## Revision history

<table>
<thead>
<tr>
<th>Manual version</th>
<th>FW version</th>
<th>HW version</th>
<th>Notes</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.0.0</td>
<td>2.1</td>
<td>• Initial release of the manual</td>
<td>January 2019</td>
</tr>
</tbody>
</table>
| 1.1            | 1.0.0      | 2.1        | • Added chapter Reference design  
                  • Added chapter Information for Ex protection | February 2019 |
| 1.2            | 1.0.0      | 2.1        | • Added known issues in chapter Firmware history | March 2019 |
| 1.3            | 1.2.0      | 2.1        | • Update for the new firmware version.  
                  • Updated chapter Firmware history | April 2019 |
| 1.4            | 1.2.0      | 2.2        | • FCC and IC certification achieved. Compliance statements and requirements added, chapter FCC Compliance Statement and following. | April 2019 |
| 1.5            | 1.2.0      | 2.2        | • Improved description of the pinout tables in chapter Pinout | June 2019 |
| 1.6            | 1.2.0      | 2.2        | • Updated known issues in chapter Firmware history | July 2019 |
| 1.7            | 1.3.0      | 2.2        | • Update for the new firmware version.  
                  • Added a section on quick help in chapter The command interface  
                  • Updated chapter Firmware history | July 2019 |
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWP</td>
<td>Network processor unit</td>
<td>802.11 network processor unit</td>
</tr>
<tr>
<td>AP</td>
<td>Access point</td>
<td>WLAN (IEEE 802.11) infrastructure node offering stations to connect to</td>
</tr>
<tr>
<td>DC</td>
<td>Duty cycle</td>
<td>Transmission time in relation of one hour. 1% means, channel is occupied for 36 seconds per hour.</td>
</tr>
<tr>
<td>FSE</td>
<td>Field Sales Engineer</td>
<td>Support and sales contact person responsible for limited sales area</td>
</tr>
<tr>
<td>0xhh [HEX]</td>
<td>Hexadecimal</td>
<td>All numbers beginning with 0x are stated as hexadecimal numbers. All other numbers are decimal.</td>
</tr>
<tr>
<td>HIGH</td>
<td>High signal level</td>
<td></td>
</tr>
<tr>
<td>LOW</td>
<td>Low signal level</td>
<td></td>
</tr>
<tr>
<td>LSB</td>
<td>Least significant bit</td>
<td></td>
</tr>
<tr>
<td>MSB</td>
<td>Most significant bit</td>
<td></td>
</tr>
<tr>
<td>PL</td>
<td>Payload</td>
<td>The real, non-redundant information in a frame/packet.</td>
</tr>
<tr>
<td>RF</td>
<td>Radio frequency</td>
<td>Describes everything relating to the wireless transmission.</td>
</tr>
<tr>
<td>STA</td>
<td>Station</td>
<td>WLAN (802.11) node in station role, can connect to an AP</td>
</tr>
<tr>
<td>UART</td>
<td></td>
<td>Universal Asynchronous Receiver Transmitter allows communicating with the module of a specific interface.</td>
</tr>
<tr>
<td>US</td>
<td>UserSettings</td>
<td>Any relation to a specific entry in the UserSettings is marked in a special font and can be found in the respective chapter.</td>
</tr>
<tr>
<td>VDD</td>
<td>Supply voltage</td>
<td></td>
</tr>
<tr>
<td>Wi-Fi</td>
<td></td>
<td>Is a Registered Trademark of the Wi-Fi Alliance for interoperability tested WLAN (IEEE 802.11) based products.</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
<td></td>
</tr>
<tr>
<td>P2P</td>
<td>Peer to Peer</td>
<td>WLAN configuration</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium access control</td>
<td></td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
<td>Network layer protocol</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
<td>Transport layer protocol</td>
</tr>
<tr>
<td>UDP</td>
<td>Use Datagram Protocol</td>
<td>Transport layer protocol</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
<td>Transport layer protocol</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
<td>Transport layer protocol</td>
</tr>
<tr>
<td>HTTP(s)</td>
<td>Hypertext transfer protocol (secure)</td>
<td>Application layer protocol</td>
</tr>
<tr>
<td>MQTT</td>
<td>Message Queuing Telemetry Transport</td>
<td>Application layer protocol</td>
</tr>
<tr>
<td>OTA</td>
<td>Over The Air</td>
<td>Update mechanism</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
<td>Application layer protocol</td>
</tr>
<tr>
<td>WEP</td>
<td>Wired Equivalent Privacy</td>
<td>802.11 security algorithm</td>
</tr>
<tr>
<td>WPA</td>
<td>Wi-Fi Protected Access</td>
<td>Wi-Fi security algorithm</td>
</tr>
<tr>
<td>WPS</td>
<td>Wi-Fi Protected Setup</td>
<td>Wi-Fi security algorithm</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
<td>Application layer protocol</td>
</tr>
<tr>
<td>mDNS</td>
<td>multicast-DNS</td>
<td>Application layer protocol</td>
</tr>
<tr>
<td>LLA</td>
<td>Link-local addressing</td>
<td>IPv4/IPv6 local addressing mechanism</td>
</tr>
<tr>
<td>DAD</td>
<td>Duplicate address detection</td>
<td>IPv4/IPv6 addressing mechanism</td>
</tr>
</tbody>
</table>
## 7. The command interface

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1. Command types</td>
<td>33</td>
</tr>
<tr>
<td>7.2. AT command characteristics</td>
<td>33</td>
</tr>
<tr>
<td>7.2.1. Request</td>
<td>33</td>
</tr>
<tr>
<td>7.2.2. Confirmations</td>
<td>34</td>
</tr>
<tr>
<td>7.2.3. Events</td>
<td>34</td>
</tr>
<tr>
<td>7.2.4. Help</td>
<td>35</td>
</tr>
</tbody>
</table>

## 8. AT commands

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1. Device commands</td>
<td>36</td>
</tr>
<tr>
<td>8.1.1. Start and stop commands</td>
<td>36</td>
</tr>
<tr>
<td>8.1.2. Test</td>
<td>36</td>
</tr>
<tr>
<td>8.1.3. Reboot</td>
<td>37</td>
</tr>
<tr>
<td>8.1.4. Factory reset</td>
<td>37</td>
</tr>
<tr>
<td>8.1.5. Sleep</td>
<td>38</td>
</tr>
<tr>
<td>8.1.6. Get</td>
<td>38</td>
</tr>
<tr>
<td>8.1.7. Set</td>
<td>39</td>
</tr>
<tr>
<td>8.2. WLAN commands</td>
<td>40</td>
</tr>
<tr>
<td>8.2.1. Set mode</td>
<td>40</td>
</tr>
<tr>
<td>8.2.2. Scan</td>
<td>41</td>
</tr>
<tr>
<td>8.2.3. Manual connection</td>
<td>41</td>
</tr>
<tr>
<td>8.2.4. Profiles</td>
<td>42</td>
</tr>
<tr>
<td>8.2.5. WLAN settings</td>
<td>44</td>
</tr>
<tr>
<td>8.2.6. WLAN policy</td>
<td>45</td>
</tr>
<tr>
<td>8.3. Network configuration commands</td>
<td>46</td>
</tr>
<tr>
<td>8.4. Socket commands</td>
<td>50</td>
</tr>
<tr>
<td>8.4.1. Sockets work flow</td>
<td>50</td>
</tr>
<tr>
<td>8.4.1.1. TCP socket</td>
<td>50</td>
</tr>
<tr>
<td>8.4.1.2. UDP socket</td>
<td>51</td>
</tr>
<tr>
<td>8.4.1.3. Multicast</td>
<td>51</td>
</tr>
<tr>
<td>8.4.2. Secure sockets</td>
<td>52</td>
</tr>
<tr>
<td>8.4.3. Socket operations</td>
<td>52</td>
</tr>
<tr>
<td>8.4.4. Socket settings</td>
<td>54</td>
</tr>
<tr>
<td>8.4.5. Socket data exchange</td>
<td>57</td>
</tr>
<tr>
<td>8.5. File system commands</td>
<td>59</td>
</tr>
<tr>
<td>8.5.1. File system operations</td>
<td>60</td>
</tr>
<tr>
<td>8.5.2. File operations</td>
<td>61</td>
</tr>
<tr>
<td>8.6. Network application commands</td>
<td>64</td>
</tr>
<tr>
<td>8.6.1. mDNS</td>
<td>64</td>
</tr>
<tr>
<td>8.6.2. SNTP client</td>
<td>65</td>
</tr>
<tr>
<td>8.6.3. HTTP client</td>
<td>66</td>
</tr>
<tr>
<td>8.6.4. MQTT client</td>
<td>70</td>
</tr>
<tr>
<td>8.6.5. Ping</td>
<td>74</td>
</tr>
<tr>
<td>8.7. Events</td>
<td>74</td>
</tr>
<tr>
<td>8.7.1. General events</td>
<td>74</td>
</tr>
<tr>
<td>8.7.2. WLAN events</td>
<td>75</td>
</tr>
<tr>
<td>8.7.3. Socket events</td>
<td>78</td>
</tr>
<tr>
<td>8.7.4. NetApp events</td>
<td>78</td>
</tr>
<tr>
<td>8.7.5. MQTT events</td>
<td>79</td>
</tr>
</tbody>
</table>
9. Provisioning
9.1. Start the provisioning mode ........................................ 81
9.2. Enter the credentials ................................................... 81

10. Typical application use cases ........................................ 84
   10.1. UDP communication ............................................... 84
   10.1.1. Prerequisites .................................................... 84
   10.1.2. UDP socket communication ................................... 84
   10.2. TCP communication ............................................... 85
   10.3. Secure socket communication ................................... 85
   10.3.1. Create an SSL/TLS server ..................................... 85
   10.3.2. Create an SSL/TLS client ..................................... 86
   10.3.3. Secure data transfer ........................................... 86
   10.4. Wi-Fi direct example .............................................. 87
   10.4.1. Prerequisites .................................................... 87
   10.4.2. Auto connection setup ....................................... 87
   10.4.3. Manual connection setup ..................................... 88
   10.5. Running a web page on the radio module ....................... 89
   10.5.1. Load the web page files to the radio module ............... 90
   10.5.2. Accessing the web site in station mode .................... 90
   10.5.3. Accessing the web site in access point mode ............. 91

11. Timing parameters ..................................................... 93
   11.1. Hard reset .......................................................... 93
   11.2. Soft reset .......................................................... 93

12. Firmware update ........................................................ 94
   12.1. Prerequisites ....................................................... 94
   12.2. Update procedure ................................................ 94
   12.2.1. Start-up ......................................................... 94
   12.2.2. Connection to the update device ............................ 95
   12.2.3. Upload the update-package .................................. 96
   12.2.4. Finalize the update ........................................... 98

13. Firmware history ....................................................... 99
   13.1. Release notes ...................................................... 99
   13.2. Known issues ....................................................... 100

14. Custom firmware ........................................................ 101
   14.1. Custom configuration of standard firmware .................... 101
   14.2. Customer specific firmware ..................................... 101
   14.3. Customer firmware ............................................... 101
   14.4. Contact for firmware requests ................................ 102

15. Design in guide ........................................................ 103
   15.1. Advice for schematic and layout ............................... 103
   15.2. Dimensioning of the micro strip antenna line ............... 105
15.3. Antenna solutions ................................................................. 106
  15.3.1. Wire antenna ............................................................... 106
  15.3.2. Chip antenna ............................................................. 107
  15.3.3. PCB antenna ............................................................. 107
  15.3.4. Antennas provided by Würth Elektronik eiSos .................. 108
    15.3.4.1. 2600130011 - Helike - 169 MHz dipole antenna .......... 108
    15.3.4.2. 2600130041 - Herse - 434 MHz dipole antenna .......... 109
    15.3.4.3. 2600130081 - Hyperion-I - 868 MHz dipole antenna .... 110
    15.3.4.4. 2600130082 - Hyperion-II - 868 MHz magnetic base antenna . 111
    15.3.4.5. 2600130021 - Himalia - 2.4 GHz dipole antenna ........ 112

16. Reference design .............................................................. 113
  16.1. EV-Board ....................................................................... 114
  16.2. Design Guide for FCC ID R7T1001102 ............................... 116
  16.3. Application mode pins ................................................... 118

17. Manufacturing information .................................................. 119
  17.1. Moisture sensitivity level .............................................. 119
  17.2. Soldering ....................................................................... 119
    17.2.1. Reflow soldering ...................................................... 119
    17.2.2. Cleaning .................................................................. 121
    17.2.3. Other notations ....................................................... 121
  17.3. ESD handling ................................................................. 121
  17.4. Safety recommendations ................................................ 122

18. Physical dimensions ............................................................. 123
  18.1. Dimensions .................................................................... 123
  18.2. Weight ......................................................................... 123
  18.3. Module drawing ............................................................. 124
  18.4. Footprint ...................................................................... 125
  18.5. Antenna free area .......................................................... 126

19. Marking .............................................................................. 127
  19.1. Lot number .................................................................... 127
  19.2. General labeling information ......................................... 128
    19.2.1. Label of Calypso ...................................................... 128

20. Information for Ex protection ................................................ 129

21. Regulatory compliance information ..................................... 130
  21.1. Important notice EU ....................................................... 130
  21.2. EU Declaration of conformity ......................................... 131
  21.3. FCC Compliance Statement ........................................... 132
  21.4. IC Compliance Statement ............................................... 132
  21.5. FCC and IC requirements to OEM integrators .................. 132
  21.6. Pre-certified antennas .................................................... 134

22. Important notes .................................................................. 135
  22.1. General customer responsibility ....................................... 135
22.2. Customer responsibility related to specific, in particular safety-relevant applications ........................................ 135
22.3. Best care and attention .................................................................................................................. 135
22.4. Customer support for product specifications ................................................................................ 135
22.5. Product improvements .................................................................................................................. 136
22.6. Product life cycle ......................................................................................................................... 136
22.7. Property rights ............................................................................................................................. 136
22.8. General terms and conditions ..................................................................................................... 136

23. Legal notice ........................................................................................................................................ 137
23.1. Exclusion of liability .................................................................................................................... 137
23.2. Suitability in customer applications ........................................................................................... 137
23.3. Trademarks .................................................................................................................................... 137
23.4. Usage restriction .......................................................................................................................... 137

24. License terms ..................................................................................................................................... 139
24.1. Limited license .............................................................................................................................. 139
24.2. Usage and obligations .................................................................................................................. 139
24.3. Ownership ...................................................................................................................................... 140
24.4. Firmware update(s) ..................................................................................................................... 140
24.5. Disclaimer of warranty .................................................................................................................. 140
24.6. Limitation of liability .................................................................................................................... 141
24.7. Applicable law and jurisdiction ................................................................................................... 141
24.8. Severability clause ....................................................................................................................... 141
24.9. Miscellaneous .............................................................................................................................. 141

A. Wi-Fi certificate .................................................................................................................................. 142

B. Error codes ....................................................................................................................................... 144
B.1. Disconnection reason codes .......................................................................................................... 144
B.2. Socket error codes .......................................................................................................................... 144
B.3. Secure socket error codes ............................................................................................................. 145
B.4. WLAN error codes ......................................................................................................................... 148
B.5. Device error codes .......................................................................................................................... 149
B.6. Network config error codes .......................................................................................................... 150
B.7. File System error codes .................................................................................................................. 150
B.8. Other error codes ........................................................................................................................... 152

C. Root certificate catalog .................................................................................................................... 154
1. Introduction

The Calypso WLAN module is a compact WLAN radio module based on IEEE 802.11 b/g/n with a fully featured TCP/IP stack. The edge castellated connections, smart antenna configuration and an easy-to-use AT-style command interface enables easy integration of Calypso into any embedded application.

The module supports IPv4 as well as IPv6 and implements several commonly used network applications like SNTP, DHPv4, DHCPv6, mDNS, HTTP(S), MQTT out-of the box. Advanced security features like up to 6 simultaneous secure sockets, secure boot, secure storage and secure OTA update provide a good basis for a secure end product.

Whether a serial cable replacement or low power IoT application with cloud connectivity, the Calypso WLAN modules offers a robust and standard compliant wireless connectivity solution for low-power and low-medium throughput applications.

WLAN will be used as a synonym for IEEE 802.11 standard compliant radio communication throughout this manual.

Calypso is Wi-Fi Certified. The Certification ID is WFA81685.

1.1. Operational description

The Calypso WLAN module is intended to be used as a radio sub-system in order to provide WLAN communication capabilities to the system.

The UART acts as the primary interface between the module and a host micro-controller.

