

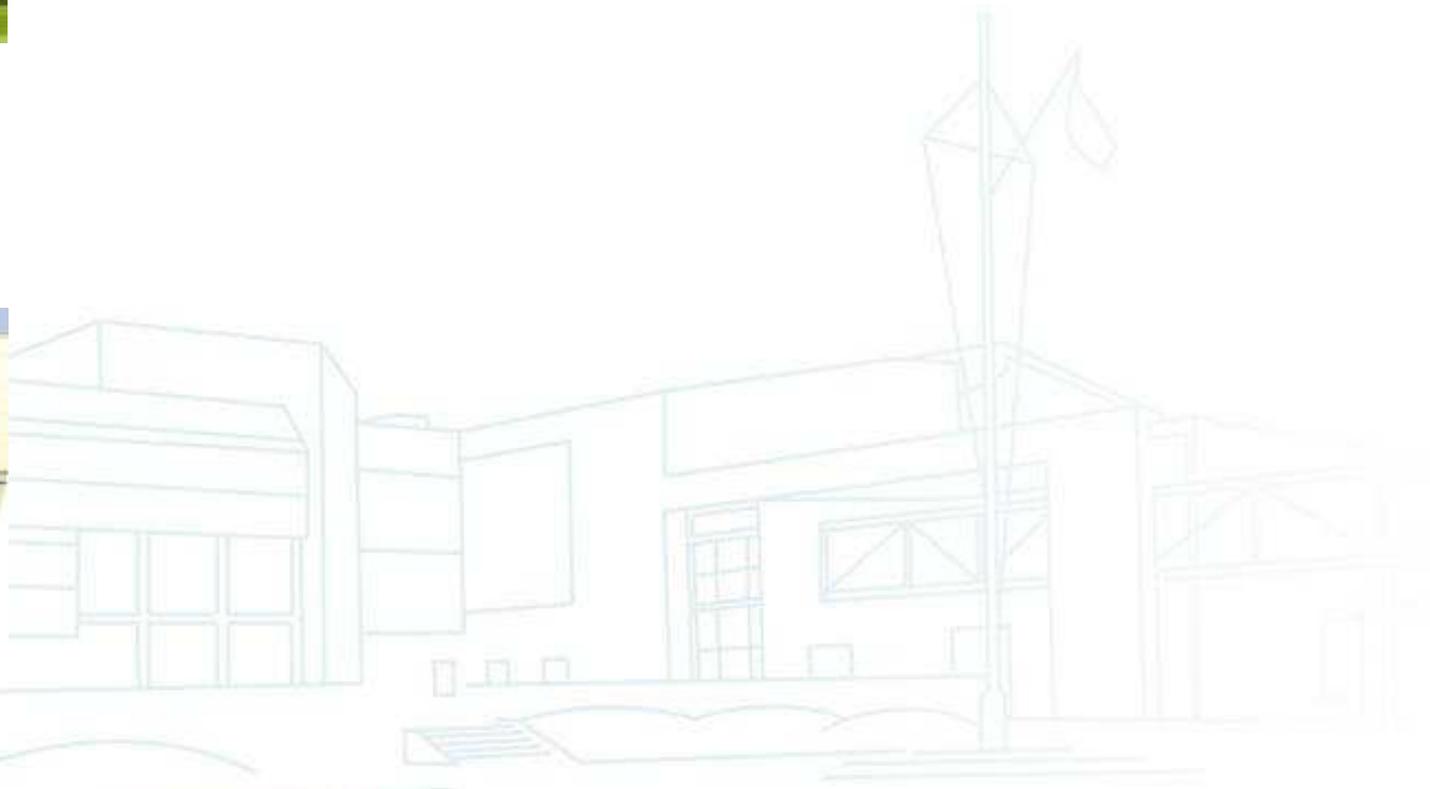


[Robot buggy autonome avec Android]

Fabrice LE BARS

Plan

- Introduction
- Constitution du robot
- Android
- IOIO



Introduction

But

- Faire un robot buggy capable de suivre une trajectoire définie par des points GPS



