CAMPOSV

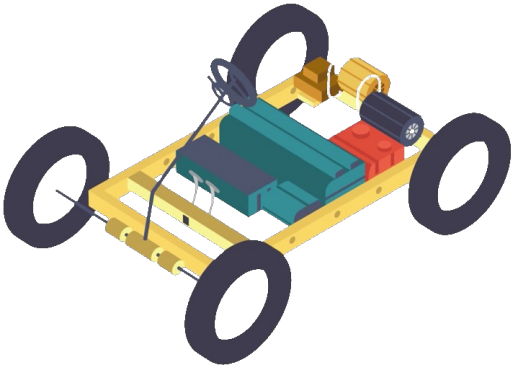
Published : 2018-03-23

LE HACKATHON CAMPOSV 14-18 MARS 2018

1. INTRODUCTION

Cette documentation a été réalisé pour le **hackathon CampOSV** du 13 au 15 mars 2018 lors de l'événement In Out 2018 sur les mobilités numériques, événement porté par Rennes Métropoles.

Le projet CampOSV, lancé en 2015, est piloté par le LabFab de l'Université de Rennes 1 en partenariat avec l'IETR et l'IMT Atlantique. Cet hackathon est le quatrième depuis le lancement du projet !



L'objectif de cet événement à l'intérieur du plus grand événement In Out est de concevoir des véhicules électriques libre. Ce projet a débuté en 2015. L'édition de mars 2018 du hackathon CampOSV est co-organisé par le LabFab de l'Université de Rennes 1 et Rennes Métropole.

Cette documentation a été écrite par une équipe de 6 personnes de Floss Manuals Francophone :

- Margaux Girard,
- Cyprien Roudet,
- Laurent Malys,
- Barthélemy Péron,
- Pierre Commenge,
- Elisa de Castro Guerra

2. JOUR 1

Accueil à 9h sur le site de PSA La Janais, Rennes. Après une conférence d'ouverture par les organisateurs du hackathon ainsi que la parole de Rennes Métropoles, nous avons suivi quelques conférences :

- sur les modèles économiques de l'innovation ouverte par **Thomas Le Texier**.
- sur la collaboration par **Alain Heureux**.
- sur l'économie du libre par **Roberto di Cosmo**.

Ensuite nous avons eu une présentations des différentes **ateliers de prototypage**.

Le hackathon est organisé autour de groupe portant sur plusieurs thématiques :

- Atelier PyOSVUE
- Atelier IoT
- Atelier HandiVéhicule
- Atelier BMS
- Atelier Robotique
- Atelier Ecolocar

Après la pause déjeuner nous nous sommes regroupés dans l'espace de travail et redécoupé par tablé.

ATELIER PYOSV

Concevoir un logiciel pour permettre la facilitation de coconception de véhicule open source.

ATELIER IOT

L'objectif est de préparer un envoie des informations remontant des informations du BMS d'une raspberry vers internet afin de pouvoir exploiter les données pour des usages spécifiques.

ATELIER HANDIVÉHICULE

L'objectif est de proposer aux personnes handicapées et à leur assistant un véhicule adapté et électrique.

ATELIER BMS

L'objectif est de prototyper un BMS libre et documenté.

ATELIER ROBOTIQUE

L'objectif est d'explorer le système embarqué en s'aidant du monde de la robotique afin de pouvoir améliorer la conception de notre véhicule.

ATELIER ECOLOCAR

L'objectif est de réutiliser ce véhicule écologique conçu par Marvin Jonhson il y a maintenant 10 ans (conçu autour de 9 panneaux solaires) afin de ré-implémenter ses excellentes idées et expériences pour le véhicule OSV.

3. JOUR 2

Premier temps de prototypage dès la matin. Les personnes sont davantages investis dans l'objectif de l'atelier.

Fin de matinée, conférences sur le Droit et la conception automobile de **Cédric Coulon**, ainsi que celle sur les Biens communs digitaux de **Marco Ciucina**. Cette dernière conférence suscite un débat très intéressant autour de la position europpéenne vis à vis de celles des autres pays.

L'après-midi, des premiers résultats commencent à apparaître et des échanges entre différents ateliers prennent corps.

4. JOUR 3

Dernier jour pour terminer les prototypages. Cette matinée passe en coup de vent, tout le monde étant afféré dans une dernière ligne droite.

Le matin conférence sur les Energies renouvelables et véhicules électriques a encore une fois suscité de nombreuses questions par les membres des projets. Merci à Bernard Multon pour autant de précision.

La journée s'est terminée plus tôt et en beauté avec la présence de Richard Stallman qui a réalisé la célèbre conférence devant un public d'étudiant privacy by design.

5. RESTITUTION

Chaque fin de journée est le moment de se retrouver tous au centre de la salle et de présenter les avancées de la journée. Ces moments convivaux permettent non seulement de clore la journée de manière satisfaisante mais également de pointer les ponts à réaliser entre chaque atelier.

ECOLOCAR

Liste des participants de l'atelier ECOLOCAR

Marvin Johnson - Association Levita : levita.fr
Maker, rêveur, ingénieur, passionné de véhicules électriques qui roulent et qui volent.
"*Pourquoi suis-je venu au Hackathon ?*" *:* Pour faire connaitre mon véhicule Ecolocar et son histoire, pour qu'il serve de démonstrateur, et pour imaginer son renouveau, ses améliorations, la suite ...

Philippe Pacotte
"*Pourquoi suis-je venu au Hackathon ?*" *:* Redessiner la carrosserie de modèle existant

Arthur Masson - arthurmasson.xyz
"*Pourquoi suis-je venu au Hackathon ?*" *:* Je suis intéressé par la mobilité de demain, l'écologie, et l'open source (le libre). Je pense qu'il y a beaucoup à faire à l'intersection de tous ces domaines, et le CampOSV est l'occasion de se pencher sur les questions inhérentes à ce paradigme.

Gregory Jamard - experiences-technologies.fr
"*Pourquoi suis-je venu au Hackathon ?*" *:* Sensible depuis toujours à l'écologie, j'ai voulu participer à ce hackathon qui allie a la fois la technologie et la mobilité intelligente.

Barthélemy Péron - graphiste, designer, artiste-chercheur - Atelier mobile de recherche et création autour du monde : geocyclab.fr
"*Pourquoi suis-je venu au Hackathon ?*" *:* Documenter ce qui se passe durant ce hackathon.
Cette doc sera sur Floss Manuals Francophone (flossmanuals.net/camposv/) et sur le wiki du projet (lafabriquedesmobilites.fr/wiki/VehiculeLibre)

Fatoumata Darame - Gabriel Couture - Alexandre Papazoglou (étudiants en master Stratégies digitales et innovation numérique)
"*Pourquoi suis-je venu au Hackathon ?*" *:* C'est un évènement qui s'inscrit dans le cadre d'un cours de notre master. Le but était d'envisager un modèle économique pour chacun des ateliers sur lesquels nous étions repartis à cette occasion.

7. ORIGINE DU PROJET

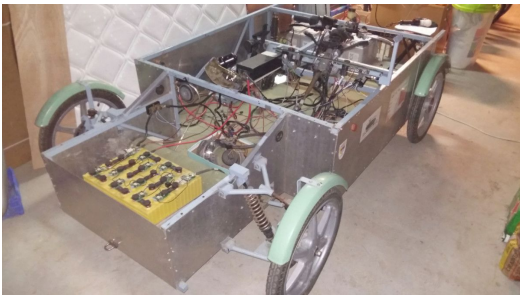
L'Ecolocar est une voiture électrique alimentée par panneaux solaires, conçue par Marvin Jonhson en 2002, selon le cahier des charges suivant :

- pouvoir embarquer deux passagers et assurer leur couchage dans l'habitacle.
- se déplacer à une vitesse limitée (45 km/h) pour profiter du paysage.
- être facilement réparable ou modifiable
- intégrer 9 panneaux solaires rigides et un système de batteries

Les neuf panneaux photovoltaïques qui alimentent l'Ecolocar ont constitué le point de départ du projet. En l'absence de contrainte d'aérodynamie, ils ont déterminé la géométrie et le design final du véhicule. Un système de pédalier mécanique devait permettre d'assurer son fonctionnement en cas de panne électrique.

Assemblée en neuf mois dans le garage de Marvin, en utilisant des pièces de voiture, moto, scooter et vélo, l'Ecolocar a ensuite passé des tests de freinage et de vitesse pour obtenir sa carte grise catégorie "quadricycle léger à moteur". Elle peut donc rouler sur la chaussée publique, à une vitesse maximale de 45 km/h.

L'Ecolocar est alors devenue le support pédagogique et médiatique de l'Ecolotour, un tour du monde de onze mois effectué par Marvin et sa compagne, et impliquant la traversée de 14 pays (dont 3 avec le véhicule).



Composants techniques :

- Batterie - 36V - 100AH
- Panneau solaire 54W (ensoleillement max) - (x9 = 486W)
- Moteur Lynch - 36 à 48V - 4kW
- Contrôleur 4QD

8. MODIFICATIONS RÉCENTES

Après une dizaine d'années d'hivernage, l'Ecolocar s'offre une seconde vie en devenant une plate-forme mobile d'expérimentation et de prototypage à disposition de personnes ou d'organismes souhaitant promouvoir des solutions de mobilité moins impactantes sur l'environnement.

Le véhicule est ici présenté raccourci (en démontant les trois panneaux solaires arrière) et il a récemment reçu :

- une nouvelle batterie
- un BMS 123SmartBMS (Battery Management System) pouvant communiquer en bluetooth
- deux écrans tactiles (pilote et copilote)
- deux Raspberry Pi

Cette nouvelle configuration permet le développement et le test d'applications en lien avec le transport de personnes dans un environnement "smart city", et en connexion avec des plateformes de données ouvertes (open data)

Ecolocar

9. OBJECTIFS DU CAMPOSV

Le hackathon **CampOSV 2018** démarre avec la présentation de l'Ecolocar par Marvin, suivie d'un tour de table pour dégager des objectifs. Chacun se présente rapidement et expose les idées ou pistes qu'il/elle souhaite développer au cours du Hackathon.

Les axes de travail suivants apparaissent alors :

Réflexion sur le design de la carrosserie :

- Modélisation du chassis existant
- Modélisation d'une nouvelle carrosserie

Branchement et finalisation des nouveaux appareils :

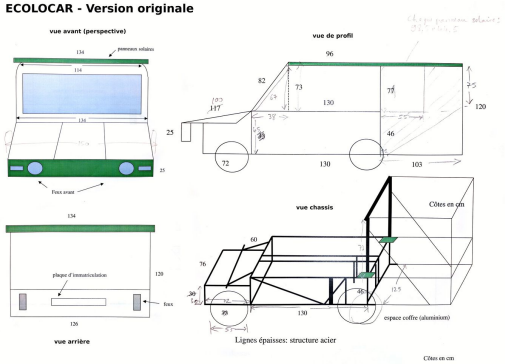
- Branchement des deux Raspberry Pi pour faire fonctionner les écrans du tableau de bord.
- Mise en fonction du nouveau BMS en discussion avec les autres ateliers du Hackathon.
- Création d'une calandre pour habiller les phares.

Mise à jour du concept "Ecolocar version 2018"

- Plateforme mobile d'expérimentation (Système logiciel open source et modulaire permettant d'ajouter simplement des services, des capteurs ?)
- Quelle est l'âme du projet ? Version "light" de l' Open source vehicle ?
- Inscription dans une gamme d'autres véhicules (vélo, dirigeable, bateau ?..)
- Questions sociales, économiques, légales...

10. MODÉLISATION 3D DE L'EXISTANT

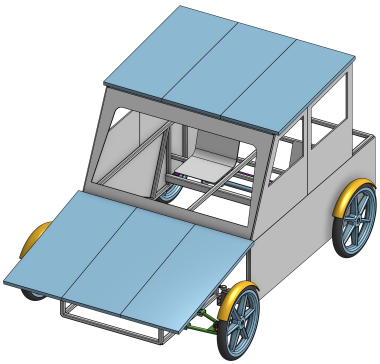
Arthur et Gregory se sont occupé de modéliser la structure existante à partir de mesures prises directement sur l'Ecolocar et de la documentation d'origine créée par Marvin en 2002.

