

B&G

WTP₃

User Manual

English

www.bandg.com

CE Certification

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference, the user is encouraged to try to correct the interference by relocating the equipment or connecting the equipment to a different circuit. Consult an authorised dealer or other qualified technician for additional help if these remedies do not correct the problem.

The Wave Technology Processor 3 (WTP 3) meets the requirements for CFR47 Part 15 of the FCC limits for Class B equipment. WTP 3 meets the standards set out in European Standard EN 60945: 1997 IEC 945: 1996 for maritime navigation and radio communication equipment and systems.

™ Trademarks

All rights reserved. No part of this manual may be reproduced or transmitted in any form or by any means including photocopying and recording, for any purpose without the express written permission of Navico UK.

Information in this document is subject to change without notice. Navico reserves the right to change or improve its products and to make changes in the content without obligation to notify any person or organisation of such changes.

Navico, B&G, Wave Technology Processor, WTP, WTP2 & WTP3 are all trademarks of Navico UK Ltd. and may not be used without the express permission of Navico UK Ltd.

Product Liability and Safety Warnings

Brookes and Gatehouse Limited accept no responsibility for the use and/or operation of this equipment. It is the user's responsibility to ensure that under all circumstances the equipment is used for the purposes for which it has been designed.

Warning: Calibration

The safe operation of this equipment is dependent on accurate and correct calibration. Incorrect calibration of this equipment may lead to false and inaccurate navigational readings placing the yacht into danger.

Warning: Navigation Hazard

The WTP3 system is an Electronic Navigation aid and is designed to assist in the navigation of your yacht. It is not designed to totally replace conventional navigation procedures and precautions and all necessary precautions should be taken to ensure that the yacht is not placed into danger.

Caution: Electrical Supply

This equipment is designed for use with a power supply source of 12V dc. The application of any other power supply may result in permanent damage to the equipment and invalidating the warranty.

Caution: Cleaning

The use of alcohol or solvent-based cleaners will damage this equipment and any warranty in force will be invalidated.

Caution: Display Installation

Displays installed into locations manufactured from conductive materials (e.g. Steel, Carbon Fibre etc.) Should be insulated from the structure to prevent damage to the casings as a result of the effects of electrolysis.

Contents

Introduction	5
CPU	6
Interface Modules	8
System Architecture	9
System Example.....	12
Deckman	13
Deckman Control Facilities	13
Output.....	13
Damping.....	13
Calibration	14
Settings	15
Bounds Checking	16
Calibrate Boatspeed	17
Advanced Deckman Controls	17
Input Selection via Deckman	18
Use of a PC.....	20
Communication Options and configuration	20
Advanced Calibration via Deckman	20
Data Files	23
Defining the variables	24
Input Configuration Files.....	27
Display Output configuration files.....	34
Deckman Output configuration files	41
Optional Files (Advanced)	47
Parameters	58
Calibration	58
Sensor Calibration	62
Filtering and Damping	63
Sensor Damping	66
Data Logging	68
Exporting the .sqlite file	69
SQLite Database Structure.....	70
Alarms.....	71
Configuring Alarms	71

Installation	73
Physical Installation	73
Fastnet Network Installation	73
Mounting Instructions.....	74
Module Wiring	76
Analogue Module Wiring	78
Serial Module Wiring	81
Fastnet Installation.....	84
Network Terminator	84
WTP3 Variables	85
Supported Compass Types	90
Wind Calculation Flowchart.....	91
Upgrading the WTP3 Software.....	92
CPU	92
Module	92
WTP3 File Structure	93
Diagnostic Messages via Terminal	94
Error Types	94
Full Startup Description	95

Introduction

The B&G Wave Technology Processor (WTP) in combination with the B&G Deckman software and B&G displays creates the world's leading yacht instrument system for Grand Prix racing and Superyachts. Central to this high performance is the WTP, a powerful processor running an embedded linux operating system that runs hundreds of times faster than standard instrument systems.

The WTP3 brings new architecture with a central CPU linked to distributed data collection networks, operating on three independent CAN channels. Using this technology increases the possibilities for data collection.

WTP3 has evolved from the previous WTP processors which are used by Volvo winners, Jules Verne record holders and top inshore racers. With key input from a wide range of professional racing teams the new WTP3 along with B&G's range of displays, sensors and software is the ultimate Grand Prix instrument system.

Data is processed in the CPU, which outputs display data, transmits serial and digital data that is output via modules, communicates with Deckman and FTP clients via LAN and operates an on-board data logger for post-sail data analysis. Powered by a single 12 Volt power supply this new system is smaller, lighter, faster and even more flexible than its predecessors giving unrivalled confidence in the data it provides.

Controlled via a PC running Deckman software, the data can be viewed via any B&G display.

B&G understand the requirements of the professional racer and our instruments have been on board all winning boats in the Volvo Ocean Race, TP52 MedCup and America's Cup, along with setting records around the globe. We provide dockside support at the world's most important regattas and events, we support our systems with the best warranty in the business and – most importantly – we understand that great products by themselves are not enough.

B&G offer a comprehensive support package for our customers and their WTP. From the system design, configuration, sea-trials, upgrades and events the B&G Technical Support team is always available to give help and advice.

So we can continue to provide the ultimate racing solution, should you wish to develop your system further our Custom Projects team can work with you to develop your system to suit your specific requirements.

CPU

The WTP CPU is the core of a WTP system. Lightweight and robust, a hard anodised case protects the CPU from the elements, making it perfectly suited to the harsh environment found onboard racing yachts.



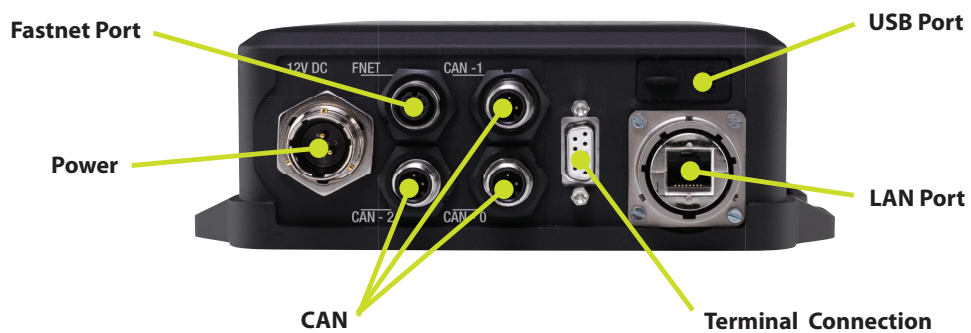
On the top of the CPU there are 7 diagnostic LED's enabling the user to see at a glance the status of the CPU and its I/O interfacing. These LED's will flash or change colour to indicate system status as detailed in the table below.



LED	DESC	STATUS	INFORMATION
PWR	Power	Solid Green	Power On
		Solid Red	Voltage too high or too low
		No Light	No Power
FNET	Fastnet	Flashing Green	Data being transmitted
		No Light or Not Flashing	No data being transmitted or received
CAN - 0	Can Channels 0, 1 & 2	Flashing Green	Data being transmitted
CAN - 1		Solid Red	Network not in use or CAN Hardware error
CAN - 2		No Light or Not Flashing	No CAN data being received or transmitted
LOG	Datalog/USB	Flashing Green	Data logging or upgrading software in progress
		Solid Green	Copying complete or software upgrade successful
		No Light	No data being logged or copied
LAN	Ethernet	Flashing Green	Data being transmitted
		No Light	No data being transmitted

Connectors

There are eight connectors on the CPU.



CONNECTOR (PORT)	DESCRIPTION
Power	12V DC
	Provides system power to fastnet and CAN
Fastnet	Display connection
	Compatible sensor support
	Pilot
CAN x 3	I/O Module networks
	Independently Configurable
	125 kbit > 1 Mbit Speed options (Default 250Kbit)
Terminal	System monitoring
LAN	100 Mbit Ethernet
	Deckman Comms
	FTP Access (File system and datalog)
USB Port	Software updates
	Configuration updates
	Data log file off-load

Interface Modules

There are 2 types of WTP module, Analogue and Serial. The modules act as the interface between analogue sensors, serial devices and other inputs and the CPU.

The correct module must be used in conjunction with its corresponding sensors. All modules are powered from the CAN Bus and can supply power to the sensors connected.

Analogue Module

The analogue module has 6 analogue inputs and 2 pulse inputs. This allows the unit to act as an interface for masthead units, speed sensors, analogue rate-gyros, potentiometer etc.

Serial Module

The serial module has 2 COM ports, each with input & output. Supported baud rates are 1,200 to 115,200. The modules support RS232, RS422, RS485 and NMEA0183 devices.

Modules can be located wherever is most convenient for the installer and can connect anywhere on the CAN networks.



On the top of the Module there is a diagnostic LED enabling the user to see at a glance the status of the module. This LED will glow green if there is power and will flash to indicate that it is sending/receiving data.

ANALOGUE INTERFACE MODULE	
PORT ALLOCATION	DESCRIPTION
4 x Analogue Input 12-Bit	3x -5V to +5V
	1x 0V to 5V
2 x Analogue Input 10-Bit	2x 0V to 5V
2 x Pulse Inputs	eg. Paddlewheel Speed Sensor or Masthead Unit

Note: If the system is to use an analogue speed input then terminals 5 & 6 may be used for this purpose.

SERIAL INTERFACE MODULE	
PORT ALLOCATION	DESCRIPTION
2 x Bi Directional COM Ports	RS232 / NMEA0183 / RS422 / RS485

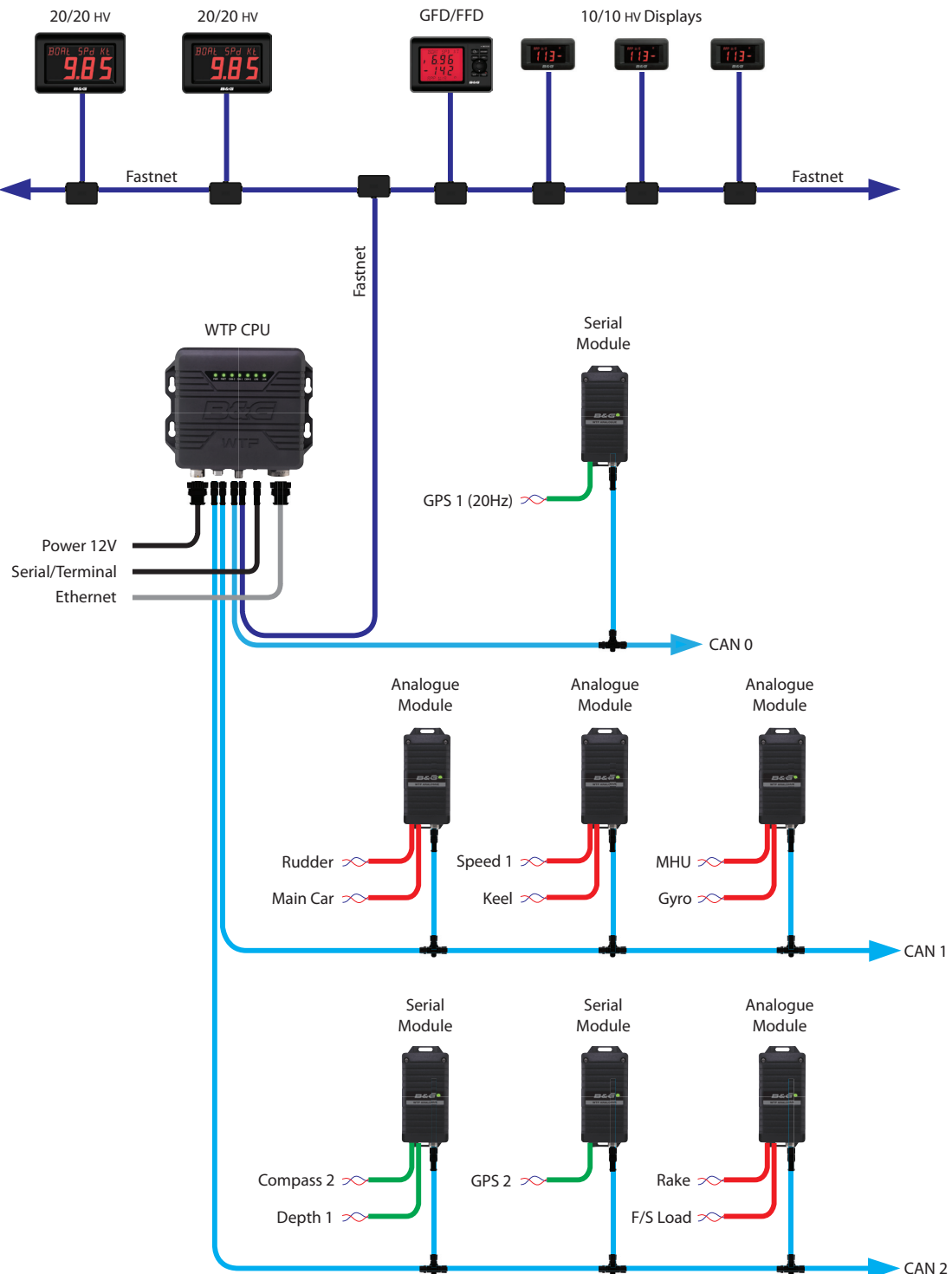
Each module contains DIP switches to enable the installer to give it a unique network address (Device number). Instructions on how to do this can be found in the installation section of this manual.

System Architecture

WTP3 interface modules are connected via multiple CAN bus networks. The system is completely flexible in how these are implemented, the sensor input is configured by the system installer based on:

- a) CAN channel used
- b) Device ID of the interface module, set by DIP switch in hardware
- c) Interface Port in use

For example a GPS input may be on Channel 0, Device 3, COM1. Below is an example of the system architecture and how the interface modules could connect to the network and CPU.



System Planning

A single CAN network configured to a baud rate of 250k is recommended to make system configuration straightforward while allowing for interfacing of NMEA2000 compatible sensors for certain functions. A dual or triple CAN network configuration is recommended where the system has network cable lengths of over 75 metres, or there is a requirement for redundant systems.

For example a dual CAN network could allow:

- a) Forward and Aft CAN networks, effectively doubling the possible network length
- b) Key systems operating on separate networks, for example one GPS unit on each network

Cable lengths

Maximum cable lengths vary depending on the installation. For example a long cable with one interface on the end will be very different to a similar length cable with 16 interfaces, all powering sensors. As such the following data is a guide only:

Guideline max. cable length per CAN channel

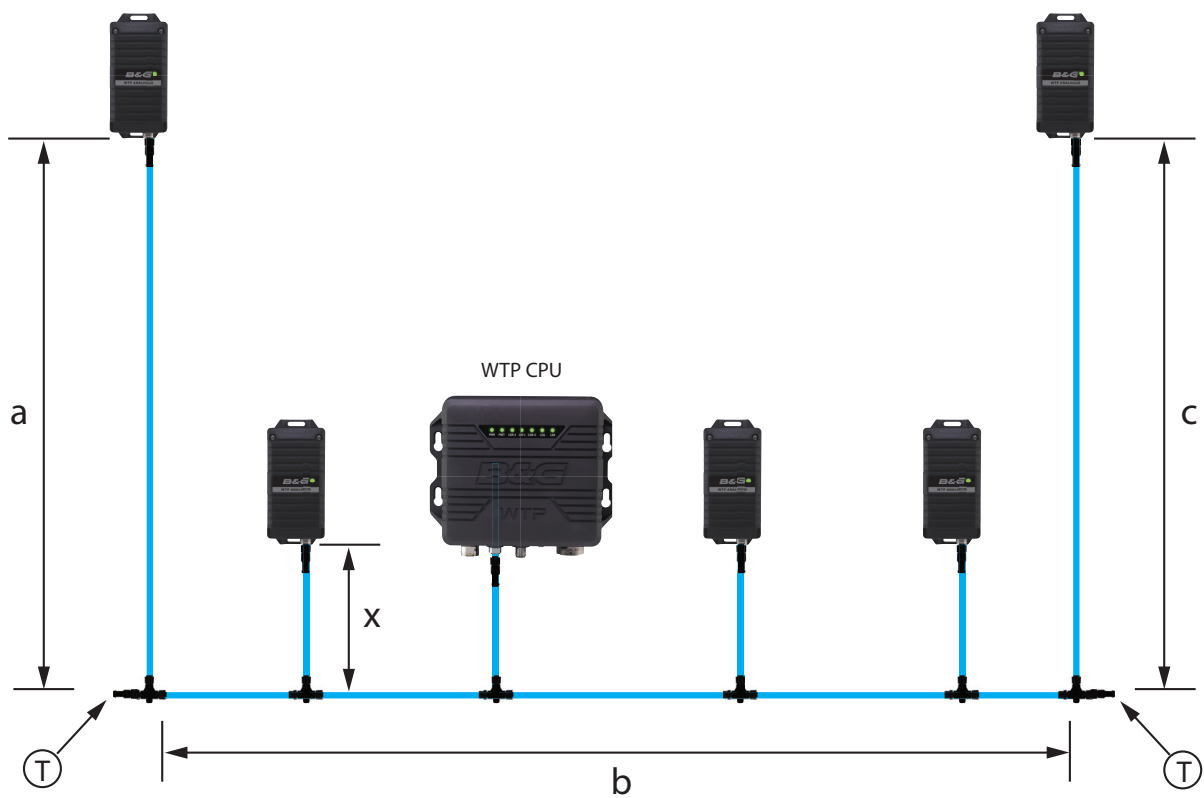
Baud Rate	Max Length
125k	150m
250k	75m
500k	35m
1M	5m

Maximum cable length is measured from the two end devices on the network (a+b+c), or the distance between terminating resistors – whichever is greater.

Guideline maximum drop cable length

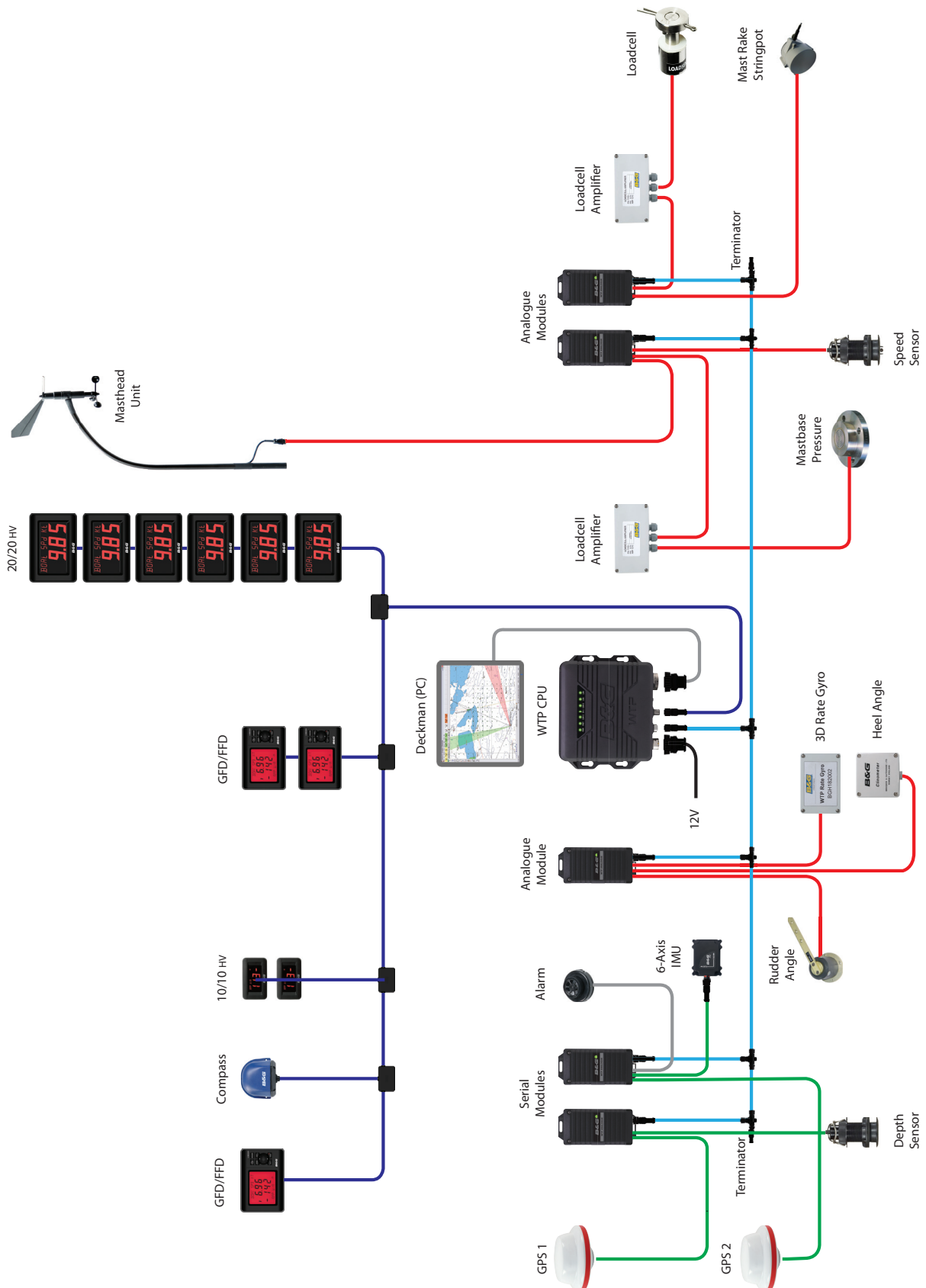
For best performance it is recommended that drop cables (x) are kept short, to approximately 0.4m. However in some cases it may be necessary to utilise longer drops, in this case we recommend the following guidelines.

Baud Rate	Max Length
125k	6m
250k	4m
500k	2m
1M	0.5m



System Example

Below is an example of a system for a typical race boat. This system show a variety of sensors and displays common on a WTP system.



Deckman

Deckman is the world's most advanced tactical navigation software and is used by winners in every field. This software is required to navigate and interface with the WTP3 processor. All configuration, damping and calibration functions are performed via Deckman software running on a 32-Bit Windows PC connected to the CPU via Ethernet

Deckman Control Facilities

To access the WTP3 control facilities in Deckman, choose gmenu instrument control. You will then see a dialog on the screen. Each box along the top of the dialog accesses a different control facility, as described below.

Output



Controls the output of Deckman variables to the WTP3 to be viewed on the displays.

In addition to the standard WTP3 system variables it is possible to output up to ten variables from Deckman to WTP3 for transmission to displays.

As supplied, WTP3 declares a Deckman menu containing four functions (remote 1, remote 2...), if more than four Deckman variables are required additional menu items will need to be added to the usermenu.d file.

To output a variable, click on the first blank line of the variables column, select the appropriate Deckman variable from the list followed by OK. The variable will be displayed on that line, with the channel column showing -1 and the current data shown in the final column.

The -1 indicates that output of this variable is currently disabled. To enable the output click on the channel number (currently -1) and assign a channel number of either 1,2,3 or 4 (on the standard configuration).

If the same channel number is assigned to more than one variable then the data will oscillate between those variables on the displays.

The screenshot shows the 'B&G WTP' window with the 'Output' tab selected. It contains a table with three columns: variables, chan, and data.

variables	chan	data
DST_STRB	1	557
DST_PORT	2	2957
TM_PORT	3	9911h
TM_STRB	-1	6839h

Damping



Allows you to alter the damping values on WTP3. The values are read directly from the WTP3 and any changes you make are sent as soon as they are entered.

Each of the menu options has a number: this is the damping applied to that instrument data. Generally, because the WTP3 uses a much faster processor and more sophisticated calculations, you will be able to use much lower damping values than with conventional systems, between 0.5 and 5 is suggested for normal variables.