Constitution du robot

Plateforme mécanique + moteurs

- Exemple : buggy radiocommandé Graupner Punisher Crawler 4WDS RTR



Carte de puissance

- Permet de contrôler les moteurs par des signaux de commande
 - Moteurs : tensions et courants élevés provenant des batteries
 - Signaux de commande : tensions et courants faibles venant directement ou indirectement du PC
Exemples : signaux PWM, I2C



Carte de puissance

- Exemple : Robbe Rokraft



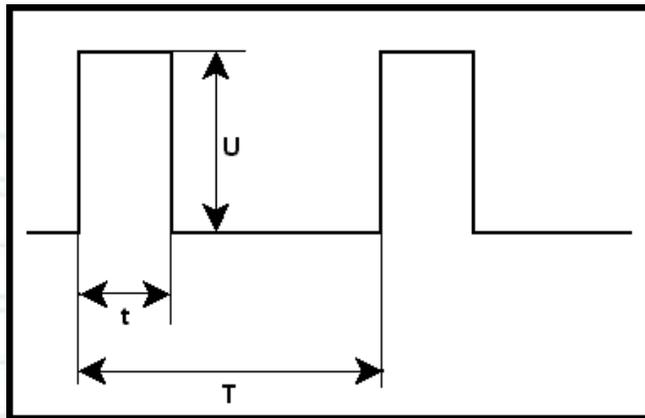
Carte de puissance

■ Exemple : Robbe Rokraft

● Fonctionnement

La puissance envoyée aux moteurs (et donc leur vitesse) dépend du signal de commande PWM

PWM = Pulse Width Modulation : modulation en largeur d'impulsion



U : tension du PWM (5 V)

t : largeur d'impulsion (entre 1 et 2 ms)

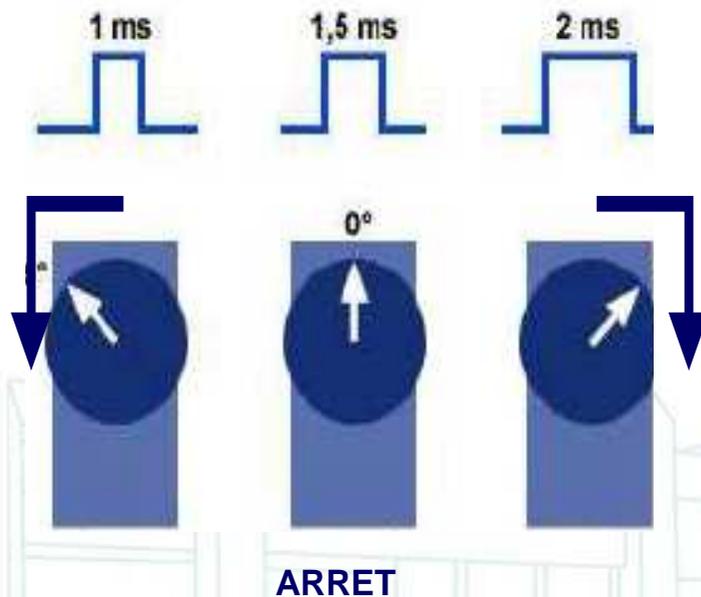
T : période (20 ms)

