



TAP Interface Specifications

This Document is for those who want to develop their own paging control software or add an interface for the **WaveWare v9 Series Paging Encoder** to their existing software applications.

A simple definition of the **TAP Protocol** is that the TAP Protocol normally requires a connect and disconnect process and normally assumes that the Host system maintains a database of pager reference numbers called IDs, and that the paging system maintains a database of all paging parameters associated with each ID. The **TAP Interface** supports paging messages up to 244 characters in length. The actual maximum length of transmitted messages in the **WaveWare SPS-5 v9 Series TAP Interface** is 245 minus the ID field length. With an ID value of 5, you can transmit up to 245 characters per message. With an ID value of 1233425120, you can transmit up to 235 characters per message.

To configure your **WaveWare v9 Series Paging Encoder** to use the TAP Paging Protocol, you may be required to configure the com port settings in the paging encoder. Please refer to **Appendix B – Com Port Settings**, for details on configuring communication protocols.

Your **WaveWare v9 Series Paging System** typically communicates with a PC or other host device via RS-232 at 9600 Baud, 8 data bits and 1 stop bit. The eighth data bit is ignored (no parity). You can configure the paging system for other serial communication parameters. Please refer to **Appendix B – Com Port Settings**, for details on serial communication parameters.

The **WaveWare v9 Series Paging Encoder** maintains an input buffer which can receive commands from the PC while a page is being transmitted. The input buffer should be able to contain approximately ten paging messages before getting full. When a command is received from the PC, the paging system responds with a message that includes error messages if the command was not understood or not properly transmitted. The first three digits of each paging system response conform to the response codes defined in the **TAP v1.8** specification.

See **Appendix D – TAP Response Codes** for a listing of the response codes.

The **WaveWare v9 Series Paging Encoder** encodes paging messages into **POCSAG** paging format and transmits the encoded paging message. If the **Carrier Detect** function is enabled, transmissions will be delayed while interfering signals are detected.

Control characters recognized by the **WaveWare v9 Paging System** in **TAP Protocol** mode include:

CARRIAGE RETURN	<CR>	\$0D
START OF TEXT	<STX>	\$02
END OF TEXT	<ETX>	\$03
END OF TRANSMISSION	<EOT>	\$04
SUBSTITUTE	<SUB>	\$1A
ESCAPE	<ESC>	\$1B

Control characters generated by the **WaveWare v9 Paging System** in **TAP Protocol** mode include:

LINE FEED	<LF>	\$0A
CARRIAGE RETURN	<CR>	\$0D
ACKNOWLEDGE	<ACK>	\$06
NEGATIVE ACKNOWLEDGE	<NAK>	\$15
ABANDON TRANSACTION	<RS>	\$1E
ESCAPE	<ESC>	\$1B
END OF TRANSMISSION	<EOT>	\$04
XON	<XON>	\$13
XOFF	<XOFF>	\$11

Tap Interface Specifications Continued on next page...

The **Operational Modes** available for paging using the **TAP Paging Protocol** include:

- System Identification Command
- Paging Session Login
- Paging Operation
- Paging Session Logout

SYSTEM IDENTIFICATION COMMAND

The System Identification command allows installation programs and other software applications to poll serial ports for the existence of a **WaveWare v9 Series Paging Encoder** using the Standard **ATI** Command. This can be used for a supervised process (also called keep-alive) where the host system periodically polls the paging system for a response. When the WaveWare paging transmitter recognizes a command formatted as **ATI<CR>**, the transmitter responds with the following message:

WaveWare Paging Encoder v9.00<CR>

(The firmware version number is subject to change)

PAGING SESSION LOGIN

The **v9 Paging Session Login** mode allows a **Host Device**, sometimes called a Remote Entry Device, to initiate communications with the paging system. With the **WaveWare v9 Series Paging Systems**, the Login process is optional. The **WaveWare v9 Series Paging System** will automatically login a Host Device and process the paging message if it recognizes a properly formatted TAP message block at any point in its operation.

The **Host Device** initiates the Login process by transmitting a carriage return **<CR>** character every two seconds until the paging system properly responds or until the Host Device times out and notifies the operator of a bad connection. The paging system will respond with **"ID=<CR>"** (Note: the quotation characters are used here only for reference and are not included in the transactions). The **"ID=<CR>"** will not be repeated or timed out by the paging system.

The **Host Device** should respond to the paging system with:

<ESC>PG1<CR> or **<ESC>pg1<CR>**

The **WaveWare v9 Series Paging System** will then respond with:

110 1.8<CR>WaveWare Paging Encoder v9.00<CR><ACK><CR>

The **"110 1.8"** message indicates that the system conforms to **TAP** specification **version 1.8** and should be backward compatible with earlier **TAP** implementations. The paging system then notifies the Host Device that it is ready to accept messages as follows:

<ESC>[p<CR>

This completes the Login process.

Tap Interface Specifications Continued on next page...

PAGING OPERATION

Paging transactions are transmitted in blocks of characters, where one transaction is sent per block. Each block sent by the Host Device is acknowledged by the paging system. The Host Device must wait for this acknowledgement before sending the next block. The **WaveWare v9 Series Paging Systems** provide acknowledgement in the form of an **<ACK>** character.

