Protocol

Modem-M64 serial protocol

This document describes the Water Linked Underwater Modem Link Layer protocol.

Terminology

- Modem - Unit for transmitting/receiving data
- Packet - Unit of data transmitted together
- ACK - Acknowledgement. The command issued was successful.
- NAK - Negative acknowledgement. The command issued failed.

Version

This document describes protocol/software version 1,0,x (major,minor,patch)

The protocol versioning follows semantic versioning in that:

- MAJOR version increments represents incompatible API changes,
- MINOR version increments represents added functionality in a backwards-compatible manner
- PATCH version increments represents backwards-compatible bug fixes

Overview

The serial communication format is 115200 8-N-1 (no hardware flow control).

Packets sent to and received from the modem start with a `w` and end with end with LF or CR+LF. The packet format is:

Start byte	Direction	Command	Options (0 to many)	Checksum	End byte
`w`	`c` or `r`	`x`	`,[option]`	`*xx`	`\n` or `\r\n`

Direction is command (`c`) for commands issued to the modem and the modem replies with direction set to response (`r`). The commands can be sent as a string or entered one char at a time from a terminal.

The protocol can support Water Linked modems with different payload sizes and extended features. To support any Water Linked modem the connection procedure is to:

- Get protocol version. Verify that the major version number is 1.
- Get payload size. Use this size when queuing packets for transmission the modem.

For Water Linked Modem-M64 the payload size is 8 bytes.

In order for two modems to communicate they must be configured to use different roles (A/B) on the same channel. The modem with role A will always transmit. The modem will role B will listen until it detects a signal from a modem of role A. It will then start transmitting responses back to the modem with role A. The modem with role B will go back to listen mode if several consecutive packets from the modem with role A cannot be decoded (ie signal is lost).

> ✏️ **Note**
>
> Payload where every byte is \0 is reserved. It is used to keep modems in sync if no data packet is queued by the user. This sync packet is filtered out by the receiver. Avoid this payload by compressing the data or otherwise ensuring atleast 1 bit is non-zero.

Checksum is optional when sending commands to the modem. The modem always returns a checksum. The checksum algorithm is CRC-8 and it is formatted as a hexadecimal number using 2 lower-case charaters (ex: `*c3`). See the checksum section of the manual for our DVL product (which uses the same CRC-8 algorithm) for further details and example code.

Commands

Commands in the table are shown **without** the checksum for readability.

Command	Description	Response	Description
`wcv`	Get protocol version	`wrv,` *[major], [minor],[patch]*	Protocol version. eg: `wrv,1,0,1`

Command	Description	Response	Description
`wcn`	Get supported payload size	`wrn`, *[size]*	Where size is supported payload size: eg: `wrn,8`
`wcc`	Get modem configuration	`wrc`, *[role]*, *[channel]*	Where role=a or b and channel=1-7 eg: `wrc,a,3`
`wcs`, *[role]*, *[channel]*	Set modem configuration - Where role=a or b and channel 1-7: eg: `wrs,b,3`	`wrs,a` or `wrs,n`	ACK or NAK
`wcl`	Get transmit queue length	`wrl`, *[q]*	Number of packets currently queued for transmission. eg: `wrl,107`
`wcf`	Flush transmit queue	`wrf,a` or `wrf,n`	ACK or NAK
`wcd`	Get diagnostic	`wrd`, *[link]*, *[packet_count]*, *[packet_loss_count]*, *[bit_error_rate]*	link=y if connection with other modem, otherwise n. eg: `wrd,y,1234, 17,3.5` *[1]*
`wcq`, *[size]*, *[payload]*	Queue packet for transmission. Number of characters in payload need to be exactly same as size. Payload can be binary. eg: `wcq,8,HelloSea` *[2]*	`wrq,a` or `wrq,n`	ACK or NAK

Command	Description	Response	Description
		`wrp, [size], [payload]`	Got packet from other modem eg: `wrp,8,Welcome!`
		`wr?`	Malformed request: Response when packet cannot be understood
		`wr!`	Malformed request: Packet does not match the given checksum

[1] `packet_count` is the number of packets received. If a packet cannot be decoded the `packet_loss_count` increments regardless of the packet being queued data or a sync packet from the other modem.

[2] Payload with just zeros (`\0`) is reseved for keeping the modems in sync if no packet is queued. It will be filtered out by the receiving modem.

Examples

Here is an example of setting up two modems and sending packets between them.

On top side modem (using role A):

Command	Response	Description
`wcv`	`wrv,1,0,1*44`	Get protocol version
`wcn`	`wrn,8*ba`	Get payload size

Command	Response	Description
`wcs,a,3`	`wrs,a*01`	Set role A and channel 3
`wcq,8,HelloSea`	`wrq,a*d7`	Send packet
		Wait packet from other modem
	`wrp,8,HelloTop*bb`	Got response

On AUV/ROV modem (using role B):

Command	Response	Description
`wcv`	`wrv,1,0,1*44`	Get protocol version
`wcn`	`wrn,8*ba`	Get payload size
`wcs,b,3`	`wrs,a*01`	Set role B and channel 3
		Wait packet from other modem
	`wrp,8,HelloSea*58`	Got packet
`wcq,8,HelloTop`	`wrq,a*d7`	Send response back