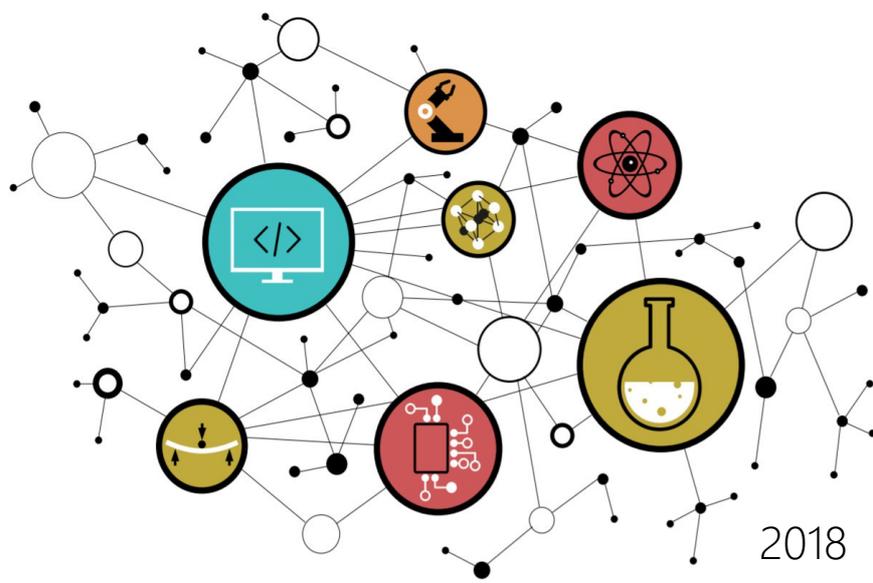


SPECIFICATIONS TECHNIQUES DE BESOIN – v1.0

ROV – Remotely Operated Vehicle



CONTACTS



Établissement

ENSICAEN
6 boulevard Maréchal Juin
CS 45053
14050 CAEN cedex 04

Référents

- hugo descoubes - MOA et directeur technique de la division Embarqué - hugo.descoubes@ensicaen.fr - +33 (0) 2 31 45 27 61 - +33 (0) 6 22 07 21 51
- matthieu denoual - AMOA et directeur technique de la division Mesure - matthieu.denoual@ecole.ensicaen.fr
- Philippe Lefebvre – Directeur technique de la division Réseau - philippe.lefebvre@ensicaen.fr
- Emmanuel Cagniot – Directeur technique de la division Logiciel - emmanuel.cagniot@ensicaen.fr
- Olivier Clouard - Directeur du Service Technique Électronique - olivier.clouard@ensicaen.fr

SOMMAIRE

1. PRESENTATION DU PROJET ROV

- 1.1. Projet ROV
- 1.2. Fonctions techniques du projet ROV
- 1.3. Schéma fonctionnel du ROVER

2. SYSTEMES MECANIQUES ET ENERGIE

- 2.1. Plateforme mobile de surface - SMP-ROV
- 2.2. Bras robot - ARM-ROV
- 2.3. Châssis aluminium
- 2.4. Stockage d'énergie - SOLAR-ROV
- 2.5. Étude LIDAR 3D - EYE-ROV

3. SYSTEMES EMBARQUES

- 3.1. Propulsion du ROVER - SMP-ROV
- 3.2. Contrôle du bras robot - ARM-ROV
- 3.3. Supervision d'énergie - SOLAR-ROV
- 3.4. Réseau de bord - NET-ROV
- 3.5. Systèmes de vision, de communication et de supervision du ROVER - CORTEX-ROV

4. APPLICATION SUR ORDINATEUR

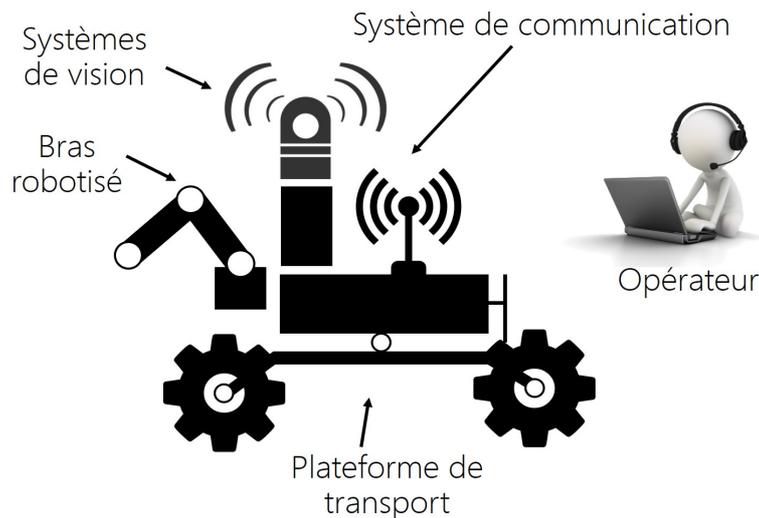
- 4.1. Système de contrôle opérateur - HAND-ROV
- 4.2. Interface opérateur et application logicielle – HMI-ROV

5. ECHEANCIER DES LIVRABLES

1. PRESENTATION DU PROJET ROV

1. PRESENTATION DU PROJET ROV

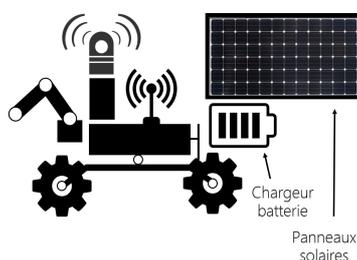
1.1. Projet ROV – Besoins et objectifs



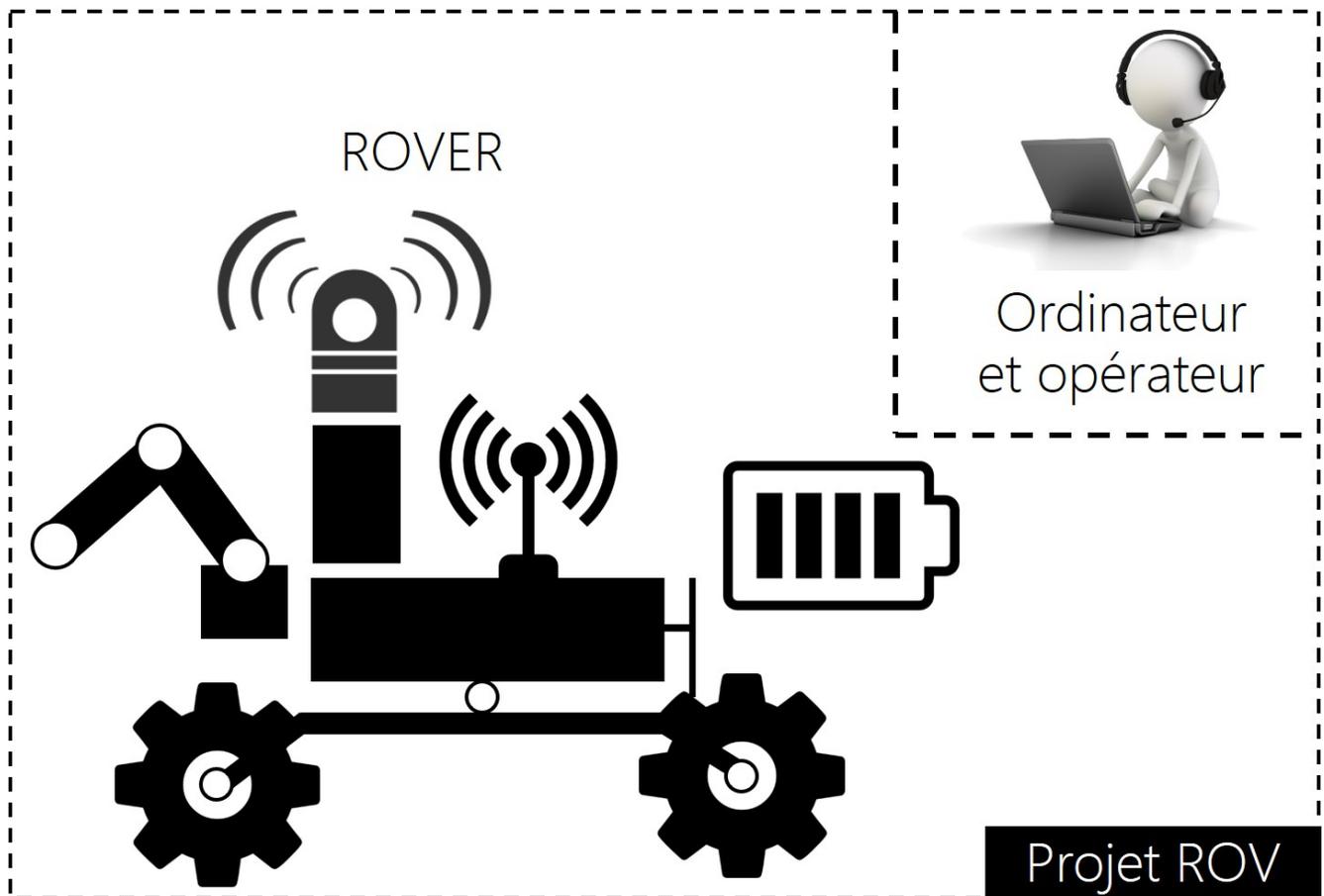
Dans une optique d'indépendance technologique, l'ENSICAEN souhaite se doter d'une solution de transport robuste, pouvant s'adapter à un grand nombre de configuration de terrain et de conditions climatiques. A travers le lancement du projet ROV (Remotely Operated Vehicle), l'objectif premier visé est l'obtention d'un démonstrateur fonctionnel validant ainsi la faisabilité du projet et de nos ambitions. Le ROVER développé proposera 5 fonctionnalités principales :

- Plateforme roulante de déplacement tout-terrain (emport de solutions de mesure)
- Contrôle d'un bras robotisé instrumenté (emport de solutions de mesure)
- Systèmes de vision (caméras fisheyes 360° et lidar)
- Systèmes de communication (wifi pour test et solutions longues distances à prévoir)
- Application de contrôle et de supervision du ROVER par ordinateur. Gestion par un opérateur (ingénieur spécialisé et analyse des mesures par les équipes de recherche)

Le projet devra être pensé dès le départ pour ses évolutions futures. Celles-ci seront explicitées plus en détail dans la suite du document et dans le document de spécifications techniques. Les évolutions futures porteront essentiellement sur des migrations technologiques des solutions de communication longue distance ainsi que sur le développement d'une solution assurant une autonomie en énergie du ROVER par panneaux solaires et un chargeur de batterie embarqué.



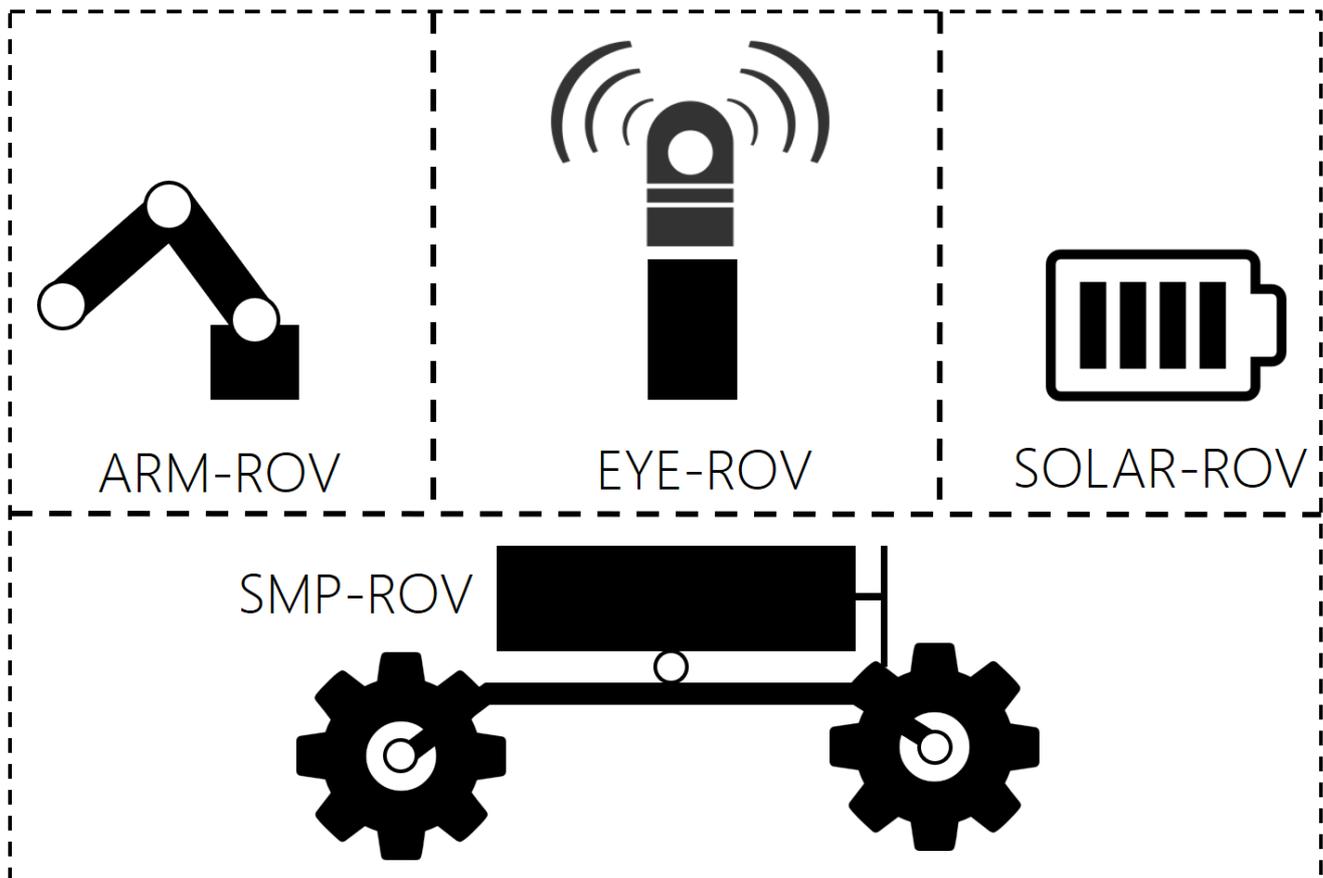
1.2. Fonctions techniques du projet ROV



Cette section du document a pour objectif de poser le jeu de vocabulaire associé aux différentes fonctions métier techniques du projet ROV. Les spécifications techniques de besoins, explicitant notamment les exigences et contraintes physiques comme technologiques suivront la partie présentation. Par la suite et durant la conduite du projet, il nous sera amené de présenter et échanger en présentant l'ensemble du système roulant (mécanique, énergie, électronique, informatique et firmware embarqués) en le nommant ROVER, dénomination plus communément rencontrée en société. Cette section du document a été pensée et est découpée selon la structure même du document de STB (Spécifications Techniques de Besoin). Elle reflète les grands pans des domaines de l'ingénierie et ainsi les différents organes du ROVER :

- Systèmes mécaniques et énergie
- Systèmes embarqués
- Application sur ordinateur

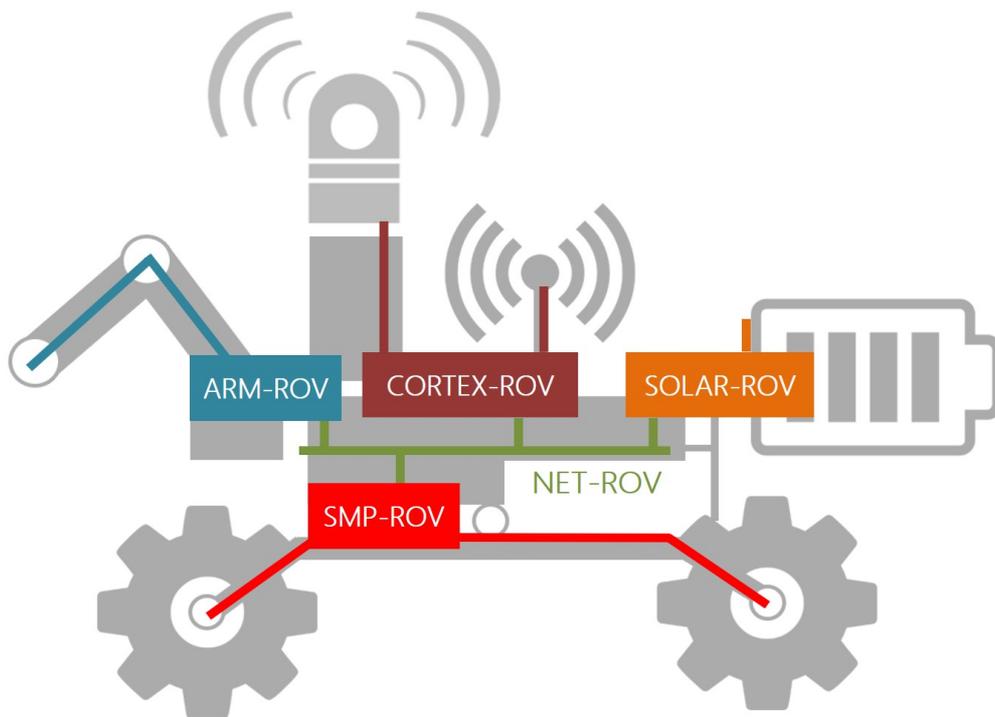
Systèmes mécaniques et énergie



Le ROVER doit être vu comme un système complet répondant à des besoins lui-même constitué d'un ensemble ensemble de sous systèmes répondant conjointement à ces mêmes besoins. Les sous systèmes du ROVER représentant les organes de captation ou d'action du produit. Les terminologies techniques facilitant par la suite les échanges entre collaborateurs sur le projet sont présentées ci-dessous :

- SMP-ROV (Surface Mobility Platform – ROV) : sous système englobant l'ensemble des solutions mécanique, électronique et informatique assurant l'emport de l'ensemble des autres fonctions métier du ROVER
- ARM-ROV : sous système englobant l'ensemble des solutions mécanique, électronique et informatique assurant l'emport de la solution ISIS de mesure sur le bras robotisé
- EYE-ROV (en étude avant projet) : sous système englobant l'ensemble des solutions mécanique, électronique et informatique assurant une cartographie 3D à 360° de l'environnement du ROVER
- SOLAR-ROV (dont étude avant projet) : sous système englobant l'ensemble des solutions mécanique, électrochimique, électronique et informatique assurant une autonomie en énergie du ROVER. Le sous système SOLAR-ROV inclus les solutions énergétiques de stockage, de supervision et la future solution de charge par panneaux solaires

Systèmes embarqués



Cette partie présente l'ensemble des systèmes embarqués (électronique et informatique) enfouis dans le ROVER et assurant notamment la captation, le transport et le traitement de l'information au sein du système. Attention, il sera important dans la suite des travaux de bien contextualiser les sous fonctions des fonctions principales présentées. Par exemple, la fonction principale ARM-ROV englobe la solution électronique et le firmware embarqué à développer, comme le bras robot lui-même nommé AL5D et utilisant un solution technologique développée par la société Lynxmotion (présenté par la suite). Observons les fonctions électroniques embarquées :

- SMP-ROV (Surface Mobility Platform – ROV) : sous système électronique et informatique assurant la propulsion du ROVER
- ARM-ROV : sous système électronique et informatique assurant un asservissement de position par rapport au sol, le contrôle du bras robot et l'emport la solution ISIS de mesure
- SOLAR-ROV (dont étude avant projet) : sous système électronique et informatique proposant une solution de supervision d'énergie par mesure du niveau de batterie (solution de charge par panneaux solaires à prévoir dans les développements)
- NET-ROV (Network ROV) : sous système de transport d'information au sein du ROVER
- CORTEX-ROV : sous système électronique et informatique interfaçant les organes de vision du ROVER (caméras fisheyes à 360° et lidar 360°). Le système CORTEX-ROV est également garant des communications avec l'ordinateur de contrôle et de supervision. Il est le maître des communications sur le réseau embarqué NET-ROV et est le centre de contrôle, de communication et de supervision global du ROVER

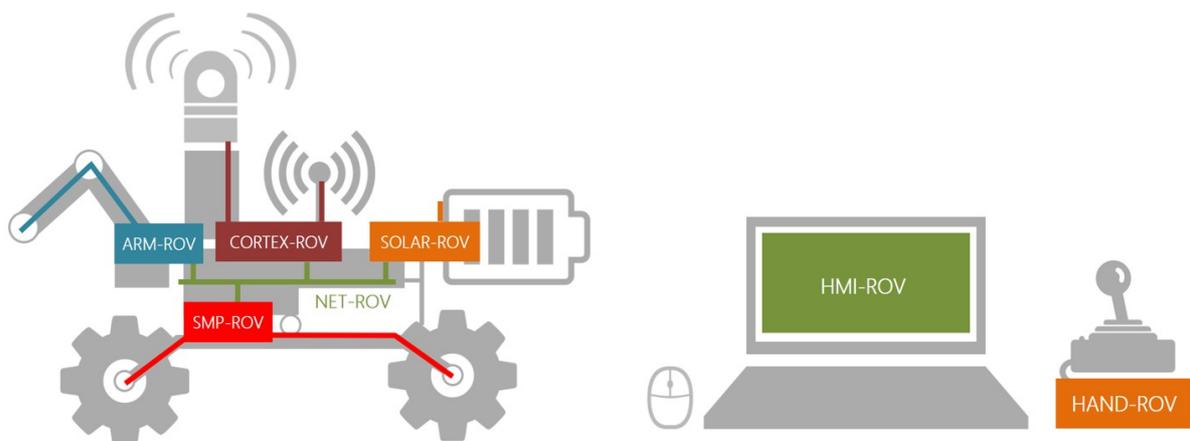
Application sur ordinateur



Cette partie présente l'ensemble des interfaces opérateur. Cette branche du projet est découpée en deux parties, l'interface de supervision/vision et l'interface de contrôle par l'opérateur.

- HAND-ROV : sous système distant de contrôle manuel du ROVER. Le système HAND-ROV inclus le contrôle du ROVER lui-même comme le contrôle du bras robot
- HMI-ROV (Human Machine Interface ROV) : sous système informatique de supervision et de vision du ROVER. L'application logicielle HMI-ROV sera chargée d'afficher voire traiter les informations retournées par les systèmes de vision (caméras fisheyes à 360° et lidar 360°). De même, en cas de défauts retournés par le ROVER, elle offrira des solutions de diagnostic et surveillance de l'état global des systèmes embarqués au sein du ROVER.

