



1

# PRÉSENTATION INTERMÉDIAIRE P22

## COMMANDE EN POSITION D'UN DRONE

Tuteur : Monsieur Komi Midzodzi PEKPE

Binômes : Lijie YAO et Lirui ZHANG

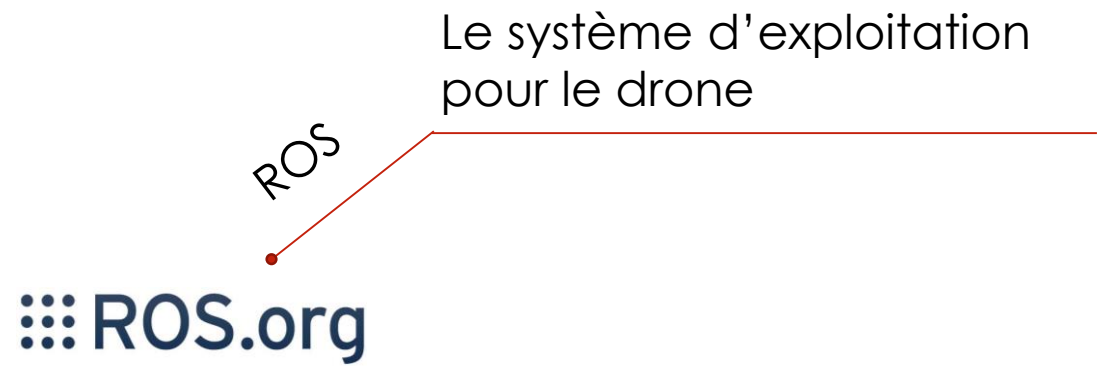
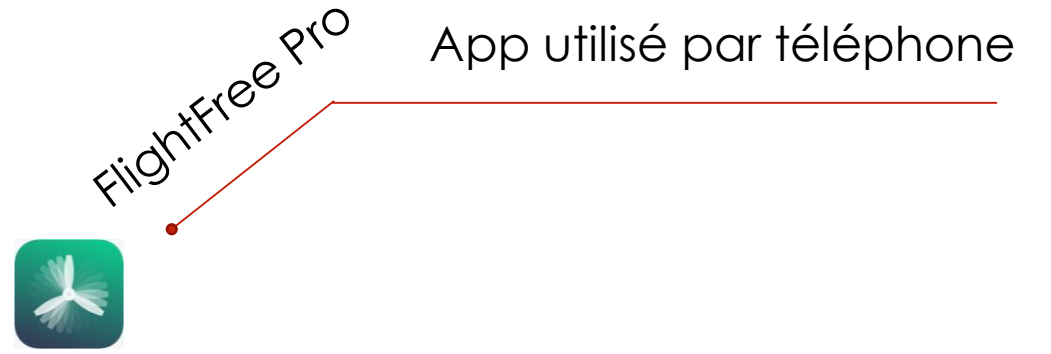
# SOMMAIRE

- L'introduction
- Ce que nous avons fait
  - L'étude sur le drone
  - L'étude de travail d'environnement , de logiciel et d'interface
  - Commande et contrôle effectué sur le drone
  - Commande et contrôle par l'interface
- Dans le futur
  - Améliorer de l'interface
  - Réguler le drone par taper et lire des données en temps réel
  - Partager le fichier entre Matlab et ROS
  - Afficher des informations de drone en temps réel sur l'écran
  - Corriger de contrôle

# L'INTRODUCTION

- Le but de notre projet est étude de faisabilité de quelques fonctions qui peut utilité pour des TP ou les recherches en tant que support.
- Comme un drone peut décoller, atterrir, avancer, reculer, pivoter, basculer et prendre des photographies aériennes, nous espérons mettre en œuvre un contrôle cinématique du drone. Il peut s'agir d'un contrôle de la position spatiale du drone, d'un contrôle basé sur le traitement d'image ou d'un contrôle de vitesse du drone. Il existe de nombreuses possibilités pour ce contrôle de la cinématique, nous étudions les fonctionnalités qui nous intéressent et essayons de mettre en œuvre cette fonction.
- Dans la première moitié de nos études, nous avons mené des recherches sur le drone, de l' environnement du travail, de logiciels et d'interface.
- Pour la seconde moitié du travail, nous serons plus intéressés par le contrôle en position du drone par une interface.

