

BERNIER Benoît
BLANDEL Alexandre
DE CEITA Alex
STEPHANT Thomas

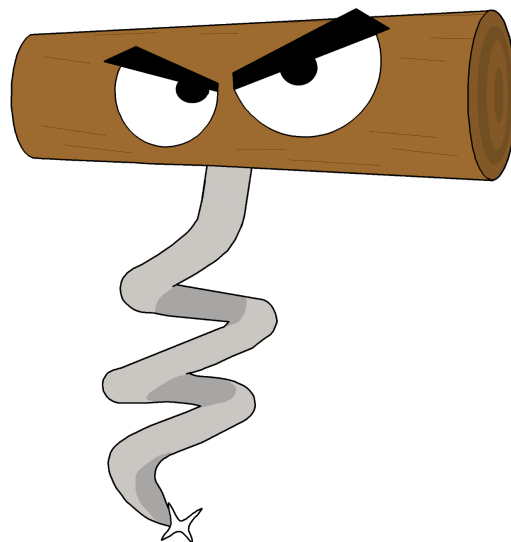


Délivrable WorkPackage 2

Simulation et paramétrage Pixhawk

30 mai 2018

Groupe : IoTireBouchon



I - Paramétrage du drone

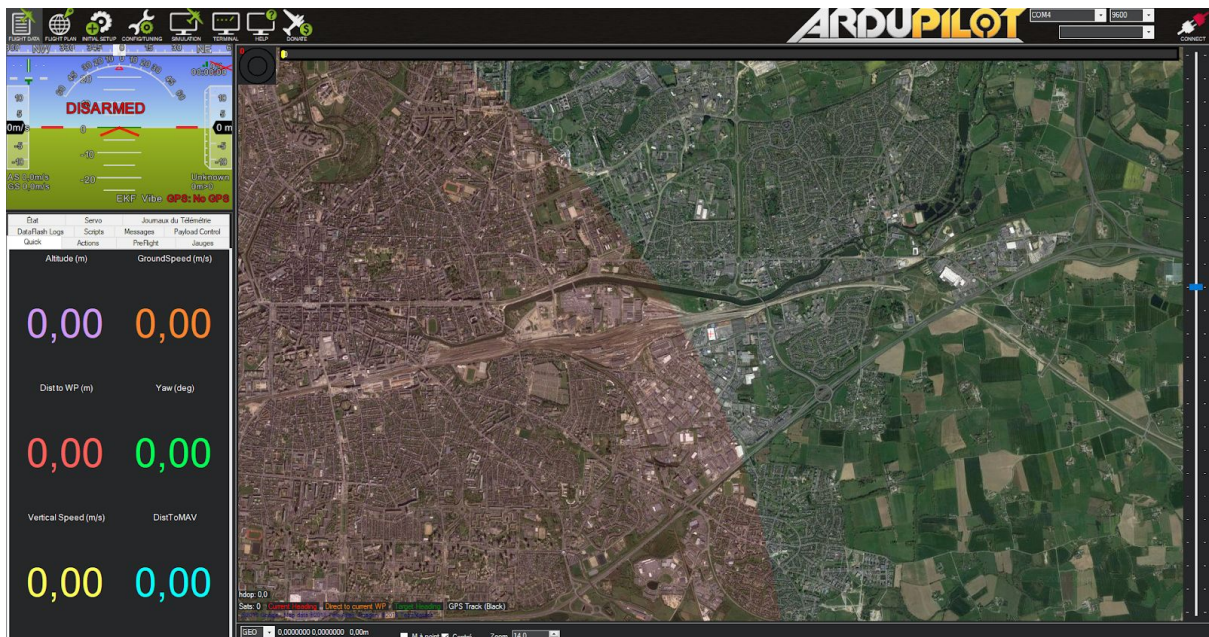
Partie 1 : Découverte de l'environnement

Dans cette première partie, nous allons présenter notre premier contact avec les différents logiciels permettant de paramétrer une carte de vol de drone. Cette phase est indispensable. En effet, le bon fonctionnement de notre drone dépend essentiellement de la configuration de la carte. Il faut donc être capable de comprendre l'environnement de travail (software) ainsi que les exigences de notre matériel (hardware et compatibilité) afin de pouvoir créer un drone fonctionnel.

1) Choix du logiciel de paramétrage : Mission Planner & APM

Nous avons choisi pour notre drone de topographie une carte Pixhawk 2.1 Cube car cette dernière supporte l'ajout de module externe tel que le LIDAR et elle permet également de récupérer des logs lors du vol dans un fichier (ce qui nous permet de consulter les données relevées à la fin du vol). Or, cette carte n'est pas livrée avec un utilitaire permettant sa configuration. Nous avons donc dû chercher un logiciel sur internet compatible avec les cartes Pixhawk.

Lors de nos différentes phases de recherches, nous avons déterminé deux potentiels logiciels permettant de configurer des cartes de vol : Mission Planner et APM Planner. Ces utilitaires sont tout d'abord gratuits et libres ce qui nous permet de faire l'économie de l'achat des licences. De plus, ces deux logiciels sont sensiblement les mêmes à une exception près: l'un fonctionne respectivement sous Windows et l'autre sous UNIX. Dans un premier temps, nous avons voulu utiliser la version sous UNIX mais nous nous sommes heurtés à des problèmes de compatibilités (notamment car APM n'est pas supporté par les récentes distributions de Ubuntu) et aussi des problèmes de drivers (la carte n'était pas reconnue). Nous avons donc décidé de nous tourner vers Mission Planner qui nous permis, après l'installation des drivers, de commencer notre travail sur la carte.



Interface de Mission Planner 1.3.52.10 sous Windows 10

Il est également important de relever que les dernières versions de Mission Planner ne sont pas stables et que nous avons dû faire plusieurs downgrade afin de trouver une version qui fonctionne avec notre carte (des messages d'erreurs logiciels apparaissent régulièrement). Nous avons donc essentiellement travaillé sous Mission Planner 1.3.52.10.

2) Première connexion de la carte sous Mission Planner

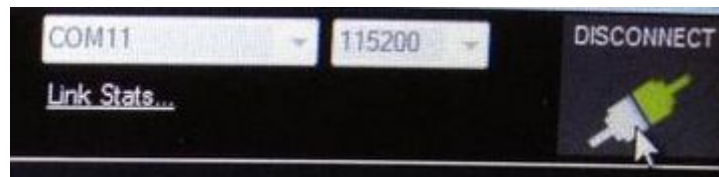
Tout d'abord, il faut savoir que la carte est connectée à l'ordinateur par un câble USB. L'ordinateur a donc besoin de télécharger les drivers associés (normalement installé lors de l'installation du logiciel) afin de récupérer les informations sur le port USB associé. Lorsque la carte est branchée sur l'ordinateur, vous devez donc être capable de la voir dans votre panneau de configuration de vos périphériques avec un numéro de port USB (COM1 ou COM2 ou COM3 ...). Récupérez ce numéro de port USB et rendez-vous sur votre interface Mission Planner (ou APM Planner) pour pouvoir configurer la carte de vol.



