ENSTA Bretagne

Réalisation d'un robot buggy autonome

Fabrice LE BARS

Plan

- Introduction
- Constitution du robot
- Equations d'état et régulation
- Android
- IOIO

Introduction

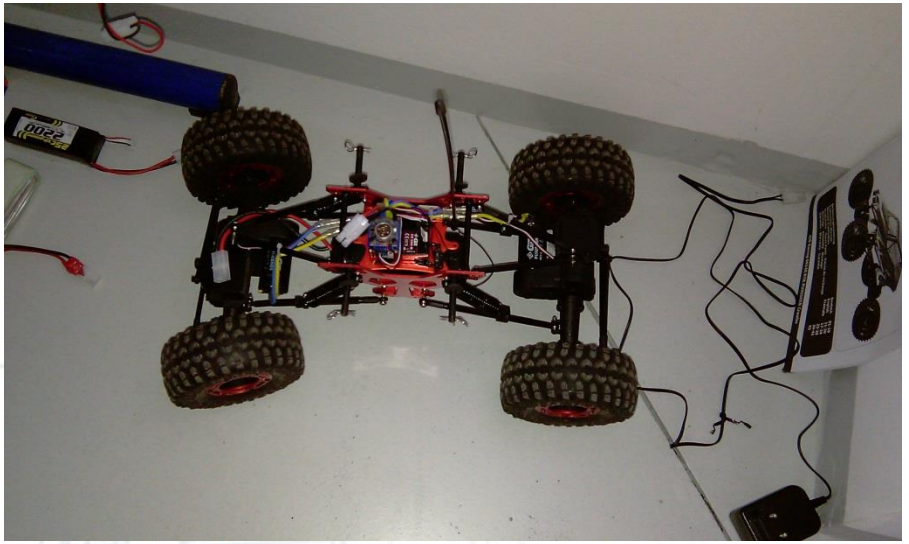
But

- 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/variateur/ESC

- 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 de l'ordinateur embarqué

    Exemples : signaux PWM (le plus courant en modélisme), I2C, etc.

Carte de puissance/variateur/ESC

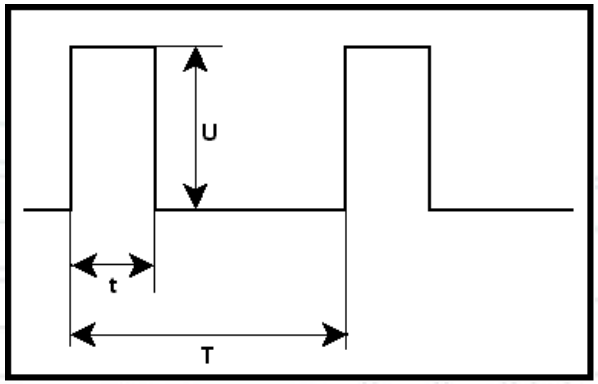
- Exemple : Robbe Rokraft (brushed motors)

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

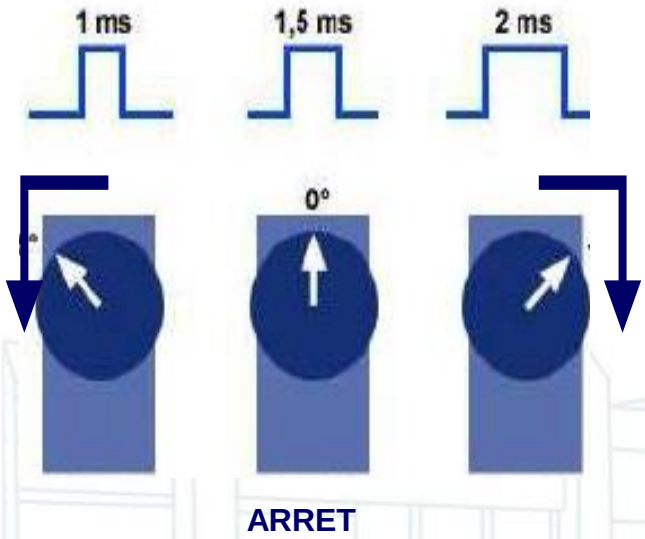


U : tension du PWM (5 V)
t : largeur d'impulsion (entre 1 et 2 ms)
T : période (20 ms)