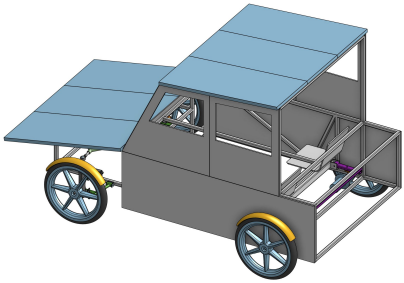


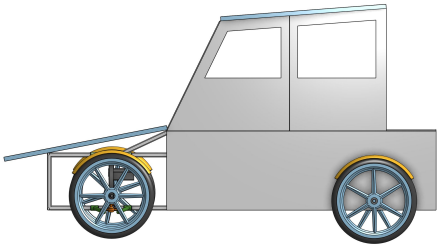
L'objectif étant de permettre de modéliser les évolutions à venir de l'Ecolocar sur la base de l'existant, panneaux solaires, sièges, cadre, carosserie et roues ont ainsi été construits en 3D sur la plate-forme OnShape.

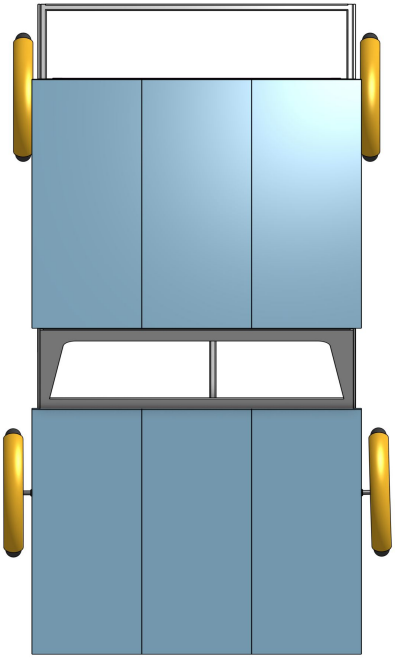
Le fichier modifiable est accessible sur cette URL :
https://cad.onshape.com/documents/61afedd481cae68a8a120006/w/8e2cbdbc7c1dded91eac8274/e/c94031e94aa713756807d217

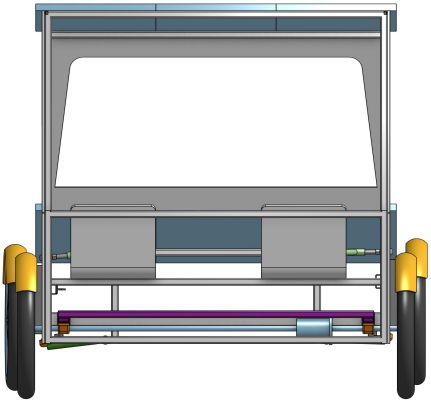
Captures du fichier créé :





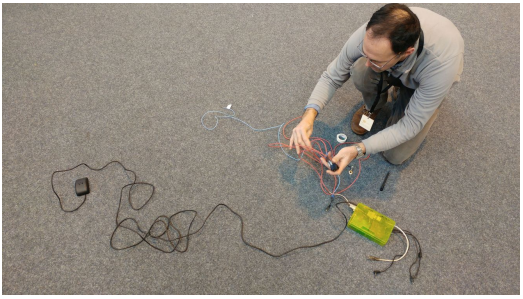


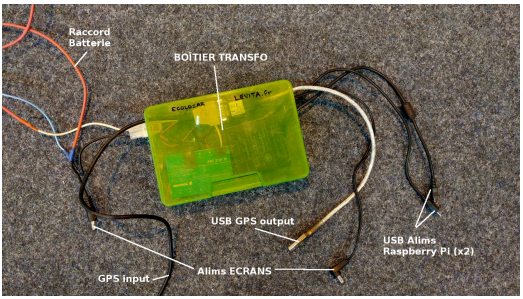




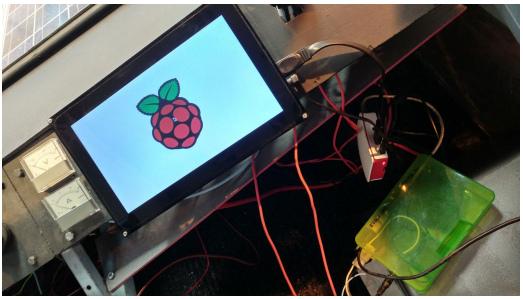
11. TABLEAUX DE BORDS

Pour faire fonctionner les deux écrans installés au préalable dans le cockpit de l'Ecolocar, Marvin s'est procuré deux Raspberry Pi et a confectionné un petit boîtier permettant de les alimenter directement depuis la batterie du véhicule.



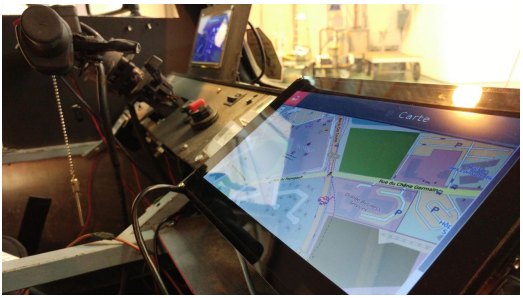


Une fois le boitier raccordé à la batterie, quatre cordons permettent d'alimenter les deux Raspberry Pi, les deux écrans, et de raccorder un GPS directement sur le Raspberry Pi de navigation.



Des scripts programmés par un ami de Marvin se lancent au démarrage des Raspberry Pi. (Ces scripts seront prochainement documentés sur le site levita.fr)

Le premier Raspberry permet d'afficher la vitesse du véhicule *(ici en simulation car non-connecté au capteur)*, des informations sur la charge de la batterie, et via une connexion internet de remonter des infos depuis le domicile de l'utilisateur.



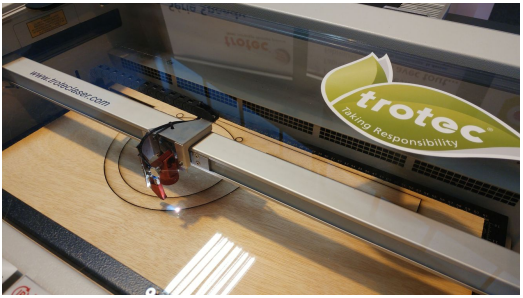
Le second Raspberry affiche une carte Open Street Map centrée sur la position de l'Ecolocar en fonction des données relayées par le GPS.

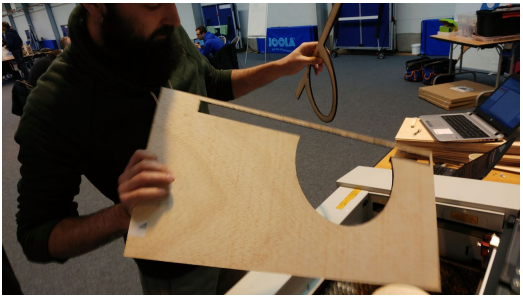
La prochaine étape de développement consistera à relier différents capteurs aux Raspberry Pi, en passant par une carte Arduino et en récupérant les infos du nouveau BMS (Battery Management System). A terme, ce petit système informatique devrait pouvoir se connecter au cloud via internet, afin d'échanger et utiliser différentes données liées aux services "Smart City".

12. CALANDRE LASERCUT

En utilisant la découpeuse laser Trotec mise à disposition sur le CampOSV, Marvin décide de réaliser une maquette de calandre afin d'habiller les phares avant de l'Ecolocar. Avec l'aide de Romain *(pour l'utilisation de la découpeuse)* et de Philippe *(pour designer la plaque centrale)*, la calandre est ainsi fabriquée en quelques heures.

Les dimensions maximales de la découpeuse obligent à construire la calandre en trois parties *(une pour chaque phare et une centrale pour l'affichage du nom, qui relie les deux premières)*





La plaque centrale dessinnée par Philippe, présentait à l'origine des décors végétaux qui n'ont pas pu être gravés en raison d'un problème technique de compatibilité avec le logiciel de contrôle de la découpeuse. Le temps à manqué pour résoudre ce problème..

FICHIERS SVG LASERCUT :

Phare Marvin Droite.svg

Phare Marvin Gauche.svg

PLAQUE ECOLOCAR.svg

13. PROJECTIONS, ÉVOLUTIONS

Philippe a repenser le design de la carrosserie, en se basant sur le chassis existant, afin d'offrir à l'Ecolocar un look plus contemporain et en accord avec les valeurs de respect de l'environnement qu'elle représente.

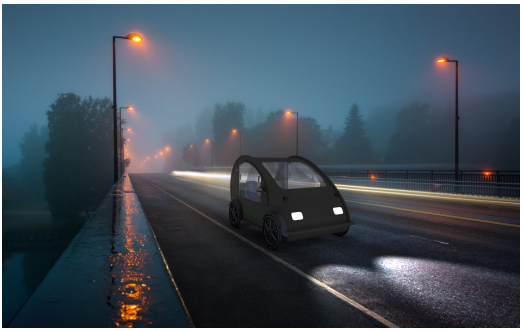
Cette nouvelle apparence a ainsi été modélisée sous le logiciel FormZ, en privilégiant des matériaux légers et simples à travailler comme le bois. Les panneaux solaires ont disparu dans cette nouvelle version, potentiellement remplaçables par des panneaux souples ou toutes autres technologies récente.

Par ailleurs, Fatoumata, Gabriel et Alexandre, étudiants en master "Stratégies digitales et innovation numérique", ont produit un dossier d'étude de marché pour présenter le concept Ecolocar dans ses aspects économiques, marketing, communication : **HACKATHON PROJET ECOLOCAR.pdf**

Captures de la modélisation 3D


103 AAB 35
écolocz


103 AAB 35




103 AAB 35

INTERNET DES OBJETS (IOT)

14. QU'EST-CE QUE L'INTERNET DES OBJETS ?
15. PARTICIPANTS
16. EXEMPLE DE CAS D'USAGE : CALCUL DE LA QUALITÉ DE L'AIR
17. QUEL CHALLENGE POUR LES IOT ?
18. LORA
19. CAS D'USAGE : REMONTER DES INFORMATIONS DU BMS
20. POINT INDUSTRIE 4.0 ATELIER IOT

14. QU'EST-CE QUE L'INTERNET DES OBJETS ?

L'internet des objets ou IdO (en anglais IoT pour Internet of Things), représente les échanges d'informations et de données provenants de dispositifs du monde réel avec le réseau internet.

L'IdO est à l'irigine du big data (mégadonnées en français) vu l'accroissement exponentiel du nombre d'objets connectés qui envoient leurs données. Il existe un grand nombre d'objet connectés, en usage dans les habitations ou même dans votre poche (ou votre sac), votre voiture etc. Cela concerne alors les systèmes embarqués.

Cependant la communication entre objets n'est pas approprié au réseau 4G ou 5G. En effet les objets peuvent être des milliers et ils peuvent échanger peu de données par jour. Il est donc plus intéressant d'utiliser un réseau longue distance et bien moins gourmand en ressource.

Le tableau suivant informe sur la porté des différents réseaux.

