UC&NL

Underwater Communication
and Navigation Laboratory
http://unavlab.com
support@unavlab.com

uWAVE

Interfacing protocol specification

version 2.0 rev. c

26-03-2020

Content

1 Introduction....3
1.1 Physical layer....3
1.2 NMEA0183 Protocol standard....4
2 UWV sentences....5
2.1 IC_D2H_ACK - device reaction....5
2.2 IC_H2D_SETTINGS_WRITE - writing new settings....5
2.3 IC_H2D_RC_REQUEST - code request to a remote subscriber....6
2.4 IC_D2H_RC_RESPONSE - answer of remote subscriber....6
2.5 IC_D2H_RC_TIMEOUT - remote subscriber timeout....7
2.6 IC_D2H_RC_ASYNC_IN - incoming message from a remote subscriber....7
2.7 IC_H2D_AMB_DTA_CFG - ambient parameters and supply voltage configuration8
2.8 IC_H2D_AMB_DTA - ambient parameters and supply voltage....9
2.9 IC_H2D_DINFO_GET - request device information....9
2.10 IC_D2H_DINFO - device information....10
3 Command mode....11
4 Identifiers....12
4.1 Error codes....12
4.2 Remote commands....13
5 Appendix....14
5.1 Command mode interfacing examples....14
Example 1 - requesting device information....14
Example 2 - requesting remote data from a remote subscribers....15
Example 3 - setting up the ambient data configuration....16

1 Introduction

1.1 Physical layer

uWAVE underwater acoustic modems support data pairing using the RS-232 physical layer standard for asynchronous interface (UART) with a 3.3V data line voltage.
The connection is made using a four-wire cable with Tx (transmitter), Rx (receiver), Vcc (power) and GND (ground) wires.
Without the use of additional repeaters and interface converters, the maximum cable length, for which the correct operation of the interface is guaranteed, is no more than 2 meters.

Default port settings[1]:

Baudrate: 9600 bit/s
Data bits: 8
Stop bits: 1
Parity: No
Hardware flow control: No

WARNING!

The modems are powered by a 5 or 12 Volt DC source, while the data line voltage is 3.3 V.

[1] Specified parameters can be changed by a request

1.2 NMEA0183 Protocol standard

The NMEA0183 standard describes the format of text (ASCII) messages at the interactive level.

Sentence example: `$PUWV0,1,0*hh<CR><LF>`

Parts of a message (sentence) NMEA0183:

- '$' - sentence start,
- 'P' - Proprietary
- 'UVW' - manufacturer identifier
- '0' - sentence identifier
- ',' - parameters separator
- '*' - checksum separator
- 'hh' - checksum in hexadecimal format (for example FF, 01). Byte-by-byte XOR for all characters between '$' and '*'.
- <CR><LF> - end of sentence

2 UWV sentences

WARNING!

If not specified, the format of the parameters should be understood literally: 'xx' means two decimal digits, if the number is less than 10, then the left position is padded with zero: '02', '09' and not '2' and '9'.

The prefix D2H in the name of the message means that it is transmitted from the device (D) to the host system (H).
The H2D prefix in the message name means that it is transmitted from the host system (Host) to the device (Device).

2.1 IC_D2H_ACK - device reaction

Sentence format $PUWV0,x,x*hh <CR><LF>	
Field/Parameter	**Description**
$	Sentence start '$'
PUWV	UWV
0	Sentence ID
cmdID	Incoming sentence ID that caused ACK
errCode	Error code (see 4.1)
*	Checksum separator NMEA
hh	Checksum NMEA
<CR><LF>	Sentence end

2.2 IC_H2D_SETTINGS_WRITE - writing new settings

Sentence format $PUWV1,x,x,x.x*hh <CR><LF>	
Field/Parameter	**Description**
$	Sentence start '$'

PUWV	UWV
1	Sentence ID
txChID	Tx code channel ID
rxChID	Rx code channel ID
STY	Salinity, PSU
isCmdMode	‘0’ - command mode by pin, ‘1’ - command mode by default (to switch back to transparent channel mode use ‘0’)
*	Checksum separator
hh	Checksum
<CR><LF>	Sentence end