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

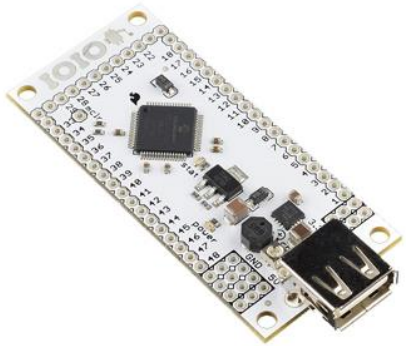
- Servomoteur = petit moteur + carte de puissance
- Commandé par PWM
- Alimentation 5 V (< 2 A en général)
- 2 types de servomoteurs :
  - Asservis en position/angle : tournent de -40 à +40° par exemple
  - Asservis en vitesse

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

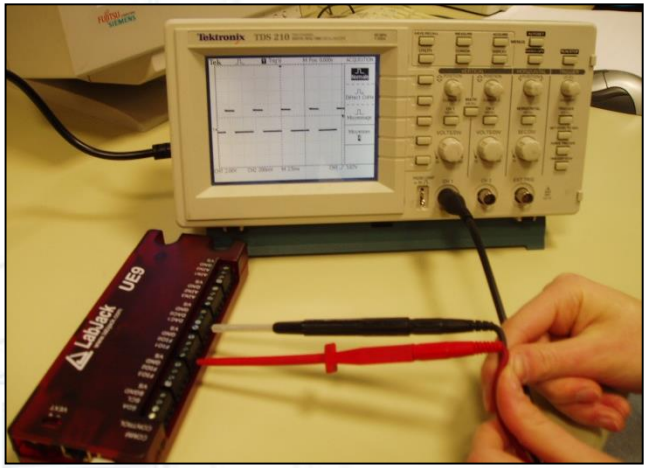
Carte d'interface

- 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 SSC-32, Parallax, Pololu, Labjack pour PC



Capteurs

- GPS, boussole, caméra…

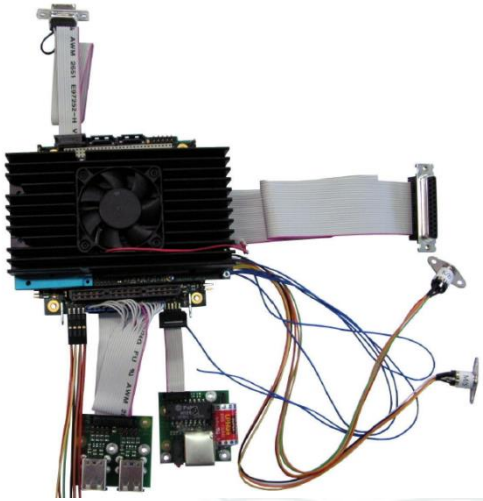
PC embarqué

- Intelligence du robot
  - Contient les programmes définissant le comportement du robot
- Exemple :
  - Smartphone / tablette

    Smartphone Samsung Galaxy S sous Android (avec GPS, boussole, caméra, Wi-Fi déjà intégrés)

PC embarqué

- Autres exemples :
  - HTPC (Home Theater PC)
  - EeePC 901 (netbook)
  - Mini ITX
  - PC/104
  - …



Computer form factors	
Name	Size (mm)
NUC	116.6 x 112 x 34.5
Compute Stick	103.3 x 12.5 x 37.6
Zotac Pico	66 x 19.2 x 115.2
eeePC 901	226 × 175.3 × 22.9
Mini TX	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 Wi-Fi USB, Wi-Fi intégré au smartphone…

