



USER MANUAL

CALYPSO

AMB5201 / 2610011025000

VERSION 2.5

OCTOBER 18, 2024

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT

MUST READ

Check for firmware updates

Before using the product, make sure you use the most recent firmware version, data sheet, and user manual. This is especially important for Wireless Connectivity products that were not purchased directly from Würth Elektronik eiSos. A firmware update on these respective products may be required.

We strongly recommend including the possibility of a firmware update in the customer system design.

Revision history

Manual version	FW version	HW version	Notes	Date
1.0	1.0.0	2.1	<ul style="list-style-type: none"> Initial release of the manual 	January 2019
1.1	1.0.0	2.1	<ul style="list-style-type: none"> Added chapter Reference design Added chapter Information for explosion protection 	February 2019
1.2	1.0.0	2.1	<ul style="list-style-type: none"> Added known issues in chapter Firmware history 	March 2019
1.3	1.2.0	2.1	<ul style="list-style-type: none"> Update for the new firmware version. Updated chapter Firmware history 	April 2019
1.4	1.2.0	2.2	<ul style="list-style-type: none"> FCC and IC certification achieved. Compliance statements and requirements added, chapter FCC Compliance Statement (US) and following. 	April 2019
1.5	1.2.0	2.2	<ul style="list-style-type: none"> Improved description of the pinout tables in chapter Pinout 	June 2019
1.6	1.2.0	2.2	<ul style="list-style-type: none"> Updated known issues in chapter Firmware history 	July 2019

1.7	1.3.0	2.2	<ul style="list-style-type: none"> • Update for the new firmware version. • Added a section on quick help in chapter <code>The command interface</code> • Updated chapter <code>Firmware history</code> 	October 2020
1.8	1.3.0	2.2	<ul style="list-style-type: none"> • Updated label in chapter <code>General labeling information</code> • Updated address of Division <code>Wireless Connectivity & Sensors</code> location 	October 2019
1.9	1.3.0	2.2	<ul style="list-style-type: none"> • Updated Declaration of EU conformity to latest Version of EN 300 328 after successfully passing corresponding delta test in chapter <code>Regulatory compliance information</code>. • Added package name in chapter <code>Footprint WE-FP-5</code>. 	October 2020
1.10	1.3.0	2.2	<ul style="list-style-type: none"> • Updated Declaration of EU conformity in chapter <code>Regulatory compliance information</code>. 	December 2020
1.11	1.3.0	2.2	<ul style="list-style-type: none"> • Updated host connection chapter (section <code>Timing and characteristics</code>). • Added further description to AT commands (section <code>WLAN settings</code>). • Updated <code>Known issues</code> 	February 2021

1.12	1.9.0	2.2	<ul style="list-style-type: none"> Removed 2610011025009 from Ordering information as there are no longer pre-cuts available. The modules are available in all quantities as cut tape. Added missing parameter [format] for the events +recv and +recvFrom . See chapter Firmware history. 	December 2021
2.0	2.0.0	2.2	<ul style="list-style-type: none"> Added a new chapter The HTTP server interface. Updated/added all informations that come with firmware v2.0.0. See chapter Firmware history. 	January 2022
2.1	2.0.0	2.2	<ul style="list-style-type: none"> Added chapter Radiation characteristic of the module's internal antenna 	December 2022
2.2	2.2.0	2.2	<ul style="list-style-type: none"> Updated the chapter Error codes Updated/added all informations that come with firmware v2.2.0. See chapter Firmware history. 	April 2023
2.3	2.2.0	2.2	<ul style="list-style-type: none"> Added certificates in addition to the required compliance statements in chapter Regulatory compliance information. 	June 2023

2.4	2.2.0	2.2	<ul style="list-style-type: none">• Added new radio certification for India in chapter 24.7	August 2023
2.5	2.2.0	2.2	<ul style="list-style-type: none">• Added RF test commands for certification in chapter RF test commands• Added description to AT command parse error codes in chapter AT command parse error codes	October 2024

Abbreviations

Abbreviation	Name	Description
0xhh [HEX]	Hexadecimal	All numbers beginning with 0x are stated as hexadecimal numbers. All other numbers are decimal.
AP	Access point	WLAN (IEEE 802.11) infrastructure node offering stations to connect to
BDM	Business Development Engineer	Support and sales contact person responsible for limited sales area
DAD	Duplicate address detection	IPv4/IPv6 addressing mechanism
DC	Duty cycle	Transmission time in relation of one hour. 1% means, channel is occupied for 36 seconds per hour.
DHCP	Dynamic Host Configuration Protocol	Application layer protocol
DNS	Domain Name System	Application layer protocol
HIGH	High signal level	
HTTP(s)	Hypertext transfer protocol (secure)	Application layer protocol
IEEE	Institute of Electrical and Electronics Engineers	
IP	Internet Protocol	Network layer protocol
LLA	Link-local addressing	IPv4/IPv6 local addressing mechanism
LOW	Low signal level	
LSB	Least significant bit	
MAC	Medium access control	
mDNS	multicast-DNS	Application layer protocol
MQTT	Message Queuing Telemetry Transport	Application layer protocol
MSB	Most significant bit	
NWP	Network processor unit	802.11 network processor unit

OTA	Over The Air	Update mechanism
P2P	Peer to Peer	WLAN configuration
PL	Payload	The real, non-redundant information in a frame/packet.
REST	Representational State Transfer	
RF	Radio frequency	Describes everything relating to the wireless transmission.
STA	Station	WLAN (802.11) node in station role, can connect to an AP
SSL	Secure Sockets Layer	Transport layer protocol
TCP	Transmission Control Protocol	Transport layer protocol
TLS	Transport Layer Security	Transport layer protocol
UART		Universal Asynchronous Receiver Transmitter allows communicating with the module of a specific interface.
UDP	Use Datagram Protocol	Transport layer protocol
US	User settings	Any relation to a specific entry in the user settings is marked in a special font and can be found in the respective chapter.
VDD	Supply voltage	
WEP	Wired Equivalent Privacy	802.11 security algorithm
Wi-Fi		Is a Registered Trademark of the Wi-Fi Alliance for interoperability tested WLAN (IEEE 802.11) based products.
WLAN	Wireless Local Area Network	
WPA	WiFi Protected Access	WiFi security algorithm
WPS	WiFi Protected Setup	WiFi security algorithm

Contents

Overview of helpful application notes	13
1. Introduction	15
1.1. Operational description	15
1.2. Block diagram	16
1.3. Ordering information	16
2. Electrical specifications	17
2.1. Recommended operating conditions	17
2.2. Absolute maximum ratings	17
2.3. Power consumption	18
2.3.1. Static	18
2.4. Radio characteristics	18
2.5. Pin characteristics	20
2.6. TX power vs current consumption	20
3. Pinout	23
4. Quick start guide	26
4.1. Antenna connection	26
4.1.1. On-board PCB antenna	26
4.1.2. External antenna	26
4.2. Minimal pin configuration	26
4.3. Power up	27
4.4. Region specific WLAN settings	27
4.5. Quick start example	28
4.5.1. Prerequisites	28
4.5.2. Hardware configuration	28
4.5.3. Setup description	29
4.5.4. Start-up	30
4.5.5. Connect to an access point	30
4.5.6. Creating a TCP server	31
4.5.7. Creating a TCP client	31
4.5.8. Data transfer	32
4.5.9. Terminating the data connection	32
5. Functional description	33
5.1. Key features	33
5.2. Modes of operation	35
5.2.1. BootUp	36
5.2.2. AT command mode	37
5.2.3. OTA update	37
5.2.4. Provisioning	37
5.2.5. Sleep	37
5.2.6. Power save	38
5.2.7. Transparent mode	39

6. Host connection	40
6.1. UART parameters	40
6.2. Hardware flow control	41
6.3. Timing and characteristics	41
7. The command interface	42
7.1. Command types	42
7.2. AT command characteristics	42
7.2.1. Request	42
7.2.2. Confirmations	43
7.2.3. Events	43
7.2.4. Help	43
8. AT commands	44
8.1. Device commands	44
8.1.1. Start and stop commands	44
8.1.2. Test	45
8.1.3. Reboot	45
8.1.4. Factory reset	45
8.1.5. Sleep	46
8.1.6. Power save	47
8.1.7. Get	47
8.1.8. Set	49
8.2. WLAN commands	50
8.2.1. Set mode	50
8.2.2. Scan	50
8.2.3. Manual connection	51
8.2.4. Profiles	52
8.2.5. WLAN settings	54
8.2.6. WLAN policy	57
8.3. Network configuration commands	58
8.4. Socket commands	62
8.4.1. Sockets workflow	62
8.4.1.1. TCP socket	62
8.4.1.2. UDP socket	63
8.4.1.3. Multicast	63
8.4.2. Secure sockets	63
8.4.3. Socket operations	64
8.4.4. Socket settings	66
8.4.5. Socket data exchange	70
8.5. File system commands	72
8.5.1. File system operations	72
8.5.2. File operations	73
8.6. Network application commands	76
8.6.1. mDNS	76
8.6.2. SNTP client	81
8.6.3. HTTP client	82
8.6.4. MQTT client	85

8.6.5.	Ping	88
8.7.	GPIO commands	89
8.8.	RF test commands	90
8.9.	Events	91
8.9.1.	Startup event	91
8.9.2.	General events	91
8.9.3.	WLAN events	92
8.9.4.	Socket events	93
8.9.5.	NetApp events	94
8.9.6.	MQTT events	95
8.9.7.	Fatal error events	96
8.9.8.	Custom events	97
9.	The HTTP server interface	98
9.1.	RESTful APIs	99
9.2.	Network processor GET APIs	100
9.2.1.	System information	101
9.2.2.	Version information	102
9.2.3.	Network information	102
9.2.4.	Ping results	105
9.2.5.	WiFi connection policy status	105
9.2.6.	WiFi profile information	105
9.2.7.	P2P information	105
9.3.	Network processor POST APIs	107
9.3.1.	Date and time	107
9.3.2.	URN configuration	107
9.3.3.	WLAN profiles	107
9.3.4.	WiFi scan	108
9.3.5.	WiFi connection policy	109
9.3.6.	IP configuration	109
9.3.7.	Ping	110
9.4.	User setting GET APIs	111
9.5.	User setting POST APIs	112
9.6.	GPIO GET APIs	115
9.7.	GPIO POST APIs	115
9.8.	File PUT API	116
9.9.	Custom GET API	116
9.10.	Custom POST API	117
10.	Provisioning	119
10.1.	Start in provisioning mode	119
10.2.	Add WLAN profile	119
10.3.	Upload files	121
11.	Typical application use cases	123
11.1.	UDP communication	123
11.1.1.	Prerequisites	123
11.1.2.	UDP socket communication	123

11.2.	TCP communication	124
11.3.	Secure socket communication	124
11.3.1.	Write certificate and key files	124
11.3.2.	Set-up SNTP client	125
11.3.3.	Create an SSL/TLS server	125
11.3.4.	Create an SSL/TLS client	126
11.3.5.	Secure data transfer	127
11.4.	WiFi direct example	128
11.4.1.	Prerequisites	128
11.4.2.	Auto connection setup	128
11.4.3.	Manual connection setup	129
11.5.	Running a web page on the radio module	130
11.5.1.	Load the web page files to the radio module	130
11.5.2.	Accessing the web site in station mode	131
11.5.3.	Accessing the web site in access point mode	131
12.	Connection to Microsoft Azure IoT Central	133
13.	Timing parameters	134
13.1.	Hard reset	134
13.2.	Soft reset	134
14.	Firmware update	135
14.1.	Prerequisites	135
14.2.	Update procedure	136
14.2.1.	Start-up	136
14.2.2.	Connection to the update device	136
14.2.3.	Upload the update package	137
14.2.4.	Finalize the update	138
15.	Firmware history	140
15.1.	Release notes	140
15.2.	New features - version \geq 2.0.0	142
15.3.	Known issues	143
16.	Custom firmware	145
16.1.	Custom configuration of standard firmware	145
16.2.	Customer specific firmware	145
16.3.	Customer firmware	145
16.4.	Contact for firmware requests	146
17.	Design in guide	147
17.1.	Advice for schematic and layout	147
17.2.	Designing the antenna connection	149
17.3.	Antenna solutions	150
17.3.1.	Wire antenna	151
17.3.2.	Chip antenna	151
17.3.3.	PCB antenna	151

17.3.4. Antennas provided by Würth Elektronik eiSos	152
17.3.4.1. 2600130021 - Himalia dipole antenna	152
18. Reference design	153
18.1. EV-Board	154
18.2. Radiation characteristic of the module's internal antenna	156
18.3. Design Guide for FCC ID R7T1001102	157
18.4. Application mode pins	159
19. Manufacturing information	160
19.1. Moisture sensitivity level	160
19.2. Soldering	160
19.2.1. Reflow soldering	160
19.2.2. Cleaning	161
19.2.3. Potting and coating	162
19.2.4. Other notations	162
19.3. ESD handling	162
19.4. Safety recommendations	163
20. Product testing	164
20.1. Würth Elektronik eiSos in-house production tests	164
20.2. EMS production tests	164
21. Physical specifications	166
21.1. Dimensions	166
21.2. Weight	166
21.3. Module drawing	167
21.4. Footprint WE-FP-5	168
21.5. Antenna free area	169
22. Marking	170
22.1. Lot number	170
22.2. General labeling information	171
23. Information for explosion protection	172
24. Regulatory compliance information	173
24.1. Important notice EU	173
24.2. EU Declaration of conformity	174
24.3. FCC Compliance Statement (US)	175
24.3.1. FCC certificate	175
24.4. IC Compliance Statement (Canada)	176
24.4.1. IC certificate	176
24.5. FCC and IC requirements to OEM integrators	177
24.6. Pre-certified antennas	178
24.7. ETA-WPC (India)	179
24.7.1. ETA-WPC certificate	179
25. References	181

26. Important notes	182
27. Legal notice	182
28. License terms	183
A. Wi-Fi certificate	185
B. Error codes	188
B.1. Disconnection reason codes	188
B.2. AT command parse error codes	188
B.3. Socket error codes	189
B.4. Secure socket error codes	189
B.5. WLAN error codes	192
B.6. Device error codes	193
B.7. Network config error codes	194
B.8. File System error codes	194
B.9. HTTP Client error codes	196
B.10. Other error codes	197
C. Root certificate catalog	199
D. TCP flow diagram	202

Overview of helpful application notes

Application note ANR007 - Calypso IoT application based on Calypso module

<http://www.we-online.com/ANR007>

The IoT demo shows how to set up a complete system, using the Calypso WiFi module connected to a computer as sensor/actuator as MQTT client, a smartphone as user interface as MQTT client, a Raspberry Pi as MQTT-broker as cloud platform and an access point for internet connectivity. This application note gives some background information and a step by step instruction to set up this demo.

Application note ANR008 - Wireless Connectivity Software Development Kit

<http://www.we-online.com/ANR008>

To ease the integration of the Würth Elektronik eiSos radio modules into an application, Würth Elektronik eiSos offers the corresponding Software Development Kit (SDK) for most commonly used host processors. This SDK contains drivers and examples in C-code to communicate with the corresponding radio module. This application note shows which SDKs are available and describes how to download and use them.

Application note ANR010 - Range estimation

<http://www.we-online.com/ANR010>

This application note presents the two most used mathematical range estimation models, Friis and two ray ground reflection, and its implementation in the range estimation tool of the RED-EXPERT.

Application note ANR023 - Calypso Cloud connectivity

<http://www.we-online.com/ANR023>

The application note shows how to set up a sensor-to-cloud demo application using the Calypso as sensor/actuator as MQTT client. This application note gives some background information and a step by step instruction to set up a demo with either Amazon AWS or Microsoft Azure cloud platform.

Application note ANR028 - Calypso transparent mode

<http://www.we-online.com/ANR028>

This application note describes a special feature of the Calypso WiFi module, the so called "transparent mode". This mode simply provides a bridge between the Calypso's UART interface and a WiFi socket, where UART data is forwarded to the WiFi socket and vice versa.

Application note ANR029 - Calypso remote GPIO control

<http://www.we-online.com/ANR029>

The Calypso WiFi module offers four remote controllable GPIOs that can be configured as input, output and PWM. This application note describes that feature which provides the possibility to perform simple and quick hostless operation for simple applications.

1. Introduction

The Calypso 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 module 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 [1] certified. The certification ID is WFA81685.

From the firmware version 2.2.0, the Calypso WiFi module driven by the *Wireless connectivity SDK* connects to the Microsoft Azure IoT Central platform as a Plug and Play device. For more details, refer to chapter 12.

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. Standalone applications, without host, can be realized with a custom firmware development.

1.2. Block diagram

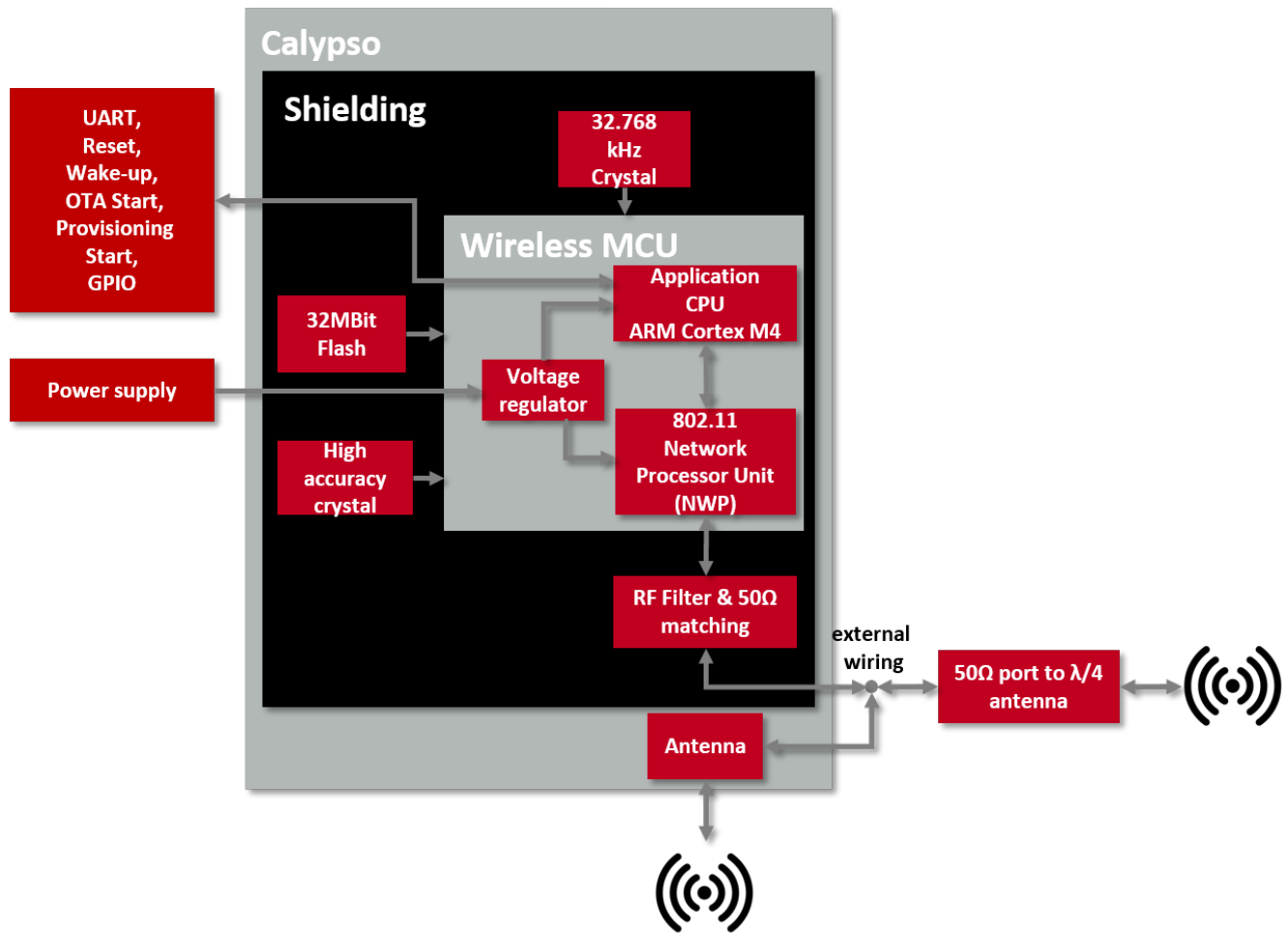


Figure 1: Block diagram

1.3. Ordering information

WE order code	Former order code	Description
2610011025000	AMB5201-TR	WLAN module in T&R packaging
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 EV-Board under the following conditions: T = 25 °C, VDDS= 3.6 V, 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	°C
Ambient thermal slew rate	-20		20	°C / min

Table 2: Recommended operating conditions



When operating at an ambient temperature of over 75 °C, the transmit duty cycle must remain below 50 % to avoid enabling the auto-protect feature of the power amplifier. If the auto-protect feature is triggered, the device takes a maximum of 60 s to restart the transmission.



The VCC brown-out threshold is 2.1V, the VCC blackout level is 1.67V. As ripples may apply when dips happen (e.g. when the TX state is entered) the current supply shall ensure at least VCC of 2.1V is always present in any operating state of the radio module.

To ensure the module's radio performance, ripple on the supply must be less than ± 300 mV.

2.2. Absolute maximum ratings

Description	Min.	Typ.	Max.	Unit
VCC	-0.5		3.8	V
Digital inputs	-0.5		VCC	V
Pin RF, Pin ANT	-0.5		2.1	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 (1 Mbit DSSS mode)		230		mA
RX current consumption (1 Mbit DSSS mode)		76		mA
Sleep mode (WLAN disconnected)		10		µA
Power save mode (WLAN station connected, socket connected, UART off)		2		mA
Peak calibration current, VCC=2.1V		670		mA
Peak calibration current, VCC=3.3 V		450		mA
Peak calibration current, VCC=3.6 V		420		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
<i>RF, ANT</i> pin input voltage			2.1	V
GPIO voltage input high	$0.65 \times VCC$		VCC	V
GPIO voltage input low	-0.5		$0.35 \times VCC$	V
<i>/RESET</i> voltage input high		VCC		V
<i>/RESET</i> voltage input low		0.6		V
GPIO voltage output high	$0.8 \times VCC$		VCC	V
GPIO voltage output low	0		$0.2 \times VCC$	V
Pin output current sunk by any I/O and control pin, drive mode dependent		2		mA
Pin output current sourced by any I/O and control pin, drive mode dependent		2		mA

Table 7: Pin characteristics, $V_{DD5} = 3.3\text{ V}$, $T = 25\text{ °C}$

2.6. TX power vs current consumption

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

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 54 Mbps (OFDM)

3. Pinout

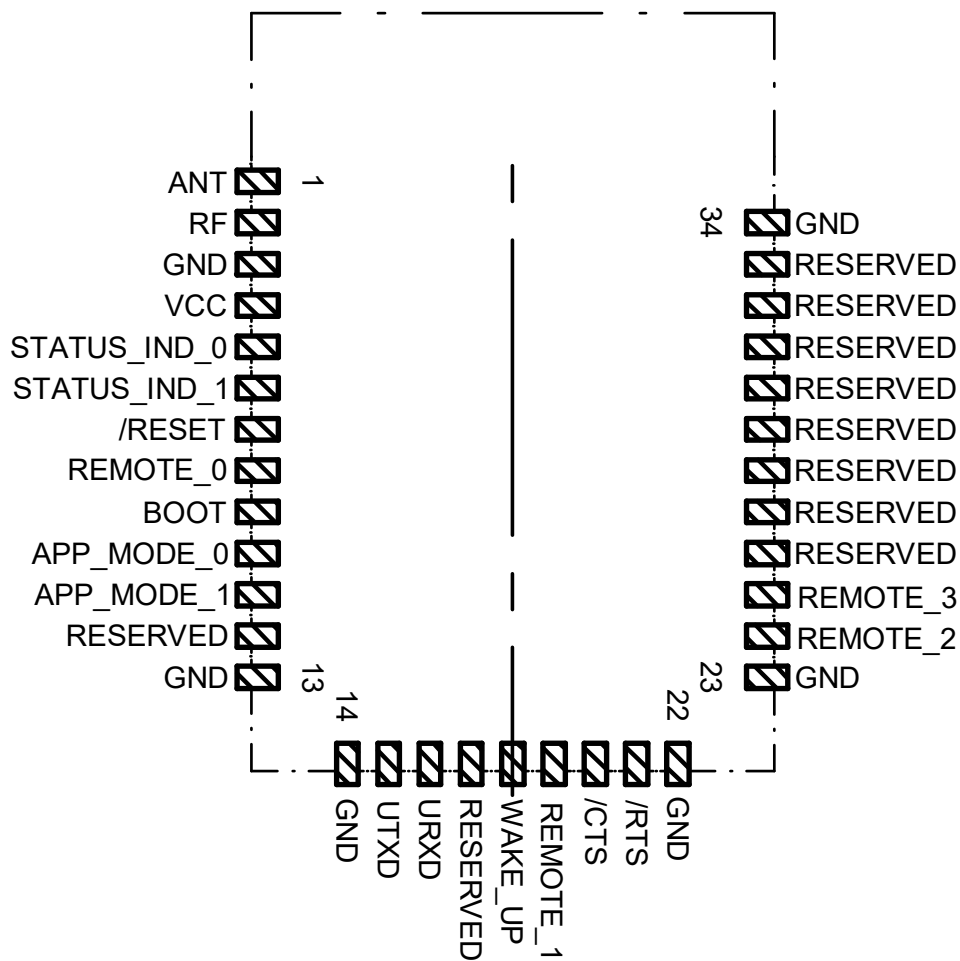


Figure 2: Pinout (top view)

Pad	Name	Chip-set pin	Description
1	<i>ANT</i>	-	RF connection to PCB antenna (see section 4.1)
2	<i>RF</i>	-	50 Ω RF connection to external antenna or on-board antenna via ANT (see section 4.1)
3	<i>GND</i>	-	Negative supply voltage
4	<i>VCC</i>	-	Positive supply voltage
5	<i>STATUS_IND_0</i>	GPIO8	Status indication LED 0, do not connect if not needed
6	<i>STATUS_IND_1</i>	GPIO9	Status indication LED 1, do not connect if not needed
7	<i>/RESET</i>	-	Reset (active low), internal pull-up (100 k Ω)
8	<i>REMOTE_0</i>	GPIO12	Remote GPIO function, do not connect if not needed
9	<i>BOOT</i>	-	Input with internal pull-down (2.7 k Ω), pull low during start-up to boot the standard application, do not connect if not needed
10	<i>APP_MODE_0</i>	GPIO22	Input, internal weak pull-down (see section 5.2), do not connect if not needed
11	<i>APP_MODE_1</i>	GPIO0	Input, internal weak pull-down (see section 5.2), do not connect if not needed
12	<i>RESERVED</i>	GPIO30	Unused, output LOW, do not connect if not needed
13	<i>GND</i>	-	Negative supply voltage
14	<i>GND</i>	-	Negative supply voltage
15	<i>UTXD</i>	GPIO2	Module UART TX, set to HIGH in case Calypso UART is disabled
16	<i>URXD</i>	GPIO1	Module UART RX, uses internal weak pull-up
17	<i>RESERVED</i>	GPIO3	Unused, output LOW, do not connect if not needed
18	<i>WAKE_UP</i>	GPIO4	Wake-up on rising edge, internal pull-down, do not connect if not needed
19	<i>REMOTE_1</i>	GPIO5	Remote GPIO function, do not connect if not needed
20	<i>/CTS</i>	GPIO6	Optionally UART CTS (see section 6.2), uses internal weak pull-down, do not connect if not needed
21	<i>/RTS</i>	GPIO7	Optionally UART RTS (see section 6.2), do not connect if not needed, set to HIGH in case Calypso UART is disabled

22	<i>GND</i>	-	Negative supply voltage
23	<i>GND</i>	-	Negative supply voltage
24	<i>REMOTE_2</i>	GPIO10	Remote GPIO function, do not connect if not needed
25	<i>REMOTE_3</i>	GPIO11	Remote GPIO function, do not connect if not needed
26	<i>RESERVED</i>	GPIO14	Unused, output LOW, do not connect if not needed
27	<i>RESERVED</i>	GPIO15	Unused, output LOW, do not connect if not needed
28	<i>RESERVED</i>	GPIO16	Unused, output LOW, do not connect if not needed
29	<i>RESERVED</i>	GPIO17	Unused, output LOW, do not connect if not needed
30	<i>RESERVED</i>	JTAG_TDI	Debug line (locked), do not connect
31	<i>RESERVED</i>	JTAG_TDO	Debug line (locked), do not connect
32	<i>RESERVED</i>	JTAG_TCK	Debug line (locked), internal pull-down, do not connect
33	<i>RESERVED</i>	JTAG_TMS	Debug line (locked), do not connect
34	<i>GND</i>	-	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 optimized 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 50 Ω RF signal on pin *RF*. In this configuration pin *ANT* of the module has to be connected to ground and pin *RF* to the external antenna via 50 Ω feed line. Refer to chapter 18 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 unconnected.