Synthèse des fonctions techniques



SMP-ROV
HAND-ROV

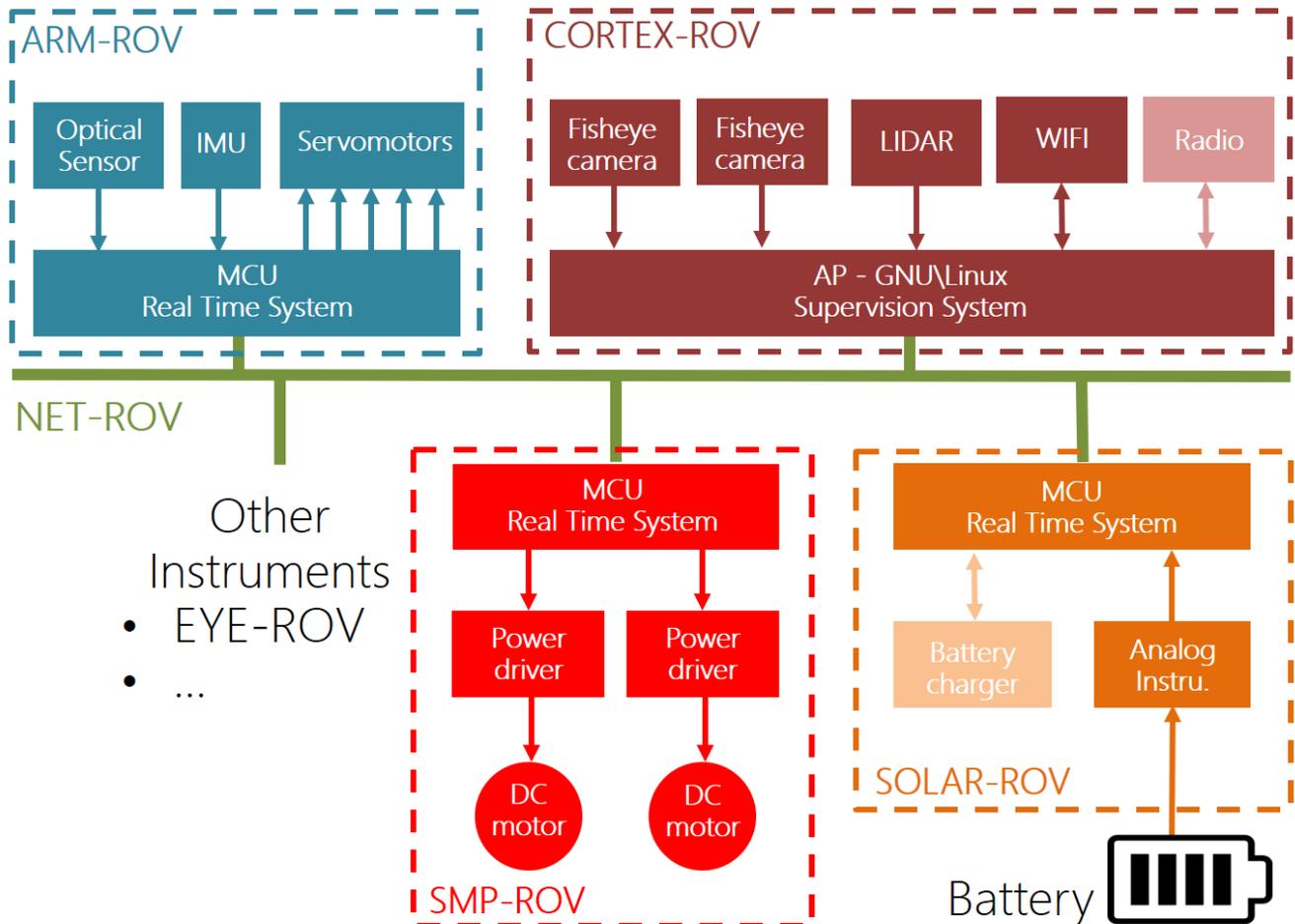
ARM-ROV
HMI-ROV

SOLAR-ROV
CORTEX-ROV

NET-ROV
EYE-ROV (étude)

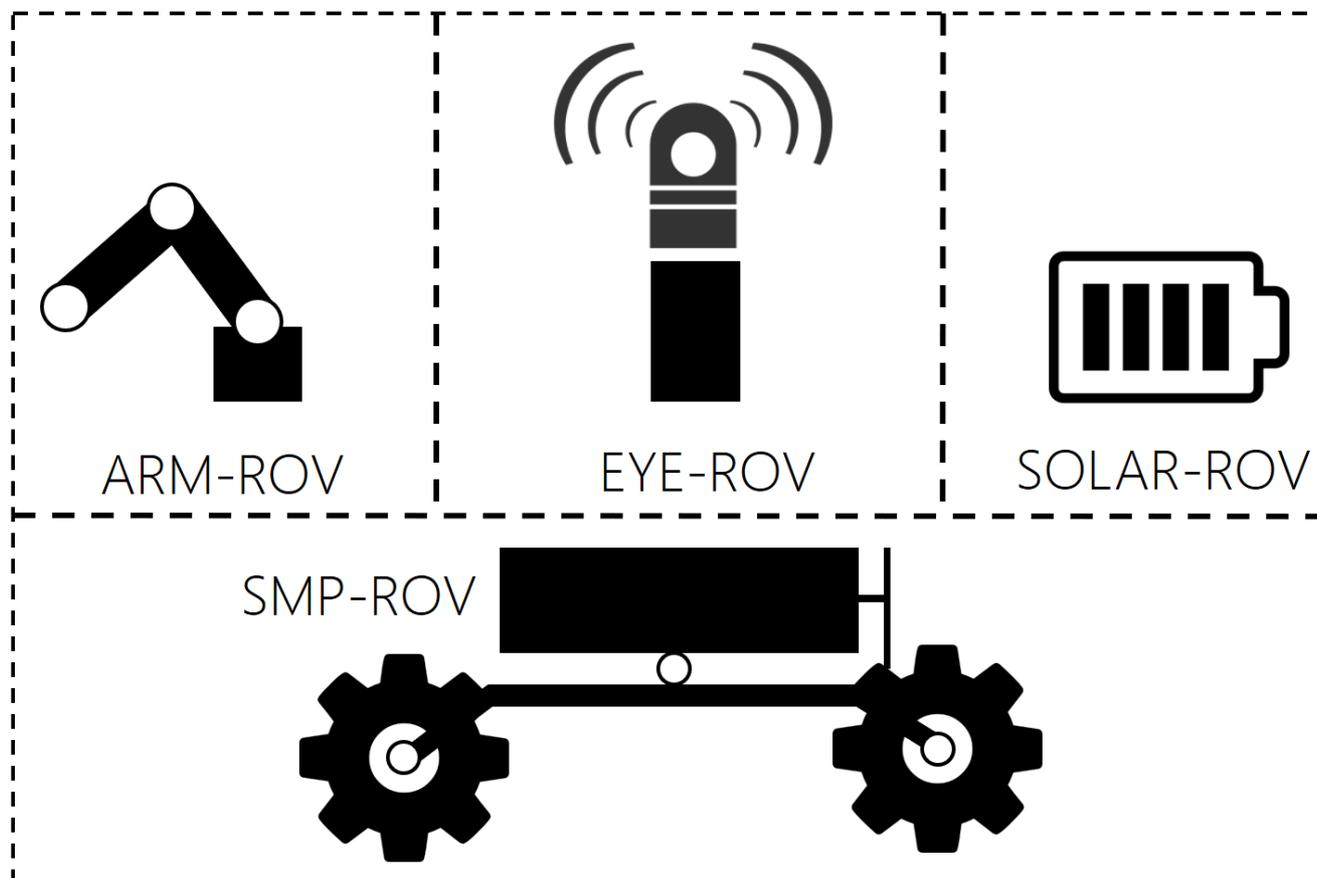
1.3. Schéma fonctionnel du ROVER

La représentation fonctionnelle ci-dessous schématise les différentes fonctionnalités capteurs, actionneurs, instrumentales, chemins et directions de l'information, passerelles de communication, entités de contrôle et de supervision des sous systèmes du ROVER. Chaque sous système sera présenté et spécifié en détail dans la suite du document de STB.



2. SYSTEMES MECANIKES ET ENERGIE

2. SYSTEMES MECANIQUES ET ENERGIE



2.1. Plateforme mobile de surface - SMP-ROV

La plateforme chargée de l'emport de la totalité des sous systèmes du ROVER est l'un des organe du produit sélectionné par nos équipes pour la totalité du projet sur une fenêtre de 3 ans voire plus. La contrainte d'usage est donc impérative. Il s'agit de la solution SMP (Surface Mobility Platform) proposée par la société GEARS.



Spécifications

- Poids : 8,16Kg
- Capacité de charge utile additionnelle : 8,16Kg
- Longueur totale : 60,96cm
- Largeur totale : 60,96cm
- Hauteur totale : 33,02cm
- Base des roues : 12,70cm
- Diamètre des pneus : 17,78cm
- Garde au sol moyenne : 12,70cm (30% hauteur totale)

Spécifications du système de motorisation



La motorisation sera assurée par 4 moteurs à courant continu (brushed motor) GM9236E854 proposés par Pittman. 2 moteurs sur le flanc droit et 2 moteurs sur le flanc gauche. La stratégie de commande se fera en boucle ouverte et sera spécifiée dans la suite du document.

- Tension : 12V
- Courant max (rotor bloqué) : 16,9A
- Rapport de réduction : 65.5:1

2.2. Bras robot - ARM-ROV



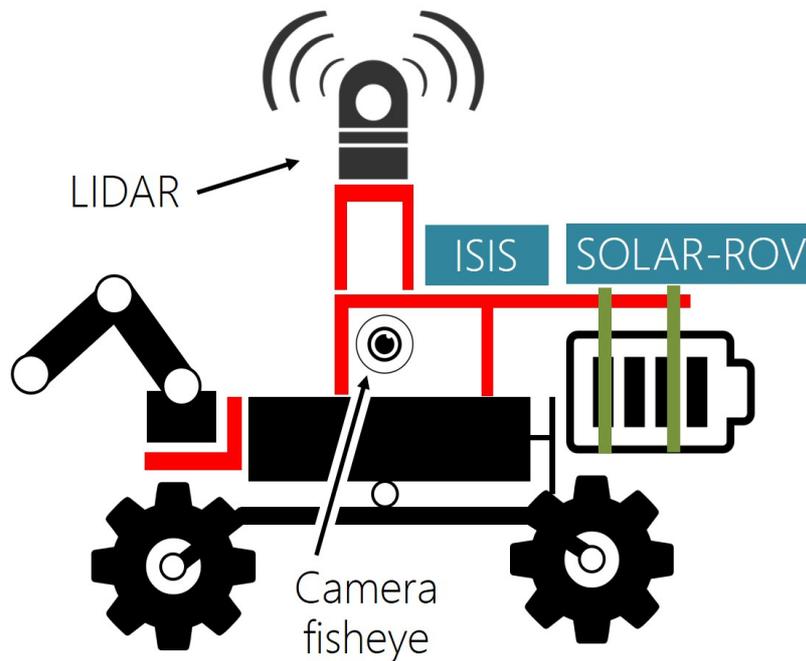
Redéveloppement de la solution électronique et logicielle d'interface

Le bras robot chargé de l'emport de la solution de mesure ISIS a été spécifié par nos équipes. Cette solution pourrait néanmoins être révisée suite à la réalisation de cette première phase du projet si ses capacités d'emport et la robustesse des solutions mécaniques ne devait pas répondre au cahier des charges fonctionnel et aux spécifications techniques associées. La solution retenue est le produit AL5D-PLTW proposé par la société Lynxmotion. L'électronique proposée par défaut avec le produit sera entièrement retirée pour redéveloppement afin de conduire à une solution interne ENSICAEN. Les solutions techniques proposées devront être spécifiées et redéveloppées par les équipes de prestation. Les tâches associées porteront sur la conception d'une nouvelle carte électronique d'interface, la migration de la solution processeur, le développement des solutions logicielles embarquées et la rédaction des documentations associées pour une adaptation au projet ROV. Ces points seront spécifiés dans la suite du document.

Spécifications

- Poids : 907g (avec électronique existante)
- Poids de levage (bras tendu) : 300g
- Nombre d'axes : 5 (avec rotation du poignet)
- Commande par servomoteurs (boucles fermées locales) :
 - 1 x HS-485HB (base)
 - 1 x HS-805BB (ou servomoteur équivalent dans l'épaule)
 - 1 x HS-755HB (coude)
 - 1 x HS-645MG (poignet)
 - 1 x HS-422HB (pince)
- Tension de travail des servomoteurs : de 4,8V à 6V
- Plage de mouvement par axe : 180°
- Précision de mouvement par axe : 0,09°

2.3. Châssis aluminium

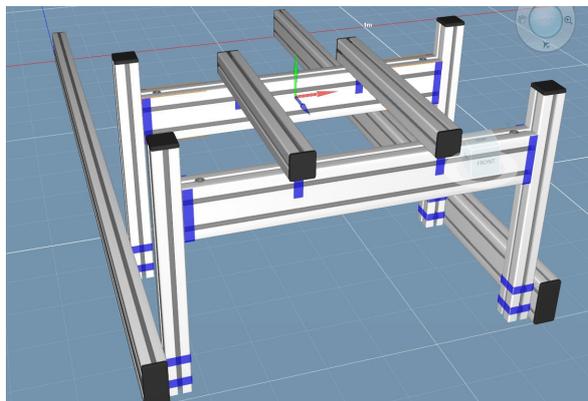


Contraintes

Une étude devra être portée par le prestataire afin de proposer une solution pouvant intégrer les sous systèmes, outils et fonctions du ROVER tout en tenant compte des évolutions futures du projet SERI. Cette étude devra notamment tenir compte des points suivants :

- Capacité de charge du SMP-ROV
- Robustesse et encombrement de la structure proposée
- Intégration de la solution ISIS
- Intégration de la solution SOLAR-ROV
- Intégration potentielle de la solution CORTEX-ROV
- Emport en poupe du ROVER de la batterie au plomb. Première solution d'accroche par ruban auto-agrippant
- Équilibre des masses proue/poupe entre solutions ARM-ROV (avec bras AL5D) et SOLAR-ROV (avec batterie)
- Passage des liens physiques du réseau de bord NET-ROV
- Passage des liens physiques du transport d'énergie vers les solutions électroniques
- Intégration des deux caméras fisheyes 195° des deux côtés du ROVER afin de garantir une vision panoramique à 360°(avec recouvrement)
- Intégration du bras robot AL5D et passage des retours mesure en tête de bras vers les sous systèmes électroniques ARM-ROV et ISIS (intégration de l'électronique de captation, de traitement et de communication dans le châssis de la plateforme GEARS SMP)
- Intégration de la solution LIDAR 2D avec évolution (sans redéfinition et modifications du châssis) vers un LIDAR 3D motorisé par bras robot
- Esthétique globale pour un éventuel démarchage d'investisseur

Solutions techniques et livrables



Des profilés 20mm (6m), équerres, capuchons divers et panneau 6mm (600x600mm) en aluminium ont été commandés et devront servir de base à l'analyse des solutions proposées. De même, il sera demandé une modélisation 3D complète du ROVER pour faciliter les phases d'intégration des solutions envisagées par les équipes de développement (photo ci-dessus non contractuelle). L'étude menée conduira à la production d'un rapport.



2.4. Stockage d'énergie - SOLAR-ROV



Batterie plomb RS
12V - 7Ah



Batterie plomb RS
12V - 3,2Ah

Le stockage d'énergie électrique sera assuré par voie électrochimique à travers l'utilisation d'une batterie au plomb 12V. Deux batteries proposées par Radiospares ont été sélectionnées dans un premier temps pour le projet. Nous serons néanmoins à l'écoute de propositions d'évolution technologique vers une solution proposant un poids et un encombrement réduit. Pour rappel, nous souhaitons faire évoluer le projet ROV d'ici un an en y ajoutant un système de recharge de batterie par panneaux solaires (SOLAR-ROV). Les éventuelles propositions devront tenir compte de ce facteur, point primordial pour nos ambitions futures.

Spécifications

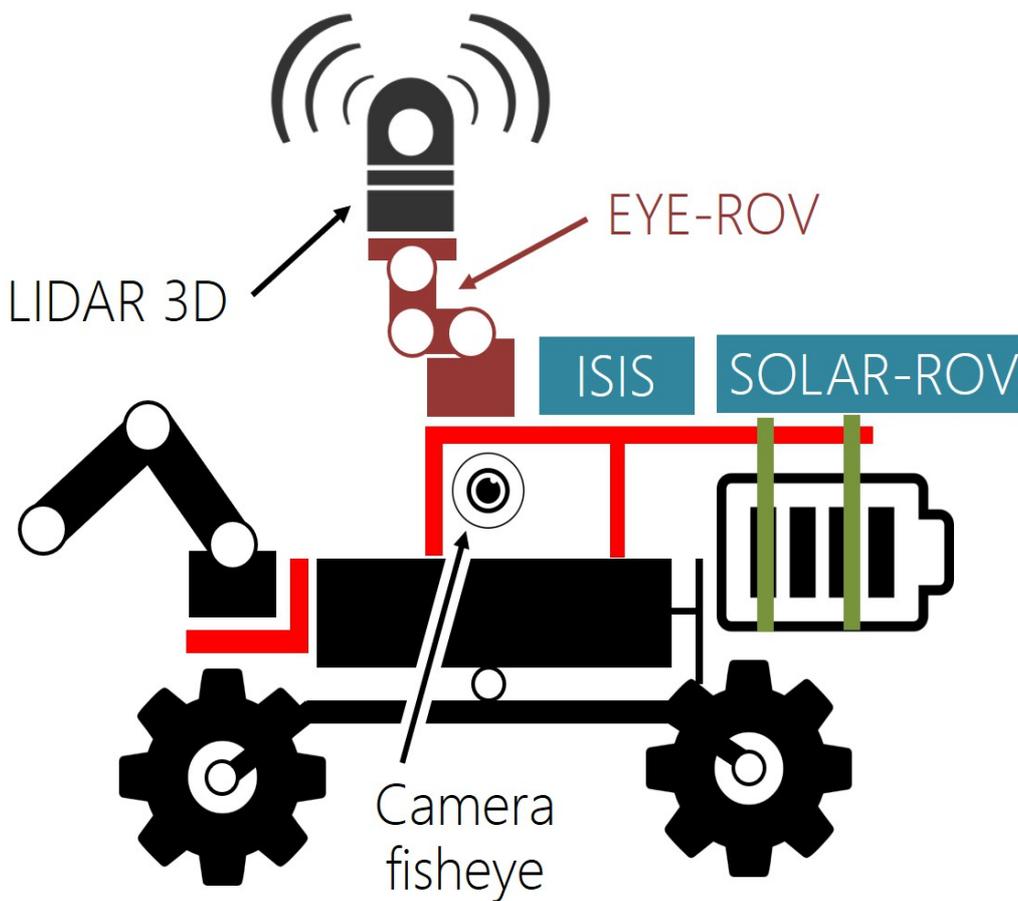
- Modèle : 12V - 7Ah
- Dimensions : 151x65x93,5mm
- Poids : 2,2Kg
- Type de terminaison : Faston F1
- Fabrication : AGM

- Modèle : 12V – 3,2Ah
- Dimensions : 134x67x60,5mm
- Poids : 1,4Kg
- Type de terminaison : Faston F1
- Fabrication : AGM

Solutions techniques et livrables

Pour cette première itération du projet, la batterie sera accrochée au châssis en poupe de ROVER à l'aide de ruban auto-agrippant. Une réflexion devra néanmoins être menée pour proposer une solution d'évolution technologique de la batterie, d'étanchéité au système de stockage et aux connectiques associées ainsi que sur une solution d'accroche au châssis pérenne. Le déploiement sur le système pourra néanmoins être repoussé à l'année suivante. L'étude menée conduira à la production d'un rapport.

2.5. Étude LIDAR 3D - EYE-ROV

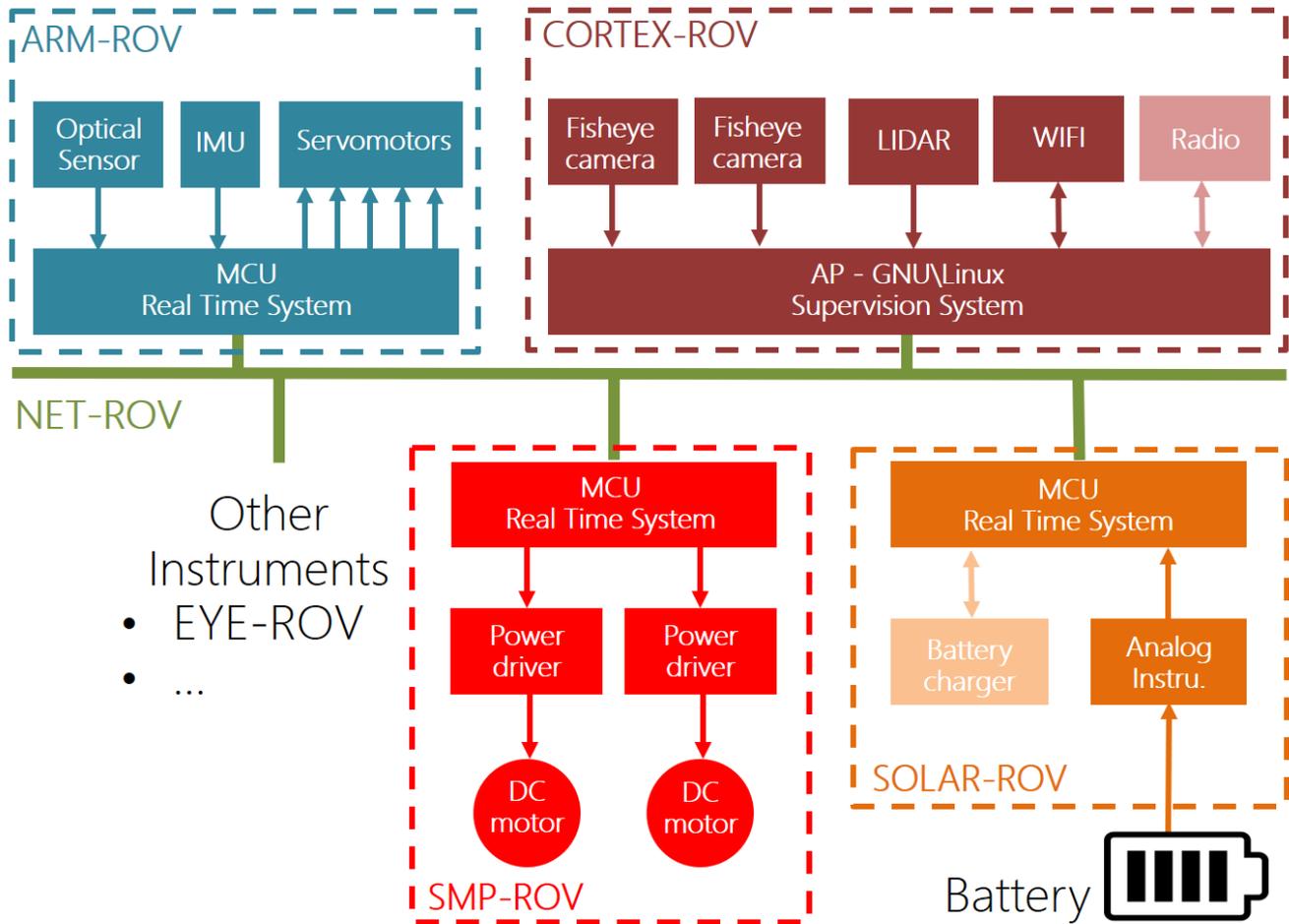


Solutions techniques et livrables

D'ici un an, la solution LIDAR 2D 360° devra évoluer vers une solution LIDAR 3D 360° sans pour autant changer la technologie LIDAR déployée dans le projet (RPLIDAR A3 25m). Une étude devra également être menée sur cette problématique. Notre pré-étude nous montre qu'un bras robotisé 2 axes devrait répondre au besoin. Néanmoins, nous souhaitons asseoir nos premières réflexions et converger vers une solution technologique pouvant être intégrée au projet existant. L'étude menée conduira à la production d'un rapport.