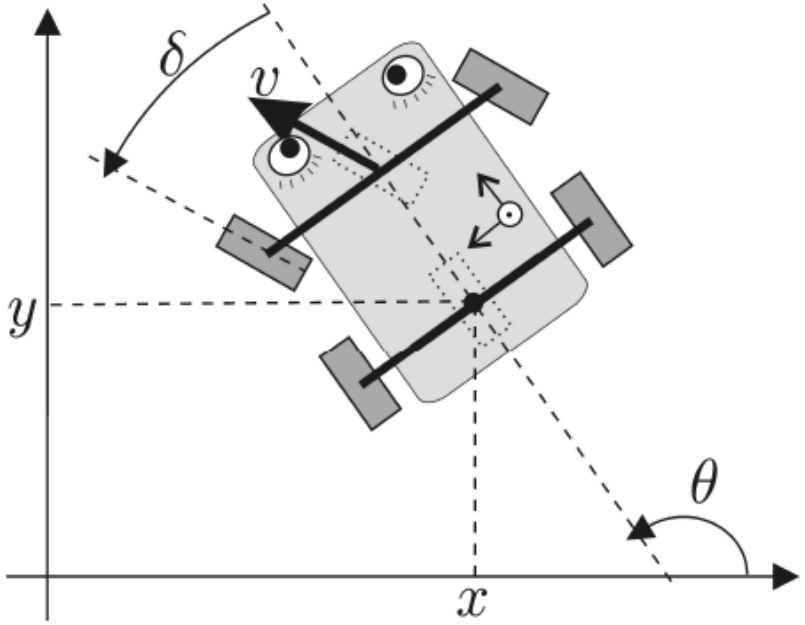
Equations d’état et régulation

Schéma du système

Modèle d'état du buggy et équations géométriques

Buggy : modèle de type voiture

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



$$\begin{aligned} v &= \alpha u_2 \\ \delta &= \beta u_1 \end{aligned}$$

L Distance entre les trains avant et arrière

Observateurs, mesures

Scénario : le buggy est dehors, capte le GPS précisément et a une boussole correcte

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

Remarques sur la boussole

- Sensible aux perturbations magnétiques dues aux objets métalliques de l'environnement proche (difficile à corriger mais on pourrait cartographier le champ magnétique)
- Sensible aux perturbations dues aux éléments constituant le robot (peut varier selon la vitesse des moteurs…). Les perturbations constantes peuvent cependant être facilement prises en compte

Remarques sur le GPS

- Ne fonctionne en général pas à l'intérieur (il faut qu'il ait une bonne « vue » des satellites dans le ciel)
- Il se peut qu'il donne des positions aberrantes lorsqu'il est à la limite de ne plus capter
- Temps de démarrage (« fix ») de plusieurs minutes variable selon les conditions

Régulation

- Si on suppose que d'une manière ou d'une autre on a une estimation de x,y,theta, on peut maintenant réfléchir à la commande pour suivre un cap ou aller à une position particulière…

PID

- Commande proportionnelle à l'erreur, à son intégrale ou à sa dérivée
- Censée marcher assez bien dans beaucoup de cas
- Voir Wikipedia PID (page en Anglais) pour un exemple simple de pseudo-code de régulation par PID et de méthode pour trouver les coefficients (Ziegler–Nichols method…)

PID

```
previous_error = setpoint - actual_position
integral = 0
start:
  error = setpoint - actual_position
  integral = integral + (error*dt)
  derivative = (error - previous_error)/dt
  output = (Kp*error) + (Ki*integral) + (Kd*derivative)
  previous_error = error
  wait(dt)
  goto start
```

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

- La boussole nous donne un angle au Nord en degrés θ
- Principe d'une 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 :

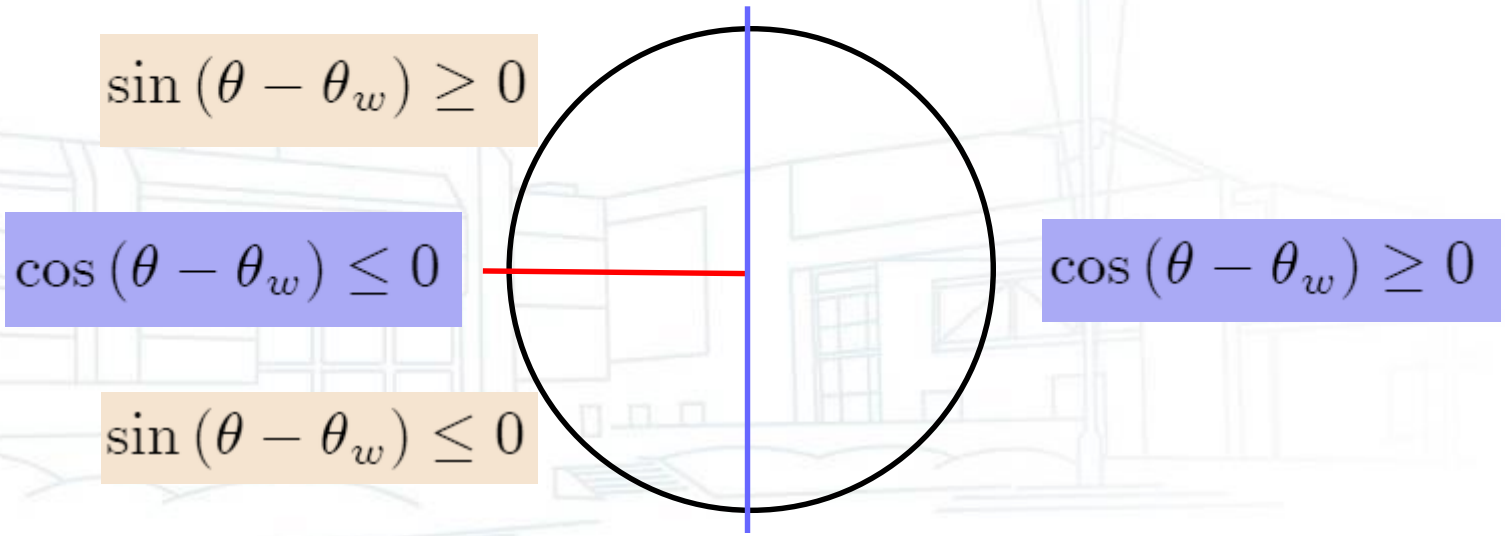
$$u_1 = K_p(\theta_w - \theta)$$
$$u_2 = u_w$$