Pin number	Pin function	Pin connection
1	<i>ANT</i>	Connect to pin 2 or <i>GND</i> (see chapter 4.1)
2	<i>RF</i>	Connect to pin 1 or external antenna (see chapter 4.1)
3,13,14,22,23,34	<i>GND</i>	<i>GND</i>
4	<i>VCC</i>	<i>VCC</i>
7	<i>/RESET</i>	Host GPIO and/or reset button
5,6	<i>STATUS_IND_x</i>	Optionally to host GPIO or LED for status indication
9	<i>BOOT</i>	Boot pin to host GPIO or <i>GND</i>
10,11	<i>APP_MODE_x</i>	Host GPIO for mode selection
15	<i>UTXD</i>	Host UART RX
16	<i>URXD</i>	Host UART TX
18	<i>WAKE_UP</i>	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 t_{reset} 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 section 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 for indicating UART idle towards Calypso. The module will send a start-up UART message once it has booted and started the application. The radio module is ready to receive AT commands 200 μ s (Δt) after the startup message has been transmitted.

For further timing information refer to chapter 13.

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 the battery status.

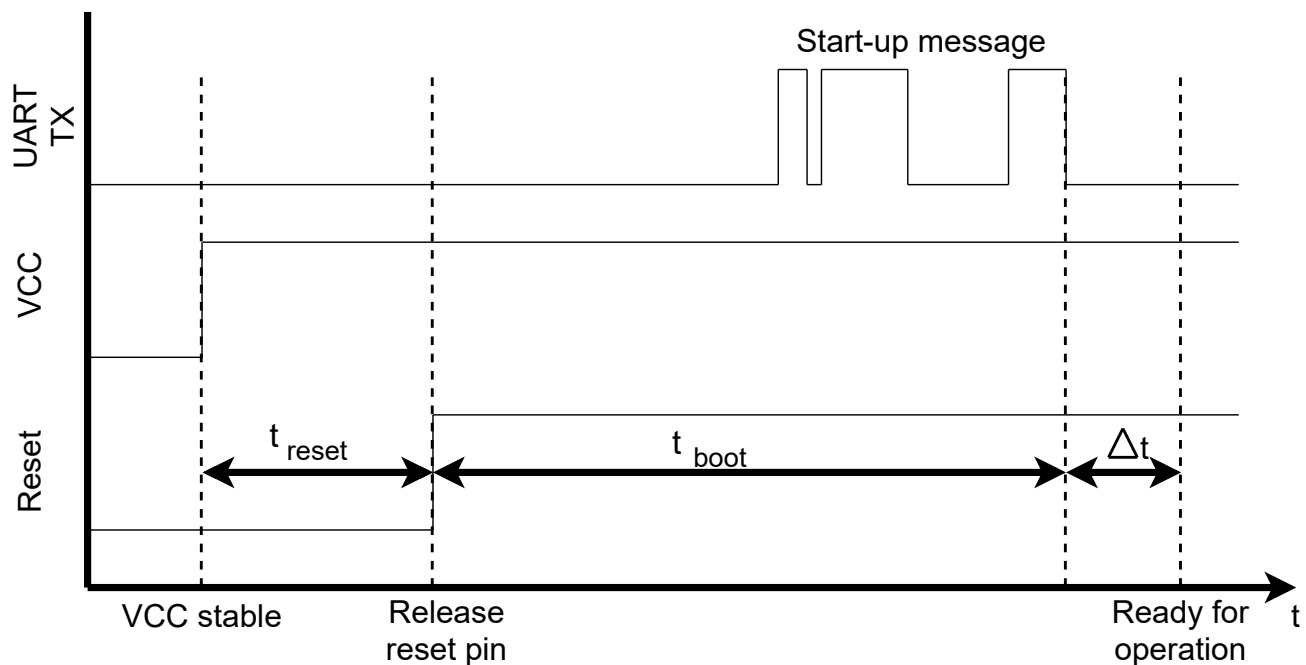


Figure 3: Power up

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 compliant with local regulations, the country code on the module has to be set-up before deployment.

Country code	Supported channels
US	1-11
EU	1-13
JP	1-13

Table 12: Country codes

By default, the country code is set to "US" as the channels allowed in this setting is supported world-wide. 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 Custom firmware).

```
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 4.1 and 4.2. It is recommended to use the Calypso EV-Kit for quick tests.

4.5.1. Prerequisites

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

1. Two Calypso EV-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. The use of Würth Elektronik eiSos's AT-Command tool is recommended ([2]).

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 EV-Board specific manual for a complete hardware description.

1. JP4 to select the USB bus as power supply. Note: This assumes that your PC can deliver enough current on it's USB Interface for two Calypso EV-Boards.
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 mode is used and hence the two *APP_MODE_x* pins shall be connected to *GND*. On the Calypso EV-Board, this can be done by connecting pin 12 and 13 on connector CON8 to *GND* (pin 1, 7 or 19 of CON8).

4.5.3. Setup description

In this example, the two Calypso modules will be connected to the access point and exchange a "hello" (see 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 EV-Board.

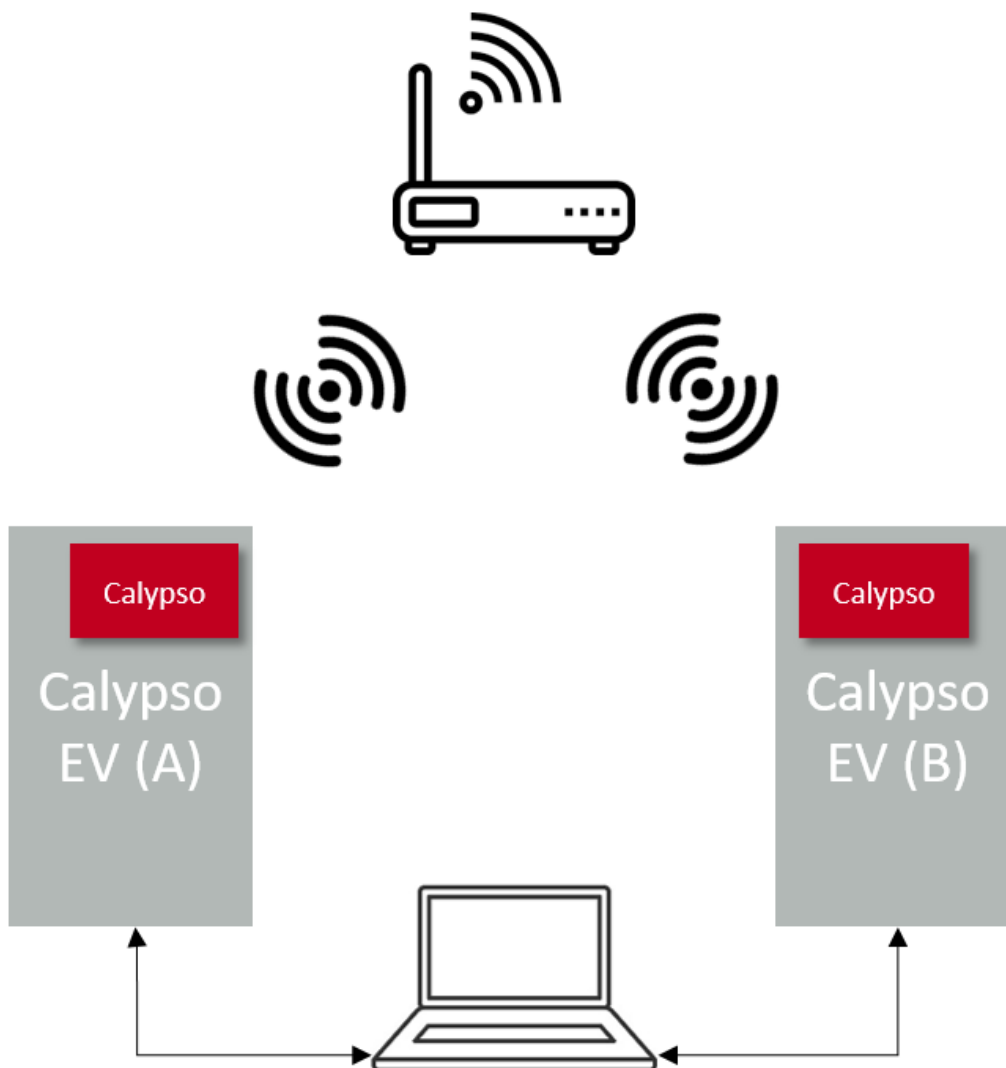


Figure 4: Quick start setup

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 EV-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, chip-ID, MAC address and the current software version.

```
+eventstartup:2610011025000,0x31000019,c8:fd:19:05:54:b4,2.0.0
```

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 indicates 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 13: 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.



Please refer to Appendix D for a detailed flow diagram of TCP server as well as client.

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 module returns a socket ID.

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

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

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

3. Now the server listens for connection requests 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 module 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 the following 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 module A to B.

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

3. At module B the data is received as follows.

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

4. Sending "hello" from module B.

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

5. Receiving the message at module A.

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

4.5.9. Terminating the data connection

1. On the client side, the connection can be terminated with the following command.

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

2. Once the client has terminated the connection, a receive command on the socket at the sever returns with 0 byte payload. This indicates that the client has terminated the connection.

```
AT+recv=1,0,10
OK
+recv:1,0,0,
OK
```

3. At this stage the server can terminate the connection with the following command,

```
AT+close=1
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 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.

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 are summarized in the form of a table. Calypso allows 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 WiFi Direct client and group owner
Channels	1-13
Security	WEP, WPA/WPA2PSK, WPA2 Enterprise (802.1x), WPA3
Provisioning	In AP mode using the on-board HTTPS server
Network layer	IPv4, IPv6
IP addressing	Static, LLA, DHCPv4, DHCPv6 with DAD
Transport layer	TCP, UDP SSLv3.0/TLSv1.0/TLSv1.1/TLSv1.2 Up to 15 simultaneous sockets of which 6 can be secure
Network applications	SNTP client HTTP(S) server mDNS, DNS-SD DHCP server Ping MQTT(S)
Software Update	Secure FOTA update with fall back mechanism in AP as well as Station modes
Security	Secure key storage Trusted root-certificate catalog Encrypted file system Secure OTA Software tamper detection Cloning protection

Table 14: Key features (Part 1)

Feature	Description
Power management	802.11 power save modes Sleep mode with timed or pin wake-up Power save mode with active WLAN and socket connection
Transparent mode	Transparent UART to WiFi mode for cable replacement applications
Remote GPIO control	Ability to configure and control GPIOs (Input, Output and PWM) remotely
RESTful APIs	Web APIs to configure various parameters as well as control GPIO over HTTP
Custom RESTful API	Web APIs to enable communication between the host MCU and HTTP client

Table 15: Key features (Part 2)

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
- Level of the *APP_MODE_x* pins during boot up
- */Reset* signal
- *WAKE_UP* signal or time event

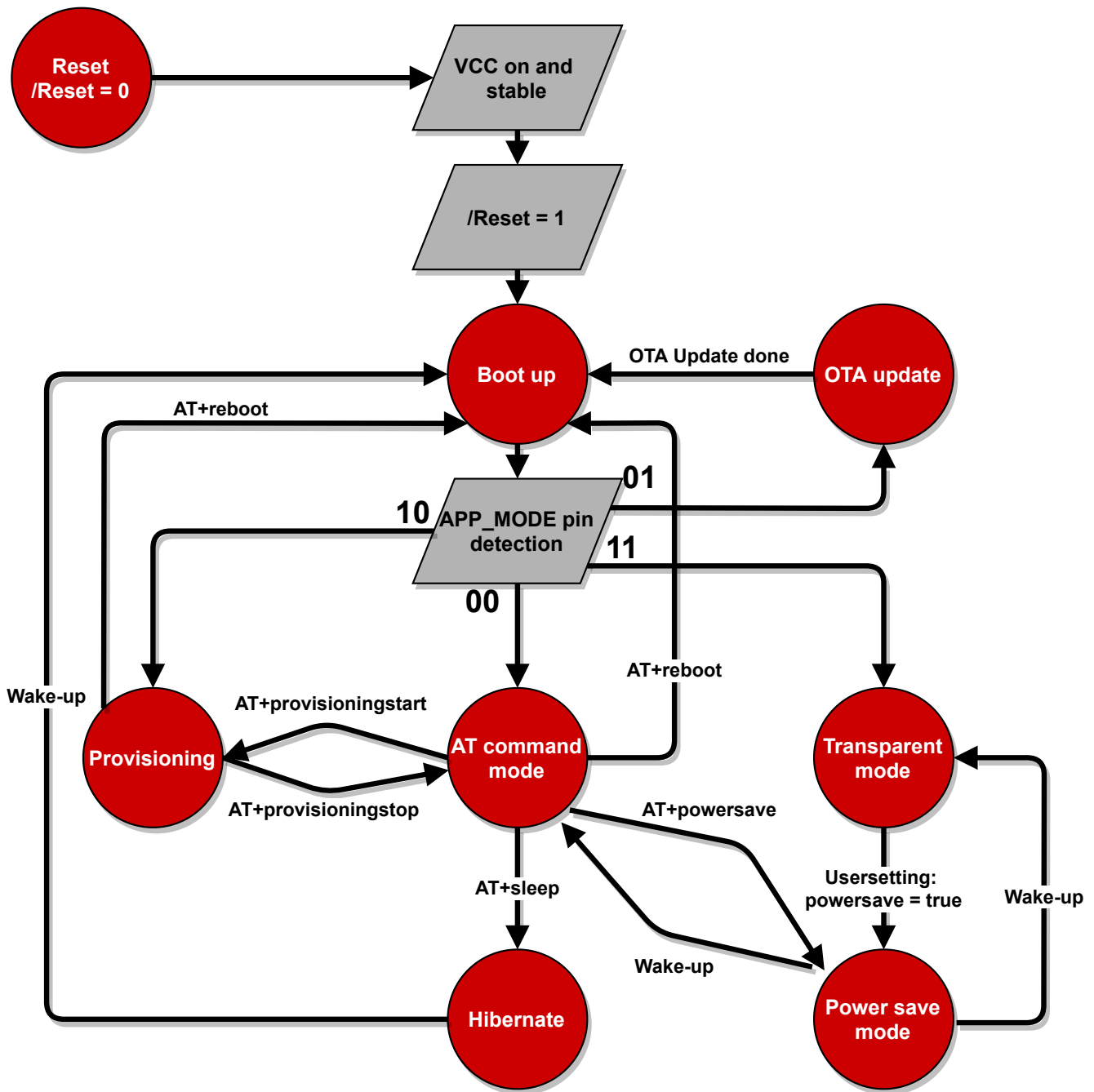


Figure 5: Modes of operation

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.

Mode index	<i>APP_MODE_1</i>	<i>APP_MODE_0</i>	Description
0	LOW	LOW	AT command normal mode, see chapter 7
1	LOW	HIGH	OTA mode, see chapter 14
2	HIGH	LOW	Provisioning mode, see chapter 10
3	HIGH	HIGH	Transparent mode, see application note ANR028 [3]

Table 16: Application modes



The application mode 3 (*APP_MODE_0* = HIGH and *APP_MODE_1* = HIGH) has changed with firmware v1.9.0. The terminal mode is replaced by the transparent mode.

Firmware version	App Mode 3 description
up to v1.3.0	Terminal mode
v1.9.0 onwards	Transparent mode

5.2.2. AT command mode

In this mode, Calypso allows the user to configure and control the module using a AT based command interface over UART. 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/Smart Device) present in the same wireless network (local OTA update). Further details regarding the OTA update mechanism can be found in chapter 14.

5.2.4. Provisioning

To enable easy provisioning when integrated into 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 HTTP server. The user can conveniently browse the settings web page and configure the module using any web browser. More details regarding provisioning can be found in chapter 10.

5.2.5. Sleep

It is essential to have a low power sleep mode for battery powered systems. Calypso offers a hibernate mode with a very low current consumption of less than 10 µA. The characteristics of this mode are as described below.

- 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.
- User configured GPIOs will be set to system default before going into hibernate and set back to user defined default configuration on wake-up.
- UART RX and TX pin needs to be properly terminated to prevent any leakage current.
 - *UTXD* - HIGH
 - *URXD* - HIGH
- The *RTS* pin has the following state:
 - UART configured without flow control *UTXD* - LOW
 - UART configured with flow control *URXD* - HIGH
- On wake up, the module starts from the reset vector meaning that the RAM contents are lost.
- Based on the WiFi connection policy (see chapter 8.2.6), the module can be set up to automatically connect to a saved access point profile and acquire an IP address.
- The socket connections are lost on entering sleep mode and have to be re-established on wake-up.

Section 8.1.5 describes the commands used to put the module to hibernate and chapter 13 describes the timing characteristics.

5.2.6. Power save

For battery powered systems with a requirement that the devices always remain online, the Calypso offers a power save feature (starting with firmware 1.9.0). This feature allows significant power saving while staying connected to the WiFi network as well as sustaining an active connection. An average of less than 2 mA current consumption can be observed, when in station mode with an active connection to a TCP server. Important characteristics of this feature are as described below.

- In this mode, the network processor is active and the application processor enters low power mode when idle.
- The UART and all other peripherals on the application processor are switched off in order to enable its idle mode.
- The connection to the AP is active when in station mode.
- The module does not enable the power save feature in AP mode
- The module does not enable the power save feature if one of the remote GPIOs is configured as PWM.

- All sockets remain active.
- The module briefly activates the UART to forward events to the host MCU.
- In order to send commands/data to the module, the module has to be manually woken up with a rising edge on the *WAKE_UP* pin.
- This feature is supported in AT-command as well as in transparent mode.
- In transparent mode, this feature can be activated by setting the user setting "power_save" to true (see application note ANR028 [3]).
- In AT command mode, the command `AT+powersave` can be used to enable this feature.
- User configured GPIOs retain their states even while the module is in the sleep state. However, due to low current state, the GPIOs are not capable of driving any load directly.



It is not recommended to connect any load (LED for example) directly to the GPIOs.



The on-board HTTP server is not available in the power save mode.

5.2.7. Transparent mode

In transparent mode, the Calypso automatically connects to a preconfigured access point and opens a socket for communication with a preconfigured remote endpoint (TCP server, TCP client or UDP endpoint). Afterwards, the Calypso acts as a transparent bridge between the UART and the created socket. This means that all data sent to the Calypso via UART is forwarded to the socket and all data received on the socket is output on the UART.

For more details see application note ANR028 [3].

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.

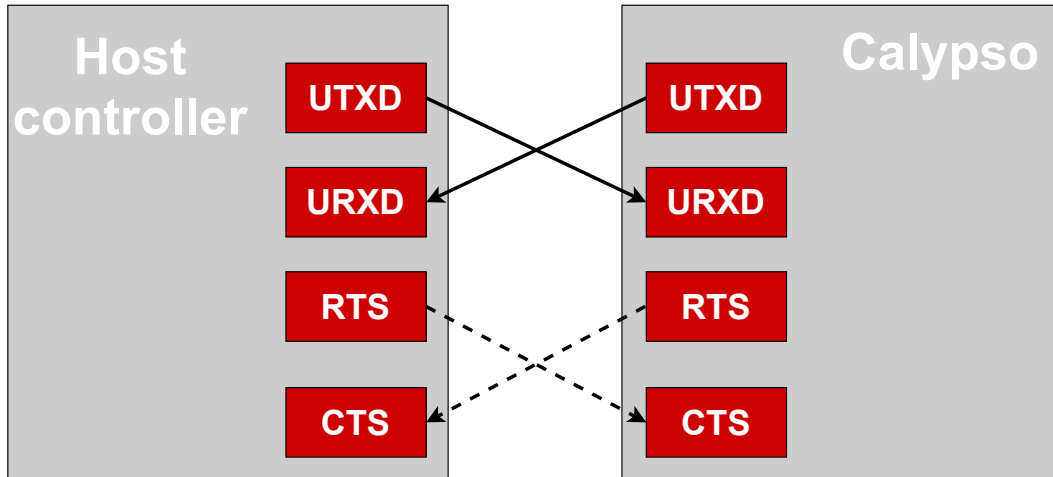


Figure 6: Host interface

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 17: UART parameters

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, parity and flow control of the UART can be configured using the corresponding commands (see section 8.1). The data format is fixed to 8 data bits and one stop bit. This results in a user data ratio of 11 UART symbols per 8 bit.

6.2. Hardware flow control

Hardware flow control is disabled by default. It is recommended not to use baud rates higher than 921600 baud if flow control is disabled.

In case flow control is enabled by using the AT+set command baud rates of up to 3 MBaud are supported.

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 output 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 module 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).

Additionally, in order to ensure proper processing of the AT commands, a short guard interval is necessary between the receipt of a confirmation/indication and the host sending the next command. As shown in figure 7, the guard interval ($t_4 - t_3$) must be at least 40 μ s.

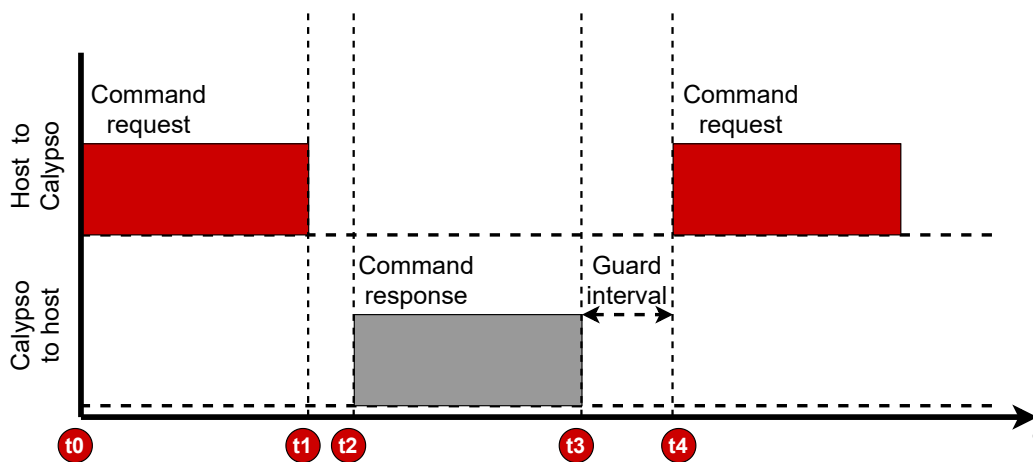


Figure 7: UART timing

7. The command interface

The command interface on the Calypso enables full control over the module using ASCII based AT styled commands, followed by a "\r\n" (hex: 0x0D0A).

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 confirmations 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 white-space between the arguments.
- Furthermore, each parameter is separated from the next with a "," delimiter. A comma shall not be followed by a white-space
- In cases where a parameter is optional or ignored, it may 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
- Bit-mask parameters are represented using "|" delimiter - e.g. x|y
- Data should be either binary or Base64 format (binary to text encoding). Further details about Base64 data encoding including reference implementation can be found in the RFC 4648 of the IETF (Internet Engineering Task Force).

7.2.2. Confirmations

The command confirmations have the following syntax,

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

On success, the confirmations contain a positive acknowledgement.

```
OK
```

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

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

7.2.3. Events

Asynchronous events can arrive at any time and are formatted as follows.

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

7.2.4. Help

The AT command interface has a built-in quick help feature. On sending a "?" character instead of parameter list, the Calypso responds with a list of parameters that are expected for the corresponding command.

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

For example

```
AT+wlanConnect=?  
[SSID],[BSSID],[SecurityType],[SecurityKey],[SecurityExtUser],[SecurityExtAnonUser],[  
SecurityExtEapMethod]  
OK
```

8. AT commands

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

The AT command set is based on ASCII coding of any data. Unless the command requires explicitly a different coding than ASCII.

8.1. Device commands

The commands in the 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
Arguments: None	

Table 18: AT+start

Request	Response
AT+stop=[timeout]	OK or error
Arguments: timeout: in milliseconds <ul style="list-style-type: none"> • 0 - Stop immediately without waiting for a response from the NWP. • 65535 - Wait indefinitely for a response from the NWP. • 0 < timeout < 65535 - Wait for timeout before forcing the NWP to stop. 	

Table 19: 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
Arguments: None	

Table 20: 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
Arguments: None	

Table 21: AT+reboot



It is recommended to use this command whenever possible instead of a 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
Arguments: None	

Table 22: AT+factoryreset



Factory reset operation can take up to 90 seconds to complete. The module responds with an "OK" only after this time period. A start-up message after the "OK" indicates the completion of the factory reset operation.



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 10 μ A. In 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 any wake up trigger, the module starts from the reset vector.

Request	Response
AT+sleep=[timeout]	OK or error
Arguments: timeout: in seconds <ul style="list-style-type: none"> • 0 - sleep forever. • 1 <= timeout <= 86400 - Wait for timeout seconds before wake-up. 	

Table 23: AT+sleep

8.1.6. Power save

The power save command enables the power saving feature on the module. In this mode, the module has an average current consumption of less than 2 mA when remaining connected to the WLAN network with an active socket. UART will be disabled. However, the module will briefly switch on the UART to forward events/data from the network to the host. The module should be woken up manually with a rising edge on the *WAKE_UP* pin in order to send further commands.



In the AT-command mode, once the module is woken up by a rising edge on the *WAKE_UP* pin, the module remains awake and ready to receive commands. Power save command must be sent to put the module back in to power save mode.

Request	Response
AT+powersave	OK or error
Arguments: None	

Table 24: AT+powersave

8.1.7. Get

The generic get command can be used to read the device parameters including version, time, UDID, UART and transparent mode settings. 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] OK or error
Arguments:		Arguments:
ID	option	value1, ..., valueX
general	version	value1: chip ID value2: MAC version (X.X.X.X) value3: PHY version (X.X.X.X) value4: NWP Version (X.X.X.X) value5: ROM version (X) value6: Calypso FW version (X.X.X)
	time	hh,mm,ss,dd,mm,yyyy
	persistent	1=enable, 0=disable
IOT	UDID	16 byte UDID (unique device identifier)
UART	baudrate	baudrate [Baud] (see chapter 6.1)
	parity	0=none, 1=even, 2=odd
	flowcontrol	true, false
	transparent_trigger (see ANR028 [3])	value1: bitmask -timer -1etx -2etx -transmit_etx example for value1: timer 2etx
	transparent_timeout	value1: timeout in [ms], range 6-1000
transparent_etx	2 byte ETX (hex), e.g. 0x0D0A	
transparent_mode (see ANR028 [3])	socket_type	-udp -tcp_server -tcp_client
	remote_address	address of the peer device, e.g. 192.178.2.1
	remote_port	port of the peer device, e.g. 5001
	local_port	port of the local device, e.g. 5001
	secure_method	-none -ssl3 -tlsv1 -tlsv1_1 -tlsv1_2 -ssl3_tlsv1_2
	power_save	true, false
	skip_date_verify	true, false
disable_cert_store	true, false	
GPIO	remote_lock	true, false

Table 25: AT+get

8.1.8. Set

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

Request		Response
AT+set=[ID],[option],[value1],...,[valueX]		OK or error
Arguments:		
ID	option	value1, ..., valueX
general	persistent	1=enable, 0=disable
	time	hh,mm,ss,dd,mm,yyyy (without preceding zeros)
UART	baudrate	baudrate [Baud] (see chapter 6.1)
	parity	0=none, 1=even, 2=odd
	flowcontrol	true, false
	transparent_trigger (see ANR028 [3])	value1: bitmask -timer -1etx -2etx -transmit_etx example for value1: timer 2etx
	transparent_timeout	value1: timeout in [ms], range 6-1000
	transparent_etx	2 byte ETX (hex), e.g. 0x0D0A
transparent_mode (see ANR028 [3])	socket_type	-udp -tcp_server -tcp_client
	remote_address	address of the peer device, e.g. 192.178.2.1
	remote_port	port of the peer device, e.g. 5001
	local_port	port of the local device, e.g. 5001
	secure_method	-none -ssl3 -tlsv1 -tlsv1_1 -tlsv1_2 -ssl3_tlsv1_2
	power_save	true, false
	skip_date_verify	true, false
	disable_cert_store	true, false
GPIO	remote_lock	true, false

Table 26: 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 (WiFi direct) mode. The mode can be selected using the following command. 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
Arguments: - STA: for station mode - AP: for access point mode - P2P: for P2P mode	

Table 27: 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 SL_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]> ... OK or error
Arguments: Index: starting index 0-29 count: number of devices, max. 30	Arguments: Each device has the following parameters listed SSID, BSSID, RSSI, Channel, Security type, hidden_ssid_enabled (0 or 1), cipher, key_management_method

Table 28: 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
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, WPA2_PLUS, WPA3 (see table 30) - SecurityKey: password (ignored if not applicable for selected SecurityType) - 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 29: AT+wlanConnect



The EAP methods with TLS supports TLS v1.0 only and not TLS v1.2

Type	Description	Password length
OPEN	No security	
WEP	WEP open security	13 or 26 characters
WEP_SHARED	WEP shared security	13 or 26 characters
WPA_WPA2	WPA-PSK and WPA2-PSK security types, or a mixed mode of WPA / WPA2-PSK security type (TKIP, AES, mixed mode)	8 to 63 characters
WPA2_PLUS	Supports connection to networks with security WPA3, WPA2+PMF (Protected Management Frames) and WPA2 (CCMP only)	8 to 63 characters
WPA3	Supports connection to WPA3 only networks	8 to 63 characters
WPA_ENT	Enterprise security	
WPS_PBC	WPS push-button security	
WPS_PIN	WPS pin code security	

Table 30: WLAN security types

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

Request	Response
AT+wlanDisconnect	OK or error

Table 31: 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 a connection using 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
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, WPA2_PLUS, WPA3 (see table 30) - 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 32: AT+wlanProfileAdd



Only one enterprise profile can be saved on to the non-volatile memory.

