Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!


## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

# AC4490

## User Guide
## Version 4.5


global solutions: local support™

Americas: +1-800-492-2320

Europe: +44-1628-858-940

Hong Kong: +852-2923-0610

Embedded Wireless Solutions Support Center:

http://ews-support.lairdtech.com

www.lairdtech.com/ramp

## REVISION HISTORY

| Version | Date | Changes | Approved By |
|---|---|---|---|
| 1.0 | | Initial Release | Chris Downey |
| 1.1 | 10/2012 | Major changes and revisions throughout document | Chris Downey |
| 2.0 | 04/5/2013 | Major changes and revisions; updated format and data | Chris Downey |
| 3.0 | -- | -- | Chris Downey |
| 4.0 | -- | -- | Chris Downey |
| 4.1 | 10 Dec 2013 | Updated download instructions for Laird Configuration Utility in section How do I configure Sync to Channel? | Chris Downey |
| 4.2 | 11 Dec 2013 | Separated into two separate docs (Hardware Integration Guide and User Guide).<br>Added a Related Documents section. | Sue White |
| 4.3 | 6 Feb 2013 | Fixed return data for EEPROM Byte Write. | Chris Downey |
| 4.4 | 15 Apr 2015 | Updated contact information and links to new website. | Sue White |
| 4.5 | 18 May 2017 | Removed references to EOL part numbers | Jonathan Kaye |

## FCC Notice

**WARNING:** 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.

## RF Exposure/Installation Instructions

**WARNING:** To satisfy FCC RF exposure requirements for mobile transmitting devices, this equipment must be professionally installed such that the end user is prevented from replacing the antenna with a non-approved antenna. The end user should also be prevented from being within 20cm of the antenna during normal use with the exception of hands, feet, wrists and ankles.

The preceding statement must be included as a CAUTION statement in manuals for OEM products to alert users on FCC RF Exposure compliance.

**Caution:** Any change or modification not expressly approved by Laird could void the user's authority to operate the equipment.

# CONTENTS

# AC4490 RF TRANSCEIVER

The compact AC4490 900 MHz transceiver can replace miles of cable in harsh industrial environments. Using field-proven frequency hopping spread spectrum (FHSS) technology which needs no additional FCC licensing in the Americas, OEMs can easily make existing systems wireless with little or no RF expertise.

## Overview

The AC4490 is a cost effective, high performance, frequency hopping spread spectrum transceiver designed for integration into OEM systems operating under FCC part 15.247 regulations for the 900 MHz ISM band.

AC4490 transceivers provide an asynchronous TTL level serial interface for OEM Host communications, which include both system and configuration data. The Host supplies system data for transmission to other Host(s). Configuration data is stored in the on-board EEPROM. All frequency hopping, synchronization, and RF system data transmission/reception is performed by the transceiver.

To boost data integrity and security, the AC4490 uses Laird's field-proven FHSS technology featuring optional Data-Encryption Standards (DES). Fully transparent, these transceivers operate seamlessly in serial cable replacement applications.

AC4490 transceivers can operate in Point-to-Point, Point-to-Multipoint, Client-Server, or Peer-to-Peer architecture. One transceiver is configured as a server with one or many client-configured transceivers synchronized to it. To establish synchronization between transceivers, the server emits a beacon; upon detecting a beacon an RF link is established and a GPIO is toggled to signify to the host that the link is active.

This document contains information about the hardware and software interface between a Laird AC4490 transceiver and an OEM Host. Information includes the theory of operation, specifications, interface definition, configuration information and mechanical drawings. The OEM is responsible for ensuring the final product meets all appropriate regulatory agency requirements listed herein before selling any product.

Additionally, this document contains a list of Related Documents and Files.

---

**Note:** Unless mentioned by name, the AC4490 module is referred to as the "radio" or "transceiver". Individual naming is used to differentiate product specific features. The host (PC / Microcontroller / Any device to which the AC4490 module is connected) will be referred to as "OEM Host".

---

# FEATURES

### Networking and Security

- Generic I/O digital lines and integrated DAC/ADC functions
- Retries and Acknowledgements
- API Commands to control packet routing and acknowledgement on a packet-by-packet basis
- Frequency Hopping Spread Spectrum for security and interference rejection
- Customizable RF Channel number and system ID
- Dynamic link analysis, remote radio discovery
- Low latency and high throughput
- Hardware Protocol Status monitoring

### Easy to Use

- Continuous 76.8 Kbps RF data stream
- Software selectable interface baud rates from 1200 bps to 115.2 Kbps
- Low cost, low power and small size ideal for high volume, portable and battery powered applications
- All modules are qualified for Industrial temperatures (-40° C to 85° C)
- Advanced configuration available using AT commands

# THEORY OF OPERATION

## RF Architecture

The AC4490 utilizes a server-client network architecture where all clients synchronize their hopping to the server. The server transmits a beacon during the first 20 milliseconds of every hop. The client transceivers listen for this beacon and, upon hearing it, assert their In_Range Low and synchronize hops with the server.

Each network consists of only one server. There should never be two servers on the same RF channel number in the same coverage area because the interference between the two servers will severely hinder RF communications. For those applications requiring collocated servers, Laird recommends using the Sync-to-Channel feature, further explained in Appendix I: Sync-to-Channel.

## Modes of Operation

The AC4490 has three different operating modes:

- Transmit mode
- Receive mode
- Command mode

If the transceiver is not communicating with another radio, it is in Receive mode actively listening for a beacon from the server. If the client determines that the beacon is from a server operating on the same RF channel and system ID, it responds by asserting In_Range Low. A transceiver enters Transmit or Command mode when the OEM host sends data over the serial interface. The state of the CMD Data pin (Pin 17) or the data contents determine which of the two modes the transceiver enters.

### Transmit Mode

All packets sent over the RF are either Addressed or Broadcast packets. Broadcast and Addressed delivery can be controlled dynamically with the API Control feature set, which can be enabled in the EEPROM configuration. To prohibit transceivers from receiving broadcast packets, Unicast Only can be enabled.

| | |
|---|---|
| Addressed Packets | When sending an addressed packet, the RF packet is sent only to the receiver specified in the destination address. To increase the odds of successful delivery, the packet uses transmit retries. Transparent to the OEM host, the sending radio sends the RF packet to the intended receiver. If the receiver receives the packet error-free it returns an RF Acknowledgement in the same 20 ms hop. If a Receive Acknowledgement is not received, the radio uses a transmit retry to resend the packet. This continues until either an acknowledgement is received or all transmit retries are used. The received packet is only sent to the OEM Host if and when it is received free of errors. |
| Broadcast Packets | When sending a broadcast packet, the RF packet is sent to every eligible transceiver on the network. To increase the odds of successful delivery, it uses broadcast attempts. Transparent to the OEM host, the sending radio sends the RF packet to the intended receiver(s). Unlike Transmit Retries, all broadcast attempts are used, regardless of when the RF packet is actually received. RF acknowledgments are not sent or received when a broadcast packet is transmitted. If the packet is received on the first attempt, the receiver ignores the remaining broadcast attempts. The received packet is only sent to the OEM host if and when it is received free of errors. |

## Receive Mode

When a transceiver is not in Transmit or Command mode, it is in Receive mode listening for data. While in Receive mode, subsequent data of up to 80 bytes can be received every hop (20 ms).

