Fait le 7/10 :

Plan:

Il s'agit de pouvoir guider un robot de type voiture grâce à un mécanisme de capteur de flexion sur des gants.

Matériel :

Les capteurs de flexion transmettent une valeur d’angle de flexion a un esp32. (environ 10)

Le véhicule possède 4 roues commander un esp32 Dev Module.

Objectif :

* Les gants transmettent des informations de flexion à un Arduino esp32.
* On traite ses informations pour les transformer en commande : avance, recule, tourne à droite/gauche , arrête.
* Ses informations sont transmises par wifi ou Bluetooth bas niveau à un autre Arduino, une voiture Esp32 qui exécute les ordres transmis.
* Transmission d’information à travers un serveur qui relire les deux Arduino.
* Elle restitue au pc les informations de température et de vision dans une page html avec le wifi.

<https://www.gotronic.fr/blog/articles/controle-dune-main-robotique/>

<https://www.robotique.tech/tutoriel/commander-une-voiture-controlee-par-la-carte-esp32-via-une-connexion-wifi/>

<https://www.tubefr.com/arduino-portable-clavier-gant.html>

<https://www.robotique.tech/tutoriel/commander-une-voiture-equipee-dune-carte-esp32-par-smartphone-via-bluetooth/>

<https://www.robotique.tech/tutoriel/commander-une-voiture-equipee-dune-camera-via-une-connexion-wifi-avec-la-carte-esp32-cam/>

<https://randomnerdtutorials.com/projects-esp32/>

Fais le 14/10 :

Quel mouvement pour quelle action.

Commande :

- Avancer , avancer droite , avance gauche, recule , recule droite, recule gauche, arrêt.

Utilisation des deux mains pour commander le véhicule.

Main Gauche : poing fermer -> arrêt ; main ouverte -> autre . possibilité de gradation dans la vitesse

On veut pouvoir s’arrêter rapidement

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Main gauche | Main Droite | Moteur :G1,G2,D1,D2 |
| Arret | Main fermée | ------------------------ | 0,0,0,0 |
| Avancer | Main ouverte | Index | 100,100,100,100 |
| Avancer droite | Main ouverte | Index + majeur | 100,100,50,50 |
| Avancer gauche | Main ouverte | Ponce + index | 50,50,100,100 |
| Reculer | Main ouverte | Main ouverte | -50,-50,-50,-50 |
| Reculer droite | Main ouverte | Pouce index majeur | -50,-50,-25,-25 |
| Reculer gauche | Main ouverte | Majeur annul. auri. | -25,-25,-50-,50 |

Structure de l’envoie et la réception des commandes :

Utilisation de MQTT et d’un réseau wifi pour la transmission des informations nécessaires.

Les capteurs du gant envoient les informations de flexion au esp32 n°1 qui transforme les informations en données : non fléchie = 0 ; fléchie = 1. L’esp traduit ensuite ces informations en l’une des 7 commande vue plus tôt qui sont envoyer avec le protocole mqtt au serveur Node RED sous un format json.

Le serveur envoie ses informations directement à l’esp n°2 (véhicule) et Node RED affiche les informations transmises dans un Dash board.

L’esp 32 exécute ses instructions et renvoies-en continue au serveur Node RED via mqtt la vidéo enregistrer par sa caméra et que de température dans deux canaux mqtt différant. (le délai de l’envoie peut-être réduit à l’arrêt). NOON.

GitHub : gmenez -> faire un GitHub avec le code etc. Et inviter menez au projet

Un esp timer camera va envoyer la vidéo indépendamment de l’esp de contrôle et l’esp des gants.

Il a une interface déjà faite, voir les exemples de Arduino -> sur une page web avec une interface indépendante de Node RED (à avoir ou pas)

Ou un esp cam ??? (pas pour l’instant.)

ATTENTION : pas de vidéo en continue !!!

Etape 1 :

Etape 2 : savoir lire les valeurs du capteur et codé les contrôles à priori avec le véhicule.

Fait le 21/10 :

Test et fonctionnement de l’esp timer cam. Attention : Si le message wifi INIT apparait, c’est le wifi qui pose un problème. (le wifi à partir de mon portable ne marche pas ??)

Montage du véhicule reporter à ce week-end ? ou à faire la semaine prochaine, voir la personne à droite au 2nd étage, qui possède du matériel électronique pour le matériel de montage.

«penser à mettre les librairies utiliser sur GitHub, ou une trace sur le rapport.