3. SYSTEMES EMBARQUES

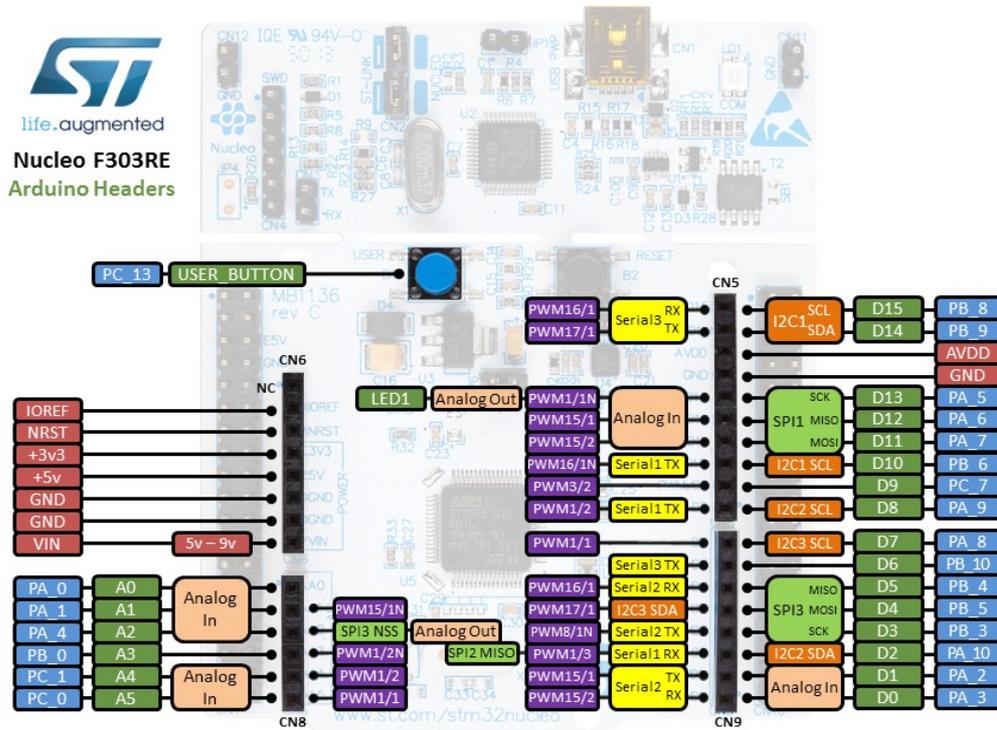
3. SYSTEMES EMBARQUES



Afin d'éviter toute redondance dans la suite des spécifications techniques de besoin, nous allons amorcer cette partie par présenter les composantes matérielles, logicielles, outils et règles communes au développement des sous systèmes esclaves ARM-ROV, SOLAR-ROV, SMP-ROV et la solution encore en cours d'étude EYE-ROV :

- Carte NUCLEO-F303RE et MCU (Micro Controller Unit) STM32F303RE
- RTOS (Real Time Operating System) FreeRTOS et wrapper CMSIS-OS
- Interface série par UART de prototypage et de test
- IDE (Integrated Development Environment) Eclipse et CubeMX
- Arborescence des projets sur STM32 et coding style
- CAO (Conception Assistée par Ordinateur) Eagle

Carte NUCLEO-F303RE et MCU STM32F303RE



La plateforme de prototypage sélectionnée par nos équipes afin de couvrir les besoins de la totalité des sous systèmes électroniques du ROVER est la solution NUCLEO-F303RE embarquant le MCU STM32F303RE proposé par la société Franco-italienne STMicroelectronics. Ce processeur offre notamment pour avantage d'intégrer une interface CAN 2.0B active (évolution future du projet pour le réseau de bord NET-ROV), 3 modules I2C 1Mbps/s, jusqu'à 5 UART, suffisamment de modules PWM, 512Kbytes de flash, 64Kbytes de RAM pour un CPU ARM cadencé à 72MHz avec FPU. Cette solution processeur sera déployée sur l'ensemble des sous systèmes esclaves ARM-ROV, SOLAR-ROV, SMP-ROV et EYE-ROV (plus instruments futurs).

RTOS FreeRTOS et wrapper CMSIS-OS

Chaque application logicielle embarquée sur les sous systèmes esclaves ARM-ROV, SOLAR-ROV, SMP-ROV et EYE-ROV devra être portée sur le système d'exploitation temps réel FreeRTOS v9.0.0. Pour des raisons de portabilité, le wrapper universel CMSIS-OS préconisé et spécifié par la société ARM devra être appliqué.



Interface série par UART de prototypage et de test

Chaque sous système esclave ARM-ROV, SOLAR-ROV, SMP-ROV et EYE-ROV proposera également une interface UART utilisé uniquement durant les phases de prototypage, de développement et de test. Cette interface a pour objectif d'offrir aux équipes de développement une solution de communication (contrôle et diagnostique) avec les fonctions en cours de développement avant le développement, les tests, la validation et le déploiement du réseau de bord NET-ROV. Cette interface de communication ne sera plus utilisée durant les phases de mise en service du ROVER. Néanmoins, les connectiques devront rester facilement accessibles en cas d'intervention sur le ROVER d'un ingénieur expert. Un interpréteur de commande unique sera développé pour la totalité des projets puis déployé avec un jeu de commandes spécifique sur chaque solution logicielle embarquée. Observons l'interface standard retourné au démarrage de chaque système. Les jeux de commandes spécifique par fonction métier (ARM-ROV, SOLAR-ROV, SMP-ROV et EYE-ROV) seront précisés dans la suite des spécifications techniques. Seule les commandes *help* et *version* seront supportées par l'ensemble des sous systèmes. Observons une séquence retournée par la console série de retour. Les parties entre chevrons seront à adapter à chaque projet.

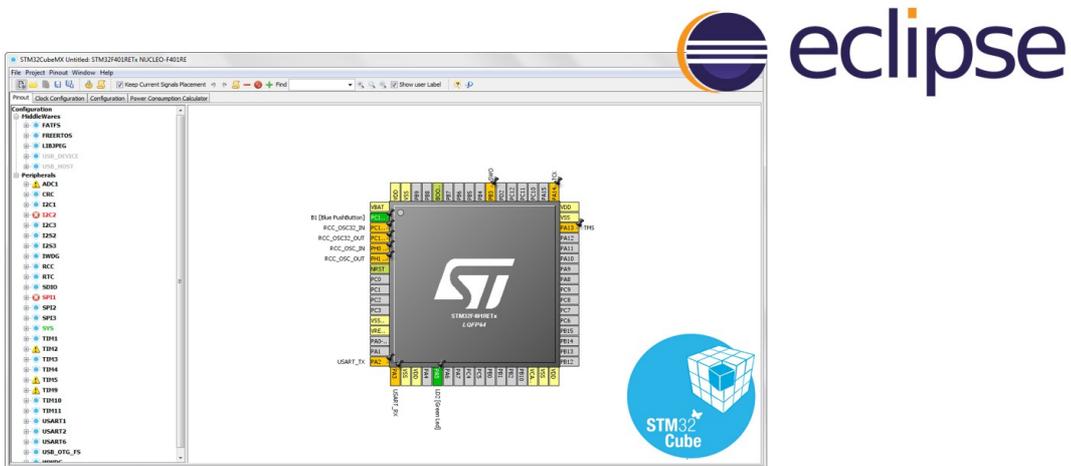
```

ENSICAEN
SERI - Space Ensicaen Rover Instruments
ROVER - <SUBSYSTEM NAME>-ROV subsystem
Version <version number>
Date <date of the last release – day/month/year>
<SUBSYSTEM NAME>-ROV # version
<version number> <date of the last release – day/month/year>
<SUBSYSTEM NAME>-ROV # help
<command name> - <short description>
<command name> - <short description>
<command name> - <short description>
...
<SUBSYSTEM NAME>-ROV # 42
unrecognized command, try help
<SUBSYSTEM NAME>-ROV #
  
```

Des câbles USB vers TTL 3,3V vous seront fournis. Les câbles proposés sont initialement dédiés au projet beagleboneblack et seront à adapter aux PCB développés pour les sous systèmes du ROVER. Les câbles proposés utilisent des composants FTDI standards du marché (support driver Linux).



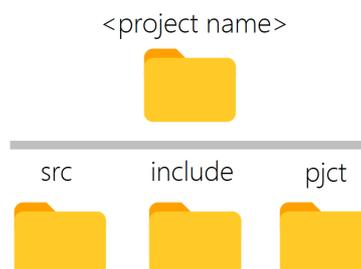
IDE Eclipse et CubeMX



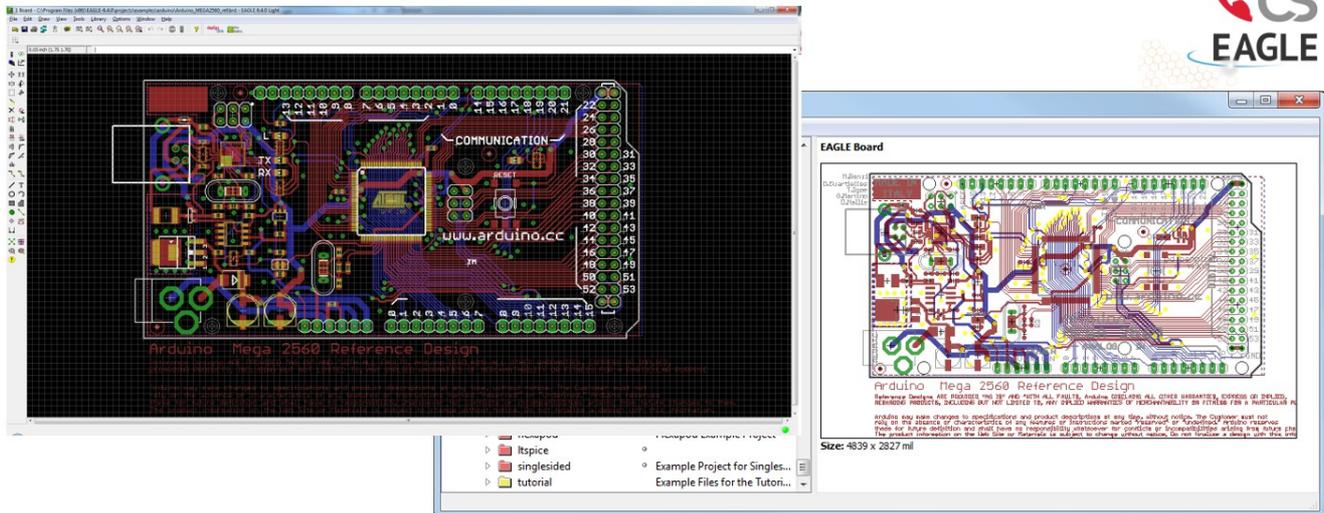
Afin de garantir une réutilisation des projets logiciels développés sur STM32 par nos équipes, nous imposons l'utilisation d'un IDE (Integrated Development Environment) sous Framework Eclipse. Cet outil standard étant déjà utilisé et maîtrisé par nos équipes. Les solutions propriétaires uVision, Keil et la solution en ligne Mbed seront donc proscrites. Les forums et outils libres préconisés par STMicroelectronics sont développés, maintenus et proposés par la société française AC6 (<http://www.openstm32.org/HomePage>). De plus, afin de faciliter les phases de configuration et génération de BSP (Board Support package), CSP (Chip Support Package) et squelettes applicatifs, nous imposons également l'utilisation de l'outil de génération automatique de code CubeMX v9.0. Néanmoins, bien pensé à toujours nettoyer les solutions proposées en ne présentant que du code applicatif utile (commentaires inutiles, code mort, etc à retirer)

Arborescence des projets sur STM32 et coding style

Chaque sous projet unitaire (PWN, ADC, CAN, I2C, UART, etc) devra respecter l'arborescence présentée ci-dessous. De même, l'arborescence globale du projet SERI sera portée sur le Gitlab ENSICAEN et englobera les parties ISIS et ROV. Elle sera spécifiée et définie conjointement avec la société de prestation et la direction technique client. Un coding style ENSICAEN vous sera fourni et à respecter impérativement. De même, des relectures fréquentes de codes seront réalisées par nos équipes. Une bibliothèque documentée doxygen globale par projet (ARM-ROV, SMP-ROV) devra être produite pour une assimilation rapide des solutions présentées.



CAO (Conception Assistée par Ordinateur) Eagle

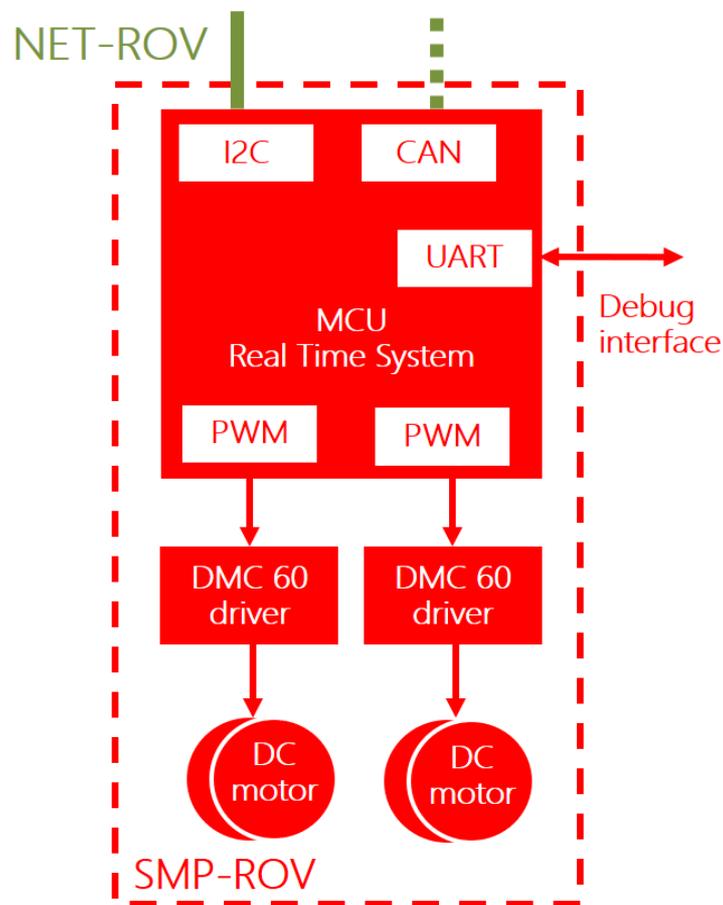


Les différentes conceptions de PCB (Printed Circuit Board) seront réalisées sous EAGLE CAD 7.7 . De même, les fichiers de sortie correspondant aux schémas électriques .SCH et schémas de routage .BRD devront impérativement respecter le format EAGLE 7.xx . Deux techniciens spécialisés seront disponibles en support pour toute question sur les outils de production. Des formations pourront également vous être dispensées par des routeurs experts maîtrisant les solutions de conception et outils de production présents à l'ENSICAEN voire dans les laboratoires de recherche. Il vous sera également potentiellement demandé de produire une modélisation 3D de chaque carte développée afin de valider l'intégration au sein du ROVER (à confirmer en fonction du niveau d'avancement du projet).



De même, avant la réalisation de toute solution PCB, les équipes de prestation devront anticiper les délais d'achat des composants électronique et les délais de production (étaier l'envoi les PCB à produire pour le Service Technique Électronique). Le service achat sera ouvert tous les jours entre 15h et 16h et se fera par l'intermédiaire du directeur du service STE (Service Technique Électronique). Il faudra compter 48h avant la livraison des produits à partir de la demande d'achat en restant attentif à la disponibilité des solutions. De même, il faudra favoriser les commandes chez nos fournisseurs partenaires Radiospares et Farnell. Il sera également potentiellement possible de réaliser une commande unique durant la totalité du projet chez Gotronic ou Conrad avec des délais de livraison supérieurs à 48h.

3.1. Propulsion du ROVER - SMP-ROV



Module de puissance



Les deux modules de puissance retenus par nos équipes d'étude sont les solutions DMC 60 dédiées à la commande de moteur à courant continu.

- Tension d'entrée : de 6V à 28V (tension nominale 12V)
- Intensité nominale : 60A en continu
- Fréquence de la sortie PWM : 15,625KHz
- Dimensions : 70x30x25,4mm

Application logicielle embarquée

Le système SMP-ROV est un système esclave. Toutes les communications seront initiées sur le réseau de bord NET-ROV par le sous système CORTEX-ROV, le maître de toutes les communications embarquées. Le développement se fera en deux phases, la première itération devant anticiper les évolutions futures. Dans un premier temps, les communications se feront sur protocole I2C. Une migration vers une solution industrielle CAN (Control Area Network) est prévue pour l'année suivante. Les solutions logicielles et électroniques sur PCB développées devront donc dès le départ anticiper cette migration et laisser libre le module CAN disponible sur le processeur STM32F303RE. Il en est de même pour les broches utilisées par ce même module qui devront être sorties sur PCB mais non utilisées dans un premier temps.

Le travail principal de la solution électronique embarquée SMP-ROV est de récupérer les ordres de déplacement (vitesse, déplacement avant, déplacement à gauche, déplacement à droite, ralentissement, marche arrière et arrêt) et de les rediriger afin de piloter les moteurs de propulsion. Réception des commandes toutes les 50ms (temps réel dur à respecter). La commande se fera en boucle ouverte. Les estimations de vitesse se feront par simple mesure de la circonférence des roues en tenant du rapport de démultiplication des moteurs. Nous rappelons que l'application développée sera portée sur système temps réel FreeRTOS avec wrapper universel CMSIS-OS.

Interface UART de prototypage et de test

Le système SMP-ROV utilisera des modules PWM 16bits (plages de commandes admissibles). L'interface série de prototypage et de test offrira, en plus des commandes standards précédemment présentées dans le document, les commandes suivantes :

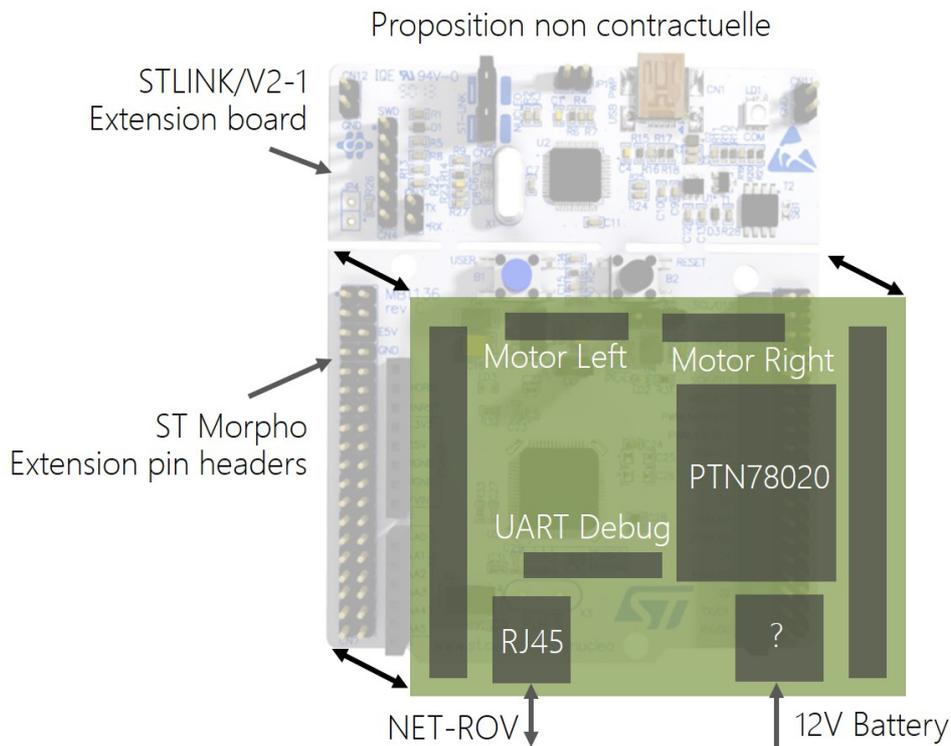
- forward : 4 moteurs en rotation avant puis saisie d'une valeur de commande
- backward : 4 moteurs en rotation arrière puis saisie d'une valeur de commande
- left : 2 moteurs de gauche en rotation puis saisie d'une valeur de commande
- right : 2 moteurs de droite en rotation puis saisie d'une valeur de commande
- stop : arrêt des 4 moteurs

```
SMP-ROV # help
forward – move forward, enter value between 0 and 65535
backward – move backward, enter value between 0 and 65535
left – move left, enter value between 0 and 65535
right – move right, enter value between 0 and 65535
stop – stop rover
version – current release
SMP-ROV # forward
1379
SMP-ROV # stop
SMP-ROV #
```

Diagnostic

Une seconde partie des développements, après validation de l'ensemble des fonctionnalités métier, sera le développement d'une solution de diagnostic dans une optique de supervision distante du système. Le système SMP-ROV devra s'auto-diagnostiquer et enregistrer les LOG d'erreurs. Ces LOG pourront alors être récupérés par le système CORTEX-ROV pour envoi à l'applicatif sur ordinateur HMI-ROV. En cas de requête de diagnostic du ROVER par l'opérateur depuis l'interface applicative, l'ensemble des fonctions métiers SMP-ROV, EYE-ROV, SOLAR-ROV et ARM-ROV seront au repos et le ROVER sera donc stoppé le temps du diagnostic. Ce point fera l'objet d'une évolution des spécifications techniques en fonction du niveau d'avancement du projet.

Circuit imprimé



Une carte fille électronique sera développée (photo ci-dessus non contractuelle) pour intégrer les différentes connectiques et interfaces de communication. Ce PCB sera connecté à la carte NUCLEO-F303RE par le connecteur d'extension ST Morpho. La solution développée devra respecter les dimensions de la carte NUCLEO 70x74,16mm (hors extension de carte pour la sonde STLINK/V2-1). Le réseau de bord NET-ROV se connectera aux systèmes esclaves par des connecteurs RJ45 en prévision de l'utilisation d'un réseau CAN de communication (paire différentielle torsadée). Des câbles ethernet blindés droits seront utilisés pour les échanges d'informations. De même, à terme sur 3 ans, nous souhaitons potentiellement implémenter une redondance sur certains sous systèmes électroniques esclaves critiques du ROVER (SOLAR-ROV, SMP-ROV, etc). Le poids du système global restant un enjeu majeure de nos développements futurs.

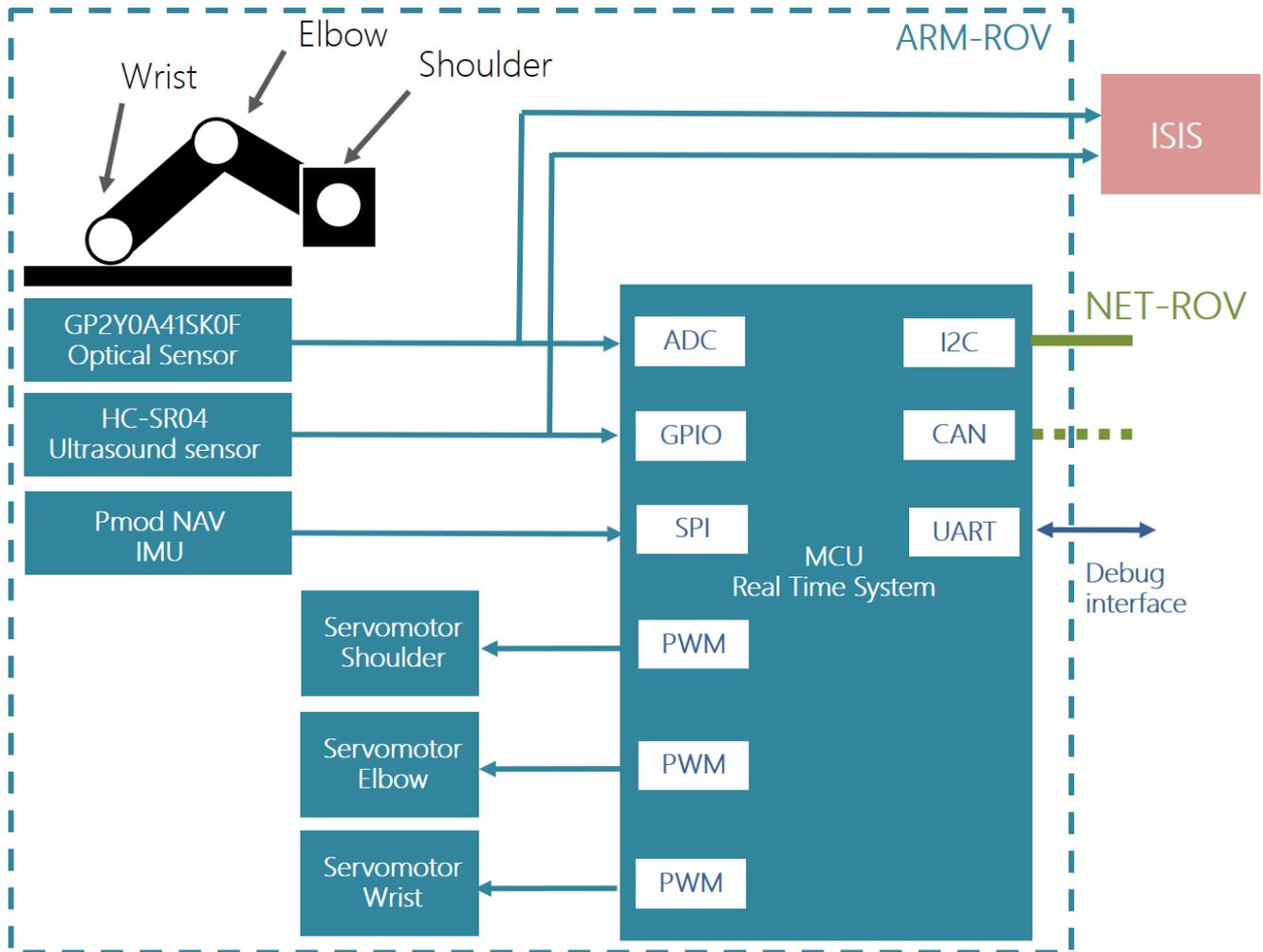
L'intégration du système électronique développé devra également s'intégrer dans un coffret étanche en ABS du marché. Solution à définir et spécifier par le prestataire. La conversion d'énergie 12V vers 5V sur le SMP-ROV sera assurée par le module PTN78020 proposé par Texas Instruments. De même, nos équipes n'ont pas encore spécifier les connecteurs et câbles assurant le transport d'énergie sur le ROVER. A proposer, définir et spécifier par le prestataire.