Technology	Frequency	Data Rate	Range	Power Usage	Cost
2G/3G	Cellular Bands	10 Mbps	Several Miles	High	High
Bluetooth/BLE	2.4Ghz	1, 2, 3 Mbps	~300 feet	Low	Low
802.15.4	subGhz, 2.4GHz	40, 250 kbps	> 100 square miles	Low	Low
LoRa	subGhz	< 50 kbps	1-3 miles	Low	Medium
LTE Cat 0/1	Cellular Bands	1-10 Mbps	Several Miles	Medium	High
NB-IoT	Cellular Bands	0.1-1 Mbps	Several Miles	Medium	High
SigFox	subGhz	< 1 kbps	Several Miles	Low	Medium
Weightless	subGhz	0.1-24 Mbps	Several Miles	Low	Low
Wi-Fi	subGhz, 2.4Ghz, 5Ghz	0.1-54 Mbps	< 300 feet	Medium	Low
WirelessHART	2.4Ghz	250 kbps	~300 feet	Medium	Medium
ZigBee	2.4Ghz	250 kbps	~300 feet	Low	Medium
Z-Wave	subGhz	40 kbps	~100 feet	Low	Medium

15. PARTICIPANTS

- **Phil coval.** Développeur.
- **Matthieu Brient**. J'ai pour objectif de rencontrer et expérimenter de nouveaux cas d'usages IoT et mobilité. Faire découvrir potentialités (et contraintes) des réseaux LPWAN et l'importance de l'interopérabilité des systèmes (Acklio/IETF)
- **Elisa de Castro Guerra** : mon objectif est de documenter ce qui se passe durant ce hackathon. Cette doc sera sur Floss Manuals Francophone (https://fr.flossmanuals.net/camposv/) et sur le wiki du projet (http://wiki.lafabriquedesmobilites.fr)/wiki/VehiculeLibre)
- **Adrien Rigobello** : mon objectif est de rencontrer et expérimenter de nouveaux cas d'usages IoT et mobilité. Cartographie des données.
- **Sébastien Gillot** : je suis venue en tant que visiteur assister aux conférences et j'ai découvert cet atelier pour apprendre à extraire des données d'un ensemble moteur électrique et batterie afin de l'exploiter avec un ordinateur. J'espère pouvoir réexploiter ces nouvelles connaissances afin de mieux libérer mon vélo électrique.
- **Veronique Sanguinetti**, mon objectif est de réfléchir sur les modèles d'affaires possible autour de l'Open Source des vehicules, comprendre les problèmes concrets posés par le hardware (doctorat en cours à ce sujet)
- Adrien et Quentin, étudiants en master stratégie digitale et innovation numérique. Notre objectif est de comprendre et savoir vulgariser pour des personnes étrangères au projet, de penser aux applications réels, élaborer un modèle économique.

16. EXEMPLE DE CAS D'USAGE : CALCUL DE LA QUALITÉ DE L'AIR

Une machine qui récupère et mesure la qualité de l'air ambiant a été conçu (démo lors du fosdem 2018).

Vous trouverez le code sur https://github.com/rzr/TizenRT.

LoRa a été utilisé pour que la machine placée sur un véhicule mobile envoie les mesures vers une antenne relais qui renvoie vers une page web (par exemple).

Vous pouvez voir une vidéo à ce sujet sur youtube : https://www.youtube.com/watch?v=S7zpBpnpfIU&feature=youtu.be#tizen-rt-lpwan-20180204rzr.

17. QUEL CHALLENGE POUR LES IOT ?

Devant ce nouveau chantier, des questions se posent et il s'agit des mêmes interrogations que celles au début d'internet.

Afin de pouvoir recevoir les données provenant des objets connectés, il faut pour cela assurer une interoperabilité entre :

- les protocoles (connectivité transparente, support de réseaux non IP)
- et les produits et les API.

Et cela de manière facile et fiable, sécure et respectueux de la vie privée.

Il faut en effet assurer l'interperabilité entre les objets et les systèmes afin de dépasser les silos créés entre les objets et les applications. Il existe plusieurs niveaux d'interopérabilité à gérer (technique, syntaxique, sémantique...) et privilégier le déploiement de standards par les instances de standardisation internationales (IETF, W3C, IEEE).

18. LORA

LoRA est à la fois un protocole de télécommunication bas débit qui diffuse par radio, et une carte électronique. Ce protocole de communication est très pertinent pour utiliser l'lenvoie des données par les objets connectés.

LE CONTEXTE

Architecture et fonctionnalités d'un réseau LPWAN (Low Power Wide Area Network) :

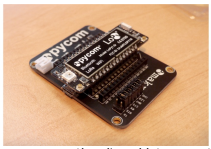
- https://datatracker.ietf.org/doc/draft-ietf-lpwan-overview/
- https://fr.wikipedia.org/wiki/LoRaWAN

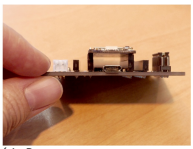
Un réseau LoRa Fabian a été déployé sur Rennes en 2016, pour en savoir plus : https://vimeo.com/151664693.

Pour information, la fréquence et MTU (maximum transmission unit) est de 1 message toutes les 3 minutes environ (duty cycle 1%). Taille des messages: 230 bytes.

DU CÔTÉ TECHNIQUE

- Objets (devices) : https://docs.pycom.io/
- LoRa Network Server: www.acklio.com/





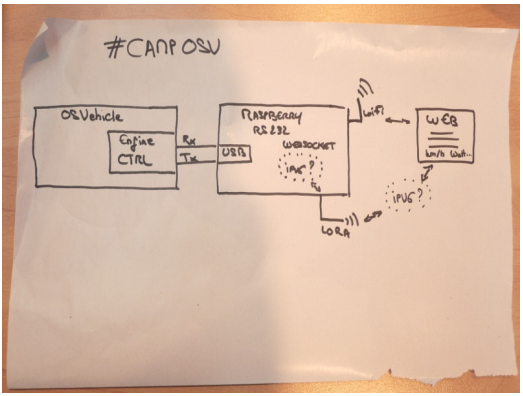
En savoir plus sur la programmation d'un objet connecté LoRa en Python (https://docs.pycom.io/)

LES PRÉ-REQUIS

- code Python: https://github.com/ltn22/PLIDO
- On préconise l'éditeur de code logiciel Atom (www.atom.io)
- Et plus spécifiquement l'utilisation de la librairie pymakr qu el'on peut installer avec les plugin.

19. CAS D'USAGE : REMONTER DES INFORMATIONS DU BMS

Lors du hackathon CampOSV, notre table a pour objectif de remonter des informations du BMS en passant par LoRA qui renverra les informations sur internet.



RENDU VISUEL DES INFORMATIONS ISSUES DU BMS

Ce rendu visuel a été réalisé en Html et JS avec un serveur en node JS avec le module WebSocket. Le graphique s'appuie sur gauge.js.

Vous trouverez le code utilisé sur ce dépôt :

https://github.com/luffah/websocket-example-osv

Ainsi qu'une démonstration visible sur vimeo :
https://vimeo.com/260345925#web-iot-automotive-20180315rzr#

ENVOIE DES INFORMATIONS VIA PYCOM ET LORA

Le troisième jour a été consacré à l'ajustement du code intégré à une raspberry afin d'envoyer en temps réel les informations sur la vitesse de la voiture. Les données sont ensuite envoyé à une carte Pycom qui utilise ensuite LoRA pour remontér les données dans des environnements contraints. En effet les réseaux LPWAN permettent de couvrir des distances plus grandes que les réseaux cellulaires plus traditionnels.

Schématiquement, nous pourrions résumé :

Véhicule BMS => Raspberry => Pycom/Lopy => LoRA Network Server => Application.

Les ressources utilisés :

- https://forum.pycom.io/topic/2514/lopy-to-raspberry-pi-3-serial-connection/18
- https://gist.github.com/matthieubrient/95f87ba079f510aed0b1ca597b33c458


EXPEMB
360
Embedded Experts
L1
L2
L3
L4

20. POINT INDUSTRIE 4.0
ATELIER IOT

Travail porté par les étudiants en stratégie innovante.

DESCRIPTION DE L'ATELIER

Systeme lora, un emetteur qui consomme très peu d'energie, sur plusieurs années, peut diffuser sur plusieurs km (faible cout energie et grande distance, inconvenient : peut envoyer seulement 6 messages par heures de 200 Kilo octets chacun).

OBJECTIFS

=> bien identifier les avantages et les limites

Le groupe a identifié des usages possibles. par exemple interagir avec des feux de signalisation, ou avec un BMS d'une batterie qui peut envoyer des infos sur la charge

RÉFLEXION

cout : moins d'un euro.

Reflechir qui fait les investissements ? la ville, les constructeurs ?

LES BESOINS AUXQUELS CELA RÉPOND

Quel est le besoin en termes de fonction et de cout ? en fonction du besoin, regarder (principe du double diamant) les technologies possibles, puis faire un cahier des charges et travailler. en automobile, un euro c'est un euro, on travaille parfois au centime. PSa a fabriqué 3,6 millions de voitures en 2017 (à verifier).

Cout de la puce + cout de l'integration, implantation dans le vehicule, stratégie comment il communique avec le vehicule (comment on le développe, temps de développement).

Mettre en place plusieurs scenarios d'usage du point de vue d'un constructeur, d'un individu, d'une ville.

Comme c'est OS, les universités, IUT etc vont pouvoir faire des TP dessus.
BMS vendu en BtoB : vendre des services pour adapter BMS et certifier securité.

PROPOSITION

Modèle de services reposant sur du hardware et du BMS Open source. Les services sont propres au BMS.
Clients : automobile.
Actuellement, les fabricants de BMS sont fabricants automobiles. Le BMS est breveté et propre à chaque batterie. Ils l'adaptent pour leur propre vehicule.

DOC JOINTE

Réalisation d'un document (il est en cours d'élaboration - semaine du 19 mars 2018).

VÉHICULE OSV ADAPTÉ AU HANDICAP

21. PARTICIPANTS DE L'ATELIER
22. ETAT DE L'ART DES VÉHICULES EXISTANTS
23. HOMOLOGATION
24. CAHIER DES CHARGES DU VÉHICULE
25. LES MOTEURS ÉLECTRIQUES KEOLIS
26. AUTRES ACCESSOIRES ET SYSTÈMES RÉUTILISABLES
27. PREMIERS ÉLÉMENTS DU PROTOTYPE
28. INDUSTRIE 4.0

21. PARTICIPANTS DE L'ATELIER

L'équipe de My Human Kit : Clara, Delphine, Yohann, John, Stéphane

Vincent (Territoires open-source)

Agathe, Pauline et Victoria, étudiantes en master stratégie digitale & innovation numérique

Séraphin, Oscar et Omar, étudiants en mécanique à Rennes 1

Pierre (Flossmanuals)

Un pad a servi à prendre des notes pendant les 3 jours du hackathon : dernière version enregistrée, le 15 mars à 16h40

22. ETAT DE L'ART DES VÉHICULES EXISTANTS

OLLI

Véhicule autonome de transport urbain, "customisable", moteur électrique, 8 passagers maximum.

web : https://localmotors.com/meet-olli/

Autonomie : de 40 à 60 km, selon la charge
Vitesse maximum : 40 km/h
Temps de charge électrique : 4.5 heures
Poids : 1840 kg
Capacité de charge : 800 kg
Dimensions : longueur 3.92m, largeur 2.05m, hauteur : 2.5m

KENGURU

Véhicule adapté individuel, moteur électrique.

web : http://www.kengurucars.co.uk/

Autonomie : 50 km
Vitesse maximum : 50 km/h
Poids : 290 kg sans les batteries
Dimensions : longueur 2.15m, largeur 1.55m, hauteur : 1.47m

ELBEE MOBILITY

Véhicule adapté individuel de catégorie quadricycle lourd, moteur à essence

web : http://web.elbeemobility.com/fr/

Vitesse maximum : 80 km/h
Dimensions : longueur 2.48m, largeur 1.33m, hauteur 1.725m
Largeur de passage d'intérieur : 770 mm
Hauteur de passage d'intérieur : 1 420 mm
Poids : 400 kg

KIMSI

Véhicule adapté, "sans permis", capable d'embarquer 2 personnes. La société fabricante, Ellectra, est en cours de rachat après un sinistre qui a ravagé ses ateliers en 2016.

web : http://kimsi.eu

Autonomie : de 80 à 100 km
Vitesse maximum : 45 km/h
Temps de charge électrique : 6 à 8 heures
Poids : ?
Capacité de charge : ?
Dimensions : longueur 3.38m, largeur 1.65m, hauteur : 1.82m
Largeur passage arrière : 0.90 m

HANDISCOOT

Scooter tricycle adapté (avec rampe d'accès pour fauteuil), moteur à essence

web : http://handiscoot.fr/

Autonomie : ?
Vitesse maximum : 45 km/h
Moteur : monocylindre 4 temps 2 soupages, cylindrée 49 cm²
Poids : 150 kg
Capacité de charge : 182 kg
Dimensions : longueur ?, largeur 1.645m, hauteur 0.955m

BEAD

Scooter tricycle adapté (avec rampe d'accès pour fauteuil), moteur électrique

web : http://www.bead.nu/

Autonomie : 45 km (batteries au plomb), 95 km (batteries Lithium-ion)
Vitesse maximum : 20 km/h
Poids : 192 kg (avec batteries au plomb)
Poids : 150 kg
Dimensions : longueur 1.63m, largeur 1.14m

XYT

Projet en démarrage.

web : http://xyt.fr/

LOL-E PRINCEPS

Il semblerait que ce véhicule ne soit plus fabriqué.

