

BE C++ Rapport: Deux bras robotiques

Réalisé par : **Hanna LILLE**
Luiz Renato RODRIGUES CARNEIRO
Bojian SHANG

En ces temps modernes, les robots occupent une place de plus en plus importante dans nos vies. Ils peuvent simplifier le travail ou le rendre plus efficace et plus précis. Par exemple, dans les hôpitaux, les médecins peuvent utiliser des bras robotisés pour effectuer des opérations à distance. Notre projet s'appuie sur cette idée. Nous avons créé deux bras robotiques, dont l'un reproduit le mouvement de l'autre. Une photo montrant les deux bras est présentée à la fin du rapport dans la figure 4.

Pour ce faire, nous avions besoin de deux ESP8266s. Le bras robotique que nous déplaçons manuellement est le client, et le bras qui va répliquer les mouvements est le serveur. Ils sont chacun connectés à un microcontrôleur. Le client est doté de trois potentiomètres, ce qui permet de mesurer le mouvement selon trois degrés de liberté. Le serveur dispose de trois servomoteurs qui, idéalement, reproduiront les mouvements exacts des potentiomètres du client. Le schéma de fonctionnement matériel avec toutes les connexions matérielles est présenté à la figure 1.

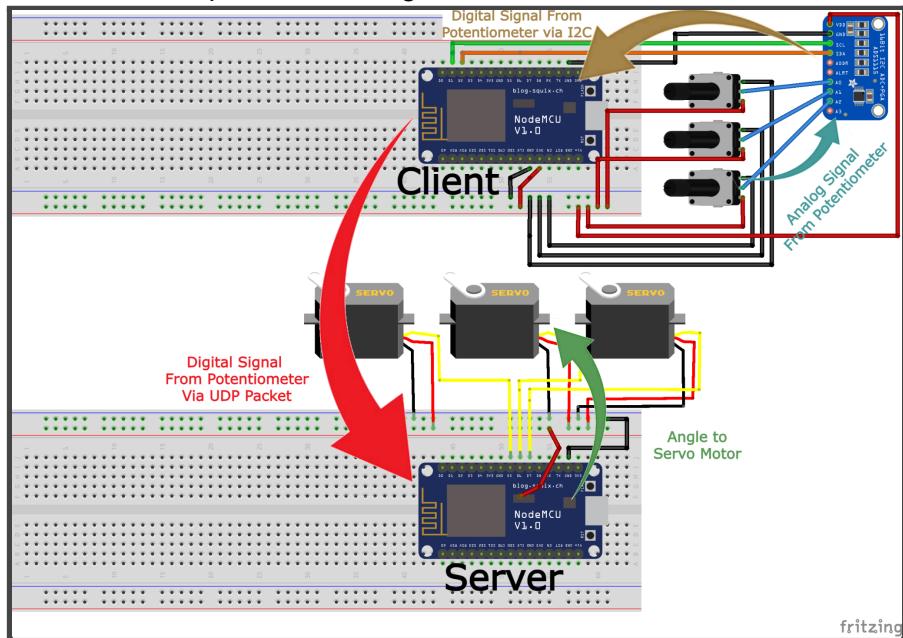


Figure 1: Le schéma de fonctionnement matériel

Une partie importante du projet est la communication entre le client et le serveur. Nous avons choisi UDP comme méthode de communication en raison du débit de connexion plus haut, de plus, c'est pas très important de recevoir tous les paquets avec connexion, puisque nous envoyons continuellement des angles et que la perte de l'un d'entre eux n'est pas si grave. Un diagramme de séquence simple du client envoyant des angles au serveur est présenté à la figure 2. Le diagramme montre que le client commence d'abord le paquet UDP, ce qui dans notre cas implique aussi l'envoi du start byte. Ensuite, il lit la valeur de chaque potentiomètre ; dans ce diagramme, une seule lecture est représentée. La valeur est enregistrée dans un map qui est un attribut du client, puis envoyée au serveur. Le paquet UDP est terminé par un end byte. Nous disposons d'un mécanisme permettant de savoir si la communication a été complètement perdue ; ce mécanisme n'apparaît pas dans le diagramme. Un compteur d'échec est utilisé pour

déTECTER si trop de paquets ont été perdus. Lorsqu'il atteint une certaine limite, une exception est lancée et la communication est déclarée perdue. Les angles provenant des potentiomètres sont de 16 bits, puisqu'ils proviennent de l'ADC. Les paquets UDP sont de 8 bits, donc nous divisons la valeur en deux bytes, MSB et LSB, avant de les envoyer. Du côté de la réception, nous devons rassembler les valeurs. L'opérateur "+" est surchargé pour cela, et l'addition du MSB et du LSB donnera la valeur rassemblée correcte.

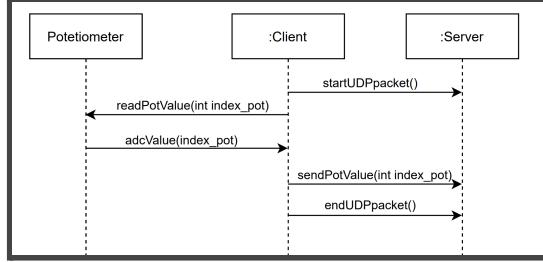


Figure 2: Le diagramme de séquence de l'envoi des angles du client au serveur

Le diagramme de classes du programme est présenté à la figure 3. Étant donné que de nombreux attributs et méthodes sont partagés entre le client et le serveur, il était naturel de créer une classe de base appelée `Client_Server`. Cette classe contient les attributs et les méthodes communs, comme les informations de réseau et de port pour la communication. La classe `Client` et la classe `Server` héritent toutes deux de la classe `Client_Server`, et elles contiennent toutes deux leurs propres attributs et méthodes. Tous les attributs et méthodes, ainsi que leur état de protection, sont visibles dans le diagramme de classes. La dernière classe de notre programme est la `Byte_Class`. Cette classe a été créée dans le seul but de surcharger un opérateur pour concaténer les deux bytes, MSB et LSB, comme expliqué précédemment. L'attribut de la `Byte_Class` est de type char, puisque le byte et le char ont la même taille.