Livrables

Observons les principaux livrables attendus. Une proposition d'échéancier global sera présentée en fin de ce même document :

- Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement.
- *Formalisation de la modélisation au standard SDL-RT sous environnement Pragmdev. Cette partie sera demandée en fonction du niveau d'avancement du projet*
- Projets de test unitaires par interface (PWM, UART et interpréteur de commande, I2C)
- Projet de test unitaire intégrant la solution RTOS FreeRTOS seul avec wrapper CMSIS-OS
- *Projet baremetal d'intégration des solutions d'interfaces sans RTOS (PWM, UART et interpréteur de commande, I2C). Cette partie n'est pas nécessaire si le livrable suivant est produit dans une version stable*
- Projet complet hors solutions de diagnostic sans intégration au réseau de bord NET-ROV
- Projet complet hors solutions de diagnostic avec intégration au système complet via le réseau de bord NET-ROV
- *En cas de retard au déploiement du réseau NET-ROV et du sous système CORTEX-ROV, développer une solution de test émulant des communication sur le réseau afin de valider les fonctions métier. A valider en fonction du niveau d'avancement du projet*
- Documentation de référence du SMP-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures
- Document d'errata du SMP-ROV
- Documentation doxygen globale au format .chm (voire PDF apprécié) de l'applicatif et bibliothèques SMP-ROV
- *Projet complet avec solutions de diagnostic encore à spécifier. Point non attendu pour cette première phase du projet. Néanmoins, les solutions développées devront dès maintenant tenir compte de cette évolution future.*
- Schéma électrique de la carte fille SMP-ROV
- Schéma de routage de la carte fille SMP-ROV
- Fichiers de de perçage de la carte fille SMP-ROV
- PCB de la carte fille SMP-ROV fonctionnel
- Documentation de référence de la carte fille SMP-ROV. Présentation de la carte fille SMP-ROV, justification du choix de placement des composants, nomenclature des composants, problématiques CEM, etc
- Prototype fonctionnel du projet SMP-ROV intégré au ROVER

3.2. Contrôle du bras robot - ARM-ROV



Application logicielle embarquée

Le système ARM-ROV est un système esclave. Toutes les communications seront initiées sur le réseau de bord NET-ROV par le sous système CORTEX-ROV, le maître de toutes les communications embarquées. Le développement se fera en deux phases, la première itération devant anticiper les évolutions futures. Dans un premier temps, les communications se feront sur protocole I2C. Une migration vers une solution industrielle CAN (Control Area Network) est prévue pour l'année suivante. Les solutions logicielles et électroniques sur PCB développées devront donc dès le départ anticiper cette migration et laisser libre le module CAN disponible sur le processeur STM32F303RE. Il en est de même pour les broches utilisées par ce même module qui devront être sorties sur PCB mais non utilisées dans un premier temps.

Le travail de la solution électronique embarquée ARM-ROV est multiple. Le système est chargé de récupérer les mesures du capteur de position optique (GP2Y0A41SK0F solution faible distance 10cm max – plage 5-10cm), du capteur de position à ultrason (HC-SR04 solution d'approche – plage 10cm et plus) et d'une IMU (Inertial Mesurement Unit) Pmod NAV. Une fois récupérées, les mesures

seront traitées afin de réaliser 2 asservissements. Un asservissement d'assiette par le contrôle du servomoteur de poignet (système de mesure ISIS toujours parallèle au sol) et un asservissement de position après traitement des retours mesure des capteurs de position (distance constante de 10cm par rapport au sol) par contrôle des servomoteurs de l'épaule et du coude. Le système ARM-ROV doit récupérer les consignes de déplacement du bras pour les rediriger vers les servomoteurs afin de piloter les différentes articulations. Les ordres de déplacement seront envoyés depuis l'ordinateur par manipulation du système HAND-ROV. Le système ARM-ROV devra assurer une réception des consignes toutes les 50ms (temps réel dur à respecter). Nous rappelons que l'application développée sera portée sur système temps réel FreeRTOS avec wrapper universel CMSIS-OS. De même, les mesures des capteurs de position seront physiquement renvoyées au système ISIS depuis le PCB d'interconnexion.

Interface UART de prototypage et de test

Le système ARM-ROV utilisera des modules PWM 16bits (plages de commandes admissibles). L'interface série de prototypage et de test offrira, en plus des commandes standards précédemment présentées dans le document, offrira les commandes suivantes :

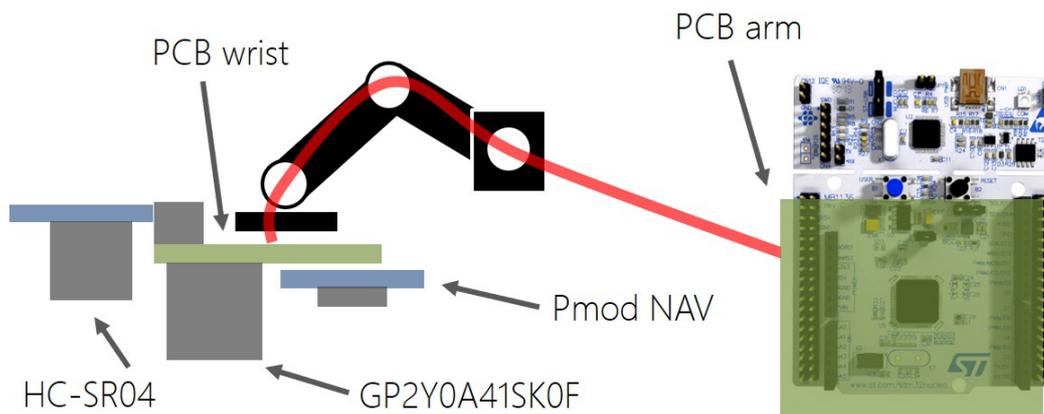
- shoulder : contrôle du servomoteur d'épaule en boucle ouverte puis saisie d'une valeur de commande
- elbow : contrôle du servomoteur de coude en boucle ouverte puis saisie d'une valeur de commande
- wrist : contrôle du servomoteur de poignet en boucle ouverte puis saisie d'une valeur de commande
- auto-wrist : validation de l'asservissement d'assiette à la position courante du bras. Hors position de repos. Attention à ne pas activer cette commande si la mobilité du poignet n'est pas assurée.
- auto-position : validation de l'asservissement de position puis saisie d'une distance de consigne en millimètres. Position minimale 50mm et maximale XXXmm (à valider)
- rest : retour du bras en position de repos (bras replié)

```
ARM-ROV # help
shoulder – move shoulder, enter value between 0 and 65535
elbow – move elbow, enter value between 0 and 65535
wrist – move wrist, enter value between 0 and 65535
auto-wrist – validation of seat control
auto-position – validation of position control, enter value in millimeters between 50 and XXX
rest – rest position
version – current release
ARM-ROV # shoulder
1379
ARM-ROV # auto-wrist
ARM-ROV # auto-position
100
ARM-ROV # rest
ARM-ROV #
```

Diagnostic

Une seconde partie des développements, après validation de l'ensemble des fonctionnalités métier, sera le développement d'une solution de diagnostic dans une optique de supervision distante du système. Le système ARM-ROV devra s'auto-diagnostiquer et enregistrer les LOG d'erreurs. Ces LOG pourront alors être récupérés par le système CORTEX-ROV pour envoi à l'applicatif sur ordinateur HMI-ROV. En cas de requête de diagnostic du ROVER par l'opérateur depuis l'interface applicative, l'ensemble des fonctions métiers SMP-ROV, EYE-ROV, SOLAR-ROV et ARM-ROV seront au repos et le ROVER sera donc stoppé le temps du diagnostic. Ce point fera l'objet d'une évolution des spécifications techniques en fonction du niveau d'avancement du projet.

Circuit imprimé



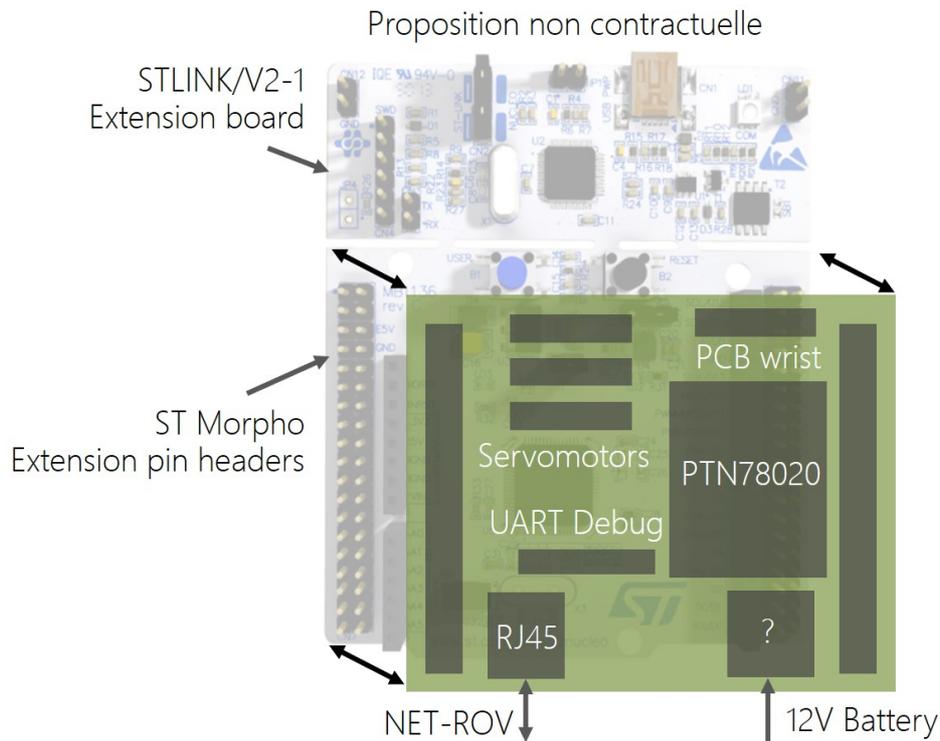
Proposition non contractuelle

Deux circuits imprimés seront développés (PCB wrist et PCB arm). Un premier nommé PCB wrist en bout de bras (au poignet) emportant les capteurs. Un second embarquant la solution électronique de traitement et de communication. Rappelons que les capacités d'emport du bras sont très limitées (300g bras tendu pince incluse, à retirer). De même, les réflexions devront être conjointement menées avec les équipes de développement ISIS. En effet, en plus des capteurs précédemment présentés (contrôle/commande), le système ARM-ROV devra porter la solution de détection de minerai spécifiée dans l'application ISIS.

PCB wrist

Le PCB wrist (photo ci-dessus non contractuelle) assurera l'interconnexion des solutions capteurs implémentées (GP2Y0A41SK0F, HC-SR04 et Pmod NAV) et la redirection des mesures vers le système ARM-ROV. Les mesures de distance (GP2Y0A41SK0F et HC-SR04) seront quant à elles retournées vers le système ISIS MODULE 1. Une attention toute particulière devra être portée au placement des capteurs et au poids de l'ensemble. Il en sera de même pour l'ajout de la solution de détection de minerai. Une réflexion devra être menée sur le câble utilisé pour transporter les mesures et donc une potentielle seconde version de carte (transport en différentiel, etc)

PCB arm



La carte fille PCB arm sera développée (photo ci-dessus non contractuelle) afin d'intégrer les différentes connectiques (capteurs et servomoteurs) et interfaces de communication. Ce PCB sera connecté à la carte NUCLEO-F303RE par le connecteur d'extension ST Morpho. La solution développée devra respecter les dimensions de la carte NUCLEO 70x74,16mm (hors extension de carte pour la sonde STLINK/V2-1). Le réseau de bord NET-ROV se connectera aux systèmes esclaves par des connecteurs RJ45 en prévision de l'utilisation d'un réseau CAN de communication (paire différentielle torsadée). Des câbles ethernet blindés droits seront utilisés pour les échanges d'informations. De même, à terme sur 3 ans, nous souhaitons potentiellement implémenter une redondance sur certains sous systèmes électroniques esclaves critiques du ROVER (SOLAR-ROV, SMP-ROV, etc). Le poids du système global restant un enjeu majeure de nos développements futurs.

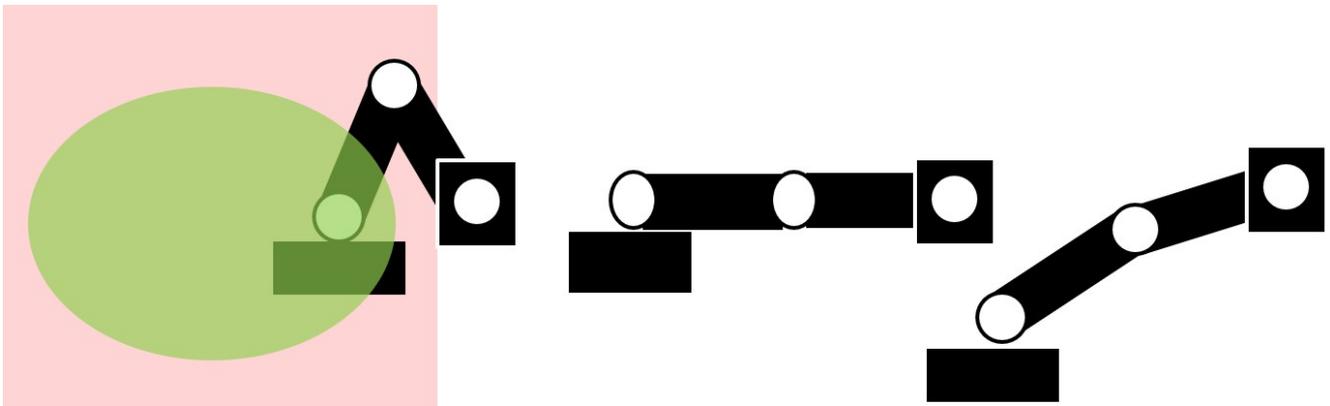
L'intégration du système électronique développé devra également s'intégrer dans un coffret étanche en ABS du marché. Solution à définir et spécifier par le prestataire. La conversion d'énergie 12V vers 5V sur le SM-ROV sera assurée par le module PTN78020 proposé par Texas Instruments. De même, nos équipes n'ont pas encore spécifier les connecteurs et câbles assurant le transport d'énergie sur le ROVER. A proposer, définir et spécifier par le prestataire.

Automatique



Nous attendons des équipes de prestation une caractérisation et une modélisation complète des procédés à asservir. Un rapport d'étude devra être produit afin de présenter les choix et justification des régulateurs implémentés (structure, synthèse, contraintes, etc). Les spécifications techniques et cahier des charges devront également être rédigés sur cette partie et donc inclus au rapport d'étude (marge de phase, marge de gain, critères, contraintes, etc). La conception sera réalisée sous environnement Matlab/Simulink. Il est à noter que les solutions NUCLEO proposées par STMicroelectronics peuvent être directement interfacées depuis Simulink via la toolbox embedded coder. Cela facilitera grandement les phases de prototypage. Un support en automatique pourra être fourni une fois le banc de test développé.

Champ d'action du bras robot



Une fois les asservissements emportés, l'architecture logicielle du système déployée (toutes les fonctionnalités opérationnelles) et les solutions électroniques embarquées (mesures ARM-ROV et ISIS), une zone de sécurité de déplacement du bras devra être garantie par le système ARM-ROV. Quelque soit les consignes reçues depuis l'applicatif HMI-ROV, cette zone (à définir) ne pourra en aucun cas être franchie. Cette zone sera fixée et limitée par les capacités de charge du bras ainsi que pour des raisons de sécurité et de survie du système électronique embarqué en bout de bras.

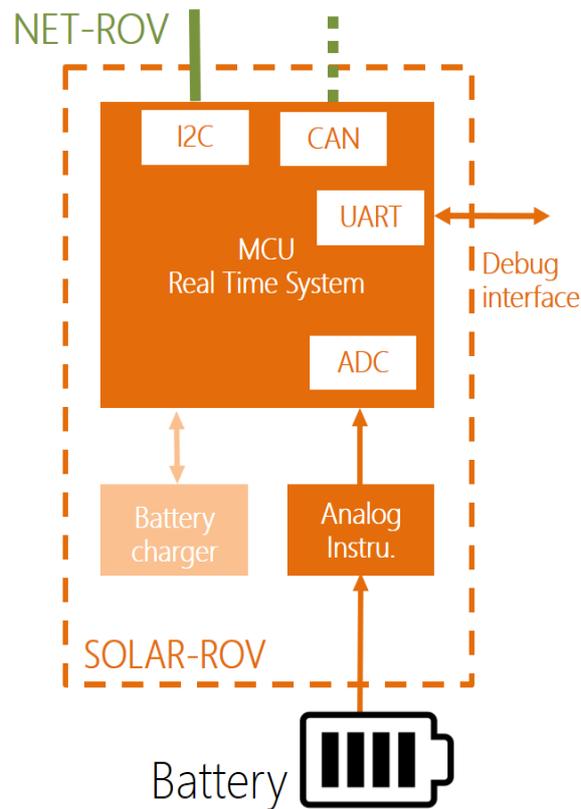
Livrables

Observons les principaux livrables attendus. Une proposition d'échéancier global sera présentée en fin de ce même document :

- Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement.
- *Formalisation de la modélisation au standard SDL-RT sous environnement Pragmdev. Cette partie sera demandée en fonction du niveau d'avancement du projet*
- Projets de test unitaires par interface (PWM, UART et interpréteur de commande, I2C, SPI, ADC et GPIO avec interruption)
- Projet de test unitaire intégrant la solution RTOS FreeRTOS seul avec wrapper CMSIS-OS
- *Projet baremetal d'intégration des solutions d'interfaces sans RTOS (PWM, UART et interpréteur de commande, I2C, SPI, ADC et GPIO avec interruption). Cette partie n'est pas nécessaire si le livrable suivant est produit dans une version stable*
- Projet complet hors solutions de diagnostic sans intégration du réseau de bord NET-ROV
- Projet complet hors solutions de diagnostic avec intégration au système complet via le réseau de bord NET-ROV
- *En cas de retard au déploiement du réseau NET-ROV et du sous système CORTEX-ROV, développer une solution de test émulant des communication sur le réseau afin de valider les fonctions métier. A valider en fonction du niveau d'avancement du projet*
- Documentation de référence du ARM-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures de développement
- Document d'errata du ARM-ROV
- Documentation doxygen globale au format .chm (voire PDF apprécié) de l'applicatif et bibliothèques ARM-ROV
- *Projet complet avec solutions de diagnostic encore à spécifier. Point non attendu pour cette première phase du projet. Néanmoins, les solutions développées devront dès maintenant tenir compte de cette évolution future.*
- Schéma électrique de la carte fille ARM-ROV PCB wrist
- Schéma de routage de la carte fille ARM-ROV PCB wrist
- Fichiers de de perçage de la carte fille ARM-ROV PCB wrist
- PCB wrist de la carte fille fonctionnel
- Schéma électrique de la carte fille ARM-ROV PCB arm
- Schéma de routage de la carte fille ARM-ROV PCB arm
- Fichiers de de perçage de la carte fille ARM-ROV PCB arm
- PCB arm de la carte fille fonctionnel
- Documentation de référence des cartes filles ARM-ROV PCB arm et PCB wrist. Présentation des cartes filles, justification du choix de placement des composants, contraintes CEM, nomenclature des composants, etc
- Simulation du régulateur pour l'asservissement d'assiette sous Matlab/Simulink
- Simulation du régulateur pour l'asservissement de position sous Matlab/Simulink

- Régulateur fonctionnel pour l'asservissement d'assiette sous Matlab/Simulink et intégré sur STM32F303RE
- Régulateur fonctionnel pour l'asservissement de position sous Matlab/Simulink et intégré sur STM32F303RE
- Rapport d'étude présentant et justifiant l'ensemble des solutions de synthèse des régulateurs, les modélisations procédés, cahiers des charges et spécifications techniques, exigences, contraintes, etc
- Prototype fonctionnel du projet ARM-ROV intégré au ROVER

3.3. Supervision d'énergie - SOLAR-ROV

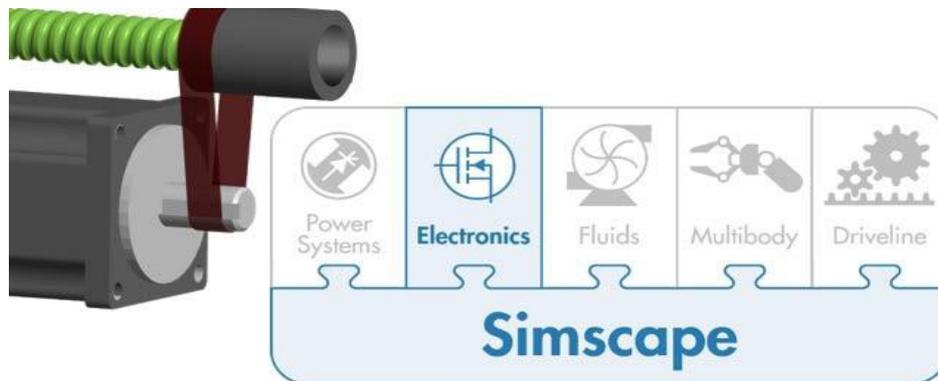


Application logicielle embarquée

Le système SOLAR-ROV est un système esclave. Toutes les communications seront initiées sur le réseau de bord NET-ROV par le sous système CORTEX-ROV, le maître de toutes les communications embarquées. Le développement se fera en deux phases, la première itération devant anticiper les évolutions futures. Dans un premier temps, les communications se feront sur protocole I2C. Une migration vers une solution industrielle CAN (Control Area Network) est prévue pour l'année suivante. Les solutions logicielles et électroniques sur PCB développées devront donc dès le départ anticiper cette migration et laisser libre le module CAN disponible sur le processeur STM32F303RE. Il en est de même pour les broches utilisées par ce même module qui devront être sorties sur PCB mais non utilisées dans un premier temps.

Le travail principal de la solution électronique embarquée SOLAR-ROV est d'assurer une supervision du niveau de batterie. L'équipe de prestation devra également proposer et développer une solution électronique analogique d'instrumentation de la récupération du niveau de tension aux bornes de la batterie. Une étude devra être menée pour proposer une technologie permettant de lire le courant absorbé par l'ensemble du ROVER. En fonction du niveau d'avancement du projet, cette solution pourra être déployée.

