Physical Radio Interface Specification

Version: 12 Nov 1998; v1.0

Authors:[1] Dave Beyer (dave@rooftop.com),
Rooftop Communications

1. Purpose

Advanced software protocols for distributed packet radio networks are being designed, tested, and fielded by a variety of organizations including: Rooftop Communications, University of California, Santa Cruz, SRI International, Bolt Beranek and Newman, and the University of California, Los Angeles. Additionally, a variety of organizations including Hughes, UCLA, Virginia Polytechnic Institute, ITT, Utilicom and Hazeltine are developing next generation, highly-programmable digital radios and antennas to provide the reliable and flexible wireless links for such networks. These future networks promise to support efficient, reliable, and secure communication of multimedia traffic over rapidly-deployed, multihop wireless infrastructures, that can serve as seamless extensions of the Internet.

This Physical Radio Interface Specification was developed, and continues to evolve, to facilitate both collaboration and independent development of the network protocols and digital radio modem hardware. The intent is to allow protocol software and digital radio modems to be easily integrated, or "mixed and matched," into distributed packet radio products (or *Internet Radios*).

Specifically, this specification is intended to:

- Define a minimum, platform-independent, interface specification between digital radio modems and the transceiver frame control module used by network controller and its protocol software,
- Foster cross-organization collaboration between protocol and digital radio developers,
- Permit the implementation and testing of protocols in the absence of actual radio hardware,
- Provide standard methods for permitting radio-specific extensions, and
- Permit easy porting of protocols between multiple radios, and vice versa.

2. Architecture

The position of the Physical Radio Interface (PRI) is presented in Figure 1.[2] The Radio Device API has been defined with the intent to avoid restricting how the "physical radio" functions are actually implemented.

Specifically, the Physical Radio Interface assumes that the following general operations are provided by the physical radio (i.e., below" the Physical Radio Interface).

- RF & IF radio stages (mixers, filters, power amplifiers, low-noise-amplifiers, ...),
- Modulation,
- Baseband spreading (direct sequence and/or fast frequency hopping), and
- Spreading preamble generation, detection, and synchronization (if any),

[1] This work was supported by the Defense Advanced Research Projects Agency (DARPA). Refer to the *Acknowledgments* section for details.

[2] This document is a companion to the "Radio Device API" document which defines the interface at the TFCI boundary between the Link Control and Transceiver Frame Control modules in Figure 1.

