LATRACE Romain 3A IT TDA – TP2

DUTERTE Basile



Projet WIFI BOT



Compte rendu

Année 2017-2018

**Professeurs**

M.HEYRMAN

M. MARROQUIN

Table des matières

I. Présentation du contexte 3

I. A- Le projet 3

I. B- Le wifibot 3

I. C- Les outils mis à disposition 4

II. Le planning 5

III. Prise en main du wifibot 6

III. A- La connexion 6

III. B- La caméra 6

III. C- Communication avec le robot 7

III. D- Les informations envoyées par le robot 8

III. E- L’interface graphique 9

IV. CONCLUSION 10

I. Présentation du contexte

# I. A- Le projet

Le projet wifibot est un module de notre formation de troisième année d’ingénieur, il consiste à travailler par binôme afin de réaliser un programme avec une interface graphique pour commander un robot en wifi.

Ce projet était à réaliser en sept séances de TP de quatre heures, et ce de manière totalement autonome, grâce aux diverses documentations fournies par le fabricant et disponible sur le site internet du wifibot : <http://www.wifibot.com>.

Plusieurs objectifs ont été fixés afin de mener à bien le projet :

* Utilisation de la bibliothèque C++ Qt
* Pouvoir diriger le robot via une connexion wifi
* Contrôler la caméra

L’atout principal de ce projet est aussi de nous initier au travail en équipe, à la gestion d’un projet et d’apprendre à gérer toutes les difficultés liées à un groupe de travail.

# I. B- Le wifibot

Le wifibot est un robot « tout-terrain », ses quatre roues motrices lui permettent de se déplacer sur différents types de surface. Plusieurs capteurs sont présents sur ce petit robot, permettant de connaitre la vitesse de rotation de chaque roue, la distance avec un obstacle…

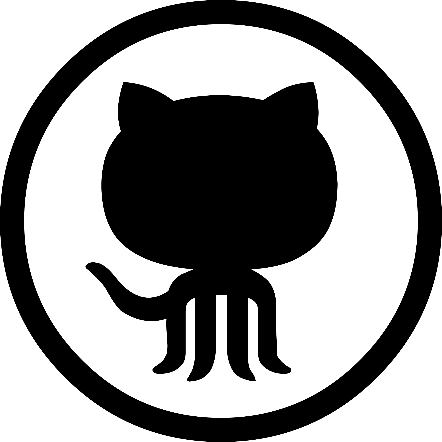
Il dispose aussi d’une webcam disposant de son propre serveur web, permettant de récupérer les images depuis le point de vue du robot.

# I. C- Les outils mis à disposition

Au cours de ce projet nous avons pu utiliser différents outils de programmation, mais aussi de gestion de projet.



Qt Creator est un des outils de Qt(bibliothèque C++ très connue), il s’agit d’un environnement de développement étudié pour faciliter le développement avec Qt, disposant même de QtDesigner, un outil permettant de réaliser des interfaces graphiques avec Qt.