The module can be fully configured and operated using a set of AT-commands over UART. Once configured, the module independently manages WLAN connectivity allowing the host controller to utilize its resources elsewhere.

Therefore, when using the standard firmware, a host MCU is required in the end product to control and access the radio module. Stand alone applications, without host, can be realized with a custom firmware development.
1.2. Block diagram

![Block diagram](image)

Figure 1: Block diagram

1.3. Ordering information

<table>
<thead>
<tr>
<th>WE order code</th>
<th>Former order code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2610011025000</td>
<td>AMB5201-TR</td>
<td>WLAN module in T&amp;R packaging</td>
</tr>
<tr>
<td>2610011025009</td>
<td>AMB5201-DEV</td>
<td>3 pcs WLAN module</td>
</tr>
<tr>
<td>2610019225001</td>
<td>AMB5201-EV</td>
<td>EV kit for WLAN module</td>
</tr>
</tbody>
</table>

Table 1: Ordering information
2. Electrical specifications

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

2.1. Recommended operating conditions

<table>
<thead>
<tr>
<th>Description</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCC</td>
<td>2.1</td>
<td>3.3</td>
<td>3.6</td>
<td>V</td>
</tr>
<tr>
<td>Temperature range</td>
<td>-40</td>
<td>25</td>
<td>85</td>
<td>°C</td>
</tr>
<tr>
<td>Ambient thermal slew rate</td>
<td>-20</td>
<td>20</td>
<td></td>
<td>°C / min</td>
</tr>
</tbody>
</table>

Table 2: Recommended operating conditions

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

2.2. Absolute maximum ratings

<table>
<thead>
<tr>
<th>Description</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCC</td>
<td>-0.5</td>
<td></td>
<td>3.8</td>
<td>V</td>
</tr>
</tbody>
</table>

Table 3: Absolute maximum ratings

2.3. Power consumption

2.3.1. Static
Table 4: Power consumption

2.4. Radio characteristics

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

Table 6: Modulation schemes and peak data rate.
2.5. Pin characteristics

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

Table 7: Pin characteristics

2.6. TX power vs current consumption

The following tables contain the typical TX power values and the corresponding typical average current for 3.6V supply voltage and 25°C ambient temperature. Cable loses of the conducted measurement are about 2dB.
<table>
<thead>
<tr>
<th>Tx power index</th>
<th>TX power [dBm]</th>
<th>Average current [mA]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13.97</td>
<td>260.15</td>
</tr>
<tr>
<td>1</td>
<td>12.59</td>
<td>255.95</td>
</tr>
<tr>
<td>2</td>
<td>11.62</td>
<td>249.5</td>
</tr>
<tr>
<td>3</td>
<td>11.53</td>
<td>251.17</td>
</tr>
<tr>
<td>4</td>
<td>10.57</td>
<td>189.35</td>
</tr>
<tr>
<td>5</td>
<td>9.47</td>
<td>184.4</td>
</tr>
<tr>
<td>6</td>
<td>8.93</td>
<td>182.3</td>
</tr>
<tr>
<td>7</td>
<td>8.96</td>
<td>182.3</td>
</tr>
<tr>
<td>8</td>
<td>8.89</td>
<td>182.27</td>
</tr>
<tr>
<td>9</td>
<td>8.88</td>
<td>182.22</td>
</tr>
<tr>
<td>10</td>
<td>8.81</td>
<td>182.29</td>
</tr>
<tr>
<td>11</td>
<td>8.86</td>
<td>182.2</td>
</tr>
<tr>
<td>12</td>
<td>8.88</td>
<td>182.17</td>
</tr>
<tr>
<td>13</td>
<td>8.89</td>
<td>182.18</td>
</tr>
<tr>
<td>14</td>
<td>8.92</td>
<td>182.2</td>
</tr>
<tr>
<td>15</td>
<td>8.93</td>
<td>182.11</td>
</tr>
</tbody>
</table>

Table 8: TX power vs current consumption, conducted measurement of continuous data transmission, rate 1Mbps (DSSS)
<table>
<thead>
<tr>
<th>Tx power index</th>
<th>TX power [dBm]</th>
<th>Average current [mA]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11.74</td>
<td>119.74</td>
</tr>
<tr>
<td>1</td>
<td>10.48</td>
<td>118.95</td>
</tr>
<tr>
<td>2</td>
<td>9.46</td>
<td>118.36</td>
</tr>
<tr>
<td>3</td>
<td>8.36</td>
<td>117.91</td>
</tr>
<tr>
<td>4</td>
<td>8.87</td>
<td>103.10</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>102.29</td>
</tr>
<tr>
<td>6</td>
<td>6.80</td>
<td>101.73</td>
</tr>
<tr>
<td>7</td>
<td>5.83</td>
<td>101.29</td>
</tr>
<tr>
<td>8</td>
<td>4.93</td>
<td>100.84</td>
</tr>
<tr>
<td>9</td>
<td>3.93</td>
<td>100.59</td>
</tr>
<tr>
<td>10</td>
<td>2.88</td>
<td>100.30</td>
</tr>
<tr>
<td>11</td>
<td>1.98</td>
<td>100.18</td>
</tr>
<tr>
<td>12</td>
<td>1.09</td>
<td>100.02</td>
</tr>
<tr>
<td>13</td>
<td>0.75</td>
<td>100</td>
</tr>
<tr>
<td>14</td>
<td>0.73</td>
<td>100</td>
</tr>
<tr>
<td>15</td>
<td>0.64</td>
<td>100</td>
</tr>
</tbody>
</table>

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

Figure 2: Pinout (top view)
<table>
<thead>
<tr>
<th>No</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ANT</td>
<td>RF connection to PCB antenna (see section 4.1)</td>
</tr>
<tr>
<td>2</td>
<td>RF</td>
<td>50Ω RF connection to external antenna or on-board antenna via ANT (see section 4.1)</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td>Negative supply voltage</td>
</tr>
<tr>
<td>4</td>
<td>VCC</td>
<td>Positive supply voltage</td>
</tr>
<tr>
<td>5</td>
<td>STATUS_IND_0</td>
<td>Status indication LED 0, do not connect if not needed</td>
</tr>
<tr>
<td>6</td>
<td>STATUS_IND_1</td>
<td>Status indication LED 1, do not connect if not needed</td>
</tr>
<tr>
<td>7</td>
<td>/RESET</td>
<td>Reset (active low), internal pull-up (100 kΩ)</td>
</tr>
<tr>
<td>8</td>
<td>GPIO12</td>
<td>Unused, output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>9</td>
<td>BOOT</td>
<td>Input with internal pull-down (2.7 kΩ), pull low during start-up to boot the standard application, do not connect if not needed</td>
</tr>
<tr>
<td>10</td>
<td>APP_MODE_0, GPIO22</td>
<td>Input, internal weak pull-down (see section 5.2), do not connect if not needed</td>
</tr>
<tr>
<td>11</td>
<td>APP_MODE_1, GPIO0</td>
<td>Input, internal weak pull-down (see section 5.2), do not connect if not needed</td>
</tr>
<tr>
<td>12</td>
<td>GPIO30</td>
<td>Unused, output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>13</td>
<td>GND</td>
<td>Negative supply voltage</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>GND</td>
<td>Negative supply voltage</td>
</tr>
<tr>
<td>15</td>
<td>UTXD</td>
<td>Module UART TX</td>
</tr>
<tr>
<td>16</td>
<td>URXD</td>
<td>Module UART RX, internal weak pull-up</td>
</tr>
<tr>
<td>17</td>
<td>GPIO3</td>
<td>Unused, output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>18</td>
<td>WAKE_UP</td>
<td>Wake-up on rising edge, internal pull-down, do not connect if not needed</td>
</tr>
<tr>
<td>19</td>
<td>GPIO5</td>
<td>Unused, output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>20</td>
<td>GPIO6/CTS</td>
<td>Optionally UART CTS (see section 6.2), internal pull-down, do not connect if not needed</td>
</tr>
<tr>
<td>21</td>
<td>GPIO7/RTS</td>
<td>Optionally UART RTS (see section 6.2), output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>22</td>
<td>GND</td>
<td>Negative supply voltage</td>
</tr>
<tr>
<td>No</td>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----</td>
<td>-----------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>23</td>
<td>GND</td>
<td>Negative supply voltage</td>
</tr>
<tr>
<td>24</td>
<td>GPIO10</td>
<td>Unused, output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>25</td>
<td>GPIO11</td>
<td>Unused, output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>26</td>
<td>GPIO14</td>
<td>Unused, output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>27</td>
<td>GPIO15</td>
<td>Unused, output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>28</td>
<td>GPIO16</td>
<td>Unused, output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>29</td>
<td>GPIO17</td>
<td>Unused, output LOW, do not connect if not needed</td>
</tr>
<tr>
<td>30</td>
<td>JTAG_TDI</td>
<td>Debug line (locked), do not connect</td>
</tr>
<tr>
<td>31</td>
<td>JTAG_TDO</td>
<td>Debug line (locked), do not connect</td>
</tr>
<tr>
<td>32</td>
<td>JTAG_TCK</td>
<td>Debug line (locked), internal pull-down, do not connect</td>
</tr>
<tr>
<td>33</td>
<td>JTAG_TMS</td>
<td>Debug line (locked), do not connect</td>
</tr>
<tr>
<td>34</td>
<td>GND</td>
<td>Negative supply voltage</td>
</tr>
</tbody>
</table>

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 $50\,\Omega$ RF signal on pin $RF$ of the module. In this configuration, pin $ANT$ of the module has to be connected to ground and pin $RF$ to the external antenna via $50\,\Omega$ feed line. Refer to chapter 16 for further information.

4.2. Minimal pin configuration

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

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

Table 11: Minimal pin configuration
4.3. Power up

Set and hold the /RESET pin to LOW. After the supply voltage to the module has stabilized, the /RESET pin shall be held LOW level for another Δt of at least 200 ms to ensure a safe start-up. Before releasing the /RESET pin, make sure that the appropriate voltage levels are applied on pins App_Mode_0 and App_Mode_1 according to the desired application mode (see 5.2). Also make sure that the host’s UART TX line to the module is configured as a logic HIGH level during module boot-up. The module will send a Start-up UART message once it has booted and started the application.

For further timing information refer chapter 11. If the module is used on a battery-powered system, using a suitable reset-IC (or a discrete RC block for a delay) is highly recommended to ensure a correct power up and stable behavior independent of battery status.

The typical time for Δt is in the order of 1 second. An additional 1.1 seconds (typical) delay is introduced on first boot due to extended calibration (e.g. after a firmware update). Upon hard-reset the firmware integrity check will add further latency into t_boot.

![Diagram of power up process]

Figure 3: Power up

4.4. Region specific WLAN settings

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

```
AT+wlanSet=general, country_code, EU
OK
```
4.5. Quick start example

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

4.5.1. Prerequisites

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

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

4.5.2. Hardware configuration

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

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

4.5.3. Setup description

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

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

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

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

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

1. In this example an access point with the following settings is used.
   - SSID: WE_calypso
   - Security method: WPA2_PSK
   - Key: calypsowlan

2. Type in the following command into the terminal to connect to the access point.
   ```
   AT+wlanConnect=WE_calypso, ,WPA_WPA2, calypsowlan , , ,
   OK
   ```

   ```
   +eventwlan:connect,WE_calypso,0x24:0xf5:0xa2:0x28:0x97:0x21
   +eventnetapp:ipv4_acquired,192.168.1.168,192.168.1.1,192.168.1.1
   ```

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

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

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

<table>
<thead>
<tr>
<th>Module</th>
<th>MAC Address</th>
<th>IP Address</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0xc8:0xfd:0x19:0x05:0x54:0xb4</td>
<td>192.168.1.168</td>
<td>TCP server</td>
</tr>
<tr>
<td>B</td>
<td>0xc8:0xfd:0x19:0x05:0x74:0x98</td>
<td>192.168.1.140</td>
<td>TCP client</td>
</tr>
</tbody>
</table>

Table 12: Quick start addresses and roles

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

4.5.6. Creating a TCP server

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

1. Create a TCP socket using the following command. The modules returns a socket ID.
   ```
   AT+socket=INET,STREAM,TCP
   +socket:0
   OK
   ```

2. Bind the TCP socket with the corresponding ID to the module IP and port:8888.
   ```
   AT+bind=0,INET,8888,192.168.1.168
   OK
   ```
3. Now the server listens for connection request on the specified port with the following command.

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

4.5.7. Creating a TCP client

Module B should be configured as a TCP client

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

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

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

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

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

4.5.8. Data transfer

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

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

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

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

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

3. At B the data is received as follows.

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

4. Sending "hello" from B

```
AT+send=0,0,5,hello
OK
```
5. Receiving the message at A

AT+recv=1,0,5
OK
+recv:1,0,5,hello
OK
5. **Functional description**

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

### 5.1. **Key features**

In this section, the features of the Calypso module is summarized in the form of a table. Calypso offers the user to configure and exploit its rich features through an easy-to-use command interface over UART.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| Radio standards         | IEEE 802.11 b/g/n station  
IEEE 802.11 b/g Access point (for provisioning only)  
Wi-Fi Direct client and group owner |
| Channels                | 1-13                                                                                                                                         |
| Security                | WEP, WPA/WPA2PSK, WPA2 Enterprise (802.1x)                                                                                                 |
| Provisioning            | In AP mode using the on-board HTTPS server                                                                                                                                                             |
| Network layer           | IPv4, IPv6                                                                                                                                     |
| IP 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                                                                                                                                         |
| Update                  | Secure OTA update with fall back mechanism                                                                                                       |
| Security                | Secure key storage  
Trusted root-certificate catalog  
Encrypted file system  
Secure OTA  
Software tamper detection  
Cloning protection                                                                                                                                  |
| Power management        | 802.11 power save power modes  
Lower power sleep mode with timed or pin wake-up                                                                                                 |

Table 13: Key features

5.2. Modes of operation

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

- Command from the host.
- Position of the App_Mode_x pins during boot up
- /Reset signal
- WAKE_UP signal or time event
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.
<table>
<thead>
<tr>
<th>APP_MODE_1</th>
<th>APP_MODE_0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>AT command normal mode, see chapter 7</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>OTA mode, see chapter 12</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Provisioning mode, see chapter 9</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>AT command terminal mode, see chapter 7</td>
</tr>
</tbody>
</table>

Table 14: Application modes

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

5.2.2. Idle

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

5.2.3. OTA update

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

5.2.4. Provisioning

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

5.2.5. Hibernate

It is essential to have a low power sleep mode especially for a battery powered systems. Calypso offer a hibernate mode with a very low current consumption of less than 10 \(\mu\)A. Section 8.1.5 describes the commands used to put the module to hibernate and chapter 11 describes the timing characteristics. On wake-up, the module starts from the reset vector meaning that the RAM contents are lost. Based on the WiFi connection policy(see8.2.6), the module can be set up to automatically connect to a saved access point profile and acquire IP address. However, the socket connections are lost on entering the sleep mode and have to be re-establish on wake-up.
6. Host connection

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

![Host Interface Diagram]

Figure 6: Host Interface

6.1. UART parameters

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud</td>
<td>115200 to 3000000</td>
<td>921600</td>
</tr>
<tr>
<td>Data bits</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Stop bits</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Parity</td>
<td>none, odd, even</td>
<td>even</td>
</tr>
<tr>
<td>Flow control</td>
<td>none, RTS/CTS</td>
<td>none</td>
</tr>
</tbody>
</table>

Table 15: 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 and the parity of the UART can be configured using corresponding commands (see section 8.1). The data format is fixed to 8 data bits and one stop bit. This results in a 11 UART symbols per 8 bit user data ratio.
6.2. Hardware flow control

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

6.3. Timing and characteristics

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

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

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

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

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

7.1. Command types

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

- Requests: The host requests the module to perform an action or start an operation. All Requests start with the "AT+" tag and end with "\r\n" (hex: 0x0D0A).

- Confirmations: On each request, the module answers with a confirmation message to give a feedback on the requested operation status. All Confirms contain the Request itself and either a "OK" or an error code. Appendix B gives a brief description of all the error codes. All confirmations end with "\r\n" (hex: 0x0D0A).

- Events: The module indicates spontaneously when a special event has occurred. All events start with the "+" tag and contain further data, error codes (see Appendix B) or status information. All events end with "\r\n" (hex: 0x0D0A).

7.2. AT command characteristics

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

7.2.1. Request

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

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

- All commands start with the prefix "AT". The delimiter "+" indicates the beginning of the command name.
• Commands can have parameters in which case the delimiter ";" separates the command name from the list of parameters.

• AT commands can be entered in upper or lower case with optional whitespace between the arguments.

• Further, each parameter is separated from the next with a "," delimiter. A comma shall not be followed by a whitespace.