Note: You can modify which variables are listed in this dialogue using the data files. See 'damping.d'

The screenshot shows the 'B&G WTP' window with the 'Damping' tab selected. It contains a table with two columns: variables and data.

variables	data
Boat_spd	0.1
Heading_1	0.5
Course	1.0
TWS	10.0
TWD	10.0

Calibration



Controls the calibration of variables by allowing you to input a calibration value to a particular variable, WTP3 has a sample set of calibration data as follows

Bspd_port and Bspd_stbd

Boat speed calibrations for the port and starboard side sensor respectively in knot. If you have only one boat speed sensor connected to both inputs enter the same value in both port and starboard. If you do not know what these values are, the boat speed can be calibrated using the cal boatspeed option.

Heading_1

Heading offset value for compass1. Once you have run the calibration routines for the compass sensor, you still need to align the unit relative to the boat. The most accurate way to do this is to sail on a known bearing towards a fixed charted object a long distance away. You then use Heading_1 to correct the compass bearing on the instruments until it matches the known bearing of the object. A positive offset will increase the reading of the compass heading.

Heel_1

Offset value for the first heel sensor. If the heel angle does not read zero when the boat is sitting upright in the water it is necessary to enter a correction value here. The heel sensor shows positive values when you are heeled to port, as on starboard tack upwind.

Trim_1

Offset value for the first trim sensor. If the trim angle does not read zero when the boat is sitting upright in the water it is necessary to enter a correction value here. The trim sensor shows positive values when the bow is trimmed down.

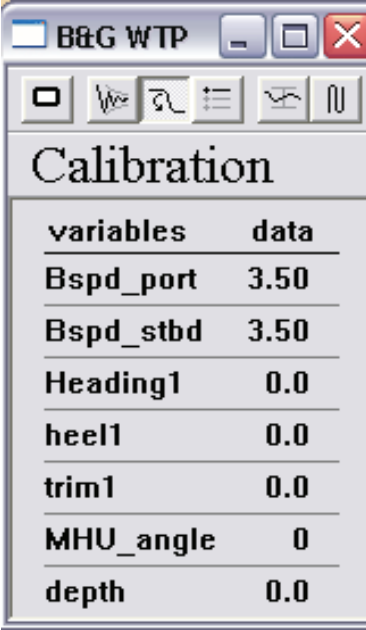
MHU_angle

Offset angle for the masthead unit sensor at the top of the mast. If you enter a positive value, it will decrease the Apparent Wind Angle on starboard tack, and increase the Apparent Wind Angle on port tack.

Depth

Offset for adjusting the depth sensor reading from the transducer position to either the keel (negative values) or the waterline (positive values).

Note: You can control which variables you are able to calibrate from this dialogue by adjusting the 'svcal.d' data file.



variables	data
Bspd_port	3.50
Bspd_stbd	3.50
Heading1	0.0
heel1	0.0
trim1	0.0
MHU_angle	0
depth	0.0

Settings



The settings dialogue controls all the normal items required to setup the WTP3.

mast_height

Used for wind calculations involving the rate gyros. This should be set to the distance from the waterline to the masthead sensor in feet.

leeway_cal

Leeway calibration value. A value between 8 and 13 is usually appropriate for most modern boats.

heel_enable

Should be set to 1 if you have a heel sensor and 0 if you do not. A heel sensor is highly recommended to achieve accurate wind data.

gyro_enable

Determines whether or not the system uses data from the rate gyro sensors to correct wind calculations. 1=use gyro data, 0=do not use gyro data. Normally there is no reason to disable this function.

variation

Magnetic variation. This is calculated automatically using data from the GPS. If the GPS is not providing this information it can be entered here. It will be overwritten by GPS information if it becomes available. Enter a positive value for East variation, negative for West.

dsp_time, dsp_up_ang, dsp_dn_ang

All refer to the switching of variables on displays according to either point of sailing or over time if this is specified in userout.d: see Data Files for more information). The dsp_time is the frequency with which the displays alternate between showing different variables, units here are 1/10th second; dsp_up_ang is the angle for the change between upwind and reaching settings; dsp_dn_ang is the angle for the change between reaching and downwind settings.

TWS_factor

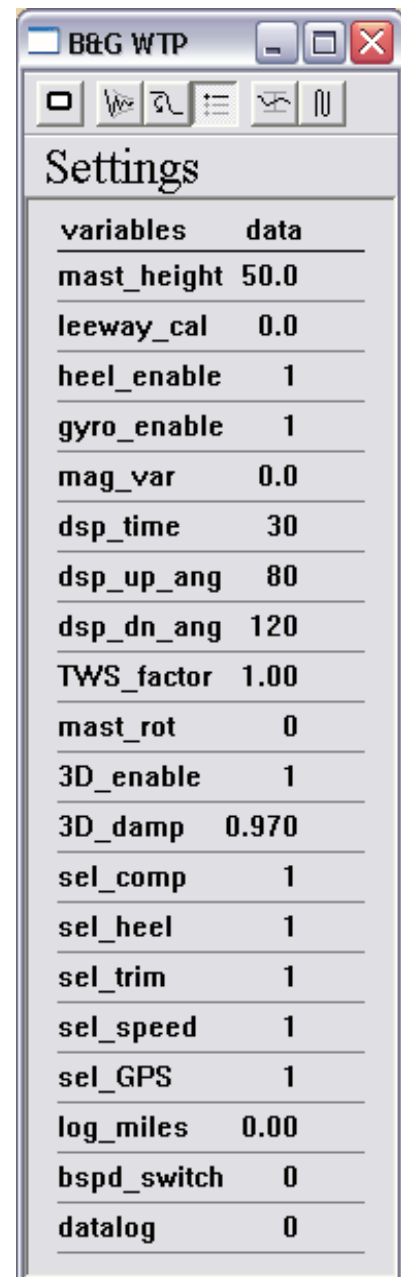
Will reduce or adjust wind speed by multiplying by this factor; this is used as an adjustment for wind weight.

mast_rot

Allows the option of using data from a mast rotation sensor. Available settings are 0 (off), 1 (on, using absolute value e.g. for Mast Twist calculations) or 2 (on, using +/- for fully rotating masts).

3D_Enable

This value is not used in WTP3, it is retained in the settings list to support legacy file sets. The 3D enable/disable is done automatically by WTP3 based on compass type, it can be overridden in the compassX.d files.



variables	data
mast_height	50.0
leeway_cal	0.0
heel_enable	1
gyro_enable	1
mag_var	0.0
dsp_time	30
dsp_up_ang	80
dsp_dn_ang	120
TWS_factor	1.00
mast_rot	0
3D_enable	1
3D_damp	0.970
sel_comp	1
sel_heel	1
sel_trim	1
sel_speed	1
sel_GPS	1
log_miles	0.00
bspd_switch	0
datalog	0

3D_damp

Damping for the 3D compass algorithms. should not be changed under normal use (0.970 is the default value).

sel_comp, sel_heel & sel_trim

Control which input is used for heading, heel and trim respectively. Refer to 'Multiple Compass, Boat Speed and GPS Inputs' for more information.

sel_speed

Controls which input is used for boat speed. Refer to 'Multiple Compass, Boat Speed and GPS Inputs' for more information.

sel_GPS

Controls which set of GPS data is used for position, SOG, COG etc. Refer to 'Multiple Compass, Boat Speed and GPS Inputs' for more information.

log_miles

Allows you to reset the Log variable to any value you wish (though zero is likely to be most useful).

bspd_switch

Controls whether Heel Angle or MWA is used for controlling the port and starboard boatspeed inputs.

datalog

Allows you to manually start or stop the data logging.

Note: Datalog manually starts or stops data logging. Data logging is only available if a valid datalog.d file is present.

Bounds Checking



To act as an additional level of data validation checking between the instrument system and the PC, WTP3 allows the user to set limits on input values.

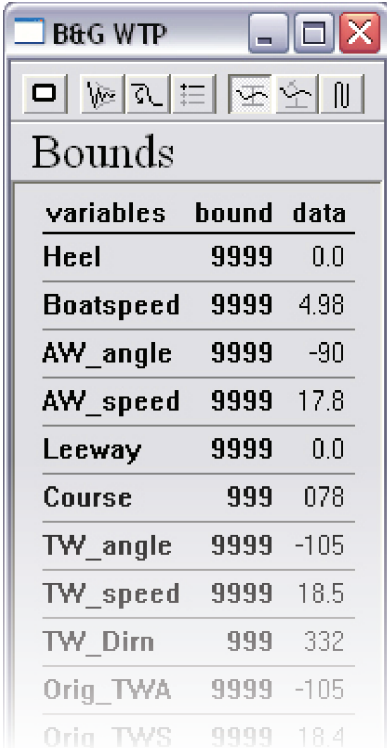
The incoming data is checked against the previous values. This display allows you to change the bounds that are used for each incoming variable; smaller values make errors less likely but increase the possibility that the numbers will 'stick' because of dramatic boat manoeuvres.

The values shown in the example should be used unless you are experiencing difficulties with a particular variable.

Reset Bounds Checking



Clicking this button will cause the next set of incoming data to overwrite the old, even if it falls outside the error bounds. This is not normally required but may be necessary under some circumstances.



variables	bound	data
Heel	9999	0.0
Boatspeed	9999	4.98
AW_angle	9999	-90
AW_speed	9999	17.8
Leeway	9999	0.0
Course	999	078
TW_angle	9999	-105
TW_speed	9999	18.5
TW_Dirn	999	332
Orig_TWA	9999	-105
Orig_TWS	9999	18.4

Calibrate Boatspeed



This function helps you to calibrate your boat speed correctly, and works in exactly the same way as the traditional method of measuring the time taken to cover a known distance. Deckman will automatically calculate the calibration values from the tests you select.

	Time	Set	Log	GPS	Del Time	Cse
1	12:10:40	0.5000	0.5070	0.0000	02:49	000
2	12:13:41			0.0000	00:11	

Click Start run at the beginning of the run, and then End run to finish. Details of each run are displayed in the table: the start time of the run, the distances from your input, the log and the GPS are shown, Del Time shows the elapsed time for the run, and the course during the run is shown on the extreme right.

In the "Use current box", you can choose what type, if any, of current information to factor into the calculations. In the Calibration distance box, you can select whether to use the distance entered by you or that received from the GPS. Click on the runs you wish to use for the calibration to send them to the Selected runs in calculation box.

When you have selected runs, a calibration value is then shown in the Calibration box. Either choose Send Cal to accept the current value or continue to do more runs and calculations.

Advanced Deckman Controls

When the Instruments Control dialog is open in Deckman, clicking the menu button will give you some controls specific to the WTP3. The Advanced Calibration or Advanced Damping options allow you to access the calibration and damping files described below. The WTP Guru option allows the advanced user to access system files, do not use the guru option if you are not familiar with the file level operation of WTP3.

See Use of a PC for further details on these functions.

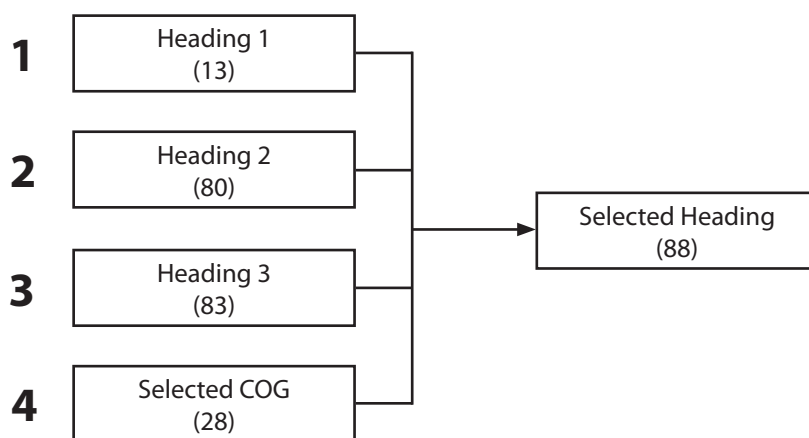
Input Selection via Deckman

Multiple Compass, Boat Speed and GPS Inputs

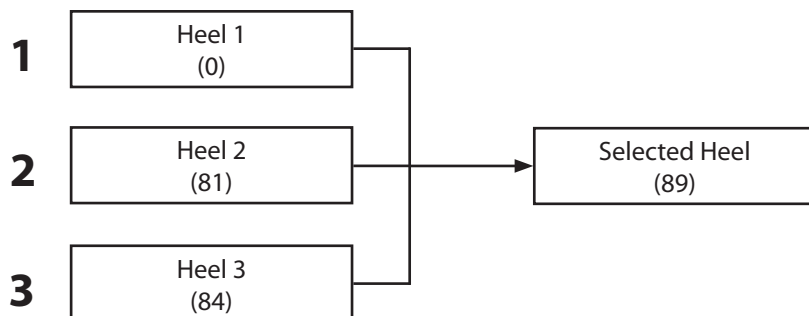
WTP3 is able to handle up to three compass inputs (as well as associated heel and trim values), two boat speed inputs and two GPS inputs. This is useful for testing purposes, to compare different sensors, or as a backup. Variables to handle data from all of these inputs exist on the system.

To select which input you wish to use in the calculations on the WTP3 use the Instrument Control option in Deckman. Whichever input is selected is then copied into the SelHdg (88), SelHeel (89), SelTrim (90), SelBoatSpd (91) or Selected GPS (multiple functions) variables, this is then used in the calculations and output to Deckman and displays as required. This selection is shown below for each of the selectable variables, in each case the standard variable number is shown in brackets.

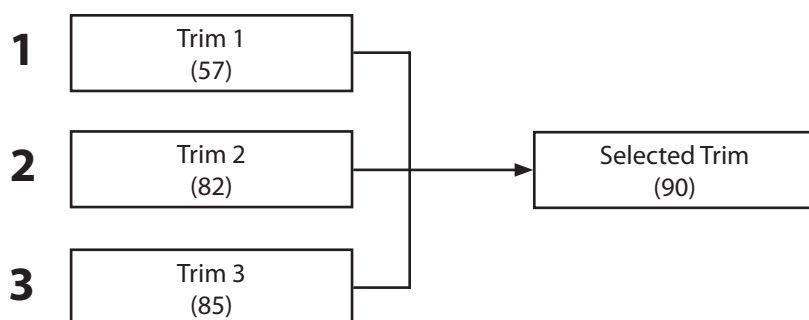
Heading selection



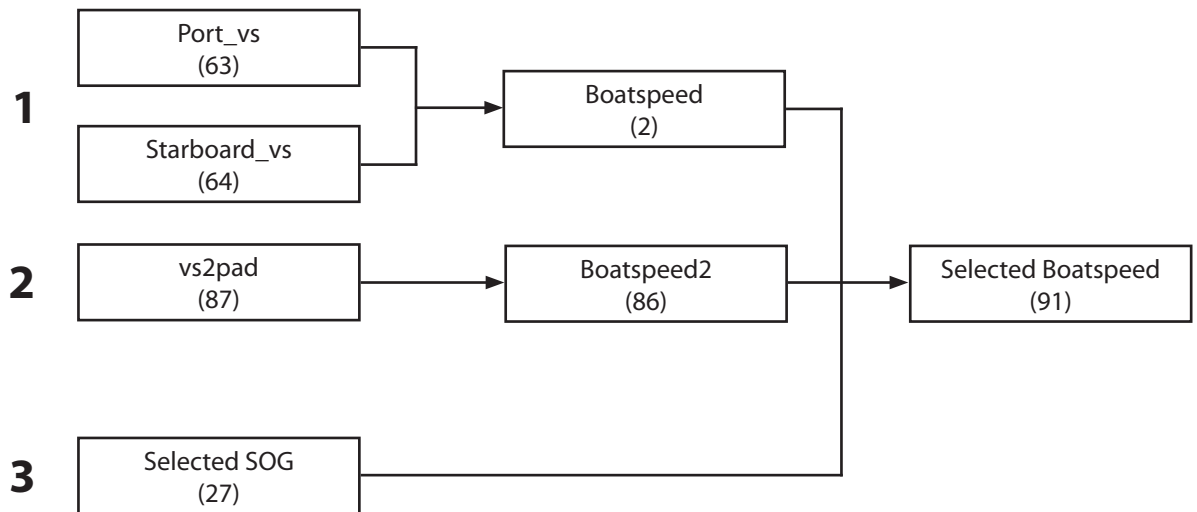
Heel selection



Trim selection



Boat Speed Selection



For boat speed there are a couple of extra steps to take account of the damping and the fact that there may be separate port and starboard paddlewheels.

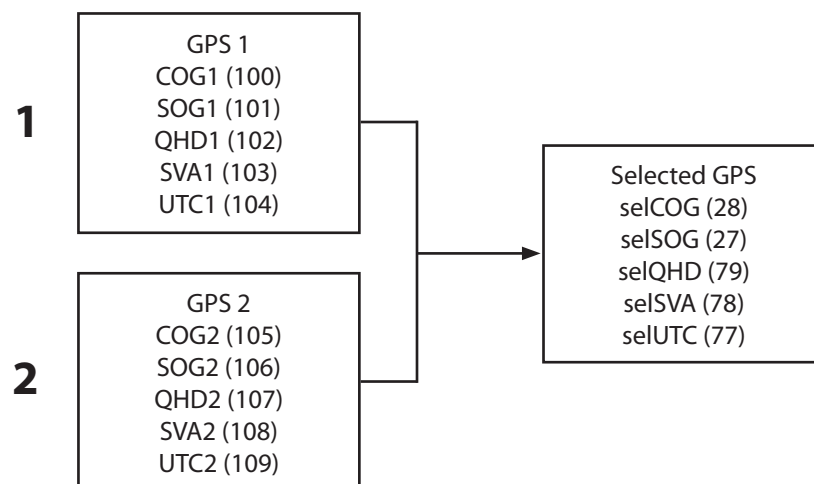
The raw data from the standard port and starboard paddle inputs (63 and 64 respectively) are combined to make Boatspeed, the WTP3 uses Heel Angle or MWA to determine which of the two inputs to use. The raw data from any additional speed sensor is used to make Boatspeed2

Note: This stage is necessary to enable users to filter the raw data from the sensors.

If you enter a 1 in sel_speed in the Settings dialog (see 'Settings') then Boatspeed (2) will be copied into SelBoatSpd (91), if you enter a 2 in the dialog then Boatspeed2 (87) will be copied and if you enter a 3 SelSOG (27) will be copied.

By default WTP3 is setup to look at Boatspeed (2).

GPS Selection



Use of a PC

Apart from the normal use of Deckman software to control WTP3 in the normal racing environment there are other times when it is necessary to communicate with WTP3 via a PC. Direct file modifications, file backup, file restore and diagnostics can be carried out via PC using Deckman, FTP or HyperTerminal (or similar terminal program).

Communication Options and configuration

Ethernet

The Ethernet interface allows much faster data transmission than a standard serial link and used as the method for interfacing Deckman. The Ethernet interface on WTP3 transmits normal function data to PCs running Deckman via the UDP protocol, UDP is available to any PC in the correct IP range on the network.

Any PC on the Ethernet can control the WTP files, calibrations etc. This is controlled via a TCP/IP protocol so that, in the case of multiple PCs running Deckman, only one copy of Deckman has access to the files at a time and only one copy of Deckman can act as the navigational server.

Note: *The navigational server option is controlled via the configure comms dialogue.*

The Ethernet port is configured by default. It is only necessary to configure the instrument type within Deckman (gmenu – change instruments) as 'WTP2 Ethernet' and set the network properties on the PCs used to match the WTP3 IP addressing. It is recommended that the fixed IP address is retained (192.168.0.2) and the PC(s) on the network are set to IP addresses 192.168.0.3 onwards. The Subnet Mask on the PC should also be set to match the WTP3, usually set to 255.255.255.0. This can be changed to suit the individual requirements of the boat.

The IP Address of the WTP3 is set in the network.d file.

Note: *For an example of this file please see: Data files.*

Advanced Calibration via Deckman

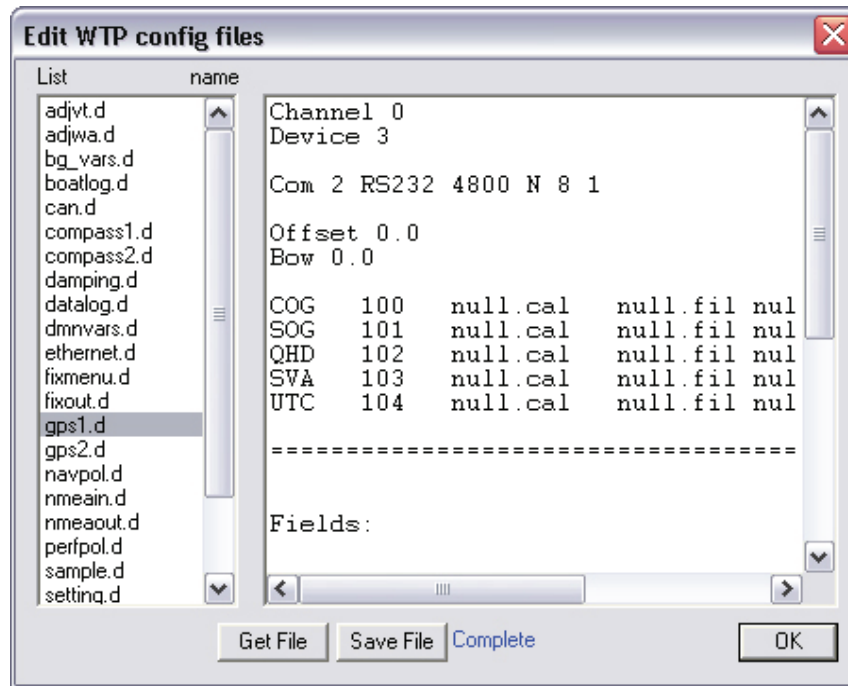
Using the Advanced Calibration, Advanced Damping and WTP Guru functions in Deckman the user can directly modify calibration, filter and system files.

Note: *It is recommended that advanced users use FTP to modify system files rather than WTP Guru and the advanced file editors within Deckman. However these utilities are very useful to take a quick look at the file content while sailing.*

Caution: *incorrect modification of the WTP3 files (especially system files accessed by the WTP Guru option) can lead to incorrect data values or system instability. Only modify files directly if you are familiar with the file level operation of WTP3. Common settings, calibration and damping can be carried out in the normal Deckman Instrument Control dialogues. It is recommended that regular backups are made of your WTP3 files using FTP.*

These functions are accessed via the menu button whilst Deckman's Instrument Control function is in use.

Use of the dialogue itself is very straight-forward, simply highlight the file you wish to view or modify in the left hand column by clicking on it with the mouse, and then click the 'Get File' button. The file content will be displayed in the right hand window. At this stage it is possible to make any modifications before clicking the 'Save File' button to save the file back to the WTP3. The 'OK' button closes the dialogue, if you haven't chosen to save the file before clicking 'OK' your changes will be lost.



Advanced file editing dialogue (WTP Guru shown)

Advanced Calibration

The Advanced Calibration function allows file edit access to any of the calibration files stored in the WTP3 /calibs/ directory.

Advanced Damping

Identical operation to Advanced Calibration except by using this option the filter files (/filters/ directory) are displayed rather than the Calibration files.

WTP Guru

WTP Guru allows access to the system data files that directly affect the configuration of inputs, variables and outputs to Deckman and displays (/data/ directory). Note that if you modify files with WTP Guru you will need to re-boot the WTP3 before they take effect – as such it is just as easy, and recommended, to use the FTP access to the files for this purpose which allows the backup of the existing files before making changes.

FTP

The WTP3 uses the File Transfer Protocol (FTP) to carry out file management tasks. Windows Explorer in the most recent versions of Windows has FTP functionality – as this is straight-forward and available to most users this is the program we will use for our examples. Other FTP clients may be used.