The **WaveWare v9 Series Paging Systems** supports **TAP** message blocks up to 253 characters in length, with a 235 to 244 character message, from 1 to 10 characters for ID (pager number), plus 5 control characters, and a 3 character checksum. (Message field length is restricted as follows: when a 1 character ID is used, a 244 character message can be accommodated, and so on, for a total of 253 characters, such that when a 4 character ID is used, a 241 character message can be accommodated. The maximum ID field length of 10 causes a maximum message field length of 235 characters). A block always carries two fields with their associated carriage returns. The message field may be empty, but its carriage return must be included in the data block. The **TAP** message block format is as follows:

<STX>ID<CR>Message<CR><ETX>Checksum<CR>

The **ID Field** is a one to ten character pager number. Leading zeroes are not required. The Message field accommodates up to 235 alphanumeric characters. The Checksum field provides a checksum for the previous portion of the **TAP** message block. See **Appendix C – TAP Checksum Calculation**, for sample code to create the checksum field.

The Message field can be formatted as follows:

- **For Alphanumeric** paging messages, all 7-bit ASCII “non-control” characters are valid except ^, ~, and _ (underscore)
- **For Numeric** paging messages, valid characters are the numbers 0 through 9, - (hyphen), and space
- **Control Characters** can be embedded into paging messages so that high end alpha pagers and paging data receivers can respond with formatted screen displays and with formatted serial output, as required. Control characters (typically **<CR>** and **<LF>**) can be embedded in messages by using the Transparency Option. The Transparency Option replaces a non-printable control character with the **SUB** (Hex 1A) character immediately followed by the control code offset by 40 Hex. See **Appendix F - Embedded Control Characters** for details on how to embed control characters in paging messages.

The **WaveWare v9 Series TAP Interface** supports ID field lengths from 1 to 10 digits. The ID field can be formatted using two different methods, as follows:

- **Method 1** - ID field lengths from 1 to 4 digits will cause a pager database lookup. ID fields of this format must contain numeric digits only. The numeric ID value will be compared to the ID values stored in the **WaveWare v9 Series Paging System's** onboard Pager Database.
- **Method 2** - ID field lengths from 5 to 10 digits will cause Extended ID processing. Extended ID processing assumes that the last 3 digits of the ID field define paging message encoding attributes, while the preceding digits define the pager cap code. Extended ID processing allows you to avoid the process of configuring a pager database in the **WaveWare v9 Series Paging Systems**. See **Appendix E – Extended ID Processing** for details on how to format the ID field.

When the **Host Device** delivers a **TAP** Message Block to the **WaveWare v9 Series Paging Systems**, if the message block is properly formatted, and a 1 to 4 digit ID field matches the pager database, the paging system will respond with an initial response code as follows:

211 Page(s) Sent Successfully<CR>

Tap Interface Specifications Continued on next page...

If your **WaveWare v9 Series Paging System** is configured to operate using **TAP Non-Verbose mode**, the paging system will respond only with an initial response code of:

211<CR>

Following the first part of the **TAP Message Block** response, if the input buffer of the paging system has capacity for another **TAP** message block, the paging system will send the following message.

<ACK><CR>

The **<ACK><CR>** response can be used as a flow control method to prevent overflow and lost messages. As alternative flow control methods, you can use either hardware or software flow control. If the paging system is configured for hardware flow control mode, the RS-232 **CTS** signal will be de-asserted when the input buffer doesn't have capacity for another **TAP Message Block**, and re-asserted when capacity exists in the input buffer. If the paging system is configured for software flow control mode, an **XOFF** Character (0x11) will be output in the response string to indicate the input buffer doesn't have capacity for another **TAP Message Block**. When capacity is available, an **XON** Character (0x13) will be output.

When the Host Device delivers a **TAP Message Block** and a checksum error occurs, the paging system will respond with an error message and a **<NAK><CR>**, which tells the Host Device to resend the transaction.

See Appendix D – TAP Response Codes for more information on paging system response messages. If the **TAP Message Block** checksum is OK, but the Message Block violates formatting rules, the paging system will respond with an error message and an **<RS><CR>**, which tells the Host Device to abandon the transaction. An example error message sequence follows:

514 Checksum Error – Exp. 2:9 Got 2:X<CR><NAK><CR>

PAGING SESSION LOGOUT

The Paging Session Logout mode allows a Host Device, sometimes called a Remote Entry Device, to stop communications with the paging system. With the **WaveWare v9 Series Paging System**, the Logout process is optional. The **WaveWare v9 Paging System** will automatically login a Host Device and process the paging message if it recognizes a properly formatted **TAP** message block at any point in its operation.

To initiate a logout (disconnect sequence) the **Host Device** should send the following to the paging system:

<EOT><CR>

Upon recognizing a logout command, the paging system will respond with:

PAGING EXCHANGE DISCONNECT<CR><ESC><EOT><CR>

This Completes the “Paging Operation”

This Completes the “TAP Interface Specifications”



TAP Checksum Calculation

Following is a sample Visual Basic™ code, that creates a string to be sent to the paging encoder. A subset of the string has a checksum calculation performed on it, and then the checksum and a control character are appended to form the string to be sent to the paging system. Refer to the TAP specification for more details on the checksum calculation.