• In cases where a parameter is optional or ignored, it could be left empty. Nevertheless the "," delimiter has to be present. An empty parameter looks like ";".

• String parameters containing spaces must be enclosed with quotation marks (""").

• All hexadecimal parameters must have a 0x prefix.

• MAC and network addresses must be entered as follows
  – MAC address - Six hexadecimal values of 8 bit each, represented as X:X:X:X:X:X (X can range from 0x00 up to 0xFF), the ":" is used as delimiter
  – IPv4 address - Four decimal values of 8 bit each, represented as X.X.X.X (X can range from 0 up to 255), the "." is used as delimiter per 8 bit
  – IPv6 address - Four hexadecimal numeric values of 32 bit each, represented as X:X:X:X (X can range from 0x00000000 up to 0xFFFFFFFF), the ":" is used as delimiter per 32 bit

• Bitmask parameters are represented using "|" delimiter - e.g. x|y

• Data should be either binary or Base64 format (binary to text encoding). 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 commands confirms have the following syntax,

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

On success, the confirms contain a positive acknowledgement.

OK

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

ERROR: <error description>, <error code>

7.2.3. Events

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

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

8.1. Device commands

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

8.1.1. Start and stop commands

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

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+start</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td>Arguments:</td>
</tr>
<tr>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 16: AT+start

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+stop=[timeout]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td>Arguments:</td>
</tr>
<tr>
<td>Timeout: in milliseconds</td>
<td>None</td>
</tr>
<tr>
<td>• 0 - Hibernate immediately without waiting for a response from the NWP.</td>
<td></td>
</tr>
<tr>
<td>• 0xFFFF - Wait indefinitely for a response from the NWP.</td>
<td></td>
</tr>
<tr>
<td>• 0 &lt; time-out &lt; 0xFFFF - Wait for timeout before forcing to hibernate.</td>
<td></td>
</tr>
</tbody>
</table>

Table 17: AT+stop

8.1.2. Test

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

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+reboot</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td>Arguments:</td>
</tr>
<tr>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 19: AT+reboot

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

### 8.1.4. Factory reset

The factory reset command restores the module to factory state.

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

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+factoryreset</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td>Arguments:</td>
</tr>
<tr>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 20: AT+factoryreset
Factory reset operation can take up to 30 seconds to complete. The module responds with an "OK" only after this time period.

Reseting 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\(\mu\)A. In the hibernate mode, the network processor is in hibernate mode and the application processor is shut down. The module wakes up automatically after a time period specified in the sleep command. Alternatively, the module can be woken up manually with a rising edge on the \textit{WAKE\_UP} pin. On wake up, the module starts from the reset vector.

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+sleep=[timeout]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>Timeout: in seconds, min 1, max 86400(24 hrs)</td>
<td>Arguments: None</td>
</tr>
</tbody>
</table>

Table 21: AT+sleep

8.1.6. Get

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

- The status bit mask is cleared once it is read. Additionally, status information is also available as events (see 7).
- The system persistent setting is enabled by default. This means that all the settings are retained after reset.
<table>
<thead>
<tr>
<th>ID</th>
<th>option</th>
<th>Return values</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>device</td>
<td>value1 : error bitmask</td>
</tr>
<tr>
<td></td>
<td>WLAN</td>
<td>value1 : bitmask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- WLANASYNCCONNECTEDRESPONSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- WLANASYNCDISCONNECTEDRESPONSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- STA_CONNECTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- STA_DISCONNECTED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- P2P_DEV_FOUND</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CONNECTION_FAILED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- P2P_NEG_REQ_RECEIVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- RX_FILTERS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- WLAN_STA_CONNECTED</td>
</tr>
<tr>
<td>BSD</td>
<td></td>
<td>value1 : bitmask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TX_FAILED</td>
</tr>
<tr>
<td>netapp</td>
<td></td>
<td>value1 : bitmask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IPACQUIRED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IPACQUIRED_V6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IP_LEASED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IP_RELEASED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IPV4_LOST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IP_COLLISION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IPV6_LOST</td>
</tr>
<tr>
<td>general</td>
<td>version</td>
<td>chip ID, FW version (X.X.X.X),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PHY version (X.X.X.X), NWP Version (X.X.X.X),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROM version</td>
</tr>
<tr>
<td></td>
<td>time</td>
<td>hh,mm,ss,dd,mm,yyyy</td>
</tr>
<tr>
<td></td>
<td>persistent</td>
<td>1=enable, 0=disable</td>
</tr>
<tr>
<td>IOT</td>
<td>UDID</td>
<td>16 byte UDID</td>
</tr>
<tr>
<td>UART</td>
<td>baudrate</td>
<td>baudrate</td>
</tr>
<tr>
<td></td>
<td>parity</td>
<td>0=None, 1=even, 2=odd</td>
</tr>
</tbody>
</table>

Table 22: AT+get

8.1.7. Set

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

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

8.2.1. Set mode

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

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

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

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+wlanSetMode=[mode]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

Arguments:
- STA : for station mode
- AP : for access point mode
- P2P : for P2P mode

Table 24: AT+wlanSetMode
8.2.2. Scan

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

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

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+wlanScan=[index],[count]</td>
<td>+wlanScan:&lt;Device[index]&gt;, . . . OK or error</td>
</tr>
</tbody>
</table>

Arguments:
- Index: starting index 0-29
- count: number of device max 30

Table 25: AT+wlanScan

8.2.3. Manual connection

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

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+wlanDisconnect</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

Table 27: AT+wlanDisconnect

8.2.4. Profiles

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

**Request**

```
AT+wlanProfileAdd=[SSID], [BSSID], [SecurityType], [SecurityKey],
[SecurityExtUser], [SecurityExtAnonUser],
[SecurityExtEapMethod],[priority]
```

**Response**

```
+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
- 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>
<thead>
<tr>
<th>Table 28: AT+wlanProfileAdd</th>
</tr>
</thead>
</table>

### AT+wlanProfileGet

**Request**

```
AT+wlanProfileGet=[index]
```

**Response**

```
+wlanProfileGet:<Profile[index]> . . .
OK or error
```

**Arguments:**
- Index: profile index 0- 6

<table>
<thead>
<tr>
<th>Table 29: AT+wlanProfileGet</th>
</tr>
</thead>
</table>

### AT+wlanProfileDel

**Request**