Connecting to WTP3

Assuming that the network is correctly configured it is only necessary to open Windows Explorer and type in the following into the address line:

ftp://192.168.0.2 (where 192.168.0.2 is the IP address of WTP3).

Backing up WTP3 files

To make a backup of the WTP3 files simply select all the files and directories and drag them (or copy/paste) to a folder on your PC.

Editing WTP3 files

To edit WTP3 files drag the individual file from the WTP3 to a local folder (e.g. the Desktop), make a backup, then modify it using a suitable text-editing tool (Wordpad, supplied with Windows, is recommended). To update the WTP3 select the modified file and drag it back into the relevant WTP3 directory in the Explorer window, confirm the file transfer if prompted.

Terminal

The WTP3 has a 'Terminal' connection that allows technicians or advanced users to diagnose operational issues. To view the diagnostic text it is necessary to use a terminal package such as HyperTerminal which is supplied with Windows.

Connect a suitable serial lead to the WTP. The only connections required for diagnostic use are Rx, Tx and Ground.

Configure your terminal program with the following information and then establish the connection (in HyperTerminal this is achieved simply by clicking the 'connect' button – other terminal programs may use different logic:

COM port	The serial port you have connected the lead to on your PC
Baud Rate	115,200
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

When the WTP3 is booting or running you will be able to see status messages on the screen, refer to 'Diagnostic messages displayed via Terminal' for full details.

Data Files

Caution: It is only recommended that advanced users or installers alter the data files directly, as described in this chapter. Most calibration, damping etc. can be controlled from Deckman as described in 'Deckman Control Facilities'.

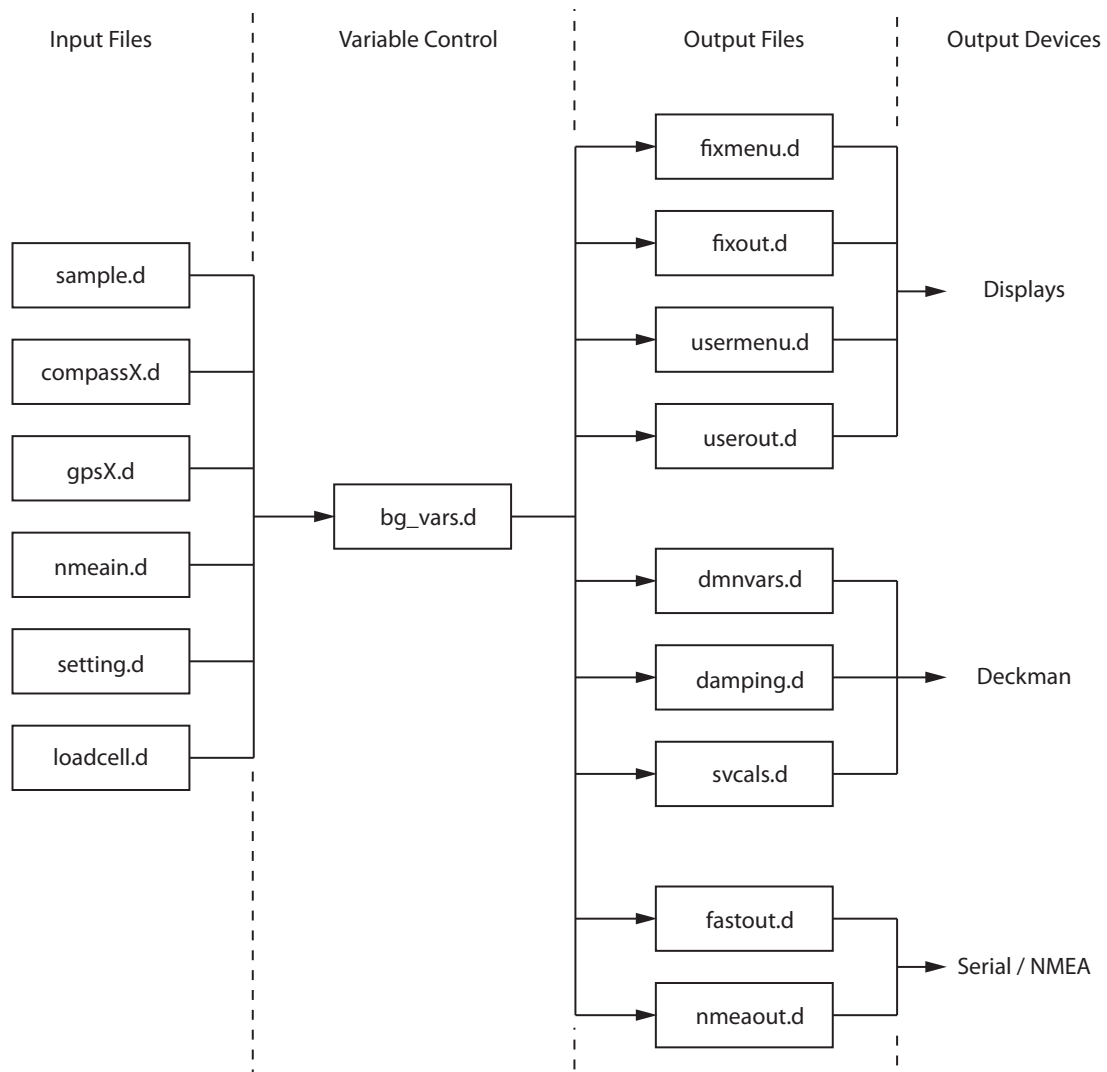
The configuration of the WTP3 can be changed to suit individual requirements by using the data files. The data files described below control how data is input onto the WTP3, stored in the variable database and output to Deckman and the displays.

All of the variables in the WTP3 database are listed in the file `bg_vars.d`. Inputs from the various components (masthead unit, paddle wheels, strain gauges and so on) are fed into the WTP3 via the `sample.d`, `compassX.d`, `gpsX.d` and `nmeain.d` files.

The structure of the menus on the displays is controlled by the `fixmenu.d` and `usermenu.d` files and the outputs to these menus are controlled by `fixout.d` and `userout.d`.

Deckman is sent data from WTP3 defined by the content of the `dmnvars.d` file and allows control of WTP3 settings, calibrations and damping values via `setting.d`, `svcal.d` and `damping.d` respectively.

There are other files, which are not shown for clarity but are detailed in the following chapter.



The structure of the main data files on the WTP3

Defining the variables

bg_vars.d

Note: It is possible to write comments within a text file. WTP will ignore any text line that starts with a # symbol. Use the # symbol at the beginning of a comment line or any text line you wish to ignore. If you use the = symbol all information thereafter on any line will be ignored.

Example.

```
# Variable No,Long Name,Short Name,Decimal Places,Absolute Value,Data Type
```

```
87 VS2pad          V2p      2      0      0
```

```
=====
```

Everything below the (=) symbol will be ignored. Only 1 (=) needs to be used but a line of = shows a clear separation.

This file is the most important on the WTP3, as it lists all of the variables in the system - if variables are not listed here then they will not be in the WTP3 at all.

```
[variables]
```

```
0 Heel1           Hg1      1      1      0
```

```
1 dotHeel         dHl      1      1      0
```

```
2 Boatspeed       VS       2      0      0
```

```
3 dotVS           dVS      2      0      0
```

```
4 SmoothVS        sVS      2      0      0
```

```
5 MHU_A_R         A_R      4      0      0
```

```
6 MHU_A_G         A_G      4      0      0
```

```
7 MHU_A_B         A_B      4      0      0
```

```
8 MW_angle        MWA      1      1      1
```

```
9 MW_speed        MWS      1      1      0
```

```
10 AW_angle       AWA      0      1      1
```

```
11 AW_speed       AWS      1      0      0
```

```
12 Leeway         Lee      1      1      0
```

```
13 Heading1       Hd1      0      0      2
```

```
14 Course         Cs       0      0      2
```

```
15 dotCourse      dCs      1      0      0
```

```
16 TW_angle       TWA      0      1      1
```

```
17 TW_speed       TWS      1      0      0
```

```
18 TW_dirn        TWD      0      0      2
```

```
19 VMG            VMG      1      1      0
```

```
20 GW_speed       GWS      1      0      0
```

```
21 GW_Dirn        GWD      0      0      2
```

```
22 Orig_TWA       TA       0      1      1
```

```
23 Orig_TWS       TS       1      0      0
```

```
24 Orig_TWD       TD       0      0      2
```

```
25 MastRot        MRo      1      1      0
```

```
26 TWD_Off        TDo      0      0      1
```

```
27 SelSOG         SOG      2      0      0
```

```
28 SelCOG         COG      0      0      2
```

```
29 VMC            VMC      2      0      0
```

30	Opt_VMC	OVC	2	0	0
31	Cse_OVMC	COC	0	0	2
32	Vs_target	Vt	2	0	0
33	Vs_targ%	Vt%	0	0	0
34	TWA_targ	WAt	0	1	1
35	Vs_perf	PPV	2	0	0
36	Vs_perf%	PP%	0	0	0
37	Vs_nav	PNV	2	0	0
38	Vs_nav%	PN%	0	0	0
39	Brg_o_Mrk	BOM	0	0	2
40	Dst_t_Mrk	DTM	2	0	4
41	Tm_t_Mrk	TTM	0	0	3
42	Curr_Rate	CrR	2	0	0
43	Curr_Dir	CrD	0	0	2
44	MCur_Rate	MCR	2	0	0
45	MCur_Dir	MCD	0	0	2
46	DCur_Rate	DCR	2	0	0
47	DCur_Dir	DCD	0	0	2
48	Battery	Bat	0	0	5
49	Rudder	Rud	1	1	0
50	Rake	Rke	1	0	0
51	gyro_hl	GHl	1	0	0
52	gyro_dhl	GdH	2	0	0
53	gyro_trm	GTm	1	0	0
54	gyro_dpt	GdP	2	0	0
55	gyro_hdg	GHg	0	0	2
56	gyro_dyw	GdY	2	0	0
57	Trim1	Tm1	1	0	0
58	forestay	frs	0	0	0
59	AnSp2	sp2	0	0	0
60	seatemp	sea	1	0	0
61	airtemp	air	1	0	0
62	Barometer	Bar	0	0	0
63	port_VS	pVS	2	0	0
64	stbd_VS	sVS	2	0	0
65	CMW_angle	CWA	1	1	1
66	CMW_speed	CWS	1	1	0
67	Depth	Dep	1	0	0
68	XTrkErr	XTE	2	0	4
69	TWA_OVMC	AOC	0	1	1
70	VMG_Targ	VGT	0	0	0
71	VMG_Targ%	VGP	0	0	0
72	OppTrkW	OTW	0	0	2
73	OppTrkG	OTG	0	0	2
74	Log	Log	2	0	4
75	pitchRMS	PMS	1	0	0
76	pitchPrd	PPd	1	0	0
77	SelUTC	UTC	1	0	0

78	SelSVA	SVA	1	0	0
79	SelQHD	QHD	1	0	0
80	Heading2	Hg2	1	0	2
81	Heel2	Hl2	1	1	0
82	Trim2	Tm2	1	0	0
83	Heading3	Hg3	1	0	2
84	Heel3	Hl3	1	1	0
85	Trim3	Tm3	1	0	0
86	BoatSpd2	VS2	1	0	0
87	VS2pad	V2p	2	0	0
88	SelHdg	SHg	1	0	2
89	SelHeel	SHl	1	1	0
90	SelTrim	STm	1	0	0
91	SelBoatSpd	VSS	1	0	0
92	Hdg2_Heave	Hv2	1	0	0
93	GGBrg	GGB	1	0	2
94	GGRng	GGR	1	0	0
95	HHDiff	HHD	1	0	1
96	MastWnd	MWM	0	0	0
97	AnSp4	AS4	0	0	0
98	AnSp5	AS5	0	0	0
99	ANSp6	AS6	0	0	0
100	gps1cog	cg1	0	0	2
101	gps1sog	sg1	2	0	0
102	gps1qhd	qh1	0	0	0
103	gps1sva	sv1	0	0	0
104	gps1utc	ut1	0	0	0
105	gps2cog	cg2	0	0	2
106	gps2sog	sg2	2	0	0
107	gps2qhd	qh2	0	0	0
108	gps2sva	sv2	0	0	0
109	gps2utc	ut2	0	0	0
110	Variation	Var	2	0	1

The lines define the variables, as follows:

Variable Number	The unique identifying number for the variable
Long Name	Descriptive name for the variable (must not contain spaces)
Short Name	Short name for the variable will only be 3 characters
Decimal Places	The number of decimal places that data is stored with
Absolute Value	Absolute value (1) or not (0), new variables should be set to 0
Data Type	Standard data (0), $\pm 180^\circ$ (1), 0 to 359° (2), time (3), distance (4)

Input Configuration Files

Defining Analogue Inputs and derived variables

sample.d

This file lists all the analogue inputs, pulse inputs, derived variables and user variables to the WTP3. The variables are split into sections according to the type.

Each column then defines a particular item: the first column gives the name of the variable (from bg_vars.d) and the last four show the variable number (from bg_vars.d) and the names of the calibration, filtering and alarm files (*.cal, *.fil and *.alm respectively).

In the [analogue] section, the second column is the CAN Channel which the Analogue Module is connected to (0, 1 or 2), the third column is the Device ID of the module and the fourth column is the analogue channel (1-6). In the [pulse] section, the second column is the CAN Channel which the Analogue Module is connected to (0, 1 or 2), the third column is the Device ID of the module and the fourth column is the pulse input port (1 or 2).

[derived] variables are fixed and should not be altered, except if possible to set calibration, filter and alarm files if these are desired.

[uservars] is a section for user defined variables. For details on this see the uservars section.

```
#can channel, device id, analogue input, variable number, cal file, fil file,
alarm file

[analogue]
MHU_A_B 0 0 4 7 null.cal null.fil null.alm
MHU_A_G 0 0 5 6 null.cal null.fil null.alm
MHU_A_R 0 0 6 5 null.cal null.fil null.alm

gyro_dhl 0 0 1 52 gdheel.cal gdheel.fil null.alm
gyro_dpt 0 0 2 54 gdpitch.cal gdpitch.fil null.alm
gyro_dyw 0 0 3 56 gdyaw.cal gdyaw.fil null.alm

rudder 0 1 1 49 null.cal null.fil null.alm
Battery 0 1 2 48 null.cal null.fil null.alm
MastRot 0 1 3 25 null.cal null.fil null.alm

spare1 0 1 4 50 null.cal null.fil null.alm
spare2 0 1 5 59 null.cal null.fil null.alm
airtemp 0 1 6 61 null.cal null.fil null.alm

[pulse]
MHU_VA 0 0 2 9 MHUVA.cal null.fil null.alm
portpad 0 1 2 63 portpad.cal portpad.fil null.alm
stbdpad 0 1 1 64 stbdpad.cal stbdpad.fil null.alm
VS2pad 0 0 1 87 vs2pad.cal vs2pad.fil null.alm

[derived]
MW_angle 8 mwa.cal null.fil null.alm
Boatspeed 2 null.cal boatspd.fil null.alm
Orig_TWD 24 null.cal orig_twd.fil null.alm
Orig_TWS 23 null.cal orig_tws.fil null.alm
TW_dirn 18 null.cal tw_dirn.fil null.alm
TW_speed 17 null.cal tw_speed.fil null.alm
Course 14 null.cal course.fil null.alm
```

VMG	19	null.cal	vmg.fil	null.alm
VMC	29	null.cal	vmc.fil	null.alm
OptVMC	30	null.cal	optvmc.fil	null.alm
CseOVMC	31	null.cal	cseovmc.fil	null.alm
TWAOVMC	69	null.cal	twaovmc.fil	null.alm
OppTrkW	72	null.cal	null.fil	null.alm
OppTrkG	73	null.cal	null.fil	null.alm
GyroHdg	55	null.cal	gyrohdg.fil	null.alm
GyroHl	51	null.cal	gyrohl.fil	null.alm
GyroTrm	53	null.cal	gyrotrm.fil	null.alm
Leeway	12	null.cal	null.fil	null.alm
pitchRMS	75	null.cal	pitchrms.fil	null.alm
pitchPrd	76	null.cal	pitchprd.fil	null.alm
CMWA	65	null.cal	cmwa.fil	null.alm
CMWS	66	null.cal	cmws.fil	null.alm
Boatspd2	86	null.cal	boatspd2.fil	null.alm
WindToMast	96	mwa.cal	null.fil	null.alm
TargetBSpd	32	null.cal	targbspd.fil	null.alm
variation	110	null.cal	null.fil	null.alm

[uservars]

Compass input

compass1.d; compass2.d; compass3.d

The compassX.d files define the inputs of serial or networked compass sensors and their associated heel and trim sensors (see Supported Compass Types). Three examples of compass configuration files are shown below:

Example A: Using a NMEA serial compass

In this example we are configuring a standard NMEA compass input with heel and trim data, the format is as follows:

```
Channel 0
Device 4

Com 1 NMEA0183 4800 N 8 1

heading1 1 13 heading1.cal heading1.fil null.alm
heel1 1 0 heel1.cal heel1.fil null.alm
trim1 1 57 trim1.cal trim1.fil null.alm
```

Line 1 defines the CAN Channel the module is connected in the format “Channel n” [where n is 0, 1 or 2]

Line 2 defines the Device ID of the Module (NOTE: this is set via the DIP switch inside the module).

Line 3 defines the COM port settings: [possible settings]

COM Port	COM port used [1 or 2]
Hardware Type	Hardware settings for the type of data [RS232, RS485, RS422, NMEA0183]
Baud Rate	Baud Rate setting to suit the input [1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600, 115200]
Parity	Parity setting to suit the input [N = no parity]
Data Bits	To suit the input [7 or 8]
Stop Bits	To suit the input [1 or 2]
Compass Type	Identifies special compass types (see compatible compass types)

All other lines define variable inputs for heading, heel and trim from this compass:

Variable Name	Name of the variable, for user information only]
COM Port	Same port as line 3 (above)
Variable Number	Variable number where data is stored (from bg_vars.d)
Calibration File	Filename of the calibration file to use
Filter File	Filename of the filter/damping file to use
Alarm File	Filename of the alarm file to use [optional]

Example B: *Using a B&G networked compass - Halcyon Gyro-Stabilised Compass.*

In this example we are configuring a B&G Halcyon Gyro-Stabilised compass sensor which is present on the B&G Fastnet network (typically attached directly to an ACP Pilot). When using a networked compass there are some specific changes to the compass file:

1. Next to 'Com' Add text 'FASTNET BGGYRO' to identify the port as Fastnet and the compass type as a Halcyon Gyro-Stabilised Compass.
2. The values in the variable lines which normally shows the COM port are modified to be the function number on the B&G Fastnet bus.

```
Com FASTNET BGGYRO

heading1      74      13      heading1.cal      heading1.fil      null.alm
heel1         52       0      heel1.cal         heel1.fil         null.alm
trim1         155     57      trim1.cal         trim1.fil         null.alm
```

Example C: *Using a B&G networked compass - Halcyon 2000 Compass*

In this example we are configuring a B&G Halcyon 2000 Compass sensor which is present on the B&G Fastnet network. When using a networked compass there are some specific changes to the compass file:

1. Next to 'Com' Add text 'FASTNET HALCYON' to identify the port as Fastnet and the compass type as a Halcyon 2000 Compass.
2. The values in the variable lines which normally shows the COM port is modified to be the function number on the B&G Fastnet bus.

```
Com FASTNET HALCYON

heading1      74      13      heading1.cal      heading1.fil      null.alm
```

Note: If 2 sources of the same data type are on the network you can proceed the function number with a node number separated by a dot (.).

```
Com FASTNET HALCYON

heading1      16      13      heading1.cal      heading1.fil      null.alm
```


GPS input

gps1.d; gps2.d

The gpsX.d files define the inputs of GPS units and the location of the antenna relative to the bow of the yacht. An example of a GPS configuration file is shown below:

```
Channel 1
Device 1

Com 1 RS232 19200 N 8 1

Offset 0.0
Bow 55.0

COG      100      null.cal    null.fil    null.alm
SOG      101      null.cal    null.fil    null.alm
QHD      102      null.cal    null.fil    null.alm
SVA      103      null.cal    null.fil    null.alm
UTC      104      null.cal    null.fil    null.alm
```

Line 1 defines the CAN Channel the module is connected to.

Line 2 defines the Device ID of the Module (NOTE: this is set via the DIP switch inside the module).

Line 3 defines the COM port settings:

COM Port	COM port used [1 or 2]
Hardware Type	Hardware settings for the type of data [RS232, RS485, RS422, NMEA0183]
Baud Rate	Baud Rate setting to suit the input [4800, 9600, 19200, 28800, 38400, 57600, 115200]
Parity	Parity setting to suit the input[N = no parity]
Data Bits	To suit the input [7 or 8]
Stop Bits	To suit the input [1 or 2]
Compass Type	Identifies special compass types (see compatible compass types)

Line 4 defines the GPS antenna position:

Offset	GPS Antenna offset from centreline (- Port, + Starboard) in feet
Bow to GPS	GPS Antenna distance from Bow in feet

The additional lines define the variables. Normally this should not be changed though, for example, you may wish to add a link to a filter file for SOG:

Variable Name	Name of the variable, for user information only]
COM Port	Same port as line 3 (above)
Variable Number	Variable number where data is stored (from bg_vars.d)
Calibration File	Filename of the calibration file to use
Filter File	Filename of the filter/damping file to use
Alarm File	Filename of the alarm file to use [optional]

Note: For NMEA0183 GPS inputs the preferred inputs are as follows:

Position: GGA, GLL (If GGA or GLL is not available then RMC can be used for Position, in this case VTG must be disabled.)
Velocity: VTG

NMEA input

nmeain.d

This file controls NMEA inputs excluding any that may be for GPS or Compass sensors. A typical use for this file is to define the decoding of Depth and Sea Temperature from an active NMEA sensor.

File example (Depth and Sea Temperature):

```
Channel 1
Device 0

Com 2 NMEA0183 4800 N 8 1

SDDBT    1    67    depth.cal    null.fil    null.alm
YXMTW    1    60    seatep.cal    null.fil    null.alm
```

Line 1 defines the CAN Channel the module is connected to.

Line 2 defines the Device ID of the Module (NOTE: this is set via the DIP switch inside the module).

Line 3 defines the COM port settings:

COM Port	COM port used [1 or 2]
Hardware Type	Hardware settings for the type of data [RS232, RS485, RS422, NMEA0183]
Baud Rate	Baud Rate setting to suit the input [4800, 9600, 19200, 28800, 38400, 57600, 115200]
Parity	Parity setting to suit the input[N = no parity]
Data Bits	To suit the input [7 or 8]
Stop Bits	To suit the input [1 or 2]

Each additional line controls the decoding of NMEA sentences, as follows:

Variable Name	Name of the variable, for user information only]
NMEA Field	Position in NMEA sentence for required data
Variable Number	Variable number where data is stored (from bg_vars.d)
Calibration File	Filename of the calibration file to use
Filter File	Filename of the filter/damping file to use
Alarm File	Filename of the alarm file to use [optional]

Where the NMEA code is all the characters between the \$ and the first comma in the NMEA sentence and the input field number is equal to the number of commas before the required value, for example the file above would decode the following input correctly (note that Depth is read in feet by default, it is converted to metres in the calibration file):

```
$SDDBT,32.81,f,10.00,M,5.46,F*hh<CR><LF>
```

(where "f" is feet, "M" is metres and "F" is Fathoms)

```
$YXMTW,18.2,C*hh<CR><LF>
```

Special Case: The 'XDR' NMEA sentence is processed slightly differently as a special case due to the possibility of multiple inputs. The following example shows a single pressure sensor input (sensor type code "P"), decoding the following sentence:

```
$IIXDR,P,1.000,B,BARO*hh<CR><LF>
```

```

Channel 1
Device 0

Com 2 NMEA0183 4800 N 8 1

IIXDR P 2 62 baro.cal null.fil null.alm

```

Line 1 defines the CAN Channel the module is connected to.