- Attention aux problèmes de modulo 2π :
  - Utiliser des sin et cos par exemple
  - Voir aussi http://www.ensta-bretagne.fr/lebars/Share/fmod_360.zip)

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

- Exemple : si l'erreur de cap est $(\theta - \theta_w)$, une commande possible sans problèmes de modulo 2π peut être :

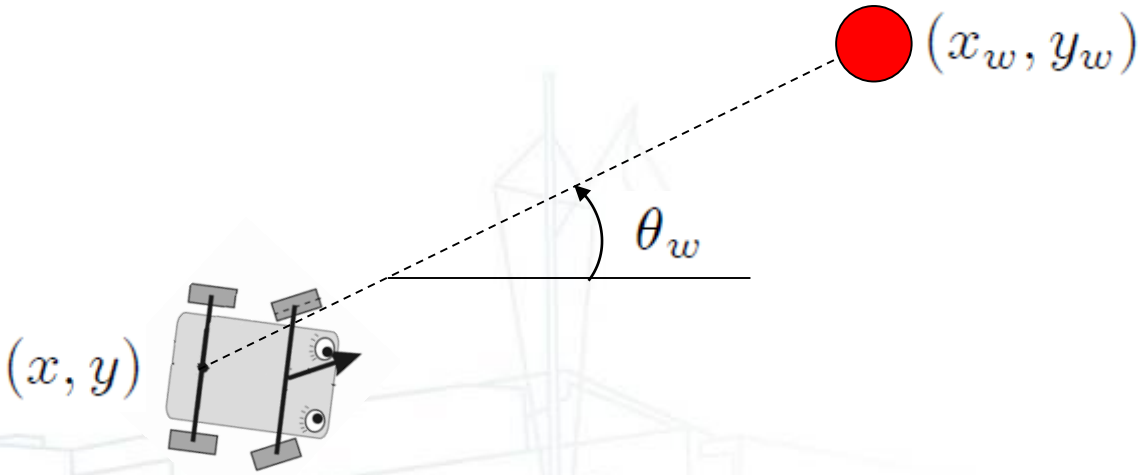
$$\delta = \begin{cases} \delta^{\max}.\sin(\theta - \theta_w) & \text{if } \cos(\theta - \theta_w) \geq 0 \\ \delta^{\max}.\text{sign}(\sin(\theta - \theta_w)) & \text{otherwise} \end{cases}$$