Chargeur de batterie



L'étude du chargeur de batterie par panneaux solaires développé et déployé l'année suivante abordera trois points principaux. Le premier traitant de l'évolution de la technologie de batterie utilisée. En effet, les batteries actuellement spécifiées utilise des technologies électrochimique au plomb. Cependant, ces solutions imposent un poids non négligeable. L'étude conduira à une évolution potentielle de la technologie de batterie utilisée.

L'étude devra également conduire à la production d'une simulation sous environnement Simscape Electronics (toolbox Matlab/Simulink proposée à l'ENSI CAEN). La simulation inclura la validation de la solution électronique (avec paramètres des technologies choisies et spécifiées) et des asservissements associés. Elle conduira également à la proposition d'une nomenclature des composants nécessaires à la réalisation de la solution électronique. Le travail débutera par une recherche bibliographique des solutions de charge de batterie en fonction des technologie visées (Ni-MH, Ni-Cd, etc).

La troisième partie de l'étude devra conduire à une proposition de technologie de panneaux solaires adaptable au ROVER, répondant à ses besoins énergétiques et disponible sur le marché. Une estimation des coûts d'achat et fournisseurs potentiels sera également proposée. La batterie utilisée devra respecter les caractéristiques suivantes : 12V, capacité de 7Ah ou supérieur, poids réduit (<1Kg), technologie à définir (hors Lithium-ion et Lithium-polymère pour des problèmes de sécurité).

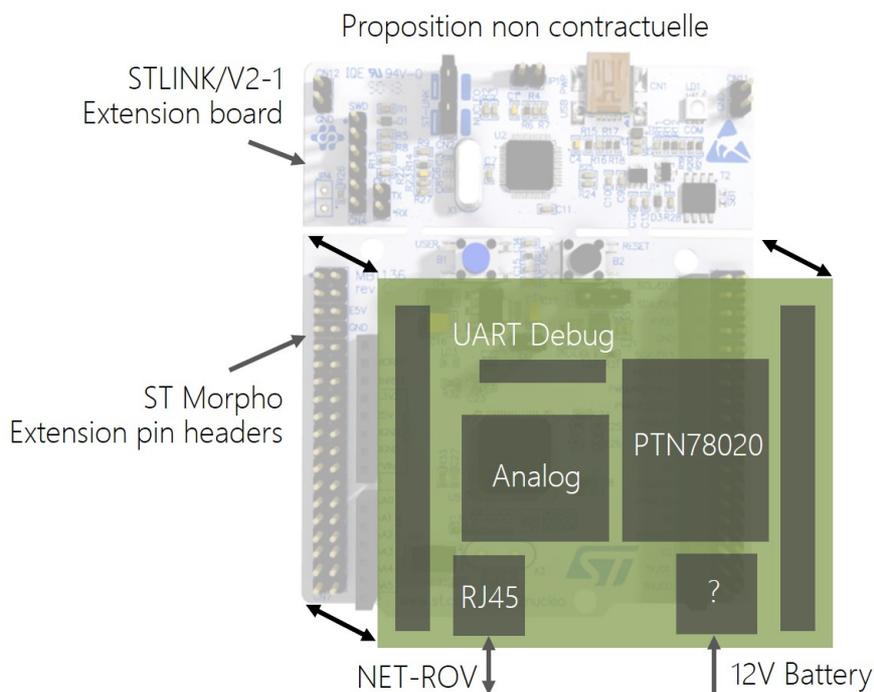
Interface UART de prototypage et de test

L'interface série de prototypage et de test offrira, en plus des commandes standards précédemment présentées dans le document, offrira les commandes suivantes :

- voltage : retourne le niveau de tension de la batterie
- current : retourne le courant absorbé par le ROVER
- power : retourne la puissance absorbée par le ROVER

```
SOLAR-ROV # help
voltage – battery voltage level
current – current absorbed by rover
power – power absorbed by rover
version – current release
SOLAR-ROV # voltage
11,9V
SOLAR-ROV # current
3,7A
SOLAR-ROV # power
44,03W
ARM-ROV #
```

Circuit imprimé



La carte fille sera développée (photo ci-dessus non contractuelle) afin d'intégrer les différentes connectiques et interfaces de communication. Ce PCB sera connecté à la carte NUCLEO-F303RE par le connecteur d'extension ST Morpho. La solution développée devra respecter les dimensions de la carte NUCLEO 70x74,16mm (hors extension de carte pour la sonde STLINK/V2-1). Le réseau de bord NET-ROV se connectera aux systèmes esclaves par des connecteurs RJ45 en prévision de l'utilisation d'un réseau CAN de communication (paire différentielle torsadée). Des câbles ethernet blindés droits seront utilisés pour les échanges d'informations. De même, à terme sur 3 ans, nous souhaitons potentiellement implémenter une redondance sur certains sous systèmes électroniques esclaves critiques du ROVER (SOLAR-ROV, SMP-ROV, etc). Le poids du système global restant un enjeu majeure de nos développements futurs.

L'intégration du système électronique développé devra également s'intégrer dans un coffret étanche en ABS du marché. Solution à définir et spécifier par le prestataire. La conversion d'énergie 12V vers 5V sur le SM-ROV sera assurée par le module PTN78020 proposé par Texas Instruments. De même, nos équipes n'ont pas encore spécifier les connecteurs et câbles assurant le transport d'énergie sur le ROVER. A proposer, définir et spécifier par le prestataire. Une solution étanche en ABS permettant l'emport de la batterie devra être proposée ainsi que la solution de fixation au châssis du ROVER. La solution électronique SOLAR-ROV sera intégrée au plus près de la batterie en poupe du ROVER.

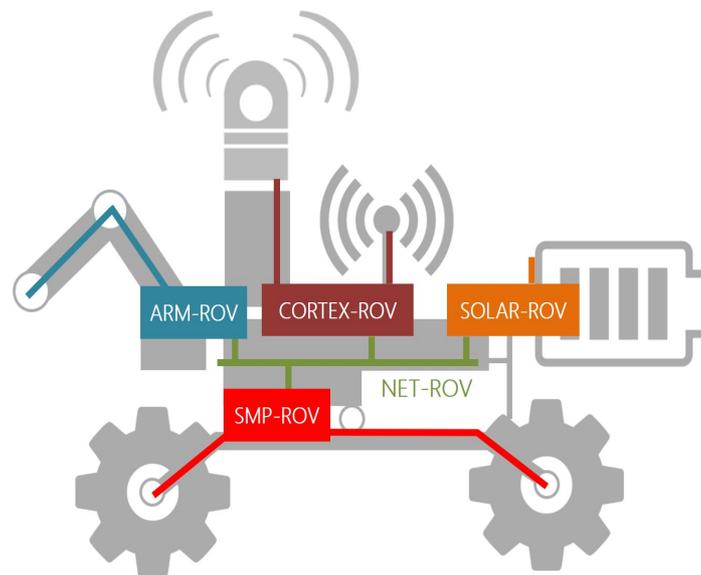
Livrables

Observons les principaux livrables attendus. Une proposition d'échéancier global sera présentée en fin de ce même document :

- Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement.
- *Formalisation de la modélisation au standard SDL-RT sous environnement Pragmadev. Cette partie sera demandée en fonction du niveau d'avancement du projet*
- Projets de test unitaires par interface (ADC, UART et interpréteur de commande, I2C)
- Projet de test unitaire intégrant la solution RTOS FreeRTOS seul avec *wrapper CMSIS-OS*
- *Projet baremetal d'intégration des solutions d'interfaces sans RTOS (ADC, UART et interpréteur de commande, I2C). Cette partie n'est pas nécessaire si le livrable suivant est produit dans une version stable*
- Projet complet hors solutions de diagnostic sans intégration du réseau de bord NET-ROV
- Projet complet hors solutions de diagnostic avec intégration au système complet via le réseau de bord NET-ROV
- *En cas de retard au déploiement du réseau NET-ROV et du sous système CORTEX-ROV, développer une solution de test émulant des communication sur le réseau afin de valider les fonctions métier. A valider en fonction du niveau d'avancement du projet*
- Documentation de référence du SOLAR-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures

- Document d'errata du SOLAR-ROV
- Documentation doxygen globale au format .chm (voire PDF apprécié) de l'applicatif et bibliothèques SOLAR-ROV
- *Projet complet avec solutions de diagnostic encore à spécifier. Point non attendu pour cette première phase du projet. Néanmoins, les solutions développées devront dès maintenant tenir compte de cette évolution future.*
- Schéma électrique de la carte fille SOLAR-ROV
- Schéma de routage de la carte fille SOLAR-ROV
- Fichiers de de perçage de la carte fille SOLAR-ROV
- PCB de la carte fille fonctionnel
- Documentation de référence de la carte fille SOLAR-ROV. Présentation du PCB, justification du choix de placement des composants, problématiques CEM, nomenclature des composants, etc
- Rapport d'étude complet concernant la solution de charge de batterie. Ce document précisera la future technologie de batterie utilisée, proposition de solutions du marché, présentation de la structure du chargeur, étude et synthèse des régulateurs d'asservissement, modélisation du procédé, nomenclature des composants, etc
- Prototype fonctionnel du projet SOLAR-ROV (supervision du niveau de tension seul) intégré au ROVER

3.4. Réseau de bord - NET-ROV



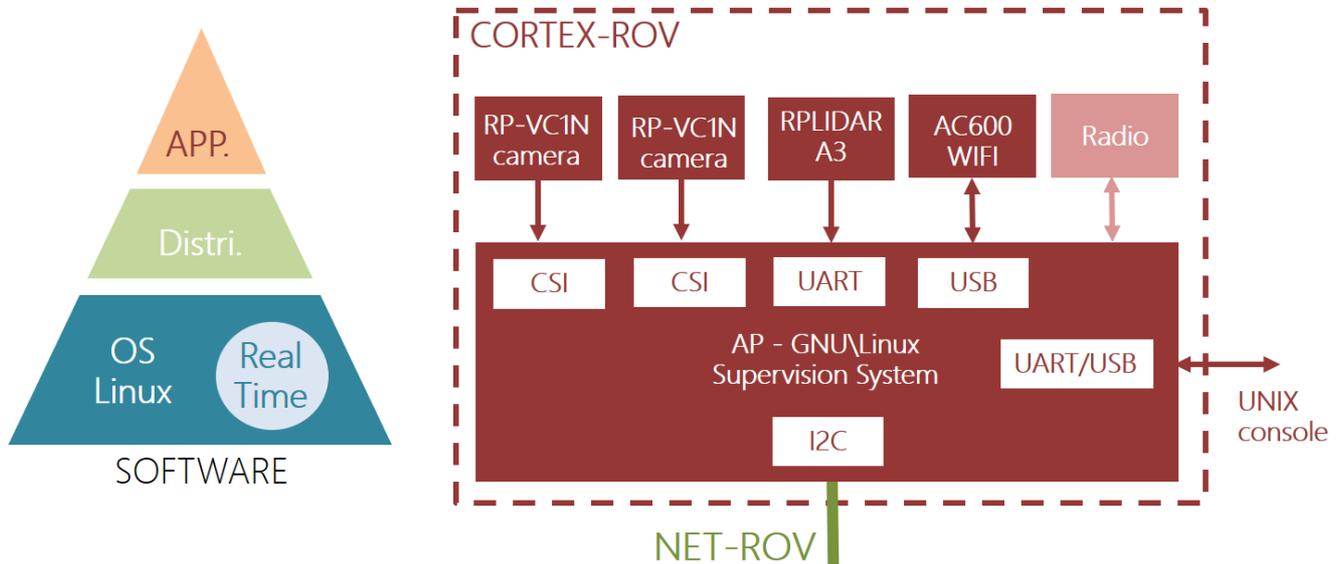
Le sous projet NET-ROV porte sur la réalisation d'un document de référence spécifiant le protocole d'échange au sein du ROVER. Il s'agira d'une communication maître/esclave. Le système CORTEX-ROV sera à l'initiative de toutes les communications. Toute transaction amènera un acquittement par le destinataire du message. Sans réponse de la part d'un système esclave, 3 réémissions successives suivront. Sans réponse à la 4^{ème} tentative (1 initiale + 3 réémissions), la fonction métier sera considérée comme non opérationnelle. Cette information devra être retournée à l'interface HMI-ROV. A chaque seconde, une trame élémentaire spécifique viendra interroger chaque sous système (ARM-ROV, SOLAR-ROV et SMP-ROV) afin de garantir sa présence et son fonctionnement. Un acquittement sera donc réalisé par chaque sous système. Le protocole devra être pensé pour être évolutif et pouvoir assumer l'interfaçage rapide de nouveaux instruments sans avoir à redéfinir les spécifications du NET-ROV. Les jeux de commandes par sous système sera également spécifié dans cette étude. De même, en plus du contrôle des fonctions métier, des trames spécifiques aux diagnostics devront être prévues. Les LOG retournés par les fonctions métier seront de simples fichiers texte. Format encore à définir.

Contraintes

CAN

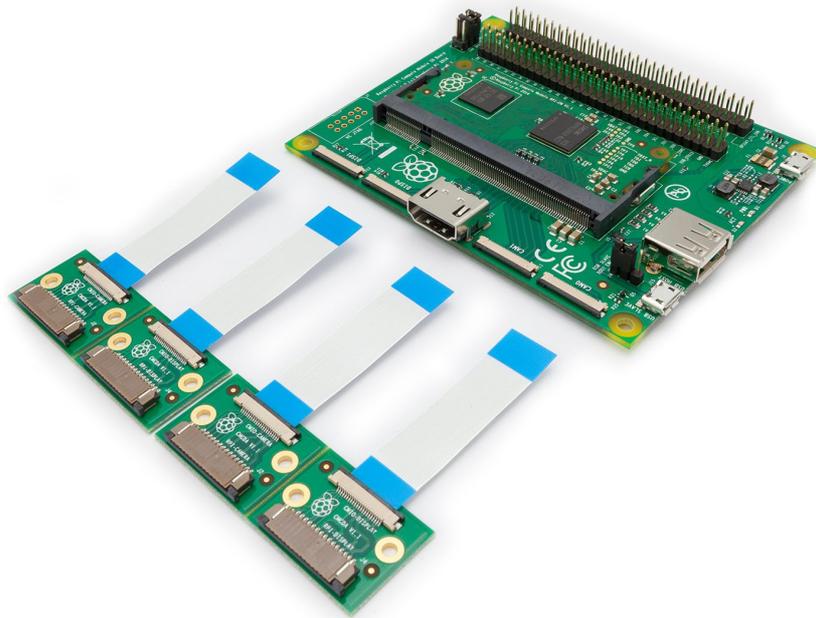
Pour cette première itération du projet, les communications se feront en I2C avec un débit de 1Mbits/s et des trames de 32bits. D'ici un an, une migration est prévue vers le protocole industriel CAN 2.0B. Les trames feront également 32bits pour un débit de 1Mbits/s. Nous demandons également au prestataire de valider la faisabilité de cette solution avec les technologies processeur STM32F303RE et solutions compute module 3 respectivement déployées sur les projets esclaves et maître.

3.5. Systèmes de vision, de communication et de supervision du ROVER - CORTEX-ROV



Application logicielle embarquée

Le sous système CORTEX-ROV est le centre de supervision de l'ensemble du ROVER. Il est le maître de toutes les communications pouvant circuler au sein du réseau de bord NET-ROV. De même, il est le point d'entrée des interfaces de vision du ROVER (2 caméras fisheyes RP-VC1N 195° 8MPs Raspberry Pi v2 et un LIDAR RPLIDAR A3 25m). Il génère également son propre point d'accès réseau dont le SSID (Service Set ID) sera ENSICAEN ROVER. L'appliquatif multi-tâches POSIX développé fera l'objet d'une modélisation complète UML. De même, le listing des dépendances des bibliothèques système et versions associées sera exigé (gestion des dépendances pour la solution OS choisie). Le système CORTEX-ROV sera chargé d'envoyer une image panoramique à 360° par seconde au HMI-ROV. Il en sera de même pour la cartographie LIDAR 2D. Néanmoins, pour la partie LIDAR, le système CORTEX-ROV stockera et construira une carte avec un maximum d'informations sur son environnement (entre chaque seconde) en l'envoi de la prochaine image. L'une des tâches principale du sous système CORTEX-ROV est de rediriger les commandes envoyées depuis le système HMI-ROV vers les organes esclaves ARM-ROV, SMP-ROV et SOLAR-ROV. Dans une optique de supervision de l'état du ROVER, il sera également apte à renvoyer les diagnostics de chaque sous système, lui inclus. Dans un premier, seul le bon fonctionnement de chaque sous système sera retourné (cf. STB NET-ROV). L'application devra être pensée pour être évolutive, notamment au regard de la solution de communication sans fil et instruments futurs portés par le ROVER. En effet, nous souhaitons migrer d'ici 2 ans vers une solution radio longue distance faible débit. De même, de nouveaux instruments (XXX-ROV) seront ajoutés au ROVER au fil des années. CORTEX-ROV et le réseau NET-ROV dont il a la charge devront anticiper ces évolutions et rester facilement adaptables. Rappelons que le système CORTEX-ROV doit garantir l'envoi des commandes et consignes toutes les 50ms (SMP-ROV et ARM-ROV) en respectant des contraintes temps réel dures. De même, il vérifiera la présence de chaque instrument à chaque seconde. Une partie spécifique pour ajouter des capacités temps réel au système d'exploitation sera explicitée par la suite.



La solution matérielle choisie est la carte compute module 3 héritée du projet Raspberry Pi. Il s'agit d'une solution industrielle, clone de la raspberry pi 3, et notamment garantie à la vente jusqu'en 2022. Il sera également demandé de réaliser une étude permettant de migrer la solution I2C vers une solution CAN 2.0B sans changer la solution processeur et la carte compute module 3 déployée. Une console UNIX (vers USB host sur ordinateur) devra également être sortie pour offrir une interface de debug et de supervision filaire pour un ingénieur expert.

Système d'exploitation durci



Les développements seront réalisés sur système GNU/Linux. Dans un premier temps, les développements pourront être portés sur une version standard grand public communément utilisée par les communautés de développeurs (exemple de raspbian, etc). Néanmoins, nous souhaitons à terme, si possible cette année, migrer vers une solution maintenue sur du long terme (au moins 20 ans) voire maintenue par nos équipes. Deux solutions sont à explorer pour validation par nos services. Nous attendons donc les retours rapides de votre étude pour déploiement, nous l'espérons, dès cette année. Présentation des solutions à explorer :

- Distribution maintenue par la Civil Infrastructure Platform (<https://www.cip-project.org/>)
- Projet Yocto pour une gestion ENSICAEN de notre solution GNU/Linux

Système d'exploitation temps réel



Le système CORTEX-ROV imposant des contraintes temps réel à respecter, en complément des spécifications et études demandées concernant l'obtention d'un système d'exploitation durci, nous souhaitons réaliser une pré-étude voire un déploiement d'une solution temps réel dès cette année. A valider en fonction du niveau d'avancement du projet. La pré-étude portera sur le choix de la solution :

- PREEMPT-RT : patch kernel
- Xenomai : co-kernel temps réel

Ce travail est donc à rattacher au travail précédemment demandé concernant le choix de la solution OS (Operating System). Nous rappelons que ces tâches ne sont pas prioritaires et que priorité est donnée à la réalisation d'un prototype fonctionnel avec solution OS du marché. Selon le niveau d'avancement du projet, des benchmarks de stress, de conformité et de mesure seront demandés.

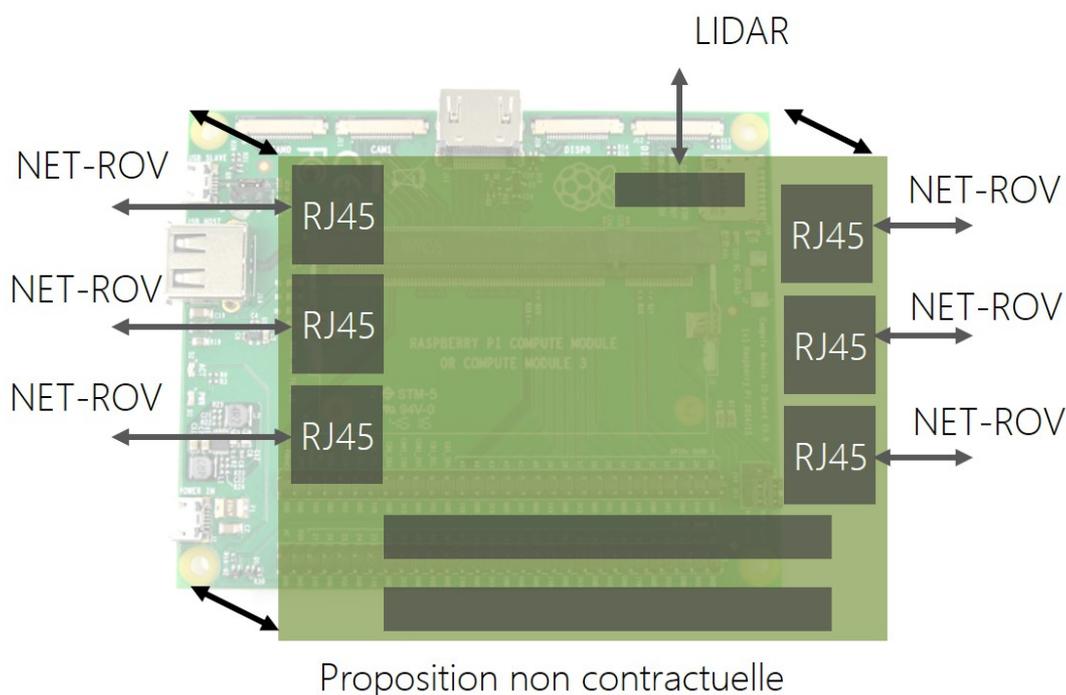
Protocole d'échange en WIFI

Une réflexion devra également être menée sur les stratégies et protocoles applicatifs d'échanges entre CORTEX-ROV et HMI-ROV. Cette étude conduira à la rédaction d'un document de référence. Nous souhaitons potentiellement à l'avenir appliquer des solutions de chiffrement des données échangées (données descendantes HMI-ROV vers CORTEX-ROV) et des techniques compression des images pour les données montantes (CORTEX-ROV vers HMI-ROV). En effet, le déploiement d'une communication par ondes radio réduira grandement les débits accessibles (transferts des images à chaque seconde). L'étude et les choix effectués devront tenir compte de ces aspects.

Nous souhaitons également valider la faisabilité d'établir un point d'accès réseau via le module WIFI AC600 DWA-172. Ce point devra être validé très rapidement par l'équipe de prestation. Si cela n'est pas possible, d'autres solutions WIFI seront fournies.



Circuit imprimé



La carte fille sera développée (photo ci-dessus non contractuelle) afin d'intégrer les différentes connectiques et interfaces de communication. Ce PCB sera connecté à la carte compute module 3 par les connecteurs d'extension J5 et J6. La solution développée devra respecter les dimensions de la carte compute module 3. Cette carte sera le point d'entrée de toutes les connexions au réseau NET-ROV. Les connexions aux systèmes esclaves se feront par des connecteurs RJ45 en prévision de l'utilisation d'un réseau CAN de communication (paire différentielle torsadée). Des câbles ethernet blindés droits seront utilisés pour les échanges d'informations. De même, à terme sur 3 ans, nous souhaitons potentiellement implémenter une redondance sur certains sous systèmes électroniques esclaves critiques du ROVER (SOLAR-ROV, SMP-ROV, etc). Le poids du système global restant un enjeu majeure de nos développements futurs. Au moins 6 connecteurs (8 maximum) d'interface RJ45 NET-ROV devront être proposés sur le PCB.