Carte de puissance

- Exemple : Robbe Rokraft

- Fonctionnement

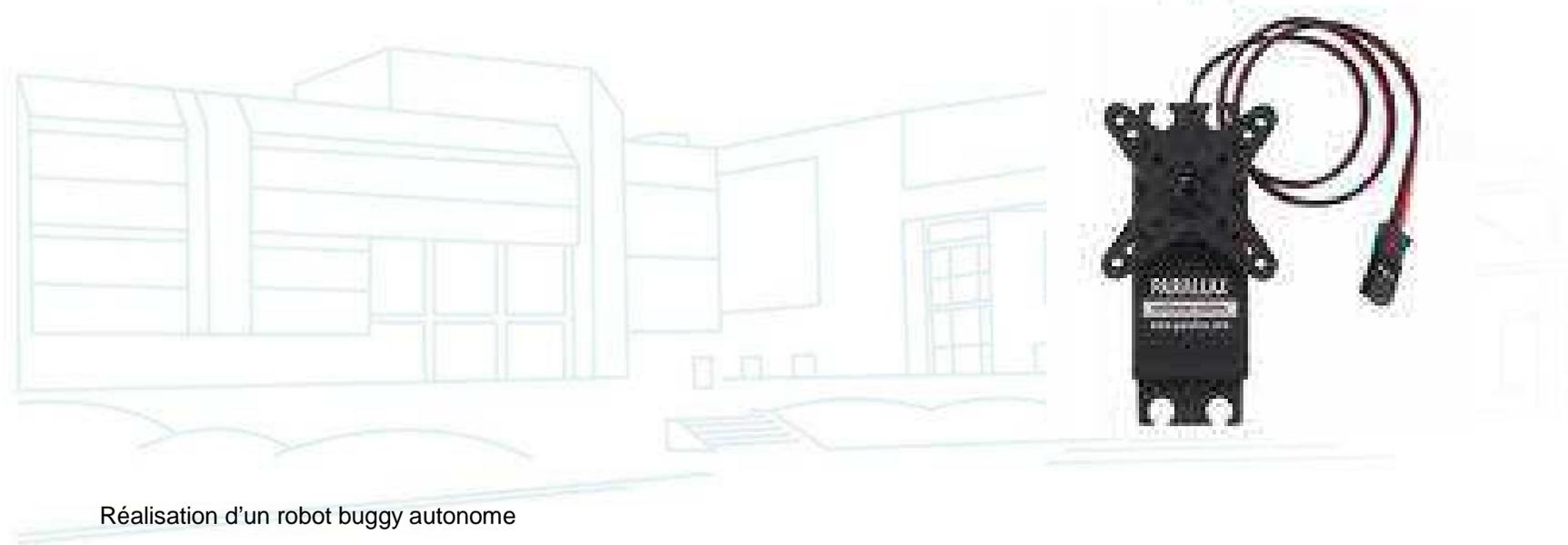
Correspondance largeur d'impulsion / vitesse de rotation



État du moteur	Largeur d'impulsion
Moteur à l'arrêt	1.5 ms
Rotation dans un sens, en accélérant	1.5 à 2.0 ms
Rotation dans le sens inverse, en décélérant	1.0 à 1.5 ms

Servomoteur

- Servomoteur = petit moteur + carte de puissance
- Commandé par PWM
- 2 types de servomoteurs :
 - Asservis en position : tournent de -40 à $+40^\circ$ par exemple
 - Asservis en vitesse



Carte d'interface

- Relie la partie informatique avec la partie électronique (capteurs, actionneurs)
 - Partie informatique : intelligence par le biais de programmes sur PC
 - Partie électronique : capteurs, actionneurs



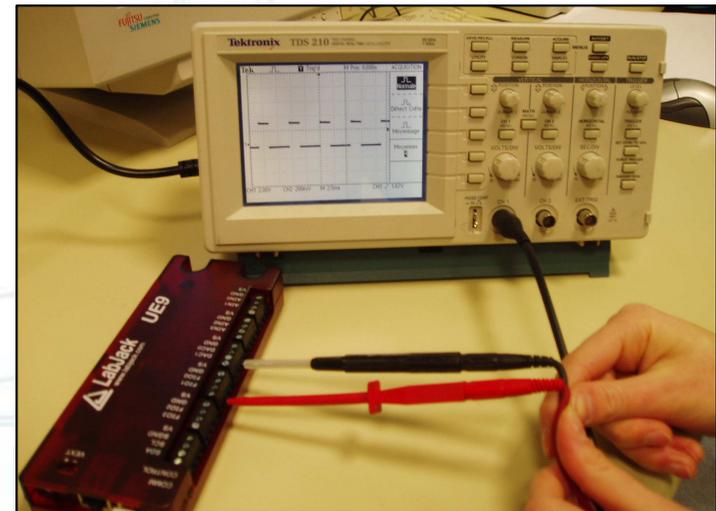
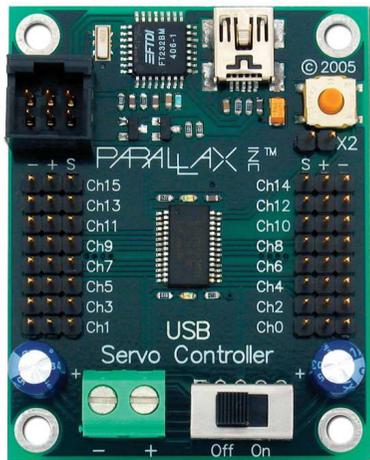
Carte d'interface