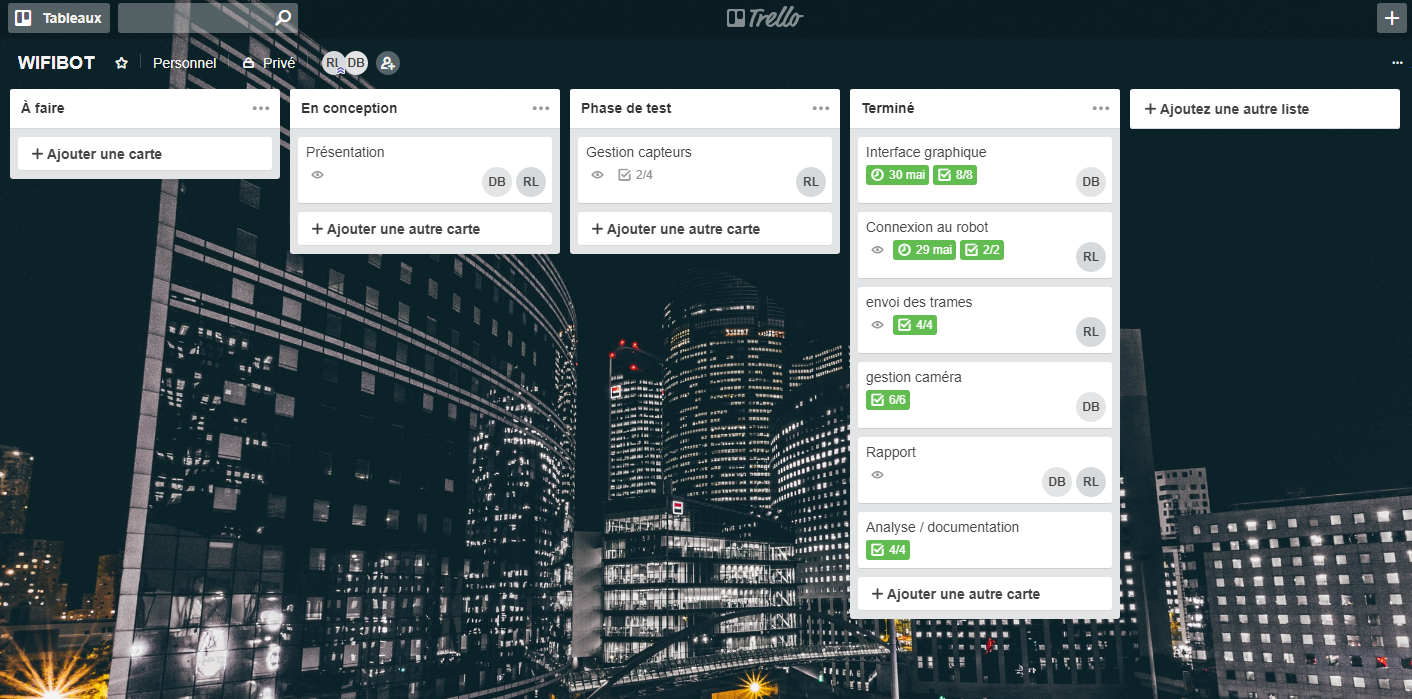


GitHub est un logiciel permettant le travail en équipe grâce au versioning. En réalisant plusieurs « branches » pour un projet, plusieurs personnes peuvent travailler sur les mêmes fichiers en même temps. Lors du déploiement, il nous rappelle les modifications que nous avons apportées, et compare avec celles des autres utilisateurs, afin d’éviter des conflits.

II. Le planning

Nous ne sommes que deux à travailler sur ce projet, mais nous avions aussi une contrainte de temps, en nombre de séances de TP. Afin de nous répartir correctement le travail à effectuer sur le robot, nous avons découpé le projet en plusieurs tâches principales, elles-mêmes découpées en sous-tâches.

Pour planifier ce projet nous avons opté pour faire un TRELLO pour avoir un bon suivi et surtout pour nous situer dans l’avancement du projet. Grâce à cet outil nous pouvons créer différentes tâches et associer telle personne à telle tâche.



* Dans la section **A faire**, toutes les tâches s’y trouvaient au tout début du projet.
* La colonne **En conception** doit contenir les tâche que nous sommes en train d’implémenter.
* L’emplacement **Phase de test** accueil les tâches qui sont normalement abouties mais qui nécessite d’être testées afin de les valider.
* Une fois la tâche terminée, nous la déplaçons dans **Terminé.**

Nous avons plus ou moins tenus nos engagements quant à la durée de chaque tâche. Puisque nous travaillions à deux, nous avons pu bien séparer la mise en place de l’interface graphique et le développement interne au logiciel. Les décalages sur la planification, notamment pour l’implémentation de fonctionnalités, sont dus à la non-disponibilité du robot (souvent occupé par un autre binôme) ne permettant pas de tester les nouveautés.

III. Prise en main du wifibot

# III. A- La connexion

Après lecture de la documentation, nous avons pu déterminer qu’il existait différentes méthodes de connexions, principalement liées aux technologies des réseaux, et plus précisément du protocole Ethernet. L’adresse IP du robot étant 192.168.1.106.