Véhicule quadricycle adapté individuel

web : https://www.facebook.com/LOL-E-492909780766040/

VEXEL QUOVIS

Ce véhicule n'est plus en fabrication

Véhicule quadricycle adapté individuel

web : http://www.quovis.com/index.htm

23. HOMOLOGATION

Pour qu'un véhicule puisse rouler sur les voies publiques, il doit être homologué par les autorités administratives. Cette démarche s'appelle la "réception", et peut concerner une série de véhicules ou un exemplaire unique.
La première après-midi du hackathon est consacrée à explorer la législation pour l'homologation de véhicules, afin de prendre en compte ces contraintes pour définir un cahier des charges adapté : quel type de véhicule peut-on imaginer qui puisse transporter une personne en fauteuil roulant lourd et son accompagnant(e).

Après ces recherches juridiques, complétées par des informations obtenues auprès de la DREAL, une piste de travail est conservée : la conception d'un quadricycle léger à moteur (catégorie administrative L6E)

Budget à prévoir : pour faire valider un dossier auprès de l'UTAC, service technique associé à l'homologation, il faut compter 4000€

INFORMATIONS COMPLÉMENTAIRES SUR L'HOMOLOGATION

Qu'est ce qui définit les conditions d'homologation d'un véhicule ?

C'est l'arrêté du 19 juillet 1954, dans sa version actualisée qui définit les conditions de réception d'un véhicule, en particulier l'annexe I pour la réception à titre isolé d'un véhicule neuf : texte de l'arrêté sur legifrance

Quelles sont les autorités administratives compétentes pour la réception des véhicules ?

Les arrêtés d'application de la directive 2007/46/CE et des règlements (UE) n° 167 et 168/2013 définissent les attributions des différents services concernés (source, mars 2018) :

- La sous-direction de la sécurité et des émissions des véhicules exerce par délégation du Ministre en charge des Transports, la fonction d'autorité compétente en matière de réception ;
- Les services de la DRIEE/DREAL/DEAL exerce la fonction de service administratif en charge des réceptions des véhicules
- L'Union technique de l'automobile, du motocycle et du cycle (UTAC), autodrome de Linas, 91310 Montlhéry est désigné par la sous-direction de la sécurité et des émissions des véhicules, comme service technique .
- L'organisme technique central (OTC) est désigné par la sous-direction de la sécurité et des émissions des véhicules, pour effectuer toutes les opérations liées à l'attribution du code national d'identification du type (CNIT) délivré aux véhicules réceptionnés par type, à leur communication aux autorités en charge de l'immatriculation, à la constitution et la maintenance de la banque de données de réception des types de ces véhicules, et à la surveillance de l'évolution des caractéristiques techniques et des performances de ces véhicules.

24. CAHIER DES CHARGES DU VÉHICULE

Durant les 3 jours du hackathon, le cahier des charges s'est affiné en prenant en compte tout d'abord l'usage, puis les contraintes réglementaires de l'homologation et les contraintes techniques de la construction automobile.

Choix du véhicule : quadricycle léger à moteur (catégorie administrative L6E)

Intentions de départ :

- Pouvoir rentrer dans le véhicule avec un fauteuil lourd, sans transfert.
- Fixation du fauteuil la plus simple possible.
- Respect de la législation (homologation, fixations du fauteuil, etc.)
- Capable d'embarquer deux personnes : conducteur avec bras valides et fauteuil lourd (ou valide), accompagnant.
- Poste de conduite ergonomique.
- Conduite sans permis.
- Conduite par guidon.
- Toutes les commandes au guidon (frein et accélération peuvent être déportés au guidon sans sortir du cadre législatif).
- Réutiliser les moteurs de vélos électriques Keolis.
- Entrée par l'arrière pour la personne en fauteuil et la personne accompagnante.
- Autonomie de 50 km (ce qui déterminera la quantité et le poids des batteries)

Contraintes liées à l'homologation en catégorie L6E :

- Poids du véhicule inférieur à 425 kg à vide.
- Charge utile inférieure à 250 kg pour des personnes (*) (300 kg pour des marchandises).
- Vitesse: entre 6 et 45 km/h.
- Transport de deux personnes maximum, donc le conducteur (*).
- Moteur de puissance inférieure à 4 kW.
- Permis AM ou BSR ou A1 ou B1 ou B ou A obligatoire pour tricycle.
- Largeur maximum : 2.55 m
- Longueur maximum : 4 m
- Colonne de direction ayant déjà fait l'objet d'essais réglementaires.

La DREAL conseille une homologation en petite série si plus d'un modèle est construit.

(*) Pour les calculs de charge, le poids d'une personne est fixé à une valeur forfaitaire de 75 kg, le poids d'une personne et d'un fauteuil roulant est fixé forfaitairement à 160 kg.

Contraintes en réflexion :

- Le siège du conducteur peut se déplacer latéralement sur le côté passager.
- Entrée par arrière pour la personne en fauteuil et la personne accompagnante.
- Portes latérales pour accompagnant.
- Les deux personnes dans l'habitacle sont elles côte à côte ou l'une derrière l'autre?

Autres éléments à prendre en compte :

- Taille d'un fauteuil : 150 x 90 x 150 (hauteur)
- Taille d'une place de parking standard : 2.5m x 5m (une place de parking handicapé fait 3 m x 5 m)
- Pour qu'une rampe amovible ne soit pas soumise à dérogation, sa pente doit être inférieure à 10% (si la longueur de la rampe est inférieure à 2 m), ou 12% si la longeur de la rampe est inférieure à 0.5 m

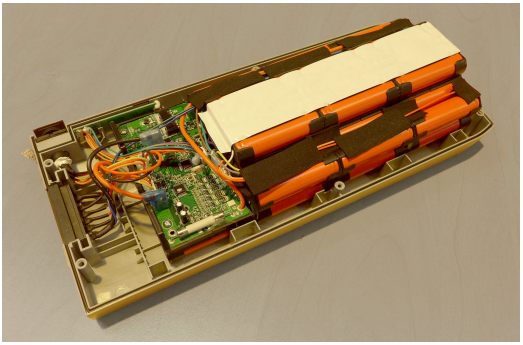
25. LES MOTEURS ÉLECTRIQUES KEOLIS

Keolis est l'opérateur privé qui fournit le service de vélos électriques à la métropole de Rennes. Après 5 ans d'utilisation, ces vélos sont retirés du service, cédés à l'association La Petite Reine et "désélectrifiés". Les batteries et les moteurs électriques sont cédés gracieusement à l'association My Human Kit par La Petite Reine.

Un des objectifs de ce hackathon est d'évaluer la réutilisation possible de ces systèmes électriques pour motoriser le véhicule.

Le moteur électrique est de type "brushless", la carte électronique fournie avec étant un circuit propriétaire, une alternative libre a été choisie, il s'agit du ESC de Benjamin Vetter.

CARACTÉRISTIQUES DES BATTERIES



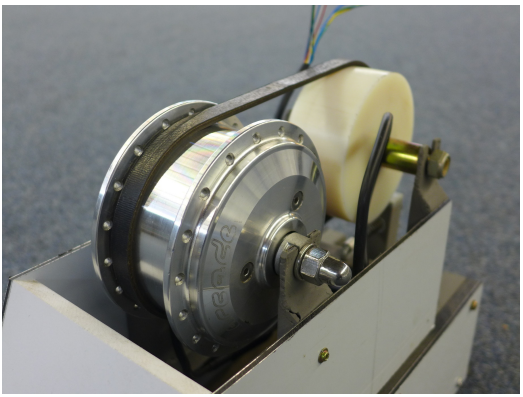
type : Li-ion

marque : Joycube

modèle : JCEB360-8.8

caractéristiques électriques : DC36.0V, 8.8 Ah, 316.8 Wh

BANC DE TEST



Un banc de test est en cours de construction pour évaluer les caractéristiques des moteurs.

CONTRÔLEUR ESC (ELECTRONIC SPEED CONTROL)

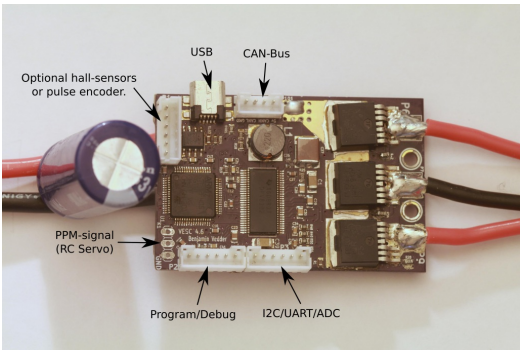


photo : Benjamin Vetter

Il s'agit d'un circuit électronique permettant de contrôler la vitesse du moteur. Le circuit utilisé est concu par Benjamin Vetter et placé sous licence Creative Common.

Informations complètes sur ce circuit : VESC

Informations complémentaires sur wikipédia (en anglais) : fonctionnement des ESC

26. AUTRES ACCESSOIRES ET SYSTÈMES RÉUTILISABLES

Pour faciliter la procédure d'homologation, il est intéressant de réutiliser des systèmes qui ont déjà faits l'objet d'essais réglementaires, par exemple pour les phares, les freins ou la colonne de direction.

FIXATIONS DE FAUTEUIL

La société Q'Straint vend des systèmes modulaires de fixations pour fauteuil roulant : sur un rail viennent se fixer des pièces d'accroche adaptée au poids du fauteuil

web : https://www.qstraint.com/fr/product-finder/

27. PREMIERS ÉLÉMENTS DU PROTOTYPE

Dimensions : 300 cm long, 150cm de large (dont 90cm d'espace libre au centre et espace de 30cm pour les étagères + batteries sur chaque côtés), 150cm de hauteur
Concept : la porte arrière fait environ 80 cm de hauteur et sert à prolonger la rampe qui démarre à 50 cm de l'avant (voir plan-voiture ci-dessous)

Chassis en alu ou acier (à voir selon le poids)
Carrosserie et bas de caisse en sandwich naval (mélange de bois contreplaqué et résine..)

SCHÉMAS

http://media.myhumankit.org/OSV/plan-voiture.jpg
http://media.myhumankit.org/OSV/attaches-rampe.jpg
http://media.myhumankit.org/OSV/rampe.jpg
http://media.myhumankit.org/OSV/Anim-rampe.mov

LICENCE(S)

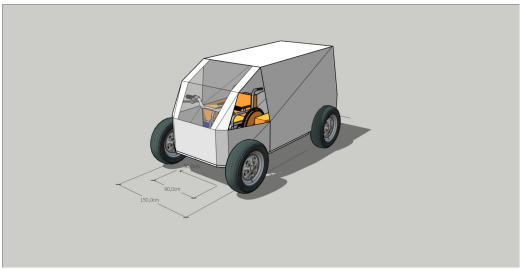
Open source hardware licence : CERN ou autre?