# L'ÉTUDE SUR LE DRONE



# L'ÉTUDE SUR LE DRONE

- Une caméra horizontale qui prend des photos et des vidéos
- Une caméra verticale qui permet le maintien d'un point fixe.
- Un capteur ultrason qui analyse l'altitude de vol jusqu'à 5 mètres.
- Un gyroscope 3 axes qui permet de calculer l'angle d'inclinaison de l'appareil.
- Un magnétomètre 3 axes qui donne la possibilité de définir la position du drone, à l'image d'une boussole.
- Un GPS et un GLONASS pour la géolocalisation du drone et aide à mesurer la vitesse de drone pour plus de stabilité à haute altitude.
- Un capteur de pression qui permet de mesurer la pression et de calculer l'altitude de vol lorsque celle-ci dépasse les 5 mètres.
- Un accéléromètre qui permet de mesurer l'orientation du drone sur 3 axes et sa vitesse linéaire.



[https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKewjzh-KTmanfAhUszlUKHelSB34QjRx6BAgBEAU&url=https%3A%2F%2Fwww.lesnumeriques.com%2Fdrone%2Fparrot-bebop-2-p29821%2Ftest.html&psig=AOvYaw18v9aMGfyQKEi\\_2To7tBi&ust=1545216225108124](https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKewjzh-KTmanfAhUszlUKHelSB34QjRx6BAgBEAU&url=https%3A%2F%2Fwww.lesnumeriques.com%2Fdrone%2Fparrot-bebop-2-p29821%2Ftest.html&psig=AOvYaw18v9aMGfyQKEi_2To7tBi&ust=1545216225108124)



# L'ÉTUDE DE TRAVAIL D'ENVIRONNEMENT

The ROS logo consists of a 3x3 grid of dots to the left of the letters "ROS" in a bold, sans-serif font.The logo for the Open Source Robotics Foundation, featuring a stylized robot head icon.