Request	Response
AT+wlanProfileGet=[index]	+wlanProfileGet:[value1], . . . , [value8] OK or error
Arguments: index: profile index, range 0 - 6	Arguments: value1 = SSID value2 = BSSID value3 = Security type value4 = Security key value5 = Security Ext User value6 = Security Ext Anon User value7 = Security EXT EAP method value8 = Priority

Table 33: AT+wlanProfileGet

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

Table 34: AT+wlanProfileDel

8.2.5. WLAN settings

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

Request		Response
AT+wlanSet=[ID],[option],[value1],...,[valueX]		OK or error
ID	option	[value1],...,[valueX]
general	COUNTRY_CODE	US (channels 1-11), EU (channels 1-13) or JP (channels 1-13)
	STA_TX_POWER	0-15 (0 = Max transmit power)
	AP_TX_POWER	0-15 (0 = Max transmit power)
	SCAN_PARAMS	value1: channel mask (Channels bits order: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14) value2: RSSI threshold
	SUSPEND_PROFILES	Suspend profile bit mask (Set bit n to suspend profile with index n)
	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	value1: append MAC address to SSID (true or false) value2: SSID string (up to 32 characters if no MAC is appended, 19 characters otherwise)
	CHANNEL	value1: WLAN Channel , range 1 - 11
	HIDDEN_SSID	0: disabled, 1: send empty SSID in beacon and ignore probe request for broadcast SSID
	SECURITY	open, WEP, WPA_WPA2
	PASSWORD	value1: password WPA: 8-63 characters, WEP: 13 or 26 characters
	MAX_STATIONS	1-4

Table 35: AT+wlanSet

Request		Response
AT+wlanGet=[ID],[option]		+wlanGet:[value1],...,[valueX] OK or error
Arguments:		Arguments: see table 35
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 36: AT+wlanGet

8.2.6. WLAN policy

This set of commands allows changes in behavior 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 following options are available (bit mask - one or more options can be set):

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

P2P - The device connects to the first available WiFi direct device.

- **Scan:** Additional to the one-shot scan, Calypso can be configured to perform periodic scans 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 connection initiation can be active or passive based on the policy set.

Request			Response
AT+wlanPolicySet=[ID],[option],[value1]			OK or error
ID	option	value1	
connection	Auto, Fast or P2P (bit mask)		
scan	Hidden_SSID	scan interval in seconds	
	No_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 37: AT+wlanPolicySet

Request	Response
AT+wlanPolicyGet=[Type]	+wlanPolicyGet:[option],[value1] OK or error
Arguments: connection, scan, PM or P2P	Arguments: (see table 37)

Table 38: 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 WiFi direct client mode, the address acquisition process begins after a successful WLAN connection is established. AP and WiFi direct modes start with a static address assigned to the module with a DHCP server available on-board.

- **IPv4 Stateful with Stateless fall-back:** 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 addresses are in the range 169.254.1.0 to 169.254.254.255.
- **Stateful (DHCPv4) only:** Wait for DHCPv4 server to assign an IP address without time-out.
- **Static:** Address 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 of 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), stateful or static.



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



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



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

	WiFi Station	WiFi AP	WiFi 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, Stateful, Static		
	Global - Stateless, Stateful, Static		

Table 39: IP addresses

Request			Response
AT+netCfgSet=[ID],[option],[value1],...,[valueX]			OK or error
ID	option	[value1],...,[valueX]	
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 address value4: DNS server address	
IPV4_AP_ADDR	STATIC	value1: IP address value2: Subnet mask value3: Default gateway value4: DNS	
IPV6_ADDR_LOCAL	STATIC	IP address	
	STATELESS STATEFUL		
IPV6_ADDR_GLOBAL	STATIC	value1: IP address	
	STATELESS	value2: DNS IP	
	STATEFUL		
IPV4_DNS_CLIENT		Secondary DNS	

Table 40: AT+netCfgSet

Request	Response
AT+netCfgGet=[configID]	+netCfgGet:[option],[value1],...,[valueX] OK or error code
Arguments:	Arguments:
GET_MAC_ADDR	MAC address
IPV4_STA_ADDR or IPV4_AP_ADDR	Method (dhcp, dhcp_lla, static), IP Address, Subnet mask, Gateway, DNS
IPV6_ADDR_LOCAL or IPV6_ADDR_GLOBAL	Method (stateless, stateful, static), IP address
IPV4_DNS_CLIENT	Secondary DNS address

Table 41: 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 provide an 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 workflow

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

- Connectionless socket: Also known as Datagram socket, this type of socket 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, thereby 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 workflow of a TCP socket (see figure 8).

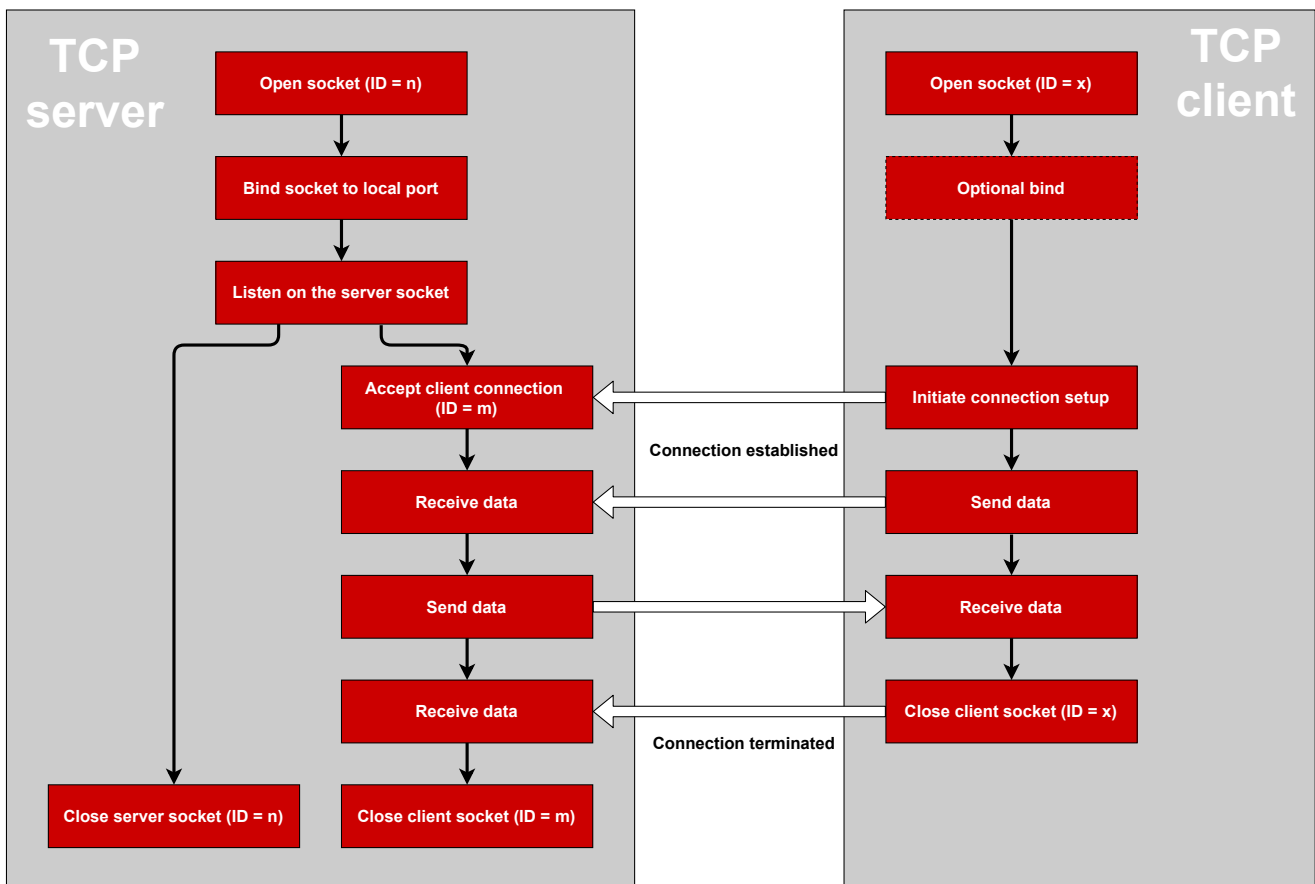


Figure 8: TCP socket workflow



Please refer to Appendix D for a detailed flow diagram including the AT commands.

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 roles as any peer can initiate communication by sending a packet with the corresponding destination address (see figure 9). 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 8).

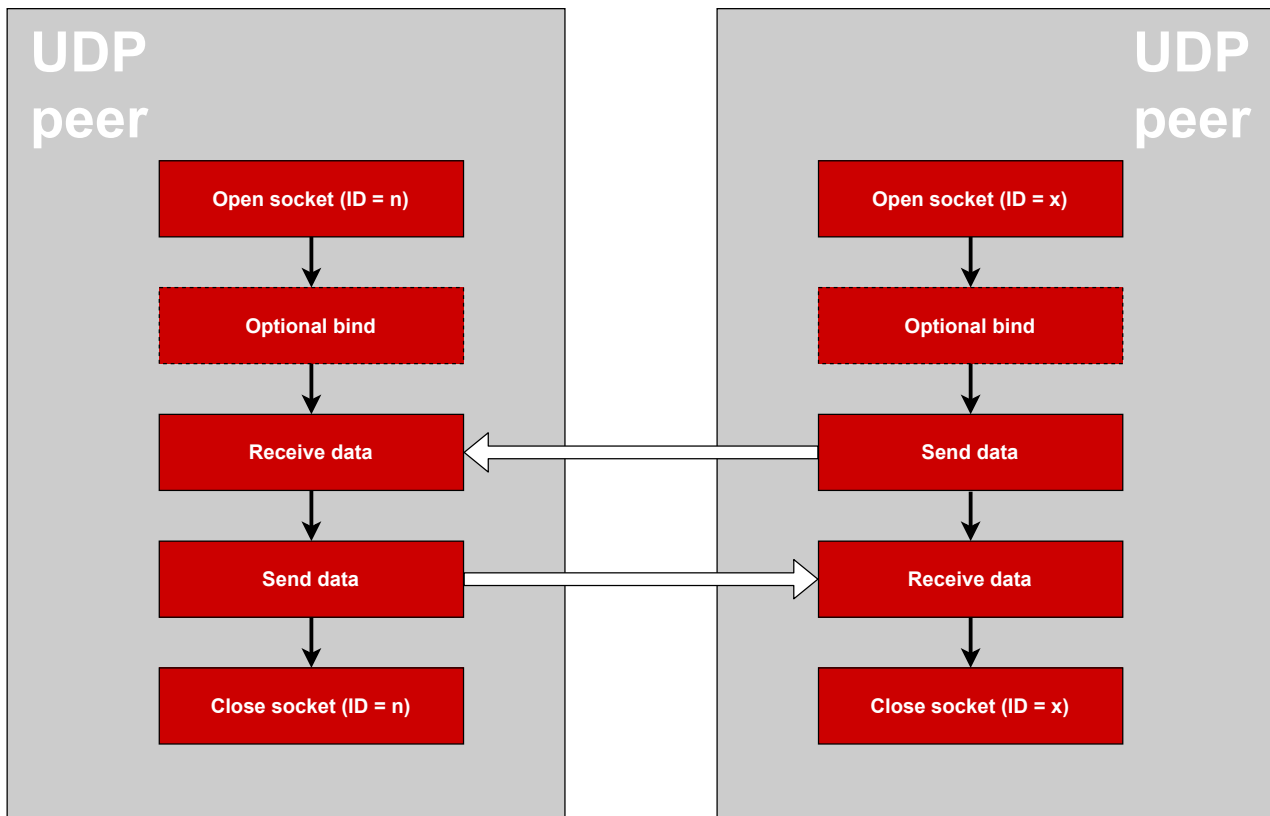


Figure 9: UDP socket work flow

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 provide features like end-to-end encryption and authentication to ensure secure communication between network peers. A sequence of messages is exchanged between a TCP

client and server leading to mutual authentication and encryption of data messages. The TLS/SSL handshake is summarized in figure 10. The SSL/TLS processes are handled in a separate execution environment and hardware acceleration is used to speed up the cryptographic operations.

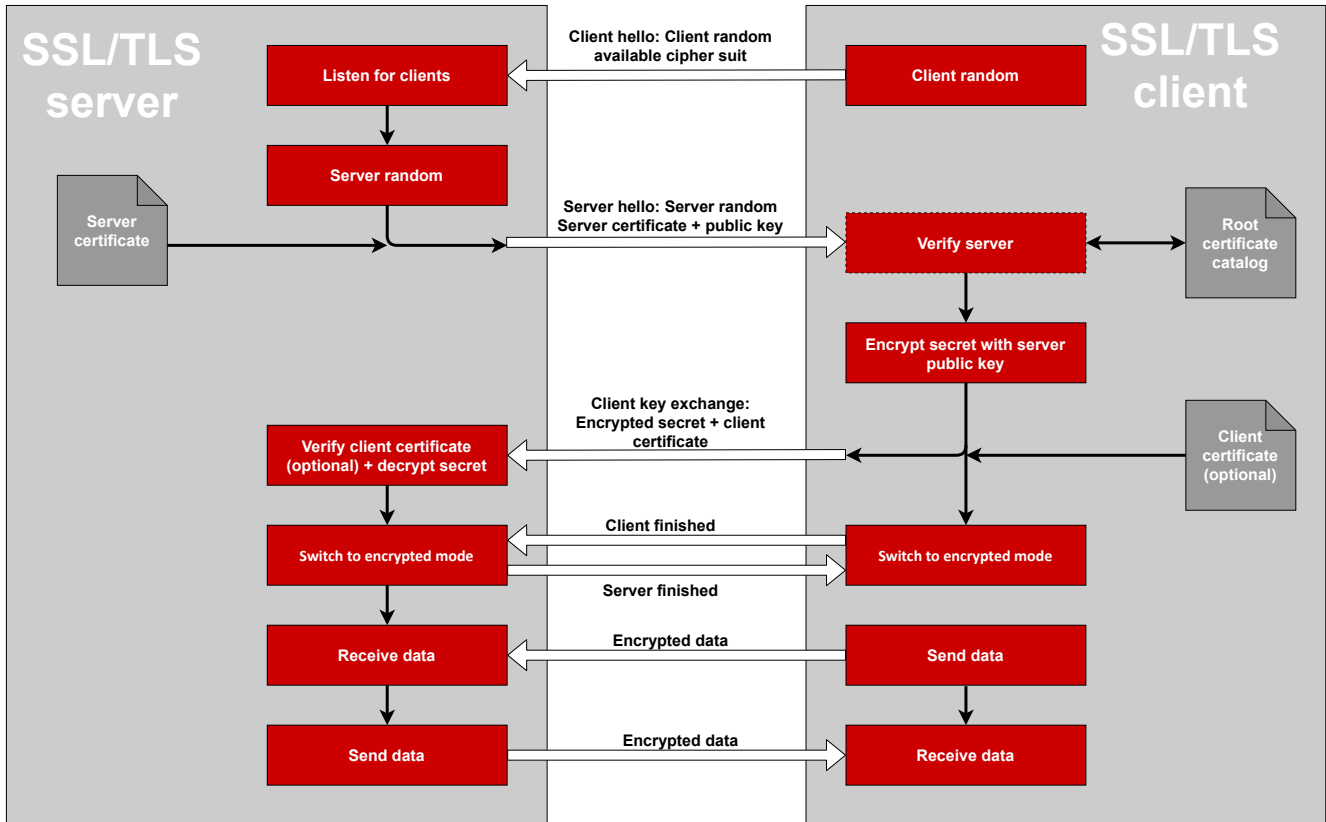


Figure 10: SSL/TLS handshake

The SSL/TLS protocol requires certificates for authentication and a trusted root certificate catalog to verify the certificates. Calypso provides a 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 are 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. The socket select command allows monitoring multiple sockets and triggering on specific events.

Local ports 80 and 8080 are reserved for the on-board HTTP server and should not be reused.

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

Table 42: AT+socket (create a socket)

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

Table 43: 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 44: AT+bind

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

Table 45: AT+listen

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

Table 46: AT+connect

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

Table 47: AT+accept

Request	Response
AT+select=[nfd],[readfds],[timeout sec],[timeout usec]	+select:[readfds] OK or error
Arguments: nfd: The highest numbered file descriptor in any of the three sets (read, write or accept) readfds: 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 48: 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],[level],[option],[value1],...,[valueX]			OK or error
level	option	[value1],...,[valueX]	
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	
	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 51	
	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		
DISABLE_CERTIFICATE_STORE: Disable the use of on-board root CA catalogue	value1: 1=disable store, 0=enable store		

Table 49: AT+setSockOpt

Request			Response
level	option	[value1],...,[valueX]	
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 50: AT+setSockOpt (Part 2)

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 51: Supported cipher methods

Request	Response
AT+getSockOpt=[socketID],[level],[option]	+getSockOpt:[value1],...,[valueX] OK or error
Arguments: socketID: socket descriptor level: SOCKET or IP option: see table 49 and 50	

Table 52: AT+getSockOpt

8.4.5. Socket data exchange

Once a socket has been created and set up, the data transfer can be performed using the send and receive commands described in this section.

Request	Response
AT+recv=[socketID],[format],[length]	OK +recv:[socketID],[format],[length],[data] OK 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 53: 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 of 1460.

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

Table 54: 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 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: data to send	

Table 55: 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 send data to remoteAddress: address of the peer to send data to format: data format 0=binary, 1=base64 (binary to text encoding) length: number of bytes to send (max 1460) data: data to send	

Table 56: AT+sendTo

8.5. File system commands

Calypso creates and maintains an encrypted file system on the serial flash present on-board. The file system provides secure storage for files like certificates, private keys and web pages. In the following, some of the features of the file system are described.

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.
- The 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).
- The maximum number of files is 240 of which 100 are reserved for system files.
- File names may have a maximum size of 180 characters.
- Files can be 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.



Minimize the number of writes to flash to ensure data endurance.



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



Care needs to be taken to have a clean and stable supply voltage especially during flash writes in battery powered applications. A drop in voltage during an 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] OK or error
Arguments: None	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 57: AT+fileGetFileList

8.5.2. File operations

In this section, the file operation commands are described.



Users shall only have one active AT+fileOpen at a time. Any AT+fileOpen shall be finalized by a AT+fileClose to ensure data integrity after filling the file using one or multiple AT+writeFile commands.

Request	Response
AT+fileOpen=[fileName],[options],[fileSize]	+fileOpen:[fileID],[secureToken] OK or error
<p>Arguments:</p> <p>fileName: full file path (max 180 chars)</p> <p>options:</p> <ul style="list-style-type: none"> - READ - Read a file (no bit mask) - WRITE - Open for writing (optionally bitmask with CREATE) - CREATE - Create a new file (optionally bitmask with WRITE or OVERWRITE) - OVERWRITE - Open an existing file (optionally bitmask with CREATE) - 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) <p>fileSize: Max file size in bytes (mandatory for CREATE option)</p>	

Table 58: AT+fileOpen

Request	Response
AT+fileClose=[fileID],[certificateFileName],[signature]	OK or error
<p>Arguments:</p> <p>fileID: ID assigned from AT+fileOpen</p> <p>certificateFileName: Full path to certificate (optional)</p> <p>signature: The signature is SHA1 (optional)</p>	

Table 59: AT+fileClose

Request	Response
AT+fileDel=[fileName],[secureToken]	OK or error
Arguments: fileName: Full path to file secureToken: Token assigned from AT+fileOpen (optional)	

Table 60: AT+fileDel

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

Table 61: AT+fileGetInfo

Request	Response
AT+fileRead=[fileID],[offset],[format],[length]	+FileRead:[format],[numberOfReadBytes],[data] OK 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 62: AT+fileRead

Request	Response
AT+fileWrite=[fileID],[offset],[format],[length],[data]	+FileWrite:[numberOfReadBytes] OK 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 (max 1460) data	

Table 63: 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. For large files the user is required to perform fragmentation. We recommend writing chunks of up to 1024 byte per AT+fileWrite command and increasing the offset parameter with each subsequent AT+fileWrite command accordingly.

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, the 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 being reserved for mDNS messages. Each module can register to up to five services.



By default, the mDNS service is enabled and the host name as well as 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=[Bitmap]	OK or error
Arguments: Bitmap: HTTP_SERVER, DHCP_SERVER, MDNS, DNS_SERVER	

Table 64: AT+netAppStart

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

Table 65: AT+netAppStop

Request	Response
AT+netAppGetHostByName=[HostName],[Family]	OK or error +NetAPPGetHostByName: [HostName],[Host IP address]
Arguments: HostName Family: INET for network protocol IPv4, INET6 for network protocol IPv6	Arguments: Host Name Host IP address: IP address according to the family (IPv4 or IPv6)

Table 66: AT+netAppGetHostByName

Request		Response
AT+netAPPGet=[ID],[option]		+netAPPGet:[value1],...,[valueX] OK or error
Arguments:		Arguments: see table 68 and table 69
ID	option	
DHCP_SERVER	BASIC	
DEVICE	URN	
	DOMAIN	
DNS_CLIENT	TIME	
HTTP_SERVER	PRIM_PORT_NUM	
	AUTH_CHECK	
	AUTH_NAME	
	AUTH_PASSWORD	
	AUTH_REALM	
	ROM_PAGES_ACCESS	
	SECOND_PORT_NUM	
	SECOND_PORT_EN	
MDNS	PRIM_PORT_SEC_EN	
	CONT_QUERY	
	QEVETN_MASK	
	TIMING_PARAMS	

Table 67: AT+netAppGet

Request			Response
AT+netAPPSet=[ID],[option],[value1],...,[valueX]			OK or error
ID	option	values	
DHCP_SERVER	BASIC	Value1: Lease time (in seconds) of the IP Address Value2: First IP Address for allocation Value3: Last IP Address for allocation	
DEVICE	URN	Value1: Device name (maximum length is 33 bytes)	
	DOMAIN	Value1: Domain name (maximum length is 63 bytes)	
DNS_CLIENT	TIME	Value1: Maximum response time in milliseconds Value2: Number of retries	

Table 68: AT+netAPPSet(1)

Request			Response
AT+netAPPSet=[ApplID],[option],[value1],...,[valueX]			OK or error
ID	option	values	
HTTP_SERVER	PRIM_PORT_NUM	Value1: Port number	
	AUTH_CHECK	Value1: 1 = enable, 0 = disable	
	AUTH_NAME	Value1: Authentication name (maximum length is 20 bytes)	
	AUTH_PASSWORD	Value1: Authentication password (maximum length is 20 bytes)	
	AUTH_REALM	Value1: Authorization realm (maximum length is 20 bytes)	
	ROM_PAGES_ACCESS	Value1: 1 = enable, 0 = disable	
	SECOND_PORT_NUM	Value1: Port number	
	SECOND_PORT_EN	Value1: 1 = enable, 0 = disable	
	PRIM_PORT_SEC_EN	Value1: 1 = enable, 0 = disable	
	PRIV_KEY_FILE	Value1: File name (maximum length is 96 bytes)	
	DEV_CERT_FILE	Value1: File name (maximum length is 96 bytes)	
	CA_CERT_FILE	Value1: File name (maximum length is 96 bytes)	
	TMP_REGISTER_SERVICE	Value1: Service name for MDNS (maximum length is 80 bytes)	
	TMP_UNREGISTER_SERVICE	Value1: Service name for MDNS (maximum length is 80 bytes)	
MDNS	CONT_QUERY	Value1: Service name (maximum length is 80 bytes)	
	QEVETN_MASK	Value1: Event mask: ipp, deviceinfo, http, https, workstation, guid, h323, ntp, objective, rdp, remote, rtsp, sip, smb, soap, ssh, telnet, tftp, xmpp, raop	
	TIMING_PARAMS	Value1: Period in ticks (100 ticks = 1 second) Value2: Repetitions Value3: Telescopic factor Value4: Retransmission interval Value5: Maximum period interval Value6: Maximum time	

Table 69: AT+netAPPSet(2)

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.



The SNTP client is disabled by default.

Request	Response
AT+netAPPGet=sntp_client,[Option]	+netAPPGet:[value1] OK or error
Arguments: Option: - enable - update_interval - time_zone - server_address	value1: 0=disabled, 1=enabled value1: minimum update interval in seconds value1: UTC ± minutes value1: list of server addresses

Table 70: SNTP get

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

Table 71: SNTP set

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

Table 72: AT+netAPPUpdateTime

8.6.3. HTTP client

Calypso allows the creation of an 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 an HTTP client are described.

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

Table 73: AT+httpCreate

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

Table 74: 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 75: AT+httpConnect



If the certificate is not formatted properly the command will return an SL_ERROR_BSD_ESECBADPRIVATEFILE (-458). To avoid that after the start line "—BEGIN CERTIFICATE—", the end line "—END CERTIFICATE—" and every 64 characters a carriage return must be part of the certificate.

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

Table 76: 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 77: AT+httpSetProxy

Request	Response
AT+httpSendReq=[index],[method],[uri],[flags],[format], [length], [data]	+httpSendReq:[status] 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, mandatory only for post/put (0=binary, 1=Base64) length: length of payload, mandatory only for post/put data: request payload, mandatory only for post/put	Arguments: status: 200 in case of success, else failure

Table 78: AT+httpSendReq

Request	Response
AT+httpReadResBody=[index], [format], [length]	+httpReadResBody:[index], [flag], [format], [length], [body] OK or error
Arguments: index: client handle format: request format (0=binary, 1=Base64) length: request data length	Arguments: index: client handle flag: 0=data end, 1=more data available format: format of returned data (0=binary, 1=Base64) length: length of returned data body: received data

Table 79: AT+httpReadResBody

Request	Response
AT+httpSetHeader=[index],[option],[flags],[format], [length], [data]	OK or error
Arguments: index: client handle option: see table 82 flags: bitmask (not_persistent, persistent) format: data format (0=binary, 1=Base64) length: length of header data (optional) data: header data (optional)	

Table 80: AT+httpSetHeader

Request	Response
AT+httpGetHeader=[index],[option], [format], [length]	+httpGetHeader:[index],[format], [length],[data] OK or error
Arguments: index: client handle option: see table 82 format: data format (0=binary, 1=Base64) length: max data length	index: client handle format: data format (0=binary, 1=Base64) length: actual data length data: header data

Table 81: 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 82: 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 of 1460.

8.6.4. MQTT client

MQTT (Message Queue Telemetry Transport) is a machine-to-machine (M2M) connectivity protocol based on a publish/subscribe transport mechanism. Features such as light-weight, low network bandwidth and scalability make it ideal for low-power, low-bandwidth IoT applications. An MQTT network consists of a broker connected to one or more clients. Clients can each subscribe to several topics or publish any topic. The broker, on the other hand, is responsible for receiving a published topic and pushing it to all the clients having subscribed to that particular topic.

Calypso offers AT commands to create an MQTT client, subscribe to topics and publish topics. The following section describes these commands.