'String that checksum is calculated on, of the form:

```
'<STX>ID<CR>Message<CR><ETX>
```

```
TXString$ = Chr(2) & txtID.Text & Chr(13) & txtMessage.Text & Chr(13) & Chr(3)
```

'Calculate and append checksum

```
j = 1
```

```
Sum = 0
```

```
Do Until j > Len(TXString$)
```

```
    B$ = Mid(TXString$, j, 1)
```

```
    D = Asc(B$)
```

```
    Sum = Sum + D
```

```
    j = j + 1
```

```
Loop
```

'Create the three characters to be transmitted to represent this checksum.

```
d3 = 48 + Sum - Int(Sum / 16) * 16
```

```
Sum = Int(Sum / 16)
```

```
d2 = 48 + Sum - Int(Sum / 16) * 16
```

```
Sum = Int(Sum / 16)
```

```
d1 = 48 + Sum - Int(Sum / 16) * 16
```

```
Check1$ = Chr$(d1)
```

```
Check2$ = Chr$(d2)
```

```
Check3$ = Chr$(d3)
```

```
Checksum$ = Check1$ & Check2$ & Check3$
```

'Create complete string to be sent to paging system, of the form:

```
'<STX>ID<CR>Message<CR><ETX>Checksum<CR>
```

```
TXString$ = TXString$ & CheckSum$ & Chr(13)
```

This Completes the "Tap Checksum Calculations"



TAP Response Codes

Following is a table of response codes supported by the **WaveWare v9 Series Paging Encoder**.

Please note that a **<NAK>** response typically means the message should be resent by the host, and an **<RS>** response typically means the message should be abandoned by the host.

<i>Response Code</i>	Definition
110 1.8	Paging system supports TAP Specification v1.8
211 Page(s) Sent Successfully	Paging message(s) successfully delivered. This message is accompanied by an <ACK>
510 Invalid character in pager ID	Pager ID field contains illegal characters. This message is accompanied by an <RS>
511 Invalid pager ID - not in database	Pager ID was legal and 1 to 4 characters in length, but didn't match database lookup. This message is accompanied by an <RS>
<i>512 Temporarily cannot deliver. Queue full. Try later</i>	Paging System input buffer is full. This message is accompanied by an <RS>
<i>514 Checksum Error. Exp XXX; Got: YYY</i>	Checksum didn't match message block. This message is accompanied by a <NAK>

Note: **<ACK>**, **<NAK>**, and **<RS>** are only representations of the lower order **ASCII** control characters output by the paging system.

<ACK> represents **Hex 06**, **<NAK>** represents **Hex 15**, and **<RS>** represents **Hex 1E**.

This Completes the "Tap Response Codes"



Embedded Control Characters

When using “WaveWare mode” it is not necessary to make use of the **<SUB>** control character. Since WaveWare mode does not use the **<CR>** control character it is only necessary to hit carriage return on your keyboard to have it show up on the pager display.

To embed a control character in a **TAP** mode paging message however, it is necessary include a **<SUB>** control character followed by an offset version of the control character you want to embed. You offset the control character by adding **40 HEX** to the control character, to make the character printable. Adding **40 HEX** to Carriage Return gives you **M**. Adding **40 HEX** to Line Feed gives you **J**. The character combination of **<SUB>M** causes a Carriage Return control character to be embedded in the encoded paging message, while **<SUB>J** causes a Line Feed control character to be embedded. The **<SUB>** character needs to be passed to the **WaveWare v9 Series Paging Encoder** as a **HEX** character.

In the **WaveWare v9 Series Paging Encoder Setup** software, you can embed a Carriage Return by entering **<CR>** in the message body, and embed a Line Feed character by entering **<LF>** in the message body. The use of the **<SUB>** character will be done by the software.

In the **Microsoft Visual Basic** programming language, you can use the following code example to define the embedded control character string to be delivered through the serial port to the paging system. In this example, we are embedding only the **Carriage Return** control character:

```
DIM CarriageReturnString As String,  
DIM MessageLine1 As String  
DIM MessageLine2 As String  
DIM TXString As String  
    CarriageReturnString = Chr(26) & "M"  
    LineFeedString = Chr(26) & "J"  
MessageLine1 = "Line 1 of test message"  
MessageLine2 = "Line 2 of test message"  
TXString = MessageLine1 & CarriageReturnString & MessageLine2  
    form1.MSComm1.Output = TXString
```

This completes the “Embedded Control Characters”



Extended ID Processing

The **WaveWare v9 Series TAP Interface** and **COMP2** interfaces support ID field lengths from 1 to 10 digits.

The ID field can be formatted as follows:

- **Standard Method** - ID field lengths from 1 to 4 digits will cause a pager database lookup. ID fields of this format must contain numeric digits only. The numeric ID value will be compared to the ID values stored in the **WaveWare v9 Series Paging Encoder's** onboard pager database.
- **Extended ID Method** - ID field lengths from 5 to 10 digits will cause Extended ID processing. Extended ID processing assumes that the last 3 digits of the ID field define paging message encoding attributes, while the preceding digits define the pager cap code. Extended ID processing allows you to avoid the process of configuring a pager database in the **WaveWare v9 Series Paging Encoder**.

Extended ID fields should be of the form:

CCCCCCERF, where **C** represents the Cap Code, **E** represents Message Encoding Type, **R** represents RF Data Rate, and **F** represents Function Code.

The **Extended ID field** can be represented in either numeric or alphanumeric format. All Extended ID fields must be between 5 and 10 digits in length to cause the paging system to perform Extended ID processing. Incorrect formatting of extended ID fields causes a **<NAK>** response combined with an explanation message.

CCCCCC is a 2 to 7 digit numeric value representing a pager cap code. Leading zeroes are optional except in the case of cap codes 8 and 9, which should be represented with at least one leading zero, e.g. 08 or 09.

E is the message encoding type, where "0" (zero), "n", or "N" represent "numeric encoding", and "1", "a", or "A" represent "alphanumeric encoding". The numeric values for **E** are provided to support numeric-only ID fields in paging control software.

R is the RF data rate, where "5" represents 512 bps, "1" represents 1200 bps, and "2" represents 2400 bps.