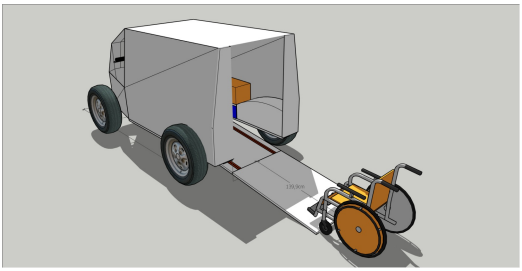
MATÉRIAUX

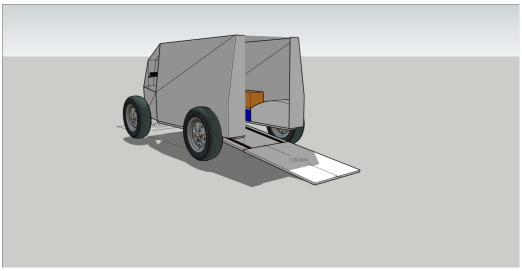
Sandwich naval (bois+résine) pour la coque
Chassis en alu ou acier

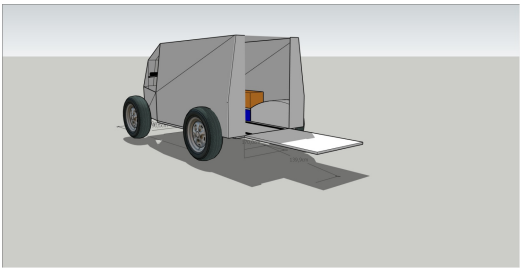
MODÉLISATION 3D

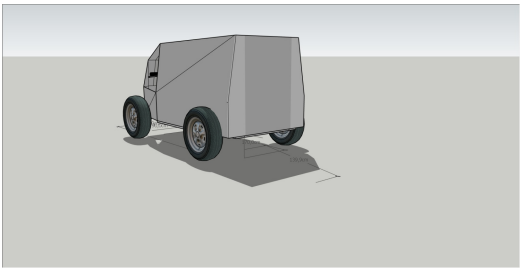
Fichier 3D sketchup : modèle 3D

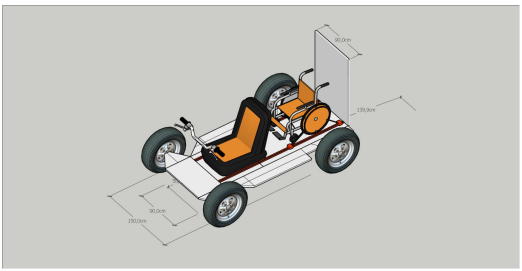


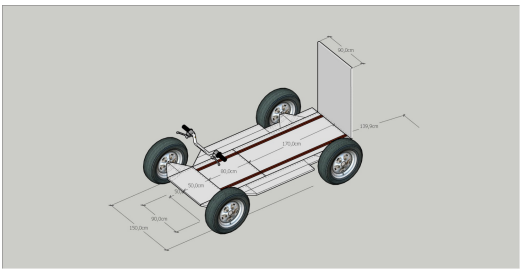


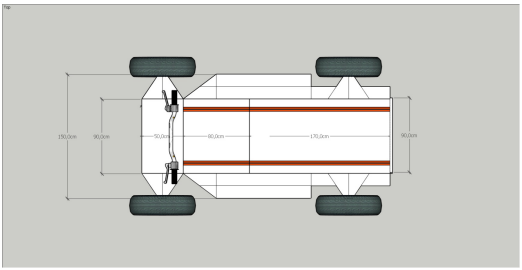


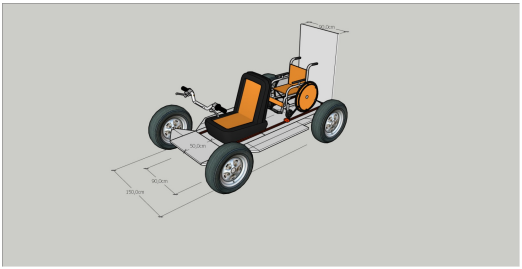












(A suivre...)

Un pad a servi à prendre des notes pendant les 3 jours du hackathon : dernière version enregistrée, le 15 mars à 16h40

28. INDUSTRIE 4.0

Pendant les 3 jours du hackathon, des étudiantes en master stratégie digitale ont accompagné le travaildu groupe pour réfléchir à différents modèles économiques adaptés. Leurs conclusions ont fait l'objet d'un document récapitulatif :

(à compléter prochainement par le document pdf)

Véronique Sanguinetti, enseignante en management de l'université de Valenciennes a également pris des notes pour définir un modèle économique adapté au projet :

législation compliquée , contraintes de poids. quadricycle : 250 kg de charge utile (donc il faudrait maxi deux personnes à 50 kg), tricycle (500 kg)
il y a le poids du fauteuil à prendre en compte (150 KG) => gros
aspects juridiques à prendre en compte pour une homologation
obligation de crashtests en fonction du nb de vehicules produits (100 par an ?)
45 km/h
+ volonté de récupérer materiaux à la casse.
orientation vers deux modèles Tricycle et Quadricycle.
=> il y a déjà des modèles qui existent, (Kimsy?) qui permettent de conduire en fauteuil roulant. (30000 euros)
le faire en open source permettrait d'adapter à un fauteuil plus large, ou une taille plus grande. , faire moins cher
chaque vehicule serait specifique,
identifier le nombre de personnes ? 100 vehicules par an ? reflechir à des manières d'évaluer plus finement ?
comment pourrait il etre produit et distribué ?
Peugeot : sort 60 vehicules par heure. il peut y avoir une difference sur une option par vehicule.
les microcars sont assez semblables, voir comment ils fonctionnent ? ont une production petite série ?
fabricant fauteuils roulants ? doivent répondre à des normes, notamment pour etre intégrés dans vehicules. seraient -il intéressés à proposer ces vehicules pour compléter leur offre
pour quels usages ? uniquement en urbain ? quel avantage par rapport à l'offre actuelle de transport aidé ?
quel horizon ? 2020 ?2035 ? en fonction du délai, il faut switcher direct sur le vehicule autonome. les reglementations auront changé, quel serait le besoin ? il restera des spécificités (il faut un accès par l'arrière, supporter un poids plus fort). voir concept car Renault qui propose un accès par arrière
MAJ 15/03/2018
finalement choix du quadricycle léger.
342 000 personnes potentiellement interessées ?
financement : plusieurs partenaires - financement public (Agefip, GMF, ...)
solutions existantes non satisfaisantes (reserverune semaine à l'avance, ...)
Uber sur Paris : pour PMR. quelques startups créées par des PMR
objectif : rendre autonome. repondre aux contraintes de poids maxi.
le chassis doit pouvoir supporter le fauteuil roulant.
ds un premier temps PMR est passager, puis devrait pouvoir être conducteur.
Demande qui est plutot captive. pb des pieces de rechange qui seraient plus cheres ?
objectif arriver à un vehicule au prix de 20000 euros au lieu de 30000 aujourd'hui ?

PYOSV

29. PRÉSENTATION DE L'ATELIER
30. DÉROULÉ
31. FONCTIONNALITÉS DE PYOSV
32. UTILISATION DE PYOSV
33. PROTOTYPAGE D'UNE PLATEFORME DE DOCUMENTATION BASÉE SUR PYOSV
34. VALORISATION

29. PRÉSENTATION DE L'ATELIER

LES PARTICIPANTS

- Guillaume Florent : développeur de pyOSV
- Bernard Ugen : professeur d'université et pythoniste, développeur de pyOSV, reverso
- Alexandre Sanchez et Mathieu Simon, ingénieurs en informatique à InriaTech
- Julie Colin, architecte
- Trois étudiants en "Stratégie digitale et innovation numérique"

CONTEXTE

Le besoin d'un outil logiciel permettant de documenter du matériel libre conçu de façon collaborative s'est fair ressentir dès le premier campOSV.

Les solutions intégrées permettant la modélisation d'objet complexe, le PLM ("product life management"), l'utilisation de librairie de pièce, le tout de façon collaborative, n'existent pas en opensource et les solutions propriétaires sont trop couteuses pour les petites structures.

Les formats d'échange historique (STEP, IGES) ne sont pas assez évolué pour permettre une modélisation aussi avancée, ne permettent pas de capturer l'intention de conception ("design intent").

Le projet s'oriente autour de plusieurs axes :

- la création d'un format de représentation géométrique (mais pas que géométrique) sous forme de graphe,
- la création d'un outil permettant de générer des pièces standard à partir de modèles paramétriques (vis, roulement à billes) : party
- un outil d'ingériérie inversé permettant d'analyser des fichiers STEP, pour les factoriser afin de les rendre exploitables avec pyOSV: reversy

Les fonctionnalités de pyOSV :

- représentation d'un objet complexe composé de plusieurs pièces liées par des contraintes
- conservation sous forme de graphe des intentions de conception

L'objectif du projet pyOSV est de fournir un outil permettant de concevoir un objet complexe libre de source ("opensource hardware") comme on le ferait dans le logiciel libre ("opensource software") notamment en utilisant des technologies permettant le versionnage et d'édition de collaborative comme git

OBJECTIF

L'objectif de l'atelier est de proposer un prototype de plateforme de documentation utilisant les fonctionnalitées de PyOSV.

30. DÉROULÉ

MARDI 13 MARS

- Présentation des participants
- présentation des fonctionnalités et prise en main de pyOSV à l'aide de notebooks python, déployés avec docker

MERCREDI 14 MARS

- long et fructueux brainstorming autour de l'interface utilisateur de la plateforme de documentation

JEUDI 16 MARS

- projection à deux ans et modèle économique

31. FONCTIONNALITÉS DE PYOSV

PYOSV

PyOSV s'appuie sur pythonocc et ccad pour générer un modèle d'objet complexe prenant en compte les interrelations, notamment géométriques et mécaniques, entre les pièces composant cet objet complexe.

Il s'agit de représenter à la fois les composantes géométriques de chaque pièces ainsi que les relations entres les pièces sous forme de contrainte notamment géométrique.

L'outil s'appuie sur 3 couches logiciels :

- opencascade, à la base, est une bibliothèque logicielle de traitement géométrique sur laquelle s'appuie de nombreux logiciels libres de CAO. C'est une librairie complète de "qualité industrielle", écrite en C
- pythonocc : pythonocc permet d'accéder aux fonctionnalité d'opencascade en python. Cette bibliothèque est développée par Thomas Paviot, partie prenante du projet pyOSV
- ccad : ccad est une interface python à pythonocc qui permet de manipuler des objets géométrique à un niveau d'abstraction plus élevé que pythonocc

pyOSV permet d'utiliser des pièce qui sont soit :

- modélisées directement en python en utilisant les foncionnalités de pythonocc/ccad.
- créée à partir de la librairie de pièces standards (projets "party")
- disponible au format STEP

Les données produites et utilisées par pyOSV sont représentée sous forme de graphe orienté.

Les objets sont des noeuds avec des relations orientés (par exemple soudure, assemblage, "je suis au dessus", "je suis en dessous"...). Chaque pièce est associées à une ou plusieurs ancres, qui expriment la relation spatiale entre deux objets et permets de remonter à l'intention de conception.

LES PROJETS ANNEXES

Différents projets sont nés en parralèle du coeur de pyOSV pour la modélisation d'objets complexes.

Party

Party permet de réaliser des modèles paramétriques de pièce afin de produire des librairies d'objets plus ou moins standard utilisables directement avec PyOSV. Chaque objet est décrit dans un format json sous forme de code python/pythonocc.

Reversy

Reversy est un outil permetant de faire de l'ingéniérie inverse sur des fichiers STEP afin de les décomposer en pièces élémentaires.

Il permet, à partir de la fouilles des données géométriques contenue dans le fichier, de reconnaitre des pièces qui ont été copiées puis utilisées en plusieurs exemplaires (par exemple des vis), et de les remettre dans un repère d'origine.