Request	Response
<p>AT+mqttCreate=[clientID], [flags], [server address], [server port], [security method], [cipher] [private key], [certificate], [CA], [DH key], [protocol], [blocking send], [data format]</p>	<p>+mqttCreate:[index] or error</p>
<p>Arguments:</p> <ul style="list-style-type: none"> -clientID: 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, SSLV3_TLSV1_2 (mandatory if sec flag) -cipher: cipher type, see table 51 (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 = binary, 1 = Base64 	<p>index: client handle used for all other MQTT operations</p>

Table 83: AT+mqttCreate

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

Table 84: AT+mqttDelete

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

Table 85: AT+mqttConnect

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

Table 86: 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 - messageLength: max 1460 - message: payload	

Table 87: AT+mqttPublish

Request	Response
AT+mqttSubscribe=[index], [number of topics], [topic1],[QoS1],[reserved1], ..., [topicX],[QoSX],[reservedX]	OK or error
Arguments: - index: client handle - number of topics: max 4 - topic: topic string - QOS: QOS0, QOS1, QOS2 - reserved: leave empty	

Table 88: 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 89: 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 90: AT+mqttSet

8.6.5. Ping

Calypso provides a 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],[packetsReceived],[roundTripTime] OK 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 91: AT+netAPPPing

8.7. GPIO commands

Read the GPIO default configuration and current values. See also application note ANR029 [4].

Request	Response
AT+gpioGet=[ID],[default]	+gpioget:[ID],[type],[value1],[value2] OK or error
Arguments: ID: ID of the GPIO default: true = default setting, false = current value	ID: ID of the GPIO -type: unused, value1: empty, value2: empty -type: input, value1: high or low, value2: nopull, pulldown or pullup -type: output, value1: high or low, value2: empty -type: pwm, value1: PWM period in milliseconds, value2: PWM ratio in percent

Table 92: AT+gpioGet

Set the GPIO default configuration and current value.

Request	Response
AT+gpioSet=[ID],[save],[type],[value1],[value2]	OK or error
Arguments: ID: ID of the GPIO save: true = save in flash, false = set volatile only type: input, output, pwm or unused value1: low or high in case of output pin, nopull, pulldown or pullup in case of input pin, PWM period in milliseconds in case of PWM pin, empty in case of unused pin value2: PWM ratio in percent in case of PWM pin, otherwise empty	

Table 93: AT+gpioSet



The flash memory used to store these settings has a limited count of write cycles. Please avoid periodic saving to flash as each time one write cycle is used.

8.8. RF test commands

Calypso supports the following test commands to perform radio transmit power tests.



The module need to be configured as a STA and disconnected from any AP to perform RF tests.

Request	Response
AT+calypso=[MODE]	OK or error
Arguments: MODE: 0 - Stop a test in progress 1 - Start continuous TX with maximum power on Channel 1 (2412 MHz) 2 - Start continuous TX with maximum power on Channel 6 (2437 MHz) 3 - Start continuous TX with maximum power on Channel 13 (2472 MHz)	

Table 94: AT+calypso



These commands configures the module to transmit continuously with maximum transmit power. This test mode is intended exclusively for use in the laboratory to perform RF tests.

8.9. 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.

8.9.1. Startup event

The startup event is output by the Calypso when the AT command application has started.

Event:	
+eventstartup:[article number],[chipID],[MAC],[FW version]	
article number	Article number of the radio module
chipID	Chip ID
MAC	MAC address of the radio module
FW version	Firmware version as string

Table 95: +eventstartup event

8.9.2. General events

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

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

Table 96: +eventgeneral event

8.9.3. WLAN events

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

Event:	
+eventwlan:[ID],[value1],...,[valueX]	
ID	[value1],...,[valueX]
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 - scanning - connecting - connected

Table 97: +eventwlan event

8.9.4. Socket events

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

Event:	
ID	[value1],..., [valueX]
+eventsock:[ID],[value1],..., [valueX]	
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 value3: Error value

Table 98: +eventsock event

8.9.5. NetApp events

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

Event:	
+eventnetapp:[ID],[value1],...,[valueX]	
ID	[value1],...,[valueX]
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 99: +eventnetapp event

8.9.6. 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	[value1],...,[valueX]
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 type 0-2
	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 100: +eventmqtt event

8.9.7. Fatal error events

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

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

Table 101: +eventfatalerror event

8.9.8. Custom events

These events are received when using the RestAPI interface to communicate with the host MCU .

Event:	
+eventcustom:[ID],[value1],...,[valueX]	
ID	[value1],...,[valueX]
0 - GPIO remotely configured	value1: ID of the configured GPIO (see section 9.7)
1 - custom POST	value1: ID, value2: value (see section 9.10)

Table 102: +eventcustom event

9. The HTTP server interface

In addition to the AT command interface, the Calypso includes a built-in HTTP server that allows the user to remotely communicate with the radio module as well as the host MCU. This HTTP server allows access to a set of resources through some RESTful APIs. These APIs provide the following functionality:

1. Remote configuration of the network processor parameters (WiFi and network).
2. Remote configuration of module parameters (UART, transparent mode, etc.).
3. Configuration and control of the remote GPIOs.
4. Access to web pages stored on the file system.
5. Allow remote access to resources on the host MCU.

This chapter describes the capabilities of the HTTP server and the related APIs in detail.



The HTTP server on board the Calypso supports version 1.0 of HTTP with a single client. It supports the HTTP standard GET and POST requests.

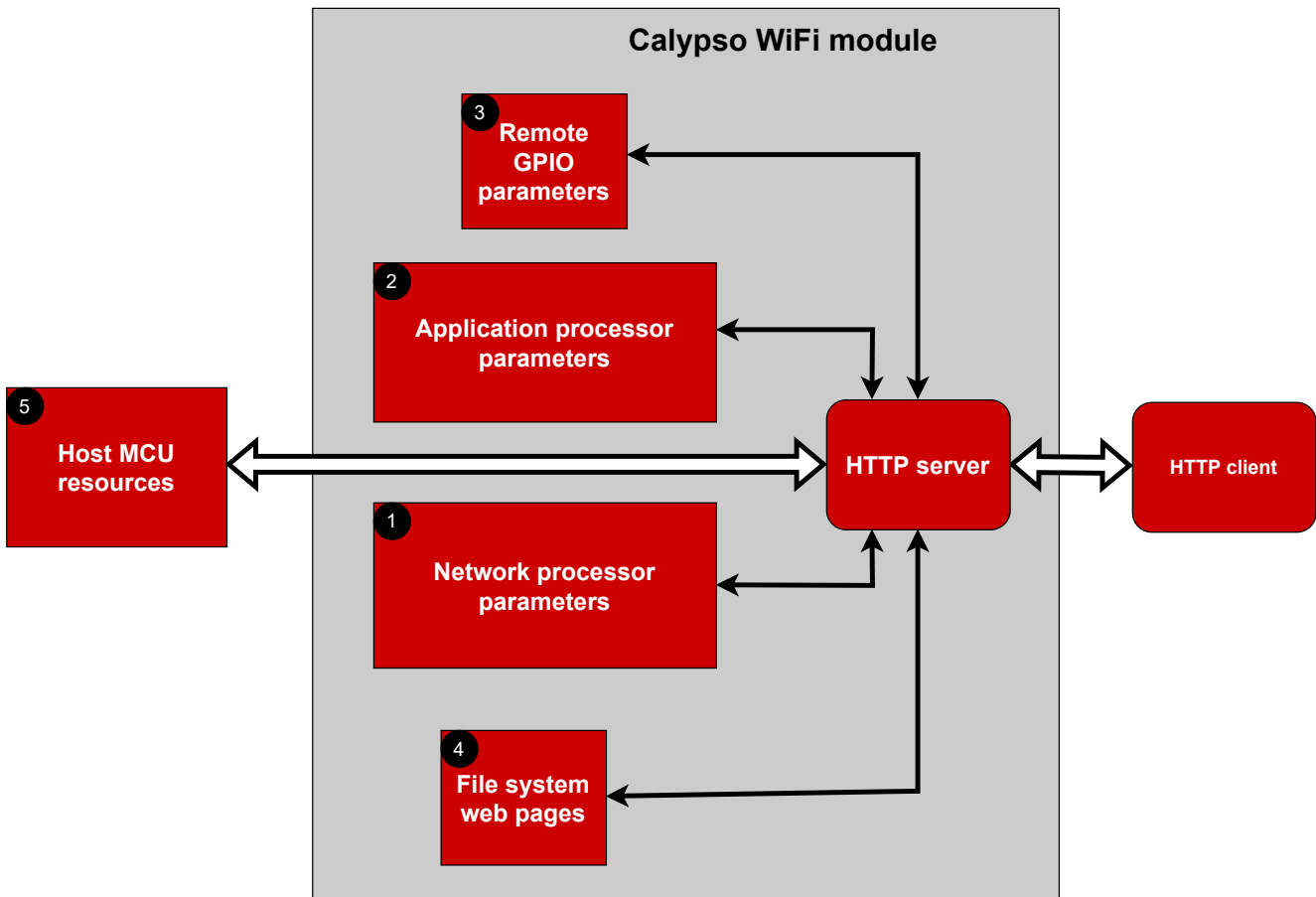


Figure 11: HTTP server

9.1. RESTful APIs

REST (Representational State Transfer) APIs or RESTful APIs provide a platform independent standard web interface in order to access resources served by the web server. The Calypso HTTP server recognizes a set of dedicated resource names and treats them as APIs. A GET request on these names makes the corresponding value available and a POST executes the API without the involvement of the host MCU.

All HTTP API requests except file upload must have the encoding of application/x-WWW-form-urlencoded:

- Each parameter must be separated from its value with an equal sign (=).
- Multiple values must be separated with a comma (,).
- Each parameter-value pair must be separated with an ampersand (&).
- Any binary data must be Base64 encoded.
- All non-alphanumeric characters including those in Base64 encoded data must be URL encoded.

The following sections describe various types of APIs supported by the Calypso radio module in detail.



The reference implementation of all of the APIs can be found on the on-board provisioning web-pages. The source files for the same can be made available on request.

The table below summarizes the availability of the APIs in different modes of operation:

API	AT command	Provisioning	Transparent	FOTA
NWP GET	Yes	Yes	Yes	No
NWP POST	No	Yes	No	No
User setting GET	Yes	Yes	Yes	No
User setting POST	No	Yes	No	No
GPIO GET	Yes	Yes	Yes	No
GPIO POST	Yes	Yes	Yes	No
File PUT	No	Yes	No	No
Custom GET	Yes	No	No	No
Custom POST	Yes	No	No	No

Table 103: API in application modes

9.2. Network processor GET APIs

The Calypso supports querying various device parameters through a mechanism called device tokens. The token names have a rigid convention "__SL_" followed by three characters of the parameter ID. For example, __SL_G_S.B is the token used to retrieve the device URN. These requests are handled directly by the network processor. The tokens can be accessed in the following ways:

- A HTTP GET request on the resource name. For example:

```
GET:/_SL_G_S.B
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
Response: calypso
```

- The token can be embedded inside any serve-able resource where they are replaced by their value when it is served. For example, if a text file is created under the path /WWW/example.txt with the content:

```
Device hardware version: __SL_G_V.D
Device network version: __SL_G_V.A
```

A GET request on calypso.net/example.txt returns the following.

```
Device hardware version: 31000019
Device network version: 3.20.0.1
```

9.2.1. System information

Token	Name	Value and usage
__SL_G_S.A	System uptime	Returns the system uptime since the last reset in the following format: 000 days 00:00:00
__SL_G_S.B	Device name (URN)	Returns device name
__SL_G_DNP	Device name	Returns device name + MAC address (as string) if the default device name is set.
__SL_G_S.C	Domain name	Returns domain name
__SL_G_S.D	Device mode (role)	Returns device role. Values: Station, Access Point, P2P
__SL_G_S.E	Device role station	Drop-down menu select/not select. Returns selected if device is station, else it returns not_selected.
__SL_G_S.F	Device role AP	Drop-down menu select/not select. Returns selected if device is AP, else it returns not_selected.
__SL_G_S.G	Device role P2P	Drop-down menu select/not select. Returns selected if device is in P2P, else it returns not_selected.
__SL_G_S.H	Device name URN (truncated to 16 bytes)	Returns the URN string name with up to 16 bytes length. Longer names will be truncated.

Table 104: System information tokens (Part 1)

Token	Name	Value and usage
__SL_G_S.I	System requires reset (after parameters change)	If system requires reset, return value will be the following string: "– Some parameters were changed, System may require reset –" else it returns an empty string. (Every internal post that was handled will cause this token to return TRUE.)
__SL_G_S.J	Get system time and date	Returned value is a string with the following format: year, month, day, hours, minutes, seconds
__SL_G_S.K	Safe mode status	If the device is in safe mode - returns "Safe Mode", if not returns empty string

Table 105: System information tokens (Part 2)

9.2.2. Version information

Token	Name	Value and usage
__SL_G_V.A	NWP version	Returns string with the version information
__SL_G_V.B	MAC version	Returns string with the version information
__SL_G_V.C	PHY version	Returns string with the version information
__SL_G_V.D	HW version	Returns string with the version information
__SL_G_REV	Revision	Returns string with the version information

Table 106: Version information tokens

9.2.3. Network information

Token	Name	Value and usage
__SL_G_N.A	STA IPv4 address	IP address in string format: xxx.yyy.zzz.ttt
__SL_G_N.B	STA IPv4 subnet mask	Subnet mask in string format: xxx.yyy.zzz.ttt
__SL_G_N.C	STA IPv4 default gateway	Default gateway in string format
__SL_G_N.D	MAC Address	MAC address in string format
__SL_G_N.E	STA IPv4 DHCP state	Returned value: Enabled / Disabled
__SL_G_N.F	STA IPv4 DHCP disable state	If DHCP is disabled, returns Checked, else returns Not_Checked. Used in the DHCP radio button
__SL_G_N.G	STA IPv4 DHCP enable state	If DHCP is enabled, returns Checked, else returns Not_Checked. Used in the DHCP radio button
__SL_G_N.L	STA IPv4 LLA enable state	If LLA option is enabled, returns Checked, else returns Not_Checked.
__SL_G_N.H	STA IPv4 DNS server	DNS server in string format: xxx.yyy.zzz.ttt
__SL_G_LV6	STA IPv6 enable	If IPv6 interface is enabled, returns Checked, else returns Not_Checked.

Table 107: Network information tokens 1 (Station or P2P client)

Token	Name	Value and usage
__SL_G_LSC	STA IPv6 local address type	Returns Checked if IPv6 local address mode is static
__SL_G_LSS	STA IPv6 local address type	Returns Checked if IPv6 local address mode is stateless
__SL_G_LSF	STA IPv6 local address type	Returns Checked if IPv6 local address mode is stateful
__SL_G_N.Z	STA IPv6 global address type	Returns Checked if IPv6 global address mode is stateful
__SL_G_N.R	STA IPv6 global address type	Returns Checked if IPv6 global address mode is stateful
__SL_G_N.O	STA IPv6 global address type	Returns Checked if IPv6 global address mode is stateful
__SL_G_N.S	STA IPv6 global address type	Returns Checked if IPv6 global address mode is stateful
__SL_G_LSK	STA Current IPv6 local address	Returns address in string format
__SL_G_LSG	STA Current IPv6 global address	Returns address in string format
__SL_G_LSP	STA IPv6 DNS server	Returns address in string format
__SL_G_LSO	STA IPv6 local address mode	Returns Disabled / Static / Stateless / Stateful
__SL_G_LSD	STA IPv6 global address mode	Returns Disabled / Static / Stateless / Stateful

Table 108: Network information tokens 2 (Station or P2P client)

Token	Name	Value and usage
__SL_G_N.I	DHCP start address	Returns address in string format: xxx.yyy.zzz.ttt
__SL_G_N.J	DHCP end address	Returns address in string format: xxx.yyy.zzz.ttt
__SL_G_N.K	DHCP Lease Time	Returns string with lease time in seconds

Table 109: Network information tokens (DHCP server)

Token	Name	Value and usage
__SL_G_N.P	AP IP address	Returns address in string format: xxx.yyy.zzz.ttt
__SL_G_N.Q	AP subnet mask	Returns address in string format: xxx.yyy.zzz.ttt
__SL_G_N.T	AP gateway address	Returns address in string format: xxx.yyy.zzz.ttt
__SL_G_N.U	AP DNS address	Returns address in string format: xxx.yyy.zzz.ttt
__SL_G_W.A	WiFi channel in AP mode	Channel number
__SL_G_W.B	SSID	SSID string
__SL_G_W.I	Is SSID public	If SSID is public (visible), returns Checked, else returns Not_Checked
__SL_G_W.J	Is SSID hidden	If SSID is hidden (invisible), returns Checked, else returns Not_Checked.
__SL_G_W.C	Security type	Returned values: Open, WEP, WPA
__SL_G_W.D	Security type open	If security type is open, returns Checked, else returns Not_Checked
__SL_G_W.E	Security type WEP	If security type is WEP, returns Checked, else returns Not_Checked
__SL_G_W.F	Security Type WPA	If security type is WPA, returns Checked, else returns Not_Checked
__SL_G_SR1, __SL_G_SR2, __SL_G_SR3, __SL_G_SR4	The configured max number of connected stations	The token representing the max number of connected stations returns Checked. Others return Not Checked
__SL_G_CN1, __SL_G_CN2, __SL_G_CN3, __SL_G_CN4	Name of the connected station in the given index	Each token returns the host name of the station in the specified index. "-" is returned if the client does not exist
__SL_G_CM1, __SL_G_CM2, __SL_G_CM3, __SL_G_CM4	MAC address of the connected station in the given index	Each token returns the MAC address of the station at the specified index. "-" is returned if the client does not exist
__SL_G_CI1, __SL_G_CI2, __SL_G_CI3, __SL_G_CI4	IP address of the connected station at the given index	Each token returns the IP address of the station at the specified index. "-" is returned if the client does not exist
__SL_G_NW1, __SL_G_NW0	Tokens for retrieving the scan results	Incrementally returns scan results pre-appended with security type (0=open, 1=WEP, 2=WPA/WPA2, 3=WPA3).

Table 110: Network information tokens (AP or P2P GO)

9.2.4. Ping results

Token	Name	Value and usage
__SL_G_T.A	IP address	Returns address in string format: xxx.yyy.zzz.ttt
__SL_G_T.B	Packet size	Packet size string
__SL_G_T.C	Number of pings	Number of pings string
__SL_G_T.D	Ping results - total sent	Number of total pings sent
__SL_G_T.E	Ping results - successful sent	Number of successful pings sent
__SL_G_T.F	Ping test duration	In seconds

Table 111: Ping result tokens

9.2.5. WiFi connection policy status

Token	Name	Value and usage
__SL_G_P.E	Auto Connect	If auto connect is enabled, returns Checked, else returns Not_Checked
__SL_G_P.F	Fast Connect	If fast connect is enabled, returns Checked, else returns Not_Checked
__SL_G_P.G	Any P2P	If any P2P is enabled, returns Checked, else returns Not_Checked

Table 112: Connection policy tokens

9.2.6. WiFi profile information

Token	Name	Value and usage
__SL_G_PNx	Return profile x SSID (x = profile index 1 - 7)	SSID string
__SL_G_PsX	Return profile x security status (x = profile index 1 - 7)	Returned values: Open, WEP, WPA, WPS, ENT, P2P_PBC, P2P_PIN or "-" for empty profile.
__SL_G_PP1x	Return profile x priority (x = profile index 1 - 7)	Profile priority: 0 - 7

Table 113: WiFi profile information tokens

9.2.7. P2P information

Token	Name	Value and usage
__SL_G_R.A	P2P device name	Device name string
__SL_G_R.B	P2P device type	Device type string
__SL_G_R.C	P2P listen channel	Returns string of the listen channel number
__SL_G_R.T	Listen channel #1	If current listen channel is #1, returns selected, else returns not_selected
__SL_G_R.U	Listen channel #6	If current listen channel is #6, returns selected, else returns not_selected
__SL_G_R.V	Listen channel #11	If current listen channel is #11, returns selected, else returns not_selected
__SL_G_R.E	P2P operation channel	Returns string of the operational channel number
__SL_G_R.W	Operational channel #1	If current operational channel is #1, returns selected, else returns not_selected
__SL_G_R.X	Operational channel #6	If current operational channel is #6, returns selected, else returns not_selected
__SL_G_R.Y	Operational channel #11	If current operational channel is #11, returns selected, else returns not_selected
__SL_G_R.L	Negotiation intent value	Returned values: Group Owner, Negotiate, Client
__SL_G_R.M	Role group owner	If intent is Group Owner, returns Checked, else returns Not_Checked
__SL_G_R.N	Role negotiate	If intent is Negotiate, returns Checked, else returns Not_Checked
__SL_G_R.O	Role client	If intent is Client, returns Checked, else returns Not_Checked
__SL_G_R.P	Negotiation initiator policy	Returned Values: Active, Passive, Random Back off
__SL_G_R.Q	Neg Initiator Active	If negotiation initiator policy is Active, returns Checked, else returns Not_Checked
__SL_G_R.R	Neg Initiator Passive	If negotiation initiator policy is Passive, returns Checked, else returns Not_Checked
__SL_G_R.S	Neg Initiator Random back off	If negotiation initiator policy is random back off, returns Checked, else returns Not_Checked

Table 114: P2P tokens

9.3. Network processor POST APIs

The POST APIs of the Calypso allow setting of parameters as well as performing actions independent of the host MCU.

9.3.1. Date and time

The device time and date can be set by posting to `/api/1/wlan/en_ap_scan/set_time`.

```
POST:/api/1/wlan/en_ap_scan/set_time
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
Content: __SL_P_S.J=yyyy,mm,dd,hh,mm,ss
```

9.3.2. URN configuration

The device URN can be set by the following POST request:

```
POST:/api/1/netapp/set_urn
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
Content: __SL_P_S.B=my-urn
```

9.3.3. WLAN profiles

WLAN connection profiles can be added by posting the parameters to either `/api/1/wlan/profile_add` or `/api/1/wlan/profile_p2p`.

```
POST:/api/1/wlan/profile_add
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
```

```
POST:/api/1/wlan/profile_p2p
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
```

Token	Name	Example
<code>__SL_P_PA</code>	The SSID of the desired AP. Must not exceed 32 characters	<code>__SL_P_PA=TargetSSID</code>
<code>__SL_P_PB</code>	Security type for the connection. 0-Open, 1-WEP, 2-WPA, 3-WPA2/WPA2+PMF, 5-WPA3, 6-WPS/Push-button, 7-WPS/Pin Keypad, 8-WPS/Pin Display	<code>__SL_P_PB=3</code>
<code>__SL_P_PC</code>	Security key or PIN code. Must not exceed 64 characters	<code>__SL_P_PC=MySecurePassword</code>
<code>__SL_P_PD</code>	Priority of the profile (0 to 15)	<code>__SL_P_PD=1</code>

Table 115: WiFi profile POST

For EAP connections, the following POST request can be performed.

```
POST:/api/1/wlan/profile_eap
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
```

Token	Name	Example
__SL_P_P.H	The SSID of the desired AP. Must not exceed 32 characters	__SL_P_P.H=TargetSSID
__SL_P_P.I	User identity. Must not exceed 64 characters	__SL_P_P.I=MyIdentity
__SL_P_P.J	Anonymous user identity. Must not exceed 64 characters	__SL_P_P.J=MyAnonIdentity
__SL_P_P.K	Connection password. Must not exceed 63 characters	__SL_P_P.K=MySecurePassword
__SL_P_P.L	Priority of the profile (0 to 15)	__SL_P_P.L=1
__SL_P_P.M	EAP method (TLS / TTLS / PEAP0 / PEAP1 / FAST)	__SL_P_P.M=TLS
__SL_P_P.N	Phase 2 authentication (None / TLS / MSCHAPV2 / PSK)	__SL_P_P.N=None
__SL_P_P.O	EAP provisioning type (0 / 1 / 2)	__SL_P_P.O=0

Table 116: WiFi EAP profile POST

A POST request to /api/1/wlan/profile_del deletes an existing profile and a post to /api/1/wlan/profile_del_all deletes all the stored profiles.

```
POST:/api/1/wlan/profile_del
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
Content: __SL_P_PRR=profile_index
```

9.3.4. WiFi scan

A WLAN scan for nearby access points may be triggered by posting to /api/1/wlan/en_ap_scan.

```
POST:/api/1/wlan/en_ap_scan
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
```

Token	Name	Example
__SL_P_SC2	Number of scan cycles to execute. Must be greater than zero and smaller than 2 ³²	__SL_P_SC2=64
__SL_P_SC1	Time between scan cycles in seconds	__SL_P_SC1=10

Table 117: WiFi Scan

9.3.5. WiFi connection policy

The connection policy of the device can be set by posting to `/api/1/wlan/policy_set`. Any combination of the parameters listed below can be present in a request. The options not present are turned off.

```
POST:/api/1/wlan/policy_set
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
```

Token	Name	Example
<code>__SL_P_PE</code>	Auto connect	<code>__SL_P_PE=</code>
<code>__SL_P_PF</code>	Fast connect	<code>__SL_P_PF=</code>
<code>__SL_P_PG</code>	Any P2P connect	<code>__SL_P_PG=</code>

Table 118: WiFi connection policy

9.3.6. IP configuration

Many IP settings can be configured from the HTTP interface by sending a POST request to either `/api/1/netapp/netcfg_sta`, `/api/1/netapp/netcfg_sta_ipv6` or `/api/1/netapp/netcfg_ap` URLs with some (or all) of the parameters listed below.

```
POST:/api/1/netapp/netcfg_sta
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
```

Token	Name	Example
__SL_P_N.A	Device IP in station mode	__SL_P_N.A=192.168.10.10
__SL_P_N.P	Device IP in AP mode	__SL_P_N.P=192.168.10.10
__SL_P_N.B	Device subnet mask in station mode	__SL_P_N.B=255.255.255.0
__SL_P_N.Q	Device subnet mask in access point mode	__SL_P_N.Q=255.255.255.0
__SL_P_N.C	Network gateway IP in station mode	__SL_P_N.C=192.168.10.1
__SL_P_N.T	Network gateway IP in AP mode	__SL_P_N.T=192.168.10.1
__SL_P_N.H	Address of primary DNS server in station mode	__SL_P_N.H=8.8.8.8
__SL_P_N.U	Address of primary DNS server in AP mode	__SL_P_N.U=8.8.8.8
__SL_P_N.D	IP acquisition mode for IPv4 address (LLA DHCP, DHCP or Static)	__SL_P_N.D=DHCP
__SL_P_I.S	IP acquisition mode for local IPv6 address (Stateless, Static or Stateful)	__SL_P_I.S=Static
__SL_P_I.L	Set the static IPv6 link-local address	__SL_P_I.L=fe80::ccaf:9519:0002:a5fd
__SL_P_I.G	IP acquisition mode for global IPv6 address (Stateless, Static or Stateful)	__SL_P_I.G=Stateless
__SL_P_I.B	Set the static IPv6 global address	__SL_P_I.B=2001:0db8:3c4d:0015:0000:0000:1a2f:1a2b
__SL_P_I.K	Set IPv6 primary DNS server	__SL_P_I.K=2001:4860:4860::8888

Table 119: Network configuration POST parameters

9.3.7. Ping

The device has a built-in ping utility for testing and troubleshooting network connectivity issues. The ping is started by posting the following parameters to /api/1/netapp/ping.

Token	Name	Example
__SL_P_T.A	IPv4 target address of ping requests	__SL_P_T.A=192.168.10.10
__SL_P_T.B	Size of the ping payload in bytes (from 1 to 1472)	__SL_P_T.B=1024
__SL_P_T.C	Number of packets to send	__SL_P_T.C=4

Table 120: Ping POST parameters

9.4. User setting GET APIs

The Calypso supports querying various parameters of the module through the user settings GET API. Only one parameter can be requested at a time. These requests are handled by the application processor of the Calypso module.

```
GET:/usersettings?category=<categoryname>&setting=<setting name>
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
Example Request: calypso.net/usersettings?category=uart&setting=baudrate
Response: 921600
```

The table below lists the set of possible values for the parameters.

