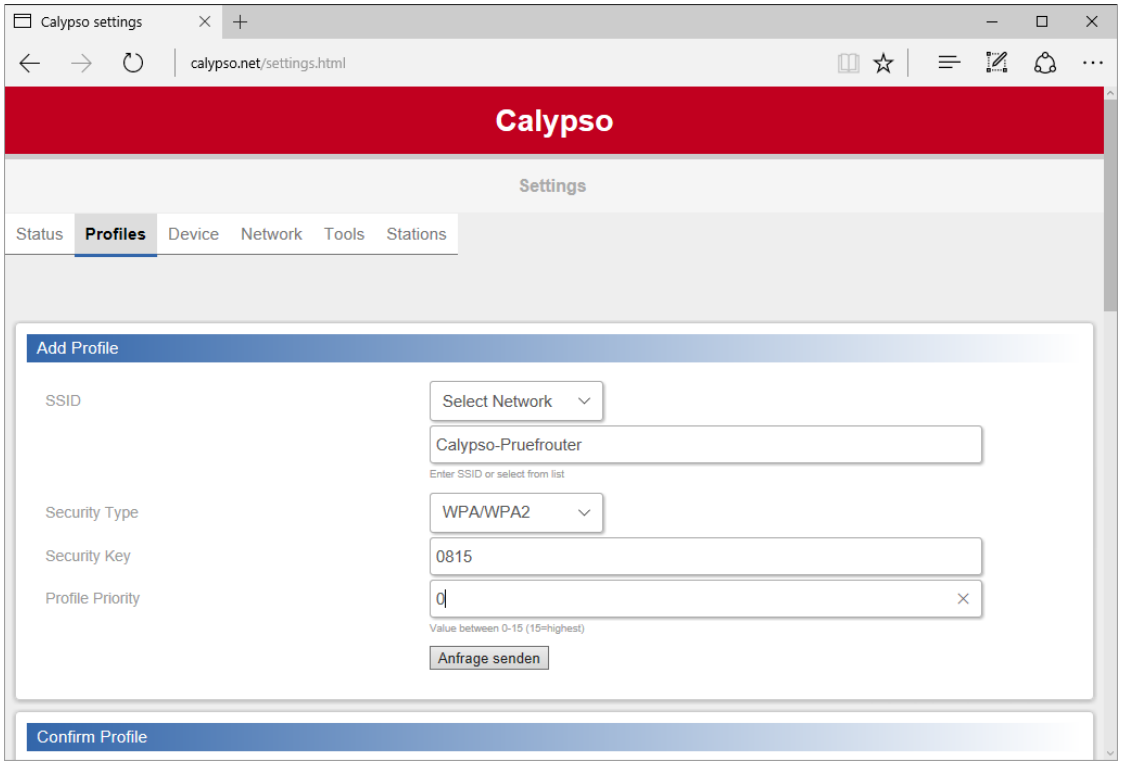


Figure 11: Provisioning main page

Press the button below to add the defined credential to the radio module. The radio module then prints on the UART the corresponding +eventwlan message.

```
+eventwlan:provisioning_profile_added,no_error,Calypso-Pruefrouter
```

In default settings the setting "WLAN policy connection" (see chapter 8.2.6) is set to "auto", meaning that the device automatically tries to connect to the AP, that is defined in the module's profiles. Thus after adding the profile to the module, a restart has to be performed. To do so send a

```
AT+reboot
```

or press the reset button.

Please make sure that the application mode pins *APP_MODE_0* and *APP_MODE_1* are set correctly, when restarting the device.

After restarting in AT command mode, the module automatically connects to the pre-defined AP.

```
+eventwlan:connect,Calypso-Pruefrouter,0x0:0x25:0x9c:0xcf:0x85:0xf0
+eventnetapp:ipv4_acquired,192.168.1.101,192.168.1.50,192.168.1.50
```

10. Typical application use cases

In this section some of the typical use cases for the Calypso module are considered and a simple example is described in each case.

10.1. UDP communication

UDP is a connectionless transport layer protocol used to exchange data between peers in an IP network. Section 8.4 describes the basics of BSD sockets and figure 8 shows the work flow for UDP communication.

10.1.1. Prerequisites

The following hardware is required to go through the quick start example.

1. Two Calypso evaluation boards.
2. An IEEE 802.11b/g/n compatible access point working in the 2.4 GHz band.
3. Computer with a serial terminal emulator like Tera Term.

Assuming that the EV boards have the hardware configuration as described in section 4.5.2, the next step in the process is to connect both the EV boards to the AP as described in section 4.5.5. In this example, the modules have the IP addresses 192.168.1.169 and 192.168.1.140.

10.1.2. UDP socket communication

1. Create a UDP socket using the following command. Note the socket ID returned for use in future commands (in this case "0").

```
AT+socket=INET,DGRAM,UDP
+socket:0
OK
```

2. Although the bind on a UDP socket is optional, it is essential here to know the destination port to send to at the peer (in this case port 8888). A bind can be done using the following command. Where "0" is the socket ID from the socket creation command above.

```
AT+bind=0,INET,8888,192.168.1.169
OK
```

3. Repeat the above steps on the second module.
4. Use the AT+sendTo commands with destination port and address to send data packets.

```
AT+sendTo=0,INET,8888,192.168.1.169,0,32,3U0fRSk9UaYx00ABvhPU1vBH7tgnGIqW
OK
```

5. To receive the data packets, use the AT+recvfrom command as shown below

```
AT+recvFrom=0,INET,8888,192.168.1.140,0,32
OK
+recvFrom:0,0,32,3U0fRSk9UaYx00ABvhPU1vBH7tgnGIqW
OK
```

10.2. TCP communication

Refer to section 4.5 for detailed description of creating a TCP server and client and data exchange between them.

10.3. Secure socket communication

SSL/TLS layer provides added security features like server authentication and end-to-end encryption. This exampled describes creation of an SSL/TLS server as well as client on Calypso EV board and exchange of data between the two.
The following hardware is required to go through the quick start example.

1. Two Calypso evaluation boards.
2. An IEEE 802.11b/g/n compatible access point working in the 2.4 GHz band.
3. Computer with a serial terminal emulator like Tera Term.
4. Server certificate and key stored on the sFlash of the server module.

Assuming that the EV boards have the hardware configuration as described in section 4.5.2, the next step in the process is to connect both the EV boards to the AP as described in section 4.5.5. In this example, the modules have the IP addresses 192.168.1.169 (SSL/TLS client) and 192.168.1.140 (SSL/TLS server).

10.3.1. Create an SSL/TLS server

The module with IP address 192.168.1.140 is configured as SSL/TLS server.

1. Create a simple TCP socket with the following command. Note the socket ID for future reference.

```
AT+socket=INET,STREAM,TCP
+socket:0
```

2. The next step is to upgrade the socket to secure by updating the socket options.

```
AT+setSockOpt=0,socket,secmethod,SSLV3_TLSV1_2
OK
```

3. The SSL/TLS server needs a certificate and the corresponding private key to be stored on sFlash. In this case the certificate "dummy-trusted-cert" and the key "dummy-trusted-cert-key" are already present in the file system and configured to be used by the SSL server as shown.

```
AT+setSockOpt=0,socket,SECURE_FILES_PRIVATE_KEY_FILE_NAME,dummy-trusted-
    cert-key
OK
AT+setSockOpt=0,socket,SECURE_FILES_CERTIFICATE_FILE_NAME,dummy-trusted-
    cert
OK
```

4. Finally, bind the socket to a port (in this example 9999) and the local IP address and listen for connection requests.

```
AT+bind=0,INET,9999,192.168.1.140
OK
AT+listen=0,10
OK
```

10.3.2. Create an SSL/TLS client

The module with IP address 192.168.1.168 is configured as SSL/TLS client. And connected to the server configured in the previous section.

1. Create a simple TCP socket with the following command. Note the socket ID for future reference.

```
AT+socket=INET,STREAM,TCP
+socket:0
```

2. The next step is to upgrade the socket to secure by updating the socket options.

```
AT+setsockopt=0,socket,SECMETHOD,SSLV3_TLSV1_2
OK
```

3. In this example, the server root CA is present the root certificate catalog of the WLAN module and client certificate verification is disabled.

4. Now the client can perform a connect to the server.

```
AT+connect=0,INET,9999,192.168.1.140
OK
```

5. The +connect event will show up once the server has accepted the connection request as described in the next section.

```
+connect:9999,192.168.1.140
OK
```

10.3.3. Secure data transfer

1. The conncetion request from the client has to be accepted by the server. Note the socket ID generated by the server for this client.

```
AT+accept=0,INET
OK
```

2. The +accept event will show up on the server side, once the server has accepted the connection request of a client. It returns the port and the IP address of the current client as well as the new socket ID generated for communication with this client (in this case socket ID "1").

```
+accept:1,inet,50020,192.168.1.169
OK
```

3. With the connection established the end-to-end encrypted data transfer can be done as shown below. The server can send a message to the client:

```
AT+send=1,0,32,YJaZ4yUGKRES7mE5ApBDo0zrFRtq56Jt
OK
```

4. Which is indicated by a +recv event in the client.

```
+recv:0,0,32,YJaZ4yUGKRES7mE5ApBDo0zrFRtq56Jt
OK
```

5. The client can reply to this message also using the AT+send (with socket ID "0" to address the server).

```
AT+send=0,0,32,iuwIHSis5xTttzffbtfhjtfh678pSHJA
OK
```

6. Which is indicated by a +recv event in the server with socket ID 1.

```
+recv:1,0,32,iuwIHSis5xTttzffbtfhjtfh678pSHJA
OK
```

7. Close the sockets using the command and corresponding Socket ID.

```
AT+close=0
+close:0
OK
```

10.4. Wi-Fi direct example

The Wi-Fi direct standard enables peer-to-peer communication between two compatible devices without the need for an infrastructure AP. Wi-Fi direct enabled devices negotiate their roles and one of them assumes the role of a Group Owner (GO) (equivalent to an AP) and the other the role of a Client. The discovery of devices is done by sending/listening broadcasting packets on channels 1, 6 and 11. This section demonstrates the Wi-Fi direct capabilities of Calypso module by connecting two Calypso EV boards over Wi-Fi direct.

10.4.1. Prerequisites

The following hardware is required to go through this Wi-Fi direct example.

1. Two Calypso evaluation boards.
2. Computer with a serial terminal emulator like Tera Term.

10.4.2. Auto connection setup

First of all, the P2P settings of both devices have to be configured. Here we use the following settings:
Connect to the first P2P device that is found.

```
AT+wlanpolicyset=connection,P2P,
```

The role (client or group owner) and negotiation request strategy (active, passive or random back-off) can be set as needed. Here we choose for simplicity to negotiate the role (client or group owner) and send the negotiation request as soon as a P2P device has been found.

```
AT+wlanpolicyset=P2P,negotiate,active
```

Set the device to P2P mode.

```
AT+wlansetmode=P2P
```

Restart the network processor.

```
AT+stop=0
```

```
AT+start
```

And scan for P2P devices.

```
AT+wlanscan=0,5
```

As soon as a P2P device has been found, the connection is setup. In case of the group owner, the output is as follows.

```
+eventwlan:p2p_devfound,calypso,0x98:0x84:0xe3:0xf6:0x8c:0x1,
+eventwlan:p2p_request,calypso,0x98:0x84:0xe3:0xf6:0x8c:0x1,pbc
+eventwlan:p2p_client_added,0x98:0x84:0xe3:0xf6:0x8c:0x1,calypso,DIRECT-GJ
+eventnetapp:dhcpv4_leased,10.123.45.2,86400,0x98:0x84:0xe3:0xf6:0x8c:0x1
```

In case of the client, the output is as follows.

```
+eventwlan:p2p_devfound,calypso,0xc8:0xfd:0x19:0x5:0x5e:0xef,
+eventwlan:p2p_request,calypso,0xc8:0xfd:0x19:0x5:0x5e:0xef,pbc
+eventwlan:p2p_connect,DIRECT-GJ,0xc8:0xfd:0x19:0x5:0x5e:0xef,calypso
+eventnetapp:ipv4_acquired,10.123.45.2,10.123.45.1,10.123.45.1
```

Now a socket can be created to transmit/receive data. Please refer to the chapters `10.1` and `10.2` to do so.
After data has been transmitted/received, the connection can be closed again.

```
AT+wlandisconnect
```

10.4.3. Manual connection setup

This chapter describes how to set-up a P2P connection between two Calypso radio modules. The goal is to establish a connection to the client (module B) initiated by the group owner (module A).

First of all, the P2P settings of module A has to be configured. Here we use the following settings: Furthermore we configure the role as "group owner" and negotiation request strategy as "active".

```
AT+wlanpolicyset=P2P,group_owner,active
```

Set the device to P2P mode.

```
AT+wlansetmode=P2P
```

Restart the network processor.

```
AT+stop=0
```

```
AT+start
```

Repeat the previous steps with module B. Use here "client" and "passive" in the `AT+wlanpolicyset` command.

```
AT+wlanpolicyset=P2P,client,passive
```

After both devices have been configured and the network processor has been restarted, start the scan for P2P devices.

```
AT+wlanscan=0,5
```

As soon as a P2P device has been found, the following message occurs.

```
+eventwlan:p2p_devfound,calypso,0x98:0x84:0xe3:0xf6:0x8c:0x1,
```

To setup a connection to the found P2P device a `AT+wlanconnect` command has to be placed, including the name of the peer device using the Push Button Configuration (PBC) for example.

```
AT+wlanconnect=calypso,,P2P_PBC,,,,
```

In case of the group owner, the output is as follows.

```
+eventwlan:p2p_devfound,calypso,0x98:0x84:0xe3:0xf6:0x8c:0x1,
+eventwlan:p2p_request,calypso,0x98:0x84:0xe3:0xf6:0x8c:0x1,pbc
+eventwlan:p2p_client_added,0x98:0x84:0xe3:0xf6:0x8c:0x1,calypso,DIRECT-GJ
+eventnetapp:dhcpv4_leased,10.123.45.2,86400,0x98:0x84:0xe3:0xf6:0x8c:0x1
```

In case of the client, the output is as follows.

```
+eventwlan:p2p_devfound,calypso,0xc8:0xfd:0x19:0x5:0x5e:0xef,
+eventwlan:p2p_request,calypso,0xc8:0xfd:0x19:0x5:0x5e:0xef,pbc
+eventwlan:p2p_connect,DIRECT-GJ,0xc8:0xfd:0x19:0x5:0x5e:0xef,calypso
+eventnetapp:ipv4_acquired,10.123.45.2,10.123.45.1,10.123.45.1
```

Now a socket can be created to transmit/receive data. Please refer to the chapters `10.1` and `10.2` to do so.
After data has been transmitted/received, the connection can be closed again.

```
AT+wlandisconnect
```

10.5. Running a web page on the radio module

The Calypso radio module offers a secure file system to store files in the radio module. In combination with the HTTP(S) server function, a custom web site can be run on the module. This chapter describes how to do so by loading a simple html file (see Code `1`) to the module's flash memory. Furthermore the customization of the web site access is demonstrated in the subsequent sections.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Simple web page</title>
</head>
<body>
This is a simple webpage
</body>
</html>
```

Code 1: Example html code

10.5.1. Load the web page files to the radio module

First of all the `AT+FileGetFileList` can be used to check the file system content of the radio module.

```
AT+FileGetFileList
+filegetfilelist:/www/help.html,3656,2
+filegetfilelist:/www/images/icon/help.png,3656,2
```

```
+filegetfilelist:/www/images/icon/menu.png,3656,2
+filegetfilelist:/www/images/icon/wireless.png,3656,2
+filegetfilelist:/www/ota.html,11848,6
+filegetfilelist:/www/settings.html,11848,6
...
OK
```

To load a file onto the radio module a new file has to be generated on the radio module by using the `AT+FileOpen` command. In this command, the file name has to be defined, as well as the maximum file size and the options (create and write in this case) of the file.

```
AT+FileOpen=/www/mytest.html,WRITE|CREATE,3656
+fileopen:1966156880,0
OK
```

It returns a file descriptor (1966156880 in this example) that has to be used in the following actions.
To load the mentioned html file onto the module the `AT+FileWrite` command can be used, that is prepended by the file descriptor, the data length and the file data itself.

```
AT+FileWrite=1966156880,0,0,104,<html><head><title>Simple web page</title></head
    ><body><div>This is a simple webpage</div></body></html>
+filewrite:104
OK
```

After the transmission of the data to the radio module has been finished the file access must be closed by an `AT+FileClose` command.

```
AT+FileClose=1966156880,,
OK
```

More complex websites can be analogously put to the secure file system. Simply, several files have to be loaded to the module before accessing the web page for the first time.

The following sub chapters demonstrate how to access the web page that has be stored on the module.

10.5.2. Accessing the web site in station mode

Before accessing the new web page, we need to start the HTTP server:

```
AT+NetAppStart=HTTP_SERVER
OK
```

Then connect the radio module and your PC to the same network and call the new web page under the module's IP using a browser. In this example its "192.168.1.104/mytest.html":

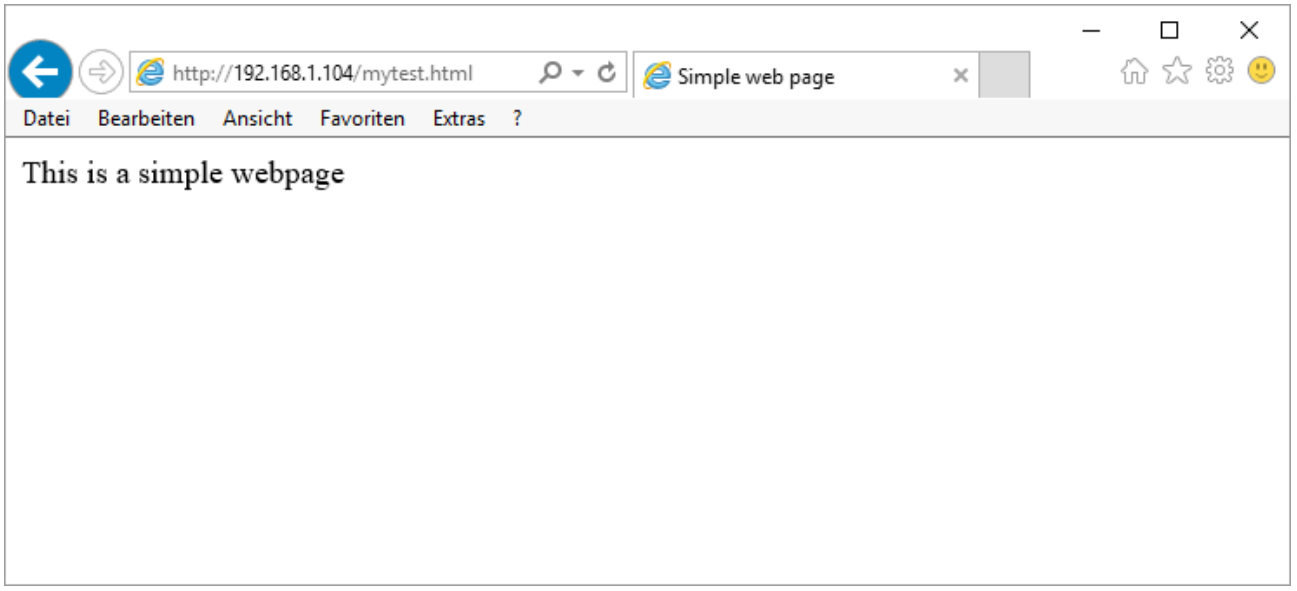


Figure 12: Test page

10.5.3. Accessing the web site in access point mode

To configure the radio module as access point we use the command:

```
AT+wlansetmode=AP
OK
```

In factory state the SSID of the radio module is "calypso" prepended by its MAC, the password is "calypsowlan" and the domain is "calypso.net".

Furthermore, we like to use an own SSID "mySSID" and a new password "mypassword" to access the wireless network. Therefore type:

```
AT+wlanset=AP,SSID,mySSID
OK
AT+wlanset=AP,password,mypassword
OK
```

Next, we would like to use our own domain "mywebpage.net":

```
AT+NetAppSet=DEVICE,DOMAIN,mywebpage.net
OK
```

Finally restart the network processor:

```
AT+stop=0
OK
AT+start
+eventnetapp:ipv4_acquired,10.123.45.1,10.123.45.1,0.0.0.0
OK
```

Now connect with your PC or smart phone to the WLAN of the Calypso radio module and call the website "mywebpage.net/mytest.html" using a browser.

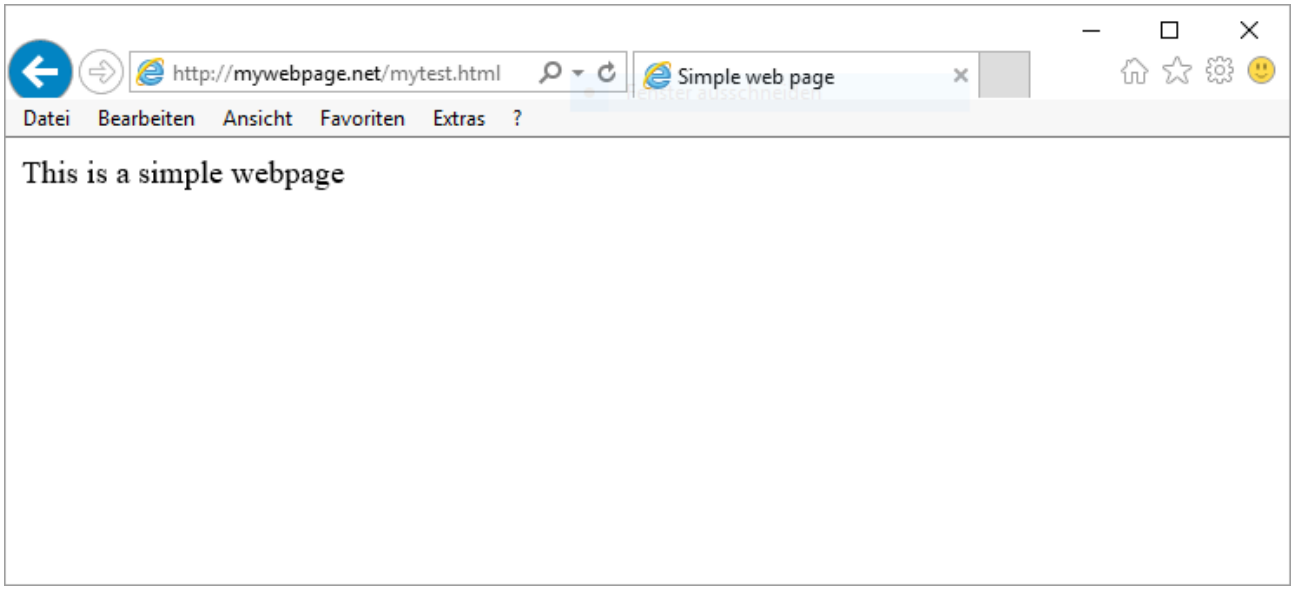


Figure 13: Test page

Please note that the radio module provides up to 4 connections in AP mode.

11. Timing parameters

This section describes the behaviour of the Calypso module during reset, sleep and wake-up operations.

11.1. Hard reset

A hard reset on the Calypso module is done by asserting a low on the */RESET*. On hard-reset, the module reloads the application from the sFlash after verifying the image to ensure the integrity of the application. This contributes towards higher start up times of the application.

Description	Typ.	Unit
Ready after reset	2	sec

Table 91: Start-up time

11.2. Soft reset

A software reset is made available through the AT command `AT+reboot` (see section `8.1`). In this case the module restarts from the reset vector. The exact same process happens after a wake-up signal from sleep mode.

Description	Typ.	Unit
Ready after reboot/wake-up	350	ms

Table 92: Start-up after reboot

It is recommended to use the AT command to reboot the device instead of a falling edge on the */Reset* pin whenever applicable.

Use `AT+stop` and `AT+start` to restart the network processor.

The fast/auto connect features ensure immediate connect to AP on reboot/wake-up.

12. Firmware update

Calypso supports secure over-the-air firmware updates to enable easy update of the module's firmware in the field. The module as a client connects to an infrastructure AP and a device (PC/tablet/smartphone) present in the same network can upload an encrypted image (provided by Würth Elektronik eiSos) using the on-board web-server.

12.1. Prerequisites

1. An infrastructure AP with known SSID key for security must be active and connectable. The AP or a device inside the AP's network must provide DHCP service to configure the connected stations. A connection to the internet is not required.
2. The module must be configured such that the credentials of the AP used for OTA are saved as Profile 0 and the connection policy is set to "AUTO" (see chapter 8.2.6).
3. The device (PC, Smartphone, ...) should be connected to the same AP and configured within the same network as the Calypso radio module. It can be any device with a browser supporting (self-)signed HTTPS content and JavaScript.

It is recommended to use the Chrome browser with JavaScript enabled. The self-signed certificate triggers a security error on the browser. Please trust the certificate and proceed to the OTA website.

4. The device used for updating the radio module shall have the compressed and encrypted firmware image for the Calypso 's OTA update in its local storage.

Using an unauthorized image may damage the module.

A maximum of 50 files, including system files, can be updated using the current update mechanism.

12.2. Update procedure

12.2.1. Start-up

Restart the module in the OTA operating mode, by setting and holding *APP_MODE_0* and *APP_MODE_1* accordingly (see chapter 5.2.1). A start-up message appears on the UART to indicate successful boot-up in OTA mode. If correctly configured, the Calypso automatically tries to connect to the AP saved as profile 0. *STATUS_IND_0* LED blinking at 1 Hz indicates WLAN connection in progress.

- In case of WLAN profile 0 being empty or no connection possible, the following message appears on the UART after a timeout of 5s. Please solve the connection issue before continuing.

```
+eventota:info,"Starting␣OTA␣update..."
+eventota:info,"Device␣is␣configured␣in␣default␣state"
+eventota:timeout,"Make␣sure␣that␣a␣valid␣AP␣profile␣is␣saved␣at␣index␣
    0"
```

- In case of the WLAN connection being successful, the following message appears and the *STATUS_IND_0* LED stays ON. In this case, the OTA procedure can be continued.

```
+eventota:info,"Starting␣OTA␣update..."
+eventota:info,"Device␣is␣configured␣in␣default␣state"
+eventota:connect,Calypso-Pruefrouter,0:25:9c:cf:85:f0
+eventota:ipacquired,192.168.1.101,192.168.1.50
```

12.2.2. Connection to the update device

Following, the module tries to ping the gateway. During this procedure the *STATUS_IND_1* LED blinks at 1 Hz. As soon as the pinging has been completed the *STATUS_IND_1* LED stays solid.

```
+eventota:info,"Pinging␣gateway,␣please␣wait..."
+eventota:info,"Ping␣completed"
+eventota:info,"Waiting␣for␣new␣ota␣upload..."
```

The message "Waiting for new ota upload..." indicates that the module is successfully connected to the network and ready to receive the update file. Make sure that the device (PC/smartphone) containing the update-package is connected to the same network. On this device, open the web-page "[modul ip]\ota.html" in a web-browser.
Example: http:\\192.168.1.101\ota.html in case the module's IP is 192.168.1.101

The browser must allow JavaScript and proxy server must be switched off.

On successful connection, the web-page with information about the module is displayed on the web-browser.

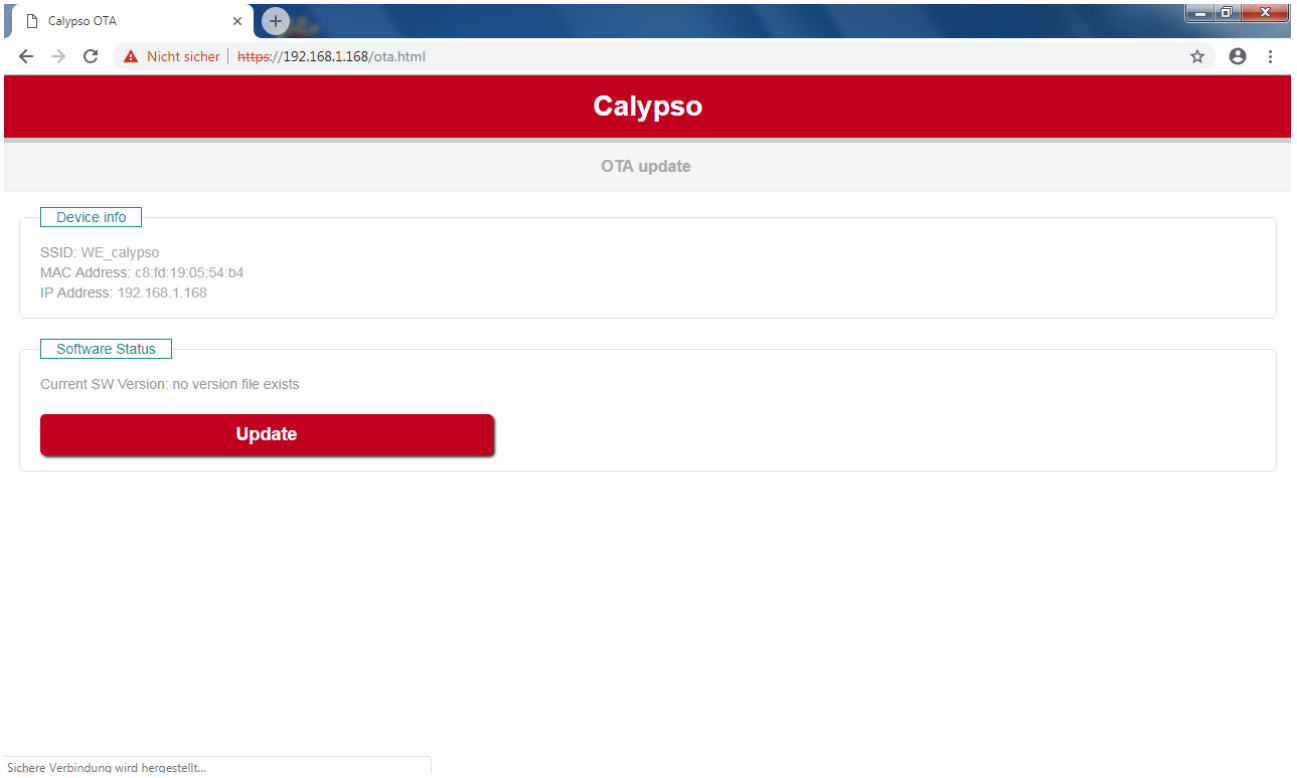


Figure 14: OTA webpage

12.2.3. Upload the update-package

On the OTA page click on update button followed by choose file button. A file browser opens up. Browse to the location where the update packet is stored and select the same. Click on upload file to start the update process. During update the module outputs the OTA states on the UART. In this state the *STATUS_IND_1* LED blinks at 2 Hz.

Using browsers other than Chrome, the progress bar is found to not update correctly. In this case refresh the page manually.

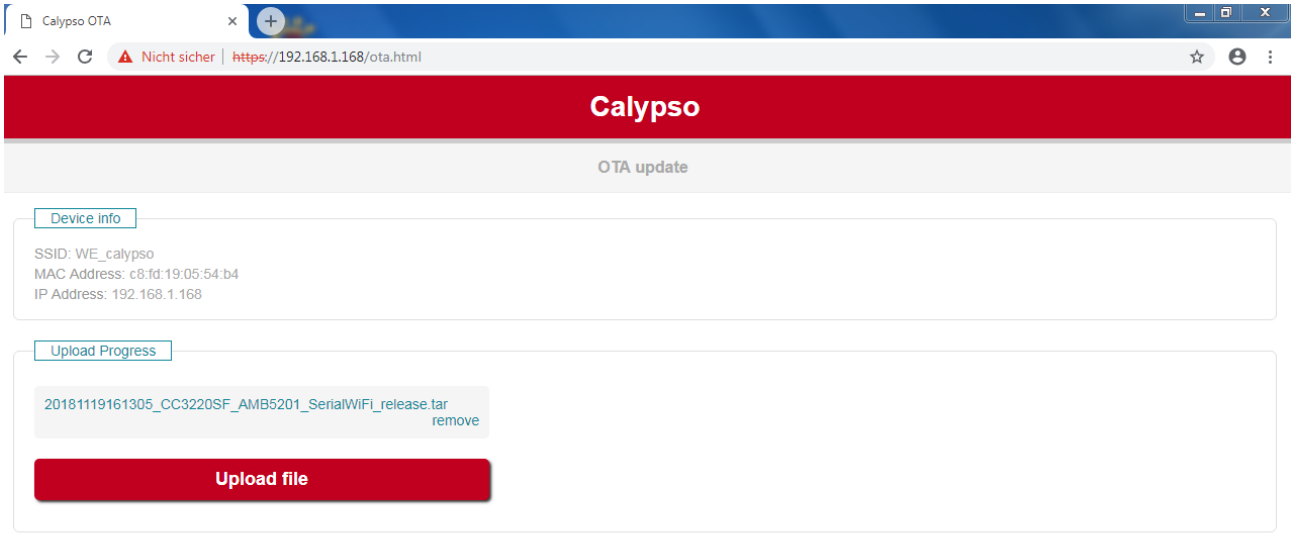


Figure 15: OTA webpage upload

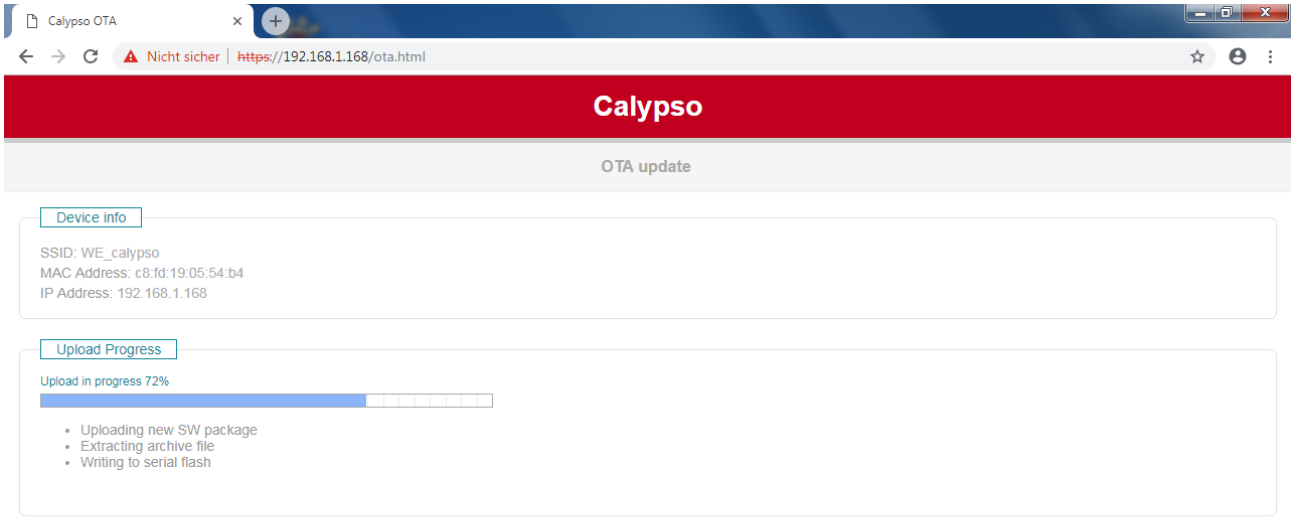


Figure 16: OTA in progress

12.2.4. Finalize the update

On completion, the module outputs the following message on the UART and reboots.

