

ENSTA BRETAGNE

PROJET ROS 4.1

ROS : implémentation du suivi de ligne
sur une voiture à l'échelle 1/10ème



Colin BAUMGARD

Corentin LEMOINE

Ludovic DIGUET

Antonin LIZÉ

Hamid HACENE

Table des matières

1	Introduction	2
1.1	Le cahier des charges	2
1.2	Les pistes envisagées	3
2	Structure du projet	3
2.1	Lancement du programme	3
2.2	Packages	3
3	Conclusion	4

1 Introduction

1.1 Le cahier des charges

Fonction	Désignation	Critères	Niveau	Flexibilité
FP1	Permettre que la voiture fasse un tour de circuit en autonomie	Vitesse	10km/h	F3: $\pm 5\text{km/h}$
		Précision max par rapport au milieu de la ligne	5cm	F1: +1cm
FC1	Doit avoir un poids adapté	Poids	4kg	F3
FC2	Doit permettre la maîtrise de la vitesse	Commande PWM		F0
FC3	Ne doit pas consommer trop de batterie			F2
FC4	Doit gommer les aspérités du terrain	Souplesse de la structure et des composants		F2

FIGURE 1 – Cahier des charges

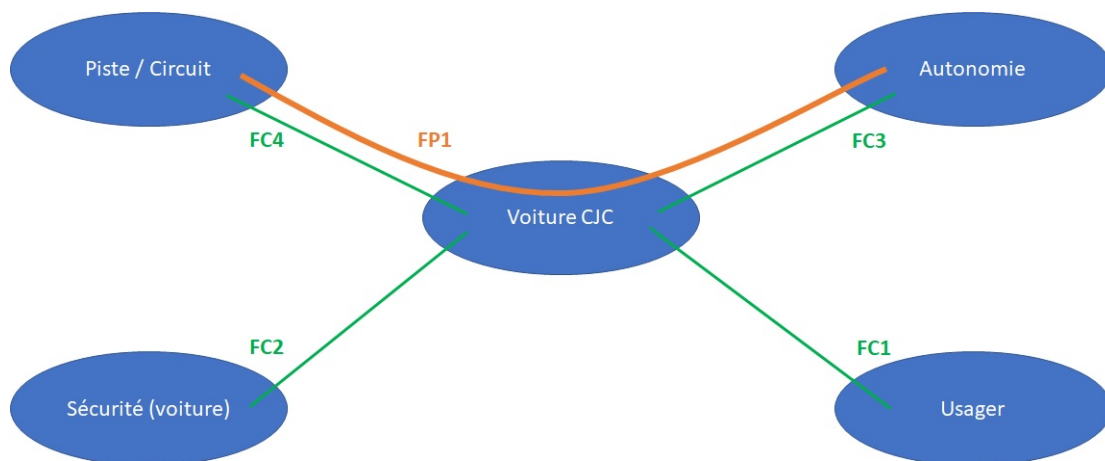


FIGURE 2 – Diagramme pieuvre

Architecture logicielle

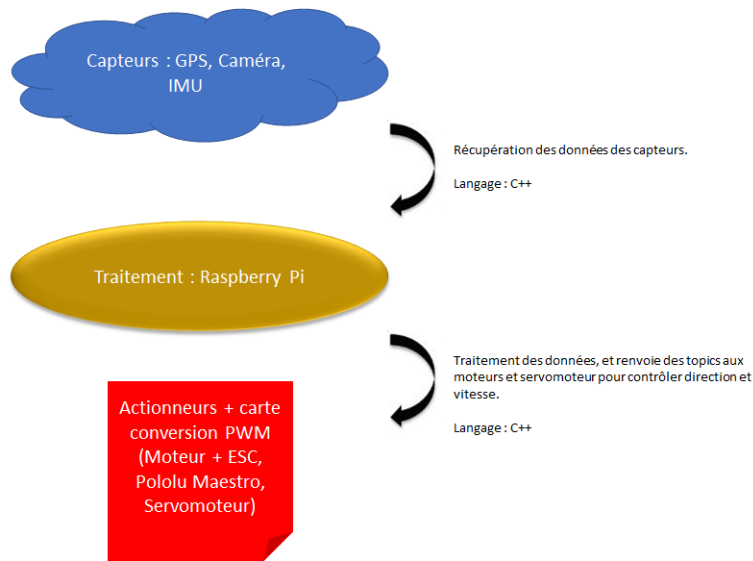


FIGURE 3 – Architecture Logicielle Matérielle

1.2 Les pistes envisagées

Lorem ipsum dolor sit amet

2 Structure du projet

2.1 Lancement du programme

Lorem ipsum dolor sit amet

```
roslaunch waypoints_follow seeMeRollin.launch
```

2.2 Packages

Drivers

Les drivers sont implémentés en utilisant le plus possible les nodes distribuées avec ROS. Le driver du GPS est celui du package *nmea_navsat_driver* ; la node utilisée est *nmea_serial_driver*. La node *navsat_transform_node* du package *robot_localization* transforme alors la donnée issue du driver en une projection dans le repère du robot.

Le package *cv_camera* nous permet de publier une image sur un topic ROS issue de la caméra de la Raspberry pi.

Pour l'envoi de signaux *PWM* vers l'ESC et le servo de direction, on a codé une node Python utilisant une bibliothèque pour contrôler la carte Maestro via la connexion série USB.

Traitement d'image

Lorem ipsum dolor sit amet

Contrôleur

Lorem ipsum dolor sit amet

Exemple latex langage c :

```
__kernel void mandelbrot(__global float2 *q,
__global ushort *output, ushort const maxiter)
{
    int gid = get_global_id(0);
    float nreal, real = 0;
    float imag = 0;
    output[gid] = 0;
    for(int curiter = 0; curiter < maxiter; curiter++) {
        nreal = real*real - imag*imag + q[gid].x;
        imag = 2* real*imag + q[gid].y;
        real = nreal;
        if (real*real + imag*imag > 4.0f)
            output[gid] = curiter;
    }
}
```

3 Conclusion

Lorem ipsum dolor sit amet