Category	Setting	Value and usage
uart	baudrate	String with the current UART baudrate
	parity	0 - none, 1 - even, 2 - odd
	flowcontrol	true if enabled, false otherwise
	transparent_trigger	Bit mask 1etx 2etx transmit_etx timer
	transparent_timeout	timeout [ms]
	transparent_etx	ETX characters in HEX
sntp	enable	0 - enable, 1 - disabled
	min_update_interval	Time update interval [s]
	timezone	Time zone [min]
	server_addresses	String containing a list of server addresses in format [n]:[address]
ssid	default_name	SSID string
	append_mac	true or false
transparent_mode	secure_method	none, SSLV3, TLSV1, TLSV1_1, TLSV1_2, SSLV3_TLSV1_2
	socket_type	udp, tcp_server, tcp_client
	remote_address	IP address string
	remote_port	Port string
	local_port	Port string
	power_save	true or false
	skip_verify_date	true or false
	disable_cert_store	true or false
gpio	remote_lock	true or false

Table 121: User setting GET

9.5. User setting POST APIs

The user settings POST APIs allow remote configuration of vital parameters of the module including UART, SNTP and transparent mode parameters. These requests are handled by the application processor. The requests return code 204 (No content) on success. Parameter Value2 must be empty when not used.

```

POST:
/usersettings?category=<categoryname>&setting=<setting name>&value1=<value1>&value2=<value2>
>
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
Example Request: calypso.net/usersettings?category=uart&setting=baudrate&value1=921600&value2=
Response: 204
    
```

Category	Setting	Value1	Value2
uart	baudrate	String with the UART baudrate to be set	
	parity	0 - none 1 - even 2 - odd	
	flowcontrol	0 - disable, 1 - enable	
	transparent_trigger	uint32 bit mask Bit 0 - 1etx Bit 1 - 2etx Bit 2 - transmit_etx Bit 3 - timer	
	transparent_timeout	timeout (6-1000) [ms]	
	transparent_etx	ETX1 char in decimal	ETX2 char in decimal
sntp	enable	0 - disable, 1 - enable	
	min_update_interval	Time update interval [s]	
	timezone	Time zone [min] (signed)	
	server_addresses	index (0,1,2)	string with server address
ssid	default_name	SSID string	
	append_mac	0 - disable, 1 - enable	

Table 122: User setting POST part 1

Category	Setting	Value1	Value2
transparent_mode	secure_method	-1 - none 0 - SSLV3 1 - TLSV1 2 - TLSV1_1 3 - TLSV1_2 4 - SSLV3_TLSV1_2	
	socket_type	0 - udp 1 - tcp_server 2 - tcp_client	
	remote_address	IP address string	
	remote_port	Port string	
	local_port	Port string	
	power_save	0 - disable, 1 - enable	
	skip_verify_date	0 - disable, 1 - enable	
	disable_cert_store	0 - disable, 1 - enable	
gpio	remote_lock	0 - disable, 1 - enable	

Table 123: User setting POST part 2

9.6. GPIO GET APIs

The Calypso offers querying remote GPIOs of the module through the GPIO GET API. Only one parameter can be requested at a time. These requests are handled by the application processor of the Calypso module.

Parameter	value
id	0, 1, 2, 3
default	0 - Get runtime value, 1 - Get default value (value on reboot)

Table 124: GPIO GET parameters

```
GET:/gpio?id=<id>&default=<0 or 1>
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
Example Request: calypso.net/gpio?id=3&default=1
Response: {"id" : 3,
"type" : "input",
"input_config" : "nopull",
"input_value" : "high"}
```

9.7. GPIO POST APIs

The GPIO POST API allows remote configuration and control of the remote GPIOs. These requests are handled by the application processor. The requests return code 204 (No content) on success. Every GPIO post generates an event with the configured GPIO ID on the UART (see section 8.9.8).

```
POST:
/gpio?id=<id>&save=<0 or 1>&type=<type>&value1=<value1>&value2=<value2>
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
Example Request: calypso.net/gpio?id=3&type=1&save=1&value1=0&value2=0
Response: 204
```

Parameter	value
id	0, 1, 2, 3
save	0 - Apply only during runtime (volatile), 1 - Apply and save as default (non-volatile)
type	see Table 126
value1	see Table 126
value2	see Table 126

Table 125: GPIO POST parameters

type	value1	value2
0 - unused	-	-
1 - input	0 - no pull 1 - pull down 2 - pull up	-
2 - output	0 - low 1 - high	-
3 - PWM	PWM period (1-200) [ms]	PWM ratio (0-100) [%]

Table 126: GPIO types with corresponding value1 and value2 parameters

9.8. File PUT API

The file PUT API allows the user to upload files such as certificates and other credentials over the web interface. All files uploaded using this API are stored in the "/user" directory. Uploading a file with the name of an existing file overwrites the file.



The amount of available memory on the file system and the number of files supported by the file system is limited.

```
PUT:calypso.net/file/?filename=<file.txt>
Host: calypso.net
Content-Type: multipart form data
```

9.9. Custom GET API

The custom GET API provides a web interface to resources stored on the host MCU. A custom GET request with an ID is forwarded to the host MCU over the UART. The host must reply to the request within 3 s. The response is then forwarded to the HTTP client.

```
GET:/custom?id=<id>
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
```

Example: The request

```
calypso.net/custom?id=greeting
```

generates the following event (see section 8.9.8) on the UART:

```
+eventhttpget:greeting
```

The host must reply to the request with the command `AT+httpcustomresponse` within 3 s:

```
AT+httpcustomresponse=0,5,hello
```

Sending the above command results in the following response on the HTTP client side:

```
Code: 200 OK
Content-Type: application/x-www-form-urlencoded
Content-Length: 5
Content: hello
```

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

Table 127: AT+httpcustomresponse

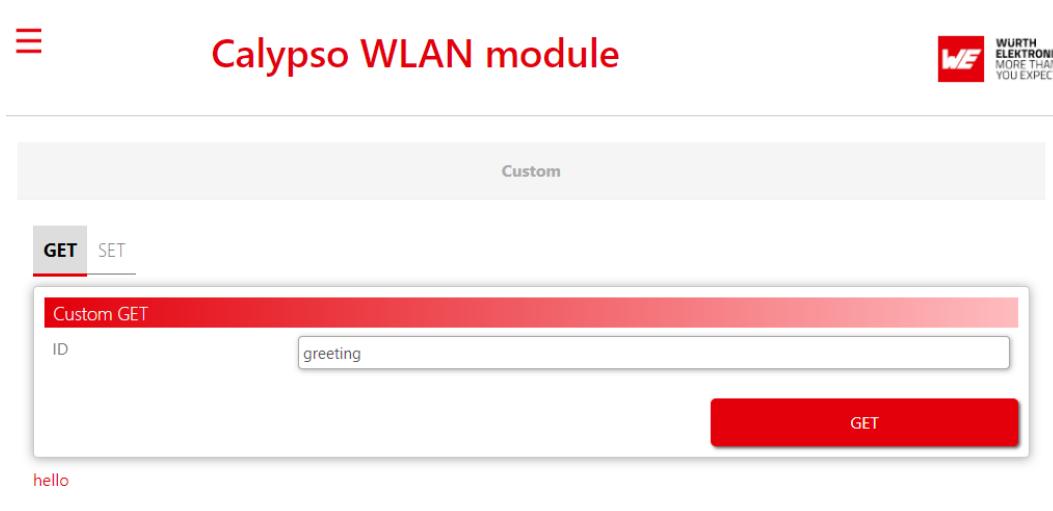


Figure 12: Custom GET webpage

9.10. Custom POST API

The custom POST API can be used to send values from the web interface to the host MCU.

```
POST:
/custom?id=<id>&value=<value>
Host: calypso.net
Content-Type: application/x-www-form-urlencoded
Example Request: calypso.net/custom?id=test&value=helloworld
Response: 204
```

The above example request, results in the following event (see section 8.9.8) on the UART interface.

```
+eventcustom:1,test,helloworld
```

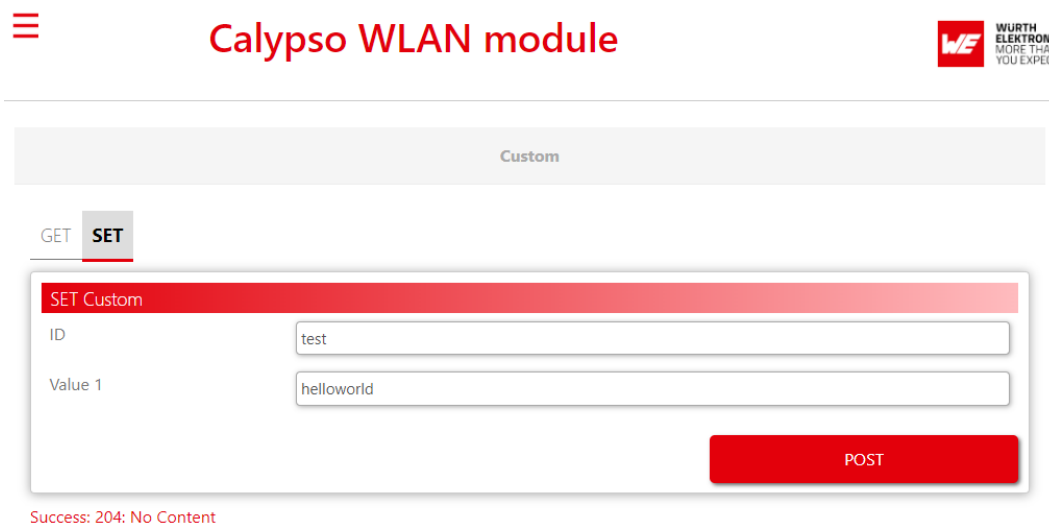


Figure 13: Custom POST webpage

10. Provisioning

To enable easy provisioning when integrated into 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 HTTP 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.

10.1. Start in 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 an interval of 1 s. The module has created an access point with an SSID "calypso_" followed 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".

10.2. Add WLAN profile

On the device connected to the Calypso AP, open the website "calypso.net" in a browser.

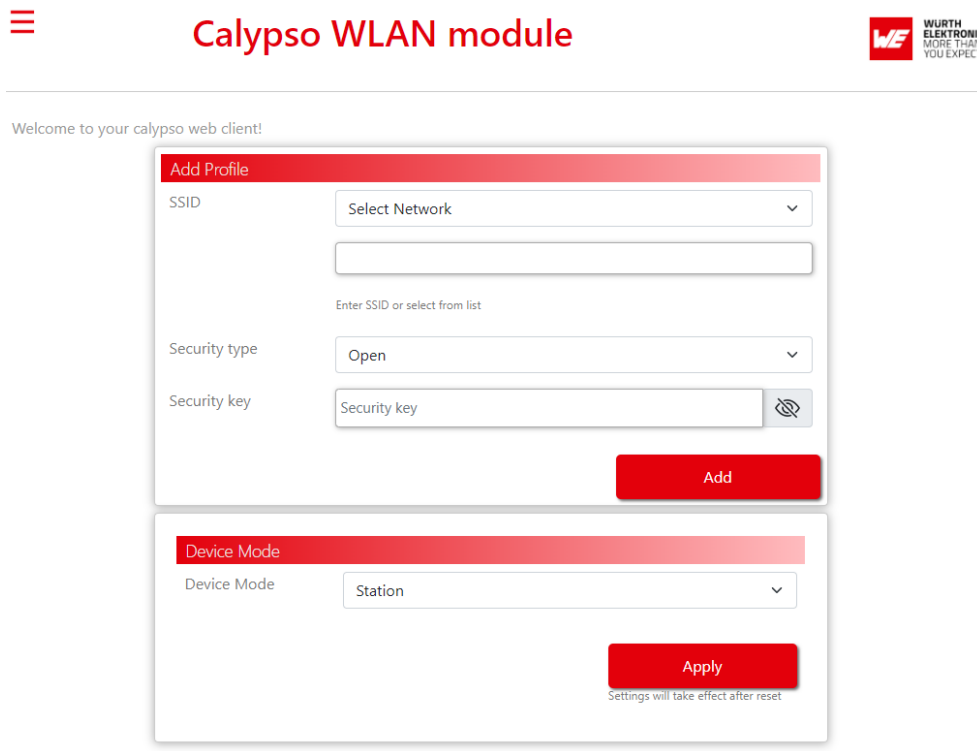


Figure 14: Provisioning main page

To save a WLAN profile in the module, select the SSID from the dropdown menu or enter the same manually in the text field. Check the correct security type, enter the key if necessary and click on the "Add" button. A pop-up appears confirming the addition of the profile.

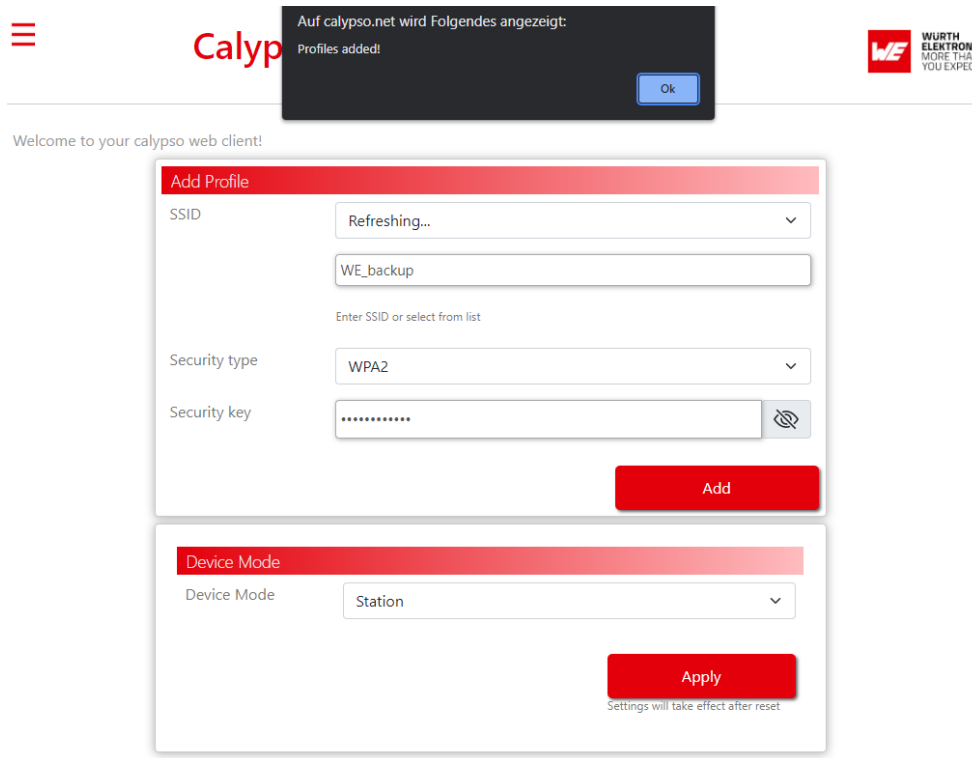


Figure 15: Provisioning main page

In provisioning mode, the module is set to start as an AP. In order to start the module in station mode, select "Station" in the "Device Mode" drop down menu and click on "Apply". In the default settings, the parameter "WLAN policy connection" (see chapter 8.2.6) is set to "auto|fast", meaning that the device automatically tries to connect to the access points defined in the module's profiles. Thus, after adding the profile to the module, a restart has to be performed. This can be done by sending the following command

```
AT+reboot
```

or pressing 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
```

10.3. Upload files

The provisioning pages on the Calypso offer the possibility to upload files such as certificates to the the on-board file system. In order to upload a file, the following steps must be performed:

- On the home page click on the "File upload" option on the menu bar.
- Clicking the "Choose file" button opens the file browser on the device.
- Browse and select the file to be uploaded.
- Click on "Upload file" to save the file to the Calypso file system.

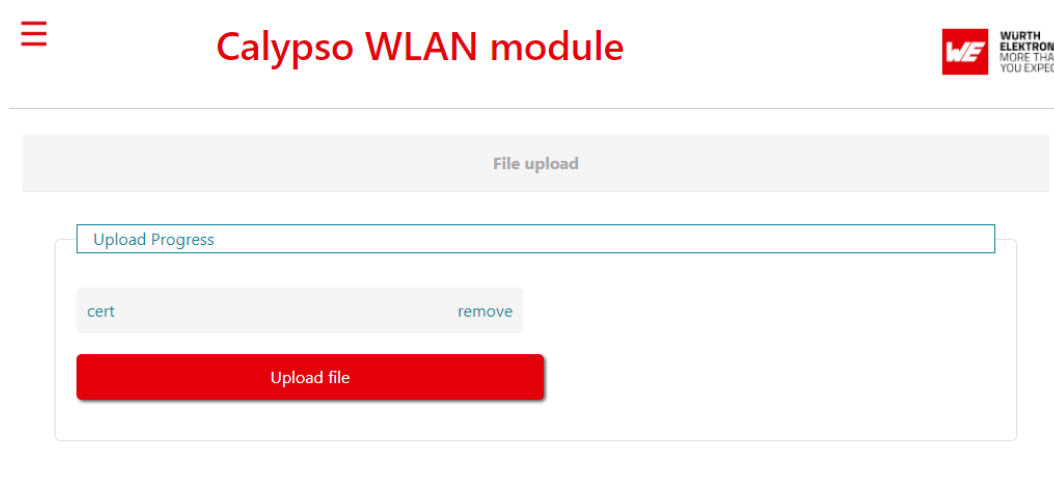


Figure 16: File upload



All files uploaded through the webpage will be saved under the path "/user".

11. 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.

11.1. 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 9 shows the work-flow for UDP communication.

11.1.1. Prerequisites

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

1. Two Calypso EV-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.

11.1.2. 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 of 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 command 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
```

11.2. TCP communication

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

11.3. Secure socket communication

SSL/TLS layer provides added security features like server authentication and end-to-end encryption. This example describes the creation of an SSL/TLS server as well as client on the 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 EV-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 is stored on the file system of the server module.
5. Root CA certificate is stored on the file system of the client module.



This example uses self-signed certificates to establish a TLS connection.

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).

11.3.1. Write certificate and key files

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 created 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=dummy-trusted-cert,WRITE|CREATE,4096
+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 aforementioned certificate file onto the module, the `AT+fileWrite` command can be used. The required command arguments include the file descriptor, the data length and the file data itself.

```
AT+fileWrite=1966156880,0,0,1024,-----BEGIN CERTIFICATE-----567587687576586979...  
...87585857487467325376986-----END CERTIFICATE-----  
+filewrite:1024  
OK
```

After the transmission of the data to the radio module has been finished, the file handle must be closed using the `AT+fileClose` command.

```
AT+fileClose=1966156880,,  
OK
```

For this example, the following files need to be written to the file system:

- Server certificate - "dummy-trusted-cert" on the server module.
- Server key - "dummy-trusted-cert-key" on the server module
- Root CA certificate - "dummy-root-ca-cert" on the client module.

11.3.2. Set-up SNTP client

SSL/TLS connection involves mutual authentication by verification of certificates. In order to validate a certificate, the system time on the module needs to be up to date. This can be done by configuring and enabling the on-board SNTP client using the following commands:

```
AT+netappset=sntp_client,enable,1  
OK  
AT+netappset=sntp_client,time_zone,60  
OK  
AT+netappset=sntp_client,server_address,0,sntp.server.com  
OK  
AT+netappupdatetime  
OK  
AT+get=general,time  
+get:18,19,30,24,11,2021  
OK
```



In case the time is not up-to-date, an error is returned when trying to establish a connection, `SL_ERROR_BSD_ESECDATEERROR (-461L)`.

11.3.3. Create an SSL/TLS server

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

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

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

2. The next step is to set the security method to be used by updating the socket options.

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

3. The certificate "dummy-trusted-cert" and the key "dummy-trusted-cert-key" are 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
```

11.3.4. 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 secure TCP socket with the following command. Note the socket ID for future reference.

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

2. The next step is to set the security method to be used by updating the socket options.

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

3. The certificate "dummy-root-ca-cert" is configured as the root CA certificate for the server certificates using the following commands.

```
AT+setsockopt=1,socket,SECURE_FILES_CA_FILE_NAME,dummy-root-ca-cert
OK
```

4. In this example, self-signed certificates are used and hence the use of root CA certificate catalogue needs to be disabled.

```
AT+setsockopt=1,socket,disable_certificate_store,
OK
```

5. The client can now connect to the server using the following command.

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


6. 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
```

11.3.5. Secure data transfer

1. The connection 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 received using the AT+recv command in the client.

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

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

```
AT+send=0,0,32,iuwIHSis5xTttzffbtfhjt678pSHJA  
OK
```

6. Which is received using the AT+recv command in the server with socket ID 1.

```
AT+recv=1,0,32  
OK  
+recv:1,0,32,iuwIHSis5xTttzffbtfhjt678pSHJA  
OK
```

7. Close the sockets using the AT+close command and corresponding socket ID.

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

11.4. WiFi direct example

The WiFi direct standard enables peer-to-peer communication between two compatible devices without the need for an infrastructure AP. WiFi 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 one assumes the role of a client. The discovery of devices is done by sending (and listening for) broadcasting packets on channels 1, 6 and 11. This section demonstrates the WiFi direct capabilities of the Calypso module by connecting two Calypso EV-Boards over WiFi direct.

11.4.1. Prerequisites

The following hardware is required to go through this WiFi direct example.

1. Two Calypso EV-Boards.
2. Computer with a serial terminal emulator like Tera Term.

11.4.2. Auto connection setup

In this section, the steps required to establish an automatic WiFi direct connection are described. First of all, the P2P settings of both devices have to be configured. The devices are configured to connect to the first P2P peer 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. For simplicity, we choose 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 and restart the network processor.

```
AT+wlanSetMode=P2P  
AT+stop=0  
AT+start
```

And scan for P2P devices.

```
AT+wlanScan=0,5
```

Note that the first scan command initiates a scan and hence returns an error code `SL_ERROR_WLAN_GET_NETWORK_LIST_EAGAIN (-2073)` before listing the available P2P devices. 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 11.1 and 11.2 to do so.

After data has been transmitted/received, the connection can be closed again.

```
AT+wlanDisconnect
```

11.4.3. Manual connection setup

This chapter describes how to manually setup 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 have to be configured. Here 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 and restart the network processor.

```
AT+wlanSetMode=P2P
AT+stop=0
AT+start
```

Repeat the previous steps with module B, using "client" instead of "group_owner" in the AT+wlanPolicySet command.

```
AT+wlanPolicySet=P2P,client,active
```

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

```
AT+wlanScan=0,5
```

Note that the first scan command initiates a scan and hence returns an error code SL_ERROR_WLAN_G (-2073) before listing the available P2P devices. 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,pcb
+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,pcb
+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 11.1 and 11.2 to do so.

After data has been transmitted/received, the connection can be closed again.

```
AT+wlanDisconnect
```

11.5. 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

11.5.1. 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 created on the radio module 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 aforementioned html file onto the module, the AT+fileWrite command can be used. The required command arguments include 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 handle must be closed using the AT+fileClose command.

```
AT+fileClose=1966156880,,  
OK
```



More complex websites can be put to the secure file system by uploading all the required files 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 been stored on the module.

11.5.2. 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, the URL is "192.168.1.104/mytest.html".

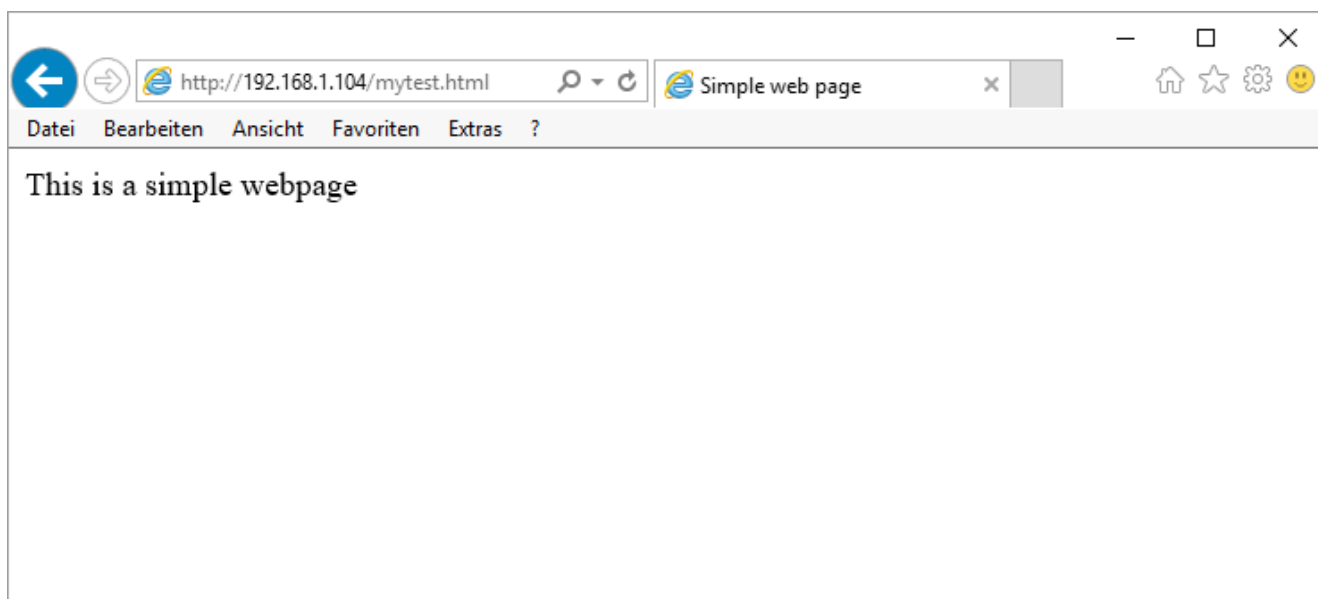


Figure 17: Test page

11.5.3. 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" followed by its MAC, the password is "calypsowlan" and the domain is "calypso.net".

Furthermore, we like to use a custom 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.

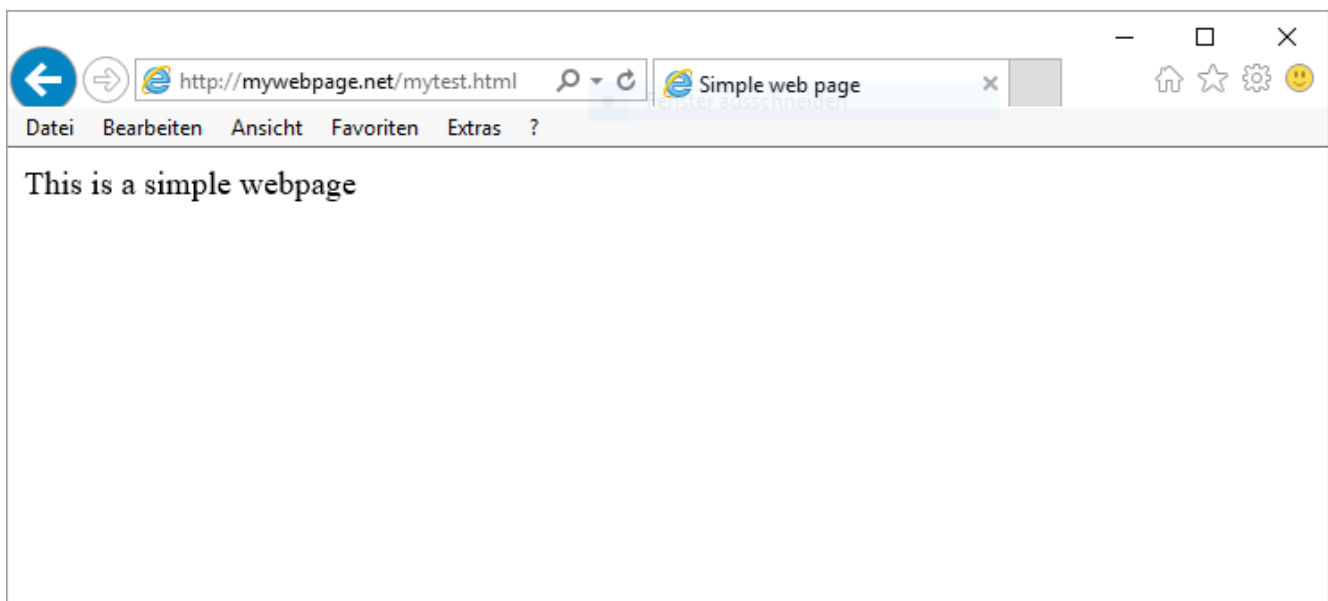


Figure 18: Test page



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

12. Connection to Microsoft Azure IoT Central

The Calypso WLAN module along with *Wireless connectivity SDK* is certified to enable Plug and Play connectivity to the Microsoft Azure's *IoT central* Platform-as-a-service. IoT Plug and Play enables solution builders to integrate any IoT device with their solutions without any manual configuration. Being IoT plug and play compatible, the Calypso WLAN module can be seamlessly integrated into any Azure based IoT solution.

Being IoT Plug and Play, the Calypso module,

- automatically enables secure connectivity to the Azure IoT Hub.
- supports secure device provisioning through the Device Provisioning Service(DPS).
- uses a standard device model that enables seamless integration into any IoT solution using the Azure digital twins
- can be used as a connectivity sub-component in any Azure certified IoT end device.

For detailed information on using the Calypso with Microsoft Azure IoT, please refer to the quick start guide of the wireless connectivity SDK under https://github.com/WurthElektronik/WirelessConnectivity-SDK_STM32.

In order to enable rapid prototyping of IoT applications, Würth Elektronik eiSos offers FeatherWing development boards that are open source and fully compatible with the Feather form factor from Adafruit. The Calypso IoT design kit with pre-installed firmware enables easy creation and evaluation of a secure end-to-end IoT solution using Microsoft's Azure IoT central. More information under, <https://we-online.com/featherwings>.

