1 Liddell St, North Shields NE30 1HE
T: +44 (0) 191 447 6883 | **E:** sales@succorfish.com

W: succorfish.com | **Skype:** succorfish

SUCCORFISH

Delphis v3.2

User Guide

1. Introduction

Delphis' are a low-cost, low-power, miniature acoustic communication and ranging device for underwater vehicles, divers and subsea instruments. Data messages may be exchanged between units and an efficient "ping" protocol is implemented for range measurement by transponder operation. If multiple units are deployed in known locations, then long baseline positioning (LBL) operation is possible.

2. Specification

Supply voltage	3- 6.5v dc (5V or 6V supply recommended)
Supply current (5V supply)	Listening: 2.5mA Receiving: 5mA Transmitting: max 300mA
Acoustic frequency	24-32kHz
Acoustic source level	~168 dB re 1uPa @ 1m
Acoustic directivity	Near omnidirectional (reduction around cable entry of potted unit).
Physical Layer	Aperiodic orthogonal code keying with BPSK modulation and error correction code.
Acoustic data rate (raw)	640 bits/s, unicast and broadcast data messages up to 64 bytes in length.
Acoustic throughput (max)	463 bits/s
Addressing	Up to 256 units (addresses 0-255)
Ranging increment	4.7cm (c=1500m/s)
Ranging variance	~10 cm
Maximum range	> 2 km
RS232 interface	9600 Baud, 8-bit, no parity, 1 stop bit, no flow control

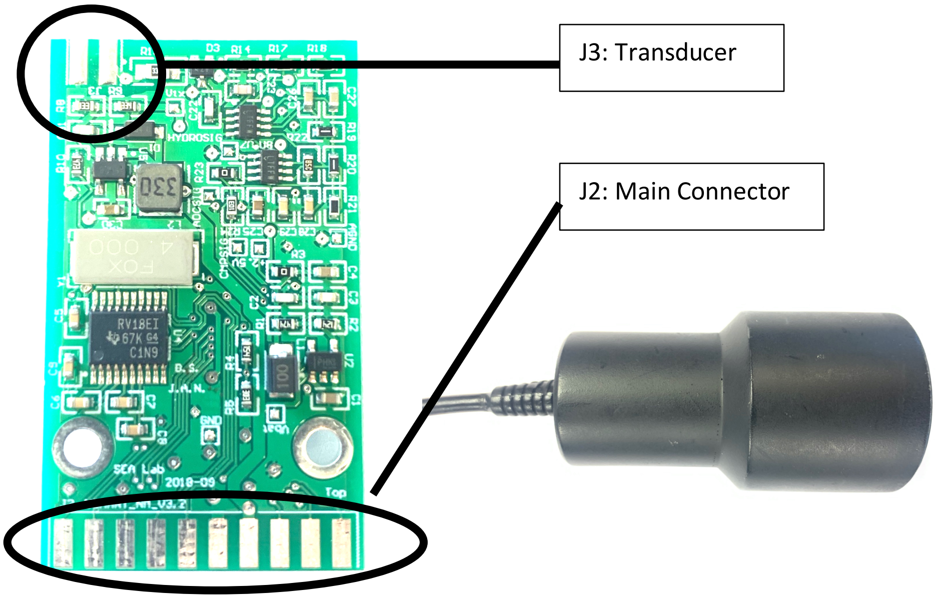
3. Components

Delphis may be supplied in 2 different forms:

1. A single polyurethane moulded unit incorporating transducer and electronics flying lead for power supply and data interface.
2. PCB for integration in existing housings with externally mounted transducer.

PCB Dimensions:

Board width: 27.0mm	Hole centres to board edge: 10.0mm
Board length: 45.0mm	Hole size: M2.5 clearance hole (2.9mm diameter)
Board thickness: 1.6mm	Hole centre to centre: 20.0mm
Header pin pitch: 2.54mm	



J2 – Main connector		J3 - Transducer	
1	Logic level UART TXD (2.5V)	1	Inner electrode (signal)
2	Hydrophone output (analog)	2	Outer electrode (GND)
3	Logic level UART RXD (2.5V)		
4	ADC input (ch 4a)		
5	RxS flag (2.5V logic)		
6	ADC input (ch 3)		
7	Vbat (+ve supply) *		
8	Vbat (+ve supply) *		
9	GND (0V) *		
10	GND (0V) *		
11	Serial TXD (RS232 output)		
12	DAC output		
13	Serial RXD (RS232 input)		
14	SPI MISO		
15	RxM flag (2.5V logic)		
16	SPI MOSI		
17	12C SCL		
18	SPI SCK		
19	12C SCA		
20	SPI CS		

Connections shown in grey are for possible future expansion (software enabled).

The transducer signal on J3 during transmission will reach 200Vp-p so electrical safety precautions must be observed when the device is activated.

* Care must be taken to connect the power supply/battery with the correct polarity as no reverse polarity protection is provided on board.

4. Basic connection

1. Connect 6V battery pack or power supply (5V recommended) between Vbat and GND on each modem. (Each modem will start up in receiving mode and draw no more than 2.5mA of current). **A short audible tone will be emitted by the modem to confirm startup.**
2. Connect unit(s) via RS232 serial cable (TXD, RXD, GND) to PC running a terminal programme (Termite is highly recommended) and configured with serial port settings (9600, 8, n, 1).

5. Serial communication protocol

5.1 Format

Serial communication format is (9600, 8, n, 1) with no flow control. All commands issued to the modem are prefixed with '$' and require no terminating characters. All responses from the modem are prefixed with '$' (for a local acknowledgement) or '#' (for the result of an executed command) and terminated with <CR><LF>. Unrecognised or invalid commands return 'E' to indicate an error.