```
+eventota:info,"Received␣OTA␣filename␣20181121135643
    _CC3220SF_AMB5201_SerialWiFi_release.tar,␣len␣=␣440320␣"
+eventota:info,"Download␣complete"
```

The boot-up after an OTA update may require additional time (up to 60 seconds) in comparison to a normal boot-up. In the browser click on finalize to complete the OTA process. After this step, the ota.html shall show the new firmware version. A module reconfiguration via AT commands or Provisioning is required after the firmware update.

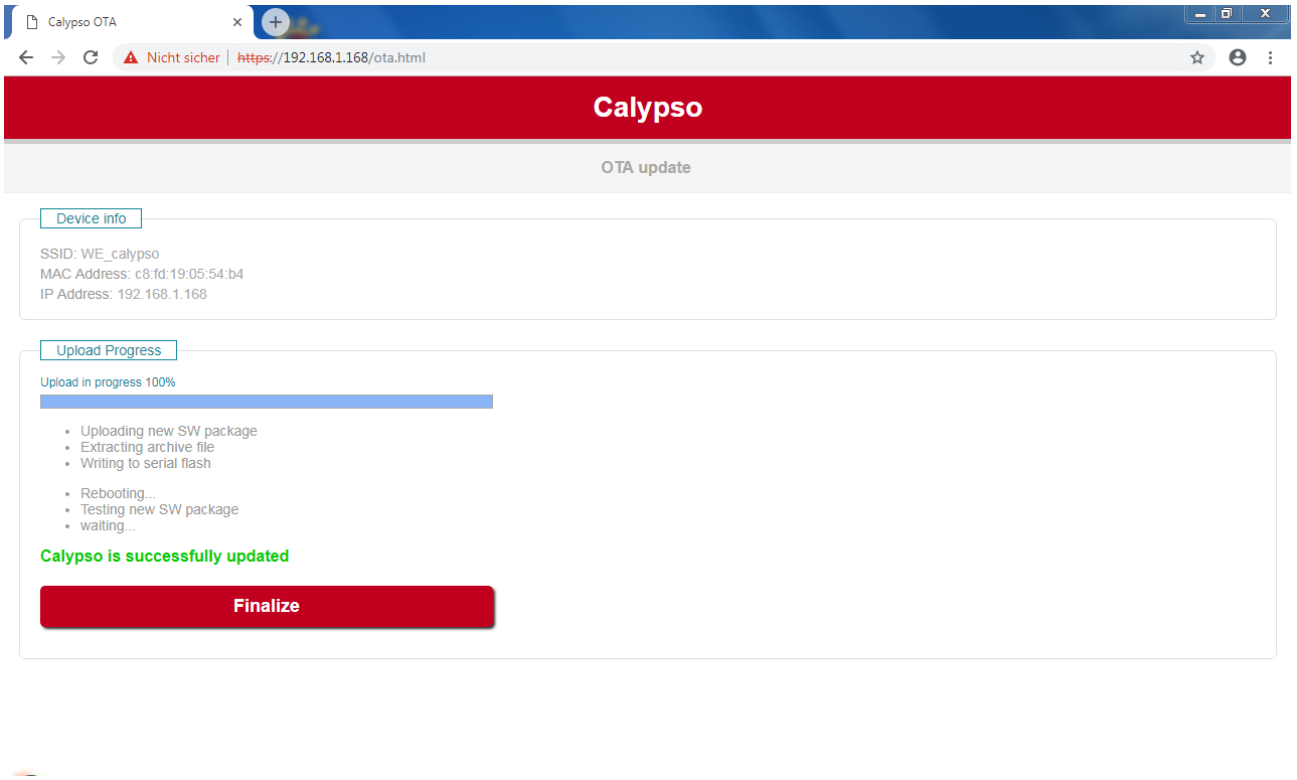


Figure 17: Finalize OTA

13. Firmware history

Version 0.x.x "Engineering"

Version 1.0.0 "Release"

The Calypso firmware is based on SimpleLink Wi-Fi CC3220 Software development kit (SDK) from Texas Instruments with the corresponding features as well as known issues. The table 93 lists the version of different components used for the current firmware version.
Known issues:

- Command `AT+MqttSet` does not work as expected. It will be fixed in subsequent firmware versions.

Description	Version
SimpleLink SDK	2.20.00.10
Service pack	sp_3.9.0.6_2.0.0.0_2.2.0.6
NWP	3.9.0.6
MAC	2.0.0.0
PHY	2.2.0.6
ROM	0

Table 93: Version

14. Custom firmware

14.1. Custom configuration of standard firmware

The configuration of standard firmware includes adoption of the non-volatile Usersettings (see chapter 6) to customer requirements and creating a customized product on base of the standard product with a unique ordering number for a specific customer that needs this configuration.
For example if the UART baud rate shall be changed from the default value to another value. This variant will result in a customer exclusive module with a unique ordering number. This will also fix the firmware version to a specific and customer tested version and thus results in a customer exclusive module with a unique ordering number.
Further scheduled firmware updates of the standard firmware will not be applied to this variant automatically. Applying updates or further functions require a customer request and customer release procedure.

14.2. Customer specific firmware

A customer specific firmware may include "Custom configuration of standard firmware" plus additional options or functions and tasks that are customer specific and not part of the standard firmware.
Further scheduled firmware updates of the standard firmware will not be applied to this variant automatically. Applying updates or further functions require a customer request and customer release procedure.
This also results in a customer exclusive module with a unique ordering number.
An example for this level of customization are functions like host-less operation where the module will perform data generation (e.g. by reading a SPI or I^2C sensor) and cyclic transmission of this data to a data collector while sleeping or being passive most of the time.
Also replacing UART with SPI as host communication interface is classified such a custom specific option.
Certification critical changes need to be re-evaluated by an external qualified measurement laboratory. These critical changes may occur when e.g. changing radio parameters, the channel access method, the duty-cycle or in case of various other functions and options possibly used or changed by a customer specific firmware.

14.3. Customer firmware

A customer firmware is a firmware written and tested by the customer himself or a 3rd party as a customer representative specifically for the hardware platform provided by a module.
This customer firmware (e.g. in form of a Intel hex file) will be implemented into the module's production process at our production side.
This also results in a customer exclusive module with a unique ordering number.
The additional information needed for this type of customer firmware, such as hardware specific details and details towards the development of such firmware are not available for the public and can only be made available to qualified customers.

The qualification(s) and certification(s) of the standard firmware cannot be applied to this customer firmware solution without a review and verification.

14.4. Contact for firmware requests

Please contact your local field sales engineer (FSE) or wireless-sales@we-online.com for quotes regarding this topics.

15. Design in guide

15.1. Advice for schematic and layout

For users with less RF experience it is advisable to closely copy the relating evaluation board with respect to schematic and layout, as it is a proven design. The layout should be conducted with particular care, because even small deficiencies could affect the radio performance and its range or even the conformity.
The following general advice should be taken into consideration:

- A clean, stable power supply is strongly recommended. Interference, especially oscillation can severely restrain range and conformity.
- Variations in voltage level should be avoided.
- LDOs, properly designed in, usually deliver a proper regulated voltage.
- Blocking capacitors and a ferrite bead in the power supply line can be included to filter and smoothen the supply voltage when necessary.

No fixed values can be recommended, as these depend on the circumstances of the application (main power source, interferences etc.).

Frequently switching the module on and off, especially with a slowly changing voltage level of the power supply, can lead to erratic behavior, in rare cases even as far as damaging the module or the firmware. The use of an external reset IC can solve this matter and shall be considered especially in battery operated scenarios.

- Elements for ESD protection should be placed on all pins that are accessible from the outside and should be placed close to the accessible area. For example, the RF-pin is accessible when using an external antenna and should be protected.
- ESD protection for the antenna connection must be chosen such as to have a minimum effect on the RF signal. For example, a protection diode with low capacitance such as the LXES15AAA1-100 or a 68 nH air-core coil connecting the RF-line to ground give good results.
- Placeholders for optional antenna matching or additional filtering are recommended.
- The antenna path should be kept as short as possible.

Again, no fixed values can be recommended, as they depend on the influencing circumstances of the application (antenna, interferences etc.).

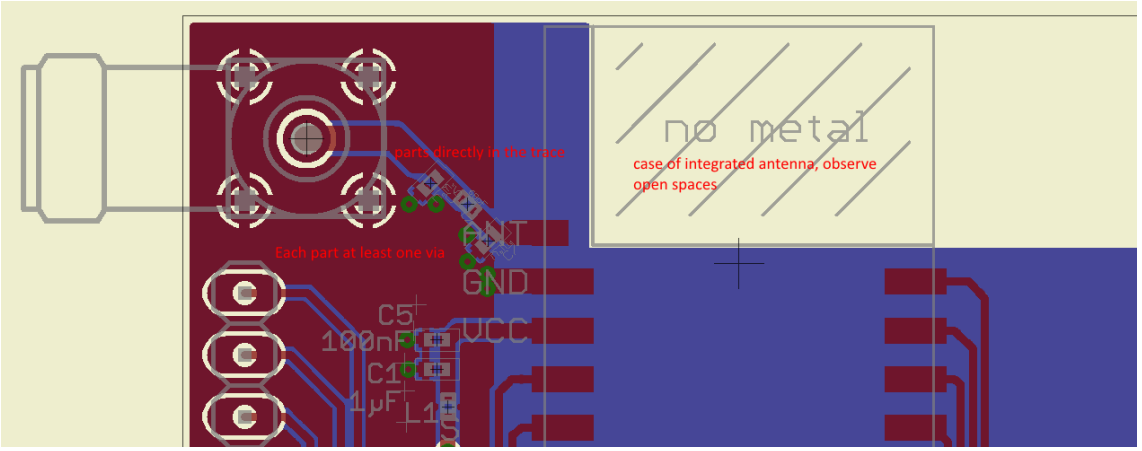


Figure 18: Layout

- To avoid the risk of short circuits and interference there should be no routing underneath the module on the top layer of the baseboard.
- On the second layer, a ground plane is recommended, to provide good grounding and shielding to any following layers and application environment.
- In case of integrated antennas it is required to have areas free from ground. This area should be copied from the evaluation board.
- The area with the integrated antenna must overlap with the carrier board and should not protrude, as it is matched to sitting directly on top of a PCB.
- Modules with integrated antennas should be placed with the antenna at the edge of the main board. It should not be placed in the middle of the main board or far away from the edge. This is to avoid tracks beside the antenna.
- Filter and blocking capacitors should be placed directly in the tracks without stubs, to achieve the best effect.
- Antenna matching elements should be placed close to the antenna / connector, blocking capacitors close to the module.
- Ground connections for the module and the capacitors should be kept as short as possible and with at least one separate through hole connection to the ground layer.
- ESD protection elements should be placed as close as possible to the exposed areas.

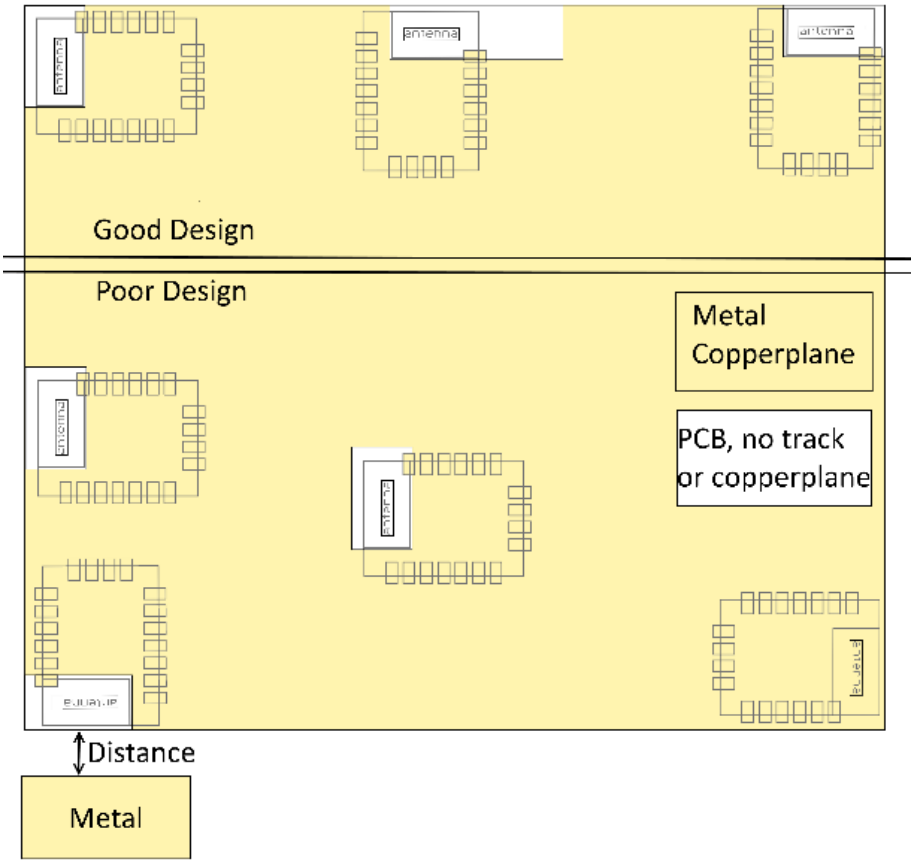


Figure 19: Placement of the module with integrated antenna

15.2. Dimensioning of the micro strip antenna line

The antenna track has to be designed as a 50Ω feed line. The width W for a micro strip can

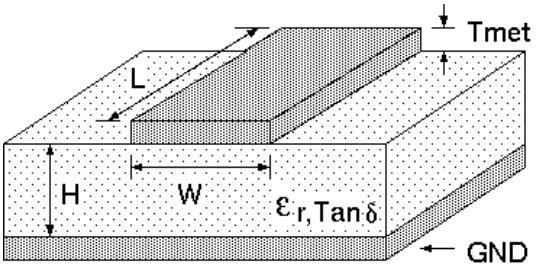


Figure 20: Dimensioning the antenna feed line as micro strip

be calculated using the following equation:

$$W = 1.25 \times \left(\frac{5.98 \times H}{e^{\frac{50 \times \sqrt{\epsilon_r} + 1.41}{87}}} - T_{met} \right) \quad (1)$$

Example:
A FR4 material with ε_r = 4.3, a height H = 1000 µm and a copper thickness of T_{met}= 18 µm

will lead to a trace width of W $\sim$ 1.9 mm. To ease the calculation of the micro strip line (or e.g. a coplanar) many calculators can be found in the internet.

- As rule of thumb a distance of about 3×W should be observed between the micro strip and other traces / ground.
- The micro strip refers to ground, therefore there has to be the ground plane underneath the trace.
- Keep the feeding line as short as possible.

15.3. Antenna solutions

There exist several kinds of antennas, which are optimized for different needs. Chip antennas are optimized for minimal size requirements but at the expense of range, PCB antennas are optimized for minimal costs, and are generally a compromise between size and range. Both usually fit inside a housing.
Range optimization in general is at the expense of space. Antennas that are bigger in size, so that they would probably not fit in a small housing, are usually equipped with a RF connector. A benefit of this connector may be to use it to lead the RF signal through a metal plate (e.g. metal housing, cabinet).
As a rule of thumb a minimum distance of $\lambda/10$ (which is 3.5 cm @ 868 MHz and 1.2 cm @ 2.44 GHz) from the antenna to any other metal should be kept. Metal placed further away will not directly influence the behavior of the antenna, but will anyway produce shadowing.

Keep the antenna away from large metal objects as far as possible to avoid electromagnetic field blocking.

The choice of antenna might have influence on the safety requirements.

In the following chapters, some special types of antenna are described.

15.3.1. Wire antenna

An effective antenna is a $\lambda/4$ radiator with a suiting ground plane. The simplest realization is a piece of wire. It's length is depending on the used radio frequency, so for example 8.6 cm 868.0 MHz and 3.1 cm for 2.440 GHz as frequency. This radiator needs a ground plane at its feeding point. Ideally, it is placed vertically in the middle of the ground plane. As this is often not possible because of space requirements, a suitable compromise is to bend the wire away from the PCB respective to the ground plane. The $\lambda/4$ radiator has approximately 40 Ω input impedance, therefore matching is not required.

15.3.2. Chip antenna

There are many chip antennas from various manufacturers. The benefit of a chip antenna is obviously the minimal space required and reasonable costs. However, this is often at the expense of range. For the chip antennas, reference designs should be followed as closely as possible, because only in this constellation can the stated performance be achieved.

15.3.3. PCB antenna

PCB antenna designs can be very different. The special attention can be on the miniaturization or on the performance. The benefits of the PCB antenna are their small / not existing (if PCB space is available) costs, however the evaluation of a PCB antenna holds more risk of failure than the use of a finished antenna. Most PCB antenna designs are a compromise of range and space between chip antennas and connector antennas.

15.3.4. Antennas provided by Würth Elektronik eiSos

15.3.4.1. 2600130041 - 434 MHz dipole antenna

Figure 21: 2600130041: 434 GHz dipole-antenna

Specification	Value
Frequency range [GHz]	
Impedance [Ω]	50
VSWR	$\leq$ 1.5
Polarization	Vertical
Radiation	Omni
Gain [dBi]	0
Antenna Cover	TPEE
Dimensions (L x d) [mm]	90 x 12
Weight [g]	
Connector	SMA plug
Operating Temp. [℃]	-40 – +85

This antenna requires a ground plane which will influence the electrical parameters.

15.3.4.2. 2600130081 - 868 MHz dipole antenna

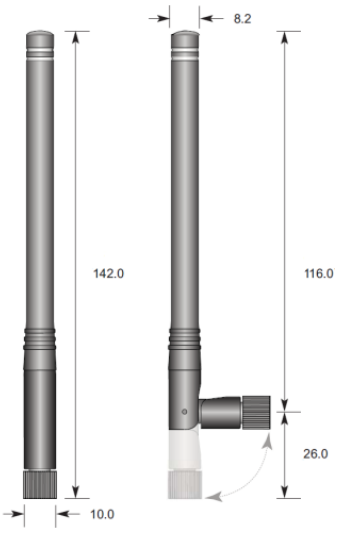


Figure 22: 2600130081: 868 MHz dipole-antenna

Ideally suited for applications where no ground plane is available.

The 2600130081 antenna can be also used for 902MHz - 928MHz range.

Specification	Value
Center frequency [MHz]	868
Frequency range [MHz]	853 – 883
Wavelength	0.5 wave
VSWR	$\leq$ 2.0
Impedance [Ω]	50
Connector	SMA (Male)
Dimensions (L x d) [mm]	142 x 10
Peak gain [dBi]	-2.3
Operating temp. [℃]	-30 – +80

15.3.4.3. 2600130082 - 868 MHz magnetic base antenna

Well suited for applications where the RF is lead through a metal wall that could serve as ground plane to the antenna.

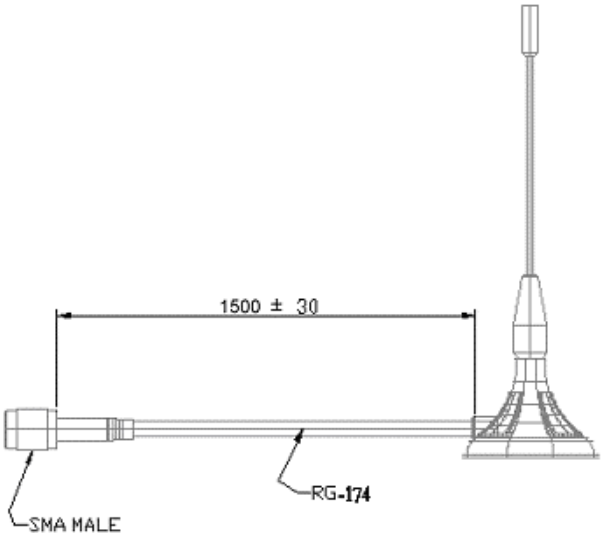


Figure 23: 2600130082: 868 MHz magnet foot antenna with 1.5 m antenna cable

The 2600130082 is a kind of λ/4 radiator and therefore needs a ground plane at the feeding point.

Specification	Value
Frequency range [MHz]	824 – 894
VSWR	≤ 2.0
Polarisation	Vertical
Impedance [Ω]	50±5
Connector	SMA (Male)
Dimensions (L x d) [mm]	89.8 x 27
Weight [g]	50±5
Operating temp. [°C]	-30 – +60

15.3.4.4. 2600130021 - 2.4 GHz dipole antenna

Figure 24: 2600130021: 2.4 GHz dipole-antenna

Due to the fact, that the antenna has dipole topology there is no need for an additional groundplane. Nevertheless the specification was measured edge mounted and 90° bent on a 100 x 100 mm ground plane.

Specification	Value
Frequency range [GHz]	2.4 – 2.5
Impedance [Ω]	50
VSWR	$\leq$ 2:1
Polarization	Linear
Radiation	Omni-Directional
Peak Gain [dBi]	2.8
Average Gain [dBi]	-0.6
Efficiency	85 %
Dimensions (L x d) [mm]	83.1 x 10
Weight [g]	7.4
Connector	SMA plug
Operating temp. [℃]	-40 – +80

16. Reference design

Calypso was tested and certified on the corresponding Calypso evaluation board. For the compliance with the EU directive 2014/53/EU Annex I, the evaluation board serves as reference design. For the FCC it serves as trace design.
This is no discrepancy due to the fact that the evaluation board itself does not fall within the scope of the EU directive 2014/53/EU Annex I as the module is tested on the evaluation board, which is also the recommended use.

Further information concerning the use of the evaluation board can be found in the manual of the Calypso evaluation board.

16.1. EV-Board

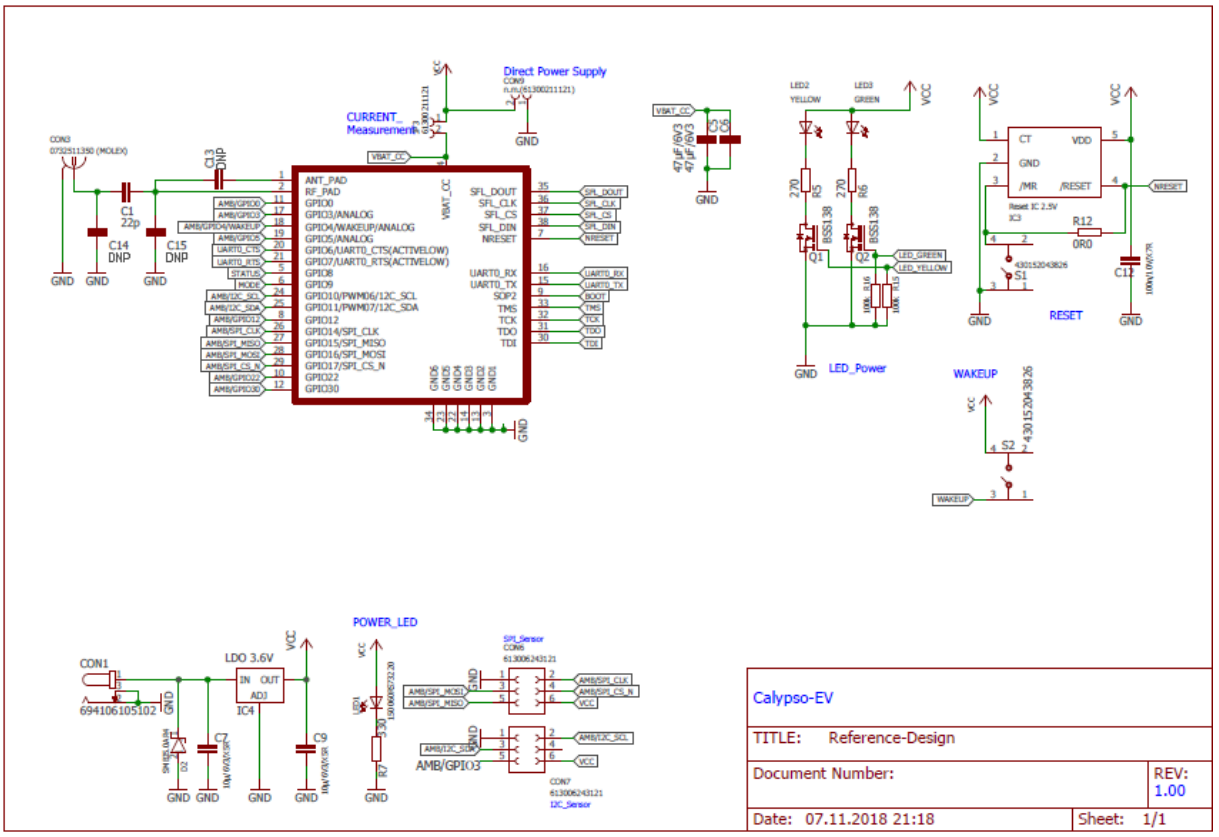


Figure 25: Reference design: Schematic, most important parts

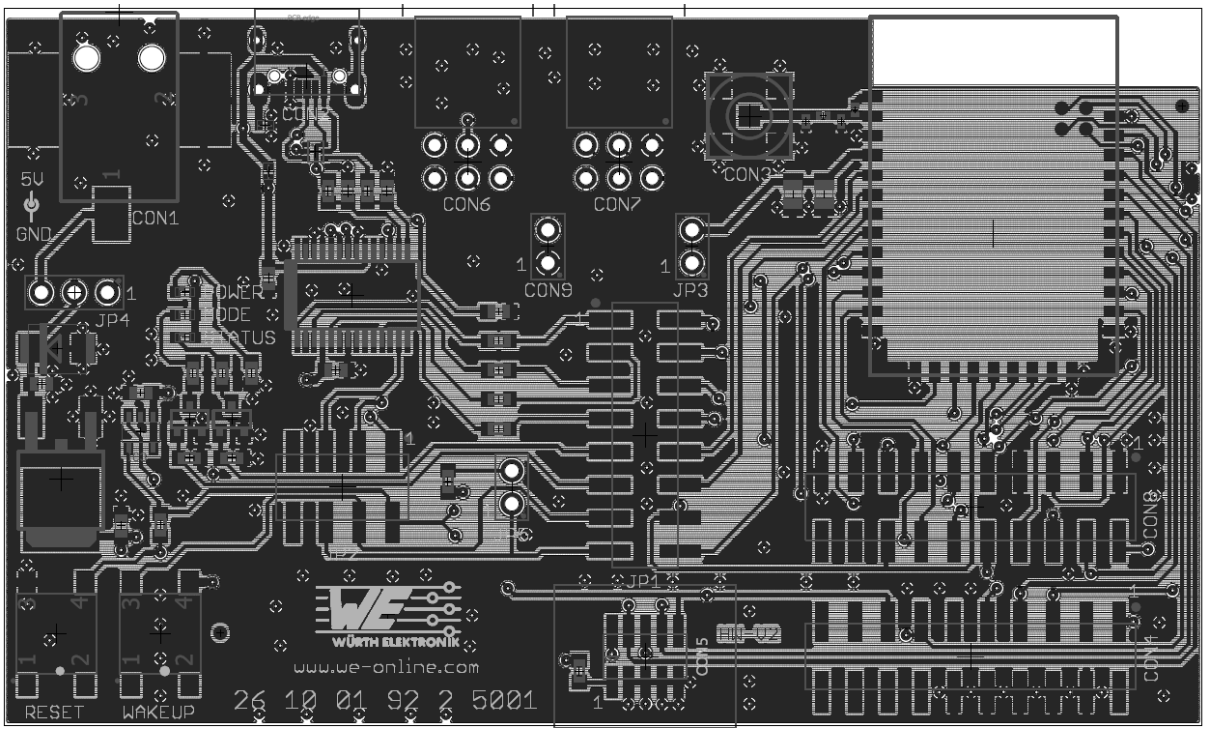


Figure 26: Reference design: Layout

16.2. Trace design

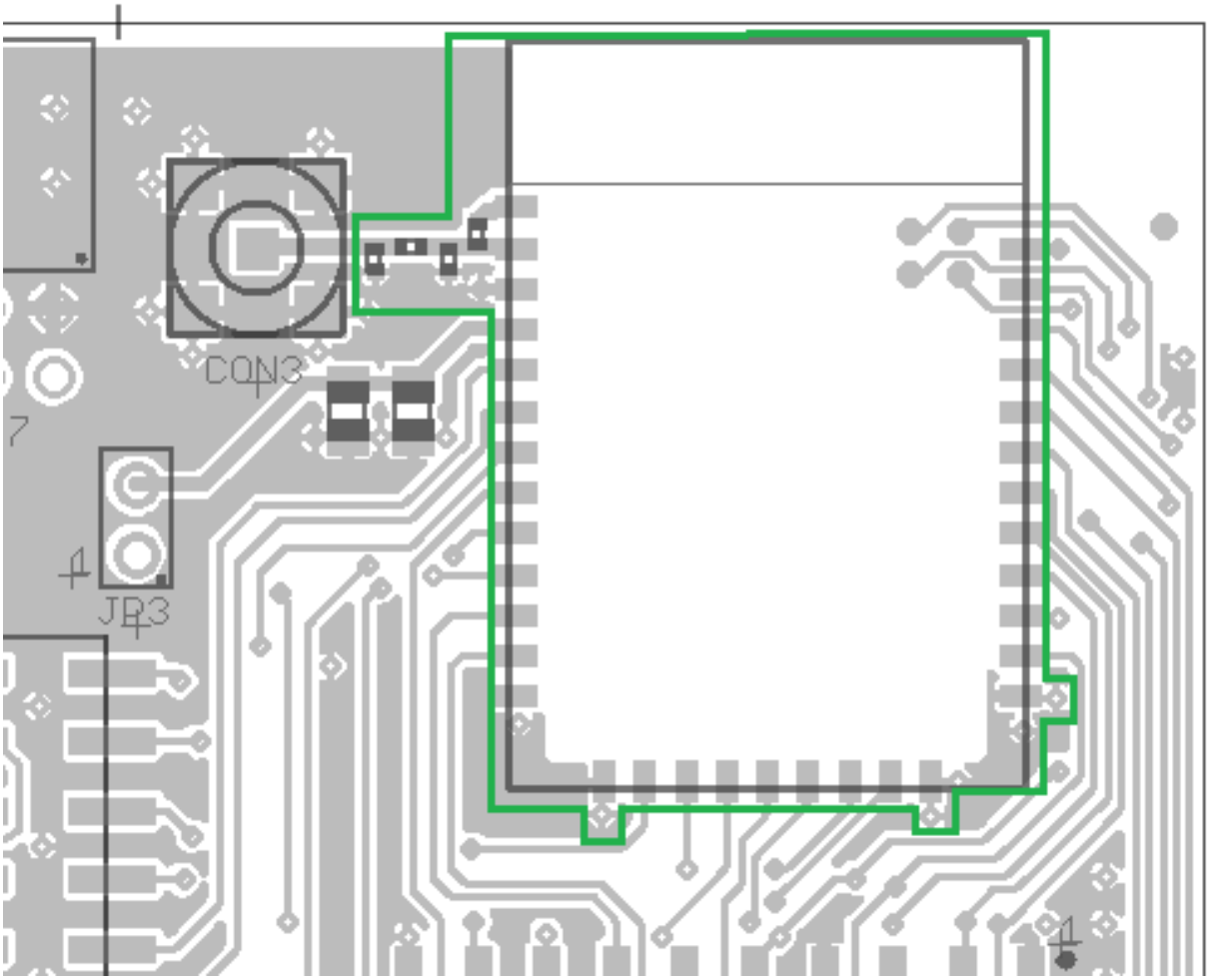


Figure 27: Trace design: Layout

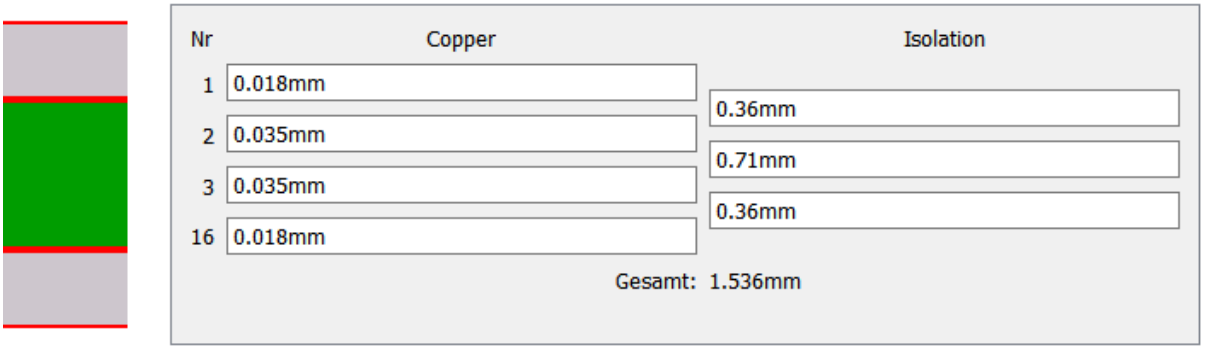


Figure 28: Reference design: Stack-up

- Top layer is used for routing and filled up with ground except underneath the module and the antenna free area.

- Second layer is ground, except the antenna free area.
- Third layer is the supply layer, except antenna free area. Some routing is allowed, not dividing the supply layer in to many or to small parts.
- Bottom layer is used for routing.

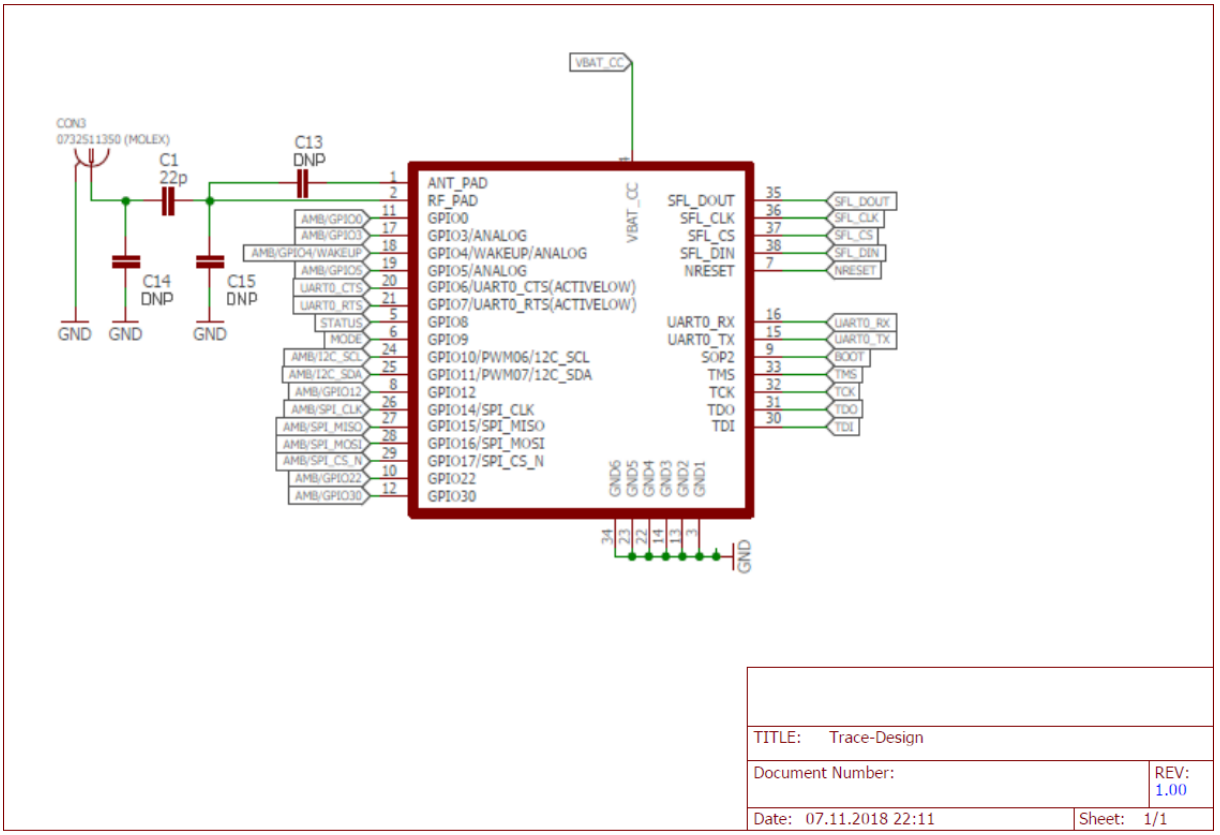


Figure 29: Trace design: Schematic

Two variants of the Calypso are certified:

- Using the on-board PCB antenna placing 22pF for C13 and not placing C1, C14 and C15.
- Placing 22pF for C13, not placing C13, C14 and C15 and connecting through a 50 Ω line a quarter-wave-length whip antenna.

To reference to the Würth Elektronik's FCC ID it is mandatory to use the trace design.

16.3. Application mode pins

The pins *APP_MODE_0* and *APP_MODE_1* define at boot time which application mode is used during operation of the module (see chapter 5.2.1).
To enable security updates of the firmware and/or HTTP server certificates via radio, the OTA mode or Provisioning mode may be used.

To manually switch to OTA mode or provisioning mode, we strongly recommend to make the pins *APP_MODE_0* and *APP_MODE_1* accessible on the custom PCB. Otherwise the update of expired HTTP server certificates and security patches cannot be applied to the radio module.

17. Manufacturing information

17.1. Moisture sensitivity level

This wireless connectivity product is categorized as JEDEC Moisture Sensitivity Level 3 (MSL3), which requires special handling.

More information regarding the MSL requirements can be found in the IPC/JEDEC J-STD-020 standard on www.jedec.org.

More information about the handling, picking, shipping and the usage of moisture/reflow and/or process sensitive products can be found in the IPC/JEDEC J-STD-033 standard on www.jedec.org.

17.2. Soldering

17.2.1. Reflow soldering

Attention must be paid on the thickness of the solder resist between the host PCB top side and the modules bottom side. Only lead-free assembly is recommended according to JEDEC J-STD020.

Profile feature		Value
Preheat temperature Min	$T_{S\ Min}$	150 ℃
Preheat temperature Max	$T_{S\ Max}$	200 ℃
Preheat time from $T_{S\ Min}$ to $T_{S\ Max}$	t_S	60 - 120 seconds
Ramp-up rate (T_L to T_P)		3 ℃ / second max.
Liquidous temperature	T_L	217 ℃
Time t_L maintained above T_L	t_L	60 - 150 seconds
Peak package body temperature	T_P	see table below
Time within 5 ℃ of actual peak temperature	t_P	20 - 30 seconds
Ramp-down Rate (T_P to T_L)		6 ℃ / second max.
Time 20 ℃ to T_P		8 minutes max.

Table 94: Classification reflow soldering profile, Note: refer to IPC/JEDEC J-STD-020E

Package thickness	Volume mm³ <350	Volume mm³ 350-2000	Volume mm³ >2000
< 1.6mm	260 °C	260 °C	260 °C
1.6mm - 2.5mm	260 °C	250 °C	245 °C
> 2.5mm	250 °C	245 °C	245 °C

Table 95: Package classification reflow temperature, PB-free assembly, Note: refer to IPC/-JEDEC J-STD-020E

It is recommended to solder this module on the last reflow cycle of the PCB. For solder paste use a LFM-48W or Indium based SAC 305 alloy (Sn 96.5 / Ag 3.0 / Cu 0.5 / Indium 8.9HF / Type 3 / 89%) type 3 or higher.

The reflow profile must be adjusted based on the thermal mass of the entire populated PCB, heat transfer efficiency of the reflow oven and the specific type of solder paste used. Based on the specific process and PCB layout the optimal soldering profile must be adjusted and verified. Other soldering methods (e.g. vapor phase) have not been verified and have to be validated by the customer at their own risk. Rework is not recommended.

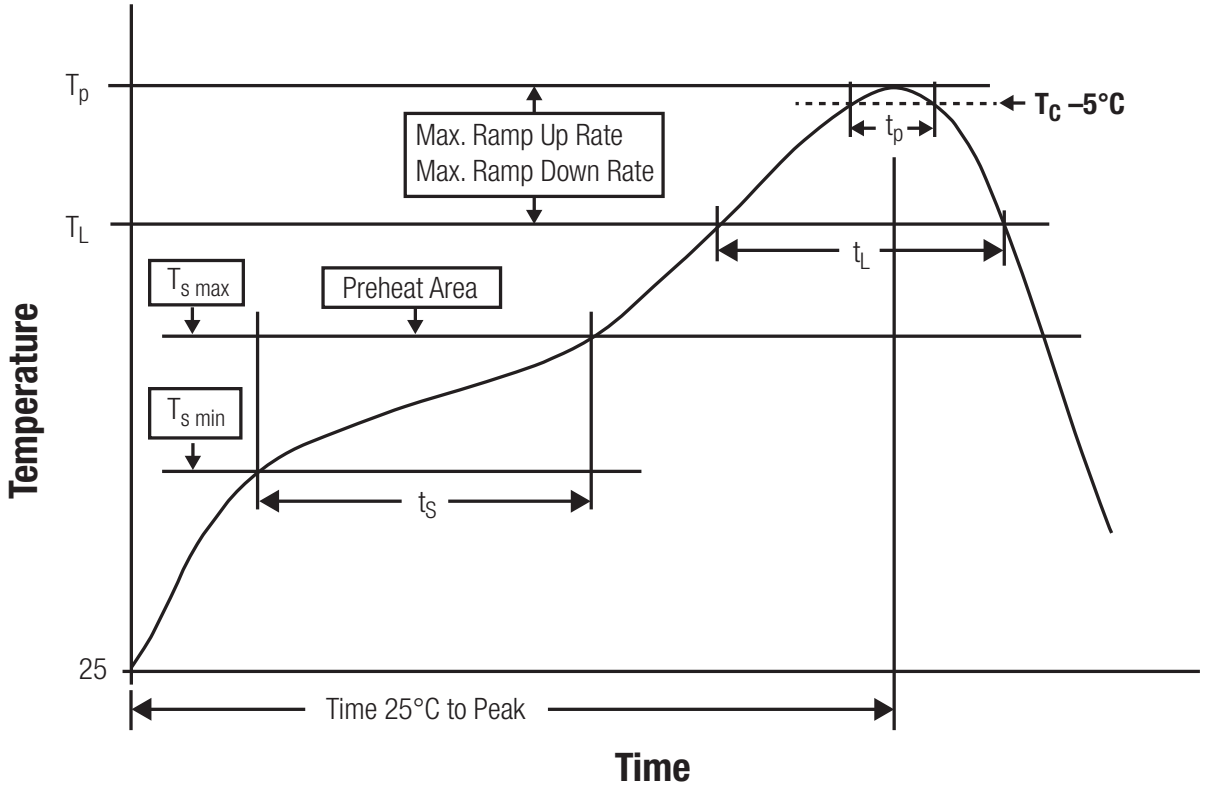


Figure 30: Reflow soldering profile

After reflow soldering, visually inspect the board to confirm proper alignment

17.2.2. Cleaning

Do not clean the product. Any residue cannot be easily removed by washing. Use a "no clean" soldering paste and do not clean the board after soldering.

- Do not clean the product with water. Capillary effects can draw water into the gap between the host PCB and the module, absorbing water underneath it. If water is trapped inside, it may short-circuit adjoining pads. The water may also destroy the label and ink-jet printed text on it.
- Cleaning processes using alcohol or other organic solvents may draw solder flux residues into the housing, which won't be detected in a post-wash inspection. The solvent may also destroy the label and ink-jet printed text on it.
- Do not use ultrasonic cleaning as it will permanently damage the part, particularly the crystal oscillators.

17.2.3. Other notations

- Conformal coating of the product will result in the loss of warranty. The RF shields will not protect the part from low-viscosity coatings.
- Do not attempt to improve the grounding by forming metal strips directly to the EMI covers or soldering on ground cables, as it may damage the part and will void the warranty.
- Always solder every pad to the host PCB even if some are unused, to improve the mechanical strength of the module.
- The part is sensitive to ultrasonic waves, as such do not use ultrasonic cleaning, welding or other processing. Any ultrasonic processing will void the warranty.

17.3. ESD handling

This product is highly sensitive to electrostatic discharge (ESD). As such, always use proper ESD precautions when handling. Make sure to handle the part properly throughout all stages of production, including on the host PCB where the module is installed. For ESD ratings, refer to the module series' maximum ESD section. For more information, refer to the relevant chapter 2. Failing to follow the aforementioned recommendations can result in severe damage to the part.

- the first contact point when handling the PCB is always between the local GND and the host PCB GND, unless there is a galvanic coupling between the local GND (for example work table) and the host PCB GND.
- Before assembling an antenna patch, connect the grounds.
- While handling the RF pin, avoid contact with any charged capacitors and be careful when contacting any materials that can develop charges (for example coaxial cable with around 50-80 pF/m, patch antenna with around 10 pF, soldering iron etc.)

- Do not touch any exposed area of the antenna to avoid electrostatic discharge. Do not let the antenna area be touched in a non ESD-safe manner.
- When soldering, use an ESD-safe soldering iron.

17.4. Safety recommendations

It is your duty to ensure that the product is allowed to be used in the destination country and within the required environment. Usage of the product can be dangerous and must be tested and verified by the end user. Be especially careful of:

- Use in areas with risk of explosion (for example oil refineries, gas stations).
- Use in areas such as airports, aircraft, hospitals, etc., where the product may interfere with other electronic components.

It is the customer's responsibility to ensure compliance with all applicable legal, regulatory and safety-related requirements as well as applicable environmental regulations. Disassembling the product is not allowed. Evidence of tampering will void the warranty.

- Compliance with the instructions in the product manual is recommended for correct product set-up.
- The product must be provided with a consolidated voltage source. The wiring must meet all applicable fire and security prevention standards.
- Handle with care. Avoid touching the pins as there could be ESD damage.

Be careful when working with any external components. When in doubt consult the technical documentation and relevant standards. Always use an antenna with the proper characteristics.

Since the module itself is not fused the voltage supply shall be fed from a limited power source according to EN 62368-1 class PS1.