13. Timing parameters

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

13.1. Hard reset

A hard reset of 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	2000	ms

Table 128: Start-up time

13.2. 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 129: 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 an AP on reboot/wake-up.

14. Firmware update

Calypso supports secure firmware-over-the-air (FOTA) updates to enable easy update of the module's firmware in the field. The FOTA can be performed in two modes:

- **Station mode (default):** In this mode, the module connects to an infrastructure AP and any 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.
- **AP mode:** In this mode, the module starts as an AP allowing a single client connection. Any WiFi enabled device can connect to this AP and upload an encrypted image (provided by Würth Elektronik eiSos) using the on-board web server.



FOTA update in AP mode is supported in firmware v1.9.0 or higher. To update the module from v1.3.0 to v1.9.0 the station mode has to be used.

14.1. Prerequisites

When using station mode,

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 module is configured to start in station mode (see chapter 8.2).
4. 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 JavaScript browser.
5. 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.

When using AP mode,

1. 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 JavaScript browser.
2. The module must be configured to start in AP mode (see chapter 8.2).
3. 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.



It is recommended to use the Chrome browser in incognito mode with JavaScript enabled.

14.2. Update procedure

14.2.1. 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.

Station 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 5 s. Please solve the connection issue before continuing.

```
+eventota:info, "Starting_FOTA_in_Station_role"
+eventota:info, "Device_is_configured_in_default_state_as_station"
+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_FOTA_in_Station_role"
+eventota:info, "Device_is_configured_in_default_state_as_station"
+eventota:connect, Calypso Test AP, 2c:91:ab:bb:ed:9a
+eventota:ipacquired, 192.168.178.45, 192.168.178.1
```

AP mode:

- If correctly configured, the Calypso creates an AP with preconfigured SSID (Default: calypso_[MAC address]). *STATUS_IND_0* LED blinking at 1 Hz indicates that the module is waiting for a client to connect

```
+eventota:info, "Starting_FOTA_in_AP_role"
+eventota:info, "Device_is_configured_in_default_state_as_AP"
```

- Once a client connects to the Calypso AP, the following events appear.

```
+eventota:connect, , 7e:7e:45:bf:8d:26
+eventota:ipacquired, 10.123.45.2, 10.123.45.1
```

