WLM10x

USER MANUAL

Laird TECHNOLOGIES®

Innovative **Technology** for a **Connected** World

www.lairdtech.com

Laird Technologies is the world leader in the design and manufacture of customized, performance-critical products for wireless and other advanced electronics applications. Laird Technologies partners with its customers to find solutions for applications in various industries such as:

Network Equipment

Telecommunications

Data Communications

Automotive Electronics

Computers

Aerospace

Military

Medical Equipment

Consumer Electronics

Laird Technologies offers its customers unique product solutions, dedication to research and development, as well as a seamless network of manufacturing and customer support facilities across the globe.

LWS-UM-WLM10x

This material is preliminary

Information furnished by Laird Technologies in this specification is believed to be accurate. Devices sold by Laird Technologies are covered by the warranty and patent indemnification provisions appearing in its Terms of Sale only. Laird Technologies makes no warranty, express, statutory, and implied or by description, regarding the information set forth herein. Laird Technologies reserves the right to change specifications at any time and without notice. Laird Technologies' products are intended for use in normal commercial and industrial applications. Applications requiring unusual environmental requirements such as military, medical life-support or life-sustaining equipment are specifically not recommended without additional testing for such application.

Limited Warranty, Disclaimer, Limitation of Liability

Revision History

Revision	Description
Version 1.0	11/10/2011 – Initial Release
Version 1.1	12/1/2011 – Minor edits to power table and ordering information

Table of contents

Overview

This document describes the WLM10x Low Power module hardware specification. The WLM10x based modules provide cost effective, low power, and flexible platform to add Wi-Fi® connectivity for embedded devices for a variety of applications, such as wireless sensors and thermostats. It combines ARM7-based processors with an RF transceiver, 802.11 MAC, security, & PHY functions, FLASH and SRAM, onboard and off module certified antenna options, and various RF front end options for end customer range needs in order to provide a WiFi and regulatory certified IEEE 802.11 radio with concurrent application processing services for variety of applications, while leverage existing 802.11 wireless network infrastructures.

Product Overview

Family of modules with different antenna options:

Dimensions: 1.450 inches by 0.900 inches by 0.143 inches (Length * Width * Height) 48-pin Dual Flat pack PCB Surface Mount Package.

Simple API for embedded markets covering large areas of applications

Compliant with IEEE 802.11and regulatory domains:

DSSS modulation for data rate of 1 Mb/s and 2 Mb/s; CCK modulation rates of 5.5 and 11 Mb/s.

Compatible with IEEE 802.11b/g/n.

Supports short preamble and short slot times.

WiFi Certified Solution

- Supports 802.11i security
  - WPA™ - Enterprise, Personal
  - WPA2™ - Enterprise, Personal
  - Vendor EAP Type(s)
    - EAP-TTLS/MSCHAPv2, PEAPv0/EAP-MSCHAPv2, PEAPv1/EAP-GTC, EAP-FAST, EAP-TLS

High-throughput hardware AES and RC4 encryption/decryption engines.

RoHS and CE compliant

FCC/IC Certified

	WLM100	**WLM101**
FCC ID	YOPGS1011MEE	YOPGS1011MEP
IC ID	9154A-GS1011MEE	9154A-GS1011MEP

Fully compliant with EU & meets the R&TTE Directive for Radio Spectrum
Japan Radio Type Approval (i.e. TELEC) pre-scan compliant

Dual ARM7 Processor Platform:

1st ARM7 processor (WLAN CPU) for WLAN software

2nd ARM7 (APP CPU) for networking software

Based on Advanced Microprocessor Bus Architecture (AMBA) system:

- AMBA High-Speed Bus (AHB).
- AMBA Peripheral Bus (APB).

On-chip WLAN boot code located in dedicated boot ROM.

Interfaces:

PCB or external antenna options.

Two general-purpose SPI interfaces (each configurable as master or slave) for external sensors, memory, or external CPU interface; one interface may be configured as a high-speed Slave-only.

Two multi-purpose UART interfaces.

Up to 23 configure able general purpose I/Os.

Single 3.3V supply option

- I/O supply voltage 1.8 ~ 3.3V option

One PWM output

I^2C master/slave interface.

Two 10-bit ADC channels, aggregate sample rate 32 kS/s.

Two alarm inputs to asynchronously awaken the chip.

Support of up to two control outputs for power supply and sensors.

Embedded RTC (Real Time Clock) can run directly from battery.

Power supply monitoring capability.

Low-power mode operations

- Sleep, Deep Sleep, and Standby

System Overview

Purpose

This document describes the operation and serial command interface for the Laird Technologies WLM10x WLAN Modules. The WLM10x module enables embedded devices with a UART/SPI interface to gain access to an IP network over an 802.11-compliant (Wi-Fi®) wireless network connection, using only serial commands.

Scope

This document reviews the architecture of the WLM10x software and provides the programmer with necessary command syntax required to manage the Wi-Fi interface and send and receive network messages. This document assumes that the reader is generally familiar with Internet Protocol (IP) networks and the operation and management of 802.11 wireless devices.

Overview

The WLM10x is used to provide Wi-Fi Capability to any devices having a serial interface. This approach offloads WLAN, TCP/IP stack and network management overhead to the Wi-Fi chip, allowing a small embedded host, based on such low-cost microcontrollers as the 8051, PIC, MSP430, or AVR to communicate with other hosts on the network using a Wi-Fi wireless link. The host processor can use serial commands to configure the WLM10x and to create wireless and network connections.

Terminology

Term	Explanation
AP	Access Point
API	Application Programmer's Interface
BSSID	Basic Service Set Identifier
CID	Connection Identifier
CPL	Clock Polarity
CPH	Clock Phase
DHCP	Dynamic Host Configuration Protocol
DIN	Data Input
DOUT	Data Output
IP	Internet Protocol
MSPI	Master SPI
MTU	Maximum Transfer Unit
PSK	Pre-shared key
RSSI	Received Signal Strength Indication
SSID	Service Set Identifier
SPI	Serial Peripheral Interface
SSPI	Slave SPI
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver/Transmitter
UDP	User Datagram Protocol
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access
PA	Power Amplifier

Table 1-1: Glossary of Terms

Standards

The following standards and conventions are considered in this design:

IEEE 802.11 b/g

ITU V.25ter AT Command Set

Chapter 2
Hardware Specifications

Architecture

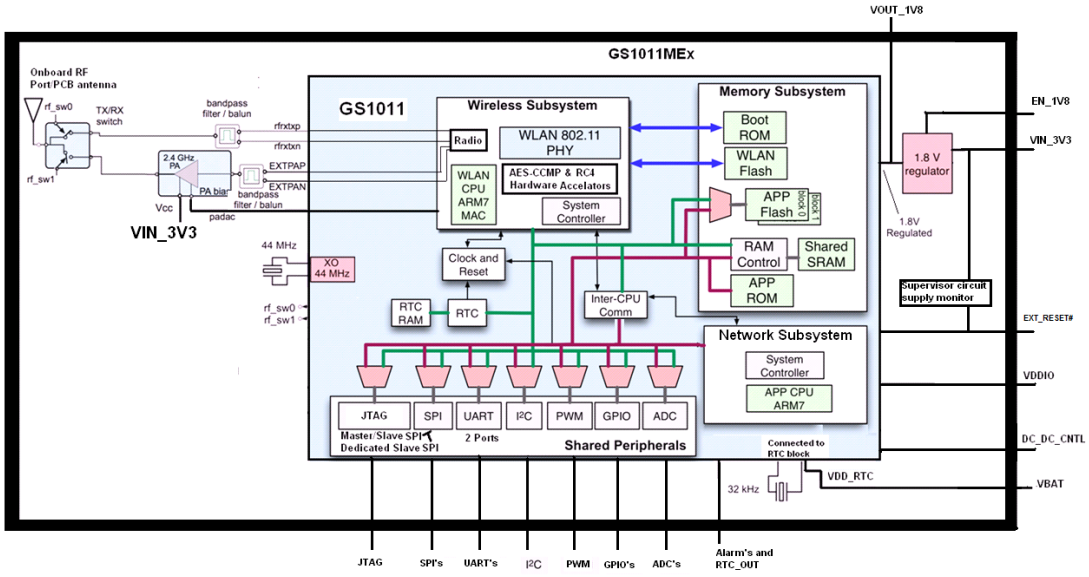


Figure 2-1: WLM101 Block Diagram

Block Diagram Description

WLM10x module is a highly integrated ultra low power Wi-Fi module that contains the following:

The module includes Gainspan GS1011 SoC, which contains media access controller (MAC), baseband processor, security, on-chip flash memory and SRAM, and an applications processor in a single package.

The module contains two ARM7-based processors, one dedicated to Networking Subsystems, and the other dedicated to Wireless Subsystems.

The module carries an 802.11 radio with onboard 32 KHz & 44 MHz crystal circuitries, RF, and certified PCB antenna or external antenna options capable of +17dBm output power at the antenna

Variety of interfaces are available such as two UART blocks using only two data lines per port with optional hardware flow controls, two SPI block (one dedicated as a serial slave with configurable hardware interrupt to the HOST), I2C with Master or slave operation, JTAG port, low-power 10-bit ADC capable of running at up to 32 Ksamples/Sec., GPIO's, and LED Drivers/GPIO with 20mA capabilities.

WLM10x contains single power supply (VIN_3V3) with optional module controlled external regulator enable control pin (DC_DC_CNTL), a separate I/O supply (VDDIO) that can be tied to the HOST supply rail without the use of external voltage translators, Real Time Clock (RTC) with battery supply monitor option (VBAT), and two external alarm inputs enable wake-up of the system on external events & outputs (ALARM and RTC_OUT) to enable periodic wake-up of the remainder of the system.

The Module carries onboard single supply monitor for under voltage supply and onboard 1.8V regulator with enable/disable capabilities for power critical applications.

Wireless Subsystem

The Wireless subsystem provides the WLAN PHY, MAC and baseband functionality. It contains the WLAN CPU, a 32-bit ARM7 TDMI-S core running at up to 44 MHz. It includes an IEEE 802.11b/g -compatible RF transceiver, which supports Direct Sequence Spread Spectrum (DSSS) 1 Mb/s and 2 Mb/s data rates, and Complementary Code Keyed (CCK) 5.5 Mb/s and 11 Mb/s data rates. The WLAN subsystem includes an integrated power amplifier. In addition, it contains hardware support for AES-CCMP encryption (for WPA2) and RC4 encryption (for WEP & WPA/WPA2 TKIP) encryption/decryption.

The WLAN subsystem contains the control logic and state machines required to drive the low power DSSS modem, and perform pre-processing (in transmit mode) or post-processing (in receive mode) of the data stream. The WLAN subsystem manages DMA accesses, data encryption/decryption using the AES algorithm, and CRC computation.

Onboard Antenna / RF port / Radio

The WLM10x modules have fully integrated RF frequency synthesizer, reference clock, and high power PA for extended range applications. Both TX and RX chain in the module incorporate internal power control loops. The WLM10x modules also incorporate onboard printed antenna option plus a variety of regulatory certified antenna options for various application needs.

Network Subsystem

APP CPU

The APP CPU is an ARM7 TDMI-S. It incorporates an AHB interface and a JTAG debug interface. The network RTOS, network stack, and customer application code can reside on this CPU.

Memory Subsystem

The GS1011 Chip contains several memory blocks:

Boot ROM blocks.

- The software contained in this ROM provides the capability to download new firmware via the SPI Slave or UART interfaces and to control the update of WLAN and APP Flash Memory.

384 KBytes of Embedded Flash to store program code.

- Three embedded Flash blocks of 128K bytes each
  - WLAN Flash (contains the WLAN Software)
  - APP Flash 0 and 1 (contain the Network/Application Software)

128 KBytes of RAM shared between the two integrated CPU's.

512 bytes of RTC memory ((retains data in all states including Standby, as long as the battery or other voltage supply is present)

Clock Circuitries

The WLM10x architecture uses a low-power oscillator (i.e. 32 kHz) to provide a minimal subset of functions when the chip is in its low-power Standby mode, and a high-speed 44 MHz oscillator to provide clock signals for the processors, bus, and interfaces during active operation. Intermediate modes of operation, in which the 44 MHz oscillator is active but some modules are inactive, are obtained by gating the clock signal to different subsystems. The Clock & Reset Controller, within the device, is responsible for generation, selection and gating of the clocks used in the module to reduce power consumption in various power states.

Real Time Clock (RTC)

To provide global time (and date) to the system, the WLM10x module is equipped with a low-power Real Time Clock (RTC).

RTC key features include:

32.768 kHz crystal support.

Two external alarm inputs to wake up the device.

Two programmable periodic outputs (one for a DC/DC regulator and one for a sensor).

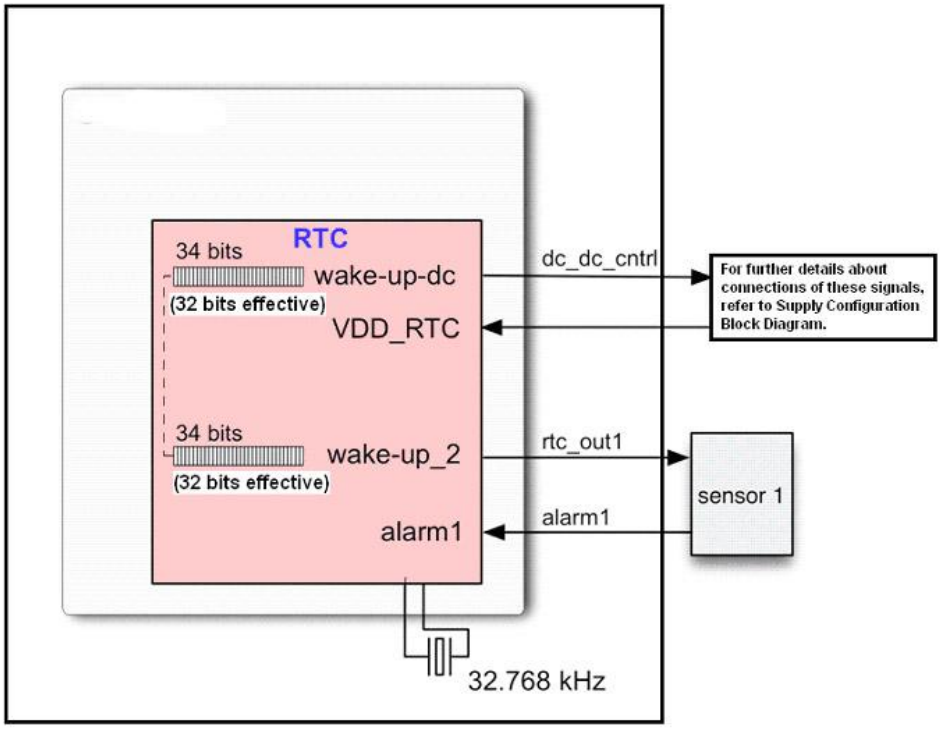
Embedded 128x32 non-volatile (battery-powered) RAM.

Embedded Power On Reset.

Real Time Counter (48 bits; 46 bits effective).

An overview of RTC block diagram is shown in the RTC Interface Diagram. The RTC contains a low-power oscillator that can use 32.768 kHz crystals. In normal operation the RTC is always powered up, even in the Standby state (see RTC Interface Diagram).

Two programmable embedded alarm counters (wrap-around) are provided to enable periodic wake-up of the remainder of the system, and one independent external component (typically a sensor). Two external alarm inputs enable wake-up of the system on external events. The global times are recorded on each external event and if the system is in the Power-ON state (see WLM10x System States figure later in this chapter), an interrupt is provided. The RTC includes a Power-On Reset (POR) circuit, to eliminate the need for an external component. The RTC contains low-leakage non-volatile (battery-powered) RAM, to enable storage of data that needs to be preserved while in Standby.



Total current consumption of the RTC in the worst case is typically less than 5 µA without data storage, using the 32.768 kHz oscillator.

Resolution of the wake-up timer is one clock cycle, and, with onboard 32KHz. CLK, each 32bit effective register can provide up to 1.5 days worth of standby duration as the longest standby period. Polarity of the rtc_out1 pin is programmable.

Real Time Counter

The Real Time Counter features:

48-bit length (with absolute duration dependent on the crystal frequency used).

Low-power design.

This counter is automatically reset by power-on-reset.

This counter wraps around (returns to “all-0” once it has reached the highest possible “all-1” value).

RTC Outputs

There are two RTC outputs (dc_dc_cntl and rtc_out1) that can be used to control external devices, such as sensors or voltage regulators.

DC_DC_CNTL

During Power-on-Reset (e.g. when the battery is connected), the dc_dc_cntl pin is held low; it goes high to indicate completion of RTC power-on-reset. This pin can be used as an enable into an external device such as voltage regulator.

RTC_OUT1

The rtc_out1 signal can be disabled or driven by the Wake-up Counter 2. This counter is 34 bits long, and operates in the same fashion as Wake-up Counter 1. The rtc_out1 signal can be configured to output the low-power crystal oscillator clock (i.e. the 32 kHz clock) instead of a simple state transition. Wake-up Counter 2 is automatically reset at Power-on-Reset.

RTC Alarm Inputs 1 and 2

The RTC inputs alarm1 and alarm2 operate as follows:

dc_dc_cntl is set to “1” (typically enabling the power supply to the rest of the WLM10x) whenever either of these inputs changes state.

The RTC counter value is stored each time either of these inputs changes state.

The inputs alarm1 and alarm2 have programmable polarity. Their task is to wake up the system (by setting dc_dc_cntl output pin to “1”) when an external event occurs.

Peripherals

SPI

There are two general-purpose SPI interfaces (each configurable as master or slave) for external sensors, memory, or external CPU interface; one interface may be configured as a high-speed Slave-only. The master SPI block provides dual synchronous serial communication interfaces. The Master SPI block can be used in one of two modes of operations: as a serial master or a serial slave. Each block provides synchronous serial communication with slave or master devices, using one of the following protocols:

- Motorola Serial Peripheral Interface (SPI).
- Texas Instruments Synchronous Serial Protocol (SSP).
- National Semiconductor Microwire Protocol.

Only Motorola Serial Peripheral Interface (SPI) timing is shown in this data sheet; however, National Semiconductor Microwire Protocol or Texas Instruments Synchronous Serial Protocol (SSP) modes are certainly supported. The SPI interface can also be used to access non-volatile external memory, such as an EEPROM block. The interface uses the SPI master mode to allow easy connection to industry-standard EEPROMs.

The shared SPI blocks provide the following general features:

- 32-bit AMBA APB interface to allow access to data, control, and status information by the host processor.
- Full-duplex serial-master or serial-slave operation.
- Two clock design:
- APB bus clock for bus interface and registers.
- Serial input clock for core logic.
- Support of external EEPROM or other non-volatile memory.
- Programmable choice of Motorola SPI, Texas Instruments Synchronous Serial Protocol or National Semiconductor Microwire.
- Programmable control of the serial bit rate of the data transfer in serial-master mode of operation.
- Programmable phase and polarity of the bit rate clock.
- Programmable transfer modes to perform transmit and receive, transmit only, receive only and EEPROM read transfers.
- Programmable data word size (8, 16, 24 & 32 bits) for each data transfer.
- Transmit and receive FIFO buffer depth 8 words (of the selected size).
- Configurable number of slave select outputs in serial-master mode of operation: 1 to 4 serial slave-select output signals can be generated.
- Combined interrupt line with independent masking of interrupts.
- Transmit FIFO overflow, transmit FIFO empty, transmit FIFO underflow, receive FIFO full, receive FIFO underflow, receive FIFO overflow, and receive FIFO timeout interrupts.
- Transmit FIFO empty and receive FIFO full interrupts provide programmable threshold values.

Both SPI blocks are configured to provide a FIFO depth of four entries.

The SPI master interface can be used to access external sensor devices, and EEPROM containing system parameters, under software control. The main purpose of the SPI slave interface is to provide control of the WLM10x from an external CPU.

SPI chip select (MPSISI_CS0 or MPSI_CS1) signals frame each data word. If the chip select is required to remain asserted for multiple data words, then a GPIO pin should be used for the chip select instead of the SPI chip select signals. For other SPI Interface Timing, please refer to ADC Parameters in Chapter 2 Hardware Specifications.

I²C

The I²C block provides a two-wire I²C serial interface. It provides the following features:

32-bit AMBA APB interface to allow access to data, control, and status Information by the host processor.

Serial 2-wire I²C bus, compliant to the I²C Bus Specification Version 2.1.

Supports only one transfer in Standard mode (100 Kb/s) and fast speed mode with a bit rate of up to 392 Kb/s.

Supports Multi-Master System Architecture through I²C bus SCL line Synchronization and Arbitration.

Transmitter and Receiver: The I²C block can act as the Transmitter or Receiver depending on the operation being performed.

Master or slave I²C operation.

7- or 10-bit addressing.

Ignores CBUS addresses (an older ancestor of I²C that used to share the I²C bus).

Interrupt or polled mode operation.

Combined interrupt line triggered by:

- Tx FIFO not FULL.
- Rx FIFO not EMPTY.
- Rx FIFO FULL (can be used to transfer data by host interface in bursts).
- Tx FIFO EMPTY (can be used to transfer data by host interface in bursts).
- Rx FIFO OVER RUN.
- Master mode to Slave Transfer Request.
- Slave Transmit Request.
- Break Interrupt (master mode): No Acknowledge received from slave for slave address or write data.

Digital debounce logic for the received SDA and SCL lines.

Hold Delay Insertion on SDA line.

UART

The WLM10x includes two UART blocks. Each UART block provides an asynchronous communication interface, using only two data lines: Rx data and Tx data. Hardware flow control using RTS/CTS signalling is provided as an option. The UART is a standard asynchronous serial interface, 16450/16550 compatible. It provides the following features:

- Operation in full-duplex mode.
- All standard bit rates up to 921.6 kbps are supported.
- RTS/CTS flow control handshake (standard 16550 approach).
- 5, 6, 7 and 8-bit character format.
- 1 or 2 stop bits (1.5 in case of a 5-bit character format).
- Parity bit: none, even, odd, mark, or space.
- 16-byte Rx and 16-byte Tx FIFOs.

The UART Serial port can be used to communicate with a PC or other devices, for debug or additional functionality.

JTAG

The JTAG ports facilitate debugging of the board and system designs. This block has the following features:

- Compliant to IEEE-1149.1 TAP ports.
- One JTAG boundary scan TAP port.
- One set of JTAG pins, which support the following mode of operation:
- APP ARM7TDMI-S Debug Mode.

GPIO & LED Driver / GPIO

The GPIO ports are referenced to VDDIO. Two GPIO pins called GPIO30_LED1 & GPO31_LED2 have the capability to sink/source 20 mA typical (VDDIO=3.3V) to connect to devices such as switch contacts or LEDs. I2C_DATA/GPIO8 & I2C_CLK/GPIO9 have the capability to sink/source 12 mA typical (VDDIO=3.3V). Other GPIO's have the capability to sink/source 4 mA typical (VDDIO=3.3V). All inputs are capable of generating processor interrupts. They can be individually programmed to provide edge- or level-triggered interrupts.

ADC

The ADC is a 10-bit, low-power, A-to-D converter capable of running at up to 32 ksps. The ADC contains an internal band-gap reference which provides a stable 1.2 V reference voltage. The ADC can be programmed to use the 1.8 V supply as the full-scale reference. The ADC uses an input clock with a maximum frequency of 1 MHz. A conversion requires 32 clock cycles.

When the internal band-gap reference is used, the reported integer *Value* at temperature *T* (ºC) is related to the voltage V_{actual} at the input pin as

$$V_{actual} = Value\left(\frac{1.2444 - 0.00014(25 - T)}{1023}\right)$$

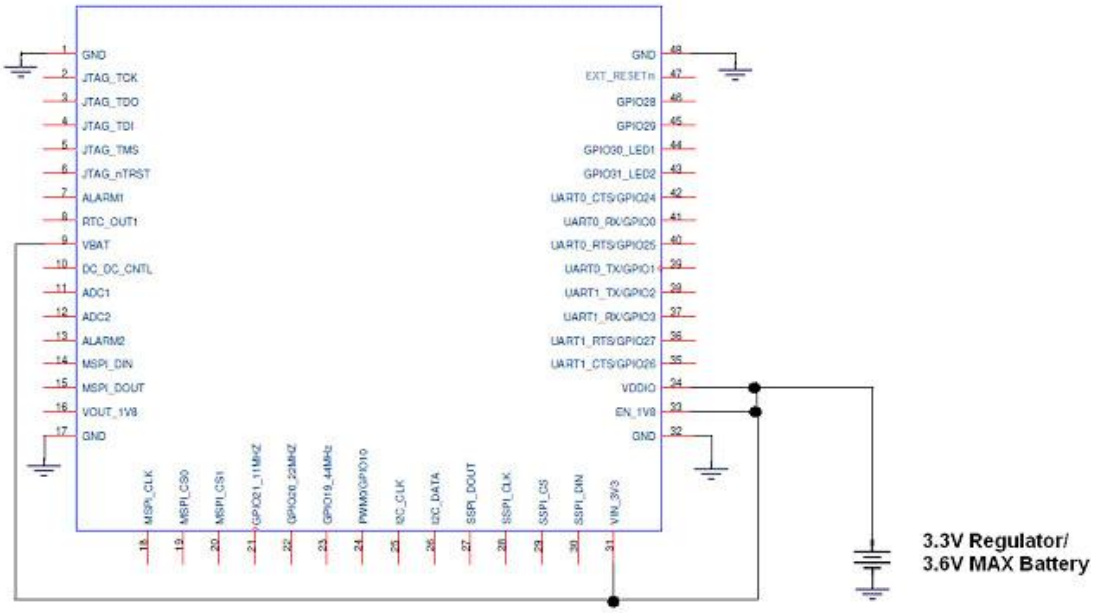
When the 1.8V supply voltage is used as the reference, the corresponding relation is:

$$V_{actual} = Value\left(\frac{V_{DD,ADC} - 0.036}{1023}\right)$$

To reduce power consumption the ADC can be disabled automatically between periodic measurements and after single measurements.