```
AT+wlanProfileDel=[index]
```

**Response**

```
OK or error
code
```

**Arguments:**
- Index: profile index 0- 6

| Table 30: AT+wlanProfileDel |
8.2.5. WLAN settings

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

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+wlanSet= [ID],[option],[value1],...,[valueX]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>option</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>COUNTRY_CODE</td>
<td>US, EU or JP</td>
</tr>
<tr>
<td></td>
<td>STA_TX_POWER</td>
<td>0-15 (0 = Max transmit power)</td>
</tr>
<tr>
<td></td>
<td>AP_TX_POWER</td>
<td>0-15 (0 = Max transmit power)</td>
</tr>
<tr>
<td></td>
<td>SCAN_PARAMS</td>
<td>value1: channel mask, value2: RSSI threshold</td>
</tr>
<tr>
<td></td>
<td>SUSPEND_PROFILES</td>
<td>Suspend profile bit mask</td>
</tr>
<tr>
<td></td>
<td>DISABLE_ENT_SERVER_AUTH</td>
<td>0 or 1 (1 = disable server auth when manually connecting to an enterprise network)</td>
</tr>
<tr>
<td>P2P</td>
<td>CHANNEL_N_REGS</td>
<td>value1: Listen channel (1/6/11), value2: Listen regulatory class (81), value3: operating channel (1/6/11), value4: operating regulatory class (81)</td>
</tr>
<tr>
<td>AP</td>
<td>SSID</td>
<td>String up to 32 characters</td>
</tr>
<tr>
<td></td>
<td>CHANNEL</td>
<td>Channels 1-11</td>
</tr>
<tr>
<td></td>
<td>HIDDEN_SSID</td>
<td>0: disabled, 1: send empty</td>
</tr>
<tr>
<td></td>
<td>SECURITY</td>
<td>open, WEP, WPA_WPA2</td>
</tr>
<tr>
<td></td>
<td>PASSWORD</td>
<td>WEP: 8-63 characters, WPA: 5-13 characters</td>
</tr>
<tr>
<td></td>
<td>MAX_STATIONS</td>
<td>1-4</td>
</tr>
</tbody>
</table>

Table 31: AT+wlanSet
Request | Response
--- | ---
AT+wlanGet=[ID],[option] | +Get:[value1],...,[valueX]
Arguments: | Arguments: see table 31

<table>
<thead>
<tr>
<th>ID</th>
<th>option</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>COUNTRY_CODE</td>
</tr>
<tr>
<td></td>
<td>STA_TX_POWER</td>
</tr>
<tr>
<td></td>
<td>SCAN_PARAMS</td>
</tr>
<tr>
<td>P2P</td>
<td>CHANNEL_N_REGS</td>
</tr>
<tr>
<td>Connection</td>
<td>Role, Status, security, SSID, BSSID, device name</td>
</tr>
<tr>
<td>AP</td>
<td>SSID</td>
</tr>
<tr>
<td></td>
<td>CHANNEL</td>
</tr>
<tr>
<td></td>
<td>HIDDEN_SSID</td>
</tr>
<tr>
<td></td>
<td>SECURITY</td>
</tr>
<tr>
<td></td>
<td>PASSWORD</td>
</tr>
<tr>
<td></td>
<td>MAX_STATIONS</td>
</tr>
<tr>
<td></td>
<td>MAX_STA_AGING</td>
</tr>
</tbody>
</table>

Table 32: AT+wlanGet

### 8.2.6. WLAN policy

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

- **Connection**: This policy defines how the device initiates and maintains a specific connection after reset. The user can use one of the following options,
  - **Auto**: The device automatically tries to connect to the stored profiles based on priority. In case of several profiles with same priority, the decision is made based on security type (WPA2>WEP>OPEN). In case of same security type, the one with the highest signal strength is chosen to be connected.
  - **Fast**: The device tries to connect to the last connected AP without transmitting a probe request.
  - **AnyP2P**: The device connect to the first available Wi-Fi direct device.

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

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

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

<table>
<thead>
<tr>
<th>ID</th>
<th>option</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection</td>
<td>Auto, Fast or P2P</td>
<td>scan interval in seconds</td>
</tr>
<tr>
<td>scan</td>
<td>Hidden_SSID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Disable_Scan</td>
<td></td>
</tr>
<tr>
<td>PM</td>
<td>Normal, low latency, low power or long sleep</td>
<td>Maximum sleep time in ms only for long sleep option</td>
</tr>
<tr>
<td>P2P</td>
<td>CLIENT, GROUP_OWNER, NEGOTIATE</td>
<td>ACTIVE, RAND_BACKOFF, PASSIVE</td>
</tr>
</tbody>
</table>

Table 33: AT+wlanPolicySet

AT+wlanPolicyGet= [Type]
+wlanPolicyGet:[option],[value]

Arguments: connection, scan, PM or P2P

8.3. Network configuration commands

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

- **IPv4 Stateful with Stateless fallback**: In this mode, the device waits for an IPv4 address from a DHCP server. On time-out, the LLA address is used. The LLA IP 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**: Addressed configured by the user.

- **IPv6 SLAAC**: The least significant 64 bits are filled with the device MAC address in EUI-64 format. In case of duplicate address (DAD failure), random 64 bits are used.

- **IPv6 Stateful (DHCPv6)**: IPv6 LLA is acquired from a DHCPv6 server. In case for DAD failure, Stateless configuration is used.

- **Static**: Preconfigured by the user. In case of DAD failure, a failure event is sent to the host.
- **Link-Global IPv6**: The IPv6 global address can be acquired similar to the LLA stateless (MSB 64 bits from RA messages), statefull or static.

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>Wi-Fi Station</th>
<th>Wi-Fi AP</th>
<th>Wi-Fi Direct</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4</td>
<td>Always enabled</td>
<td>static</td>
<td>client - like station GO - like AP</td>
</tr>
<tr>
<td></td>
<td>one address- DHCP, LLA, Static</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPv6</td>
<td>disabled</td>
<td>not supported</td>
<td>not supported</td>
</tr>
<tr>
<td></td>
<td>two addresses- Local, Stateless, Statefull, Static</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Global - Stateless, Statefull, Static</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 35: IP addresses
<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+netCfgSet=[ID],[option],[value1],...,[valueX]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>option</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF</td>
<td>STATE (Enable/disable) bitmask</td>
<td>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</td>
</tr>
<tr>
<td>SET_MAC_ADDR</td>
<td></td>
<td>MAC Address</td>
</tr>
<tr>
<td>IPV4_STA_ADDR</td>
<td>STATIC, DHCP_LLA, RELEASE_IP_OFF, RELEASE_IP_SET, DHCP</td>
<td>For static only, value1 : IP address, value2 : subnet mask, value3 : Default gateway, value4 : DNS</td>
</tr>
<tr>
<td>IPV6_ADDR_LOCAL</td>
<td>STATIC, STATELESS, STATEFUL</td>
<td>IP address</td>
</tr>
<tr>
<td>IPV6_ADDR_GLOBAL</td>
<td>STATIC, STATELESS, STATEFUL</td>
<td>value1: IP address, value2 :DNS IP</td>
</tr>
<tr>
<td>IPV4_DNS_CLIENT</td>
<td></td>
<td>Secondary DNS</td>
</tr>
</tbody>
</table>

Table 36: AT+netCfgSet
### Table 37: AT+netCfgGet

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+netCfgGet= [configID]</td>
<td>+netCfgGet:[option],[value1],...,[valueX] OK or error code</td>
</tr>
<tr>
<td>Arguments:</td>
<td>Arguments:</td>
</tr>
<tr>
<td>GET_MAC_ADDR</td>
<td>Mac address</td>
</tr>
<tr>
<td>IPV4_STA_ADDR or IP_AP_ADDR</td>
<td>Method, IP Address, Subnet mask, Gateway, DNS</td>
</tr>
<tr>
<td>IPV6_ADDR_LOCAL or IPV6_ADDR_GLOBAL</td>
<td>Method, IP address</td>
</tr>
<tr>
<td>IP_DNS_CLIENT</td>
<td>Secondary DNS address</td>
</tr>
</tbody>
</table>

Calypso reference manual version 1.7  ©July 2019
www.we-online.com/wireless-connectivity  49
8.4. Socket commands

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

8.4.1. Sockets work flow

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

- Connectionless socket: Also know as Datagram sockets, this allows data exchange between network entities without establishing a connection. This results in minimal connection latency but cannot ensure data integrity or packet order.

- Connection-oriented socket: Stream sockets establish a connection between two entities before data exchange there by ensuring data integrity and packet order.

8.4.1.1. TCP socket

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

![TCP socket work flow diagram]

Figure 7: TCP socket work flow
8.4.1.2. UDP socket

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

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 are exchanged between a TCP client and server leading to mutual authentication and encryption of data messages. The TLS/SSL handshake is summarized in the figure 9. The SSL/TLS processes are handled in an separate execution environment and hardware acceleration is used to speed up the cryptographic operations.
The SSL/TLS protocol requires certificates for authentication and a trusted root certificate catalog to verify the certificates. The calypso provides secure key storage option through the encrypted file system (see section 8.5). A trusted root certificate catalog is present on board with a set of well known trusted root CAs (see appendix C).

8.4.3. Socket operations

In this section, the AT commands used to perform various operations on a socket is described. A socket can be created using the command `AT+socket` and the socket descriptor (socketID) returned by this command can be used to perform all the other socket operations. Socket select command allows monitoring multiple sockets and triggering on specific events.
<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+socket=[family],[type],[protocol]</td>
<td>+socket:[socketID] OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>- family : INET or INET6</td>
<td></td>
</tr>
<tr>
<td>- Type : STREAM or DGRAM</td>
<td></td>
</tr>
<tr>
<td>- protocol : TCP, UDP or SEC</td>
<td></td>
</tr>
</tbody>
</table>

Table 38: AT+socket (create a socket)

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+close=[socketID]</td>
<td>+close:[socketID] OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>socketID : socket descriptor</td>
<td></td>
</tr>
</tbody>
</table>

Table 39: AT+close (close a socket)

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+bind=[socketID],[family],[localPort],[localAddress]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>socketID: socket descriptor</td>
<td></td>
</tr>
<tr>
<td>- family : INET or INET6</td>
<td></td>
</tr>
<tr>
<td>- localPort : Local port</td>
<td></td>
</tr>
<tr>
<td>- localAddress: Local IP address</td>
<td></td>
</tr>
</tbody>
</table>

Table 40: AT+bind

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+listen=[socketID],[backlog]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>socketID : socket descriptor</td>
<td></td>
</tr>
<tr>
<td>backlog : max length of connect request queue</td>
<td></td>
</tr>
</tbody>
</table>

Table 41: AT+listen
### 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.
<table>
<thead>
<tr>
<th>level</th>
<th>option</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOCKET</td>
<td>KEEPALIVE: enable/disable TCP keep active message</td>
<td>value1: 1=enable, 0=disable</td>
</tr>
<tr>
<td></td>
<td>KEEPALIVETIME: keep alive timeout</td>
<td>value1: timeout in seconds</td>
</tr>
<tr>
<td></td>
<td>RX_NO_IP_BOUNDARY: enable/disable RX IP boundary</td>
<td>value1: 1=enable, 0=disable</td>
</tr>
<tr>
<td></td>
<td>RX_NO_IP_BOUNDARY: enable/disable RX IP boundary</td>
<td>value1: 1=enable, 0=disable</td>
</tr>
<tr>
<td></td>
<td>RCVTIMEO: timeout value that specifies maximum amount of time an input function waits until it completes</td>
<td>value1: seconds value2: microseconds</td>
</tr>
<tr>
<td></td>
<td>RCVBUF: TCP maximum receive window size</td>
<td>value1: size in bytes</td>
</tr>
<tr>
<td></td>
<td>NONBLOCKING: Set socket to non blocking</td>
<td>value1: 1=enable, 0=disable</td>
</tr>
<tr>
<td></td>
<td>SECMETHOD: Sets security method to TCP socket</td>
<td>value1: SSLV3, TLSV1, TLSV1_1, TLSV1_2, SS-LV3_TLSV1_2 (highest possible)</td>
</tr>
<tr>
<td></td>
<td>SECURE_MASK: Set specific ciphers as bit mask (default= all ciphers)</td>
<td>value1: cipher type see table 47</td>
</tr>
<tr>
<td></td>
<td>SECURE_FILES_CA_FILE_NAME: Map secured socket to CA file by name</td>
<td>value1: absolute file path</td>
</tr>
<tr>
<td></td>
<td>SECURE_FILES_PRIVATE_KEY_FILE_NAME: Map secured socket to private key by name</td>
<td>value1: absolute file path</td>
</tr>
<tr>
<td></td>
<td>SECURE_FILES_CERTIFICATE_FILE_NAME: Map secured socket to certificate file by name</td>
<td>value1: absolute file path</td>
</tr>
<tr>
<td></td>
<td>SECURE_FILES_D-H_KEY_FILE_NAME: Map secured socket to Diffie Hellman file by name</td>
<td>value1: absolute file path</td>
</tr>
<tr>
<td></td>
<td>SECURE_DOMAIN_NAME_VERIFICATION: Set a domain name, to check in SSL client connection</td>
<td>value1: Domain name</td>
</tr>
</tbody>
</table>

Table 45: AT+setSockOpt
<table>
<thead>
<tr>
<th>Request</th>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>level</td>
<td>option</td>
<td>value</td>
</tr>
<tr>
<td>IP</td>
<td>MULTICAST_TTL: Set the time-to-live value of outgoing multicast packets</td>
<td>value1: Number of hops</td>
</tr>
<tr>
<td></td>
<td>ADD_MEMBERSHIP:UDP socket, join a multicast group</td>
<td>Value1: IPv4 multicast address Value2: Multicast interface address</td>
</tr>
<tr>
<td></td>
<td>DROP_MEMBERSHIP:UDP socket, leave a multicast group</td>
<td>Value1: IPv4 multicast address Value2: Multicast interface address</td>
</tr>
</tbody>
</table>

Table 46: AT+setSockOpt continued.

<table>
<thead>
<tr>
<th>Supported Cipher methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL_RSA_WITH_RC4_128_SHA</td>
</tr>
<tr>
<td>SSL_RSA_WITH_RC4_128_MD5</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_RC4_128_SHA</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_GCM_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_GCM_SHA384</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_128_GCM_SHA384</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA384</td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA384</td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256</td>
</tr>
</tbody>
</table>

Table 47: 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 45 and 46
timeout usec: Time in microseconds

Table 48: AT+getSockOpt

8.4.5. Socket data exchange

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

Request | Response
---|---
AT+recv=[socketID], [format], [length] | OK
+recv:[socketID], [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 49: AT+recv

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

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+recvFrom=[socketID],[family],[remotePort], [remoteAddress], [format], [length]</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>+recvFrom:[socketID], [length],[data]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments:</th>
</tr>
</thead>
<tbody>
<tr>
<td>socketID : socket descriptor</td>
</tr>
<tr>
<td>family : INET or INET6</td>
</tr>
<tr>
<td>remotePort : Port of the peer to connect to</td>
</tr>
<tr>
<td>remoteAddress : Address to connect to</td>
</tr>
<tr>
<td>format : data format 0=binary, 1=base64 (binary to text encoding)</td>
</tr>
<tr>
<td>length : Max number of bytes to receive</td>
</tr>
</tbody>
</table>

Table 50: AT+recvFrom

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

### AT+send

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+send=[socketID], [format], [length],[data]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments:</th>
</tr>
</thead>
<tbody>
<tr>
<td>socketID : socket descriptor</td>
</tr>
<tr>
<td>format : data format 0=binary, 1=base64 (binary to text encoding)</td>
</tr>
<tr>
<td>length : number of bytes to send (max 1460)</td>
</tr>
<tr>
<td>data: date to send</td>
</tr>
</tbody>
</table>

Table 51: AT+send

Firmware version 1.2.0 and older has known issues when transmitting binary data. See chapter Firmware history for more details.
### Request

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AT+sendTo=[socketID],[family],[remotePort],[remoteAddress],[format],[length],[data]</code></td>
<td>OK or error</td>
</tr>
</tbody>
</table>

**Arguments:**
- `socketID` : socket descriptor
- `family` : INET or INET6
- `remotePort` : Port of the peer to connect to
- `remoteAddress` : Address to connect to
- `format` : data format 0=binary, 1=base64 (binary to text encoding)
- `length` : number of bytes to send (max 1460)
- `data` : date to send

**Table 52: AT+sendTo**

---

**Firmware version 1.2.0 and older has known issues when transmitting binary data. See chapter Firmware history for more details.**

---

### 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. Here are some of the features of the file system.

The storage capacity for additional content in the radio module’s file system is limited to the available capacity in the file system itself.

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

Minimize the number of writes to flash to ensure data endurance

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

Switch off the NWP when performing a file operation

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

8.5.1. File system operations
Table 53: AT+fileGetFileList

8.5.2. File operations

In this section, the file operation commands are described.
### AT+fileOpen=[fileName],[options],[fileSize]

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+fileOpen=[fileName],[options],[fileSize]</td>
<td>+fileOpen:[fileID],[secureToke] OK or error</td>
</tr>
</tbody>
</table>

**Arguments:**
- fileName : full file path
- options:
  - READ - Read a file (no bit mask)
  - WRITE - Open for write (optionally bitmask with CREATE)
  - CREATE - Create a new file (optionally bitmask with WRITE or OVERWRITE)
  - CREATE_FAILSAFE
  - CREATE_SECURE
  - CREATE_NOSIGNATURE (for secure files only)
  - CREATE_STATIC_TOKEN (for secure files only)
  - CREATE_VENDOR_TOKEN (for secure files only)
  - CREATE_PUBLIC_WRITE (for secure files only)
  - CREATE_PUBLIC_READ (for secure files only)
- fileSize : Max file size in bytes (mandatory for CREATE option)

**Table 54: AT+fileOpen**

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

## AT+fileClose=[fileID],[certificateFileName],[signature]

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+fileClose=[fileID],[certificateFileName],[signature]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

**Arguments:**
- fileID : ID assigned from AT+FileOpen
- certificateFileName: Full path to certificate (optional)
- signature : The signature is SHA1 (optional)

**Table 55: AT+fileClose**
### Table 56: AT+fileDel

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+fileDel=[fileName],[secureToke]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

**Arguments:**
- **FileName:** Full path to file
- **secureToken:** Token assigned from AT+fileOpen (optional)

### Table 57: AT+fileGetInfo

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+fileGetInfo=[fileName],[secureToke]</td>
<td>+FileGetInfo:[Flags],[FileSize], [Allocated-Size],[Tokens],[storageSize], [WriteCounter]</td>
</tr>
</tbody>
</table>

**Arguments:**
- **FileName:** Full path to file
- **secureToken:** Token assigned from AT+fileOpen (optional)

OK or error

### Table 58: AT+fileRead

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+fileRead=[fileID],[offset],[format],[length]</td>
<td>+FileRead:[format], [numberOfReadBytes], [data]</td>
</tr>
</tbody>
</table>

**Arguments:**
- **fileID:** ID assigned from AT+fileOpen
- **offset:** Offset to specific read block
- **Format:** 0=binary, 1=Base64
- **Length:** Number of bytes to read

OK or error
### Request

| 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 |

**Table 59: AT+fileWrite**

---

**Firmware version 1.2.0 and older has known issues when transmitting binary data. See chapter Firmware history for more details.**

---

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

---

### 8.6. Network application commands

#### 8.6.1. mDNS

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

---

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

---

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

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

SNTP client is disabled by default.

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

Table 62: AT+netAPPGet
### AT+netAPPSet

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+netAPPSet=[AppID],[option], [value1]..., [valueX]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

**Arguments:**
- **AppID**: sntp_client
- **Options**:
  - `enable`, value: 0=disabled, 1=enabled
  - `update_interval`, value: minimum update interval in seconds
  - `time_zone`, value: UTC +/- minutes
  - `server_address`, value1: server index (0-2), value2: server address (IP addresses or URL)

Table 63: AT+netAPPSet

### AT+netappUpdateTime

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+netappUpdateTime</td>
<td>OK or error</td>
</tr>
<tr>
<td>Synchronize device time with SNTP server</td>
<td></td>
</tr>
</tbody>
</table>

Table 64: AT+netappUpdateTime

#### 8.6.3. HTTP client

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

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+httpCreate</td>
<td>+HttpCreate:[index]</td>
</tr>
<tr>
<td>OK or error</td>
<td>Arguments: index - client handle for all further operations</td>
</tr>
</tbody>
</table>

Table 65: AT+httpCreate
<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+httpDestroy=[index]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>index - client handle</td>
<td></td>
</tr>
</tbody>
</table>

Table 66: AT+httpDestroy

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+httpConnect=[index],[host],[flags],[private key], [cert], [ca]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>index - client handle</td>
<td></td>
</tr>
<tr>
<td>host - host name</td>
<td></td>
</tr>
<tr>
<td>flags - bitmask (ignore_proxy, host_exist)</td>
<td></td>
</tr>
<tr>
<td>private key - full path (optional)</td>
<td></td>
</tr>
<tr>
<td>certificate - full path (optional)</td>
<td></td>
</tr>
<tr>
<td>ca- full path (optional)</td>
<td></td>
</tr>
</tbody>
</table>

Table 67: AT+httpConnect

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+httpDisconnect=[index]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>index - client handle</td>
<td></td>
</tr>
</tbody>
</table>

Table 68: AT+httpDisconnect

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+httpSetProxy=[family],[port],[address]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>family - INET or INET6</td>
<td></td>
</tr>
<tr>
<td>port - proxy server port</td>
<td></td>
</tr>
<tr>
<td>address - proxy server address</td>
<td></td>
</tr>
</tbody>
</table>

Table 69: AT+httpSetProxy
Table 70: AT+httpSendReq

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+httpSendReq=[index],[method],[uri],[flags],[format], [length], [data]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

Table 71: AT+httpReadResBody

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+httpReadResBody=[index], [format], [length]</td>
<td>+HttpReadResBody:[index], [flag], [format], [length], [body]</td>
</tr>
</tbody>
</table>

Table 70: AT+httpSendReq

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

Table 71: AT+httpReadResBody

Firmware version 1.2.0 and older has known issues when transmitting binary data. See chapter Firmware history for more details.
Firmware version 1.2.0 and older has known issues when transmitting binary data. See chapter Firmware history for more details.

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+httpSetHeader=[index],[option],[flags],[format], [length], [data]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

Arguments:
- index - client handle
- option - see table 74
- flags - bitmask (not_persistent, persistent)
- format - data format(0=Bin,1=Base64)
- length - (optional)
- data - (optional)

Table 72: AT+httpSetHeader

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+httpGetHeader=[index],[option], [format], [length]</td>
<td>+HttpGetHeader:[index],[format], [length],[data]</td>
</tr>
</tbody>
</table>

Arguments:
- index - client handle
- option - see table 74
- format - data format(0=Bin,1=Base64)
- length - max data length
- data - value

Table 73: AT+httpGetHeader
<table>
<thead>
<tr>
<th>Header options</th>
</tr>
</thead>
<tbody>
<tr>
<td>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, resWarning, 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</td>
</tr>
</tbody>
</table>

Table 74: HTTP header options

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

### 8.6.4. MQTT client

MQTT (Message Queue Telemetry Transport) is a machine-to-machine (M2M) connectivity protocol based on publish/subscriber transport mechanism. Features like lightweight, low network bandwidth, scalability makes it ideal for low-power, low-bandwidth IoT applications. A MQTT network consists of a broker connected to multiple clients. Clients can each subscribe to several topics or publish on any topic. The broker on the other hand is responsible for receiving a published topic and pushing it to all the subscribed nodes.

Calypso offers AT commands to create an MQTT client, subscribe as well as publish topics. The following section describes these commands.
### Request

<table>
<thead>
<tr>
<th>AT+mqttCreate=[clientID], [flags], [server address], [server port], [security method], [cipher] [private key], [CA], [DH key], [protocol], [blocking send], [data format]</th>
<th>+mqttCreate:[index] or error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arguments:</strong></td>
<td><strong>index-client handle used for all other MQTT operations</strong></td>
</tr>
<tr>
<td>-client ID: MQTT client ID string</td>
<td></td>
</tr>
<tr>
<td>-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</td>
<td></td>
</tr>
<tr>
<td>-server address: IP or URL</td>
<td></td>
</tr>
<tr>
<td>-server port: 0-65535</td>
<td></td>
</tr>
<tr>
<td>-security method: SSLV3, TLSV1, TLSV1_1, TLSV1_2, TLSV1_1, SSLV3_TLSV1_2 (mandatory if sec flag)</td>
<td></td>
</tr>
<tr>
<td>-cipher: cipher type see table 47 (optional)</td>
<td></td>
</tr>
<tr>
<td>-private key: Full path to key file (optional)</td>
<td></td>
</tr>
<tr>
<td>-certificate: Full path to certificate (optional)</td>
<td></td>
</tr>
<tr>
<td>-CA: Full path to CA (optional)</td>
<td></td>
</tr>
<tr>
<td>-DH key: Full path to Diffie Hellman key (optional)</td>
<td></td>
</tr>
<tr>
<td>-protocol: v3_1 = MQTT version 3.1, v3_1_1 = MQTT version 3.1.1</td>
<td></td>
</tr>
<tr>
<td>-blocking send: 0 = do not wait for server response, 1 = wait for server response</td>
<td></td>
</tr>
<tr>
<td>-data format: set globally for all further commands, 0 = bin, 1 = Base64</td>
<td></td>
</tr>
</tbody>
</table>

**Table 75: AT+mqttCreate**

### Request

<table>
<thead>
<tr>
<th>AT+mqttDelete=[index]</th>
<th>OK or error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arguments:</strong></td>
<td></td>
</tr>
<tr>
<td>index: client handle</td>
<td></td>
</tr>
</tbody>
</table>

**Table 76: AT+mqttDelete**
### Table 77: AT+mqttConnect

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+mqttConnect=[index]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>index: client handle</td>
<td></td>
</tr>
</tbody>
</table>

### Table 78: AT+mqttDisconnect

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+mqttDisconnect=[index]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>index: client handle</td>
<td></td>
</tr>
</tbody>
</table>

### Table 79: AT+mqttPublish

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+mqttPublish=[index],[topic],[QOS],[retain], [messageLength],[message]</td>
<td>OK or error</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>- index: client handle</td>
<td></td>
</tr>
<tr>
<td>- topic: topic string</td>
<td></td>
</tr>
<tr>
<td>- QOS: QOS0, QOS1, QOS2</td>
<td></td>
</tr>
<tr>
<td>- retain: 0 = do not retain, 1 = retain</td>
<td></td>
</tr>
<tr>
<td>- message length: max 1460</td>
<td></td>
</tr>
<tr>
<td>- message: payload</td>
<td></td>
</tr>
</tbody>
</table>

Firmware version 1.2.0 and older has known issues when transmitting binary data. See chapter Firmware history for more details.
### Table 80: AT+mqttSubscribe

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+mqttSubscribe=[index], [number of Topics], [topic1],[QoS],[reserved1], . . . , [topicX],[QoS],[reservedX]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

**Arguments:**
- **index**: client handle
- **number of topics**: max 4
- **topic**: topic string
- **QOS**: QOS0, QOS1, QOS2
- **reserved**: leave empty

### Table 81: AT+mqttUnsubscribe

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+mqttUnsubscribe=[index], [number of Topics], [topic1],[reserved1], . . . , [topicX],[reservedX]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

**Arguments:**
- **index**: client handle
- **number of topics**: max 4
- **topic**: topic string
- **reserved**: leave empty

### Table 82: AT+mqttSet

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+mqttSet=[index],[option], [value1],. . . ,[valueX]</td>
<td>OK or error</td>
</tr>
</tbody>
</table>

**Arguments:**
- **index**: client index

<table>
<thead>
<tr>
<th><strong>option</strong></th>
<th><strong>value</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>username string</td>
</tr>
<tr>
<td>password</td>
<td>password string</td>
</tr>
<tr>
<td>will</td>
<td>topic, QOS, retain, Message-Length, message</td>
</tr>
<tr>
<td>keepalive</td>
<td>value in seconds</td>
</tr>
<tr>
<td>clean</td>
<td>0 = persistent connection, 1 = clean connection</td>
</tr>
</tbody>
</table>

Table 82: AT+mqttSet
8.6.5. Ping

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

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+netAPPPing=[family], [destination], [size], [delay], [timeout], [max], [flags]</td>
<td>+netAPPPing:[packetsSent], [packetReceived], [RoundTripTime] OK or error</td>
</tr>
</tbody>
</table>

Arguments:
- family : INET or INET6
- destination : Destination IP address (0 to stop an ongoing ping)
- size : Size of ping in bytes
- delay : Delay between pings in milliseconds
- timeout : Timeout for each ping in milliseconds
- max : Number of pings to send (0 = forever)
- flag : 0 = report once all pings are done, 1 = report after every ping, 2= Stop after first successful ping

Table 83: AT+netAPPPing

8.7. Events

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

8.7.1. General events

The general event may be received in relation to general device operation.
### Table 84: `+eventgeneral` event

<table>
<thead>
<tr>
<th>ID</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>reset_request</td>
<td>value1=Code</td>
</tr>
<tr>
<td></td>
<td>value2=Software module</td>
</tr>
<tr>
<td></td>
<td>- other</td>
</tr>
<tr>
<td></td>
<td>- wlan</td>
</tr>
<tr>
<td></td>
<td>- netcfg</td>
</tr>
<tr>
<td></td>
<td>- netapp</td>
</tr>
<tr>
<td></td>
<td>- security</td>
</tr>
<tr>
<td>error</td>
<td>value1=Code</td>
</tr>
<tr>
<td></td>
<td>value2=Software module</td>
</tr>
<tr>
<td></td>
<td>- other</td>
</tr>
<tr>
<td></td>
<td>- wlan</td>
</tr>
<tr>
<td></td>
<td>- netcfg</td>
</tr>
<tr>
<td></td>
<td>- netapp</td>
</tr>
<tr>
<td></td>
<td>- security</td>
</tr>
</tbody>
</table>

**8.7.2. WLAN events**

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

**+eventwlan:** [ID]. [Value 1]. . . . [Value X]

<table>
<thead>
<tr>
<th>ID</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect</td>
<td>value1 = SSID&lt;br&gt;value2 = BSSID</td>
</tr>
<tr>
<td>disconnect</td>
<td>value1 = SSID&lt;br&gt;value2 = BSSID&lt;br&gt;value3 = Reason, see chapter B.1</td>
</tr>
<tr>
<td>sta_added</td>
<td>value1 = MAC</td>
</tr>
<tr>
<td>sta_removed</td>
<td>value1 = MAC</td>
</tr>
<tr>
<td>p2p_connect</td>
<td>value1 = SSID&lt;br&gt;value2 = MAC&lt;br&gt;value3 = GO device name</td>
</tr>
<tr>
<td>p2p_disconnect</td>
<td>value1 = SSID&lt;br&gt;value2 = MAC&lt;br&gt;value3 = Reason, see chapter B.1&lt;br&gt;value4 = GO device name</td>
</tr>
<tr>
<td>p2p_client_added</td>
<td>value1 = MAC&lt;br&gt;value2 = GO device name&lt;br&gt;value3 = Own SSID</td>
</tr>
<tr>
<td>p2p_client_removed</td>
<td>value1 = MAC&lt;br&gt;value2 = GO device name&lt;br&gt;value3 = Own SSID</td>
</tr>
<tr>
<td>p2p_devfound</td>
<td>value1 = GO device name&lt;br&gt;value2 = MAC&lt;br&gt;value3 = WPS</td>
</tr>
<tr>
<td>p2p_request</td>
<td>value1 = GO device name&lt;br&gt;value2 = MAC&lt;br&gt;value3 = WPS</td>
</tr>
<tr>
<td>p2p_connectfail</td>
<td>value1 = Status&lt;br&gt;- disconnected&lt;br&gt;- scaning&lt;br&gt;- connecting&lt;br&gt;- connected</td>
</tr>
</tbody>
</table>

Table 85: `+eventwlan` event
### Event:

```
+eventwlan:[ID],[value1],...,[valueX]
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>provisioning_status</td>
<td>value1=Provisioning status</td>
</tr>
<tr>
<td></td>
<td>value2=Role</td>
</tr>
<tr>
<td></td>
<td>value3=Status</td>
</tr>
<tr>
<td></td>
<td>- disconnected</td>
</tr>
<tr>
<td></td>
<td>- scanning</td>
</tr>
<tr>
<td></td>
<td>- connecting</td>
</tr>
<tr>
<td></td>
<td>- connected</td>
</tr>
<tr>
<td></td>
<td>value4=SSID</td>
</tr>
<tr>
<td>provisioning_profile_added</td>
<td>value1=Provisioning status</td>
</tr>
<tr>
<td></td>
<td>value2=SSID</td>
</tr>
</tbody>
</table>

Table 86: +eventwlan event
8.7.3. Socket events

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

<table>
<thead>
<tr>
<th>Event:</th>
<th>+eventsocket:[ID],[value1]...,[valueX]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Values</td>
</tr>
<tr>
<td>tx_failed</td>
<td>value1=SD</td>
</tr>
<tr>
<td></td>
<td>value2=Status</td>
</tr>
<tr>
<td>async_event</td>
<td>value1=SD</td>
</tr>
</tbody>
</table>
|         | value2=Type
|         | - ssl_accept                   |
|         | - rx_frag_too_big               |
|         | - other_side_close_ssl          |
|         | - connected_secured             |
|         | - wrong_root_ca                 |
|         | value4=SSID                     |
|         | value3=Value                    |
|         | value4=SSID                     |
|         | value3=Value                    |

Table 87: +eventsocket event

8.7.4. NetApp events

The NetApp event may be received in relation to network processor operation.
<table>
<thead>
<tr>
<th>Event</th>
<th>ID</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>+eventnetapp:[ID],[value1],...,[valueX]</td>
<td>ipv4_acquired</td>
<td>value1=Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value2=Gateway</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value3=DNS</td>
</tr>
<tr>
<td></td>
<td>ipv6_acquired</td>
<td>value1=Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value2=DNS</td>
</tr>
<tr>
<td></td>
<td>ip_collision</td>
<td>value1=Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value2=DHC MAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value3=Conflict MAC</td>
</tr>
<tr>
<td></td>
<td>dhcpv4_leased</td>
<td>value1=Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value2=Lease time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value3=BSSID</td>
</tr>
<tr>
<td></td>
<td>dhcpv4_released</td>
<td>value1=Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value2=BSSID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value3=Reason</td>
</tr>
<tr>
<td></td>
<td>ipv4_lost</td>
<td>value1=Status</td>
</tr>
<tr>
<td></td>
<td>dhcp_ipv4_acquire_timeout</td>
<td>value1=Status</td>
</tr>
<tr>
<td></td>
<td>ipv6_lost</td>
<td>value1=IP lost</td>
</tr>
</tbody>
</table>

Table 88: +eventnetapp event

### 8.7.5. MQTT events

The MQTT event may be received in relation to one of the MQTT operations performed by the module.
### 8.7.6. Fatal error events

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

<table>
<thead>
<tr>
<th>Event:</th>
<th>+eventfatalerror:[ID],[value1],...,[valueX]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Values</td>
</tr>
<tr>
<td>device_abort</td>
<td>value1=Code</td>
</tr>
<tr>
<td></td>
<td>value2=Value</td>
</tr>
<tr>
<td>driver_abort</td>
<td></td>
</tr>
<tr>
<td>sync_loss</td>
<td></td>
</tr>
<tr>
<td>no_cmd_ack</td>
<td>value1=Code</td>
</tr>
<tr>
<td>cmd_timeout</td>
<td>value1=Code</td>
</tr>
</tbody>
</table>

Table 90: +eventfatalerror event
9. Provisioning

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

The web pages for provisioning require JavaScript.

9.1. Start the provisioning mode

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

1. When starting the module in AT command mode the command
   
   AT+provisioningStart
   
   starts the provisioning.

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

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

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

9.2. Enter the credentials

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

Figure 10: Provisioning main page

Click on the "Settings" button to open the settings menu. Several tabs are available to read the current status of the radio module and to configure it. To save a WLAN profile in the radio module go to the "Profiles" tab and enter the credentials of the access point the radio module is supposed to connect to.
Press the button below to add the defined credential to the radio module. The radio module then prints on the UART the corresponding +eventwlan message.

+eventwlan:provisioning_profile_added,no_error,Calypso–Pruefrouter

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

AT+reboot

or press the reset button.

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

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

+eventwlan:connect,Calypso–Pruefrouter,0x0:0x25:0x9c:0xcf:0x85:0xf0
+eventnetapp:ipv4_acquired,192.168.1.101,192.168.1.50,192.168.1.50
10. Typical application use cases

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

10.1. UDP communication

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

10.1.1. Prerequisites

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

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

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

10.1.2. UDP socket communication

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

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

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

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

3. Repeat the above steps on the second module.

4. Use the AT+sendTo commands with destination port and address to send data packets.

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

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

   AT+recvFrom=0,INET,8888,192.168.1.140,0,32
   +recvFrom:0,0,32,3U0fRSk9UaYx00ABvhPU1vBH7tgnGIqW
   OK
10.2. TCP communication

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

10.3. Secure socket communication

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

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

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

10.3.1. Create an SSL/TLS server

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

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

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

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

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

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

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

4. Finally, bind the socket to a port (in this example 9999) and the local IP address and listen for connection requests.
### 10.3.2. Create an SSL/TLS client

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

1. Create a simple TCP socket with the following command. Note the socket ID for future reference.
   ```
   AT+socket=INET,STREAM,TCP
   +socket:0
   ```

2. The next step is to upgrade the socket to secure by updating the socket options.
   ```
   AT+setSockopt=0,socket,SECMETHOD,SSLV3_TLSV1_2
   OK
   ```

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

4. Now the client can perform a connect to the server.
   ```
   AT+connect=0,INET,9999,192.168.1.140
   OK
   ```

5. The `+connect` event will show up once the server has accepted the connection request as described in the next section.
   ```
   +connect:9999,192.168.1.140
   OK
   ```

### 10.3.3. Secure data transfer

1. The 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,YJaZ4yUGKRES7mE5ApBD0zrFRtq56Jt
   OK
   ```
4. Which is received by a AT+recv command in the client.

\[
\text{AT+recv}=0,0,32 \\
\text{OK}
\]

\[
+\text{recv}:0,0,32,\text{YJaZ4yUGKRES7mE5ApBDo0zrFRtq56Jt} \\
\text{OK}
\]

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

\[
\text{AT+send}=0,0,32,\text{iuwHSis5xTttzfftffhjfh678pSHJA} \\
\text{OK}
\]

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

\[
\text{AT+recv}=1,0,32 \\
\text{OK}
\]

\[
+\text{recv}:1,0,32,\text{iuwHSis5xTttzfftffhjfh678pSHJA} \\
\text{OK}
\]

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

\[
\text{AT+close}=0 \\
+\text{close}:0 \\
\text{OK}
\]

10.4. Wi-Fi direct example

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

10.4.1. Prerequisites

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

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

10.4.2. Auto connection setup

First of all, the P2P settings of both devices have to be configured. Here we use the following settings:

Connect to the first P2P device that is found.

\[
\text{AT+wlanPolicySet}=\text{connection}\!,\text{P2P},
\]

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

Set the device to P2P mode.

AT+wlanSetMode=P2P

Restart the network processor.

AT+stop=0

AT+start

And scan for P2P devices.

AT+wlanScan=0,5

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

+eventwlan: p2p_devfound, calypso, 0x98:0x84:0xe3:0xf6:0x8c:0x1,
+eventwlan: p2p_request, calypso, 0x98:0x84:0xe3:0xf6:0x8c:0x1, pbc
+eventwlan: p2p_client_added, 0x98:0x84:0xe3:0xf6:0x8c:0x1, calypso, DIRECT-GJ
+eventnetapp: dhcpv4leased, 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, 0xc8:0xdf:0x19:0x5:0x5e:0xef,
+eventwlan: p2p_request, calypso, 0xc8:0xdf:0x19:0x5:0x5e:0xef, pbc
+eventwlan: p2p_connect, DIRECT-GJ, 0xc8:0xdf:0x19:0x5:0x5e:0xef, calypso
+eventnetapp: ipv4_acquired, 10.123.45.2, 10.123.45.1, 10.123.45.1

Now a socket can be created to transmit/receive data. Please refer to the chapters 10.1 and 10.2 to do so.

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

AT+wlanDisconnect

10.4.3. Manual connection setup

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

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

AT+wlanPolicySet=P2P, group_owner, active

Set the device to P2P mode.

AT+wlanSetMode=P2P

Restart the network processor.

AT+stop=0

AT+start

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

AT+wlanPolicySet=P2P, client, passive
After both devices have been configured and the network processor has been restarted, start the scan for P2P devices.

AT+wlanScan=0,5

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

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

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

AT+wlanConnect=calypso,,P2P_PBC,,

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

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

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

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

Now a socket can be created to transmit/receive data. Please refer to the chapters 10.1 and 10.2 to do so.

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

AT+wlanDisconnect

10.5. Running a web page on the radio module

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

Code 1: Example html code

<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>
10.5.1. Load the web page files to the radio module

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

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

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

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

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

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

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

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

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

```
!
```

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

10.5.2. Accessing the web site in station mode

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

```
AT+netAppStart=HTTP_SERVER
OK
```

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

```
```
10.5.3. Accessing the web site in access point mode

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

```
AT+wlanSetMode=AP
OK
```

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

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

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

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

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

Finally restart the network processor:

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

Now connect with your PC or smart phone to the WLAN of the Calypso radio module and call the website "mywebpage.net/mytest.html" using a browser.
Please note that the radio module provides up to 4 connections in AP mode.
11. Timing parameters

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

11.1. Hard reset

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

<table>
<thead>
<tr>
<th>Description</th>
<th>Typ.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ready after reset</td>
<td>2</td>
<td>sec</td>
</tr>
</tbody>
</table>

Table 91: Start-up time

11.2. Soft reset

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

<table>
<thead>
<tr>
<th>Description</th>
<th>Typ.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ready after reboot/wake-up</td>
<td>350</td>
<td>ms</td>
</tr>
</tbody>
</table>

Table 92: Start-up after reboot

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

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

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

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

12.1. Prerequisites

1. An infrastructure AP with known SSID key for security must be active and connectable. The AP or a device inside the AP’s network must provide DHCP service to configure the connected stations. A connection to the internet is not required.

2. The module must be configured such that the credentials of the AP used for OTA are saved as Profile 0 and the connection policy is set to "AUTO" (see chapter 8.2.6).

3. The device (PC, Smartphone, ...) should be connected to the same AP and configured within the same network as the Calypso radio module. It can be any device with a browser supporting (self-)signed HTTPS content and JavaScript. It is recommended to use the Chrome browser with JavaScript enabled. The self-signed certificate triggers a security error on the browser. Please trust the certificate and proceed to the OTA website.

4. The device used for updating the radio module shall have the compressed and encrypted firmware image for the Calypso’s OTA update in its local storage. Using an unauthorized image may damage the module.

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

12.2. Update procedure

12.2.1. Start-up

Restart the module in the OTA operating mode, by setting and holding `APP_MODE_0` and `APP_MODE_1` accordingly (see chapter 5.2.1). A start-up message appears on the UART to indicate successful boot-up in OTA mode. If correctly configured, the Calypso automatically tries to connect to the AP saved as profile 0. `STATUS_IND_0` LED blinking at 1 Hz indicates WLAN connection in progress.
• In case of WLAN profile 0 being empty or no connection possible, the following message appears on the UART after a timeout of 5s. Please solve the connection issue before continuing.

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

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

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

12.2.2. Connection to the update device

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

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

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

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

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

On the OTA page click on update button followed by choose file button. A file browser opens up. Browse to the location where the update packet is stored and select the same. Click on upload file to start the update process. During update the module outputs the OTA states on the UART. In this state the $STATUS\_IND\_1$ LED blinks at 2 Hz.

Using browsers other than Chrome, the progress bar is found to not update correctly. In this case refresh the page manually.
Figure 15: OTA webpage upload

Figure 16: OTA in progress
12.2.4. Finalize the update

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

