

0

T R A N S C E I V E R

9

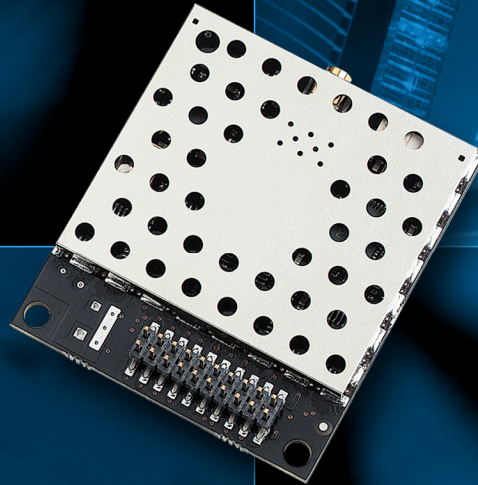
7

4

C

A

AEROCOMM'S MOTTO OF "INSTANT WIRELESS" EXPRESSES OUR COMMITMENT TO FACILITATING FAST AND EASY RF INTEGRATION. OUR COMPREHENSIVE DEVELOPMENT TOOLS AND ENGINEERING SUPPORT HELP OEMS ACCELERATE TIME-TO-MARKET.



U S E R  
M A N U A L

Version 1.4



**Copyright © 2006 AeroComm, Inc. All rights reserved.**

The information contained in this manual and the accompanying software programs are copyrighted and all rights are reserved by AeroComm, Inc. AeroComm, Inc. reserves the right to make periodic modifications of this product without obligation to notify any person or entity of such revision. Copying, duplicating, selling, or otherwise distributing any part of this product or accompanying documentation/software without the prior consent of an authorized representative of AeroComm, Inc. is strictly prohibited.

All brands and product names in this publication are registered trademarks or trademarks of their respective holders.

**This material is preliminary**

Information furnished by AeroComm in this specification is believed to be accurate. Devices sold by AeroComm are covered by the warranty and patent indemnification provisions appearing in its Terms of Sale only. AeroComm makes no warranty, express, statutory, and implied or by description, regarding the information set forth herein. AeroComm reserves the right to change specifications at any time and without notice.

AeroComm's 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**

For a period of one (1) year from the date of purchase by the OEM customer, AeroComm warrants the OEM transceiver against defects in materials and workmanship. AeroComm will not honor this warranty (and this warranty will be automatically void) if there has been any (1) tampering, signs of tampering; 2) repair or attempt to repair by anyone other than an AeroComm authorized technician.

This warranty does not cover and AeroComm will not be liable for, any damage or failure caused by misuse, abuse, acts of God, accidents, electrical irregularity, or other causes beyond AeroComm's control, or claim by other than the original purchaser.

In no event shall AeroComm be responsible or liable for any damages arising: From the use of product; From the loss of use, revenue or profit of the product; or As a result of any event, circumstance, action, or abuse beyond the control of AeroComm, whether such damages be direct, indirect, consequential, special or otherwise and whether such damages are incurred by the person to whom this warranty extends or third party.

If, after inspection, AeroComm determines that there is a defect, AeroComm will repair or replace the OEM transceiver at their discretion. If the product is replaced, it may be a new or refurbished product.

## DOCUMENT INFORMATION

<u>Revision</u>	<u>Description</u>
Version 1.0	2/21/05 - Initial Release Version
Version 1.1	3/4/05 - Updated Session Count Truth Table
Version 1.2	4/26/05 - Updated Transmit Mode Section
Version 1.3	3/17/06 - Corrected API Send Data Complete. Added Australian Channels. Added 1x1 documentation. Added Appendices I - IV.
Version 1.4	6/25/06 - Updated API Section. Added Serial Communications. Added Max Power backup EEPROM byte - address 0x8E. Added Product ID EEPROM bytes - addresses 0x90 - 0x9F. Updated Compliancy Information. Updated Appendices I - IV.

# AC4790 TRANSCEIVER MODULE

1

The compact AC4790 900MHz transceiver can replace miles of cable in harsh industrial environments. Using field-proven FHSS technology which needs no additional FCC licensing in the Americas, OEMs can easily make existing systems wireless with little or no RF expertise.

## AC4790 FEATURES

### NETWORKING AND SECURITY

- Masterless: True peer-to-peer, point-to-multipoint, point-to-point
- 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
- Software controlled sensitivity
- Hardware Protocol Status monitoring
- Two generic input and output digital lines and integrated ADC functions

### 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 80°C)
- Advanced configuration available using AT commands

## OVERVIEW

The AC4790 is a member of AeroComm's ConnexRF OEM transceiver family. The AC4790 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.

AC4790 transceivers operate in a Masterless architecture. The unique feature of this architecture is its dynamic Session extension and Collision Avoidance mechanism, which uses a proprietary scoring system to promote contention free communication and ensure that each node has fair access to the network. This instinctive dynamic peer-to-peer networking architecture enables several transceiver pairs to carry on simultaneous conversations on the same network.

## AC4790 TRANSCEIVER MODULE

### *Overview*

To boost data integrity and security, the AC4790 uses AeroComm's field-proven FHSS technology featuring optional Data-Encryption Standards (DES). Fully transparent, these transceivers operate seamlessly in serial cable replacement applications. Communications include both system and configuration data via an asynchronous TTL or optional RS-485 serial interface for OEM Host communications. Configuration data is stored in an on-board EEPROM and most parameters can be changed on the fly. All frequency hopping, synchronization, and RF system data transmission/reception is performed by the transceiver.

This document contains information about the hardware and software interface between an AeroComm AC4790 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.

Note: Unless mentioned specifically by name, the AC4790 modules will be 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 AC4790 module is connected) will be referred to as "OEM Host".

# SPECIFICATIONS

# 2

**Table 1: AC4790 Specifications**

General																									
20 Pin Interface Connector	Molex 87759-0030, mates with Samtec SMM-110-02-S-D																								
RF Connector	Johnson Components 135-3711-822																								
Antenna	AC4790-1x1: Customer must provide AC4790-200: MMCX Connector or integral antenna AC4790-1000: MMCX Connector																								
Serial Interface Data Rate	Baud rates from 1200 bps to 115,200 bps																								
Power Consumption (typical)	<table border="1"> <thead> <tr> <th rowspan="2"></th> <th colspan="4">Duty Cycle (TX=Transmit; RX=Receive)</th> </tr> <tr> <th>10%TX</th> <th>50%TX</th> <th>100%TX</th> <th>100%RX</th> </tr> </thead> <tbody> <tr> <td>AC4790-1x1:</td> <td>33mA</td> <td>54mA</td> <td>80mA</td> <td>28mA</td> </tr> <tr> <td>AC4790-200:</td> <td>38mA</td> <td>68mA</td> <td>106mA</td> <td>30mA</td> </tr> <tr> <td>AC4790-1000:</td> <td>130mA</td> <td>650mA</td> <td>1300mA</td> <td>30mA</td> </tr> </tbody> </table>		Duty Cycle (TX=Transmit; RX=Receive)				10%TX	50%TX	100%TX	100%RX	AC4790-1x1:	33mA	54mA	80mA	28mA	AC4790-200:	38mA	68mA	106mA	30mA	AC4790-1000:	130mA	650mA	1300mA	30mA
	Duty Cycle (TX=Transmit; RX=Receive)																								
	10%TX	50%TX	100%TX	100%RX																					
AC4790-1x1:	33mA	54mA	80mA	28mA																					
AC4790-200:	38mA	68mA	106mA	30mA																					
AC4790-1000:	130mA	650mA	1300mA	30mA																					
Channels	3 Channel Sets comprising 56 total channels																								
Security	One byte System ID. 56-bit DES encryption key.																								
Interface Buffer Size	Input/Output:256 bytes each																								
Transceiver																									
Frequency Band	902 – 928 MHz																								
RF Data Rate	76.8 kbps fixed																								
RF Technology	Frequency Hopping Spread Spectrum																								
Output Power	<table border="1"> <thead> <tr> <th></th> <th>Conducted (no antenna)</th> <th>EIRP (3dBi gain antenna)</th> </tr> </thead> <tbody> <tr> <td>AC4790-1x1:</td> <td>10mW typical</td> <td>20mW typical</td> </tr> <tr> <td>AC4790-200:</td> <td>100mW typical</td> <td>200mW typical</td> </tr> <tr> <td>AC4790-1000:</td> <td>743mW typical</td> <td>1486mW typical</td> </tr> </tbody> </table>		Conducted (no antenna)	EIRP (3dBi gain antenna)	AC4790-1x1:	10mW typical	20mW typical	AC4790-200:	100mW typical	200mW typical	AC4790-1000:	743mW typical	1486mW typical												
	Conducted (no antenna)	EIRP (3dBi gain antenna)																							
AC4790-1x1:	10mW typical	20mW typical																							
AC4790-200:	100mW typical	200mW typical																							
AC4790-1000:	743mW typical	1486mW typical																							
Supply Voltage	AC4790-1x1: 3.3V, $\pm 50$ mV ripple AC4790-200: 3.3 – 5.5V, $\pm 50$ mV ripple AC4790-1000*: Pin 10: 3.3 – 5.5V $\pm 50$ mV ripple Pin 11: 3.3 $\pm 3\%$ , $\pm 100$ mV ripple * Pins 10 and 11 may be tied together, provided the supply voltage never falls below 3.3 V and is capable of supplying 1.5 A of current.																								
Sensitivity	-100dBm typical @ 76.8kbps RF Data Rate																								
EEPROM write cycles	20000																								
Initial Transceiver Sync time/Hop period	25ms / 50 ms																								

**Table 1: AC4790 Specifications**

Transceiver (Cont'd)		
Range, Line of Site (based on 3dBi gain antenna)	AC4790-1x1: AC4790-200: AC4790-1000:	Up to 1 mile Up to 4 miles Up to 20 miles
Environmental		
Temperature (Operating)	-40°C to 80°C	
Temperature (Storage)	-50°C to +85°C	
Humidity (non-condensing)	10% to 90%	
Physical		
Dimensions	Transceiver with MMCX Connector: 1.65" x 1.9" x 0.20" Transceiver with Integral Antenna: 1.65" x 2.65" x 0.20" AC4790-1x1: 1.00" x 1.00" x 0.162"	
Certifications		
	<b>AC4790-200</b>	<b>AC4790-1000</b>
FCC Part 15.247	KQLAC4490-100	KQLAC4490
Industry Canada (IC)	2268C-AC4490	2268C-AC44901000

## PIN DEFINITIONS

The AC4790 has a simple interface that allows OEM Host communications with the transceiver. The table below shows the connector pin numbers and associated functions. The I/O direction is with respect to the transceiver. All outputs are 3.3VDC levels and inputs are 5VDC TTL (with the exception of AC4790-1x1 and AC4790-1000 transceivers which have 3.3V inputs). All inputs are weakly pulled High and may be left floating during normal operation (with the exceptions listed for the AC4790-1x1).

**Table 2: AC4790 Pin Definitions**

Module Pin	1x1 Pin	Type	Signal Name	Function
1	4	O	GOO	Session status if Protocol Status is enabled. Otherwise, generic output.
2	6	O	TXD	Transmitted data out of the transceiver

**SPECIFICATIONS**

*Pin Definitions*

**Table 2: AC4790 Pin Definitions**

Module Pin	1x1 Pin	Type	Signal Name	Function
		I/O	RS485 A (True) <sup>1</sup>	Non-inverted RS-485 representation of serial data
3	7	I	RXD	Data input to the transceiver
		I/O	RS485 B (Invert) <sup>1</sup>	Mirror image of RS-485 A
4	5 <sup>2</sup>		GI0	Generic Input pin
5	3	GND	GND	Signal Ground
6		O	Do Not Connect	Has internal connection, for <b>AeroComm use only</b> .
7	9	O	CTS	Clear to Send – Active Low when the transceiver is ready to accept data for transmission.
8	10 <sup>2</sup>	I	RTS	Request to Send – When enabled in EEPROM, the OEM Host can take this High when it is not ready to accept data from the transceiver. NOTE: Keeping RTS High for too long can cause data loss.
9	19	O	GO1	Received Acknowledge status pin if Protocol Status is enabled. Otherwise, generic output.
10	2	PWR	VCC1	AC4790-1x1: 3.3V, ±50mV ripple AC4790-200: 3.3 – 5.5V, ±50mV ripple ( <b>Pin 10 is internally connected to Pin 11</b> ) AC4790-1000: 3.3 – 5.5V, ±50mV ripple
11	11	PWR	VCC2	AC4790-1x1: 3.3V, ±50mV ripple AC4790-200: 3.3 – 5.5V, ±50mV ripple ( <b>Pin 11 is internally connected to Pin 10</b> ) AC4790-1000: 3.3V ±3%, ±100mV ripple
12	23	I	Test	Test Mode – When pulled logic Low and then applying power or resetting, the transceiver’s serial interface is forced to a 9600, 8-N-1 rate. To exit, the transceiver must be reset or power-cycled with Test Mode logic High.
13	12	O	RSSI	Received Signal Strength - An analog output giving an instantaneous indication of received signal strength. Only valid while in Receive Mode.
14	21 <sup>2</sup>	I	GI1	Generic Input pin
15	16	I	UP_RESET	RESET – Controlled by the AC4790 for power-on reset if left unconnected. After a stable power-on reset, a logic High pulse will reset the transceiver.
16	13	GND	GND	Signal Ground
17	17	I	CMD/Data	When logic Low, the transceiver interprets OEM Host data as command data. When logic High, the transceiver interprets OEM Host data as transmit data.



**Table 2: AC4790 Pin Definitions**

Module Pin	1x1 Pin	Type	Signal Name	Function
18	15 <sup>3</sup>	I	AD In	10 bit Analog Data Input
19	1,8,20 24-28	N/C	Do Not Connect	Has internal connection, for <b>AeroComm use only</b> .
20	18	O	Session Status	When logic Low, the transceiver is in Session
N/A	14	RF	RF Port	RF Interface
N/A	22	I	Reset	Active Low version of UP_RESET. If RESET is used, UP_RESET should be left floating and if UP_RESET is used, RESET should be left floating.