2.3 IC_H2D_RC_REQUEST - code request to a remote subscriber

Sentence format $PUWV2,x,x,x*hh <CR><LF>	
Field/Parameter	**Description**
$	Sentence start '$'
PUWV	UWV
2	Sentence ID
txChID	Tx code channel ID
rxChID	Rx code channel ID for the request
rcCmdID	Command ID (see 4.2)
*	Checksum separator
hh	Checksum
<CR><LF>	Sentence end

2.4 IC_D2H_RC_RESPONSE - answer of remote subscriber

Sentence format $PUWV3,x,x.x,x.x,x.x,x.x,x.x*hh <CR><LF>	
Field/Parameter	**Description**
$	Sentence start '$'
PUWV	UWV

3	Sentence ID
rcCmdID	Command ID (see 4.2)
propTime	Signal propagation time, sec
MSR	Mean main lobe to side-peak ratio, dB
Value	Requested value
Azimuth	Horizontal angle of arrival[2]
*	Checksum separator
hh	Checksum
<CR><LF>	Sentence end

2.5 IC_D2H_RC_TIMEOUT - remote subscriber timeout

Sentence format `$PUWV4,x*hh <CR><LF>`	
Field/Parameter	**Description**
$	Sentence start '$'
PUWV	UWV
4	Sentence ID
rcCmdID	Command ID (see 4.2)
*	Chechsum separator
hh	Checksum
<CR><LF>	Sentence end

2.6 IC_D2H_RC_ASYNC_IN - incoming message from a remote subscriber

Sentence format `$PUWV5,x,x.x,x*hh <CR><LF>`	
Field/Parameter	**Description**
$	Sentence start '$'
PUWV	UWV
5	Sentence ID

[2] For uWAVE USBL devices only, otherwise empty

rcCmdID	Command ID (see 4.2)
MSR	Mean main lobe to side-peak ratio, dB
Azimuth	Horizontal angle of arrival[3]
*	Checksum separator
hh	Checksum
<CR><LF>	Sentence end

2.7 IC_H2D_AMB_DTA_CFG - ambient parameters and supply voltage configuration

This message configures the modem's output of the readings of the built-in pressure/temperature sensor and supply voltage. After configuration, the modem can transmit these readings using the IC_D2H_AMB_DTA message (Section 2.8)

Sentence format `$PUWV6,x,x,x,x,x,x*hh <CR><LF>`	
Field/Paramter	**Description**
$	Sentence start '$'
PUWV	UWV
6	Sentence identifier
IsSaveToFlash	1 - store settings in internal Flash, 0 - do not store
PeriodMs	IC_D2H_AMB_DTA period in msec., 0 - disabled, 1 - tandem (send immediately after any outcoming message to the host system), or value from 500 to 60000 (0.5 - 60 sec.)
IsPressure	1 - pressure output enabled, 0 - disabled
IsTemperature	1 - temperature output enabled, 0 - disabled
IsDepth	1 - depth output enabled, 0 - disabled
IsVCC	1 - supply voltage output enabled, 0 - disabled
*	Checksum separator NMEA
hh	Checksum NMEA

[3] For uWAVE USBL devices only, otherwise empty

<CR><LF>	Sentence end

2.8 IC_H2D_AMB_DTA - ambient parameters and supply voltage

Sentence format $PUWV7,x.x,x.x,x.x,x.x*hh <CR><LF>	
Field/Parameter	**Description**
$	Sentence start '$'
PUWV	UWV
7	Sentence identifier
Pressure_mBar	Pressure in mBar
Temperature_C	Temperature in °C
Depth_m	Depth in meters
VCC_V	Supply voltage in Volts
*	Checksum separator NMEA
hh	Checksum NMEA
<CR><LF>	Sentence end

2.9 IC_H2D_DINFO_GET - request device information

Sentence format $PUWV?,x*hh <CR><LF>	
Field/Parameter	**Description**
$	Sentence start '$'
PUWV	UWV
?	Sentence ID
Reserved	Reserved
*	Checksum separator
hh	Checksum
<CR><LF>	Sentence end