L'idée est d'avoir en sortie un fichier "factorisé", c'est à dire que chaque pièce n'est définie qu'une seule fois avant d'être éventuellement utilisées à plusieurs emplacement de l'objet complexe.

Il permet également d'extraire des données relationnelles entre les objets.

32. UTILISATION DE PYOSV

INSTALLATION

La façon la plus simple d'utiliser pyOSV pour l'instant est d'utiliser des notebooks empaqueté sous forme d'image Docker.

Docker est un utilitaire permettant de déployer simplement des projets web.

Sur un ordinateur doté d'une distribution linux récente (par Exemple Ubuntu 16.04, Debian 8 ou 9) disposant de 20Go d'espace libre :

- installer Docker (community edition)

En fonction de la distribution Linux utilisée:
https://docs.docker.com/install/linux/docker-ce/ubuntu/
https://docs.docker.com/install/linux/docker-ce/debian/
https://docs.docker.com/install/linux/docker-ce/centos/
https://docs.docker.com/install/linux/docker-ce/fedora/

- clôner pyosv-binderhub

$ git clone https://github.com/osv-team/pyosv-binderhub

- Installer l'image pyosv

$ cd pyosv-binderhub
$ ./install_pyosv.sh

Attendre plusieurs heures ..... la construction de l'image compile PythonOCC et d'autres projets, en
plus du téléchargement d'une image de base assez lourde.

- Test de fonctionnement

$ /start_pyosv.sh

prompt Docker container > cd /home/jovyan
prompt Docker container > jupyter notebook
Copier l'url affichée en résultat de la commande jupyter notebook dans un navigateur
Exécuter un notebook (/home/jovyan/work/notebooks) et vérifier que tout se passe sans erreur (les
warnings de deprecation sont normaux).

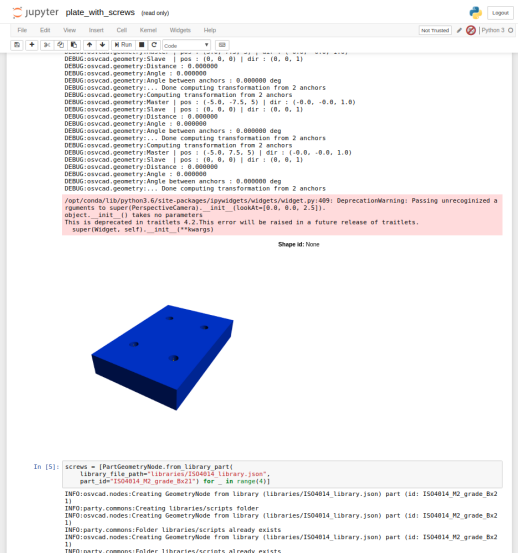
UN EXEMPLE DE NOTEBOOK

Un des notebooks donné en exemple pour le campOSV est une plaque avec 4 vis nommé "plate_with_screws"

Après avoir importé les dépendances, on crée une plaque avec 4 trous à l'aide de l'exécution du script python "plate_with_holes.py" :

```
# coding: utf-8

r"""Flat plate with holes Python creation script"""

from __future__ import division

from ccad.model import prism, filling, ngon, cylinder, translated, box

units = 'mm'

e = 5
l = 20
w = 30

hole_d = 2

hole_positions = ((l/4, -w/4), (l/4, w/4), (-l/4, -w/4), (-l/4, w/4))

plate = translated(box(l, w, e), (-l /2, -w/2, 0))

cylinders = list()

for (x, y) in hole_positions:
    cylinders.append(translated(cylinder(hole_d / 2., e), (x, y, 0)))

for c in cylinders:
    plate -= c
part = plate

anchors = dict()
for i, (x, y) in enumerate(hole_positions, 1):
    anchors[str(i)] = {"position": (x, y, e),
                       "direction": (0., 0., -1.),
                       "dimension": hole_d,
                       "description": "%s mm hole" % hole_d}
```

Voici le résultat obtenu dans le notebook :

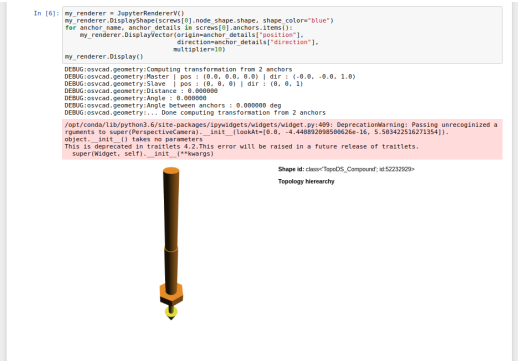


On crée ensuite 4 vis à partir de library_part. Ce programme permet de créer des pièces standards comme les vis avec leurs ancres qui permettront ensuite de les placer par rapport à la plaque.

Le code qui permet d'obtenir une liste contenant 4 vis est le suivant :

```
screws = [PartGeometryNode.from_library_part(
    library_file_path="libraries/ISO4014_library.json",
    part_id="ISO4014_M2_grade_8x21") for _ in range(4)]
```

L'ancre est représenté dans la figure suivante par la flèche jaune.



On crée ensuite un noeud "assemblage" avec la plaque comme origine.

```
A = AssemblyGeometryNode(root=plate_gn)
```

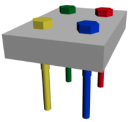
On ajoute à cet assemblage chacune des 4 vis en précisant l'ancre à laquelle elle doit s'attacher.

```
for i, screw in enumerate(screws, 1):
    A.add_edge(plate_gn,
               screw,
               object=ConstraintAnchor(anchor_name_master=str(i),
                                       anchor_name_slave=1,
                                       distance=0,
                                       angle=0))
```

Il n'y a plus qu'à construire l'objet :

```
A.build()
```

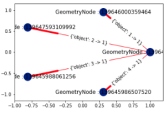
Quelques instructions sont encore nécessaire pour créer le rendu graphique. On obtient ainsi l'assemblage suivant ainsi que le graphe affiché en bas de la capture d'écran.



View the graph representing this simple assembly

In [16]:
```
A.show_plot()
```

```
DEBUG:matplotlib.font_manager:findfont: Matching :family=sans-serif:style=normal:variant=normal:weight=normal:stretc
h=normal:size=10.0 to DejaVu Sans ('/opt/conda/lib/python3.6/site-packages/matplotlib/mpl-data/fonts/ttf/DejaVuSans.
ttf') with score of 0.050000
DEBUG:matplotlib.font_manager:findfont: Matching :family=sans-serif:style=normal:variant=normal:weight=normal:stretc
h=normal:size=12.0 to DejaVu Sans ('/opt/conda/lib/python3.6/site-packages/matplotlib/mpl-data/fonts/ttf/DejaVuSans.
ttf') with score of 0.050000
```



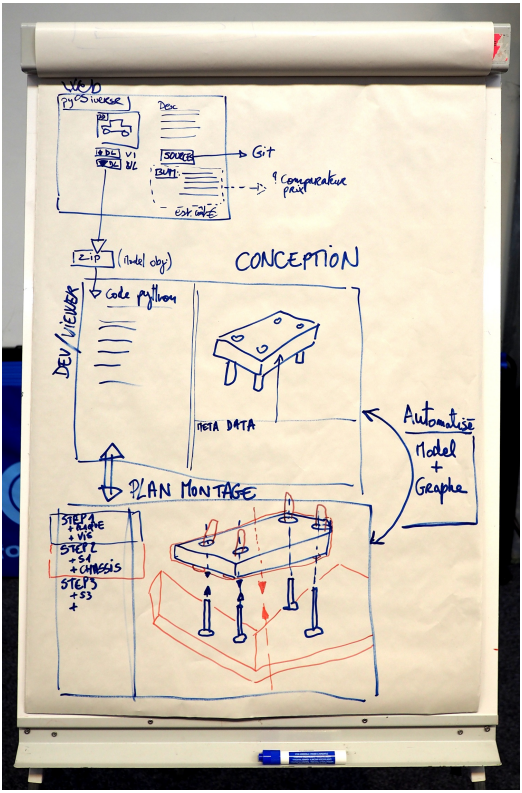
On retrouve les 5 pièces composants l'assemblage, chacune des quatre vis étant lié à la plaque.

33. PROTOTYPAGE D'UNE PLATEFORME DE DOCUMENTATION BASÉE SUR PYOSV

L'idée de départ est de transférer les méthodes utilisées dans le développement open source de logiciel au développement opensource d'objet, notamment en s'appuyant sur git.

Il s'agit de prototyper une version de gitlabCE qui permettrait d'intégrer pyOSV.

Dans un premier temps, il est envisagé d'ajouter une vue 3d dans une interface gitlab classique. Finalement, l'interface utilisateur de gitlab ne semblait pas forcément adapté à un public ne connaissant pas git. Il est alors envisagé de séparer l'interface en deux parties.



INTERFACE CONCEPTEUR

L'interface concepteur est orientée vers les personnes souhaitant modifier les objets avec un outil de gestion de version du type git.

Elle est séparée en plusieurs vues :

- Une vue au format notebook qui permet d'éditer et d'exécuter du code pyOSV
- Une vue 3d de rendu de l'objet
- Une vue représentant le graphe de l'objet

INTERFACE UTILISATEUR / MAKER

L'interface utilisateur/maker s'approche plus d'une page objet du site thingiverse, qui permet de voir l'objet, de télécharger le fichier au format zip.

Elle donne également accès à un plan de montage (séparé en étapes de montage) généré automatiquement à partir des données du graphe de l'objet.

GÉNÉRATION D'INSTRUCTION DE MONTAGE

Une partie du mercredi après midi a été consacré à discuter de l'opportunité de créer automatiquement des procedures de montage à partir du graphe d'objet. Il s'agit de réaliser de la fouille de données sur l'objet pour retrouver les différentes étapes de montage ou de fabrication et de les exporter sous forme d'un manuel d'instruction por le montage.

Le modèle nodale hierarchique nous fait penser qu'il ne serait pas nécessaire d'ajouter beaucoup de méta données "manuellement". Il s'agit de définir un algorithme qui permet, à partir du graphe de l'objet de retrouver les différentes étapes du montage en utilisant le fait que les liens qui unissent chacune des pièces par rapport à leurs ancres soient orientés.

GÉNÉRATION DE BOM

Le graphe et les métadonnées de chaque pièce constituant l'objet permettent aussi de créer automatiquement une "BOM" (pour "Bills of materials") qui est la liste des pièces à fabriquer ou à se procurer auprès d'un fournisseur pour réaliser l'objet et qui accompagne souvent la documentation des projets "open hardware".

34. VALORISATION

IDENTIFICATION DE DIFFÉRENTS TYPE D'UTILISATEURS

- l'utilisateur du projet
- le concepteur,
- le développeur (de la plateforme)

Identification de certains effets de réseaux.
-> Ce sont les concepteurs qui apportent la valeur ajoutée permettant d'attirer les utlisateurs ("makers").

Se concentrer sur les concepteurs. Importance de la communication

MODÈLES DE VALORISATION

- comparateurs de prix ou market place pour les BOMs : toutes les parties sont gagnantes, négociation pour la répartition des commissions, par en faveur de la plateforme dans un premier temps.

fournisseur "traditionnel" vs marketplace : quid de la gestion des stocks

- valorisation des données personnelles : niet
- modèle freemium : vente à des entreprises qualifiées, besoin de formation

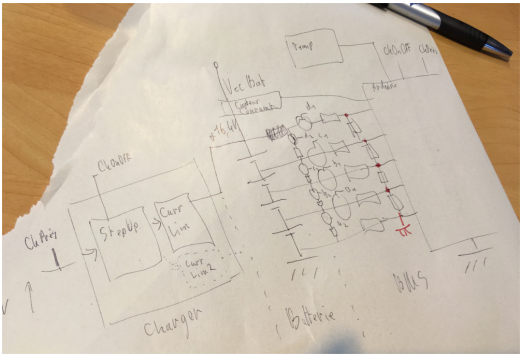
FONCTIONNEMENT D'UN BMS

35. FONCTIONNEMENT D'UN BMS

Le BMS (Battery Management System) est un système de gestion de batterie qui sécurise et recharge les batteries. Il permet de surveiller la tension basse des cellules, de gérer l'équilibrage à la recharge et de mesurer la température de la batterie. Ces paramètres sont importants afin d'avoir un bon fonctionnement de la batterie et d'augmenter sa durée de vie. L'objectif ici sera de réaliser un gestionnaire de batterie sur table qui puisse par la suite permettre à l'utilisateur de voir le niveau de la batterie de sa voiture et de contrôler la charge et la décharge de la batterie.

