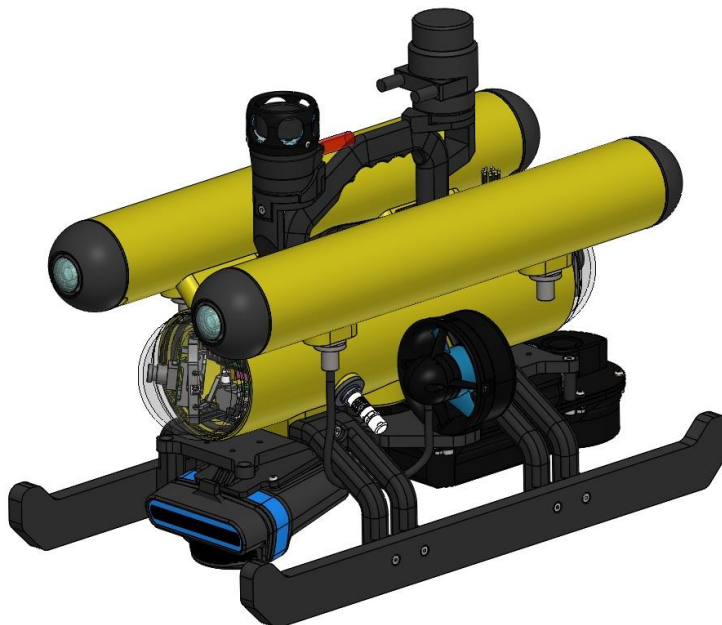


SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE



Client	DGA TN
Ref. Doc	ST-OM-18634-02
Date	15/05/2018
Version	DRAFT



SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE

Doc N° : **ST-OM-18653-02**

Version : **DRAFT**

Date : **15/05/2018**

Page : 2/12

Version	Rédigé par :	Vérifié par :	Approuvé par :
Nom : Visa : Date :	A. CHAUVET	P. MARTY	P. MARTY
CLASSIFICATION DOCUMENT			
<u>Confidentialité</u> <input type="checkbox"/> Non confidentiel <input checked="" type="checkbox"/> Confidentiel Industrie <input type="checkbox"/> Confidentiel Défense		<u>Statut</u> <input checked="" type="checkbox"/> Pour commentaires <input type="checkbox"/> Pour approbation <input type="checkbox"/> Pour exécution	
REVISION			
Version	Date	Objet	
DRAFT	15/05/18	Communiqué avant recette	



SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE

Doc N° : **ST-OM-18653-02**

Version : **DRAFT**

Date : **15/05/2018**

Page : 3/12

TABLE DES MATIERES

1	PREREQUIS	6
1.1	INSTALLATION D'UBUNTU	6
1.2	INSTALLATION DE ROS	6
1.3	DESCRIPTION DE L'API	6
1.4	OUTILS ROS	6
1.5	INSTALLATION DES NODES	7
1.6	LANCEMENT DES NODES	7
2	DESCRIPTION DES NODES	7
2.1	NODE ROV	7
2.1.1	Données publiées	7
2.1.2	Données souscrites	9
2.2	NODE USBL	10
2.2.1	Données publiées	10
2.2.2	Données souscrites	11
2.2.3	Messages personnalisés	11
2.3	NODE DVL	13
2.3.1	Données publiées	13
2.3.2	Données souscrites	13
2.4	NODE MICRON SONAR	13
2.4.1	Données publiées	13
2.4.2	Visualisation	13
2.4.3	Paramétrage	14



SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE

Doc N° : **ST-OM-18653-02**

Version : **DRAFT**

Date : **15/05/2018**

Page : 4/12

Attention !

Lisez attentivement le présent manuel ainsi que celui concernant l'utilisation du ROV avant d'utiliser Le SDK ROS pour la première fois

1 PREREQUIS

Le développement ainsi que les tests ont été effectués sur un système linux (distribution Ubuntu 16.04) et ROS Lunar. Nous recommandons d'utiliser à minima ces versions ou des versions ultérieures.