## Command Mode

A radio enters Command mode when data is received over the serial interface from the OEM host and contains the **AT+++** (Enter AT Command mode) command or when the state of the CMD/DATA pin is transitioned low. Once in Command mode, the radio interprets all data received as command data. Command data can be either EEPROM configuration or on-the-fly commands.

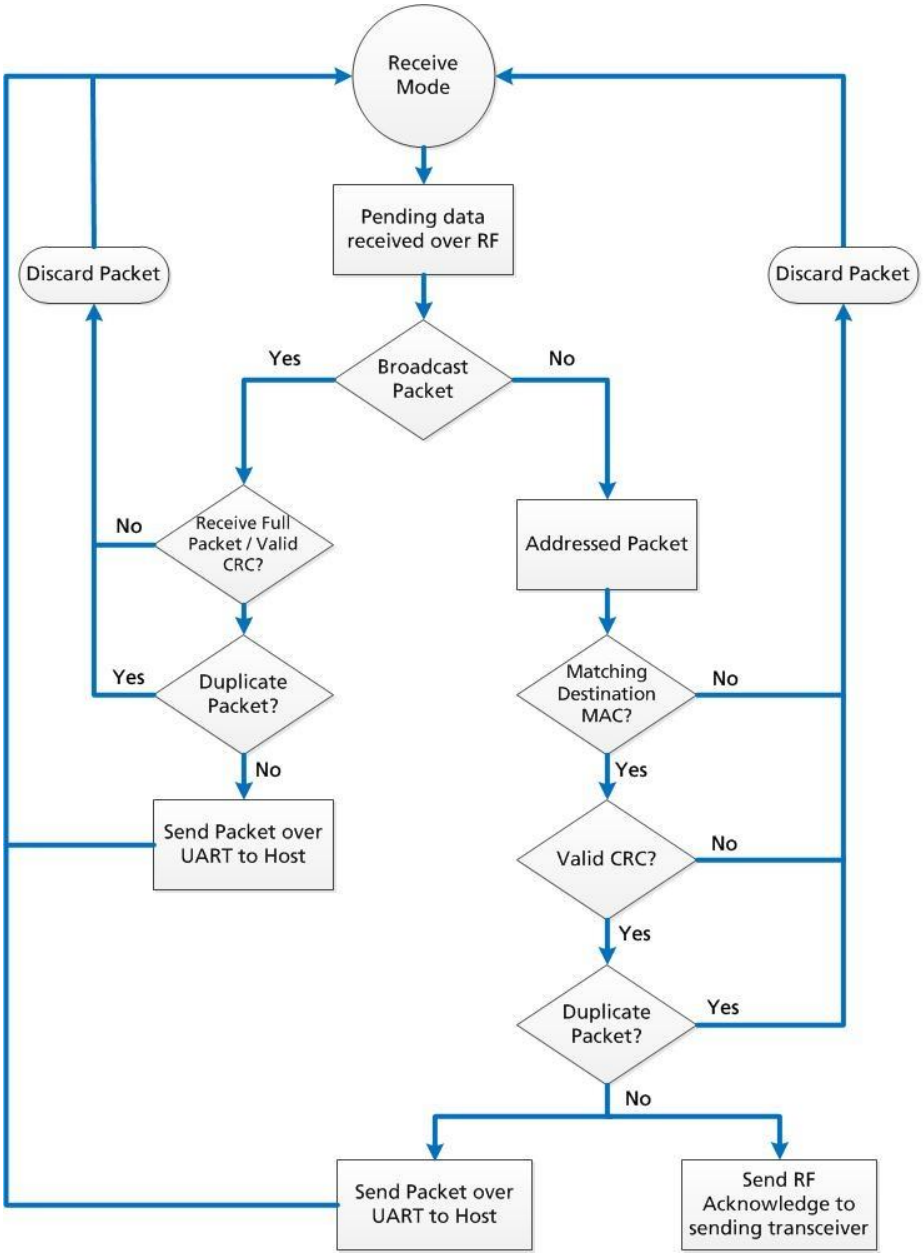


*Figure 1: Pending RF Buffer Flow*

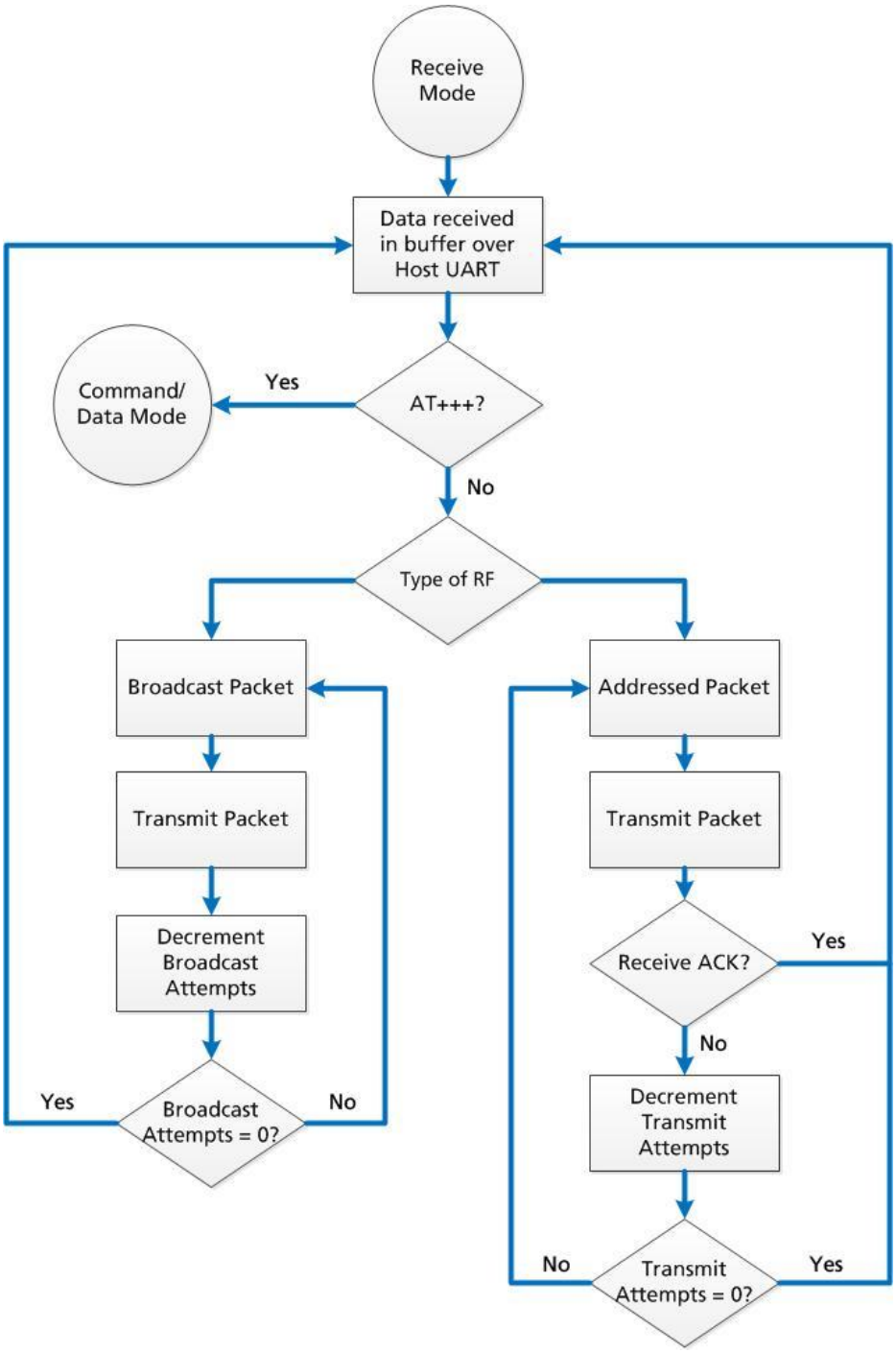


*Figure 2: Pending Data in Buffer Flow*

# AC4490 CONFIGURATION

## AT Commands

The AT Command mode implemented in the AC4490 creates a virtual version of the Command/Data pin. The "Enter AT Command Mode" Command asserts this virtual pin Low (to signify Command Mode) and the "Exit AT Command Mode" Command asserts this virtual pin High (to signify Data). Once this pin has been asserted Low, all On-the-Fly CC Commands documented in the manual are supported.

**Note:** The Command/Data RX Disable feature can be enabled in firmware versions 8.6+. When enabled in EEPROM, the radio disables the RF receiver while pin 17 (Command/Data) is Low. To ensure that the radio is not in the middle of transmitting data to the OEM Host, the host should be prepared to receive data for up to 20ms after taking pin 17 Low.

| | |
|---|---|
| On-the-Fly Control Commands | The AC4490 transceiver contains static memory that holds many of the parameters that control the transceiver operation. Using the "CC" command set allows many of these parameters to be changed during system operation. Because the memory these commands affect is static, when the transceiver is reset, these parameters will revert back to the settings stored in the EEPROM. While in CC Command mode using pin 17 (Command/Data), the RF interface of the transceiver is still active. Therefore, it can receive packets from remote transceivers while in CC Command mode and forward these to the OEM Host. |