Pour le fonctionnement de ce BMS, il est nécessaire d'avoir :
- une batterie de 4 cellules lithium 18650
- un chargeur qui va permettre de charger les cellules de la batteries
- un limiteur de courant
- le BMS

Ces différents éléments seront ensuite assemblés ensemble pour aboutir à un gestionnaire pouvant contrôler la batterie.



36. LES PARTICIPANTS DU BMS

Guillaume Le Gall : Ingénieur de Recherche IMT Rennes

Damien Albert : Architecte cloud

Baptiste Gaultier : Ingénieur R&D IMT

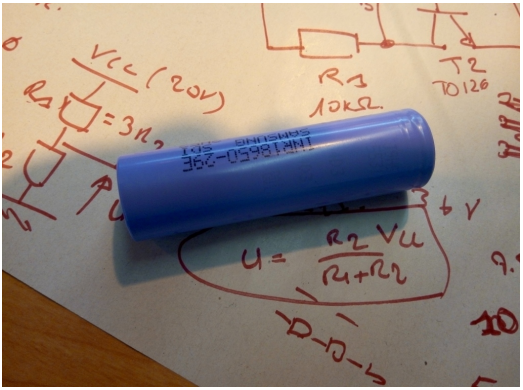
Fabienne Nouvel : Enseignante Chercheur IETR

Laura Barthélémy, Justine et : étudiantes en master stratégie digitale & innovation numérique

Cyprien Roudet et Margaux Girard : docMasters

Un pad a été crée pour les 3 jours : pad

37. LES BATTERIES À BASE DE CELLULES 18650



Les cellules 18650 sont des accumulateurs au Lithium-ion

Chaque cellule peut délivrer 3.7v. Elles peuvent être montées en série pour augmenter la tension ou en parallèle pour augmenter la capacité.

Elles sont rechargeables. Pour les recharger il faut appliquer une tension de 4.2V maximum sinon les cellules explosent.

Elles sont très courantes dans nos environnements (ordinateurs portables, vélo electriques, etc.).

Pour le montage réalisé ici, 4 cellules seront montées en série pour obtenir une tension de 14.8V.

Chargeur BMS

38. CRÉER UN CHARGEUR (COMPOSANT LM2586)

Pour charger des cellules, il faut leur envoyer du courant. Chaque cellule 18650, a besoin de 4.2V pour se recharger. Ici pour recharger 4 cellules, il faudra avoir besoin de 4.2V * 4 = 16.8V.

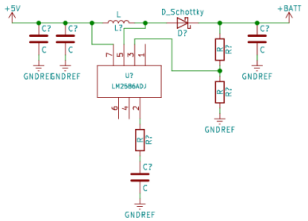
Un port USB peut fournir 5V de courant. La battere doit pouvoir se recharger à partir d'un port USB, le chargeur recevra donc 5V en entrée de circuit et fournirra 16.8V en sortie.

Pour obtenir cette tension de 16.8V, le composant LM2586-ADJ est utilisé. Ce composant est un régulateur de tension a pour rôle de maintenir à sa sortie une tension constante, indépendamment de la tension d'entrée. La tension obtenu est bien de 16.8V.

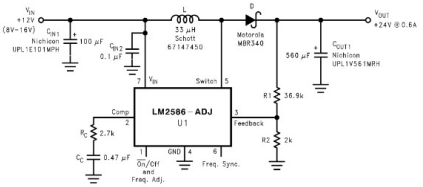
Vidéo :

Chargeur BMS

Shéma Kicad :



Shéma du datasheet du composant :

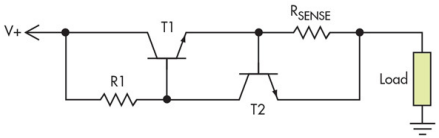


39. LIMITEUR DE COURANT

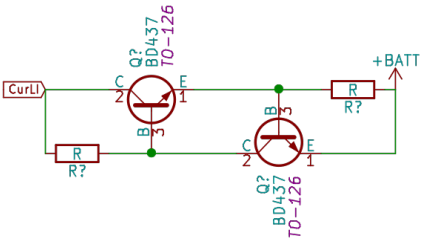
Un limiteur de courant est ajouté à la suite du chargeur. Ce limiteur de courant va limiter la valeur du courant et ainsi d'éviter de charger trop rapidement les cellules.

Lien vers la documentation du limiteur de courant

Limiteur de courant



V+ : valeur récupérée à la sortie du chargeur
T1 et T2 : transistor MOSFET TO-126
R1 : 10 kOhms
Rsense : 1 Ohm



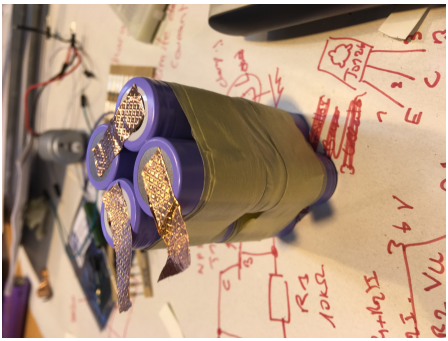
La tension à la sortie du chargeur a été mesurée à 16.9V avec le limiteur de courant la tension est descendu à 16.4V. Le limiteur de courant a bien fait son travail, il réduit la vitesse de chargement. Cette valeur de tension correspond à ce qui était attendu pour une batterie de 4 cellules puisque une cellule à son seuil de tension est de 4.2V (4.2V*4cellules = 16.8V).

40. LA BATTERIE 4 CELLULES 18650

MISE EN SÉRIE DES 4 CELLULES

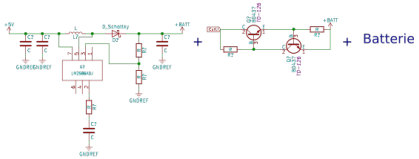
Pour la batterie, les 4 cellules en série. Elles sont connectées entre elles par un ruban conducteur.

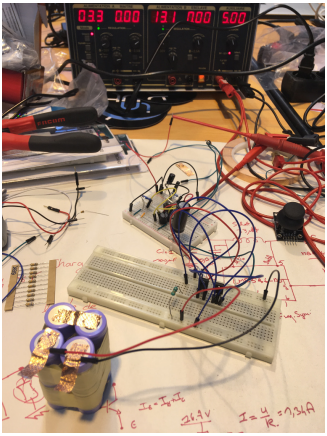
La tension mesurée en sortie est alors de 14.04V pour des cellules chargées à 3.5V chacune (Les cellules sont donc déchargées - Chargées elles auront une tension à 3.7V).



CHARGEMENT DE LA BATTERIE

Le chargeur, le limiteur de courant et la batterie sont reliés entre eux. Le chargement de la batterie peut se faire.

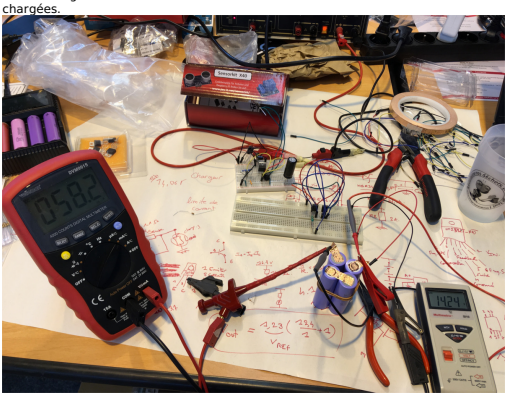




En laissant en charge pendant 1h la tension de la batterie est montée à 14.16V.

Batterie avant le chargement : 14.04V
Batterie après 45min de charge : 14.12V
Batterie après 1h de charge : 14.16V

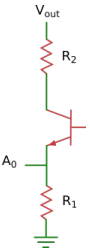
=> Le chargeur fonctionne sans détecter si les batteries sont bien chargées.



41. MESURE DES TENSIONS

Pour pouvoir gérer le niveau de charge de la batterie, il est nécessaire de connaître la tension de chaque cellule de la batterie. La mesure de tension pour chaque cellule se fait par un pont diviseur de tension. Les sorties de chaque cellule sont branchées sur l'arduino (A0, A1, A2 et A3) qui va mesurer la tension des cellules.

Schéma du circuit de mesure de tension d'une cellule.



R1 = 1.2kOhm
R2 = 2.2kOhms
Transistors Mosfet utilisés : TO-92

Pour mesure la tension de la cellule, la formule a utilisé est celle-ci
VA0 = R1*I

Vout = (R1+R2)*I + Vt

Où I = (Vout - Vt) / (R1+R2)

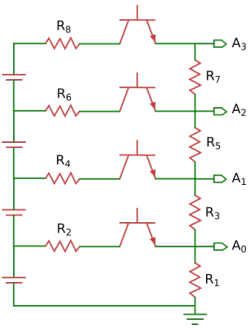
Donc VA0 = R1*(Vout - Vt) / (R1+R2)

Vt étant la tension mesurée au transistor

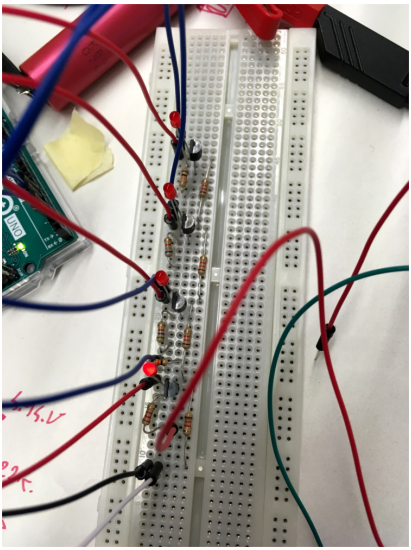
La tension Vout obtenu est de 4.14V et la tension au transistor de 0.6V

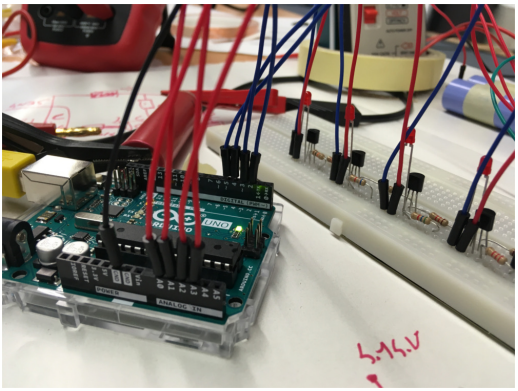
VA0 = 1.2*(4.14-0.6) / (2.2+1.2) = 1.25V

Schéma du circuit de mesure de tension pour 4 cellules.



R1, R3, R5 et R7 : 1,2kOhm
R2 : 2.2kOhms
R4 : 5.6kOhms
R6 : 6.8 kOhms
R8 : 10kOhms





Code Arduino :

```
#include <Arduino.h>
// the setup function runs once when you press reset or power the
board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
     Serial.begin(9600);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);


  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
}

// the loop function runs over and over again forever
void loop() {

  digitalWrite(2, HIGH);   // turn the LED on (HIGH is the voltage
level)
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a
voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you read:
  Serial.println(voltage);
  delay(2000);

  digitalWrite(2, LOW);   // turn the LED on (HIGH is the voltage
level)
  delay(1000);
  int sensorValue2 = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a
```

```
voltage (0 - 5V):
  float voltage2 = sensorValue2 * (5.0 / 1023.0);
  // print out the value you read:
  Serial.println(voltage2);
}
```

Fin du code

Ce module n'a pas été entièrement abouti par manque de temps, il est possible que les résultats obtenus soient erronés.