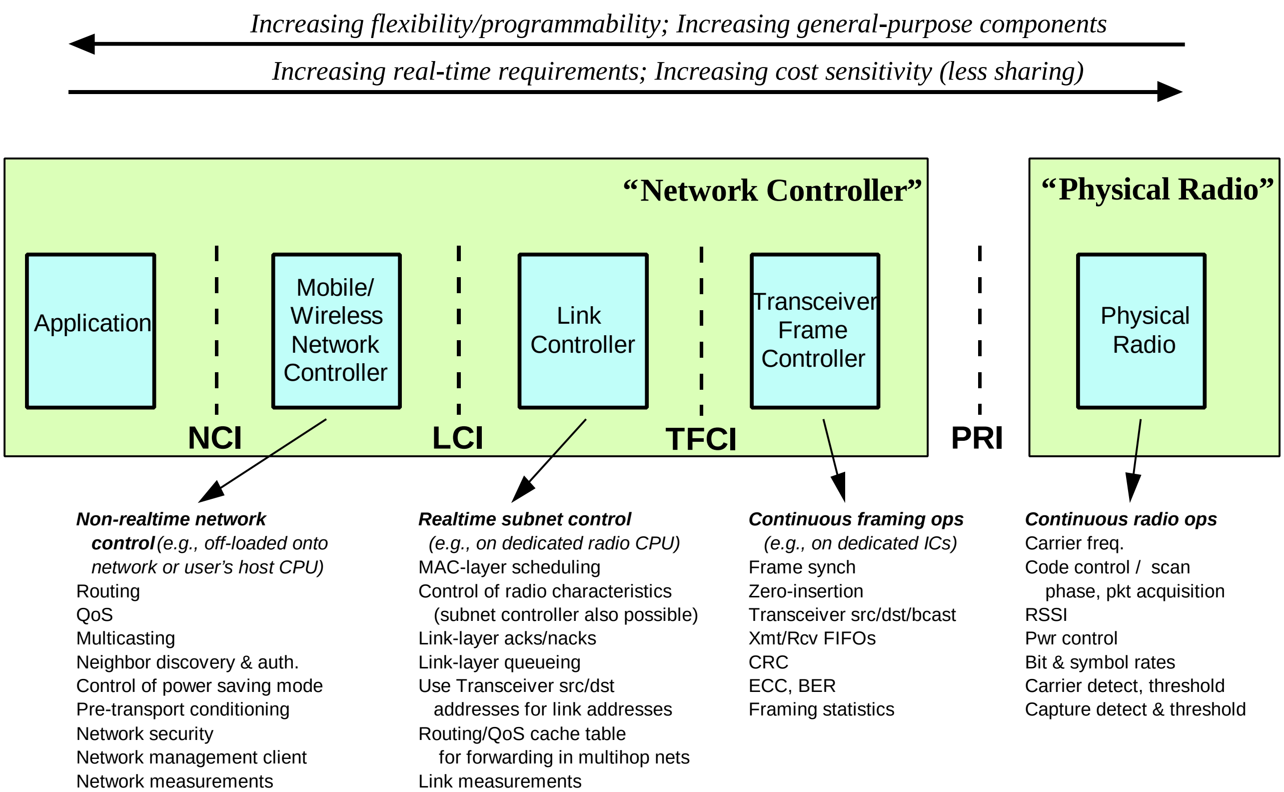

Figure 1: Position of the Physical Radio Interface

The Physical Radio Interface assumes that the following operations are performed by the software protocols (i.e., "above" the PRI):

- Framing (start/stop flags, zero-bit insertion, ...), and
- Error detection and/or correction (CRC computation, interleaving, error control coding).
- Media Access Control (MAC) protocols ("channel" scheduling and synchronization, avoidance of hidden terminal collisions),
- Link-layer protocols (reliable delivery of packets between neighbors or of local broadcast packets, fair sharing of link resources among neighbors, discovery and authentication of new neighbors),
- Network-layer protocols (efficient routing free of persistent loops despite mobility and dynamics, routing and queuing according to the service and priority requirements of the traffic, efficient multicasting, security of network control traffic), and
- Internetwork-layer protocols (wireless-to-wired Internet routing issues, network management, Internet-compatible interfaces).

Although this document is intended to provide a model for the interface to the physical radio, it is recognized that deviations and/or extensions from this interface may be appropriate for certain system architectures. For example, interleaving or error-control-coding may be best performed on the serial bit stream by digital signal processor components within the "Physical Radio" module. However, even for these cases, this specification should provide a baseline interface from which only an addendum need be added which details these deviations or extensions for a particular physical radio.

3. *Logical Functionality*

This section introduces the logical functionality of the Physical Radio Interface. The PRI is divided into two main "ports":

- Data Port
- Command Port

The Data Port identifies the circuits required to transfer the serial bit stream between the physical radio and the transceiver frame controller. The Command Port identifies other commands, variables, and signaling "primitives" required for the network controller to control and measure the transmission and reception characteristics of the physical radio.

The radio should support simultaneous communication on both the Command and Data Ports. However, for compatibility with some current radios, the interface specification also supports the concept of command and data "modes" where communication on the associated port is only permitted when in the corresponding mode.

3.1 Data Port

The PRI's Data Port defines the serial, bit-stream interface between the network controller and the physical radio. Rather than inventing new terms, the Data Port borrows the standard terminology of the RS-232, RS-422, and similar serial interface definitions. Using this standard terminology has a number of advantages:

- The specification is more easily understood by someone with knowledge of RS-232 terminology,
- Mapping the PRI Data Port specification onto RS-232, RS-422, RS-530 and similar serial interface connectors becomes trivial, and
- Many existing physical radio manufacturers use this terminology to define the interface to their current radio modems.

The Physical Radio module communicates user data in the form of a synchronous (clocked) bit stream. To send a packet, the network controller would assert the request-to-send (RTS) circuit. Upon receiving RTS, the radio will drop any current receive activity, start ramping up the transmit circuitry, and perform any other operations required at the start of a transmission (such as transmitting a spreading preamble to permit synchronization at the receiver). When the radio is ready to start transmitting data bits, it will assert the clear-to-send (CTS) circuit, and will activate the transmit data clock (TxClk). The network controller will then start transmitting data bits with each cycle of the TxClk signal. The physical radio should transmit these bits onto the RF channel with minimum latency through the physical radio module (preferably on the order of one bit period). Immediately after the network controller transmits the final bit in the packet (or burst of multiple back-to-back packets, it's all the same to the radio), it will deassert the RTS signal. The radio will respond by deasserting the CTS signal, ramping down the transmitter, and reverting to receive-mode.

For packet receptions, the radio will first activate the carrier/capture-detect (CD), and soon afterwards (see below for details), will activate the receive data clock (RxClk) to clock in each data bit in the received stream. At the end of the packet, the RxClk will be deactivated, followed by the deassertion of CD.

All operations required to transmit and receive a serial bit stream are performed by the radio. The network controller is responsible for any collision avoidance, link addressing, error checking, and retransmissions. In particular, the radio should not wait for a "clear channel" before responding to an RTS. The network controller is responsible for performing any carrier-sense multiple access (CSMA) or other MAC-layer algorithms. Also, the radio should not attempt to filter any received bit streams according to a link address. Thus, whenever the radio obtains synchronization of an incoming bit stream, it should forward that bit stream to the network controller (after stripping off the radio-dependent synchronization preamble, if any). All address recognition and related operations are performed by the network controller.

3.2 Command Port

The PRI's Command Port is defined by three basic types of *primitives*, as described in the following table and illustrated in Figure 2.