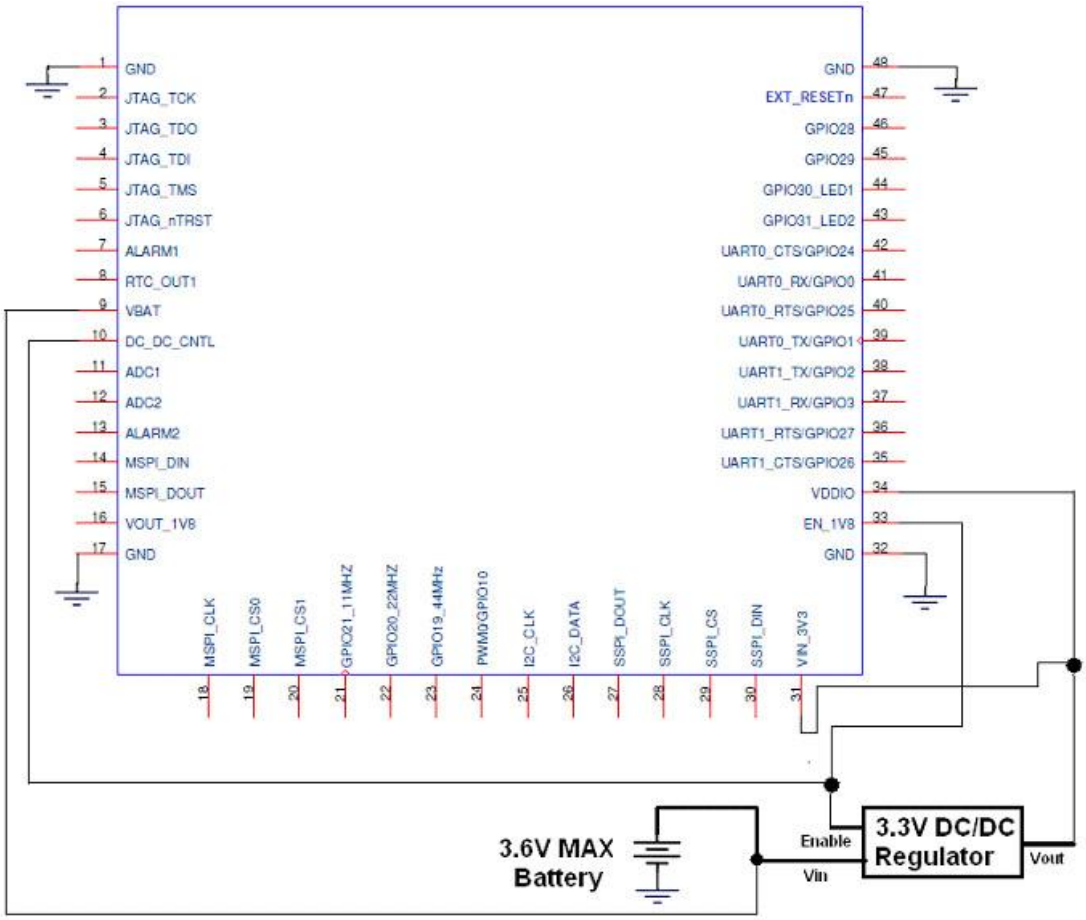
Power Supply

In this section, diagrams are shown for various application power supply connection.



Notes:

1. This connection applies generally for designs that target to keep system power on always and use 3.3V I/O
2. Always On is obtained by tying EN_1V8 to 1 which is the enable for the 1.8V voltage regulator.
3. In this state system can still go to deep sleep state and take advantage of low power consumption, but system will not go into the lowest power consumption state (i.e standby state).



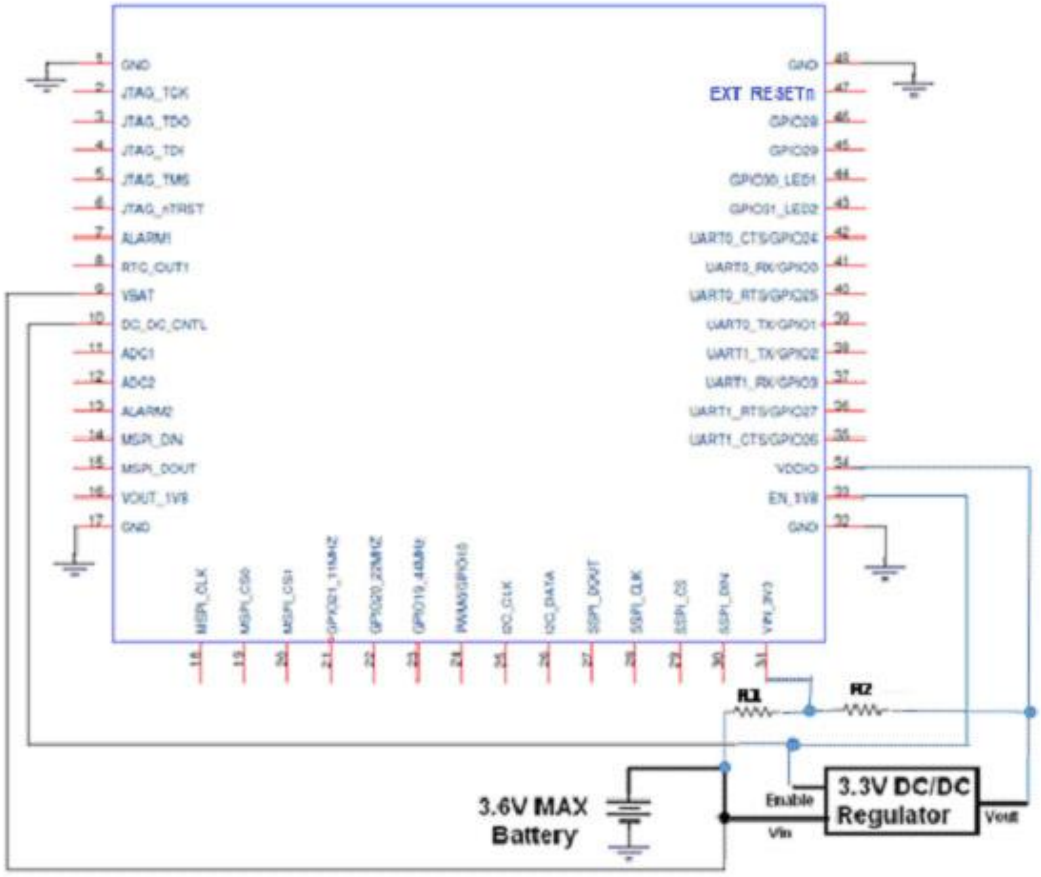
Applications that require Standby Mode and use WLM10x MUST use this connection configuration to take advantage of the lowest power consumption during standby mode. In this connection it is important to note the following:

1. WLM10x, module PA is supplied with VIN_3V3 and in-rush current for PA transmission; thus, the 3.3V DC/DC Regulator may have to be an Up/Down regulator depending on the battery used
2. For WLM10x, VDDIO and VIN_3V3 power MUST be shut OFF in standby mode so there is no leakage from PA device and thus achieve the lowest current consumption.
3. Since power to VIN_3V3 is turned off, when power is applied again, the WLM10x device will be kept in reset until the voltage supervisor device releases reset.

Power Supply Connection Summary

Module	Standby Support required	1V8_EN Connection	VIN_3V3 in Standby	VDDIO	VDDIO in Standby	Standby Wake-Up Latency	Refer to Figure Titled in this Chapter
WLM10x	Yes	DC_DC_CNTL	OFF	3.3V	OFF	< 15ms	WLM10xE Battery Powered with 3.3V IO and Standby Support
WLM10x	No (always on or deep sleep)	Supply Voltage	ON	3.3V	ON	NA	WLM10x Always ON Power Supply Connections

For designs that plan to use standby and would like to use a single baseboard that supports the WLM10x module, then they should use the following example for connections.



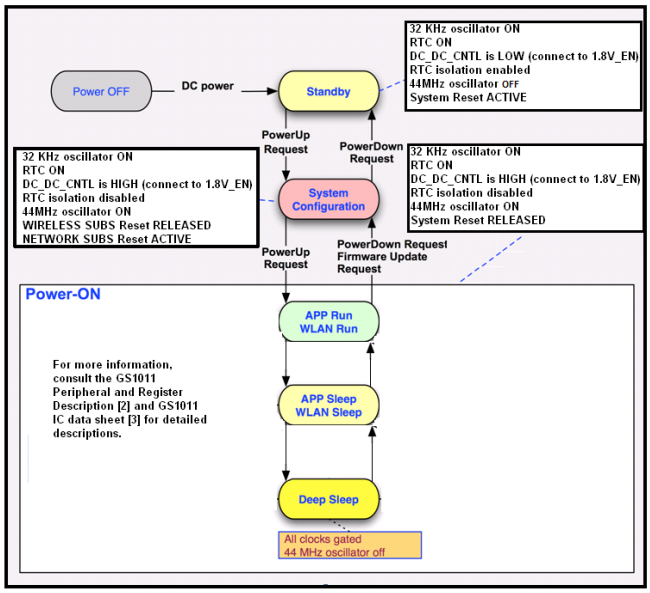
Resistor R2 stuffing options for WLM10x module

Module	*Stuffing Option*	*Equivalent Circuit*
WLM10x	R2 Only	See Figure Titled: WLM10x Battery Powered with 3.3V IO and Standby Support in this chapter

Table 2-2: Stuffing Options

System States

The following figure shows the power management/clock states of the WLM10x system.



The system states of the WLM10x system are as follows:

Power OFF: No power source connected to the system. I/Os should not be driven high by an external device during this state.

Standby: When supply voltage is stable and the RTC is powered up, and the system transitions from the Power OFF state to the Standby state. In this state only the 32.768 kHz clock is running, and the RTC is powered directly by the battery or DC supply. VDDIO to the system is off and 1.8V regulator is off as DC_DC_CNTL to the 1.8V enable of the internal regulator is de-asserted. In this state only the RTC RAM retains the state, SRAM, CPUs and I/Os are all in OFF state, as there is no 1.8V being supplied to the WLM10x device. VDDIO should be off, and the I/Os should not be driven high by an external device during this state.

This is the lowest-power-consumption state. In a typical application, the system returns to the Standby state between periods of activity, to keep the average power very low and enable years of operation using conventional batteries.

Note: During first battery plug, i.e when power is applied first time to the RTC power rail (VBAT), the RTC startup up latency will be at least couple of hundred ms as it is waiting for stabilization of the 32KHz crystal

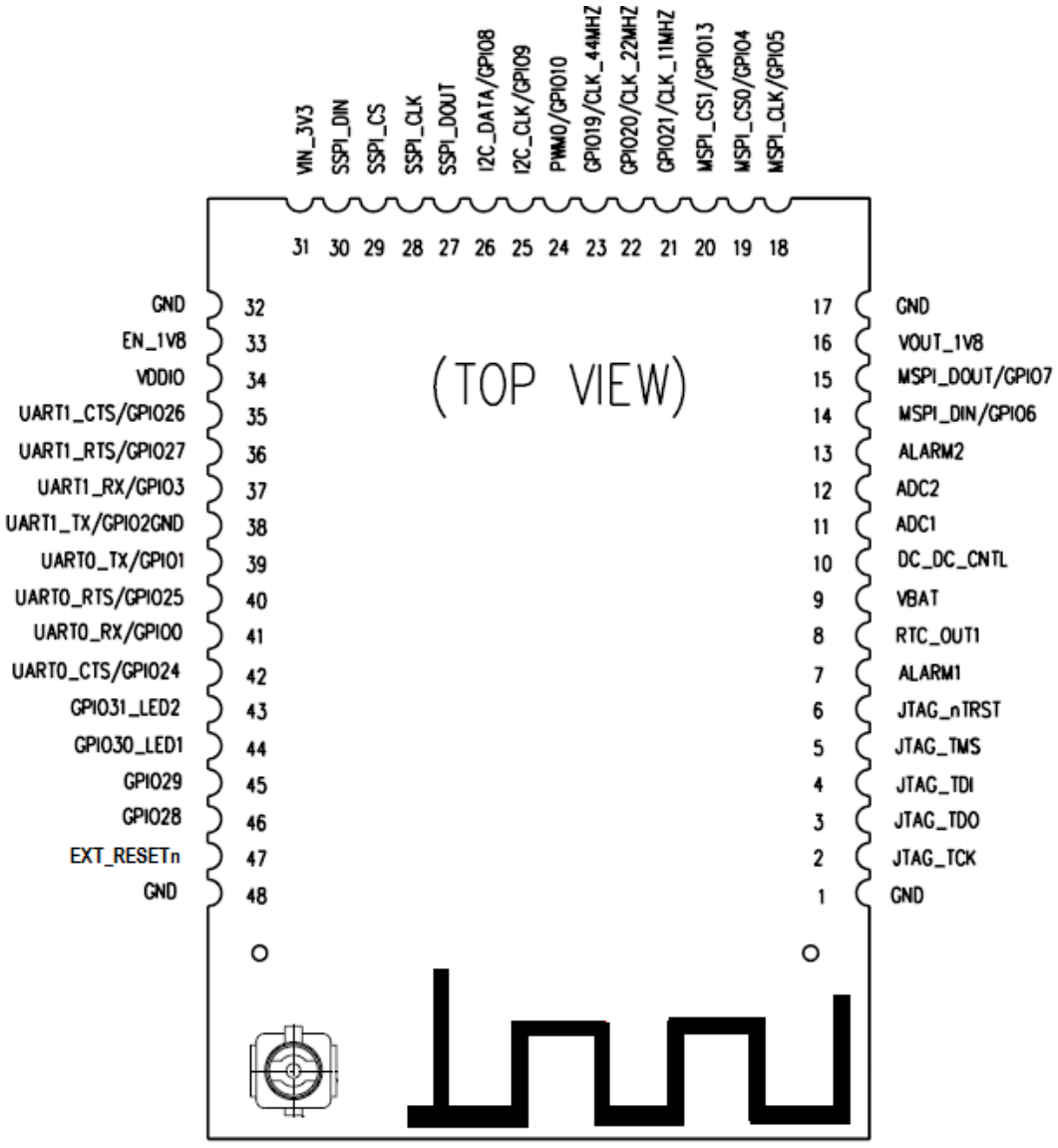
System Configuration: When a power-up is requested, the system transitions from the Standby state to the System Configuration state. In this state, the WLAN CPU is released from reset by the RTC. The APP CPU remains in the reset state during System Configuration. The WLAN CPU then executes the required system configurations, releases the APP CPU from reset, and transitions to the Power-ON state.

The System Configuration state is also entered on transition from the Power-ON state to the Standby state, to complete necessary preparations before shutting off the power to the core system. Finally, the System Configuration state is used for firmware updates.

Power-ON: This is the active state where all system components can be running. The Power-ON state has various sub-states, in which unused parts of the system can be in sleep mode, reducing power consumption. Sleep states are implemented by gating the clock signal off for a specific system component. The Deep-Sleep sub-state, in which all clocks are gated off, allows minimum power consumption while permitting rapid resumption of normal operation. In this state, the 1.8V supply is on, the SRAM retains its state and the I/Os retain their state. The 44 MHz reference oscillator is turned off to further reduce power consumption. The IO pins can generate interrupt and wake up the system from deep sleep

Pin-out and Signal Description

WLM10x Device Pin-out Diagram (Module top view)



PIN OUT DIAGRAM

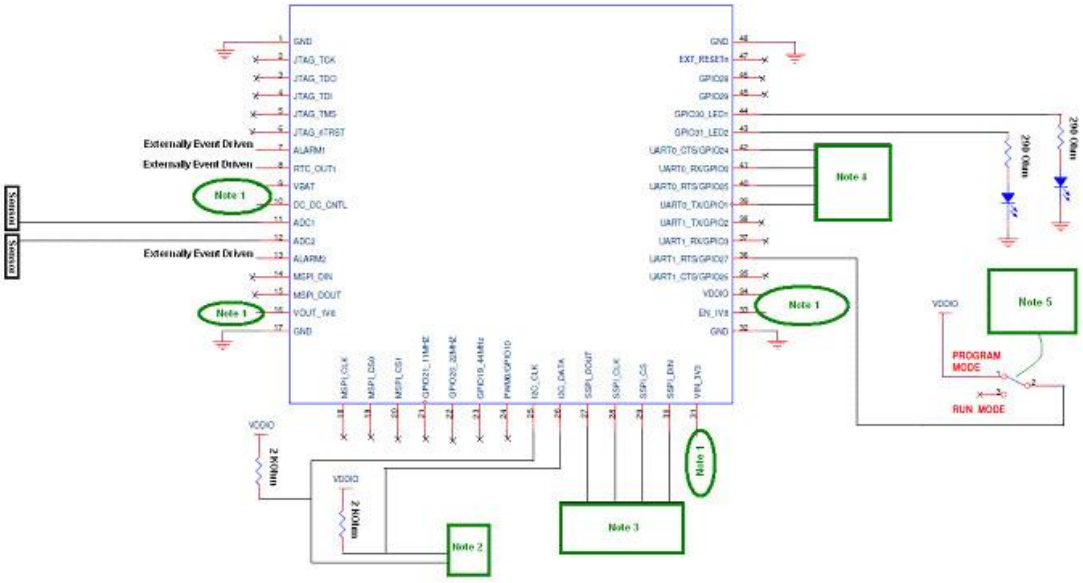
WLM10x Module Pins Description

Pins	Name	Voltage Domain	Internal Bias after hardware reset	Signal State	Description
1	GND	0V	Not Applicable	Analog port	Ground
2	JTAG_TCK	VDDIO	Pull-up (See Note 1)	Digital Input	Joint Test Action Group Test Clock
3	JTAG_TDO	VDDIO	Not Applicable	Digital Output	Joint Test Action Group Test Data Out
4	JTAG_TDI	VDDIO	Pull-up (See Note 1)	Digital Input	Joint Test Action Group Test Data In
5	JTAG_TMS	VDDIO	Pull-up (See Note 1)	Digital Input	Joint Test Action Group Test Mode Select
6	JTAG_nTRST	VDDIO	Pull-up (See Note 1)	Digital Input	Joint Test Action Group Test Mode Reset Active Low
7	ALARM1	VBAT	Pull-down (See Note 1)	Digital Input	Embedded Real Time Clock Wake Up Input 1
8	RTC_OUT1	VBAT	Not Applicable	Digital Output	Embedded Real Time Clock Wake Up Output 1
9	VBAT	VBAT	Not Applicable	Analog port	Embedded Real Time Clock Power Supply
10	DC_DC_CNTL	VBAT	Not Applicable	Digital Output	VIN_3V3 Regulator Control Output
11	ADC1	VDD18 (internal)	Not Applicable	Analog Output	General Analog to Digital Converter 1
12	ADC2	VDD18 (internal)	Not Applicable	Analog Output	General Analog to Digital Converter 2
13	ALARM2	VBAT	Pull-down (See Note 1)	Digital Input	Embedded Real Time Clock Wake Up Input 2
14	MSPI_DIN / GPIO6	VDDIO	Pull-down	Digital Input	Master Serial Peripheral Interface Bus Data Input / General Purpose Input Output
15	MSPI_DOUT / GPIO7	VDDIO	Pull-down	Digital Output	Master Serial Peripheral Interface Bus Data Output / General Purpose Input Output
16	VOUT_1V8	VIN_3V3 (internally regulated)	Not Applicable	Analog port	Internal 1.8V Vout
17	GND	0V	Not Applicable	Analog port	Ground
18	MSPI_CLK / GPIO5	VDDIO	Pull-down	Digital Input / Output	Master Serial Peripheral Interface Bus Clock / General Purpose Input Output
19	MSPI_CS0 / GPIO4	VDDIO	Pull-down	Digital Input / Output	Master Serial Peripheral Interface Bus Chip Select 0 / General Purpose Input Output
20	MSPI_CS1 / GPIO13	VDDIO	Pull-down	Digital Output	Master Serial Peripheral Interface Bus Chip Select 1 / General Purpose Input Output
21	GPO21_11MHZ	VDDIO	Pull-down	Digital Input	Internal Clock Circuitry Test Point / General Purpose Input Output
22	GPO20_22MHZ	VDDIO	Pull-down	Digital Input	Internal Clock Circuitry Test Point / General Purpose Input Output
23	GPO19_44MHZ	VDDIO	Pull-down	Digital Input	Internal Clock Circuitry Test Point / General Purpose Input Output
24	PWM0 / GPIO10	VDDIO	Pull-down	Digital Output	Pulse Width Modulator / General Purpose Input Output
25	I2C_CLK/GPIO9	VDDIO	Pull-down (Note 4)	Digital Input / Output	Inter-Integrated Circuit Clock / General Purpose Input Output
26	I2C_DATA/GPIO8	VDDIO	Pull-down (Note 4)	Digital Input / Output	Inter-Integrated Circuit Data / General Purpose Input Output
27	SSPI_DOUT	VDDIO	Pull-up (See Note 1)	Digital Input	SPI Slave Transmit Data Output to the HOST
28	SSPI_CLK	VDDIO	Pull-up (See Note 1)	Digital Input	SPI Slave Clock Input from the HOST
29	SSPI_CS	VDDIO	Pull-up (See Note 1)	Digital Input	SPI Slave Chip Select Input from the HOST
30	SSPI_DIN	VDDIO	Pull-down (See Note	Digital Input	SPI Slave Receive Data Input from the HOST

			1)		
31	VIN_3V3	VIN_3V3	Not Applicable	Analog port	Single Supply Port
32	GND	0V	Not Applicable	Analog port	Ground
33	EN_1V8	VDDIO	Need to be driven HIGH or LOW externally	Digital Input	Internal 1.8V regulator enable port-Active High
34	VDDIO	VDDIO	Not Applicable	Analog port	All I/O voltage domain (can be tied to VIN_3V3 or tied to HOST I/O supply)
35	UART1_CTS / GPIO26	VDDIO	Pull-down	Digital Input	Universal Asynchronous Receiver Transmitter 1 Clear to Send Input (See Note 6) / General Purpose Input Output
36	UART1_RTS / GPIO27	VDDIO	Pull-down (See Note 2)	Digital Output	Universal Asynchronous Receiver Transmitter 1 Request to Send Output (See Note 6) / General Purpose Input Output
37	UART1_RX / GPIO3	VDDIO	Pull-down	Digital Input	Universal Asynchronous Receiver Transmitter 1 Receive Input / General Purpose Input Output
38	UART1_TX/ GPIO2	VDDIO	Pull-down	Digital Output	Universal Asynchronous Receiver Transmitter 1 Transmitter Output / General Purpose Input Output
39	UART0_TX / GPIO1	VDDIO	Pull-down	Digital Output	Universal Asynchronous Receiver Transmitter 0 Transmitter Output / General Purpose Input Output
40	UART0_RTS / GPIO25	VDDIO	Pull-down	Digital Output	Universal Asynchronous Receiver Transmitter 0 Request to Send Output (See Note 6) / General Purpose Input Output
41	UART0_RX / GPIO0	VDDIO	Pull-down	Digital Input	Universal Asynchronous Receiver Transmitter 0 Receive Input / General Purpose Input Output
42	UART0_CTS / GPIO24	VDDIO	Pull-down	Digital Input	Universal Asynchronous Receiver Transmitter 0 Clear to Send Input (See Note 6) / General Purpose Input Output
43	GPO31_LED2	VDDIO	Pull-down	Digital Output	Light Emitting Diode Driver / General Purpose Input Output
44	GPIO30_LED1	VDDIO	Pull-down	Digital Output	Light Emitting Diode Driver / General Purpose Input Output
45	GPIO29	VDDIO	Pull-down (See Note 3)	Digital Input / Output	General Purpose Input Output
46	GPIO28	VDDIO	Pull-down (See Note 3)	Digital Input / Output	General Purpose Input Output
47	EXT_RESETn (See Note 5)	VDDIO	Pull-up	Digital Open Drain Input / Output	Module Hardware Reset Input and Power Supply Reset Monitor Indictor Active Low
48	GND	0V	Not Applicable	Analog port	Ground

Notes:

1. These pins have onboard hardware configured pull-ups/downs and cannot be changed by software.
2. If UART1_RTS is high during boot, then the WLAN will wait for Flash download via UART0. For development purposes, route this pin to a test point on the board so it can be pulled up to VIN_3V3.
3. GPIO 28 and 29 are sampled at reset to establish JTAG configuration for debugging. These signals should not be driven from an external device. If using JTAG, configure these pins as outputs.
4. If I2C interface is used, provide 2K Ohm pull-ups, to VDDIO, for pins 25 and 26 (I2C_CLK and I2C_DATA)
5. EXT_RESETn is a active low signal. It is an output during power up, indicating to the system when WLM10x device is out of power-on-reset (i.e 32 KHz crystal has stabilized. After power-on-reset, this pin is an input. It is not necessary to assert reset to the WLM10x after power on, since the WLM10x has a built-in power on reset. Also, the EXT_RESETn signal does not clear the RTC RAM or the SRAM.
6. CTS and RTS signals indicate it is clear to send or ready to send when they are LOW. If signals are high, indicates device is not ready.



Notes

1. For the noted pin configurations, please refer to data sheet power supply section.
2. If I2C interface is used, provide 2KOhm pull-ups, to VDDIO, for pins 25 and 26 (I2C_CLK and I2C_DATA). If not used, leave pins 25 and 26 as No Connects.
3. Connect to external HOST SPI (can be left as No Connects if not used).
4. Connect to external serial HOST UART (can be left as No Connections if not used)
5. This switch enables the programming of WLM10x onboard flash. Switch is recommended for development purposes and is not needed for production.

Electrical Characteristics

Absolute Maximum Ratings

Conditions beyond those cited in the following Absolute Maximum Ratings table may cause permanent damage to the WLM10x, and must be avoided.