42. EQUILIBRAGE DES CELLULES

PROBLÉMATIQUE :

Lorsque les cellules ont des différences de tension, par exemple :

Cellule 1 : 3.7V

Cellule 2 : 3.7V

Cellule 3 : 3.4V <= Cellule déséquilibrée

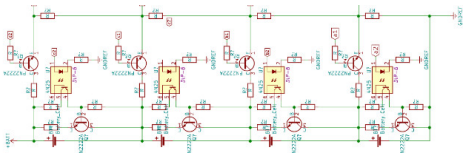
Cellule 4 : 3.7V

Il faut alors rééquilibrer les cellules (que les tensions soient toutes égales).

Sachant que le chargeur n'est capable de recharger que l'ensemble des cellules, il faut alors décharger les 3 autres cellules pour qu'elles aient une tension de 3.4V.

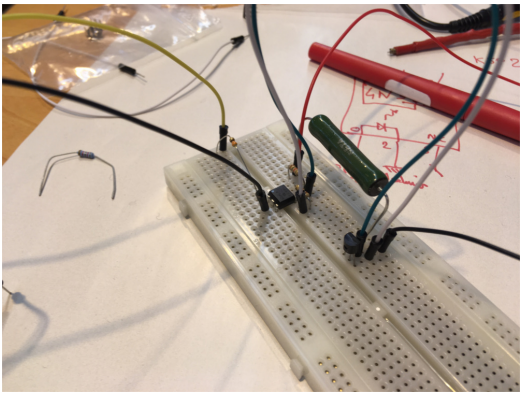
Décharge

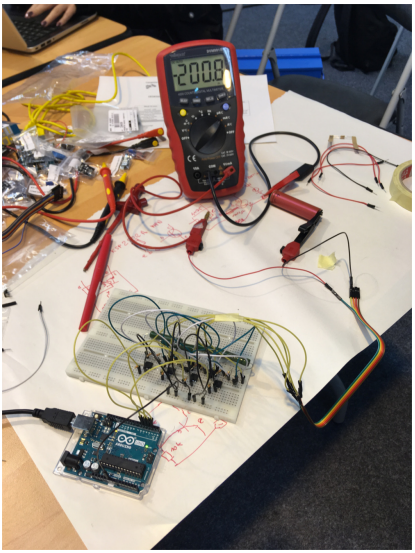
Schéma de montage :



Ce schéma de montage comprend la décharge pour l'équilibrage mais également le schéma du montage des mesures de tension.

Ce montage est constitué d'un optocoupleur 4n35, d'une résistance 330Ohms, de 2 résistances 10kOhms et d'une résistance bobinée de 8W.
L'optocoupleur permet d'isoler le circuit à une cellule afin d'avoir le courant. Le principe est que lorsqu'un courant passe dans la diode de l'optocoupleur, la diode brille en émettant une lumière infrarouge alors le courant peut traverser le transistor. Le transistor est alors considéré comme un interrupteur fermé.





La cellule se décharge de 200mA en quelques secondes.

Le code Arduino est tout simple, il décharge sucéssivement les 4 cellules pendant quelques secondes (lorsque la led est allumée).

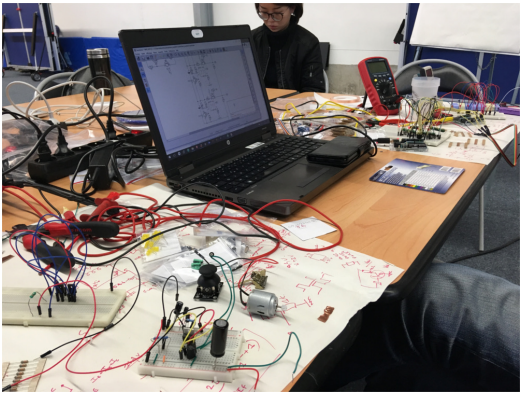
```
// the setup function runs once when you press reset or power the
board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);

}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(5, HIGH);   // turn the LED on (HIGH is the voltage
level)
  delay(1000);
  digitalWrite(4, HIGH);             // wait for a second
  delay(1000);
  digitalWrite(3, HIGH);             // wait for a second
  delay(1000);
  digitalWrite(2, HIGH);             // wait for a second
  delay(1000);

  digitalWrite(5, LOW);    // turn the LED off by making the voltage
LOW
  delay(1000);                       // wait for a second
  digitalWrite(4, LOW);    // turn the LED off by making the voltage
LOW
  delay(1000);                       // wait for a second
  digitalWrite(3, LOW);    // turn the LED off by making the voltage
LOW
  delay(1000);                       // wait for a second
```

```
  digitalWrite(2, LOW);    // turn the LED off by making the voltage
LOW
  delay(1000);                       // wait for a second

}
```

Fin du code

43. CONCLUSION

Plusieurs modules ont été réalisés au cours de ce prototypage qui, en étant assemblés, pourraient être une base pour la réalisation d'un BMS open source.



D'autres modules non réalisés ici sont nécessaire pour la sécurité de la batterie, tels que le capteur de température qui assure la température de la batterie afin que cette dernière ne surchauffe ou un interrupteur On/Off sur le chargeur pour décider si la batterie doit être chargée ou non.

ROBOTIQUE OPENSOURCE

44. PRÉSENTATION DE L'ATELIER ROBOTIQUE OPENSOURCE
45. INTRODUCTION À LA ROBOTIQUE
46. INITIATION À LA MODÉLISATION AVEC FREECAD

44. PRÉSENTATION DE L'ATELIER ROBOTIQUE OPENSOURCE

LES PARTICIPANTS

- Gilles : animateur de l'atelier, développeur linux embarqué chez "Savoir Faire Linux" et roboticien amateur
- François, électronicien qui se forme au numérique
- Christian, étudiant en master
- Laurine, Adrien et Nicolas : étudiant en économie et stratégie digitale

DÉROULÉ DE L'ATELIER

MARDI 13 MARS

présentation des participants et de la robotique opensource.

Prise en main de freecad et modélisation d'une coque de téléphone.

MERCREDI 14 MARS

Quelques essais avec le robot et impression 3d

JEUDI 15 MARS

Initiation à Riot-os et réglage du robot

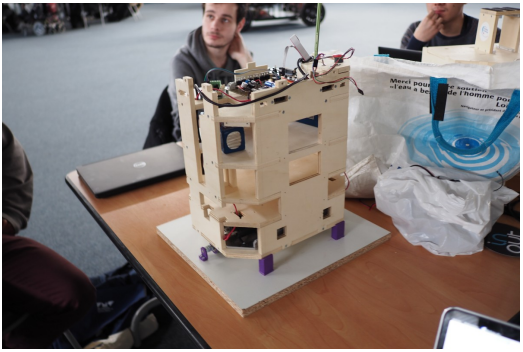
45. INTRODUCTION À LA ROBOTIQUE

Un robot est une machine qui fonctionne en autonomie avec ou sans opérateur. Il n'est pas forcément doué d'intelligence artificielle, mais suit une machine d'état.

Un robot à trois composantes principales : mécanique, éléctronique et logicielle.

LA COMPOSANTE MÉCANIQUE

La partie mécanique peut être réalisée de nombreuses manières différentes, notamment en terme de matériaux. La base mécanique de "Cortex", le robot de démo amené par Gilles, a une base en bois conçu avec inkscape et usiné à l'aide d'une CNC. Seule la partie support de la motorisation est réalisé en aluminium.

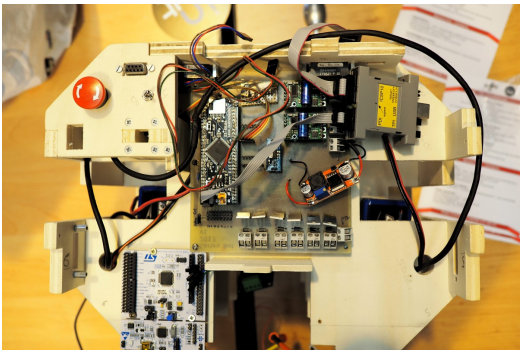


Le mode de déplacement varie également en fonction de différentes contraintes, notamment le fait que le sol soit parfaitement plat. Dans le cas de cortex, qui est conçu pour participer à la coupe de france de robotique, le sol est effectivement plat et 2 roues suffisent à orienter et déplacer le robot.

LA COMPOSANTE ÉLECTRONIQUE

Le "cerveau" du robot est réalisé à l'aide d'un microcontrôleur, généralement 8bits, comme celui utilisé par les cartes arduino.

Le microcontrôleur est lié à des capteurs (distance, ...) et à des effecteurs par l'intermédiaire d'une étage de puissance (mosFET, ponts en H).



Tout cela est alimenté par plusieurs régulateurs de tension (5V, 3.3V).

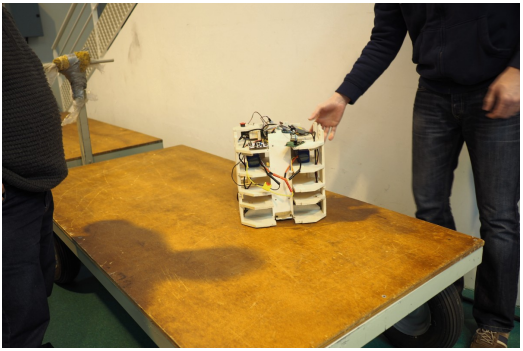
LA COMPOSANTE LOGICIELLE

On privilégie des solutions légères sans système d'exploitation complet.

Gilles et son association de robotique se sont mis à utiliser RiotOS, un système d'exploitation léger dédié à l'internet des objets.

DES ESSAIS AVEC CORTEX

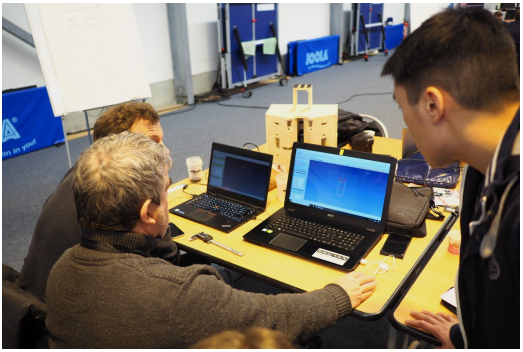
Les participants ont pu faire des essais avec Cortex, le robot que Gilles à amené pour l'atelier.



Après quelques test mercredi, la journée du jeudi a été consacrée au réglage de certains paramètres de l'asservissement des moteurs afin d'adapter leur comportement au poids du robot.

46. INITIATION À LA MODÉLISATION AVEC FREECAD

L'atelier a été l'occasion pour les participants de l'initier à la modélisation avec FreeCAD, un logiciel libre de conception assistée par ordinateur (CAO)



Les participants on choisi de modéliser une coque de téléphone, qui a pu être imprimée avec l'imprimante 3d mise à disposition.

ANNEXES

47. RESSOURCES GÉNÉRALES

47. RESSOURCES GÉNÉRALES

Vous trouverez éparpillé un peu partout sur les projets des intervenants des hackathons des deux années précédentes, un peu de documentation :

- Site principal : https://camposv-labfab-ur1.ietr.fr/
- Un wiki recense des informations sur les éditions précédentes de campOSV : http://wiki.lafabriquedesmobilites.fr/wiki/VehiculeLibre
- Collection de vidéos de campOSV 2016 : https://www.youtube.com/playlist?list=PLoCPQEftaSqPeKA3sgqDnLhOpOCe7AybE
- Le chassis Tabby EVO : https://www.openmotors.co/
- https://elinux.org/OSVehicle
- http://wiki.lafabriquedesmobilites.fr/wiki/VehiculeLibre