1. When ordered with a RS-485 interface (not available on the AC4790-1x1).
2. Must be tied to VCC or GND if not used. Should never be permitted to float.
3. If used, requires a shunt 0.1 $\mu$ F capacitor at pin 15 followed by a series 1k resistor.

## ELECTRICAL SPECIFICATIONS

**Table 3: Input Voltage Characteristics**

Signal Name	AC47901x1 / AC4790-1000M				AC4790-200X				Unit
	High Min.	High Max.	Low Min.	Low Max.	High Min.	High Max.	Low Min.	Low Max.	
RS485A/B	N/A	12	-7	N/A	N/A	12	-7	N/A	V
RXD	2.31	3.3	0	0.99	2	5.5	0	0.8	V
GI0	2.31	3.3	0	0.99	2	5.5	0	0.8	V
RTS	2.31	3.3	0	0.99	2	5.5	0	0.8	V
$\overline{\text{Test}}$	2.31	3.3	0	0.99	2	5.5	0	0.8	V
GI1	2.31	3.3	0	0.99	2	5.5	0	0.8	V
UP_RESET	0.8	3.3	0	0.6	0.8	5	0	0.6	V
Command/Data	2.31	3.3	0	0.99	2	5.5	0	0.8	V
AD In	N/A	3.3	0	N/A	N/A	3.3	0	N/A	V

**Table 4: Output Voltage Characteristics**

Signal Name	Module Pin	1x1 Pin	Type	High Min.	Low Max.	Unit
GO0	1	19	O	2.5 @ 8mA	0.4 @ 8mA	V
TXD	2	6	O	2.5 @ 2mA	0.4 @ 2mA	V
RS485A/B	2,3	N/A	I/O	3.3 @ 1/8 Unit Load	N/A	V
CTS	7	9	O	2.5 @ 2mA	0.4 @ 2mA	V
GO1	9	19	O	2.5 @ 2mA	0.4 @ 2mA	V
RSSI	13	12	O	See Figure 1	See Figure 1	V
Session Status	20	18	O	2.5 @ 2mA	0.4 @ 2mA	V
GO0	1	19	O	2.5 @ 8mA	0.4 @ 8mA	V

# THEORY OF OPERATION

---

## MASTERLESS ARCHITECTURE

---

The Masterless architecture is a true peer-to-peer architecture, where any module that has data to transmit will initiate a communication Session with a transceiver(s) within its range, transmit data and exit the Session. This architecture eliminates the need for a master which dictates data flow control, hence reducing additional system overhead and greatly improving efficiency.

---

## MODES OF OPERATION

---

The AC4790 has three different operating modes; Receive, Transmit, & Command Mode. If the transceiver is not communicating with another radio, it will be in Receive Mode actively listening for a sync pulse from another transceiver. If the radio determines that it is a broadcast or addressed sync pulse, it will respond by going into session with the radio. A transceiver will enter Transmit or Command mode when the OEM Host sends data over the serial interface. The state of the Command/Data pin (Pin 17) or the data contents determine which of the two modes will be entered.

### 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 byte and corresponding on-the-fly commands. 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 destination address. To increase the odds of successful delivery, Transmit retries are utilized. Transparent to the OEM Host, the sending radio will send the RF packet to the intended receiver. If the receiver receives the packet free of errors, it will return an RF acknowledge within the same 50 ms hop. If a receive acknowledgement is not received, the radio will use a transmit retry to resend the packet. The radio will continue sending the packet until either (1) an acknowledgement is received or (2) all transmit retries have been used. The received packet will only be 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 out to every eligible transceiver on the network. To increase the odds of successful delivery, Broadcast attempts are utilized. Transparent to the OEM Host, the sending radio will send 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 and without RF acknowledgements. If the packet is received on the first attempt, the receiver will ignore the remaining broadcast attempts. The received packet will only be sent to the OEM Host if and when it is received free of errors.

## THEORY OF OPERATION

### *Modes of Operation*

When a radio has data to transmit, it sends out a sync pulse to initiate a session with one or more radios. This 25 ms sync pulse is sent during the first half of each 50 ms hop and transparent to the OEM Host. Once a Session has been established, the radio transmits the data during the remaining 25 ms of the current hop. The radio will stay in Transmit mode until its Session Count expires. When sending addressed packets, Session Count is defined as Session Count Refresh (EEPROM address 0xC4) + number of transmit retries (EEPROM address 0x4C). When sending broadcast packets, Session Count is equal to Session Count Refresh (EEPROM address 0xC4) + number of broadcast attempts (EEPROM address 0x4D). Once the radio exits the Session it returns to the default Receive Mode.

### Receive Mode

If a transceiver detects a sync pulse while in Receive Mode, it will join the Session and begin receiving data. While in Receive Mode, subsequent data of up to 128 bytes can be received every hop (50 ms).

When a transceiver is in Session, its Session Count is decremented by one every hop. When the Session Count reaches zero, the transceiver exits the Session. In order to continue receiving data, the transceivers update their Session Count every time data or an RF acknowledge is received. The SLock0 and SLock1 settings control Session Count as shown below.

**Table 5: Session Count Truth Table**

Case	Slock0	Slock1	Transceiver Receiving an Addressed Packet	Transceiver Receiving a Broadcast Packet
1	0	0	Radio loads its Current Session Count with its Session Count Refresh	Radio loads its Current Session Count with its Session Count Refresh
2	0	1	Radio loads its Current Session Count with (its Transmit Retries + its Session Count Refresh)	Radio loads its Current Session Count with (its Broadcast Attempts + its Session Count Refresh)
3*	1	0	Radio loads its Current Session Count with the remote radio's Session Count	Radio loads its Current Session Count with the remote radio's Session Count
4	1	1	Radio loads its Current Session Count with the remote radio's Current Session Count	Radio loads its Current Session Count with the remote radio's Current Session Count

\* EEPROM Default

Note 1: For Broadcast/Addressed packets, the Session Count for Full Duplex is 2x the value of Session Count in Half Duplex.

Note 2: It is best to have all transceivers with the same Session Count Refresh (EEPROM Address 0xC4) value. Session Count Refresh must not be set to 0x00.

**Case 1:** In this case, a radio loads its Session Count with its Session Count Refresh. This is suitable for Half Duplex communication where immediate response is not received from the remote radio.

## THEORY OF OPERATION

### *Modes of Operation*

**Case 2:** In this case, a radio loads its Session Count with (its Session Count Refresh + its Transmit Retries). This case is suitable for applications where there are high levels of interference and it is likely that transmit retries will be necessary to maintain reliable communications.

When an addressed packet or a response to a broadcast packet is sent, the sending radio will listen for a successful acknowledgement. If an acknowledgement is not sent, the radio will resend the packet until either an acknowledgement is received or it has exhausted all available transmit retries. If two radios are on the last hop of the current session and a retry is required, it is possible that once the current session has ended the receiving radio could go into session with a different radio and miss the final packet of the previous session. Adding the radios Transmit retries to its Current Session Count will ensure that the radio does not exit the session when the remote radio is using a Transmit Retry.

**Case 3:** In this case a radio loads its Session Count with the remote radio's Session Count. This is suitable for full duplex applications as the Session is extended as long as there is communication.

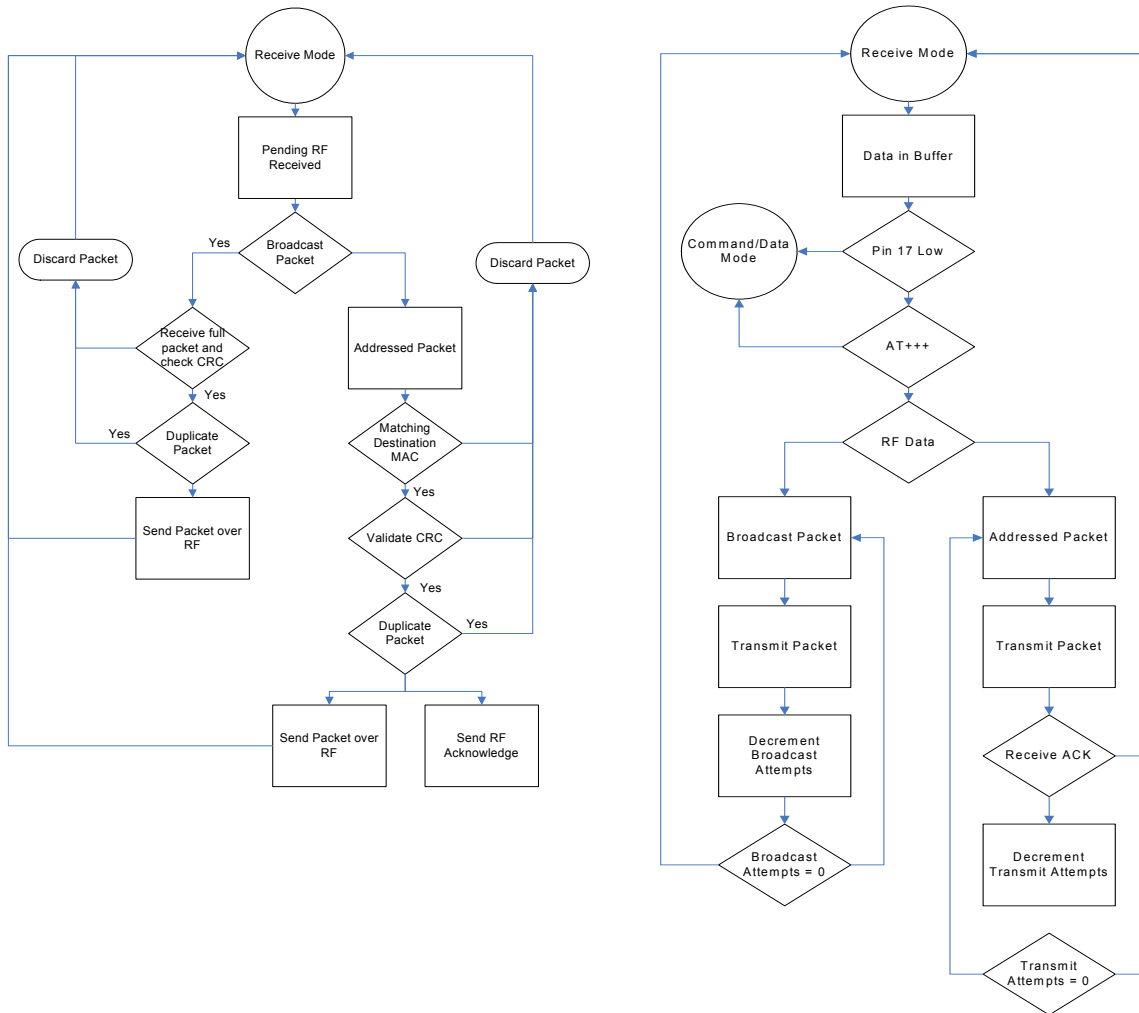
Note: This is the default case with which the radio ships and works well for almost all applications.

**Case 4:** In this case, a radio loads its Session Count with the remote radio's current Session Count. This is suitable for daisy chain applications and large networks in which radios cannot stay in session longer than needed.

### **Command Mode**

A radio will enter Command Mode when data is received over the serial interface from the OEM Host and either the Command/Data pin (pin 17) is logic Low or the received data contains the "AT+++" (Enter AT Command Mode) command. Once in Command Mode, all data received by the radio is interpreted as command data. Command Data can be either EEPROM Configuration or On-The-Fly commands.

Figure 1: Pending RF and Data in Buffer Flow



## API CONTROL

API Control is a powerful feature that the Masterless Protocol offers. When enabled, the API Transmit Packet, API Send Data Complete and 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. This ability of the protocol makes it ideal for any legacy system. 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.

### API Transmit Packet

API Transmit Packet is a powerful command that allows the OEM Host to send data to a single or multiple (broadcast) transceivers on a packet-by-packet basis. This can be useful for many applications; including polling and/or mesh networks. Refer to the API Appendix for further details.

API Transmit Packet is enabled when bit-1 of the API Control byte is enabled. The OEM Host should use the following format to transmit a packet over the RF.

0x81	Payload Data Length (0x01 - 0x80)	Session Count Refresh	Transmit Retries/Broadcast Attempts	Destination MAC (2,1,0)	Payload Data
------	--------------------------------------	-----------------------	-------------------------------------	-------------------------	--------------

- 1 If the OEM Host does not encode the header correctly, the transceiver will send the entire string (up to 0x80 bytes) and will look for the header in the next data.
- 2 Although the 7 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.
- 3 Setting the MAC to 0xFF 0xFF 0xFF will broadcast the packet to all available transceivers.

### API Send Data Complete

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

For applications where data loss is not an option, the OEM Host may wish to monitor the acknowledgement process using the API Send Data Complete. If an acknowledgement is not received (Failure), the OEM Host can send the packet to the transceiver once again.

API Send Data Complete is enabled when bit-2 of the API Control byte is enabled. The transceiver sends the OEM Host the following data upon receiving an RF acknowledge or exhausting all attempts.

0x82	RSSI	RSSI*	0x00: Failure 0x01: Success
------	------	-------	--------------------------------

- 1 The RSSI is how strong the remote transceiver heard the local transceiver; RSSI\* is how strong the local transceiver heard the remote transceiver.
- 2 Successful RF Acknowledge updates the Success/Failure bit.
- 3 A success will always be displayed when sending broadcast packets after all broadcast attempts have been exhausted.

### API Receive Packet

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. When API Receive Packet is enabled, all packets received by the transceiver will 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 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 - 0x80)	RSSI	RSSI*	Source MAC (2,1,0)	Payload Data
------	--------------------------------------	------	-------	-----------------------	--------------

**ENGINEER'S TIP**

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 order may get reversed when the API Send Data Complete is missed and is being resent after the API Receive Packet is received.



# SERIAL INTERFACE

In order for the OEM Host and a transceiver to communicate over the serial interface they need to have the same serial data rate. Refer to the following sections to ensure that the OEM Host data rate matches the serial interface baud rate.