L'intégration du système électronique développé devra également s'intégrer dans un coffret étanche en ABS du marché. Solution à définir et spécifier par le prestataire. Une attention toute particulière sera apportée au placement des caméras fisheyes 360° RP-VC1N afin d'offrir une vision panoramique optimale. La solution châssis devra déjà être spécifiée voire montée par les équipes de prestation. La conversion d'énergie 12V vers 5V sur le CORTEX-ROV sera assurée par le module PTN78020 proposé par Texas Instruments. De même, nos équipes n'ont pas encore spécifier les connecteurs et câbles assurant le transport d'énergie sur le ROVER. A proposer, définir et spécifier par le prestataire.

Livrables

Observons les principaux livrables attendus. Une proposition d'échéancier global sera présentée en fin de ce même document :

- Modélisation logicielle de l'applicatif POSIX en UML si développement par langage orienté objet. Sinon, utiliser le même formalisme (unique et standard au projet) que pour les sous systèmes ARM-ROV, SMP-ROV et CORTEX-ROV. Une modélisation SysML pourra potentiellement être demandées.
- Projets de test unitaires par interface (caméras fisheyes 360°, LIDAR, I2C et WIFI en access point)
- Projet complet hors solutions de diagnostic sans réseau de bord NET-ROV
- Projet complet hors solutions de diagnostic avec réseau de bord NET-ROV
- Documentation de référence du CORTEX-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures
- Document d'errata du CORTEX-ROV
- Documentation doxygen globale au format .chm (voire PDF apprécié) de l'applicatif et bibliothèques CORTEX-ROV
- *Projet complet avec solutions de diagnostic encore à spécifier. Point non attendu pour cette première phase du projet. Néanmoins, les solutions développées devront dès maintenant tenir compte de cette évolution future.*
- Schéma électrique de la carte fille CORTEX-ROV
- Schéma de routage de la carte fille SCORTEX -ROV
- Fichiers de de perçage de la carte fille CORTEX-ROV
- PCB de la carte fille fonctionnel
- Documentation de référence de la carte fille CORTEX-ROV. Présentation de la carte fille, justification du choix de placement des composants, nomenclature des composants, etc
- Document de spécification du réseau de bord NET-ROV
- Document de spécification du réseau WIFI
- Rapport d'étude pour le déploiement d'une distribution durcie, voire avec kernel temps réel, et maintenue sur le long terme. L'étude sur le déploiement sur une solution kernel temps réel devra apparaître dans ce rapport d'étude.
- *Déploiement d'une solution système durcie et maintenue sur le long terme en fonction du niveau d'avancement du projet*
- *Document de maintenance de la distribution. Par exemple, documentation concernant le listing des dépendances de l'application (version des packages) et justification de la configuration du projet Yocto fourni, versions kernel choisie, etc. Ce document pourra être rattaché au document d'étude pour le déploiement. A réaliser en fonction du niveau d'avancement du projet*
- Prototypage fonctionnel du projet CORTEX-ROV avec réseau NET-ROV opérationnel intégré au ROVER

4. APPLICATION SUR ORDINATEUR

4. APPLICATION SUR ORDINATEUR



4.1. Système de contrôle opérateur - HAND-ROV



La solution de contrôle HAND-ROV du ROVER par l'opérateur sera assurée par le joystick Thrustmaster Hotas Warthog. L'application logicielle sera développée sur système GNU/Linux. De même, l'application sera, si nécessaire, entièrement manipulable sans souris et depuis les interfaces de contrôle du HAND-ROV. Les différentes commandes et fonctionnalités présentées par la suite concernant le HAND-ROV seront à rattacher à la présentation du projet HMI-ROV présentant l'interface graphique utilisateur et les actions sur l'IHM (Interface Homme Machine) relatives aux actions de l'opérateur sur le HAND-ROV. Nous attendons de la part du prestataire des retours critiques et propositions d'évolutions quant aux commandes et fonctionnalités initialement proposées par l'ENSICAEN.

De même, le travail sur le HAND-ROV débutera par une validation du support driver Linux et de chaque commande et interface proposée (interrupteurs, joysticks, boutons poussoirs, etc). En cas de non support complet du HAND-ROV, le listing des commandes supportées sera demandé ainsi que l'identification des sources du driver utilisé (kernel.org ou autre dépôt en ligne).

Fonctionnalités HAND-ROV Right

J1 : Déplacement du ROVER
J1 : Déplacement du bras robot



PB2 : Sélection d' environnement
(LIDAR, vidéo 360°, diagnostique, etc)

PB3 : Capture vidéo
(démarrer/arrêter)



J2 : Déplacement du curseur
sur HMI-ROV

J3 : Glissement de la vidéo 360° dans
la fenêtre principale (gauche/droite)



PB4 : Sélection d' une zone pour agrandissement
sur la fenêtre principale (début/fin)

Les fonctionnalités présentées dans cette partie sont à rattacher à la présentation du projet HMI-ROV dans la suite du document de spécifications (Jx ou Joystick x, PBx ou Push Button x, SW x ou Switch x et STx ou Stick x). Voici un descriptif de chaque fonctionnalité :

- J1 (Rover) : Selon le mode sélectionné par l'opérateur via SW1 (mode ROVER), ordres de déplacement du ROVER (avant, gauche, arrière et droite). En position de repos de J1, le ROVER sera à l'arrêt.
- J1 (bras robot) : Selon le mode sélectionné par l'opérateur via SW1 (mode ARM), ordres de déplacement du bras robot en translation (avant, arrière). En position de repos de J1, le bras restera à la dernière position occupée. En restant en position arrière, le bras finira sa course en position de repos. Pour rappel, une zone de sécurité du champ de déplacement du bras est à garantir par le système ARM-ROV. Quelque soit les ordres envoyés par l'opérateur depuis le HAND-ROV, cette zone ne pourra être franchie. Ce point est impératif.
- J2 : Joystick de déplacement du curseur de sélection de zone sur l'HMI-ROV. De futurs usages seront à prévoir, toujours dans une optique de déplacement du curseur applicatif.
- J3 : Joystick de déplacement de la fenêtre principale (vers la gauche ou la droite) durant la sélection de l'environnement de vision vidéo à 360°. La rotation se fera sous forme d'une rotation circulaire sans fin. L'avant du Rover sera repéré par un indicateur (triangle noir).

- PB2 : Sélection de l'environnement de supervision et de l'affichage en fenêtre principale (LIDAR, vidéo 360° et diagnostique) par action sur le bouton poussoir (appuyer puis relâcher). 1 seconde de temps mort est laissé à l'opérateur pour commuter d'environnement. Par exemple, 3 actions successives avec moins d'une seconde d'attente entre chaque action n'aura aucun effet sur l'HMI et le ROVER. Autre exemple, si l'utilisateur utilise l'environnement LIDAR, 2 actions successives avec moins d'une seconde d'attente entre chaque action le fait basculer en environnement de diagnostique. Une action seule l'aurait fait basculer en environnement vidéo 360°.
- PB3 : Démarrage d'une capture vidéo (fenêtre principale) par une simple action sur le bouton poussoir PB3 (appuyer puis relâcher). Une seconde action permet d'arrêter la capture. Le fichier sera alors automatiquement sauvegardé sur le machine host.
- PB4 : Sélection d'une zone pour agrandissement. Une première action sur le bouton poussoir PB4 (appuyer puis relâcher) permet de fixer le début de la zone (en haut à gauche), après manipulation de J2 afin de sélectionner un zone, une seconde action sur PB4 (appuyer puis relâcher) permet de fixer la fin de la zone à agrandir (en bas à droite). La zone à afficher apparaît automatiquement en fenêtre principale, le bandeau contextuel d'environnement se met à jour une fois l'affichage réalisé. Une troisième action permettra de repasser en plein écran.

Fonctionnalités HAND-ROV Left

ST1 : Consigne de vitesse ROVER 1KM/h
ST1 : Consigne de position du bras

ST2 : Consigne de vitesse ROVER 3Km/h Max.



SW1 : Sélection du mode J1
(contrôle ROVER ou bras robot)

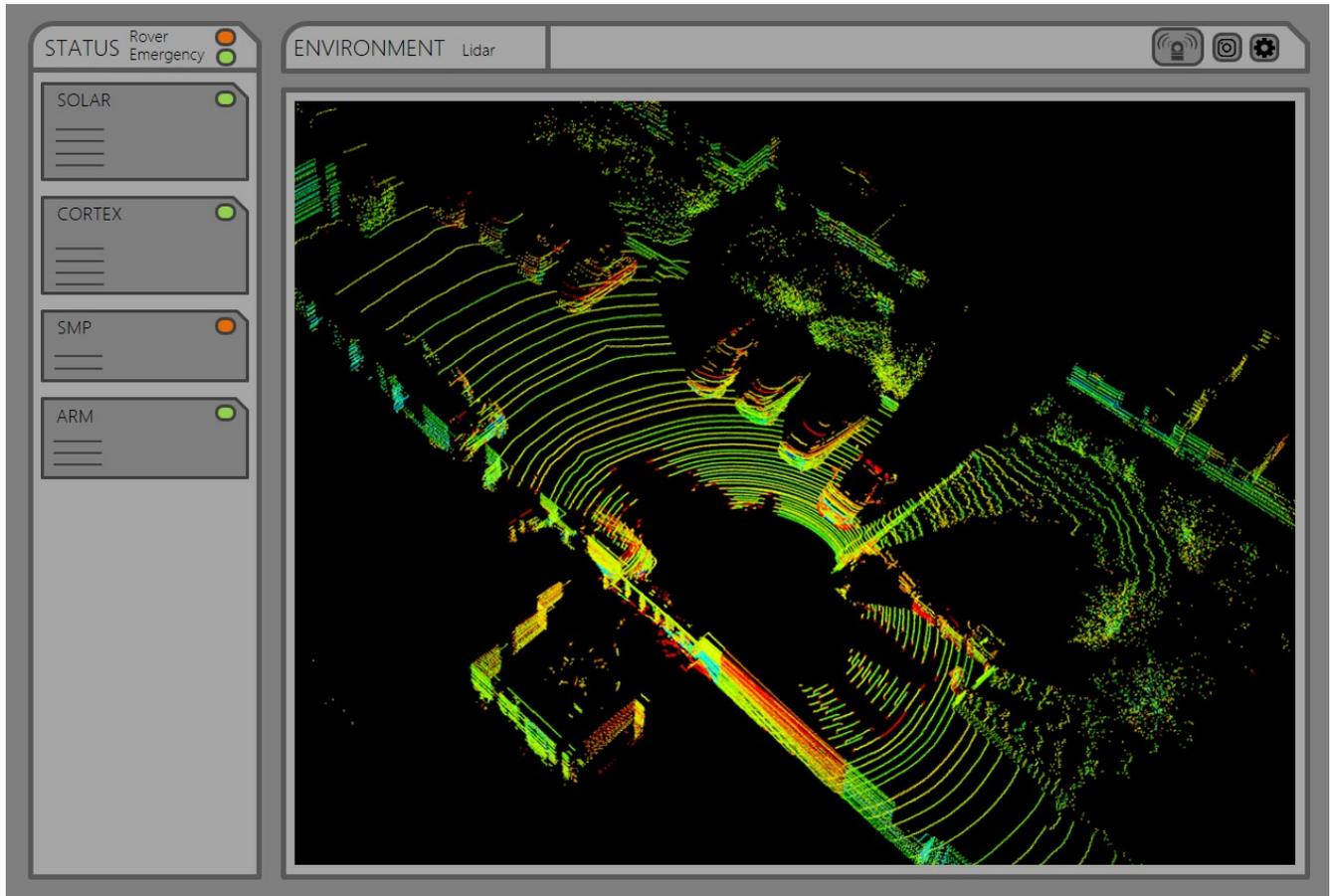
PB1 : Arrêt d'urgence

Les fonctionnalités présentées dans cette partie sont à rattacher à la présentation du projet HMI-ROV dans la suite du document de spécifications (Jx ou Joystick x, PBx ou Push Button x, SW x ou Switch x et STx ou Stick x). Voici un descriptif de chaque fonctionnalité :

- SW1 : Sélection de mode ROVER ou ARM pour l'utilisation du joystick J1 et du stick ST1

- PB1 : Une simple action sur ce bouton poussoir met le ROVER à l'arrêt, le bras robot au repos, quelque soit les positions des différentes interfaces de commandes du HAND-ROV. L'opérateur devra alors suivre une procédure pour remettre toutes les commandes en sécurité (ST1 repos, ST2 repos, J1/J2/J3 relâchés, SW1 mode ROVER et PB2/PB3/PB4 relâchés). Une seconde action sur PB1 relance toute les fonctions en mode normal de fonctionnement. L'opérateur peut alors reprendre la conduite du ROVER.
- ST1 (Rover) : Selon le mode sélectionné par l'opérateur via SW1 (mode ROVER), ST2 au repos sinon ST1 inactif, permet de fixer une première limite maximum de consigne à 1Km/h maximum. Si J1 est enfoncé en position de butée, cette vitesse est progressivement atteinte. Cela permet de limiter les vitesses de déplacement ordonnées par l'opérateur.
- ST1 (Bras robot) : Selon le mode sélectionné par l'opérateur via SW1 (mode ARM), ST2 au repos et inactif dans tous les cas dans ce mode, permet de fixer la consigne de l'asservissement de position du bras robot. Plus ST1 est enfoncé, plus le système de mesure sera proche du sol. La limite est fixée à 5cm et ne peut être franchie. Ce point est impératif. En position de repos de ST1, la consigne sera fixée à 20cm (à valider).
- ST2 : Selon le mode sélectionné par l'opérateur via SW1 (mode ROVER), ST1 obligatoirement en vitesse maximum 1Km/h (sinon, action sur ST2 inactive), permet de fixer une seconde limite maximum de consigne de vitesse à 3Km/h maximum (limite fixée par le cahier des charges fonctionnel, point contractuel). Si J1 est enfoncé en position de butée, cette vitesse est progressivement atteinte. Cela permet de limiter les vitesses de déplacement ordonnées par l'opérateur.

4.1. Interface opérateur et application logicielle - HMI-ROV

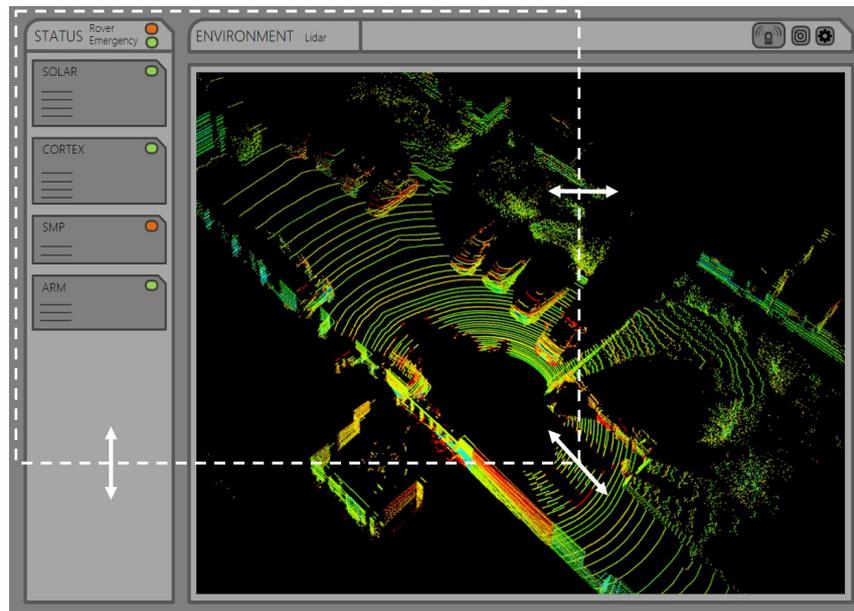


Interface graphique

L'interface graphique HMI-ROV proposée à l'opérateur en charge du contrôle du ROVER proposera 3 fenêtres principales. Attention, l'image ci-dessus n'a pour objectif que d'illustrer un rendu global et est donc non contractuelle. Il en sera de même pour la totalité des images présentées dans la suite des spécifications techniques de besoin. Chaque fenêtre sera présentée plus en détail dans la suite du document :

- fenêtre d'état du ROVER (STATUS) : Fenêtre présentant l'état non exhaustif global du ROVER à l'opérateur
- Fenêtre d'environnement (ENVIRONMENT) : Fenêtre contextuelle permettant la commutation de la fenêtre principale et offrant un retour visuel des actions possibles sur la fenêtre principale ainsi que l'état de ces actions
- Fenêtre principale : Fenêtre offrant les retours utiles métier d'informations à l'opérateur (cartographie LIDAR, vision à 360°, diagnostic complet du ROVER, autres environnements futurs, etc) et permettant un retour visuel potentiel des actions menées sur informations visualisées (sélection d'une zone, etc)

Fenêtre d'état du ROVER - STATUS

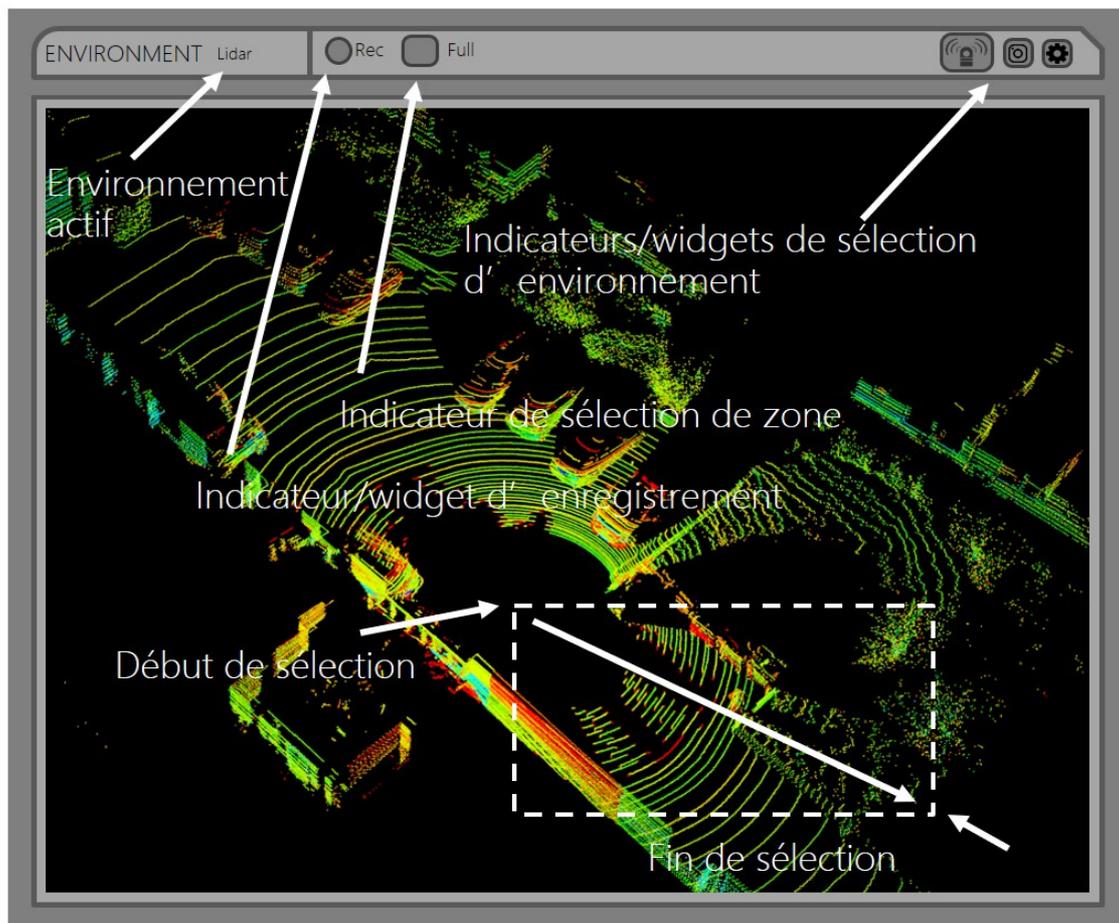


La fenêtre d'état du ROVER présente l'état global non exhaustif du ROVER à l'opérateur. La taille sur l'écran de cette fenêtre doit être garantie et rester fixe quelque soit l'écran utilisé pour interfacier l'HMI-ROV. Cette fenêtre n'a pas pour objectif de retourner un diagnostic temps réel complet du ROVER mais seulement les informations de supervision principales. Pour cette première itération du projet, seul un indicateur visuel devra à minima être proposé par cette interface. Vert, au défaut détecté, sous système entièrement opérationnel. Orange, erreurs ou warning détectés, mais sous système toujours opérationnel. Rouge, sous système non opérationnel ou arrêt d'urgence demandé

Par la suite, la fenêtre dédiée au retour d'état du sous système associé retournera les principales grandeurs utiles à l'opérateur permettant une vision haute du bon fonctionnement global du ROVER. Prenons un exemple non contractuel ci-dessous de retour retourné par le sous système SOLAR-ROV lorsque la charge de la batterie solaire sera implémenté d'ici un an. Les indicateurs en tête de fenêtre présente respectivement l'état global du Rover (vert si toutes les fonctions du Rover sont opérationnelles, orange ou rouge si au moins une fonction présente un warning ou une erreur) et l'éventuel arrêt d'urgence demandé par l'opérateur via le HAND-ROV (Emergency).



Fenêtre d'environnement - ENVIRONNEMENT



La fenêtre d'environnement a pour objectif d'offrir un retour visuel de l'environnement de supervision demandé par l'opérateur. Le bandeau de contrôle (widgets) et de supervision (indicateurs) de l'environnement actif sur la fenêtre principale sera contextuel en fonction du retour demandé. Il devra toujours offrir la même hauteur/épaisseur (longueur à adapter à l'écran utilisé), seule la fenêtre principale s'adaptera en hauteur et en largeur à technologie de l'écran. Le bandeau d'environnement sera découpé en deux zones, gauche et droite.

- A droite (dans le bandeau) : indicateurs/widgets visuels simples et ergonomiques présentant les environnements disponibles et celui actif (icône plus gros par exemple, propositions à faire par le prestataire)
- A gauche (dans le bandeau) : indicateurs/widgets contextuels relatifs à l'environnement sélectionné. Il seront présentés par la suite en fonction de l'environnement choisi

Dans cette première phase du projet, 3 environnements principaux seront à déployer. Néanmoins, prévoir dans la modélisation de l'appliquatif les évolutions des futures fonctionnalités et donc des potentiels futurs environnements à déployer (cartographie d'analyse du sol, bilan énergétique consommation/production du système d'énergie, etc). Les 3 environnements à développer sont présentés dans la suite du document. Les contraintes et limitations seront présentées en fin de document.

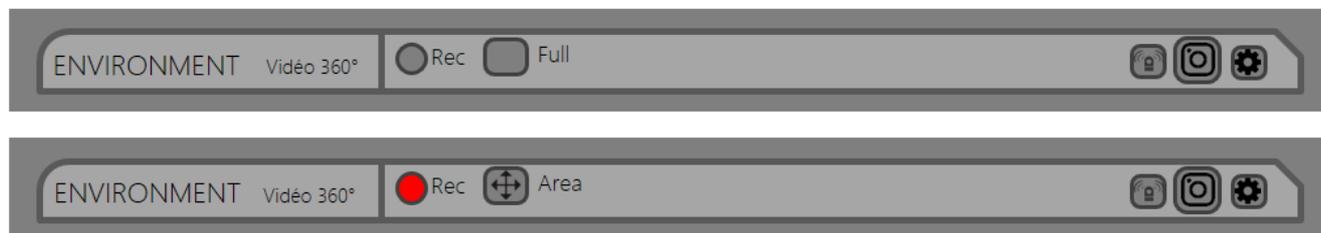
Fenêtre d'environnement LIDAR



Le premier environnement proposé offrira la possibilité d'afficher en temps réel (fréquence d'affichage de 1s) les cartographies LIDAR 2D retournée par le ROVER. Deux opérations seront possibles sur les retours d'images. Un mode d'enregistrement et un mode de saisie d'une zone (zoom numérique). Une fois sélectionné, le mode d'enregistrement réalisera l'enregistrement sur la machine host du retour vidéo affiché en fenêtre principale, avec ou sans zoom sur une zone. Chaque fichier vidéo sera horodaté et enregistré au format MPEG-4. La sélection du répertoire de destination sera présenté par la suite. Les deux images présentées ci-dessus offrent un exemple d'indicateurs visuels de retour pour l'opérateur (à proposer par le prestataire). En effet, celui-ci devra toujours être apte à savoir si une capture vidéo est en cours ou si l'image affichée correspond à la sélection d'une zone (zoom). Le retour d'image LIDAR occupera la totalité de la fenêtre principale. Les zones mortes seront unies et utiliseront le même code couleur que le fond de l'application. L'affichage en fenêtre principale sera présenté dans la suite du document.