2.10 IC_D2H_DINFO - device information

Sentence format
`$PUWV!,c--c,c--c,x,c--c,x,x.x,x,x,x*hh <CR><LF>`

Field/Parameter	**Description**
$	Sentence start '$'
PUWV	UWV
!	Sentence ID
Serial number	Device serial number
System moniker	System name
System version	System version
Core moniker	Communication subsystem
Core version	Communication subsystem version
acBaudrate	Data transmission speed, baud
rxChID	Rx code channel ID
txChID	Tx code channel ID
maxChannels	Total number of possible code channel IDs
styPSU	Water salinity, PSU (can be set by the user)
isPTS	‘1’ - device has a pressure/temperature sensor, ‘0’ - otherwise
isCmdMode	‘1’ - command mode by default, ‘0’ - command mode by command pin.
*	Checksum separator
hh	Checksum
<CR><LF>	Sentence end

3 Command mode

uWAVE modems provide the user with a so-called "transparent channel" when all data supplied to the input without changes and analysis are transmitted to the hydroacoustic channel, after which they are received by another modem and in the unchanged form are given to the user at the receiving side. In this regard, to be able to configure modems, as well as to measure the propagation time to remote subscribers, there is a command mode.

Modems analyze input data only in command mode. To switch to the command mode, the “service” core should be pulled to +3.3 V. After that, the “service” core should be pulled to the ground to exit the service mode.

Also, the command mode can be enabled by default using the IC_H2D_SETTINGS_WRITE sentence, when isCmdMode parameter equals to 1. To return to control by “service” core, IC_H2D_SETTINGS_WRITE sentence can be used with isCmdMode parameter equals to 0.

WARNING!

The core "service" is pulled ONLY to 3-5 V or ground, connecting it to a higher voltage will cause a FATAL and NON-GUARANTEE failure of the device.

WARNING!

Before switching on the device, the "service" core should be pulled to the ground, otherwise, the device will enter the software update mode.

4 Identifiers

4.1 Error codes

Error	Value	Description
LOC_ERR_NO_ERROR	0	Request accepted
LOC_ERR_INVALID_SYNTAX	1	Syntax error
LOC_ERR_UNSUPPORTED	2	Request not supported
LOC_ERR_TRANSMITTER_BUSY	3	Transmitter is busy
LOC_ERR_ARGUMENT_OUT_OF_RANGE	4	Specified parameter out of range
LOC_ERR_INVALID_OPERATION	5	Invalid request
LOC_ERR_UNKNOWN_FIELD_ID	6	Unknown field identifier
LOC_ERR_VALUE_UNAVAILIBLE	7	Requested parameter is not available at the moment
LOC_ERR_RECEIVER_BUSY	8	Receiver is busy (waiting for a remote answer)
LOC_ERR_TX_BUFFER_OVERRUN	9	Transmitter buffer is full
LOC_ERR_CHKSUM_ERROR	10	Checksum error

4.2 Remote commands

Command	Value	Description
RC_PING	0	Ping
RC_PONG	1	Pong
RC_DPT_GET	2	Request a depth value of a remote subscriber
RC_TMP_GET	3	Request a temp. Value of a remote subscriber
RC_BAT_V_GET	4	Request a battery voltage of a remote subscriber
RC_ERR_NSUP	5	Remote subscriber answered - request not supported
RC_ACK	6	Remote subscriber answered - request accepted
RC_USR_CMD_000	7	User command
RC_USR_CMD_001	8	User command
RC_USR_CMD_002	9	User command
RC_USR_CMD_003	10	User command
RC_USR_CMD_004	11	User command
RC_USR_CMD_005	12	User command
RC_USR_CMD_006	13	User command
RC_USR_CMD_007	14	User command
RC_USR_CMD_008	15	User command

5 Appendix

5.1 Command mode interfacing examples

In the following examples, the messages that come **to** a modem are prefixed with ‘<<’, and the messages that come **from** the modem are prefixed with ‘>>’. Any comments in the examples are prefixed with ‘//’.

All the data that is actually transferred to/from modem is marked with **bold**.

It is assumed that the modem is connected to a host system, and the command mode is enabled.

