

# Rapport IoT Projet : Robot mobile Connecté

## Table des matières

<a href="#">Table des matières</a> .....	2
<a href="#">Description</a> .....	3
<a href="#">Explication</a> .....	3
<a href="#">Python - Détecter la main</a> .....	3
<a href="#">LabVIEW - Intermédiaire</a> .....	3
<a href="#">Robot - Récepteur</a> .....	4
<a href="#">Présentation du LEGO</a> .....	4
<a href="#">Construction pour ce projet</a> .....	4
<a href="#">Utilisation</a> .....	4
<a href="#">Construction du robot</a> .....	5
<a href="#">Python</a> .....	5
<a href="#">LabVIEW</a> .....	5
<a href="#">Conclusion</a> .....	6
<a href="#">Illustration</a> .....	6

## Description

Le projet IoT « Robot mobile Connecté » a pour but de créer un robot connecté en Bluetooth ou wifi qui a la capacité de se déplacer de manière basique et qui détecte les murs devant et derrière lui. Les déplacements que le robot doit effectuer sont indiqués grâce au mouvement de la main de l'utilisateur, qui est détecté grâce à une caméra. Dans ce cas-ci, on utilise comme base pour le robot le Lego Mindstorms EV3. Pour la détection de la main via la caméra, on utilise un programme python et pour faire la liaison entre le robot et les mouvements de la main, on utilise le logiciel LabVIEW.

## Explication

Le fonctionnement des différentes parties de ce projet (le programme LabVIEW, le programme Python et le robot lego Mindstorms EV3) les unes avec les autres se fait comme ceci :

Tout d'abord, le programme python détecte la main de l'utilisateur, s'il y en a une, et stocke l'état de la main dans un fichier appelé « data.txt ».

Ensuite, le programme LabVIEW va lire la nouvelle donnée stockée dans le fichier « data.txt » et déterminer le mouvement que le robot doit faire.

Une fois que le programme LabVIEW a fini cela, il enverra au robot les indications appropriées grâce à une communication Bluetooth. Il ne restera plus au robot qu'à bouger les moteurs indiqués.

## Python - Détecter la main

Le programme python « HandTracking61305 » est là pour capter et sauvegarder les informations captées grâce à la caméra par rapport à une main.

Pour plus d'informations, veuillez-vous référer directement au code et au fichier README qui l'accompagne.

## LabVIEW - Intermédiaire

Pour réaliser ce projet avec le programme LabVIEW, il faut créer un projet spécifique dit 'Project Robot'<sup>1</sup>.

---

<sup>1</sup> cf. Illustration A.

Pour établir la connexion entre LabVIEW et l'EV3, appuyez sur le bouton 'No Target' puis 'Find NXT/EV3...'. N'oubliez pas d'activer votre Bluetooth et le Bluetooth de l'EV3.

Ensuite, il faut spécifier au programme quelle sortie est utilisée et par quoi elle est occupée. Pour cela, appuyez sur le bouton 'Open Schematic Editor'. De là, vous pouvez spécifier quelle entrée va servir à quoi et pour quoi. Vous pouvez aussi tester les différents moteurs et capteurs<sup>2</sup>.

Le programme LabVIEW, une fois lancé, va aller lire les données qui ont été stockées par le programme python. Il va lire les fichiers 'data.txt' et 'status.txt'. LabVIEW n'écrit que dans le fichier 'status.txt' quand une nouvelle donnée de 'data.txt' est différente de celle qui est dans 'status.txt'.

Après cela, le programme va prendre l'information qui se trouve dans le fichier 'status.txt' et la traduire pour envoyer la bonne instruction à l'EV3.

## Robot - Récepteur

### Présentation du LEGO

Le LEGO MINDSTORMS EV3<sup>3</sup> est un lego qui permet de construire divers robots, il existe plus de 5 formes officielles et encore bien plus inventées par la communauté.

Ce qui fait la différence avec les autres lego, ce sont ses moteurs et capteurs mais surtout pour l'EV3. La pièce centrale est une brique composée d'un processeur ARM9, un port USB pour une connexion wifi, un lecteur de carte Micro SD. Il y a aussi cinq boutons, un écran, quatre ports pour moteurs et quatre ports pour capteurs.

### Construction pour ce projet

Pour ce projet, l'inspiration vient du robot Wall-E, pour atteindre cette similitude la base de la structure vient du modèle KRAZ3 de LEGO<sup>4</sup>. Il y a eu des modifications simplement pour se rapprocher de l'inspiration mais aussi pour éviter des constructions inutiles du modèle KRAZ3.

## Utilisation

Cette partie est dédiée à l'explication étape par étape de la mise en place et de l'utilisation du projet tel qu'il est. Cela se déroule en trois grandes parties.