Serial commands should be sent to the modem as a contiguous string with no more than ~ 2ms between bytes. If the delay between bytes exceeds this the serial handler will time out and return 'E'.

5.2 Node addressing

Each modem must be allocated a unique 8-bit node address (0-255) which is stored in non-volatile memory on the device. This is set and queried via a modem command as described in section 5.2. An error will be returned if you attempt to send a message to the address of the sending unit e.g. entering $P000 on unit 000.

5.2 Modem commands

Command String	Description	Responses
	Status and setup commands	
$Axxx	**Set node address** to xxx (ascii decimal e.g. 123)	#Axxx – confirms node address has been set to xxx and stored in non-volatile memory.
$?	**Query status**	#AxxxVyyyyy – where xxx is node address and yyyyy is a 16-bit supply voltage monitor value. To convert to a voltage: v = yyyyy * 15/65536
$Pxxx	**Ping** unit with address xxx	$Pxxx is returned immediately to acknowledge command.

		#RxxxTyyyyy is then returned if response is received from unit xxx. Range to unit xxx is given by R = yyyyy * c * 3.125e-5 where c is the sound velocity (assume 1500 m/s if no data is available). If no response is received, #TO is returned after 4s. Another ping request may be sent earlier than this and timing is reset.
$Vxxx	**Get supply voltage** on unit xxx	$Vxxx to acknowledge command. #Bxxx06Vyyyyy is then returned if a response is received from unit xxx. yyyyy is as in the $? command.

Data transfer commands

$Bnnddd...	**Broadcast data**. Send nn bytes (ddd...) to all units in range. Data (d) can be any printable or non-printable byte value. nn is the number of transmitted bytes as two ASCII decimal digits (02- 64).	$Bnn is returned immediately to confirm nn bytes have been broadcast. All units in range will output #Bxxxnnddd... (where xxx is the transmitting unit) when this message has been received.
$Uxxxnnddd...	**Unicast data**. Send nn bytes (ddd...) to unit xxx. Data (d) can be any printable or non-printable byte value.	$Uxxxnn is returned immediately to confirm nn bytes sent to unit xxx. Unit xxx will output #Unnddd... when message has been received.
$Mxxxnnddd...	**Unicast data and acknowledge**. Send nn bytes (ddd...) to unit xxx and request an ack to confirm delivery. Data (d) can be any printable or non-printable byte value.	$Mxxxnn is returned immediately to confirm nn bytes sent to unit xxx. Unit xxx will output #Unnddd... when message has been received. If message has been delivered the sending unit will receive #RxxxTyyyyy where yyyyy is the range

		count as in the ping command.
	Test and debug commands	
$Txxx	**Request test message** from unit xxx.	$Txxx is returned immediately to acknowledge command. Unit xxx will broadcast the 64 byte message "Hello! This is a Delphis v3 DSSS test transmission at 640 bps."
$Exxxnnddd...	**Echo data**. Request echo test of nn bytes (ddd...) from unit xxx. Data (d) can be any printable or non-printable byte value.	$Exxxnn is returned immediately to confirm nn bytes sent to unit xxx. Unit xxx will transmit a broadcast message containing the same data but **will not output to its serial port**.
$Q	**Get quality indicator.**	$Cx is returned where x is the number of bytes corrected by the error correction code for the last data packet detected (this does not apply to ping responses or acknowledgements). $C- is returned if the last packet failed to decode.

5.2 Acoustic packet durations

The following information can be used to calculate the expected transmission times for various acoustic message exchanges. Acoustic propagation delays should be added in calculating the expected duration of bidirectional data exchanges and this also excludes the time taken for serial data transmission at 9600 Baud. All values are in seconds:

Ping request/response	0.105
Command message ($Vxxx, $Txxx)	0.105
Data message (n bytes)	0.105 + (n+16) * 0.0125

5.3 Acoustic packet format

Data packet	Frame header	Message header	2-64 bytes payload	16 parity bytes
	30ms	75ms	25 – 800 ms	200ms

Command packet (105ms)	Frame header	Message header
	30ms	75ms

The acoustic packet format for both data packets and ping/command packets is shown above. All packets have a robust frame header waveform which the modems detect, followed by a message header which contains message length, address and control information. The message header has error control coding to maximise reliability. The variable length data payload has a fixed amount of redundancy (16 bytes) for an error correction code which enables up to 8 bytes to be corrected in any packet. If all errors cannot be corrected then a packet is rejected. Communication performance may be monitored by using the $Q command to report the number of errors corrected – packet reliability can be increased if necessary by reducing the payload size (and hence increasing the error rate that may be corrected).

5.4 Acoustic receive flags RxS and RxM

When the start of any acoustic packet is detected by Delphis, the RxS flag is raised. The timing of this rising edge coincides precisely with the detection of the packet header waveform and so it may be used for time difference of arrival (TDOA) estimates where multiple Delphis' are placed in an array. The RxS flag returns to zero at the end of the acoustic packet.

When a Delphis receives a unicast data message addressed to that unit, the RxM flag is raised for a short period corresponding to the transmissions of the received serial data. This signal may be used, for example, to wake up connected circuitry from a low power state.

5.5 In air acoustic testing

Delphis can communicate through air over at least 5m in a typical office/lab environment. For best results in air, modems should be aligned end to end i.e. with the circular faces pointed at each other (this does not apply in water where the transducer beam pattern is more omnidirectional). Ranging information is accurate if you apply the speed of sound in air which is c = 340 m/s.