Parameter	Symbol	Minimum	Typical	Maximum	Unit
Storage temperature	T_{ST}	-55		+125	ºC
RTC Power Supply	Vbat	-0.5		4.0	V
I/O Supply voltage	VDDIO	-0.5		4.0	V
Single Supply Port	VIN_3V3	2.7	3.3	4.0	V

Notes:

1. For limitations on state voltage ranges, please consult the section Power supply located in Chapter 1.

Operating Conditions

Parameter	Symbol	Minimum	Typical	Maximum	Unit
Extended temp. range	T_A	-40		+85	ºC
RTC Power Supply	Vbat	1.6	3.3	3.6	V
I/O Supply voltage	VDDIO	1.7	3.3	3.6	V
Single Supply Port WLM10x (as configured per Figure titled WLM10x Battery Powered with 3.3V IO and Standby Support located in Chapter 1)	VIN_3V3	3.0	3.3	3.6	V

Internal 1.8V Regulator

VIN_3V3=VDDIO=Vbat=3.3V Temp=25ºC fOSC=3.0MHz

Parameter	Symbol	Test Conditions	Minimum	Typical	Maximum	Unit
Output Voltage	VOUT_1V8			1.8		V
Maximum Output Current	IVOUT_1V8			30	50	mA
Oscillation Frequency	fOSC		1.6		3.45	MHz
1.8V Regulator Enable "H" Voltage	EN_1V8		1.0		VIN_3V3	V
1.8V Regulator Enable "L" Voltage	EN_1V8		0		0.25	V

I/O DC Specifications

Digital Input Specifications

Parameter	Symbol	Min.	Typical	Max.	Unit	Note
Input Low Voltage	V_{IL}	-0.3		0.25* VDDIO	V	
Input High Voltage	V_{IH}	0.8* VDDIO		VDDIO	V	

Table 2-3: Digital Input Parameters

Digital Output Specification

Parameter	Symbol	Min.	Typical	Max.	Unit	Note
Output Low Voltage	V_{OL}	0		0.4	V	With 4 mA load
Output High Voltage	V_{OH}	2.4V		VDDIO	V	V_{DDIO}=3.0V, DC current load 4.0 mA
		1.3V				V_{DDIO} =1.62 V, DC current load 2.0 mA
Output rise time @ VDDIO=3.3V	t_{TLH}			7	ns	With 4 mA, 33 pF load
Output fall time @ VDDIO=3.3V	t_{THL}			7	ns	With 4 mA, 33 pF load

Table 2-4: Digital Output Parameters

I/O Digital Specifications (Tri-State)

Parameter	Symbol	Min.	Typical	Max.	Unit	Note
Input Low Voltage	V_{IL}	-0.3		0.25* VDDIO	V	
Input High Voltage	V_{IH}	0.8* VDDIO		VDDIO	V	
Schmitt trig. Low to High threshold point	V_{T+}	1.5			V	
Schmitt trig. High to Low threshold point	V_{T-}			1	V	
Input Leakage Current	I_L			5	µA	Pull up/down disabled
Tri-State Output Leakage Current	Oz_L			5	µA	Pull up/down disabled
Pull-Up Resistor	R_u	0.05		1	MΩ	
Pull-Down Resistor	R_d	0.05		1	MΩ	
Output Low Voltage	V_{OL}	0		0.4	V	With 4/12/20 mA load
Output High Voltage	V_{OH}	2.4V		VDDIO	V	V_{DDIO}=3.0V With 4/12/20 mA load
		1.3 V				V_{DDIO} =1.62 V With 2/6/10 mA load
Output rise time @ VDDIO =3.3V	t_{ToLH}			7	ns	With 4/12/20 mA load, 33 pF
Output fall time @ VDDIO = 3.3V	t_{ToHL}			7	ns	With 4/12/20 mA load, 33 pF
Input rise time	t_{TiLH}			7	ns	
Input fall time	t_{TiHL}			7	ns	

Table 2-5: I/O Digital Parameters

RTC Input Specifications (with Schmitt Trigger)

Parameter	Symbol	Min.	Typical	Max.	Unit	Note
I/O Supply Voltage	V_{DDRTC}	1.2		Vbat	V	
Input Low Voltage	V_{IL}	-0.3		0.25*V_{DDRTC}	V	
Input High Voltage	V_{IH}	0.8*V_{DDRTC}		V_{DDRTC}	V	
Schmitt trig. Low to High threshold point	V_{T+}	0.57*V_{DDRTC}		0.68*V_{DDRTC}	V	
Schmitt trig. High to Low threshold point	V_{T-}	0.27*V_{DDRTC}		0.35*V_{DDRTC}	V	
Input Leakage Current	I_L		260		pA	

Table 2-6: RTC Input Parameters

RTC Output Specifications

Parameter	Symbol	Min.	Typical	Max.	Unit	Note
I/O Supply Voltage	V_{DDRTC}	1.2		Vbat	V	
Output Low Voltage	V_{OL}	0		0.4	V	
Output High Voltage	V_{OH}	0.8*V_{DDRTC}		V_{DDRTC}	V	
Output rise time	t_{TLH}	19		142	ns	50 pF load
Output fall time	t_{THL}	21		195	ns	50 pF load
Input Leakage Current	I_L		730		pA	

Table 2-7: RTC Output Parameters

Power Consumption

Test Conditions: VDD33=VDDIO=Vbat=3.3V Temp=25ºC

System state	Current (Typ.)
Standby	7uA
Deep Sleep	200uA
Receive (-81 dBm RX sens. @ 11Mb/Sec).	140 mA
Transmit (+18 dBm at antenna port @ 11Mb/Sec.)	250 mA

Table 2-8: Power Consumption in Different States

Radio Parameters

Test Conditions: VIN_3V3=VDDIO=Vbat=3.3V Temp=25ºC; 100-byte packet

Parameter	Min.	Typical	Max.	Unit	Notes
RF Frequency range	2412		2497	MHz	
Radio bit rate	1		11	Mbps	Ch 14 max is 2 Mbps
TX/RX specification for WLM10x					
Output power (average)		18		dBm	Modulated signal at antenna port; 11Mb/Sec.
Spectrum Mask				dBr	Modulated signal at antenna port
F0 +/- 11 MHz	-30				
Offset >= 22 MHz	-50				
Receive Sensitivity at antenna port		-83		dBm	11 Mbps CCK, 8% PER
		-86			5.5 Mbps CCK, 8% PER
		-90			2 Mbps QPSK, 8% PER
		-92			1 Mbps BPSK, 8% PER

Table 2-9: Radio Parameters

ADC Parameters

Test Conditions: VIN_3V3=VDDIO=Vbat=3.3V Temp=25ºC

Parameter	Min.	Typical	Max.	Unit	Notes
ADC Resolution	-	10	-	Bits	
ADC Sample Freq	1.024	-	31.25	ksps	
ADC input Clock Freq	32.768	-	1000	kHz	
ADC Full Scale Voltage	-	VOUT_1V8 – 0.036	VOUT_1V8	V	Reference =VOUT_1V8
	1.169	1.24	1.311		Reference = bandgap
	3.527	3.73	3.913		Measuring Vbat
Conversion Time		32		Clocks	Based on internally generated 1MHz or 32.768 KHz Clocks
ADC Integral Non-Linearity (INL)	-2.0	-	2.0	LSB	
ADC Differential non-linearity (DNL)	-1.0	-	1.0	LSB	
AVDD Power Supply current (operational)	-	400	800	µA	
ADC Offset Error	-10	-	10	mV	
ADC Gain Error	-10	-	10	mV	
Settling Time		-	1	µS	
Input resistance	1	-	-	MOhm	
Input Capacitance	-	10	-	pF	
Bandgap Output Voltage (Vref) (T = 25 ºC)	1.179	1.24	1.301	V	

SPI Interface Timing

Test Conditions: VIN_3V3=VDDIO=Vbat=3.3V Temp=25ºC

Motorola SPI, clock polarity SPO = 0, clock phase SPH = 0

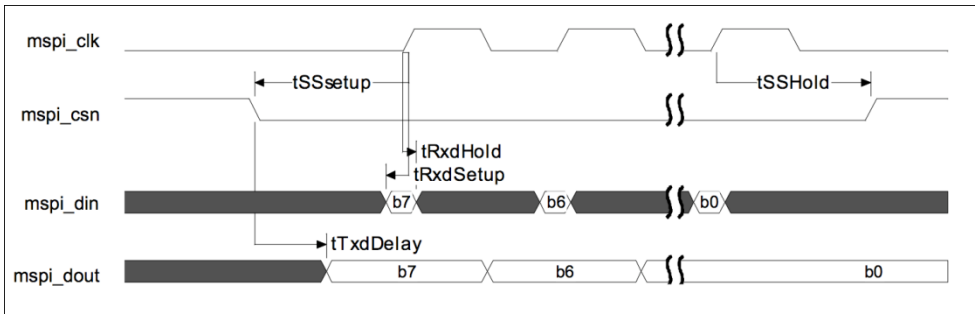


Figure 2-2: Timing Diagram, Master mode, SPO=SPH=0.

Parameter	Description	Min.	Max.	Unit
tSSetup	Minimum time between falling edge of Select line and first rising edge of SPI clock	1		MSPI clock period
tTxdDelay	Delay in Master asserting TX line after falling edge of Select line		2 core SPI clock periods + 3 nsec	mixed
tRxdSetup	Time before rising edge of SPI clock by which received data must be ready	30		nsec
tRxdHold	Time for which received data must be stable after rising edge of SPI clock	10		nsec
tSSHold	Time for which the Select line will be held after the sampling edge for the final bit to be transferred	1		MSPI clock period

Table 2-10: Timing Parameters, Master mode, SPO=SPH=0.

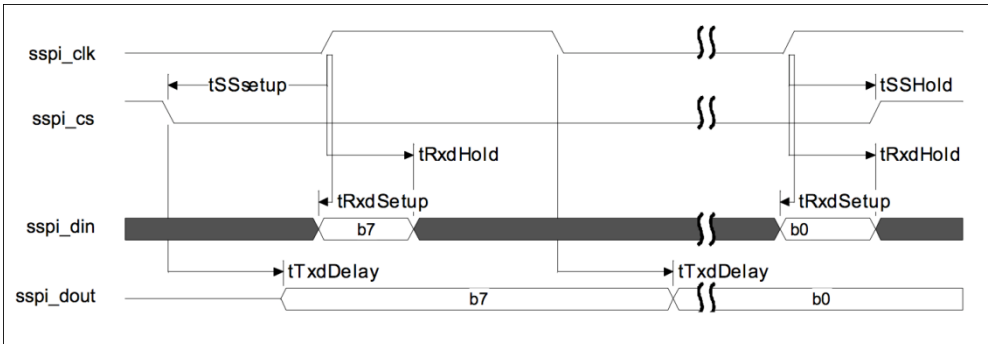


Table 2-11: Timing Parameters, Master mode, SPO=SPH=0.

Parameter	Description	Minimum	Maximum	Unit
tSSetup	Minimum time between falling edge of Select line and first rising edge of SPI clock.	4 core SPI clock periods + 68 ns		mixed
tTxdDelay	Delay in Slave asserting TX line after falling edge of SPI clock, or the first bit after falling edge of the Select line.		4 core SPI clock periods + 68 ns	mixed
tRxdSetup	Time before rising edge of SPI clock by which received data must be ready	15		ns
tRxdHold	Time for which received data must be stable after rising edge of SPI clock	3 core SPI clock periods + 14 ns		mixed
tSSHold	Time for which the Select line will be held after the sampling edge for the final bit to be transferred	3 core SPI clock periods + 14 ns		mixed

Table 2-12: Timing Parameters, Slave mode, SPO=SPH=0.

Motorola SPI, clock polarity SPO = 0, clock phase SPH = 1

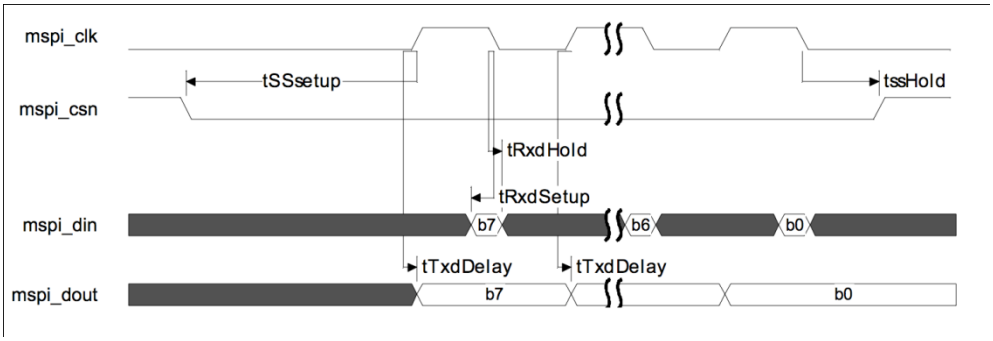


Figure 2-3: Timing Diagram, Master, SPO=0, SPH=1.

Parameter	Description	Min.	Maximum	Unit
tSSetup	Minimum time between falling edge of select line and first rising edge of SPI clock.	1.5		MSPI clock period
tTxdDelay	Delay in Master asserting TX line after rising edge of SPI clock.		0	ns
tRxdSetup	Time before falling edge of SPI clock by which received data must be ready.	30		ns
tRxdHold	Time for which received data must be stable after falling edge of SPI clock.	10		ns
tSSHold	Time for which the Select line will be held low after the sampling edge for the final bit to be transferred.	0.5		MSPI clock period

Table 2-13: Timing Parameters, Master mode; SPO=0, SPH=1.

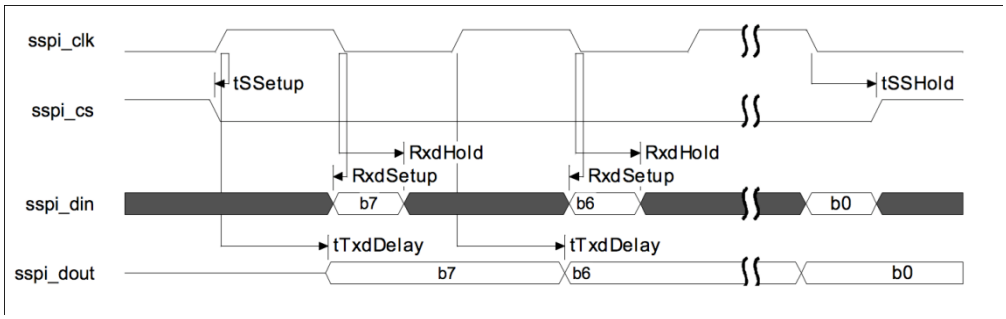


Figure 2-4: Timing Diagram, Slave, SPO=0, SPH=1.

Parameter	Description	Minimum	Maximum	Unit
tSSetup	Minimum time between falling edge of select line and first rising edge of SPI clock.	15		ns
tTxdDelay	Delay in Slave asserting TX line after rising edge of SPI clock.		4 core SPI clock periods + 68 ns	mixed
tRxdSetup	Time before falling edge of SPI clock by which received data must be ready.	15		ns
tRxdHold	Time for which received data must be stable after falling edge of SPI clock.	3 core SPI clock periods + 14 ns		mixed
tSSHold	Time for which the Select line will be held low after the sampling edge for the final bit to be transferred.	3 core SPI clock periods + 14 ns		mixed

Table 2-14: Timing Parameters, Slave mode, SPO=0, SPH=1.

Motorola SPI, clock polarity SPO = 1, clock phase SPH = 0

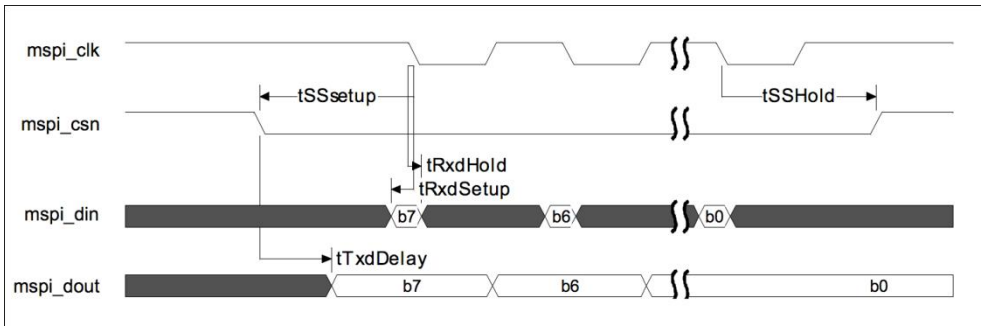
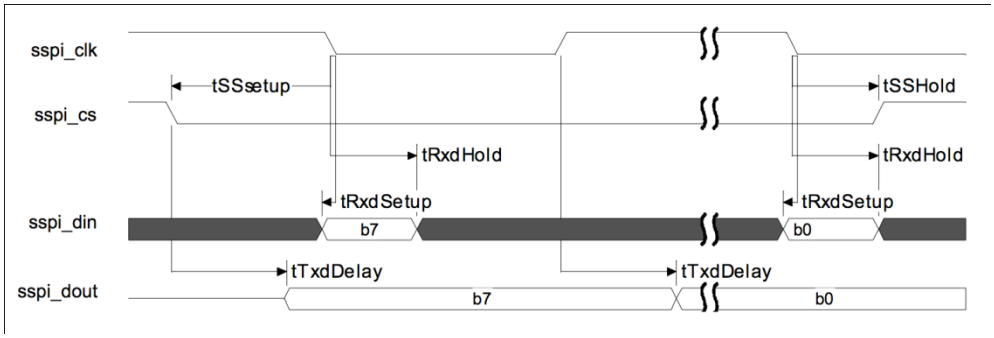


Figure 2-5: Timing Diagram, Master mode, SPO=1, SPH=0.

Parameter	Description	Minimum	Maximum	Unit
tSSetup	Minimum time between falling edge of select line and first falling edge of SPI clock.	1		MSPI clock period
tTxdDelay	Delay in Master asserting TX line after falling edge of Select line.		2 core SPI clock periods + 3 ns	mixed
tRxdSetup	Time before falling edge of SPI clock by which received data must be ready.	30		ns
tRxdHold	Time for which received data must be stable after falling edge of SPI clock.	10		ns
tSSHold	Time for which the Select line will be held low after the sampling edge for the final bit to be transferred.	1		MSPI clock period

Table 2-15: Timing Parameters, Master mode, SPO=1, SPH=0.



Parameter	Description	Minimum	Maximum	Unit
tSSetup	Minimum time between falling edge of Select line and first falling edge of SPI clock.	4 core SPI clock periods + 68 ns		Mixed
tTxdDelay	Delay in Slave asserting TX line after rising edge of SPI clock, or the first bit after falling edge of the Select line.		4 core SPI clock periods + 68 ns	Mixed
tRxdSetup	Time before falling edge of SPI clock by which received data must be ready.	15		ns
tRxdHold	Time for which received data must be stable after falling edge of SPI clock.	3 core SPI clock periods + 14 ns		Mixed
tSSHold	Time for which the Select line will be held low after the sampling edge for the final bit to be transferred.	3 core SPI clock periods + 14 ns		MSPI clock period

Table 2-16: Timing Parameters, Slave mode, SPO=1, SPH=0.

Motorola SPI, clock polarity SPO = 1, clock phase SPH = 1

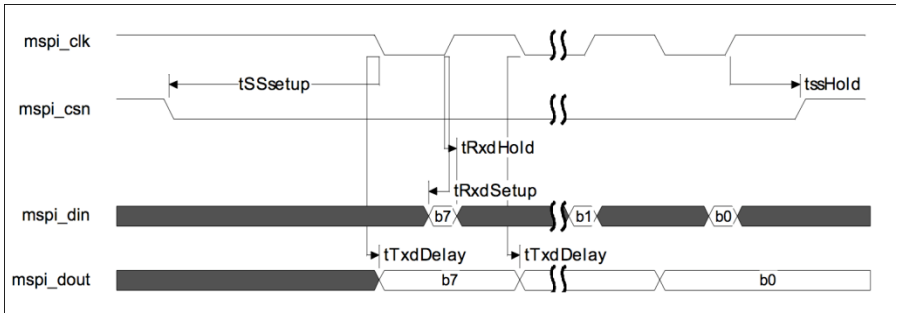


Figure 2-6: Timing Diagram, Master mode, SPO=SPH=1.

Parameter	Description	Minimum	Maximum	Unit
tSSetup	Minimum time between falling edge of select line and first falling edge of SPI clock.	1.5		MSPI clock period
tTxdDelay	Delay in Master asserting TX line after falling edge of SPI clock.		0	ns
tRxdSetup	Time before rising edge of SPI clock by which received data must be ready.	30		ns
tRxdHold	Time for which received data must be stable after rising edge of SPI clock.	10		ns
tSSHold	Time for which the Select line will be held low after the sampling edge for the final bit to be transferred.	0.5		MSPI clock period

Table 2-17: Timing Parameters, Master mode, SPO=SPH=1.

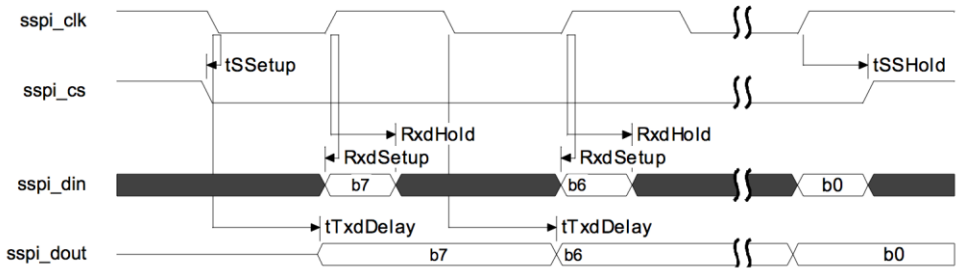


Figure 2-7: Timing Diagram, Slave mode, SPO=SPH=1.

Parameter	Description	Min.	Max.	Unit
tSSetup	Minimum time between falling edge of select line and first falling edge of SPI clock.	15		ns
tTxdDelay	Delay in Slave asserting TX line after falling edge of SPI clock.		4 core SPI clock periods + 68 ns	Mixed
tRxdSetup	Time before rising edge of SPI clock by which received data must be ready.	15		ns
tRxdHold	Time for which received data must be stable after rising edge of SPI clock.	3 core SPI clock periods + 14 ns		Mixed
tSSHold	Time for which the Select line will be held low after the sampling edge for the final bit to be transferred.	3 core SPI clock periods + 14 ns		Mixed

Table 2-18: Timing Parameters, Master mode, SPO=SPH=1

Chapter 3
AT Command Reference

Interface Architecture

The overall architecture of the WLM10x interface is depicted in the following figure titled Overall Architecture of the Module. Tx and Rx Data Handlers pass messages to, and from, the TCP/IP network. Commands related to management of the WLM10x interface and the network connections are intercepted by a Command Processor. A Serial Data Handler translates data to and from a UART/SPI-compatible format.

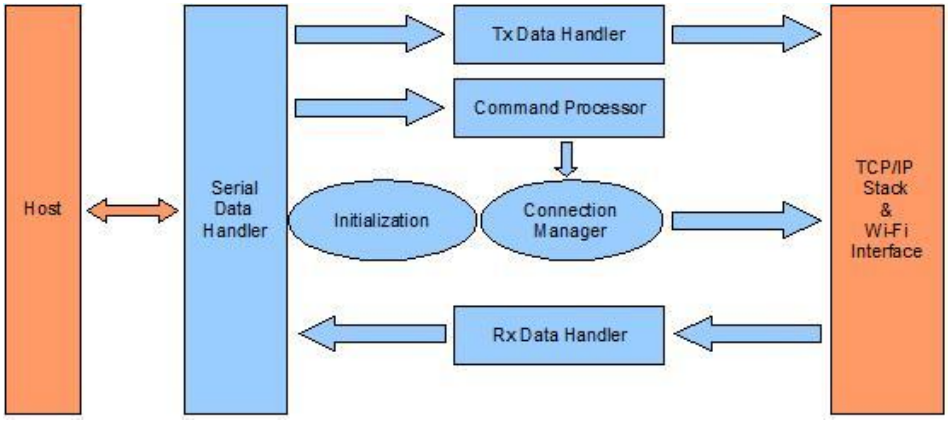


Figure 3-1: Overall Architecture of the Module

The system is composed of the following modules:

- System Initialization
- Command Processor
- Data Handlers
- Serial Data Handler
- Network Connection Manager
- Wireless Connection Management

The software for the WLM10x is mainly driven using a state machine. Upon powering on, the required initialization of all the modules is performed and then the state machine is entered. This state machine is event-driven and processes the events received from either the serial port or from the Wi-Fi / Network interface as well as internal events from its own modules. The state machine calls the appropriate handler for a given event per the current state.

The WLM10x has three distinct operating modes (see the following figure titled Operating Modes of the Module). In the default ***command processing operating*** mode, commands to configure and manage the interface are sent over the serial interface. In the default mode, the node accepts commands entered by the Host CPU and processes each of the commands. All commands are available in this mode. The User can establish a data connection here and send data.

In ***auto connection*** mode, data sent over the serial interface is transparently sent over the IP network to a single, pre-configured IP address/port pair, where data from that address is transparently sent over the UART/SPI to the serial host. With Auto mode, the IP Layer connections are already established and the data is sent directly to the target destination. In this mode, the node does not accept all commands. To accept commands the node needs to be brought back to “Command Processing” mode by pressing an escape sequence.

In ***data processing*** mode, data can be sent to, or received from, any of 16 possible connections. Each connection consists of a TCP or UDP path to a destination IP address and port. Auto connection mode is entered using a serial command and terminated using a special escape sequence.

For each mode, configuration parameters are stored in non-volatile memory. In addition to factory-default parameter values, two user-defined profiles (0 and 1) are available. The parameter set to be used is determined by a user command.

Module Description

System Initialization

Upon startup, the WLM10x interface performs the following actions, depicted graphically in the following figure titled Operating Modes of the Module.

- During the initialization process, the module will search for a saved configuration file. The configuration file include the auto connection settings, default profile and profile settings. If a saved configuration file is available, it is loaded from non-volatile memory. If no saved configuration file, the default settings will be applied. If there are no saved parameters, the factory-default configuration is loaded.
- The WLM10x application is initialized based on the profile settings.
- If auto connection is enabled, the interface will attempt to associate with the specified network, previously set by the user. Once associated, it will establish a TCP or UDP connection within the specified parameters. If successful, the interface will enter the Auto Connect mode, where all data received on the serial port is transmitted to the network destination and vice versa.
- If auto-connection is disabled or fails, the interface enters the command processing state.

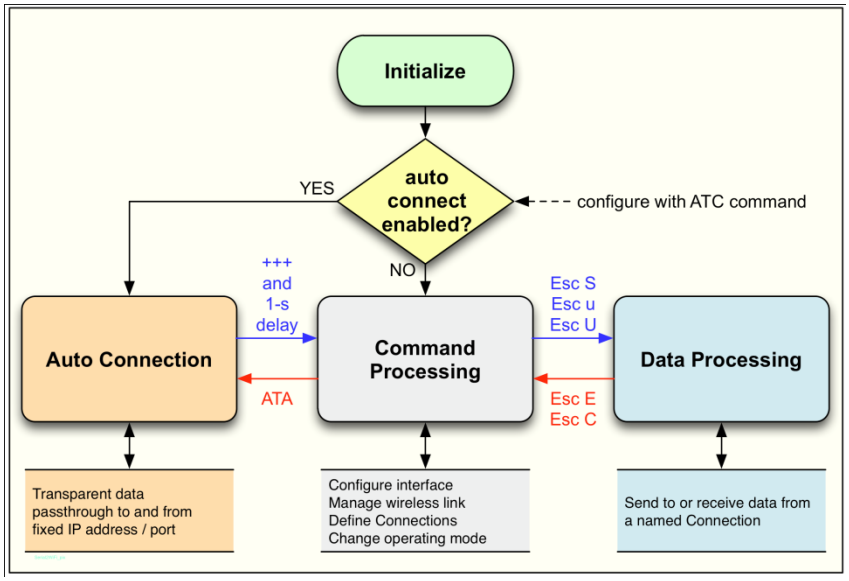


Figure 3-2: Operating Modes of the Module

Upon power-up, the UART interface defaults to 9600 baud, using 8 bit characters with no parity bits and one stop bit. Similarly SPI interface defaults to Mode#0 (CPL=0, CPH=0) Any changes to this configuration that were made in a previous session using the `ATB` command (UART Parameters) will be lost when power is lost. To make changes in the UART/SPI parameters that will persist across power cycling, the relevant changes must be saved into the power-on profile using `AT&W` (Save Profile) and `AT&Y` (Selection of Default Profile).

External PA Auto Detection

Upon startup, the WLM10x interface performs an auto detection of External PA. This detection is done through the GPIO pin 12. If this GPIO is “high” during startup, meaning the external PA is present; the module enables the external PA and forces the module to go into and out of standby mode for a moment just to make any changes effective for the external PA configuration.

Network Configurations

Once associated, the module supports instances of four types of network entities: TCP client, TCP server, UDP client and UDP server. Each client, or server, is associated with one or more of a possible 16 ***Connection Identifiers***, where the CID is a single hexadecimal number. More than one such entity can exist simultaneously; and a TCP server can have multiple connections, each with its own CID. When the module is in Auto Connect mode (Auto Connection), the entity called for by the Profile is created automatically upon startup. In Command modes, servers and clients are created using specific serial commands (Connection Management Configuration).

A TCP client (Figure titled Creation and Use of a TCP Client) is created with the serial command `AT+NCTCP` (TCP Clients). The client attempts to create a TCP network connection with the destination IP address and port specified within the command. If successful, it issues a CONNECT response with the CID of the client. Data can then be sent to the remote server using the `<Esc>S` sequence (Data Handling) with the appropriate CID. Data from the server is passed back to the Host, with the CID to identify its source.