Suivi de waypoints GPS

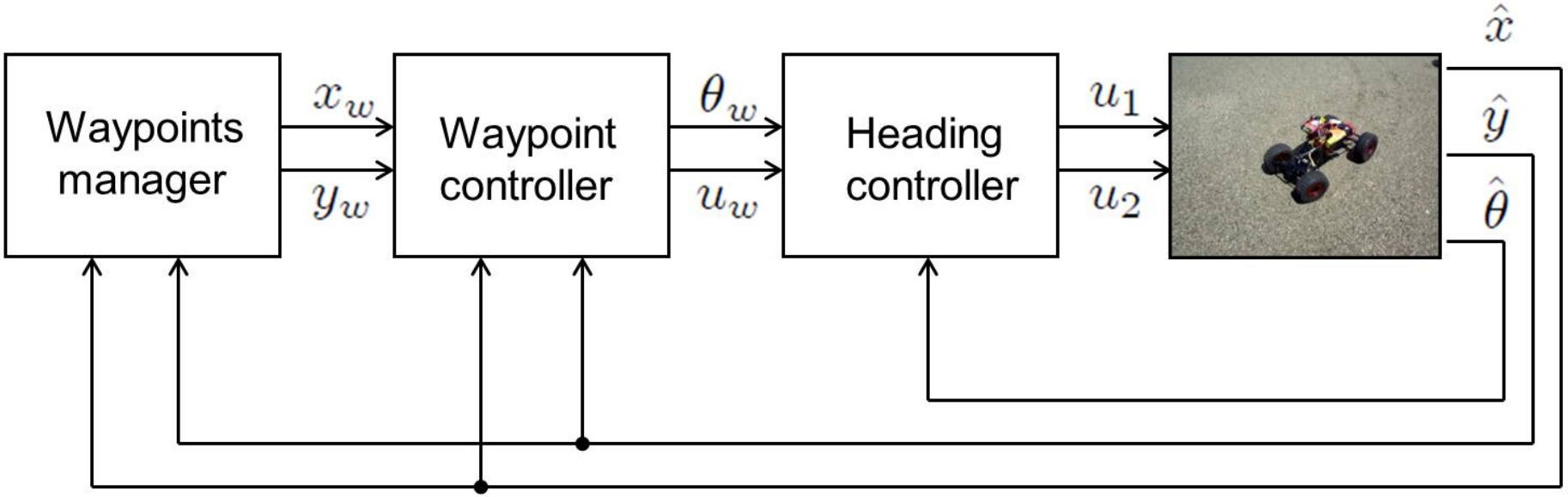
- On peut prendre pour cap voulu :

$$\theta_w = \arctan_2 \left(y - y_w, x - x_w \right)$$



- atan2 est une fonction MATLAB comme arctan, mais qui retourne un angle dans $[-\pi, \pi]$ au lieu de $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$

Schéma du système pour le suivi de waypoints GPS



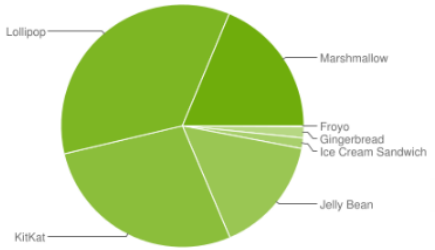
Android

Android, un système d'exploitation pour smartphones et tablettes

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

4 Smartphones sur 5
Autres : iOS, Windows Phone

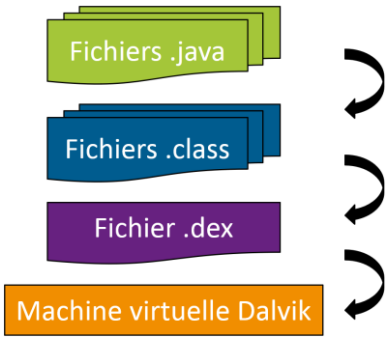
Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.5%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.4%
4.1.x	Jelly Bean	16	5.6%
4.2.x		17	7.7%
4.3		18	2.3%
4.4	KitKat	19	27.7%
5.0	Lollipop	21	13.1%
5.1		22	21.9%
6.0	Marshmallow	23	18.7%



Data collected during a 7-day period ending on September 5, 2016.
Any versions with less than 0.1% distribution are not shown.