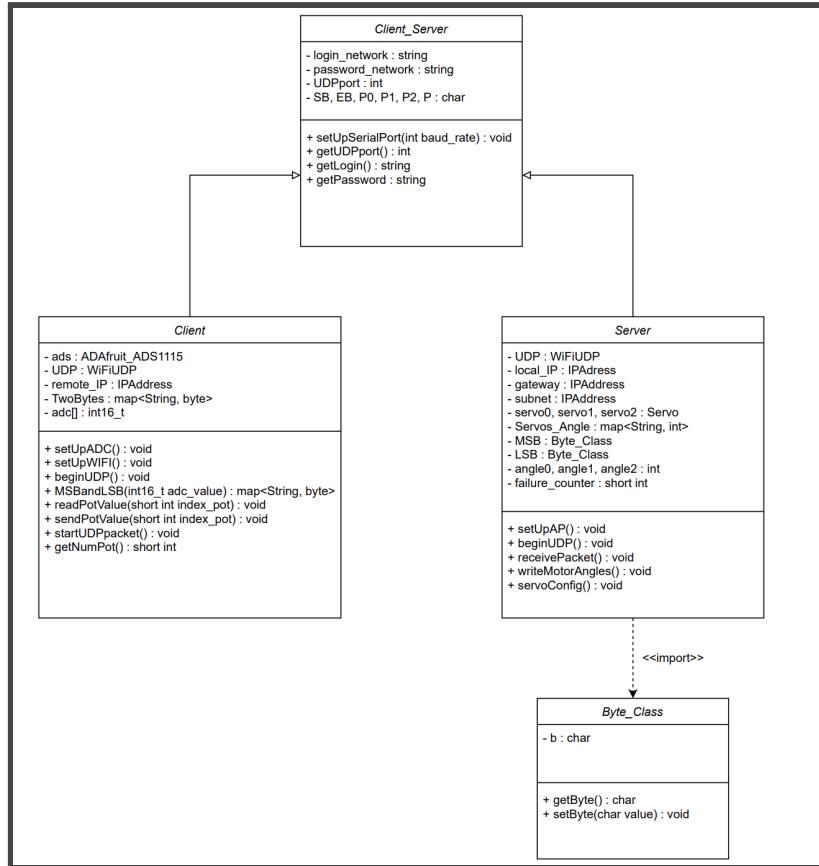


Figure 3: Le diagramme de classe

Ce projet nous a appris à concevoir notre propre concept orienté objet et à l'implémenter en code. L'objectif de faire fonctionner deux bras robotiques communiquant entre eux, l'un répliquant l'autre, a été atteint avec succès. Il y a eu quelques limitations matérielles, comme la limitation de rotation des moteurs et des potentiomètres. Nous avons également remarqué qu'il y avait un peu de bruit dans certains des capteurs, ainsi qu'une limitation du poids qu'ils pouvaient porter. Afin d'améliorer potentiellement le projet à l'avenir, nous rendrions le code du serveur plus générique, afin qu'il puisse se connecter à un nombre arbitraire de servomoteurs, comme le client peut se connecter à un nombre arbitraire de potentiomètres. Une autre amélioration de notre projet serait d'utiliser des matériaux plus robustes, comme des pièces imprimées en 3D. Le projet a fait l'objet d'un travail constant et a évolué à un bon rythme. Nous avons commencé par nous assurer que la communication UDP fonctionnait, puis nous avons conçu la conception orientée objet et modifié le code pour réaliser la conception. Nous sommes tout à fait satisfaits du résultat du projet, et nous voyons clairement sa pertinence. Un bras robotique performant, permettant un contrôle à distance, peut être utile dans de nombreux domaines, tels que la médecine, l'enseignement, les travaux de construction, etc. Dans certains cas, il peut même être plus sûr d'employer un bras robotique que d'envoyer un humain quelque part.

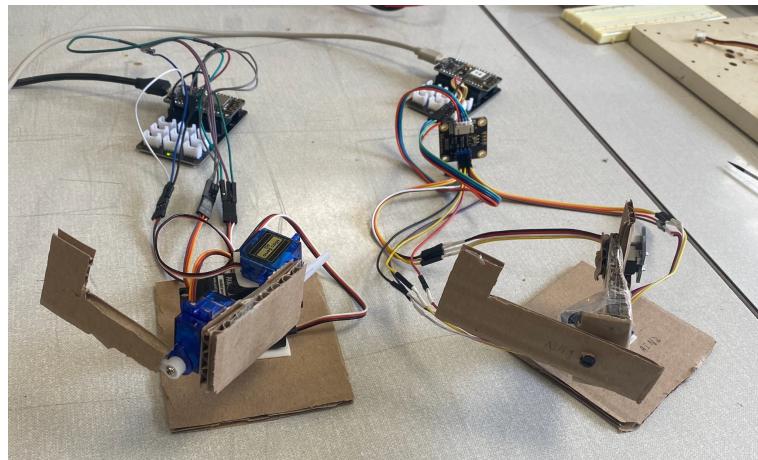


Figure 4: Une photo des deux bras robotiques