Line 2 defines the Device ID of the Module (NOTE: this is set via the DIP switch inside the module).

Line 3 defines the COM port settings:

COM Port	COM port used [1 or 2]
Hardware Type	Hardware settings for the type of data [RS232, RS485, RS422, NMEA0183]
Baud Rate	Baud Rate setting to suit the input [4800, 9600, 19200, 28800, 38400, 57600, 115200]
Parity	Parity setting to suit the input[N = no parity]
Data Bits	To suit the input [7 or 8]
Stop Bits	To suit the input [1 or 2]

The additional lines control the decoding of the NMEA sentences, as follows:

Variable Name	Name of the variable, for user information only]
NMEA Field	Position in NMEA sentence for required data
Variable Number	Variable number where data is stored (from bg_vars.d)
Calibration File	Filename of the calibration file to use
Filter File	Filename of the filter/damping file to use
Alarm File	Filename of the alarm file to use [optional]

Note: *If a checksum is present on an incoming NMEA sentence it will be checked and the sentence discarded if incorrect, if checksums are not present the sentence will be accepted "as is". It is recommended to implement checksums where possible to avoid incorrect data being accepted.*

Display Output configuration files

Fixed Fastnet menus

fixmenu.d

This file controls the configuration of the standard function menus onto the displays – the menu items which are standard parts of the WTP3 system but are not declared normally by the display are declared here.

Modifying this file is not recommended. It is suggested that users adjust the *usermenu.d* file to alter network output settings. The format of the file is shown below for completeness.

POL_SPD_KT	0102	3	7e	5
TARG_SPDKT	0102	4	7d	5
REACHINGPC	0102	5	7c	5
MEAS_W/A_@	0106	5	5a	5
MEAS_W/SKT	0106	6	57	5
TARG_TWA_@	0106	8	53	5
HEEL_____@	0107	1	34	5
TRIM_____@	0107	2	9B	5
YAW_RTE_@S	0112	1	44	5
PTCH_RTE@S	0112	2	9E	5
ROLL_RTE@S	0112	3	3C	5

The format is:

Menu item name	This name is defined in the menus
Menu number	Defines which menu contains the function (see <i>usermenu.d</i> for full list)
Order in chain	The position of the function in the menu
Fastnet Function No.	The Fastnet function number of the function in hexadecimal
Node	The Node number that the menu is declared from (for system compatibility)

Note: *The text in the first column (e.g. POL_SPD_KT, etc.) is not seen on the displays, and is only to make it easier to recognise what the numbers refer to; if custom titles for the menu items are required then the items should be sent using *userout.d* as described in the next section.*

Fixed Fastnet output

fixout.d

This file controls the standard data outputs onto the B&G Fastnet network – variables such as Boat Speed, Wind data, Heading etc. (which are common to all systems) are defined here.

Modifying this file is not recommended. It is suggested that the users adjust the userout.d file if it is necessary to alter network output settings. The format of the file is shown below for completeness.

```
11

4 0 0 5
51 /AWA_____M 10 0
55 /TWS_____KT 17 1
4d /AWS_____KT 11 1
59 /TWA_____@ 16 1

4 0 0 5
57 /MWS_____KT 9 1
5a /MWA_____@ 8 0
49 /HDG_____@M 88 0
4a /HDGR_____@M 88 0

4 0 0 5
4e /AWSR_____KT 11 1
6d /TWD_____@M 18 0
6e /TWDR_____@M 18 0
52 /AWAR_____@ 10 0

3 0 0 1
41 /VS_____KT 91 2
42 /VSR_____KT 91 2
7F /VMG_____KT 19 2

4 1 0 5
44 /YAW_RTE_@S 56 1
9E /PTCH_RTE@S 54 1
3C /ROLL_RTE@S 52 1
75 /TIMER____MS -1 0

4 1 1 5
7e /POL_SPD_KT 35 1
7d /TARG_SPDKT 32 1
7c /REACHINGKT 36 1
53 /TARG_TWA_@ 34 0
```

```

2  9  1  1
C1 /DEPTH____M    67  1
1F /SEA_TEMP_C    60  1

3  9  3  5
87 /BAROMETRMB    62  1
82 /LEEWAY____@   12  1
9a /OPP_TACK@M    72  0

3  9  5  9
e8 /DTW_GC__NM    40  1
e6 /BTW_GC__@M    39  0
ee /XTE____NM     68  2

4  4  0  5
9C /MAST_ANG_@    25  0
9D /WA_MAST____@  96  0
9B /TRIM____@    90  1
34 /HEEL____@    89  1

4  4  2  9
ea /COG____@M     28  0
eb /SOG____KT     27  1
84 /TIDE_SET@M    43  0
83 /TIDE_RTEKT    42  1

```

Line 1 defines the number of Transmit Groups in the file

The first line of each Transmit Group defines:

Number of variables	Up to a maximum of 5 in each group
Transmit delay*	Sets the update rate (0=10Hz, 1=5Hz, 2=3.3Hz, 4=2Hz, 9=1Hz)
Transmit offset*	Allows spacing of data transmission on network to optimise bandwidth
Transmit Node ID	Allows full system compatibility with other Fastnet devices

*Example of Transmit delay and offset on Fastnet Traffic can be seen in the userout.d section.

Further lines in each group:

Fastnet Function No.	Must match the number defined in the menu (see fixmenu.d)
Function name	Name displayed on the display (not transmitted in fixout.d, reference only)
Variable number	From bg_vars.d
Decimal Places	The number of decimal places shown on displays

Note: The fixout.d file does not support context switching or time based oscillation of functions – userout.d should be used for this purpose.

Defining Custom Fastnet Menus

usermenu.d

This file enables you to either add a function to an existing menu or add a new menu with associated functions.

In the file example shown below we have added two new menus called DECKMAN and SAILS with functions and we have also added two functions to the existing PARAMTR menu. Note that the Deckman functions are all named RemoteX as the Deckman software will send the relevant function text with the function – here we are just defining a placeholder in the menu.

```
DECKMAN      01b1  03
REMOTE1      01b1  01  F0
REMOTE2      01b1  02  F1
REMOTE3      01b1  03  F2
REMOTE4      01b1  04  F3

SAILS        01b2  04
MAIN_POS     01b2  01  A1
JIB_CAR_P    01b2  02  A2
JIB_CAR_S    01b2  03  A3

CWA_____@  0112  04  A8
CWS_____@  0112  05  A9
```

The file format here is best dealt with by looking at part of the example above:

```
SAILS        01b2  04
MAIN_POS     01b2  01  A1
JIB_CAR_P    01b2  02  A2
JIB_CAR_S    01b2  03  A3
```

The first line creates a new menu called SAILS, format as follows:

Menu name	As is appears on the display - "SAILS" in the example above
Menu ID number	New menus use ID numbers 01b1, 01b2, 01b3 etc.
Menu position	Identifies where the menu appears in the Display - here it is 4th in the chain

The following lines add functions to the menu – here we are adding three functions to the menu, the Mainsheet Traveller position and Jib Car positions (port and starboard). The format is as follows:

Function Name	As is appears on the display - e.g. "MAIN_POS" in the example above*
Menu group ID	The ID number of the menu group the function is to appear in
Function order	The position in the menu the function should appear (simple 01, 02, 03 etc.)
Fastnet Function No.	The hexadecimal value of the fastnet function number

* If you wish to display units use " " for a space, "@" for a degrees sign.

Function Name	As is appears on the display - e.g. "MAIN_POS" in the example above*
Menu group ID	The ID number of the menu group the function is to appear in
Function order	The position in the menu the function should appear (simple 01, 02, 03 etc.)
Fastnet Function No.	The hexadecimal value of the fastnet function number

New functions should use Fastnet function numbers a1-a4 and a8-ae. If further function numbers are required please contact B&G.

New menus use ID numbers 01b1, 01b2, 01b3 etc. Existing menus are numbered as follows:

SPEED	0102
LOG	0103
DEPTH	0104
NAVIGATE	0105
WIND	0106
PERFORM	0107
WAYPOINT	0108
MOTOR	0109
TEMP	010a
TIME	010b
MISC	010c
PARAMTR	0112
EXTERNAL	0113

Note: Any items added in any of the menu or output files need to be defined correctly in *bg_vars.d* etc. so that the function exists in WTP3 in the first place.

Defining Custom Fastnet Outputs

userout.d

This file controls how additional “user” data is sent from the WTP3 to the displays and allows you to have different variables shown according to your point of sailing and/or to have oscillating variables on a time basis.

Any variable detailed in this output must have been defined in *bg_vars.d* and a menu item defined using *usermenu.d* to enable you to access the data from a display. The only exception to this rule is remote Deckman outputs that are dealt with automatically by WTP3.

```

2

3 9 7
A1 /MAIN_POS 115 2
A2 /JIB_CAR_P 116 2
A3 /JIB_CAR_S 117 2

2 9 8
A8 /CWA_____@ 65 1
A9 /CWS_____KT 66 1

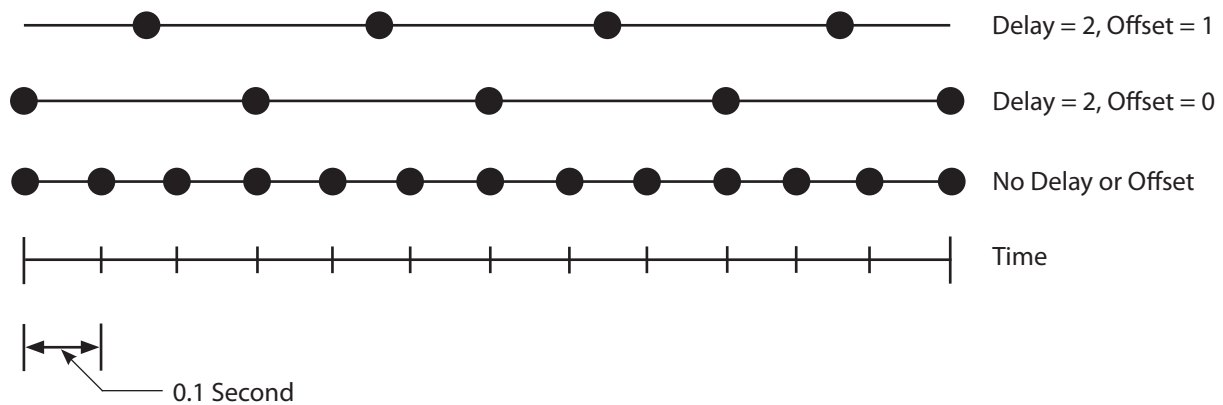
```

The first line of the file contains a single number which defines the number of transmit groups that follow; in the example above we have 2 transmit groups.

The first line of each group defines:

Number of variables	Up to a maximum of 5 in each group
Transmit delay	Sets the update rate (0=10Hz, 1=5Hz, 2=3.3Hz, 4=2Hz, 9=1Hz)
Transmit offset	Allows spacing of data transmission on network to optimise bandwidth

Example of Transmit delay and offset on Fastnet Traffic:



Further lines in each group:

Fastnet Function No.	Must match the number defined for the menu (usermenu.d) / Function name
Function name	Name displayed on the display once data is transmitted, see below for options
Variable Number	From bg_vars.d
Decimal Places	The number of decimal places shown on displays

Variable Switching

The userout.d file can be configured so that variables oscillate on a time basis, on a context sensitive basis (where the variables switch whether the boat is sailing upwind, reaching, downwind or is in the pre-start) or on a port and starboard basis.

These options are shown in the modified example below:

```

2
4 1 0
A1 /MAIN_POS 111 2
A2 /JIB_CAR_P 112 2 JIB_CAR_S 113 2
A3 /AWA 10 0 /TWA 16 0 /TWA 16 0 /TIMER -1 0
A4 /TARG_SPDKT 32 1 ?FORESTAY_T 58 2

2 9 8
A8 /CWA_____@ 65 1
A9 /CWS_____KT 66 1

```

Here we have three changes:

1. The two Jib Car variables oscillate automatically – the time delay is set via the osc_time value in Deckman (Instrument Control, Settings)
2. We have added an output that switches on the sailing context: AWA upwind, TWA when reaching or downwind and Timer during the start sequence. The True Wind Angles for switching between upwind, reaching and downwind variables are set using the dsp_up_ang and dsp_dn_ang values in Deckman (Instrument Control, Settings).
3. We have added an output that changes from tack to tack. When sailing on Port Tack, the display will show Target Boat Speed. When we tack over to Starboard, the display will change and show Forestay. This is controlled from the Wind Angle.

The format for the context switching is: /upwind section [/reach section] [/downwind section] [/start section], the sections within [brackets] are optional. If information is not specified for all of these sections, the information for the upwind section will be repeated for all missing sections.

It is also possible (though uncommon) to combine these functions, the following line would oscillate the Jib Car variables upwind and display TWA when either reaching or downwind with Timer in the pre-start:

```
A2 /JIB_CAR_P 116 2 JIB_CAR_S 117 2 /TWA 16 0 /TWA 16 0 /TIMER -1 0
```

Note: *All items output from the userout.d file are output to the network on Node 10.*

Deckman Output configuration files

Data output to Deckman

dmnvars.d

This file defines which variables are output to Deckman.

```
88
18
17
91
2

89
10
11

88
18
17
91
2

22
23
90

77
78
79

88
18
17
91
2

24
16

88
18
17
91
2
```

Each line defines a single variable that is output to Deckman; the operation of this file varies slightly depending on whether the communication with Deckman is serial or Ethernet.

For Ethernet communication each unique item in the file is sent at the rate detailed in ethernet.d (10Hz by default), additional repeated variables are ignored.

Settings control in Deckman

setting.d

This file defines the settings which are controllable from Deckman, these values are fixed in the source code so must not be changed. It is not necessary to modify this file directly.

mast_height	S	0	5	1	107
leeway_cal	S	1	4	1	0.0
heel_enable	S	2	4	0	1
gyro_enable	S	3	4	0	1
mag_var	S	4	5	1	0.0
dsp_time	S	5	4	0	30
dsp_up_ang	S	6	4	0	80
dsp_dn_ang	S	7	4	0	120
TWS_factor	S	8	4	2	1.00
mast_rot	S	9	4	0	0
3D_enable	S	10	4	0	1
3D_damp	S	11	5	3	0.970
sel_comp	S	12	4	0	2
sel_heel	S	13	4	0	1
sel_trim	S	14	4	0	1
sel_speed	S	15	4	0	1
sel_GPS	S	16	4	0	1
log_miles	S	17	5	2	0
bspd_switch	S	18	4	0	0
datalog	S	19	0	0	0

Each line defines a different setting item. All these items can be modified from within the Instrument Control option in Deckman.

Description of each item:

mast_height	Mastheight (feet)
leeway_cal	Leeway calibration value
heel_enable	Use heel in calculations (0 off, 1 on)
gyro_enable	Use gyro in calculations (0 off, 1 on)
mag_var	Magnetic variation (+East, -West)
dsp_time	Time (in tenths of seconds) for oscillating variables (see userout.d)
dsp_up_ang	Angle for switch between upwind and reaching variables
dsp_dn_ang	Angle for switch between reaching and downwind variables
TWS_factor	TWS adjustment value
mast_rot	Use mast rotation (0 off, 1 absolute value, 2 full rotation)
3D_enable	Use 3D gyro correction in heading (0 off, 1 on)
3D_damping	3D damping parameter (DO NOT CHANGE should be 0.970)
sel_comp	Compass select (1, 2 or 3)
sel_heel	Heel select (1, 2 or 3)
sel_trim	Trim select (1, 2 or 3)
sel_speed	Boatspeed select (1 p/s paddle, 2 single i/p, 3 SOG)
sel_GPS	GPS select
log_miles	Total distance travelled, nm
bspd_switch	Boatspeed switching mode (0 heel, 1 mwa)
datalog	Datalog (0 off, 1 on)

Note: *Order of parameters is fixed and must not change*

Each line is the same format, as follows:

Item name	As it appears in Deckman
"S"	Denotes a Setting item
Item ID	Numeric ID, in sequence (fixed in the application, do not change)
Width of field	Width of display field in characters (inc. decimal point)
Decimal places	Number of decimal places required
Value	The value of the setting item

Note: *The standard leeway calculation is:*

$$\text{Leeway} = \frac{K \times \text{Heel}}{\text{Boatspeed}^2}$$

Where K is the leeway constant, as set in the setting.d file.

Calibration control in Deckman

svcats.d

This file defines which variables have calibration control available in Deckman. The following format is the default file, it is flexible for the user to add/delete items as required.

Bspd_port	K	63	4	2	2
Bspd_stbd	K	64	4	2	2
Heading1	K	13	4	1	0
heel1	K	0	3	1	0
trim1	K	57	3	1	0
MHU_angle	K	8	3	0	0
depth	K	67	4	1	0

Each line defines a separate calibration option, format as follows:

Variable Name	As it appears in Deckman
"K"	Denotes a calibration value
Variable Number	The variable number to calibrate (from bg_vars.d)
Width of field	Width of display field in characters (inc. decimal point)
Decimal places	Number of decimal places required
Calibration Type	Sets type: intercept (0), slope (1), inverted slope (2), set value (3)

Note: *There are no actual damping values in this file - these are stored in the relevant *.cal file.*

Changing calibration settings through Deckman that are listed in the svcats.d file only controls the calibration in the first line of the *.cal file.

Additional calibration settings within the file (e.g. a table) must be edited directly within the file itself. The way in which the first line of the *.cal file is altered by Deckman is controlled by the Calibration Type setting within svcats.d. The first line of the *.cal file is always a straightforward linear calibration and Deckman can alter this in four ways

0: Change the intercept - changes the value of the calibrated output when the input is zero but keeps the gradient of the calibration the same. This effectively offsets the output by the amount the intercept is changed by and is commonly used for sensors such as depth, rake or rudder where the zero position may have to be changed.

1 & 2: Change the slope or inverse slope – alters the gradient of the calibration without changing the offset. The inverse slope option is typically used in boatspeed calibration where the slope is normally presented as its inverse in Hz/Knot.

3: Set the value – alters the slope to match the output to the value entered without changing the intercept. This is often used on load sensors where the intercept is known to be zero tonnes at zero volts and then the sensor is attached to a known load for calibration.

Damping control in Deckman

damping.d

This file defines which variables have damping control available in Deckman. The following format is the default file, it is flexible for the user to add/delete items as required.

```
boatspeed  D    2  4  1
heading1   D   13  4  1
gyrohdg    D   55  4  1
course     D   14  4  1
TW_speed   D   17  4  1
TW_dirn    D   18  4  1
```

Each line defines a separate damping option, format as follows:

Variable Name	As it appears in Deckman
"D"	Denotes a damping value
Variable Number	The variable number to calibrate (from bg_vars.d)
Width of field	Width of display field in characters (inc. decimal point)
Decimal places	Number of decimal places required

Note: *there are no actual damping values in this file - these are stored in the relevant *.fil file.*

Ethernet Configuration

network.d

network.d defines the IP address and subnet used by the WTP3.

```
Ipaddress=192.168.0.2
Netmask=255.255.255.0
```

Note: *This file may contain: ipaddress=x.x.x.x & netmask=x.x.x.x as shown above OR ipaddress=dhcp as shown below.*

If it is desired to use IP addresses assigned by an external DHCP server then you should change the file as below.

```
ipaddress=dhcp
```

Note: *4. If dhcp networking is selected, then the operating system will attempt to gather network configuration from a dhcp server. If no dhcp server is found, then the device will fall back to 'zeroconf' networking, where an un-used ip in the 169.254.x.x range is used.*

ethernet.d

ethernet.d defines the settings for the data transmission on Ethernet.

```
networkON      1
UDPfrequency   10
UDPprotocol    1
multicastaddr   234.1.1.1
multicastport   5602
```

Each line defines a separate item, format as follows:

networkON	Use Ethernet communications (1)
UDPfrequency	Sets the frequency (Hz) that data is sent to Deckman on Ethernet (max 10)
UDPprotocol	1
multicastaddr	The network address that the WTP3 data is sent from (default value shown)
multicastport	The port used for WTP3 data (default value shown)

Note: Many PCs will require firewall settings to be altered to allow WTP3 UDP multicast data to be accepted on port 5602.

Polar Tables

navpol.d; perfpol.d

These files contains the polar table information used within WTP3, it is not normal to modify these files directly - they are modified when the table is altered in Deckman to suit your boat.

	v1	a1	v2	a2	v3	a3	v4	a4	v5	a5
2.0	1.45	70	1.48	80	1.48	110	1.29	125	0.58	180
4.0	2.66	60	2.73	70	2.96	110	2.56	130	1.28	180
6.0	5.40	51	6.04	70	6.09	110	5.56	138	3.65	180
7.0	6.57	50	6.78	70	6.91	110	6.34	138	4.16	180
8.0	6.91	50	7.42	70	7.52	110	7.18	138	4.74	180
10.0	7.14	48	8.08	70	8.29	110	8.04	138	5.31	180
12.0	7.53	46	8.42	70	8.80	110	8.43	142	5.95	180
14.0	7.76	45	8.67	70	9.14	110	8.70	146	6.46	180
16.0	7.94	44	8.87	70	9.54	110	8.99	149	6.91	180
20.0	8.23	43	9.20	70	10.32	110	9.46	156	7.74	180
25.0	8.25	43	9.28	70	10.37	110	9.50	157	7.81	180

True Wind Correction Tables

adjwa.d; adjvt.d

These files define the corrections applied to True Wind Angle (adjwa.d) and True Wind Speed (adjvt.d) data. The format of both files is the same, it is not normal to modify these files directly - they are updated when the table is altered in Deckman.

	v1	a1	v2	a2	v3	a3
2.0	0.0	60	0.0	90	0.0	130
4.0	0.0	60	0.0	90	0.0	130
6.0	0.0	50	0.0	90	0.0	130
8.0	0.0	45	0.0	90	0.0	130
10.0	0.0	40	0.0	90	0.0	130
12.0	0.0	40	0.0	90	0.0	130
16.0	0.0	40	0.0	90	0.0	130
20.0	0.0	40	0.0	90	0.0	130
24.0	0.0	40	0.0	90	0.0	130
28.0	0.0	40	0.0	90	0.0	130
32.0	0.0	40	0.0	90	0.0	130

Log Mileage

boatlog.d

This file simply contains the total mileage travelled, the value can be modified using settings in Deckman's Instrument Control dialogue.

CAN Baud Rate

can.d

can.d file is pre-loaded on the WTP. As standard the Baud Rate is set to 250 for all channels on the network.

If you wish to use different Baud Rates for channels then you can create a can.d file as shown below where you can set different rates for each channel.

Channel 0	250
Channel 1	125
Channel 2	500

Note: Baud Rate speeds should be, 125, 250, 500 or 1000 (KBits per second).

See system architecture for details.

Data Logging

datalog.d

See: Data Logging

Optional Files (Advanced)

Pre-Start True Wind Correction Tables

adjstrwa.d; adjstrvt.d

These files define the corrections applied to True Wind Angle (adjstrwa.d) and True Wind Speed (adjstrvt.d) data when the WTP3 is in pre-start mode. The format of both files is the same as the normal correction tables.

If these files are not present the normal tables will be used at all times (this is the default setting). The pre-start wind files can be loaded and modified in Deckman using the Adjust start wind angle and Adjust start wind speed options in the Start screen menu.

NMEA output

nmeaout.d

When present this file controls the NMEA output from the WTP3. A typical use for this file is to output wind and other instrument data onto another NMEA enabled device such as a chartplotter. It is possible to define multiple NMEA out ports in the system. For each NMEA out port you can define multiple NMEA sentences to be transmitted and their frequency.