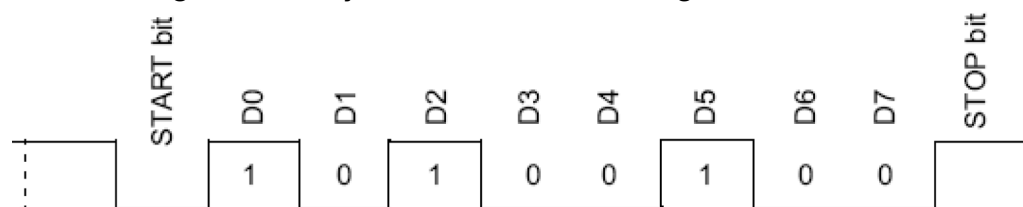
## SERIAL COMMUNICATIONS

The AC4790 is a TTL device which can be interfaced to a compatible UART (microcontroller) or level translator to allow connection to serial devices. UART stands for Universal Asynchronous Receiver Transmitter and its main function is to transmit or receive serial data.

### Asynchronous Operation

Since there is no separate clock in asynchronous operation, the receiver needs a method of synchronizing with the transmitter. This is achieved by having a fixed baud rate and by using START and STOP bits. A typical asynchronous mode signal is shown below.

**Figure 2: Asynchronous Mode Signal**



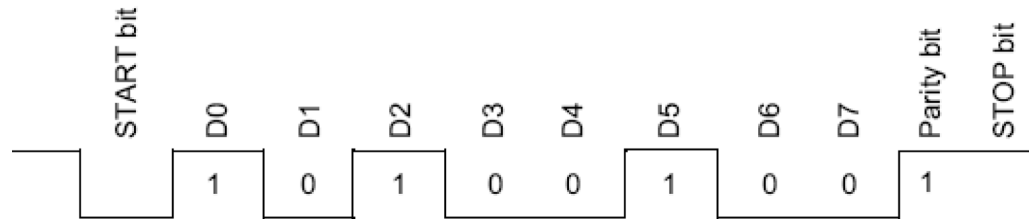
The UART outputs and inputs logic level signals on the TX and RX pins. The signal is high when no data is being transmitted and goes low when transmission begins.

The signal stays low for the duration of the START bit and is followed by the data bits; LSB first. The STOP bit follows the last data bit and is always high. After the STOP bit has completed, the START bit of the next transmission can occur.

### Parity

A parity bit is used to provide error checking for a single bit error. When a single bit is used, parity can be either even or odd. Even parity means that the number of ones in the data and parity sum to an even number and vice-versa. The ninth data bit can be used as a parity bit if the data format requires eight data bits and a parity bit as shown below.

**Figure 3: Even Parity Bit**



**Note:** Enabling parity cuts throughput and the interface buffer in half.

## OEM HOST DATA RATE

The OEM Host Data Rate is the rate with which the OEM Host and transceiver communicate over the serial interface. This rate is independent of the RF baud rate, which is fixed at 76.8 kbps. Possible values range from 1200 bps to 115,200 bps. Note: Enabling Parity cuts throughput in half and the Interface Buffer size in half. The following asynchronous serial data formats are supported:

**Table 6: Supported Serial Formats**

Data Bits	Parity	Stop Bits	Transceiver Programming Requirements
8	N	1	Parity Disabled
7	N	2	Parity Disabled
7	E, O, M, S	1	Parity Disabled
9	N	1	Parity Enabled
8	N	2	Parity Enabled
8	E, O, M, S	1	Parity Enabled
7	E, O, M, S	2	Parity Enabled

Mark (M) corresponds to 1 & Space (S) corresponds to 0

## SERIAL INTERFACE BAUD RATE

This two-byte value determines the baud rate used for communicating over the serial interface to a transceiver. The Table below lists values for some common baud rates. Baud rates below 1200 baud are not supported. For a baud rate to be valid, the calculated baud rate must be within  $\pm 3\%$  of the OEM Host baud rate. If the Test pin (Pin 12) is pulled logic Low at reset, the baud rate will be forced to 9,600. The RF baud rate is fixed at 76.8 Kbps and is independent of the interface baud rate. For Baud Rate values other than those shown below, the following equations can be used:

$$\text{BAUD} = \frac{14.7456 \times 10^6}{64 \times \text{DesiredBaud}}$$

BaudH = Always 0

BaudL = Low 8 bits of BAUD (base 16)

**Table 7: Baud Rate / Interface Timeout**

Baud Rate	BaudL (0x42)	BaudH (0x43)	Minium Interface Timeout (0x58)	Stop Bit Delay (0x3F)
115,200	0xFE	0x00	0x02	0xFF
57,600 <sup>1</sup>	0xFC	0x00	0x02	0x03
38,400	0xFA	0x00	0x02	0x08
28,800	0xF8	0x00	0x02	0x0E
19,200	0xF4	0x00	0x03	0x19
14,400	0xF0	0x00	0x04	0x23
9,600	0xE8	0x00	0x05	0x39
4800	0xD0	0x00	0x09	0x7A
2400	0xA0	0x00	0x11	0xFC
1200	0x40	0x00	0x21	0x00 <sup>2</sup>

1. 57,600 is the default baud rate

2. 0x00 will yield a stop bit of 421 uS. The stop bit at 1200 baud should actually be 833 uS.

## INTERFACE TIMEOUT / RF PACKET SIZE

**Interface Timeout** (EEPROM address 0x58), in conjunction with **RF Packet Size** (EEPROM address 0x5B), determines when a buffer of data will be sent out over the RF as a complete RF packet, based on whichever condition occurs first.

**Interface Timeout** – Interface Timeout specifies a maximum byte gap between consecutive bytes. When that byte gap is exceeded, the bytes in the transmit buffer are sent out over the RF as a complete packet. Interface Timeout is adjustable in 0.5ms increments and has a tolerance of ±0.5ms. Therefore, the Interface Timeout should be set to a minimum of 2. The default value for Interface Timeout is 0x04 (2ms) and should be adjusted accordingly when changing the transceiver baud rate.

**RF Packet Size** – When the number of bytes in the transceiver transmit buffer equals RF Packet Size, those bytes are sent out as a complete RF packet. It is much more efficient to send a few large packets rather than several short packets as every packet the transceiver sends over the RF contains extra header bytes which are not included in the RF Packet Size. RF packet size can

be set to a maximum of 0x80 and must be set to a minimum of 0x06 in order to send the Enter AT Command mode command.

## FLOW CONTROL

Flow control refers to the control of data flow between transceivers. It is the method used to handle data in the transmit/receive buffer and determines how data flow between the transceivers is started and stopped. Often, one transceiver is capable of sending data much faster than the other can receive and flow control allows the slower device to tell the faster device when to pause and resume data transmission.

When a transceiver has data to send, it sends a Ready To Send signal and waits for a Clear To Send response from the receiving unit. If the receiving radio is ready to accept data it will assert its  $\overline{\text{CTS}}$  low.  $\overline{\text{CTS}}$  will be reasserted when the buffer contains the number of bytes specified by CTS\_OFF (EEPROM address 0x5D). These signals are sent apart from the data itself on separate wires.

### ENGINEER'S TIP

#### Can I implement a design using just Txd, Rxd and Gnd (Three-wire Interface)?

Yes. However, it is strongly recommended that your hardware monitor the CTS pin of the radio. CTS is taken High by the radio when its interface buffer is getting full. Your hardware should stop sending at this point to avoid a buffer overrun (and subsequent loss of data).

You can perform a successful design without monitoring CTS. However, you need to take into account the amount of latency the radio adds to the system, any additional latency caused by Transmit Retries or Broadcast Attempts, how often you send data, non-delivery network timeouts and interface data rate. Polled type networks, where the Server host requests data from the Client host and the Client host responds, are good candidates for avoiding the use of CTS. This is because no one transceiver can monopolize the RF link. Asynchronous type networks, where any radio can send to another radio at any point in time, are much more difficult to implement without the use of CTS.

## HALF DUPLEX / FULL DUPLEX

When Half Duplex communication is chosen, the AC4790 will send a packet out over the RF whenever it can. This can cause packets sent by multiple transceivers at the same time to collide with each other over the RF. To prevent this, Full Duplex communication can be chosen. Full Duplex shares the bandwidth intelligently to enable two-way collision-free communication without any collision. This is done by calculating the amount of time until the next hop to ensure that it has time to send the packet; if there is enough time, it will send the packet and if not, it will wait until its next appropriate hop. The radio which initiates the session transmits during the even hops while the remaining radio(s) will transmit during the odd hops. Although the RF hardware is still technically half duplex, the bandwidth sharing it makes the transceiver seem full duplex. Enabling Full Duplex can cause overall throughputs to be cut in half.

## SYSTEM TIMING & LATENCY

Care should be taken when selecting transceiver architecture, as it can have serious effects on data rates, latency, and overall system throughput. The importance of these three characteristics will vary from system to system and should be a strong consideration when designing the system.

### ENGINEER'S TIP

#### In High-density applications, what amount of latency should be expected?

It is not easy to predict the exact amount of latency in high-density applications. There are many variables that affect system latency. The three variables that most affect the latency are the network load, the distance between transceivers, and whether the transceivers are operating in a broadcast or addressed mode. There is no fixed answer as to how much latency will be introduced in the system when considering high-density applications. In these cases we can just offer qualitative analysis of the latency in high-density applications. As the network load increases, then the number of collisions that will occur increases. As the number of collisions increase, then the system latency increases. As the distance between the transceivers increases, so to does the system latency. Finally, when transceivers operate in addressed mode they will retry sending a packet up to the number of time specified in the transmit retry parameter specified in the EEPROM. As the number of retries increases, the system latency will increase also.

## SYSTEM THROUGHPUT

When operating as shown below, an AC4790 transceiver is capable of achieving the listed throughput. However, in the presence of interference or at longer ranges, the transceiver may be unable to meet the specified throughput.

**Table 8: Maximum System Throughput**

Rf Status	Half Duplex Throughput (bps)	Full Duplex Throughput (bps) each way
Radio not in continuous session	25k	12.5k
Radio continuously in session	45k	22.5k

## RANDOM BACKOFF

**Random Back-Off** – The transceivers utilize a Carrier Sense Multiple Access (CSMA) protocol with random back-off and a selectable back-off seed. In the event of a collision, the transceiver will back off and retry the packet. Specifically, when two transceivers detect a collision, each transceiver will choose a random number of packet times that it will wait before retrying the

packet. This random number is selected from a pool of numbers defined by the back-off seed and consists of a number between 1 and 2, 1 and 4, 1 and 8, 1 and 16, 1 and 32, 1 and 64, 1 and 128 and 1 and 256. In a very dense network, where more than two transceivers could experience a collision, it is important to have a higher random back-off seed.

### ENGINEER'S TIP

#### What effects will Random Backoff have on system latency?

As the random backoff value increases, the overall system latency increases.

Worst case latency (Half Duplex) = 50 ms \* Number of retries \* Max. random value

Worst case latency (Full Duplex) = 100 ms \* Number of retries \* Max. random value

## NETWORKING

**System ID** - System ID (EEPROM address 0x76) is similar to a password character or network number and makes network eavesdropping more difficult. A transceiver will not establish a Session or communicate with a transceiver operating on a different System ID or Channel Number.

**RF Channel Number** - Channels 0x00 - 0x0F and 0x30 - 0x37 hop on 26 different frequencies. Channels 0x10 - 0x2F use 50 different frequencies.

**Table 9: RF Channel Number Settings**

Channel Set <sup>1</sup>	RF Channel Number Range (0x40)	Frequency Details & Regulatory requirements	Countries
0 (AC4790 - 1x1 AC4790 - 200)	0x00 - 0x0F	902 - 928 MHz (26 hop bins)	US / Canada
1 (AC4790 - 1x1 AC4790 - 1000)	0x10 - 0x2F	902 - 928 MHz (50 hop bins)	US / Canada
2 (AC4790 - 1x1 AC4790 - 200 AC4790 - 1000)	0x30 - 0x37	915 - 928 MHz (22 hop bins)	US / Canada (-1x1 / -200) Australia(-1x1/-200/-1000)

1. All Channels in a Channel Set use the same frequencies in a different order.

**DES (Data Encryption Standard)** - DES (Data Encryption Standard) – Encryption is the process of encoding an information bit stream to secure the data content. The DES algorithm is a common, simple and well-established encryption routine. An encryption key of 56 bits is used to encrypt the packet. The receiver must use the exact same key to decrypt the packet; otherwise garbled data will be produced.

To enable DES, EEPROM Byte 0x45, bit 6 must be set to a value of 1. To disable DES, set bit 6 to a value of 0. The 7 byte (56 bits) Encryption/Decryption Key is located in EEPROM Bytes 0xD0 – 0xD6. **It is highly recommended that this Key be changed from the default.**

## MAX POWER

---

Max Power provides a means for controlling the RF output power of the AC4790. Output power and current consumption can vary by as much as  $\pm 10\%$  per transceiver for a particular Max Power setting. Contact AeroComm for assistance in adjusting Max Power.

### ENGINEER'S TIP

The max power is set during Production and may vary slightly from one transceiver to another. The max power can be set as low as desired but should not be set higher than the original factory setting. A backup of the original power setting is stored in EEPROM address 0x8E.

# HARDWARE INTERFACE

# 5

Below is a description of all hardware pins used to control the AC4790.

## PIN DEFINITIONS

---

### Generic I/O

Both GIn pins serve as generic input pins. When Protocol Status (byte 0xC2 of EEPROM) is disabled, GO0 & GO1 serve as generic outputs. When Protocol Status is enabled, pins GO0 and GO1 alternatively serve as the Session Status and Receive Acknowledge Status pins, respectively. Reading and writing of these pins can be performed using CC Commands.

### HARDWARE PROTOCOL STATUS

When the GO0 pin is configured as the Session Status pin, GO0 is normally Low. GO0 will go High when a Session is initiated and remain High until the end of the Session. When the GO1 pin is configured as the Receive Acknowledge Status pin, GO1 is normally Low and GO1 will go High upon receiving a valid RF Acknowledgement and will remain High until the end (rising edge) of the next hop.