---

<sup>2</sup> cf. Illustration B

<sup>3</sup> <https://www.lego.com/fr-be/product/lego-mindstorms-ev3-31313>

<sup>4</sup> [https://www.lego.com/cdn/product-assets/product.bi.additional.extra.pdf/31313\\_X\\_KRAZ3.pdf](https://www.lego.com/cdn/product-assets/product.bi.additional.extra.pdf/31313_X_KRAZ3.pdf)

Premièrement, la construction du robot Mindestorms EV3, puis le lancement du programme de détection de main et pour finir le programme qui fait l'intermédiaire entre le robot et le programme python.

## Construction du robot

La structure du robot en lego importe peu, il faut simplement respecter les moteurs et capteurs. Dans notre cas, le moteur de la roue droite se trouve sur le port C et le moteur de la roue gauche sur le port B<sup>5</sup>. Le capteur à infrarouge se trouve sur le port numéro un et le capteur avec un bouton poussoir se trouve sur le port numéro deux.

Attention, si vous changez le placement des ports pour les moteurs et les capteurs dans le logiciel LabVIEW<sup>6</sup>, vous devez aussi l'adapter dans la vraie vie.

Une fois le robot construit et si le câblage est similaire dans le logiciel et sur le robot, cette partie est achevée.

## Python

Ouvrez votre environnement de développement python, quel qu'il soit (PyCharm, IDLE, etc...) et sélectionnez le programme « HandTracking61305 ». Favorisez l'utilisation de python3.8 pour être sûr du bon déroulement du programme<sup>7</sup>.

Normalement, une fois les instructions précédentes respectées, il vous suffit de run le programme et votre caméra devrait s'ouvrir en détectant une main. Vous pouvez aussi vérifier si, une fois le programme quitté, les données ont bien été enregistrées dans le fichier « data.txt ».

## LabVIEW

Une fois l'installation et la configuration de LabVIEW<sup>8</sup> effectuées il vous suffit de lancer le logiciel et de sélectionner le fichier « lotprolectg61305. lvrbt ». Vous arrivez sur la fenêtre qui vous permet de créer la connexion Bluetooth entre votre appareil et le robot EV3. C'est aussi sur cette fenêtre que vous devez indiquer au logiciel quel moteur et quel capteur est connecté à quelles entrées grâce au bouton 'Open Schematic Editor '<sup>9</sup>. Il vous reste à placer les chemins absolus qui mènent au fichier « data.txt » et « status.txt » aux emplacements indiqués dans le programme LabVIEW.

---

<sup>5</sup> La notion de droite et gauche est par rapport au robot.

<sup>6</sup> cf. Illustration B.

<sup>7</sup> N'hésitez pas à consulter le fichier ReadMe pour plus de précisions.

<sup>8</sup> Choisissez la versions 15.0 (32 bits).

<sup>9</sup> cf. Illustration B

## Conclusion

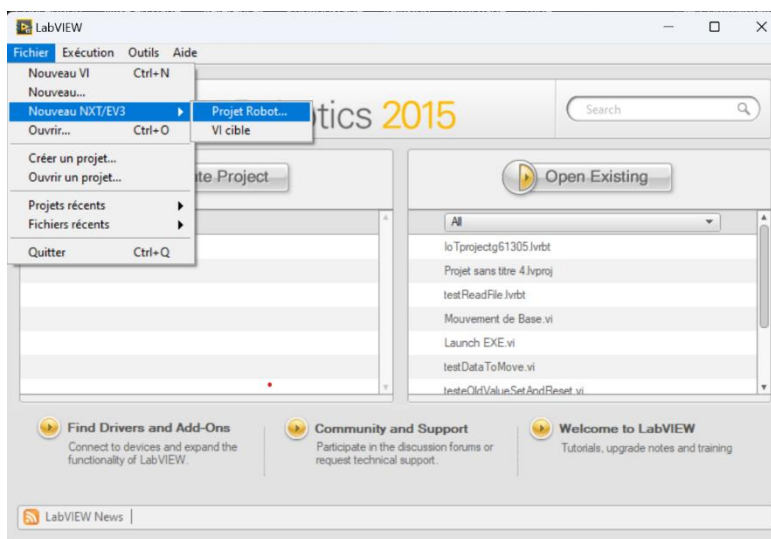
En définitive, ce projet a démontré que la communication entre un robot LEGO Mindestorms EV3 et les mouvements d'une main était possible grâce à l'utilisation d'un programme python et un programme LabVIEW. Ce qui montre la facilité d'évolution de ce type d'interaction avec les objets connectés.

Dans ce cas assez basique, il n'y a pas vraiment d'utilité de cette communication. Pourtant, si on applique une logique similaire dans d'autres domaines (la santé, les usines, etc.), cela pourrait avoir un impact non négligeable.

Cependant, il y a encore des domaines à améliorer, notamment en termes de réactivité et de robustesse du système face à des variations de luminosité et de conditions d'éclairage. De plus, l'ajout de fonctionnalités supplémentaires telles que la reconnaissance de gestes plus complexes pourrait rendre le système encore plus polyvalent.

## Illustration

A) Créer le bon type de projet :



B) Menu permettant de faire la connexion et spécifier les entrées et sorties :