Connexion entre la carte de vol et Mission Planner

La partie en haut à droite de l'interface de Mission Planner est réservée à la connexion des cartes. Le premier menu déroulant représente le type de connexion entre l'ordinateur et la carte (protocole UDP ou TCP, port USB...). Dans notre cas, comme nous sommes connectés par câble USB, on choisit le port COM associé à notre carte. Le deuxième champ représente la vitesse de transmission, dans notre

cas nous avons mis 115200. Dans le dernier champs vous devriez voir le nom de la carte si elle a déjà été paramétré sur Mission Planner. Si les étapes se sont bien déroulées, votre carte de vol devrait émettre un son via le buzzer ou changer de couleur afin de signaler qu'elle est appareillée. De plus, l'utilitaire Mission planner vous signalera que la carte a bien été connectée.



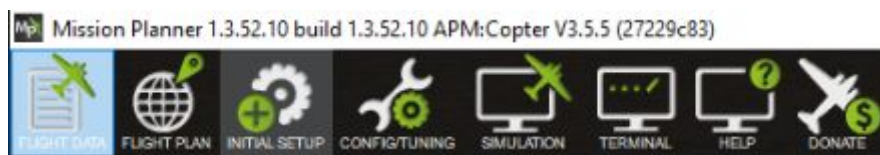
Appareillage de la carte à Mission Planner

Nous n'avons pas rencontré de gros problèmes lors de la phase d'appareillage. Le seul imprévu que nous avons relevé est l'oubli de la carte SD dans la carte de vol Pixhawk qui nous permettait pas de nous connecter ainsi que des problèmes de drivers ou de stabilités de logiciels.

3) Présentation de l'interface

Nous allons maintenant présenter l'interface de Mission Planner. L'objectif est de comprendre où se situe les panneaux de configuration ainsi que les informations importantes sur l'état de notre drone.

Tout d'abord, en haut à droite, on peut voir les différents menus de Mission Planner :



Présentation des menus de Mission Planner

- (1) Flight data : permet d'avoir la carte GPS ainsi que les différentes informations tels que l'inclinaison du drone, les positions géographiques, l'altitude... et également si il y a un problème de configuration ou de connexion (par exemple pas de GPS → no GPS).
- (2) Les parties flight plan et initial setup permettent de configurer les aspects importants ainsi les pièces principales du drone. C'est dans cette partie que l'on choisi le firmware de la carte de vol, le type de drone (tricopter, quadcopter... et le sens des branches) ou encore le compass (sens inclinaison) et les calibration de la manette.



Présentation des panneau de configuration de base de Mission Planner

- (3) Dans l'onglet configuratings, on est capable de paramétrer en détail les paramètres de notre drone. Les configurations sont affichées sous forme de listes avec des noms de variables. L'interface n'est pas très intuitive, il faut savoir ce qu'on cherche ainsi que tous les champs associés au terme à modifier afin de réussir la configuration. Par exemple, la configuration du GPS nécessite le changement de plusieurs variables.

Liste de variables associés à la configuration du GPS

Command	Value	Units	Options	Desc
EKF_GPS_TYPE	0		0:GPS 3D Vel and 2D Pos 1:GPS 2D vel and 2D pos 2:GPS 2D pos 3:No GPS	This controls use of GPS measurements - 0 = use 3D velocity & 2D position, 1 = use 2D velocity and 2D position, 2 = use 2D position, 3 = inhibit GPS use - this can be useful when flying with an optical flow sensor in an environment where GPS quality is poor and subject to large multipath errors.
GPS_AUTO_CONFIG	1		0:Disables automatic configuration 1:Enable automatic configuration	Controls if the autopilot should automatically configure the GPS based on the parameters and default settings
GPS_AUTO_SWITCH	1		0:Disabled 1:UseBest 2:Blend	Automatic switchover to GPS reporting best lock
GPS_BLEND_MASK	5			Determines which of the accuracy measures Horizontal position, Vertical Position and Speed are used to calculate the weighting on each GPS receiver when soft switching has been selected by setting GPS_AUTO_SWITCH to 2
GPS_BLEND_TC	10	s	5 0 30 0	Controls the slowest time constant applied to the calculation of GPS position and height offsets used to adjust different GPS receivers for steady state position differences.
GPS_DELAY_MS	0	ms	0 250	Controls the amount of GPS measurement delay that the autopilot compensates for. Set to zero to use the default delay for the detected GPS type.
GPS_DELAY_MS2	0	ms	0 250	Controls the amount of GPS measurement delay that the autopilot compensates for. Set to zero to use the default delay for the detected GPS type.
GPS_GNSS_MODE	0		0:Leave as currently configured 1:GPS-NoSBAS 3:GPS+SBAS 4:Galileo-NoSBAS 6:Galileo+SBAS 8:Beidou 51:GPS+IMES+QZSS+SBAS (Japan Only) 64:GLONASS 66:GLONASS+SBAS 67:GPS+GLONASS+SBAS	Bitmask for what GNSS system to use on the first GPS (all unchecked or zero to leave GPS as configured)
GPS_GNSS_MODE2	0		0:Leave as currently configured 1:GPS-NoSBAS 3:GPS+SBAS 4:Galileo-NoSBAS 6:Galileo+SBAS 8:Beidou 51:GPS+IMES+QZSS+SBAS (Japan Only) 64:GLONASS 66:GLONASS+SBAS 67:GPS+GLONASS+SBAS	Bitmask for what GNSS system to use on the second GPS (all unchecked or zero to leave GPS as configured)
GPS_HDOP_GOOD	140		100 900	GPS Hdop value at or below this value represent a good position. Used for pre-arm checks
GPS_INJECT_TO	127		0:send to first GPS 1:send to 2nd GPS 127:send to all	The GGS can send raw serial packets to inject data to multiple GPSes.
GPS_MIN_DGPS	100		0:Any 50:FloatRTK 100:IntegerRTK	Sets the minimum type of differential GPS corrections required before allowing to switch into DGPS mode.
GPS_MIN_ELEV	-100	deg	-100 90	This sets the minimum elevation of satellites above the horizon for them to be used for navigation. Setting this to -100 leaves the minimum elevation set to the GPS modules default.
GPS_NAVFILTER	8		0:Portable 2:Stationary 3:Pedestrian 4:Automotive 5:Sea 6:Airborne 1G 7:Airborne2G 8:Airborne4G	Navigation filter engine setting
GPS_POS1_X	0	m		X position of the first GPS antenna in body frame. Postive X is forward of the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_POS1_Y	0	m		Y position of the first GPS antenna in body frame. Postive Y is to the right of the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_POS1_Z	0	m		Z position of the first GPS antenna in body frame. Postive Z is down from the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_POS2_X	1	m		X position of the second GPS antenna in body frame. Postive X is forward of the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_POS2_Y	1	m		Y position of the second GPS antenna in body frame. Postive Y is to the right of the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_POS2_Z	1	m		Z position of the second GPS antenna in body frame. Postive Z is down from the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_RATE_MS	200	ms	50 200 100:10Hz 125 8Hz 200:5Hz	Controls how often the GPS should provide a position update. Lowering below 5Hz is not allowed
GPS_RATE_MS2	200	ms	50 200 100:10Hz 125 8Hz 200:5Hz	Controls how often the GPS should provide a position update. Lowering below 5Hz is not allowed
GPS_RAW_DATA	0		0:Ignore 1:Always log 2:Stop logging when disarmed (SBF only) 5:Only log every five samples (uBlox only)	Handles logging raw data; on uBlox chips that support raw data this will log RXM messages into datflash log; on Septentrio this will log on the equipment's SD card and when set to 2, the autopilot will try to stop logging after disarming and restart after arming
GPS_SAVE_CFG	0		0:Do not save config 1:Save config 2:Save only when needed	Determines whether the configuration for this GPS should be written to non-volatile memory on the GPS. Currently working for uBlox 6 series and above.
GPS_SBAS_MODE	2		0:Disabled 1:Enabled 2:NoChange	This sets the SBAS (satellite based augmentation system) mode if available on this GPS. If set to 2 then the SBAS mode is not changed in the GPS. Otherwise the GPS will be reconfigured to enable/disable SBAS. Disabling SBAS may be worthwhile in some parts of the world where an SBAS signal is available but the baseline is too long to be useful.
GPS_SBP_LOGMASK	-256		0:None (0x0000) -1:All (0xFFFF) -256:External only (0xF00)	Masked with the SBP msg_type field to determine whether SBR1/SBR2 data is logged
GPS_TYPE	2		0:None 1:AUTO 2:uBlox 3:MTK 4:MTK19 5:NMEA 6:SRF 7:HIL 8:SwifNav 9:UAVCAN 10:SBF 11:GSOF 12:QURT 13:ERB 14:MTK19S 15:uBlox	GPS type