- Exemple : carte IOIO pour smartphone/tablette Android
 - Se branche sur le port USB du smartphone et est contrôlé par des programmes exécutés sur le smartphone
 - Peut générer des signaux PWM, I2C
 - Peut générer et lire des signaux numériques
 - Peut lire des petites tensions (venant de capteurs analogiques tels que des télémètres, odomètres, boussoles...)
 - ...



Carte d'interface

- Autres exemples : Cartes Parallax, Pololu, Labjack pour PC



Capteurs

- GPS, boussole, caméra...



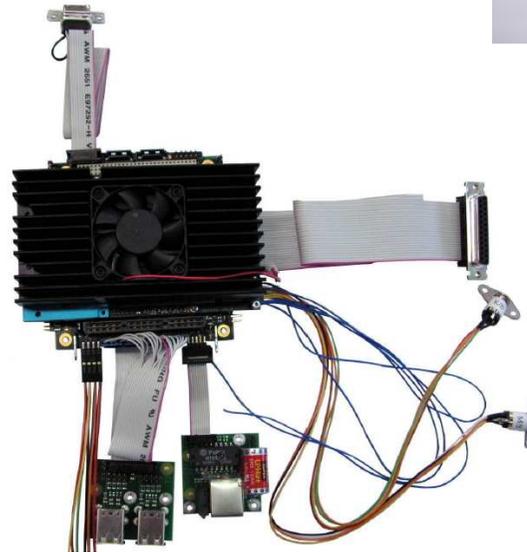
PC embarqué

- Intelligence du robot
 - Contient les programmes définissant le comportement du robot
- Exemple :
 - Smartphone / tablette
 - Smartphone Samsung Galaxy S sous Android (avec GPS, boussole, caméras, Wifi déjà intégrés)



PC embarqué

- Autres exemples :
 - EeePC 901 (netbook)
 - Mini ITX
 - PC/104
 - ...



Computer form factors	
Name	Size (mm)
eeePC 901	226 × 175.3 × 22.9
Mini-ITX	170×170
Nano-ITX	120×120
Pico-ITX	100×72
PC/104	96×90

Périphérique de communication

- Relie le robot au PC de commande
- Exemple : clé Wifi USB, Wifi intégré au smartphone...



Android

Android, un système d'exploitation pour smartphone et tablettes

- OS de Google
- Basé sur un noyau Linux modifié
- Ne contient pas les commandes et outils habituels sous Linux
- Est fait pour être programmé en Java sous Eclipse



Android, un système d'exploitation pour smartphone et tablettes

- Application Android
 - **Activity** : fenêtre décrite par classe Java+fichier XML
 - Contient une **Activity** principale : classe correspondant à la fenêtre principale de l'application. Celle-ci peut provoquer l'ouverture d'autres fenêtres
 - Peut utiliser des services systèmes (**LocationManager**, **SensorsManager...**)
 - **AsyncTask** : thread
 - **Fragments** : petites fenêtres temporaires, boîtes de dialogue

Package Explorer

- Buggy
 - src
 - fr.enstabretagne.buggy
 - MainActivity.java
 - gen [Generated Java Files]
 - Android 4.1.2
 - Android Dependencies
 - assets
 - bin
 - libs
 - res
 - drawable-hdpi
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - layout
 - activity_main.xml
 - menu
 - values
 - strings.xml
 - styles.xml
 - values-v11
 - values-v14
 - xml
 - AndroidManifest.xml
 - ic_launcher-web.png
 - proguard-project.txt
 - project.properties
- HelloTOIO
- IOIOLib
- MyFirstApp

