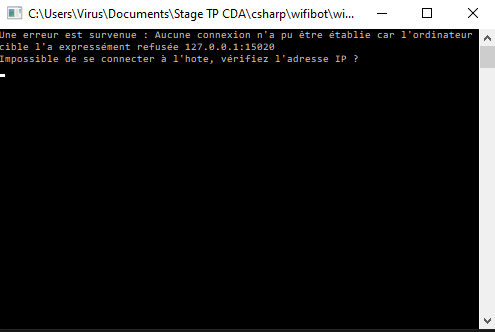
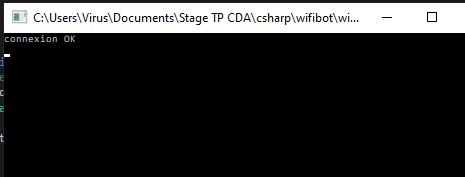
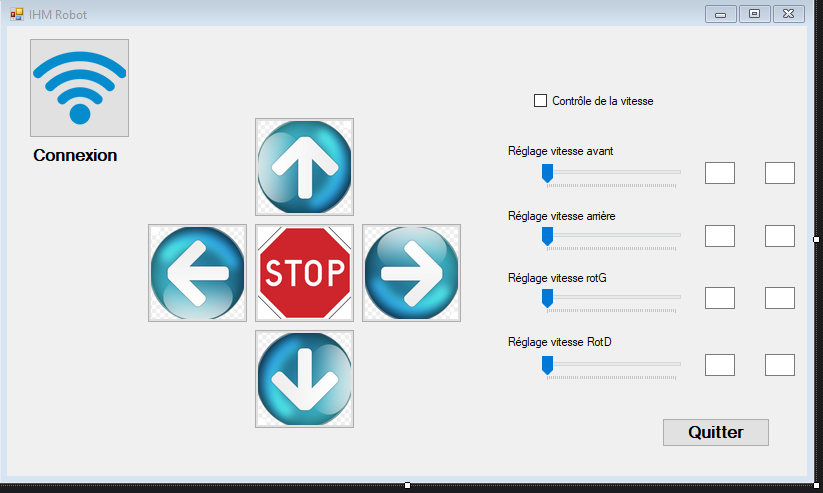
Test liaison OK :

Test connexion via la class dans une appli console.

* Test si le logiciel n’est pas lancé 
* Test avec un mauvais port TCP 15120. A suivre
* Test si tout est ok
* Test avant arrière Ok, voir film.
* Ajout des boutons de directions Ok,
* Ajout variables pour actionner les différentes directions (pouvoir changer les données facilement
* Ajout d’un timer (créé une methode « void timer\_Tick(object sender, EventArgs e) »
* Test des différents boutons, avec ajout d’un bouton Connexion au robot et un bouton Quitter.
* Comment récupérer l’erreur dans le programme principal en cas de non connexion. En console nous avions console.writeline. La comment faire ? Messagebox ne fonctionne pas dans une class.
* Après recherche j’ai décidé d’ajouter ceci « using System.Windows.Forms; »  dans la class. Cela m’a permis d’utiliser l’attribut MessageBox.Show.
* Ensuite, comme nous devons initialisés le Robot à « Stop » j’ai mis un try catch sur la boucle d’initialisation ce qui fait qu’il recup l’erreur et je redonnes le focus au bouton Quitter après le MessageBox
* Voir comment mettre en place ces boutons spéciaux Que l’on apelle Track\_Bar.
* Les boutons sont créés et sont bien mis a jour. Il me reste a traduire cette vitesse choisie en binaire pour connaître l’hexa à passer au robot.
* Il faut donc créer une fonction qui fera le calcul
* Après quelques vérif, ce n’est pas 64 la vitesse max mais 63 (100000) nous avons 64 positions 0 – 63
* J’ai ajouté un checkbox pour la vitesse contrôlée ou pas. Plus des textbox permettant de vérifier le chiffre decimal qui sera envoyé en byte au Robot.
* Je fais la mise à jour à chaque fois que l’on scroll sur les trackbars.
* Me reste à tester la connexion au cas où on l’a perd et éventuellement retenter la connexion.
* 
* .