```
+eventota:info,"Received OTA_filename_20181121135643_CC3220SF_AMB5201_SerialWiFi_release.tar, len=440320"
+eventota:info,"Download complete"
```

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

![Figure 17: Finalize OTA](image.png)
13. Firmware history

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

<table>
<thead>
<tr>
<th>Description</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimpleLink SDK</td>
<td>2.20.00.10</td>
</tr>
<tr>
<td>NWP</td>
<td>3.9.0.6</td>
</tr>
<tr>
<td>MAC</td>
<td>2.0.0.0</td>
</tr>
<tr>
<td>PHY</td>
<td>2.2.0.6</td>
</tr>
<tr>
<td>ROM</td>
<td>0</td>
</tr>
</tbody>
</table>

13.1. 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.
## 13.2. Known issues

<table>
<thead>
<tr>
<th>Index</th>
<th>Details</th>
<th>Affected versions</th>
</tr>
</thead>
</table>
| KI001 | **Description:** AT+mqttSet does not work as expected.  
**Workaround:** Firmware update to ≥ v1.2.0 |
| KI002 | **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.  
**Workaround:** Firmware update to ≥ v1.2.0 |
| KI003 | **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+netappSend, AT+mqttPublish and AT+mqttSet.  
**Workaround:** Use base64 encoding or firmware update to ≥ v1.3.0 |
| KI004 | **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 A.P 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.  
**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 14). |

KI001 Affected versions ≤ 1.0.0  
KI002 Affected versions ≤ 1.1.0  
KI003 Affected versions ≤ 1.2.0  
KI004 Affected versions ≤ 1.3.0
14. Custom firmware

14.1. Custom configuration of standard firmware

The configuration of standard firmware includes adoption of the non-volatile Usersettings (see chapter 6) to customer requirements and creating a customized product on base of the standard product with a unique ordering number for a specific customer that needs this configuration.

For example if the UART baud rate shall be changed from the default value to another value. This variant will result in a customer exclusive module with a unique ordering number. This will also fix the firmware version to a specific and customer tested version and thus results in a customer exclusive module with a unique ordering number.

Further scheduled firmware updates of the standard firmware will not be applied to this variant automatically. Applying updates or further functions require a customer request and customer release procedure.

14.2. Customer specific firmware

A customer specific firmware may include "Custom configuration of standard firmware" plus additional options or functions and tasks that are customer specific and not part of the standard firmware.

Further scheduled firmware updates of the standard firmware will not be applied to this variant automatically. Applying updates or further functions require a customer request and customer release procedure.

This also results in a customer exclusive module with a unique ordering number.

An example for this level of customization are functions like host-less operation where the module will perform data generation (e.g. by reading a SPI or 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.

14.3. Customer firmware

A customer firmware is a firmware written and tested by the customer himself or a 3rd party as a customer representative specifically for the hardware platform provided by a module. This customer firmware (e.g. in form of a Intel hex file) will be implemented into the module’s production process at our production side.

This also results in a customer exclusive module with a unique ordering number.

The additional information needed for this type of customer firmware, such as hardware specific details and details towards the development of such firmware are not available for the public and can only be made available to qualified customers.
The qualification(s) and certification(s) of the standard firmware cannot be applied to this customer firmware solution without a review and verification.

14.4. Contact for firmware requests

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

15.1. Advice for schematic and layout

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

- A clean, stable power supply is strongly recommended. Interference, especially oscillation can severely restrain range and conformity.

- Variations in voltage level should be avoided.

- LDOs, properly designed in, usually deliver a proper regulated voltage.

- Blocking capacitors and a ferrite bead in the power supply line can be included to filter and smoothen the supply voltage when necessary.

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

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

- Elements for ESD protection should be placed on all pins that are accessible from the outside and should be placed close to the accessible area. For example, the RF-pin is accessible when using an external antenna and should be protected.

- ESD protection for the antenna connection must be chosen such as to have a minimum effect on the RF signal. For example, a protection diode with low capacitance such as the LXES15AAA1-100 or a 68 nH air-core coil connecting the RF-line to ground give good results.

- Placeholders for optional antenna matching or additional filtering are recommended.

- The antenna path should be kept as short as possible.
Again, no fixed values can be recommended, as they depend on the influencing circumstances of the application (antenna, interferences etc.).

![Figure 18: Layout](image)

- To avoid the risk of short circuits and interference there should be no routing underneath the module on the top layer of the baseboard.

- On the second layer, a ground plane is recommended, to provide good grounding and shielding to any following layers and application environment.

- In case of integrated antennas it is required to have areas free from ground. This area should be copied from the evaluation board.

- The area with the integrated antenna must overlap with the carrier board and should not protrude, as it is matched to sitting directly on top of a PCB.

- Modules with integrated antennas should be placed with the antenna at the edge of the main board. It should not be placed in the middle of the main board or far away from the edge. This is to avoid tracks beside the antenna.

- Filter and blocking capacitors should be placed directly in the tracks without stubs, to achieve the best effect.

- Antenna matching elements should be placed close to the antenna / connector, blocking capacitors close to the module.

- Ground connections for the module and the capacitors should be kept as short as possible and with at least one separate through hole connection to the ground layer.

- ESD protection elements should be placed as close as possible to the exposed areas.
15.2. Dimensioning of the micro strip antenna line

The antenna track has to be designed as a 50Ω feed line. The width W for a micro strip can be calculated using the following equation:

\[
W = 1.25 \times \left( \frac{5.98 \times H}{\epsilon_r \sqrt{\epsilon_r + 1.41}} - T_{\text{met}} \right)
\]  

(1)

Example:
A FR4 material with \( \epsilon_r = 4.3 \), a height \( H = 1000 \ \mu m \) and a copper thickness of \( T_{\text{met}} = 18 \ \mu m \)
will lead to a trace width of $W \sim 1.9$ mm. To ease the calculation of the micro strip line (or e.g. a coplanar) many calculators can be found in the internet.

- As rule of thumb a distance of about $3 \times W$ 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.

### 15.3. Antenna solutions

There exist several kinds of antennas, which are optimized for different needs. Chip antennas are optimized for minimal size requirements but at the expense of range, PCB antennas are optimized for minimal costs, and are generally a compromise between size and range. Both usually fit inside a housing.

Range optimization in general is at the expense of space. Antennas that are bigger in size, so that they would probably not fit in a small housing, are usually equipped with a RF connector. A benefit of this connector may be to use it to lead the RF signal through a metal plate (e.g. metal housing, cabinet).

As a rule of thumb a minimum distance of $\lambda/10$ (which is 3.5 cm @ 868 MHz and 1.2 cm @ 2.44 GHz) from the antenna to any other metal should be kept. Metal placed further away will not directly influence the behavior of the antenna, but will anyway produce shadowing.

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

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

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

#### 15.3.1. Wire antenna

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

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

15.3.3. PCB antenna

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

15.3.4.1. 2600130011 - Helike - 169 MHz dipole antenna

Figure 21: 169 MHz dipole-antenna

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency range [MHz]</td>
<td>169</td>
</tr>
<tr>
<td>Impedance [Ω]</td>
<td>50</td>
</tr>
<tr>
<td>VSWR</td>
<td>( \leq 2.1 )</td>
</tr>
<tr>
<td>Gain [dBi]</td>
<td>1</td>
</tr>
<tr>
<td>Dimensions (L x d) [mm]</td>
<td>320 x 15</td>
</tr>
<tr>
<td>Weight [g]</td>
<td>42</td>
</tr>
<tr>
<td>Connector</td>
<td>SMA plug</td>
</tr>
<tr>
<td>Operating Temp. [°C]</td>
<td>-40 – +85</td>
</tr>
</tbody>
</table>

This antenna requires a ground plane which will influence the electrical parameters.
15.3.4.2. 2600130041 - Herse - 434 MHz dipole antenna

![Image of 434 MHz dipole-antenna](image)

Figure 22: 434 MHz dipole-antenna

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency range [MHz]</td>
<td>433</td>
</tr>
<tr>
<td>Impedance [Ω]</td>
<td>50</td>
</tr>
<tr>
<td>VSWR</td>
<td>≤ 1.5</td>
</tr>
<tr>
<td>Polarization</td>
<td>Vertical</td>
</tr>
<tr>
<td>Radiation</td>
<td>Omni</td>
</tr>
<tr>
<td>Gain [dBi]</td>
<td>0</td>
</tr>
<tr>
<td>Antenna Cover</td>
<td>TPEE</td>
</tr>
<tr>
<td>Dimensions (L x d) [mm]</td>
<td>90 x 12</td>
</tr>
<tr>
<td>Weight [g]</td>
<td>9.6</td>
</tr>
<tr>
<td>Connector</td>
<td>SMA plug</td>
</tr>
<tr>
<td>Operating Temp. [°C]</td>
<td>-40 – +80</td>
</tr>
</tbody>
</table>

This antenna requires a ground plane which will influence the electrical parameters.
15.3.4.3. 2600130081 - Hyperion-I - 868 MHz dipole antenna

Figure 23: 868 MHz dipole-antenna

Ideally suited for applications where no ground plane is available.

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

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center frequency [MHz]</td>
<td>868</td>
</tr>
<tr>
<td>Frequency range [MHz]</td>
<td>853 – 883</td>
</tr>
<tr>
<td>Wavelength</td>
<td>0.5 wave</td>
</tr>
<tr>
<td>VSWR</td>
<td>≤ 2.0</td>
</tr>
<tr>
<td>Impedance [Ω]</td>
<td>50</td>
</tr>
<tr>
<td>Connector</td>
<td>SMA (Male)</td>
</tr>
<tr>
<td>Dimensions (L x d) [mm]</td>
<td>142 x 10</td>
</tr>
<tr>
<td>Peak gain [dBi]</td>
<td>-2.3</td>
</tr>
<tr>
<td>Operating temp. [°C]</td>
<td>-30 – +80</td>
</tr>
</tbody>
</table>
15.3.4.4. 2600130082 - Hyperion-II - 868 MHz magnetic base antenna

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

![Hyperion-II - 868 MHz magnetic base antenna](image)

**Figure 24: 868 MHz magnet foot antenna with 1.5 m antenna cable**

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

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency range [MHz]</td>
<td>824 – 894</td>
</tr>
<tr>
<td>VSWR</td>
<td>$\leq$ 2.0</td>
</tr>
<tr>
<td>Polarisation</td>
<td>Vertical</td>
</tr>
<tr>
<td>Impedance [Ω]</td>
<td>50±5</td>
</tr>
<tr>
<td>Connector</td>
<td>SMA (Male)</td>
</tr>
<tr>
<td>Dimensions (L x d) [mm]</td>
<td>89.8 x 27</td>
</tr>
<tr>
<td>Weight [g]</td>
<td>50±5</td>
</tr>
<tr>
<td>Operating temp. [°C]</td>
<td>-30 – +60</td>
</tr>
</tbody>
</table>
15.3.4.5. 2600130021 - Himalia - 2.4 GHz 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.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency range [GHz]</td>
<td>2.4 – 2.5</td>
</tr>
<tr>
<td>Impedance [Ω]</td>
<td>50</td>
</tr>
<tr>
<td>VSWR</td>
<td>≤ 2:1</td>
</tr>
<tr>
<td>Polarization</td>
<td>Linear</td>
</tr>
<tr>
<td>Radiation</td>
<td>Omni-Directional</td>
</tr>
<tr>
<td>Peak Gain [dBi]</td>
<td>2.8</td>
</tr>
<tr>
<td>Average Gain [dBi]</td>
<td>-0.6</td>
</tr>
<tr>
<td>Efficiency</td>
<td>85 %</td>
</tr>
<tr>
<td>Dimensions (L x d) [mm]</td>
<td>83.1 x 10</td>
</tr>
<tr>
<td>Weight [g]</td>
<td>7.4</td>
</tr>
<tr>
<td>Connector</td>
<td>SMA plug</td>
</tr>
<tr>
<td>Operating temp. [°C]</td>
<td>-40 – +80</td>
</tr>
</tbody>
</table>

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 dipol antenna in a closed housing, so that only through professional installation it is possible to remove it.
16. Reference design

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

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

Figure 26: Reference design: Schematic, most important parts
Figure 27: Reference design: Layout
16.2. Design Guide for FCC ID R7T1001102

Figure 28: Close-up: Layout

<table>
<thead>
<tr>
<th>Nr</th>
<th>Copper</th>
<th>Isolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.018mm</td>
<td>0.36mm</td>
</tr>
<tr>
<td>2</td>
<td>0.035mm</td>
<td>0.71mm</td>
</tr>
<tr>
<td>3</td>
<td>0.035mm</td>
<td>0.36mm</td>
</tr>
<tr>
<td>16</td>
<td>0.018mm</td>
<td></td>
</tr>
</tbody>
</table>

Gesamt: 1.536mm

Figure 29: 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 into many or to small parts.

• Bottom layer is used for routing.

Figure 30: 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 15.3.4.5 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.
16.3. Application mode pins

The pins `APP_MODE_0` and `APP_MODE_1` define at boot time which application mode is used during operation of the module (see chapter 5.2.1). The OTA mode enable 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.
17. Manufacturing information

17.1. Moisture sensitivity level

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

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

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

17.2. Soldering

17.2.1. Reflow soldering

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

<table>
<thead>
<tr>
<th>Profile feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preheat temperature Min</td>
<td>$T_{S_{\text{Min}}}$</td>
</tr>
<tr>
<td>Preheat temperature Max</td>
<td>$T_{S_{\text{Max}}}$</td>
</tr>
<tr>
<td>Preheat time from $T_{S_{\text{Min}}}$ to $T_{S_{\text{Max}}}$</td>
<td>$t_S$</td>
</tr>
<tr>
<td>Ramp-up rate ($T_L$ to $T_P$)</td>
<td></td>
</tr>
<tr>
<td>Liquidous temperature</td>
<td>$T_L$</td>
</tr>
<tr>
<td>Time $t_L$ maintained above $T_L$</td>
<td>$t_L$</td>
</tr>
<tr>
<td>Peak package body temperature</td>
<td>$T_P$</td>
</tr>
<tr>
<td>Time within 5°C of actual peak temperature</td>
<td>$t_P$</td>
</tr>
<tr>
<td>Ramp-down Rate ($T_P$ to $T_L$)</td>
<td></td>
</tr>
<tr>
<td>Time 20°C to $T_P$</td>
<td></td>
</tr>
</tbody>
</table>

Table 93: Classification reflow soldering profile, Note: refer to IPC/JEDEC J-STD-020E
### Table 94: Package classification reflow temperature, PB-free assembly, Note: refer to IPC-/JEDEC J-STD-020E

<table>
<thead>
<tr>
<th>Package thickness</th>
<th>Volume mm³</th>
<th>Volume mm³</th>
<th>Volume mm³</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 1.6mm</td>
<td>260 °C</td>
<td>260 °C</td>
<td>260 °C</td>
</tr>
<tr>
<td>1.6mm - 2.5mm</td>
<td>260 °C</td>
<td>250 °C</td>
<td>245 °C</td>
</tr>
<tr>
<td>&gt; 2.5mm</td>
<td>250 °C</td>
<td>245 °C</td>
<td>245 °C</td>
</tr>
</tbody>
</table>

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 31: Reflow soldering profile](Image)

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

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

- Do not clean the product with water. Capillary effects can draw water into the gap between the host PCB and the module, absorbing water underneath it. If water is trapped inside, it may short-circuit adjoining pads. The water may also destroy the label and ink-jet printed text on it.

- Cleaning processes using alcohol or other organic solvents may draw solder flux residues into the housing, which won’t be detected in a post-wash inspection. The solvent may also destroy the label and ink-jet printed text on it.

- Do not use ultrasonic cleaning as it will permanently damage the part, particularly the crystal oscillators.

17.2.3. Other notations

- Conformal coating of the product will result in the loss of warranty. The RF shields will not protect the part from low-viscosity coatings.

- Do not attempt to improve the grounding by forming metal strips directly to the EMI covers or soldering on ground cables, as it may damage the part and will void the warranty.

- Always solder every pad to the host PCB even if some are unused, to improve the mechanical strength of the module.

- The part is sensitive to ultrasonic waves, as such do not use ultrasonic cleaning, welding or other processing. Any ultrasonic processing will void the warranty.

17.3. ESD handling

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

- the first contact point when handling the PCB is always between the local GND and the host PCB GND, unless there is a galvanic coupling between the local GND (for example work table) and the host PCB GND.

- Before assembling an antenna patch, connect the grounds.

- While handling the RF pin, avoid contact with any charged capacitors and be careful when contacting any materials that can develop charges (for example coaxial cable with around 50-80 pF/m, patch antenna with around 10 pF, soldering iron etc.)
• Do not touch any exposed area of the antenna to avoid electrostatic discharge. Do not let the antenna area be touched in a non ESD-safe manner.

• When soldering, use an ESD-safe soldering iron.

17.4. Safety recommendations

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

• Use in areas with risk of explosion (for example oil refineries, gas stations).

• Use in areas such as airports, aircraft, hospitals, etc., where the product may interfere with other electronic components.

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

• Compliance with the instructions in the product manual is recommended for correct product set-up.

• The product must be provided with a consolidated voltage source. The wiring must meet all applicable fire and security prevention standards.

• Handle with care. Avoid touching the pins as there could be ESD damage.

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

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

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

18.1. Dimensions

<table>
<thead>
<tr>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>19 * 27.5 * 3 mm</td>
</tr>
</tbody>
</table>

Table 95: Dimensions

18.2. Weight

<table>
<thead>
<tr>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 g</td>
</tr>
</tbody>
</table>

Table 96: Weight
18.3. Module drawing

Figure 32: Module dimensions [mm]
18.4. Footprint

Figure 33: Footprint and dimensions [mm]
18.5. Antenna free area

To avoid influence and mismatching of the antenna the recommended free area around the antenna should be maintained. As rule of thumb a minimum distance of metal parts to the antenna of $\lambda/10$ should be kept (see figure 33). Even though metal parts would influence the characteristic of the antenna, but the direct influence and matching keep an acceptable level.
19. Marking

19.1. Lot number

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

![Lot number structure diagram](image)

Figure 34: Lot number structure

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

Table 97: Lot number details

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

The module labels may include the following fields:

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

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

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

For FCC compliant integration the label requirements described in chapter 21.5 have to be considered.

19.2.1. Label of Calypso