The possible NMEA out sentences are:

GLL	Geographic position
VTG	Course and speed over ground
VHW	Boatspeed and heading
MWD	True wind direction and speed
VWR	Apparent wind speed angle
VWT	True wind speed and angle
MTW	Sea temperature
XDR	Transducer measurements
HDG	Heading
MWV_T	True wind speed and angle
MWV_A	Apparent wind speed and angle

```
Channel 0
Device 0
Com 2 NMEA0183 4800 N 8 1
```

```
GLL 10
VTG 10
VHW 5
MWD 5
VWR 5
VWT 1
MTW 1
XDR 1
HDG 1
```

```
Channel 1
Device 0
Com 1 NMEA0183 4800 N 8 1
```

```
GLL 1
VTG 2
VHW 2
MWD 1
```

```
Channel 1
Device 1
Com 2 NMEA0183 4800 N 8 1
```

```
GLL 1
MWD 1
```

Note: Multiple outputs can be listed in one file. The proceeding number following the NMEA sentence denotes the frequency at which the WTP will transmit that NMEA sentence. Sentences can be transmitted at 10, 5, 2 or 1 Hz.

Fast Serial Output

fastout.d

This file enables a high-speed serial output containing the variables listed in the file, as an example the file below would output the following string:

```
Channel 1
Device  0

Com 1 RS232 57600 N 8 1

93 x 5 1
94 y 5 1
13 a 5 1
55 b 5 1
80 c 5 1
51 d 4 1
81 e 4 1
53 f 4 1
82 g 4 1
92 h 4 2
16 i 4 0
2  j 5 2
```

Line 1 defines the COM port settings:

COM Port	COM port used
Baud Rate	Baud Rate setting to suit the input [4800, 9600, 19200, 28800, 38400, 57600, 115200]
Parity	Parity setting to suit the input[N = no parity]
Data Bits	To suit the input [7 or 8]
Stop Bits	To suit the input [1 or 2]

The additional lines control the output variables, as follows:

Variable WTP3	variable number
Tag Character	A unique character to identify the output variable
Field Width	The width of the output field
Decimal places	Number of decimal places required on the output

From the above file, the WTP3 will transmit data in the following format:

```
1140700826.78,18178469,-467165,18178472,-467167
j0.00
i28
h0.0
g0.0
f-9.7
e0.0
d1.1
c0.0
b8.9
a0.0
y0.0
```

Note: Multiple outputs can be listed in one file

The output file is made up as:

```
timestamp,lat1,long1,lat2,long2  
id variable  
id variable  
etc
```

Where:

timestamp	seconds (since 1 Jan 1970)
lat1	GPS1 Latitude (Degrees x 360000), bow position
long1	GPS1 Longitude
lat2	GPS2 Latitude
long2	GPS2 Longitude

Decoded example:

1140700826.78	23/02/2006 13.20
18178469	GPS1 Latitude: 50° 29.744' N
-467165	GPS1 Longitude: 1° 17.861' W
18178472	GPS2 Latitude: 50° 29.75' N
-467167	GPS2 Longitude: 1° 17.861' W

Serial Loadcell Configuration

loadcell.d

The addition of this file allows the WTP3 to accept serial inputs from loadcell systems which are generally used for large numbers of loadcells. If a small number of loadcells are installed it is normal to use an analogue device and to connect to analogue inputs on the WTP3.

The loadcell input accepted via *loadcell.d* is as follows (additional loadcells increment n as their identifier):

```
#n[+-]XXX.XX<CR><LF>
```

```
Channel 1  
Device 0  
  
Com 1 rs232 9600 N 8 1  
  
forestay 0 58 forestay.cal null.fil null.alm
```

Note: *Multiple outputs can be listed in one file*

Line 1 defines the COM port settings:

COM Port	WTP3 COM port used
Baud Rate	Baud Rate setting
Parity	Parity setting to suit the input (usually 'N' for no parity)
Data Bits	7 or 8 to suit the input
Stop Bits	1 or 2 to suit the input

The additional lines control the input variables, as follows:

Variable Name	variable name for reference
Sentence ID	"n" in the example sentence above
Variable Number	Variable number where data is added to (from bg_vars.d)
Calibration File	Filename of the calibration file to use
Filter File	Filename of the filter/damping file to use
Alarm File	Filename of the alarm file to use [optional]

Fastnet data

fastnet.d

Allows for the provision of data on fastnet to be received by the WTP3, calibrated, filtered and mapped to a variable in bg_vars. For example, rudder angle or loadcells.

An example of this is shown below.

Rudder	D7	18	49	rudder.cal	rudder.fil	null.alm
--------	----	----	----	------------	------------	----------

Note: The variable name is used for reference only and is not used in the WTP system. The variable name must not contain spaces.

<name> <B&G function number in hex> <node> <variable bg_vars.d> <cal file> <fil file> <alm file>

Example on using the data files

Imagine you wanted to add a linear displacement transducer to your system to tell you the forward or aft position of the mast foot. This would give out a voltage that would need to be fed into an analogue module.

In this example we will add a variable called 'mastfoot' which we will input into an analogue module set to channel 0, device id 1, analogue input channel 3.

In `bg_vars.d` we add a line to define the new variable, in this case the next variable number is 111, so we add the following line setting variable 111 to be mastfoot and having two decimal places.

```
111 mastfoot mst 2 0 0
```

In `sample.d` we need to define the sampling of the analogue channel, so we modify the line for the analogue section to look for channel 0, device id 1, analogue input channel 3 as shown below.

```
mastfoot 0 1 3 111 mastfoot.cal mastfoot.fil null.alm
```

This defines that the function called 'mastfoot' is sampled on analogue module on channel 0, Device 1, analogue Input 3 and mapped onto variable 111 being calibrated and filtered with the listed files with the possibility of an optional alarm file.

At this stage we would like to show the variable on the displays, as an example we will add the function 'MASTFOOT' to the PARAMETER menu. First we need to add the menu item in `usermenu.d`, for this we add one line:

```
MASTFOOT 0112 a1 4
```

This new menu item, 'MASTFOOT', would now be displayed in the 'PARAMETER' menu (0112), using fastnet function number 'a1', the '4' defines the location of the menu item within the item list.

We now need to output the data onto the network so that when you select the item from the menu there is data to display, this is done using the `userout.d` file. In this example we will output the data twice per second, given that there are currently no outputs at this rate we need to add another transmit group to the existing file:

Existing file:

```
1
2 9 7
A8 /CWA_____@ 65 1
A9 /CWS_____KT 66 1
```

Modified file:

```
2

2 9 7
A8 /CWA_____@ 65 1
A9 /CWS_____KT 66 1

1 4 1
A1 /MASTFOOT 111 2
```

As can be seen, in addition to adding the extra transmit group we have also modified the first line of the file to read '2', which identifies the number of transmit groups that follow.

The final thing to do would be to create new calibration and damping files (mastfoot.cal and mastfoot.fil) in the relevant folders with appropriate values, and, if required, add the new variable into damping.d and/or svcals.d to allow damping and calibration from Deckman (see Chapter 5: Parameters for more information on these). If calibration or filtering of the variable is not required it is normal to use null.cal and null.fil respectively as the calibration and damping filenames.

User variables

You are able to create your own data variables, using several methods.

- Taking data from existing variables and applying calibrations and filtering as desired
- 2D variable lookup table
- JavaScript

User Variables are available for distributing to system displays or Deckman in the same manner as a normal variable.

Defining User variables

Whichever method is being used, any new user variables must be defined by adding them to the variable list (bg_vars.d).

In bg_vars.d we add a line to define each new variable, in the example below the next available variable number was 111 so we have added new lines defining variables 111-113 which we will use in the following examples. See bg_vars.d section for further information.

```
108  gps2sva          sv2  0  0  0
109  gps2utc          ut2  0  0  0
110  Variation        Var  2  0  1
111  MA_TWD           MWD  0  0  2
112  TargRudder       TRd  1  0  1
113  TargDagger       TDg  1  0  0
```

The content of the new user variable is then defined via sample.d, the methodology varies depending on how the user variable is being created (standard, lookup or javascript), the following section describes the options.

Option 1: Standard User Variables

Standard User Variables are new variables based on one existing variable (additional variables may be included in the calibration).

In sample.d we add the new User Variable to the [uservars] section:

```
[uservars]
MA_TWD      111  18  null.cal  ma_twd.fil  null.alm
```

Where:

MA_TWD	Name of our User Variable (note in this example we have changed the name from uservar1)
111	Number of our User Variable, from bg_vars.d
18	The variable number that we are basing our new Variable on (18 is TWD in bg_vars.d)
null.cal	Calibration file for this variable (replace with uservar1.cal if calibration is required)
ma_twd.fil	Filter file for this variable
null.alm	file for this variable (replace with uservar1.alm if alarm is required)

Note: To have no calibration, filtering or alarm for a user variable, we use *null.cal*, *null.fil* or *null.alm* as the filenames – WTP3 ignores any file named “null”.

At this stage the User Variable “MA_TWD” will simply be equal to variable 18 (TWD), we modify it by applying calibrations or filters.

For example if we wanted a 10 second moving average of TWD then we would use the following filter file (ma_twd.fil):

```
6      100
```

Where:

6	Filter Type 6, a moving average filter
100	10 seconds of filtering. The value is in 1/10th of a second

Now we have two TWD variables:

- The original TWD variable (18)
- A 10 second moving average of TWD (111)

Option 2: 2D Variable Lookup Table

2D lookup tables allow the user to produce a new variable based on a linear interpolation of two existing variables, for example a design team may have provided an expected rudder angle based on True Wind Angle and True Wind Speed.

The lookup table is defined in a *.tab file which is located in the calibs directory.

In sample.d we add the new User Variable to the [uservars] section:


```
[uservars]

TargRudder 112 -1 targ_rud.tab null.fil null.alm
```

Where:

TargRudder	Name of our user variable
112	Number of our User Variable, from bg_vars.d
-1	A placeholder for consistency with other uservar types (table lookups are always "-1")
targ_rud.tab	Lookup table file in use
null.fil	Filter file (none in use in this example)
null.alm	Alarm file (none in use in this example)

The table file (targ_rud.tab in this example) is in the following format:

```
TWA 16      30      90      150
TWS 17
4           6.0     4.0     4.0
8           5.0     4.0     3.0
12          4.0     2.0     1.5
16          3.5     2.5     1.0
18          3.0     2.5     1.0
22          2.5     1.0     1.0
26          2.0     1.0     1.0
30          1.0     1.0     1.0
```

Where:

TWA 16	is the X Axis Variable name and number defined in bg_vars
TWS 17	is the Y Axis Variable name and number defined in bg_vars
30, 90, 150	are the X Axis Column headers
4, 8, 12, 16...	are the Y Axis Row Header
6, 0...	are the results of the lookup that populate the new user variable

Notes on Interpolation of table data

The output of any user variable lookup table is a 2D linear interpolation of the input values. Below the minimum values entered in the table the result is interpolated to zero on both X and Y axes, however there is no extrapolation above the maximum values in a table (i.e. the output can never exceed the values given by the maximum X and Y values).

Option 3: JavaScript

Advanced users are able to execute a JavaScript and the resultant value returned from the script is stored in the User Variable. Scripts are calculated at 10Hz, prior to the main wind & navigation calculations.

In sample.d we add the new script based User Variable to the [uservars] section:

```
[uservars]

TargDagger 113 -1 dagger.js null.fil null.alm
```

TargDagger	is the name of the variable
113	is the variable number defined in bg_vars.d
-1	A placeholder for consistency with other uservar types (scripts are always "-1")
dagger.js	is the JavaScript file stored in the 'scripts' directory. Filename must end with 'js'
null.fil	Filter file (none in use in this example)
null.alm	Alarm file (none in use in this example)

Return Value

The uservar will contain the result of the Javascript evaluation. Therefore, the Javascript must return a floating point value. The script will return a value to be stored in bg_vars with the keyword 'return'

For example, return x;

Where x is a floating point value.

Read Access to BG_VARS

A user creating a JavaScript has read access to all variables defined in bg_vars. A user can reference a variable and access its value via the object interface 'bgvars'.

For example, bgvars.value(17) will return to the script the floating point value of variable 17.

Errors

Syntax errors in scripts will be reported as soon as a script is evaluated. This will be reported in the terminal window on WTP3. Scripts that do not evaluate correctly will not be executed by the WTP.

Print

Any output from the JavaScript using the 'print' command will be directed to the terminal window on WTP3.

Worked Example

Here we are using JavaScript to provide target daggerboard position based on a fictitious algorithm, the algorithm is:

$$\text{Target Daggerboard} = 0.25 \times \frac{(0.1 * \text{ABS}(\text{TWS})) + (\text{ABS}(\text{Heel})) + (\text{ABS}(\text{Keel Angle}))}{\text{Boat Speed}}$$

The JavaScript required is:

```
var TargDaggerBoard;  
  
TargDaggerBoard = 0.25 * ((0.1 * Math.abs(bgvars.value(17))) +  
  ((Math.abs(bgvars.value(89)) + Math.abs(bgvars.value(59))) /  
  bgvars.value(91)) );  
  
Return TargDaggerBoard;
```

Where 17 is TWS, 89 is Heel Angle, 59 is Keel Angle and 91 is Boat Speed.

Note: WTP3 supports ECMAScript scripting language, as defined in standard ECMA-262. For an overview of ECMAScript, see the ECMAScript reference. If you are not familiar with the ECMAScript language, there are several existing tutorials and books that cover this subject, such as *JavaScript: The Definitive Guide*.

Parameters

Note: It is only recommended that advanced users alter the parameter files directly as described in this chapter. Most simple calibration, damping etc. can be controlled from Deckman as described in Chapter 2: Basic Operation.

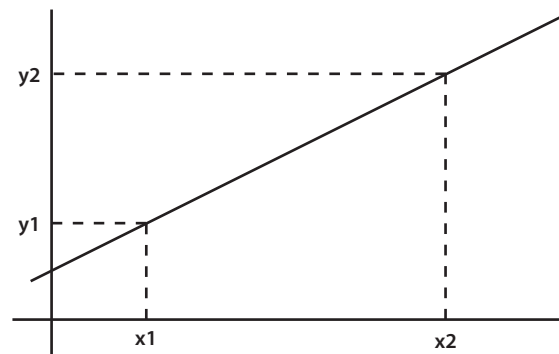
Calibration

Each variable requiring calibration has its own calibration file (.cal file extension), all calibration files are located in the Calibs directory.

Various calibration functions are available to the WTP3 but most of them are variations on linear as in the diagram.

The first parameter describes the type of calibration:

ID	Calibration Type
0	Null Calibration
1	Ordinary Linear
2	Linear for 0° to 360°
3	Linear for -180° to 179°
4	Table
5	Table for 0° to 360°
6	Table for -180° to 180°



The next two numbers are x1, y1; and the final two are x2, y2. x is the independent variable (which is actually a voltage or pulses per second for boat speed and wind speed) and y is the Dependent variable that we need to calibrate.

We will use the example of calibrating a compass. A possible simple calibration file (heading.cal) might look like:

```
2 0 20 5 25
```

This would add a 20° offset, it is unlikely that you would ever have to apply such a large offset to a compass the large numbers are just to illustrate the functionality below.

Functions 4, 5 and 6 are more complicated. The first line of the file is the same as for calibrations 1, 2 and 3; after this you create one or more tables to further calibrate the variable, and these operate on whatever the output is from the first line. The first way that this can be done is with one simple table of corrections.

Advanced calibration example 1

A sample file might look like:

```
5 0 20 5 25  
  
Table -1 +  
0      10  
180    -10  
360     10
```

The first line of the file still works the same as before but the result is then further calibrated from the table. The -1 following the word Table indicates that the corrections are applied directly to the output from the initial calibration. Next, the addition sign (+) after the -1 indicates that the corrections in the table are to be added. Then, the numbers in the left column indicate values of the incoming data, while the right column indicates the correction to be applied, with interpolation for values between the defined values. The table below indicates the result of this two-stage calibration:

ORIGINAL DATA	RESULT OF FIRST LINE	RESULT AFTER TABLE
0	020	028
90	110	108
180	200	192
340	000	010

You can see the interpolation for headings between those at which the corrections are specified and that the calibrations in the table are applied to the output from the first line of the file rather than the original input.

It is also possible to calibrate your variable with respect to another variable in the WTP3 database.

Advanced calibration example 2

For example, you could enter:

```
5 0 20 5 25

table -1 +
0      10
180    -10
360     10

table 0 +
-20    10
0       0
20     10
```

The first line and first table of this are identical to above, but the outcome of the first table is then further modified by the second table. In the example, the -1 after the word table indicates that the correction was applied to the variate itself. Entering any other number after the word table means that we are calibrating with respect to another variable in the WTP3's database, with the variables referred to by the bg_vars identification number (see bg_vars.d).

In the example above, the 0 refers to the bg_vars identification number for Heel. The second table therefore applies corrections depending on the angle of heel: the first column is the angle of heel and the second is the correction to be applied to the compass. The result of the second table in the above example would be:

INPUT FROM 1ST TABLE	ANGLE OF HEEL	RESULT AFTER 2ND TABLE
50	30	65
50	10	55
50	0	50
50	-20	60
50	-30	65

Here, the offset to the compass heading is altered by the heel angle. Of course, the corrections in the previous table will continue to be applied before the corrections with respect to heel.

Here you can see that, as well as interpolating within the calibration points you enter, the WTP3 will also extrapolate outside them.

Advanced calibration example 3

It is also possible to multiply, subtract and divide in your corrections. For example, a table to alter boat speed with respect to angle of heel might look like:

```
4  0  0  1  1

table 0 *
-40  0.95
-30  0.95
-25  0.975
-20  0.99
-15  1.0
15  1.0
20  0.99
25  0.975
30  0.95
40  0.95
```

This table is, therefore, taking the standard input from boat speed (which would be specified in the `bg_vars.d` file) and applying a correction based on angle of heel (`bg_vars` identification number 0 after table). So far, this is operating the same as the example above. Next, however, we have a multiplication sign (*), which indicates that boat speed is to be multiplied by the values in the table. Then the table works as before for calibrating with respect to another variable: the left column indicates the value at which the calibrations to be applied while the right column is the multiplication factor.

The example above would act to reduce boat speed with increasing angle of heel.

Other identifiers and operators

As well as the word `table`, it is also possible to use two other identifiers:

Variable	The following number refers to the <code>bg_vars</code> number and a mathematical symbol indicates what operation is to be performed.
Constant	To specify a constant value to use in the calculation; a mathematical symbol indicates what operation is to be performed.

There are also a number of mathematical operators that you can use:

+	Add
-	Subtract
*	Multiply
/	Divide
=	Assigns a value to the variable you are calibrating

Calibration example 4

The line

```
constant 3.3 *
```

would mean that we are multiplying by a constant value of 3.3.

Calibration example 5

As an example we will attempt to recreate the leeway calculation that WTP3 does as standard.

The standard calculation is based on the formula:

$$\text{Leeway} = \frac{K \times \text{Heel}}{\text{Boatspeed}^2}$$

Refer to the setting.d file for more information.

Let us suppose we wish to recreate this but artificially limit heel to 25 degrees and using a leeway constant of 6.4. The file (leeway.cal) would look like this:

```
4 0 0 1 0

table 89 =
-30 -25
-25 -25
 25  25
 30  25

constant 6.4 *

variable 91 /
variable 91 /
```

The first line of this is taking the input from sample.d for Leeway and ensuring it is set to 0, the output of the first line is 0 no matter what the input (refer to example 1 above). The table then takes the value of Selected Heel (bg_vars variable 89) and the equals signs means that we are assigning values for leeway based on heel.

The effect of this table would be that, for Heel values up to 25, the value assigned would be exactly the same as the Heel angle. Above 25, the assigned value will stay at a constant of 25 since when the system will interpolate between 25 and 30 the assigned value remains 25, and beyond 30, the extrapolation will still give the value 25. The next line will multiply by the leeway constant of 6.4 and each of the final two lines will divide by selected boat speed (bg_vars variable 91) to complete the formula.

Summary

The following provides a summary of the operation of the calibration tables:

- The calibrations are applied sequentially, so that those specified first in a file will be applied before those specified later.
- Identifiers recognised are table, constant and variable.
- Number '-1' indicates that the calibrations are applied directly to the variable.
- Any other number indicates that the calibration is with respect to a different variable in the WTP3 database, with the number being the variable number from bg_vars.d.

Sensor Calibration

Rate Gyros

During assembly the output of each channel is measured as mV per degree per second. So if, for example, the measured response of the gyro was 111.1 mV/degree/sec then a 1.0 volt input would indicate a pitch or roll rate of 9.0 °/s. A reading from the analogue board of 0V indicates a rate of 0 °/s, therefore appropriate calibration values would be 0.0, 0.0) and (1.0, 9.0).

There should be no need to change the pre-set values, unless you want to see the effect of removing one or more of the sensors.

Boat Speed

This calibration is expressed as Hertz per knot (Hz/kt) so for a calibration of 3.50 Hz/kt enter (0.0, 0.0) and (3.50, 1.0). Both port and starboard should be calibrated.

1	0.00	0.00	3.50	1.00
---	------	------	------	------

Wind Speed

The B&G systems use $W = (1/A)H + B$ where A is calibration in Hz/kt, H is the anemometer frequency and B is an offset. So for the standard of A=1.04 and B=1.04 the equivalent WTP3 calibration is (0, 1.04) and (10.4, 11.04).

1	0.00	1.04	10.4	11.04
---	------	------	------	-------

Mast Rotation

The pre-set value for this is for it not to be used. It is straightforward to use if you have the facility, and is a requirement for accurate wind data if you have a rotating mast.

The options are for mast rotation correction to be off (0), in absolute value mode for mast twist (1) or in mast rotation mode (2) for fully rotating masts.

If the mast rotation correction is used (set to either 1 or 2) then it is necessary to have a suitable input configured for the mast angle information on variable number 25.

Filtering and Damping

Damping Types

Like calibration, all the variables that require filtering have their own filter file in the filters directory. The various damping functions are specified by the first number in the damping file (ID in the table below).

ID	DAMPING TYPE	NOTES
1	Ordinary exponential	One parameter: inverse of required damping time in secs/10
2	Exponential, for 360°	One parameter: inverse of required damping time in secs/10
3	Exponential, for 180°	One parameter: inverse of required damping time in secs/10
4	Two term Kalman filter	Refer to B&G
5	Band pass - mainly for rate gyros	Refer to B&G
6	k term moving average	Refer to B&G
7	3rd order Chebyshev low pass; ripple fraction 0.1	One parameter: damping in secs
8	As 7 for 360	One parameter: damping in secs
9	As 7 for 180	One parameter: damping in secs
10	3rd order Chebyshev band pass; fixed coefs	DO NOT CHANGE: used for rate gyros, see below
11	Non-linear	See explanation below
12	Non-linear, for 360°	See explanation below
13	Non-linear, for 180°	See explanation below
14	RMS calculation	Root mean square: e.g. for calculating wave amplitude
15	Period calculation	Period calculation: e.g. for calculating time between waves
16	Ordinary exponential dependent damping	Table with independent variable and inverse of required damping time in secs/10
17	Exponential dependent damping, for 360°	Table with independent variable and inverse of required damping time in secs/10
18	Exponential dependent damping, for 180°	Table with independent variable and inverse of required damping time in secs/10

Damping example 1: Exponential Damping

Damping functions 11, 12 and 13 are exponential functions that will cause the data to move more quickly if the difference between the new data and the last value moves outside a bound.

The first damping number in the filter file is as for functions 1-3 (i.e. - inverse of required damping time in secs/10); the second specifies the bound - outside this value, the damping becomes less until at 8 times the bound value there is almost no damping at all. These functions are particularly useful for boat speed and heading when coming out of a tack.