### TXD & RXD

#### SERIAL TTL

The AC4790-200 accepts 3.3 or 5VDC TTL level asynchronous serial data on the RXD pin and interprets that data as either Command Data or Transmit Data. Data is sent from the transceiver, at 3.3V levels, to the OEM Host via the TXD pin. **Note:** The AC4790-1000 & AC4790-1x1 transceivers ONLY accept 3.3V level signals.

#### RS-485

When equipped with an onboard RS-485 interface chip, TXD and RXD become the half duplex RS-485 pins. The transceiver interface will be in Receive Mode except when it has data to send to the OEM Host. TXD is the non-inverted representation of the data (RS485A) and RXD is a mirror image of TXD (RS485B). The transceiver will still use RTS (if enabled).

### CTS

The AC4790 has an interface buffer size of 256 bytes. If the buffer fills up and more bytes are sent to the transceiver before the buffer can be emptied, data loss will occur. The transceiver prevents this loss by asserting CTS High as the buffer fills up and taking CTS Low as the buffer is emptied. CTS On and CTS Off control the operation of CTS. CTS On specifies the amount of bytes that must be in the buffer for CTS to be disabled (logic High). Even while CTS is disabled, the OEM Host can still send data to the transceiver, but it should do so carefully.

**Note:** The CTS On/Off bytes of the EEPROM can be set to 1, in which case CTS will go high as data is sent in and low when buffer is empty.



## RTS

With RTS disabled, the transceiver will send any received data to the OEM Host as soon as it is received. However, some OEM Hosts are not able to accept data from the transceiver all of the time. With RTS enabled, the OEM Host can prevent the transceiver from sending it data by disabling RTS (logic High). Once RTS is enabled (logic Low), the transceiver can send packets to the OEM Host as they are received.

**Note:** Leaving RTS disabled for too long can cause data loss once the transceiver's 256 byte receive buffer fills up.

## Test / 9600 Baud

When pulled logic Low before applying power or resetting, the transceiver's serial interface is forced to a 9600, 8-N-1 (8 data bits, No parity, 1 stop bit). To exit, the transceiver must be reset or power-cycled with Test pin logic High. This pin is used to recover transceivers from unknown baud rates only. It should not be used in normal operation. Instead the transceiver Interface Baud Rate should be programmed to 9600 baud if that rate is desired for normal operation. The Test/9600 pin should be used for recovery purposes only as some functionality is disabled in this mode.

## RSSI

### INSTANTANEOUS RSSI

Received Signal Strength Indicator is used by the OEM Host as an indication of instantaneous signal strength at the receiver. The OEM Host must calibrate RSSI without an RF signal being presented to the receiver. Calibration is accomplished by following the steps listed below.

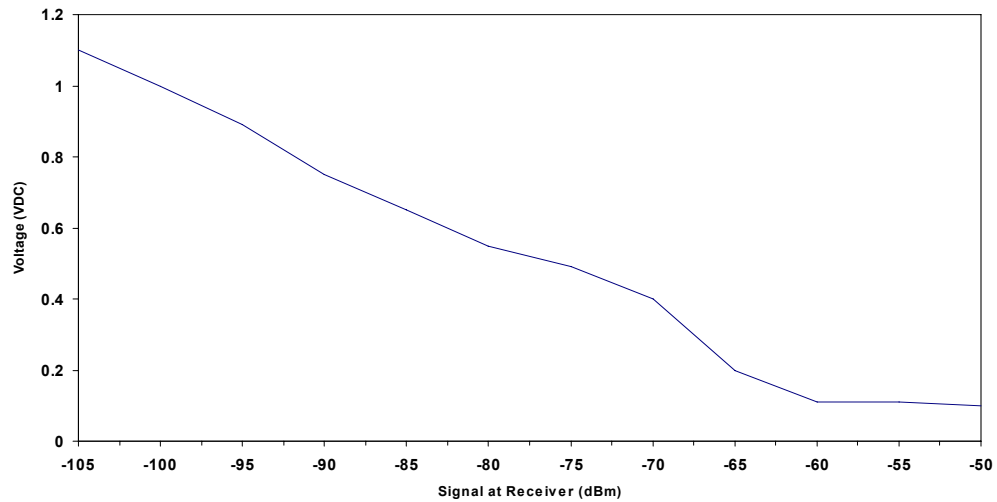
- 1) Power up only one transceiver in the coverage area.
- 2) Measure the RSSI signal to obtain the minimum value with no other signal present.
- 3) Power up another transceiver and begin sending data from that transceiver to the transceiver being measured. Make sure the two transceivers are separated by approximately ten feet.
- 4) Measure the peak RSSI, while the transceiver is in Session, to obtain a maximum value at full signal strength.

### VALIDATED RSSI

As RSSI is only valid when the local transceiver is receiving an RF packet from a remote transceiver, instantaneous RSSI can be very tricky to use. Therefore, the transceiver stores the most recent valid RSSI value. The OEM Host issues the Report Last Good RSSI command to request that value. Additionally, validated RSSI can be obtained from Receive Packet and Send Data Complete API commands and from the Probe command. Validated RSSI is not available at the RSSI pin. The following equation approximates the RSSI curve:

$$\text{Signal Strength (dBm)} = (-46.9 \times \text{VRSSI}) - 53.9$$

**Figure 4: RSSI Voltage vs. Received Signal Strength**



### **UP\_Reset**

UP\_Reset provides a direct connection to the reset pin on the AC4790 microprocessor and is used to force a soft reset. For a valid reset, reset must be asserted High for a minimum of 10ms.

### **Command/Data**

When logic High, the transceiver interprets incoming OEM Host data as transmit data to be sent to other transceivers and their OEM Hosts. When logic Low, the transceiver interprets OEM Host data as command data.

### **AD In**

AD In can be used as a cost savings to replace Analog-to-Digital converter hardware. Reading of this pin can be performed locally using the Read ADC command found in the On-the-Fly Control Command Reference.

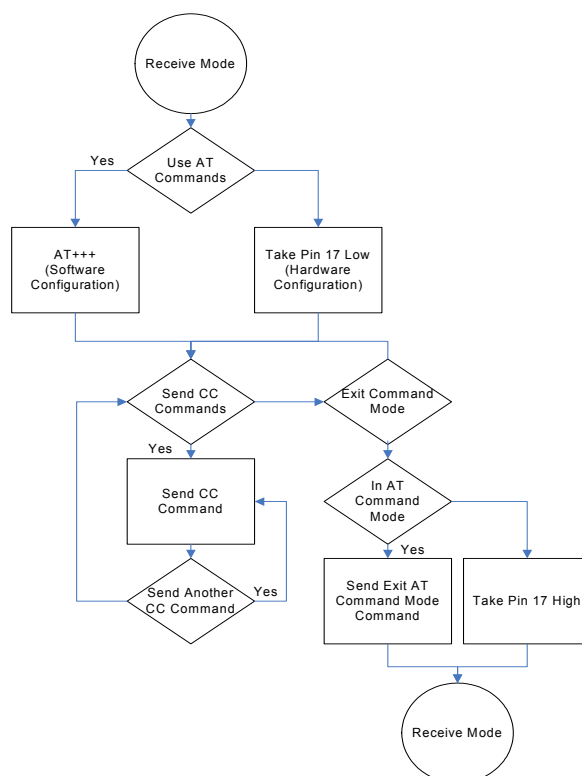
### **Session Status**

Reports logic Low during a Session and logic High when not in Session. The inverse of this pin can be obtained from pin GO0 when Protocol Status is enabled.

# CONFIGURING THE AC4790

The AC4790 can be configured using the CC Configuration Commands. The CC Commands can be issued using either Hardware or Software Configuration. To use Hardware Configuration, pin 17 of a transceiver must be asserted Low. Software Configuration can be used by entering AT Command Mode before issuing the CC Commands.

**Figure 5: AC4790 Configuration Flow**



## AT COMMANDS

---

The AT Command mode implemented in the AC4790 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.

### On-the-Fly Control Commands

The AC4790 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.

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. If the OEM Host has sent a CC Command to the transceiver and an RF packet is received by the transceiver, the transceiver will send the CC Command response to the OEM Host before sending the packet. 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.

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 back to the user and discards the rest.

**Table 10: Command Quick Reference**

Command Name	Command (All Bytes in Hex)						Return (All Bytes in Hex)			
AT Enter CommandMode	0x41	0x54	0x2B	0x2B	0x2B	0x0D	0xCC	0x43	0x4F	0x4D
Exit AT Command-Mode	0xCC	0x41	0x54	0x4F	0x0D	-	0xCC	0x44	0x41	0x54
Status Request	0xCC	0x00	0x00	-	-	-	0xCC	Firmware Version	0x00 - 0x03	-
Change Channel	0xCC	0x01	New Channel		-	-	0xCC	New Channel		-
Broadcast Packets	0xCC	0x08	0x00: Broadcast 0x01: Addressed			-	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
Auto Destination	0xCC	0x15	bit-0: Auto Destination bit-4: Enable Auto Destination			-	0xCC	bit-0: Auto Destination bits-1-7: 0		
Read API Control	0xCC	0x16	-	-	-	-	0xCC	API Control		-
Write API Control	0xCC	0x17	API Control		-	-	0xCC	API Control		-
Read Digital Inputs	0xCC	0x20	-	-	-	-	0xCC	bit-0: GI0 bit-1: GI1	-	-
Read ADC	0xCC	0x21	0x01: AD In 0x02: Temp 0x03: RSSI		-	-	0xCC	MSB of 10 bit ADC	LSB of 10 bit ADC	
Write Digital Outputs	0xCC	0x23	bit-0: GO0 bit-1: GO1		-	-	0xCC	bit-0: GO0 bit-1: GO1	-	-
Set Max Power	0xCC	0x25	New Max Power			-	0xCC	Max Power	-	-
Enter Probe	0xCC	0x8E	0x00: Enter Probe 0x01: Exit Probe			-	0xCC	0x00 or 0x01	-	-
Read Temperature	0xCC	0xA4	-	-	-	-	0xCC	Temp (C)	-	-
EEPROM Byte Read	0xCC	0xC0	Start Address		Length		0xCC	Starting Address	Length	Data
EEPROM Byte Write	0xCC	0xC1	Start Address		Length	Data	Starting Address		Length	Data written
Soft Reset	0xCC	0xFF	-	-	-	-	-	-	-	-

## 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.

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

### Firmware Version Request

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

Command: 0xCC 0x00 0x00

Number of Bytes Returned: 3

Response: 0xCC Version XX

Parameter Range: XX = 0x00 - 0x03 (Ignore)

### Change Channel

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

Command: 0xCC 0x01 Channel

Number of Bytes Returned: 2

Response: 0xCC Channel

### 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. Setting bit-7 of API Control to 1 can also enable Broadcast Packets.

Command: 0xCC 0x08 Data1

Number of Bytes Returned: 2

Response: 0xCC Data1

Parameter Range: Data1 = 0x00 for Addressed, 0x01 for 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: 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: 0x00 - 0xFF corresponding to 3 LSB's of destination MAC Address

### Auto Destination

The Host issues this command to change the Auto Destination setting. When issuing this command, the Auto Destination setting will only be changed if the corresponding enable bit is set (Control1 Parameter, EEPROM address 0x56, bit-4).

Command: 0xCC 0x15 Data1  
 Number of Bytes Returned: 2  
 Response: 0xCC Data2  
 Parameter Range: Data1 = bit-0: Auto Destination, bit-4: Enable Auto Destination modification; Data2 = bit-0: New Auto Destination setting, bits 2 - 7:0

### Read API Control

The OEM Host issues this command to read the API Control byte.

Command: 0xCC 0x16  
 Number of Bytes Returned: 2  
 Response: 0xCC API Control

### Write API Control

The OEM Host issues this command to write the API Control byte.

Command: 0xCC 0x17  
 Number of Bytes Returned: 2  
 Response: 0xCC API Control

### 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. The radios are shipped at maximum allowable power.

Command: 0xCC 0x25 Max Power  
 Number of Bytes Returned: 2  
 Response: 0xCC Max Power

### Read Temperature

The OEM Host issues this command to read the onboard temperature sensor. The transceiver reports the temperature in °C where 0x00 - 0x80 corresponds to 0 - 80 °C and where 0xD8 - 0x00 corresponds to -40 - 0 °C.

Command: 0xCC 0xA4  
 Number of Bytes Returned: 2  
 Response: 0xCC Temperature  
 Parameter Range: Temperature = 0xD8 - 0x80

### 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 Data1

---

Parameter Range: Data1 = bit-0: GI0, bit-1: GI1

---

### Read Radio Table

The OEM Host issues this command to read the Radio Table that resides on the transceiver. The Radio Table stores information for up to the last 8 transceivers that it received a packet from. This information can be useful for determining alternative data paths.

---

Command: 0xCC 0x18

---

Number of Bytes Returned: Varies

---

Response: 0xCC #Transceivers MAC2 MAC1  
MAC0 RSSI RSSI\* StaleCount MAC2 MAC1  
MAC0...etc.

---

**Stale Count:** The Stale Count Reload (0x04) determines the amount of time that a transceiver stays active in the Radio Table. The Stale Count (min: 0x00; max: dependent on EEPROM setting) for a radio is set to 0 when a packet is received; and then incremented by one every 100 ms thereafter. When the Stale Count of a transceiver reaches the Stale Count Reload (0x04), the transceiver is considered stale. A Radio Table can hold information for up to 8 different transceivers; however if the table is full and a ninth radio appears, the first stale radio is replaced with the new radio. If none of the radios are stale, the oldest radio is replaced by the new radio.