Palette

Form Widgets

- TextView Large Medium Small Button
- Small OFF CheckBox
- RadioButton CheckedTextView
- Spinner
- Slider
- Switch

Text Fields

Layouts

Composite

Images & Media

Time & Date

Transitions

Advanced

Other

Custom & Library Views

Graphical Layout

Nexus One

MainActivity

Android 17

Buggy

NMEA

gpsStatus

utc

lat

long

alt

speed

bearing

azimuth

pitch

roll

OFF

Structure

Outline

- LinearLayout
 - nmeaTextView - "NMEA"
 - gpsStatusTextView - "gpsStatus"
 - utcTextView - "utc"
 - latTextView - "lat"
 - longTextView - "long"
 - altTextView - "alt"
 - speedTextView - "speed"
 - bearingTextView - "bearing"
 - azimuthTextView - "azimuth"
 - pitchTextView - "pitch"
 - rollTextView - "roll"
 - button (ToggleButton)
 - seekBar1
 - seekBar2
 - seekBar3

Properties

<No properties>

Problems Javadoc Declaration Console LogCat Emulator Control Devices

Android

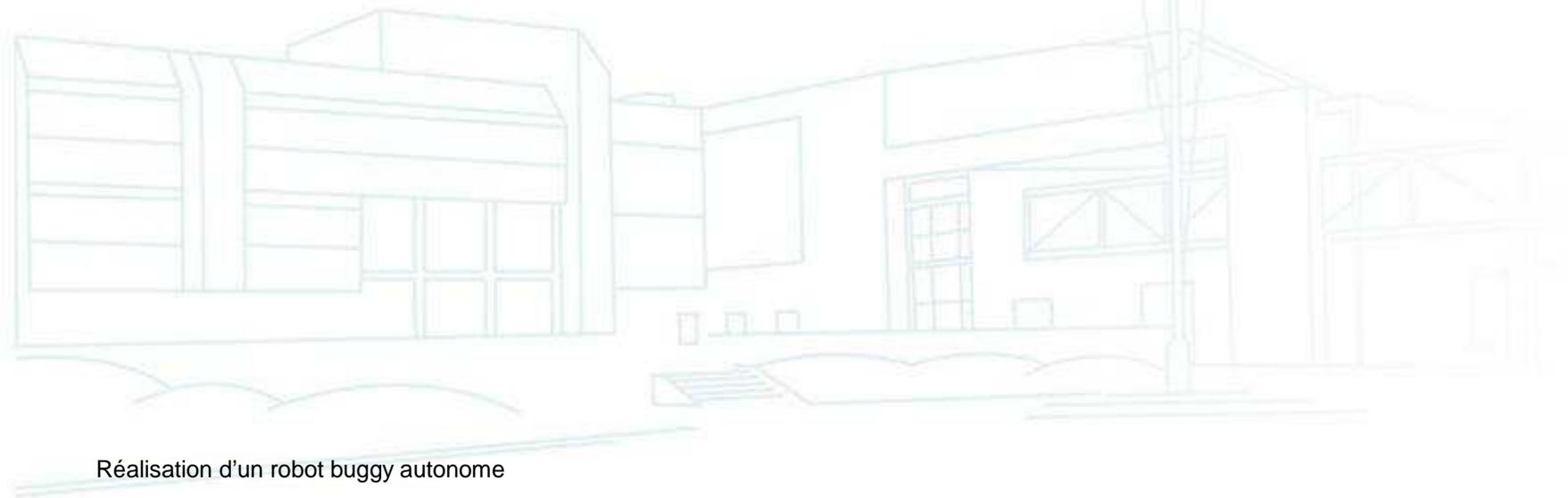
Android, un système d'exploitation pour smartphone et tablettes

- Hello World
 - <http://developer.android.com/training/basics/firstapp/index.html>
- Guides de programmation
 - <http://developer.android.com/guide/components/index.html>
- Documentation
 - <http://developer.android.com/reference/packages.html>