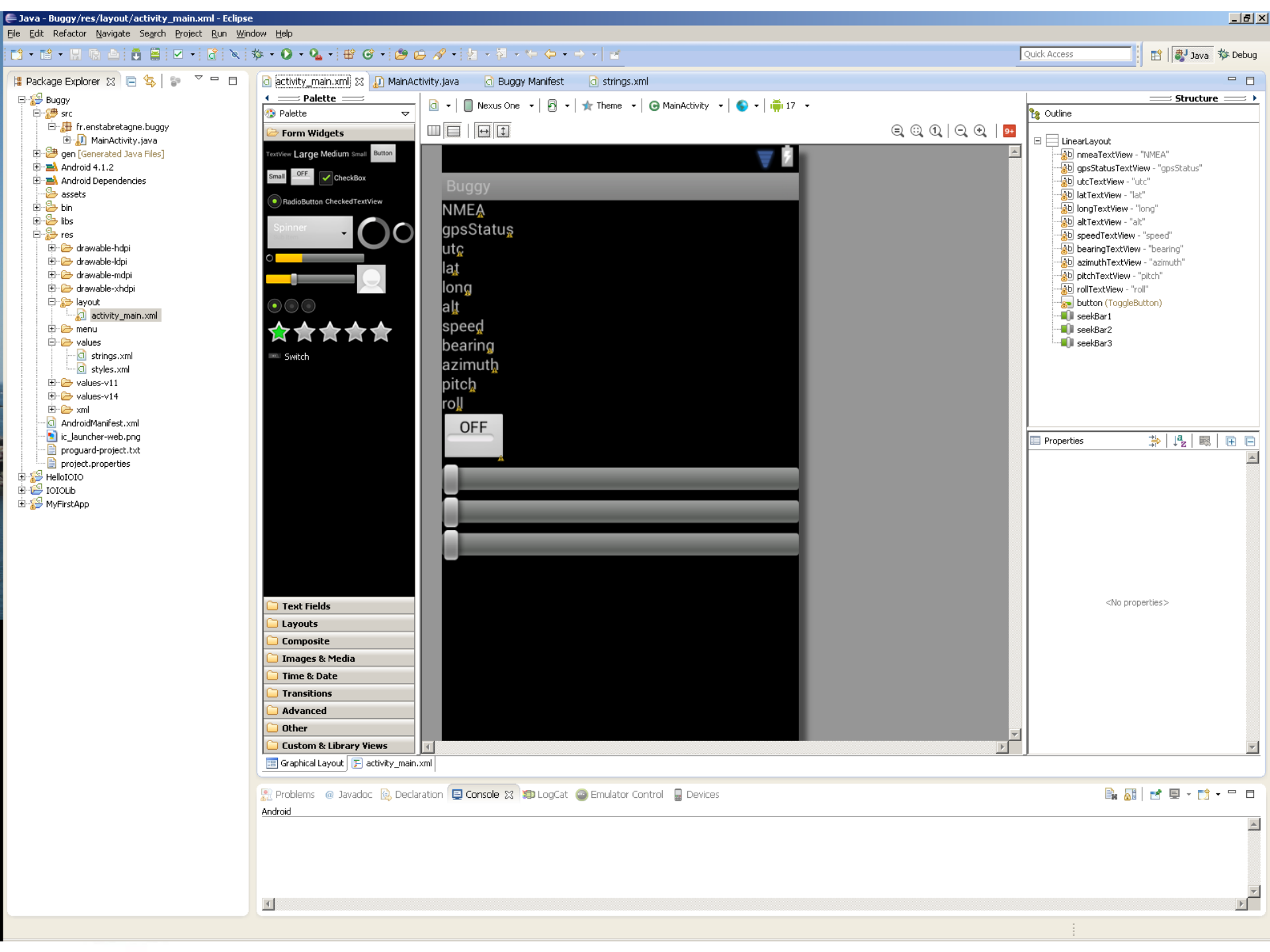
Android, un système d'exploitation pour smartphones et tablettes