F defines the function code to be delivered, where "1" is function code 1, "2" is function code 2, "3" is function code 3, and "4" is function code 4. A function code of "0" is also valid. A function code of "0" causes the default function code for a particular Message Encoding Type ("**E**") to be substituted. The default value for **F** is "4" when the **E** value is "alphanumeric encoding", and "1" when the **E** value is "numeric encoding".

Following are valid example Extended ID fields causing Extended ID processing:

08A20 – Causes delivery of a standard (function code 4) alphanumeric message to cap code 0000008 at 2400 bps

08A24 – Same result as above

0008124 – Same result as above

000008124 – Same result as above

11A53 – Causes delivery of an alphanumeric message with function code 3 to cap code 0000011 at 512 bps

0000011A53 – Same result as above

11N53 – Same result as above, but with numeric encoding

038N52 – Causes delivery of a numeric message with function code 2 to cap code 0000038 at 512 bps



COMP1 Interface Specifications

This Document is for those who want to attach the **WaveWare v9 Series Paging Encoders** to existing systems that output raw data or simple data strings that are normally used in printing status logs on serial printers. The **COMP1 Protocol** is designed to emulate one of the operational modes of the Motorola People Finder™ paging system.

A simple definition of the **COMP1 Protocol** is that the **COMP1 Protocol** allows raw **ASCII Data** to be sent to all pagers in the Group one list of the **WaveWare v9 Series Paging Encoder's** Group Database. The group lists must be defined in **TAP** mode using the **WaveWare v9 Paging Encoder Setup** software. The **COMP1** Interface supports paging messages up to 255 characters in length, contact monitoring, and timed messaging.

To configure your **WaveWare v9 Series Paging Encoder** to use the **COMP1** paging protocol, you may be required to configure the com port settings in the paging encoder. Please refer to **Appendix B – Com Port Settings**, for details on configuring communication protocols.

Your **WaveWare v9 Series Paging Encoder** will typically communicate with a PC or other host device via RS-232 at 9600 Baud, 8 data bits and 1 stop bit. The eighth data bit is ignored (no parity). You can configure the paging system for other serial communication parameters.

The paging encoder maintains an input buffer, which can receive commands from the PC while paging messages are being transmitted. The paging encoder encodes paging messages into **POCSAG** paging format and transmits the encoded paging message through the internal radio transmitter. If the Carrier Detect function is enabled, transmissions will be delayed while interfering signals are detected.

Control characters recognized by the paging encoder in **COMP1 Protocol** mode include:

CARRIAGE RETURN	<CR>	\$0D
LINEFEED	<LF>	\$0A

Control characters generated by the **WaveWare Paging Encoder** in **COMP1 Protocol** mode when software flow control mode is active include:

XON	<XON>	\$13
XOFF	<XOFF>	\$11

In **COMP1 Interface Protocol**, the **WaveWare v9 Series Paging Encoder** does not provide feedback response signals on the serial port, other than flow control signals, when in Non-Verbose mode. **COMP1** is defined as a one way (simplex) protocol. The Verbose mode available with the **WaveWare v9 Series Paging Encoders** provide a serial port response, to allow software development and troubleshooting using a terminal software program.

The **COMP1** operation is as follows:

When incoming data is detected on the serial port, one of three things will happen:

- 1) A carriage return character is detected in the **Data stream**
- 2) A time period of approximately 10 seconds elapses after receipt of the last character on the serial port
- 3) 256 characters or more accumulate in the serial port input buffer. If any of these three events occur, the paging system will strip any carriage return and linefeed characters and transmit the remainder of the **Data stream** to all pagers defined in the encoder's pager database.

If the serial port input buffer becomes filled with greater than 1000 characters, the paging system will output either a hardware or software flow control response, depending upon how Hardware/Software Flow Control is configured. Once the serial port input buffer drops below 450 characters, the paging system will use flow control signals to indicate that it is no longer busy and that additional data can be delivered to the serial port.

NOTE: When using **COMP1** mode, please note that all data is transmitted to the pagers listed in group one of the **WaveWare v9 Series Paging Encoder's** pager database. When setting up for **COMP1** operation, you should limit the pager quantity in the pager database to no more than a few pagers, in order to avoid excessive paging transmissions.

This Completes the "COMP1 Interface Specifications"



COMP2 Interface Specifications

This Document is for those who want to attach the **WaveWare v9 Series Paging Encoders** to existing systems that output simplified paging system control commands. The **COMP2** protocol is designed to emulate one of the operational modes of the Motorola People Finder™ paging system. There are two possible response modes for **COMP2**, the Verbose and Non-Verbose modes. The Verbose response mode has been designed to have human readable responses to message inputs and certain errors. The Non-Verbose response mode has no readable responses in it.

A simple definition of the **COMP2 Protocol** is that the **COMP2 Protocol** allows a message formatted as **PagerID<CR>Message<CR>** to cause the Message to be delivered to the associated **PagerID** defined in the **WaveWare v9 Series Paging Encoder's** pager database. The **COMP2** interface supports paging messages up to 253 characters in length, group paging, contact monitoring, and timed messaging. The total length of the control string, including **<CR>** and **<LF>** characters cannot exceed 256 characters. The **PagerID** field can be from 1 to 10 characters in length. **PagerID** field lengths from 5 to 10 digits are assumed to be Extended Pager IDs (See **Appendix E – Extended Pager ID Processing**). Carriage Return is the normal field delimiter in **COMP2 Protocol**, but the tilde (~) character is also supported as a delimiter.