Il faudra également anticiper les évolutions futures du projet et donc l'apparition de nouvelles actions et indicateurs sur la zone contextuelle (à gauche) du bandeau. Les évolutions principales futures porteront sur l'application d'algorithmes de traitement d'image (reconnaissance de formes, classification, apprentissage, etc).

Fenêtre d'environnement vidéo 360°



Le second environnement proposé offrira la possibilité d'afficher en temps réel (fréquence d'affichage de 1s) les retours vidéo à 360° retournée par le ROVER à travers les caméra fisheyes. Deux opérations seront possibles sur les retours d'images. Un mode d'enregistrement et un mode de saisie d'une zone (zoom numérique). Une fois sélectionné, le mode d'enregistrement réalisera l'enregistrement sur la machine host du retour vidéo affiché sur la fenêtre principale, avec ou sans zoom sur une zone. Chaque fichier vidéo sera horodaté et enregistré au format MPEG-4. La sélection du

répertoire de destination sera présenté par la suite. Les deux images présentées ci-dessus offrent un exemple d'indicateurs visuels de retour pour l'opérateur (à proposer par le prestataire). En effet, celui-ci devra toujours être apte à savoir si une capture vidéo est en cours ou si l'image affichée correspond à la sélection d'une zone. Le retour d'image vidéo à 360° n'affichera qu'une partie de la totalité de l'image de l'environnement physique du ROVER. Aucune zone morte ne sera affichée. L'affichage en fenêtre principale sera présenté dans la suite du document.

Fenêtre d'environnement de diagnostic et de configuration

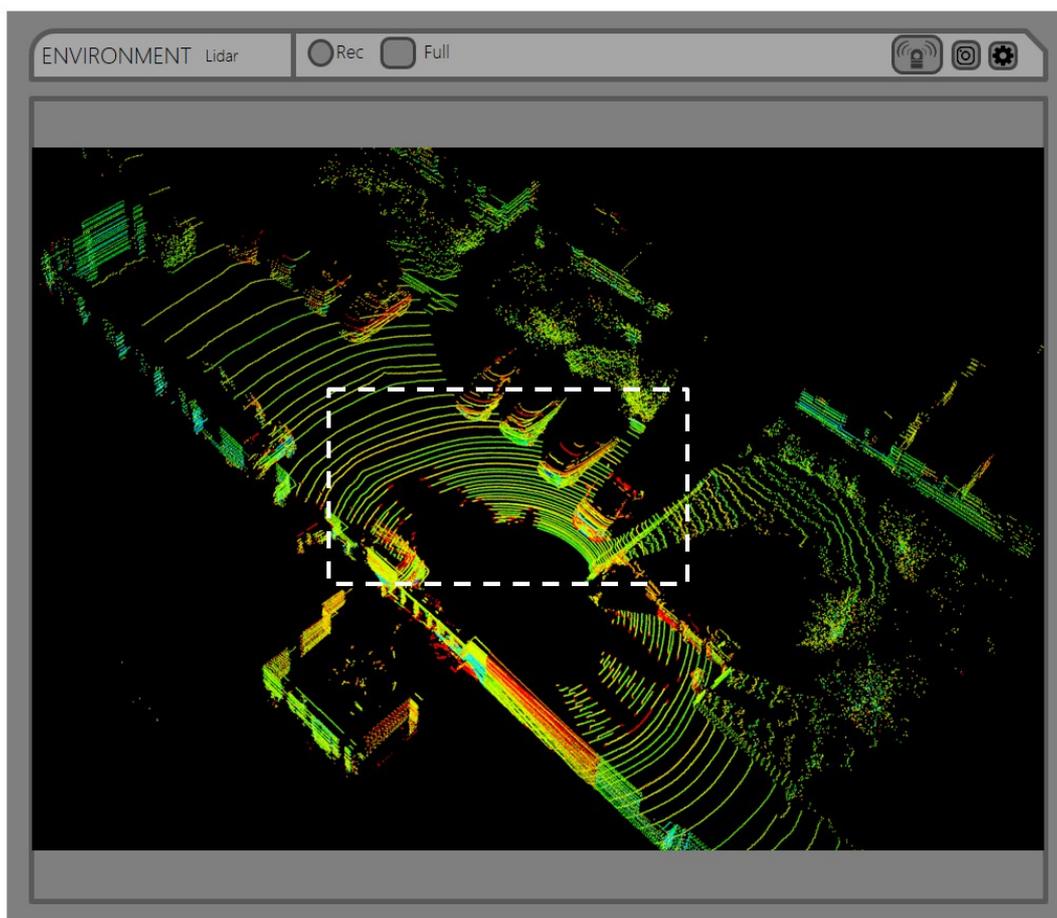


Le troisième environnement proposé offrira un retour opérateur exhaustif de l'ensemble des fonctionnalités de diagnostic, d'acquiescement et de configuration du projet ROV dans son ensemble. Cela inclut notamment les sous projets HMI-ROV et HAND-ROV. L'affichage en fenêtre principale sera présenté par la suite. Le bandeau contextuel d'environnement n'affichera que l'icône d'environnement de diagnostic, configuration et paramétrage.

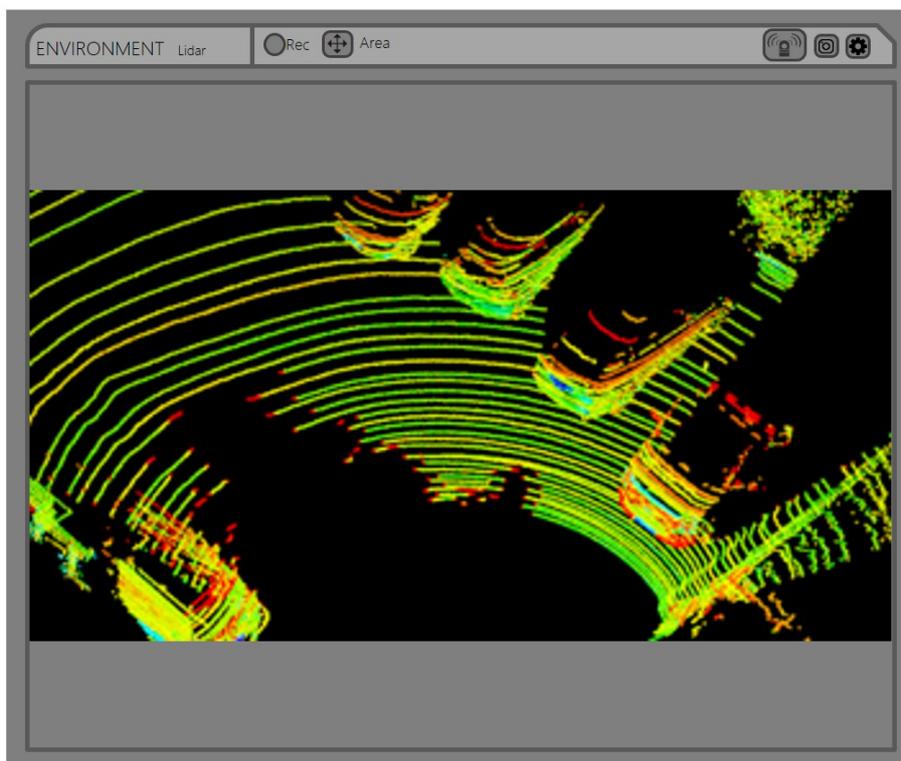
Fenêtre principale

Environnement LIDAR

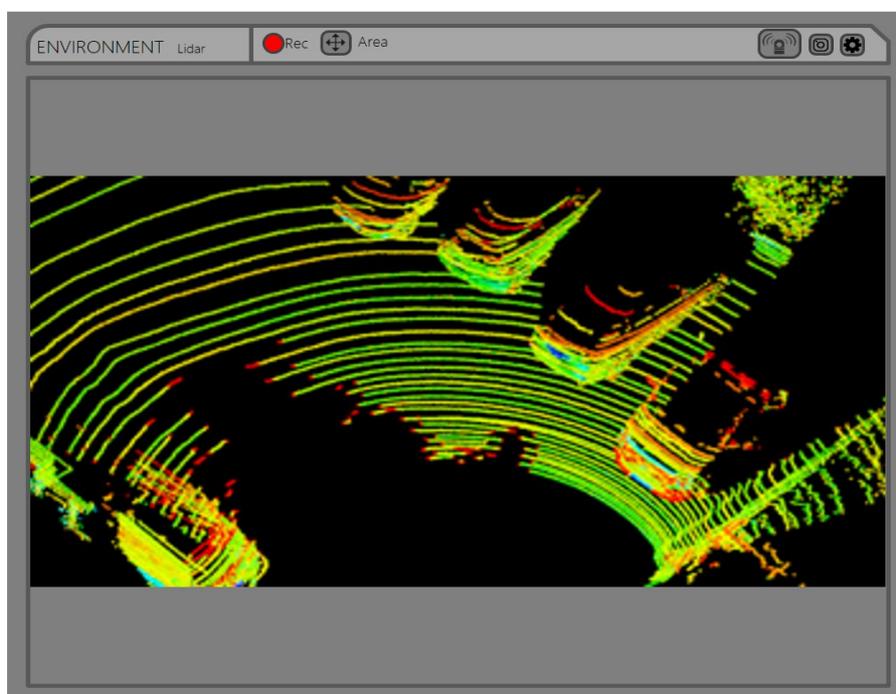
En cas de sélection par l'opérateur de l'environnement LIDAR, Le retour d'image occupera la totalité de la fenêtre principale. Les zones mortes seront unies et utiliseront le même code couleur que le fond de l'application. En cas de sélection d'une zone (pour zoom numérique) par l'opérateur, la zone sélectionnée devra explicitement apparaître. L'affichage de la zone après zoom occupera la totalité de la fenêtre principale. Les zones mortes seront unies et utiliseront le même code couleur que le fond de l'application. En cas de capture vidéo par l'opérateur, nous rappelons que l'icône d'enregistrement devra explicitement indiquer une capture en cours. Les 3 images suivantes illustrent la séquence présentée ci-dessus.



Sélection d'une zone



Affichage temps réel d'une zone sélectionnée

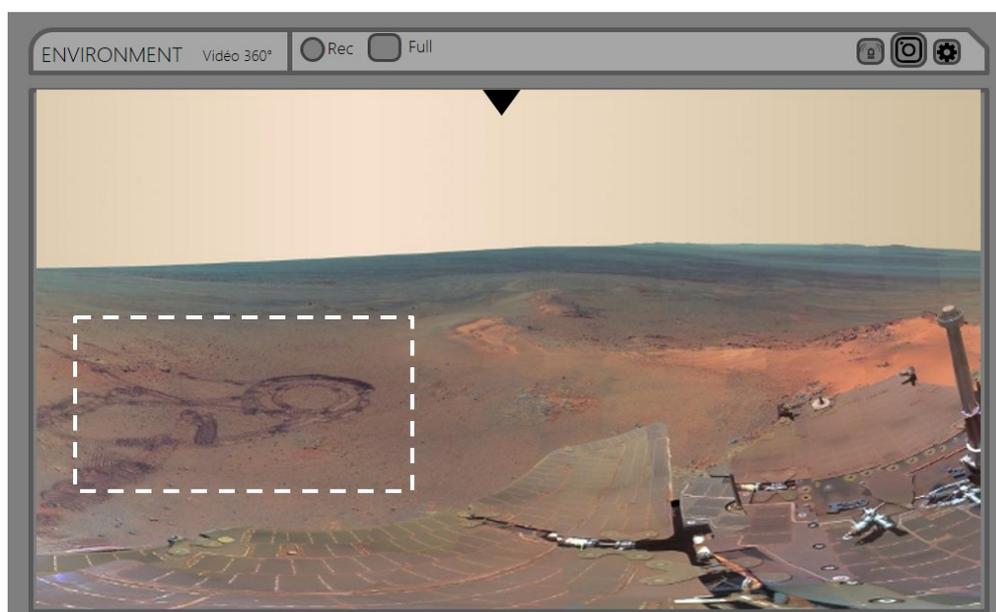


Enregistrement de la vidéo affichée en fenêtre principale

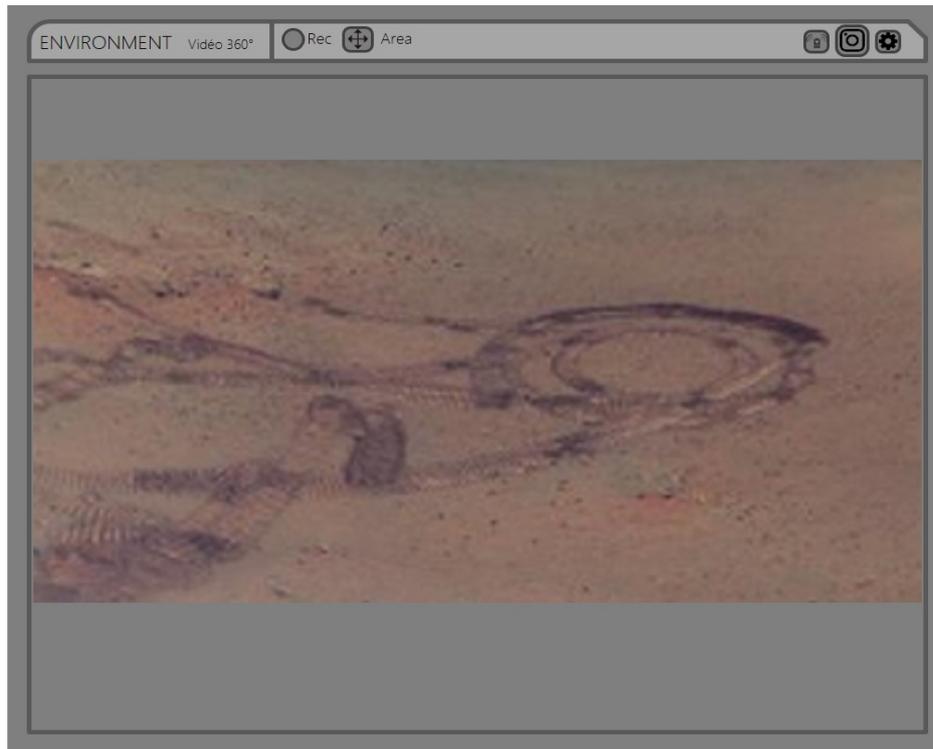
Environnement vidéo 360°



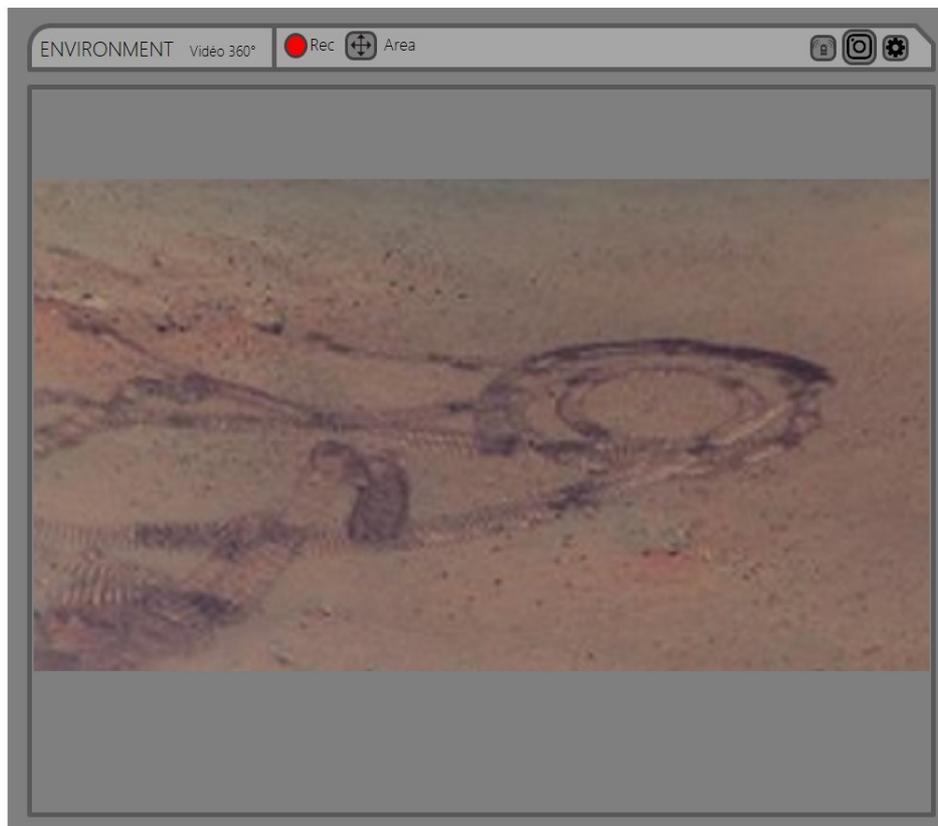
En cas de sélection par l'opérateur de l'environnement vidéo 360°, Le retour d'image vidéo à 360° n'affichera qu'une partie de la totalité de l'image de l'environnement physique du ROVER. Aucune zone morte ne sera affichée. Le glissement de la vidéo se fera par l'interface HAND-ROV ou part souris (maintient du clic gauche et glissement de la vidéo). Un indicateur visuel (triangle noir) pointera toujours l'avant du Rover. En cas de sélection d'une zone (pour zoom numérique) par l'opérateur, la zone sélectionnée devra explicitement apparaître. L'affichage de la zone après zoom occupera la totalité de la fenêtre principale. Les zones mortes seront unies et utiliseront le même code couleur que le fond de l'application. En cas de capture vidéo par l'opérateur, nous rappelons que l'icône d'enregistrement devra explicitement indiquer une capture en cours. Les 3 images suivantes illustrent la séquence présentée ci-dessus.



Sélection d'une zone

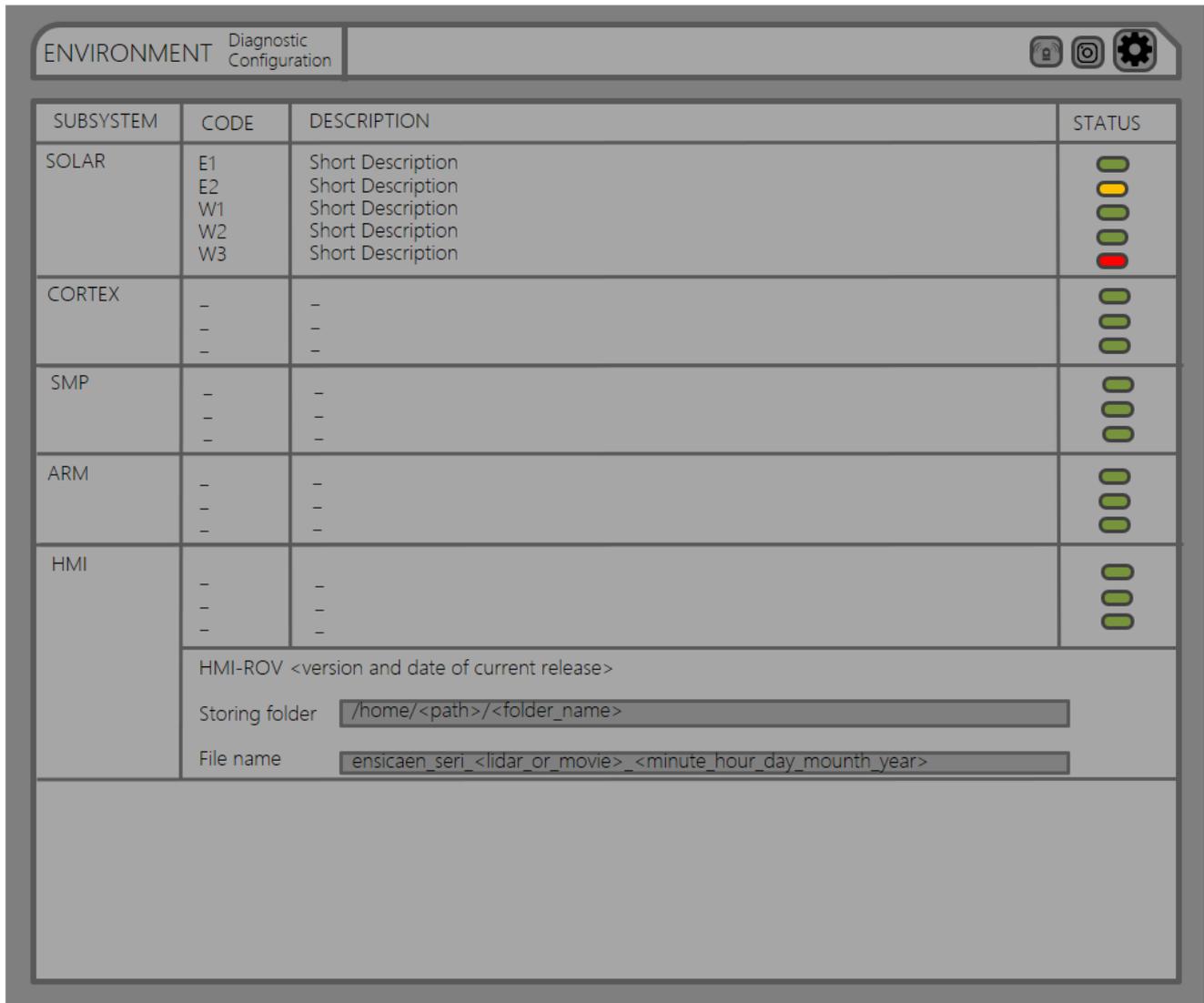


Affichage temps réel d'une zone sélectionnée



Enregistrement de la vidéo affichée en fenêtre principale

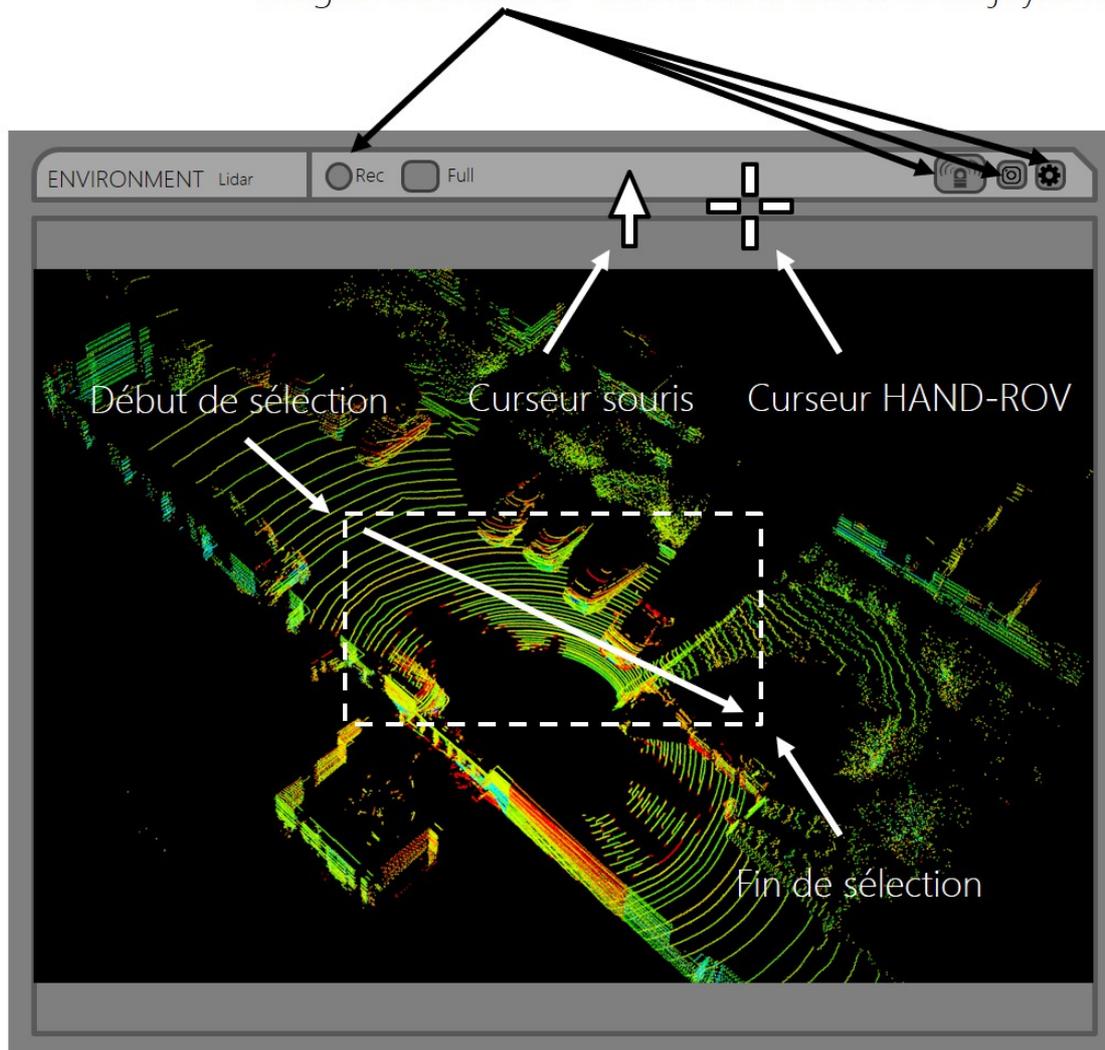
Environnement de diagnostic et de configuration



En cas de sélection par l'opérateur de l'environnement de diagnostic et de configuration, le ROVER devra se stopper et les fonctions métiers SMP-ROV, ARM-ROV et CORTEX-ROV (seulement les parties LIDAR et vidéo) devront se mettre au repos. Dans cette première phase du projet, concernant les sous systèmes du ROVER, l'interface proposera un squelette vide. Les fonctions de diagnostic n'étant pas une priorité. Néanmoins, la section de paramétrage de l'application HMI-ROV se fera dans cette fenêtre. La sélection du répertoire de destination des différentes capture LIDAR et vidéo 360° (Storing folder), ainsi que le nom des vidéos, contextuel en fonction de l'environnement sélectionné (non prioritaire), se feront dans cette fenêtre.