Commands — Asynchronous network controller-to-physical radio primitives for performing immediate, typically non-persistent actions. Example command primitives include: reset the radio, and drop receive capture.

Responses — Synchronous radio-to-controller responses to command or variable operations (using return codes). For this physical radio interface, synchronous responses are optional, with reliance on PhyRadSigEvent signals to report error conditions.

Variables — Persistent radio state or long-term measurement primitives that support one or more of the set, get, or increment synchronous access operations. Control variables include the raw channel bit rate, coding rate, center frequency, transmit power, and carrier-detect threshold. Measurement variables include the received signal strength and noise level.

Signals — Asynchronous network controller-to-physical radio primitives for reporting recent, typically non-persistent events. The PRI Command Port currently only specifies a general PhyRadSigEvent signal, for reporting various "return code" conditions, and which can be easily extended to report other radio-specific events.

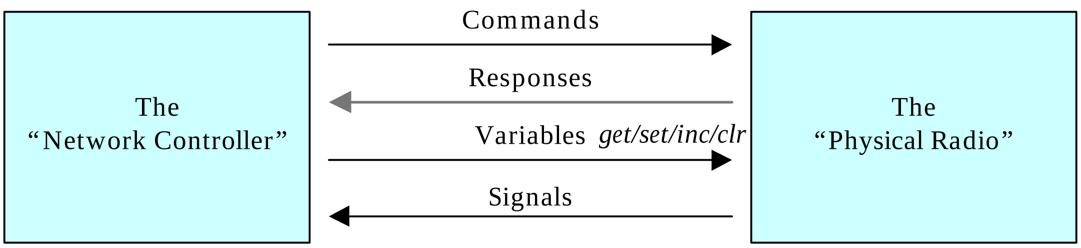

Figure 2: Basic Types of Command Port Primitives

Each primitive can be *qualified* to give more specific instructions such as specifying the radio "channel" to which the command should be applied (for radios that support multiple channels), and specifying which radio section (e.g., xmt or rcv) the operation should be applied. Of course, these qualifiers will only be relevant to radios that support these capabilities.

The Physical Radio implementation will vary on how these primitive operations are implemented. For example, they may consist of a set of registers and a hardware interrupt line used to communicate the occurrence of a signal primitive (with further information available in the register). Or, they may be implemented using a second, high-speed serial "command port" for communicating these operations via command, variables, and signal primitive-identifiers plus values.

The PRI also defines a minimal set of abstract return codes to provide a standard means for the radio to indicate the success or failure status for primitive operations or to report asynchronous error conditions with PhyRadSigEvent.

4. *Data Port Specification*

This section specifies the Data Port interface of the PRI. The interface is defined by a set of circuits modeled after the standard RS-232, RS-422, RS-530 and similar serial interfaces. In this terminology, the direction of the circuits is specified such that the Physical Radio is serving as the *Data Communications Equipment* (DCE) and the network controller as the *Data Terminal Equipment* (DTE).

For each circuit, its name, acronym, degree of requirement (mandatory, highly desirable, etc.) and general description are provided. Sequence diagrams are also provided for typical receive and transmit operations.

Note that the Data Port is used to communicate serial data between the network controller and a single physical radio channel. If a radio is capable of transmitting on multiple channels simultaneously, multiple data ports would be used.

4.1 Data Port Circuits

Transmit Data **TD**

Requirement: Mandatory

Direction: ® To Radio

Description: Used to transfer a serial bit stream from the network controller to the physical radio. Active while **CTS** is asserted and clocked by **TxClk**.

Receive Data **RD**

Requirement: Mandatory

Direction: ¬ From Radio

Description: Used to transfer a serial bit stream from the physical radio to the network controller. Active while **CD** is asserted and clocked by **RxClk**.

Transmit Clock **TxClk**

Requirement: Mandatory

Direction: ¬ From Radio

Description: Used to clock the serial bit stream on **TD** being transmitted by the network controller for the physical radio. Can only be active while **CTS** is asserted.

Receive Clock **RxClk**

Requirement: Mandatory

Direction: ¬ From Radio

Description: Used to clock the serial bit stream on **RD** being transmitted by the physical radio for the network controller. Can only be active while **CD** is asserted.

Request To Send **RTS**

Requirement: Mandatory

Direction: ® To Radio

Description: Causes the radio to abort any receiver activity, ramp-up the transmitter, and perform any other operations required before serial data can be transmitted. Note that by toggling **RTS,** this circuit can also be used by the network controller to abort any receive activity (e.g., code synchronization) and start a fresh search for receive data.

Clear To Send **CTS**

Requirement: Mandatory

Direction: ¬ From Radio

Description: Indicates that the radio is ready to start transmitting data, as generated by the network controller on the **TD** circuit, and clocked by **TxClk**.

Carrier or Capture Detect **CD**

Requirement: Highly Desirable

Direction: ¬ From Radio

Description: Depending on the PhyRadVarCdMode variable (see below), this is used to indicate that the radio is either receiving a strong RF carrier signal (presumably generated by a neighbor radio), or has gained synchronization of the incoming bit (or spreading code) sequence.