Après avoir changer les paramètres de configuration, il faut ensuite appliquer les changements en utilisant le bouton Write. Il est également possible de charger / sauvegarder des fichiers de configuration. Cela nous a permis de travailler sur plusieurs machines.

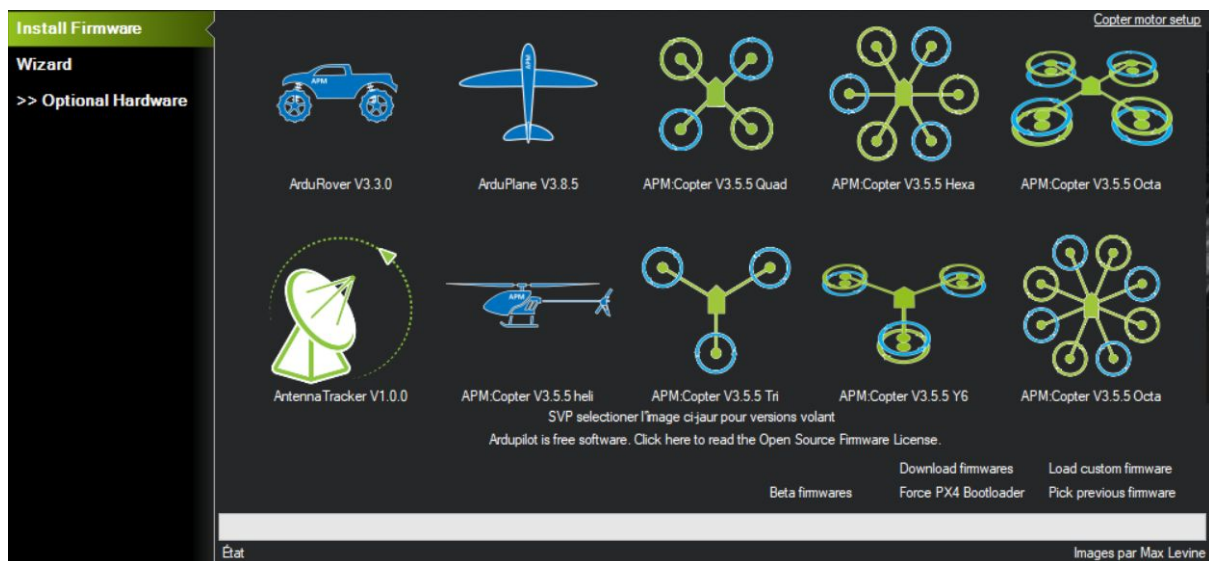
- (4) L'onglet Simulation, comme son nom l'indique permet de simuler un vol d'un drone autonome (avec une carte sur laquelle on place des balises à visiter). Nous n'avons pas utiliser cette fonction car notre drone n'a pour vocation d'être autonome et aussi car notre objectif est de s'entraîner à piloter le drone avec une manette RC. La simulation sera réalisé en passant par ROS et Gazebo. Cette partie sera abordée dans la seconde partie de ce rapport.
- (5) Le terminal intégré à Mission Planner permet de faire des tests de composants. ainsi que d'utiliser des commandes pour configurer le drone. Malheureusement, le terminal n'était pas utilisable dans cette version de Mission Planner. En effet, des messages d'erreurs nous empêchaient de travailler. Ces bugs sont également sur les autres versions de Mission Planner que nous avons utilisé.

Partie 2 : Configuration des paramètres globaux

Nous allons maintenant nous intéresser aux configurations que nous avons entré dans Mission Planner afin de pouvoir paramétrer notre drone. La première étape est d'ajouter un firmware dans la carte de vol.

1) Ajout du premier firmware

Avant de se connecter à la carte Pixhawk (avec le bouton Connect en haut à gauche), il faut insérer un firmware. Il ne faut pas choisir ce firmware au hasard, il doit à la fois être compatible avec la carte et correspondre au châssis souhaité. Ces firmwares n'ont pas les mêmes extensions en fonction du type de drone (.hex pour un hexacopteur par exemple). Si le firmware n'est pas adapté à la carte, cela risque d'endommager la carte.



Installation du firmware sur la carte de vol

Dans notre cas, comme nous voulons avoir un quadcopter, nous avons choisi le firmware proposé par défaut dans Mission Planner : APM:Copter V3.5.5 Quad. Nous sommes à présent capable de configurer notre drone.

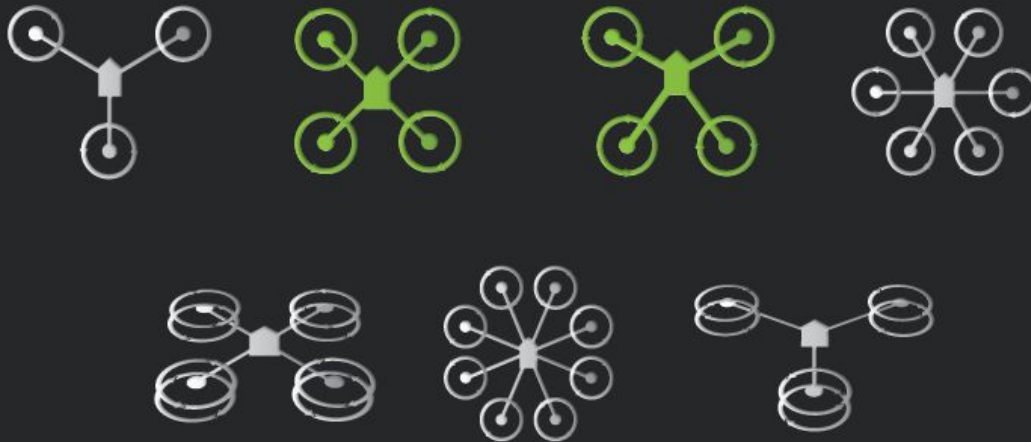
2) Première installation

La première installation se fait normalement relativement vite. En effet, Mission Planner propose un Wizard qui permet de configurer les pièces principales du drone.