**Table 11: Received Signal Strength**

Signal Strength (dBm)	RSSI Value (Hex)	Signal Strength (dBm)	RSSI Value (Hex)
4	0x0E	-62	0x2B
-2 to 1	0x0D	-66	0x40
-12 to -6	0x0C	-69	0x55
-36 to -22	0x0B	-72	0x62
-42 to -39	0x0C	-76	0x71
-46	0x0D	-79	0x78
-49	0x0E	-82	0x84
-52	0x11	-86	0x9A
-56	0x17	-89	0xAD
-59	0x1C	-92	0xBD



## 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) \* 3.3V  
**Temperature (°C)** = ((Analog Voltage - 0.3) / 0.01) - 30  
**RSSI value (dBm)** = -105 + (0.22 \* (0x3FF - 10 bits))

Command: 0xCC 0x21 Data1

Number of Bytes Returned: 3

Response: 0xCC Data2 Data3

Parameter Range: Data1 = 0x00: AD In, 0x01: Temperature, 0x02: RSSI; Data2 = MSB of requested 10-bit ADC value; Data3 = LSB of requested 10-bit ADC value

## 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 Data1

Number of Bytes Returned: 2

Response: 0xCC Data1

Parameter Range: Data1 = bit-0: GO0, bit-1: GO1

## Probe

Enabling bit-6 of API Control will enable this command. When the OEM Host issues this command, the transceiver sends out a query every 500 ms. The transceivers, upon receiving the query, 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.

Apart from the transceiver response, there are two other responses that provide crucial information to the OEM Host. This information can be used to monitor the network and determine alternate routing paths.

### Probe Report

Remote transceiver's response to its OEM host upon receiving a Probe query.

**Note:** Only valid when Probe Report (address 0xC9) is set to 0xE3.

### Transceiver's Response

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

Command: 0xCC 0x8E Data1

Number of Bytes Returned: 2

Response: 0xCC Data1

Parameter Range: 0x00 = Disable Probe, 0x01 = Enable Probe

Command: N/A

Number of Bytes Returned: 5

Response: 0x86 RSSI MAC3 MAC2 MAC1

Parameter Range: MAC3 MAC2 MAC1 = 3 LSB's of radio sending the Probe query

Command: N/A

Number of Bytes Returned: 6

Response: 0x87 RSSI RSSI\* MAC3 MAC2 MAC1

Parameter Range: RSSI = How strong remote heard local transceiver; RSSI\* = How strong local heard remote transceiver

### EEPROM Byte Read

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

Command: 0xCC 0xC0 Data1 Data2

Number of Bytes Returned: 4+

Response: 0xCC Data1 Data2 Data3

Parameter Range: Data1 = EEPROM address; Data2 = Length (0x00 - 0x80); Data3 = 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).

Command: 0xCC 0xC1 Data1 Data2

Number of Bytes Returned: 4+

Response: 0xCC Data1 Data2 Data

Parameter Range: Data1 = EEPROM address; Data2 = Length (0x00 - 0x80); Data3 = 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 values stored in the EEPROM.

Command: 0xCC 0xFF

Number of Bytes Returned: None

Response: None

# EEPROM PARAMETERS

The OEM Host can program various parameters that are stored in EEPROM which become active after a power-on reset. The table below gives the locations and descriptions of the parameters that can be read/written by the OEM Host. Factory default values are also shown. 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.

**Table 12: EEPROM Parameters**

Parameter	EEPROM Address	Length (Bytes)	Range	Default	Description
<b>Product ID</b>	0x00	40			40 bytes - Product identifier string. Includes revision information for software and hardware.
<b>Stop Bit Delay</b>	0x3F	1	0x00 - 0xFF	0xFF	For systems employing the RS-485 interface or Parity, the serial stop bit might come too early. Stop bit delay controls the width of the last bit before the stop bit occurs. 0xFF = Disable Stop Bit Delay (12 us) 0x00 = (256 * 1.6 us) + 12 us 0x01 - 0xFE = (value * 1.6 us) + 12 us
<b>Channel Number</b>	0x40	1	0x00 - 0x37	1x1: 0x00 200: 0x00 1000: 0x10	Set 0 = 0x00 - 0x0F (US/Canada): 1x1/200 Set 1 = 0x10 - 0x2F (US/Canada): 1x1/1000 Set 2 = 0x30 - 0x37 (US/Canada): 1x1/200; Australia: 1x1/200/1000
<b>Baud Rate Low</b>	0x42	1	0x00 - 0xFF	0xFC	Low byte of the interface baud rate. Default baud rate is 57,600.
<b>Baud Rate High</b>	0x43	1	0x00	0x00	High byte of interface baud. Always 0x00
<b>Control 0</b>	0x45	1		0x00	Settings are: bit-7: 0 bit-6: DES Enable 0 = Disable 1 = Enable bits 5-0: 0
<b>Transmit Retries</b>	0x4C	1	0x01 - 0xFF	0x10	Maximum number of times a packet is sent out when Addressed packets are selected.
<b>Broadcast Attempts</b>	0x4D	1	0x01 - 0xFF	0x04	Maximum number of times a packet is sent out when Broadcast packets are selected.
<b>Stale Count Reload</b>	0x4F	1	0x01 - 0xFF	0x40	Determines the amount of time that a transceiver will keep a radio active in its Receive Table. This value is reset every time a packet is received from that radio.

## EEPROM PARAMETERS

### *Command Descriptions*

**Table 12: EEPROM Parameters**

Parameter	EEPROM Address	Length (Bytes)	Range	Default	Description
<b>Control 1</b>	0x56	1		0x43	Settings are: bit-7: Aerocomm Use Only bit-6: Aerocomm Use Only bit-5: Aerocomm Use Only bit-4: Auto Destination 0 = Use destination address 1 = Use auto destination bit-3: Aerocomm Use Only bit-2: RTS Enable 0 = Ignore RTS 1 = Transceiver obeys RTS bit-1: Duplex 0 = Half Duplex 1 = Full Duplex bit-0: Auto Config 0 = Use EEPROM values 1 = Auto Configure values
<b>Interface Timeout</b>	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).
<b>RF Packet Size</b>	0x5B	1	0x01 - 0x80	0x80	Used in conjunction with Interface Timeout; specifies the maximum size of an RF packet.
<b>CTS On</b>	0x5C	1	0x01 - 0xFF	0xD2	CTS will be deasserted (High) when the transmit buffer contains at least this many characters.
<b>CTS Off</b>	0x5D	1	0x00 - 0xFE	0xAC	Once CTS has been deasserted, CTS will be reasserted (Low) when the transmit buffer is contains this many or less characters.
<b>Max Power</b>	0x63	1	0x00 - 0x60	Set in Production & can vary	Used to increase/decrease the output power. The transceivers are shipped at maximum allowable power.
<b>Parity</b>	0x6F	1	0xE3, 0xFF	0xFF	0xE3 = Enable Parity 0xFF = Disable Parity <b>Note:</b> Enabling parity cuts throughput and the interface buffer size in half.
<b>Destination ID</b>	0x70	6	0x00 - 0xFF		Specifies destination for RF packets
<b>System ID</b>	0x76	1	0x00 - 0xFF	0x01	Similar to network password. Radios must have the same system ID to communicate with each other.
<b>RS-485 DE</b>	0x7F	1	0xE3, 0xFF	0xFF	0xE3 = GO0 is active Low DE for control of external RS-485 hardware 0xFF = Disable RS-485 DE
<b>MAC ID</b>	0x80	6	0x00 - 0xFF		Factory programmed unique IEEE MAC address.

## EEPROM PARAMETERS

### *Command Descriptions*

**Table 12: EEPROM Parameters**

Parameter	EEPROM Address	Length (Bytes)	Range	Default	Description
<b>Original Max Power</b>	0x8E	1		Set in Production and can vary	Copy of original max power EEPROM setting. This address may be referenced but should not be modified.
<b>Product ID</b>	0x90	15			0x90 - 0x93: Product ID 0x94 - 0x95: Prefix (CL, CN, or AC) 0x96 - 0x99: Power (200M, 200A, 1000, 1x1) <b>Note:</b> There will be a period in front of the 1x1 to keep the field at four bytes 0x9A - 0x9C: Interface (232, 485, TTL) 0x9D - 0x9E: Setup script (01 is stock) 0x9F: Reserved for future use; always 0xFF
<b>API Control</b>	0xC1	1		0x10	Settings are: bit-7: Broadcast packets 0 = Addressed Packets 1 = Broadcast Packets bit-6: Probe 0 = Disable Probe 1 = Enable Probe bit-5: SLock1 0 = Disable 1 = Enable bit-4: SLock0 0 = Disable 1 = Enable bit-3: Unicast Packets 0 = Broadcast or Addressed delivery 1 = Addressed packets only bit-2: Send Data Complete Enable 0 = Disable 1 = Enable bit-1: API Transmit Packet Enable 0 = Disable 1 = Enable bit-0: API Receive Packet Enable 0 = Disable 1 = Enable
<b>Protocol Status</b>	0xC2	1	0x00 - 0xFF	0xE3	Determines if the GO0 & GO1 server as generic output or as protocol status.
<b>Session Count Refresh</b>	0xC4	1	0x00 - 0xFF	0x08	Specifies the number of hops a transceiver stays in session with another transceiver

**EEPROM PARAMETERS**  
*Command Descriptions*

**Table 12: EEPROM Parameters**

Parameter	EEPROM Address	Length (Bytes)	Range	Default	Description
<b>Random Back-Off</b>	0xC3	1	0x00 - 0xFF	0x00	The random amount of time a transceiver waits when a collision occurs before resending the packet again.  0x00: Disable Random Backoff 0x01: Wait 1-2 packet times, then retry 0x03: Wait 1-4 packet times, then retry 0x07: Wait 1-8 packet times, then retry 0x0F: Wait 1-16 packet times, then retry 0x1F: Wait 1-32 packet times, then retry 0x3F: Wait 1-64 packet times, then retry 0x7F: Wait 1-128 packet times, then retry 0xFF: Wait 1-256 packet times, then retry
<b>Sesense Adjust</b>	0xC8	1	0x00 - 0xFF	Set in Production and can vary	The minimum RSSI required by a transceiver to establish a session upon hearing a long beacon.
<b>Probe Report</b>	0xC9	1	0x00 - 0xFF	0xE3	When set to 0xE3, upon receiving a probe the transceiver sends a Probe Report to the OEM Host.
<b>DES Key</b>	0xD0	7	0x00 - 0xFF		56-bit Data Encryption key

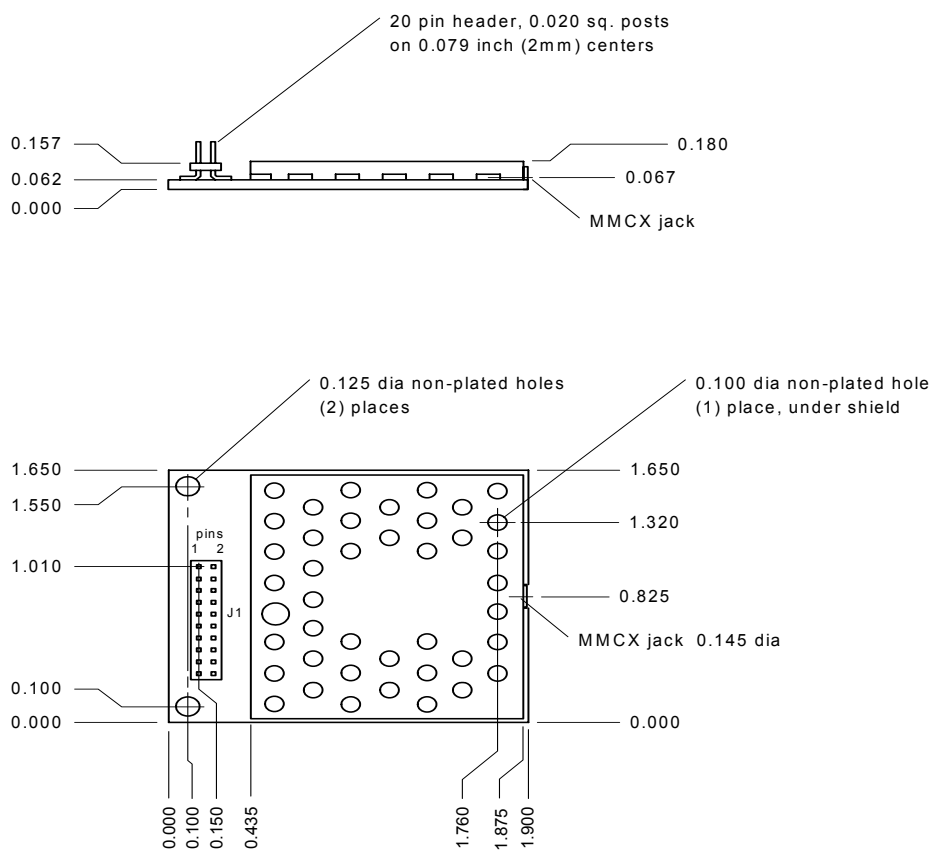
# DIMENSIONS

## MECHANICAL DRAWINGS

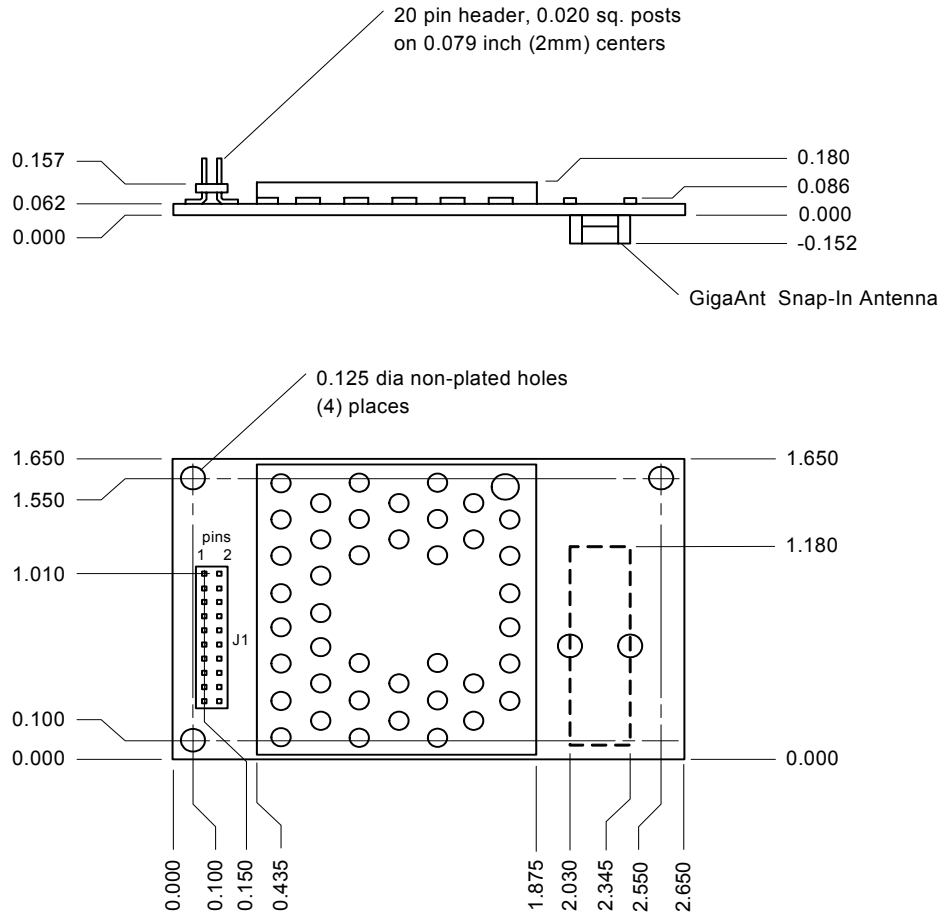
**Interface Connector** - 20 pin OEM Interface connector (Molex 87759-0030, mates with Samtec SMM-110-02-S-D)