Open Source Robotics Foundation

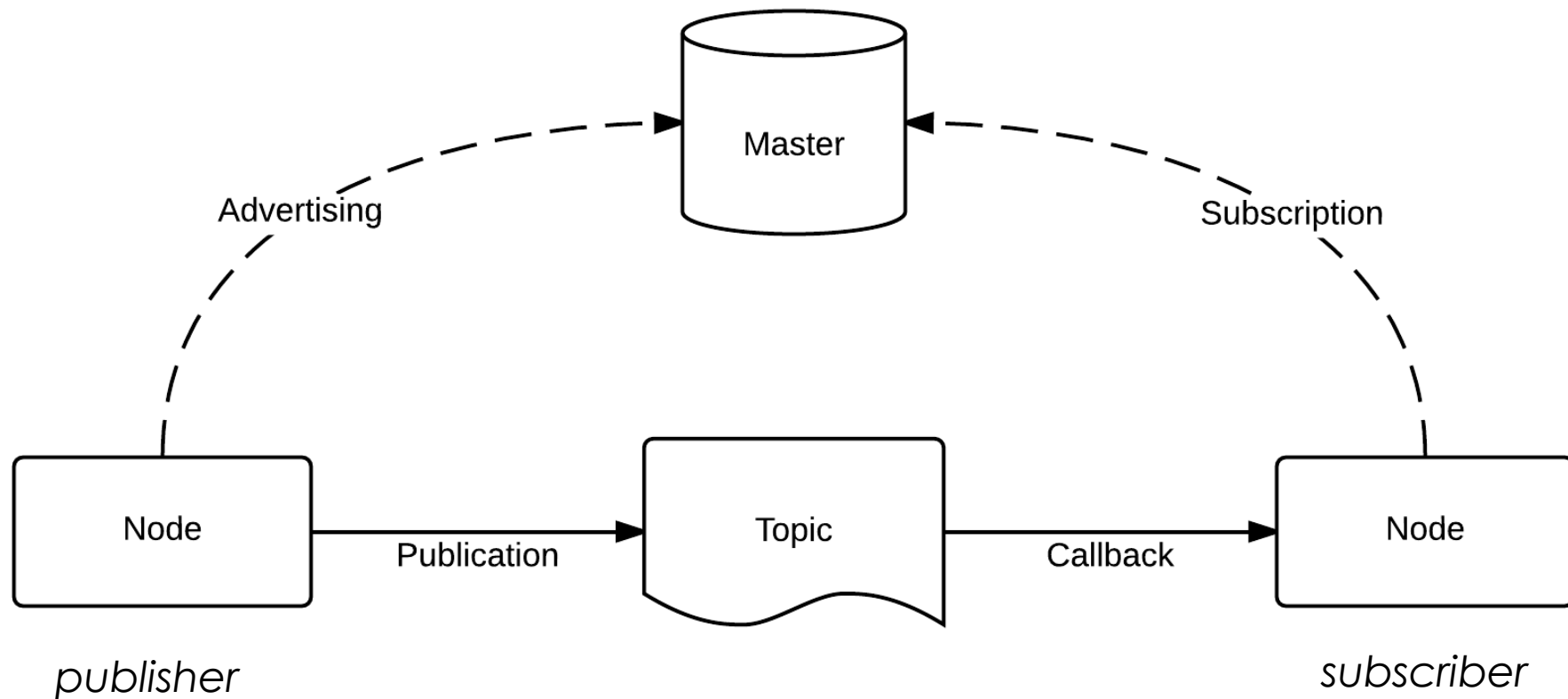
C'est quoi

Un méta-système d'exploitation qui peut fonctionner sur un ou plusieurs ordinateurs et qui fournit plusieurs fonctionnalités.

Pourquoi

Pour faire contrôler le drone.  
Il y a deux ROS dans le drone, un pour les capteurs et un autre pour les commandes.

# L'ÉTUDE DE TRAVAIL D'ENVIRONNEMENT



# L'ÉTUDE DE LOGICIEL ET D'INTERFACE

Connecter, piloter, recevoir des flux, enregistrer et télécharger des médias (photo et vidéo), envoyer et lire des plans de vol sur pilote automatique et mettre à jour notre drone.

**Parrot**  
For Developers

SDK Parrot

Un drive développé sur SDK Parrot, utilisé pour lancer le pilote, envoyer de commandes à Bebop, lecture de Bebop, configuration de Bebop et du pilote

Bebop Autonomy



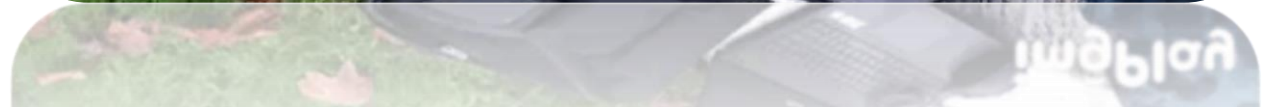
Python Pyparrot

Une interface de python pour Bebop basé sur Anaconda. Utilisé pour communiquer et contrôler le drone, utilisé également pour récupérer des données des capteurs.



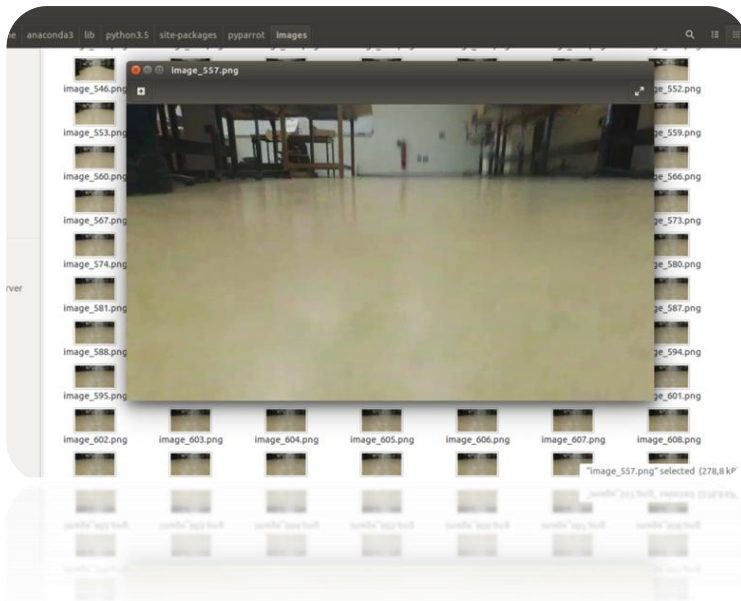
# COMMANDE ET CONTRÔLE EFFECTUE SUR LE DRONE<sup>9</sup>