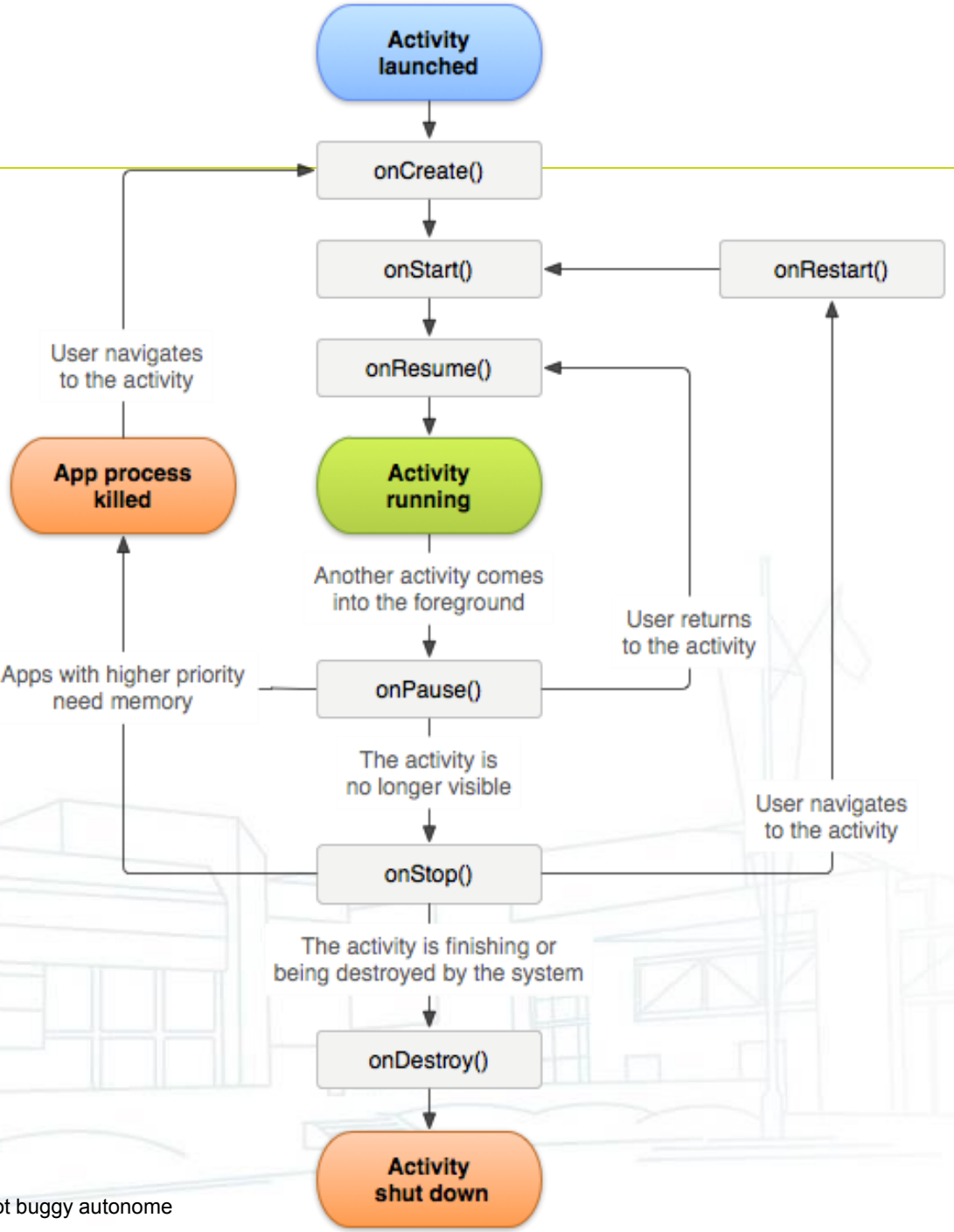
- Application Android
  - Android package **.apk** : archive contenant tous les fichiers nécessaires à l'application, le lancer installera l'application
  - Executable **.dex** : format optimisé d'executable regroupant les **.class**
  - **AndroidManifest.xml** : contient notamment les **permissions** et les **infos de versions d'Android** ciblées
  - Une application contient une **Activity** principale : classe correspondant à la fenêtre principale de l'application. Celle-ci peut provoquer l'ouverture d'autres fenêtres



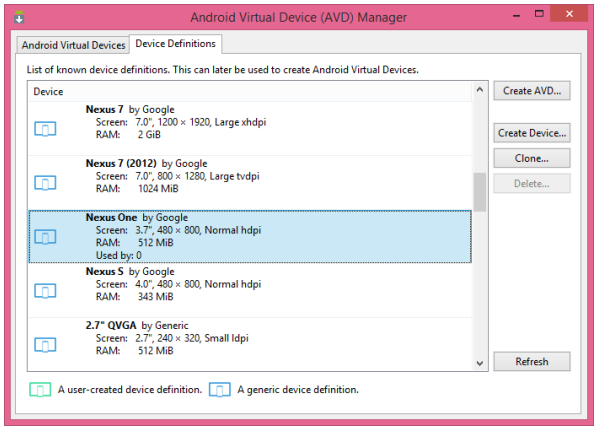
Android, un système d'exploitation pour smartphones et tablettes

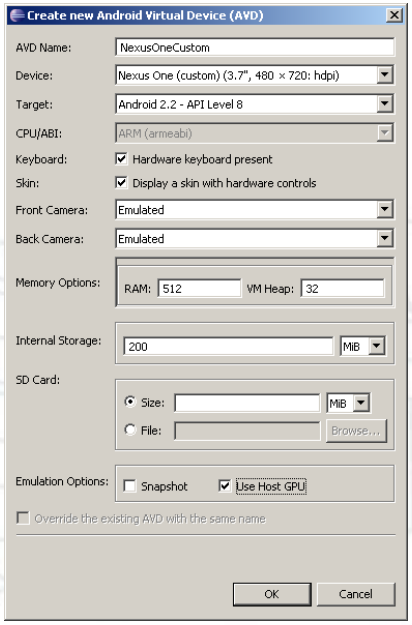
- Application Android
  - **Activity** : fenêtre décrite par classe Java+fichiers XML (voir **src/MainActivity.java**, **res/layout/activity_main.xml**, **res/values/strings.xml**)
  - Peut utiliser des services systèmes (**LocationManager**, **SensorsManager**…)
  - **AsyncTask** : thread
  - **Fragments** : petites fenêtres temporaires, boîtes de dialogue
  - **Toast** : popup temporaire
  - **Intent** : message pour demander une action à une autre **Activity**
  - **Bundle** : permet de passer des paramètres à l'action demandée par l'**Intent**