Data Terminal Ready **DTR**

Requirement: Highly Desirable

Direction: ® To Radio

Description: Used to indicate that the network controller is present and ready to receive data. No receive data should be forwarded to the network controller while this line is not asserted (i.e., **CD** and **RxClk** should never be active while **DTR** is not asserted). For radios that are limited to operation in one of two modes ("Command Mode" or "Data Mode"), toggling this circuit should cause the radio to switch from Data Mode to Command Mode.

Data Set Ready **DSR**

Requirement: Desirable

Direction: ¬ From Radio

Description: Used to indicate that the physical radio is present. For radios that are limited by operation in one of two modes ("Command Mode" or "Data Mode") , this circuit may optionally be used to indicate which mode the radio is in currently, with asserted meaning Data Mode.

4.2 Data Port Sequence Diagrams

The following diagrams present typical sequence diagrams for the transmit and receive operations on the Data Port. For these diagrams, we use positive logic for all control and data circuits (i.e., “high voltage” means asserted) and the positive transition for clocks are used for clocking the data. However, note that the actual polarity for all of these circuits will depend on the implementation specifics (RS-232, RS-422, positive logic TTL, negative logic TTL, etc.).

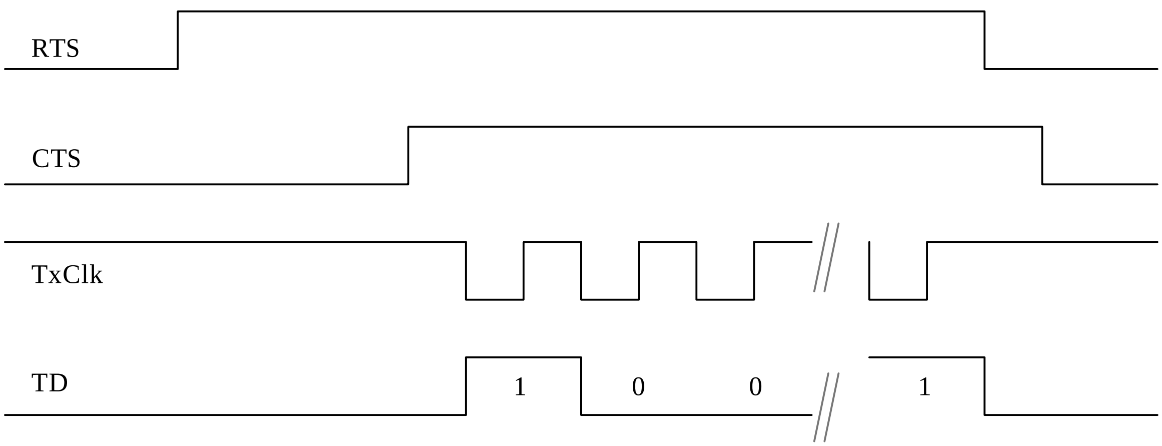

Figure 3: Typical Data Port Sequence for Transmitting Data

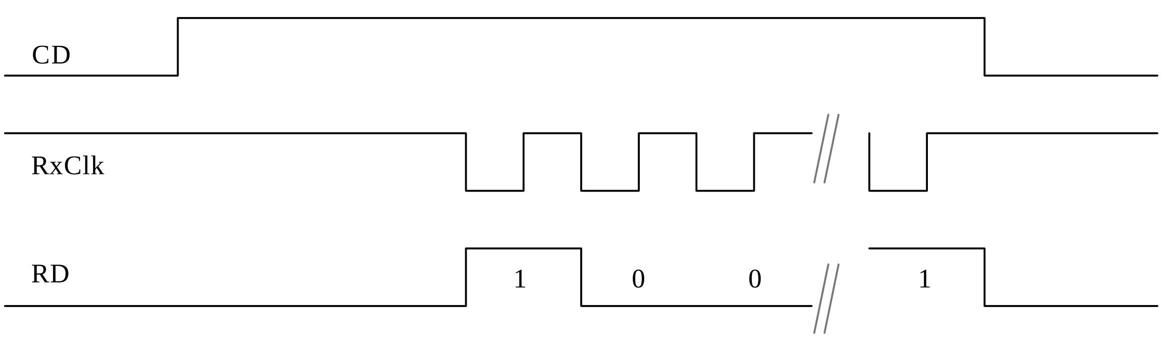

Figure 4: Typical Data Port Sequence for Receiving Data

5. *Command Port Specification*

This section identifies and describes the command, variable, and signal primitives of the PRI's Command Port. Each primitive is labeled with Mandatory, Highly desirable, Desirable, or Optional, indicating the degree of requirement by the network controller. A "Data" field indicates the generic input and/or output data communicated across the API by each primitive. Also, each primitive can be qualified by one or more of the following:

get	Indicates that the primitive should support *get* operations. If only get (and not set or inc) is specified, then the variable is read-only.
set	Indicates that the primitive should support *set* operations. If only *set* or *inc* (and not get) is specified, then the variable is write-only.
inc	Indicates that the primitive should support *increment* operations. Increment operations instruct the radio device to add the (possibly negative) value passed to the current value.
clr	Indicates that the primitive should support *clear* operation. Clear will return the state of variables to their initialized state.
xmt/rcv	Indicates that the primitive should be supported for both the transmitter and receiver sections individually, for radios that can support it.
chNum	Indicates that the primitive should be supported on a channel-specific basis, for radios that support multiple simultaneous channels.