IOIO

- Documentation (à lire en priorité pour savoir comment la brancher)
 - <https://github.com/ytai/ioio/wiki>
- HelloIOIO
 - <http://www.sparkfun.com/tutorials/280>

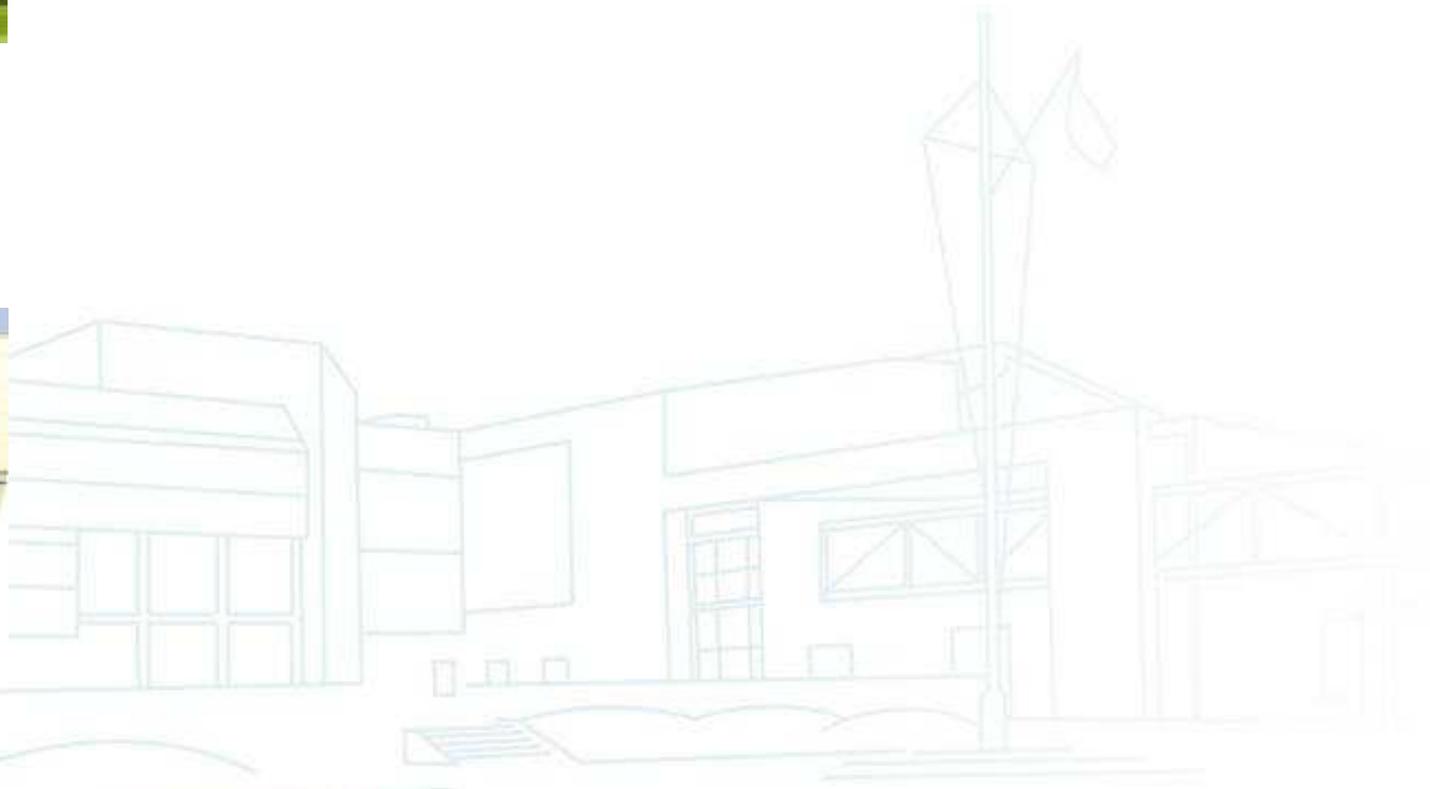


- Alimentation dans notre cas
 - Via BEC de la carte de puissance Rokraft (convertit la tension des batteries en 5V)
 - Cette alimentation remonte vers le smartphone via le port USB
- Entrée sortie-sorties utilisées
 - 3 PWM : 1 pour les 2 moteurs de traction et propulsion, 1 pour l'essieu directeur avant, 1 pour l'essieu directeur arrière (ce dernier est optionnel)