14.2.2. Connection to the update device

Thereafter, 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 ON.

```
+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 OTA webpage as follows.

- In station mode - "[module ip]/ota.html". For example, <http://192.168.1.101/ota.html> in case the module's IP is 192.168.1.101
- In AP mode - "calypso.net/ota.html".



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

On successful connection, the webpage with information about the module is displayed in the web browser.

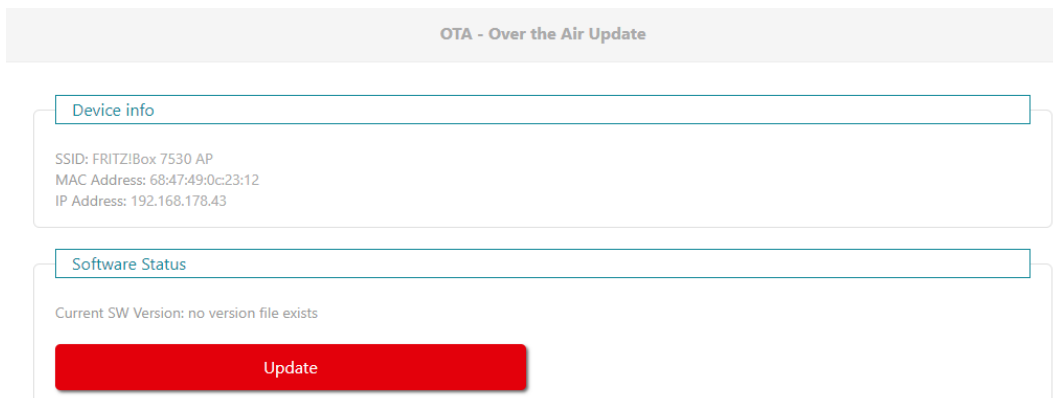


Figure 19: OTA webpage

14.2.3. Upload the update package

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



Using browsers other than Chrome, the progress bar is occasionally found to not update correctly.

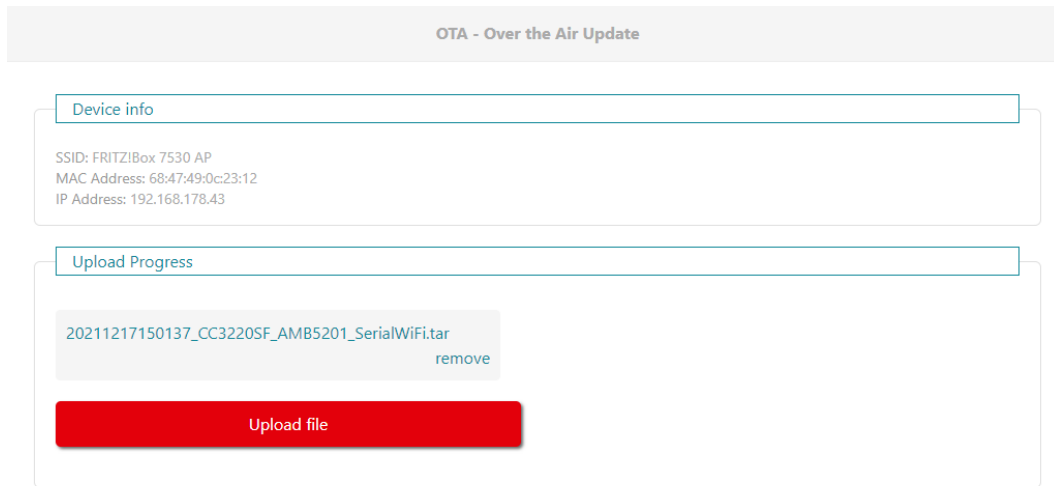


Figure 20: OTA webpage upload

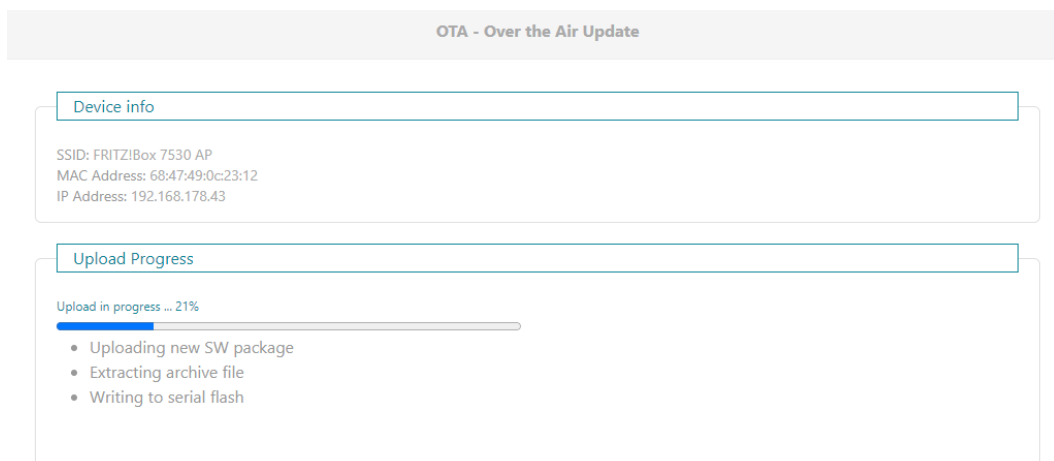


Figure 21: OTA in progress

14.2.4. 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.



During update from version 1.3.0 to version 1.9.0 or higher, the finalize button does not appear automatically due to a switch from HTTPS to HTTP. Please observe the logs over the UART and load the page "http://[module ip]/ota.html" to check if the update was successful.

OTA - Over the Air Update

Device info

SSID: FRITZ!Box 7530 AP
MAC Address: 68:47:49:0c:23:12
IP Address: 192.168.178.43

Upload Progress

Upload in progress ... 100%

- Uploading new SW package
- Extracting archive file
- Writing to serial flash

- Rebooting...
- Testing new SW package
- waiting...

Calypso is successfully updated

Finalize

Figure 22: Finalize OTA

15. Firmware history

The Calypso firmware is based on the SimpleLink WiFi CC3220 software development kit (SDK) from Texas Instruments with the corresponding features as well as known issues. A list of the versions of different components used for the current Calypso firmware version is as shown below.

Description	Version
SimpleLink SDK Version	5.20.00.06
NWP Version	3.20.0.1
MAC Version	2.7.0.0
PHY Version	2.2.0.7
ROM Version	0

15.1. Release notes

Version 0.x.x "Engineering"

Version 1.0.0 "Release"

- First release of the product.

Version 1.1.0 "Internal Release"

- Fixed issue KI001

Version 1.2.0 "Release"

- Fixed issue KI002
- The root catalog for secure communication was updated as a maintenance action. The new catalog adds support for certificates signed by Amazon.

Version 1.3.0 "Release"

- Fixed issue KI003
- The root catalog for secure communication was updated as a maintenance action. The new catalog removes retired certificates.

Version 1.9.0 "Closed beta"

- Replaced UART terminal mode with the transparent mode with the following *APP_MODE* pin configuration
 - *APP_MODE_0* = HIGH
 - *APP_MODE_1* = HIGH
- New WiFi security modes WPA2(CCMP), WPA3
- Added UART flow control and support of 3 MBaud
- Added remote GPIO operation

- Change AT+wlanSet AT command for option SSID
- Additional power save mode implemented
- FOTA in AP as well as station mode
- RESTful API for remote configuration and control
- Fixed KI005

Version 2.0.0 "Release"

- Bug fixes and performance optimization
- New and updated features and functions, see chapter [New features - version ≥ 2.0.0](#)

Version 2.1.0 "Internal release"**Version 2.2.0 "Release"**

- Fixed issue KI007
- Added features to support easy connectivity to Microsoft Azure IoT central.

15.2. New features - version \geq 2.0.0

Feature	Description
UART-Wi-Fi-bridge	A transparent mode that makes use of a pre-configured connection and allows to upgrade a serial cable to IP based traffic.
Power save feature	The Calypso stays connected and available/online via WiFi but will consume an average current of below than 2 mA during passive phases. This option is also available in the transparent mode.
WiFi Security modes	Added WPA3, WPA2 (CCMP), WPA Enterprise (802.1x) security modes.
Remote control pins	The Calypso has 4 Remote control Pins available as digital input/output. Two of these pins can also be used as PWM.
WiFi and network configuration via website	RESTful APIs support, example website included.
GPIO configuration and control via website	RESTful APIs support, example website included.
Customizable web pages	Custom APIs to display custom values (ex: sensor values) on the website and send commands to host MCU from the website.
Enhance FOTA support	Support for FOTA in Soft-AP mode (No AP infrastructure needed).
Redesigned web pages	New enhanced web pages for provisioning and FOTA.
Browser compatibility	Enhanced browser compatibility for web pages hosted on the Calypso.
General updates	TI-RTOS, SDK, Drivers and Service pack updates.
Improved overall performance	Lower latency, higher throughput, support to 3 MBaud on the UART interface with flow control

Table 130: Key features v2.0.0

15.3. Known issues

Index	Details	Affected versions
KI-001	<p>Description: AT+mqttSet does not work as expected.</p> <p>Workaround: Firmware update to \geq v1.2.0</p>	\leq 1.0.0
KI-002	<p>Description: Responses to AT commands or events that use a bit-field as a parameter, had a missing delimiter (","), in the case that the bit-field had a value 0 (i.e. all bits '0'). For example, the response to AT+fileGetFileList had a missing delimiter ("") in cases where the bit-field "file properties" is 0.</p> <p>Workaround: Firmware update to \geq v1.2.0</p>	\leq 1.1.0
KI-003	<p>Description: When transmitting data in binary format, specific data bytes (0x20 and 0x22), if present in the payload, cause errors. This effect is seen on the following AT commands: AT+sendTo, AT+send, AT+httpSendReq, AT+httpSetHeader, AT+fileWrite, AT+mqttPublish and AT+mqttSet.</p> <p>Workaround: Use base64 encoding or firmware update to \geq v1.3.0</p>	\leq 1.2.0
KI-004	<p>Description: The module enters unstable state when using the fast connect feature and connecting to certain access points. This behaviour is observed in cases where the module and the AP are configured to use more than one WPA2 authentication method. The issue is generally observed in access points with fast roaming or 802.11r support.</p> <p>Workaround: Either disable fast connect in case such an access point is used or make sure a maximum of one WPA2 security method is enabled in the access points configuration. A custom firmware with a use-case specific fix can be implemented on request (see chapter 16).</p>	\leq 1.3.0
KI-005	<p>Description: Calypso does not respond to commands on the UART when sent with a very short guard interval (see section 6.3).</p> <p>Workaround: Use a guard interval of at least 1 ms or update to firmware version \geq 1.9.0.</p>	\leq 1.3.0

<p>KI-006</p>	<p>Description: The command <code>AT+wlanget=connection</code> returns wrong WiFi security type WEP in cases where the actual security type is WPA2+ or WPA3. This is due to a enumeration mapping inside the NWP and cannot be changed/corrected by the Application.</p> <p>Workaround: As the user is well aware of the security mode and credentials for any AP a connection is intended to, the user shall accept the return value WEP in case of WPA2+ or WPA3 as used security mode.</p>	<p>≤ 2.0.0</p>
<p>KI-007</p>	<p>Description: Files uploaded using the file upload API lose their integrity.</p> <p>Workaround: Use AT commands to upload files or update to firmware version $\geq 2.2.0$.</p>	<p>≤2.0.0</p>

16. Custom firmware

16.1. Custom configuration of standard firmware

The configuration of the standard firmware includes adoption of the non-volatile user settings (see chapter 6) to customer requirements and creating a customized product based on the standard product.

This variant will result in a customer exclusive module with a unique ordering number. It will also freeze 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 release procedure.

16.2. 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 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 I²C 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.

16.3. 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 site.

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.

16.4. Contact for firmware requests

Please contact your Business Development Engineer (BDM) or WCS@we-online.com for quotes regarding these topics.

17. Design in guide

17.1. Advice for schematic and layout

For users with less RF experience it is advisable to closely copy the relating EV-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.).



The use of an external reset IC should be considered if one of the following points is relevant:



- The slew rate of the power supply exceeds the electrical specifications.
- The effect of different current consumptions on the voltage level of batteries or voltage regulators should be considered. The module draws higher currents in certain scenarios like start-up or radio transmit which may lead to a voltage drop on the supply. A restart under such circumstances should be prevented by ensuring that the supply voltage does not drop below the minimum specifications.
- Voltage levels below the minimum recommended voltage level may lead to malfunction. The reset pin of the module shall be held on LOW logic level whenever the VDD is not stable or below the minimum operating Voltage.
- Special care must be taken in case of battery powered systems.

- 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 8231606A 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.).

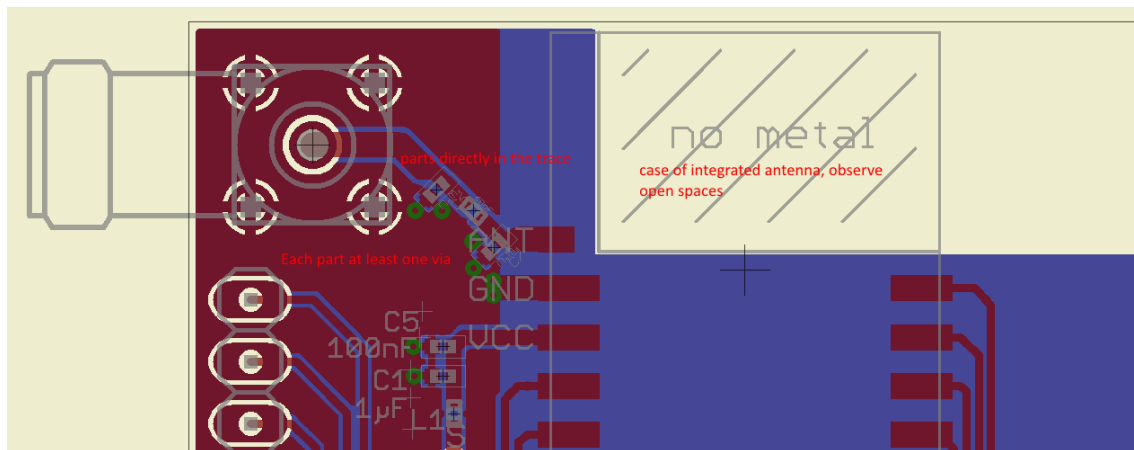


Figure 23: Layout

- 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 EV-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.

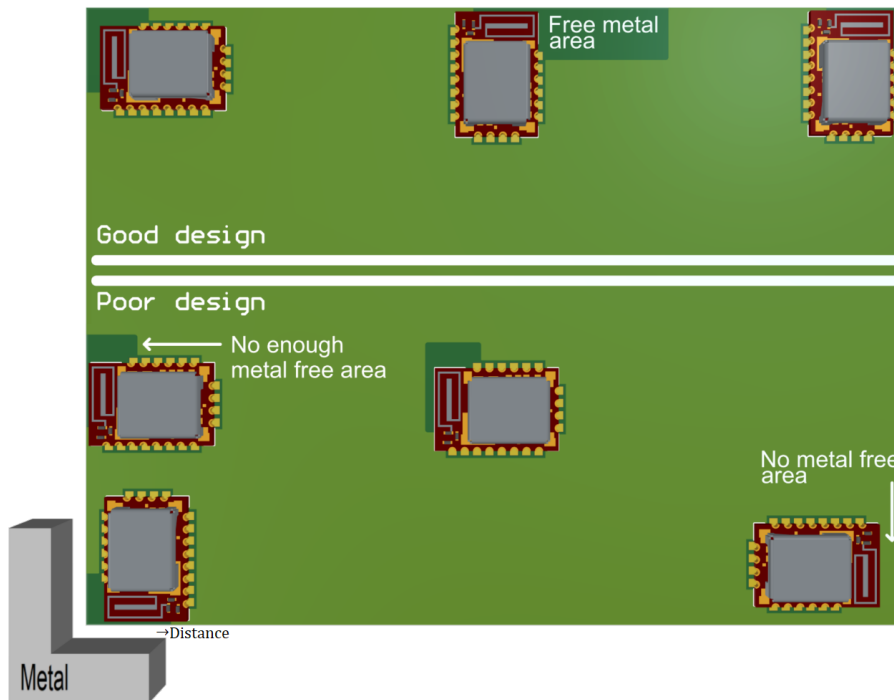


Figure 24: Placement of the module with integrated antenna

17.2. Designing the antenna connection

The antenna should be connected with a 50 Ω line. This is needed to obtain impedance matching to the module and avoids reflections. Here we show as an example how to calculate the dimensions of a 50 Ω line in form of a micro strip above ground, as this is easiest to calculate. Other connections like coplanar or strip line are more complicated to calculate but can offer more robustness to EMC. There are free calculation tools available in the internet.

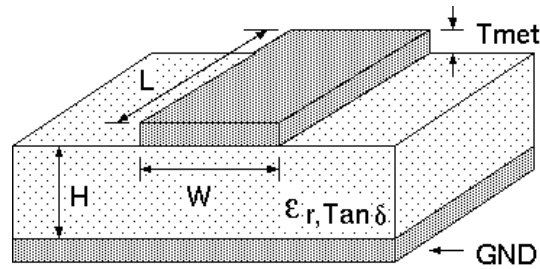


Figure 25: Dimensioning the antenna connection as micro strip

The width W for a micro strip can 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 $\epsilon_r = 4.3$, a height $H = 1000 \mu\text{m}$ and a copper thickness of $T_{met} = 18 \mu\text{m}$ will lead to a trace width of $W \sim 1.9 \text{ mm}$. 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 $3 \times W$ 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.

17.3. 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 as far as possible from large metal objects to avoid electro-magnetic field blocking.

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

17.3.1. 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.

17.3.2. 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.

17.3.3. 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 EV 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.

17.3.4. Antennas provided by Würth Elektronik eiSos

17.3.4.1. 2600130021 - HIMALIA dipole antenna



Figure 26: HIMALIA dipole antenna

Due to the fact that the antenna has dipole topology, there is no need for an additional ground plane. 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. [$^{\circ}\text{C}$]	-40 – +80

Special care must be taken for FCC certification when using this external antenna to fulfill the requirement of permanently attached antenna or unique coupling, for example by using the certified dipole antenna in a closed housing, so that it is possible to remove it only through professional installation.

18. Reference design

Calypso was tested and certified on the corresponding Calypso EV-Board. For the compliance with the EU directive 2014/53/EU Annex I, the EV-Board serves as reference design. For the FCC it serves as trace design.

This is no discrepancy due to the fact that the EV-Board itself does not fall within the scope of the EU directive 2014/53/EU Annex I as the module is tested on the EV-Board, which is also the recommended use.

Further information concerning the use of the EV-Board can be found in the manual of the Calypso EV-Board.

18.1. EV-Board

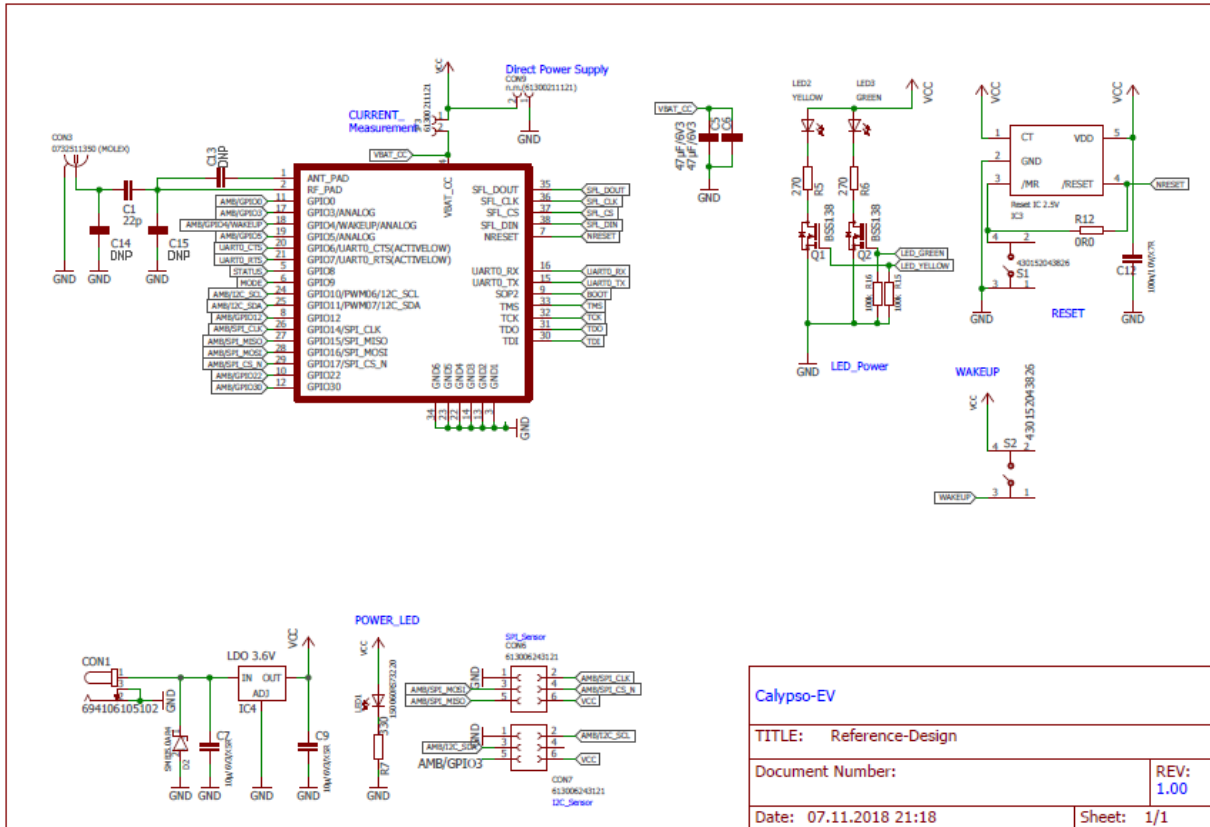


Figure 27: Reference design: Schematic, most important parts

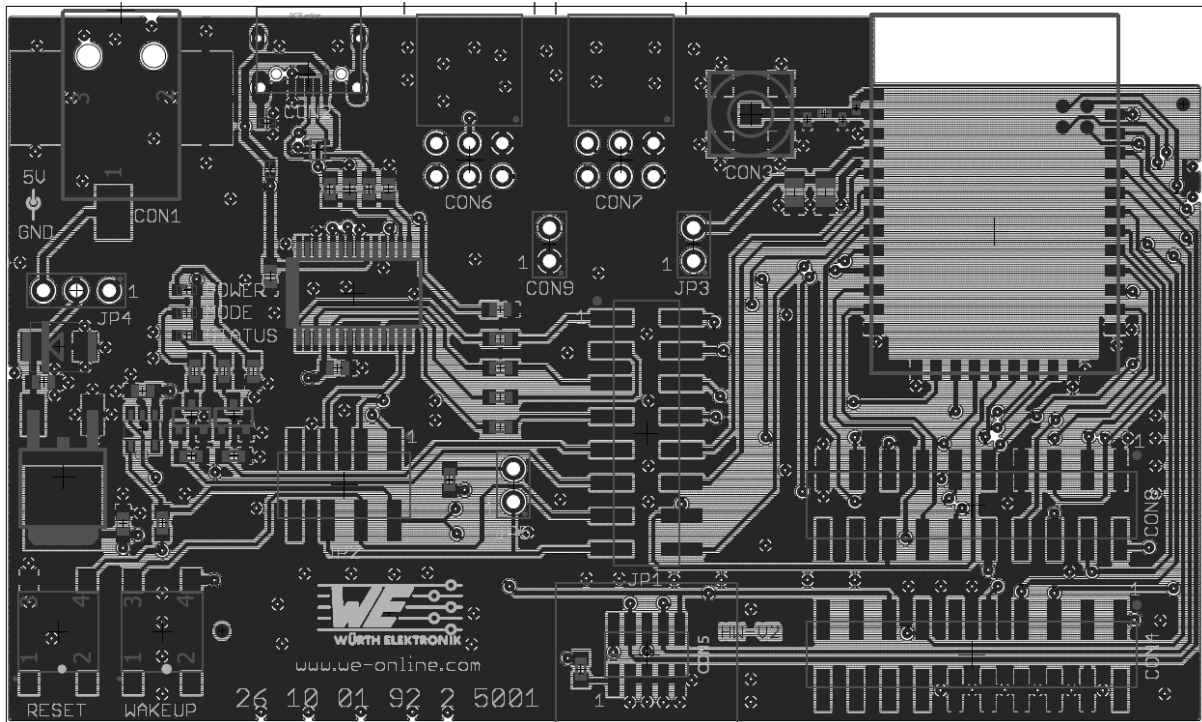


Figure 28: Reference design: Layout

18.2. Radiation characteristic of the module's internal antenna

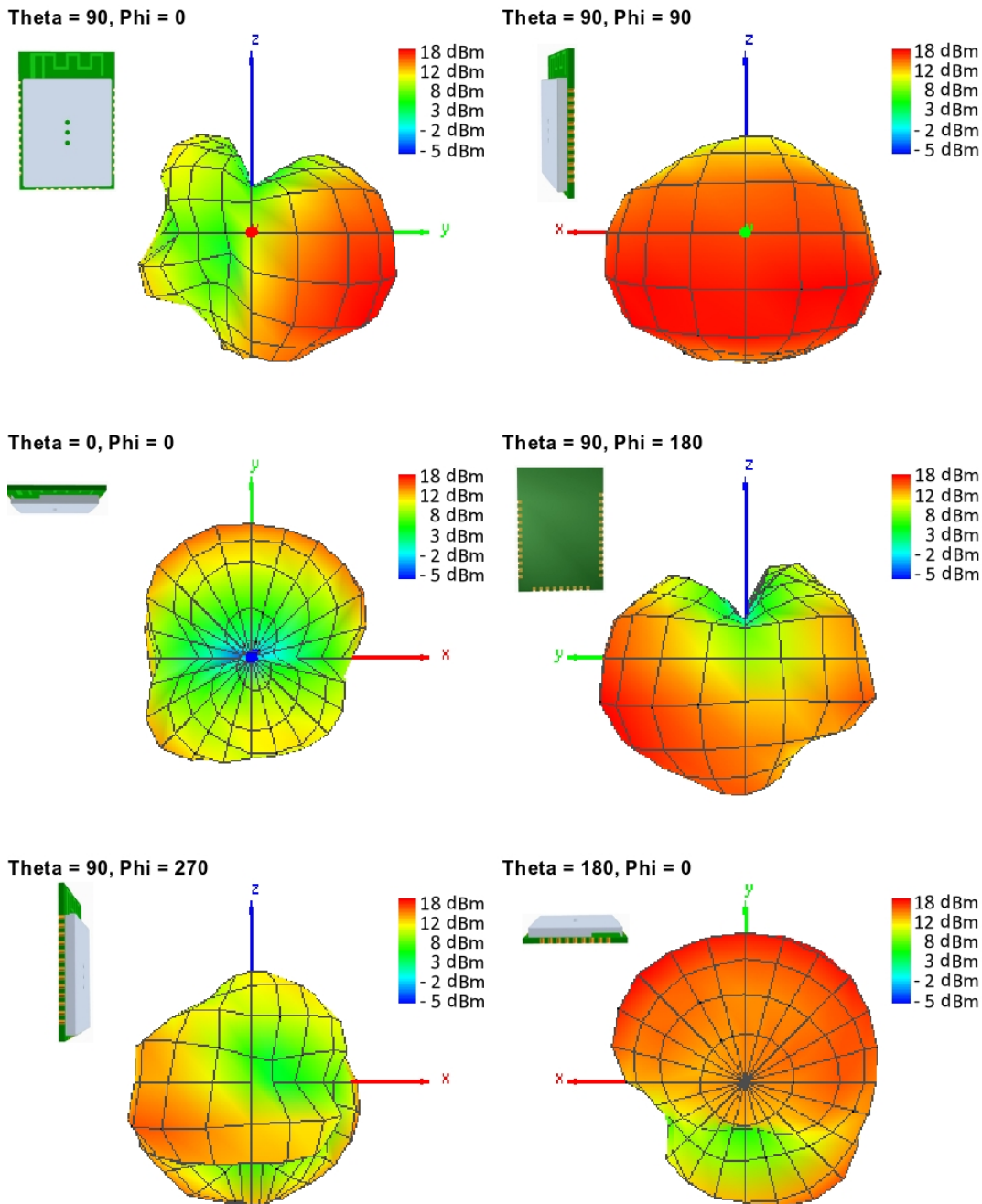


Figure 29: Antenna characteristic of the module with its integrated antenna measured on the official EV-Board



It is important to be aware that size and shape of the ground plane as well as the placement of module has influence on the radiation pattern.

18.3. Design Guide for FCC ID R7T1001102

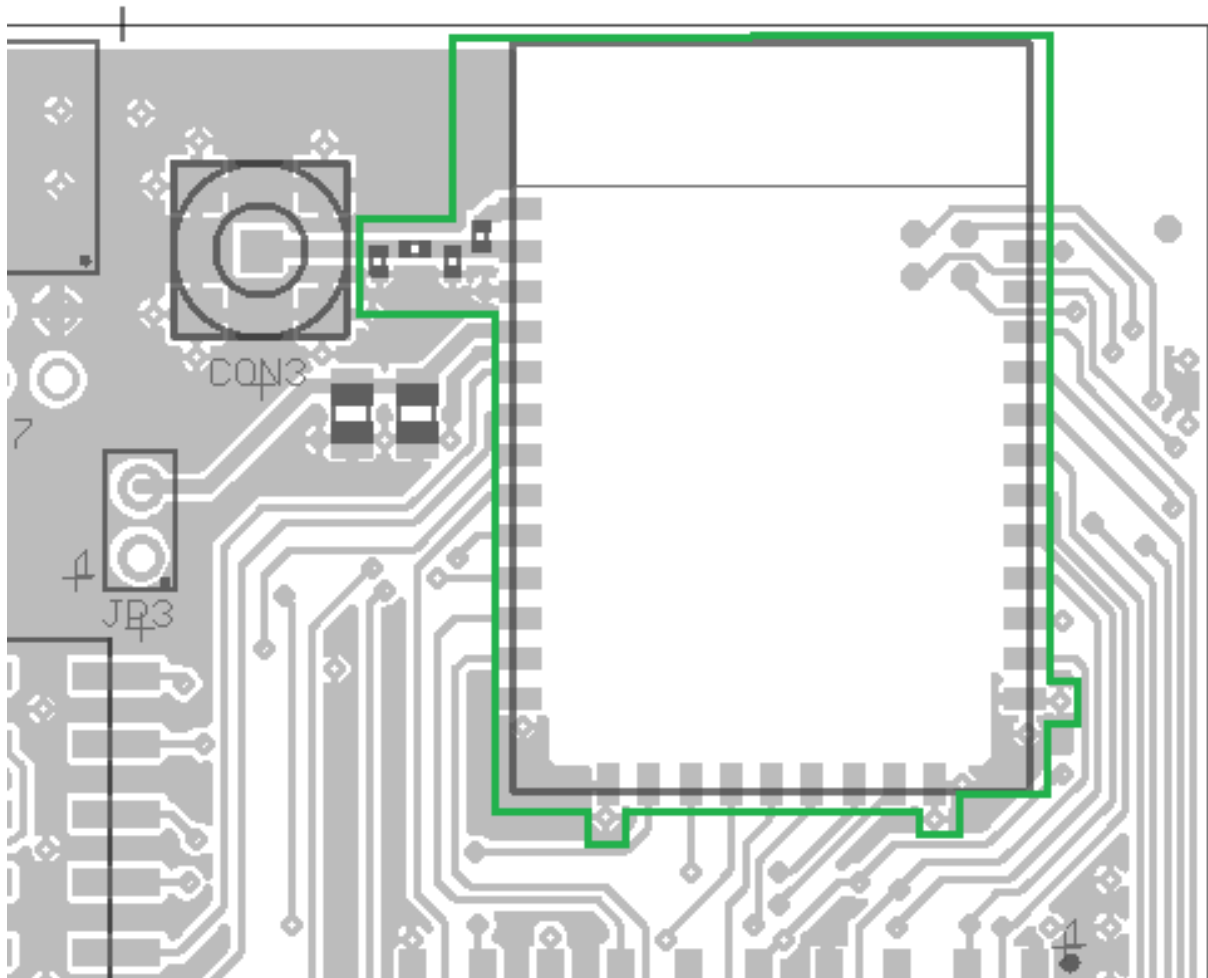


Figure 30: Close-up: Layout

Nr	Copper	Isolation
1	0.018mm	0.36mm
2	0.035mm	0.71mm
3	0.035mm	0.36mm
16	0.018mm	
Gesamt: 1.536mm		

Figure 31: 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.

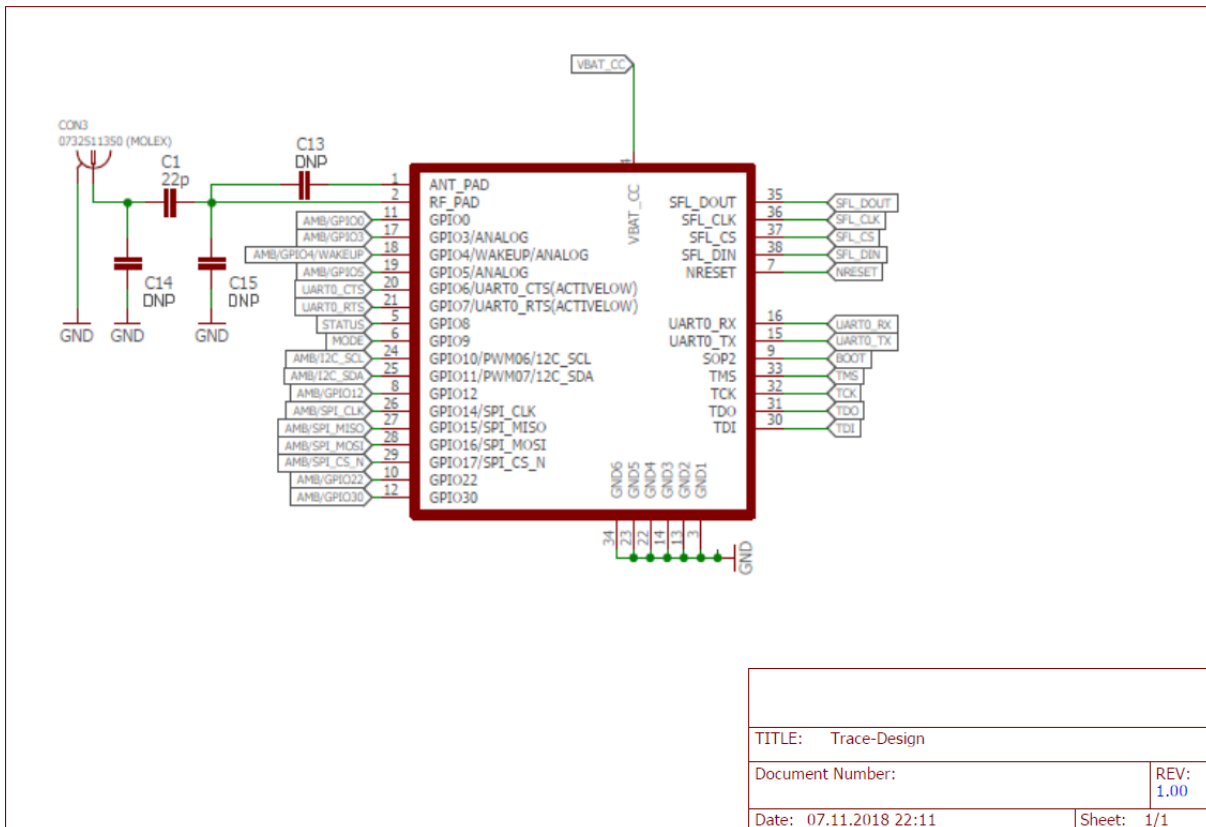


Figure 32: Close-up: Schematic

Two variants of the Calypso are certified:

- **Integrated PCB antenna:** Not placing C1, C14 and C15, but placing 0 Ω at C13. C13 connects the RF pad, the radio signal to/from the transceiver, to the ANT pad, the connection to the module’s integrated PCB antenna.
- **External antenna:** Placing 22pF at C1, not placing C13, C14 and C15 and connecting C1 with a 50 Ω line to a dipole antenna. For the certification the antenna from 17.3.4.1 was used with a peak gain of 2.8 dBi.

Special care must be taken when using an external antenna to fulfil the requirement for FCC Certification of permanently attached antenna or unique coupling for example by using the certified dipole antenna in a closed housing, so that only through professional installation it is possible to remove it.

18.4. 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).

The OTA mode enables security updates of the firmware and/or HTTP server certificates via radio and provisioning mode may be used for configuring the module.



To manually switch to OTA mode or provisioning mode, it is strongly recommended to make the pins *APP_MODE_0* and *APP_MODE_1* accessible on the custom PCB. Otherwise the update of certificates and the firmware is not possible.

19. Manufacturing information

19.1. 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.

19.2. Soldering

19.2.1. 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	$T_{S \text{ Min}}$	150 °C
Preheat temperature Max	$T_{S \text{ Max}}$	200 °C
Preheat time from $T_{S \text{ Min}}$ to $T_{S \text{ Max}}$	t_S	60 - 120 seconds
Ramp-up rate (T_L to T_P)		3 °C / second max.
Liquidous temperature	T_L	217 °C
Time t_L maintained above T_L	t_L	60 - 150 seconds
Peak package body temperature	T_P	245 °C
Time within 5 °C of actual peak temperature	t_P	20 - 30 seconds
Ramp-down Rate (T_P to T_L)		6 °C / second max.
Time 20 °C to T_P		8 minutes max.

Table 132: Classification reflow soldering profile, 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.

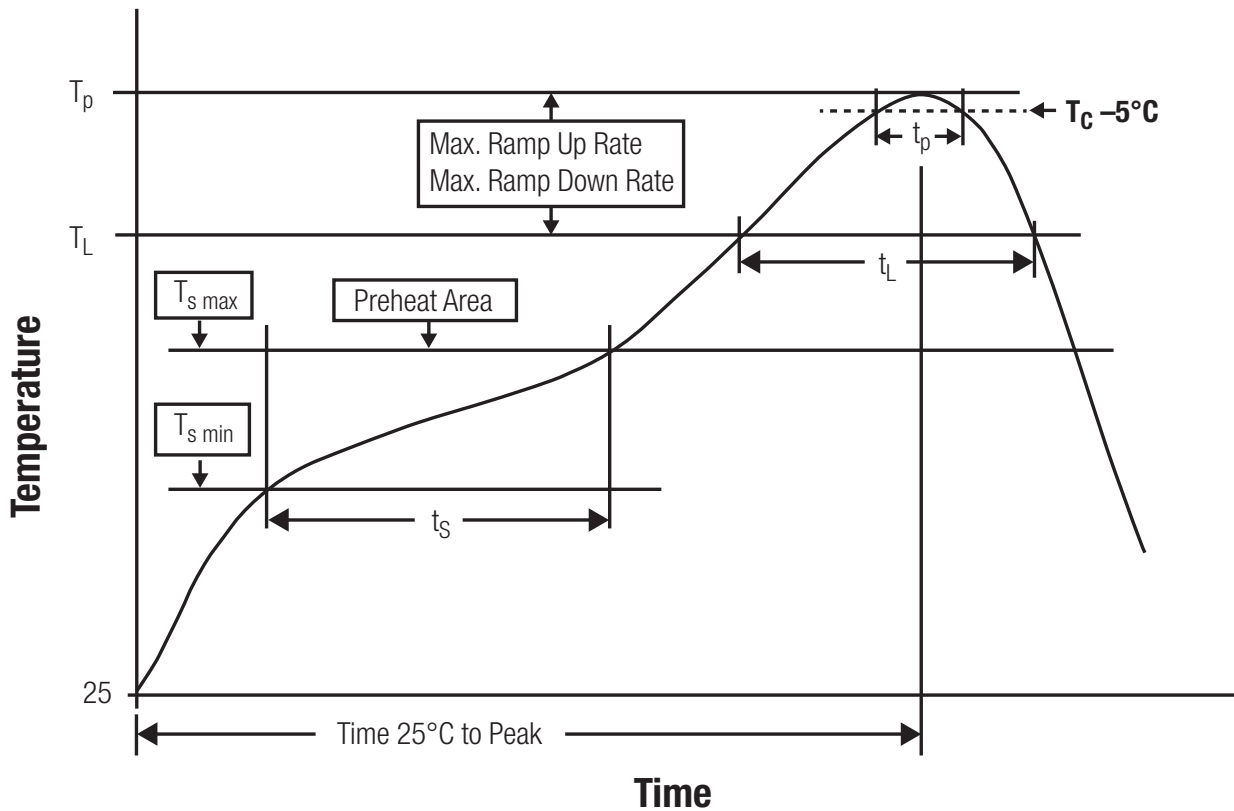


Figure 33: Reflow soldering profile

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

19.2.2. 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.

19.2.3. Potting and coating

- If the product is potted in the customer application, the potting material might shrink or expand during and after hardening. Shrinking could lead to an incomplete seal, allowing contaminants into the component. Expansion could damage components. We recommend a manual inspection after potting to avoid these effects.
- Conformal coating or potting results in loss of warranty.
- The RF shield will not protect the part from low-viscosity coatings and potting. An undefined amount of coating and potting will enter inside the shielding.
- Conformal coating and potting will influence the parts of the radio front end and consequently influence the radio performance.
- Potting will influence the temperature behaviour of the device. This might be critical for components with high power.

19.2.4. Other notations

- 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.

19.3. 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.

19.4. 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.



Würth Elektronik eiSos radio modules with high output power of up to 500 mW generate a large amount of heat while transmitting. The manufacturer of the end device must take care of potentially necessary actions for his application.

20. Product testing

20.1. Würth Elektronik eiSos in-house production tests

To achieve a high quality standard, Würth Elektronik eiSos follows a philosophy of supplying fully tested radio modules. At the end of the production process, every unit undergoes an optical inspection. Here the quality of soldering, edge castellation and edge milling is monitored.

If this has been passed, the radio modules are handed over to the automatic test equipment for the electrical characterization. This includes:

- Voltage and current tests to ensure proper electrical performance
- RF characteristics (frequency, spectrum, TX power) measurement and calibration
- Radio communication tests
- Firmware and serial number programming
- Host interface communication tests

The automated testing process is logged for internal quality control. The gained measurement data of each unit is analysed to detect defective parts and investigate the corresponding root cause. Defective radio modules are discarded, in order to guarantee a 100% failure-free delivery to customers.

20.2. EMS production tests

The rigorous in-series production testing ensures that EMS don't need to duplicate firmware tests or measurements. This streamlines the process and eliminates the need for additional testing over analogue and digital interfaces during device production. When it comes to device testing, the ideal focus should be on module assembly quality:

- All module pins are soldered properly on the base PCB
- There are no short circuits
- The mounting process did not damage the module
- The communication between host and radio module is working
- The antenna is connected properly

Simple "Go/No go" tests, like checking the RSSI value, give already a hint if the power supply and antenna have been connected properly.

In addition to such standard testing procedures, radio module integrators have the flexibility to perform additional dedicated tests to thoroughly evaluate the device. Specific tests they can consider are:

- Measure module current consumption in a specified operating state. Deviations from expected results (compared to a "Golden Device") can signal potential issues.

- Perform functional tests, including communication checks with the host controller and verification of interfaces.
- Assess fundamental RF characteristics (modulation accuracy, power levels, spectrum). Verify that the device meets expected performance standards.

21. Physical specifications

21.1. Dimensions

Dimensions
19 * 27.5 * 3 mm

Table 133: Dimensions

21.2. Weight

Weight
3 g

Table 134: Weight

21.3. Module drawing

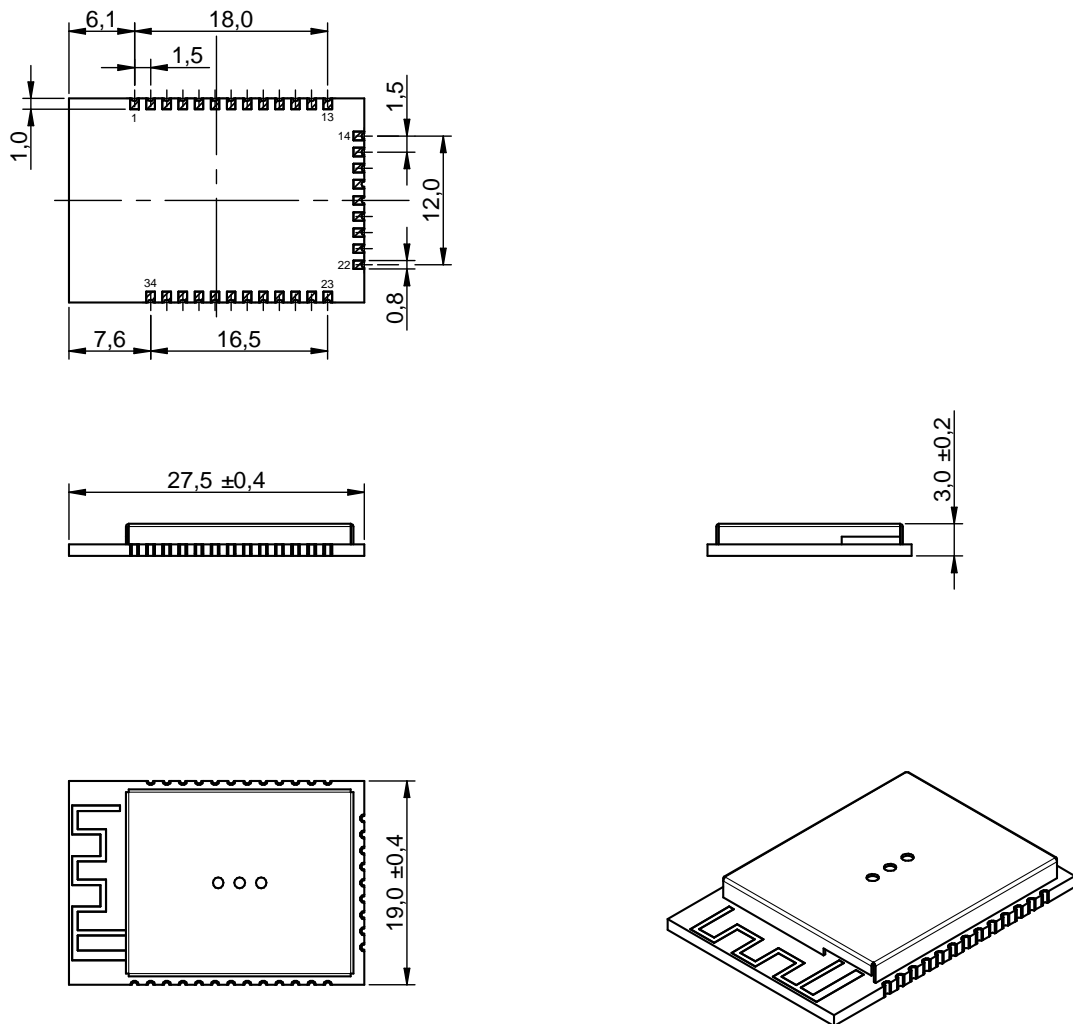


Figure 34: Module dimensions [mm]

21.4. Footprint WE-FP-5

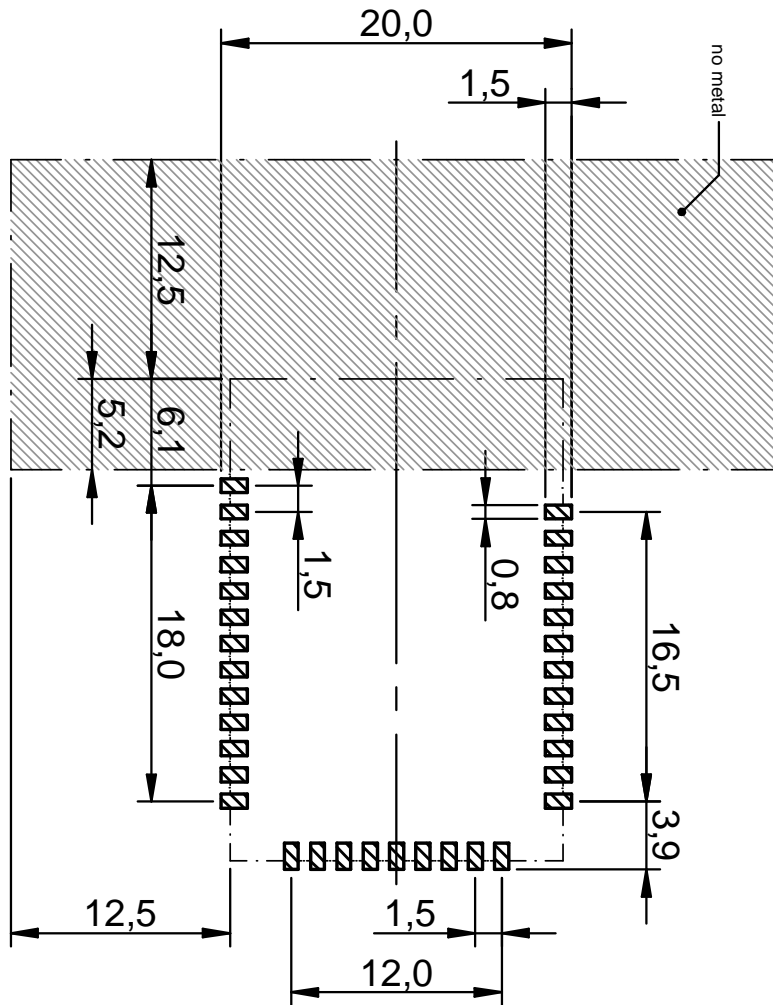


Figure 35: Footprint WE-FP-5 and dimensions [mm]

21.5. 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 $\lambda/10$ should be kept (see figure 35). Even though metal parts would influence the characteristic of the antenna, but the direct influence and matching keep an acceptable level.

22. Marking

22.1. 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.

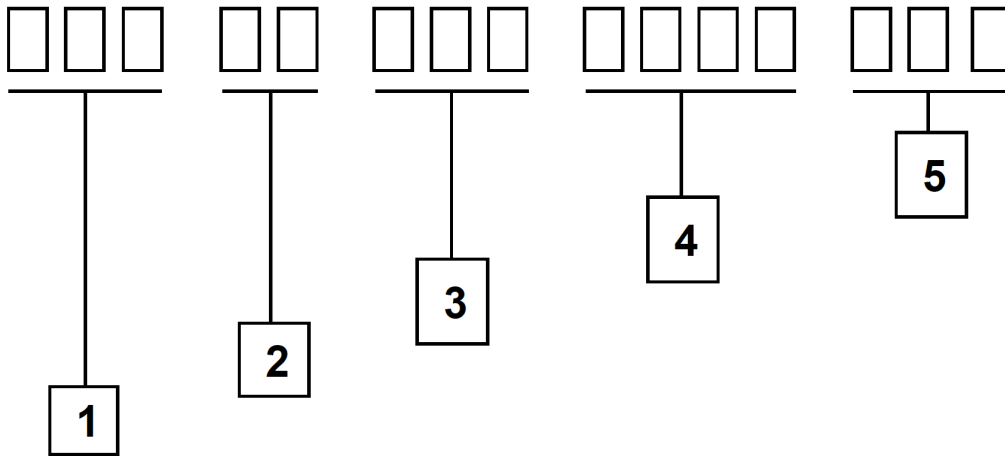


Figure 36: Lot number structure

Block	Information	Example(s)
1	eiSos internal, 3 digits	438
2	eiSos internal, 2 digits	01
3	Radio module 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	Radio module firmware version, 3 digits	V3.2 = 302, V5.13 = 513

Table 135: 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.

22.2. General labeling information

Labels of Würth Elektronik eiSos radio modules include several fields. Besides the manufacturer identification, the product's *WE* order code, serial number and certification information are placed on the label. In case of small labels, additional certification marks are placed on the label of the reel.

The information on the label are fixed. Only the serial number changes with each entity of the radio module. For Calypso the label is as follows:


2610011025000 
1001102
FCC ID: R7T1001102
IC: 5136A-1001102
SN: 129004197

Figure 37: Label of the Calypso

23. Information for explosion protection

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

- The module itself has no internal fuse.
- The maximum output power of the module is 18 dBm.
- The total amount of capacitance of all capacitors is 91.1 μF .
- The total amount of inductance of all inductors is 15.4 μH .

24. Regulatory compliance information

24.1. Important notice EU

The use of RF frequencies is limited by national regulations. The Calypso has been designed to comply with 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 RED directive, restrictions (e.g. in terms of duty cycle or maximum allowed RF power) may apply.

24.2. 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 harmonisation legislation Directive 2014/53/EU and 2011/65/EU with its amending Annex II EU 2015/863 . Following harmonised norms or technical specifications have been applied:

EN 300 328 V2.2.2 (2019-07)
EN 301 489-1 V2.2.3 (2019-11)
EN 301 489-17 V3.2.4 (2020-09)
EN 62479 : 2010
EN 62368-1:2014 + AC:2015 + A11:2017

i.A. G. Exlerdt

Trier, 21th of December 2020

Place and date of issue

24.3. FCC Compliance Statement (US)

FCC ID: R7T1001102

This device complies with Part 15 of the FCC Rules.

Operation is subject to the following two conditions:

- (1) this device may not cause harmful interference, and
 - (2) this device must accept any interference received, including interference that may cause undesired operation.
- (FCC 15.19)

Modifications (FCC 15.21)

Caution: Changes or modifications for this equipment not expressly approved by Würth Elektronik eiSos may void the FCC authorization to operate this equipment.

24.3.1. FCC certificate

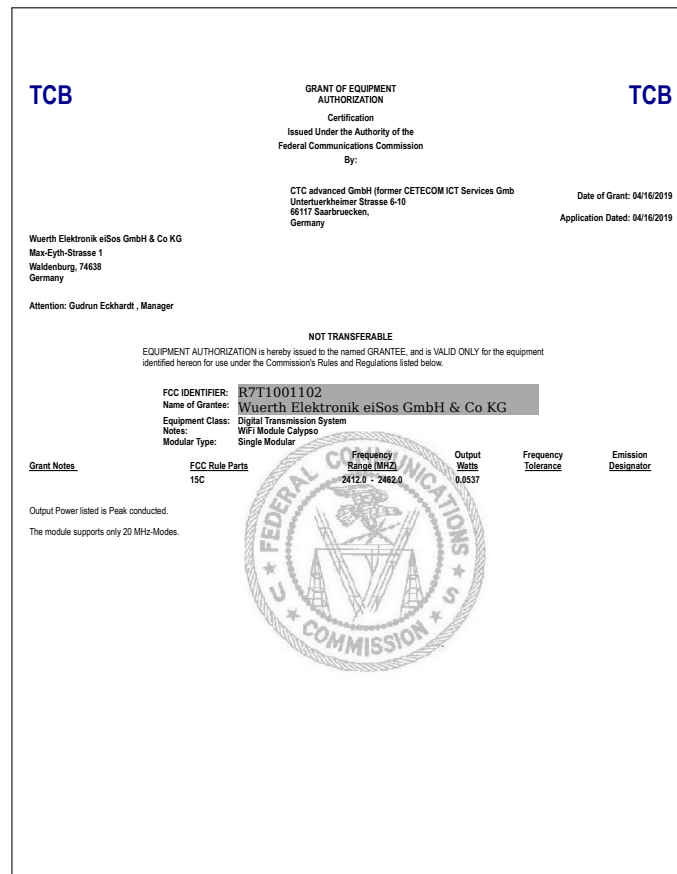


Figure 38: FCC certificate

24.4. IC Compliance Statement (Canada)

Certification Number: 5136A-1001102

HVIN: 1001102

This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes : (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

24.4.1. IC certificate



Figure 39: IC certificate

24.5. FCC and IC requirements to OEM integrators

This module has been granted modular approval. OEM integrators for host products may use the module in their final products without additional FCC/IC (Industry Canada) certification if they meet the following conditions. Otherwise, additional FCC/IC approvals must be obtained. The host product with the module installed must be evaluated for simultaneous transmission requirements.

- The users manual for the host product must clearly indicate the operating requirements and conditions that must be observed to ensure compliance with current FCC/IC RF exposure guidelines.
- To comply with FCC/IC regulations limiting both maximum RF output power and human exposure to RF radiation, the maximum antenna gain including cable loss in a mobile-only exposure condition must not exceed 6Bi.
- A label must be affixed to the outside of the host product with the following statements:
This device contains FCCID: R7T1001102
This equipment contains equipment certified under ICID: 5136A-1001102
- The final host / module combination may also need to be evaluated against the FCC Part 15B criteria for unintentional radiators in order to be properly authorized for operation as a Part 15 digital device.
- The final host / module combination may also need to be evaluated against the FCC Part 15C criteria for intentional radiators according KDB 996369.
- If the final host / module combination is intended for use as a portable device (see classifications below) the host manufacturer is responsible for separate approvals for the SAR requirements from FCC Part 2.1093 and RSS-102.

OEM requirements:

The OEM must ensure that the following conditions are met.

- End users of products, which contain the module, must not have the ability to alter the firmware that governs the operation of the module. The agency grant is valid only when the module is incorporated into a final product by OEM integrators.
- The end-user must not be provided with instructions to remove, adjust or install the module.
- The Original Equipment Manufacturer (OEM) must ensure that FCC labeling requirements are met. This includes a clearly visible label on the outside of the final product. Attaching a label to a removable portion of the final product, such as a battery cover, is not permitted.
- The label must include the following text:
Contains FCC ID: R7T1001102
The enclosed device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:
(i.) this device may not cause harmful interference and

(ii.) *this device must accept any interference received, including interference that may cause undesired operation.*

When the device is so small or for such use that it is not practicable to place the statement above on it, the information required by this paragraph shall be placed in a prominent location in the instruction manual or pamphlet supplied to the user or, alternatively, shall be placed on the container in which the device is marketed. However, the FCC identifier or the unique identifier, as appropriate, must be displayed on the device.

- The user manual for the end product must also contain the text given above.
 - Changes or modifications not expressly approved could void the user's authority to operate the equipment.
 - The OEM must ensure that timing requirements according to 47 CFR 15.231(a-c) are met.
 - The OEM must sign the OEM Modular Approval Agreement.
 - The module must be used with only the following approved antenna(s).

24.6. Pre-certified antennas

The Calypso is pre-certified with the following antennas.

Product	Certified antenna
2610011025000	PCB antenna included in the Calypso
2610011025000	Dipole antenna as specified in chapter 17.3.4.1

It is only possible to connect an antenna by soldering. It is mandatory to follow chapter 18.3 when connecting an antenna. Special care must be taken when using an external antenna to fulfill the requirement of permanently attached antenna or unique coupling for example by using the certified dipole antenna in a closed housing, so that only through professional installation it is possible to remove it.

7.	Applicable Gazette Notification(s)	1. 45 (E) Dated 28-01-2005	
8.	RF Test Report details:-		
	Name&Address /Country of accredited laboratory issuing the RF test report	Accreditation Certificate Reference/Number	Test Report No. and Date
	CTC advanced GmbH & Untertuerkheimer Strasse 6 10 66117 Saarbruecken / Germany	D-PL-12076-01-03	1-9813/20-05-02 & 28-09-2018

II. Terms and Conditions

- (i). This certificate will not be valid in case any change in the above parameters and not conforming to the Gazette Notification mentioned in sl.no 7 above.
- (ii). Use of such equipment has been exempted from licensing requirement vide Gazette Notification mentioned in sl. no. 7, on Non-interference,Non-protectionand sharing (non-exclusive) basis.
- (iii). Use of such equipment in case not conforming to above notification will require a specific wireless operating license, as applicable from this Ministry.
- (iv). Field units of WPC Wing reserve the right for sample check/audit carried out for the purpose of RF analysis/spectrum monitoring in view to avoid interference to other wireless users and ensure compliance of technical parameters mentioned in sl no. 5,6&7.
- (v). This certificate is valid only for equipment which are exempted from import licensing requirements as per the Import Policy of DGFT and for import of such device, a self-declaration based, system generated (Saralsanchar) Import undertaking/ permission is required.
- (vi). The applicant is liable for prosecution under Indian Law in case of any wrong declaration/ submission of ingenuine RF test report(s) for issue of ETA through Self-Declaration.

Note:

- 1. Once ETA through self-declaration is generated for a model, subsequently it may be utilized by other person(s) for import/usage purpose in India.
- 2. The importers of above model shall comply with other import related requirements, if any, with Customs.

This is Self-generated certificate. Hence, no signature is required. It may be downloaded/verified from the website <https://saralsanchar.gov.in>.

Figure 41: ETA-WPC certificate page 2

25. References

- [1] WiFi Alliance. WiFi Specification. <https://www.wi-fi.org/discover-wi-fi/specifications>.
- [2] Würth Elektronik. PC Tool Calypso. <https://www.we-online.com/at-commander>.
- [3] Würth Elektronik. Application note 28 - Calypso transparent mode. <http://www.we-online.com/ANR028>.
- [4] Würth Elektronik. Application note 29 - Calypso remote GPIO. <http://www.we-online.com/ANR029>.

26. Important notes

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

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.

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 source code and firmware parts contained in or used with or for products in the wireless connectivity and sensor 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.

Best care and attention

Any product-specific data sheets, manuals, application notes, PCNs, warnings and cautions must be strictly observed in the most recent versions and matching to the products revisions. These documents can be downloaded from the product specific sections on the wireless connectivity and sensors homepage.

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 Business Development Engineer (BDM) or the internal sales person in charge should be contacted who will be happy to support in this matter.

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 case of further queries regarding the PCN, the Business Development Engineer (BDM), the internal sales person or the technical support team in charge should be contacted. The basic responsibility of the customer as per section 26 and 26 remains unaffected.

All software like "wireless connectivity SDK", "Sensor SDK" or other source codes as well as all PC software tools are not subject to the Product Change Notification information process.

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 Business Development Engineer (BDM) 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.

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.

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.

27. Legal notice

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.

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 information 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.

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.

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.

28. License terms

These License terms will take effect upon the purchase and usage of the Würth Elektronik eiSos GmbH & Co. KG wireless connectivity products. You hereby agree that these license terms are applicable to the product and the incorporated software, firmware and source codes (collectively, "Software") made available by Würth Elektronik eiSos in any form, including but not limited to binary, executable or source code form. The software included in any Würth Elektronik eiSos wireless connectivity product is purchased to you on the condition that you accept the terms and conditions of these license terms. You agree to comply with all provisions under these license terms.

Limited license

Würth Elektronik eiSos hereby grants you a limited, non-exclusive, non-transferable and royalty-free license to use the software and under the conditions that will be set forth in these license terms. You are free to use the provided software only in connection with one of the products from Würth Elektronik eiSos to the extent described in these license terms. You are entitled to change or alter the source code for the sole purpose of creating an application embedding the Würth Elektronik eiSos wireless connectivity product. The transfer of the source code to third parties is allowed to the sole extent that the source code is used by such third parties in connection with our product or another hardware provided by Würth Elektronik eiSos under strict adherence of these license terms. Würth Elektronik eiSos will not assume any liability for the usage of the incorporated software and the source code. You are not entitled to transfer the source code in any form to third parties without prior written consent of Würth Elektronik eiSos.

You are not allowed to reproduce, translate, reverse engineer, decompile, disassemble or create derivative works of the incorporated software and the source code in whole or in part. No more extensive rights to use and exploit the products are granted to you.

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 wireless connectivity 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.**

Ownership

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

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.

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.

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, software and source code. 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.

Applicable law and jurisdiction

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

Severability clause

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

Miscellaneous

Würth Elektronik eiSos reserves the right at any time to change these terms at its own discretion. It is your responsibility to check at Würth Elektronik eiSos homepage for any updates. Your continued usage of the products will be deemed as the acceptance of the change.

We recommend you to be updated about the status of new firmware and software, which is available on our website or in our data sheet and manual, and to implement new software in your device where appropriate.

By ordering a product, you accept these license terms 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_CONNECTING (208)
SL_WLAN_DISCONNECT_MISSING_CERT (209)
SL_WLAN_DISCONNECT_CERTIFICATE_EXPIRED (210)