Modules with high output power of up to 500mW, as for example the Thebe family, generate a high amount of warmth while transmitting. The manufacturer of the end device must take care of potentially neccessary actions for his application.

18. Physical dimensions

18.1. Dimensions

Dimensions
19 * 27.5 * 3 mm

Table 96: Dimensions

18.2. Weight

Weight
3 g

Table 97: Weight

18.3. Module drawing

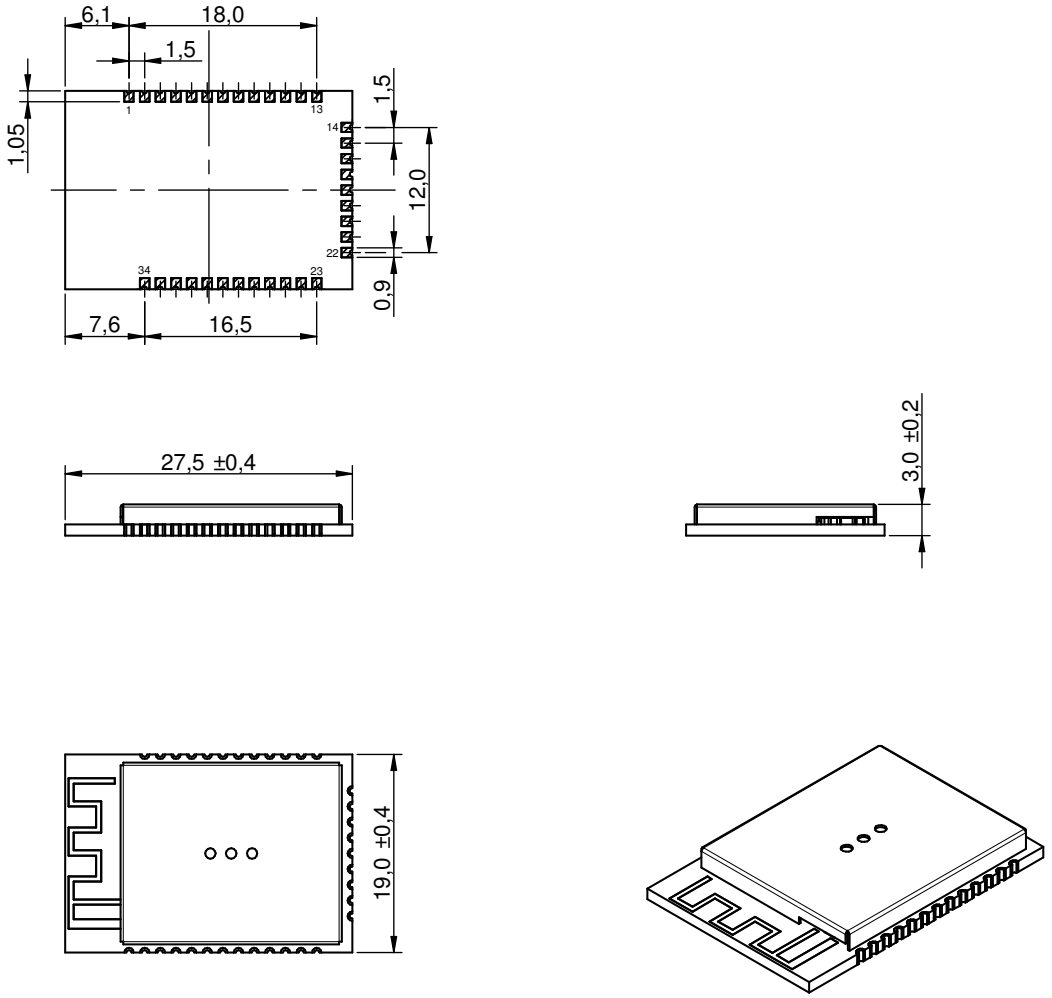


Figure 31: Module dimensions [mm]

18.4. Footprint

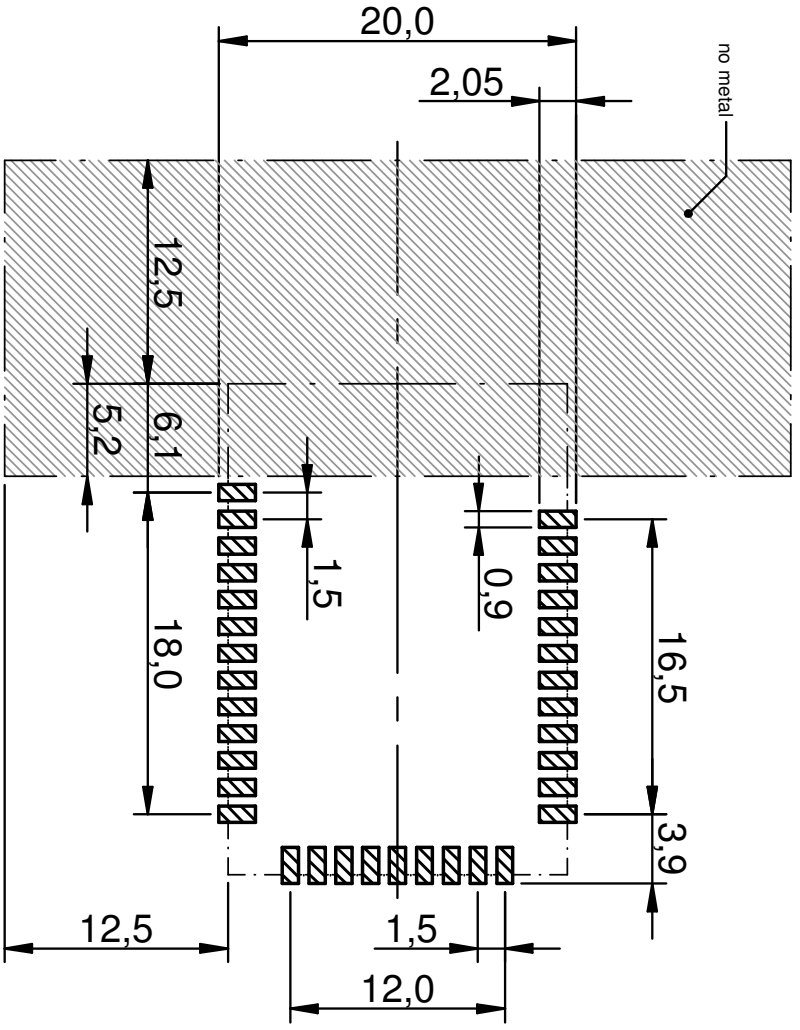


Figure 32: Footprint and dimensions [mm]

18.5. Antenna free area

To avoid influence and mismatching of the antenna the recommended free area around the antenna should be maintained. As rule of thumb a minimum distance of metal parts to the antenna of λ/10 should be kept (see figure 32). Even though metal parts would influence the characteristic of the antenna, but the direct influence and matching keep an acceptable level.

19. Marking

19.1. Lot number

The 15 digit lot number is printed in numerical digits as well as in form of a machine readable bar code. It is divided into 5 blocks as shown in the following picture and can be translated according to the following table.

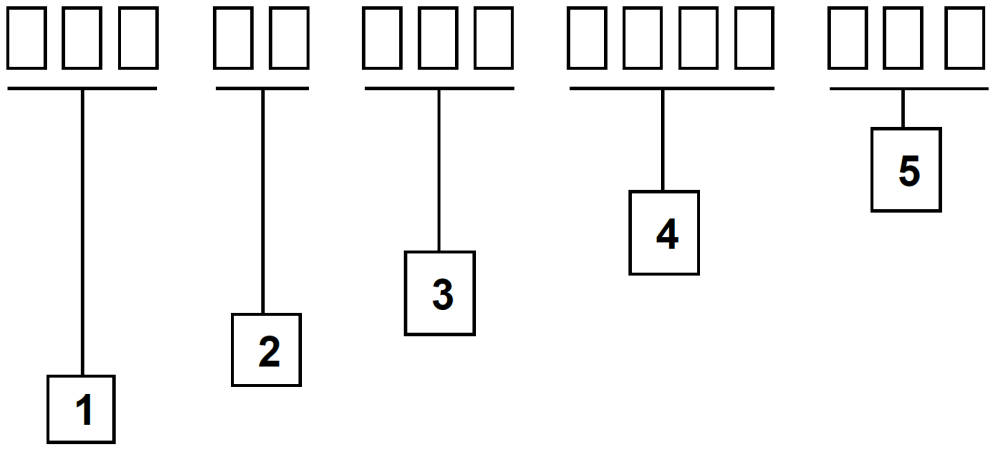


Figure 33: Lot number structure

Block	Information	Example(s)
1	eiSos internal, 3 digits	439
2	eiSos internal, 2 digits	01
3	Hardware version, 3 digits	V2.4 = 024, V12.2 = 122
4	Date code, 4 digits	1703 = week 03 in year 2017, 1816 = week 16 in year 2018
5	Firmware version, 3 digits	V3.2 = 302, V5.13 = 513

Table 98: Lot number details

As the user can perform a firmware update the printed lot number only shows the factory delivery state. The currently installed firmware can be requested from the module using the corresponding product specific command. The firmware version as well as the hardware version are restricted to show only major and minor version not the patch identifier.

19.2. General labeling information

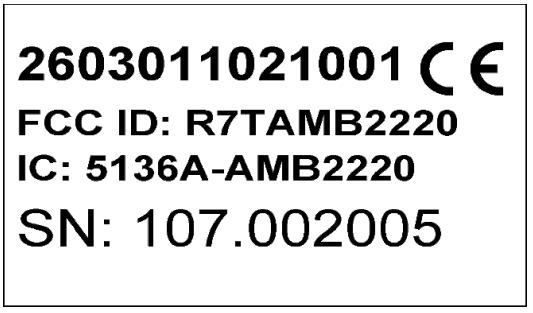
The module labels may include the following fields:

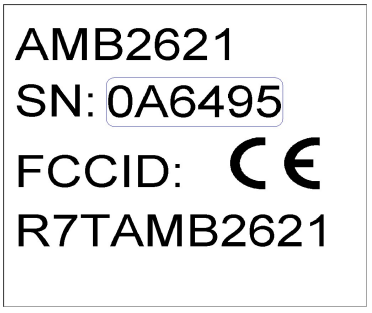
- Manufacturer identification WE, Würth Elektronik or Würth Elektronik eiSos
- WE Order Code and/or article alias
- Serial number or MAC address
- Certification identifiers (CE, FCC ID, IC, ARIB,...)
- Barcode or 2D code containing the serial number or MAC address

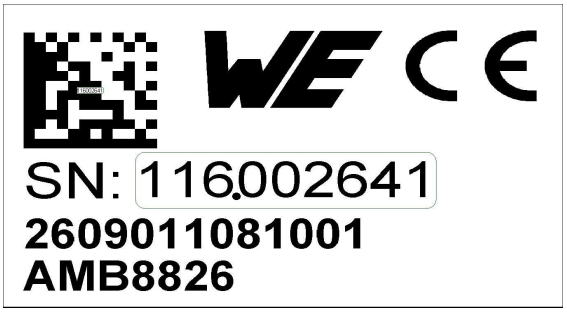
The serial number includes the product ID (PID) and an unique 6 digit number. The first 1 to 3 digits represent the PID, then the "." marks the start of the 6 digit counter to create a unique product marking.

In case of small labels, the 3 byte manufacturer identifier (0x0018DA) of the MAC address is not printed on the labels. The 3 byte counter printed on the label can be used with this 0018DA to produce the full MAC address by appending the counter after the manufacturer identifier.

19.2.1. Example labels of Würth Elektronik eiSos products







20. Information for Ex Protection

In case the end product should be used in Ex protection areas the following information can be used:

- The module itself is unfused.
- The maximum output power of the module is 18dBm.
- The total amount of capacitivity of all capacitors is 91.1µF.
- The total amount of inductivity of all inductors is 15.4µH.

21. Regulatory compliance information

21.1. Important notice EU

The use of RF frequencies is limited by national regulations. The Calypso has been designed to comply with the R&TTE directive 1999/5/EC and the RED directive 2014/53/EU of the European Union (EU).
The Calypso can be operated without notification and free of charge in the area of the European Union. However, according to the R&TTE / RED directive, restrictions (e.g. in terms of duty cycle or maximum allowed RF power) may apply.

21.2. EU Declaration of conformity

EU DECLARATION OF CONFORMITY

Radio equipment: **Calypso / 2610011025000**

The manufacturer: Würth Elektronik eiSos GmbH & Co. KG
Max-Eyth-Straße 1
74638 Waldenburg

This declaration of conformity is issued under the sole responsibility of the manufacturer.

Object of the declaration: Calypso / 2610011025000

The object of the declaration described above is in conformity with the relevant Union harmonization legislation: Directive 2014/53/EU and 2011/65/EU.
Following harmonized norms or technical specifications have been applied:

EN 300 328 V2.1.1 (2016-11)
EN 301 489-1 V2.2.0 (Draft)
EN 301 489-17 V3.2.0 (Draft)
EN 62479 : 2010
EN 62368-1:2014 + AC:2015 + A11:2017

i.A. G. Erlandt

Trier, 31th of October 2018
Place and date of issue

22. Important information

The following conditions apply to all goods within the wireless connectivity product range of Würth Elektronik eiSos GmbH & Co. KG :

22.1. General customer responsibility

Some goods within the product range of Würth Elektronik eiSos GmbH & Co. KG contain statements regarding general suitability for certain application areas. These statements about suitability are based on our knowledge and experience of typical requirements concerning the areas, serve as general guidance and cannot be estimated as binding statements about the suitability for a customer application. The responsibility for the applicability and use in a particular customer design is always solely within the authority of the customer. Due to this fact, it is up to the customer to evaluate, where appropriate to investigate and to decide whether the device with the specific product characteristics described in the product specification is valid and suitable for the respective customer application or not. Accordingly, the customer is cautioned to verify that the documentation is current before placing orders.

22.2. Customer responsibility related to specific, in particular safety-relevant applications

It has to be clearly pointed out that the possibility of a malfunction of electronic components or failure before the end of the usual lifetime cannot be completely eliminated in the current state of the art, even if the products are operated within the range of the specifications. The same statement is valid for all software and firmware parts contained in or used with or for products in the wireless connectivity product range of Würth Elektronik eiSos GmbH & Co. KG . In certain customer applications requiring a high level of safety and especially in customer applications in which the malfunction or failure of an electronic component could endanger human life or health, it must be ensured by most advanced technological aid of suitable design of the customer application that no injury or damage is caused to third parties in the event of malfunction or failure of an electronic component.

22.3. Best care and attention

Any product-specific datasheets, manuals, application notes, PCN's, warnings and cautions must be strictly observed in the most recent versions and matching to the products firmware revisions. This documents can be downloaded from the product specific sections on the wireless connectivity homepage.

22.4. Customer support for product specifications

Some products within the product range may contain substances, which are subject to restrictions in certain jurisdictions in order to serve specific technical requirements. Necessary information is available on request. In this case, the field sales engineer or the internal sales person in charge should be contacted who will be happy to support in this matter.

22.5. Product improvements

Due to constant product improvement, product specifications may change from time to time. As a standard reporting procedure of the Product Change Notification (PCN) according to the JEDEC-Standard, we inform about major changes in hard- or firmware. In case of further queries regarding the PCN, the field sales engineer, the internal sales person or the technical support team in charge should be contacted. The basic responsibility of the customer as per section 22.1 and 22.2 remains unaffected.

22.6. Product life cycle

Due to technical progress and economical evaluation we also reserve the right to discontinue production and delivery of products. As a standard reporting procedure of the Product Termination Notification (PTN) according to the JEDEC-Standard we will inform at an early stage about inevitable product discontinuance. According to this, we cannot ensure that all products within our product range will always be available. Therefore, it needs to be verified with the field sales engineer or the internal sales person in charge about the current product availability expectancy before or when the product for application design-in disposal is considered. The approach named above does not apply in the case of individual agreements deviating from the foregoing for customer-specific products.

22.7. Property rights

All the rights for contractual products produced by Würth Elektronik eiSos GmbH & Co. KG on the basis of ideas, development contracts as well as models or templates that are subject to copyright, patent or commercial protection supplied to the customer will remain with Würth Elektronik eiSos GmbH & Co. KG . Würth Elektronik eiSos GmbH & Co. KG does not warrant or represent that any license, either expressed or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, application, or process in which Würth Elektronik eiSos GmbH & Co. KG components or services are used.

22.8. General terms and conditions

Unless otherwise agreed in individual contracts, all orders are subject to the current version of the "General Terms and Conditions of Würth Elektronik eiSos Group", last version available at *www.we-online.com*.

23. Legal notice

23.1. Exclusion of liability

Würth Elektronik eiSos GmbH & Co. KG considers the information in this document to be correct at the time of publication. However, Würth Elektronik eiSos GmbH & Co. KG reserves the right to modify the information such as technical specifications or functions of its products or discontinue the production of these products or the support of one of these products without any written announcement or notification to customers. The customer must make sure that the information used corresponds to the latest published information. Würth Elektronik eiSos GmbH & Co. KG does not assume any liability for the use of its products. Würth Elektronik eiSos GmbH & Co. KG does not grant licenses for its patent rights or for any other of its intellectual property rights or third-party rights. Notwithstanding anything above, Würth Elektronik eiSos GmbH & Co. KG makes no representations and/or warranties of any kind for the provided information related to their accuracy, correctness, completeness, usage of the products and/or usability for customer applications. Information published by Würth Elektronik eiSos GmbH & Co. KG regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof.

23.2. Suitability in customer applications

The customer bears the responsibility for compliance of systems or units, in which Würth Elektronik eiSos GmbH & Co. KG products are integrated, with applicable legal regulations. Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of Würth Elektronik eiSos GmbH & Co. KG components in its applications, notwithstanding any applications-related in-formation or support that may be provided by Würth Elektronik eiSos GmbH & Co. KG . Customer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences lessen the likelihood of failures that might cause harm and take appropriate remedial actions. The customer will fully indemnify Würth Elektronik eiSos GmbH & Co. KG and its representatives against any damages arising out of the use of any Würth Elektronik eiSos GmbH & Co. KG components in safety-critical applications.

23.3. Trademarks

AMBER wireless is a registered trademark of Würth Elektronik eiSos GmbH & Co. KG . All other trademarks, registered trademarks, and product names are the exclusive property of the respective owners.

23.4. Usage restriction

Würth Elektronik eiSos GmbH & Co. KG products have been designed and developed for usage in general electronic equipment only. This product is not authorized for use in equipment where a higher safety standard and reliability standard is especially required or where a failure of the product is reasonably expected to cause severe personal injury or death, unless the parties have executed an agreement specifically governing such use. Moreover,

Würth Elektronik eiSos GmbH & Co. KG products are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation (automotive control, train control, ship control), transportation signal, disaster prevention, medical, public information network etc. . Würth Elektronik eiSos GmbH & Co. KG must be informed about the intent of such usage before the design-in stage. In addition, sufficient reliability evaluation checks for safety must be performed on every electronic component, which is used in electrical circuits that require high safety and reliability function or performance. By using Würth Elektronik eiSos GmbH & Co. KG products, the customer agrees to these terms and conditions.

24. License agreement for Würth Elektronik eiSos GmbH & Co. KG connectivity product firmware and software

Agreement between You and Würth Elektronik eiSos GmbH & Co. KG

The following terms of this license agreement for the usage of the Würth Elektronik eiSos GmbH & Co. KG wireless connectivity product firmware are a legal agreement between you and Würth Elektronik eiSos GmbH & Co. KG and/or its subsidiaries and affiliates (collectively, "Würth Elektronik eiSos "). You hereby agree that this license agreement is applicable to the product and the incorporated software and firmware (collectively, "Firmware") made available by Würth Elektronik eiSos in any form, including but not limited to binary, executable or source code form.
The Firmware included in any Würth Elektronik eiSos wireless connectivity product is purchased to you on the condition that you accept the terms and conditions of this license agreement. You agree to comply with all provisions under this license agreement.

24.1. Limited license

Würth Elektronik eiSos hereby grants you a limited, non-exclusive, non-transferable and royalty-free license to use the Firmware under the conditions that will be set forth in this license agreement. You are free to use the provided Firmware only in connection with one of the products from Würth Elektronik eiSos to the extent described in this license agreement.
You are not entitled to change or alter the provided Firmware.
You are not entitled to transfer the Firmware in any form to third parties without prior written consent of Würth Elektronik eiSos .
You are not allowed to reproduce, translate, reverse engineer, read out, decompile, disassemble or create derivative works of the incorporated Firmware in whole or in part.
No more extensive rights to use and exploit the Firmware granted to you.

24.2. Usage and obligations

The responsibility for the applicability and use of the Würth Elektronik eiSos wireless connectivity product with the incorporated Firmware in a particular customer design is always solely within the authority of the customer. Due to this fact, it is up to you to evaluate and investigate, where appropriate, and to decide whether the device with the specific product characteristics described in the product specification is valid and suitable for your respective application or not.
You are responsible for using the Würth Elektronik eiSos Product with the incorporated Firmware in compliance with all applicable product liability and product safety laws. You acknowledge to minimize the risk of loss and harm to individuals and bear the risk for failure leading to personal injury or death due to your usage of the product.
Würth Elektronik eiSos ' products with the incorporated Firmware are not authorized for use in safety-critical applications, or where a failure of the product is reasonably expected to cause severe personal injury or death. Moreover, Würth Elektronik eiSos ' products with the incorporated Firmware are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation (automotive control, train

control, ship control), transportation signal, disaster prevention, medical, public information network etc. You shall inform Würth Elektronik eiSos about the intent of such usage before design-in stage. In certain customer applications requiring a very high level of safety and in which the malfunction or failure of an electronic component could endanger human life or health, you must ensure to have all necessary expertise in the safety and regulatory ramifications of your applications. You acknowledge and agree that you are solely responsible for all legal, regulatory and safety-related requirements concerning your products and any use of Würth Elektronik eiSos ' products with the incorporated Firmware in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by Würth Elektronik eiSos . YOU SHALL INDEMNIFY WÜRTH ELEKTRONIK EISOS AGAINST ANY DAMAGES ARISING OUT OF THE USE OF WÜRTH ELEKTRONIK EISOS' PRODUCTS WITH THE INCORPORATED FIRMWARE IN SUCH SAFETY-CRITICAL APPLICATIONS.

24.3. Ownership

The incorporated Firmware created by Würth Elektronik eiSos is and will remain the exclusive property of Würth Elektronik eiSos .

24.4. Firmware update(s)

You have the opportunity to request the current and actual firmware for a bought wireless connectivity Product within the time of warranty. However, Würth Elektronik eiSos has no obligation to update a modules firmware in their production facilities, but can offer this as a service on request. The upload of firmware updates falls within your responsibility, e.g. via ACC or another software for firmware updates. Firmware updates will not be communicated automatically. It is within your responsibility to check the current version of a firmware in the latest version of the product manual on our website. The revision table in the product manual provides all necessary information about firmware updates. There is no right to be provided with binary files, so called "firmware images", those could be flashed through JTAG, SWD, Spi-Bi-Wire, SPI or similar interfaces.

24.5. Disclaimer of warranty

THE FIRMWARE IS PROVIDED "AS IS". YOU ACKNOWLEDGE THAT WÜRTH ELEKTRONIK EISOS MAKES NO REPRESENTATIONS AND WARRANTIES OF ANY KIND RELATED TO, BUT NOT LIMITED TO THE NON-INFRINGEMENT OF THIRD PARTIES' INTELLECTUAL PROPERTY RIGHTS OR THE MERCHANTABILITY OR FITNESS FOR YOUR INTENDED PURPOSE OR USAGE. WÜRTH ELEKTRONIK EISOS DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT RELATING TO ANY COMBINATION, MACHINE, OR PROCESS IN WHICH THE WÜRTH ELEKTRONIK EISOS' PRODUCT WITH THE INCORPORATED FIRMWARE IS USED. INFORMATION PUBLISHED BY WÜRTH ELEKTRONIK EISOS REGARDING THIRD-PARTY PRODUCTS OR SERVICES DOES NOT CONSTITUTE A LICENSE FROM WÜRTH ELEKTRONIK EISOS TO USE SUCH PRODUCTS OR SERVICES OR A WARRANTY OR ENDORSEMENT THEREOF.

24.6. Limitation of liability

Any liability not expressly provided by Würth Elektronik eiSos shall be disclaimed.
You agree to hold us harmless from any third-party claims related to your usage of the Würth Elektronik eiSos ' products with the incorporated Firmware. Würth Elektronik eiSos disclaims any liability for any alteration, development created by you or your customers as well as for any combination with other products.

24.7. Applicable law and jurisdiction

Applicable law to this license agreement shall be the laws of the Federal Republic of Germany. Any dispute, claim or controversy arising out of or relating to this license agreement shall be resolved and finally settled by the court competent for the location of Würth Elektronik eiSos ' registered office.

24.8. Severability clause

If a provision of this license agreement is or becomes invalid, unenforceable or null and void, this shall not affect the remaining provisions of the agreement. The parties shall replace any such provisions with new valid provisions that most closely approximate the purpose of the agreement.

24.9. Miscellaneous

This license agreement constitutes the entire understanding and merges all prior discussions between the parties relating to this license agreement.
No ancillary verbal agreements have been made and no such agreements shall be valid. Any additions and amendments to this license agreement shall require the written form in order to be binding.
We recommend you to be updated about the status of new firmware, which is available on our website or in our data sheet, and to implement new firmware in your device where appropriate. In case only firmware is provided, we expressly exclude the automatic receipt of PCN information. Thus, new firmware will also not be provided automatically.
By ordering a wireless connectivity Product, you accept this license agreement in all terms.

A. Wi-Fi certificate

The section contains the Wi-Fi certificate for Calypso .

Wi-Fi CERTIFIED™ Interoperability Certificate

This certificate lists the features that have successfully completed Wi-Fi Alliance interoperability testing.
Learn more: www.wi-fi.org/certification/programs

Certification ID: WFA81685 **Page 1 of 2**

Date of Last Certification	January 22, 2019
Company	Wurth Elektronik eiSos GmbH & CO. KG
Product	Calypso
Model Number	261001102500x
Product Identifier(s)	AMB5201 (Other)
Category	Other
Subcategory	Industrial (communications & input)
Hardware Version	Product: 2.1, Wi-Fi Component: 3.1
Firmware Version	Product: 1.0.0, Wi-Fi Component: 31.2.0.0.0
Operating System	ThreadX, version: 2.20.00.10
Frequency Band(s)	2.4 GHz

Summary of Certifications

CLASSIFICATION	PROGRAM
Connectivity	Wi-Fi CERTIFIED™ b, g, n
	WPA™ – Enterprise, Personal
	WPA2™ – Enterprise, Personal
	Wi-Fi Direct®
Optimization	WMM®
Access	Wi-Fi Protected Setup™

Wi-Fi CERTIFIED™ Interoperability Certificate

Certification ID: WFA81685 **Page 2 of 2**

Security

WPA™ – Enterprise, Personal
WPA2™ – Enterprise, Personal
EAP Type(s)
EAP-TLS
EAP-TTLS/MSCHAPv2
PEAPv0/EAP-MSCHAPv2
EAP-FAST

Wi-Fi CERTIFIED™ b

Wi-Fi CERTIFIED™ g

Wi-Fi CERTIFIED™ n

2.4 GHz
1 Spatial Stream 2.4 GHz
Short Guard Interval
Greenfield Preamble

Wi-Fi Direct®

2.4 GHz

WMM®

Wi-Fi Protected Setup™

2.4 GHz
PIN
Push-Button (PBC)

B. Error codes

The section briefly describes the meaning of error codes returned by Calypso in response to commands.

B.1. Disconnection reason codes