A **COMP2 Protocol superset** is also available; where a message formatted as **PagerIDX<CR>Message<CR>** will cause the message to be delivered to the **PagerID** using Function Code value of X. Values of X can include A, B, C, or D. An example of a function code controlled COMP2 message using Function Code A and Pager ID 101 is **101A<CR>Test Message<CR>**

To configure your **WaveWare v9 Series Paging Encoder** to use the **COMP2 Paging Protocol**, you may be required to configure the Com Port of the paging encoder. Please refer to **Appendix B – Com Port Settings**, for details on configuring communication protocols.

Your **WaveWare v9 Series Paging Encoder** will typically communicate with a PC or other host device via RS-232 at 9600 Baud, 8 data bits and 1 stop bit. The eighth data bit is ignored (no parity). You can configure the paging system for other serial communication parameters. Please refer to **Appendix B – Com Port Settings**, for details on serial communication parameters.

The paging encoder maintains an input buffer, which can receive commands from the PC while paging messages are being transmitted. The paging encoder encodes paging messages into **POCSAG** paging format and transmits the encoded paging message through the attached radio transmitter. If the Carrier Detect function is enabled, transmissions will be delayed while interfering signals are detected.

Control characters recognized by the **WaveWare v9 Series Paging Encoders** in **COMP2 Protocol** mode include:

CARRIAGE RETURN	<CR>	\$0D
LINEFEED	<LF>	\$0A
TILDE	~	\$7E
DELETE		\$7F or \$FF

Control characters generated by the **WaveWare v9 Series Paging Encoders** in **COMP2 Protocol** mode when software flow control mode is active include:

CARRIAGE RETURN	<CR>	\$0D
XON	<XON>	\$13
XOFF	<XOFF>	\$11
ABANDON TRANSACTION	<RS>	\$1E

The **WaveWare v9 Series Paging Encoders**, when using **COMP2 Protocol** mode, are controlled using control strings formatted as follows:

Single Message Example using **Carriage Return** as delimiter:

PagerID<CR>Message<CR>

Single Message Example using **Tilde** as delimiter:

PagerID~Message~

Tone/Vibe Only Example:

PagerID<CR><CR>

Multiple Message Example:

PagerID1<CR>Message1<CR>PagerID2<CR>Message2<CR>

Forced Resynchronization Example:

PagerID<CR>Message<CR>

Multiple Messages using Forced Resynchronization Example:

PagerID<CR>Message 1<CR>PagerID<CR>Message 2<CR>

Function Code Control Example using Function Code B:

PagerIDB<CR>Message<CR>

When incoming data is detected on the serial port, **one of three things will happen:**

- 1)** The data in the serial port input buffer contains properly formatted **COMP2** command strings.
- 2)** A time period of approximately 10 seconds elapses after receipt of the last character on the serial port.
- 3)** 256 characters or more accumulate in the serial port input buffer. The serial port data is analyzed character by character to see if a properly formatted command string exists. If the serial port data contains one or more properly formatted **COMP2** command strings, then the paging message(s) will be transmitted to the specified Pager ID(s) as long as those Pager ID(s) exist in the **WaveWare v9 Paging Encoder's** Pager Database. If a **Pager ID** doesn't exist in the pager database, an error message will be output in verbose mode by the paging encoder as follows:

Invalid character in pager ID<CR><CR><RS><CR>

Any portions of the serial port input buffer that don't meet the **COMP2** message formatting requirements are discarded. The original **COMP2** Protocol has an inherent weakness of using the **<CR>** character to delimit both the **Pager ID** Field and the Message Field. This approach allows the possibility of an extra or missing **<CR>** character to cause the host system and the paging system to get out of synch with each other, causing multiple paging messages to be rejected by the paging system. A more robust version of the **COMP2** protocol uses Three **** characters as indicators of the beginning of a **COMP2** command string. The use of three **** characters at the beginning of each output message assures that out-of-synch conditions will not occur. The forced resynchronization method is optional. As additional synchronization insurance, the **WaveWare COMP2** implementation has been designed to stay in synch even if one or more **<CR>** characters immediately precede a properly formatted **COMP2** command string.

If the **10 Second Timeout** occurs while waiting for all of the characters required for a properly formatted **COMP2** message to arrive on the serial port, the data in the serial port input buffer will be discarded. If 256 or more characters arrive in the paging encoder's serial port input buffer, and the first 256 characters contain all of the elements of a properly formatted **COMP2** Command String except for the trailing **<CR>** character, then the **<CR>** will be assumed and only the first 256 characters of the input buffer will be processed as a valid **COMP2** paging message.

If the serial port input buffer becomes filled with greater than **600** characters, the paging encoder will output either a hardware or software flow control response, depending upon how the system is configured. Once the serial port input buffer drops below **450** characters, the paging encoder will use flow control signals to indicate that it is no longer busy and that additional data can be delivered to the serial port.

This Completes the "COMP2 Interface Specifications"



Scope Interface Specifications

This Document is for those who want to develop their own paging control software or add an interface for the **WaveWare v9 Series Paging Encoder** to their existing software applications. You can choose to use the **Scope Paging Protocol** defined in this appendix, the **TAP Paging Protocol**, defined in **Appendix A – TAP Interface Specifications, COMP1**, or **COMP2**, or **WaveWare Proprietary Paging Protocols** defined elsewhere in this manual.

A simple definition of the difference between the **Scope and TAP Protocols** is that the **Scope Protocol** is a dedicated connection that doesn't require a login process and it assumes that the **Host Device**, e.g. nurse call system, maintains a database of all paging parameters associated with a pager, including Cap Code, Encoding Method, and RF Data Rate. The **TAP Protocol** requires a connect and disconnect process and assumes that the **Host Device** only needs to maintain a database of pager reference numbers called **PIN's**, and that the paging system maintains a database of all paging parameters associated with each **PIN**.