| AT Command Mode | While in CC Command mode using AT Commands, the RF interface of the transceiver is active, but packets sent from other transceivers will not be received. The transceiver uses Interface Timeout/RF Packet Size to determine when a CC Command is complete. Therefore, there should be no delay between each character as it is sent from the OEM Host to the transceiver or the transceiver will not recognize the command.<br><br>The link between the OEM host and the transceiver does not need to resync when exiting Command Mode. Acknowledgements will be sent while in Command Mode, but the packet will be dumped on the receiving end. However, if an RF packet is received before the Interface Timeout expires on a CC Command, the transceiver will send the packet to the OEM Host before sending the CC Command response.<br><br>When an invalid command is sent, the radio scans the command to see if it has a valid command followed by bytes not associated with the command, in which case the radio discards the invalid bytes and accepts the command. In all other cases, the radio returns the first byte of the invalid command to the user and discards the rest. |

*Table 1: Command Quick Reference*

| Command Name | Command (All Bytes in Hex) | | | | | | Return (All Bytes in Hex) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Enter Command Mode | 0x41 | 0x54 | 0x2B | 0x2B | 0x2B | 0x0D | 0xCC | 0x43 | 0x4F | 0x4D |
| Exit Command Mode | 0xCC | 0x41 | 0x54 | 0x4F | 0x0D | - | 0xCC | 0x44 | 0x41 | 0x54 |
| Status Request | 0xCC | 0x00 | 0x00 | - | - | - | 0xCC | Firmware Version | 0x00: Server<br>0x01: Client in range<br>0x03: Out of range | |
| Change Channel | 0xCC | 0x02 | New Channel | | - | - | 0xCC | New Channel | | - |
| Change Server/Client | 0xCC | 0x03 | 0x00: Server<br>0x03: Client | | - | - | 0xCC | Firm-ware Version | 0x00: Server<br>0x03: Client | |
| Change Sync Channel | 0xCC | 0x05 | New Sync Channel | | - | - | 0xCC | New Sync Channel | - | - |