Select your frame type

UAV SELECTION: COPTER

Please select your frame type from below, so we can upload the correct firmware to your autopilot



<< Back

Next >>

Progress... 2 of 16

Choix du type de châssis dans le Wizard

Calibrate your Accelerometer

ACCELEROMETER CALIBRATION

Here we will calibrate the minimum and maximum values seen by your accelerometer. Please click start and follow the

Start

Continue



<< Back

Next >>

Progress... 5 of 16

Calibration du Compas

Les pièces principales configurées dans le Wizard de Mission Planner sont : le type de châssis, le compas (accéléromètre), le GPS, le type de batterie, le type d'ESC, le LIDAR (si il y en a un → optionnel) et enfin la manette RC. Ces configurations prennent quelques minutes si le matériel est compatible et si on suit la documentation des matériels et de Mission Planner.

Partie 3 : Configuration et test des composants

Nous avons rencontré quelques problèmes lors de la configuration des différents composants. Pour les résoudre, nous avons utilisé une seconde carte, une Pixhawk 1 (la première génération de la carte) ainsi que le matériel d'autres groupes. L'objectif était de vérifier notre configuration et de voir si il y avait des problèmes de matériels.

Dans cette partie, nous allons présenter les étapes de configuration du GPS, des moteurs et du LIDAR. Ces composants ont nécessité beaucoup de temps et de recherche afin de les faire fonctionner correctement. Nous allons présenter comment nous avons configuré nos matériels et comment nous avons résolu nos problèmes.

1) Installation, configuration et test du GPS

Tout d'abord, nous avons mis le GPS sur le port GPS2 de notre carte de vol. Nous avons utilisé l'adaptateur fourni avec la carte de vol pour le connecter. En effet, la connectique pour le Pixhawk n'est pas la même que les autres cartes de vol ce qui limite l'utilisation de composants.



Connexion du GPS sur la carte de vol

Sur le port GPS 1, nous avons connecté le switch qui permet d'armer les moteurs. Le switch est composé d'une deuxième partie qui est censé être relié au GPS. Dans notre cas, ce câble n'est pas compatible avec la carte. Nous avons donc câblé notre GPS dans le port GPS2.

Nous avons ensuite suivi les instructions de l'installation rapide de Mission Planner. Le GPS n'était toujours pas détecté par Mission Planner. Nous avons donc dû modifier les paramètres dans l'onglet configuratings. Voici la liste des paramètres que nous avons modifié (en utilisant la spécification de la carte de vol et du GPS).

EK2_GPS_TYPE	0		0:GPS 3D Vel and 2D Pos 1:GPS 2D vel and 2D pos 2:GPS 2D pos 3:No GPS	This controls use of GPS measurements : 0 = use 3D velocity & 2D position, 1 = use 2D velocity and 2D position, 2 = use 2D position, 3 = inhibit GPS use - this can be useful when flying with an optical flow sensor in an environment where GPS quality is poor and subject to large multipath errors.
GPS_AUTO_CONFIG	1		0:Disables automatic configuration 1:Enable automatic configuration	Controls if the autopilot should automatically configure the GPS based on the parameters and default settings
GPS_AUTO_SWITCH	1		0:Disabled 1:UseBest 2:Blend	Automatic switchover to GPS reporting best lock
GPS_BLEND_MASK	5			Determines which of the accuracy measures Horizontal position, Vertical Position and Speed are used to calculate the weighting on each GPS receiver when soft switching has been selected by setting GPS_AUTO_SWITCH to 2
GPS_BLEND_TC	10	s	5.0 30.0	Controls the slowest time constant applied to the calculation of GPS position and height offsets used to adjust different GPS receivers for steady state position differences.
GPS_DELAY_MS	0	ms	0 250	Controls the amount of GPS measurement delay that the autopilot compensates for. Set to zero to use the default delay for the detected GPS type.
GPS_DELAY_MS2	0	ms	0 250	Controls the amount of GPS measurement delay that the autopilot compensates for. Set to zero to use the default delay for the detected GPS type.
GPS_GNSS_MODE	0		0:Leave as currently configured 1:GPS-NoSBAS 3:GPS+SBAS 4:Galileo-NoSBAS 6:Galileo+SBAS 8:Beidou 51:GPS+IMES+QZSS+SBAS (Japan Only) 54:GLONASS 66:GLONASS+SBAS 67:GPS+GLONASS+SBAS	Bitmask for what GNSS system to use on the first GPS (all unchecked or zero to leave GPS as configured)
GPS_GNSS_MODE2	0		0:Leave as currently configured 1:GPS-NoSBAS 3:GPS+SBAS 4:Galileo-NoSBAS 6:Galileo+SBAS 8:Beidou 51:GPS+IMES+QZSS+SBAS (Japan Only) 54:GLONASS 66:GLONASS+SBAS 67:GPS+GLONASS+SBAS	Bitmask for what GNSS system to use on the second GPS (all unchecked or zero to leave GPS as configured)
GPS_HDOP_GOOD	140		100 900	GPS Hdop value at or below this value represent a good position. Used for pre-arm checks
GPS_INJECT_TO	127		0:send to first GPS 1:send to 2nd GPS 127:send to all	The GGS can send raw serial packets to inject data to multiple GPSes.
GPS_MIN_DGPS	100		0:Any 50:RoarRTK 100:IntegerRTK	Sets the minimum type of differential GPS corrections required before allowing to switch into DGPS mode.
GPS_MIN_ELEV	-100	deg	-100 90	This sets the minimum elevation of satellites above the horizon for them to be used for navigation. Setting this to -100 leaves the minimum elevation set to the GPS modules default.
GPS_NAVFILTER	8		0:Portable 2:Stationary 3:Pedestrian 4:Automotive 5:Sea 6:Airborne1G 7:Airborne2G 8:Airborne4G	Navigation filter engine setting
GPS_POS1_X	0	m		X position of the first GPS antenna in body frame. Positive X is forward of the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_POS1_Y	0	m		Y position of the first GPS antenna in body frame. Positive Y is to the right of the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_POS1_Z	0	m		Z position of the first GPS antenna in body frame. Positive Z is down from the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_POS2_X	1	m		X position of the second GPS antenna in body frame. Positive X is forward of the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_POS2_Y	1	m		Y position of the second GPS antenna in body frame. Positive Y is to the right of the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_POS2_Z	1	m		Z position of the second GPS antenna in body frame. Positive Z is down from the origin. Use antenna phase centroid location if provided by the manufacturer.
GPS_RATE_MS	200	ms	50 200100:10Hz 125:8Hz 200:5Hz	Controls how often the GPS should provide a position update. Lowering below 5Hz is not allowed
GPS_RATE_MS2	200	ms	50 200100:10Hz 125:8Hz 200:5Hz	Controls how often the GPS should provide a position update. Lowering below 5Hz is not allowed
GPS_RAW_DATA	0		0:Ignore 1:Always log 2:Stop logging when disarmed (SBF only) 5:Only log every five samples (uBlox only)	Handles logging raw data: on uBlox chips that support raw data this will log RXM messages into dataflash log; on Septentrio this will log on the equipment's SD card and when set to 2, the autopilot will try to stop logging after disarming and restart after arming
GPS_SAVE_CFG	0		0:Do not save config 1:Save config 2:Save only when needed	Determines whether the configuration for this GPS should be written to non-volatile memory on the GPS. Currently working for UBlox 6 series and above.
GPS_SBAS_MODE	2		0:Disabled 1:Enabled 2:NoChange	This sets the SBAS (satellite based augmentation system) mode if available on this GPS. If set to 2 then the SBAS mode is not changed in the GPS. Otherwise the GPS will be reconfigured to enable/disable SBAS. Disabling SBAS may be worthwhile in some parts of the world where an SBAS signal is available but the baseline is too long to be useful.
GPS_SBP_LOGMASK	-256		0:None (0x0000) -1:All (0xFFFF) -256:External only (0xFF00)	Masked with the SBP msg_type field to determine whether SBR1/SBR2 data is logged
GPS_TYPE	2		0:None 1:AUTO 2:uBlox 3:MTK 4:MTK19 5:NMEA 6:SRF 7:HIL 8:SwiftNav 9:UAVCAN 10:SBF 11:GSOF 12:QURT 13:ERB 14:MAV 15:NOVA	GPS type

De plus, nous avons eu des problèmes d'association entre le GPS et les satellites. Le drone n'arrivait pas à se raccrocher à un satellite. La solution a été de sortir hors des locaux (il y avait 2 étages au dessus de nous en béton) et de s'écarter des bâtiments.

