Boxfish Research Limited, 234A Orakei Road, Remuera, Auckland 1050, New Zealand | +64 9 600 1910 | info@boxfish.nz

Automation Computer Software

1.0 Introduction

This document provides an overview of the Boxfish Research automation software, which runs on the automation computer. In particular, the bespoke ROS2 packages are the focus of this document.

The automation computer software uses ROS2 as a framework. The bespoke ROS2 packages are located in *~/projects/rostwo/src*. These packages are built using *colcon build* in *~/projects/rostwo*. Configuration besides launch file parameters and defaults in source code, may be found in *~/.ros*. Some of the specialist dependencies may be found in *~/Downloads, /opt* and otherwise the home area.

2.0 Package Descriptions

2.1 serial_hub_interface

The *serial_hub_interface* package provides commands to and feedback interfaces from the automation computer serial port that is connected to the ROV Serial Hub. The main node in this package is *ros_nmea* (ros_nmea.cpp).

Subscribed topics are:

- drone_twist (geometry_msgs::msg::Twist), which is translated to “PROVSTEER” commands for controlling the vehicle pose.
- current_cmds_mask (std_msgs::msg::Int32), which is a bit mask defined in *auto_control/include/common_parameters.hpp* in *CMD_REPEAT* used to control:
    - Thruster enabling
    - Internal 12 VDC power
    - Internal 24 VDC power
    - Stability enabling
    - Depth hold enabling
    - Levelling
    - Return to surface enabling
- light_cmd (std_msgs::msg::Int32), which controls lights with values from 0-100% intensity
- heading_cmd (std_msgs::msg::Int32), which translates to a compass heading target in degrees
- depth_cmd (std_msgs::msg::Float64), which translates to a depth target in metres
- joy (sensor_msgs::msg::Joy), which provides a bare-bones joystick interface

Published topics are:

- pubpp_usbl_raw (auv_interfaces::msg::UsblPubpp), which provides a USBL beacon_id with the range (m), azimuth (radians), elevation (radians) and fit error read from the USBL in the USBL’s coordinate system
- pubatt_usbl_raw (auv_interfaces::msg::UsblPubatt), which provides a USBL beacon id with roll, pitch and yaw (all in radians) read from the USBL in the USBL’s coordinate system

- hchdm (std_msgs::msg::Float64), which publishes the heading measurements from the compass in degrees

2.2 spinnaker
The *spinnaker* package provides images from the FLIR cameras, published on *sensor_msgs/msg/Image* topics.

2.3 auto_control
The *auto_control* package provide system wide definitions in *auto_control/include/common_parameters.hpp.*

2.4 auv_interfaces
The *auv_interfaces* package provides msgs used by the automation system. The key messages are *UsblPubatt.msg* and *UsblPubpp.msg* for USBL data.

USBL NMEA Commands (Available from Serial Hub)

USBL Attitude (PUBATT)

$PUBATT,<1>,<2>,<3>,<4>*HHHHHHHH<CR><LF>

<1>	Beacon ID (not implemented atm)	1
<2>	Roll to 1DP	-179.3
<3>	Pitch to 1DP	1.0
<4>	Yaw to 1DP	33.4

USBL Polar Position (PUBPP)

$PUBPP,<1>,<2>,<3>,<4>,<5>*HHHHHHHH<CR><LF>

<1>	Remote Beacon ID	2
<2>	Range of Remote to 1DP (m)	18.9
<3>	Azimuth of Remote to 1DP (deg)	152.4
<4>	Elevation of Remote to 1DP (deg)	-21.9
<5>	USBL Fit Error to 2DP (see usbl doc)	1.08

USBL Rectangular Position (PUBRP)

$PUBRP,<1>,<2>,<3>,<4>*HHHHHHHH<CR><LF>

<1>	Remote Beacon ID	2
<2>	Position of Remote Easting to 1DP (m)	-15.2
<3>	Position of Remote Northing to 1DP (m)	-8.4
<4>	Depth of Remote to 1DP (m)	0.3

Depth will use remote beacons depth sensor if an enhanced usbl message is being transmitted

USBL Transmission Error (PUBTE)

$PUBTE,<1>,<2>,<3>*HHHHHHHH<CR><LF>

<1>	Beacon Id (unimplemented)	1
<2>	Protocol (NAV/DAT/PING/ECHO)	NAV
<3>	Error Type	TO

	FO = Formatting Error BU = Transceiver Busy TO = Response Timeout RE = General Reception Error SE = Sender Error (wrong mesage rx'd)	

Compass NMEA Commands (Available from Serial Hub)

Set Output Mode (PCPOM)

$PCPOM,<1>*HHHHHHHH<CR><LF>

<1> is 0 or 1 (1 is auto mode, 0 reply only mode)

Calibrate Compass Begin (PCPCA)

$PCPCA*HHHHHHHH<CR><LF>

Calibrate Compass End & Save (PCPSC)

$PCPSC*HHHHHHHH<CR><LF>

Set Compass Baud Rate (PCPBR)

$PCPBR,<1>*HHHHHHHH<CR><LF>

<1>	Baudrate (2400 - 115200)	2400

Set Compass Declination (PCPSD)

$PCPSD,<1>*HHHHHHHH<CR><LF>

<1>	Declination (-89.9 to 89.9)	30.2

Status Messages

Compass Heading (HCHDM)

$HCHDM,<1>,<2>*HHHHHHHH<CR><LF>

<1>	Heading to 1dp	233.2
<2>	Magnetic (M)	M

conforms to nmea standard message

Compass Acknowledge Command (PCPAK)

$PCPAK,<1>*HHHHHHHH<CR><LF>

<1>	Command Execution Successful (1 = true, 0 = false)	1

Issued in reply to commands sent to the compass

DVL Basic NMEA Commands (Available from Serial Hub)

DVL Velocities (PDVLVL)

$DPVDVL,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,<10>,<11>*HHHHHHHH<CR><LF>

<1>	X velocity	12.1
<2>	Y velocity	-0.2
<3>	Z velocity	4.3
<4>	Velocity is valid	1
<5>	Altitude	10.3
<6>	Figure of merit	0.1
<7>	Covar	.8
<8>	Time of validity	16
<9>	Time of transmission	18
<10>	Time since last velocity report	16.1
<11>	Report is valid	1

Altitude (SDDPT) [this is a standard altimeter message created from the DVL Velocities report altitude]

$SDDPT,<1>,<2>,<3>*HHHHHHHH<CR><LF>

<1>	Altitude (m)	10.32
<2>	Offset (m)	0.0

<3>	Maximum range (m)	50

DVL Transducer Info (PDVLIN)

$DPVDIN,<1>,<2>,<3>,<4>,<5>HHHHHHHH<CR><LF>

<1>	Transducer report id	12
<2>	Transducer report velocity	5.2
<3>	Transducer report distance	4.3
<4>	Transducer report RSSI	10
<5>	Transducer report NSD	10

DVL Dead Reckoning Report (PDVLDR)

$DPVDLDR,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>HHHHHHHH<CR><LF>

<1>	Time stamp	12 .1
<2>	x	52.2
<3>	y	4.9
<4>	z	3.3
<5>	Pos std	10.1
<6>	Roll	-0.9
<7>	Pitch	1.2
<8>	Yaw	14.7
<9>	Status	1