



## **UHF TagModem™ INSTRUCTIONS V 1.1**

The UHF TagModem enables communication between any RS-232 device and a UHF EPC RFID Reader. The current version of the TagModem uses EPC protocol Class 1 Generation 1, informally known as the “Alien Protocol” used by Alien Technology RFID readers. The TagModem sends and receives data in packets of 8 bytes. These 8 bytes are then transmitted by the tagmodem to the reader as if they were the EPC ID of the tagmodem. Packets sent to the tagmodem must have a one byte SOF (Start of Frame) and a one byte EOF (End of Frame).

The UHF TagModem receives packets from a host device through a simplified Unix style command prompt. The prompt is represented by the character '>'. Using the Unix analogy, each command string sent to the TagModem creates a “process.” In this case, only one kind of process exists: repeatedly transmitting an eight byte packet of data to an RFID reader. This process is considered terminated when the TagModem receives a “Quiet” Command from the RFID reader (i.e., this command serves as an acknowledgement that the TagModem’s packet was received). When a process terminates, the character ‘q’ is returned, followed by a new line and a new command prompt.

Also as in Unix, if a command string sent to the TagModem is terminated by an ampersand (&), it will continue to accept new commands while the current process is ongoing. If the ampersand is not sent, the TagModem will not accept any new commands until the current process has terminated (i.e., a new data packet cannot be sent to the TagModem until an acknowledgement was received for the previous data packet). Finally, a process can be immediately terminated by typing in CTRL-X.

The user should be aware of the operating mode of the RFID reader when using the TagModem, since certain operating modes do not send the “Quiet” command to the tag. In this case, the TagModem should be used without the ampersand character.

## OPERATING MODES

There are two modes in which the UHF TagModem can receive data from a host device: ASCII mode and binary mode. The mode is determined by the SOF in the packet of bytes sent from the host.

### ASCII Mode:

The SOF for ASCII mode is ASCII character 'E'. Each data byte should be transmitted to the tagmodem as two ASCII characters, one for each hexadecimal digit in the byte. For example the data byte 0x4e would be sent as the ASCII character for '4' (0x34) followed by the ASCII character for 'e' (0x65). The EOF is a carriage return (0x0D). The total number of bytes is 18. If an error is made while sending a packet to the tagmodem from a host device (e.g. an invalid character is sent, or a carriage return is received before the correct number of bytes have been sent), an 'X' is returned by the tagmodem to the host device, and the tagmodem ignores the partial command.

If a quiet command is received by the TagModem, a 'q' is returned (assuming the command was not terminated by an ampersand).

### Examples:

```
>E0123456789abcdef<CR>
```

```
q  
>
```

```
>Edeadbeefdeadbeef&<CR>
```

```
>
```

```
>E039ab4<CR>
```

```
X  
>
```

```
>E123G
```

```
X  
>
```

## Binary Mode:

The SOF for Binary mode is the hexadecimal character 0x02. Each byte is sent as is (so 0x4e is represented as 0x4e). The EOF is a carriage return (0x0D).

Errors are not as easily detected by the TagModem in binary mode as it is ASCII mode. This is because only alphanumeric characters '0'-'9' and 'a'-'f' are acceptable data bytes. In binary mode, any hexadecimal value between 0x00 and 0xFF can be used. In the next version of the firmware, a checksum will be added such that the integrity of the data packet received by the TagModem can be confirmed. However, in this version, the only way an error can be detected is if the command string does not either contain a <CR>, or a "&<CR>" in the last position of the string, then an X is returned. If an error occurs in communication (for example one of the data bytes is lost), currently there is no means for the host device to know this.

As in ASCII mode, upon successful transmission of a data packet, the TagModem returns the ASCII character 'q' (again assuming the command was not terminated by an ampersand).

In the following examples, each raw binary byte is represented as a hexadecimal number preceded by a lower case x. For example, the decimal integer value 80 would be represented as 0x50. Also note that <CR> and x0D are identical, as are '&' and 0x26.

### Example:

```
>x02x01x23x45x67x89xABxCDxEFx0D
```

```
q
```

```
>
```

```
>x02xDExADxBExEFxDExADxBExEFx26x0D
```

```
>
```

In the next version of the firmware, a checksum will be added.

For any further questions on the functionality, please contact Rich Redemske at Tagsense: [redemske@tagsense.com](mailto:redemske@tagsense.com)