Voici une capture de la carte GPS que nous avons pris près du bassin en face de l'Istic.



2) Installation, configuration et test des moteurs

Après avoir configuré notre GPS, nous avons testé les moteurs afin de pouvoir les fixer sur notre châssis. Nous avons également modifié les paramètres des ESC à partir de la documentation.

ESC Calibration (AC3.3+)

Calibrate ESCs

Remove Props!

After pushing this button:

- Disconnect USB and battery
- Plug in battery
- when LEDs flash, push Safety Switch (if present)
- ESCs should beep as they are calibrated
- restart flight controller normally

ESC Type: Normal

Output PWM Min: 0 Leave as 0 to use RX input range

Output PWM Max: 0 Leave as 0 to use RX input range

Spin when Armed: 0.220 speed when motors are armed but throttle is at zero (idle)

Spin minimum: 0.300 minimum speed of motors while in flight (slightly higher than "Spin when Armed")

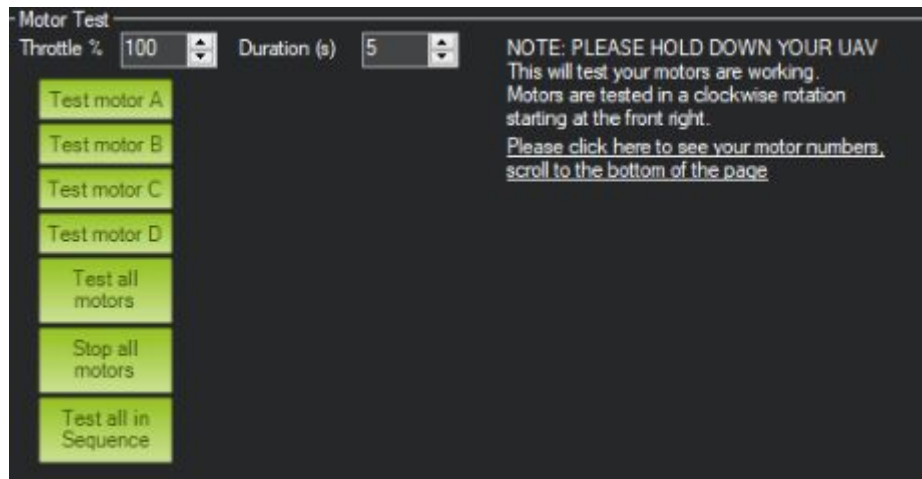
Spin Maximum: 0.950 maximum speed of motors while in flight (almost all escs have a deadzone at the top)

Paramètres liés aux ESC

COMPASS_MOT_X	0	mGauss/A	-1000 1000	Multiplied by the current throttle and added to the compass's x-axis values to compensate for motor interference (Offset per Amp or at Full Throttle)
COMPASS_MOT_Y	0	mGauss/A	-1000 1000	Multiplied by the current throttle and added to the compass's y-axis values to compensate for motor interference (Offset per Amp or at Full Throttle)
COMPASS_MOT_Z	0	mGauss/A	-1000 1000	Multiplied by the current throttle and added to the compass's z-axis values to compensate for motor interference (Offset per Amp or at Full Throttle)
MOT_BAT_CURR_MAX	0	A	0 200	Maximum current over which maximum throttle is limited (0 = Disabled)
MOT_BAT_CURR_TC	5	s	0 10	Time constant used to limit the maximum current
MOT_BAT_VOLT_MAX	0	V	6 35	Battery voltage compensation maximum voltage (voltage above this will have no additional scaling effect on thrust). Recommend 4.4 * cell count, 0 = Disabled
MOT_BAT_VOLT_MIN	0	V	6 35	Battery voltage compensation minimum voltage (voltage below this will have no additional scaling effect on thrust). Recommend 3.5 * cell count, 0 = Disabled
MOT_HOVER_LEARN	2		0:Disabled 1:Learn 2:LearnAndSave	Enable/Disable automatic learning of hover throttle
MOT_PWM_MAX	0	PWM	0 2000	This sets the max PWM value in microseconds that will ever be output to the motors, 0 = use input RC3_MAX
MOT_PWM_MIN	0	PWM	0 2000	This sets the min PWM output value in microseconds that will ever be output to the motors, 0 = use input RC3_MIN
MOT_PWM_TYPE	0		0:Normal 1:OneShot 2:OneShot125 3:Brushed	This selects the output PWM type, allowing for normal PWM continuous output, OneShot or brushed motor output
MOT_SAFE_DISARM	0		0:PWM enabled while disarmed 1:PWM disabled while disarmed	Disables motor PWM output when disarmed
MOT_SPIN_ARM	0.22		0.0:Low 0.1:Default 0.2:High	Point at which the motors start to spin expressed as a number from 0 to 1 in the entire output range. Should be lower than MOT_SPIN_MIN.
MOT_SPIN_MAX	0.95		0.9:Low 0.95:Default 1.0:High	Point at which the thrust saturates expressed as a number from 0 to 1 in the entire output range
MOT_SPIN_MIN	0.3		0.0:Low 0.15:Default 0.3:High	Point at which the thrust starts expressed as a number from 0 to 1 in the entire output range. Should be higher than MOT_SPIN_ARM.
MOT_SPOOL_TIME	0.5	s	0 2	Time in seconds to spool up the motors from zero to min throttle.
MOT_THST_EXPO	0.65		0 25 0.8	Motor thrust curve exponent (from 0 for linear to 1.0 for second order curve)
MOT_THST_HOVER	0.35		0 2 0.8	Motor thrust needed to hover expressed as a number from 0 to 1
MOT_YAW_HEADROOM	200	PWM	0 500	Yaw control is given at least this pwm in microseconds range
THROW_MOT_START	0		0:Stopped 1:Running	Used by THROW mode. Controls whether motors will run at the speed set by THR_MIN or will be stopped when armed and waiting for the throw.

Paramètres liés aux moteurs

Dans un second temps, nous avons testé les moteurs via l'interface de test de Mission Planner. L'objectif était de voir si les moteurs fonctionnaient individuellement et ensemble.



Interface de test des moteurs

Nous avons remarqué que les moteurs ne fonctionnent qu'au dessus de 25%.

3) Installation, configuration et test du LIDAR

Enfin, à la réception de notre LIDAR, nous avons remarqué que le port dédié au LIDAR était le même que celui déjà utilisé par le GPS sur la carte Pixhawk. Nous avons donc essayé de changer les câbles afin de mettre le GPS sur un autre port (comme on a pu voir sur d'autres cartes de vol). L'objectif est d'utiliser le port I2C ou le port télémétrique afin de récupérer les données de notre LIDAR.