The Scope Interface supports paging messages up to 512 characters in length while the **TAP** interface supports paging messages up to 256 characters in length, group paging, and contact monitoring, and timed messaging. Each protocol has its advantages and disadvantages, but both accomplish the process of causing **POCSAG** encoded paging messages to be delivered to local area pagers.

To configure your **WaveWare v9 Series Paging Encoders** to use the **Scope Paging Protocol**, you may be required to configure the Com Port on the encoder board using the **v9 Encoder Setup Software**. Please refer to **Appendix B – Com Port Settings**, for details on configuring communication protocols.

Your **WaveWare v9 Series Paging Systems** will typically communicate with a **Host Device** via RS-232 at 9600 Baud, 8 data bits and 1 stop bit. The eighth data bit is ignored (no parity). You can configure the paging encoder for other serial communication parameters. Please refer to **Appendix B – Com Port Settings**, for details on serial communication parameters.

The paging system maintains an input buffer which can receive commands from the **Host Device** while a page is being transmitted. The input buffer should be able to contain approximately ten paging messages before getting full. When a command is received from the **Host Device**, the paging encoder responds with a message that includes error messages if the command was not understood or not properly transmitted. The normal response will echo the paging request command along with a status message.

The paging system can use hardware handshaking (**CTS**) to control the flow of paging message commands from the **Host Device**. Alternatively, the paging control software could use the response messages as a means of handshaking between the paging encoder and the paging control software.

The paging encoder encodes paging messages into **POCSAG** paging format and passes the encoded paging message to the attached radio transmitter. If the **Carrier Detect** function is enabled, transmissions will be delayed while interfering signals are detected.

PAGING MESSAGE COMMAND

The **Scope Protocol Paging Message** command is formatted as follows:

AN1234567DMessage<CR>

A Alphanumeric character representing the type of paging message to be transmitted

A = Alphanumeric

N = Numeric

N Alphanumeric character representing the data rate, in bits per second, at which the paging message is to be transmitted

N = 512 bps

F = 1200 bps

S = 2400 bps

1234567 7 digits representing the pager cap code, including leading zeroes as required

D Alphanumeric character representing the function code value to be delivered with the message

A = One Beep

B = Two Beeps

C = Three Beeps

D = Four Beeps

Example 1:

AN0046180DThis is a test<CR>

Will send the message "**This is a test**" at 512 bps to an alphanumeric pager with Cap Code 0046180.

Note: The POCSAG alphanumeric character set is the entire ASCII 7 bit character set.

Example 2:

NN0765155B412-3433<CR>

Will send the message "**412-3433**" at 512 bps to a numeric pager with Cap Code 0765155.

Note: The POCSAG numeric character set allows Hyphen (-), space, left bracket ([), and right bracket (]) characters in addition to the normal numeric character set.

Example 3:

AF0145678C<CR>

Will send a three beep tone/vibe only message at 1200 bps to a tone/vibe only, a numeric, or an alphanumeric pager with Cap Code 0145678 and multiple tone/vibe cadence capability. Certain paging receiver models may not be designed to respond to multiple tone/vibe cadence messages.

Example 4:

AS0145678B<CR>

Will send a two beep tone/vibe only message at 2400 bps to a paging receiver with Cap Code 0145678.

RESPONSE TO ALL COMMANDS

The **v9 Series Paging Systems** will respond immediately to Paging Message commands with a status message in one of the following formats:

Page Queued Okay<CR><LF>

The **Page Queued Okay** response is sent by the paging encoder to the host system immediately following a command if the command was understood by the paging encoder. Following are some example error messages output with Scope protocol.

Error – Invalid Pager ID<CR><LF>

Error – Invalid pager type. Expected A or N<CR><NAK><CR><LF>

Error – Beep code invalid. Expected A,B,C,D<CR><NAK><CR><LF>

<NAK> (Hex code 15) is used to indicate non-acknowledgment of a command from the host system.

This Completes the “Scope Interface Specifications”



WaveWare Interface Specifications

This Document is for those who want to develop their own paging control software or add an interface for the **WaveWare v9 Series Paging Encoder** to their existing software applications. You can choose to use the **WaveWare Paging Protocol** defined in this appendix, the **TAP Paging Protocol**, defined in **Appendix A – TAP Interface Specifications**, **COMP1**, or **COMP2**, or **Scope Paging Protocols** defined elsewhere in this manual.

A simple definition of the difference between the **WaveWare and TAP Protocols** is that the **WaveWare Protocol** is a dedicated connection that doesn't require a login process and it assumes that the **Host Device** maintains a database of all paging parameters associated with a pager, including Cap Code, Encoding Method, and RF Data Rate. The **TAP Protocol** requires a connect and disconnect process and assumes that the **Host Device** only needs to maintain a database of pager reference numbers called **PIN's**, and that the paging system maintains a database of all paging parameters associated with each **PIN**.

The WaveWare Interface supports paging messages up to 512 characters in length while the **TAP** interface supports paging messages up to 256 characters in length, group paging, and contact monitoring, and timed messaging. Each protocol has its advantages and disadvantages, but both accomplish the process of causing **POCSAG** encoded paging messages to be delivered to local area pagers.

To configure your **WaveWare v9 Series Paging Encoder** to use the WaveWare paging protocol, you may be required to configure the Com Port on the encoder board using the **v9 Encoder Setup Software**. Please refer to **Appendix B – Com Port Settings**, for details on configuring communication protocols.