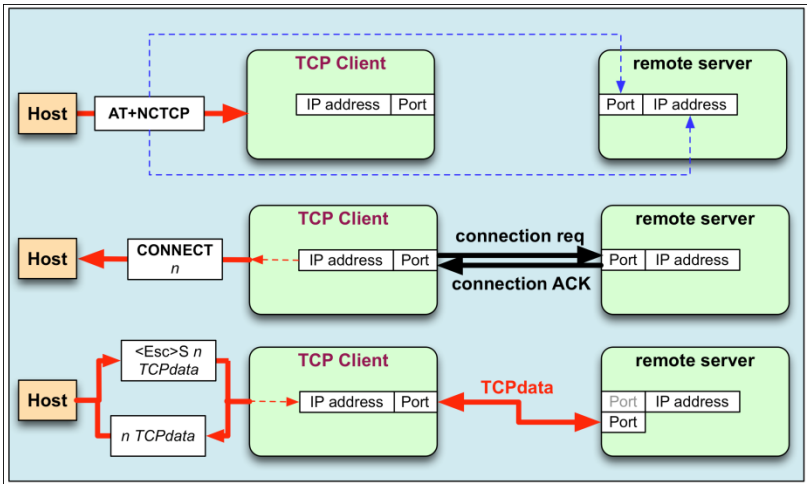


Figure 3-3: Creation and Use of a TCP Client

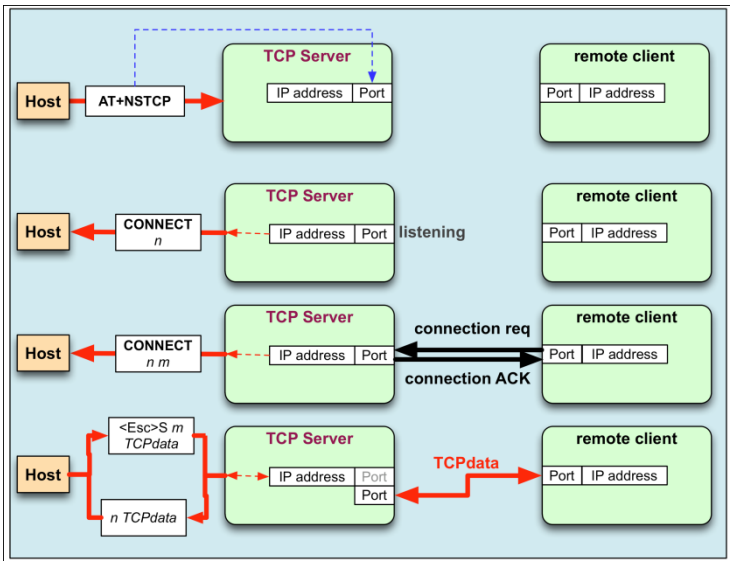


Figure 3-4: Creation and Use of a TCP Server

The figure titled Creation and Use of a TCP Server schematically depicts the corresponding sequence for a TCP server. A server is created with the serial command `AT+NSTCP`; it receives a CID, but listens passively until a remote client requests a connection.If that connection is successfully created, a second `CONNECT` message and a new CID are provided to the Host. It is this second CID that is used to send data to the remote client and identify received data from that client. A TCP server may support multiple clients, each with a unique CID.

A UDP client's life is depicted in the figure titled Creation and Use of a UDP Client. The client is created with the serial command `AT+NCUDP` and receives a CID. The UDP client is associated with a specific destination port and address.

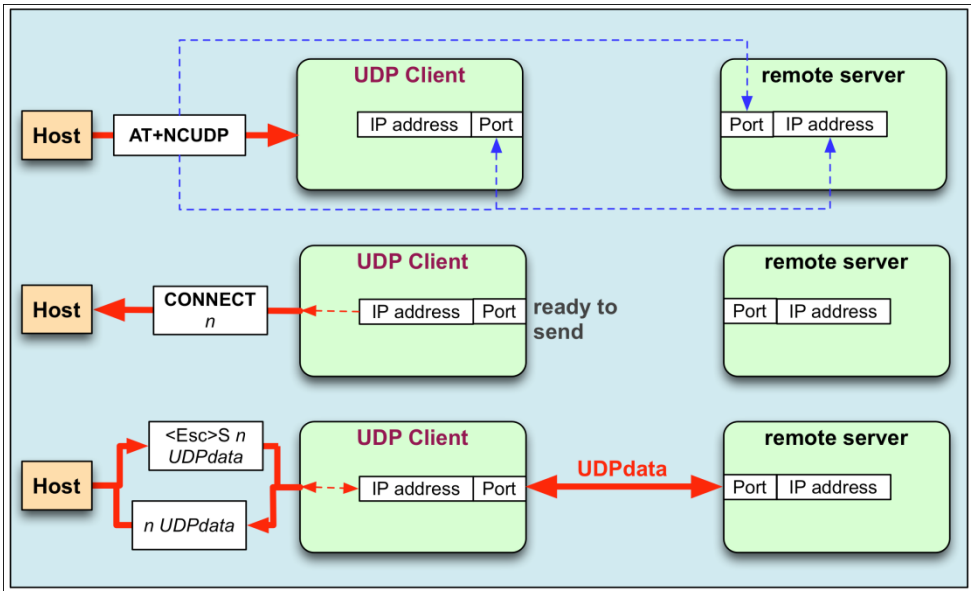


Figure 3-5: Creation and Use of a UDP Client

Finally, the figure titled Creation and Use of a UDP Server shows a UDP server. The server is created with `AT+NSUDP` and is assigned a CID. Individual clients do not receive unique CIDs; data sent using the UDP server must be accompanied with the destination IP address and port, and data received via the server is modified with the identifying source address and port number.

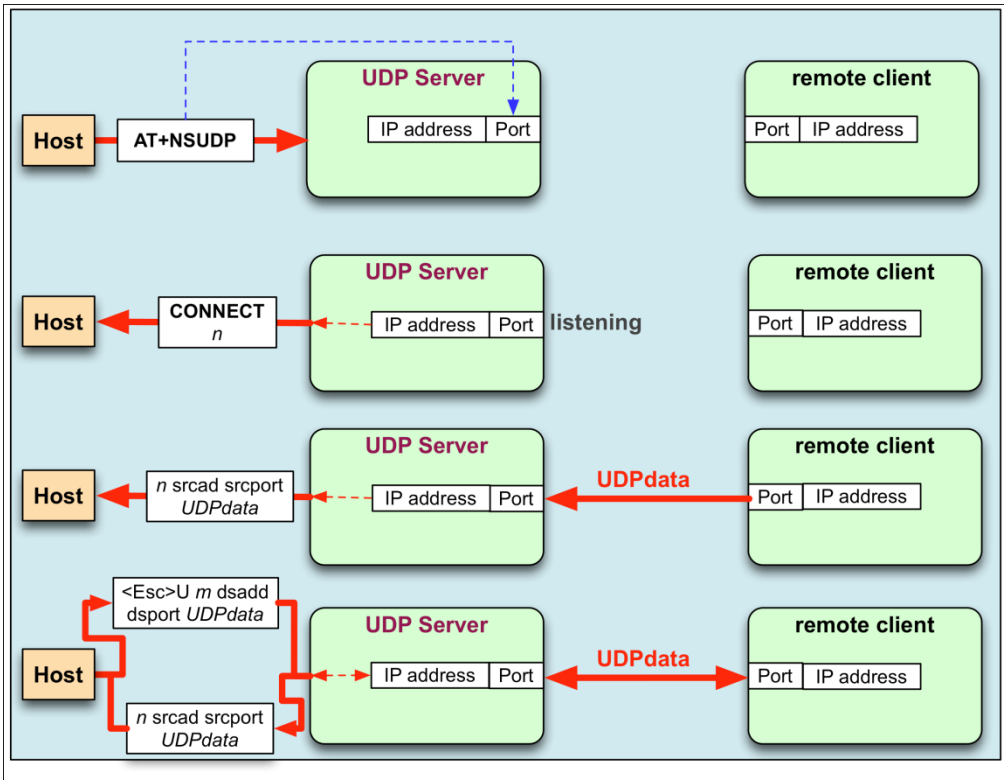


Figure 3-6: Creation and Use of a UDP Server

Profile Definition

The configuration parameter values that define the behaviour of the Module are grouped into Profiles. These profiles are stored in non-volatile memory when not in use. The default configuration supports two Profiles. The contents of a profile are listed in the following table titled Profile Parameters.

Parameter	Values	Reference
General Wireless Parameters		
802.11 Operating Mode	BSS, IBSS, Limited AP	Mode
Transmit Power Configuration		Transmit Power
802.11 Transmit Retry Count		Set Retry Count
Power Save Mode	Enabled, Disabled	Enable/Disable 802.11 Power Save Mode
802.11 Radio Mode	Enabled, Disabled	Enable/Disable 802.11 Radio
Auto Connect Mode, Wireless Interface Settings		
802.11 Operating Mode	BSS, IBSS	Wireless Parameters
Operating Channel	1 to 14	Wireless Parameters
SSID Parameter	Any valid SSID	Wireless Parameters
BSSID Parameter	Any valid BSSID	Wireless Parameters
Maximum Scan Time		Serial to WI-FI Configuration
Auto Connect Mode, Network Interface Settings		
Mode	Server, Client	Network Parameters
Protocol	TCP, UDP	Network Parameters
Server Port Number	Any valid port	Network Parameters
Server IP Address	Any valid IP address	Network Parameters
Wireless Interface Security Configuration		
Authentication Mode	Open, Shared	Authentication Mode
PSK Valid	Valid, Invalid	WPA-PSK and WPA2-PSK Key Calculation
PSK-SSID	Any valid SSID; used for PSK key computation.	WPA-PSK and WPA2-PSK Key Calculation
WEP Key Configuration		WEP Keys
WPA Passphrase		WPA-PSK and WPA2-PSK Passphrase
TCP/IP Configuration		
DHCP Mode	Enabled, Disabled	DHCP Support
IP Address	Valid IP address	Static Configuration of Network Parameters
Net Mask Address	Valid mask	Static Configuration of Network Parameters
Default Gateway Address	Valid IP address	Static Configuration of Network Parameters
DNS1	Valid DNS1 IP address	Static Configuration of DNS (Client)
DNS2	Valid DNS2 IP address	Static Configuration of DNS (Client)
UART Configuration		
Echo Mode	Enabled, Disabled	Echo

Verbose Mode	Enabled, Disabled	Verbose
Bits Per Character	5,6,7,8	UART Parameters
Number of Stop Bits	1,2	UART Parameters
Parity Type	No, Odd, Even	UART Parameters
Software Flow Control Mode	Enabled, Disabled	Software Flow Control
Hardware Flow Control Mode	Enabled, Disabled	Hardware Flow Control
Baud Rate		UART Parameters
Limits and Timeouts		
Network Connection Timeout	Units of 10 milliseconds	Serial to WI-FI Configuration
Auto Association Timeout	Units of 10 milliseconds	Serial to WI-FI Configuration
TCP Connection Timeout	Units of 10 milliseconds	Serial to WI-FI Configuration
Association Retry Count		Serial to WI-FI Configuration
Nagle Wait Time	Units of 10 milliseconds	Serial to WI-FI Configuration
SPI Configuration		
SPI clock polarity and clock phase	0,1	SPI Parameters

Table 3-1: Profile Parameters

Command Processing Mode

In command mode, the application receives commands over the serial port. Commands are processed line by line. “Verbose Mode”, when referring to commands being executing, refers to the displaying of status of any command executed in ASCII (human readable) format. When the verbose mode is disabled, the output will simply be in numeric digits, each digit indicating a particular status. Verbose Mode is enabled by default.

- If “echo” is enabled then each character is echoed back on the serial port
- Each command is terminated with a ***carriage return*** <CR> or ***line feed*** <LF>
- Each response is started with a carriage return <CR> and line feed<LF>, with the exception of the responses to the following commands:
    a) The response to the following group of commands starts with a line feed <LF> only:
        - AT+WA
        - AT+NSTAT
        - AT+WPAPSK=<SSID>,<Passphrase>
        - AT+NSET=<IP Address>,<Subnet Mask>,<Gateway IP Address>
        - AT+TRACEROUTE=<IP Address>
        - AT+PING=<IP Address>
        - ATA
        - AT+NDHCP after association
    b) The response to the following group of commands starts with a line feed and carriage return: <LF><CR>.
        - AT+SETTIME=<dd/mm/yyyy>,<hh:mm:ss>
        - AT+HTTPOPEN=<IP Address>
- Each response is terminated with a carriage return <CR> and line feed <LF>

If the characters “A” and “/” are entered at the beginning of a line (after <CRLF>), then the previous command is executed

Once a complete line (ending with <CR or LF>) is entered, then the command contained therein is processed and an appropriate response returned

Unless otherwise specified, if verbose mode is enabled, then the response to a successful command is the characters “OK”. The response to an unsuccessful command is the word “ERROR”, followed by a detailed error message, if available. If verbose mode is disabled, command responses is numerical with OK having a value of 0 and error codes represented by positive integers.

The commands are described under the paragraph heading Commands for Command Processing Mode in this chapter. Possible response codes are described under the paragraph heading Response Codes in this chapter.

Auto Connection

If auto connection is enabled, then upon startup the Module will:

Attempt to associate to or from the specified network, for a maximum time of *Auto Associate Timeout* (see Serial to WI-FI Configuration in this chapter). On successful association, attempt to establish a network connection based on the specified parameters

On successful connection establishment, enter the pass-through auto connect mode

On failure, enter the command processing state

In TCP client mode, the connection is considered established only when the client successfully connects to the server specified in the parameters. The client address may be fixed or obtained from a DHCP server. The client port is selected at random during creation of the client. The connection is attempted for a maximum time based on the *Network Connection Timeout,* specified in units of 10 milliseconds (see Serial to WI-FI Configuration in this chapter). Data is sent to, and received from, this server. If the connection is terminated, auto-connect mode also terminates and the command processing state is entered.

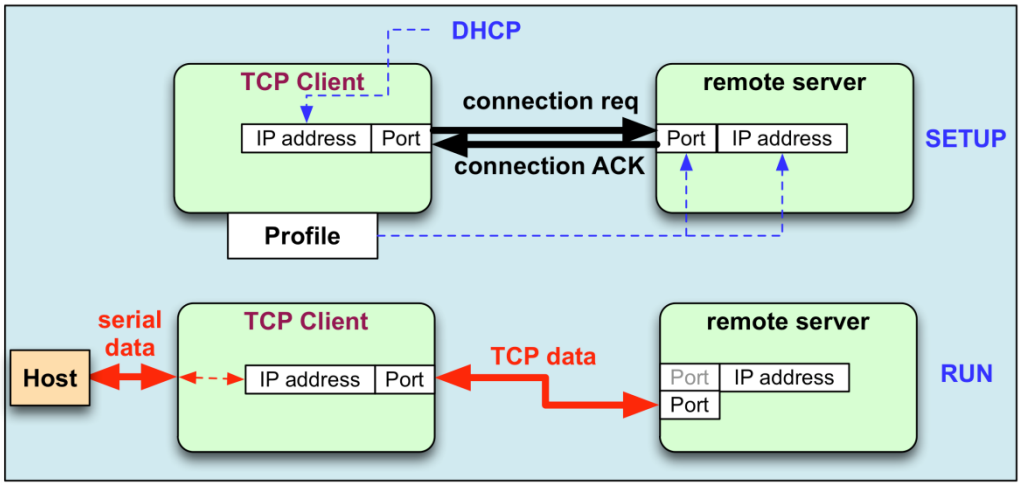


Figure 3-7: TCP Client Operation in Auto Connect Mode

The TCP server IP address may be fixed in the profile or obtained from DHCP. The port for connection attempts to be made is obtained from the profile. In TCP server mode, the connection is considered established when the first client connects to the server. Data is sent to, and received from, this client. If the client disconnects, the module waits for the next client to connect.

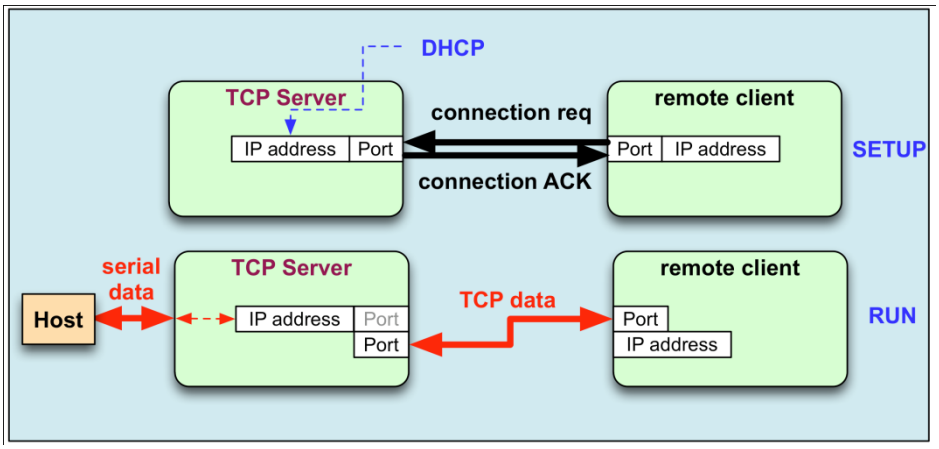


Figure 3-8: TCP Server Operation in Auto Connect Mode

In UDP client mode, the connection is considered established when the client is created. The client IP address may be fixed or obtained from DHCP. The client port number is set at random upon creation of the client. Data is sent to and received from the configured server.

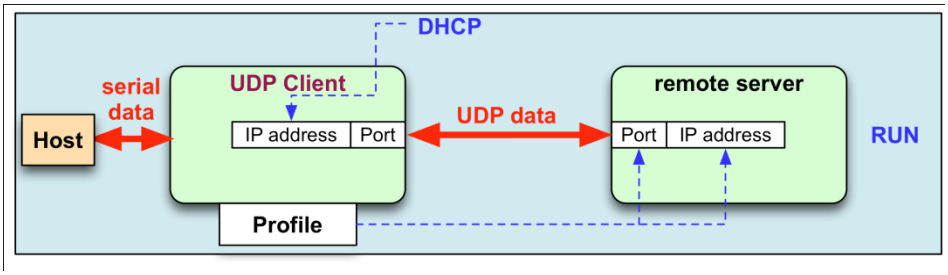


Figure 3-9: UDP Client Operation in Auto Connect Mode

In UDP server mode, the connection is considered established when data is received from any client. The UDP server IP address may be fixed or obtained by DHCP. The port is set by the profile. Data received from any client is output on the serial port and data received on the serial port is transmitted to the client based on the last packet was received.

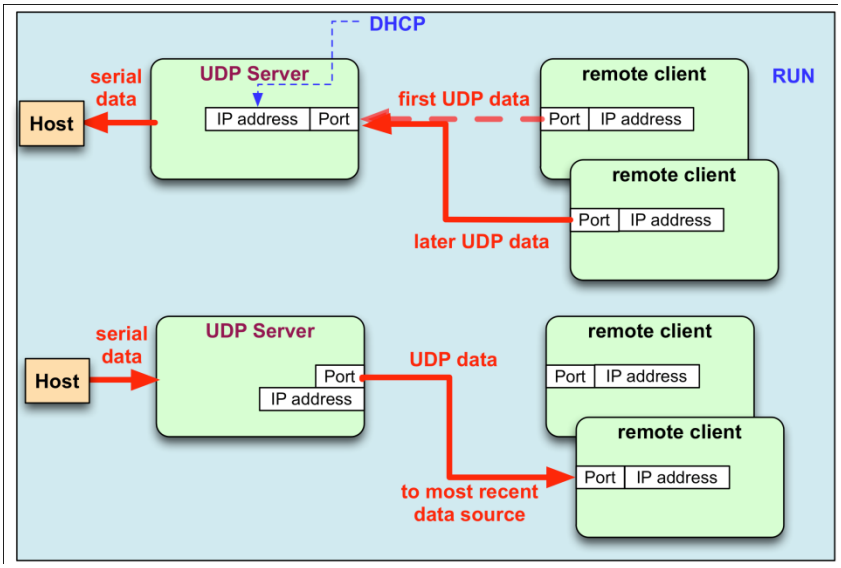


Figure 3-10: UDP Server Operation in Auto Connect Mode

In TCP and UDP server mode, even where no connection is established, the serial host may take control of the WLM10x interface by issuing a specific escape sequence, described in the following paragraph heading Auto Connection Operation.

Auto Connection Operation

Auto Connect Mode acts as a cable replacement so that the interface acts like a serial interface and no commands or user intervention are required for connection management. The node automatically establishes the wireless and network connections by using parameter values from the current active Profile and transfers data transparently between the Host and Target in data mode. No status information is sent to the Host.

In auto connection mode the Module:

Receives characters from the serial port and transmits them over the Wi-Fi connection

Receives data from the Wi-Fi connection and transmits it on the serial port

The serial host may gain control of the interface by issuing the ***escape sequence*** "+++", followed by a one-second gap where no characters are received on the serial port. When this sequence is encountered, the Module suspends auto connection mode and resumes command processing. The Host then may make changes in the network configuration or other parameters as needed. However, the Module does not accept any new TCP/UDP client/server or auto connection requests since auto connection exists in the background. The ATO command (terminated by the ASCII character "O", not the number 0) is used to return to auto connection mode.

In auto connection mode, the Nagle Algorithm Wait Time (paragraph heading Serial to WI-FI Configuration in this chapter) can be used to buffer any characters to be sent, in order to avoid sending a large number of packets with small payloads onto the network. The wait time is specified in units of 10 milliseconds. This functionality is available for both UDP and TCP connections.

Data Handling

In Data Processing Mode, data transfers are managed using various ***escape sequences***. Each escape sequence starts with the ASCII character 27 (0x1B); this is equivalent to the ESC key. The encoding of data and related commands are described in the following pages. This encoding is used for both transmitted and received data.

The network destination, or destination source, for a given data packet is established by means of a ***Connection Identifier***, and represented as a single hexadecimal number. Data is transferred on a per CID basis. Data is normally buffered until the end-of-data escape sequence is received. However, if the amount of data exceeds the size of the data buffer, the data received, thus far, is sent immediately. The data buffer size depends on the implementation, but is usually one MTU (1400 bytes).

The process of sending a data packet is depicted in the following figure titled Data Processing Flow. The sequence Esc S or Esc U is sent to initiate the data transfer. This sequence is followed by a single-digit CID; if the CID is valid, the subsequent characters are assembled into a data stream, terminated by Esc E, Esc C, Esc S or Esc U. With a terminating sequence, the data is sent via the requested network connection and the system either returns to command processing or to further data processing.

Escape sequences like Esc S, Esc u and Esc U support only ASCII data handling while Esc Z, Esc Y and Esc y supports all types of data (ASCII, Binary etc.) handling.

Please refer to Chapter 4 for a complete description of all the Escape sequences used for data handling.

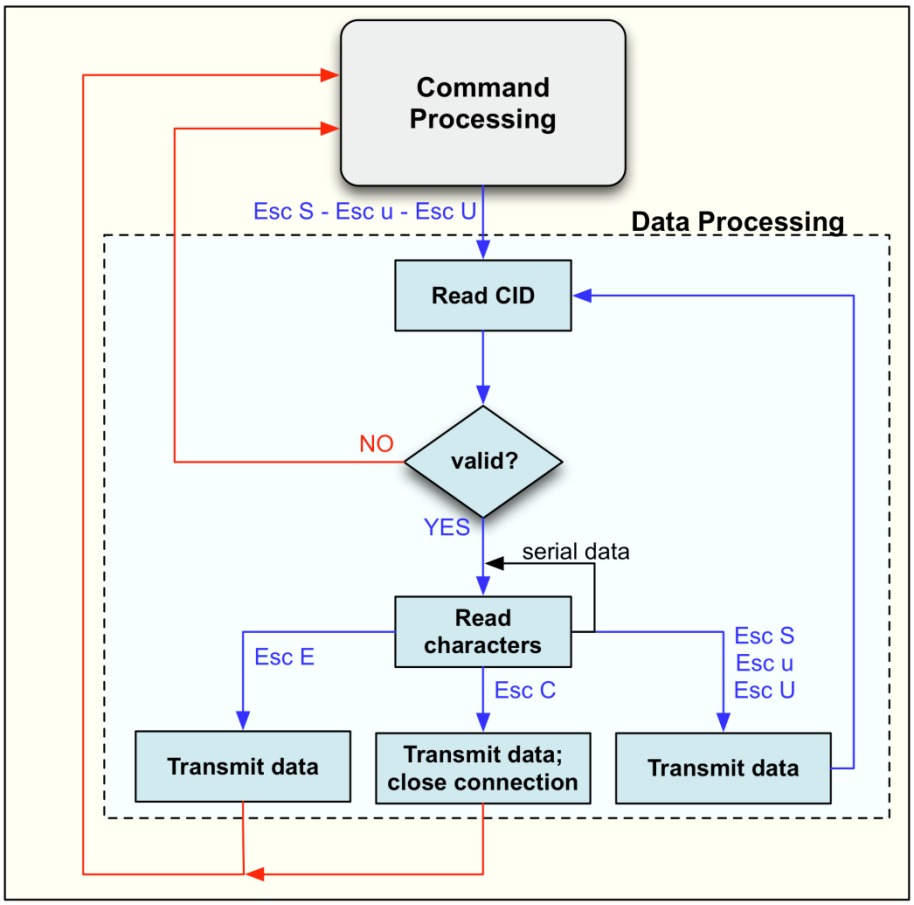


Figure 3-11: Data Processing Flow

Operation	Escape Sequence	Description
Send and Return to Command Mode Sequence	<Esc>C	This sequence causes transmission of the data received on the serial interface on a TCP server/client or UDP client connection. After, the currently selected connection is closed and the interface returns to Command Mode. Any buffered data is sent before the connection is closed. This can be issue from the serial host once the data transmissions start on a socket using <Esc>S<CID> sequence.
Success Indication	<Esc>O	"OK": This sequence is sent to the serial host by the WLM10x Module upon successful completion of the <Esc>S<CID>, <Esc>E, <Esc>U<CID> or <Esc>C commands.
Failure Indication	<Esc>F	"FAILURE": This sequence is sent to the host by the WLM10x Module if an <Esc>S, <Esc>E, <Esc>U, or <Esc>C command failed.

The contents of < > are either a byte or byte stream, except for <Esc>; literals outside brackets are ASCII characters.

Table 3-2: Data Handling Responses At Completion

Bulk data Tx and Rx

In Bulk Data Mode, data transfers are managed using ***escape sequences* (*Esc Z, Esc Y and Esc y*)**. Each escape sequence starts with the ESC key (ASCII character 27 (0x1B)). Encoding is used for both transmitted and received data. Enable bulk data by using command "AT+BDATA=" (1 is enable and 0 is disable).

The format of a bulk data frame for TCP client, TCP server, or UDP client is:

<Esc>Z<CID><Data Length xxxx 4 ascii char><data>

The contents of < > are a byte or byte stream.

CID is connection identifier (UDP, TCP, etc.; as derived when TCP socket is created by issuing the command: AT+NCTCP, for example.)

Data Length is 4 ascii char represents decimal value i.e. 1400 byte (0x31 0x34 0x30 0x30).

The Data Length range should be 1 to 1400 bytes.

User Data size **must match** the specified Data Length. Ignore all command or esc sequence in between data pay load. User should send the specified length of data to the module irrespective of any asynchronous events happened on the module so that the module can start receiving next commands.

For example, if CID value is 3, then:

To send a 5 byte user data (e.g. ABCDE) for a TCP client connection, the format will be:

<ESC>Z30005ABCDE

To send a 512 byte user data for a TCP client connection, the format will be:

<ESC>Z30512<512 bytes of user data>

To send data on UDP server, the bulk data frame format is:
<Esc>Y<CID><Ip address>:<port>:<Data Length xxxx 4 ascii char><data>

When receiving data on UDP server, the format of a bulk data frame is:
<Esc>y<CID><IP address><space><port><horizontal tab><Data Length xxxx 4 ascii char><data>

Operation	Escape Sequence	Description
Bulk Data transfer on TCP Server/Client and UDP Client connection	<Esc>Z<CID>Data Len 4 digit ascii<Data>	To improve data transfer speed, one can use this bulk data transfer. This escape sequence is used to send and receive data on a TCP Client/Server and UDP client connection. Example: <Esc>Z40005Hello where 4 is the CID, 0005 is the 5 byte data length and Hello is the data to be sent.
Bulk Data Send on UDP sever connection	<Esc>Y<CID> remote address: remote port:Data Len 4 digit ascii<Data>	This escape sequence is used when sending UDP data on a UDP server connection. When this command is used, the remote address and remote port is transmitted in ASCII text encoding and terminated with a ':' character. Example: <Esc>Y4192.168.1.1:52:0005Hello where 4 is the CID, 0005 is the 5 byte data length and Hello is the data to be sent.
Bulk Data Receive on UDP Server Connection	<Esc>y<CID> remoteaddress<space>remote port<horizontal tab>Data length in 4 digit ascii<Data>	This escape sequence is used when receiving UDP data on a UDP server connection. When this sequence is used, the remote address and remote port is transmitted in ASCII text encoding and separated be a space ( ) character. Example: <Esc>y4192.168.1.1<space>52<horizontal tab>0005Hello where 4 is the CID, 0005 is the 5 byte data length and Hello is the data received.