For example we might use non-linear damping on our heading so that when it is changing rapidly after a manoeuvre it is less damped than when we are sailing a steady course. A typical filter file would look like:

```
12 0.1 4
```

This gives a damping of 1 second in normal use (inverse of 0.1 divided by 10); however, when difference between the new data and the last value is greater than 4°, the damping gradually reduces until at 32° difference, no damping is applied.

Damping example 2: Dependent Damping

Damping functions 16, 17 and 18 are exponential functions similar to functions 1, 2 and 3 except that the time constant for the damping can be determined with respect to another variable in the WTP3 database.

During a manoeuvre True Wind Direction (TWD) can be unsteady. If we wish to use some damping to display TWD more steadily based upon data from before the start of the manoeuvre then we could filter it based on the value of Yaw Rate using damping function 17 (Exponential 360°). To do this we could create the following filter file (twd.fil):

```
17
Table 56
0 0.1
3 0.02
6 0.0125
9 0.01
10 0.01
```

The first line of the file indicates the damping type to be used. In the example, the 56 after the word Table indicates that the damping will be calculated with respect to Yaw Rate as 56 is the variable number for gyro_dyw (see bg_vars).

Therefore, in this example, the table controls the time constant for the exponential damping depending on the Yaw Rate of the vessel: the first column is the Yaw Rate and the second is the time constant to be used (as in functions 1,2 and 3 i.e. inverse of required damping time in secs/10). This file would result in the following damping being applied:

YAW RATE (°/S)	DAMPING (S)
0	1
3	5
6	8
>9	10

Wind

To fully understand the filtering of the wind functions it is necessary to consider the order in which WTP3 calculates the various functions and where filtering is applied.

When the wind is measured it is initially corrected for masthead unit offset and mast rotation (or twist), then the rate-gyro corrections for pitching and rolling are applied and then the triangulation with Boat Speed is done and Course added to get the Original Wind speed and Wind Direction.

The adjustment tables for wind shear and gradient are applied to get True Wind speed and True Wind Direction and then these variables are filtered. From these filtered variables, the True Wind Angle is calculated and a back triangulation is done to calculate Apparent Wind Speed and Apparent Wind Angle.

Therefore, the order of wind calculations is:

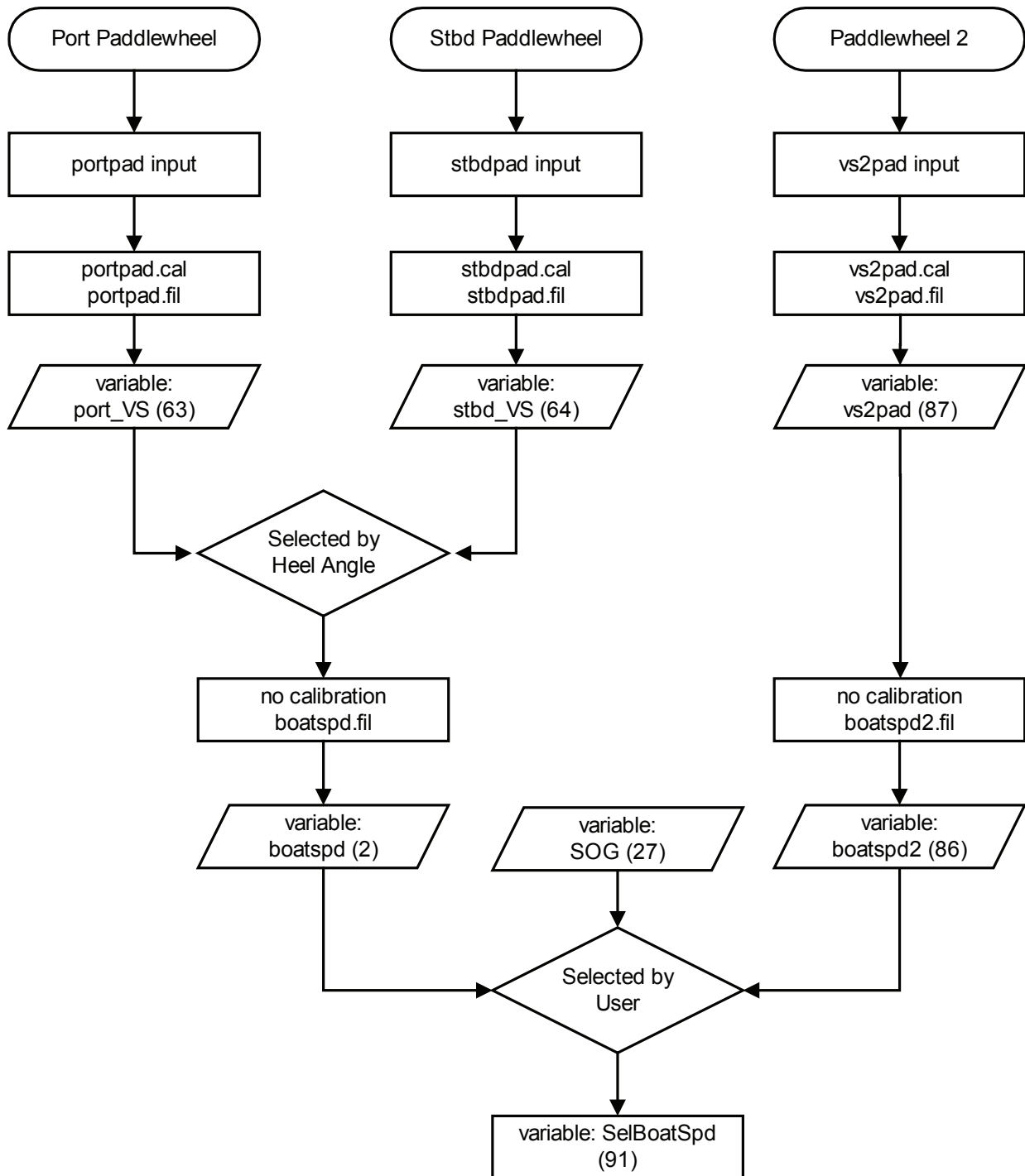
measured wind ► corrected wind ► original wind ► true wind ► apparent wind

It is the data from the last two steps of the calculation that you actually see on the displays. The various stages in the calculation can be seen in more detail in Wind Calculation flowchart.

Sensor Damping

Boat Speed

To understand the filtering of the boatspeed functions it is necessary to consider the order in which WTP3 calculates the various functions and where filtering is applied, this is shown in the flowchart below:



Note: When shipped from the factory, Boat Speed is shown to two decimal places. However, due to the extra responsiveness of the WTP3 system, especially when tacking, it may be desirable to change the displayed value to one decimal place in the fixout.d file.

Rate Gyros

The rate gyro filters are specified in `gdheel.fil`, `gdpitch.fil` and `gdyaw.fil` for heel, pitch and yaw respectively.

The rate gyros are susceptible to drift and so a band-pass filter used. The values in these files should not be altered. They should read:

5 0.5 .001

Whenever the WTP3 is switched on, the measured Pitch and Roll are likely to have values that are well away from zero and it will take 15 minutes or so for the numbers to settle down.

This has an enormous effect on wind speed and angle but is perfectly normal. The WTP3 therefore ignores the inputs from the rate gyros for 15 minutes after power on.

Data Logging

WTP3 will log data internally in the form of a SQLite database. To enable and configure data logging it is necessary to add a datalog.d file to the current config directory. This file contains the criteria to start and stop the logging of data at desired frequencies.

An example of the datalog.d file can be found below. The first line 'START' is the trigger to start data logging. START parameters are: Racetimer (starts logging on activation of the race timer), Alarm (starts logging on the trigger of any alarm), POWERON (Starts logging as soon as the WTP is switched on or Off (Disables the data logging file)).

The second line 'STOP' is the length of time you wish to log data for. Data can be logged in minutes or seconds. To set the length of time write the word Minutes or Seconds followed by a space and the number of minutes or seconds you wish to log data for.

Note: A 'STOP' line is only required if you wish to have a time limit on the data log function. If no 'STOP' line is entered then the WTP will continue to log data until it is stopped via Deckman, the WTP is switched off or the data log transfer procedure is started to USB stick.

The logging of data from the CPU is flexible and variables are separated into sections [10Hz], [5Hz], [2Hz] & [1Hz]. For each section it is possible to list multiple variable numbers from bg_vars you wish to log.

Example: datalog.d

```
START racetimer
STOP minutes 70

[10Hz]
111
112
113

[5Hz]
0
2
8
9
13
16

[2Hz]
19
22
23
24

[1Hz]
19
27
28
32
34
```

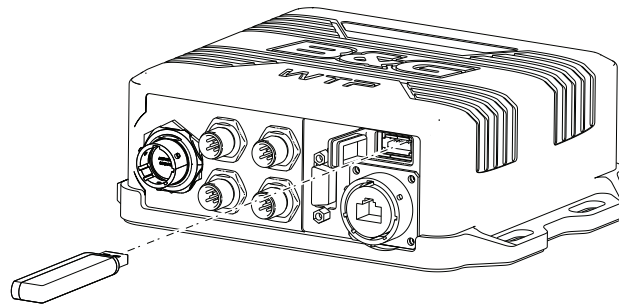
Exporting the .sqlite file

The data log is an SQLite database. For reference on SQLite go to www.sqlite.org

The .sqlite file that is stored on the CPU can be exported by copying the file via the FTP to your PC or auto transfer by inserting a memory stick.

Exporting .sqlite file via Memory Stick

1. Ensure WTP CPU is on
2. Plug a memory stick into the USB port



3. As soon as the memory stick is inserted into the CPU it will transfer the LOG file to the USB stick. If the CPU is in the middle of logging data it will immediately stop logging and begin to transfer the LOG file. During the file transfer process the LOG LED on the top of the unit will go out.

Once complete the LED will turn solid green. If the transfer fails then the LED will turn red. A failure is likely to be caused due to lack of space on the USB stick. Try to free up enough space on the memory stick or use a stick with more memory. Repeat steps 1 & 2.



4. Once complete remove the memory stick and transfer to a PC. If you continue to experience difficulties transferring the data log file you can access it via the FTP.
5. To restart data logging the CPU must be restarted or reset via Deckman.

Note: The data log file can be viewed using an sql database browser.

SQLite Database Structure

The database used in WTP3 consists of the following tables

Logs

The logs table contains data that identifies the individual data logging sessions:

ID	unique identifier for each logging session
Start	start date and time of the session
Stop	stop date and time of the session if stopped via duration or manual control. "Power off" stop times are not recorded
Notes	Records assistance text, identifying how the log session was started e.g. "Power On"

Datalog

The datalog table contains the variable data configured by the user in datalog.d

LogID	unique identifier for each logging session, from Logs table
Var	variable number of the data, from bg_vars.d
Calibrated	calibrated value of the variable at the time of logging
Raw	raw, pre-calibrated, value of the variable at the time of logging
Time	date and time of the sample
PositionID	unique identifier for each position log, from Tracks table

Tracks

The tracks table contains a record of positions recorded at 1Hz. Each data log record contains a positionID field to allow cross reference of each data log with a position.

PositionID	unique identifier for each position log
Latitude	latitude value in seconds (Degrees * 360000)
Longitude	current longitude value in seconds (Degrees * 360000)

Variables

The variables table contains a lookup of the name of each logged variable from bg_vars.d

Var	variable number from bg_vars.d
Name	name of variable from bg_vars.d (e.g. "TW_speed")

Alarms

The alarms table contains a history of alarms raised.

AlarmID	unique identifier for each alarm raised
Var	variable number from bg_vars.d
Raised	DATETIME
Value	real

Alarms

WTP3 provides an alarm output via the digital output terminals in the Serial Interfaces.

In the case of an alarm being triggered all digital outputs which are configured to Alarm output will be enabled, Displays will show an alarm on the assigned function. Alarms are cancelled via the display, or if the value drops outside the alarm range.

Configuring Alarms

Configuring a function to use an alarm file

Alarms are configured by assigning an alarm file to a function in any configuration files where .cal and .fil files are specified (e.g. sample.d).

The following example shows a nmeain.d file, used for NMEA0183 input of Depth data, with the correct format for calling a depth alarm file.

```
Channel 0
Device 7

Com 1 NMEA0183 4800 N 8 1

SDDBT 1 67 depth.cal null.fil depth.alm
YXMTW 1 60 seatemp.cal null.fil
```

If no alarm is required on a function the alarm file reference should be set to null.alm (a non-existent file)

File Structure

Alarm files are of the following format, any Alarm files are stored in the alarms directory, all alarm files have the extension .alm

The following file is an example alarm file for a depth alarm.

```
Depth      67    C1
Hi         40.0  Off
Lo         2.5   On
```

Where:

Depth	Text label for user information only, it is not used by the system. Must not contain spaces.
67	The function number to apply the alarm to (from bg_vars.d). In this example 67 = Depth.
C1	Fastnet function used to display the alarm on displays (hexadecimal). C1 = Depth.
Hi 40.0 Off	Configures a High ("Hi") alarm, at a value of 40.0 which is currently disabled ("Off")
Lo 2.5 On	Configures a Low ("Lo") alarm, at a value of 2.5 which is currently enabled ("On")

Either the Hi or Lo lines may be omitted from the file, so if only the low value alarm was required the following

file content is valid:

```
Depth 67 C1
Lo    2.5 On
```

Note: Ensure that the Fastnet function number (C1 in the example) is output in either `fixmenu.d` or `usermenu.d`, if it is not present then the function will not be available on a display, so cannot be cancelled via display.

Advanced Alarms

Use of Javascript

Advanced users can use Javascript to construct more complex alarm criteria. The JavaScript can be used for defining complex alarm conditions for each 'High' and 'Low' condition. The JavaScript must return a Boolean 'true or false' to indicate if the alarm condition passes or fails. A value of false indicates the alarm should be raised. The JavaScript has read access to all the variables defined in `bg_vars.d` in its evaluation.

A JavaScript file is referenced from the `.alm` file in the following format:

```
TWS 17 55
HI -script highwind.js ON
```

Where:

-script highwind.js	References the JavaScript 'highwind.js' stored in the Scripts directory.
Hi	High Alarm
On	Alarm enabled. If OFF the alarms is disabled and the JavaScript is not evaluated.

```
var TWA = bgvars.value(16);
//Define variable TWA and Get latest value from bg_vars

var TWS = bgvars.value(17);
//Define variable TWS and Get latest value from bg_vars

TWA = Math.abs(TWA);
//Absolute value of TWA to deal with -ve value for port

if( (TWS > 20) && (TWA > 150) ) //Condition

    return false;          //FAIL - Raise the alarm

else

    return true; //PASS - Alarm Passes.
```

Installation

Physical Installation

Processor

The WTP3 unit should be installed in a dry place with easy accessibility. The enclosure is water resistant to IP67 but will not survive prolonged immersion.

The engine box is NOT a good place to install your instrument system processors; it is hot and electrically noisy.

The WTP3 unit does not contain orientation sensitive components so it is NOT necessary to mount the unit vertically, however it is recommended to orientate the unit with all cable exits downwards.

Rate Gyro Box

The Rate Gyro box should be orientated as carefully as possible along the fore and aft axis of the yacht and in the horizontal, with the cable gland facing forwards, as indicated by the arrows. If you do not fix it down initially you will be able to check that the wiring is correct by rotating the box along the fore and aft and athwartships axes and seeing that the roll, pitch and yaw rate values are updated correctly. These rate values are displayed in the 'PARAMETR' menu by default.

Pitch is taken to be positive when the top of the mast is swinging forwards; roll is taken to be positive when the top of the mast is rolling from starboard to port; yaw is taken to be positive when the boat is moving in an anti-clockwise direction (i.e. turning to port).

Heel

Heel should read positive on starboard tack (i.e. with the boat heeling to port).

The Heel sensor (B&G part 690-00-004) should be installed on an aft facing bulkhead.

Fastnet Network Installation

General Layout

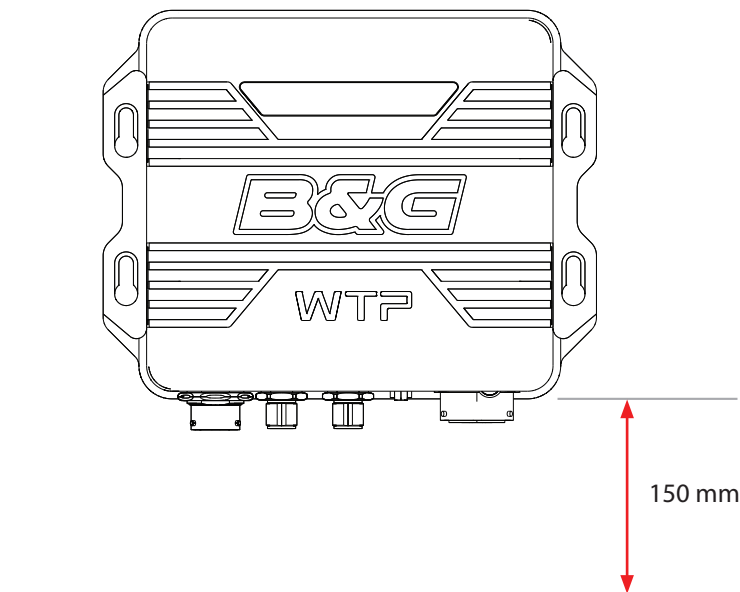
The Fastnet network cabling should be such that the network cable run is predominantly in a linear layout with a definite start point and end point (which are terminated). "Star" shaped layouts are inefficient, may cause incorrect operation and should be avoided.

Mounting Instructions

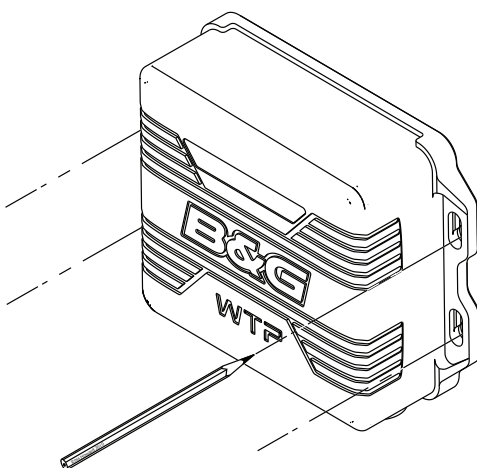
WTP3 CPU

Step 1: Find a suitable location to position the CPU.

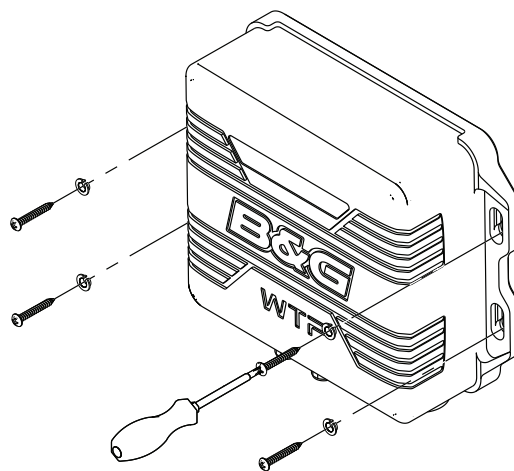
Mount the CPU vertically. Ensure that there is at least 150mm clearance between the connectors and any surface to enable easy access to cables



Step 2: Mark the hole positions



Step 3: Drill pilot holes and fix into position with 4 self tapping screws and split ring insulating washers

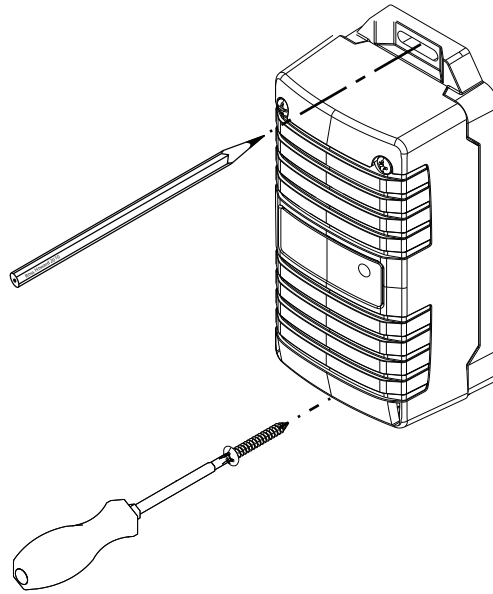


WTP3 Module

Step 1: Find a suitable location to position the module.

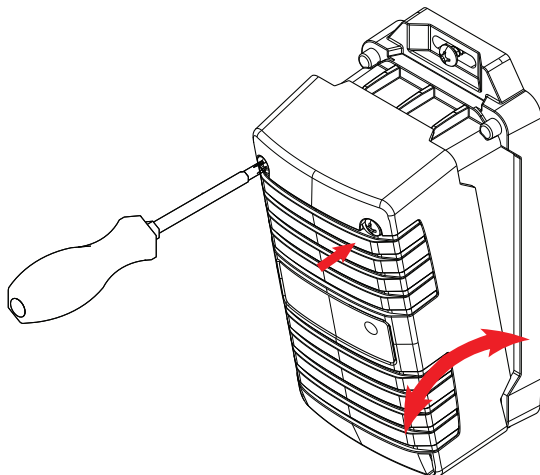
Mount the module vertically. Ensure that there is at least 100mm clearance between the connector and grommet and any surface to enable easy access to cables

Step 2: Mark the hole positions, drill pilot holes and fix into position with 2 self tapping screws



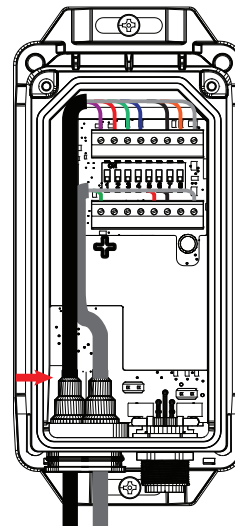
Step 3: To remove the lid, unscrew the 2 lid screws.

Note: The lid hinges at the bottom edge of the module. To close the lid you need to locate the bottom edge first.



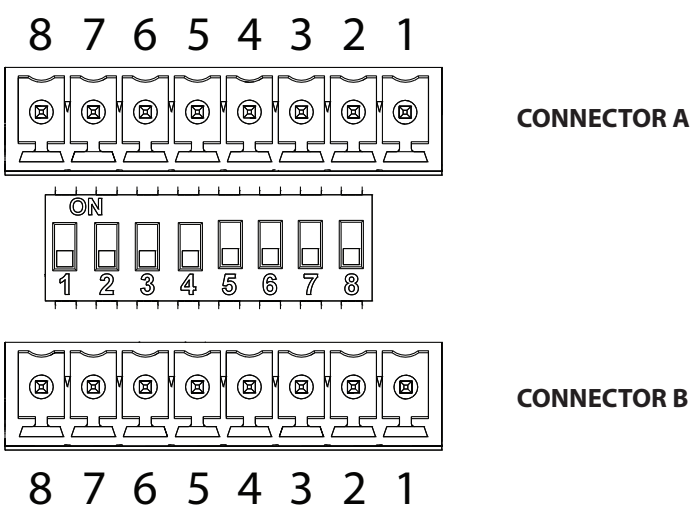
Step 4: Cut the grommet indicated to a length that creates a tight fit for the cable to be fed. Feed the cable through and wire into the module plug.

Ensure the cables are positioned to the left of the centre pillar and the plug can reach the connectors without placing tension on the wires



Module Wiring

The analogue and serial module connector terminals have differing functions. Below is a list of the connectors their terminal numbers and their functions.