```
/* WLAN Disconnect Reason Codes */
 SL_WLAN_DISCONNECT_UNSPECIFIED (1)
 SL_WLAN_DISCONNECT_AUTH_NO_LONGER_VALID (2)
 SL_WLAN_DISCONNECT_DEAUTH_SENDING_STA_LEAVING (3)
 SL_WLAN_DISCONNECT_INACTIVITY (4)
 SL_WLAN_DISCONNECT_TOO_MANY_STA (5)
 SL_WLAN_DISCONNECT_FRAME_FROM_NONAUTH_STA (6)
 SL_WLAN_DISCONNECT_FRAME_FROM_NONASSOC_STA (7)
 SL_WLAN_DISCONNECT_DISS_SENDING_STA_LEAVING (8)
 SL_WLAN_DISCONNECT_STA_NOT_AUTH (9)
 SL_WLAN_DISCONNECT_POWER_CAPABILITY_INVALID (10)
 SL_WLAN_DISCONNECT_SUPPORTED_CHANNELS_INVALID (11)
 SL_WLAN_DISCONNECT_INVALID_IE (13)
 SL_WLAN_DISCONNECT_MIC_FAILURE (14)
 SL_WLAN_DISCONNECT_FOURWAY_HANDSHAKE_TIMEOUT (15)
 SL_WLAN_DISCONNECT_GROUPKEY_HANDSHAKE_TIMEOUT (16)
 SL_WLAN_DISCONNECT_REASSOC_INVALID_IE (17)
 SL_WLAN_DISCONNECT_INVALID_GROUP_CIPHER (18)
 SL_WLAN_DISCONNECT_INVALID_PAIRWISE_CIPHER (19)
 SL_WLAN_DISCONNECT_INVALID_AKMP (20)
 SL_WLAN_DISCONNECT_UNSUPPORTED_RSN_VERSION (21)
 SL_WLAN_DISCONNECT_INVALID_RSN_CAPABILITIES (22)
 SL_WLAN_DISCONNECT_IEEE_802_1X_AUTHENTICATION_FAILED (23)
 SL_WLAN_DISCONNECT_CIPHER_SUITE_REJECTED (24)
 SL_WLAN_DISCONNECT_DISASSOC_QOS (32)
 SL_WLAN_DISCONNECT_DISASSOC_QOS_BANDWIDTH (33)
 SL_WLAN_DISCONNECT_DISASSOC_EXCESSIVE_ACK_PENDING (34)
 SL_WLAN_DISCONNECT_DISASSOC_TXOP_LIMIT (35)
 SL_WLAN_DISCONNECT_STA_LEAVING (36)
 SL_WLAN_DISCONNECT_STA_DECLINED (37)
 SL_WLAN_DISCONNECT_STA_UNKNOWN_BA (38)
 SL_WLAN_DISCONNECT_STA_TIMEOUT (39)
 SL_WLAN_DISCONNECT_STA_UNSUPPORTED_CIPHER_SUITE (40)
 SL_WLAN_DISCONNECT_USER_INITIATED (200)
 SL_WLAN_DISCONNECT_AUTH_TIMEOUT (202)
 SL_WLAN_DISCONNECT_ASSOC_TIMEOUT (203)
 SL_WLAN_DISCONNECT_SECURITY_FAILURE (204)
 SL_WLAN_DISCONNECT_WHILE_CONNNECTING (208)
 SL_WLAN_DISCONNECT_MISSING_CERT (209)
 SL_WLAN_DISCONNECT_CERTIFICATE_EXPIRED (210)
```

B.2. Socket error codes