Unlike the Radio Device API, it is not assumed that single get-and-set operations can be performed on variable primitives (even those that can support both operations individually) due to the more limited capabilities of the PRI's Command Port relative to the function-call interface of the Radio Device API.

5.1 Commands

This section lists the synchronous commands that can be issued by the controller to the radio.

PhyRadCmdReset Command

Requirement: Mandatory

Qualifiers:

Data:

Description: A command used to reset the radio. Any receive or transmit packet buffers should be returned to the protocols using the PhyRadSigRcvPkt and PhyRadSigXmtPkt signals with the RadioRetPktRcvFail or RadioRetPktXmtFail return code. If performed through a function call, the function should "block" until the reset operation has completed.

PhyRadCmdDropCapture Command

Requirement: Highly desirable (for DS radios only)

Qualifiers: *chNum*

Data:

Description: Commands the radio to drop code-synchronization of an incoming chip stream. Return receiver to searching for code-synchronization. Note that this may instead (or also) be implemented using the **RTS** circuit on the Data Port. Refer to the description for **RTS**.

PhyRadCmdProcExec Command

Requirement: Optional

Qualifiers: xmt/rcv

Data: Test or other radio procedure to execute

Description: Request that the radio executes a self-test or other procedure. The result is returned in PhyRadSigProcResults.

PhyRadCmdNativeConsole Command

Requirement: Optional

Qualifiers:

Data: String to be delivered to radio's "console", and the returned response string.

Description: Send a command string to the radio's native "console" and return the response string, if any. The radio's native console refers to the radio's ASCII command interpreter that may be available to the user by connecting a dumb terminal (or terminal emulator) directly to a serial port (or "Command Port") on the radio.

5.2 Variables

This section lists the radio state variables that should be available to the controller.

PhyRadVarVersion Variable

Requirement: Optional
Qualifiers: *get*
Data: Returned string or number.
Description: A read-only variable that provides a type and version string.

PhyRadVarBitRate Variable

Requirement: Highly Desirable
Qualifiers: *get/set, chNum, xmt/rcv*
Data: Raw channel bit rate
Description: The data rate in terms of units or a table index as defined by the documentation for the specific physical radio.

PhyRadVarXmtPower Variable

Requirement: Highly Desirable
Qualifiers: *get/set, chNum*
Data: Transmission power
Description: The RF transmission power in terms of units or a table index as defined by the documentation for the specific physical radio.

PhyRadVarFreq Variable

Requirement: Highly Desirable
Qualifiers: *get/set, chNum, xmt/rcv*
Data: Center frequency
Description: The RF center frequency in terms of units or a table index as defined by the documentation for the specific physical radio.

PhyRadVarCarrierThresh Variable

Requirement: Desirable
Qualifiers: *get/set, chNum*
Data: Carrier-detection threshold
Description: The receive carrier detection threshold, in terms of units or a table index as defined by the documentation for the specific physical radio.

PhyRadVarRcvSignal Variable

Requirement: Highly Desirable
Qualifiers: *get, chNum*
Data: Receive signal power
Description: The current receive signal power measurement, in terms of units or a table index, and averaged over some period, as defined by the documentation for the specific physical radio.

PhyRadVarRcvNoise Variable

Requirement: Desirable
Qualifiers: *get, chNum*
Data: Receive noise power
Description: The current receive noise power measurement, in terms of units or a table index, and averaged over some period, as defined by the documentation for the specific physical radio.

PhyRadVarCode Variable

Requirement: Highly Desirable (for DS radios)
Qualifiers: *get/set, chNum, xmt/rcv*
Data: PN code
Description: The pseudo-random code in terms of units or a table index as defined by the documentation for the specific physical radio.

PhyRadVarTestMode Variable

Requirement: Desirable
Qualifiers: *get/set, chNum*
Data: Test mode (e.g., PhyRadTestLoopbackBaseband, PhyRadTestLoopbackIf, PhyRadTestLoopbackRf, PhyRadTestXmtContinuous)
Description: Used for debugging, this variable indicates the current test mode of the radio. The test mode must be "disabled" by default after power-up or reset.

PhyRadVarCaptureState Variable

Requirement: Desirable
Qualifiers: *get, chNum*
Data: Receive capture state
Description: Indicates whether the radio receiver is currently synchronized to an incoming bit or code sequence.

PhyRadVarPowerMode Command

Requirement: Highly Desirable
Qualifiers: *set/get chNum*
Data: Power-down mode to enter.
Description: Used to put the radio into a low-power, "sleep" or "standby" modes and to return it to active normal power mode.

