



UBECOME
INNOVATE TOGETHER



ROBOTS MOBILES ET MODULAIRES
SOLUTIONS PROFESSIONNELLES

COMPTE RENDU

Ben ammar Rouaa

I. Introduction

1. Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de ma mission et qui m'ont aidé lors de la rédaction de ce rapport

Je désire remercier **Mr. OUAJDI KORBAA**, mon maître de stage, qui m'a fourni les outils nécessaires au bon déroulement de mon stage ainsi pour son encouragement et sa confiance.

De plus, je tiens à remercier **Mr. FAYCEL BAAZIZ** d'avoir nous donner l'occasion à faire notre stage au sein de la société UBECOME.

Je remercie également **Mr. TRIKI Bayrem**, pour ses conseils qu'ils ont pu me prodiguer au cours de ces deux mois.

Je remercie **JeanLuc MACHANGAMA** pour sa confiance et les connaissances qu'il a su partager avec nous. Je le remercie aussi pour sa disponibilité et la qualité de son encadrement.

2. Sommaire

I.	Introduction	1
1.	Remerciements	1
2.	Sommaire.....	1
3.	Tableau des figures	2
II.	Cadre générale.....	2
1.	Introduction	2
2.	Etude technologique	2
a.	Environnement de travail de programmation :	2
b.	Les technologies utilisées.....	2
III.	Réalisation	3
❖	Challenge 1.....	3
❖	Challenge2 :.....	5
IV.	Conclusion :.....	9

3. Tableau des figures

Figure 1:Creation des topics	4
Figure 2:Ceration de l'api web.....	4
Figure 3:Recuperation de la position et de l'etat du turtlesim	5
Figure 4:Recuperation de l'etat de données capteurs	5
Figure 5:Integré l'image du map dans le fichier YAML	6
Figure 6:Publication de la carte dans /map	6
Figure 7:Les positions de turtlesim dans le simulateur.....	7
Figure 8:Récuperation des positions	7
Figure 9:upload file	8
Figure 10:Les fichiers dans Mlab	8

II. Cadre générale

1. Introduction

Ce rapport présente le travail que j'ai effectué pendant les mois de février et mars. Les tâches réalisées s'est avéré très intéressante et très enrichissante. Le but de ce rapport est de faire une présentation exhaustive de ces tâches.

2. Etude technologique

a. Environnement de travail de programmation :

- ✚ Linux
- ✚ Programmation sous ROS version indigo UBUNTU 14
- ✚ Programmation sous ROS version melodic UBUNTU 18

b. Les technologies utilisées

- ✚ MongoDB (de l'anglais humongous qui peut être traduit par « énorme ») est un système de gestion de base de données orientée documents, répartitionnable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données.
- ✚ MongoLab : mLab est un service de base de données cloud entièrement géré qui héberge des bases de données MongoDB
- ✚ HTML5 est la dernière version du langage de développement web HTML.

- ✚ JavaScript : C'est un langage de programmation de scripts principalement utilisé côté serveur.
- ✚ JSON (JavaScript Object Notation) : C'est un format léger d'échange de données. Il s'agit d'un format texte complètement indépendant de tout langage.
- ✚ Node.js : est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge
- ✚ Express.js Express est une infrastructure web middleware et de routage, qui a des fonctions propres minimales : une application Express n'est ni plus ni moins qu'une succession d'appels de fonctions middleware.
- ✚ Service Web REST (Representational State Transfer) : C'est un service Web basé sur le style d'architecture REST. En effet, un service web est un programme informatique permettant la communication et l'échange de données entre les applications et les systèmes hétérogènes dans des environnements distribués.
- ✚ Mongoose: est une bibliothèque de modélisation pour MongoDB et nodejs.
- ✚ EJS: Nous utiliserons ejs (templates javascript intégrés) comme moteur de visualisation de notre application, ce qui nous permettra de générer un balisage HTML avec du javascript simple.
- ✚ GridFsStorage: Ceci sera utilisé pour créer un moteur de stockage.

III. Réalisation

Après avoir pris une connaissance sur ROS j'ai commencé la réalisation des tâches demandées sous forme de challenge :

❖ Challenge 1

- créer un topic ROS à partir d'un message de position fourni par ROS
- créer un topic ROS d'état de fonctionnement fourni par ROS
- créer un topic ROS d'état de donnée capteurs fourni par ROS
- créer un code qui modifie en live l'état du topic

Monter une API qui trigge selon les différents états de chaque topic,

Créer une interaction entre les topics et l'interface web (changement de valeur de chaque topic).