```
/* BSD SOCKET ERRORS CODES */

SL_ERROR_BSD_SOC_ERROR (-1L) /* Failure */
SL_ERROR_BSD_EINTR (-4L) /* Interrupted system call */
SL_ERROR_BSD_E2BIG (-7L) /* length too big */
SL_ERROR_BSD_INEXE (-8L) /* socket command in execution */
SL_ERROR_BSD_EBADF (-9L) /* Bad file number */
```

```
SL_ERROR_BSD_ENSOCK (-10L) /* The system limit on the total number of open
    socket, has been reached */
SL_ERROR_BSD_EAGAIN (-11L) /* Try again */
SL_ERROR_BSD_EWOULDBLOCK  SL_ERROR_BSD_EAGAIN
SL_ERROR_BSD_ENOMEM (-12L) /* Out of memory */
SL_ERROR_BSD_EACCES (-13L) /* Permission denied */
SL_ERROR_BSD_EFAULT (-14L) /* Bad address */
SL_ERROR_BSD_ECLOSE (-15L) /* close socket operation failed to transmit all
    queued packets */
SL_ERROR_BSD_EALREADY_ENABLED (-21L) /* Transceiver - Transceiver already ON.
    there could be only one */
SL_ERROR_BSD_EINVAL (-22L) /* Invalid argument */
SL_ERROR_BSD_EAUTO_CONNECT_OR_CONNECTING (-69L) /* Transceiver - During
    connection, connected or auto mode started */
SL_ERROR_BSD_CONNECTION_PENDING (-72L) /* Transceiver - Device is connected,
    disconnect first to open transceiver */
SL_ERROR_BSD_EUNSUPPORTED_ROLE (-86L) /* Transceiver - Trying to start when WLAN
     role is AP or P2P GO */
SL_ERROR_BSD_EDESTADDRREQ (-89L) /* Destination address required */
SL_ERROR_BSD_EPROTOTYPE (-91L) /* Protocol wrong type for socket */
SL_ERROR_BSD_ENOPROTOOPT (-92L) /* Protocol not available */
SL_ERROR_BSD_EPROTONOSUPPORT (-93L) /* Protocol not supported */
SL_ERROR_BSD_ESOCKTNOSUPPORT (-94L) /* Socket type not supported */
SL_ERROR_BSD_EOPNOTSUPP (-95L) /* Operation not supported on transport endpoint
    */
SL_ERROR_BSD_EAFNOSUPPORT (-97L) /* Address family not supported by protocol */
SL_ERROR_BSD_EADDRINUSE (-98L) /* Address already in use */
SL_ERROR_BSD_EADDRNOTAVAIL (-99L) /* Cannot assign requested address */
SL_ERROR_BSD_ENETUNREACH (-101L) /* Network is unreachable */
SL_ERROR_BSD_ENOBUFS (-105L) /* No buffer space available */
SL_ERROR_BSD_EOBUFF  SL_ENOBUFS
SL_ERROR_BSD_EISCONN (-106L) /* Transport endpoint is already connected */
SL_ERROR_BSD_ENOTCONN (-107L) /* Transport endpoint is not connected */
SL_ERROR_BSD_ETIMEDOUT (-110L) /* Connection timed out */
SL_ERROR_BSD_ECONNREFUSED (-111L) /* Connection refused */
SL_ERROR_BSD_EALREADY (-114L) /* Non blocking connect in progress, try again */
```

B.3. Secure socket error codes

```
/* ssl tls security start with -300 offset */
SL_ERROR_BSD_ESEC_CLOSE_NOTIFY (-300L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_UNEXPECTED_MESSAGE (-310L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_BAD_RECORD_MAC (-320L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_DECRYPTION_FAILED (-321L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_RECORD_OVERFLOW (-322L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_DECOMPRESSION_FAILURE (-330L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_HANDSHAKE_FAILURE (-340L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_NO_CERTIFICATE (-341L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_BAD_CERTIFICATE (-342L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_UNSUPPORTED_CERTIFICATE (-343L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_CERTIFICATE_REVOKED (-344L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_CERTIFICATE_EXPIRED (-345L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_CERTIFICATE_UNKNOWN (-346L) /* ssl/tls alerts */