1.1 Installation d'Ubuntu

- Télécharger l'image Ubuntu à cette adresse : <https://ubuntu-fr.org/telechargement>
- Créer un disque ou une clé USB d'amorçage (suivre le manuel ici : https://doc.ubuntu-fr.org/tutoriel/obtenir_cd_ubuntu)
- Installer Ubuntu en suivant le manuel suivant : https://doc.ubuntu-fr.org/tutoriel/installer_ubuntu_avec_le_live_cd

1.2 Installation de ROS

Pour installer la suite ROS se référer au manuel <http://wiki.ros.org/lunar/Installation/Ubuntu>.

Au paragraphe 1.4 sélectionner la version « **Desktop-Full Install** ».

Il est indispensable de suivre les instructions jusqu'au bout.

1.3 Description de l'API

L'API est constituée de plusieurs nodes ROS permettant la communication (envoi de commandes ou réception d'information) avec le ROV et les différents instruments qu'il emporte. Il existe un node par équipement (sonar multifaisceaux, sonar à balayage, USBL, DVL, ROV).

Le démarrage des différents nodes a été regroupé dans une seule commande : **roslaunch**

1.4 Outils ROS

Les tests de l'API ROS ont été réalisés avec les outils rqt_publisher, rqt_gui ainsi qu'RVIZ pour la visualisation des images sonar.

Ces outils accessibles via les commandes :



SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE

Doc N° : **ST-OM-18653-02**

Version : **DRAFT**

Date : **15/05/2018**

Page : 5/12

```
roslaunch rqt_publisher rqt_publisher
```

```
roslaunch rqt_gui rqt_gui
```

```
roslaunch rviz
```

Ils permettent respectivement : d'injecter des données, de visualiser les données échangées sur le réseau ROS de visualiser graphiquement les données.

1.5 Installation des nodes

- 1) Dézipper le fichier « workspace.zip » (situé sur la clé USB fournie) dans le dossier « workspace » de l'arborescence ROS
- 2) se placer dans le dossier précédemment créé et lancer la commande :

```
catkin init
```

afin d'initialiser le workspace contenant les nodes CORAL.

- 3) Toujours dans le même dossier taper la commande :

```
catkin make
```

afin de compiler l'intégralité des nodes.

1.6 Lancement des nodes

Un fichier « .launch » est disponible afin de lancer le node principal (de contrôle du ROV) et RosCore.

```
roslaunch driver_rov coral.launch
```

Il est ensuite possible d'ajouter d'autres nodes pour les capteurs (DVL, sonar, USBL) en utilisant les commandes suivantes:

```
roslaunch driver_rov coral.launch
```

```
roslaunch tritech_micron tritech_micron.launch
```

```
roslaunch dvl dvl_node
```

```
roslaunch seatrac_pinger_node seatrac_read
```

2 DESCRIPTION DES NODES

2.1 Node ROV

2.1.1 Données publiées

Topic	Type de donnée	Donnée	Valeur	Désignation
"/ROV/Battery"	sensor_msgs::BatteryState			Etat de charge de la batterie (en %)
	float32	capacity		
	float32[]	cell_voltage		



SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE

Doc N° : **ST-OM-18653-02**

Version : **DRAFT**

Date : **15/05/2018**

Page : 6/12

	float32	charge		
	float32	current		
	float32	design_capacity		
	std_msgs/Header	header		
	string	location		
	float32	percentage	[0;100]%	
	uint8	power_supply_health		
	uint8	power_supply_status		
	uint8	power_supply_technology		
	bool	present		
	string	serial_number		
	float32	voltage		
"/ROV/Depth"	std_msgs::Float32		[0;-150]	Profondeur d'immersion en mètre
"/ROV/Depth_consigne"	std_msgs::Float32		[0;-150]	Valeur de consigne utilisée par l'asservissement en profondeur
"/ROV/Cap"	std_msgs::Float32		[0-359,9]	Cap de l'engin en °
"/ROV/Cap_consigne"	std_msgs::Float32		[0-359,9]	Valeur de consigne utilisée par l'asservissement en cap
"/ROV/Temperature"	sensor_msgs::Temperature			Température de l'environnement du ROV
	std_msgs/Header	header		
	float64	temperature	[-40;100]	
	float64	variance	0	
"/ROV/Command"	sensor_msgs::Joy			Commandes envoyées par la console à destination du ROV
	std_msgs/Header	header		
	float32[6]	axes[0-2] : Commandes moteurs (Vertical, Rotation, Translation) axes[3-4] : Commandes Pan & Tilt (Pan, Tilt) axes[5] : Commande puissance des LEDs	axes[0-4] : [0-1024] axes[5] : [0;255]	
	int32[12]	buttons[0] : Verrou P&T buttons[5] : Camera AV/AR buttons[6] : Autodepth buttons[7] : Boost moteur buttons[8] : Auto Forward buttons[9] : Auto Head buttons[10] : Auto down buttons[11] : Auto up	[0;1]	