The contents of < > are either a byte or byte stream, except for <Esc>; literals outside brackets are ASCII characters.

Table 3-3: Escape Sequences

Raw Data Handling (BACNET Support Only)

In Raw Data Mode, data transfers are managed using ***escape sequences***. Each escape sequence starts with the ASCII character 27 (0x1B), the equivalent to the ESC key. The encoding of data is described below. Encoding is used for both transmitted and received data. The Raw Ethernet Support Enable command (paragraph heading Enable/Disable Raw Ethernet Support in this chapter) must be issued before sending or receiving raw data through the Module.

The format of a raw-data frame is:
<Esc>:R:<Length>:<DstAddr><SrcAddr><EtherType><Raw-Payload>

The contents of < > are a byte or byte stream.

Length is the size of DstAddr, SrcAddr, EtherType and Raw-Payload

DstAddr is the destination MAC address

SrcAddr is the source MAC address

EtherType is the type of the Ethernet packet. For example, for BACNET-over-Ethernet, EtherType is 0x0000.

Raw-Payload is the raw data

Unsolicited Data Handling

In Unsolicited Data Mode (data transmission without association), data transfer is managed using ***escape sequences***. Each escape sequence starts with the ASCII character 27 (0x1B), equivalent to the ESC key. The encoding of data is described below. This encoding is used for transmitted data only. The unsolicited data transmission Enable command (paragraph heading Enable/Disable Raw Ethernet Support in this chapter) must be issued before sending unsolicited data through the Module.

The format of an unsolicited data frame is:

<ESC>D/d<PayLoad>

The PayLoad contents are byte or byte stream.

Software Flow Control

Software flow control works only with ASCII data transfers and cannot be used for binary data.

If software flow control is enabled, and the interface receives an XOFF character from the serial host, it stops sending to the host until it receives an XON character. If the Module is receiving data over the wireless connection during the time that XOFF is enabled, it is possible for the wireless buffer to become full before XON is received. In such a case, data from the network will be lost.

If software flow control is enabled, then the interface sends an XOFF character to the host when it will be unable to service the serial port. The XON character is sent when the interface is once again able to accept data over the serial port.

Note: With initialization, the Module treats the serial channel as clear with no restrictions on data transmission or reception; no explicit XON is transmitted by the Module or required from the Host, even if flow control is enabled.

Hardware Flow Control

The Hardware Flow control is a handshake mechanism between the Serial host and WLM10x module on UART interface, using two additional CTS and RTS connections. This feature prevents the UART hardware FIFO overflow on WLM10x module due to high speed data transmission from/to the WLM10x module. If hardware flow control is enabled, an RTS/CTS handshake will occur between the serial host and the Module. This is a hardware feature and available only for UART interface.

The WLM10x module uses both CTS and RTS signals as “low” to indicate the readiness to send or receive data from serial host.

Serial Data Handling

The Serial Data Handler receives and transmits data to and from the hardware serial controller. Data read from the serial port is passed to:

- The command processor in command mode
- The Tx data handler in data mode
- The auto connection mode processor for data transfer in auto connection mode

Then Data is transferred on the serial port from:

- The command processor in order to output responses to commands
- The Rx data handler in order to output incoming packets
- The auto connection handler in order to output incoming data
- The connection manager in order to output status indications
- The wireless connection manager in order to output status indications

When configured in Auto Connection Mode, the Module enters directly into Data Processing Mode after the completing the connection without sending any status information to the Host.

Connection Management

The connection management module is responsible for processing connection-related events. The interface provides UDP and TCP sockets (similar to the familiar BSD network sockets). Each socket may represent either a server or client connection. Each connection has a unique, single-digit hexadecimal value (0 to F), for the CID. The allowed maximum number of connections (up to 16) may be specified at compile time. Note that this single pool of CID's is used for TCP, UDP, Server and Client connections.

Packet Reception

When a packet is received on any open connection, and the application is not currently in auto-connect mode, the packet is transferred on the UART/SPI in the form described in paragraph heading Data Handling above in this chapter. Received data payloads are encoded with the appropriate Escape sequence. The connection ID is used to inform the serial host of the origin of an IP data packet. The source IP address and port are provided along with the data when a UDP packet is received.

If auto-connect mode is enabled and a packet is received on the auto-connected CID, the packet data is sent without modification over the UART/SPI to the serial host.

Remote Close

If a TCP connection is terminated by disconnection from the remote end, an unsolicited ASCII-format response of the form `DISCONNECT Connection ID` is sent to the serial host, and the specified CID should be considered unavailable. If the connection ends because the remote server has shut down, the unsolicited response `ERROR: SOCKET FAILURE Connection ID` will be sent to the host. Note that a data packet from the remote client or server containing the same ASCII characters `CLOSE Connection ID` is treated as data rather than a command and forwarded to the serial host.

TCP Server Connections

Upon deployment of incoming TCP connections on a socket, the incoming connection is allowed if the limit on the maximum number of connections has not been reached. There is an unsolicited response of the form `CONNECT <server CID> <new CID> <ip> <port>`, where:

Server CID is the CID of the server where the connection has arrived

New CID is the CID allocated for this client connections

IP and port of the client encoded in the binary encoding used for UDP server data packets described in paragraph heading Data Handling in this chapter is sent to the serial host. The host can use the IP address to ascertain the source of the TCP connection request. The TCP server has no timeout limitation for an incoming connect request. It waits indefinitely, until a CLOSE command is received.

Note that if Verbose mode is disabled (paragraph heading Verbose in this chapter), the word CONNECT in the unsolicited response is replaced by the number 7.

Wireless Network Management

Scanning

The WLM10x interface can instruct the Wi-Fi radio to scan for access points and ad hoc networks with a specified SSID, BSSID and/or channel for a specified scan time. Scanning can be performed to find networks with a specific SSID or BSSID, networks operating on a specific radio channel or a combination of these constraints.

Association

The WLM10x interface performs all the actions required to join an infrastructure IP network:

- Scan for a specific AP (AT+WS, paragraph heading Scanning)
- Authenticate the specified network using the configured authentication mode (AT+WAUTH, paragraph heading WI-FI Security Configuration)
- Associate to the AP (AT+WA, paragraph heading Associate with a Network, or Start an Ad Hoc or Infrastructure (AP) Network)
- Perform security negotiation if required
- Change state to Wireless Connected
- Initialize the networking stack using the configured static IP address or via DHCP (AT+NDHCP, paragraph heading DHCP Support)

In ad hoc mode, the interface can:

- Scan for a specified Ad-hoc Network
- Join the ad hoc network, if it exists
- If the ad hoc network does not exist, create a new ad hoc network to join
- Perform security negotiation, if required
- Change state to Wireless Connected
- Initialize the networking stack using the configured static IP address or via DHCP

Response Codes

The possible responses sent by the Module to the serial host are enumerated in the following table.

No	ASCII CHAR	Response	ASCII STRING	Meaning
1	0	WLM10X_SUCCESS	"OK"	Command Request Success.
2	1	WLM10X_FAILURE	"ERROR"	Command Request Failed.
3	2	WLM10X_EINVAL	"ERROR: INVALID INPUT"	Invalid Command or Option or Parameter.
4	3	WLM10X_SOCK_FAIL	"ERROR: SOCKET FAILURE <CID>\r\n"	Socket Operation Failed.
5	4	WLM10X_ENOCID	"ERROR: NO CID"	All allowed CID's in use, so there was no CID to assign to the new connection.
6	5	WLM10X_EBADCID	"ERROR: INVALID CID"	Invalid Connection Identifier.
7	6	WLM10X_ENOTSUP	"ERROR: NOT SUPPORTED"	Operation or Feature not supported.
8	7	WLM10X_CON_SUCCESS	"\r\nCONNECT <CID>\r\n"	TCP/IP connection successful. <CID> = the new CID in hexadecimal format.
9	8	WLM10X_ECIDCLOSE	"\r\nDISCONNECT <CID>\r\n"	TCP/IP connection with the given CID is closed. This response is sent to the host when a connection is closed either by the remote device or by the serial host.
10	9	WLM10X_LINK_LOST	"DISASSOCIATED"	Not associated to a wireless network.
11	A	WLM10X_DISASSO_EVT	"\r\nDisassociation Event\r\n"	Wireless network association lost.
12	B	WLM10X_STBY_TMR_EVT	"\n\rOut of StandBy-Timer\r\n"	Wake up from Standby due to RTC timer expiration.
13	C	WLM10X_STBY_ALM_EVT	"\n\rOut of StandBy-Alarm\r\n"	Wake up from Standby due to receipt of an Alarm signal.
14	D	WLM10X_DPSLEEP_EVT	"\r\nOut of Deep Sleep\r\n"	Wake from Deep Sleep
15	E	WLM10X_BOOT_UNEXPECTED_EVT	"\r\nUnExpected Warm Boot(Possibly Low Battery)\r\n"	Unexpected reset. Possible reasons: external reset or low battery
16	F	WLM10X_ENOIP	"ERROR: IP CONFIG FAIL"	IP configuration has failed. This message also can come asynchronously when there is a DHCP renew fails.

Table 3-4: Response Codes

NO	Message	Subtype	Meaning
1	ERROR: SOCKET FAILURE <CID>	0	Socket Operation Failed
2	CONNECT <CID>	1	TCP/IP connection successful. <CID> = the new CID in hexadecimal format.
3	DISCONNECT <CID>	2	TCP/IP connection with the given CID is closed. This response is sent to the host when a connection is closed by the remote device.
4	Disassociation Event	3	Wireless network association lost.
5	Out of StandBy-Timer	4	Wake up from Standby due to RTC timer expiration.
6	Out of StandBy-Alarm	5	Wake up from Standby due to receipt of an Alarm signal.
7	Out of Deep Sleep	6	Wake from Deep Sleep.
8	UnExpected Warm Boot (Possibly Low Battery)	7	Unexpected reset. Possible reasons: external reset or low battery.
9	ERROR: IP CONFIG FAIL	8	IP configuration has failed. This message comes asynchronously when there is a DHCP renew fails.

Table 3-5 Enhanced Asynchronous Messages

Exception Messages

The possible exception messages sent by the Module to the serial host are enumerated in the following table.

No	ASCII String	Meaning
1	\n\rAPP Reset-Wlan SW Reset\r\n	Module reset due to WLAN processor software reset.
2	"\n\rAPP Reset-APP SW Reset\r\n"	Module reset due to app processor software reset...
3	\n\rAPP Reset-Wlan-Wd\r\n	Module reset due to WLAN processor watchdog.
4	\n\rAPP Reset-App-Wd\r\n	Module reset due to app processor watchdog
5	\n\rAPP Reset-Wlan Except\r\n	Module reset due to WLAN processor software abort or assert.
6	\n\rAPP Reset-FW-UP-FAILURE\r\n	Module reset due to firmware upgrade failure.
7	\n\rAPP Reset-FW-UP-SUCCESS\r\n	Module reset due to firmware upgrade success.
8	\n\rAPP Reset-FW-UP-RECOVERY\r\n	Module reset due to firmware upgrade failure with one of the flash image updated successfully.

Table 3-6 Exception Messages

If the exception is due to one of the WLAN wd/SW Reset/Except, then the module send memory dump information of its WLAN registers to the serial host starts with the message \r\n---MEM-DUMP-START:\r\n and end with the message \n\r---MEM-DUMP-END:\r\n.

Boot Messages

The possible boot messages sent by the Module to the serial host are enumerated in the following table.

No	ASCII String	Meaning
2	\ r\nSerial2WiFi APP-Ext.PA\r\n	Normal Serila2WiFi module boot message with external PA.
3	\r\n Factory Default CheckSum Error\r\n	The factory default section contains invalid data. This comes along with the above boot message.

Table 3-7 Boot Messages

SSID and PassPhrase

Rules:

The WLM10X module accepts the following ASCII characters for SSID and passphrase.

Category	Accepted Characters
Numerical	0-9
Alphabets	a-z and A-Z
Special characters	SP ! # $ % & ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~ ”

Note: SP = space

Table 3-8 Acceptable SSID and Passphrase Characters

1. The SSID or PassPhrase parameter may be captured within or without double quotation marks (“SSID”).
2. The quotation mark (“) may not be used as the first character of the SSID or passphrase.
3. If comma (,) is a part of the SSID, then SSID parameter needs to be framed with double quotation marks (“SS,ID”).

Expected SSID	Input SSID	Remarks
TEST	TEST	Valid (satisfies rule 2)
TEST	“TEST”	Valid (satisfies rule 2)
TE”ST	TE”ST	Valid (satisfies rule 3)
TE”ST	“TE”ST”	**Invalid** (*breaks rule 3*)
TE,ST	“TE,ST”	Valid (satisfies rule 4)
TE,ST	TES,T	**Invalid** (*breaks rule 4*)
TE,S”T	“TE,S”T”	**Invalid** (*breaks rule 3 and 4*)

Table 3-9 Valid SSID Format

Commands for Command Processing Mode

This section provides a list of WLM10x commands and their effects. Formatting and processing of commands was described in paragraph heading Command Processing Mode in this chapter. Parameters are generally ASCII characters, e.g. ATEn with n=1 is the series of ASCII characters ‘A’, ‘T’, ‘E’, and ‘1’. Where some parameters are optional, mandatory parameters are denoted by < > and optional parameters by [ ]. If a parameter is mandatory, any associated sub-parameters are also mandatory; sub-parameters of an optional parameter are optional. Parameters must always be provided in the order given in the command description. When an optional parameter is not supplied, the comma delimiters must still be included in the command. Every command starts with the characters “AT”; any other initial characters will cause an error to be returned.

Command Response: In most cases, valid commands return the characters `OK` if verbose mode is enabled and `0` verbose mode is not enabled. Invalid inputs return `ERROR: INVALID INPUT` if verbose is enabled and `2` if it is not. Exceptions to this rule are noted explicitly below.

Commands for Command Processing Mode

Command Interface

Interface Verification

The command AT can be issued to verify that the interface is operating correctly; it should return a successful response `OK` (or `0` if verbose mode is disabled).

Echo

Command: **ATEn**

Response: *If* n *is 0, echo is disabled and if* n *is 1, echo is enabled.*

Description: If echo is enabled, every character received on the serial port is transmitted back on the serial port. By default echo is enabled in WLM10x module.

Verbose

Command: **ATVn**

Response: *If* n *is 0, verbose responses is disabled.*

If n *is 1, verbose responses is enabled.*

Description: If verbose mode is disabled, the status response is in the form of numerical response codes. If verbose mode is enabled, the status response is in the form of ASCII strings. Verbose Mode is enabled by default.

UART Interface Configuration

UART Parameters

Command: **ATB=<baudrate>[[,<bitsperchar>][,<parity>][,<stopbits>]]**

Response: *OK*

Description: This is the command to set the UART Parameters. All standard baud rates are supported.

Allowed baud rates include: 9600, 19200, 38400, 57600, 115200, 230400,460800 and 921600.

Parity is *n* for no parity, *e* for even parity and *o* for odd parity.

Allowed values are 5, 6, 7 or 8 bits/character, with 1 or 2 stop bits

The new UART parameters take effect immediately. However, they are stored in RAM and will be lost when power is lost unless they are saved to a profile using `AT&W` (paragraph heading Save Profile in this chapter). The profile used in that command must also be set as the power-on profile using `AT&Y` (paragraph heading Selection of Default Parameters in this chapter).

Software Flow Control

Command: **AT&Kn**

Response: *If* n *is 0, software flow control is disabled.*

If n *is 1, software flow control is enabled.*

OK

Description: This command is used to configure software flow control. The use of software flow control is described in paragraph heading Software Flow Control in this chapter.

Hardware Flow Control

Command: **AT&Rn**

Response: *If* n *is 0, hardware flow control is disabled.*

If n *is 1, hardware flow control is enabled.*

OK

Description: The use of software flow control is described in paragraph heading Hardware Flow Control in this chapter.

SPI Interface Configuration

SPI Parameters

Set The SPI Clock Phase And Clock Polarity Parameter

Command: **AT+SPICONF=<clockpolarity>, <clockphase>**

Response: *If clock polarity is 0, then inactive state of serial clock is low.*

If clock polarity is 1, then inactive state of serial clock is high.

If clock phase is 0, then data is captured on the first toggling edge of the serial clock (clock phase zero), after the falling edge of slave select signal.

If clock phase is 1, then data is captured on the second edge of the serial clock (clock phase 180), after the falling edge of slave select signal.

Default is clock polarity 0 and clock phase 0.

OK

Description: The new SPI parameters take effect after node reset/restart. However, they are stored in RAM and will be lost when power is lost unless they are saved to a profile using AT&W (paragraph heading Save Profile in this chapter). The profile used in that command must also be set as the power-on profile using AT&Y (paragraph heading Selection of Default Profile in this chapter).

Serial to Wi-Fi Configuration

Command: **ATSn=p**

Response: n *is the parameter id to set and* p *is the value to set the parameter to. The parameters available are described in the following table.*

OK

Description:

Parameter ID	Name	Description	Citation
0	Network Connection Timeout	The maximum amount of time allowed establishing the network connection in Auto Connect Mode. Measured in units of 10 milliseconds. Allowed values: 1 to 65535 (but the TCP/IP stack limits the maximum timeout value). Default value: 1000 (10 seconds). If the connection attempt is a TCP client connection, and TCP Connection Timeout below is less than Network Connection Timeout, the value of Network Connection Timeout will be ignored.	Serial to WI-FI Configuration
1	Auto Associate Timeout	The maximum amount of time allowed associating to the desired wireless network in Auto Connect Mode, in units of 10 milliseconds. Allowed values: 0 to 65535. Default value: 500 (5 seconds).	Serial to WI-FI Configuration
2	TCP Connection Timeout	The maximum amount of time allowed establishing a TCP client connection, in units of 10 milliseconds. Allowed values: 0 to 65535 (but the TCP/IP stack limits the maximum timeout value). Default value: 500 (5 seconds). Note that 0 corresponds to the default TCP/IP stack timeout (75 seconds).	TCP Clients
3	Association Retry Count	Not currently supported.	Profile Definitions
4	Nagle Algorithm Wait Time	The maximum time for serial data sent in Auto Connect Mode to be buffered, in units of 10 milliseconds. Allowed values: 0 to 65535 (but the amount of data is limited by available buffer size). Default value: 10 (100 ms).	Auto Connection Operation
5	Scan Time	The maximum time for scanning in one radio channel, in units of milliseconds. Allowed values: 0 to 65535 (but at the high limit a 14-channel scan will consume 2.6 hours!). Default value: 20 (20 ms).	Scanning

Table 3-10: Configuration Parameters for Network Association.

Identification Information

Command: **ATIn**

Response: *OK*

n *is the ID of the information to obtain. The responses are listed in the following table. These responses are provided as ASCII strings.*

Description:

Information ID	Description
0	OEM identification
1	Hardware version
2	Software version

Table 3-11: Application Information.

Serial to Wi-Fi Configuration Profiles

Module configuration parameters can be stored and recalled as a Profile; see 0 for a detailed description of the profile parameters.

Save Profile

Command: **AT&Wn**

Response: *n shall either be 0 for profile 0 or 1 for profile 1. (Higher values are allowed if more profiles are configured at compile time.)*

ERROR (1, if verbose disabled), if the operation failed.

Description: Upon deployment of this command, the current configuration settings are stored in non-volatile memory under the specified profile. Note that, in order to ensure that these parameters are restored after power cycling the module, the command `AT&Y` (paragraph heading Selection of Default Profile in this chapter) must also be issued, using the same profile number selected here.

Load Profile

Command: **ATZn**

Response: *OK*

Description: n shall either be 0 for profile 0 or 1 for profile 1. (Higher values are allowed if more profiles are configured at compile time.) Upon deployment of this command, the currently configured settings are set to those stored in non-volatile memory under the specified profile. This command returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface. The WLM10x module uses profile 0 as the default profile.

Selection of Default Profile

Command: **AT&Yn**

Response: *OK*

Response will be ERROR or 1, based on verbose settings, if a valid input cannot be executed.

Description: n shall either be 0 for profile 0 or 1 for profile 1. (Higher values are allowed if more profiles are configured at compile time.)

The settings from the profile that is chosen as the default profile are loaded from non-volatile memory when the device is started.

Restore to Factory Defaults

Command: **AT&F**

Response: *OK*

Description: Upon deployment of this command, the current configuration variables are reset to the factory defaults. These defaults are defined by macro values in the configuration header, and can be modified at compile time. Issuing this command resets essentially all configuration variables ***except*** the IEEE MAC address. Only the command AT+NMAC (paragraph heading MAC Address Configuration in this chapter) changes the MAC address.

Output Current Configuration

Command: **AT&V**

Response: *OK*

Description: Upon deployment of this command, the current configuration and the configuration of the saved profiles are output on the serial port in ASCII format. The details of the profile parameters are described in paragraph heading Profile Definitions in this chapter.

Wi-Fi Interface Configuration

MAC Address Configuration

Command: **AT+NMAC=<MAC ADDRESS>**

Response: *No Response*

Description: Upon deployment of this command, the Module sets the IEEE MAC address as specified. The format of the MAC address is an 8-byte colon-delimited hexadecimal number. An example is shown below:

```
AT+NMAC=00:1d:c9:00:01:a2
```

The MAC address is used in the 802.11 protocol to identify the various nodes communicating with an Access Point and to route messages within the local area (layer 2) network. Fixed MAC addresses issued to network interfaces are hierarchically structured and are intended to be globally unique. Before issuing a MAC address to a given Module, ensure that no other local device is using that address.

The MAC address supplied in the `AT+NMAC` command is saved to flash memory, and will be used on each subsequent cold boot (from power off) or warm boot (from Standby).

The alternative command

```
AT+NMAC2=<MAC ADDRESS>
```

Stores the MAC address in RTC RAM. Each warm boot (from Standby) will use the MAC address stored in RTC RAM (from the most recent `AT+NMAC2=` command), but if power to the device is lost, the next cold boot will use the MAC address stored in flash memory (from the most recent `AT+NMAC=` command). This command is particularly useful in cases where writing to flash memory is undesirable.

Output MAC Address

Command: **AT+NMAC=?**

Response: *No Response*

Description: Upon deployment of the command, the Module outputs the current MAC address of the wireless interface to the serial port, in addition to the usual command responses (paragraph heading Commands for Command Processing Mode in this chapter) . The alternate command is

AT+NMAC2=?

may also be used, and returns the same value.

Regulatory Domain Configuration

Command: **AT+WREGDOMAIN=<Regulatory Domain>**

Response: *OK*

Description: This command sets the regulatory domain as per the Regulatory Domain parameter passed. The supported regulatory domains are:

- FCC → supported Channel range is 1 to 11.
- ETSI → supported Channel range is 1 to 13.
- TELEC → supported Channel range is 1 to 14.

The corresponding values for this regulatory domain that needs to be passed as the parameter are:

- FCC : 0
- ETSI : 1
- TELEC : 2

The default regulatory domain is FCC. The Regulatory domain set is required only once since it is being updated in the flash.

Regulatory Domain Information

Command: **AT+WREGDOMAIN=?**

Response: *No Response*

Description: This is the command to get the configured regulatory domain in the WLM10x

Upon reception of the command, the module outputs the current Regulatory domain of the wireless interface to the serial port as the following format:

REG_DOMAIN=FCC or ETSI or TELEC, in addition to the standard command responses.

Scanning For Access Points Or Ad Hoc Networks

Command: **AT+WS[=<SSID>[,<BSSID>][,<Channel>][,<Scan Time>]]**

Response: *No response.*

Description: Upon deployment of the command, the module scans for networks with the specified parameters, and displays the results. Scanning can be performed to find networks with specific SSID or in a particular operating channel, or a combination of these parameters. Scanning for a specific SSID employs active scanning, in which probe requests are transmitted with the SSID fields being filled appropriately.

The SSID is a string containing between 1 and 32 ASCII characters, Refer to paragraph heading Boot Messages in this chapter for details.

This command does not support scan based on the BSSID.

The Scan Time is in units of Milliseconds with a range of 0-65535.

Upon completion, the module reports the list of networks and information for each network along with the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) one per line, in the following format to the serial interface

<space><BSSID>,<space><SSID>,<space><Channel>,<space><space><Type><space>,<space><RSSI><space>,<space><Security>

Also this sends out the total number of networks found as follows (after send out the above information to the serial interface).

“No. Of AP Found:<n><CR><LF>”

Where n is the total number of networks found during scan.

`Type` is `INFRA` for an infrastructure network and `ADHOC` for an ad hoc network.

Set the Wireless Mode

Command: **AT+WM=n**

Response: *If* n *is 0, the mode is set to* infrastructure*; if* n *is 1, the mode is set to* ad hoc.

If n is 2, the mode is set to limited AP so that the module can act as a limited wireless Access Point. WLM10x Module uses infrastructure(0) as the default mode.

Description:

Associate with a Network, or Start an Ad Hoc or Infrastructure (AP) Network

Command: **AT+WA=<SSID>[,[<BSSID>][,<Ch>]]**

Response: *No response.*

Description: In infrastructure mode (paragraph heading Mode in this chapter, n is 0), the module will attempt to associate with the requested network. In ad hoc mode (paragraph heading Mode in this chapter, n is 1), if a network with the desired SSID or channel or both is not found, then a new network is created. However, if the BSSID was specified in the request and the applicable BSSID is not found, the Module will report an error and will not create an ad hoc network.

In AP mode (paragraph heading Mode in this chapter, n is 2), the module creates an infrastructure network (limited AP) with the SSID passed

The SSID is a string containing between 1 and 32 ASCII characters. Refer to paragraph heading Boot Message in this chapter for details.

Upon completion, the module reports its IP address to the serial interface in the following format:

<LF><4 spaces>IP<14 spaces>SubNet<9 spaces>Gateway<3 spaces>

<space><IP address>:<space><SubNet address>:<space><Gateway address>

In addition to the usual status responses, this command will return `ERROR` or `1` (depending on verbose status) if a valid command was issued but association failed.

In adhoc and AP modes, the radio should be on in active mode (paragraph heading Network Parameters in this chapter).

Disassociation

Command: **AT+WD**

An equivalent command is ATH

Response: *No response.*

Description: Upon deployment of this command, the interface disassociates from the current infrastructure or ad hoc network, if associated. This command returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface.

Associate To An AP Using WPS

Command: **AT+WWPS=<METHOD>[,PIN]**

METHOD is push button (1) or pin (2).

PIN is the pin for PIN method.

Response: *No response.*

Description: Upon execution of this command, the module uses either push button or pin method as per the METHOD parameter to associate to the WPS enabled AP. The PIN is optional and is valid for pin method only.