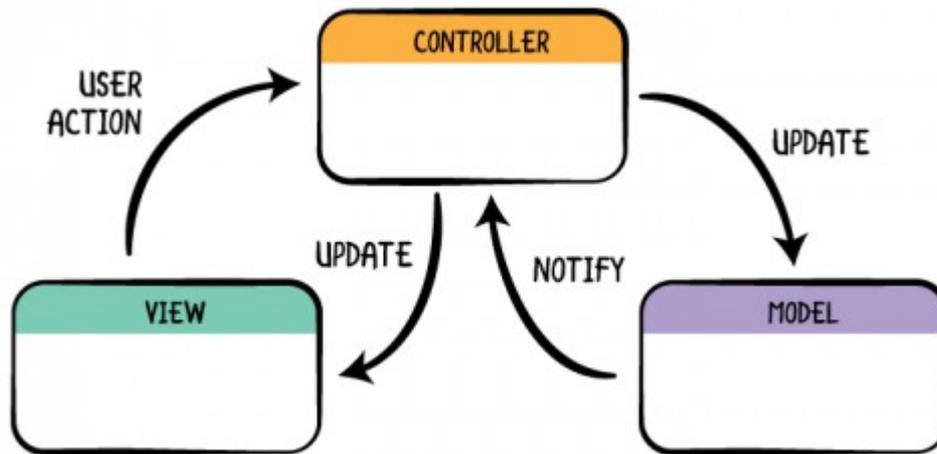
Souris et joystick HAND-ROV

Widgets ou zones d' action de la souris et du joystick HAND-ROV



L'interface graphique pourra entièrement être manipulée par le système HAND-ROV pour le contrôle et la supervision du ROVER via le joystick J2. Ce point ne s'applique pas au paramétrage de l'HMI-ROV (répertoire de destination et nom des fichiers de sortie). Les curseur souris et HAND-ROV seront graphiquement dissociés (flèche classique pour la souris et croix noire et blanche pour le HAND-ROV). Dans les deux cas, souris et HAND-ROV, sans déplacement des curseurs et arrêt de déplacement, après 3 secondes les curseurs disparaîtront de l'interface.

Contraintes et limitations



Architecture MODEL VIEW CONTROLLER/PRESENTER

Fenêtre d'état

La fenêtre d'état ne proposera jamais plus de 7 sous fenêtres correspondant à 7 sous systèmes du ROVER. Chaque sous fenêtre ne proposera jamais plus de 7 informations à afficher en temps réel (actualisation à chaque seconde).

Fenêtre d'environnement et fenêtre principale

Le bandeau contextuel ne proposera jamais plus de 7 environnements à sélectionner (zone à droite du bandeau). De même, par environnement, le bandeau ne proposera jamais plus de 7 indicateurs ou widgets contextuels (zone à gauche du bandeau). Les cartographies LIDAR proposeront un fond noir et des contours de formes blancs.

Contraintes temporelles

Les données montantes du CORTEX-ROV vers l'HMI-ROV (cartographies LIDAR, images à 360°, diagnostics, informations pour la fenêtre d'état, évolutions futures, etc) seront actualisées toutes les secondes. Contrainte temporelle impérative à respecter. A terme, les données seront compressées (images) et chiffrées. Les données descendantes de l'HMI-ROV vers le CORTEX-ROV (consignes de vitesse, limitations de vitesse, consignes de déplacement du bras, limitations de distance du bras par rapport au sol, commutations d'environnement, demandes de diagnostic, etc) seront envoyées toutes les 50ms. Contrainte temporelle impérative à respecter. A terme, les données descendantes seront chiffrées.

Graphismes et police

Graphismes et couleurs

Deux nuances de gris seront utilisées par l'interface afin de faciliter l'utilisation de l'interface en milieu sombre, gris foncé et gris clair. Les cadres proposeront un gris très foncé proche du noir. Le texte sera en noir. Les graphismes et l'ergonomie globale devront rester très simple.



Gris foncé



Gris clair



Gris très foncé

Police

La police utilisée pour l'ensemble de l'HMI-ROV sera Calibri Ligth. Les titres de fenêtre et de section seront écrits en majuscule. Les autres textes en minuscule. Les tailles seront à spécifier une fois les premiers rendus proposés et feront l'objet d'une extension des spécifications techniques.

5. ECHEANCIER DES LIVRABLES

5. ECHEANCIER DES LIVRABLES

Cette dernière partie des spécifications fixe un échéancier spéculatif des livrables sur la totalité de la durée du projet. Afin de garantir la faisabilité du projet ROV, ces deadlines seront à respecter dans la mesure du possible et feront l'objet de discussions et de validation entre direction des équipes de prestation et client (chefs de projet maîtrise d'ouvrage). A chaque livrable sera associé un niveau de priorité :

- P0 : Priorité impérative
- P1 : Priorité haute
- P2 : Priorité moyenne
- P3 : Priorité basse

Les rendus seront attendus à chaque date proposée pour 23h. Dans l'attente du déploiement du dépôt collaboratif de gestion de version Gitlab le week-end du 15 septembre, les premiers livrables seront livrés sous forme d'une archive. L'arborescence de l'archive fera l'objet d'une réflexion commune et sera très proche voire identique à l'arborescence déployée sur le futur dépôt de gestion de version collaboratif.

Lundi 10 septembre

- Cadrage et amorçage du projet

Mardi 11 septembre

- Cadrage et amorçage du projet

Mercredi 12 septembre

SMP-ROV

- P0 : Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement - version 1
- P0 : Projets de test unitaires par interface (PWM, UART et interpréteur de commande, I2C)
- P0 : Projet de test unitaire intégrant la solution RTOS FreeRTOS seul avec wrapper CMSIS-OS
- P0 : Schéma électrique de la carte fille SMP-ROV - version 1
- P0 : Schéma de routage de la carte fille SMP-ROV - version 1
- P0 : Fichiers de de perçage de la carte fille SMP-ROV - version 1

ARM-ROV

- P0 : Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement - version 1
- P0 : Projets de test unitaires par interface (PWM, UART et interpréteur de commande, I2C, SPI, ADC et GPIO avec interruption)
- P0 : Projet de test unitaire intégrant la solution RTOS FreeRTOS seul avec wrapper CMSIS-OS

SOLAR-ROV

- P0 : Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement - version 1
- P0 : Projets de test unitaires par interface (ADC, UART et interpréteur de commande, I2C)
- P0 : Projet de test unitaire intégrant la solution RTOS FreeRTOS seul avec wrapper CMSIS-OS
- P0 : Schéma électrique de la carte fille SOLAR-ROV - version 1
- P0 : Schéma de routage de la carte fille SOLAR-ROV - version 1
- P0 : Fichiers de de perçage de la carte fille SOLAR-ROV - version 1
- P0 : Validation de la solution analogique de mesure du niveau batterie

CORTEX-ROV

- P1 : Projets de test unitaires par interface (LIDAR et WIFI en access point)

HMI-ROV

- P1 : Premier rendu visuel de l'interface graphique - version 1

Gestion et communication

- P0 : Plan d'exécution complet prévisionnel du projet - version 1
- P0 : Proposition d'actions de communication (Facebook et journée portes ouvertes)

Jeudi 13 septembre

ARM-ROV

- P0 : Schéma électrique de la carte fille ARM-ROV PCB arm - version 1
- P0 : Schéma de routage de la carte fille ARM-ROV PCB arm - version 1
- P0 : Fichiers de de perçage de la carte fille ARM-ROV PCB arm - version 1
- Validation du modèle et simulation du régulateur pour l'asservissement d'assiette sous Matlab/Simulink
- Validation du modèle et simulation du régulateur pour l'asservissement de position sous Matlab/Simulink

CORTEX-ROV

- P0 : Modélisation logicielle de l'applicatif POSIX en UML si développement par langage orienté objet. Sinon, utiliser le même formalisme (unique et standard au projet) que pour les sous systèmes ARM-ROV, SMP-ROV et CORTEX-ROV. Une modélisation SysML pourra potentiellement être demandées - version 1
- P1 : Projets de test unitaires par interface (caméras fisheyes 360° et I2C)

HMI-ROV

- P1 : Validation des fonctionnalités utiles du HAND-ROV
- P0 : Modélisation et architecture logicielle - version 1

Mécanique

- P1 : Proposition de châssis mécanique, placement des solutions électroniques et sous systèmes du Rover, solution d'emport de la batterie, solution d'emport du bras robot, passage des câbles, etc

Gestion et communication

- P0 : Plan d'exécution complet du projet - version stable
- P0 : déploiement d'une première version du dépôt collaboratif de gestion de versions Gitlab. Validation de l'arborescence du dépôt
- P1 : Planification des actions de communication (Facebook et journée portes ouvertes)

Vendredi 14 septembre

SMP-ROV

- P0 : Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement - version stable
- P3 : *Projet baremetal d'intégration des solutions d'interfaces sans RTOS (PWM, UART et interpréteur de commande, I2C). Cette partie n'est pas nécessaire si le livrable suivant est produit dans une version stable*
- P1 : Projet complet hors solutions de diagnostic sans intégration au réseau de bord NET-ROV

ARM-ROV

- P0 : Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement - version stable
- P3 : *Projet baremetal d'intégration des solutions d'interfaces sans RTOS (PWM, UART et interpréteur de commande, I2C, SPI, ADC et GPIO avec interruption). Cette partie n'est pas nécessaire si le livrable suivant est produit dans une version stable*
- P1 : Projet complet hors solutions de diagnostic sans intégration du réseau de bord NET-ROV
- P1 : Schéma électrique de la carte fille ARM-ROV PCB wrist - version 1
- P1 : Schéma de routage de la carte fille ARM-ROV PCB wrist - version 1
- P1 : Fichiers de de perçage de la carte fille ARM-ROV PCB wrist - version 1

SOLAR-ROV

- P0 : Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement - version stable
- P3 : *Projet baremetal d'intégration des solutions d'interfaces sans RTOS (ADC, UART et interpréteur de commande, I2C). Cette partie n'est pas nécessaire si le livrable suivant est produit dans une version stable*
- P1 : Projet complet hors solutions de diagnostic sans intégration du réseau de bord NET-ROV

CORTEX-ROV et NET-ROV

- P0 : Modélisation logicielle de l'applicatif POSIX en UML si développement par langage orienté objet. Sinon, utiliser le même formalisme (unique et standard au projet) que pour les sous systèmes ARM-ROV, SMP-ROV et CORTEX-ROV. Une modélisation SysML pourra potentiellement être demandées - version stable
- P1 : Document de spécification du réseau de bord NET-ROV - version 1
- P1 : Document de spécification du réseau WIFI - version 1
- P0 : Schéma électrique de la carte fille CORTEX-ROV - version 1
- P0 : Schéma de routage de la carte fille SCORTEX -ROV - version 1
- P0 : Fichiers de de perçage de la carte fille CORTEX-ROV - version 1

HMI-ROV

- P0 : Modélisation et architecture logicielle - version stable

Samedi 15 septembre

SMP-ROV

- Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement - version stable
- P0 : Schéma électrique de la carte fille SMP-ROV - version stable
- P0 : Schéma de routage de la carte fille SMP-ROV - version stable
- P0 : Fichiers de de perçage de la carte fille SMP-ROV - version stable
- P0 : PCB de la carte fille SMP-ROV fonctionnel
- P1 : Documentation de référence de la carte fille SMP-ROV. Présentation de la carte fille SMP-ROV, justification du choix de placement des composants, nomenclature des composants, problématiques CEM, etc

ARM-ROV

- Modélisation logicielle de l'applicatif temps réel. Formalisme graphique non standardisé. Néanmoins, il devra être commun en l'ensemble des applications C temps réel développées. Harmonisation entre équipes de développement - version stable
- P0 : Schéma électrique de la carte fille ARM-ROV PCB arm - version stable
- P0 : Schéma de routage de la carte fille ARM-ROV PCB arm - version stable
- P0 : Fichiers de de perçage de la carte fille ARM-ROV PCB arm - version stable

SOLAR-ROV

- P0 : Schéma électrique de la carte fille SOLAR-ROV - version stable
- P0 : Schéma de routage de la carte fille SOLAR-ROV - version stable
- P0 : Fichiers de de perçage de la carte fille SOLAR-ROV - version stable
- P0 : PCB de la carte fille fonctionnel
- P1 : Documentation de référence de la carte fille SOLAR-ROV. Présentation du PCB, justification du choix de placement des composants, problématiques CEM, nomenclature des composants, etc

CORTEX-ROV

- P1 : Document de spécification du réseau de bord NET-ROV - version stable
- P1 : Document de spécification du réseau WIFI - version stable

HMI-ROV

- P0 : Validation de toutes les fonctionnalités utiles du HAND-ROV et validation de l'instrumentation de l'HMI-ROV
- P0 : Premier démonstrateur partiel avec retour vidéo intégré

Mécanique

- P0 : Validation de la structure du châssis mécanique, placement des solutions électroniques et sous systèmes du Rover, solution d'emport de la batterie, passage des câbles

Gestion et communication

- Organisation de la journée portes ouvertes – version 1

Lundi 17 septembre

SMP-ROV

- P1 : *En cas de retard au déploiement du réseau NET-ROV et du sous système CORTEX-ROV, développer une solution de test émulant des communication sur le réseau afin de valider les fonctions métier. A valider en fonction du niveau d'avancement du projet*
- P0 : Projet complet hors solutions de diagnostic avec intégration au système complet via le réseau de bord NET-ROV

ARM-ROV

- P1 : *En cas de retard au déploiement du réseau NET-ROV et du sous système CORTEX-ROV, développer une solution de test émulant des communication sur le réseau afin de valider les fonctions métier. A valider en fonction du niveau d'avancement du projet*
- P0 : Projet complet hors solutions de diagnostic avec intégration au système complet via le réseau de bord NET-ROV

SOLAR-ROV

- P1 : *En cas de retard au déploiement du réseau NET-ROV et du sous système CORTEX-ROV, développer une solution de test émulant des communication sur le réseau afin de valider les fonctions métier. A valider en fonction du niveau d'avancement du projet*
- P0 : Projet complet hors solutions de diagnostic avec intégration au système complet via le réseau de bord NET-ROV

CORTEX-ROV

- P1 : Projet complet hors solutions de diagnostic sans réseau de bord NET-ROV

HMI-ROV

- P1 : Projet complet hors solutions de diagnostic avec émulation du système CORTEX-ROV

Gestion et communication

- Organisation de la journée portes ouvertes

Mardi 18 septembre

SMP-ROV

- P0 : Prototype fonctionnel complet

ARM-ROV

- P0 : PCB wrist de la carte fille fonctionnel
- P0 : PCB arm de la carte fille fonctionnel
- Régulateur fonctionnel pour l'asservissement d'assiette sous Matlab/Simulink et intégré sur STM32F303RE
- Régulateur fonctionnel pour l'asservissement de position sous Matlab/Simulink et intégré sur STM32F303RE

SOLAR-ROV

- P0 : Prototype fonctionnel complet

CORTEX-ROV

- P1 : Projet complet hors solutions de diagnostic avec réseau de bord NET-ROV
- P0 : PCB de la carte fille fonctionnel

HMI-ROV

- P0 : Prototype fonctionnel complet

Mécanique

- P0 : Prototype fonctionnel complet

Mercredi 19 septembre

SMP-ROV

- PO : Documentation de référence du SMP-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures – version 1
- PO : Document d'errata du SMP-ROV – version 1

ARM-ROV

- PO : Prototype fonctionnel complet
- PO : Documentation de référence du ARM-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures de développement – version 1
- PO : Document d'errata du ARM-ROV – version 1
- PO : Documentation de référence des cartes filles ARM-ROV PCB arm et PCB wrist. Présentation des cartes filles, justification du choix de placement des composants, contraintes CEM, nomenclature des composants, etc – version 1

SOLAR-ROV

- PO : Documentation de référence du SOLAR-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures – version 1
- PO : Document d'errata du SOLAR-ROV – version 1

CORTEX-ROV

- PO : Prototype fonctionnel complet
- PO : Documentation de référence du CORTEX-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures – version 1
- PO : Document d'errata du CORTEX-ROV – version 1

HMI-ROV

- PO : Application logicielle HMI-ROV fonctionnelle – version 1
- PO : Documentation de référence du HMI-ROV. Présentation de la modélisation et architecture logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures – version 1
- PO : Document d'errata du HMI-ROV – version 1

Mécanique

- PO : Document de référence présentant l'ensemble des solutions mécaniques déployées – version 1

Gestion et communication

- Organisation des présentations techniques

Jeudi 20 septembre

SMP-ROV

- P3 : *Formalisation de la modélisation au standard SDL-RT sous environnement Pragmadev. Cette partie sera demandée en fonction du niveau d'avancement du projet*
- P2 : Documentation doxygen globale au format .chm (voire PDF apprécié) de l'applicatif et bibliothèques SMP-ROV

ARM-ROV

- P3 : *Formalisation de la modélisation au standard SDL-RT sous environnement Pragmadev. Cette partie sera demandée en fonction du niveau d'avancement du projet*
- P2 : Documentation doxygen globale au format .chm (voire PDF apprécié) de l'applicatif et bibliothèques ARM-ROV

SOLAR-ROV

- P3 : *Formalisation de la modélisation au standard SDL-RT sous environnement Pragmadev. Cette partie sera demandée en fonction du niveau d'avancement du projet*
- P2 : Documentation doxygen globale au format .chm (voire PDF apprécié) de l'applicatif et bibliothèques SOLAR-ROV
-

CORTEX-ROV

- P2 : Documentation doxygen globale au format .chm (voire PDF apprécié) de l'applicatif et bibliothèques CORTEX-ROV
- P3 : *Déploiement d'une solution système durcie et maintenue sur le long terme en fonction du niveau d'avancement du projet*
- P3 : *Document de maintenance de la distribution. Par exemple, documentation concernant le listing des dépendances de l'application (version des packages) et justification de la configuration du projet Yocto fourni, versions kernel choisie, etc. Ce document pourra être rattaché au document d'étude pour le déploiement. A réaliser en fonction du niveau d'avancement du projet*

Vendredi 21 septembre

- **PO : Démonstrateur complet intégré du projet ROV**
- **PO : Livraison du dépôt collaboratif Gitlab avec l'ensemble des réalisations**

SMP-ROV

- PO : Prototype fonctionnel du projet SMP-ROV intégré au ROVER
- PO : Documentation de référence du SMP-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures
- PO : Document d'errata du SMP-ROV

ARM-ROV

- PO : Prototype fonctionnel du projet ARM-ROV intégré au ROVER
- PO : Documentation de référence du ARM-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures de développement
- PO : Document d'errata du ARM-ROV
- PO : Documentation de référence des cartes filles ARM-ROV PCB arm et PCB wrist. Présentation des cartes filles, justification du choix de placement des composants, contraintes CEM, nomenclature des composants, etc
- PO : Régulateur fonctionnel pour l'asservissement d'assiette sous Matlab/Simulink et intégré sur STM32F303RE
- PO : Régulateur fonctionnel pour l'asservissement de position sous Matlab/Simulink et intégré sur STM32F303RE
- PO : Rapport d'étude présentant et justifiant l'ensemble des solutions de synthèse des régulateurs, les modélisations procédés, cahiers des charges et spécifications techniques, exigences, contraintes, etc

SOLAR-ROV

- PO : Documentation de référence du SOLAR-ROV. Présentation de la modélisation logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures
- PO : Document d'errata du SOLAR-ROV
- PO : Documentation doxygen globale au format .chm (voire PDF apprécié) de l'applicatif et bibliothèques SOLAR-ROV
- P2 : Rapport d'étude complet concernant la solution de charge de batterie. Ce document précisera la future technologie de batterie utilisée, proposition de solutions du marché, présentation de la structure du chargeur, étude et synthèse des régulateurs d'asservissement, modélisation du procédé, nomenclature des composants, etc
- PO : Prototype fonctionnel du projet SOLAR-ROV intégré au ROVER

CORTEX-ROV

- PO : Prototype fonctionnel du projet CORTEX-ROV avec réseau NET-ROV opérationnel intégré au ROVER
- PO : Documentation de référence du CORTEX-ROV. Présentation de la modélisation

logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures

- P0 : Document d'errata du CORTEX-ROV
- P1 : Rapport d'étude pour le déploiement d'une distribution durcie, voire avec kernel temps réel, et maintenue sur le long terme. L'étude sur le déploiement sur une solution kernel temps réel devra apparaître dans ce rapport d'étude.
- P0 : Documentation de référence de la carte fille CORTEX-ROV. Présentation de la carte fille, justification du choix de placement des composants, nomenclature des composants, etc

HMI-ROV

- P0 : Application logicielle HMI-ROV fonctionnelle
- P0 : Documentation de référence du HMI-ROV. Présentation de la modélisation et architecture logicielle, des interfaces, problèmes rencontrés, justification des solutions déployées, versions des outils, etc. Ce document devra entièrement être pensé dans une optique de réutilisation par les équipes futures
- P0 : Document d'errata du HMI-ROV

Mécanique

- P0 : Solutions mécaniques opérationnelles
- P0 : Document de référence présentant l'ensemble des solutions mécaniques déployées

Gestion et communication

- Organisation des présentations commerciales