"/ROV/Moist"	sensor_msgs::RelativeHumidity			Humidité relative de l'air contenu dans le ROV (une augmentation soudaine traduit une entrée d'eau)
	std_msgs/Header	header		
	float64	relative_humidity	[0-100]	
	float64	variance	0	
"/ROV/Roll"	std_msgs::Float32		[0-359,9]	Attitude du ROV (roll en degré)
"/ROV/Pitch"	std_msgs::Float32		[0-359,9]	Attitude du ROV (pitch en degré)

2.1.2 Données souscrites

Topic	Type de donnée	Donnée	Valeur	Désignation
"/ROV/rawMotorCmd"	sensor_msgs/Joy			Message permettant la prise de contrôle en direct des moteurs
	std_msgs/Header	header		
	float32[3]	axes	[-100,0;100,0]	Consigne de puissance des moteurs (Vertical, Babord, Tribord)
	int32[1]	buttons	[0,1]	Activation de la prise de contrôle directe des moteurs
"/ROV/rovCmd"	sensor_msgs/Joy			Message permettant la prise de contrôle du ROV en fonctionnement normal
	std_msgs/Header	header		
	float32[7]	axes	axes[0-4] --> [-100,0;100,0] axes[5-6] : [0,1,2,3]	axes[0-2] : Puissance moteur (Vertical, Rotation, Translation) axes[3-4] : Position Pan & Tilt (Pan, Tilt) axes[5] : Puissance lumineuse de l'éclairage axes[6] : Modes moteurs (paramétrage asservissement)



SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE

Doc N° : **ST-OM-18653-02**

Version : **DRAFT**

Date : **15/05/2018**

Page : 8/12

	int32[14]	button s	[0,1]	buttons[0] : Activation de la prise de contrôle moteur buttons[1] : Activation de la prise de contrôle caméra buttons[2] : Verrouillage Pan&Tilt buttons[3] : Selection caméra Avant/Arrière buttons[4] : Activation de la prise de contrôle des LEDs buttons[5] : Selection des LEDs Avant/Arrière buttons[6] : Allumage/Extinction LEDs buttons[7] : Activation de la prise de contrôle des modes automatiques buttons[8] : Boost moteur buttons[9] : Auto Up buttons[10] : Auto Down buttons[11] : Auto Depth buttons[12] : Auto Head buttons[13] : Auto Forward
Param "ROVIP"	std::string			Adresse IP de la carte ROV
Param "ROVPORT"	int			Port de communication permettant la prise de contrôle du ROV (ne doit pas être modifié)
Param "ConsoleIP"	std::string			Adresse IP de la carte Console
Param "ConsolePort"	int			Port de communication permettant l'espionnage des communications entre ROV et Console (ne doit pas être modifié)

2.2 Node USBL

2.2.1 Données publiées

Topic	Type de donnée	Donnée	Valeur	Désignation
"Seatrac/imu/data"	sensor_msgs::Imu			IMU du transpondeur seatrac
"Seatrac/geometry/data"	geometry_msgs::PoseStamped			Quaternion représentant la position
"Seatrac/data"	seatrac_read_node::seatrac			Message personnalisé Seatrac description



SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE

Doc N° : **ST-OM-18653-02**

Version : **DRAFT**

Date : **15/05/2018**

Page : 9/12

				dans le fichier seatrak.msg
"beacon/data"	seatrak_read_node::beacon			Message personnalisé Beacon description dans le fichier beacon.msg
"reponse/raw"	std_msgs::String			Reponse en raw du port serie

2.2.2 Données souscrites

Topic	Type de donnée	Donnée	Valeur	Désignation
Ping_beacon_2	std_msgs::Bool		[True;False]	Envoi d'un ping à chaque réception d'un True