| Command Name | Command (All Bytes in Hex) | | | | | | Return (All Bytes in Hex) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Sleep Walk Power Down | 0xCC | 0x06 | - | - | - | - | 0xCC | Channel | - | - |
| Sleep Walk Wake Up | 0xCC | 0x07 | - | - | - | - | 0xCC | Channel | - | - |
| Broadcast Packets | 0xCC | 0x08 | 0x00: Addressed<br>0x01: Broadcast | | | - | 0xCC | 0x00 or 0x01 | - | - |
| Write Destination Address | 0xCC | 0x10 | Byte 4 of Dest. MAC | | Byte 5 | Byte 6 | 0xCC | Byte 4 of Dest.MAC | Byte 5 | Byte 6 |
| Read Destination Address | 0xCC | 0x11 | - | - | - | - | 0xCC | Byte 4 of Dest. MAC | Byte 5 | Byte 6 |
| Forced Calibration | 0xCC | 0x12 | 0x00 | 0x00 | - | - | 0xCC | Firm-ware Version | 0x00: Server in Normal Operation<br>0x01: Client in Normal Operation<br>0x02: Server in Acquisition Sync<br>0x03: Client in Acquisition Sync | |
| Auto Destination | 0xCC | 0x15 | bit-0: Auto Destination<br>bit-1: Auto Channel<br>bit-4: Enable Auto Destination<br>bit-5: Enable Auto Channel | | | | 0xCC | bit-0: Auto Destination<br>bit1: Auto Channel<br>bits-2-7: 0 | | |
| Read Digital Inputs | 0xCC | 0x20 | - | - | - | - | 0xCC | bit-0: GI0<br>bit-1: GI1 | - | - |
| Read ADC | 0xCC | 0x21 | 0x01: AD In<br>0x02: Temp<br>0x03: RSSI | | - | - | 0xCC | MSB of 10 bit ADC | LSB of 10 bit ADC | |
| Report Last Valid RSSI | 0xCC | 0x22 | - | - | - | - | 0xCC | RSSI | - | - |
| Write Digital Outputs | 0xCC | 0x23 | bit-0: GO0<br>bit-1: GO1 | | - | - | 0xCC | bit-0: GO0<br>bit-1: GO1 | - | - |
| Write DAC | 0xCC | 0x24 | Update Period | Duty Cycle | - | - | 0xCC | Update Period | Duty Cycle | - |
| Set Max Power | 0xCC | 0x25 | New Max Power | | | - | 0xCC | Max Power | - | - |
| Report Last Packet RSSI | 0xCC | 0x26 | - | - | - | - | 0xCC | RSSI | - | - |
| Long Range Mode[1] | 0xCC | 0x27 | 0x00: Normal Mode (Disabled)<br>0x01: Long Range Mode (Enabled) | | | | 0xCC | 0x00: Normal Mode (Disabled)<br>0x01: Long Range Mode (Enabled) | | |
| Transmit Buffer Empty | 0xCC | 0x30 | - | - | - | - | 0xCC | 0x00 | - | - |
| Disable Sync to Channel | 0xCC | 0x85 | - | - | - | - | 0xCC | Channel | - | - |
| Deep Sleep Mode | 0xCC | 0x86 | - | - | - | - | 0xCC | Channel | - | - |
| Enter Probe | 0xCC | 0x8E | 0x00: Enter Probe<br>0x01: Exit Probe | | | - | 0xCC | 0x00 or 0x01 | - | - |
| Read Temperature | 0xCC | 0xA4 | - | - | - | - | 0xCC | Temp (C) | - | - |
| Read Temperature at last calibration | 0xCC | 0xA5 | - | - | - | - | 0xCC | Temp (C) | | |
| EEPROM Byte Read | 0xCC | 0xC0 | Start Address | | Length | | 0xCC | Start Address | Length | Data |
| EEPROM Byte Write | 0xCC | 0xC1 | Start Address | | Length | Data | Starting Address | | Length | Data written |
| Soft Reset | 0xCC | 0xFF | - | - | - | - | - | - | - | - |

1. Available only on AC4490LR-1000 transceivers.

# Command Descriptions

## Enter AT Command Mode

Prior to sending this command, the OEM Host must ensure that the transceiver's RF transmit buffer is empty. If the buffer is not empty, the radio will interpret the command as data and it will be sent over the RF. This can be accomplished by waiting up to one second between the last packet and the AT command.

**Note:** RF Packet Size must be set to a minimum of 6 to use this command.

| |
|---|
| Command: <0x41> <0x54> <0x2B> <0x2B> <0x2B> <0x0D> |
| Number of Bytes Returned: 4 |
| Response: <0xCC> <0x43> <0x4F> <0x4D> |

## Exit AT Command Mode

The OEM Host should send this command to exit AT Command mode and resume normal operation.

| |
|---|
| Command: <0xCC> <0x41> <0x54> <0x4F> <0x0D> |
| Number of Bytes Returned: 4 |
| Response: <0xCC> <0x44> <0x41> <0x54> |

## Status Request

The OEM Host issues this command to request the status of the transceiver.

| |
|---|
| Command: <0xCC> <0x00> <0x00> |
| Number of Bytes Returned: 3 |
| Response: <0xCC> <Version> <Status> |
| Parameter Range:<br><Version> = Firmware version of radio<br><Status> = 0x00: Server<br>0x01: Client in Range<br>0x03: Client out of Range |

## Change Channel

The OEM Host issues this command to change the channel of the transceiver.

| |
|---|
| Command: 0xCC 0x02 <Channel> |
| Number of Bytes Returned: 2 |
| Response: 0xCC <Channel> |
| Parameter Range: <Channel> = RF Channel in use |

## Change Server/Client

The OEM Host issues this command to change the mode of the transceiver from server to client and vice versa.

| | |
|---|---|
| Command: <0xCC> <0x03> <Mode> | |
| Number of Bytes Returned: 3 | |
| Response: <0xCC> <Version> <Mode> | |
| Parameter Range: | |
| <Mode> = | 0x00: Server<br>0x03: Client |
| <Version> = | Firmware version of radio |

## Change Sync Channel

The OEM Host issues this command to change the sync channel byte and enable sync to channel. See Appendix I: Sync-to-Channel for more information.

**Note:** Valid only for server transceivers.

| |
|---|
| Command: <0xCC> <0x05> <Channel> |
| Number of Bytes Returned: 3 |
| Response: <0xCC> <Channel> |
| Parameter Range: <Channel> = Sync Channel |

## Sleep Walk Power-Down

After the host issues this command, the client transceiver will issue its In_Range line logic high after entering power down. A client in Power Down will remain in sync with a server for a minimum of 2 minutes. To maintain synchronization with the server, the client should re-sync at least once every 2 minutes. This is done by sending the Power Down Wake Up command and waiting for the In_Range line to issue logic low. Once this occurs, the client is in sync with the server and can be put back into power-down mode.

**Note:** This command is valid only for client transceivers.

| |
|---|
| Command: <0xCC> <0x06> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Channel> |
| Parameter Range: <Channel> = RF Channel currently being used |

## Sleep Walk Power-Down Wake Up

The OEM Host issues this command to bring the client transceiver out of Power Down mode.

**Note:** This command is valid only for client transceivers.

| |
|---|
| Command: <0xCC> <0x07> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Channel> |
| Parameter Range: <Channel> = RF Channel currently being used |

## Broadcast Packets

The OEM Host issues this command to change the transceiver operation between Addressed Packets and Broadcast Packets. If Addressed Packets are selected, the transceiver will send all packets to the transceiver designated by the Destination Address programmed in the transceiver. If Broadcast Packets are selected, the transceiver will send its packets to all transceivers on that network.

| | |
|---|---|
| Command: <0xCC> <0x08> <Mode> | |
| Number of Bytes Returned: 2 | |
| Response: <0xCC> <Mode> | |
| Parameter Range: <Mode> = | 0x00: Addressed<br>0x01: Broadcast |

## Write Destination Address

The OEM Host issues this command to the transceiver to change the Destination Address.

**Note:** Only the three Least Significant Bytes of the MAC Address are used for packet delivery.

| |
|---|
| Command: <0xCC> <0x10> <MAC3> <MAC2> <MAC1> |
| Number of Bytes Returned: 4 |
| Response: <0xCC> <MAC3> <MAC2> <MAC1> |
| Parameter Range: <MAC> = 0x00 - 0xFF corresponding to 3 LSB's of destination MAC Address |

## Read Destination Address

The OEM Host issues this command to the transceiver to read the destination address.