Your **WaveWare v9 Series Paging Systems** will typically communicate with a **Host Device** via RS-232 at 9600 Baud, 8 data bits and 1 stop bit. The eighth data bit is ignored (no parity). You can configure the paging encoder for other serial communication parameters. Please refer to **Appendix B – Com Port Settings**, for details on serial communication parameters.

The paging system maintains an input buffer which can receive commands from the **Host Device** while a page is being transmitted. The input buffer should be able to contain approximately ten paging messages before getting full. When a command is received from the **Host Device**, the paging encoder responds with a message that includes error messages if the command was not understood or not properly transmitted. The normal response will echo the paging request command along with a status message.

The paging system can use hardware handshaking (**CTS**) to control the flow of paging message commands from the **Host Device**. Alternatively, the paging control software could use the response messages as a means of handshaking between the paging encoder and the paging control software.

The paging encoder encodes paging messages into **POCSAG** paging format and passes the encoded paging message to the attached radio transmitter. If the **Carrier Detect** function is enabled, transmissions will be delayed while interfering signals are detected.

The Commands available for serial port controlled paging using the **WaveWare Paging Protocol** include:

- System Identification
- Paging Message
- Setup
- Status

PAGING MESSAGE COMMAND

The **Paging Message** command is formatted as follows:

Example 1:

<SOH>TBCC...C<STX>XXXX...XXX<ETX><EOT>

Example 2:

<SOH>T,B,CC...C,<STX>XXXX...XXX<ETX><EOT><CR><LF>

Note: The commas, the <CR> character, and the <LF> character are optional in the **Paging Message** command. They are used for appearance purposes to separate subparts of the command string and for formatting the echoed output while troubleshooting the system.

<SOH> (Hex code 01) is used to mark the start of a message. This character resets the paging system and can be used to abort an incomplete paging command at any point prior to transmission.

T Alphanumeric character representing the type of paging message to be transmitted

A = Alphanumeric

N = Numeric

1 = One Beep (alphanumeric pagers)

2 = Two Beeps (alphanumeric pagers)

3 = Three Beeps (alphanumeric pagers)

4 = Four Beeps (alphanumeric pagers)

B Numeric character representing the data rate, in bits per second, at which the paging message is to be transmitted

5 = 512 bps

1 = 1200 bps

2 = 2400 bps

Note: This numeric character may have one or no leading commas separating it from the paging message type.

C...C One to seven decimal numeric digits representing the cap code of the paging receiver to which the message will be transmitted. Cap Codes are seven digit strings. Cap Codes that begin with leading zeroes do not require the leading zeroes to be included in this numeric string, if you use comma delimiters. If comma delimiters are not used, you must include leading zeroes.

Note: This numeric string may have one or no leading commas separating it from the data rate.

<STX> (Hex code 02) is used to mark the beginning of a string of message characters to be transmitted. For paging message types of 1 to 4 Beeps (defined by the 'T' character above), this character and all following characters are ignored by the paging transmitter, until another **<SOH>** is encountered.

This character may have one or no leading commas separating it from the cap code.

X...X A string of 0 to 256 alphanumeric message characters to be transmitted. For paging message types of 1 to 4 Beeps, this character and all following characters are ignored by the paging system, until another **<SOH>** is encountered.

<ETX> (Hex code 03) is used to mark the end of a string of message characters to be transmitted. For paging message types of 1 to 4 Beeps, this character and all following characters are ignored by the paging system, until another **<SOH>** is encountered.

<EOT> (Hex code 04) is used to mark the end of a Transmit Paging Message command string. All following characters are ignored by the paging system until a **<SOH>** character is encountered.

<CR><LF> (CR is Hex code 0D, LF is Hex code 0A)

These characters are ignored by the paging system unless contained within a message string and are used only for formatting output on a PC.

The **<CR>** and **<LF>** characters are optional.

Example 1:

<SOH>A,5,46180,<STX>This is a test<ETX><EOT><CR><LF>

Will send the message "This is a test" at 512 bps to an alphanumeric pager with cap code 0046180.

Note: The POCSAG alphanumeric character set is the entire ASCII 7 bit character set.

Example 2:

<SOH>N,5,0765155,<STX>412-3433<ETX><EOT><CR><LF>

Will send the message "412-3433" at 512 bps to a numeric pager with Cap Code 0765155.

Note: The POCSAG numeric character set allows Hyphen (-), space, left bracket ([), and right bracket (]) characters in addition to the normal numeric character set.

Example 3:

<SOH>N1765155<STX>412-3433<ETX><EOT>

Will send the message "412-3433" at 1200 bps to a numeric pager with cap code 0765155. Note the lack of commas and the implied leading zeroes on the cap code.

Example 4:

<SOH>3,1,145678<EOT>

Will send a three beep tone/vibe only message at 1200 bps to a tone/vibe only, a numeric, or an alphanumeric pager with cap code 0145678 and multiple tone/vibe cadence capability. Certain paging receiver models may not be designed to respond to multiple tone/vibe cadence messages.

Example 5:

<SOH>2,1,145678<STX>412-3433<ETX><EOT>

Will send a two beep tone/vibe only message at 1200 bps to a paging receiver with cap code 0145678. Note that the portion of the string after the cap code is ignored by the paging system because the paging message type was a paging message type of one to four beeps.

Example 6:

<SOH>450006123<EOT><CR><LF>

Will send a four beep tone only message at 512 bps to a paging receiver with cap code 0006123.

