**Setup**

**Windows Setup:**

* 1. On the Host Computer, download the [DJI Assistant 2 for Autopilot](https://www.dji.com/downloads/softwares/assistant-dji-2-for-autopilot).
  2. [Download](https://www.dji.com/goapp) the DJI GO App to your mobile device.
  3. Refer to this [link](https://developer.dji.com/onboard-sdk/documentation/development-workflow/environment-setup.html#all-platforms), and follow the instructions mentioned under the topics under All Platforms -> Update Firmware, Enable OSDK API, Onboard SDK Application registration.
  4. The DJI ROS SDK has some services, which are of custom message type, and are not shipped by default by the ROS toolbox. Note that you may have to install Python, CMAKE and C++ compiler to use these custom types in MATLAB. Refer [ROS System Requirements](https://www.mathworks.com/help/ros/gs/ros-system-requirements.html) for more information.

**Onboard Computer Setup**

1. The Onboard Computer could be any Single Board Computer running Ubuntu OS. This example is run on [NVIDIA Jetson Nano](https://developer.nvidia.com/embedded/jetson-nano-developer-kit) running Ubuntu 18.04.
2. The DJI Onboard SDK is supported on Ubuntu Platforms only for versions 16.04 and above. Ubuntu 18.04 is the recommended version. Ensure the supported Ubuntu OS is installed on your compute board.
   1. Follow the instructions mentioned at this [link](https://developer.dji.com/onboard-sdk/documentation/development-workflow/environment-setup.html#linux-with-ros) under Linux with ROS. Refer the below steps for detailed instructions.
      1. Install C, C++ Compiler and Development Tools by installing *build-essential.* Execute the below steps on your terminal
         1. *sudo apt update*
         2. *sudo apt install build-essential*
      2. Install CMake 2.8.3 or newer by executing the below on your terminal
         1. *sudo apt-get -y install cmake*
      3. Install the ROS distribution compatible with the Ubuntu OS running on your compute board. Check this [link](https://www.ros.org/reps/rep-0003.html) for ROS compatibilities. In this example, we have used [ROS Melodic](http://wiki.ros.org/melodic/Installation/Ubuntu) distribution and recommend the same.
3. Install DJI SDK Core Library. Before using the DJI SDK ROS package, we first need to install the DJI SDK core library
   1. Download the necessary files from git repo by executing the below commands on the terminal:
      1. *cd ~*
      2. *git clone https://github.com/dji-sdk/Onboard-SDK*
      3. *cd Onboard-SDK*
   2. Checkout Version 3.7, as this version is being tested with MATLAB and Simulink.
      1. *git checkout 3.7*
   3. Create a build directory and cd into it:
      1. *mkdir build*
      2. *cd build*
   4. Compile and install DJI SDK core library
      1. *cmake ..*
      2. *make djiosdk-core*
      3. *sudo make install*
4. Setup the DJI SDK ROS package. Now that we have the core sdk library installed and ready to go, we can start setting up the DJI SDK ROS package.
   1. If you don't have a catkin workspace, create one as follows:
      1. *mkdir catkin\_ws*
      2. *cd your\_catkin\_ws*
      3. *mkdir src*
      4. *cd src*
      5. *catkin\_init\_workspace*
   2. To download and install the dji\_sdk ROS package, go to your catkin workspace.
      1. *cd ~/your\_catkin\_ws/src/*
   3. Download the DJI SDK ROS package from our git repository.
      1. *git clone https://github.com/dji-sdk/Onboard-SDK-ROS.git*
   4. Checkout Version 3.7, as this version is being tested with MATLAB and Simulink.
      1. *git checkout 3.7*
   5. Run 'catkin\_make' and compile.
      1. *cd ~/your\_catkin\_ws/*
      2. *catkin\_make*
5. Now follow all the instructions in this [wiki](http://wiki.ros.org/dji_sdk/Tutorials/Getting%20Started) page from step 2.2 onwards, to setup the DJI Onboard ROS SDK on your compute board and to verify if the setup is working.
6. Follow the steps in this [tutorial](http://wiki.ros.org/dji_sdk/Tutorials/Running%20the%20flight%20control%20sample%20code) to run the sample flight control application to ensure the setup is fine.
7. The explanation of the usage of the DJI subscribers, publishers and services are documented in the [dji\_sdk](http://wiki.ros.org/dji_sdk) wiki page.