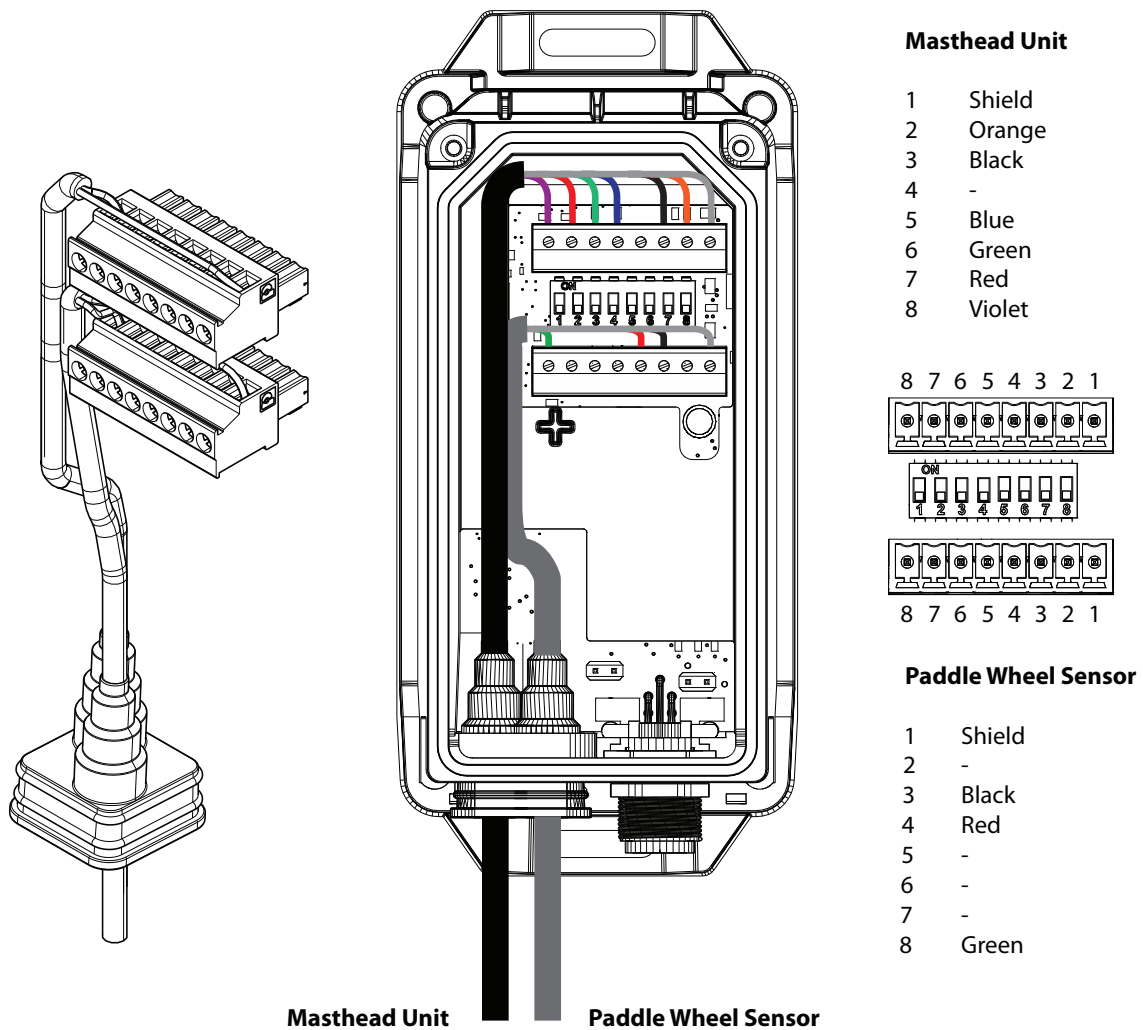


ANALOGUE MODULE		
TERMINAL	CONNECTOR A	CONNECTOR B
1	Shield	Shield
2	12V	12V
3	0V	0V
4	5V	5V
5	Analogue 1	Analogue 4
6	Analogue 2	Analogue 5
7	Analogue 3	Analogue 6
8	Pulse 1	Pulse 2

SERIAL MODULE		
TERMINAL	CONNECTOR A	CONNECTOR B
1	Shield	Shield
2	5V	12V
3	0V	0V
4	Digital	Digital
5	Tx +	Tx +
4	Tx -	Tx -
7	Rx +	Rx +
8	Rx -	Rx -

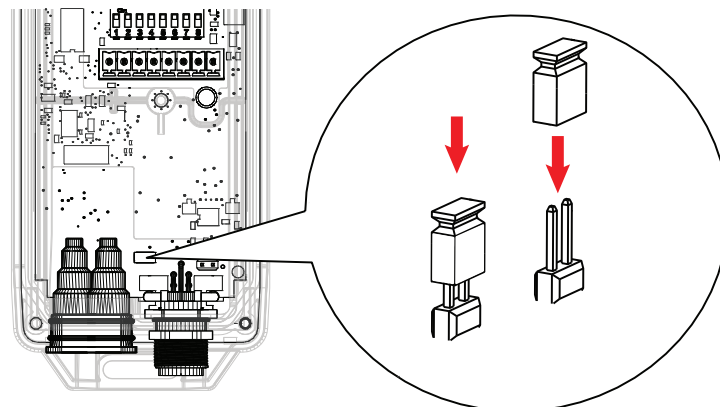
Analogue Module Wiring Example

Below is an example of how to wire a analogue module with a masthead unit and paddle wheel sensor.



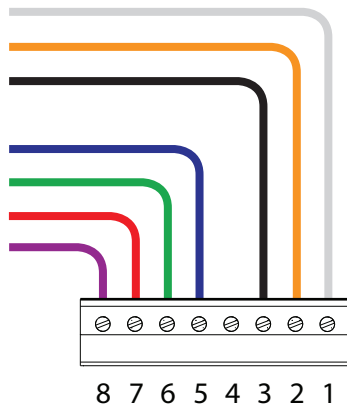
Module Jumpers

If a sensor or combination of sensors connected to an analogue or serial module draw more than 300 mA current total, then a terminal link must be placed onto both jumpers located above the can input, inside the module as shown below. This bypasses the isolated power supply within the module.



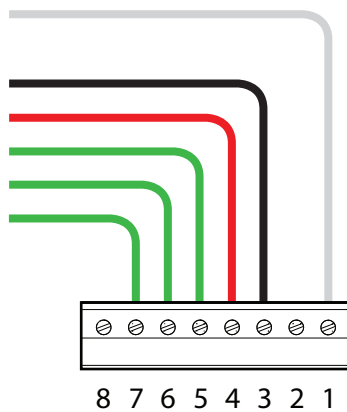
Analogue Module Wiring

Masthead Unit



TERMINAL	COLOUR	INPUT
1	-	Screen
2	Orange	12 Volts
3	Black	0 Volts
4	N/A	N/A
5	Blue	Wind Angle Phase
6	Green	Wind Angle Phase
7	Red	Wind Angle Phase
8	Violet	Wind Speed

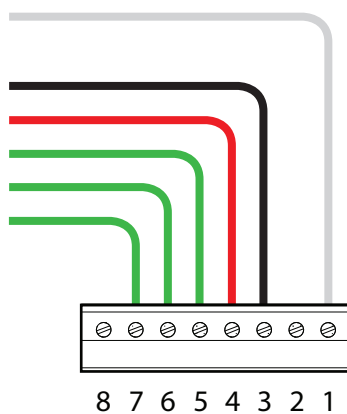
5V Analogue Input / -5V to +5V Signal



CONNECTOR 1 (TOP TERMINALS)			
TERMINAL	COLOUR	INPUT	CHANNEL
1	-	Screen	-
2	N/A	N/A	-
3	Black	0 Volts	-
4	Red	5 Volts	-
5	Sensor Dependent	Signal -5V to +5V	Analogue 1
6	Sensor Dependent	Signal -5V to +5V	Analogue 2
7	Sensor Dependent	Signal -5V to +5V	Analogue 3
8	N/A	N/A	-

Note: -5 to 0 Volt sensors must be wired into the top connector.

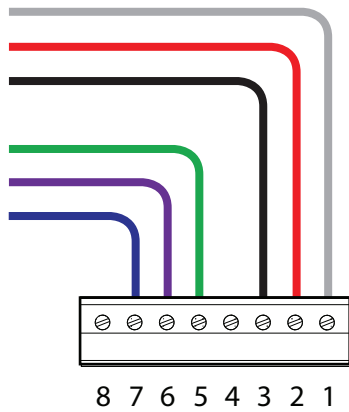
5V Analogue Input / 0V to +5V Signal



CONNECTOR 2 (BOTTOM TERMINALS)			
TERMINAL	COLOUR	INPUT	CHANNEL
1	-	Screen	-
2	N/A	N/A	-
3	Black	0 Volts	-
4	Red	5 Volts	-
5	Sensor Dependent	Signal 0V to +5V	Analogue 4
6	Sensor Dependent	Signal 0V to +5V	Analogue 5
7	Sensor Dependent	Signal 0V to +5V	Analogue 6
8	N/A	N/A	-

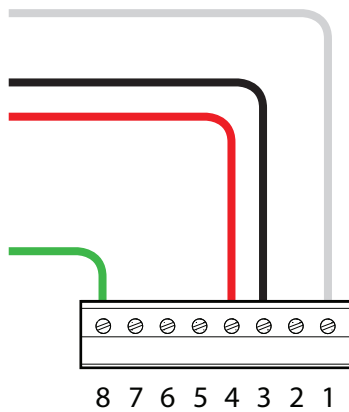
Note: 0 Volt to 5 Volt Sensors can be wired into either the top or the bottom connector terminals. If you plug a 0 - 5 Volt analogue input into connector 1 then the analogue channels will be 1, 2 & 3, if connected to connector 2 then they will be channels 4, 5 & 6.

Rate Gyro



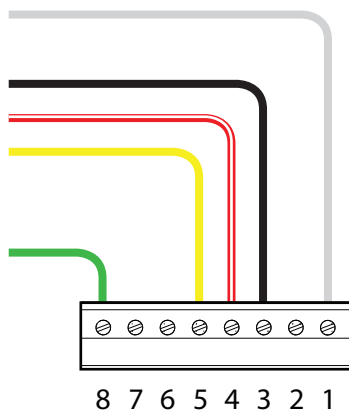
TERMINAL	COLOUR	INPUT
1	-	Screen
2	Red	12 Volts
3	Black	0 Volts
4	N/A	N/A
5	Green	Roll Rate
6	Violet	Pitch Rate
7	Blue	Yaw Rate
8	N/A	N/A

Paddle Wheel



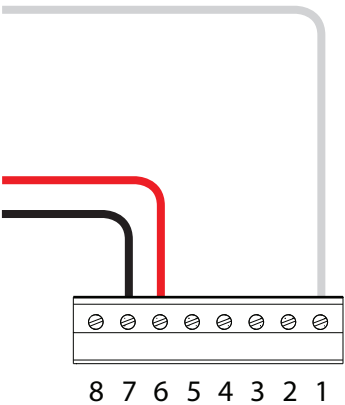
TERMINAL	COLOUR	INPUT
1	-	Screen
2	N/A	N/A
3	Black	0 Volts
4	Red	5 Volts
5	N/A	N/A
6	N/A	N/A
7	N/A	N/A
8	Green	Speed

Paddle Wheel & Sea Temperature



TERMINAL	COLOUR	INPUT
1	-	Screen
2	N/A	N/A
3	Black	0 Volts
4	Red & White	5 Volts
5	Yellow	Sea Temperature
6	N/A	N/A
7	N/A	N/A
8	Green	Speed

Analogue Speed Sensor

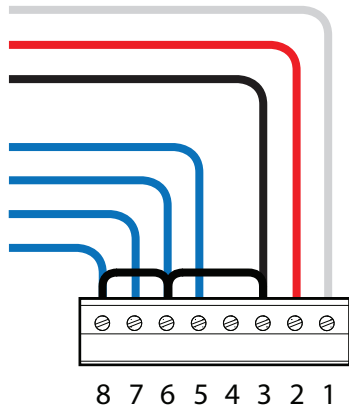


CONNECTOR 2 (BOTTOM TERMINALS)			
TERMINAL	COLOUR	INPUT	CHANNEL
1	-	Screen	-
2	N/A	N/A	-
3	N/A	N/A	-
4	N/A	N/A	-
5	N/A	N/A	Analogue 4
6	Red	Analogue Input +	Analogue 5
7	Black	Analogue Input -	Analogue 6
8	N/A	N/A	-

Note: Analogue speed sensors must be wired into the bottom connector (connector 2)

Serial Module Wiring

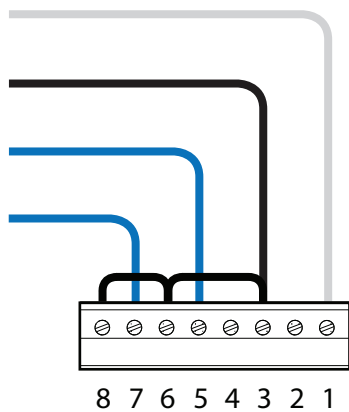
NMEA0183 - GPS Antenna



TERMINAL	COLOUR	INPUT
1	-	Screen
2	Red	12 Volts
3	Black	0 Volts
4	N/A	N/A
5	Sensor Dependent	TX +
6	Sensor Dependent	TX -
7	Sensor Dependent	RX +
8	Sensor Dependent	RX -

Note: A link wire may be required between terminals 3 & 6 and 6 & 8 depending on the sensor.

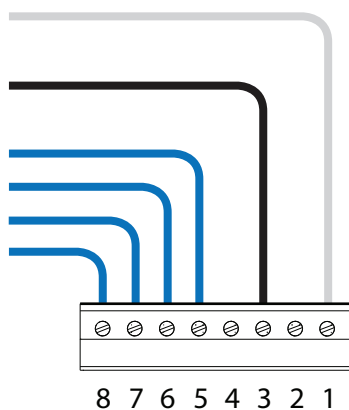
RS232



TERMINAL	COLOUR	INPUT
1	-	Screen
2	N/A	N/A
3	Black	0 Volts
4	N/A	N/A
5	Sensor Dependent	TX +
6	N/A	N/A
7	Sensor Dependent	RX +
8	N/A	N/A

Note: A link wire must be inserted between terminals 3 & 6 and 6 & 8.

RS485 & RS422

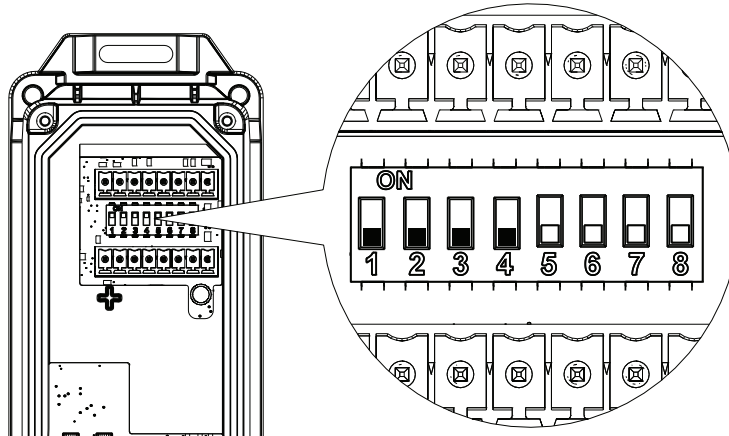


TERMINAL	COLOUR	INPUT
1	-	Screen
2	N/A	N/A
3	Black	0 Volts
4	N/A	N/A
5	Sensor Dependent	TX +
6	Sensor Dependent	TX -
7	Sensor Dependent	RX +
8	Sensor Dependent	RX -

Setting the Module Address

When adding any module to the network you will need to allocate a unique address to that module so that it can be referenced with the WTP.

To set the address there are 8 dip switches inside the module located between the two terminals as shown below. As default all of these switches are set to the 'OFF' position, making the address code zero.



The address allocation switches work on a binary code system where:-

Switch 1	=	1
Switch 2	=	2
Switch 3	=	4
Switch 4	=	8

Note: Only switches 1 to 4 are used to set the module address. Switches 5 to 8 are not used and should be left in the 'OFF' position at all times.

Binary Code Example



The address is set by moving a combination of switches and adding the binary value of each switch in the 'ON' position together. This binary code system makes it possible to set a total of 16 address codes 0 to 15.

The example above shows switches 1, 4 and 8 in the 'ON' position. So the binary total will be $1+4+8 = 13$. This would set the address to 13.

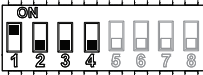
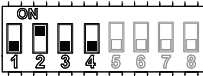
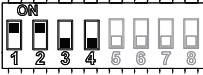
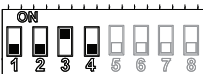
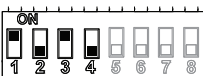
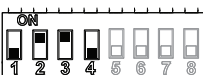
Note: Each module must have a unique address allocated to it.

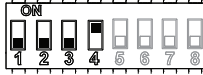
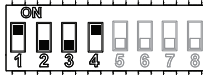
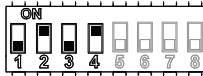
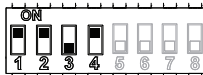
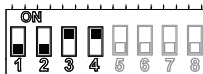
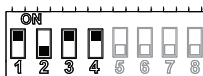
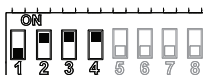
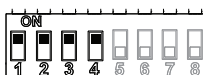
Module Hot-Swap

If any changes are required on the system it is recommended that the system power is turned off. However, if necessary, sensor interfaces can be replaced while the system is powered on. If replacing a module ensure that you set the address of the new module to match the unit being replaced, then the system will not require any re-configuration.

Dip Switch Binary Address Codes

Below is a table showing all of the possible binary code dip switch positions.

Address	Dip Switch Selection
0	
1	
2	
3	
4	
5	
6	
7	

Address	Dip Switch Selection
8	
9	
10	
11	
12	
13	
14	
15	

Fastnet Installation

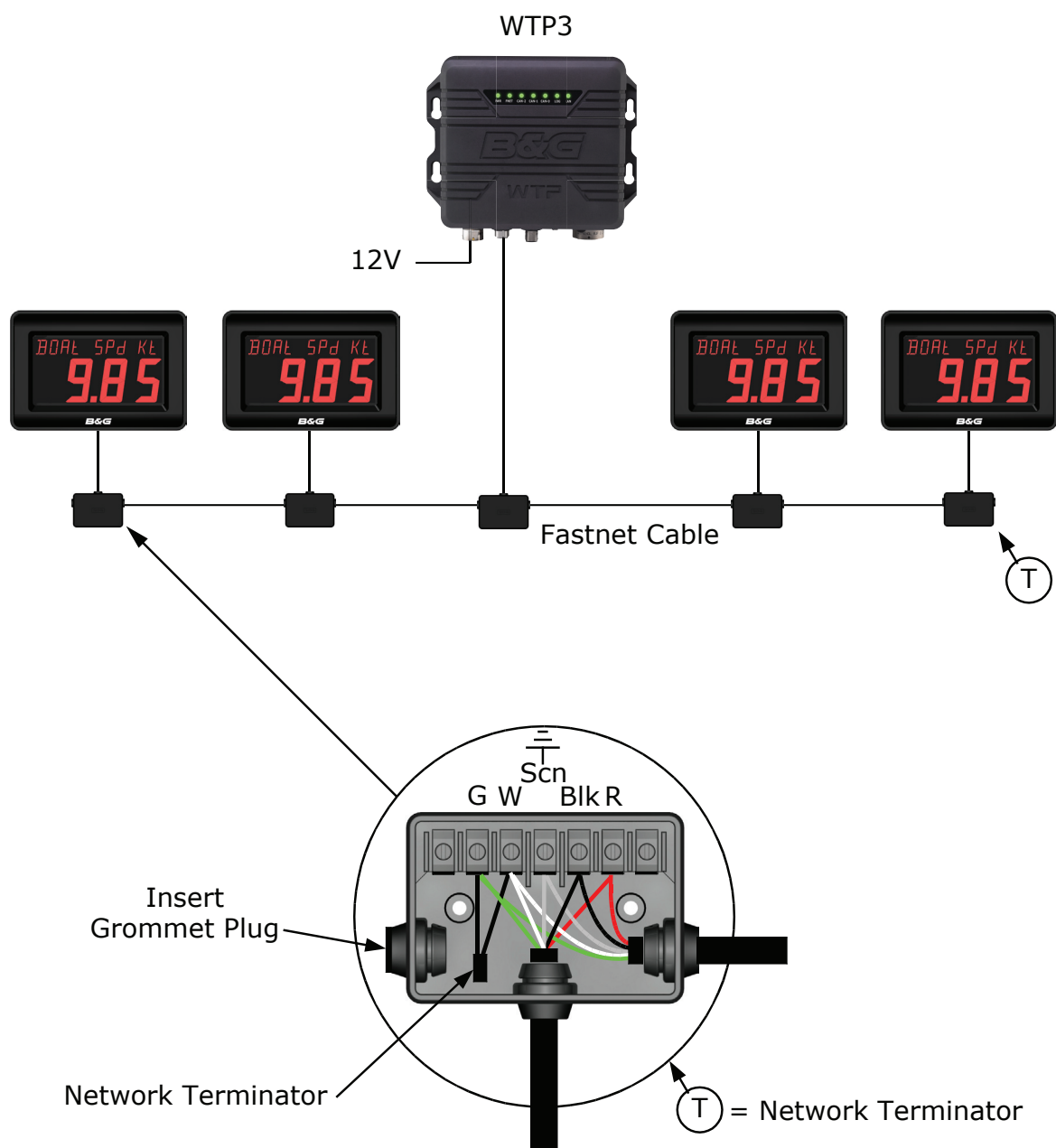
Network Terminator

The Network Terminator (B&G part 239-00-099) is a black two-wired component with a resistance of 100 Ohms. Two are supplied with insulating sleeving to prevent shorting of the wires.

A Network Terminator must be fitted across the Green and White Fastnet databus wires of the last unit of junction box at each end of the network cable. (Refer to the examples below)

When adding more displays or units to the system, ensure that the terminator is moved to the ends of the Fastnet databus cable.

Never fit more than two terminators to the system. All systems, no matter how large or small, must have two terminators installed across the Green and White wires.