Example 1 - requesting device information

```
<< $PUWV?,0*27<CR><LF>

// PUWV? = IC_H2D_DINFO_GET

>> $PUWV!,3A001E000E51363437333330,STRONG,256,uWAVE
[JULY],257,78.27,0,0,28,0.0,1,0*18<CR><LF>

// PUWV! = IC_D2H_DINFO

// 3A001E000E51363437333330 = serial number,

// STRONG = system moniker

// 256 = 0x0100 system version is 01.00

// uWAVE [JULY] = core moniker,

// 257 = 0x0101 core version is 01.01

// 78.27 = acoustic channel baudrate, bit/sec

// 0 = Tx channel ID

// 0 = Rx channel ID

// 28 = total number of channels available

// 0.0 = salinity, PSU

// 1 = built-in pressure/temperature sensor is present

// 0 = command mode by default is disabled
```

Example 2 - requesting remote data from a remote subscribers

```
<< $PUWV2,0,0,2*28

// PUWV2 = IC_H2D_RC_REQUEST

// 0 = Tx channel ID (target Rx channel ID)

// 0 = Rx channel ID (target Tx channel ID)

// 2 = Request ID = RC_DPT_GET

>> $PUWV0,2,0*36

// PUWV0 = IC_D2H_ACK

// 2 = ACK is for PUWV2 request

// 0 = Error code = LOC_ERR_NO_ERROR

>> $PUWV3,0,2,0.00020,22.75,0.000,*1B

// PUWV3 = IC_D2H_RC_RESPONSE

// 0 = remote modem’s Rx channel ID

// 2 = Request ID = RC_DPT_GET

// 0.00020 = Propagation time, sec

// 22.75 = MSR (Main lobe to side-peak ratio), dB

// 0.000 = received value (in this case remote modem’s depth in meters)

<< $PUWV2,0,0,3*29

// PUWV2 = IC_H2D_RC_REQUEST

// 0 = Tx channel ID (target Rx channel ID)

// 0 = Rx channel ID (target Tx channel ID)

// 3 = Request ID = RC_TMP_GET

>> $PUWV0,2,0*36

// PUWV0 = IC_D2H_ACK

// 2 = ACK is for PUWV2 request
// 0 = Error code = LOC_ERR_NO_ERROR

>> $PUWV3,0,3,0.00030,26.31,27.300,*29

// PUWV3 = IC_D2H_RC_RESPONSE

// 0 = remote modem’s Rx channel ID

// 2 = Request ID = RC_TMP_GET

// 0.00030 = Propagation time, sec

// 26.31 = MSR (Main lobe to side-peak ratio), dB

// 27.300 = received value (in this case remote modem’s temperature in °C)
```

Example 3 - setting up the ambient data configuration

```
<< $PUWV6,0,1000,1,1,1,1*03<CR><LF>

// PUWV6 = IC_H2D_AMB_DTA_CFG

// 0 = isSaveToFlash = false

// 1000 = transmit ambient data every 1000 msec

// 1 = isPressure = true

// 1 = isTemperature = true

// 1 = isDepth = true

// 1 = isVCC = true

>> $PUWV0,6,0*32<CR><LF>

// PUWV0 = IC_D2H_ACK

// 6 = ACK is for PUWV6 request

// 0 = Error code = LOC_ERR_NO_ERROR

>> $PUWV7,1025.2,29.9,-0.014,5.0*18

. . .

>> $PUWV7,1026.3,29.9,-0.002,5.0*1D

// PUWV7 = IC_D2H_AMB_DTA
// 1026.3 = current pressure, mBar

// 29.9 = current temperature, °C

// -0.002 = current depth, m

// 5.0 = current supply voltage, V

<< $PUWV6,0,0,0,0,0,0*32

// PUWV6 = IC_H2D_AMB_DTA_CFG

// 0 = isSaveToFlash = false

// 0 = do not transmit ambient data

// 0 = isPressure = false

// 0 = isTemperature = false

// 0 = isDepth = false

// 0 = isVCC = false

>> $PUWV0,6,0*32

// PUWV0 = IC_D2H_ACK

// 6 = ACK is for PUWV6 request

// 0 = Error code = LOC_ERR_NO_ERROR
```