2.2.3 Messages personnalisés

1) Message SEATRAC

Type de donnée	Donnée	Valeur	Désignation	Matériel
uint8	Msg_id		ID	Seatrak
uint8	Status_output		status	
uint64	timestamp		timestamp	
uint16	env_supply		tension	
int16	env_temp		température	
int32	env_pressure		pression	
int32	env_depth		profondeur	
uint16	env_vos			
int16	att_yaw		ATTITUDE	
int16	att_pitch			
int16	att_roll			
uint8	mag_cal_buf			
bool	mag_cal_valid			
uint32	mag_cal_age			
uint8	mag_cal_fit			
int16	acc_lim_min_x		ACCELEROMETRE	
int16	acc_lim_min_y			
int16	acc_lim_min_z			
int16	acc_lim_max_x			
int16	acc_lim_max_y			
int16	acc_lim_max_z			
int16	ahrs_raw_acc_x		AHRS brute	



SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE

Doc N° : **ST-OM-18653-02**

Version : **DRAFT**

Date : **15/05/2018**

Page : 10/12

int16	ahrs_raw_acc_y			
int16	ahrs_raw_acc_z			
int16	ahrs_raw_mag_x			
int16	ahrs_raw_mag_y			
int16	ahrs_raw_mag_z			
int16	ahrs_raw_gyro_x			
int16	ahrs_raw_gyro_y			
int16	ahrs_raw_gyro_z			
float32	ahrs_comp_acc_x			
float32	ahrs_comp_acc_y			
float32	ahrs_comp_acc_z			
float32	ahrs_comp_mag_x			
float32	ahrs_comp_mag_y			
float32	ahrs_comp_mag_z			
float32	ahrs_comp_gyro_x			
float32	ahrs_comp_gyro_y			
float32	ahrs_comp_gyro_z			

AHRS

2) Message beacon

Type de donnée	Donnée	Valeur	Désignation	Matériel
uint8	msg_id			
uint8	dest_id			
uint8	src_id			
uint8	flags			
uint8	msg_type			
int16	attitude_yaw			
int16	attitude_pitch			
int16	attitude_roll			
int16	depth_local		Profondeur	
int16	vos			
int16	rssi			
uint32	range_count			
int32	range_time			
uint16	range_dist		Distance à la cible	
uint8	usbl_channels			

Beacon



SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE

Doc N° : **ST-OM-18653-02**

Version : **DRAFT**

Date : **15/05/2018**

Page : 11/12

int16	usbl_azimuth		
int16	usbl_elevation		
int16	usbl_fit_error		
int16	position_easting		
int16	position_northing		
int16	position_depth		

2.3 Node DVL

2.3.1 Données publiées

Topic	Type de donnée	Donnée	Valeur	Désignation
"DVL/Raw"	std_msgs::String			Réponse en raw du port TCP , la string est soit en hex ou string suivant le port et donc le type de message choisi.

2.3.2 Données souscrites

Topic	Type de donnée	Donnée	Valeur	Désignation
Param DVL ip	std::string			Adresse IP du dvl
Param DVL port	int32[1]			Port de réponse du dvl (varie en fonction du type de message)

2.4 Node Micron Sonar

2.4.1 Données publiées

Topic	Type de donnée	Donnée	Valeur	Désignation
tritech_micron/scan	PointCloud			Données du scan pour la position actuelle de la tête du sonar
tritech_micron/heading	PoseStamped			Position actuelle de la tête du sonar
tritech_micron/config	TritechMicronConfig			Configuration du sonar publiée lors d'un changement

2.4.2 Visualisation

Les données peuvent être visualisées facilement en utilisant « rviz ».

Pour cela ajouter le topic « tritech_micron/scan » en tant que message « PointCloud ».



SDK ROS Mini-ROV CORAL

MANUEL DE REFERENCE

Doc N° : **ST-OM-18653-02**

Version : **DRAFT**

Date : **15/05/2018**

Page : 12/12

Veiller à régler le « Decay Time » pour qu'il corresponde au temps nécessaire au sonar pour faire un tour.

2.4.3 Paramétrage

Le node est configurable en modifiant directement les paramètres dans les fichiers `cfg/Scan.cfg`.

Il est également possible de les modifier en temps réel en utilisant le système « `dynamic_reconfigure` ». Pour cela utiliser la commande :

```
roslaunch rqt_reconfigure rqt_reconfigure
```