```
2610011025000
1001102
FCC ID: R7T1001102
IC: 5136A-1001102
SN: 129000328
```
20. Information for Ex protection

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

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

21.1. Important notice EU

The use of RF frequencies is limited by national regulations. The Calypso has been designed to comply with the R&TTE directive 1999/5/EC and the RED directive 2014/53/EU of the European Union (EU).

The Calypso can be operated without notification and free of charge in the area of the European Union. However, according to the R&TTE / RED directive, restrictions (e.g. in terms of duty cycle or maximum allowed RF power) may apply.
21.2. EU Declaration of conformity

EU DECLARATION OF CONFORMANCE

Radio equipment: Calypso / 2610011025000

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

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

Object of the declaration: Calypso / 2610011025000

The object of the declaration described above is in conformity with the relevant Union harmonization legislation: Directive 2014/53/EU and 2011/65/EU.

Following harmonized norms or technical specifications have been applied:

- EN 300 328 V2.1.1 (2016-11)
- EN 301 489-1 V2.2.0 (Draft)
- EN 301 489-17 V3.2.0 (Draft)
- EN 62479 : 2010

Trier, 31th of October 2018

Place and date of issue
21.3. FCC Compliance Statement

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.

21.4. IC Compliance Statement

Certification Number: 5136A-1001102
PMN: 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.

21.5. 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).

21.6. Pre-certified antennas

The Calypso is pre-certified with the following antennas.

<table>
<thead>
<tr>
<th>Product</th>
<th>Certified antenna</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calypso</td>
<td>PCB antenna included in the Calypso</td>
</tr>
<tr>
<td>Calypso-1</td>
<td>Dipole antenna as specified in chapter 15.3.4.5</td>
</tr>
</tbody>
</table>

It is only possible to connect an antenna by soldering. It is mandatory to follow chapter 16.2 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.
22. Important notes

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

22.1. General customer responsibility

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

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

It has to be clearly pointed out that the possibility of a malfunction of electronic components or failure before the end of the usual lifetime cannot be completely eliminated in the current state of the art, even if the products are operated within the range of the specifications. The same statement is valid for all software sourcecode 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.

22.3. Best care and attention

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

22.4. Customer support for product specifications

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

Due to constant product improvement, product specifications may change from time to time. As a standard reporting procedure of the Product Change Notification (PCN) according to the JEDEC-Standard, we inform about major changes. In case of further queries regarding the PCN, the field sales engineer, the internal sales person or the technical support team in charge should be contacted. The basic responsibility of the customer as per section 22.1 and 22.2 remains unaffected. All wireless connectivity module driver software “wireless connectivity SDK” and it’s source codes as well as all PC software tools are not subject to the Product Change Notification information process.

22.6. Product life cycle

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

22.7. Property rights

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

22.8. General terms and conditions

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

23.1. Exclusion of liability

Würth Elektronik eiSos GmbH & Co. KG considers the information in this document to be correct at the time of publication. However, Würth Elektronik eiSos GmbH & Co. KG reserves the right to modify the information such as technical specifications or functions of its products or discontinue the production of these products or the support of one of these products without any written announcement or notification to customers. The customer must make sure that the information used corresponds to the latest published information. Würth Elektronik eiSos GmbH & Co. KG does not assume any liability for the use of its products. Würth Elektronik eiSos GmbH & Co. KG does not grant licenses for its patent rights or for any other of its intellectual property rights or third-party rights.

Notwithstanding anything above, Würth Elektronik eiSos GmbH & Co. KG makes no representations and/or warranties of any kind for the provided information related to their accuracy, correctness, completeness, usage of the products and/or usability for customer applications. Information published by Würth Elektronik eiSos GmbH & Co. KG regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof.

23.2. Suitability in customer applications

The customer bears the responsibility for compliance of systems or units, in which Würth Elektronik eiSos GmbH & Co. KG products are integrated, with applicable legal regulations. Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of Würth Elektronik eiSos GmbH & Co. KG components in its applications, notwithstanding any applications-related 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.

23.3. Trademarks

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

23.4. Usage restriction

Würth Elektronik eiSos GmbH & Co. KG products have been designed and developed for usage in general electronic equipment only. This product is not authorized for use in equipment where a higher safety standard and reliability standard is especially required or where
a failure of the product is reasonably expected to cause severe personal injury or death, unless the parties have executed an agreement specifically governing such use. Moreover, Würth Elektronik eiSos GmbH & Co. KG products are neither designed nor intended for use in areas such as military, aerospace, aviation, nuclear control, submarine, transportation (automotive control, train control, ship control), transportation signal, disaster prevention, medical, public information network etc. Würth Elektronik eiSos GmbH & Co. KG must be informed about the intent of such usage before the design-in stage. In addition, sufficient reliability evaluation checks for safety must be performed on every electronic component, which is used in electrical circuits that require high safety and reliability function or performance. By using Würth Elektronik eiSos GmbH & Co. KG products, the customer agrees to these terms and conditions.
24. License terms

This 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 this license terms is 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 this license terms. You agree to comply with all provisions under this license terms.

24.1. 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 this 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 this 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 this 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.

24.2. Usage and obligations

The responsibility for the applicability and use of the Würth Elektronik eiSos wireless connectivity product with the incorporated Firmware in a particular customer design is always solely within the authority of the customer. Due to this fact, it is up to you to evaluate and investigate, where appropriate, and to decide whether the device with the specific product characteristics described in the product specification is valid and suitable for your respective application or not.

You are responsible for using the Würth Elektronik eiSos 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.

24.3. Ownership

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

24.4. Firmware update(s)

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

24.5. Disclaimer of warranty

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

Any liability not expressly provided by Würth Elektronik eiSos shall be disclaimed. You agree to hold us harmless from any third-party claims related to your usage of the Würth Elektronik eiSos’ products with the incorporated Firmware, 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.

24.7. Applicable law and jurisdiction

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

24.8. Severability clause

If a provision of this license 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.

24.9. Miscellaneous

Würth Elektronik eiSos reserves the right at any time to change this 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 wireless connectivity product, you accept this license terms in all terms.
A. Wi-Fi certificate

The section contains the Wi-Fi certificate for Calypso.

<table>
<thead>
<tr>
<th>Certification ID: WFA81685</th>
<th>Page 1 of 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of Last Certification</td>
<td>January 22, 2019</td>
</tr>
<tr>
<td>Company</td>
<td>Wurth Elektron eiSos GmbH &amp; CO. KG</td>
</tr>
<tr>
<td>Product</td>
<td>Calypso</td>
</tr>
<tr>
<td>Model Number</td>
<td>261001102500x</td>
</tr>
<tr>
<td>Product Identifier(s)</td>
<td>AMB5201 (Other)</td>
</tr>
<tr>
<td>Category</td>
<td>Other</td>
</tr>
<tr>
<td>Subcategory</td>
<td>Industrial (communications &amp; input)</td>
</tr>
<tr>
<td>Hardware Version</td>
<td>Product: 2.1, Wi-Fi Component: 3.1</td>
</tr>
<tr>
<td>Firmware Version</td>
<td>Product: 1.0.0, Wi-Fi Component: 31.2.0.0.0</td>
</tr>
<tr>
<td>Operating System</td>
<td>ThreadX, version: 2.20.00.10</td>
</tr>
<tr>
<td>Frequency Band(s)</td>
<td>2.4 GHz</td>
</tr>
</tbody>
</table>

Summary of Certifications

<table>
<thead>
<tr>
<th>CLASSIFICATION</th>
<th>PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectivity</td>
<td>Wi-Fi CERTIFIED™ b, g, n</td>
</tr>
<tr>
<td></td>
<td>WPA™ – Enterprise, Personal</td>
</tr>
<tr>
<td></td>
<td>WPA2™ – Enterprise, Personal</td>
</tr>
<tr>
<td></td>
<td>Wi-Fi Direct®</td>
</tr>
<tr>
<td>Optimization</td>
<td>WMM®</td>
</tr>
<tr>
<td>Access</td>
<td>Wi-Fi Protected Setup™</td>
</tr>
</tbody>
</table>
**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

```c
/* 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_DISASSOC_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_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. Socket error codes

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

B.3. Secure socket error codes

/* ssl tls security start with −300 offset */
SL_ERROR_BSD_ESEC_CLOSE_NOTIFY (−300L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_UNEXPECTED_MESSAGE (−310L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_BAD_RECORD_MAC (−320L) /* ssl/tls alerts */
SL_ERROR_BSD_ESEC_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_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_CANCELED (-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_BADCERTIFICATE_HASH_VALUE (-414L) /* ssl/tls alerts */
/* proprietary secure */
SL_ERROR_BSD_ESECGENERAL (-450L) /* error secure level general error */
SL_ERROR_BSD_ESECDECRYPT (-451L) /* error secure level, decrypt recv packet fail */
SL_ERROR_BSD_ESECCLOSED (-452L) /* secure layer is closed by other size, tcp is still connected */
SL_ERROR_BSD_ESECNOVERIFY (-453L) /* Connected without server verification */
SL_ERROR_BSD_ESECNOCAFILE (-454L) /* error secure level CA file not found */
SL_ERROR_BSD_ESECDECLOSED (-455L) /* error secure level, No memory space available */
SL_ERROR_BSD_ESECBADCAFILE (-456L) /* error secure level bad CA file */
SL_ERROR_BSD_ESECBADCERTFILE (-457L) /* error secure level bad Certificate file */
SL_ERROR_BSD_ESECBADPRIVATEFILE (-458L) /* error secure level bad private file */
SL_ERROR_BSD_ESECBADDHFILE (-459L) /* error secure level bad DH file */
SL_ERROR_BSD_ESECINSECUNTOMANYSSLOPENED (-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_ESECINSECUTESNOTALLOWEDWHENLISTENING (-463L) /* the operation is not allowed when listening, do before listen */
SL_ERROR_BSD_ESECHEADERREVOKED (-464L) /* connected but on of the certificates in the chain is revoked */
SL_ERROR_BSD_ESECUNKNOWNROOTCA (-465L) /* connected but the root CA used to validate the peer is unknown */
SL_ERROR_BSD_ESECWONGPEERELECT (-466L) /* wrong peer cert (server cert) was received while trying to connect to server */
SL_ERROR_BSD_ESECADMINDISCONNECTEDUNCOMPLETERECORD (-467L) /* the other side disconnected the TCP layer and didn’t send the whole ssl record */
SL_ERROR_BSD_ESEC_BUFFER_E (-632L) /* output buffer too small or input too large */
SL_ERROR_BSD_ESEC_ALGO_ID_E (-633L) /* setting algo id error */
SL_ERROR_BSD_ESEC_PUBLIC_KEY_E (-634L) /* setting public key error */
SL_ERROR_BSD_ESEC_DATE_E (-635L) /* setting date validity error */
SL_ERROR_BSD_ESEC_SUBJECT_E (-636L) /* setting subject name error */
SL_ERROR_BSD_ESEC_ISSUER_E (-637L) /* setting issuer name error */
SL_ERROR_BSD_ESEC_CA_TRUE_E (-638L) /* setting CA basic constraint true error */
SL_ERROR_BSD_ESEC_EXTENSIONS_E (-639L) /* setting extensions error */
SL_ERROR_BSD_ESEC_ASN_PARSE_E (-640L) /* ASN parsing error, invalid input */
SL_ERROR_BSD_ESEC_ASN_VERSION_E (-641L) /* ASN version error, invalid number */
SL_ERROR_BSD_ESEC_ASN_GETINT_E (-642L) /* ASN get big int error, invalid data */
SL_ERROR_BSD_ESEC ASN RSA KEY E (−643L) /* ASN key init error, invalid input */
SL_ERROR_BSD_ESEC ASN OBJECT ID E (−644L) /* ASN object id error, invalid id */
SL_ERROR_BSD_ESEC ASN TAG NULL E (−645L) /* ASN tag error, not null */
SL_ERROR_BSD_ESEC ASN EXPECT 0 E (−646L) /* ASN expect error, not zero */
SL_ERROR_BSD_ESEC ASN BITSTR E (−647L) /* ASN bit string error, wrong id */
SL_ERROR_BSD_ESEC ASN UNKNOWN OID E (−648L) /* ASN oid error, unknown sum id */
SL_ERROR_BSD_ESEC ASN DATE SZ E (−649L) /* ASN date error, bad size */
SL_ERROR_BSD_ESEC ASN BEFORE_DATE E (−650L) /* ASN date error, current date before */
SL_ERROR_BSD_ESEC ASN AFTER_DATE E (−651L) /* ASN date error, current date after */
SL_ERROR_BSD_ESEC ASN SIG OID E (−652L) /* ASN signature error, mismatched oid */
SL_ERROR_BSD_ESEC ASN TIME E (−653L) /* ASN time error, unknown time type */
SL_ERROR_BSD_ESEC ASN INPUT E (−654L) /* ASN input error, not enough data */
SL_ERROR_BSD_ESEC ASN SIG CONFIRM E (−655L) /* ASN sig error, confirm failure */
SL_ERROR_BSD_ESEC ASN SIG HASH E (−656L) /* ASN sig error, unsupported hash type */
SL_ERROR_BSD_ESEC ASN SIG KEY E (−657L) /* ASN sig error, unsupported key type */
SL_ERROR_BSD_ESEC ASN DH KEY E (−658L) /* ASN key init error, invalid input */
SL_ERROR_BSD_ESEC ASN NTRU KEY E (−659L) /* ASN ntru key decode error, invalid input */
SL_ERROR_BSD_ESEC ASN CRIT EXT E (−660L) /* ASN unsupported critical extension */
SL_ERROR_BSD_ESEC ECC BAD ARG E (−670L) /* ECC input argument of wrong type */
SL_ERROR_BSD_ESEC 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 DECRIPT 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 */
B.4. WLAN error codes