```

B.2. AT command parse error codes

```

STRMPL_ERROR_PARAM_MISSING (-1)
STRMPL_ERROR_MEM_ALLOCATION (-2)
STRMPL_ERROR_DELIM_MISSING (-3)
STRMPL_ERROR_WRONG_PARAM (-4)
STRMPL_ERROR_WRONG_SIZE (-5)

```

B.3. 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_EOBUFS 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.4. 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_ESECDCLOSED (-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_ESECBCADCAFILE (-456L) /* error secure level bad CA file */
SL_ERROR_BSD_ESECBCADCERTFILE (-457L) /* error secure level bad Certificate file */
SL_ERROR_BSD_ESECBCADPRIVATEFILE (-458L) /* error secure level bad private file */
SL_ERROR_BSD_ESECBCADDHFILE (-459L) /* error secure level bad DH file */
SL_ERROR_BSD_ESECTOOMANYSSLOPENED (-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_O_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.5. 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.6. 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.7. 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.8. 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_THRESHOLD (-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_ROLLBACK_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.9. HTTP Client error codes

```

/*Internal send buffer is not big enough*/
#define HTTPClient_ESENDBUFSMALL (-3001)

/* Buffer inserted into HTTPClient_getOpt() is not big enough.*/
HTTPClient_EGETOPTBUFSMALL (-3002)

/* Response received from the server is not a valid HTTP/1.1 or HTTP/1.0 response*/
HTTPClient_ERESPONSEINVALID (-3003)

/* Operation could not be completed. Try again.*/
HTTPClient_EINPROGRESS (-3004)

/* Input domain name length is too long to be read into buffer.*/
HTTPClient_EDOMAINBUFSMALL (-3005)

/* Allocation failed during the CB creation.*/
HTTPClient_ECBALLOCATIONFAILED (-3006)

/* Body size is too small.*/
HTTPClient_EBODYBUFSMALL (-3008)

/* Invalid de-referencing a NULL pointer.*/
HTTPClient_ENULLPOINTER (-3009)

/* Request header allocation failed.*/

```

```

HTTPClient_EREQUESTHEADERALLOCFALLED          (-3010)

/* Request header wasn't found in the req header list.*/
HTTPClient_EREQHEADERNOTFOUND                 (-3011)

/* Host request header wasn't found.*/
HTTPClient_EHOSTNOTFOUND                      (-3012)

/* Client is already connected.*/
HTTPClient_ECLIENTALREADYCONNECTED           (-3013)

/* Response is not redirectable.*/
HTTPClient_ERESPONSEISNOTREDIRECTABLE         (-3014)

/* Send couldn't be completed.*/
HTTPClient_ESENDERROR                         (-3015)

/* Location Header fields value couldn't be read completely*/
HTTPClient_EREDIRECTLOCATIONFAIL               (-3016)

/* TLS downgrade is forbidden.*/
HTTPClient_ETLSDOWNGRADEISFORBIDDEN          (-3017)

/* Wrong API parameter.*/
HTTPClient_EWRONGAPIPARAMETER                 (-3018)

/* HOST already exist.*/
HTTPClient_EHOSTHEADERALREADYEXIST            (-3019)

/* Client is disconnected.*/
HTTPClient_ENOCONNECTION                       (-3020)

/* URI is not absolute*/
HTTPClient_ENOTABSOLUTEURI                    (-3021)

/* Error during creation of security attribut*/
HTTPClient_ECANTCREATESECATTRIB               (-3022)

/* General internal error*/
HTTPClient_EINTERNAL                          (-3023)

/* Buffer inserted into getHeaderByName(..) is not big enough.*/
HTTPClient_EGETCUSOMHEADERBUFSMALL           (-3024)

/* Custom response header name on getHeaderByName(..) dosn't set before.*/
HTTPClient_ENOHEADERNAMEDASINSERTED          (-3025)

```

B.10. Other error codes

```

SL_POOL_IS_EMPTY (-2000L)
SL_ESMALLBUF (-2001L)
SL_EZEROLEN (-2002L)
SL_INVALPARAM (-2003L)
SL_BAD_INTERFACE (-2004L)
SL_API_ABORTED (-2005)
SL_RET_CODE_INVALID_INPUT (-2006L)
SL_RET_CODE_SELF_ERROR (-2007L)
SL_RET_CODE_NWP_IF_ERROR (-2008L)

```

```
SL_RET_CODE_MALLOC_ERROR (-2009L)
SL_RET_CODE_PROTOCOL_ERROR (-2010L)
SL_RET_CODE_DEV_LOCKED (-2011L)
SL_RET_CODE_DEV_ALREADY_STARTED (-2012L)
SL_RET_CODE_API_COMMAND_IN_PROGRESS (-2013L)
SL_RET_CODE_PROVISIONING_IN_PROGRESS (-2014L)
SL_RET_CODE_NET_APP_PING_INVALID_PARAMS (-2015L)
SL_RET_CODE_SOCKET_SELECT_IN_PROGRESS_ERROR (-2016L)
SL_RET_CODE_STOP_IN_PROGRESS (-2017L)
SL_RET_CODE_DEV_NOT_STARTED (-2018L)
SL_RET_CODE_EVENT_LINK_NOT_FOUND (-2019L)

/* 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
Amazon Root CA 1
Amazon Root CA 2
Amazon Root CA 3
Amazon Root CA 4
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 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

D. TCP flow diagram

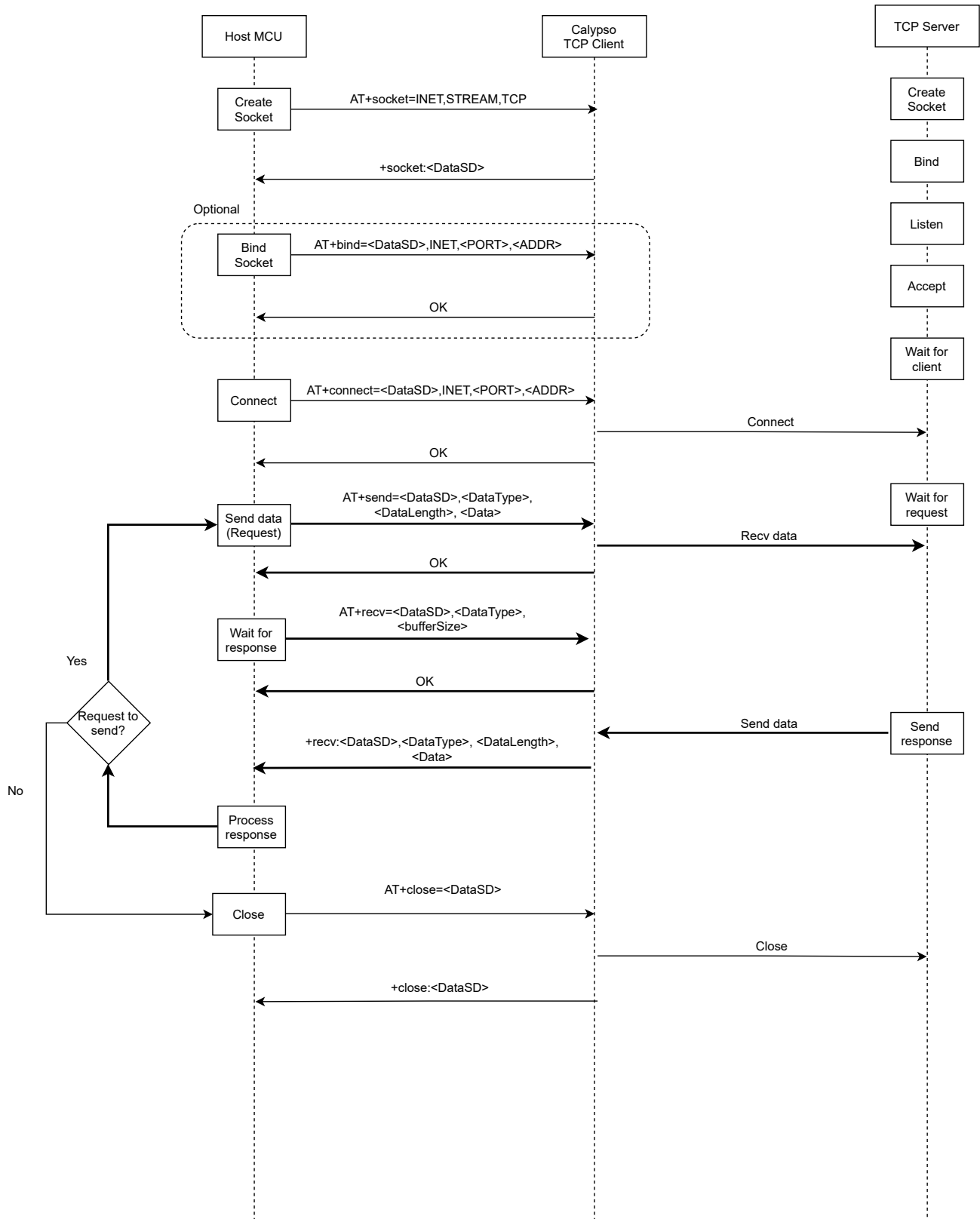


Figure 42: Calypso TCP client work flow

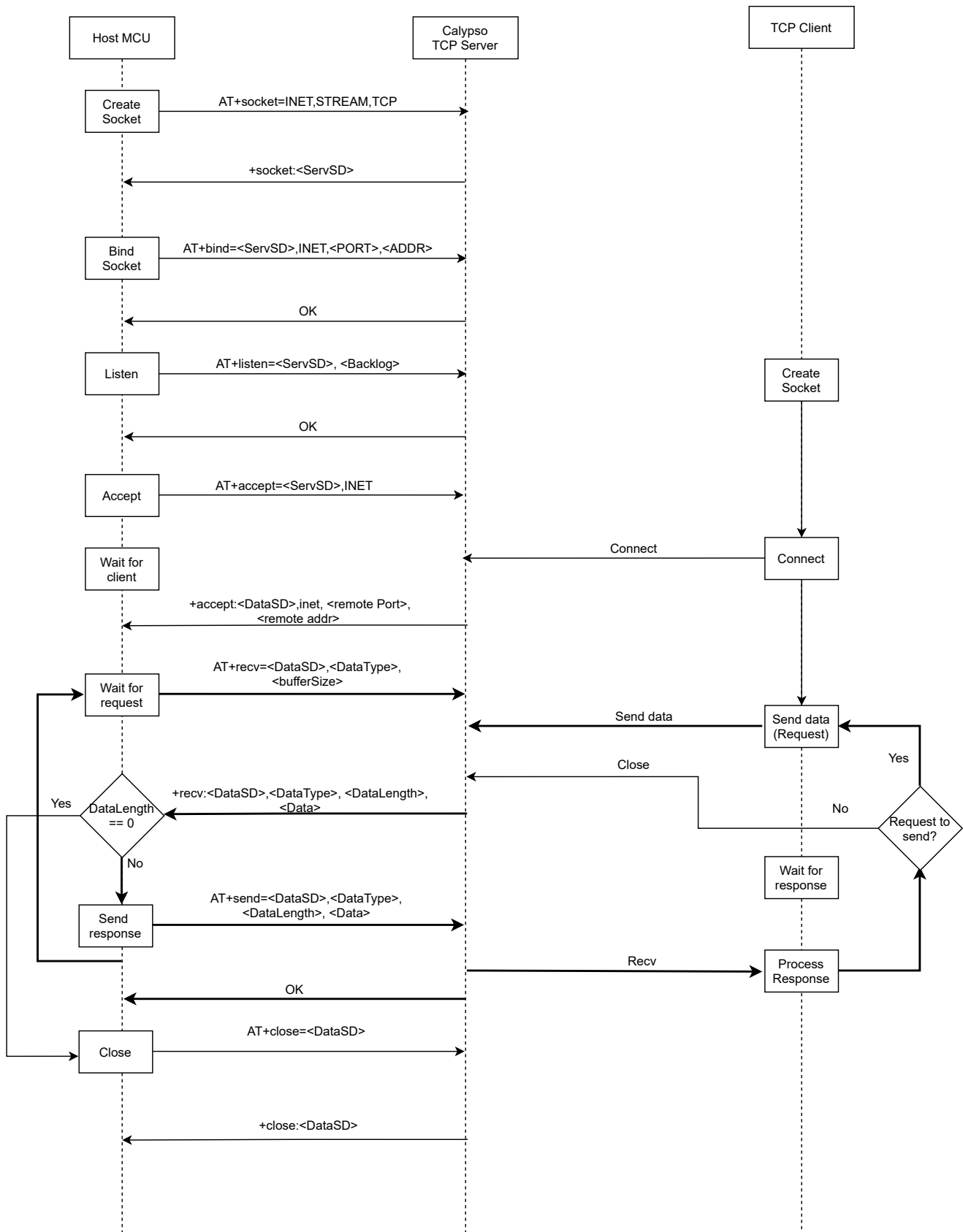


Figure 43: Calypso TCP server work flow

List of Figures

1.	Block diagram	16
2.	Pinout (top view)	23
3.	Power up	27
4.	Quick start setup	29
5.	Modes of operation	36
6.	Host interface	40
7.	UART timing	41
8.	TCP socket workflow	62
9.	UDP socket work flow	63
10.	SSL/TLS handshake	64
11.	HTTP server	99
12.	Custom GET webpage	117
13.	Custom POST webpage	118
14.	Provisioning main page	120
15.	Provisioning main page	121
16.	File upload	122
17.	Test page	131
18.	Test page	132
19.	OTA webpage	137
20.	OTA webpage upload	138
21.	OTA in progress	138
22.	Finalize OTA	139
23.	Layout	148
24.	Placement of the module with integrated antenna	149
25.	Dimensioning the antenna connection as micro strip	150
26.	Himalia dipole antenna	152
27.	Reference design: Schematic, most important parts	154
28.	Reference design: Layout	155
29.	Antenna characteristic of the module with its integrated antenna measured on the official EV-Board	156
30.	Close-up: Layout	157
31.	Reference design: Stack-up	157
32.	Close-up: Schematic	158
33.	Reflow soldering profile	161
34.	Module dimensions [mm]	167
35.	Footprint WE-FP-5 and dimensions [mm]	168
36.	Lot number structure	170
37.	Label of the Calypso	171
38.	FCC certificate	175
39.	IC certificate	176
40.	ETA-WPC certificate page 1	179
41.	ETA-WPC certificate page 2	180
42.	Calypso TCP client work flow	202
43.	Calypso TCP server work flow	203

List of Tables

1.	Ordering information	16
2.	Recommended operating conditions	17
3.	Absolute maximum ratings	17
4.	Power consumption	18
5.	Radio characteristics	18
6.	Modulation schemes and peak data rate.	19
7.	Pin characteristics, VDD5 = 3.3 V, T = 25 °C	20
8.	TX power vs current consumption, conducted measurement of continuous data transmission, rate 1Mbps (DSSS)	21
9.	TX power vs current consumption, conducted measurement of continuous data transmission, rate 54 Mbps (OFDM)	22
10.	Pinout	25
11.	Minimal pin configuration	26
12.	Country codes	28
13.	Quick start addresses and roles	30
14.	Key features (Part 1)	34
15.	Key features (Part 2)	34
16.	Application modes	37
17.	UART parameters	40
18.	AT+start	44
19.	AT+stop	44
20.	AT+test	45
21.	AT+reboot	45
22.	AT+factoryreset	46
23.	AT+sleep	47
24.	AT+powersave	47
25.	AT+get	48
26.	AT+set	49
27.	AT+wlanSetMode	50
28.	AT+wlanScan	51
29.	AT+wlanConnect	51
30.	WLAN security types	52
31.	AT+wlanDisconnect	52
32.	AT+wlanProfileAdd	53
33.	AT+wlanProfileGet	54
34.	AT+wlanProfileDel	54
35.	AT+wlanSet	55
36.	AT+wlanGet	56
37.	AT+wlanPolicySet	57
38.	AT+wlanPolicyGet	58
39.	IP addresses	59
40.	AT+netCfgSet	60
41.	AT+netCfgGet	61
42.	AT+socket (create a socket)	65
43.	AT+close (close a socket)	65
44.	AT+bind	65

45.	AT+listen	65
46.	AT+connect	66
47.	AT+accept	66
48.	AT+select	66
49.	AT+setSockOpt	67
50.	AT+setSockOpt (Part 2)	68
51.	Supported cipher methods	69
52.	AT+getSockOpt	69
53.	AT+recv	70
54.	AT+recvFrom	70
55.	AT+send	71
56.	AT+sendTo	71
57.	AT+fileGetFileList	73
58.	AT+fileOpen	74
59.	AT+fileClose	74
60.	AT+fileDel	75
61.	AT+fileGetInfo	75
62.	AT+fileRead	75
63.	AT+fileWrite	76
64.	AT+netAppStart	77
65.	AT+netAppStop	77
66.	AT+netAppGetHostByName	77
67.	AT+netAppGet	78
68.	AT+netAPPSet(1)	79
69.	AT+netAPPSet(2)	80
70.	SNTP get	81
71.	SNTP set	81
72.	AT+netAPPUpdateTime	82
73.	AT+httpCreate	82
74.	AT+httpDestroy	82
75.	AT+httpConnect	82
76.	AT+httpDisconnect	83
77.	AT+httpSetProxy	83
78.	AT+httpSendReq	83
79.	AT+httpReadResBody	84
80.	AT+httpSetHeader	84
81.	AT+httpGetHeader	84
82.	HTTP header options	85
83.	AT+mqttCreate	86
84.	AT+mqttDelete	86
85.	AT+mqttConnect	87
86.	AT+mqttDisconnect	87
87.	AT+mqttPublish	87
88.	AT+mqttSubscribe	87
89.	AT+mqttUnsubscribe	88
90.	AT+mqttSet	88
91.	AT+netAPPping	89
92.	AT+gpioGet	89

93.	AT+gpioSet	90
94.	AT+calypso	90
95.	+eventstartup event	91
96.	+eventgeneral event	92
97.	+eventwlan event	93
98.	+eventsock event	94
99.	+eventnetapp event	95
100.	+eventmqtt event	96
101.	+eventfatalerror event	96
102.	+eventcustom event	97
103.	API in application modes	100
104.	System information tokens (Part 1)	101
105.	System information tokens (Part 2)	101
106.	Version information tokens	102
107.	Network information tokens 1 (Station or P2P client)	102
108.	Network information tokens 2 (Station or P2P client)	103
109.	Network information tokens (DHCP server)	103
110.	Network information tokens (AP or P2P GO)	104
111.	Ping result tokens	105
112.	Connection policy tokens	105
113.	WiFi profile information tokens	105
114.	P2P tokens	106
115.	WiFi profile POST	107
116.	WiFi EAP profile POST	108
117.	WiFi Scan	108
118.	WiFi connection policy	109
119.	Network configuration POST parameters	110
120.	Ping POST parameters	111
121.	User setting GET	112
122.	User setting POST part 1	113
123.	User setting POST part 2	114
124.	GPIO GET parameters	115
125.	GPIO POST parameters	115
126.	GPIO types with corresponding value1 and value2 parameters	116
127.	AT+httpcustomresponse	117
128.	Start-up time	134
129.	Start-up after reboot	134
130.	Key features v2.0.0	142
132.	Classification reflow soldering profile, Note: refer to IPC/JEDEC J-STD-020E	160
133.	Dimensions	166
134.	Weight	166
135.	Lot number details	170



Contact

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

Max-Eyth-Straße 1
74638 Waldenburg
Germany

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

WÜRTH ELEKTRONIK MORE THAN YOU EXPECT