**MMCX Jack** - Antenna Connector (Johnson Components 135-3711-822)AC4790 (with MMCX connector) Mechanical

**Figure 6: AC4790 (with MMCX connector) Mechanical**

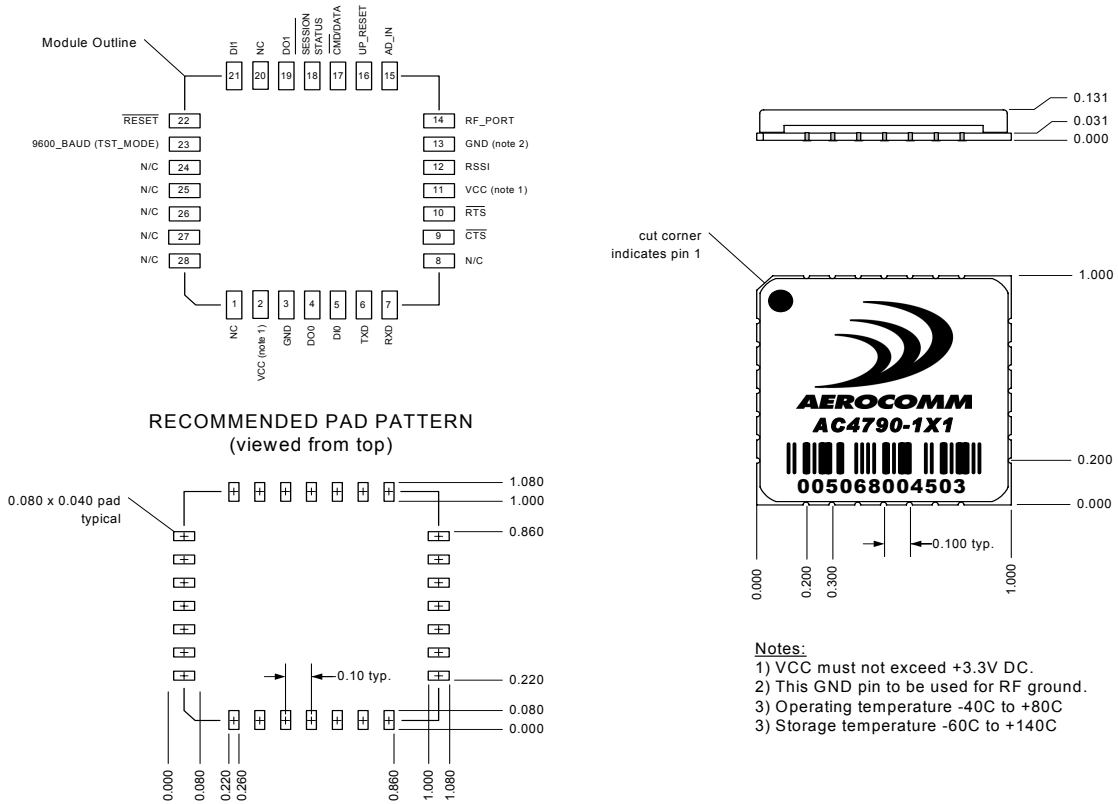


**Figure 7: AC4790 with integral gigaAnt Antenna (on bottom) Mechanical**





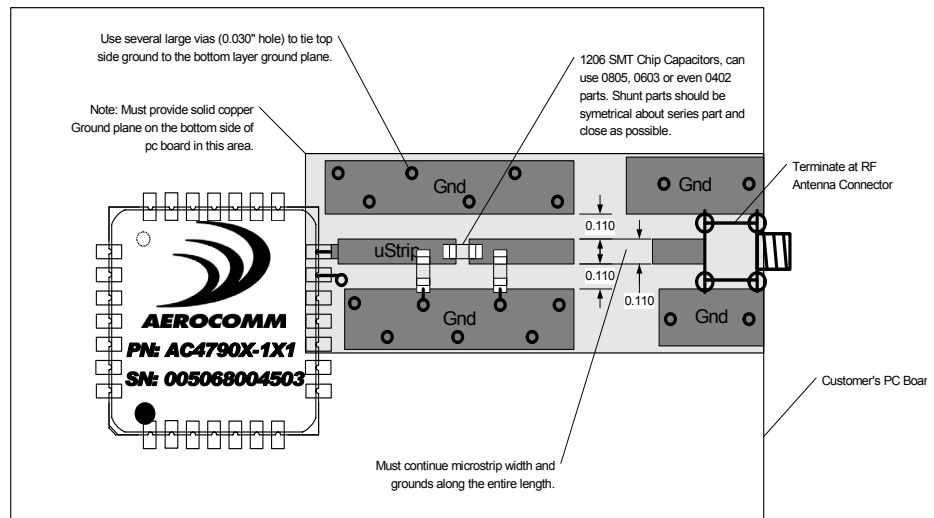
**Figure 8: AC4790-1x1 Mechanical**



- Notes:**
- 1) VCC must not exceed +3.3V DC.
  - 2) This GND pin to be used for RF ground.
  - 3) Operating temperature -40C to +80C
  - 3) Storage temperature -60C to +140C

**Figure 9: AC4790-1x1 PCB Considerations**

Note: Keep distance between 1x1 Module and antenna connector as short as possible for better performance.



PCB Thickness Notes:

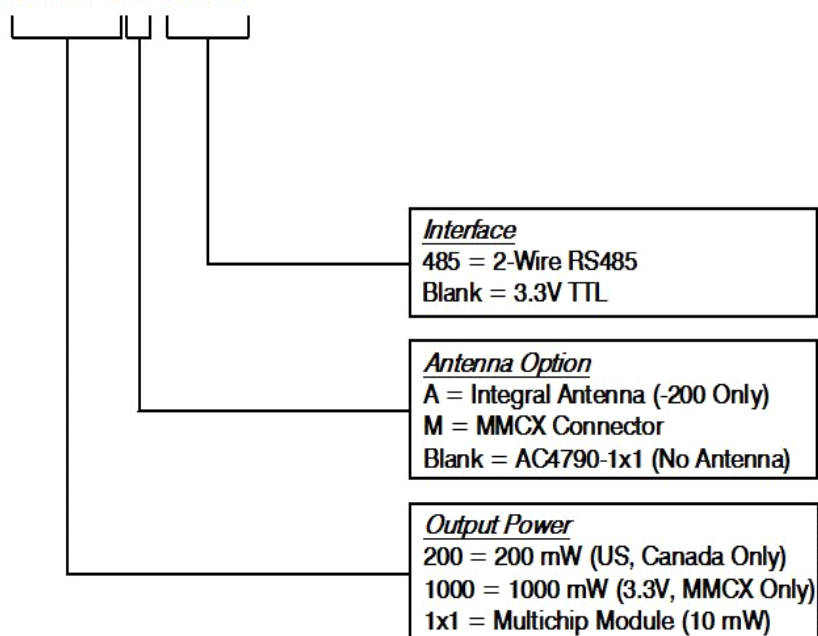
For 0.062 thick PC board microstrip width and spacing is 0.110 inches.

For 0.031 thick PC board microstrip width and spacing is 0.055 inches.

# ORDERING INFORMATION

## PRODUCT PART NUMBER TREE

AC4790-XXXXX-XXX



## DEVELOPER KIT PART NUMBERS

All of the above part numbers can be ordered as a development kit by prefacing the part number with "SDK-". As an example, part number AC4790-200A can be ordered as a development kit using the part number: SDK-AC4790-200A.

All developer's kits include (2) transceivers, (2) development boards, (2) 7.5 VDC unregulated power supplies, (2) serial cables, (2) USB cables, (2) antennas, configuration/testing software and integration engineering support.

# COMPLIANCE INFORMATION

## AC4790-1X1

Due to the RF antenna trace residing on the OEM Host PCB, the FCC will not grant modular approval for the AC4790-1x1 and requires the OEM to submit their completed design for approval. Contact AeroComm for the approval procedure.

## AGENCY IDENTIFICATION NUMBERS

Agency compliancy is a very important requirement for any product development. AeroComm has obtained modular approval for its products so the OEM only has to meet a few requirements to be eligible to use that approval. The corresponding agency identification numbers and approved antennas are listed below.

**Table 13: Agency Identification Numbers**

Part Number	US/FCC	Canada/IC
AC4790-200	KQLAC4490-100	2268C-AC4490
AC4790-1000	KQL-AC4490-1000	2268C-AC44901000

## APPROVED ANTENNA LIST

The following antennas are approved for use with the AC4790 as identified. The OEM is free to choose another vendor's antenna of like type and equal or lesser gain as a listed antenna and still maintain compliance.

**Table 14: AC4790 Approved Antennas**

AeroComm Part Number	Manufacturer Part Number	Manufacturer	Type	Gain (dBi)	200A	200M	1000M
0600-00019	S467-FL-5-RMM-915S	Nearson	1/2 Wave Dipole	2	-	X	X
0600-00025	S467-FL-5-RMM-915	Nearson	1/2 Wave Dipole	2	-	X	X
0600-00024	S467-AH-915	Nearson	1/2 Wave Dipole	2	-	X	X
0600-00027	S467-AH-915R	Nearson	1/2 Wave Dipole	2	-	X	X
0600-00028	S161-AH-915R	Nearson	1/2 Wave Dipole	2.5	-	X	X

**COMPLIANCY INFORMATION**  
*FCC / IC Requirements for Modular Approval*

**Table 14: AC4790 Approved Antennas**

AeroComm Part Number	Manufacturer Part Number	Manufacturer	Type	Gain (dBi)	200A	200M	1000M
0600-00029	S161-AH-915	Nearson	1/2 Wave Dipole	2.5	-	X	X
0600-00030	S331-AH-915	Nearson	1/4 Wave Dipole	1	-	X	X
	1020B5812-04 Flavus	gigaAnt	Microstrip	-0.5	X	-	-

## FCC / IC REQUIREMENTS FOR MODULAR APPROVAL

In general, there are two agency classifications of wireless applications; portable and mobile.

**Portable** - Portable is a classification of equipment where the user, in general, will be within 20 cm of the transmitting antenna. Portable equipment is further broken down into two classes; within 2.5 cm of human contact and beyond 2.5 cm (**Note:** Ankles, feet, wrists, and hands are permitted to be within 2.5 cm of the antenna even if the equipment is designated as being greater than 2.5 cm). The AC4790 is not agency approved for portable applications. The OEM is required to have additional testing performed to receive this classification. Contact AeroComm for more details.

**Mobile** - Mobile defines equipment where the user will be 20 cm or greater from the transmitting equipment. The antenna must be mounted in such a way that it cannot be moved closer to the user with respect to the equipment, although the equipment may be moved. (**Note:** Ankles, feet, wrists, and hands are permitted to be within 20 cm of mobile equipment).

## OEM EQUIPMENT LABELING REQUIREMENTS

**WARNING:** The OEM must ensure that FCC labeling requirements are met. This includes a clearly visible label on the outside of the OEM enclosure specifying the appropriate AeroComm FCC identifier for this product as well as the FCC notice below. The FCC identifiers are listed above.

**Contains FCC ID: KQLAC4490-100 / KQLAC4490**  
 The enclosed 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.

Label and text information should be in a size of type large enough to be readily legible, consistent with the dimensions of the equipment and the label. However, the type size for the text is not required to be larger than eight point.

## ANTENNA REQUIREMENTS

---

**WARNING:** This device has been tested with an MMCX connector with the above listed antennas. When integrated into the OEM's product, these fixed antennas require professional installation preventing end-users from replacing them with non-approved antennas. Any antenna not listed in the above table must be tested to comply with FCC Section 15.203 for unique antenna connectors and Section 15.247 for emissions. Contact AeroComm for assistance.

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

## WARNINGS REQUIRED IN OEM MANUALS

---

**WARNING:** This equipment has been approved for mobile applications where the equipment should be used at distances greater than 20 cm from the human body (with the exception of hands, feet, wrists, and ankles). Operation at distances of less than 20 cm is strictly prohibited and requires additional SAR testing.

## CHANNEL WARNING

---

The OEM must prevent the end-user from selecting a channel not approved for use by the FCC.

# APPENDIX I - SAMPLE POWER SUPPLY

Below is a simple switching power supply that provides enough current to easily power any Aerocomm OEM module. It utilizes low cost, off the shelf components that fit into a small area. This supply has an input voltage range of +6 volts to +18 volts and will output +3.4 volts at 1.5 amps.

Included is a schematic, bill of materials with manufacture's name and part numbers and a sample PCB layout. It is important to follow the layout suggestions and use large areas of copper to connect the devices as shown in the layout. It is also important to hook up the ground traces as shown and use multiple vias to connect input and output capacitors to the bottom side ground plane.