- Décollage et atterrissage
- Vol en droite



# COMMANDE ET CONTRÔLE EFFECTUE SUR LE DRONE

- Utiliser des capteurs
- Utiliser de caméra



```

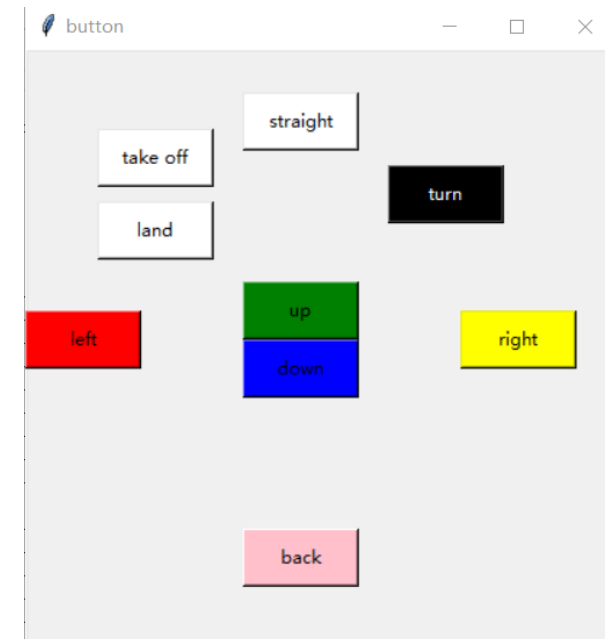
parrotbebop2@parrotbebop2-GL502VML: ~/test/20181212
DONE - disconnecting
80
{'AllStatesChanged': True, 'VideoRecordingTimestamp_stopTimestamp': 1547996, 'CameraSettingsChanged_panMax': 35.0, 'HomeTypeChosenChanged_type': 'FOLLOWEE', 'Orientation_tilt': -13, 'MassStorageStateListChanged_name': 'b'i', 'AttitudeChanged_yaw': -3.1073977947235107, 'HomeTypeChanged_type': 'FOLLOWEE', 'CameraSettingsChanged_tiltMax': 17.0, 'AltitudeChanged_altitude': 0.0, 'AttitudeChanged_pitch': 0.016698559746146202, 'TimelapseChanged_minInterval': 1.0, 'MassStorageContent_mass_storage_id': 0, 'AvailabilityStateChanged_AvailabilityState': 0, 'WifiSignalChanged_rssi': -51, 'AlertStateChanged_state': 'none', 'ConnectedAccessories_swVersion': 'b'\xff', 'PictureStateChangedV2_error': 'ok', 'HomeTypeAvailabilityChanged_available': 0, 'GpsLocationChanged_altitude_accuracy': -58, 'RunIdChanged_runId': 'b'\x00', 'defaultCameraOrientationV2_tilt': -13.0, 'CameraSettingsChanged_fov': 78.0, 'HomeTypeAvailabilityChanged_type': 'FOLLOWEE', 'NavigateHomeStateChanged_reason': 'userRequest', 'GPSFixStateChanged_fixed': 0, 'MotorErrorStateChanged_motorError': 'noError', 'MotorSoftwareVersionChanged_version': 'b'6', 'moveToChanged_altitude': 500.0, 'VelocityRange_max_pan': 20.0, 'WifiSecurity_key': 'b'\x00', 'DeprecatedMassStorageContentChanged_nbCrashLogs': 1, 'MassStorageInfoStateListChanged_mass_storage_id': 0, 'HomeChanged_altitude': 500.0, 'VideoResolutionsChanged_type': 'rec1080_stream480', 'PictureStateChangedV2_state': 'ready', 'MassStorageInfoStateListChanged_internal': 1, 'DeprecatedMassStorageContentChanged_nbVideos': 65, 'PositionChanged_longitude': 500.0, 'VideoFramerateChanged_framerate': '30_FPS', 'HomeChanged_longitude': 500.0, 'moveToChanged_status': 'RUNNING', 'MagnetocalibrationRequiredState_required': 0, 'MagnetocalibrationAxisToCalibrateChanged_axis': 'none', 'MotorFlightsStatusChanged_lastFlightDuration': 8, 'MassStorageContent_nbRawPhotos': 0, 'AttitudeChanged_roll': 0.006098572630435228, 'MassStorageContent_nbVideos': 65, 'MassStorageStateListChanged_mass_storage_id': 0, 'PilotedPOI_longitude': 500.0, 'Orientation_pan': 0, 'WifiSecurity_keyType': 'plain', 'CameraSettingsChanged_panMin': -35.0, 'OrientationV2_pan': 0.0, 'defaultCameraOrientationV2_pan': 0.0, 'ComponentStateListChanged_State': 0, 'MassStorageInfoStateListChanged_plugged': 1, 'ConnectedAccessories_id': 255, 'MassStorageContentForCurrentRun_nbVideos': 1, 'TimelapseChanged_maxInterval': 300.0, 'VideoRecordingModeChanged_mode': 'time', 'TimelapseChanged_interval': 8.0, 'SensorsStatesListChanged_sensorName': 'vertical_camera', 'HomeChanged_latitude': 500.0, 'moveToChanged_heading': 0.0, 'GpsLocationChanged_longitude': 500.0, 'MassStorageContent_nbPhotos': 0, 'BatteryStateChanged_battery_percent': 80, 'TimelapseChanged_enabled': 0, 'MassStorageContentForCurrentRun_nbPhotos': 0, 'DeprecatedMassStorageContentChanged_nbPuds': 11, 'PilotedPOI_latitude': 500.0, 'MassStorageContent_nbPuds': 11, 'moveToChanged_latitude': 500.0, 'MotorFlightsStatusChanged_totalFlightDuration': 2235, 'MavlinkFilePlayingStateChanged_filepath': 'b'\x00', 'ConnectedAccessories_uid': 'b'\xff', 'DeprecatedMassStorageContentChanged_mass_storage_id': 0, 'GpsLocationChanged_altitude': 500.0, 'AlertSound_state': 'stopped', 'MavlinkFilePlayingStateChanged_state': 'stopped', 'MassStorageInfoStateListChanged_used_size': 4996, 'defaultCameraOrientation_tilt': -13, 'OrientationV2_tilt': -13.0, 'MotorErrorLastErrorChanged_motorError': 'noError', 'VideoEventChanged_event': 'stop', 'VelocityRange_max_tilt': 20.0, 'MassStorageContent_nbCrashLogs': 1, 'GpsLocationChanged_latitude': 500.0, 'moveToChanged_longitude': 500.0, 'VideoEnableChanged_enabled': 'disabled', 'DeviceLibARCommandsVersion_version': 'b'4', 'PositionChanged_altitude': 500.0, 'SpeedChanged_speedY': 0.0, 'SensorsStatesListChanged_sensorState': 0, 'DeprecatedMassStorageContentChanged_nbPhotos': 0, 'WifiSecurityChanged_type': 'open', 'defaultCameraOrientation_pan': 0, 'MagnetocalibrationStartedChanged_started': 0, 'ConnectedAccessories_list_flags': 255, 'GpsLocationChanged_longitude_accuracy': -48, 'MassStorageInfoStateListChanged_size': 7338, 'WifiSecurity_type': 'open', 'CameraSettingsChanged_tiltMin': -83.0, 'MotorFlightsStatusChanged_nbFlights': 85, 'MassStorageInfoStateListChanged_full': 0, 'MotorErrorStateChanged_motorIds': 0, 'VideoStateChangedV2_state': 'stopped', 'SpeedChanged_speedZ': 0.0, 'PilotedPOI_status': 'UNAVAILABLE', 'MavlinkFilePlayingStateChanged_type': 'flightPlan', 'MassStorageContentForCurrentRun_nbRawPhotos': 0, 'SpeedChanged_speedX': 0.0, 'ComponentStateListChanged_component': 'TakeOff', 'VideoRecordingTimestamp_startTimestamp': 1537877, 'FlyingStateChanged_state': 'landed', 'moveToChanged_orientation_mode': 'NONE', 'VideoStateChangedV2_error': 'ok', 'GpsLocationChanged_latitude_accuracy': -58, 'NavigateHomeStateChanged_state': 'unavailable', 'VideoEventChanged_error': 'ok', 'MassStorageContentForCurrentRun_mass_storage_id': 0, 'PositionChanged_latitude': 500.0, 'PilotedPOI_altitude': 500.0, 'ConnectedAccessories_accessory_type': 'UNKNOWN_ENUM_VALUE'}
disconnecting
parrotbebop2@parrotbebop2-GL502VML: ~/test/20181212