Voici les configurations du LIDAR dans l'onglet configuratings.

EK2_RNG_USE_HGT	-1	%	-1.70	The range finder will be used as the primary height source when below a spe
RNGFND_MAX_CM	1200	cm		Maximum distance in centimeters that rangefinder can reliably read
RNGFND_MIN_CM	20	cm		Minimum distance in centimeters that rangefinder can reliably read

Configuration de la distance minimal et maximal du LIDAR (spécification constructeur)

RNGFND_TYPE	0	0:None 1:Analog 2:MaxbotixI2C 3:LidarLiteV2-I2C 5:PX4-PWM 6:BBB-PRU 7:LightWareI2C 8:LightWareSerial 9:Bebop 10:MAVLink 11:uLanding 12:LeddarOne 13:MaxbotixSerial 14:TeraRangerI2C 15:LidarLiteV3-I2C 16:VL53L0X	What type of rangefinder device that is connected
-------------	---	---	---

Choix du type du LIDAR (dans notre cas Pixhawk → PX4)

Nous avons comme erreur "Bad Lidar Health". Après recherches, nous nous sommes rendus compte que par défaut, le LIDAR envoyait les données sous cette forme :

Byte0 -1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x59 59	Dist_L	Dist_H	Strength_L	Strength_H	Mode	0x00	Checksum
Data code explanation							
Byte0	0x59, frame header, same for each frame						
Byte1	0x59, frame header, same for each frame						
Byte2	Dist_L distance value lower by 8 bits						
Byte3	Dist_L distance value higher by 8 bits						
Byte4	Strength_L low 8 bits						
Byte5	Strength_L high 8 bits						
Byte6	Mode, distance mode, represented respectively by 02 (short distance) and 07 (long distance), automatically switchable by default.						
Byte7	Spare byte, 00 by default						
Byte8	Checksum is the low 8 bits of the cumulative sum of the numbers of the first 8 bytes.						

Nous devons passer en "pixhawk data format" : *The data output is in the format of character string and its unit is m(meter). For example, if the measurement distance is 1.21m, the string 1.21 will be output, followed by the escape character \r\n.*

Pour ce faire, nous pouvons utiliser l'utilitaire proposé par Benewake et entrer une ligne de commande afin de passer dans ce mode. Nous devons pour cela acheter un convertisseur TTL/USB, que nous n'avons pas. Nous avons donc mis en suspens le travail avec le LIDAR.

II - Simulation du drone

1) Installer ROS indigo (compatible Ubuntu 14.x ET Gazebo 2)

<http://wiki.ros.org/indigo/Installation/Ubuntu>

a- sources

```
> sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >
/etc/apt/sources.list.d/ros-latest.list'
```

b- keys

```
> sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key
421C365BD9FF1F717815A3895523BAEEB01FA116
```

c- installation

```
> sudo apt-get update
> sudo apt-get install ros-indigo-desktop-full
```

2) Configurer ROS

a- initialiser rosdep

```
> sudo rosdep init
> rosdep update
```

b- rosinstall

```
> sudo apt-get install python-rosinstall
```

c- environnement setup

```
> echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
> source ~/.bashrc
```

```
> mkdir -p ~/catkin_ws/src
> cd ~/catkin_ws/
> catkin_make
```

```
> source devel/setup.bash
```

fermez le terminal, ouvrez en un autre.

3) Installer Gazebo 2.x

a- installer

```
> curl -sSL http://get.gazebosim.org | sh
```

b- maj

il est possible que la version installée soit inférieure à 2.2.5. le vérifier en ouvrant gazebo (tapez gazebo dans le terminal), aller dans help, about.

Le cas échéant, tapez :

```
> sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu trusty main" > /etc/apt/sources.list.d/gazebo-latest.list'
```

```
> wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -
```

puis effectuez une maj de ubuntu

```
> sudo apt-get update && sudo apt-get upgrade
```

c- prise en main rapide

https://www.google.fr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwikvfrghc7ZAhVBU1AKHfQqCgUQjBAINDAC&url=http%3A%2F%2Fgazebo.org%2Ftutorials%3Ftut%3Dquick_start&usg=AOvVaw0d9Vlq2mM7FSKmAyN4SZI6

4) Lier ROS et GAZEBO

http://gazebosim.org/tutorials?tut=ros_installing

a- > sudo apt-get install ros-indigo-gazebo-ros-pkgs ros-indigo-gazebo-ros-control

b- > rosrn gazebo_ros gazebo

c- s'assurer que > rostopic list (dans un autre terminal) donne entre autre

```
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/set_link_state
/gazebo/set_model_state
```

d- s'assurer que > rosservice list donne entre autre

```
/gazebo/apply_body_wrench
/gazebo/apply_joint_effort
/gazebo/clear_body_wrenches
/gazebo/clear_joint_forces
/gazebo/delete_model
/gazebo/get_joint_properties
/gazebo/get_link_properties
/gazebo/get_link_state
/gazebo/get_loggers
/gazebo/get_model_properties
/gazebo/get_model_state
/gazebo/get_physics_properties
/gazebo/get_world_properties
/gazebo/pause_physics
/gazebo/reset_simulation
/gazebo/reset_world
/gazebo/set_joint_properties
/gazebo/set_link_properties
/gazebo/set_link_state
/gazebo/set_logger_level
/gazebo/set_model_configuration
/gazebo/set_model_state
/gazebo/set_parameters
/gazebo/set_physics_properties
/gazebo/spawn_gazebo_model
/gazebo/spawn_sdf_model
/gazebo/spawn_urdf_model
/gazebo/unpause_physics
/rosout/get_loggers
```

/rosout/set_logger_level

5) Prise en main ROS+GAZEBO

a- Infos, détails, ressources ROS (FR) :

<http://generationrobots.developpez.com/tutoriels/presentation-robot-operating-system/>

b- Tuto TurtleBot

<http://wiki.ros.org/turtlebot/Tutorials/indigo/Turtlebot%20Installation>

```
> sudo apt-get install ros-indigo-turtlebot ros-indigo-turtlebot-apps ros-indigo-turtlebot-interactions  
ros-indigo-turtlebot-simulator ros-indigo-kobuki-ftdi ros-indigo-rocon-remocon ros-indigo-rocon-qt-library  
ros-indigo-ar-track-alvar-msgs
```

```
> roslaunch turtlebot_ [taper tab pour voir les options]  
> roslaunch turtlebot_ gazebo turtlebot_world.launch
```

(roslaunch lance le fichier launch et turtlebot_gazebo et le nom du package ou se trouve le fichier world.launch)

Le lancement de gazebo va prendre du temps pour initialiser le world

pour controller le turtlebot

```
>roslaunch turtlebot_teleop keyboard_teleop.launch
```

5bis) [facultatif] Tutos ROS en cas de problèmes

tuto video ROS + turtle bot

<https://www.youtube.com/watch?v=9U6GDonGFHw>

tuto vidéo ROS (FR) :

<https://www.youtube.com/watch?v=aOqCcOGFh2s>

tuto vidéo ROS :