PhyRadVarCdMode Variable

Requirement: Desirable (for DS radios)
Qualifiers: *get/set, chNum, rcv*
Data: CD Mode
Description: Determines how the Data Port's **CD** circuit is used. Alternatives are to indicate when the received carrier signal power has risen above **PhyRadVarCarrierThresh,** or when the receiver had "captured" synchronization of the incoming bit or code sequence (the default). In any case, the CD signal should always be asserted when the radio is clocking receive data to the controller.

PhyRadVarCodeRate Variable

Requirement: Desirable (for DS radios)
Qualifiers: *get/set, chNum, xmt/rcv*
Data: PN code rate
Description: The pseudo-random code rate (e.g., chips/sec) in terms of units or a table index as defined by the documentation for the specific physical radio.

PhyRadVarCodeOffset Variable

Requirement: Optional (for DS radios)
Qualifiers: *get/set, chNum, xmt/rcv*
Data: PN code offset
Description: The offset in a pseudo-random code sequence, in terms of units or a table index as defined by the documentation for the specific physical radio.

PhyRadVarSymbolRate Variable

Requirement: Desirable
Qualifiers: *get/set, chNum, xmt/rcv*
Data: Symbol rate of modulator
Description: Select the rate of symbols transmitted or received by the modululator.

PhyRadVarModulationType Variable

Requirement: Desirable
Qualifiers: *get/set, chNum, xmt/rcv*
Data: Modulation Type
Description: Select the modulation type (e.g., BPSK, QPSK, ...)

PhyRadVarScrambleMode Variable

Requirement: Desirable
Qualifiers: *get/set, chNum, xmt/rcv*
Data: Scramble mode
Description: Selects the scrambling mode for the baseband module.

5.3 Asynchronous Signals

This section lists the asynchronous signals generated by the radio.

PhyRadSigEvent Signal

Requirement: Highly Desirable
Qualifiers: *chNum*
Data: A number indicating the error or event, as defined in the documentation
Description: A signal indicating that a radio error or significant event has occurred.

PhyRadSigAntSel Signal

Requirement: Desirable
Qualifiers: *chNum*
Data: ID of antenna to switch to
Description: A signal used by the radio to command the network controller to switch the active antenna (used for diversity purposes when the radio doesn't directly control the antenna selection switch).

PhyRadSigCarrier Signal

Requirement: Desirable
Qualifiers: *chNum*
Data: On/Off state of carrier detection
Description: A signal used to indicate that a receive signal has exceeded PhyRadVarCarrierThresh threshold. This signal is useful when the Data Port's "CD" circuit is used to indicate "Receive Capture" (the default) rather than "Receive Carrier Detect."

PhyRadSigProcResults Signal

Requirement: Optional
Qualifiers: *chNum*
Data: Identifier of self-test or other procedure executed along with result of procedure.
Description: Issued in response to a PhyRadCmdProcExec command.

5.4 Return Codes

To the extent permitted by the interface implementation, each Command, Variable, and Signal over the Command Port should be tagged with a return code that the radio can use to indicate the result or status of the operation. Table 1 lists standard names and definitions for return codes.

Table 1: Physical Radio Return Codes

Return Code	Description
PhyRadRetOk	Operation accepted or successfully performed.
PhyRadRetFail	General request failure.
PhyRadRetTimeOut	Request timed out.
PhyRadRetHwFail	Hardware failure.
PhyRadRetInvState	Operation not permitted in current state.
PhyRadRetInvCmd	Invalid command or command not implemented by this radio.
PhyRadRetInvVar	Invalid variable or variable not implemented by this radio.
PhyRadRetInvSig	Invalid signal or signal not implemented by this radio.
PhyRadRetInvQual	Invalid qualifier.
PhyRadRetInvParam	Invalid paramater used with primitive operation.
PhyRadRetRcvError	Error detected while receiving a packet.
PhyRadRetXmtErro	Error detected while transmitting a packet.

For instance, for a bus/register implementation, a numeric return code may be available in a "result" register following each operation. For an AT Command Port implementation, this return code may be available in an S register, or be a numeric code tacked onto the "ERROR" ASCII result (e.g., ERROR 5).

5.5 Summary of Primitives

Table 2 summarizes the names, degree of requirement (M-Mandatory, H-Highly desirable, D-Desirable, O-Optional), qualifiers, and data for each of the Physical Radio's Command Port primitives.

