-A- Get code and build

Using https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki/CatkinBuild#catkin-build-and-rosinstall as template to build

wstool to get all code from different repositories

catkin to build everything

Main difference with dVRK build is to use a different wstool .rosinstall file:
> wstool merge https://raw.githubusercontent.com/jhu-saw/sawIntuitiveDaVinci/devel/ros/isi.rosinstall ros
To setup environment, use: https://github.com/jhu-cisst/cisst/wiki/Compiling-cisst-and-SAW-with-CMake#setting-up-your-environment-variables-for-ros

-B- Run code for isi_api (data from da Vinci)

roscore --> master (always on)

rosrun isi_ros isi_console (this has to be turned on)

rostopic --> more for debug

ex) rostopic echo /ECM1/measured_cp (display messages published from the endoscopic camera)

rosbag --> for data collection

-C- video

Launch files copied from https://github.com/jhu-dvrk/dvrk-ros/tree/devel/dvrk_robot/launch to local directory ~/catkin_ws/src/cisst-saw/sawIntuitiveDaVinci/ros/launch

To run frame capture (left/right/pose):

1. roslaunch isi_ros gscam_decklink_stereo.launch rig_name:=daVinci
2. rosbag record --duration xx /daVinci/left/decklink/camera/image_raw

/daVinci/right/decklink/camera/image_raw /ECM1/measured_cp

To check that videos are ok, use:

1. rosrun rqt_image_view rqt_image_view
2. rosbag play file_name.bag

-D- Notes

Image size: 1920 * 1080

-E- Arduino

http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World

- check usb port : dmesg|grep tty

rosrun rosserial_python serial_node.py _port:=/dev/ttyACM0

sudo chmod a+rw /dev/ttyACM0

-F- LabVIEW

http://wiki.ros.org/ROS/Tutorials/MultipleMachines

http://wiki.ros.org/ROS/NetworkSetup

-G- TO RUN THE WHOLE SYSTEM

1. roslaunch get_3d_pos stereo_proc.launch rig_name:=daVinci

-H- git token