RESPONSE TO ALL COMMANDS

The **v9 Series Paging Systems** will respond immediately to all commands, including Paging Message commands, Setup Commands, and Status Commands, by echoing the Paging Message command back to the PC, and appending the echoed command with a status message in one of the following formats:

<SOH><ACK>V..V,QQ,D,C,II<EOT>

The **ACK** response is sent by the paging encoder to the host system immediately following a command if the command was understood by the paging encoder.

<SOH><NAK>EE<EOT>

The **NAK** response is sent by the paging encoder to the host system immediately following a command if the command was not understood by the paging encoder or if certain errors occur.

<SOH> (Hex code 01) is used to mark the start of a message.

<ACK> (Hex code 06) is used to indicate acknowledgement of valid reception of a command from the host system.

V..V Alphanumeric string that indicates the version number of the firmware in the paging encoder. The version number is followed by a comma character.

QQ A one to two numeric character string that indicates the quantity of paging messages that have been received by the paging encoder and not yet transmitted (Input Queue).

The **QQ** value is followed by a comma character. This parameter should be monitored if you intend to use software handshaking between your paging control software and the paging encoder. The paging encoder Input Queue overflows if the **QQ** value tries to exceed 14. As an example, with software handshaking, your software should stop sending data at a **QQ** value of 12 and resume sending data when the **QQ** value reaches 10 or less. You can query the size of the Input Queue by submitting a Status Command, described on the following pages.

D Numeric character that indicates whether the carrier detect circuit in the paging encoder is currently detecting a carrier signal or not. Valid values are:

0 = Carrier Signal Not Detected
1 = Carrier Signal Detected

Note: The carrier detect signal status is passed to the PC whether or not the paging encoder has been programmed to avoid transmitting when a carrier signal has been detected (see Programming Command below).

C A single numeric value used to indicate the **On/Off** state of the Carrier Detect mode. The numeric character is followed by a comma.

Valid values (**C**) are:

0 = Off
1 = On

II A two hexadecimal digit value that is reserved for future use.

<EOT> (Hex code 04) is used to mark the end of a message

<NAK> (Hex code 15) is used to indicate non-acknowledgment of a command from the host system.

<EE> Two numeric characters that indicate the particular error that caused no acknowledgment of a command from the host system.

The possible error values are:

01 = **Invalid Message Type.** Indicates a message type other than A, N, 1-4 was transmitted.

02 = **Invalid Data Rate.** Indicates a data rate other than 5, 1, or 2 was transmitted.

03 = **Invalid Cap Code.** Indicates an invalid cap code was transmitted. Invalid cap codes are: 0-7, 2007664-2007671, 2045056-2045063, 2097144+.

04 = **Input Buffer Overflow.** Indicates the serial data input buffer in the paging encoder has overflowed. This condition would likely be caused by the host system not recognizing the CTS signal.

05 = **Paging Buffer Overflow.** Indicates the encoded paging message output buffer in the paging encoder has overflowed. This condition would likely be caused by excessive carrier signal detection while the Carrier Detect mode was enabled.

06 = **Invalid Status/Setup Command.** Indicates that the command was recognized as a Status or Setup command but the entire command string was not fully interpreted by the paging encoder.

07 = **EEPROM Write Error.** Indicates an internal system error in the paging encoder with regard to the proper storage of configuration information. This condition would likely indicate that paging encoder repair is required.

08 = **Invalid Command Format.** Indicates that the command was not fully recognized as a valid paging encoder command.

Example 1:

```
<SOH>A,5,46180,<STX>This is a test<ETX><EOT><SOH><ACK>V8.01,00,0,1,1A<EOT>
```

Echoes the paging message command and acknowledges proper receipt of a command by appending the **ACK**, indicating paging encoder firmware version is 8.01, no paging messages in queue, no carrier signal is being detected, and **Carrier Detect** mode is set to “**ON**”. The last field does not currently provide useful information.

Example 2:

```
<SOH>4,5,0000006<EOT><SOH><NAK>03<EOT><CR><LF>
```

Echoes the paging message command and indicates an error in receiving a command due to an invalid paging receiver capcode in a Paging Message command. Includes a **<CR>** and **<LF>** at the end of the response string if they were included in the paging message command.

SETUP COMMAND

You may program the **WaveWare v9 Series Paging Encoders** to monitor for interference using the **Carrier Detect** mode. If the **Carrier Detect** mode is turned “**ON**”, the paging encoder will hold paging messages in queue until an offending carrier signal is no longer detected at the transmission frequency of the paging encoder. The settings programmed by the Setup command will be remembered by the paging encoder even if power is removed from the paging encoder.

The Setup command is formatted as follows:

```
<SOH>S,X,V<EOT>
```

Where the S indicates Setup Mode, X is the command type, and V is the value. The comma delimiters are optional.

X A single numeric value used to indicate the command type. The numeric character is followed by a comma.

Valid values (**X**) are:

1 = [Carrier Detect Mode](#)

V A single numeric value used to indicate the value of the command setting to be saved. The numeric character is followed by an **EOT** character.

Valid values (**V**) are:

0 = [Off](#)

1 = [On](#)

Example 1:

<SOH>S,1,0<EOT> will set the Carrier Detect function to the “**Off**” status.

Example 2:

<SOH>S11<EOT> will set the Carrier Detect function to the “**On**” status.

STATUS COMMAND

You may request the paging encoder to respond with its current programmed settings using the Status command. The Status command is formatted as follows:

<SOH>S?<EOT>

Where the **S?** Requests the paging encoder to respond with the status of its current programmed settings. The Status Command can be used in supervisory controlled applications to ensure the encoder is still functioning, without causing unwanted paging transmissions.

This Completes “Status Commands”

This Completes the “WaveWare Interface Specifications”