**Note:** Only the three Least Significant Bytes of the MAC Address are used for packet delivery.

| |
|---|
| Command: <0xCC> <0x11> |
| Number of Bytes Returned: 4 |
| Response: <0xCC> <MAC3> <MAC2> <MAC1> |
| Parameter Range: <MAC> = 0x00 - 0xFF corresponding to 3 LSB's of destination MAC Address |

## Auto Calibration (Forced Recalibration)

When enabled, Auto Calibrate causes the radio to measure the temperature every 30 to 60 seconds. If the temperature changes more than 30°C from the last calibration, the radio will initiate a recalibration.

During the recalibration, the radio will not assert CTS high. Recalibration can take up to 3 seconds and the command response will not be sent to the OEM Host until recalibration is complete.

**Note:** If Auto Calibration is *disabled*, the CL4490 radio may fail to lock onto frequency. If it does, the radio timeouts after 5 ms and performs a recalibration.

| | | |
|---|---|---|
| Command: <0xCC> <0x12> <0x00> <0x00> | | |
| Number of Bytes Returned: 3 | | |
| Response: <0xCC> <Version> <Status> | | |
| Parameter Range: | <Version> = | Firmware version of radio |
| | <Status> = | 0x00: Server in range |
| | | 0x01: Client in range |
| | | 0x02: Server out of range |
| | | 0x03: Client out of range |

## Auto Destination/Auto Channel

The Host issues this command to change the Auto Destination & Auto Channel settings. When issuing this command, the Auto Destination/Auto Channel settings is only changed if the corresponding enable bit is set.

| | |
|---|---|
| Command: <0xCC> <0x15> <Auto Dest> | |
| Number of Bytes Returned: 2 | |
| Response: <0xCC> <Auto Dest> | |
| Parameter Range: | |
| <Auto Dest>= | bit 7: Ignored |
| | bit 6: Ignored |
| | bit 5: Enable Auto Chan. Modification |
| | bit 4: Enable Auto Dest. Modification |
| | bit 3: Ignored |
| | bit 2: Ignored |
| | bit 1: Auto Channel |
| | bit 0: Auto Destination |

## Read Digital Inputs

The OEM Host issues this command to read the state of both digital input lines.

| |
|---|
| Command: <0xCC> <0x20> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Digital In> |
| Parameter Range:<br><Digital In> = bit-0: GI0<br>bit-1: GI1 |

## Read ADC

The OEM Host issues this command to read any of the three onboard 10-bit A/D converters. Because the RF is still active in On-the-Fly Command Mode, the transceiver will not process the command until there is no activity on the network. The Read RSSI command is therefore useful for detecting interfering sources but will not report the RSSI from a remote transceiver on the network. The equations for converting these 10 bits into analog values are as follows:

**Analog Voltage** = (10 bits / 0x3FF) * 3.3 V

**Temperature (°C)** = ((Analog Voltage - 0.3) / 0.01) - 30

**Instantaneous RSSI value (dBm)** = -105 + (0.22 * (0x3FF - 10 bits))

| |
|---|
| Command: <0xCC> <0x21> <Port> |
| Number of Bytes Returned: 3 |
| Response: <0xCC> <Hi ADC> <Lo ADC> |
| Parameter Range:<br><Port> = 0x00: AD In<br>0x01: Temperature<br>0x02: Instantaneous RSSI<br><Hi ADC> = MSB of requested 10-bit ADC value<br><Lo ADC> = LSB of requested 10-bit ADC value |

## Report Last Valid RSSI

Since RSSI values are only valid when the local transceiver is receiving an RF packet from a remote transceiver, instantaneous RSSI can be tricky to use. Therefore, the transceiver stores the most recent valid RSSI value as measured the last time the transceiver received a packet or beacon. The Host issues this command to retrieve that value.

**Note:** This value will default to 0xFF on a client and 0x00 on a server if no valid RSSI measurement has been made since power-up.

| |
|---|
| Command: <0xCC> <0x22> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Last Valid RSSI> |

*Table 2: Received Signal Strength*

| RSSI (dBm) | Hex Value | RSSI (dBm) | Hex Value |
|---|---|---|---|
| -92 | C0 | -72 | 63 |
| -91 | BC | -71 | 5F |
| -90 | BB | -70 | 5B |
| -89 | B9 | -69 | 58 |
| -88 | B8 | -68 | 54 |
| -87 | AE | -67 | 4F |
| -86 | A9 | -66 | 4B |
| -85 | A2 | -65 | 47 |
| -84 | 92 | -64 | 43 |
| -83 | 8D | -63 | 3D |
| -82 | 86 | -62 | 2A |
| -81 | 82 | -60 | 25 |
| -80 | 7D | -58 | 1A |
| -79 | 79 | -56 | 16 |
| -78 | 75 | -54 | 13 |
| -77 | 72 | -52 | 11 |
| -76 | 6F | -50 | 0E |
| -75 | 6B | -48 | 0D |
| -74 | 68 | -46 | 0C |
| -73 | 66 | -44 | 0B |

**Note:** The receiver is saturated after -45 dBm and cannot accurately measure the RSSI above -45 dBm.

## Write Digital Outputs

The OEM Host issues this command to write both digital output lines to particular states.

**Note:** This command should only be used when Protocol Status (0xC2) is not set to 0xE3.

| |
|---|
| Command: <0xCC> <0x23> <Digital Out> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Digital Out> |
| Parameter Range:<br><Digital Out>= bit-0: GO0<br>bit-1: GO1 |

## Write DAC

The OEM Host issues this command to write DA_Out to a particular voltage. The transceiver uses a PWM (Pulse Width Modulator) to generate the analog voltage. The theory behind a PWM is that a binary pulse is generated with a fixed rate (<Data 1>) and duty cycle (<Data 2>). As such, this pin toggles between High & Low. This signal is filtered via an on-board R-C circuit and an analog voltage is generated.

Duty cycle specifies the ratio of time in one cycle that the pulse spends High proportionate to the amount of time it spends Low. So, with a duty cycle of 50% (0x80), the pulse is High 50% of the time and Low 50% of the time; therefore the analog voltage would be half of 3.3V or 1.65V. A broad filter has been implemented on the transceiver and there is no advantage to using a slower update period. Generally, a faster update period is preferred.

| |
|---|
| Command: <0xCC> <0x24> <Data1> <Data2> |
| Number of Bytes Returned: 3 |
| Response: <0xCC> <Data1> <Data2> |
| Parameter Range: <Data1> = Period of PWM, Hex value<br><Data2> = Duty Cycle of PWM, Hex value |
| $Data1 = \frac{(T_{period} * 14.7456^{6})}{255} - 1$ ; Where $T_{period}$ == Period of the PWM in seconds<br>Data1 value is then converted from decimal to hex |
| $Data2 = DC_{percentage} * 255$ ; Where $DC_{percentage}$ == duty cycle of the PWM in decimal percent (50% = 0.5) |

**Note:** The duty cycle is represented at this pin as an analog voltage. 50% duty cycle is half of 3.3V or 1.65V.

## Set Max Power

The OEM Host issues this command to limit the maximum transmit power emitted by the transceiver. This can be useful to minimize current consumption and satisfy certain regulatory requirements.