- **Création des topics**

```
rouaa@ubuntu:~$ cd catkin_ws/
rouaa@ubuntu:~/catkin_ws$ cd src/
rouaa@ubuntu:~/catkin_ws/src$ cd pose_message/
rouaa@ubuntu:~/catkin_ws/src/pose_message$ cd src/
rouaa@ubuntu:~/catkin_ws/src/pose_message/src$ python move.py
Set your x goal: 1
Set your y goal: 1
Set your tolerance: 0.2
^C2019-04-07T15:07:48.173-0700 [.....] mydatabase.mycollecti
on 0B/40B (0.0%)
2019-04-07T15:07:48.357-0700 connected to: Cluster0-shard-0/cluster0-shard-00
-00-iyxed.mongodb.net:27017,cluster0-shard-00-01-iyxed.mongodb.net:27017,cluster
0-shard-00-02-iyxed.mongodb.net:27017
2019-04-07T15:07:51.171-0700 [#####] mydatabase.mycollecti
on 40B/40B (100.0%)
2019-04-07T15:07:51.430-0700 [#####] mydatabase.mycollecti
on 40B/40B (100.0%)
2019-04-07T15:07:51.430-0700 imported 1 document
rouaa@ubuntu:~/catkin_ws/src/pose_message/src$ █
```

Figure 1:Creation des topics

- **Création de l'api web**

J'ai donc créé un service Web via Mongodb Stitch qui renvoie la dernière valeur dans la base de données

```
// Try running in the console below.

exports = function(payload) {
  const mongodb = context.services.get("mongodb-atlas");
  const mycollection = mongodb.db("mydatabase").collection("mycollection");

  var array = mycollection.find({}).toArray();
  return array ;
};
```

Figure 2:Ceration de l'api web

- **Récupération des valeurs dans la page web**

Les valeurs données par l'utilisateur seront affichées dans la page Web

J'ai fait quelques tests sur les positions des turtlesim afin de vérifier son état (en mouvement ou en pause)

Si le test est vrai: un bouton rouge s'affichera dans la page Web pour indiquer que le tortue est dans la même position

Si le test est faux: un bouton vert apparaîtra dans la page Web pendant un temps limité (2,5 secondes) indiquant que le turtlesim est en mouvement



Figure 3:Recuperation de la position et de l'état du turtlesim

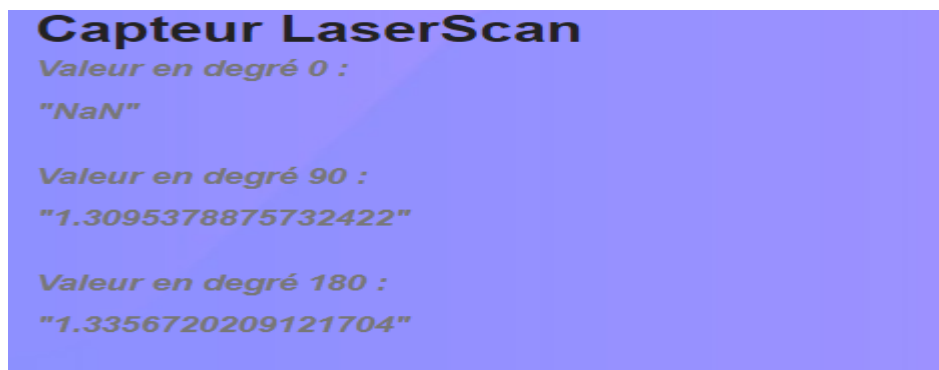


Figure 4:Recuperation de l'état de données capteurs

❖ Challenge2 :

- créer un Noeud ROS qui publie la carte du turtle sim sur /map
- créer l'api web qui récupere cette carte
- créer l'api web qui récupere les données de velocity et position du turtle
sim cmd_vel (geometry_msgs/Twist) pose (turtlesim/Pose)
- Cloner l'affichage du turtle sim sur l'api WEB
- l'api saura sauvegarder les données en local de carte et de position en local.
- l'api saura lire des données de la dernière map et de la dernière position sauvegardées localement,
- **Noeud ROS qui publie la carte du turtle sim sur /map**

```
image: /home/rouaa/catkin_ws/src/carte/map/map.pmg
resolution: 0.050000
origin: [0.000000, 0.000000, 0.000000]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.196
```

Figure 5: Intégré l'image du map dans le fichier YAML

```

**
* /rostdistro: melodic
* /rosversion: 1.14.3
**

NODES
  map_server (map_server/map_server)
  turtlesim_node (turtlesim/turtlesim_node)

auto-starting new master
process[master]: started with pid [2731]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 316b9e6c-4d48-11e9-9d92-000c29b2ab16
process[rosout-1]: started with pid [2742]
started core service [/rosout]
process[turtlesim_node-2]: started with pid [2745]
process[map_server-3]: started with pid [2746]

```

Figure 6:Publication de la carte dans /map

- la techno de ROS n'est pas assez mature pour aller dans ce sens. les données ros sont tous BRUT.
- **Solution proposée**