IOIO

- USB
- PWM
- 5 V
- 12 V





Modèle d'état du buggy et équations géométriques

Buggy : modèle de type voiture

$$\begin{cases} \dot{x} &= v \cos \delta \cos \theta \\ \dot{y} &= v \cos \delta \sin \theta \\ \dot{\theta} &= \frac{v \sin \delta}{L} \end{cases}$$

$$\begin{cases} y_1 &= x \\ y_2 &= y \\ y_3 &= \theta \end{cases}$$

$$v = \alpha u_1$$

$$\delta = \beta u_2$$

L Distance entre les trains avant et arrière

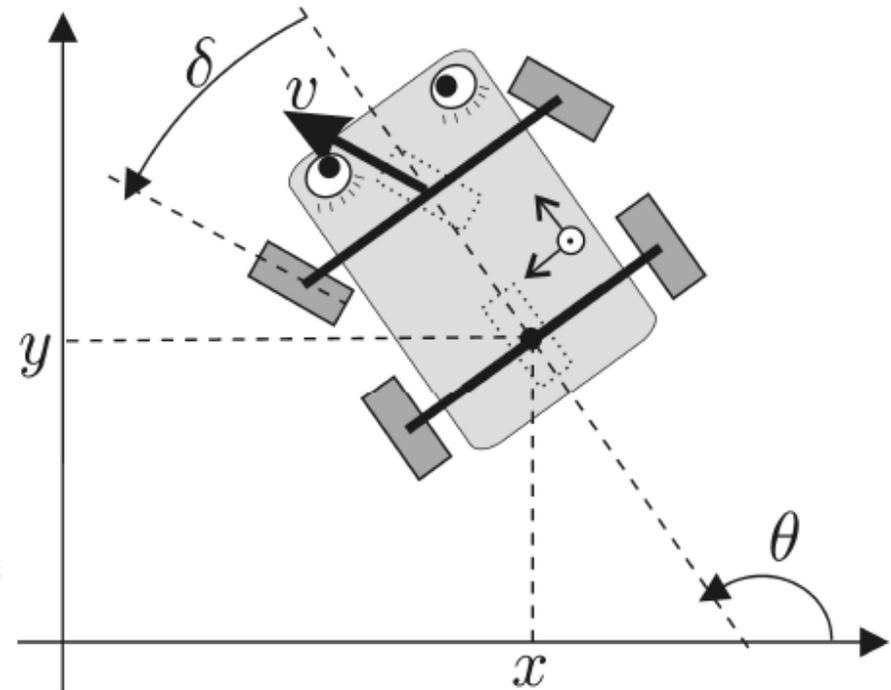
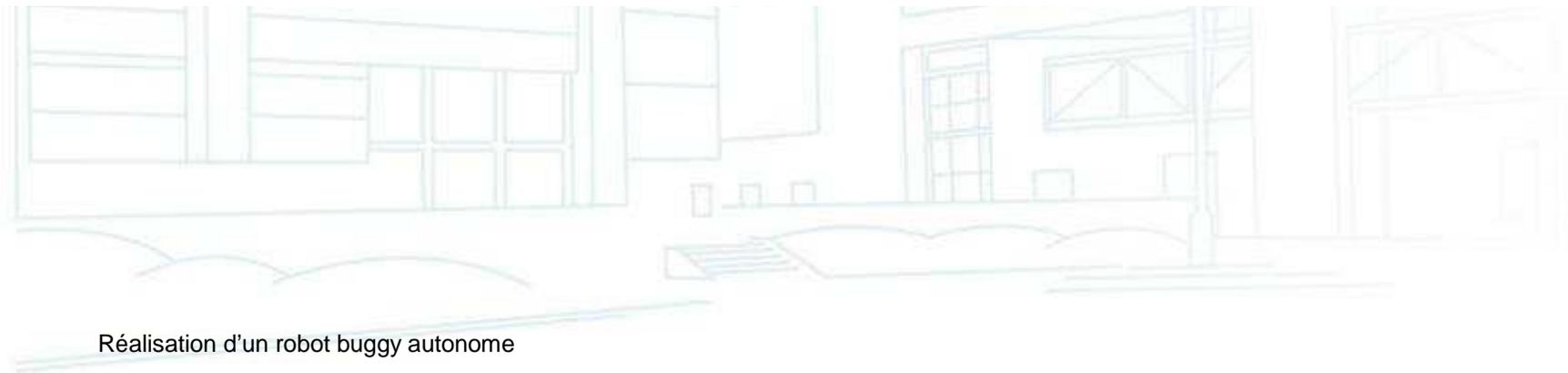
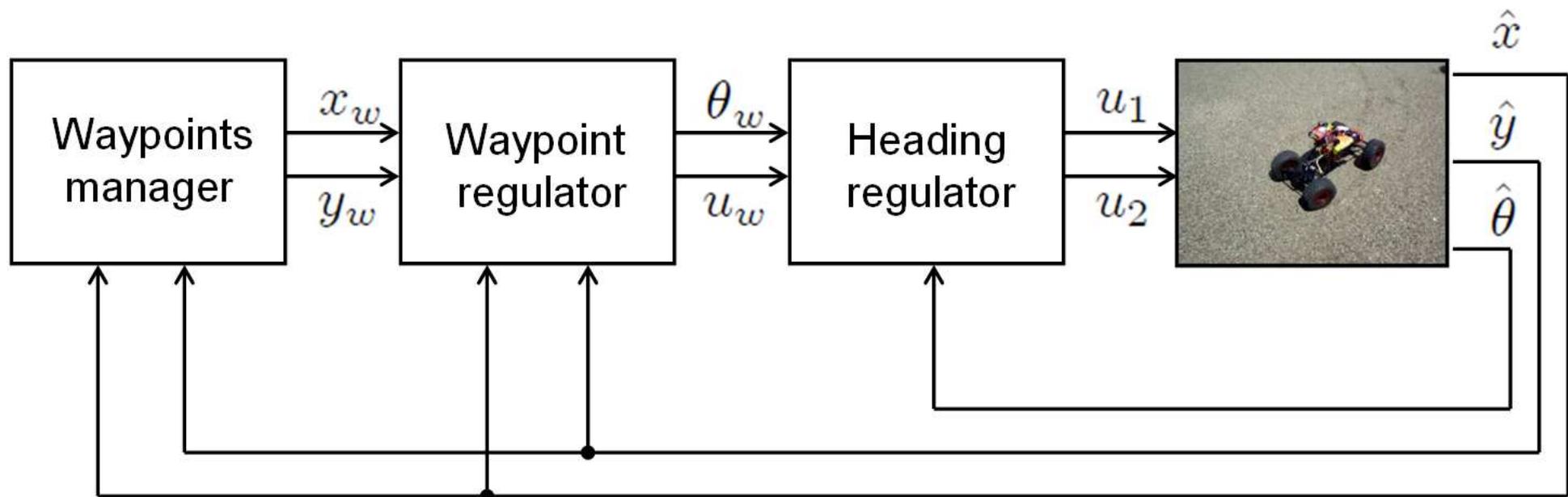


Schéma du système pour le suivi de waypoints GPS



Régulation à une orientation voulue grâce à la boussole, à une vitesse arbitraire

- La boussole nous donne un angle au Nord en degrés $\hat{\theta}$
- Régulation à un cap voulu θ_w :

- Commande bang-bang: on fait tourner le robot à la vitesse de rotation maximale lorsqu'il est tourné dans le mauvais sens par rapport au cap voulu
- Proportionnelle à l'erreur autrement :

$$u_1 = K_p (\theta_w - \theta)$$

$$u_2 = u_w$$

- Attention aux problèmes de modulo 2π : utiliser des sin et cos par exemple