**Note:** The radios are shipped at maximum allowable power.

| |
|---|
| Command: <0xCC> <0x25> <Max Power> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Max Power> |
| Parameter Range: <Max Power>= New Max Power setting |

## Output Power

*Table 3: AC4490-200*

| EEPROM Value (Hex) | Current (mA) | dBm | mW |
|---|---|---|---|
| 0 | 61 | -22 | 0.01 |
| 1 | 64 | -9 | 0.13 |
| 2 | 65 | -3 | 0.5 |
| 3 | 66 | 1 | 1.26 |
| 4 | 67.5 | 4 | 2.51 |
| 5 | 70 | 7 | 5.01 |
| 6 | 73 | 9 | 7.94 |
| 7 | 77 | 10.5 | 11.22 |
| 8 | 83 | 12 | 15.85 |
| 9 | 88 | 13.5 | 22.39 |
| A | 93.5 | 14.5 | 28.18 |
| B | 99 | 15.5 | 35.48 |
| C | 105 | 16.5 | 44.67 |
| D | 110.5 | 17 | 50.12 |
| E | 114.5 | 17.5 | 56.23 |
| F | 117.5 | 18.5 | 70.79 |
| 1E | 126 | 19.5 | 89.13 |

| EEPROM Value (Hex) | Current (mA) | dBm | mW |
|---|---|---|---|
| 60 | 127.5 | 20 | 100 |

*Table 4: AC4490-1000*

| EEPROM Value (Hex) | Current (mA) | dBm | mW |
|---|---|---|---|
| 0 | 430 | -1 | 0.79 |
| 1 | 430 | 9 | 7.94 |
| 2 | 470 | 14 | 25.12 |
| 3 | 520 | 16.5 | 44.67 |
| 4 | 560 | 19 | 79.43 |
| 5 | 650 | 20.5 | 112.2 |
| 6 | 740 | 22 | 158.49 |
| 7 | 780 | 23 | 199.53 |
| 8 | 870 | 24.5 | 281.84 |
| 9 | 950 | 25 | 316.23 |
| A | 1000 | 26 | 398.11 |
| B | 1080 | 26.5 | 446.68 |
| C | 1130 | 27 | 501.19 |
| D | 1170 | 27.5 | 562.34 |
| E | 1260 | 28 | 630.96 |
| F | 1300 | 28 | 743 |

## Long Range Mode

The OEM Host issues this command to temporarily enable or disable Long Range Mode in the transceiver.

**Note:** Only available on AC4490LR-1000 transceivers with firmware v6.7+.

| |
|---|
| Command: <0xCC> <0x27> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Mode> |
| Parameter Range:<br><Mode> = 0x00: Disable Long Range Mode<br>0x01: Enable Long Range Mode |

## Transmit Buffer Empty

The OEM Host issues this command to determine when the RF transmit buffer is empty. The Host will not receive the transceiver response until that time.

| |
|---|
| Command: <0xCC> <0x30> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <0x00> |

## Disable Sync-to-Channel

The OEM Host issues this command to disable Sync-to-Channel mode. See Appendix I: API for more information.

**Note:** This command is valid only for servers.

| |
|---|
| Command: <0xCC> <0x85> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Channel> |
| Parameter Range: <Channel> = RF Channel currently being used |

## Deep Sleep Mode

The OEM Host issues this command to put the transceiver into Deep Sleep mode. Once in Deep Sleep mode, the transceiver disables all RF communications and will not respond to any further commands until being reset or power-cycled.

**Note:** This command is valid for both servers and clients.

| |
|---|
| Command: <0xCC> <0x86> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Channel> |
| Parameter Range: <Channel> = RF Channel currently being used |

## Read Temperature

The OEM Host issues this command to read the onboard temperature sensor. The transceiver reports the temperature in °C where 0x00 - 0x50 corresponds to 0 - 80 °C and 0xD8 - 0x00 corresponds to - 40 - 0 °C.

| |
|---|
| Command: <0xCC> <0xA4> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Temp> |
| Parameter Range: <Temp> = Temperature of module |

## Read Temperature at Last Calibration

The OEM Host issues this command to read the temperature of the radio at the time of its last calibration. The transceiver reports the temperature in °C where 0x00 - 0x80 corresponds to 0 - 80° C and where 0xD8 - 0x00 is the two's complement representation corresponding to -40 - 0 °C.

**Note:** 0xD8 is a twos complement representation of -40 – 0.

| |
|---|
| Command: <0xCC> <0xA5> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Temp> |
| Parameter Range: <Temp> = Temperature at last calibration |

## Probe

When the OEM Host issues this command, the transceiver sends out a query every 500 ms. The transceivers randomly choose a query to respond to. After responding to a probe, the transceiver will wait at least 10 seconds before responding to another probe.

**Note:** This command can only be sent from a server radio.

| |
|---|
| Command: <0xCC> <0x8E> <Probe> |
| Number of Bytes Returned: 2 |
| Response: <0xCC> <Probe> |
| Parameter Range:<br><Probe> = 0x00: Disable Probe Mode<br>0x01: Enable Probe Mode |

### Transceiver's Response

Upon hearing the remote transceiver's probe acknowledge, the transceiver sends a response to the OEM Host.

| |
|---|
| Command: N/A |
| Number of Bytes Returned: 5 |
| Response: <0xCC> <Data> <MAC3> <MAC2> <MAC1> |
| Parameter Range:<br><Data> = bit-7: 0 Client<br>bit-7: 1 Server<br>bits 6-0: RF Channel |

## EEPROM Byte Read

Upon receiving this command, a transceiver will respond with the desired data from the addresses requested by the OEM Host. See EEPROM Parameters.

| |
|---|
| Command: <0xCC> <0xC0> <Start> <Length> |
| Number of Bytes Returned: 4+ |
| Response: <0xCC> <Start> <Length> <Data> |
| Parameter Range:<br><Start> = Address to begin reading from<br><Length> = Length of bytes to read<br><Data> = Requested data |

## EEPROM Byte Write

Upon receiving this command, a transceiver will write the data byte to the specified address but will not echo it back to the OEM Host until the EEPROM write cycle is complete (up to 10 ms).

Multiple byte writes of up to 128 bytes are allowed. An EEPROM boundary exists between addresses 0x7F and 0x80. No single EEPROM write command shall write to addresses on both sides of that EEPROM boundary. See EEPROM Parameters.