SL_ERROR_BSD_ESEC_ILLEGAL_PARAMETER (-347L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_ACCESS_DENIED (-349L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_DECODE_ERROR (-350L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_DECRYPT_ERROR1 (-351L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_EXPORT_RESTRICTION (-360L) /* ssl/tls alerts */
```

```
SL_ERROR_BSD_ESEC_PROTOCOL_VERSION (-370L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_INSUFFICIENT_SECURITY (-371L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_INTERNAL_ERROR (-380L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_USER_CANCELLED (-390L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_NO_RENEGOTIATION (-400L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_UNSUPPORTED_EXTENSION (-410L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_CERTIFICATE_UNOBTAINABLE (-411L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_UNRECOGNIZED_NAME (-412L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_BAD_CERTIFICATE_STATUS_RESPONSE (-413L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_BAD_CERTIFICATE_HASH_VALUE (-414L) /* ssl/tls alerts */
/* propriety secure */
SL_ERROR_BSD_ESECGENERAL (-450L) /* error secure level general error */
SL_ERROR_BSD_ESECDECRYPT (-451L) /* error secure level, decrypt recv packet fail
    */
SL_ERROR_BSD_ESECCLOSED (-452L) /* secure layrer is closed by other size , tcp
    is still connected */
SL_ERROR_BSD_ESECSNOVERIFY (-453L) /* Connected without server verification */
SL_ERROR_BSD_ESECNOCAFILE (-454L) /* error secure level CA file not found*/
SL_ERROR_BSD_ESECMEMORY (-455L) /* error secure level No memory space available
    */
SL_ERROR_BSD_ESECBADCAFILE (-456L) /* error secure level bad CA file */
SL_ERROR_BSD_ESECBADCERTFILE (-457L) /* error secure level bad Certificate file
    */
SL_ERROR_BSD_ESECBADPRIVATEFILE (-458L) /* error secure level bad private file
    */
SL_ERROR_BSD_ESECBADDHFILE (-459L) /* error secure level bad DH file */
SL_ERROR_BSD_ESECT00MANYSSLOPENED (-460L) /* MAX SSL Sockets are opened */
SL_ERROR_BSD_ESECDATEERROR (-461L) /* connected with certificate date
    verification error */
SL_ERROR_BSD_ESECHANDSHAKETIMEDOUT (-462L) /* connection timed out due to
    handshake time */
SL_ERROR_BSD_ESECTXBUFFERNOTEMPTY (-463L) /* cannot start ssl connection while
    send buffer is full */
SL_ERROR_BSD_ESECRXBUFFERNOTEMPTY (-464L) /* cannot start ssl connection while
    recv buffer is full */
SL_ERROR_BSD_ESECSSLDURINGHANDSHAKE (-465L) /* cannot use while in hanshaking */
SL_ERROR_BSD_ESECNOTALLOWEDWHENLISTENING (-466L) /* the operation is not allowed
     when listening , do before listen */
SL_ERROR_BSD_ESECCERTIFICATEREVOKED (-467L) /* connected but on of the
    certificates in the chain is revoked */
SL_ERROR_BSD_ESECUNKNOWNROOTCA (-468L) /* connected but the root CA used to
    validate the peer is unknown */
SL_ERROR_BSD_ESECWRONGPEERCERT (-469L) /* wrong peer cert (server cert) was
    received while trying to connect to server */
SL_ERROR_BSD_ESECTCPDISCONNECTEDUNCOMPLETERECORD (-470L) /* the other side
    disconnected the TCP layer and didn't send the whole ssl record */

SL_ERROR_BSD_ESEC_BUFFER_E (-632L) /* output buffer too small or input too large
    */
SL_ERROR_BSD_ESEC_ALGO_ID_E (-633L) /* setting algo id error */
SL_ERROR_BSD_ESEC_PUBLIC_KEY_E (-634L) /* setting public key error */
SL_ERROR_BSD_ESEC_DATE_E (-635L) /* setting date validity error */
SL_ERROR_BSD_ESEC_SUBJECT_E (-636L) /* setting subject name error */
SL_ERROR_BSD_ESEC_ISSUER_E (-637L) /* setting issuer name error */
SL_ERROR_BSD_ESEC_CA_TRUE_E (-638L) /* setting CA basic constraint true error */
SL_ERROR_BSD_ESEC_EXTENSIONS_E (-639L) /* setting extensions error */
SL_ERROR_BSD_ESEC_ASN_PARSE_E (-640L) /* ASN parsing error, invalid input */
SL_ERROR_BSD_ESEC_ASN_VERSION_E (-641L) /* ASN version error, invalid number */
SL_ERROR_BSD_ESEC_ASN_GETINT_E (-642L) /* ASN get big int error, invalid data */
```

```
SL_ERROR_BSD_ESEC_ASN_RSA_KEY_E (-643L) /* ASN key init error, invalid input */
SL_ERROR_BSD_ESEC_ASN_OBJECT_ID_E (-644L) /* ASN object id error, invalid id */
SL_ERROR_BSD_ESEC_ASN_TAG_NULL_E (-645L) /* ASN tag error, not null */
SL_ERROR_BSD_ESEC_ASN_EXPECT_0_E (-646L) /* ASN expect error, not zero */
SL_ERROR_BSD_ESEC_ASN_BITSTR_E (-647L) /* ASN bit string error, wrong id */
SL_ERROR_BSD_ESEC_ASN_UNKNOWN_OID_E (-648L) /* ASN oid error, unknown sum id */
SL_ERROR_BSD_ESEC_ASN_DATE_SZ_E (-649L) /* ASN date error, bad size */
SL_ERROR_BSD_ESEC_ASN_BEFORE_DATE_E (-650L) /* ASN date error, current date
    before */
SL_ERROR_BSD_ESEC_ASN_AFTER_DATE_E (-651L) /* ASN date error, current date after
     */
SL_ERROR_BSD_ESEC_ASN_SIG_OID_E (-652L) /* ASN signature error, mismatched oid
    */
SL_ERROR_BSD_ESEC_ASN_TIME_E (-653L) /* ASN time error, unknown time type */
SL_ERROR_BSD_ESEC_ASN_INPUT_E (-654L) /* ASN input error, not enough data */
SL_ERROR_BSD_ESEC_ASN_SIG_CONFIRM_E (-655L) /* ASN sig error, confirm failure */
SL_ERROR_BSD_ESEC_ASN_SIG_HASH_E (-656L) /* ASN sig error, unsupported hash type
     */
SL_ERROR_BSD_ESEC_ASN_SIG_KEY_E (-657L) /* ASN sig error, unsupported key type
    */
SL_ERROR_BSD_ESEC_ASN_DH_KEY_E (-658L) /* ASN key init error, invalid input */
SL_ERROR_BSD_ESEC_ASN_NTRU_KEY_E (-659L) /* ASN ntru key decode error, invalid
    input */
SL_ERROR_BSD_ESEC_ASN_CRIT_EXT_E (-660L) /* ASN unsupported critical extension
    */
SL_ERROR_BSD_ESEC_ECC_BAD_ARG_E (-670L) /* ECC input argument of wrong type */
SL_ERROR_BSD_ESEC_ASN_ECC_KEY_E (-671L) /* ASN ECC bad input */
SL_ERROR_BSD_ESEC_ECC_CURVE_OID_E (-672L) /* Unsupported ECC OID curve type */
SL_ERROR_BSD_ESEC_BAD_FUNC_ARG (-673L) /* Bad function argument provided */
SL_ERROR_BSD_ESEC_NOT_COMPILED_IN (-674L) /* Feature not compiled in */
SL_ERROR_BSD_ESEC_UNICODE_SIZE_E (-675L) /* Unicode password too big */
SL_ERROR_BSD_ESEC_NO_PASSWORD (-676L) /* no password provided by user */
SL_ERROR_BSD_ESEC_ALT_NAME_E (-677L) /* alt name size problem, too big */
SL_ERROR_BSD_ESEC_ASN_NO_SIGNER_E (-688L) /* ASN no signer to confirm failure */
SL_ERROR_BSD_ESEC_ASN_CRL_CONFIRM_E (-689L) /* ASN CRL signature confirm failure
     */
SL_ERROR_BSD_ESEC_ASN_CRL_NO_SIGNER_E (-690L) /* ASN CRL no signer to confirm
    failure */
SL_ERROR_BSD_ESEC_ASN_OCSP_CONFIRM_E (-691L) /* ASN OCSP signature confirm
    failure */
SL_ERROR_BSD_ESEC_VERIFY_FINISHED_ERROR (-704L) /* verify problem on finished */
SL_ERROR_BSD_ESEC_VERIFY_MAC_ERROR (-705L) /* verify mac problem */
SL_ERROR_BSD_ESEC_PARSE_ERROR (-706L) /* parse error on header */
SL_ERROR_BSD_ESEC_UNKNOWN_HANDSHAKE_TYPE (-707L) /* weird handshake type */
SL_ERROR_BSD_ESEC_SOCKET_ERROR_E (-708L) /* error state on socket */
SL_ERROR_BSD_ESEC_SOCKET_NODATA (-709L) /* expected data, not there */
SL_ERROR_BSD_ESEC_INCOMPLETE_DATA (-710L) /* don't have enough data to complete
    task */
SL_ERROR_BSD_ESEC_UNKNOWN_RECORD_TYPE (-711L) /* unknown type in record hdr */
SL_ERROR_BSD_ESEC_INNER_DECRYPT_ERROR (-712L) /* error during decryption */
SL_ERROR_BSD_ESEC_FATAL_ERROR (-713L) /* recvd alert fatal error */
SL_ERROR_BSD_ESEC_ENCRYPT_ERROR (-714L) /* error during encryption */
SL_ERROR_BSD_ESEC_FREAD_ERROR (-715L) /* fread problem */
SL_ERROR_BSD_ESEC_NO_PEER_KEY (-716L) /* need peer's key */
SL_ERROR_BSD_ESEC_NO_PRIVATE_KEY (-717L) /* need the private key */
SL_ERROR_BSD_ESEC_RSA_PRIVATE_ERROR (-718L) /* error during rsa priv op */
SL_ERROR_BSD_ESEC_NO_DH_PARAMS (-719L) /* server missing DH params */
SL_ERROR_BSD_ESEC_BUILD_MSG_ERROR (-720L) /* build message failure */
SL_ERROR_BSD_ESEC_BAD_HELLO (-721L) /* client hello malformed */
```

```
SL_ERROR_BSD_ESEC_DOMAIN_NAME_MISMATCH (-722L) /* peer subject name mismatch */
SL_ERROR_BSD_ESEC_WANT_READ (-723L) /* want read, call again */
SL_ERROR_BSD_ESEC_NOT_READY_ERROR (-724L) /* handshake layer not ready */
SL_ERROR_BSD_ESEC_PMS_VERSION_ERROR (-725L) /* pre m secret version error */
SL_ERROR_BSD_ESEC_WANT_WRITE (-727L) /* want write, call again */
SL_ERROR_BSD_ESEC_BUFFER_ERROR (-728L) /* malformed buffer input */
SL_ERROR_BSD_ESEC_VERIFY_CERT_ERROR (-729L) /* verify cert error */
SL_ERROR_BSD_ESEC_VERIFY_SIGN_ERROR (-730L) /* verify sign error */
SL_ERROR_BSD_ESEC_LENGTH_ERROR (-741L) /* record layer length error */
SL_ERROR_BSD_ESEC_PEER_KEY_ERROR (-742L) /* can't decode peer key */
SL_ERROR_BSD_ESEC_ZERO_RETURN (-743L) /* peer sent close notify */
SL_ERROR_BSD_ESEC_SIDE_ERROR (-744L) /* wrong client/server type */
SL_ERROR_BSD_ESEC_NO_PEER_CERT (-745L) /* peer didn't send key */
SL_ERROR_BSD_ESEC_ECC_CURVETYPE_ERROR (-750L) /* Bad ECC Curve Type */
SL_ERROR_BSD_ESEC_ECC_CURVE_ERROR (-751L) /* Bad ECC Curve */
SL_ERROR_BSD_ESEC_ECC_PEERKEY_ERROR (-752L) /* Bad Peer ECC Key */
SL_ERROR_BSD_ESEC_ECC_MAKEKEY_ERROR (-753L) /* Bad Make ECC Key */
SL_ERROR_BSD_ESEC_ECC_EXPORT_ERROR (-754L) /* Bad ECC Export Key */
SL_ERROR_BSD_ESEC_ECC_SHARED_ERROR (-755L) /* Bad ECC Shared Secret */
SL_ERROR_BSD_ESEC_NOT_CA_ERROR (-757L) /* Not a CA cert error */
SL_ERROR_BSD_ESEC_BAD_PATH_ERROR (-758L) /* Bad path for opendir */
SL_ERROR_BSD_ESEC_BAD_CERT_MANAGER_ERROR (-759L) /* Bad Cert Manager */
SL_ERROR_BSD_ESEC_OCSP_CERT_REVOKED (-760L) /* OCSP Certificate revoked */
SL_ERROR_BSD_ESEC_CRL_CERT_REVOKED (-761L) /* CRL Certificate revoked */
SL_ERROR_BSD_ESEC_CRL_MISSING (-762L) /* CRL Not loaded */
SL_ERROR_BSD_ESEC_MONITOR_RUNNING_E (-763L) /* CRL Monitor already running */
SL_ERROR_BSD_ESEC_THREAD_CREATE_E (-764L) /* Thread Create Error */
SL_ERROR_BSD_ESEC_OCSP_NEED_URL (-765L) /* OCSP need an URL for lookup */
SL_ERROR_BSD_ESEC_OCSP_CERT_UNKNOWN (-766L) /* OCSP responder doesn't know */
SL_ERROR_BSD_ESEC_OCSP_LOOKUP_FAIL (-767L) /* OCSP lookup not successful */
SL_ERROR_BSD_ESEC_MAX_CHAIN_ERROR (-768L) /* max chain depth exceeded */
SL_ERROR_BSD_ESEC_NO_PEER_VERIFY (-778L) /* Need peer cert verify Error */
SL_ERROR_BSD_ESEC_UNSUPPORTED_SUITE (-790L) /* unsupported cipher suite */
SL_ERROR_BSD_ESEC_MATCH_SUITE_ERROR (-791L) /* can't match cipher suite */
```

B.4. WLAN error codes

```
/* WLAN ERRORS CODES */
 SL_ERROR_WLAN_KEY_ERROR (-2049L)
 SL_ERROR_WLAN_INVALID_ROLE (-2050L)
 SL_ERROR_WLAN_PREFERRED_NETWORKS_FILE_LOAD_FAILED (-2051L)
 SL_ERROR_WLAN_CANNOT_CONFIG_SCAN_DURING_PROVISIONING (-2052L)
 SL_ERROR_WLAN_INVALID_SECURITY_TYPE (-2054L)
 SL_ERROR_WLAN_PASSPHRASE_TOO_LONG (-2055L)
 SL_ERROR_WLAN_EAP_WRONG_METHOD (-2057L)
 SL_ERROR_WLAN_PASSWORD_ERROR (-2058L)
 SL_ERROR_WLAN_EAP_ANONYMOUS_LEN_ERROR (-2059L)
 SL_ERROR_WLAN_SSID_LEN_ERROR (-2060L)
 SL_ERROR_WLAN_USER_ID_LEN_ERROR (-2061L)
 SL_ERROR_WLAN_PREFERRED_NETWORK_LIST_FULL (-2062L)
 SL_ERROR_WLAN_PREFERRED_NETWORKS_FILE_WRITE_FAILED (-2063L)
 SL_ERROR_WLAN_ILLEGAL_WEP_KEY_INDEX (-2064L)
 SL_ERROR_WLAN_INVALID_DWELL_TIME_VALUES (-2065L)
 SL_ERROR_WLAN_INVALID_POLICY_TYPE (-2066L)
 SL_ERROR_WLAN_PM_POLICY_INVALID_OPTION (-2067L)
 SL_ERROR_WLAN_PM_POLICY_INVALID_PARAMS (-2068L)
 SL_ERROR_WLAN_WIFI_NOT_CONNECTED (-2069L)
 SL_ERROR_WLAN_ILLEGAL_CHANNEL (-2070L)
 SL_ERROR_WLAN_WIFI_ALREADY_DISCONNECTED (-2071L)
```

```
 SL_ERROR_WLAN_TRANSCEIVER_ENABLED (-2072L)
 SL_ERROR_WLAN_GET_NETWORK_LIST_EAGAIN (-2073L)
 SL_ERROR_WLAN_GET_PROFILE_INVALID_INDEX (-2074L)
 SL_ERROR_WLAN_FAST_CONN_DATA_INVALID (-2075L)
 SL_ERROR_WLAN_NO_FREE_PROFILE (-2076L)
 SL_ERROR_WLAN_AP_SCAN_INTERVAL_TOO_LOW (-2077L)
 SL_ERROR_WLAN_SCAN_POLICY_INVALID_PARAMS (-2078L)
SL_ERROR_WLAN_INVALID_COUNTRY_CODE (-2164L)
SL_ERROR_WLAN_NVMEM_ACCESS_FAILED (-2165L)
SL_ERROR_WLAN_OLD_FILE_VERSION (-2166L)
SL_ERROR_WLAN_TX_POWER_OUT_OF_RANGE (-2167L)
SL_ERROR_WLAN_INVALID_AP_PASSWORD_LENGTH (-2168L)

SL_ERROR_WLAN_PROVISIONING_ABORT_PROVISIONING_ALREADY_STARTED (-2169L)
SL_ERROR_WLAN_PROVISIONING_ABORT_HTTP_SERVER_DISABLED (-2170L)
SL_ERROR_WLAN_PROVISIONING_ABORT_PROFILE_LIST_FULL (-2171L)
SL_ERROR_WLAN_PROVISIONING_ABORT_INVALID_PARAM (-2172L)
SL_ERROR_WLAN_PROVISIONING_ABORT_GENERAL_ERROR (-2173L)
SL_ERROR_WLAN_MULTICAST_EXCEED_MAX_ADDR (-2174L)
SL_ERROR_WLAN_MULTICAST_INVAL_ADDR (-2175L)
SL_ERROR_WLAN_AP_SCAN_INTERVAL_TOO_SHORT (-2176L)
SL_ERROR_WLAN_PROVISIONING_CMD_NOT_EXPECTED (-2177L)


SL_ERROR_WLAN_AP_ACCESS_LIST_NO_ADDRESS_TO_DELETE (-2178L) /* List is empty, no
    address to delete */
SL_ERROR_WLAN_AP_ACCESS_LIST_FULL (-2179L) /* access list is full */
SL_ERROR_WLAN_AP_ACCESS_LIST_DISABLED (-2180L) /* access list is disabled */
SL_ERROR_WLAN_AP_ACCESS_LIST_MODE_NOT_SUPPORTED (-2181L) /* Trying to switch to
    unsupported mode */
SL_ERROR_WLAN_AP_STA_NOT_FOUND (-2182L) /* trying to disconnect station which is
     not connected */
```

B.5. Device error codes