In addition to the usual status responses this command returns the following information to the serial host on success case:

- SSID=<ssid>
- CHANNEL=<channel>
- PASSPHRASE=<passphrase> for wpa/wpa2 security;
- WEP KEY=<wep key> for WEP security;
- WEPKEYINDEX=<key index> for WEP security

The above information is send to the serial interface with one information element per line.

This command returns `ERROR` or `1` (depending on verbose status) if a valid command was issued but WPS failed.

On success case the serial host should issue the AT+NDHCP=0/1 to establish the L3 connection.

Status

Command: **AT+NSTAT=?**

Response: *No response.*

Description: Upon deployment of this command, the module reports the current network configuration to the serial host:

- MAC address;
- WLAN state;
- SSID;
- Mode;
- Security;
- Channel;
- BSSID;
- Network configuration: IP Address, Subnet mask, Gateway address, DNS1 address, DNS2 address;
- TX count;
- RX count.
- RSSI value

in addition to the usual status response.

The alternate command

AT+WSTATUS

may also be used, Upon deployment of this command, the module reports the current network configuration to the serial host:

- Mode;
- Channel;
- SSID;
- BSSID;
- Security;

if the module associated to an Access Point. If no association is present, the error message NOT ASSOCIATED is returned, in addition to the standard command response.

Get RSSI

Command: **AT+WRSSI=?**

Response: *No response.*

Description: Upon deployment of this command, the current RSSI value (in dBm) is output on the serial port in ASCII format, in addition to the standard command response.

Get Transmit Rate

Command: **AT+WRATE=?**

Response: *No response.*

Description: Upon deployment of this command, the current transmit rate used is output on the serial port in ASCII format along with the standard command response

Set Retry count

Command: **AT+WRETRY=<retrycount>**

Response: *No response.*

Description: Upon deployment of this command, the current wireless retry count is set to the supplied value. The transmission retry count determines the maximum number of times a data packet is retransmitted, if an 802.11 ACK is not received. (Note that the count includes the initial transmission attempt.) The valid range is 4 to 7 with default value 5.

This command returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface.

Wi-Fi Security Configuration

Authentication Mode

Command: **AT+WAUTH=n**

Response: *n is:*

- *None*
- *1 – Open*
- *2 – Shared with WEP*

Description: Note that this command configures the authentication mode, but any required encryption keys must be set using the key commands described below. This authentication mode command is specific to WEP encryption; if WPA/WPA2 operation is employed, the authentication mode may be left at the default value " None". This command returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface.

Security Configuration

Command: **AT+WSEC= n**

Response: Where *n* is:

- 0 – Auto security (All)
- 1 – Open security
- 2 – Wep security
- 4 – Wpa-psk security
- 8 – Wpa2-psk security
- 16 – Wpa Enterprise
- *32 – Wpa2 Enterprise*

Description: The WLM10x module supports a strict security configuration. The WLM10x module supports either one of the above value with default security configuration as auto. This strict security compliance is not applicable for WPS feature. This command returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface.

Set WEP Keys

Command: **AT+WWEPn=<key>**

Response: *n is the key index, between 1 and 4, and key are either 10 or 26 hexadecimal digits corresponding to a 40-bit or 104-bit key. Some examples:*

```
AT+WWEP1=123456abdc
```

```
AT+WWEP3=abcdef1234567890901234567890
```

Description: Upon receiving a valid command, the relevant WEP key is set to the value provided. This command returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface.

Set The WPA-PSK And WPA2-PSK Passphrase

Command: **AT+WWPA=<passphrase>**

Response: No Response

Description: The passphrase is a string containing between 8 and 63 ASCII characters, used as a seed to create the WPA ***pre-shared key*** (PSK).

If the comma (,) is a part of the passphrase, then the passphrase parameter is to be framed in double quotation marks (“passphrase”). Refer to paragraph heading Boot Messages in this chapter for details.

Upon receiving the command, the PSK passphrase is reset to the value provided. This command returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface.

Compute And Store The Value Of The WPA/WPA2 PSK, Derived From The Passphrase And SSID Value

Command: **AT+WPAPSK=<SSID>,<PASSPHRASE>**

Response: *<LF>Computing PSK from SSID and PassPhrase".*

Followed by:

OK or *0*, if there are no errors.

Invalid inputs will result in:

ERROR: INVALID INPUT or *2*

Description: Computation of the PSK from the passphrase is complex and consumes substantial amounts of time and energy. To avoid recalculating this quantity every time the module associates, the module provides the capability to compute the PSK once and store the resulting value. The key value is stored in the SRAM copy of the current profile; the profile needs to be saved in flash memory for this value to persist during a transition to Standby.

The passphrase is a string containing between 8 and 63 ASCII characters, used as a seed to create the PSK. The SSID is a string of between 1 and 32 ASCII characters. Refer to paragraph heading SSID and PASSPHRASE in this chapter for details.

Each Parameter of the above command separated by comma (,). If the comma(,) is a part of the SSID or PASSPHRASE, then SSID and PASSPHRASE parameters is to be framed in double quotation marks ("SSID","PASSPHRASE").

When the command is issued, the module immediately responds with "<LF>Computing PSK from SSID and PassPhrase". Computation of the passphrase can be time-consuming! When it is complete, the module will issue the usual OK or 0. Invalid inputs will result in ERROR: INVALID INPUT or 2, as usual.

Upon receiving the command, the module computes the PSK from the SSID and passphrase provided, and stores those values in the current profile. The current profile parameters PSK Valid, PSK-SSID, and WPA Passphrase are updated, and can be queried with AT&V (paragraph heading Output Current Configuration in this chapter). The next time the module associates to the given SSID, the PSK value is used without being recalculated.

After the PSK has been computed, the commands AT&W (to save the relevant profile) and AT&Y (to ensure that the profile containing the new PSK is the default profile) should be issued. The PSK will then be available when the module awakens from Standby. Refer to paragraph heading Selection of Default Profile in this chapter for more information on profile management.

Configure The WPA / WPA2 PSK Key Directly

Command: **AT+WPSK=<PSK>**

Response: *No Response*

Description: This command directly sets the pre-shared key as provided. The argument is a 32-byte key, formatted as an ASCII hexadecimal number; any other length or format is considered invalid. Example:

```
AT+WPSK=
0001020304050607080900010203040506070809000102030405060708090001
```

After the PSK has been entered, the commands AT&W (to save the relevant profile) and AT&Y (to ensure that the profile containing the new PSK is the default profile) should be issued. The PSK will then be available when the module awakens from Standby. Refer paragraph heading Selection of Default Profile for more information on profile management.

Configure The EAP-Security

Command: **AT+ WEAPCONF=<Outer Authentication>,<Inner Authentication>,<user name>,<password>**

Response: *No Response*

Description: Upon execution of this command, the module will set the Outer authentication, Inner authentication, user name and password for EAP Security.

The valid outer authentication values are:

Eap-FAST: 43

Eap-TLS: 13

Eap-TTLS: 21

Eap-PEAP: 25

The valid Inner Authentication values are:

Eap-MSCHAP: 26

Eap-GTC: 6

Configure Certificate For EAP-TLS

Command: AT+ WEAP=< Type >,< Format >,< Size >,< Location >

<ESC>W <data of size above>

- Type: CA certificate(0)/ Client certificate(1)/ Private Key(2)
- Format: Binary(0)/Hex(1)
- Size: size of the file to be transferred.
- Location: Flash(0)/Ram(1)

Response: If the operation is successful, there will be no response

ERROR, 1 (verbose disabled), if the operation failed

Description: This command enables the module to receive the certificate for EAP-TLS. This command stores the certificate in flash or RAM, depending on the parameter.

Configure The Certificate For SSL/HTTPS Connection

Command: AT+ TCERTADD=<Name>,<Format>,<Size>,<Location>

<ESC>W <data of size above>

- Name: Name of the certificate
- Format: Binary(0)/Hex(1)
- Size: Size of the file to be transferred.
- Location : Flash (0)/Ram(1)

Response: If the operation is successful, there will be no response

ERROR, 1 (verbose disabled), if the operation failed

Description: This command enables the module to receive the certificate for SSL/HTTPS connection. It stores the certificate in flash or ram depends on the parameter.

Certificate Deletion

Command: **AT+TCERTDEL=<certificate name>**

Response: If the operation is successful, there will be no response

ERROR, 1 (verbose disabled), if the operation failed

Description: This command deletes the SSL/HTTPS/EAP-TLS certificate stored in flash/ram by name.

In the case of EAP-TLS certificate names are:

- TLS_CA
- TLS_CLIENT
- TLS_KEY

Enable/Disable 802.11 Radio

Command: **AT+WRXACTIVE=n**

Response: *If* n *is 0, the radio is disabled and if* n *is 1, the radio is enabled with default setting as disabled.*

Description: This command returns the standard command response (OK) to the serial interface. If `WRXACTIVE` = 1, the 802.11 radio receiver is always on. This minimizes latency and ensures that packets are received at the cost of increased power consumption. The WLM10x module cannot enter Deep Sleep (paragraph heading Enable/Disable Module Deep Sleep) even if it is enabled (`PSDPSLEEP=1`). Power Save mode (paragraph heading Enable/Disable 802.11 Power Save Mode) can be enabled but will not save power, since the receiver is left on. If `WRXACTIVE` = 0, the receiver is switched off after association is complete. If Power Save mode is not enabled (`WRXPS` not issued or `WRXPS`=0), the receiver will not be turned on again unless `WRXACTIVE` = 1 is received. Packets will not be received, and disassociation could occur. If Power Save mode is enabled (`WRXPS`=1) prior to issuing `WRXACTIVE` = 0, the receiver will be turned off, but will turn on again when it is time to listen for the next beacon from the Access Point. If Deep Sleep is also enabled, the receiver will turn off, and the module will enter Deep Sleep when all pending tasks are completed, but again the system will be awakened to listen to the next beacon. If a transition to Standby is requested and occurs (paragraph heading Request Standby Mode), the module will remain in Standby for the requested period, and will ***not*** awaken to receive a beacon during that time.

Enable/Disable 802.11 Power Save Mode

Command: **AT+WRXPS=n**

Response: *If* n *is 0, Power Save is disabled and if* n *is 1, Power Save is enabled with default setting as enabled.*

Description: This command returns the standard command response (OK) to the serial interface. In 802.11 Power Save Mode, the node (in this case, the WLM10x module) will inform the Access Point that it will become inactive, and the Access Point will buffer any packets addressed to that node. In this case, the WLM10x module radio receiver is turned off between beacons. The node will awaken to listen to periodic beacons from the Access Point that contains a Traffic Indication Map (TIM) that will inform the Station if packets are waiting for it. Buffered packets can be retrieved at that time, using ***PSPoll*** commands sent by the node. In this fashion, power consumed by the radio is reduced (although the benefit obtained depends on traffic load and beacon timing), at the cost of some latency.

The latency encountered depends in part on the timing of beacons, set by the Access Point configuration. Many Access Points default to 100msec between beacons; in most cases this parameter can be adjusted.

Enable/Disable Multicast Reception

Command: **AT+MCSTSET=n**

Response: No Response

Description: If n = 0, multicast reception is disabled; if n = 1, multicast reception is enabled. By default the multicast reception is enabled.

Transmit Power

Command: **AT+WP=<power>**

Response: No Response

Description: On reception of this command, the transmit power is set to the supplied value. The desired power level shall be specified in ASCII decimal format. The value of the parameter can range 2 to 15, with default value of 2 (for maximum RF output).

Sync Loss Interval

Command: **AT+WSYNCINTRL=<n>**

Response: *n is the number of beacon interval.*

Description: On execution of this command the module sets the sync loss interval for n times the beacon interval so that if the module does not receive the beacon for this time it informs the user this event as “Dissociation event”. The default value of sync loss interval is 30.This command accept the sync loss interval from 1 to 65535.

Enable The External PA

Command: **AT+EXTPA=<n>**

Response: *n=1 to enable the external PA*

n=0 to disable external PA

Description: If enabled ,this command forces the module to standby and comes back immediately and causing all configured parameters and network connection will be lost. Paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface.

Configure The Keep-Alive Timer Interval

Command: **AT+PSPOLLINTRL=<n>**

Response: No Response

Description: On execution of this command, the module will set the keep-alive time interval for n seconds. This keep-alive timer will fire for every n seconds once the module is associated. This timer will keep the module in associated state even there is no activity between AP and module. The default vale is 45 seconds. This command accepts keep-alive timer interval from 0 to 65535 seconds. The value 0 disables this timer.

Network Interface

Network Parameters

Note that IP addresses in the network commands are to be given in ASCII dotted-decimal format.

Enable Or Disable DHCP (DHCP Support)

Command: **AT+NDHCP=n**

Response: *If* n *is 0, DHCP is disabled.*

If n *is 1, DHCP is enabled.*

Description: If the interface is associated with a network, enabling DHCP will cause an attempt to obtain an IP address using DHCP from that network. Thus issuing this command with n=1 will cause the module to attempt to refresh an existing DHCP address. If the module is not associated when the command is received, future associations will attempt to employ DHCP. If the module fails to obtain an address via DHCP it will return an error response `ERROR: IP CONFIG FAIL` if verbose is enabled, or `F`(0x0F) if verbose is disabled

If the interface is not associated, this command returns the standard command response (OK) else it returns the ip address information along with the standard command response in the following format:

<LF><4 spaces>IP<14 spaces>SubNet<9 spaces>Gateway<3 spaces><CR><LF>

<space><IP address>:<space><SubNet address>:<space><Gateway address>

By default, DHSCP is disabled.

Static Configuration of Network Parameters

Command: **AT+NSET=<Src Address>,<Net-mask>,<Gateway>**

Response: *No Response*

Description: Upon deployment of this command, any previously-specified network parameters are overridden, and the module is configured to use the newly-specified network parameters for the current association, if associated, and for any future association. The use of DHCP is disabled if the network parameters are configured statically. The DNS address can be set using `AT+DNSSET` (paragraph heading Static Configuration of DNS (Client)).

Module Support DHCP Server Start/Stop

Command: **AT+DHCPSRVR=1/0**

Response: *1 is for start the server and 0 is for stop the server.*

Description: Prior to start the server, The module should be configured with a valid static ip address (using command described in paragraph heading Static Configuration of Network Parameters, both Src address and Gateway should be same) and created or configure to create a limited AP network.

This DHCP server can support maximum 8 client connections with server ip as the statically configured Ip address and client ip address starts from the next ip address of the configured static ip address.

Module Support DNS Server And The Command To Start/Stop The Server

Command: **AT+DNS=1/0,<url>**

Response: *1 is for start the server and 0 is for stop the server.*

Description: URL is the DNS name associated to the DNS IP address.

Prior to start the server, the DHCP server (paragraph heading DHCP Server) should be started and created or configure to create a limited AP network. This DNS server use the same DHCP server ip address as it ip address.

This command returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface.

DNS Lookup (Client) Retrieve IP Address From A Host Name

Command: **AT+DNSLOOKUP=<URL>,[<RETRY>,<TIMEOUT-S>]**

Response: *No Response*

Description: Where `URL` is the hostname to be identified. Upon deployment of this command, the Module queries the DNS server to obtain the IP address corresponding to the hostname provided in `URL`, and returns the address if found. Retry and timeout are optional; if they are not given, or if 0 values are provided, the default value of 2 is used. Timeout is in seconds.

The retry range is 0 to 10 and timeout range is 0 to 20.

In addition to the standard command response , the interface returns ERROR (1 , if verbose disabled) if a valid command was issued but DNS lookup failed.

Statically Configure The DNS IP Addresses

Command: **AT+DNSSET=<DNS1 IP>,[<DNS2 IP>]**

Response: *No Response*

Description: This command sets the values of the DNS server addresses to be used by the module. The second address, DNS2 IP, is optional but should not be same as DNS1 IP. This command returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface.

This static configuration of DNS set will take effect only in the case of static IP address on the module.

Store The Network Context And Configuration Prior To A Transition To Standby

Command: **AT+STORENWCONN**

Response: *If the operation is successful, there will be no response.*

"DISASSOCIATED" or *9* (based on verbose setting) if the interface is not associated state.

Description: This command will preserve network connection parameters (layer 2 and layer 3 information) in RTC memory when the Laird Technologies module is sent to Standby mode using the Request Standby command (paragraph heading Request Standyby Mode). Note that CID's are lost when the transition to Standby occurs.

Restore Network Context

Command: **AT+RESTORENWCONN**

Response: *No Response*

Description: This command reads the layer 3 (IP) network connection parameters saved by Store Network Context (paragraph heading Static Configuration of Network Parameters), and re-establishes the connection that existed before the transition to Standby. If needed, the node will re-associate and re-authenticate with the specified SSID. In addition to the usual status responses, this command returns `ERROR` or `1` (based on verbose setting) is it is called prior to storing the network connection, or after storing the network connection but before a transition to Standby has occurred.

Connection Management Configuration

All connection commands, except for the transport of Raw Ethernet data (paragraph heading Enable/Disable Raw Ethernet Support), use the embedded TCP/IP Network Stack functions to perform the required actions. Connection identifiers, denoted as `<CID>` below, are to be sent as single hexadecimal characters in ASCII format.

Open A TCP Client Connection

Command: **AT+NCTCP=<Dest-Address>,<Port>**

Response: *No Response*

Description: Upon deployment of this command, the interface attempts to open a socket and connect to the specified address and port. The connection attempt shall timeout if a socket has not been opened after a delay equal to TCP Connection Timeout.

On successful connection, the interface sends `CONNECT<space><CID>` to the serial host along with the standard response, where CID is the newly allocated connection identifier. `ERROR` or `1` is returned if a timeout occurs.

Note:
By default the TCP keep alive option is disabled but the user can enable it using the command described in paragraph heading SOCKET Options Configuration.
The default TCP retransmission timeout is infinite, but the user can change it using the command described in paragraph heading SOCKET Options Configuration.
To detect the abnormal disconnection in L3 Layer after establishing the TCP connection on the WLM10X module, the user should configure the proper values of the above two timeouts.

Open A UDP Client Connection

Command: **AT+NCUDP=<Dest-Address>,<Port>[<,Src.Port>]**

Response: *No Response*

Description: Dest-Address is the destination (server) ip address

Port is the destination (server) port

Upon deployment of this command, the interface opens a UDP socket capable of sending data to the specified destination address and port. If a source port is provided, the socket will bind to the specified port. On successful completion, the interface sends `CONNECT<space><CID>` to the serial host, followed by standard response. where CID is the newly allocated connection identifier.

The port range 0xBAC0 (47808) to 0xBACF (47823) may not be used for destination port

Start A TCP Server

Command: **AT+NSTCP=<Port>**

Response: *No Response*

Description: Upon deployment of this command, the interface opens a socket on the specified port and listens for connections. On successful creation of the server, `CONNECT<space><CID>`followed by standard command response (paragraph heading Commands for Command Processing Mode in this chapter) is sent to the serial host, where `CID` is the newly allocated connection identifier, followed by `OK` or 0. Up to 16 total CID's can be supported by the application, so a TCP server can support up to 15 distinct client connections, if no other entity has assigned CID's.

Start A UDP Server

Command: **AT+NSUDP=<Port>**

Response: *No Response*

Description: Upon deployment of this command, the interface:

- Allocates a CID for this connection. If no CID is available, the command fails.
- If a valid CID was allocated, a UDP socket is opened on the specified port.
- If the socket is successfully created, `CONNECT<space><CID>` is sent to the serial host, followed by standard command response. where CID is the allocated connection identifier.

The port range 0xBAC0 (47808) to 0xBACF (47823) may not be used.

Output The Current CID Configuration

Command: **AT+CID=?**

Response: *No Response*

Description: This command returns the current CID configuration for all existing CID's:

- CID number, In decimal format.
- CID type;
- Protocol;
- Local port;
- Remote port;
- Remote IP address

followed by the usual status response. If no valid CID's are present, the message "<space>`No valid Cids"` is sent to serial interface, followed standard command response.

Closing a Connection

Command: **AT+NCLOSE=<CID>**

Response: *No Response*

Description: Upon deployment of this command, the connection associated with the specified CID is closed, if it is currently open. On completion of this command the CID is free for use in future connections. If an invalid CID is provided, the command returns `ERROR: INVALID CID` or `5`, depending on verbose status else it returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter)

Closing All Connections

Command: **AT+NCLOSEALL**

Response: *No Response*

Description: Upon execution of this command, all open connections are closed and returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter).

Configure A Socket Which Is Identified By A CID

Command: **AT+SETSOCKOPT=<CID>,<Type>,<Parameter>,<Value>,<Length>**

Response: *No Response*

Description: Upon execution of this command the module configure the socket identified by CID with the value passed.

CID: is the socket identifier received after opening a connection.

Type: is the type of the option to be set

- SOCKET: 65535
- IP : 0
- TCP: 6

Parameter: The Option name to be set. Accepts hex values.

- TCP_MAXRT : 10(Hex)
- TCP_KEEPALIVE: 4001(Hex)
- SO_KEEPALIVE: 8(Hex)
- TCP_KEEPALIVE_CNT: 4005(Hex)

Value: The value to be set. This in seconds (Ex: 30 → 30 seconds)

Length: The length of the value in bytes (Ex: in above case it is 4, basically it tells the type of the value is integer, Short or Char)

Integer → 4
Short →2
Char →1

This command returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter) to the serial interface.

Ex:

Set the TCP retransmission timeout to 20 seconds is
AT+SETSOCKOPT=0,6,10,20,4

Where 0 is the CID.

Similarly, to enable the TCP Keepalive is:

AT+SETSOCKOPT= 0,65535,8,1,4 → Enable SO_KEEPALIVE option at base socket level. Without enabling this TCP_KEEPALIVE will not work.
AT+SETSOCKOPT= 0,6,4001,600,4→ Enable TCP_KEEPALIVE option at TCP level with timeout as 600 seconds.

Note: The default keepalive count is 8 so the minimum keepalive timeout is 8*75=600 seconds. To reduce the keepalive timeout further, set the Keepalive count first to an appropriate value and set the keepalive timeout.

Ex: To set the keep alive timeout to 75 seconds:

AT+SETSOCKOPT =0,6,4005,1,4 → Configure TCP Keep Alive Probe Sending count at just 1.

AT+SETSOCKOPT= 0,6,4001,75,4 → Enable TCP_KEEPALIVE option at TCP level with 75 seconds as Keep Alive timeout.

Open An SSL Connection

Command: **AT+SSLOPEN=<CID>,[<certificate name>]**

Response: *No response if successful.*

ERROR if the operation fails.

Description: Upon execution of this command, the module opens an SSL connection over the TCP connection identified by the CID. For this SSL connection, the module uses the certificate stored in memory that is identified by the certificate name. Prior issuing this command, a valid TCP connection should exists with connection identifier as CID

Closing SSL connection

Command: **AT+SSLCLOSE=<CID>**

Response: *No response if successful.*

ERROR if the operation fails.

Description: Upon reception of this command, the module closes the existing SSL connection identified by CID.

HTTP Client Configuration

Command: **AT+HTTPCONF=<Param>,<Value>**

Response: *No response if successful.*

ERROR if the operation fails.

Description: Upon reception of this command the module configures the HTTP parameters. The ‘param’ is the HTTP header and is one of the following:

- GSN_HTTP_HEADER_AUTHORIZATION (2)
- GSN_HTTP_HEADER_CONNECTION (3)
- GSN_HTTP_HEADER_CONTENT_ENCODING (4)
- GSN_HTTP_HEADER_CONTENT_LENGTH (5)
- GSN_HTTP_HEADER_CONTENT_RANGE (6)
- GSN_HTTP_HEADER_CONTENT_TYPE (7)
- GSN_HTTP_HEADER_DATE (8)
- GSN_HTTP_HEADER_EXPIRES (9)
- GSN_HTTP_HEADER_FROM (10)
- GSN_HTTP_HEADER_HOST (11)
- GSN_HTTP_HEADER_IF_MODIFIED_SINCE (12)
- GSN_HTTP_HEADER_LAST_MODIFIED (13)
- GSN_HTTP_HEADER_LOCATION (14)
- GSN_HTTP_HEADER_PRAGMA (15)
- GSN_HTTP_HEADER_RANGE (16)
- GSN_HTTP_HEADER_REFERER (17)
- GSN_HTTP_HEADER_SERVER (18)
- GSN_HTTP_HEADER_TRANSFER_ENCODING (19)
- GSN_HTTP_HEADER_USER_AGENT (20)
- GSN_HTTP_HEADER_WWW_AUTHENTICATE (21)
- GSN_HTTP_REQUEST_URL (23)

The ‘value’ is a string that depends on the above parameters.

HTTP Client Configuration Removal

Command: **AT+HTTPCONFDEL=<Param>**

Response: *No response if successful.*

ERROR if the operation fails.

Description: Upon reception of this command the module removes the HTTP configuration specified by the param. The 'param' is the HTTP header and is one of the following:

- GSN_HTTP_HEADER_AUTHORIZATION (2)
- GSN_HTTP_HEADER_CONNECTION (3)
- GSN_HTTP_HEADER_CONTENT_ENCODING (4)
- GSN_HTTP_HEADER_CONTENT_LENGTH (5)
- GSN_HTTP_HEADER_CONTENT_RANGE (6)
- GSN_HTTP_HEADER_CONTENT_TYPE (7)
- GSN_HTTP_HEADER_DATE (8)
- GSN_HTTP_HEADER_EXPIRES (9)
- GSN_HTTP_HEADER_FROM (10)
- GSN_HTTP_HEADER_HOST (11)
- GSN_HTTP_HEADER_IF_MODIFIED_SINCE (12)
- GSN_HTTP_HEADER_LAST_MODIFIED (13)
- GSN_HTTP_HEADER_LOCATION (14)
- GSN_HTTP_HEADER_PRAGMA (15)
- GSN_HTTP_HEADER_RANGE (16)
- GSN_HTTP_HEADER_REFERER (17)
- GSN_HTTP_HEADER_SERVER (18)
- GSN_HTTP_HEADER_TRANSFER_ENCODING (19)
- GSN_HTTP_HEADER_USER_AGENT (20)
- GSN_HTTP_HEADER_WWW_AUTHENTICATE (21)
- GSN_HTTP_REQUEST_URL (23)

HTTP Client Connection Open

Command: **AT+HTTPOPEN=<host >[, <Port Number>, <SSL Flag>, <certificate name>,<proxy>]**

Response: *It returns the normal response code and the CID of the HTTP client connection on success.*

Description: This command opens an HTTP client on the module and connects to the server specified by the host name or IP address.

- Host: Host is either the Fully Qualified Domain Name of the Server or the IP address of the server
- Port Number: Port number of the server to which the HTTP client will open the connection. The client can specify the port when the server is running on a non-standard port. Default is the standard port – 80 for HTTP and 443 for HTTPS.
- SSL Flag: 0 – SSL Disabled, 1 – SSL Enabled. Default is SSL Disabled
- Certificate Name: The name of the CA Certificate to be used for Server Certificate Authentication in case SSL is enabled. The CA Certificate must be provisioned before this.

It uses the certificate configured on the module identified by the certificate name.

- Proxy: This flag is used only during HTTPS connection through proxy 1 – The HTTPS connection is through proxy server.

Get/Post HTTP Data On The HTTP Client Connection

Command: **AT+HTTPSEND=<CID>,<Type>,<Timeout>,<Page>[,Size of the content]**

ESC<H><Content of above size>

Response: *Response: Receive is implicit in AT+HTTPSEND based on the HTTPS Server's response to the sent data. Received data is asynchronous and should be handled accordingly.*

The response from the server is sent to the host in one or more chunks with max size of 1024 bytes. Each chunk is of the format:

<Esc>H<1 Byte - CID><4 bytes – Length of the data><data>

The data part of first chunk of the response will have the status line at the beginning . The status line contains the status code and the status phrase . This will be in the format:

<status code><space><status phrase>\r\n

After the last chunk, OK/ERROR is sent to the host.