Table 2: Summary of Physical Radio Command Port Primitives

Commands	**Rqmt**	**Qualifiers**	**Data**
PhyRadCmdReset	M		
PhyRadCmdDropCapture	H*	*chNum*	
PhyRadCmdProcExec	O	*chNum*	Self-test or procedure to execute
PhyRadCmdNativeConsole	O		Cmd string and response string
Variables	**Rqmt**	**Qualifiers**	**Data**
PhyRadVarVersion	O	*get*	Version string or number
PhyRadVarBitRate	H	*get/set, chNum, xmt/rcv*	Raw channel bit rate
PhyRadVarXmtPower	H	*get/set, chNum*	Transmission power
PhyRadVarFreq	H	*get/set, chNum, xmt/rcv*	Center frequency
PhyRadVarCarrierThresh	D	*get/set, chNum*	Carrier-detect threshold
PhyRadVarRcvSignal	H	*get, chNum*	Receive signal power
PhyRadVarRcvNoise	D	*get, chNum*	Receive noise power
PhyRadVarCode	H*	*get/set, chNum, xmt/rcv*	PN code
PhyRadVarTestMode	D	*get/set, chNum*	Test mode
PhyRadVarCaptureState	D	*get/set, chNum*	Capture state of receiver
PhyRadVarPowerMode	H	*get/set chNum*	Power-down mode for radio device
PhyRadVarCdMode	D	*get/set, chNum*	Use of CD for Capture or Carrier
PhyRadVarCodeRate	D*	*get/set, chNum, xmt/rcv*	PN code rate
PhyRadVarCodeOffset	O*	*get/set, chNum, xmt/rcv*	PN code offset
PhyRadVarSymbolRate	D	*get/set, chNum, xmt/rcv*	Symbol rate
PhyRadVarModulationType	D	*get/set, chNum, xmt/rcv*	Modulation type
PhyRadVarScrambleMode	D	*get/set, chNum, xmt/rcv*	Baseband bit scrambling mode
Signals	**Rqmt**	**Qualifiers**	**Data**
PhyRadSigEvent	H	*chNum*	Error or event string or number
PhyRadSigAntSel	D	*chNum*	Radio's signal to switch antennas
PhyRadSigCarrier	D	*chNum*	Receive carrier detected[3]
PhyRadSigProcResults	O	*chNum*	Results of self-test or other procedure

* For direct-sequence spreading radios.

[3] Useful when the Data Port's "CD" line is used to indicate "Receive Capture" rather than "Carrier Detect."

6. *Physical Radio Implementations*

This section provides two brief overviews of example implementation methods for the Physical Radio Interface.[4]

6.1 Radio Daughter Board Implementation

This implementation is based on communication between the network controller and a radio daughter board over a 40-pin SAMTEC[5] connector, part number TFM-120-02-S-D on the radio (and SFM-120-02-S-D on the network controller). The following table presents the pin-list for this implementation. The synchronous serial Data Port (shaded) is complete except for DSR (radio is always assumed ready). The Command Port primitives are communicated using either a serial protocol and the CmdEn, CmdSel0, CmdSel1, CmdRW, CmdClk, and CmdData circuits (shaded) to clock buffer address plus data bits to and from the radio, or using individual control and status circuits ("Cntl-x" & "Stat-x") (shaded). For these individual circuits, suggested use is noted in italics in the Table.

Figure 5 presents typical sequencing of a "read" operation over the command port. The first set of CmdData bits identifies the command, or variable register (i.e., the API primitive), and the second set of bits is used to transfer the data. The CmdData bit must be valid for the duration of a positive-then-negative cycle of CmdClk. The number of identifier bits is the same for any command port operation, but the number of data bits may vary depending on the command. The identifier and data bits are transferred with the most significant bit (MSB) first. For example, assuming 8 bits are used for the identifier, then command identifier 0x3E would be transferred in the order 00111110.

For read operations, the Network Controller initially drives the CmdData line (while CmdRW is low), and then the Physical Radio drives this line to communicate the data bits (while CmdRW is high).

The sequencing for write operations are identical except that the CmdRW line remains low for the duration of the operation (i.e., CmdRW line would match the CmdEn line below), and the driver for the CmdData line is the Network Controller throughout the operation.

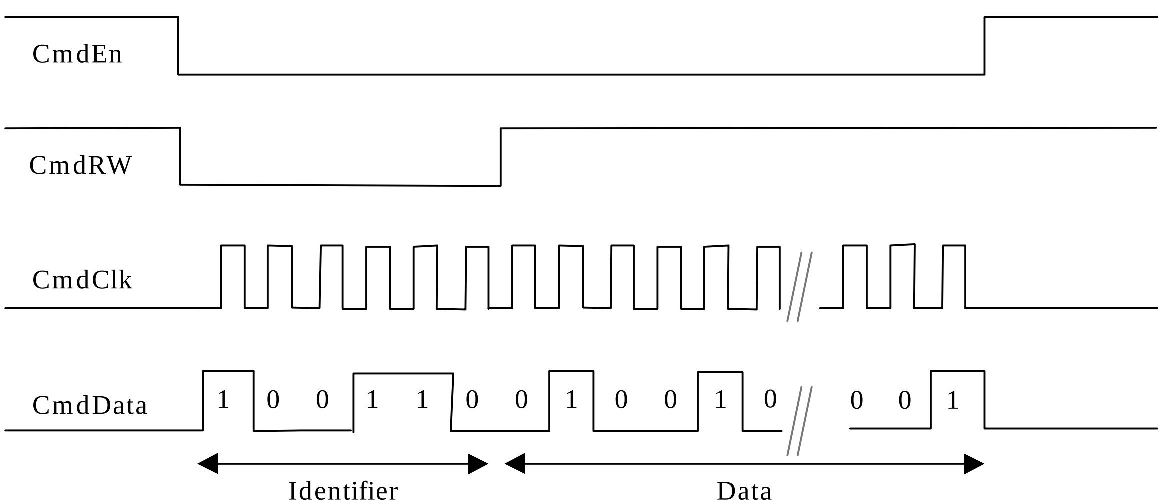

Figure 5: Typical Serial Command Port Sequence for Daughter Board Implementation

4 The implementations in this section are also all based on working implementations used by Rooftop Communications to communicate with different physical radios.