If the input voltage will be less than 12 volts then C1 and C2 can be replaced with a single 100uF 20 volt capacitor (same part number as C7). This will reduce board space and lower costs further. If you are powering an AC5124 module, R1 can be changed to a 373 ohm 1% resistor. This will change the output to +5 volts at 1.0 amps.

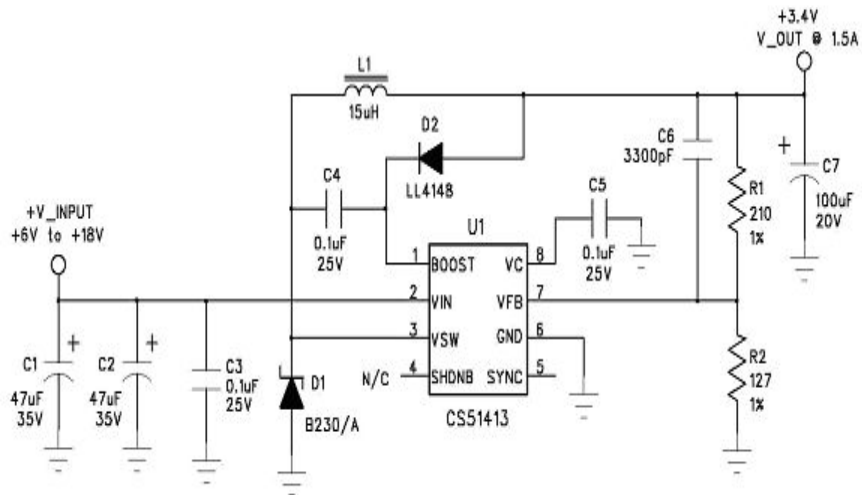
## BILL OF MATERIALS

**Table 15: Power Supply Bill of Materials**

Qty	Reference	Value	Description	Mfg.	Mfg. part number
1	R1	210	Res, 0603, 210, 1/16W, 1%	KOA	RK73H1JT2100F
1	R2	127	Res, 0603, 127, 1/16W, 1%	KOA	RK73H1JT1270F
2	C1 C2	47uF	Cap, Tant, 7343, 47uF, 35V	AVX	TPSE476M035R0200
3	C3 C4 C5	0.1uF	Cap, Cer, 0603, 0.1uF, Y5V, 25V	Murata	GRM39Y5V104Z025AD
1	C6	3300pF	Cap, Cer, 0603, 3300pF, X7R, 50V	Murata	GRM39X7R332K050AD
1	C7	100uF	Cap, Tant, 7343, 100uF, 20V	Kemet	T491X107K020A5
1	D1	B230/A	Diode, SMB, B230/A, 2A, Schott-key	Diodes, Inc.	B230/A
1	D2	LL4148	Diode, MELF, LL4148, Switch Diode	Diodes, Inc.	LL4148
1	L1	15uH	Xfmr, 2P, SMT, 15uH, 2A	Coiltronics	UP2.8B150
1	U1	CS51413	IC, CS51413, 8P, SO, Switch Reg Ctrl.	On-Semi-cond.	CS51413

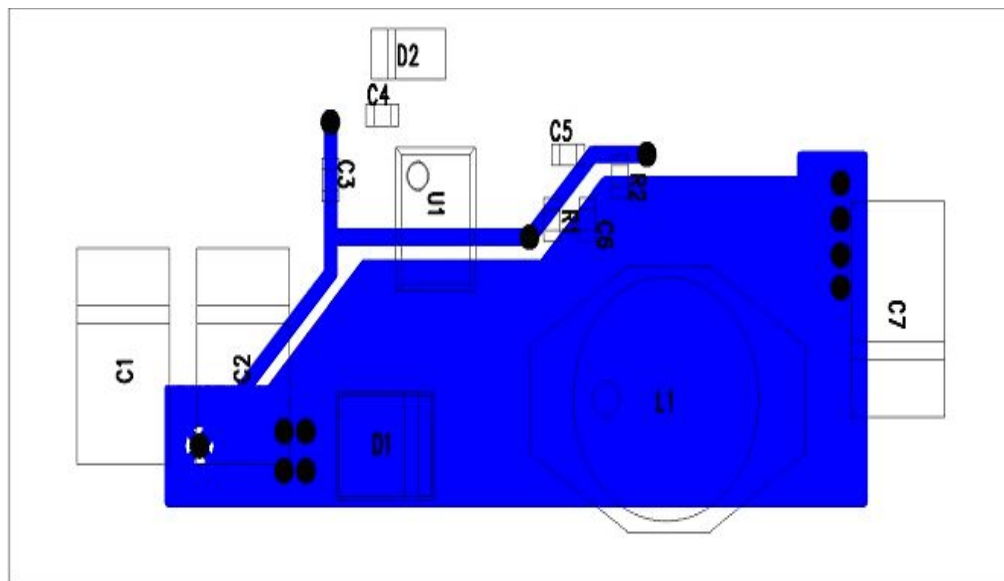
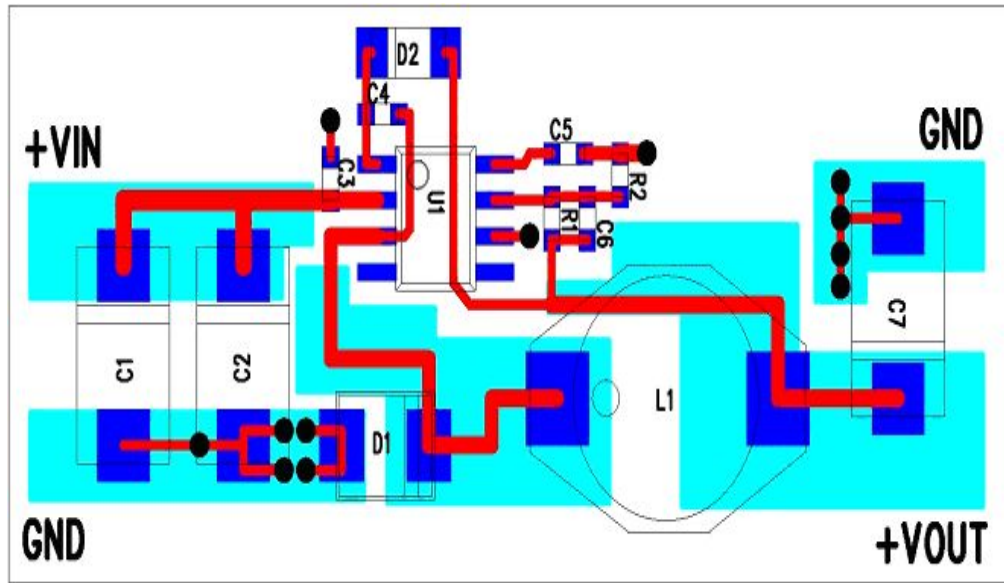
## SCHEMATIC

### Switching Power Supply





## PCB LAYOUT



## APPENDIX II - 5V TO 3.3V LEVELS

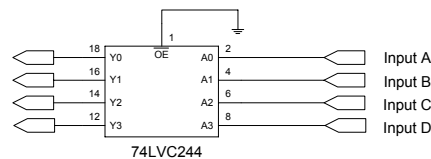
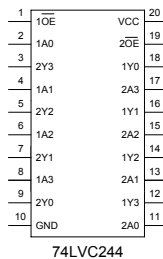


All inputs on the AC4790-200 & AC4790-1000 are weakly pulled high via 10 kohm resistors. The AC4790-200 has 5V inputs while the AC4790-1000 & AC4790-1x1 have 3.3V inputs. The AC4790-200 uses an octal buffer to drop the 5V to the required 3.3V level; the -1000 and -1x1 leave this to the OEM.

Some of the most common voltage conversion methods are described below.

### Voltage Level Conversion IC's

This is the easiest and most efficient method. Aerocomm recommends the TI SN74LVC244A Octal Buffer/Driver. Inputs can be driven from either 3.3 or 5V systems, allowing the device to be used in a mixed 3.3/5V system.



### Passive Resistor Voltage Divider

While a resistor voltage divider can successfully drop the 5V to the required 3.3V, it will draw static current all of the time. Typically this method is only suitable for one-way 5V to 3.3V conversion. When choosing the resistor values, one needs to include the radio's internal 10 kohm resistors on the input signals.

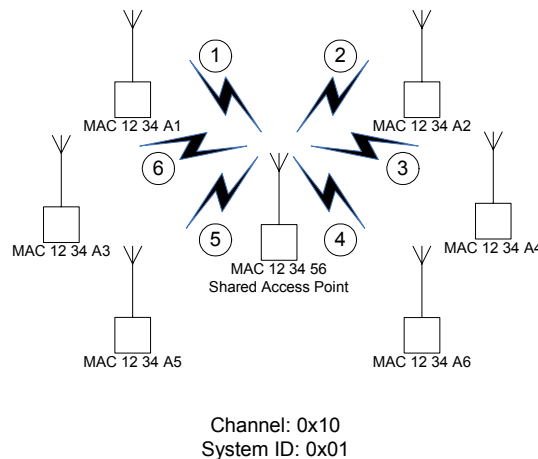
# APPENDIX III - API



The API feature set of the AC4790 provides powerful packet routing capabilities to the OEM Host. The number of API configurations is endless as individual radios can all be configured differently to suit the OEM Host's varying needs. Some of the most common implementations are described in the following pages.

## POLLING NETWORK

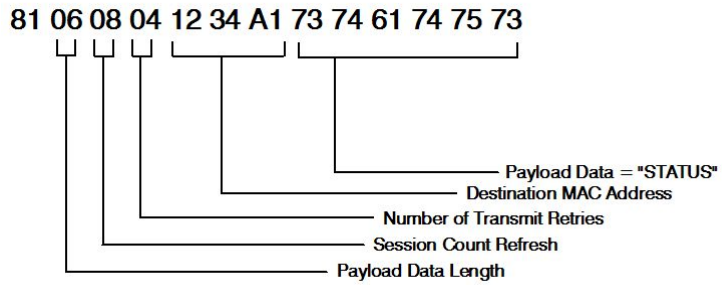
Many applications require multiple locations to report back to a single access point. One solution would be to enter Command mode, change the transceiver's destination address and then exit Command mode to resume normal operation. When it is time to communicate with another transceiver, the process would be repeated; costing time and inevitably reduction in throughput as unnecessary commands are issued. As an alternative, the Transmit API command can be used to control packet routing on a packet-by-packet basis.



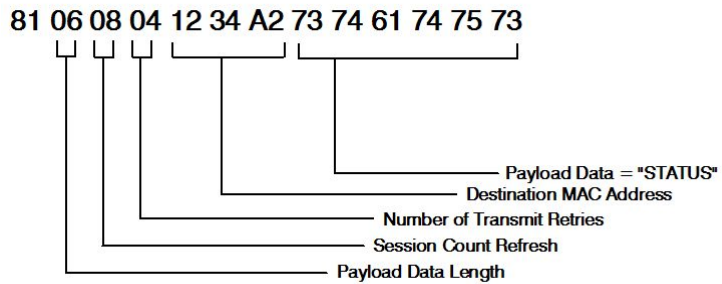
The simplest implementation consists of a smart Shared Access Point (SAP) with a microcontroller or processor of some type which has transmit API enabled. The SAP controls which transceiver(s) each packet is routed to. Broadcast packets should be used when all remotes are to receive the same message and addressed packets when communication with a single remote only is desired. An example of each is shown in the following pages.

### Addressed Transmit API

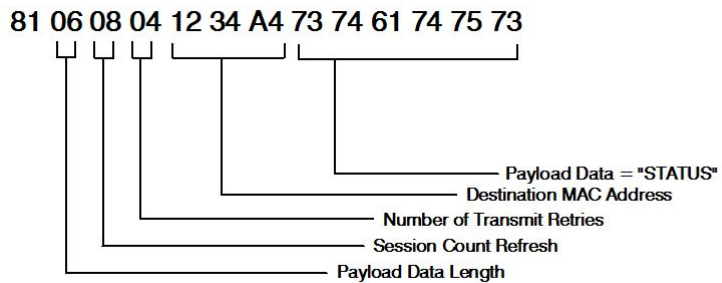
- 1 To poll radio 1, the SAP transmits the packet using the following format:



- 2 To poll radio 2, the SAP transmits the packet using the following format:



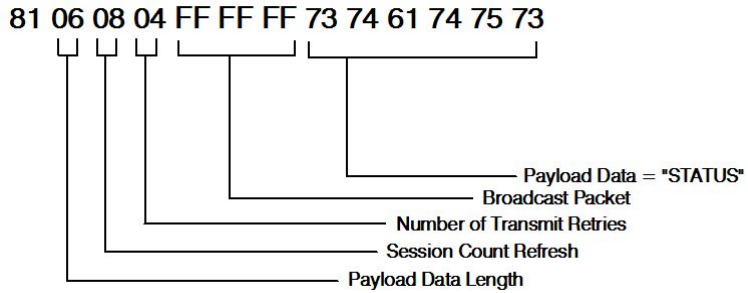
- 3 To poll radio 2, the SAP transmits the packet using the following format:



- 4 This continues until all radios have successfully been polled by the SAP.

### Broadcast Transmit API

To send out a universal poll request or data packet, the OEM may wish to utilize the broadcast portion of the Transmit API command. The Broadcast command is similar to the addressed command; only with the Destination MAC Address set to all 0xFF.

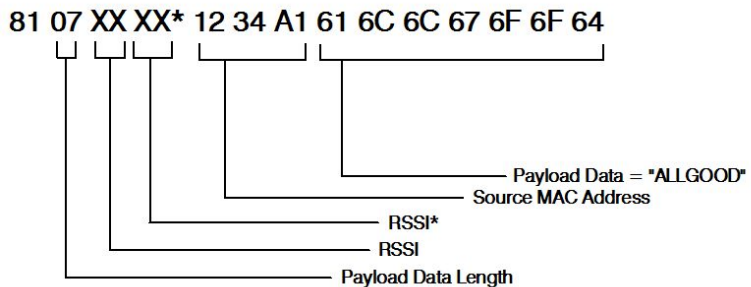


The remote response is dependent on the OEM's specific needs and equipment. In many cases, remote radios are connected to dumb devices without the intelligence to filter out or append specific portions of a packet that is transmitted or received. Since the 7 bytes of overhead in the Transmit API command are not sent over the RF, the remotes will receive only the payload data, "STATUS". If auto destination is enabled on the remote radio, the transceiver will automatically change its destination address to that of the radio it last received a packet from. When the remote device sends its response, it will therefore automatically be routed back to the SAP.