Nous avons choisi la méthode de connexion recommandée, le mode connecté TCP. Ce mode de connexion est relativement efficace, car une fois connecté au robot, nous pouvons garder cette connexion pour envoyer et recevoir des trames à des intervalles réguliers.

Pour communiquer avec ce mode, nous devrons utiliser des sockets TCP, et d’après la documentation, il faudra utiliser le port 15020.

# III. B- La caméra

La gestion de la caméra peut être divisée en deux parties :

* Les mouvements de la caméra
* Récupération du flux vidéo

Pour contrôler la caméra, nous avons utilisé des liens http en envoyant des requêtes avec l’objet QNetworkAccessManager.

Nous avons obtenus le retour caméra à l’aide de QWebEngineView en chargeant l’URL du serveur http sur le port 8080.

# III. C- Communication avec le robot

Une chose importante à savoir, c’est que pour communiquer au robot des directives de déplacement, ou changer son comportement, nous avons besoin d’envoyer une trame à un intervalle relativement rapide, pour éviter d’être déconnecté rapidement. Pour ce faire, nous bombardons le wifibot avec une trame vide (sans directives) afin de garder la main sur le robot.

D’après la documentation, chaque trame devra suivre un schéma précis, et comporter 9 octets :

1. Début de trame : valeur 255
2. La taille de la trame, qui sera toujours de 7
3. Premier octet pour la vitesse des roues gauches
4. Deuxième octet des roues gauches
5. Premier octet pour la vitesse des roues droites
6. Deuxième octet des roues droites
7. Le 7ème octet est un peu plus compliqué:
   1. Bit de capteurs (0)
   2. Bit du relai 2 (0)
   3. Bit du relai 3 (0)
   4. Bit de la période de rotation des roues ou PID : 50 ms ou 10 ms. (0)
   5. Sens de rotation des roues droites : marche avant (1) ou marche arrière (0)
   6. Gestion et contrôle de la rotation de chaque roue droite (1)
   7. Sens de rotation des roues gauches : marche avant (1) ou marche arrière (0)
   8. Gestion et contrôle de chaque roue gauche (1)
8. Premier octet du CRC : code de vérification permettant de valider que la trame ait été reçue de manière intacte
9. Deuxième octet du CRC

Nous avons utilisé un outil de Qt appelé « QBytesArray ». Ce dernier permet de créer un tableau d’octet, il est compatible avec l’outil « QTCPSocket » pour envoyer une trame directement. On peut ainsi ajouter un à un les octets au tableau après avoir calculé leur valeur, et une fois que cela est terminé, utiliser la commande « write » pour envoyer la requête.

# III. D- Les informations envoyées par le robot

Il est possible de récolter plusieurs informations sur le wifivot comme le niveau de la batterie et notamment les donnée des capteurs. Il nous suffit d’attendre une trame sur le port 15020, le robot les envoie automatiquement.