Description: This command sends a get or post HTTP request to the server. The content can be transferred using the escape sequence mentioned previously.

- CID : HTTP client identifier.
- Type: GSN_HTTP_METHOD_GET (1) / GSN_HTTP_METHOD_POST (3)
- Page: The page/script being accessed e.g. /index.html
- Timeout: timeout value in seconds.
- Size: Actual Content size, Optional in case of GET

In case the HTTP connection is opened with SSL encryption enabled, this command encrypt the data based with encrypt key in SSL connection structure for the specific CID. This encryption happens before Network Layer and the Encrypted data will be sent through the network layer

Close The HTTP Client Connection

Command: **AT+HTTPCLOSE=<CID>**

Response: *No response.*

Description: Upon execution of this command the module closes the HTTP client connection identified by the CID.

Enable Or Disable Support Of Raw Ethernet Data

Command: **AT+NRAW=<0|1|2>**

Response: `OK`

The results of this command are summarized in the following table:

Information ID	Description
0	Disable Raw Ethernet frame transmission / reception.
1	Enable Raw Ethernet frames with NON-SNAP 802.2LLC headers.
2	Enable all Raw Ethernet frames.

Table 3-12: Raw Ethernet Support Options.

Description: When selection 1 is chosen, 802.3 frames are presumed to include an 802.2 header which is not a SNAP header. These frames are used, for example, for sending BACNET data over Ethernet. A frame of this type has the format:

<ESC>R:<Length>:<DstAddr><SrcAddr>0x0000<Raw-Payload>

On the receiving side, frames with 802.2 headers which are not a SNAP header, are sent directly to serial interface and DATA Frames with UDP port range 0xBAC0 to 0xBACF will be ignored.

When selection 2 is chosen, the 802.2 header (presumed to be a SNAP header) is removed, and a raw Ethernet II frame payload is expected, as per the format below:

<ESC>R:<Length>:<DstAddr><SrcAddr><EtherType><Raw-Payload>

On the receiving side, frames with 802.2 headers that are not SNAP headers and DATA Frames with UDP port ranges 0xBAC0 to 0xBACF are sent directly to serial interface.

This frame format is used for sending IP data over BACNET.

Length is size of DstAddr, SrcAddr, EtherType and Payload.

If the Module receives DATA Frames, where the 802.2 LLC headers' SSAP and DSAP are not both 0xAA, these frames are presumed to be 802.3 frames, and are sent to the Module's serial port as described above.

If the Module received DATA Frames with UDP port range 0xBAC0 to 0xBACF, they are presumed to be BACNET/IP frames, BacNet Ip frame, and are sent to the Module's serial port as described above.

Unsolicited Data Transmission

Command: **AT+UNSOLICITEDTX=<Frame Control>,<Sequence Control>,<Channel>,<Rate>,<WmmInfo>,**

<Receiver Mac>,<Bssid of AP>,<Frame Length>

Response: *No response if successful.*

Description: This command enables the unsolicited data transmission with the parameters configured. After issuing this command, the user needs to send the payload data as following:

<ESC>D/d <PayLoad of the above Frame length>

- Frame Control: is the 802.11 frame control field. It should be limited to all data frames and management frames like beacons, association requests and probe responses.
- Sequence Control: is the sequence number of the frame. This field consists of 12 bits (LSB) fragment number and 4 bit (MSB) sequence number (0-65535).
- Channel: is the channel on which the data to be sent.
- Rate: is the rate at which the data to be send and the possible values are:

  RATE_1MBPS = 130,

  RATE_2MBPS = 132,

  RATE_5_5MBPS = 139,

  RATE_11MBPS = 150
- WmmInfo: is the wmm information to be sent.
- Receiver Mac: is the remote MAC address of the frame to be sent.
- Bssid: is bssid of the AP.
- Frame Length: is the length of the payload. The maximum size of the frame is limited to 1400 bytes.

Battery Check

Battery Check Start

Command: **AT+BCHKSTRT=<Batt.chk.freq>**

Response: *No response if successful.*

ERROR if the operation fails.

Description: The unit of Batt.chk.freq is in number of packets send out from the WLM10x module.

The valid range for the parameter Batt.chk.freq is between 1 and 100. Upon deployment of this command, the module performs a check of the battery voltage each Batt.chk.freq number of sent packets, and stores the resulting value in nonvolatile memory; only the most recent value is stored. Note that battery checks are performed during packet transmission to ensure that they reflect loaded conditions. Battery checks can be used to ensure that a battery-powered system is provided with sufficient voltage for normal operation. Low supply voltages can result in data corruption when profile data is written to flash memory.

Set The Battery Warning/Standby Level To Enable The Module's Internal Battery Measuring Logic

Command: **AT+ BATTLVLSET=<Warning Level>,<Warning Freq>,<Standby Level>**

Response: *No response if successful.*

Description: Upon execution of this command the module's internal battery level monitoring logic starts. This command should be executed before the battery check start command (paragraph heading Battery Check Start).

Warning Level: The battery voltage, in millivolts. When the module battery voltage is less than this level, it sends the message "Battery Low" to the serial interface.

Warning Freq: is the frequency at which the module sends the "Battery Low" message to the serial interface once the module's battery check detected low battery.

Standby Level: The battery voltage, in millivolts, When the module battery voltage reaches this level, it sends the message "Battery Dead" to the serial interface and goes to long standby.

Set/Reset The Battery Check Period After Battery Check Has Been Started

Command: **AT+BCHK=< Batt.chk.freq >**

Response: *No response if successful.*

Description: The valid range for the parameter Batt.chk.freq is between 1 and 100. Upon receipt, the module records the new value of the battery check frequency so that module performs the battery voltage check with the new value set. This command returns standard command response (paragraph heading Commands for Command Processing Mode in this chapter).

The same command can be used to get the current configured battery check period, the usage as follows

AT+BCHK=?

Battery Check Stop

Command: **AT+BCHKSTOP**

Response: *No response if successful.*

Description: Upon deployment of this command, battery check is halted.

Battery Value Get

Command: **AT+BATTVALGET**

Response: *This command should return a message with the latest value, e.g.* `Battery Value: 3.4 V`*, followed by the standard command response.*

Description: This command will retrieve the results of battery check operations. If this command is issued before issuing the command to start battery checks, it returns `ERROR` or `1,` depending on the current verbose setting.

Power State Management

Enable/Disable Module Deep Sleep

Command: **AT+PSDPSLEEP**

Response: *This command does not return any response code to the serial interface. The WLM10x module sends the message "Out of Deep Sleep" once it comes out from deep sleep.*

Description: This command will enable the Laird Technologies module's power-saving Deep Sleep processor mode.

When enabled, the module will enter the power-saving Deep Sleep mode when no actions are pending. In Deep Sleep mode, the processor clock is turned off, and module power consumption is reduced to less than 1 mW (about 0.1 mA at 1.8 V). Note that other components external to the module may continue to dissipate power during this time, unless measures are taken to ensure that they are also off or disabled.

The processor can be awakened by sending data on the serial port. However, several milliseconds are required to stabilize the clock oscillator when the system awakens from Deep Sleep. Since the clock oscillator must stabilize before data can be read, the initial data will not be received; "dummy" (discardable) characters or commands should be sent until an indication is received from the application.

Request Standby Mode

Command: **AT+PSSTBY=x[,<DELAY TIME>,<ALARM1 POL>,<ALARM2 POL>]**

Response: *No response.*

Description: This command will request a transition to ultra-low-power Standby operation.

The parameters are:

- `x` is the Standby time in milliseconds. If a delay time (see below) is provided, the Standby count begins after the delay time has expired.
- `DELAY TIME` is the delay in milliseconds from the time the command is issued to the time when the module goes to Standby.
- `ALARM1 POL` is the polarity of the transition at pin 31 of the module which will trigger an alarm input and waken the Laird Technologies module from Standby. A value of 0 specifies a high-to-low transition as active; a value of 1 specifies low-to-high.
- `ALARM2 POL` is the polarity of the transition at pin 36 that triggers an alarm input, using the same convention used for Alarm1.

The parameters `DELAY TIME`, `ALARM1 POL`, and `ALARM2 POL` are optional. Specifying an alarm polarity also enables the corresponding alarm input.

This command does not return any response code to the serial interface. When this command is issued, the Laird Technologies module will enter the ultra-low-power Standby state (after the optional delay time if present), remaining there until *x* milliseconds have passed since the command was issued, or an enabled alarm input is received. Any current CID's are lost on transition to Standby. On wakeup, the module sends the message Out of Standby-<reason of wakeup> or the corresponding error code (paragraph heading TCP Server Connections), depending on verbose status.

In Standby, only the low-power clock and some associated circuits are active. Serial messages sent to the UART port will not be received. The radio is off and packets cannot be sent or received. Therefore, before requesting a transition to Standby, the requesting application should ensure that no actions are needed from the interface until the requested time has passed, or provide an alarm input to awaken the module when needed. The alarm should trigger about 10 msec prior to issuance of any serial commands.

The Standby clock employs a 34-bit counter operating at 131,072 Hz, so the maximum possible Standby time is 131,072,000 milliseconds, or about 36.4 hours. Standby is not entered until all pending tasks are completed, and a few milliseconds are required to store any changes and enter the Standby state; a similar delay is encountered in awaking from Standby at the end of the requested time. Therefore, we do not recommend Standby times less than about 32 milliseconds.

Wireless Parameters

Command: **AT+WAUTO=<mode>,<SSID>,<BSSID>,[channel]**

Response: *No response.*

Description: This command will set the auto connection wireless parameters for the current profile.

Mode is 0 for Infrastructure and 1 for Ad-hoc mode;

SSID is the SSID of the AP or Ad-hoc Network to connect to;

BSSID is the BSSID of the AP or Ad-hoc Network to connect to;

Channel is the operating channel.

All other parameters required to configure the wireless connection are taken from the current Profile (paragraph heading Profile Definition).

Network Parameters

Command: **AT+NAUTO=<Type>,<Protocol>,<Destination IP>,<Destination Port>**

Response: *No response.*

Description: This command will set the network parameters for auto connection operation for the current profile.

- Type is 0 for Client and 1 for Server;
- Protocol is 0 for UDP and 1 for TCP;
- Destination IP is the IP address of the remote system (optional if the Module is acting as a server);
- Destination Port is the port number to connect to on the remote system.

Enable Auto Connection

Command: **ATCn**

Response: *No response.*

Description: *n* is 0 to disable auto connection or 1 to enable auto connection.

Upon receipt of this command, the configuration setting in non-volatile memory is modified according to the parameter value in the command; the resulting change (if any) takes effect on the next reboot, or the next issuance of an ATA command.

Initiate Auto Connect

Command: **ATA**

Response: *The module responds with the IP address, subnet mask, and Gateway IP address, followed by CONNECT<space>CID and* `OK` *or* `0` *(per verbose status), if the connection is successful. If the connection attempt is unsuccessful the module returns* `ERROR` *or* `1` *(per verbose status).*

Description: On reception of this command, the interface initiates the auto connection procedure as described in paragraph heading Auto Connection in this chapter, using the parameters specified by the `AT+WAUTO` and `AT+NAUTO` commands (paragraph heading Network Parameters). The module responds with the IP address, subnet mask, and Gateway IP address, followed by CONNECT<space>CID and `OK` or `0` (per verbose status), if the connection is successful. If the connection attempt is unsuccessful the module returns `ERROR` or `1` (per verbose status). After the connection is established, the module enters the data transfer mode described in paragraph heading Auto Connection in this chapter.

If the module is already associated with a wireless network, the alternative command ATA2 below may be used.

Initiate Auto Connect – TCP/UDP Level

Command: **ATA2**

Response: *If the connection successful it returns CONNECT<space>CID followed by standard command response If a valid command input was received, but the connection cannot be established due to a socket bound failure, the message ERROR: SOCKET FAILURE or 3 (per verbose settings) is returned.*

Description: This command will initiate auto connection when the Module is already associated with an Access Point.

This command requires a pre-existing wireless association. On reception of this command, the interface establishes a network connection to a TCP or UDP server with the parameters specified by the AT+NAUTO command (paragraph heading Network Parameters). This command assumes a pre-existing association and should not be issued unless such exists.

Return to Auto Connect Mode

Command: **ATO**

Response: `OK`

Description: If the interface receives this command after it has exited the auto connect mode with +++, it shall return to auto connect mode. If the connection no longer exists, the interface attempts to re-establish the previous connection, and returns to data mode if the reconnection is successful. If the module was not previously connected when this command is received, it returns an error.

Web Provisioning

Command: **AT+WEBPROV=<user name>,<passwd>**

Response: *This command returns standard command response (paragraph heading Commands for Command Processing Mode in this chapter) or ERROR, if the operation fails.*

Description: The module supports provisioning through web pages. This command will start web provisioning.

Prior to issuing this command the module should be in an ad hoc or limited AP network with a valid ip address. Upon reception of this command the module starts a web server. It returns the normal response code OK or ERROR depends on the success or failure condition.

Once the module returns the success response (“OK”), the user can open a webpage on the PC (where the ad hoc network was created) with the IP address of the module and the HTTP client application (e.g. Internet Explorer).

If the module is configured as limited AP, the DHCP and DNS server should be started prior to issuing this command. Once the module returns the success response (“OK”), the user can open a webpage on the PC or smartphone that is connected to the limited AP.

User can configure both L2 and L3 level information on the provisioning web pages. Submit button stores all the configured information in the module and logout/boot button presents all provisioned information to the serial host and resets the module.

The size of the username and password is limited to 16 characters.

The provisioned information sends to serial host is:

- SSID=<ssid>
- CHNL=<channel>
- CONN_TYPE=<connType> /* either BSS or IBSS */
- MODE=<mode> /* 0 –> 802.11b */
- WEP_ID=<wep ID>
- WEP_KEY=<wep key>
- PSK_PASS_PHRASE=<psk PassPhrase>
- EAP_USER_NAME=<eap User name>
- EAP_PASS_WORD=<eap PassWord>
- PRIVATE_KEY_LEN=<private Key Length>
- PRIVATE_KEY=<private key file> /* private key file is stream of bytes of length= private Key Length
- CLIENT_CERT_LEN=<client Certificate Length>
- CLIENT_CERT =<client certificate> /* client certificate is stream of bytes of length= client Certificate
- CA_CERT_LEN=<CA certificate Length>
- CA_CERT =<CA certificate> /* CA certificate is stream of bytes of length= CA Certificate
- DHCP_ENBL=<0/1>
- STATIC_IP=<static IP address>
- SUBNT_MASK=<subnet Mask>
- GATEWAY_IP=<gateway>
- AUTO_DNS_ENBL=<0 /1>
- PRIMERY_DNS_IP=<primary DNS server IP>
- SECNDRY_DNS_IP<secondary DNS IP>
- NEW_USER_NAME<new User Name>
- NEW_PASS=<new Password>

Web Provisioning (Logo)

Command: **AT+WEBLOGOADD=<size>**

<Esc>L<Actual File content>

Response: *No response.*

Description: The module supports adding the Logo that will appear on the web pages used for provisioning.

<size> is measured in bytes and the maximum size is 1788 bytes. This command is typically done at the manufacturing line in the factory. This command can be done only once. There is no command to delete the Logo.

RF Tests

The module supports different types of frame transmission for RF capability measurement. It supports asynchronous data transmission/reception and modulated/un-modulated wave transmission.

Asynchronous Frame Transmission

Command: **AT+RFFRAMETXSTART=<Channel>,<Power>,<Rate>,<No.Of.Times>,<Fr.Intrvel>, <FrameControl>,<DurationId>,<Sequence Control>,<frameLen>,<Preamble>,<Scrambler>[,<DstMac>,<Src Mac>]**

Response: *OK*

Description: This command will enable the asynchronous frame transmission.

This command enables the asynchronous data transmission with the parameter configured. After issuing this command the user needs to send the payload data as following:

<ESC>A/a <PayLoad of the above Frame length>

- Channel: the channel on which the data to be send.
- Power: the power in db at which the frame to be sent. The value of this parameter can range from 2 to 15.
- Rate: the rate at which the data can be sent and the possible values are:

  RATE_1MBPS = 2,

  RATE_2MBPS = 4,

  RATE_5.5MBPS = 11,

  RATE_11MBPS = 22
- No. Times: the number of asynchronous frames to be sent. (1-65535)
- Fr. Interval: the interval between each frame, in microseconds. (1-65535)
- Frame Control: expects only the lower byte (B0...B7) of 802.11 frame control field, which includes protocol version, Type and Subtype. All the higher order bits (B8...B15) are made zero for this command.

For example. frame control field of beacon frame is: 128

Higher Byte B15 – B8	Sub Type B7-B4	Type B3- B2	Protocol Version B1 – B0
00000000	1000	00	00

1. Note: This command is intended to transfer only data & a few Management frames like Beacon/Probe request/Probe response/Association request.

Table 3-13: Frame Control Information

- DurationId: duration id information to be sent. (0-65535)
- Sequence Control: the sequence number of the frame. This field consists of 12 bits(LSB) fragment number and 4 bit (MSB)sequence number. (0-65535)
- frameLen: the length of the payload. The maximum size of the frame is limited to 1400 bytes.
- Preamble: the short (1) or long (0) preamble.
- Scrambler: the ON(0) or OFF(1) scrambler field of the frame
- DstMac: the MAC address through which the frame to be send.
- Src Mac: MAC address for the WiFi Bridge.

Example:

AT+RFFRAMETXSTART=1,3,4,2,200,0,11,0,30,0,1,00:1d:c9:00:07:a2

<ESC>A12345678901234567890123456789O

Please check the wireless sniffer to see the frame on air.

The AT+RFSTOP (paragraph heading SPI Interface Handling) command should be issued prior to successive frame transmission command.

CSMA/CA is not executed before transmitting this command; hence it could destroy the network.

Asynchronous Frame Reception

Command: **AT+RFRXSTART=<Channel>[,<Sendtouser>]**

- Channel: is the channel on which the data to be received.
- Sendtouser: is a flag (0/1) which instructs the module to send the received data to the serial interface.

Response: `OK`

Description: The Frame Transmission/Reception Stop command (paragraph heading Frame Transmission/Reception Stop) will send the status information of the received frames to the serial interface.

Example:

AT+RFRXSTART=1,1 → this will send the received data to the serial interface

AT+RFRXSTART=1,0 → this will not send the received data to the serial interface

In both case the received frame information is stored in SRAM and once issue the command AT+RFSTOP sends the received frame information to the user through serial. We recommend using the second option.

Modulated/Un-Modulated Wave Transmission

Command: **AT+RFWAVETXSTART=<Modulated>,<Channel>,<Rate>,<PreambleLong>,<ScamblerOff>,<Cont.Tx>, <Power>,<Ssid>**

Response: *OK*

The response will be ERROR if the command fails.

Description:

- Modulated : is the flag to tell whether the wave transmission should be modulated(1) or un-modulated (0)
- Channel: is the channel on which the data to be received.
- Rate: the rate at which the wave transmission should happen.

  TX_RATE 1mbps = 0,

  TX_RATE 2 mbps = 1,

  TX_RATE 5.5 mbps = 2,

  TX_RATE 11 mbps = 3,
- PreambleLong: is long preamble (1) or short preamble (0).
- ScamblerOff: is the scrambler field OFF (0) or ON (1).
- Cont.Tx: is the wave transmission is continuous (1) or not (0).
- Power: is the power in db at which the wave transmission should happen. The value of this parameter can range 2 to 15.
- Ssid: is the ssid of the network created for the wave transmission.

Example:

AT+RFWAVETXSTART=1,4,2,1,1,1,3,aaa -→(modulated)

AT+RFWAVETXSTART=0,4,3,0,1,1,3,bbb --→(un-modulated)

Frame Transmission/Reception Stop

Command: **AT+RFSTOP**

Response: *OK*

Description: This command will stop any of the RF tests transmission/reception.

Upon reception of this command the module stops any of the frame transmission/reception RF tests started. This command sends the status information of the received asynchronous frames to the serial interface other than the normal command response if this command issued for the asynchronous frame reception stop.

Example:

AT+RFSTOP (if this command issued after AT+ RFRXSTART, then it sends the following information to the serial interface)

Total frames received =xxxx

Correct frames received =xxxx

Incorrect frames received =xxx

FCS Error frames received =xxx

Enhanced Asynchronous Notification

Command: **AT+ASYNCMSGFMT=n**

n is:

0 – Disable this feature

1 – Enable this feature

Response: Refer to Command Processing Mode in this chapter.

Description: WLM10x Module supports an enhanced asynchronous notification method. This command will enable/disable this feature.

Enabling this feature results with all asynchronous messages going to the serial interface with a header. Also during these asynchronous message transfer WLM10x module make the gpio 19 high. The asynchronous message format is as shown below:

<ESC><TYPE><SUBTYPE><LENGTH><MESSAGE>

TYPE – Type of message and the length is one byte. For asynchronous message , it is 0x41 (Ascii value A)

SUBTYPE – Message subtype and the length of this field is one byte. Normally this field contains the ascii value of the subtype message. Refer to paragraph heading Enhanced Asynchronous Messages for subtype values.

LENGTH – Length of the asynchronous message in hex. This field length is 2 bytes.

MESSAGE – Exact asynchronous message as string. Refer to paragraph heading Enhanced Asynchronous Messages for all enhanced asynchronous messages.

Node Start Up Handling

Command: **N/A**

Response:

Description: For proper synchronization between host micro controller (MCU) and WLM10x node, the following steps must be followed:

In case of UART interface, during boot up host MCU shall send dummy ‘AT’ command and wait for response from the WLM10x node. The host MCU must continuously send these dummy ‘AT’ commands till ‘OK’ response is received from WLM10x node.

In case of SPI interface, during boot up host MCU must check the status of host wake-up signal (GPIO#28 of WLM10x node). Once host wake-up signal is HIGH, then host MCU can send the ‘AT’ commands.

If for some reason host MCU getting reset, then WLM10x module must be explicitly reset using EXT_RESET pin and the MCU should wait for the wake-up signal(GPIO#28) become high in case of SPI interface. However if reset provision is not available, then host MCU must continuously send dummy ‘AT’ commands till ‘OK’ response is received from WLM10x module.

Firmware Upgrade

Command: **AT+FWUP= <SrvIp>,<SrvPort>,<SrcPort>,[<retry>]**

Response: *When a valid command has been received, the module returns the message:* `Firmware upgrade is going on, Please wait....` *followed with the status message OK or 0, which applies only to the validity of the command. After attempting to upgrade the firmware, the node sends an additional message describing the result of the actual firmware upgrade attempt.*

After a successful firmware upgrade, the module will reset and boot up using the updated firmware; when startup is complete, it will issue the message `APP Reset-FW-UP-SUCCESS`.

If the firmware upgrade attempt failed, the module will reset and boot up with the old firmware, and issue the message `APP Reset-FW-UP-FAILURE`.

If the firmware upgrade attempt failed after successful upgrade of one flash image (flash0), the module will reset and boot up, issue the message APP Reset-FW-UP-RECOVERY, associate back to the network with previous settings and try to upgrade the firmware again. The retry count decides how many times this can be done.

If the node is not associated, the module returns `ERROR` *or* `1`*, based on verbose settings.*

Description: This command starts the firmware upgrade procedure over the wireless link.

- SrvIp is the IP address of the firmware upgrade server;
- SrvPort is the server port number to be used for firmware upgrade;
- SrcPort is the module port number to be used for firmware upgrade.
- Retry is the number of times the node will repeat the firmware upgrade attempt if failures are encountered. The default value is 10 and the retry count ranges from 0 to 0xffffffff.

SPI Interface Handling

Command:

Response:

Description: In the case of SPI interface, the WLM10x node acts as slave and will communicate to master SPI controller. By default, SPI interface supports Motorola protocol with clock polarity 0 and clock phase 0. For more detailed specification of SPI frame format and timing characteristics refer to WLM10x data sheet.

Since SPI data transfer works in full duplex mode, its required to make use of special octet to indicate idle data. Similarly if host MCU is sending data at higher rate flow control mechanism is required. In order differentiate these special control codes (such as idle pattern , flow control codes and other control octets) from user data, byte stuffing mechanism is incorporated.

SPI transmit data handling procedure:

The SPI data transfer layer makes use of an octet (or byte) stuffing procedure. The Control Escape octet is defined as binary 11111011 (hexadecimal **0xFB**), most significant bit first. Each special control pattern is replaced by a two octet sequences consisting of the Control Escape octet followed by the original octet exclusive-or'd (XOR) with hexadecimal **0x20**. Receiving implementations must correctly process all Control Escape sequences.

Escaped data is transmitted on the link as follows:

Pattern	Encoded as	Description
0xFD	0xFB 0xDD	***Flow control XON***
0xFA	0xFB 0xDA	***Flow control XOFF***
0x00	0xFB 0x20	Inactive link detection
0xFB	0xFB 0xDB	Control ESCAPE
0xF5	0xFB 0xD5	***IDLE character***
0xFF	0xFB 0xDF	Inactive link detection
0xF3	0xFB 0xD3	SPI link ready indication

Table 3-14: Escaped Data

One dedicated GPIO signal (***GS_SPI _HOST_WAKEUP***: ***GPIO#28)*** is available for data ready indications from Slave WLM10x node to Master Host controller. This ***GS_SPI _HOST_WAKEUP*** signal is asserted high during valid data transmission period, so that the host (master SPI) starts pulling out data by giving SPI clock and ***GS_SPI _HOST_WAKEUP*** signal is de-asserted once transmission is completed. Master host controller must provide clock as long as ***GS_SPI_HOST_WAKEUP*** signal is active.

Special character (***GS_SPI _IDLE***) will be transmitted during idle period (if there is no more data to transmit) and must be dropped at the receiving Host.

SPI receive data handling procedure:

Since byte stuffing is used, each Control Escape octet must be removed and the next immediate octet is exclusive-or'd (XOR) with hexadecimal **0x20**. If received buffer has reached the upper water mark, then ***XOFF*** character will be sent out informing the host to stop transmitting actual data. After receiving ***XOFF*** character host must stop transmitting actual data and can send IDLE bytes, until the XON is received. Once the host receives ***XON***, then it may resume the valid data transmissions.

Special control byte ***IDLE*** will be dropped at receiver.

Pin Connection for SPI Interface

Host MCU	WLM10X Node	Remarks
MSPI_DOUT	SSPI_DIN	
MSPI_DIN	SSPI_DOUT	
MSPI_SS	SSPI_SS	
MSPI_CLK	SSPI_CLK	
GPIO	GPIO#28	Host wake-up signal
Ground	Ground	

Table 3-15: Pin Connection for SPI Interface

Factory Defaults

Command:

Response:

Description: The WLM10x module stores the factory defaults to its flash; currently supporting only the MAC addresses as factory the default. If the factory default MAC address location contains a valid address, then the WLM10x module reads and uses it as the MAC address, otherwise it use the default MAC as it MAC address.

The factory default location starts at 122Kbytes of second application flash and the WLM10x stores the factory default in the following format:

Checksum(1 byte)	Length (1 byte)	Mac address (6 byte)

Checksum : the simple byte wise xor of both length and MAC address.

Length : the length in bytes of MAC address and length (here it is 7).