Depending on the API configuration of the SAP, the packet will be received in one of two formats:

### Receive API

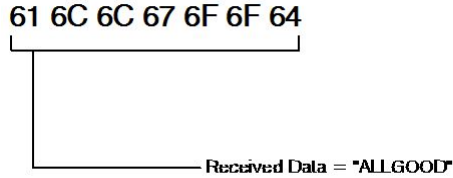
When Receive API is enabled, the transceiver will receive the reply data + the MAC address of the source radio and two RSSI values; RSSI is how strong the remote transceiver heard the local transceiver and RSSI\* is how strong the local heard the remote transceiver.



It may be useful to the OEM Host to determine which radio each packet originated from. When Receive API is enabled, every packet received by the transceiver will be received in the above format.

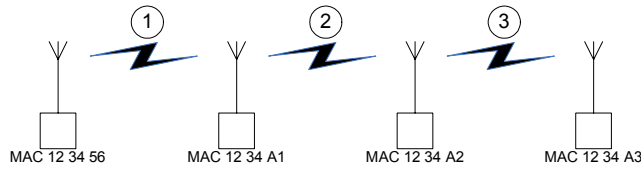
**Normal Receive Mode (non-API)**

If Receive API is not enabled, the transceiver will receive the reply data only (i.e. "ALLGOOD") from each transceiver.

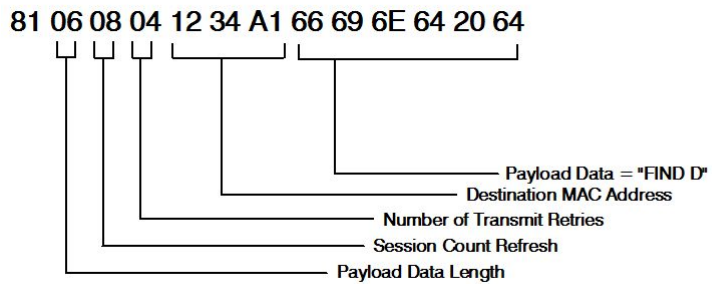


**DAISY CHAIN / REPEATER NETWORK**

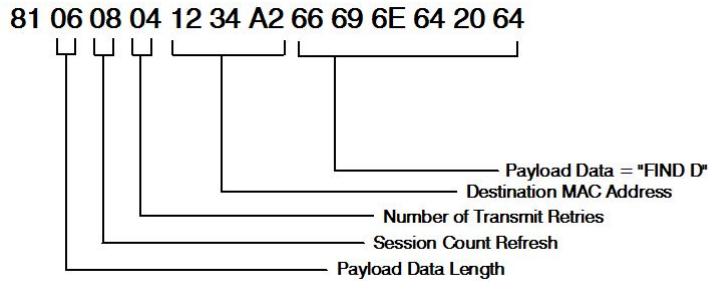
For applications spanning long distances and cases where the desired radio is not within range of the sending radio, a daisy chaining type network can be implemented. With the use of API commands, a processor and external buffer, a daisy chain or repeater can easily be implemented to store and forward the data to the desired radio. The example below assumes that radio A has a packet which needs to be received by radio D (far right).



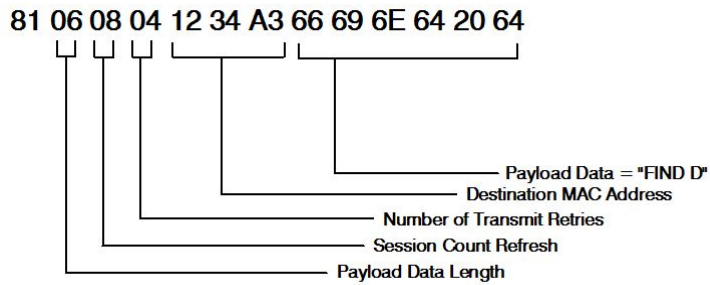
1 Radio A transmits the string "FIND D" to Radio B using the Transmit API command.



- 2 Radio B receives the packet "FIND D", and stores it in the buffer until the current session with Radio A has ended. Once the current session ends, Radio B forwards the packet from its buffer to Radio C.



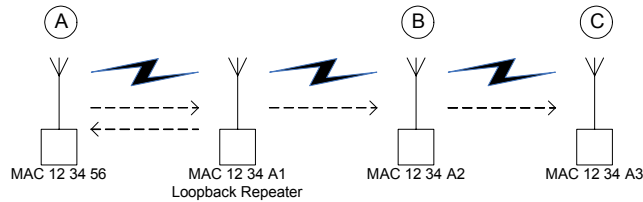
- 3 Radio C receives the packet "FIND D", and stores it in the buffer until the current session with Radio B has ended. Once the current session ends, Radio C forwards the packet from its buffer to Radio D.



- 4 Radio D receives the packet "FIND D" and sends the appropriate response back down the line to Radio A.

## LOOPBACK REPEATER

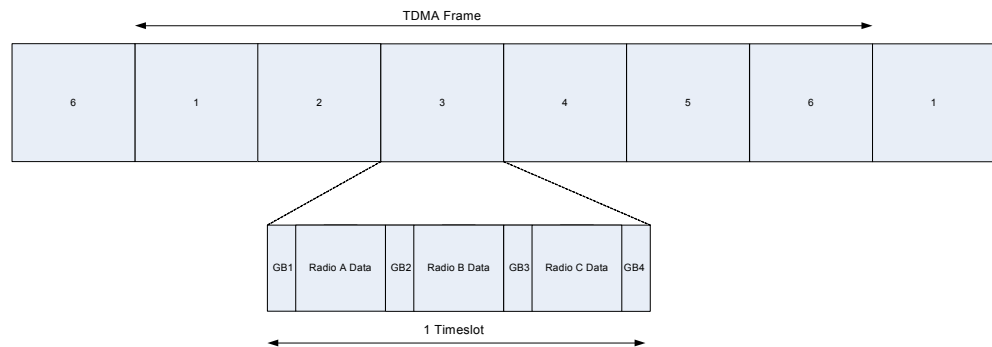
The simplest repeater to implement is a loopback repeater. A loopback repeater can be created by connecting the transceiver's RXD and TXD lines together. When the radio receives data, it will retransmit the data to all available transceivers on the network. It is important not to have two loopback repeaters in range of each other as they will continuously transmit data back and forth.



If radios B & C in the above picture are not within range of radio A, they will not be able to receive or respond to communications from radio A. A loopback repeater can be added between the three such that it is in range of both radio A and radios B & C. When the repeater receives a packet from radio A, it will transmit the packet out to radios B & C. If the repeater is set to Broadcast mode, radio A will receive a copy of each packet that it sends. If the repeater has a specific destination address (i.e. 12 34 A2), then radio A will not receive the packet as its MAC address will not match the specified destination address.

## TIME DIVISION MULTIPLE ACCESS NETWORK

For a more intelligent network, a TDMA system can be implemented. In this system various radios transmit data to a Shared Access Point (SAP) during an assigned time interval. The system is synchronous so that only one radio is transmitting at a time and has full access to the SAP's bandwidth. In a TDMA network, each radio must store its data for the amount of time between its transmissions or bursts. A typical format for data passing through a SAP is shown below. A frame consists of arriving bursts from remote radios and each frame is then divided into multiple time slots. The bursts can be of varying lengths and can be longer for heavy-traffic stations. To prevent overlaps, guard intervals can be inserted to absorb small timing errors in burst arrivals.





**Example:**

- **Shared Access Point (SAP) sends broadcast packet which includes a sync pulse**
- **Remote radios hear the sync pulse and join the session**
- **Radio A transmits during time interval  $t = 1$**
- **Radio B transmits during time interval  $t = 2$**
- **Radio N transmits during time interval  $t = N - 1$**

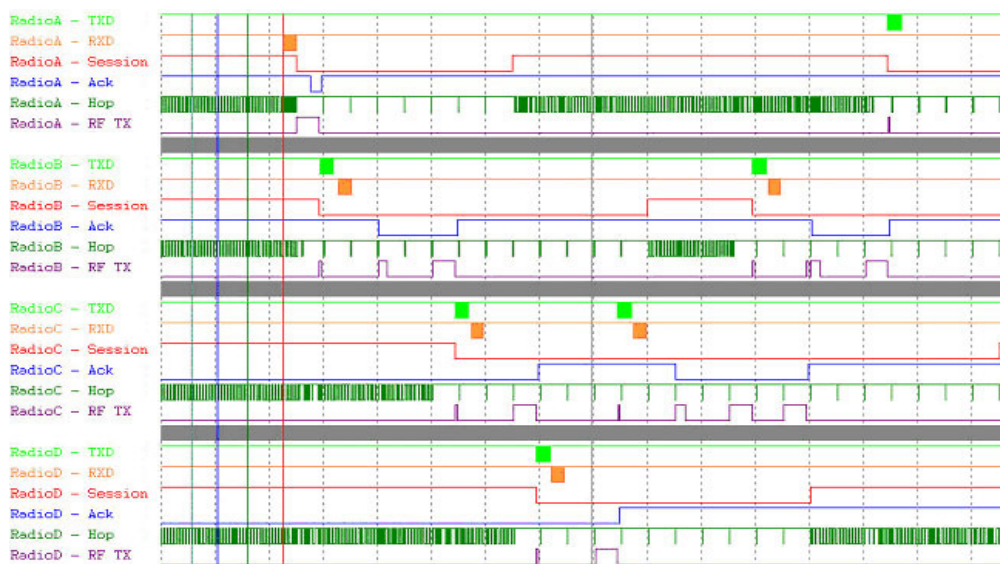
This type of implementation requires careful planning and should allow enough time for retries if necessary. When full duplex is enabled, the radio which initiated the Session (SAP) will transmit during the even numbered hops and the remote radios will transmit only during odd numbered hops.

# IV

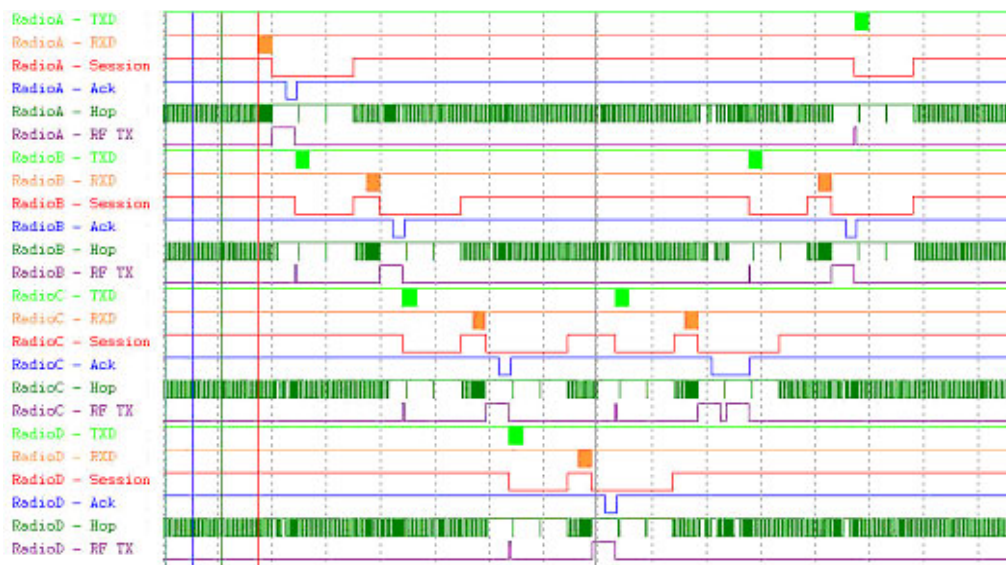
## APPENDIX IV - API TIMING DIAGRAMS

### TIMING DIAGRAMS

Session Count = 8, Retries = 3



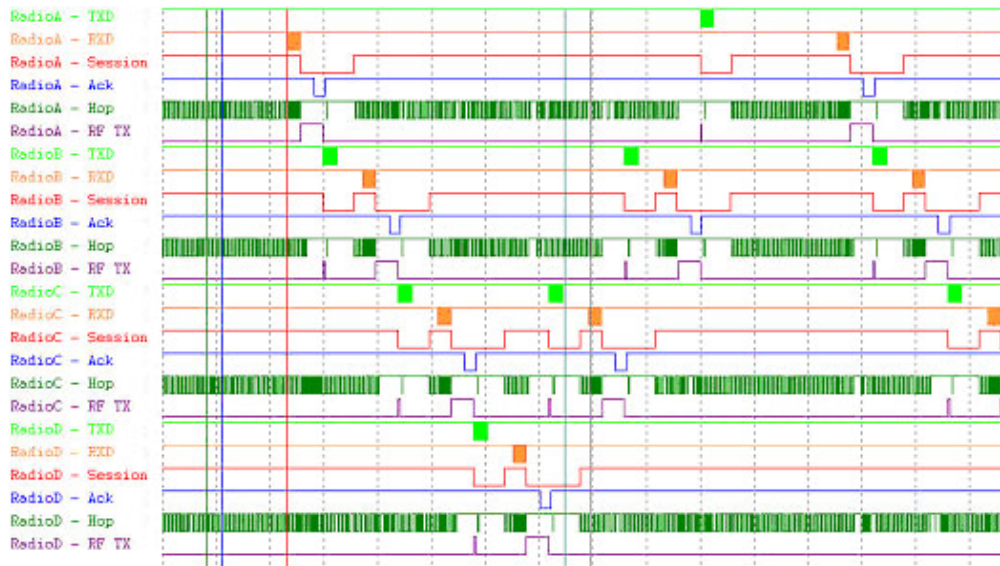
Session Count = 3, Retries = 3



# APPENDIX IV - API TIMING DIAGRAMS

## Timing Diagrams

Session Count = 2, Retries = 2



Session Count = 1, Retries = 1