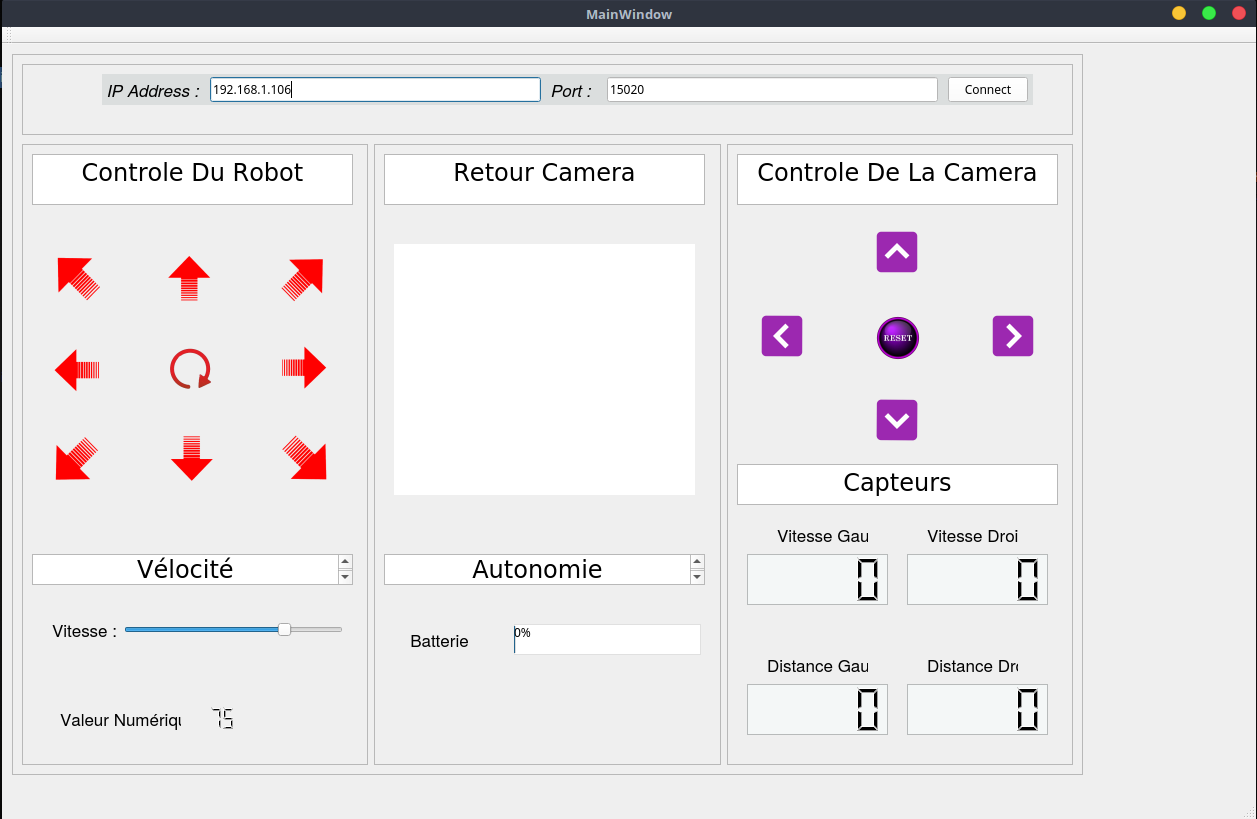
L’ensemble des informations lié aux capteurs tient sur 21 octets :

1. La vitesse des roues gauches en tics/sec : 2 octets
2. Le niveau de la batterie, en 10-1 Volts
3. Capteurs adc4
4. Adc3
5. Odométrie gauche : 4 octets
6. Vitesse des roues droites : 2 octets
7. Adc0
8. Adc1
9. Odométrie droite : 4 octets
10. Consommation instantanée du robot (en Ampère)
11. Version du firmware
12. CRC : 2 octets

# III. E- L’interface graphique

L’interface a été développée de façon à être simple d’utilisation pour qu’elle soit accessible à tous. Celle-ci se décompose en quatre parties :

1



3

2

4

1

Se connecter au robot

2

Pilotage du wifibot. Nous avons également utilisé les touches du clavier car nous trouvions cette option plus intuitive. Sous les boutons de contrôle de mouvement, il y a un slider pour gère la vitesse du robot.

3

Contrôler les mouvements de la caméra et récupérer le flux vidéo

4

Informations sur le robot. La barre de la batterie devrait changer de couleur en fonction de l’état de charge du wifibot.

IV. CONCLUSION

Durant ce 2ème semestre nous avons donc eu l’occasion de réaliser un projet complet. Nous avons commencé par organiser notre projet, afin de structurer notre temps. Nous nous sommes ensuite partagés le travail technique afin que chacun apporte ses connaissances et ses idées.

Lors de ce projet, nous avons pu concrétiser nos connaissances théoriques et les appliquer par la pratique. Nous nous sommes rendus compte, qu’il y avait un décalage entre la théorie et la pratique et cela nous a permis d’obtenir une expérience que l’on pourra mettre à profil dans notre vie professionnelle et peut-être dans un futur projet.