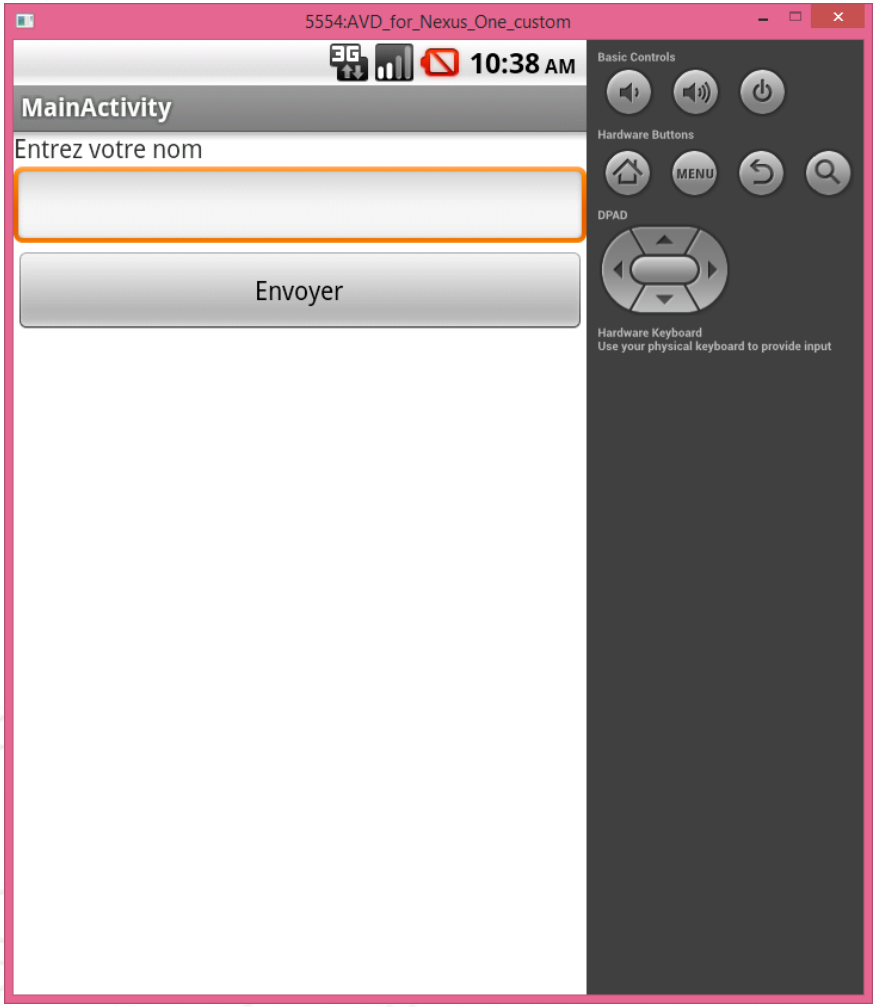

Java - Buggy/res/layout/activity_main.xml - Eclipse
File Edit Refactor Navigate Search Project Run Window Help
Quick Access
Java
Debug
Package Explorer
Buggy
src
fr.enstabretagne.buggy
MainActivity.java
gen [Generated Java Files]
Android 4.1.2
Android Dependencies
assets
bin
libs
res
drawable-hdpi
drawable-ldpi
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
HelloIOIO
IOIOLib
MyFirstApp
activity_main.xml
MainActivity.java
Buggy Manifest
strings.xml
Palette
Form Widgets
Text Fields
Layouts
Composite
Images & Media
Time & Date
Transitions
Advanced
Other
Custom & Library Views
Nexus One
Theme
MainActivity
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>
Graphical Layout
activity_main.xml
Problems
Javadoc
Declaration
Console
LogCat
Emulator Control
Devices
Android


Activity launched
onCreate()
onStart()
onRestart()
onResume()
Activity running
User navigates to the activity
App process killed
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Android, un système d'exploitation pour smartphones et tablettes







Android, un système d'exploitation pour smartphones 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

IOIO

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

IOIO

- 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
- Entrées-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