```



# COMMANDE ET CONTRÔLE EFFECTUE PAR L'INTERFACE

- Connecter et déconnecter le service de wifi
- Décollage et atterrissage
- Capteurs
- Caméra vidéo
- Vol et rotation
- Faire certaines limites(altitude, vitesse etc.)
- Pause ou sommeil
- Créer une interface pour contrôler (appuyer chaque bouton et déterminer le valeur de variable dans le terminal pour réaliser les différents fonctions)



# Démonstration

# LE PLAN DANS LE FUTUR

- Améliorer l'interface pour afficher les paramètres de statut de vol.
  - Utiliser la boîte à outils tkinter de Python GUI.
  - Afficher les paramètres de drone sur l'écran.
  - Renouveler les variables affichées.
- Partager le fichier avec Matlab.
  - Enregistrer les variables d'état en vol dans un fichier, puis utiliser Matlab pour lire le fichier afin de gérer les erreurs.

# LE PLAN DANS LE FUTUR

- Corriger des erreurs de vol.
  - Si nous pouvons collecter les paramètres de l'état de vol via le capteur, nous pouvons faire une comparaison avec notre commande, car des erreurs peuvent se produire pendant le vol. Nous envisageons ensuite d'utiliser un contrôleur d'erreur (PID?) pour corriger ces erreurs.



Merci de votre attention