5 SAMTEC, Inc., New Albany, Indiana, 812/944-6733.

Description	Pin	Direction[dir]	Notes *[suggestions]*
Gnd	1		
Stat-1	2	¬	*[PhyRadSigCarrier; rcv signal detect]*
CmdSel0	3	®	Optional chip select 0 for cmds
CmdSel1	4	®	Optional chip select 1 for cmds
CmdRW	5	®	Selects cmd read or write
Gnd	6		
TxClk	7	¬	
CmdEn	8	®	Enables cmd read/write
CmdData	9	«	Cmd serial data circuit
CmdClk	10	®	Cmd serial data clock
+5V	11		
Gnd	12		
RD	13	¬	
RxClk	14	¬	
Gnd	15		
DTR	16	®	
RTS	17	®	
Cntl-3	18	®	*[PhyRadCmdReset; Active low radio reset]*
CTS	19	¬	
Gnd	20		
Gnd	21		
Stat-2	22	¬	*[PhyRadSigAntSel; Radio's antenna selection]*
Cntl-4	23	®	*[PhyRadVarPowerMode; rcvr power enable]*
Cntl-5	24	®	*[PhyRadVarBit/Code/SymbolRate; Band{width} select 0]*
Cntl-6	25	®	*[PhyRadVarBit/Code/SymbolRate; Band{width} select 1]*
Gnd	26		
TD	27	®	
Cntl-7	28	®	*[PhyRadCmdDropCapture; Drop radio rcv-capture]*
CD	29	¬	
Cntl-8	30	®	*[PhyRadVarScrambleMode; Scramble enable]*
+5V	31		
Cntl-9	32	®	*[PhyRadVarPowerMode; Transmitter pwr enable]*
Stat-RSSI	33	¬	The only analog signal
+5V	34		
Cntl-10	35	®	*[PhyRadVarPowerMode; Standby power]*
Cntl-11	36	®	
TxEnableOut	37	®	Tied to pin 39 to enable Tx[6]
Cntl-12	38	®	*[PhyRadVarTestMode; Loopback test]*
TxEnableIn	39	¬	Tied to pin 37 to enable Tx
Gnd	40		

[dir] "¬ " indicates a circuit generated by the Physical Radio, and "® " indicates one generated by the Network Controller.

[6] Optionally used by radio to gate transmitter. With these pins are tied together on the network controller board (the normal state), then the radio can gate it's "transmitter enable" signal by running it out onto pin 39, in from pin 37, and then over to the transmitter.

6.2 RS-530/RS-422 Implementation

In this implementation, a RS-530/RS-422 differential, synchronous serial interface cable is used to connect the network controller with the physical radio. The standard RS-422 circuits are used for the Data Port (shaded). The Command Port primitives are communicated using asynchronous serial, “Hayes modem style” AT commands over the auxiliary data pins 18 and 25 (shaded).

Circuit Description	Controller DB-25-M pin	Direction[dir]	Radio DB-25-F pin
Protective Gnd	1		1
TD (A)	2	®	2
RD (A)	3	¬	3
RTS (A)	4	®	4
CTS (A)	5	¬	5
DSR (A)	6	¬	6
Signal Gnd	7		7
CD (A)	8	¬	8
RxClk (B)	9	¬	9
CD (B)	10	¬	10
TxClk2 (B)[unused]	11	®	11
TxClk (B)	12	¬	12
CTS (B)	13	¬	13
TD (B)	14	®	14
TxClk (A)	15	¬	15
RD (B)	16	¬	16
RxClk (A)	17	¬	17
AuxTD	18	→	18
RTS (B)	19	®	19
DTR (A)	20	®	20
Open	21		21
DSR (B)	22	¬	22
DTR (B)	23	®	23
TxClk2 (A)[unused]	24	®	24
AuxRD	25	←	25

[dir] “¬ ” indicates a circuit generated by the Physical Radio, and “® ” indicates one generated by the Network Controller.
[unused] Unused circuits.

7. Acknowledgments

The development of this document was supported by the Defense Advanced Research Projects Agency (DARPA) through the Global Mobile (GloMo) program's Wireless Internet Gateways (WINGS) project (contract no. DAAB07-95-C-D157). WINGS is a collaborative effort by the University of California, Santa Cruz and Rooftop Communications.