WTP3 Variables

VARIABLE NUMBER	NAME	SHORT NAME	DESCRIPTION	NOTES	NORMAL FASTNET FUNC NO.
0	Heel	HI	Heel		
1	dotHeel	dHI	Rate of change of heel	Not implemented	
2	Boatspeed	VS	Boat speed		41
3	dotVS	dVS	Rate of change of boat speed, i.e. acceleration	Not implemented	
4	SmoothVS	sVS	Moving average of boat speed	Not implemented	
5	MHU_A_R	A_R	Masthead unit – red phase		
6	MHU_A_G	A_G	Masthead unit – green phase		
7	MHU_A_B	A_B	Masthead unit – blue phase		
8	MW_angle	MWA	Measured wind angle		5A
9	MW_speed	MWS	Measured wind speed		57
10	AW_angle	AWA	Apparent wind angle		51
11	AW_speed	AWS	Apparent wind speed		4D
12	Leeway	Lee	Leeway		82
13	Heading	Hdg	Magnetic compass heading	Not including leeway	49
14	Course	Cse	Course	Heading and leeway combined	69
15	dotCourse	dCs	Rate of change of course		
16	TW_angle	TWA	True wind angle		59
17	TW_speed	TWS	True wind speed		55
18	TW_dirn	TWD	True wind direction		6D
19	VMG	VMG	Velocity made good		7F
20	GW_speed	GWS	Ground wind speed		

VARIABLE NUMBER	NAME	SHORT NAME	DESCRIPTION	NOTES	NORMAL FASTNET FUNC NO.
21	GW_dirn	GWD	Ground wind direction		
22	Orig_TWA	ta	Original true wind angle		
23	Orig_TWS	ts	Original wind speed		
24	Orig_TWD	td	Original True Wind Direction		
25	MastRot	MRo	Mast Rotation		9C
26	TWD_Off	wdo	True Wind Direction offset		
27	selSOG	SOG	Selected SOG	selected from GPS1 or 2	EA
28	selCOG	COG	Selected COG	selected from GPS1 or 2	EB
29	VMC	VMC	Velocity Made Good relative to Course		EC
30	Opt_VMC	OVC	Optimum VMC		
31	Cse_OVMC	COC	Course for Optimum VMC		
32	Vs_target	TS	Target Boat Speed		7D
33	Vs_targ%	T%	Boat Speed as a percentage of Target Boat Speed		
34	TWA_targ	AT	Target True Wind Angle		53
35	Vs_perf	PPV	Boat speed derived from performance polar		7E
36	Vs_perf%	PP%	Boat speed as a percentage of performance polar		33
37	Vs_nav	PNV	Boat speed derived from navigation polar		
38	Vs_nav%	PN%	Boat speed as a percentage of navigation polar		
39	Brg_o_Mrk	BM	Bearing of mark		E6
40	Dst_t_Mrk	DM	Distance to mark		E8
41	Tm_t_Mrk	TM	Time to mark		35
42	Curr_Rate	CrR	Current rate	Written from Deckman	
43	Curr_Dir	CrD	Current direction	Written from Deckman	

VARIABLE NUMBER	NAME	SHORT NAME	DESCRIPTION	NOTES	NORMAL FASTNET FUNC NO.
44	MCur_Rate	MCR	Measured current rate	Written from Deckman	
45	MCur_Dir	MCD	Measured current direction	Written from Deckman	
46	DCur_Rate	DCR	Diamond current rate	Written from Deckman	
47	DCur_Dir	DCD	Diamond current direction	Written from Deckman	
48	Battery	Bat	Battery volts		8D
49	Rudder	Rud	Rudder Angle		0C
50	Rake	Rke	Mast Rake		CA
51	gyro_hl	GHI	Gyro Heel		
52	gyro_dhl	GdH	Roll (rate of change of gyro heel)	Input from rate gyro	3C
53	gyro_trm	GTm	Gyro trim		
54	gyro_dpt	GdP	Pitch (rate of change of trim)	Input from rate gyro	9C
55	gyro_hdg	GHg	Gyro heading		49
56	gyro_dyw	GdY	Yaw (rate of change of heading)	Input from rate gyro	44
57	trim	trm	Trim		9B
58	forestay	frs	Forestay load		CC
59	keel	kel	Keel angle		C9
60	seatemp	sea	Sea temperature		1F
61	airtemp	air	Air temperature		1D
62	barom	bar	Barometer		87
63	port_VS	pVS	Port boat speed paddlewheel		
64	stbd_VS	sVS	Starboard boat speed paddlewheel		
65	CMW_angle	CWA	Corrected Measured Wind Angle		A8
66	CMW_speed	CWS	Corrected Measured Wind Speed		A9

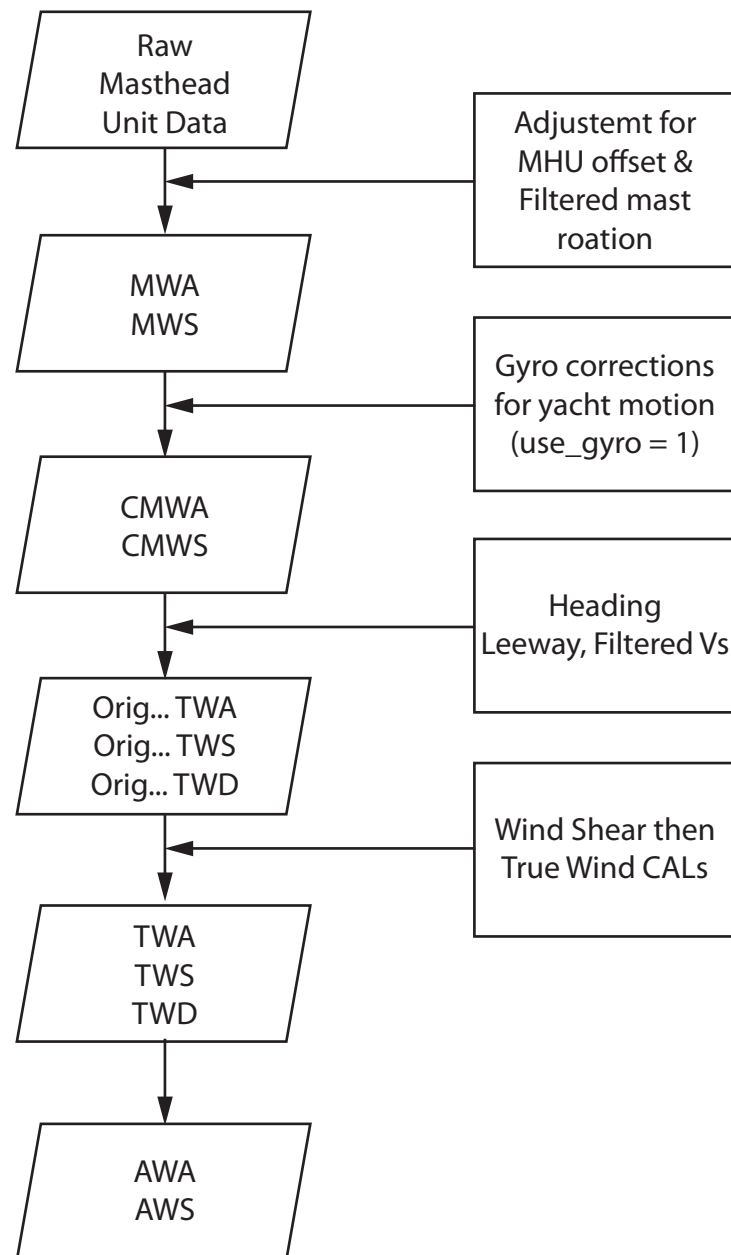
VARIABLE NUMBER	NAME	SHORT NAME	DESCRIPTION	NOTES	NORMAL FASTNET FUNC NO.
67	Depth	Dep	Depth		0B
68	XTrkErr	XTE	Cross track error		EE
69	TWA_OVMC	AOC	True Wind Angle for Optimum VMC		
70	VMG_Targ	VGT	Target VMG		
71	VMG_Targ%	VGP	VMG as a percentage of Target VMG		32
72	OppTrkW	OTW	Opposite tack track (wind)	Calculated from wind direction, polars and tidal information	9A
73	OppTrkG	OTG	Opposite tack track (COG)	Calculated from COG, polar table and tidal information	
74	Log	Log		Can be reset from the Settings dialog	CD
75	pitchRMS	PMS	Pitch Root Mean Squared	Gives an indication of wave amplitude	
76	pitchPrd	PPd	Pitch period	Gives an indication of wave period	
77	selUTC	UTC	Time in seconds since midnight (0000 hours), UTC	selected from GPS1 or 2	DD
78	selSVA	SVA	Digit 1: number of satellites, Digits 2-4: age of GPS fix in seconds	selected from GPS1 or 2	
79	selQHD	QHD	Digit 1: quality of GPS fix Digits 2-4: HDOP in metres	selected from GPS1 or 2	
80	Hdg2	HG2	Heading 2		
81	Hdg2_hl	H2H	Heel 2		
82	Hdg2_trm	H2T	Trim 2		
83	Hdg3	Hg3	Heading 3		
84	Hdg3_hl	H3H	Heel 3		
85	Hdg3_trm	H3T	Trim 3		
86	BoatSpd2	VS2	Boat speed 2		
87	VS2pad	V2p	Boat speed 2, raw data.		
88	SelHdg	SHg	Selected heading		49
89	SelHeel	SHI	Selected heel		34

VARIABLE NUMBER	NAME	SHORT NAME	DESCRIPTION	NOTES	NORMAL FASTNET FUNC NO.
90	SelTrim	Stm	Selected trim		96
91	SelBoatSpd	VSS	Selected boat speed		41
92	Hdg2_Heave	Hve	Heave from Compass 2	Requires EM series compass	
93	GGBrg	GGB	GPS1 to GPS2 bearing	For comparison of GPS inputs	
94	GGRng	GGR	GPS1 to GPS2 range	For comparison of GPS inputs	
95	HHDiff	HHd	Heading 1 to Heading 2 difference	For compass comparison	
96	MastWind	MWM	Wind Angle measured relative to mast	For rotating mast systems	9D
97	FwdRud	FWR	Forward Rudder Angle		28
98	Code0	Cd0	Code 0 load		A3
99	Vang	Vng	Vang position		CB
100	gps1cog	cg1	COG from GPS1	From a position fixer. These functions are used when GPS 1 is selected	
101	gps1sog	sg1	SOG from GPS1		
102	gps1qhd	qh1	QHD from GPS1		
103	gps1sva	sv1	SVA from GPS1		
104	gps1utc	ut1	UTC from GPS1		
105	gps2cog	cg2	COG from GPS2	From a position fixer. These functions are used when GPS 2 is selected	
106	gps2sog	sg2	SOG from GPS 2		
107	gps2qhd	qh2	QHD from GPS 2		
108	gps2sva	sv2	SVA from GPS 2		
109	gps2utc	ut2	UTC from GPS 2		
110	Variation	Var	Magnetic Variation	Received from either GPS source	
111				New functions can be added from here onwards	

Supported Compass Types

COMPASS SENSOR	INPUT SENTENCE	HDG	HEEL	TRIM	HEAVE	RATES	LABEL IN FILE
B&G Halcyon 2000	B&G Fastnet	Y	N	N	N	N	HALCYON
B&G Halcyon Gyro	B&G Fastnet	Y	Y	Y	N	N	BGGYRO
B&G IMU	Binary	Y	Y	Y	N	Y	BGIMU
Crossbow AHRS	Binary	Y	Y	Y	N	Y	XBAHRS
CSI Vector GPS	\$PSAT,HPR,hhmmss.ss,h.h,p,p,r,r,*KK	Y	Y	Y	N	N	PSAT
Furuno SC60	\$PFEC,GPatt,hhh.h,+pp.p+rr.r (Ver. 1.5)	Y	Y	Y	N	N	<no label>
	\$PFEC,GPatt,hhh.h,+pp.p+rr.r*KK (Ver. 2.0)	Y	Y	Y	N	N	<no label>
Honeywell HMR3000	\$PTNTHPR,h.h,a,p,p,a,r,r,a*KK	Y	Y	Y	N	N	HMR3000
Keppel HPR03	\$HPR,h.h,p,p,r,r	Y	Y	Y	N	N	KEPPEL
KVH GyroTrac	%pppp,rrrr,hhhh	Y	Y	Y	N	N	<no label>
NMEA 0183 Heading Sensor	\$xxHDT,h.h,T	Y	N	N	N	N	<no label>
	\$xxHDM,h.h,M	Y	N	N	N	N	<no label>
	\$xxHDG,h.h,d,d,a,v,v,a*KK	Y	N	N	N	N	<no label>
PNI Corp. TCM2	\$Ch.hPp.pRr.rXxx.xxYyy.yyZzz.zzTcc.c	Y	Y	Y	N	N	TCM2
PNI Corp. TCM2 (Heel/Trim sensor only)	\$Pp.pRr.r	N	Y	Y	N	N	TCM2
PRDID Proprietary NMEA	\$PRDID,p,p,r,r,h.h*KK	Y	Y	Y	N	N	PRDID
Simrad EM-series input format (EM1000, EM3000)	Binary	Y	Y	Y	N	N	EM3000
Xsens	Binary	Y	Y	Y	N	N	XSENS
Octans	\$HEHDT,x.xx,T*hh \$PHTRO,x.xx,a,y.yy*hh \$PHLIN,x.xxx,y.yyy,z.zzz*hh \$PHROT,x.xxx,y.yyy,z.zzz*hh	Y	Y	Y	N	Y	OCTANS
Gladiator / Landmark	Binary	Y	Y	Y	N	Y	GLD
Microstrain 3DM-GX3	Binary	Y	Y	Y	N	Y	MS3DM
TOGS/S	\$HEHDT,x.xx,T*hh \$PHTRO,x.xx,a,y.yy*hh \$PHLIN,x.xxx,y.yyy,z.zzz*hh \$PHROT,x.xxx,y.yyy,z.zzz*hh	Y	Y	Y	N	Y	TOGS

Wind Calculation Flowchart



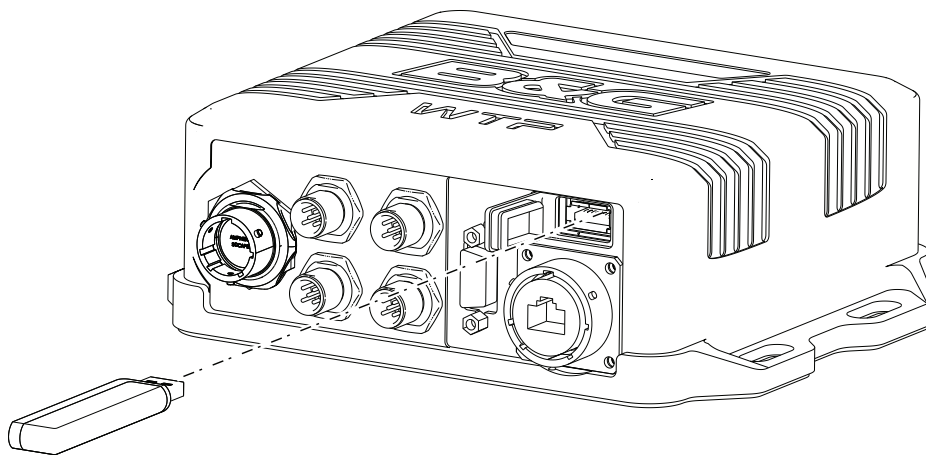
Note: If use_gyro is set to '0' (OFF) then the CMWA, CMWS stage is bypassed.

Upgrading the WTP3 Software

CPU

To upgrade your WTP3 CPU Software to the latest version all you will need is the .upd file (upgrade file) and a USB memory stick. (Must be FAT or FAT32 format, NTFS is not supported)

1. Save .upd file to a USB Stick
2. Turn off power to the WTP CPU
3. Place USB stick into the USB port in the front of the CPU



4. Turn on power to the WTP CPU
5. The CPU will automatically recognise the .upd file and begin the upgrade process
6. To indicate the upgrade is in progress the LOG LED on the top of the CPU will flash green
7. When the LOG LED turns solid green the upgrade process is complete
8. Remove the USB stick, the CPU will automatically reboot



Note: A solid red light on the LOG LED indicates the upgrade was unsuccessful. Check your .upd file , USB stick and repeat the process. If you continue to experience difficulties updating the software please seek assistance from a B&G WTP specialist.

Module

To upgrade a module to the latest software version you will need to use the Navico Software Upgrade Tool. Follow the Navico SWUP (Software Upgrade) tool procedure)

WTP3 File Structure

When a user logs in via FTP into the CPU, The user will be presented with the following file structure:

1. wtp_config.d
Details the set of config files to use via a directory name.
2. network.d
Details IP address to use (see Ethernet Configuration)
3. wtp3-datalog.sqlite is the SQL data logging database.
4. Bootlog directoy - contains a set of text files. Each file contains the terminal output from the power cycle. To be used for diagnostics.
5. wtp_files directory - Contains the standard set of WTP3 files
6. Any additional WTP3 file directories added by the user.

Below is an example of the wtp_config.d file where test1 is the name of the directory where the current config file is stored.

```
[Config]
Dir test1
```

Diagnostic Messages via Terminal

On boot the terminal window will display several messages detailing the boot up and configuration file load of the unit, these are summarized below. If you require support on a WTP3 installation it is helpful to have a copy of the configuration files in use and a log of a system boot to send to the WTP support specialist.

Error Types

Error messages are categorised into four types:

- Info
- Warning
- Critical
- Fatal

Info

General message displayed for information, e.g. software version number

Warning

An indication that a configuration error exists. The error does not need the system to stop, but requires user attention.

For example, if a configuration file calls for a device that is not visible on the network:

```
Warning: nmeain.d : Serial Device 15 on CAN Channel 1 does not exist
```

Critical

An indication that an error exists that requires immediate user attention. Critical errors will not stop the system, but the system is unlikely to operate correctly until the issue is resolved.

Fatal

Fatal errors will stop the WTP3 application from running. FTP access is normally retained. The user must rectify the issue to use WTP3.

For example, if the file directory referred to in wtp_config.d for the user files does not exist:

```
Fatal : Config File Path does not exist : mydirectory
```

Configuration Consistency

At startup the system will attempt to cross reference the config files for validity, ensuring the user has not:

- called the same COM port in multiple files, using different port settings
- called the same COM port in multiple files with different functions (e.g compass and GPS)
- assigned multiple analogue inputs to a single variable

In this case the system will output Warning messages to the user, as per the following examples:

```
Heel1 MHU_A_B Multiple Analogue Definition : 0 1 2
```

- "heel1" and "mhu_a_b" functions have been configured to the same port (channel 0, device 1, port 2):

MHU_VA portpad Multiple Pulse Definition : 0 1 1

- "MHU_VA" and "portpad" functions are configured to the same input (channel 0, device 1, input 1):

Multiple Devices defined on the same Channel, Device & Port with differing settings 1 1 2

- there are two devices (unknown) on the same physical serial port but with different hardware (RS232 etc.), baud rate, data, stop or parity settings. In this case serial port 2 on channel 1, device 1 in the conflict

Full Startup Description

Software Versions

The two key versions for diagnostic purposes are:

Platform S/W Version : "(1.1 0.5 0.8)" [Operating System Version]
B&G WTP3 S/W Version : 1.0.00 [WTP3 Software Version]

Configuration Files

Config File Path : Default_Config [folder name for file set in use]

CAN initialisation

Initialising NDP2K Stack
CAN Channel 0 : 250Kbs [baud rate set for CAN channel 0]
CAN Channel 1 : 250Kbs [baud rate set for CAN channel 1]
CAN Channel 2 : 250Kbs [baud rate set for CAN channel 2]

Progress

This section details the start of various tasks, including configuration file loading.

NDP2K Thread Running
WTP3 sets custom speed on s3c_serial0. This is deprecated.
Initialising Variable Manager
Loading Surface Tables
Sampler : Loaded Settings
Sampler : Loaded Boat Log
Serial Thread Running
Sampler Thread Running
Deckman Comms Thread Running
Fastnet Manager Running
External Message Server started listening on port : 8080
Fastnet : User Menus Loaded
Fastnet : B&G Menus Loaded
Fastnet : Loading User Variables
Fastnet : Loading B&G Variables

Ethernet configuration

Identifies CPU IP address and start of UDP task:

```
My Local IP Address for eth0 : "192.168.0.2" IP Address of CPU UDP Thread
Running
```

CAN Modules present

Details the modules connected to each CAN Channel, allowing the system engineer to confirm the devices visible on each CAN network, their address, type, serial number and software version.

```
***** CAN Device List *****
```

Channel 0 :

Device : 2, WTP3 ANALOG IO MODULE, 123456789001, S/W Version R1.00 (10 Dec2010)

Device : 1, WTP3 ANALOG IO MODULE, 123456789002, S/W Version R1.00 (10 Dec2010)

Device : 3, WTP3 SERIAL IO MODULE, 123456789003, S/W Version R1.00 (10 Dec2010)

Channel 1 :

Device : 1, ZG50 B&G _Active GPS ant., 1FFFFF#, 01200_E 0005

Device : 1, WTP3 SERIAL IO MODULE, 123456789004, S/W Version R1.00 (10 Dec2010)

```
*****
```

Compass Configuration

Details the compass types installed and the CAN Channel, Device ID, Port settings and compass variables used for each input.

Installing Compass...

Compass 1 HALCYON Fastnet

heading1 Node: -1 Function: 116 Variable: 13

heel1 Node: -1 Function: 0 Variable: 0

trim1 Node: -1 Function: 0 Variable: 57

Compass 2 BGIMU Channel : 0 Device : 8 Comm Port: 0, Baud: 28800

heading2 Variable: 80

heel2 Variable: 81

trim2 Variable: 82

Roll Variable: 111

Pitch Variable: 112

Yaw Variable: 113

Accx Variable: 114

Accy Variable: 115

Accz Variable: 116

GPS Configuration

Details the GPS types installed and the CAN Channel, Device ID and Port settings used for each input.
Installing GPS...

GPS : Channel: 1, Device: 4, Comm Port: 2, Baud : 4800

NMEA0183 Configuration

Details the NMEA0183 interfaces. Including CAN Channel, Device ID and Port settings used.

```
Installing NMEA0183 Input...
```

```
NMEA0183 In : Channel : 1, Device : 4, Port: 1, Baud: 4800
```

```
Installing NMEA0183 Output...
```

```
NMEA0183 Out : Channel : 1, Device : 4, Port: 1, Baud: 4800
```

Loadcell Configuration

Details of serial Loadcell inputs. Including CAN Channel, Device ID and Port settings used.

```
LOADCELL Channel : 1, Device : 5, Port: 1, Baud: 4800
```

Fast Output Configuration (fastout.d)

Details of the output configuration of the [optional] fast output function. Including CAN Channel, Device ID and Port settings used.

```
FASTOUT Channel : 1, Device : 6, Port: 1, Baud: 57600
```

File Consistency Checks

On boot the system checks the consistency of files called in configuration files (e.g. sample.d) with the existence of the files on the unit. The example warnings below indicate two files are referenced in a file but do not physically exist:

```
Loading Sample.d.....
```

```
Warning: QFileSystemWatcher: failed to add paths: wtp_default/filters/vmg.fil
```

```
Warning: QFileSystemWatcher: failed to add paths: wtp_default/filters/vmc.fil
```

```
Sampler: Loaded Sample File
```

```
Sampler: Initialised derived variables
```

This may be due to the file not being present, the filename being misspelt or the use of uppercase characters in the configuration files – the operating system automatically renames all filenames to lowercase.

Datalog Task Start

Simple comment to indicate if the datalog task has started (also see Datalog Status, below)

```
Data logger Running
```

Alarm Task Start

Simple comment to indicate if the alarm handler task has started

```
Alarm Manager Running
```

Startup confirmation

Following the loading of the configuration files the unit will report:

```
WTP3 Startup Complete.....
```

Serial Port Config

Each serial port is configured with the settings defined in the configuration files. The following message types are displayed if the serial devices are configured correctly:

```
Serial Device : Channel 0 Device 8 Port 0 Configured Successfully
Serial Device : Channel 1 Device 4 Port 0 Configured Successfully
Serial Device : Channel 1 Device 4 Port 1 Configured Successfully
```

Data Time-out

The following warnings are printed after the startup is complete. They indicate the data in the system is missing or gone stale. i.e. the data has not been updated within the allowable time-out period.

```
Warning: "Heel1" is stale, 2.5Sec expired since last update
```

If this warning appears shortly after startup it usually indicates the data has been defined but no physical data is on the system. For example, "Heel1" has been defined in sample.d as being an analogue input on CAN channel 0, Device 1, analogue channel 1. So if the following warning appears it may indicate that we have either referenced the wrong analogue module (either CAN channel, Device ID or analogue channel) or the device is not transmitting data on the CAN channel for some other reason.

Datalog Status

Details the configuration of the datalog manager following startup.

Datalog Manager Initialised	[datalog manager started]
Start : Racetimer	[datalog will start with racetimer]
Duration: 60	[datalog will run for 60 minutes]

Alarm Status

Details the configuration of any alarms (.alm files) in use following startup.

Alarm Config :

Depth 67 ENABLED	[Depth alarm, variable 67 is enabled]
TW_Speed 17 DISABLED	[TWS alarm, variable 17 is disabled]