/* WLAN ERRORS CODES*/
SL_ERROR_WLAN_KEY_ERROR (−2049L)
SL_ERROR_WLAN_INVALID_ROLE (−2050L)
SL_ERROR_WLAN_PREFERRED_NETWORKS_FILE_LOAD_FAILED (−2051L)
SL_ERROR_WLAN_CANNOT_CONFIG_SCAN_DURING_PROVISIONING (−2052L)
SL_ERROR_WLAN_INVALID_SECURITY_TYPE (−2054L)
SL_ERROR_WLAN_PASSPHRASE_TOO_LONG (−2055L)
SL_ERROR_WLAN_EAP_WRONG_METHOD (−2057L)
SL_ERROR_WLAN_PASSWORD_ERROR (−2058L)
SL_ERROR_WLAN_EAP_ANONYMOUS_LEN_ERROR (−2059L)
SL_ERROR_WLAN_SSID_LEN_ERROR (−2060L)
SL_ERROR_WLAN_USER_ID_LEN_ERROR (−2061L)
SL_ERROR_WLAN_PREFERRED_NETWORK_LIST_FULL (−2062L)
SL_ERROR_WLAN_PREFERRED_NETWORKS_FILE_WRITE_FAILED (−2063L)
SL_ERROR_WLAN_ILLEGAL_WEP_KEY_INDEX (−2064L)
SL_ERROR_WLAN_INVALID_DWELL_TIME_VALUES (−2065L)
SL_ERROR_WLAN_INVALID_POLICY_TYPE (−2066L)
SL_ERROR_WLAN_PM_POLICY_INVALID_OPTION (−2067L)
SL_ERROR_WLAN_PM_POLICY_INVALID_PARAMS (−2068L)
SL_ERROR_WLAN_PREFERRED_NETWORKS_FILE_WRITE_FAILED (−2069L)
SL_ERROR_WLAN_ILLEGAL_CHANNEL (−2070L)
SL_ERROR_WLAN_WIFI_ALREADY_DISCONNECTED (−2071L)
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL_ERROR_WLAN_TRANSCEIVER_ENABLED (−2072L)</td>
<td>wlan_transceiver_enabled error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_GET_NETWORK_LIST_EAGAIN (−2073L)</td>
<td>wlan_get_network_list_eagain error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_GET_PROFILE_INVALID_INDEX (−2074L)</td>
<td>wlan_get_profile_invalid_index error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_FAST_CONN_DATA_INVALID (−2075L)</td>
<td>wlan_fast_conn_data_invalid error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_NO_FREE_PROFILE (−2076L)</td>
<td>wlan_no_free_profile error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_AP_SCAN_INTERVAL_TOO_LOW (−2077L)</td>
<td>wlan_ap_scan_interval_too_low error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_SCAN_POLICY_INVALID_PARAMS (−2078L)</td>
<td>wlan_scan_policy_invalid_params error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_INVALID_COUNTRY_CODE (−2164L)</td>
<td>wlan_invalid_country_code error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_NVMEM_ACCESS_FAILED (−2165L)</td>
<td>wlan_nvmem_access_failed error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_OLD_FILE_VERSION (−2166L)</td>
<td>wlan_old_file_version error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_TX_POWER_OUT_OF_RANGE (−2167L)</td>
<td>wlan_tx_power_out_of_range error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_INVALID_AP_PASSWORD_LENGTH (−2168L)</td>
<td>wlan_invalid_ap_password_length error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_PROVISIONING_ABORT_PROVISIONING_ALREADY_STARTED (−2169L)</td>
<td>wlan_provisioning_abort_provisioning_already_started error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_PROVISIONING_ABORT_HTTP_SERVER_DISABLED (−2170L)</td>
<td>wlan_provisioning_abort_http_server_disabled error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_PROVISIONING_ABORT_PROFILE_LIST_FULL (−2171L)</td>
<td>wlan_provisioning_abort_profile_list_full error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_PROVISIONING_ABORT_INVALID_PARAM (−2172L)</td>
<td>wlan_provisioning_abort_invalid_param error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_PROVISIONING_ABORT_GENERAL_ERROR (−2173L)</td>
<td>wlan_provisioning_abort_general_error error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_MULTICAST_EXCEED_MAX_ADDR (−2174L)</td>
<td>wlan_multicast_exceed_max_addr error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_MULTICAST_INVAL_ADDR (−2175L)</td>
<td>wlan_multicast_inval_addr error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_AP_SCAN_INTERVAL_TOO_SHORT (−2176L)</td>
<td>wlan_ap_scan_interval_too_short error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_PROVISIONING_CMD_NOTEXPECTED (−2177L)</td>
<td>wlan_provisioning_cmd_not_expected error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_AP_ACCESS_LIST_NO_ADDRESS_TO_DELETE (−2178L)</td>
<td>wlan_ap_access_list_no_address_to_delete error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_AP_ACCESS_LIST_FULL (−2179L)</td>
<td>wlan_ap_access_list_full error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_AP_ACCESS_LIST_DISABLED (−2180L)</td>
<td>wlan_ap_access_list_disabled error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_AP_ACCESS_LIST_MODE_NOT_SUPPORTED (−2181L)</td>
<td>wlan_ap_access_list_mode_not_supported error</td>
</tr>
<tr>
<td>SL_ERROR_WLAN_AP_STA_NOT_FOUND (−2182L)</td>
<td>wlan_ap_sta_not_found error</td>
</tr>
</tbody>
</table>

**B.5. Device error codes**

```c
/* DEVICE ERRORS CODES*/
SL_ERROR_SUPPLICANT_ERROR (−4097L)
SL_ERROR_HOSTAPD_INIT_FAIL (−4098L)
SL_ERROR_HOSTAPD_INIT_IF_FAIL (−4099L)
SL_ERROR_WLAN_DRV_INIT_FAIL (−4100L)
SL_ERROR_FS_FILE_TABLE_LOAD_FAILED (−4102L) /* init file system failed */
SL_ERROR_MDNS_ENABLE_FAIL (−4103L) /* mdns enable failed */
SL_ERROR_ROLE_STA_ERR (−4107L) /* Failure to load MAC/PHY in STA role */
SL_ERROR_ROLE_AP_ERR (−4108L) /* Failure to load MAC/PHY in AP role */
SL_ERROR_ROLE_P2P_ERR (−4109L) /* Failure to load MAC/PHY in P2P role */
SL_ERROR_CALIB_FAIL (−4110L) /* Failure of calibration */
SL_ERROR_FS_CORRUPTED_ERR (−4111L) /* FS is corrupted, Return to Factory Image or Program new image should be invoked (see sl_FsCtl, sl_FsProgram) */
SL_ERROR_FS_ALERT_ERR (−4112L) /* Device is locked, Return to Factory Image or Program new image should be invoked (see sl_FsCtl, sl_FsProgram) */
SL_ERROR_RESTORE_IMAGE_COMPLETE (−4113L) /* Return to factory image completed, perform reset */
SL_ERROR_UNKNOWN_ERR (−4114L)
SL_ERROR_GENERAL_ERR (−4115L) /* General error during init */
SL_ERROR_WRONG_ROLE (−4116L)
SL_ERROR_INCOMPLETE_PROGRAMMING (−4117L) /* Error during programming, Program new image should be invoked (see sl_FsProgram) */
```
SL_ERROR_PENDING_TXRX_STOP_TIMEOUT_EXP (-4118L) /* Timeout expired before completing all TX/RX */
SL_ERROR_PENDING_TXRX_NO_TIMEOUT (-4119L) /* No Timeout, still have pending TX/RX */
SL_ERROR_INVALID_PERSISTENT_CONFIGURATION (-4120L) /* persistency configuration can only be set to 0 (disabled) or 1 (enabled) */

B.6. Network config error codes

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

B.7. File System error codes

/* FS ERRORS CODES */
SL_FS_OK (0L)
SL_ERROR_FS_EXTRACTION_WILL_START_AFTER_RESET (-10241L)
SL_ERROR_FS_NO_CERTIFICATE_STORE (-10242L)
SL_ERROR_FS_IMAGE_SHOULD_BE_AUTHENTICATE (-10243L)
SL_ERROR_FS_IMAGE_SHOULD_BE_ENCRYPTED (-10244L)
SL_ERROR_FS_IMAGE_CANT_BE_ENCRYPTED (-10245L)
SL_ERROR_FS_DEVELOPMENT_BOARD_WRONG_MAC (-10246L)
SL_ERROR_FS_DEVICE_NOT_SECURED (-10247L)
SL_ERROR_FS_SYSTEM_FILE_ACCESS_DENIED (-10248L)
SL_ERROR_FS_IMAGE_EXTRACT_EXPECTING_USER_KEY (-10249L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_CLOSE_FILE (-10250L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_WRITE_FILE (-10251L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_OPEN_FILE (-10252L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_GET_IMAGE_HEADER (-10253L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_GET_IMAGE_INFO (-10254L)
SL_ERROR_FS_IMAGE_EXTRACT_SET_ID_NOT_EXIST (-10255L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_DELETE_FILE (-10256L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_FORMAT_FS (-10257L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_LOAD_FS (-10258L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_GET_DEV_INFO (-10259L)
SL_ERROR_FS_IMAGE_EXTRACT_FAILED_TO_DELETE_STORAGE (-10260L)
SL_ERROR_FS_IMAGE_EXTRACT_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_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.UNKNOWN (−10287L)
SL_ERROR_FS_BUNDLE_NOT_CONTAIN_FILES (−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_ASMETRIC_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_FILE_WRITE_NVMEM_HEADER (−10317L)
SL_ERROR_FS_FAILED_TO_ALLOCATE_MEM (−10318L)
SL_ERROR_FS_OPERATION_BLOCKED_BY_VENDOR (−10320L)
SL_ERROR_FS_FAILED_TO_READ_NVMEM_FILE_SYSTEM (−10321L)
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_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_PRODUCTION (−10352L)
SL_ERROR_FS_CONFIG_FILE_RET_READ_FAILED (−10353L)
SL_ERROR_FS_CONFIG_FILE_CHECKSUM_ERROR_SECURITY_ALERT (−10354L)
SL_ERROR_FS_CONFIG_FILE_NO_SUCH_FILE (−10355L)
SL_ERROR_FS_IMAGE_HEADER_READ_FAILED (−10357L)
SL_ERROR_FS_CERT_STORE_DOWNGRADE (−10358L)
SL_ERROR_FS_PROGRAMMING_IMAGE_NOT_VALID (−10359L)
SL_ERROR_FS_PROGRAMMING_IMAGE_NOT_VERIFIED (−10360L)
SL_ERROR_FS_RESERVE_SIZE_IS_SMALLER (−10361L)
SL_ERROR_FS_WRONG_ALLOCATION_TABLE (−10362L)
SL_ERROR_FS_ILLEGAL_SIGNATURE (−10363L)
SL_ERROR_FS_FILE_ALREADY_OPENED_IN_PENDING_STATE (−10364L)
SL_ERROR_FS_INVALID_TOKEN_SECURITY_ALERT (−10365L)
SL_ERROR_FS_NOT_SECURE (−10366L)
SL_ERROR_FS_RESET_DURING_PROGRAMMING (−10367L)
SL_ERROR_FS_CONFIG_FILE_RET_WRITE_FAILED (−10368L)
SL_ERROR_FS_FILE_IS_ALREADY_OPENED (−10369L)
SL_ERROR_FS_FILE_IS_OPEN_FOR_WRITE (−10370L)
SL_ERROR_FS_ALERT_CANT_BE_SET_ON_NON_SECURE_DEVICE (−10371L) /* Alerts can be configured on non-secure device */
SL_ERROR_FS_WRONG_CERTIFICATE_FILE_NAME (−10372L)

B.8. Other error codes

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

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

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

SL_ERROR_LENGTH_ERROR_PREFIX (−30734L)
SL_ERROR_WAKELOCK_ERROR_PREFIX (−30735L)
SL_ERROR_DRV_START_FAIL (−30736L)
SL_ERROR_VALIDATION_ERROR (−30737L)
SL_ERROR_SETUP_FAILURE (−30738L)
SL_ERROR_HTTP_SERVER_ENABLE_FAILED (−30739L)
SL_ERROR_DHCP_SERVER_ENABLE_FAILED (−30740L)
SL_ERROR_WPS_NO_PIN_OR_WRONG_PIN_LEN (−30741L)
C. Root certificate catalog

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

ACEDICOM Root
Actalis Authentication Root CA
AddTrust Class 1 CA Root
AddTrust External CA Root
AddTrust Qualified CA Root
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
Bypass 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
SZAIFIR ROOT CA
SZAIFIR 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
List of Figures

1. Block diagram .................................................. 10
2. Pinout (top view) .................................................. 17
3. Power up .......................................................... 21
4. Quick start setup ................................................. 23
5. Modes of operation .............................................. 29
6. Host Interface .................................................. 31
7. TCP socket work flow ........................................... 50
8. UCP socket work flow ......................................... 51
9. SSL/TLS handshake .............................................. 52
10. Provisioning main page ....................................... 82
11. Provisioning main page ....................................... 83
12. Test page ....................................................... 91
13. Test page ....................................................... 92
14. OTA webpage .................................................. 96
15. OTA webpage upload ......................................... 97
16. OTA in progress ............................................... 97
17. Finalize OTA .................................................. 98
18. Layout .......................................................... 104
19. Placement of the module with integrated antenna .......... 105
20. Dimensioning the antenna feed line as micro strip .... 105
21. 169 MHz dipole-antenna .................................... 108
22. 434 MHz dipole-antenna .................................... 109
23. 868 MHz dipole-antenna .................................... 110
24. 868 MHz magnet foot antenna with 1.5 m antenna cable 111
25. 2.4 GHz dipole-antenna .................................... 112
26. Reference design: Schematic, most important parts .... 114
27. Reference design: Layout ..................................... 115
28. Close-up: Layout ............................................... 116
29. Reference design: Stack-up .................................. 116
30. Close-up: Schematic .......................................... 117
31. Reflow soldering profile ..................................... 120
32. Module dimensions [mm] ..................................... 124
33. Footprint and dimensions [mm] ............................... 125
34. Lot number structure .......................................... 127

List of Tables

1. Ordering information .......................................... 10
2. Recommended operating conditions ....................... 11
3. Absolute maximum ratings .................................. 11
4. Power consumption ............................................ 12
5. Radio characteristics .......................................... 12
6. Modulation schemes and peak data rate. ................. 13
7. Pin characteristics ............................................. 14
8. TX power vs current consumption, conducted measurement of continuous data transmission, rate 1Mbps (DSSS) 15
9. TX power vs current consumption, conducted measurement of continuous data transmission, rate 54Mbps (OFDM) . ........................................ 16
10. Pinout .......................................................................................... 19
11. Minimal pin configuration ............................................................ 20
12. Quick start addresses and roles .................................................. 24
13. Key features ................................................................................ 28
14. Application modes ......................................................................... 30
15. UART parameters ........................................................................ 31
16. AT+start ...................................................................................... 36
17. AT+stop ........................................................................................ 36
18. AT+test ....................................................................................... 37
19. AT+reboot ................................................................................... 37
20. AT+factoryreset .......................................................................... 37
21. AT+sleep ..................................................................................... 38
22. AT+get ....................................................................................... 39
23. AT+set ....................................................................................... 40
24. AT+wlanSetMode ......................................................................... 40
25. AT+wlanScan ............................................................................... 41
26. AT+wlanConnect .......................................................................... 42
27. AT+wlanDisconnect ..................................................................... 42
28. AT+wlanProfileAdd ...................................................................... 43
29. AT+wlanProfileGet ...................................................................... 43
30. AT+wlanProfileDel ....................................................................... 43
31. AT+wlanSet ............................................................................... 44
32. AT+wlanGet ............................................................................... 45
33. AT+wlanPolicySet ........................................................................ 46
34. AT+wlanPolicyGet ........................................................................ 46
35. IP addresses ................................................................................ 47
36. AT+netCfgSet .............................................................................. 48
37. AT+netCfgGet ............................................................................ 49
38. AT+socket (create a socket) .......................................................... 53
39. AT+close (close a socket) ............................................................... 53
40. AT+bind ..................................................................................... 53
41. AT+listen ................................................................................... 53
42. AT+connect ............................................................................... 54
43. AT+accept .................................................................................. 54
44. AT+select ................................................................................... 54
45. AT+setSockOpt ........................................................................... 55
46. AT+setSockOpt continued ............................................................. 56
47. Supported cipher methods ............................................................. 56
48. AT+getSockOpt .......................................................................... 57
49. AT+recv ...................................................................................... 57
50. AT+recvFrom ............................................................................... 58
51. AT+send ..................................................................................... 58
52. AT+sendTo .................................................................................. 59
53. AT+fileGetFileList ...................................................................... 61
54. AT+fileOpen ................................................................................ 62
55. AT+fileClose ............................................................................... 62
56. AT+fileDel .................................................................................. 63