```
/* DEVICE ERRORS CODES*/
SL_ERROR_SUPPLICANT_ERROR (-4097L)
SL_ERROR_HOSTAPD_INIT_FAIL (-4098L)
SL_ERROR_HOSTAPD_INIT_IF_FAIL (-4099L)
SL_ERROR_WLAN_DRV_INIT_FAIL (-4100L)
SL_ERROR_FS_FILE_TABLE_LOAD_FAILED (-4102L) /* init file system failed */
SL_ERROR_MDNS_ENABLE_FAIL (-4103L) /* mDNS enable failed */
SL_ERROR_ROLE_STA_ERR (-4107L) /* Failure to load MAC/PHY in STA role */
SL_ERROR_ROLE_AP_ERR (-4108L) /* Failure to load MAC/PHY in AP role */
SL_ERROR_ROLE_P2P_ERR (-4109L) /* Failure to load MAC/PHY in P2P role */
SL_ERROR_CALIB_FAIL (-4110L) /* Failure of calibration */
SL_ERROR_FS_CORRUPTED_ERR (-4111L) /* FS is corrupted, Return to Factory Image
    or Program new image should be invoked (see sl_FsCtl, sl_FsProgram) */
SL_ERROR_FS_ALERT_ERR (-4112L) /* Device is locked, Return to Factory Image or
    Program new image should be invoked (see sl_FsCtl, sl_FsProgram) */
SL_ERROR_RESTORE_IMAGE_COMPLETE (-4113L) /* Return to factory image completed,
    perform reset */
SL_ERROR_UNKNOWN_ERR (-4114L)
SL_ERROR_GENERAL_ERR (-4115L) /* General error during init */
SL_ERROR_WRONG_ROLE (-4116L)
SL_ERROR_INCOMPLETE_PROGRAMMING (-4117L) /* Error during programming, Program
    new image should be invoked (see sl_FsProgram) */
```

```
SL_ERROR_PENDING_TXRX_STOP_TIMEOUT_EXP (-4118L) /* Timeout expired before
    completing all TX\RX */
SL_ERROR_PENDING_TXRX_NO_TIMEOUT (-4119L) /* No Timeout , still have pending TX\
    RX */
SL_ERROR_INVALID_PERSISTENT_CONFIGURATION (-4120L) /* persistency configuration
    can only be set to 0 (disabled) or 1 (enabled) */
```

B.6. Network config error codes

```
/* NETCFG ERRORS CODES*/
SL_ERROR_STATIC_ADDR_SUBNET_ERROR (-8193L)
SL_ERROR_INCORRECT_IPV6_STATIC_LOCAL_ADDR (-8194L) /* Ipv6 Local address perfix
    is wrong */
SL_ERROR_INCORRECT_IPV6_STATIC_GLOBAL_ADDR (-8195L) /* Ipv6 Global address
    perfix is wrong */
SL_ERROR_IPV6_LOCAL_ADDR_SHOULD_BE_SET_FIRST (-8196L) /* Attempt to set ipv6
    global address before ipv6 local address is set */
```

B.7. File System error codes

```
/* FS ERRORS CODES*/
SL_FS_OK  (0L)
SL_ERROR_FS_EXTRACTION_WILL_START_AFTER_RESET (-10241L)
SL_ERROR_FS_NO_CERTIFICATE_STORE (-10242L)
SL_ERROR_FS_IMAGE_SHOULD_BE_AUTHENTICATE (-10243L)
SL_ERROR_FS_IMAGE_SHOULD_BE_ENCRYPTED (-10244L)
SL_ERROR_FS_IMAGE_CANT_BE_ENCRYPTED (-10245L)
SL_ERROR_FS_DEVELOPMENT_BOARD_WRONG_MAC (-10246L)
SL_ERROR_FS_DEVICE_NOT_SECURED  (-10247L)
SL_ERROR_FS_SYSTEM_FILE_ACCESS_DENIED (-10248L)
SL_ERROR_FS_IMAGE_EXTRACT_EXPECTING_USER_KEY (-10249L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_CLOSE_FILE (-10250L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_WRITE_FILE (-10251L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_OPEN_FILE (-10252L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_GET_IMAGE_HEADER (-10253L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_GET_IMAGE_INFO (-10254L)
SL_ERROR_FS_IMAGE_EXTRACT_SET_ID_NOT_EXIST (-10255L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_DELETE_FILE (-10256L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_FORMAT_FS (-10257L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_LOAD_FS (-10258L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_GET_DEV_INFO (-10259L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_DELETE_STORAGE (-10260L)
SL_ERROR_FS_IMAGE_EXTRACT_INCORRECT_IMAGE_LOCATION (-10261L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_CREATE_IMAGE_FILE (-10262L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_INIT (-10263L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_LOAD_FILE_TABLE (-10264L)
SL_ERROR_FS_IMAGE_EXTRACT_ILLEGAL_COMMAND (-10266L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_WRITE_FAT (-10267L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_RET_FACTORY_DEFAULT (-10268L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_READ_NV (-10269L)
SL_ERROR_FS_PROGRAMMING_IMAGE_NOT_EXISTS (-10270L)
SL_ERROR_FS_PROGRAMMING_IN_PROCESS (-10271L)
SL_ERROR_FS_PROGRAMMING_ALREADY_STARTED (-10272L)
SL_ERROR_FS_CERT_IN_THE_CHAIN_REVOKED_SECURITY_ALERT (-10273L)
SL_ERROR_FS_INIT_CERTIFICATE_STORE (-10274L)
SL_ERROR_FS_PROGRAMMING_ILLEGAL_FILE (-10275L)
SL_ERROR_FS_PROGRAMMING_NOT_STARTED (-10276L)
SL_ERROR_FS_IMAGE_EXTRACT_NO_FILE_SYSTEM (-10277L)
```

```
SL_ERROR_FS_WRONG_INPUT_SIZE (-10278L)
SL_ERROR_FS_BUNDLE_FILE_SHOULD_BE_CREATED_WITH_FAILSAFE (-10279L)
SL_ERROR_FS_BUNDLE_NOT_CONTAIN_FILES (-10280L)
SL_ERROR_FS_BUNDLE_ALREADY_IN_STATE (-10281L)
SL_ERROR_FS_BUNDLE_NOT_IN_CORRECT_STATE (-10282L)
SL_ERROR_FS_BUNDLE_FILES_ARE_OPENED (-10283L)
SL_ERROR_FS_INCORRECT_FILE_STATE_FOR_OPERATION (-10284L)
SL_ERROR_FS_EMPTY_SFLASH (-10285L)
SL_ERROR_FS_FILE_IS_NOT_SECURE_AND_SIGN (-10286L)
SL_ERROR_FS_ROOT_CA_IS_UNKOWN (-10287L)
SL_ERROR_FS_FILE_HAS_NOT_BEEN_CLOSE_CORRECTLY (-10288L)
SL_ERROR_FS_WRONG_SIGNATURE_SECURITY_ALERT (-10289L)
SL_ERROR_FS_WRONG_SIGNATURE_OR_CERTIFIC_NAME_LENGTH (-10290L)
SL_ERROR_FS_NOT_16_ALIGNED (-10291L)
SL_ERROR_FS_CERT_CHAIN_ERROR_SECURITY_ALERT (-10292L)
SL_ERROR_FS_FILE_NAME_EXIST (-10293L)
SL_ERROR_FS_EXTENDED_BUF_ALREADY_ALLOC (-10294L)
SL_ERROR_FS_FILE_SYSTEM_NOT_SECURED (-10295L)
SL_ERROR_FS_OFFSET_NOT_16_BYTE_ALIGN (-10296L)
SL_ERROR_FS_FAILED_READ_NVMEM (-10297L)
SL_ERROR_FS_WRONG_FILE_NAME (-10298L)
SL_ERROR_FS_FILE_SYSTEM_IS_LOCKED (-10299L)
SL_ERROR_FS_SECURITY_ALERT (-10300L)
SL_ERROR_FS_FILE_INVALID_FILE_SIZE (-10301L)
SL_ERROR_FS_INVALID_TOKEN (-10302L)
SL_ERROR_FS_NO_DEVICE_IS_LOADED (-10303L)
SL_ERROR_FS_SECURE_CONTENT_INTEGRITY_FAILURE (-10304L)
SL_ERROR_FS_SECURE_CONTENT_RETRIVE_ASYMETRIC_KEY_ERROR (-10305L)
SL_ERROR_FS_OVERLAP_DETECTION_THRESHHOLD (-10306L)
SL_ERROR_FS_FILE_HAS_RESERVED_NV_INDEX (-10307L)
SL_ERROR_FS_FILE_MAX_SIZE_EXCEEDED (-10310L)
SL_ERROR_FS_INVALID_READ_BUFFER (-10311L)
SL_ERROR_FS_INVALID_WRITE_BUFFER (-10312L)
SL_ERROR_FS_FILE_IMAGE_IS_CORRUPTED (-10313L)
SL_ERROR_FS_SIZE_OF_FILE_EXT_EXCEEDED (-10314L)
SL_ERROR_FS_WARNING_FILE_NAME_NOT_KEPT (-10315L)
SL_ERROR_FS_MAX_OPENED_FILE_EXCEEDED (-10316L)
SL_ERROR_FS_FAILED_WRITE_NVMEM_HEADER (-10317L)
SL_ERROR_FS_NO_AVAILABLE_NV_INDEX (-10318L)
SL_ERROR_FS_FAILED_TO_ALLOCATE_MEM (-10319L)
SL_ERROR_FS_OPERATION_BLOCKED_BY_VENDOR (-10320L)
SL_ERROR_FS_FAILED_TO_READ_NVMEM_FILE_SYSTEM (-10321L)
SL_ERROR_FS_NOT_ENOUGH_STORAGE_SPACE (-10322L)
SL_ERROR_FS_INIT_WAS_NOT_CALLED (-10323L)
SL_ERROR_FS_FILE_SYSTEM_IS_BUSY (-10324L)
SL_ERROR_FS_INVALID_ACCESS_TYPE (-10325L)
SL_ERROR_FS_FILE_ALREADY_EXISTS (-10326L)
SL_ERROR_FS_PROGRAM_FAILURE (-10327L)
SL_ERROR_FS_NO_ENTRIES_AVAILABLE (-10328L)
SL_ERROR_FS_FILE_ACCESS_IS_DIFFERENT (-10329L)
SL_ERROR_FS_INVALID_FILE_MODE (-10330L)
SL_ERROR_FS_FAILED_READ_NVFILE (-10331L)
SL_ERROR_FS_FAILED_INIT_STORAGE (-10332L)
SL_ERROR_FS_FILE_HAS_NO_FAILSAFE (-10333L)
SL_ERROR_FS_NO_VALID_COPY_EXISTS (-10334L)
SL_ERROR_FS_INVALID_HANDLE (-10335L)
SL_ERROR_FS_FAILED_TO_WRITE (-10336L)
SL_ERROR_FS_OFFSET_OUT_OF_RANGE (-10337L)
SL_ERROR_FS_NO_MEMORY (-10338L)
```

```
SL_ERROR_FS_INVALID_LENGTH_FOR_READ (-10339L)
SL_ERROR_FS_WRONG_FILE_OPEN_FLAGS (-10340L)
SL_ERROR_FS_FILE_NOT_EXISTS (-10341L)
SL_ERROR_FS_IGNORE_COMMIT_ROLLBAC_FLAG (-10342L) /* commit rollback flag is not
    supported upon creation */
SL_ERROR_FS_INVALID_ARGS (-10343L)
SL_ERROR_FS_FILE_IS_PENDING_COMMIT (-10344L)
SL_ERROR_FS_SECURE_CONTENT_SESSION_ALREADY_EXIST (-10345L)
SL_ERROR_FS_UNKNOWN (-10346L)
SL_ERROR_FS_FILE_NAME_RESERVED (-10347L)
SL_ERROR_FS_NO_FILE_SYSTEM (-10348L)
SL_ERROR_FS_INVALID_MAGIC_NUM (-10349L)
SL_ERROR_FS_FAILED_TO_READ_NVMEM (-10350L)
SL_ERROR_FS_NOT_SUPPORTED (-10351L)
SL_ERROR_FS_JTAG_IS_OPENED_NO_FORMAT_TO_PRDUCTION (-10352L)
SL_ERROR_FS_CONFIG_FILE_RET_READ_FAILED (-10353L)
SL_ERROR_FS_CONFIG_FILE_CHECSUM_ERROR_SECURITY_ALERT (-10354L)
SL_ERROR_FS_CONFIG_FILE_NO_SUCH_FILE (-10355L)
SL_ERROR_FS_CONFIG_FILE_MEMORY_ALLOCATION_FAILED (-10356L)
SL_ERROR_FS_IMAGE_HEADER_READ_FAILED (-10357L)
SL_ERROR_FS_CERT_STORE_DOWNGRADE (-10358L)
SL_ERROR_FS_PROGRAMMING_IMAGE_NOT_VALID (-10359L)
SL_ERROR_FS_PROGRAMMING_IMAGE_NOT_VERIFIED (-10360L)
SL_ERROR_FS_RESERVE_SIZE_IS_SMALLER (-10361L)
SL_ERROR_FS_WRONG_ALLOCATION_TABLE (-10362L)
SL_ERROR_FS_ILLEGAL_SIGNATURE (-10363L)
SL_ERROR_FS_FILE_ALREADY_OPENED_IN_PENDING_STATE (-10364L)
SL_ERROR_FS_INVALID_TOKEN_SECURITY_ALERT (-10365L)
SL_ERROR_FS_NOT_SECURE (-10366L)
SL_ERROR_FS_RESET_DURING_PROGRAMMING (-10367L)
SL_ERROR_FS_CONFIG_FILE_RET_WRITE_FAILED (-10368L)
SL_ERROR_FS_FILE_IS_ALREADY_OPENED (-10369L)
SL_ERROR_FS_FILE_IS_OPEN_FOR_WRITE (-10370L)
SL_ERROR_FS_ALERT_CANT_BE_SET_ON_NON_SECURE_DEVICE (-10371L) /* Alerts can be
    configured on non-secure device. */
SL_ERROR_FS_WRONG_CERTIFICATE_FILE_NAME (-10372L)
```

B.8. Other error codes

```
/* GENERAL ERRORS CODES*/
SL_ERROR_INVALID_OPCODE (-14337L)
SL_ERROR_INVALID_PARAM (-14338L)
SL_ERROR_STATUS_ERROR (-14341L)
SL_ERROR_NVMEM_ACCESS_FAILED (-14342L)
SL_ERROR_NOT_ALLOWED_NWP_LOCKED (-14343L) /* Device is locked, Return to Factory
     Image or Program new image should be invoked (see sl_FsCtl, sl_FsProgram) */


/* SECURITY ERRORS CODE */
SL_ERROR_LOADING_CERTIFICATE_STORE (-28673L)

/* Device is Locked! Return to Factory Image or Program new
image should be invoked (see sl_FsCtl, sl_FsProgram) */
SL_ERROR_DEVICE_LOCKED_SECURITY_ALERT (-28674L)

SL_ERROR_LENGTH_ERROR_PREFIX (-30734L)
SL_ERROR_WAKELOCK_ERROR_PREFIX (-30735L)
SL_ERROR_DRV_START_FAIL (-30736L)
SL_ERROR_VALIDATION_ERROR (-30737L)
```

```
SL_ERROR_SETUP_FAILURE (-30738L)
SL_ERROR_HTTP_SERVER_ENABLE_FAILED (-30739L)
SL_ERROR_DHCP_SERVER_ENABLE_FAILED (-30740L)
SL_ERROR_WPS_NO_PIN_OR_WRONG_PIN_LEN (-30741L)
```

C. Root certificate catalog

The following list of root CA can be verified using the on-board root certificate catalog.

ACEDICOM Root
Actalis Authentication Root CA
AddTrust Class 1 CA Root
AddTrust External CA Root
AddTrust Qualified CA Root
ANF Global Root CA
Apple Root CA - G2
Apple Root CA - G3
Apple Root CA
Apple Root Certificate Authority
ApplicationCA2 Root
Atos TrustedRoot 2011
Autoridad de Certificacion Firmaprofesional CIF A62634068
Baltimore CyberTrust Root
Buypass Class 3 Root CA
CA Disig Root R1
CA WoSign ECC Root
Certigna
Certinomis - Root CA
CFCA EV ROOT
Chambers of Commerce Root - 2008
China Internet Network Information Center EV Certificates Root
Cisco Root CA 2048
Class 2 Primary CA
COMODO Certification Authority
COMODO ECC Certification Authority
COMODO RSA Certification Authority
ComSign Global Root CA
ComSign Secured CA
Cybertrust Global Root
D-TRUST Root Class 3 CA 2 EV 2009
DigiCert Assured ID Root CA
DigiCert Assured ID Root G2
DigiCert Assured ID Root G3
DigiCert Global Root CA
DigiCert Global Root G2
DigiCert Global Root G3
DigiCert High Assurance EV Root CA
DigiCert Trusted Root G4
DST Root CA X3
EE Certification Centre Root CA
Entrust Root Certification Authority - EC1
Entrust Root Certification Authority - G2
Entrust Root Certification Authority
Equifax Secure Certificate Authority

GeoTrust Global CA
GeoTrust Primary Certification Authority - G2
GeoTrust Primary Certification Authority - G3
GeoTrust Primary Certification Authority
GeoTrust Universal CA 2
GeoTrust Universal CA
GlobalSign ECC Root CA - R4
GlobalSign ECC Root CA - R5
GlobalSign Root CA - R2
GlobalSign Root CA - R3
GlobalSign Root CA
Go Daddy Root Certificate Authority - G2
Hellenic Academic and Research Institutions RootCA 2011
Hongkong Post Root CA 1
IdenTrust Commercial Root CA 1
KISA RootCA 1
Microsec e-Szigno Root CA 2009
OISTE WISeKey Global Root GB CA
QuoVadis Root CA 2 G3
Root CA Generalitat Valenciana
S-TRUST Universal Root CA
SecureSign RootCA11
SecureTrust CA
Staat der Nederlanden EV Root CA
Staat der Nederlanden Root CA - G2
Staat der Nederlanden Root CA - G3 Starfield Class 2 Certification Authority
Starfield Root Certificate Authority - G2
Starfield Services Root Certificate Authority - G2
StartCom Certification Authority G2
StartCom Certification Authority
Swisscom Root CA 1
Swisscom Root CA 2
Swisscom Root EV CA 2
SwissSign Gold Root CA - G3
SwissSign Platinum Root CA - G3
SwissSign Silver Root CA - G3
SZAFIR ROOT CA
SZAFIR ROOT CA2
T-TeleSec GlobalRoot Class 3
TeliaSonera Root CA v1
Thawte Premium Server CA
thawte Primary Root CA - G2
thawte Primary Root CA - G3
thawte Primary Root CA
TWCA Global Root CA
UCA Global Root
UCA Root
VeriSign Class 1 Public Primary Certification Authority - G3
VeriSign Class 2 Public Primary Certification Authority - G3

VeriSign Class 3 Public Primary Certification Authority - G3
VeriSign Class 3 Public Primary Certification Authority - G4
VeriSign Class 3 Public Primary Certification Authority - G5
VeriSign Class 4 Public Primary Certification Authority - G3
VeriSign Universal Root Certification Authority
Visa Information Delivery Root CA

List of Figures

List of Tables

WÜRTH ELEKTRONIK

more than you expect

Internet of Things

Monitoring & Control

Automated Meter Reading

Contact:
Würth Elektronik eiSos GmbH & Co. KG
Division Wireless Connectivity & Sensors

Rudi-Schillings-Str. 31
54296 Trier
Germany

Tel.: +49 651 99355-0
Fax.: +49 651 99355-69
www.we-online.com/wireless-connectivity