**Note:** Only the last byte written will be displayed in the command response.

| |
|---|
| Command: <0xCC> <0xC1> <Start> <Length> <Data> |
| Number of Bytes Returned: 4+ |
| Response: <Starting Address> <Length> <Data written> |
| Parameter Range:<br><Start> = Address to begin writing from<br><Length> = Length of bytes to write<br><Data> = Last byte of data written |

### Reset

The OEM Host issues this command to perform a soft reset of the transceiver. Any transceiver settings modified by CC commands will revert to the default values stored in the EEPROM.

| |
|---|
| Command: <0xCC> <0xFF> |
| Number of Bytes Returned: None |
| Response: None |

## API Control

API Control is a powerful feature offered by the AC4490. When enabled, the API Receive Packet, API Transmit Packet, API Send Data Complete and Enhanced API Receive Packet features provide dynamic packet routing and packet accounting ability to the OEM host, thereby eliminating the need for extensive programming on the OEM host side. API operation utilizes specific packet formats; specifying various vital parameters used to control radio settings and packet routing on a packet-by-packet basis. The API features can be used in any combination that suits the OEM's specific needs.

### Receive API Packet

**Note:** Implemented in firmware v.6.3 and later.

By default, the source MAC is not included in the received data string sent to the OEM host. For applications where multiple radios are sending data, it may be necessary to determine the origin of a specific data packet. Receive API Packet can be enabled to determine the sender of a message. This causes the receiving radio to add a header to the received packet detailing the length of the data packet and the sender's MAC address. The format of the Receive API Packet is:

| | | | |
|---|---|---|---|
| 0x83 | Payload Data Length | Sender's MAC | Payload Data |

**Note:** If Receive API is enabled, the Enhanced API Receive feature should be disabled by clearing bit-0 of the Enhanced API control byte, EEPROM address 0xC6.

## Enhanced API Receive Packet

**Note:** Implemented in firmware v.6.7 and later.

When Enhanced API Receive Packet is enabled, all packets received by the transceiver include the MAC address of the source radio as well as an RSSI indicator which can be used to determine the link quality between the two. API Receive Packet is enabled when bit-0 of the Enhanced API Control byte is enabled. Upon receiving a packet the radio sends its OEM Host the packet in the following format:

| | | | | | |
|---|---|---|---|---|---|
| 0x81 | Payload Data Length (0x01 – 0x50) | Laird Use | RSSI* | Source MAC (2, 1, 0) | Payload Data |

**Note:** When both API Send Data Complete and API Receive Packet are enabled, the Send Data Complete will be received before the transceiver sees the Receive API Packet. This may be reversed when the API Send Data Complete is missed and is being resent after the API Receive Packet is received.

**Note:** If Enhanced API Receive is enabled, the Receive API feature should be disabled by setting EEPROM byte 0xC1 to 0xFF.

## API Transmit Packet

**Note:** Implemented in firmware v6.7 and later.

API Transmit Packet is a powerful command that allows the OEM host to dynamically send data to a single or multiple (broadcast) transceiver(s) on a packet-by-packet basis. API Transmit Packet is enabled when bit-1 of the Enhanced API Control byte (EEPROM byte 0xC6) is enabled. The OEM host must use the following format to transmit a packet over the RF when using Transmit API packets:

| | | | | | |
|---|---|---|---|---|---|
| 0x81 | Payload Data Length (0x01 – 0z50) | Laird Use | Transmit Retries/ Broadcast Attempts | Destination MAC (2, 1, 0) | Payload Data |

- If the OEM Host does not encode the header correctly, the transceiver sends the entire string (up to 80 bytes) and looks for the header in the next data.
- Although the seven bytes of overhead are not sent over the RF, they are kept in the buffer until the packet is sent. Keep this in mind so as not to overrun the 256-byte buffer.
- Setting the Destination MAC to **0xFF 0xFF 0xFF** broadcasts the packet to all available transceivers.

**Note:** If the OEM host does not properly encode the header of the Tx API packet, the string (up to 80 bytes) is sent to the MAC address from the header of the last known good Tx API encoded packet.

## API Send Data Complete

**Note:** Implemented in v6.7 of the firmware and later.

API Send Data complete can be used as a software acknowledgement indicator. When a radio sends an addressed packet, it looks for a received acknowledgement (transparent to OEM host). If an acknowledgement is not received, the packet is retransmitted until one is received or all retries are used.

API Send Data Complete is enabled when bit-2 of the Enhanced API Control byte (EEPROM byte 0xC6) is enabled. The transceiver sends the OEM host the following data upon receiving an RF acknowledge or exhausting all attempts:

| | | | |
|---|---|---|---|
| 0x82 | Laird Use | RSSI* | 0x00: Failure<br>0x01: Success |

- **RSSI*** is how strong the local transceiver heard the remote transceiver.
- Successful RF Acknowledge updates the Success/Failure bit.
- A success is always displayed when sending broadcast packets (after all broadcast attempts have been exhausted).

# EEPROM PARAMETERS

The OEM Host can program various parameters that are stored in EEPROM which become active after a power-on reset. Table 5 gives the locations and descriptions of the parameters that the OEM Host can read or write. Factory default values are also shown.

**WARNING:** Do not write to any EEPROM addresses other than those listed below. Do not copy one transceiver's EEPROM to another transceiver as doing so may cause the transceiver to malfunction. Only the Configuration Utility should be used to copy one configuration into another transceiver.

*Table 5: EEPROM Parameters*

| Parameter | EEPROM Address | Length (Bytes) | Range | Default | Description |
|---|---|---|---|---|---|
| Product ID | 0x00 | 40 | | | Product identifier string. Includes revision information for software and hardware. |
| Page Refresh | 0x3D | 1 | 0x01 - 0xFF | 0x18 | Specifies the maximum amount of time a transceiver will report In-Range without having heard a server's beacon (equal to hop period * value).<br>**Note:** Do not set to 0x00. |
| Stop Bit Delay | 0x3F | 1 | 0x00 - 0xFF | 0xFF | For systems employing Parity, the serial stop bit might come too early. Stop Bit Delay controls the width of the last bit before the stop bit occurs.<br>0xFF = Disable Stop Bit Delay (12 µs)<br>0x00 = (256 * 1.6 µs) + 12 µs<br>0x01 - 0xFE = (value * 1.6 µs) + 12 µs |
| Channel Number | 0x40 | 1 | 0x00 - 0x37 | 1x1: 0x00<br>200: 0x00<br>1000: 0x10 | Set 0 = 0x00 - 0x0F (US/Canada): 1x1/200<br>Set 1 = 0x10 - 0x2F (US/Canada): 1x1/1000<br>Set 2 = 0x30 - 0x37 Australia: 1x1/200/1000 |
| Server/Client Mode | 0x41 | 1 | 0x01 - 0x02 | 0x02 | 0x01 = Server<br>0x02 = Client |
| Baud Rate Low | 0x42 | 1 | 0x00 - 0xFF | 0xFC | Low byte of the interface baud rate. Default baud rate is 57600. |