<https://www.youtube.com/watch?v=MD255BS0YH4>

Autres outils pour ROS

<http://jenkins.ros.org/>

6) Simulation de drones

```
> git clone https://github.com/tu-darmstadt-ros-pkg/hector_quadrotor.git  
> sudo apt-get install ros-hydro-hector-quadrotor-demo  
> sudo apt-get install ros-indigo-teleop-twist-keyboard  
> rosrn teleop_twist_keyboard teleop_twist_keyboard.py
```

http://wiki.ros.org/hector_quadrotor

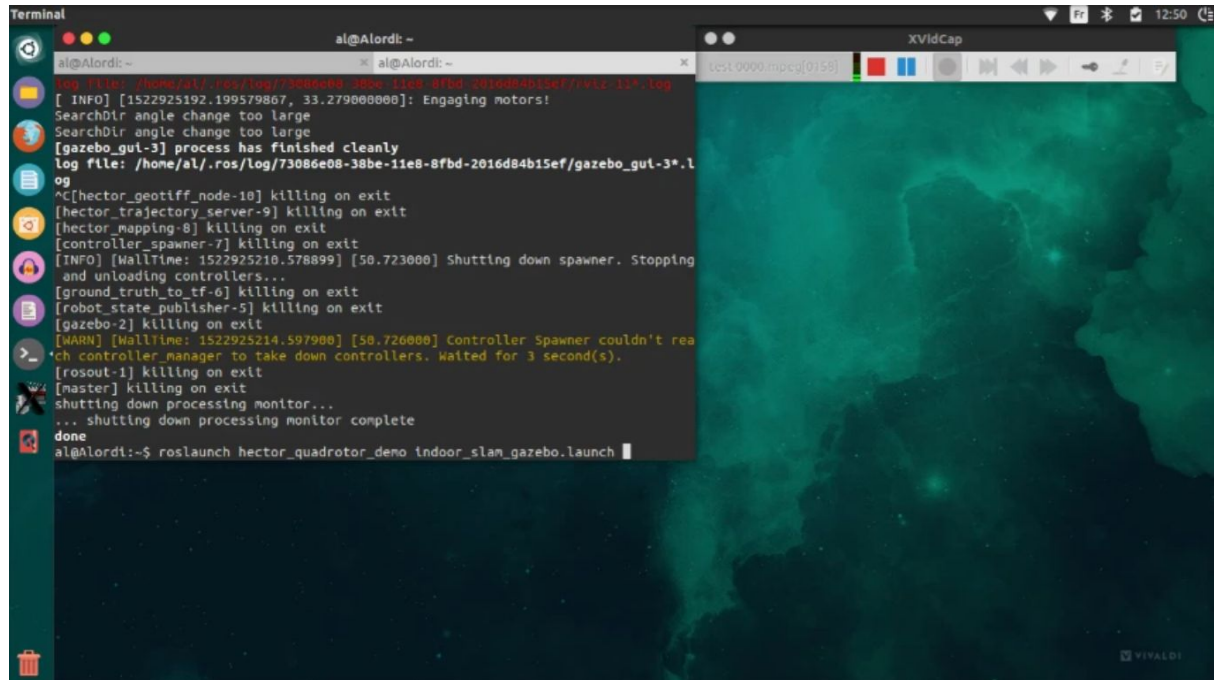
http://wiki.ros.org/tum_simulator

Encore un autre DRONE : https://github.com/wilselby/ROS_quadrotor_simulator

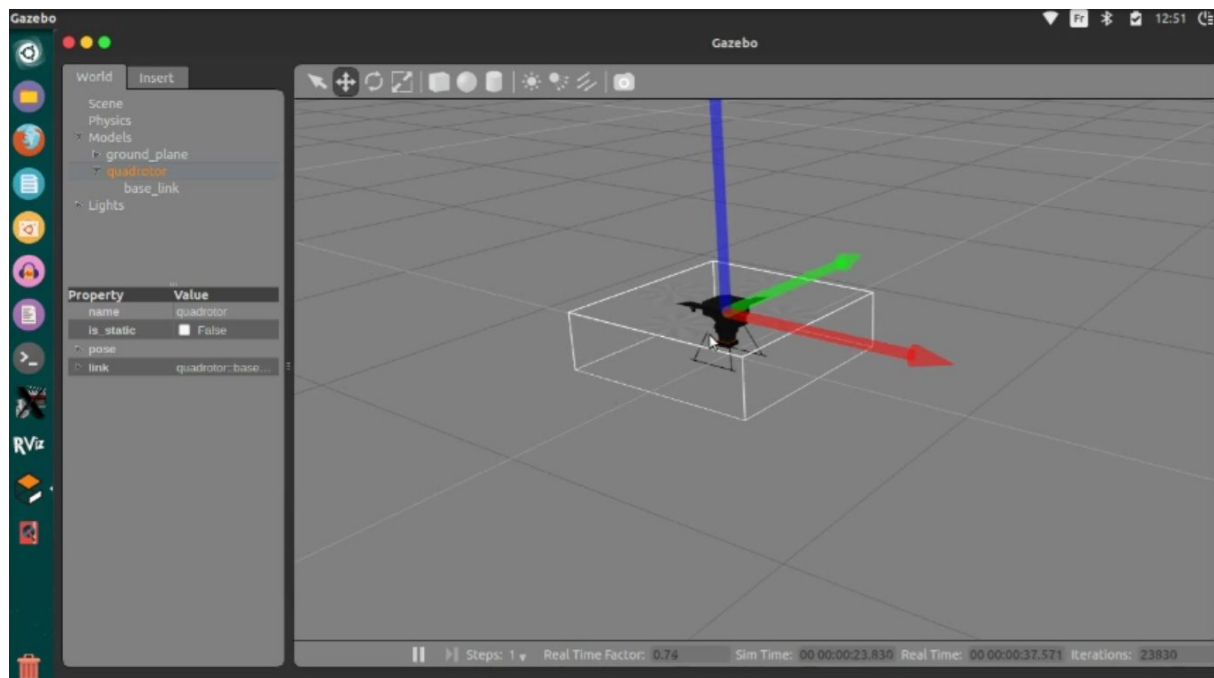
Nous avons réalisé une vidéo résumé du lancement de la simulation visionnable ici:

<https://youtu.be/N4swrVIOgh8>

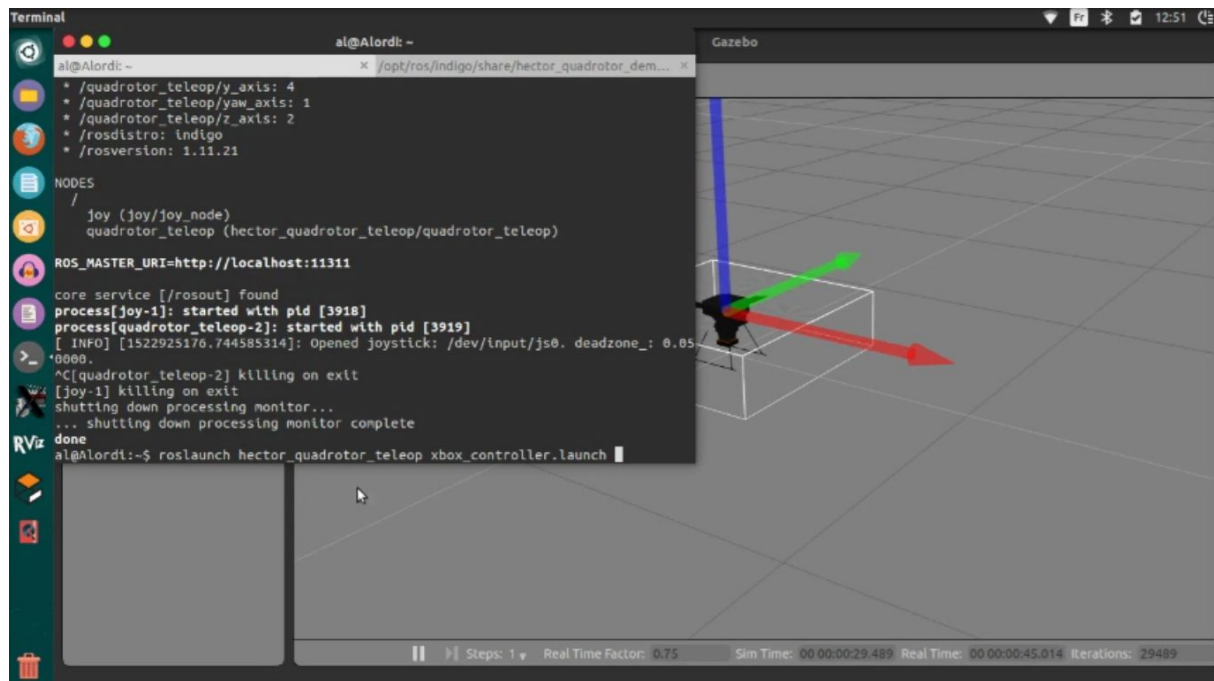
Lancement de ROS/GAZEBO/Rviz



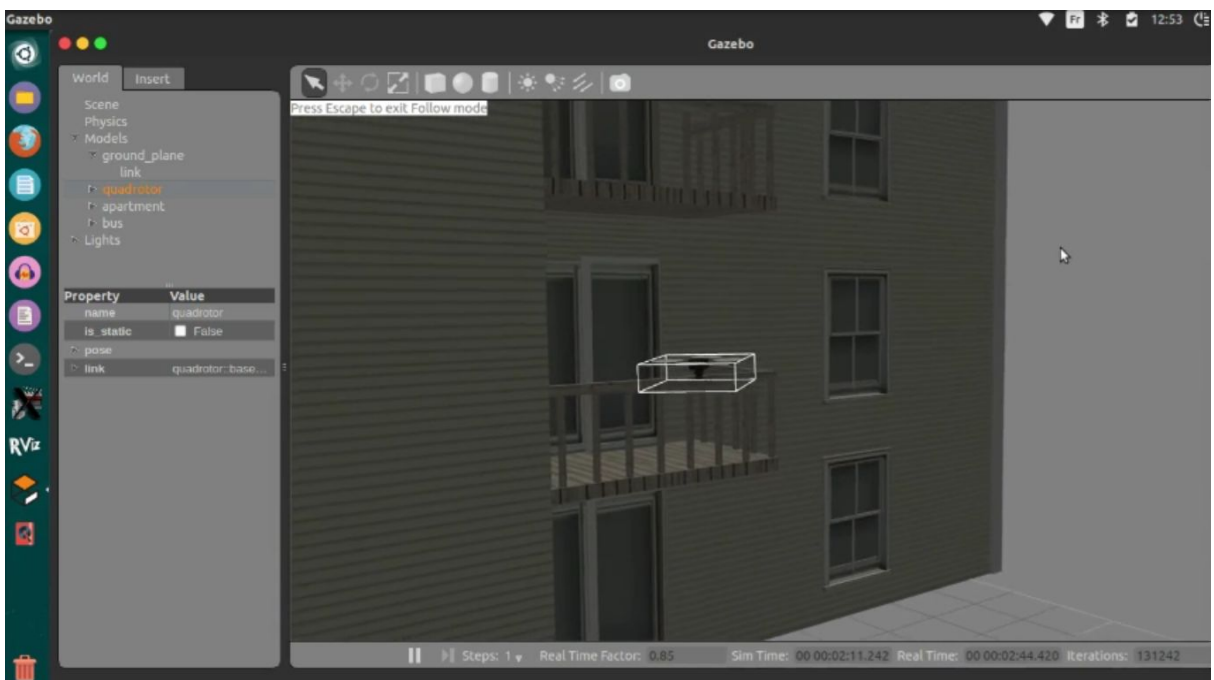
Apparition du drone dans gazebo au bout de quelques secondes, suppression du decor pour conserver la RAM



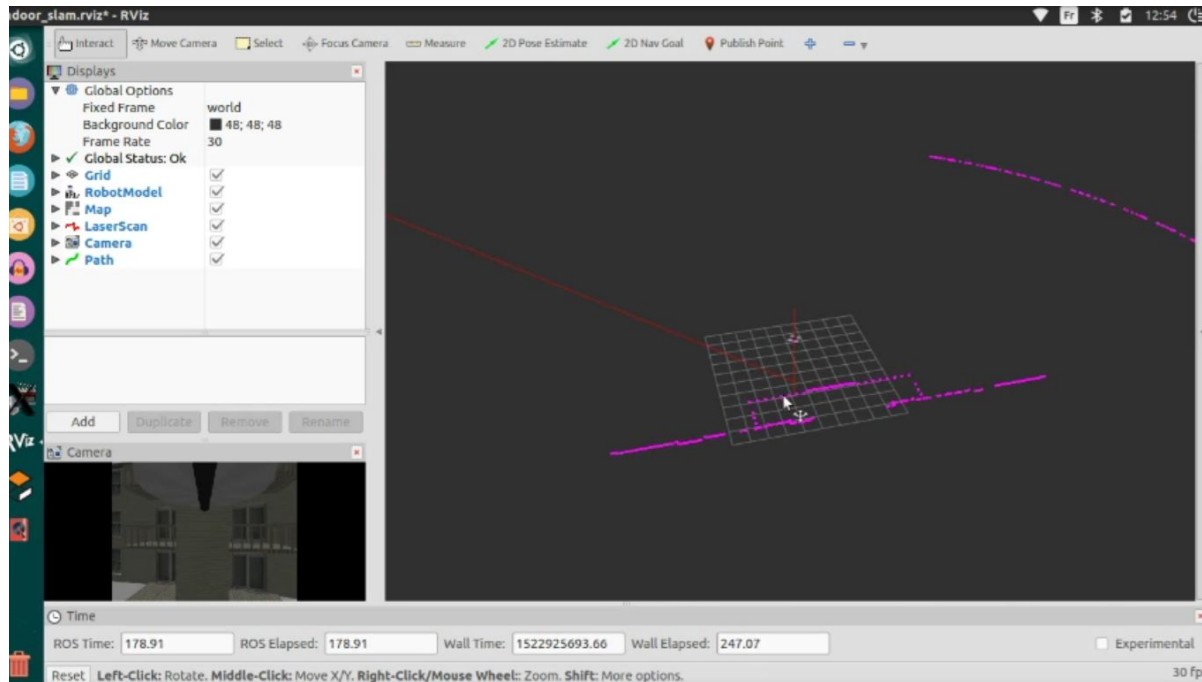
Lancement du xbox controller



Contrôle du drone



Lidar sur Rviz



7) Configuration du drone

opt/ros/indigo/share/

/opt/ros/indigo/share/hector_quadrotor_gazebo/

8) Problèmes rencontrés

- La radio télécommande ne peut pas être connectée à l'ordinateur ni par câble ni par radio et il faudrait coder une interface entre les logs de la manette et ROS.
- La modification des paramètres du drone (masse, taille, puissance) est complexe et représenterait donc un investissement de temps peu rentable car on ne pourrait pas y connecter la vraie manette.

Nous suspendons donc ici le workpackage Simulation.