Mac Address : the MAC address. The user can override the factory default MAC address by using the AT commands mentioned in paragraph heading MAC Address Configuration.

Set System Time

Command: **AT+SETTIME=<dd/mm/yyyy>,<HH:MM:SS>**

Response:

Description: Upon execution of this command the module set its system time to the time specified as the parameters and returns the standard command response.

Get System Time

Command: **AT+ GETTIME=?**

Response:

Description: Upon reception of this command the module sends the current system time in milliseconds since epoch(1970) followed by the standard command response to the serial interface. The time format comes on the serial interface as follows:

“Current Time in msec since epoch=xxxxxxx”

GPIO Out HIGH/LOW

Command: **AT+DGPIO=<GPIO-NO>,<SET/RESET(0/1)>**

Response:

Description: This command will set/reset (high/low) a gpio pin. This command sets the Gpio 'GPIO-NO' pin level to high or low as per the SET/RESET parameter and returns the standard command response (paragraph heading Commands for Command Processing Mode in this chapter)

Note: Only the Gpio Pins which are not mixed with the any used IOs like UART/SPI etc. that can be set high/low with this command.

The supported Gpios and the corresponding numbers are:
Gpio10 : 10
Gpio11 : 11
Gpio30 : 30
Gpio31 : 31

Error Counts

Command: **AT+ERRCOUNT=?**

Response: *OK*

The error counts include:

- *Watchdog reset counts*
- *Software reset counts*
- *Wlan abort/assert counts*

Description: This command will get the error count statistics.

Version

Command: **AT+VER=?**

Response: *OK*

Serial-to-Wi-Fi version;
Laird Technologies Embedded Platform Software version;
WLAN firmware version.

Description: This command will output the current version information.

Ping

Command: **AT+PING=<Ip>,[[<Trails>],[<Interval>],[<Len>],[<TOS>],[<TTL>],[<PAYLOAD>]]**

Response: *No response*

Description: This command will initiate a network *ping.*

Upon deployment of this command the device sends a *ping* to the remote machine specified by the IP address.

- Ip is the IP address of the server to which the command is directed;
- Trails indicate the number of *ping* requests to send. The default value is 0; in this case, *ping* will continue until terminated as described below.
- Interval is the interval in milliseconds between each *ping* request; the valid range is 1000-99000. The default value is 3000.
- Len is the length of the *ping* packet; the valid range is 0 to 1024.The default value is 56.
- TOS is the type of service; the valid range is 0-99. The default value is 0.
- TTL is the time to live; the valid range is 0-2047. The default value is 30.
- Payload is the data to be sent in each *ping* packet. The payload length should be in the range 0-16; the payload may contain valid alphanumeric characters (0-9, a-e).

To terminate a Ping sequence, issue `<Esc> C`.

Trace Route

Command: **AT+TRACEROUTE=<Ip>,[[<Interval>],[<MaxHops>],[<MinHops>],[<TOS>]]**

Response: *Upon reception of this command the module starts the trace route operation and returns* `OK`

<LF>Tracing Route to<space><Ip address><space>over a max hops<space>< `MaxHops`*><CR><LF>*

Description: This command will start a *trace route* operation.

Parameters:

- Ip is the IP address of the remote server;
- Interval is the interval in milliseconds between each request; the valid range is 1000-99000. The default value is 1000.
- MaxHops is the maximum time-to-live; the valid range is 2-99. The default value is 30.
- MinHops is the minimum time-to-live; the value given should be greater than 1 and less than MaxHops. The default value is 1.
- TOS is the type of service; the valid range is 0-99. The default value is 0.

During this trace route operation the module sends the ping delays and the next hop ip address information to the serial interface one at a line in the following format:

<CR><LF><current TTL ><2 space><1st RTT in 4 bytes>ms<2 space><2nd RTT in 4 bytes>ms<2 space><3rd RTT in 4 bytes>ms<2 space><ip address of hop>

Once the trace route operation complete, the module sends the message"<CR><LF><CR><LF> Trace Complete<CR><LF>" to the serial interface.

Memory Trace

Command: **AT+MEMTRACE**

Response: *OK.*

Description: This command will display the module memory trace information.

The memory trace information contains the following :

- Number Of Allocation
- Number Of Free
- Current Used Memory in bytes
- Peak Memory Usage in bytes
- Memory Details of currently used allocations in the following format:

  <address>,<line number>,<size>,<module name>

- Number of Allocations to be freed

Chapter 4
Data Handling using Esc Sequences

Data handling using Esc Sequences on UART Interface

Flow Control	Data Mode (Data Type)	Connection Type	Description and Escape Command Sequence
SW or HW	Normal (ASCII Text)	TCP client TCP server	This escape sequence selects the specified Connection ID as the current connection. This switches the connection to be used without exiting from the Data mode of operation. Use this sequence to send data from a TCP server, TCP client or UDP client (must be done before data can be received by that client). WLM10x send and receive sequence: <Esc>S<CID><data><Esc>E Example: To send user data (e.g. Hello) on CID 1, the format will be: <Esc>S1Hello<Esc>E
SW or HW	Normal (ASCII Text)	UDP client	If UDP client is configured with unicast destination server IP address, then: WLM10x send and receive sequence: <Esc>S<CID><data><Esc>E If UDP client is configured with broadcast destination server IP address (i.e. 255.255.255.255), then: WLM10x expects to receive the following data sequence from Host: <Esc>S<CID><data><Esc>E WLM10x sends the following data sequence to Host: <Esc>u<CID><IPAddress><space><port><horizontal tab<data><Esc>E
SW or HW	Normal (ASCII Text)	UDP server	This escape sequence is used when sending and receiving UDP data on a UDP server connection. When this command is used, the remote address and remote port is transmitted. WLM10x expects to receive the following data sequence from Host: <Esc>U<CID><IP Address>:<port>:<data><Esc>E WLM10x sends the following data sequence to Host: <Esc>u<CID><IPAddress><space><port><horizontal tab<data><Esc>E Example: When WLM10x sends data (e.g. Hello) on CID 0, the format will be: <Esc>u0192.168.0.101<space>1001<horizontal tab>Hello<Esc>E
SW or HW	Normal (Binary)	NA	Binary data transfer with software or hardware flow control are not supported with ESC sequence.

Flow Control	Data Mode (Data Type)	Connection Type	Description and Escape Command Sequence
SW or HW	Bulk (ASCII Text)	TCP client TCP server	To improve data transfer speed , one can use this bulk data transfer. This sequence is used to send and receive data on TCP client, TCP server, or UDP client connection. WLM10x send and receive sequence: <Esc>Z<CID><data length><data> Example: To send a 5 byte user data (e.g. Hello) on CID 1, the format will be: <Esc>Z10005Hello
SW	Bulk (ASCII Text or Binary)	UDP client	If UDP client is configured with an unicast destination server IP address, then WLM10x send and receive sequence: <Esc>Z<CID><Data Length><data> If UDP client is configured with a broadcast destination server IP address (i.e. 255.255.255.255), then: WLM10x expects to receive the following data sequence from Host: <Esc>Z<CID><Data Length><data> WLM10x sends the following data sequence to Host: <Esc>y<CID><IPAddress><Space><Port><horizontal tab><data length><data>
SW or HW	Bulk (ASCII Text)	UDP server	This escape sequence is used when sending and receiving UDP bulk data on a UDP server connection. When this command is used, the remote address and remote port is transmitted. WLM10x expects to receive the following data sequence from Host: <Esc>Y<CID><IP address>:<port>:<data length><data> WLM10x sends the following data sequence to Host: <Esc>y<CID><IPAddress><Space><Port><horizontal tab><data length><data> Example: When receiving a 5 byte user data (e.g. Hello) on CID 1, the format will be: <Esc>y0192.168.0.101<space>1001<horizontal tab>0005Hello
HW	Bulk (Binary)	TCP client TCP server UDP client	To improve data transfer speed , one can use this bulk data transfer. This sequence is used to send and receive data on TCP client, TCP server, or UDP client connection. WLM10x send and receive sequence: <Esc>Z<CID><data length><data> Example: To send a 5 byte user data (e.g. Hello) on CID 1, the format will be: <Esc>Z10005Hello
SW	Bulk (Binary)	NA	Binary data transfer with software flow control not supported.

DATA HANDLING USING ESC SEQUENCES

Data Handling Using Esc Sequences On SPI Interface

Data Mode (Data Type)	Connection Type	Description and Escape Command Sequence
Normal (ASCII Text)	TCP client TCP server	1. Data transfer is transparent due to byte stuffing at SPI driver level. 2. Byte stuffing must be incorporated in Host controller as per the Adaptor guide. WLM10x send and receive sequence: <Esc>S<CID><data><Esc>E or Auto mode
Normal (ASCII Text)	UDP client	If UDP client is configured with an unicast destination server IP address, then: WLM10x send and receive sequence: <Esc>S<CID><data><Esc>E If UDP client is configured with a broadcast destination server IP address (i.e. 255.255.255.255), then: WLM10x expects to receive the following data sequence from MCU: <Esc>S<CID><data><Esc>E WLM10x sends the following data sequence to MCU: <Esc>u<CID><IP Address><space><port><horizontal tab<data><Esc>E
Normal (ASCII Text)	UDP server	This escape sequence is used when sending and receiving UDP data on a UDP server connection. When this command is used, the remote address and remote port is transmitted. WLM10x expects to receive the following data sequence from Host: <Esc>U<CID><IP Address>:<port>:<data><Esc>E WLM10x send the following data sequence to Host: <Esc>u<CID><IP Address><space><port><horizontal tab<data><Esc>E Example: When receiving user data (e.g. Hello) on CID 0, the format will be: <Esc>u0192.168.0.101<space>1001<horizontal tab>Hello<Esc>E
Normal (Binary)	NA	Binary data transfer with software flow control is not supported with ESC sequence.
Normal (ASCII Text or Binary)	NA	Hardware flow control is not supported.
Bulk (ASCII Text or Binary)	TCP client TCP server	1. Data transfer is transparent due to byte stuffing at SPI driver level. 2. Byte stuffing must be incorporated in Host controller as per the Adaptor guide. WLM10x send and receive sequence: <Esc>Z<CID><Data Length><data> Example: To send a 5 byte user data (e.g. Hello) on CID 1, the format will be: <Esc>Z10005Hello
Bulk (ASCII Text or Binary)	UDP client	If UDP client is configured with an unicast destination server IP address, then: WLM10x sends and receives the following data sequence: <Esc>Z<CID><Data Length><data> If UDP client is configured with a broadcast destination server IP address (i.e. 255.255.255.255), then: WLM10x expects to receive the following data sequence from Host: <Esc>Z<CID><Data Length><data> WLM10x sends the following data sequence to Host: <Esc>y<CID><IP Address><Space><Port><horizontal tab><data length><data>
Bulk (ASCII Text or Binary)	UDP server	This escape sequence is used when sending and receiving UDP bulk data on a UDP server connection. When this command is used, the remote address and remote port is transmitted. WLM10x receives from Host the following data sequence: <Esc>Y<CID><IP address>:<port>:<data length><data> WLM10x sends the following data sequence to Host: <Esc>y<CID><IP Address><Space><Port><horizontal tab><data length><data> Example: When receiving a 5 byte user data (e.g. Hello) on CID 1, the format will be: <Esc>y0192.168.0.101<space>1001<horizontal tab>0005Hello

Chapter 5
Mechanical Considerations

Package and Layout Guidelines

WLM10x Recommended PCB Footprint and Dimensions

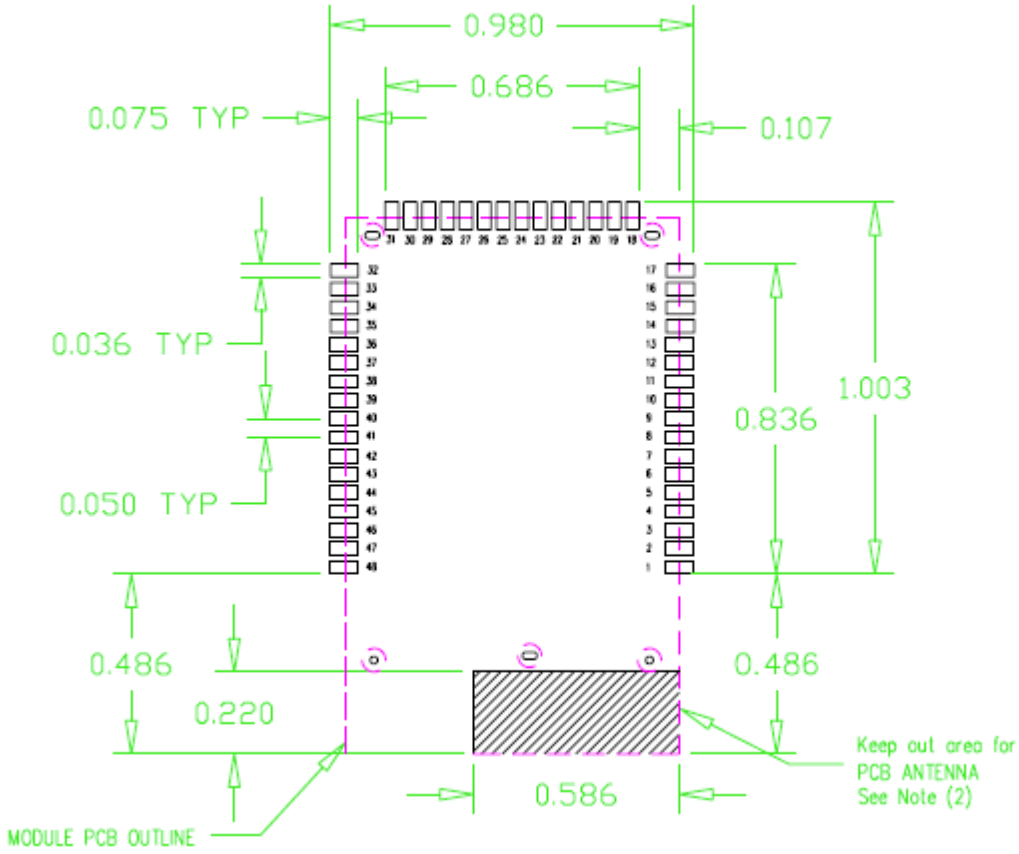


Figure 9-12: WLM10x Module Recommended PCB Footprint (dimensions are in inches)

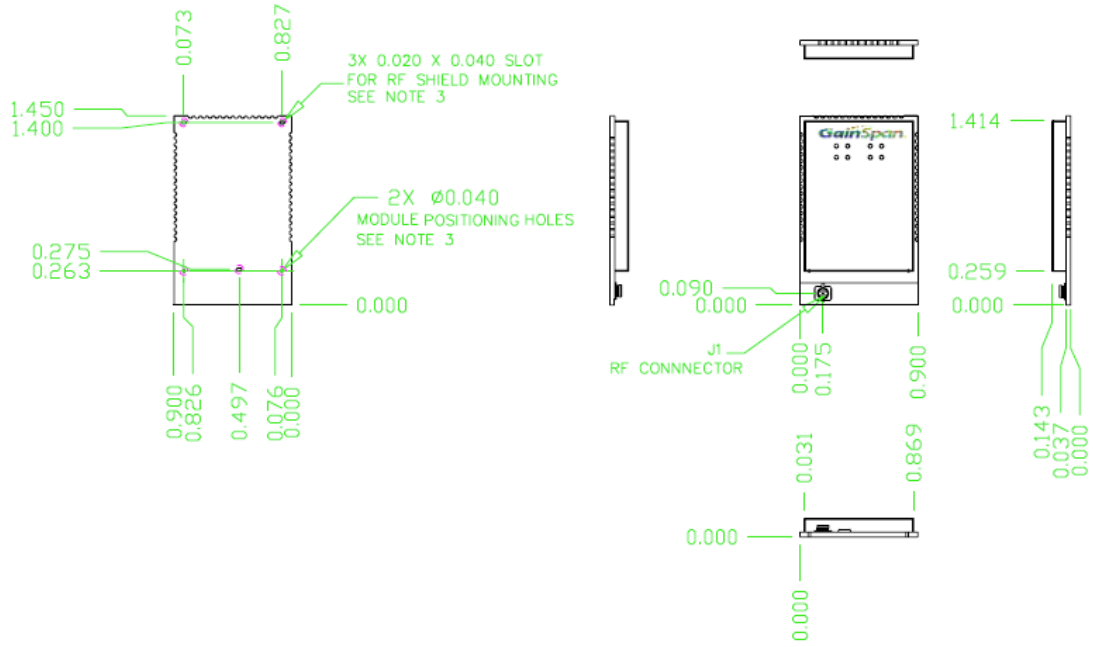


Figure 9-13: WLM10x Module Dimensions (in inches)

WLM10xxP Layout Guidelines

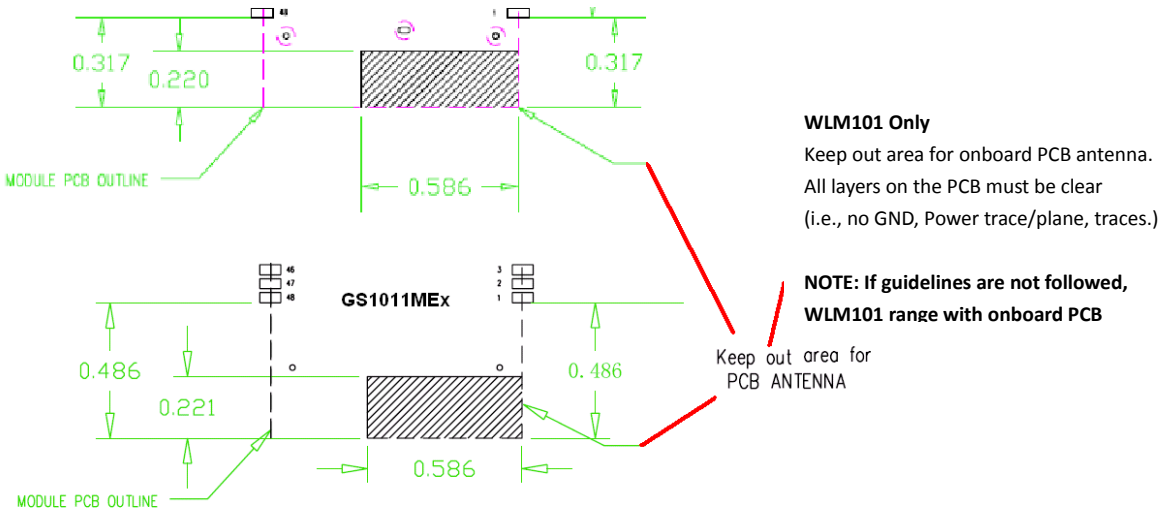


Figure 9-14: WLM101 Module Onboard Antenna End Application Board
Keep-Out Layout Guidelines (Dimensions are in inches)

Notes:

1. All Dimensions are in inches. Tolerances shall be ±0.010 inches.
2. Absolutely no metal trace or ground layer underneath this area
3. It is recommended not to run circuit traces underneath the module especially near these holes; The RF shield mounting holes are grounded.
4. In performing SMT or manual soldering of the module to the base board, first align the row of pins from #18 thru 31 onto the base board and then match the other two rows.

In addition to the guidelines note the following suggestions:

WLM10x

- External Bypass capacitors for all module supplies should be as close as possible to the module pins.
- Never place the antenna very close to metallic objects.
- The external dipole antennas need a reasonable ground plane area for antenna efficiency.

WLM101 onboard PCB antenna specific

- The PCB antenna keep out area, as shown in Figure 5-4, must be adhered to (i.e. No ground, power trace/plane, traces; all layers of PCB, in the keep out area, must be clear), or the over the air range of the WLM10xxP will be compromised.
- Do not use a metallic or metalized plastic for the end product enclosure.
- Recommendation is to keep plastic enclosure clearance of 1cm from top and bottom of the WLM10x PCB antenna keep-out area, if possible. 5-mm (0.2 in) clearance shall be the minimum as shown in the Figure 9-15titled Recommended Clearance Above And Below The WLM101 Trace Antenna in the following figure.

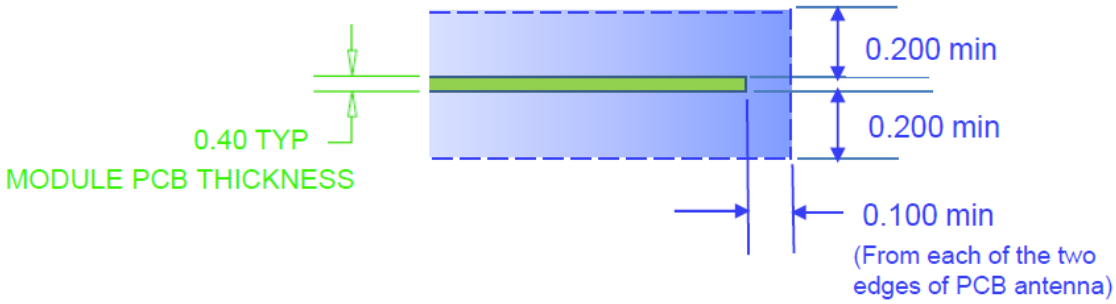


Figure 9-15 Recommended Clearance Above And Below The WLM101 Trace Antenna

Surface Mount Assembly

The reflow profile1 is shown in the following Figure titled Reflow Temperature Profile. Recommended reflow parameters are summarized in the following table titled Recommended Reflow Parameters**.**

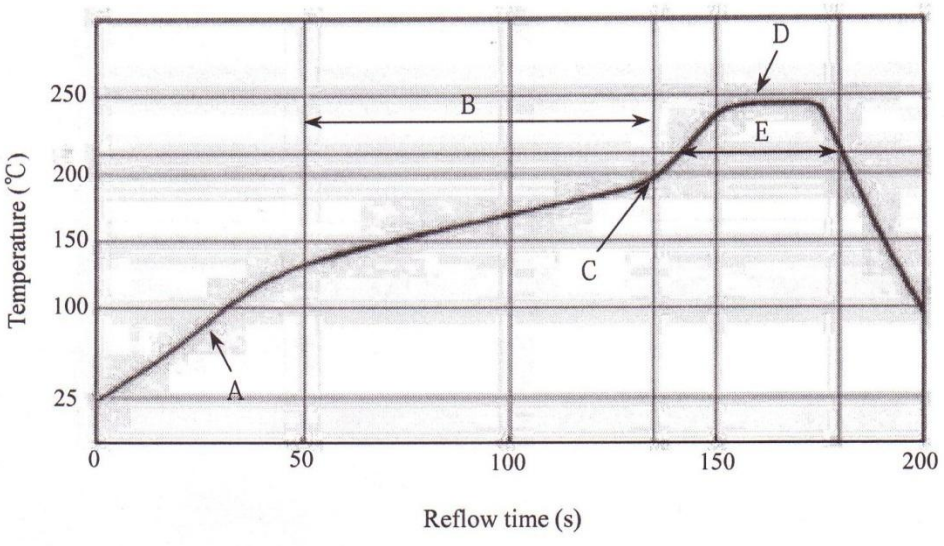


Figure 9-16: Reflow temperature profile.

PreHeat	
Temperature Ramp up rate for (A)[2]	2~4 °C/s
Pre-heat time (B)[3]	60 to 120 seconds
Pre-heat ending temperature (C)[4]	180 to 200 ° C
Heating [5]	
Peak Temperature range (D)	230 to 240 °C
Melting time that is the time over 220 °C (E)	20 to 40 seconds

Table 9-1: Recommended Reflow Parameters.

Notes:

1. Perform adequate test in advance as the reflow temperature profile will vary accordingly to the conditions of the parts and boards, and the specifications of the reflow furnace.
2. Be careful about rapid temperature rise in preheat zone as it may cause excessive slumping of the solder paste.
3. If the preheat is insufficient, rather large solder balls tend to be generated. Conversely, if performed excessively, fine balls and large balls will generate in clusters at a time.
4. If the temperature is too low, non-melting tends to be caused in the area with large heat capacity after reflow.
5. Be careful about sudden rise in temperature as it may worsen the slump of solder paste.

Be careful about slow cooling as it may cause the positional shift of parts and decline in joining strength at times.

Chapter 6
FCC Regulatory Statements

Regulatory Notes

Federal Communication Commission Interference Statement
This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

FCC Caution: To assure continued compliance, (example - use only shielded interface cables when connecting to computer or peripheral devices). Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

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.

IMPORTANT NOTE:

FCC & IC Radiation Exposure Statement:
This equipment complies with FCC & IC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter. This device is intended only for OEM integrators under the following conditions:

1) The antenna must be installed such that 20 cm is maintained between the antenna and users, and

2) The transmitter module may not be co-located with any other transmitter or antenna.

As long as 2 conditions above are met, further transmitter test will not be required. However, the OEM integrator is still responsible for testing their end-product for any additional compliance requirements required with this module installed (for example, digital device emissions, PC peripheral requirements, etc.).

IMPORTANT NOTE: In the event that these conditions cannot be met (for example certain laptop configurations or co-location with another transmitter), then the FCC & IC authorizations are no longer considered valid and the FCC & IC IDs cannot be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining separate FCC & IC authorizations.

FCC REGULATORY STATEMENTS

End Product Labeling

This transmitter module is authorized only for use in device where the antenna may be installed such that 20 cm may be maintained between the antenna and users (for example access points,routers, wireless ADSL modems, and similar equipment). The final end product must be labeled in a visible area with the corresponding FCC ID number.

IC Certification — Canada

The labeling requirements for Industry Canada are similar to those of the FCC. A visible label on the outside of the final product must display the IC labeling. The user is responsible for the end product to comply with IC ICES-003 (Unintentional radiators)

Manual Information That Must be Included

The user's manual for end users must include the following information in a prominent location.

IMPORTANT NOTE: To comply with FCC & IC RF exposure compliance requirements, the antenna used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter.

Other notes:

Laird Technologies modules have been built or under development for near body exposure applications. The 20cm statement is a standard note because absorption rate testing (commonly known as SAR or Specific absorption rate) is not modularly transferable for FCC/IC. Thus, if a radio is being used against the body, the end user is still responsible to test for regulatory near body exposure testing (for USA, please refer to the following):

- FCC Part 1.1037
- FCC Part 2.1091 Mobile Devices
- FCC Part 2.1093 Portable Devices
- FCC Part 15.247 (b) (4)

Chapter 7
Ordering Information

Device Description	Ordering Number
Low power 802.11b WiFi module with PA, PCB Trace Antenna	WLM101
Low power 802.11b WiFi module with PA, Antenna Connector	WLM100

Chapter 8
Limitations

This device and associated software are not designed, manufactured or intended for use or resale for the operation of application in a hazardous environment, or requiring fail-safe performance, or in which the failure of products could lead directly to death, personal injury, or severe physical or environmental damage (collectively, "high risk applications"). You agree and acknowledge that you have no license to, and shall not (and shall not allow a third party to) use the technology in any high risk applications, and licensor specifically disclaims any warranty regarding, and any liability arising out of, high risk applications.

Chapter 9
References

[1]	
[2]	IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 802.11-2007, IEEE, www.ieee.org
[3]	V.250, Serial asynchronous automatic dialing and control, and V.251, Procedure for DTE-controlled call negotiation, International Telecommunications Union, www.itu.int
[4]	**Communications Networks,** A. Leon-Garcia and I. Widjaja, McGraw-Hill 2000, p. 582
[5]	**AN023 – Battery Level Check,** Laird Technologies Corporation

global solutions: local support™

USA: +1.800.492.2320

Europe: +44.1628.858.940

Asia: +852.2268.6567

wirelessinfo@lairdtech.com

www.lairdtech.com/wireless