| Parameter | EEPROM Address | Length (Bytes) | Range | Default | Description |
|---|---|---|---|---|---|
| Baud Rate High | 0x43 | 1 | 0x00 | 0x00 | High byte of interface baud. Always 0x00. |
| Control 0 | **0x45** | 1 | 0x00 - 0xFF | 0x14 | Settings are:<br>bit-7: One Beacon Mode<br>0 = Disable One Beacon Mode<br>1 = Enable One Beacon Mode<br>bit-6: DES Enable<br>0 = Disable Encryption<br>1 = Enable Encryption<br>bit-5: Sync-to-Channel<br>0 = Disable Sync-to-Channel<br>1 = Enable Sync-to-Channel<br>bit-4: Laird Use Only<br>bit-3: Laird Use Only<br>bit-2: Laird Use Only<br>bit-1: RF Delivery<br>0 = Transmit using Addressed packets<br>1 = Transmit using Broadcast packets<br>bit-0: Laird Use Only |
| Frequency Offset | **0x46** | 1 | 0x00 - 0xFF | 0x01 | Protocol parameter used in conjunction with Channel Number to satisfy unique regulations. |
| $\overline{\text{CMD}}$ /Data RX Disable | **0x4B** | 1 | 0xE3, 0xFF | 0xFF | oxE3 = Enable $\overline{\text{CMD}}$ /Data RX Disable<br>0xFF = Disable $\overline{\text{CMD}}$ /Data RX Disable |
| Transmit Retries | **0x4C** | 1 | 0x01 - 0xFF | 0x10 | Maximum number of times a packet is transmitted when Addressed packets are selected.<br>**Note:** Do not set to 0. |
| Broadcast Attempts | **0x4D** | 1 | 0x01 - 0xFF | 0x04 | Number of times each packet is transmitted when Broadcast packets are selected.<br>**Note:** Do not set to 0. |
| API Control | **0x56** | 1 | 0x00 - 0xFF | 0x43 | Settings are:<br>bit-7: Laird Use Only<br>bit-6: Laird Use Only<br>bit-5: Unicast Only<br>0 = Disable Unicast Only<br>1 = Enable Unicast Only<br>bit-4: Auto Destination<br>0 = Use destination address<br>1 = Use auto destination<br>bit-3: Client Auto Channel<br>0 = Disable Auto Channel<br>1 = Enable Auto Channel<br>bit-2: RTS Enable |

| Parameter | EEPROM Address | Length (Bytes) | Range | Default | Description |
|---|---|---|---|---|---|
| | | | | | 0 = Ignore RTS<br>1 = Transceiver obeys RTS<br>bit-1: Duplex<br>0 = Half Duplex<br>1 = Full Duplex<br>bit-0: Auto Config<br>0 = Use EEPROM values<br>1 = Auto Configure values |
| Interface Timeout | 0x58 | 1 | 0x02 - 0xFF | 0x04 | Specifies a byte gap timeout, used in conjunction with RF Packet Size to determine when a packet coming over the interface is complete (0.5 ms per increment). |
| Sync Channel | 0x5A | 1 | 0x00 - 0xFF | 0x01 | Used to synchronize the hopping of collocated systems to minimize interference. |
| RF Packet Size | 0x5B | 1 | 0x01 - 0x80 | 0x80 | Used in conjunction with Interface Timeout; specifies the maximum size of an RF packet.<br>**Note:** Must be set to a minimum of 6 in order to send the Enter AT command. |
| $\overline{\text{CTS}}$ On | 0x5C | 1 | 0x01 - 0xFF | 0xD2 | $\overline{\text{CTS}}$ will be deasserted (High) when the transmit buffer contains at least this many characters. |
| $\overline{\text{CTS}}$ On Hysterisis | 0x5D | 1 | 0x00 - 0xFE | 0xAC | Once $\overline{\text{CTS}}$ has been deasserted, $\overline{\text{CTS}}$ will be reasserted (Low) when the transmit buffer is contains this many or less characters. |
| Max Power | 0x63 | 1 | 0x00 - 0x60 | Set in Production & can vary | Used to increase/decrease the output power.<br>**Note:** The transceivers are shipped at maximum allowable power. |
| Modem Mode | 0x6E | 1 | 0xE3, 0xFF | 0xFF | oxE3 = Enable Modem Mode<br>0xFF = Disable Modem Mode |
| Parity | 0x6F | 1 | 0xE3, 0xFF | 0xFF | 0xE3 = Enable Parity<br>0xFF = Disable Parity<br>Note: Enabling parity cuts throughput and the interface buffer size in half. |
| Destination ID | 0x70 | 6 | 0x00 - 0xFF | 0xFF | Specifies destination for RF packets |
| System ID | 0x76 | 1 | 0x00 - 0xFF | 0x01 | Similar to network password. Radios must have the same system ID to communicate with each other. |
| | | | | | |
| | | | | | |
| | | | | | |

| Parameter | EEPROM Address | Length (Bytes) | Range | Default | Description |
|---|---|---|---|---|---|
| MAC ID | 0x80 | 6 | 0x00 - 0xFF | | Factory programmed unique IEEE MAC address. |
| Original Max Power | 0x8E | 1 | | Set in Production & can vary | Copy of original max power EEPROM setting. This may be referenced but should not be modified. |
| Product ID | 0x90 | 15 | | | 0x90 - 0x93: Product ID<br>0x94 - 0x95: Prefix (CL, CN, or AC)<br>0x96 - 0x99: Power (200M, 200A, 1000, 1x1)<br>**Note:** There will be a period in front of the 1x1 to keep the field at four bytes.<br>0x9A - 0x9C: Interface (232, 485, TTL)<br>0x9D - 0x9E: Setup script (01 is stock)<br>0x9F: Reserved for future use; always 0xFF |
| Protocol Status / Receive ACK | 0xC0 | 1 | 0xE3, oxFF | oxFF | oxE3 = GO0 outputs the Protocol Status and GO1 outputs the Received Acknowledgement signal<br>0xFF = Disable Protocol Status / Receive ACK |
| Receive API | 0xC1 | 1 | 0xE3, 0xFF | 0xFF | 0xE3 = Enabled<br>0xFF = Disabled |
| Enhanced API Ctrl. | 0xC6 | 1 | | 0xF8 | Settings are:<br>bit-7: Enhanced API Control Enable<br>0 = Enable Enhanced API Control<br>1 = Disable Enhanced API Control<br>bit-6: Laird Use Only<br>bit-5: Laird Use Only<br>bit-4: Laird Use Only<br>bit-3: Laird Use Only<br>bit-2: Send Data Complete Enable<br>0 = Disable<br>1 = Enable<br>bit-1: API Transmit Packet Enable<br>0 = Disable<br>1 = Enable<br>bit-0: Enhanced API Receive Packet Enable<br>0 = Disable<br>1 = Enable |
| Auto Calibrate | 0xCC | 1 | 0xE3, 0xFF | 0xFF | oxE3 = Enable Auto Calibrate<br>0xFF = Disable Auto Calibrate |
| DES Key | 0xD0 | 7 | 0x00 - 0xFF | | 56-bit Data Encryption key |