* Moniteur de carte m5stack
* [VCP Drivers - FTDI (ftdichip.com)](VCP%20Drivers%20-%20FTDI%20(ftdichip.com)) : https://ftdichip.com/drivers/vcp-drivers/

spécification de l’esp timer camera : [m5stack/TimerCam-arduino: TimerCam Arduino Library (github.com)](https://github.com/m5stack/TimerCam-arduino)

le moniteur pour les capteurs de flexion fonction, appeler test\_flex.

Remarque, la différence de valeur donner est plus grande lors de flexion “à l'envers” qu’à l'endroit !!! (gp : D33)

Chose faite :

7/10 : recherche pour le state of the art et la réalisation technique du projet

14/10 :

* suite et fin de la recherche, récupération du matériel
* mise en place d’un schéma a priori du fonctionnement de la liaisons gant -> robot
* Schéma théorique du projet

Objectif 21/10 :

Test du fonctionnement de la caméra, montage de la voiture, voir les câblages

Tester de lire les valeurs du capteur, se familiariser avec le matériel

Pas de montage possible sans matériel et sans accès au labo par exemple.

Fait le 21/10 : réalisation du fichier test\_flex.ino qui permet de comprendre comment fonctionne le capteur de flexion et les différentes valeurs que l’on peut attendre du capteur.

Récupération dans du state of art un fichier CameraWebServeur pour tester la caméra et c’est fonctionnalité.

Fait le 28/10 : début du montage du véhicule :

* Pas mal de problème de gestion de la place et du câble management et de comment monter le véhicule. Récupération d’un code pour tester les moteurs.

Fait le 4/11 : Finalisation du montage du véhicule sans la caméra et réalisation du câblage final entre les moteurs et les servo-moteur.

J’ai réalisé un code teste pour l’activation des différents modèles mais je me suis buté sur un problème de nombre de port disponible sur l’esp pour contrôler mes 4 moteur indépendamment.

De nombreux port qui sembler disponible

Pour ne pas prendre de retard, j’ai décidé de commander les 4 moteur 2 part 2 pour le moment. La même commande sera donner au moteur de droite et à ceux de gauche.

Le dossier avancer\_reculer commande au robot d’avancer, de reculer puis de s’arrêter pour tester la puissance du moteur.

A faire le 18/11 : Réaliser un broker mqtt pour commander le véhicule à distance :

* Mettre en place le broker

Fait le 18/11 : Connexion robot – broker réaliser. Commande basic du véhicule avec des ligne de code à travers un verseur mqtt de test (à changer à voir).

Envois de message par un json (parser json pour lire les valeurs), possibilité de rajouter des infos sur le véhicule ou possiblement de la vitesse demander

Ce qui reste à faire :

* Coder la traduction, signal gant -> commande. (3h)
* Calibrage des capteurs (2h)
* Réalisation de deux paradigmes différents : 1er ,commande absolue, pas de contrôle de vitesse (simple et déjà possible), 2ème ,commande avec un valeur de vitesse pour un meilleur contrôle (à coder et implémenter du côté du robot). (2~3h )
* Installation de la caméra et d’un 2nd étage sur le véhicule (1h)
* Adapter le code de lecture de l’image pour renvoyer un flux d’image régulier (~2 ou 3 fps) (2~3h)
* Installer puis gérer le retour des informations de températures. (4~5h)
* Créer une heatmap avec les informations retourner (3~4h)

Rapport, diagramme de Gantt ??

<https://github.com/bblanchon/ArduinoJson/blob/6.x/examples/JsonParserExample/JsonParserExample.ino>

protocole du gant + vision (vidéo) à réfléchir, pour la suite.

Dans le rapport, des images et des caractéristique, le schéma etc…

Fait le 25/11 : rédaction de rapport mid terme

Objectif 02/12 : Commencer le code de l’Arduino qui gérera le gant

Commande « finale » :

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Main gauche | Main Droite | Moteur :G1,G2,D1,D2 |
| Arret | Main fermée | ------------------------ | 0,0,0,0 |
| Avancer | Pouce | Main fermé | 220,220,220,220 |
| Avancer droite | Pouce | Index | 255,255,180,180 |
| Avancer gauche | Pouce | Index + majeur | 180,180,255,255 |
| Reculer | Pouce | Pouce | -220,-220,-220,-220 |
| Reculer droite | Pouce | Pouce index | -255,-255,-180,-180 |
| Reculer gauche | Pouce | Pouce index majeur | -180,-180,-255-,255 |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| Mouvement Caméra | Index | ---------------------------- | ---------------------------- |
| Envoie info température | Majeur index | ---------------------------- | ---------------------------- |

Fait 02/12 : Début du code du gant. Fin du montage des gants

Objectif 09/12 : Faire le calibrage des capteurs et compléter/finir le code du gant.