J'ai crée un Publisher `pubTsim` qui va utilise le topic `/turtle1/cmd_vel`

j'ai crée un subscriber file topic turtle/pose

J'ai crée un api web pour récupérer les positions et les afficher dans l'interface web

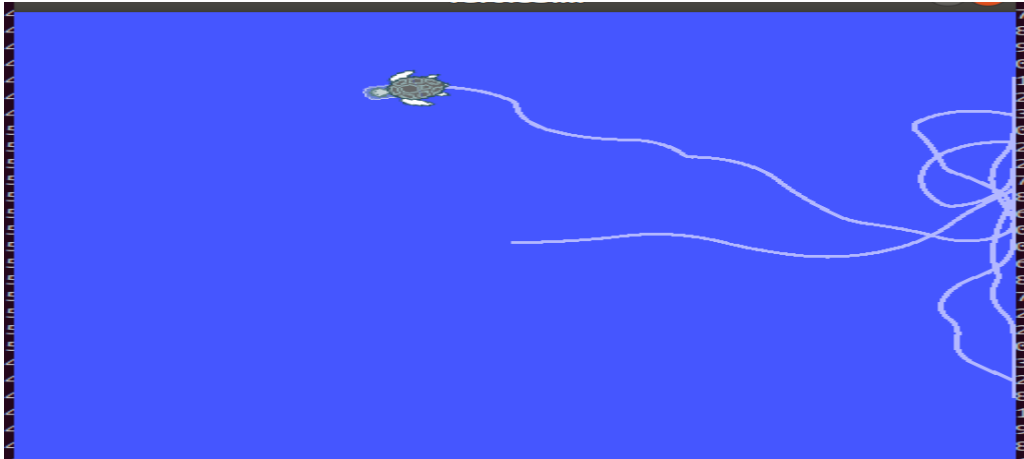


Figure 7: Les positions de turtlesim dans le simulateur



Figure 8: Récupération des positions

- Connexion et/ou la récupération à distance de fichiers sur un autre ordinateur à partir d'une page WEB :

J'ai utilisé :

- Un fichier HTML pour saisir le fichier et le visualiser.
- Un gestionnaire de soumission pour envoyer les fichiers au serveur

- Express pour recevoir le fichier téléchargé et le renvoyer au client.

Mongo File Uploads

4d2be42fc0426879c6862bb1fbc8d732.md

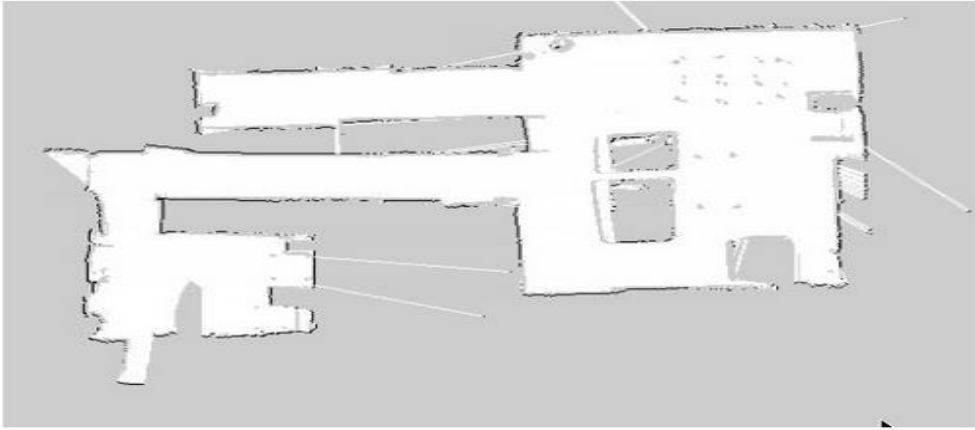


Figure 9:upload file

```
records / page 10 ▾
{
  "uploadDate": {
    "$date": "2019-04-04T08:33:34.889Z"
  },
  "md5": "37e54e0d9fd1de4c63cd8ee0590b4e0d",
  "filename": "4d2be42fc0426879c6862bb1fbc8d732.md",
  "contentType": "text/markdown"
}
{
  "uploadDate": {
    "$date": "2019-04-04T08:55:52.413Z"
  },
  "md5": "fc9cee28ccdf0e2955ce542c4be16d3b",
  "filename": "d17c7b35c1e63e14fe4d8556f83b2457.jpeg",
  "contentType": "image/jpeg"
}
```

Figure 10:Les fichiers dans Mlab

IV. Conclusion :

Je souhaiterais remercier l'ensemble des intervenants pour le déroulement de la mission dans les Bonnes conditions.

Les apports que j'ai tiré de cette expérience professionnelle peuvent être regroupés autour de trois idées principales : les compétences acquises, les difficultés rencontrés et solutions apportées.