

saw3Gear Installation Guide (Windows 7)

1. Install the 3Gear hardware/software according to their website.
 - a. We used 3Gear's "legacy" setup with 2 Kinects; they have since moved on to a "single camera devkit," which hasn't been tested with the saw3Gear. Instructions for the legacy setup can be found here:
<http://www.threegear.com/latest/doc/installation-win64-2camera.html>
 - b. Using the older (Feb 2013) software that saw3Gear was tested on, hand calibration had to be done using "user specific hand models." Newer software may be able to bypass this step.
 - c. Test the 3Gear system using handdriver.bat.
2. Install ICE 3.5.0 according to website: <http://www.zeroc.com/download.html>
 - a. Add ICE to your user path: C:\Program Files (x86)\ZeroC\Ice-3.5.0\bin\x64\
 - b. NOTE: We weren't able to compile CISST with ICE 3.4 on our particular Windows machine, but if you are able to do so, then this might work better if you plan on connecting with an Ubuntu system. At this time, there is no Ubuntu package for ICE 3.5, and the two versions of ICE are not cross-compatible.
3. Install CISST according to the website:
<https://trac.lcsr.jhu.edu/cisst/wiki/DownloadAndInstallationFAQ#OnWindows>
 - a. We built using Visual Studio 2010
 - b. When using CMake, we checked the box for CISST Multitask and unchecked StereoVision. We left the other values as their defaults.
 - a. Set CMake options for ICE:
 - i. ICE_HOME: /path/to/Ice-3.5.0
 - ii. ICE_ICE_H_INCLUDE_DIR: /path/to/Ice/include
 - iii. ICE_SLICE2CPP: /path/to/Ice/bin/x64/slice2cpp.exe
 - c. Compile the code in Visual Studio, in Release Mode
4. Build saw3Gear
 - a. Set CMake Options:
 - i. CISST_USE_FILE:
/path/to/cisst/build/cisst/share/cisst-1.0/cmake/UseCisst.cmake
 - ii. cisst_DIR: /path/to/cisst/build/cisst
 - b. Compile using Visual Studio 2012, in Release Mode
5. Installation is complete!

saw3Gear User Guide (Windows 7)

1. This guide will cover how to run either of saw3Gear's two programs:
 - a. saw3GearCollector simply collects the data passed to CISST and logs it in a CSV.
 - b. saw3GearIce connects to the cisstGlobalComponentManager, which allows it to send data to different machines using Zeroc's Internet Communications Engine.

2. Start the 3Gear hand driver by running handdriver.bat in a command prompt window.
3. If you are running the cisstGlobalComponentManager on this machine (we did not do so for our setup using cisstToRos):
 - a. Set CISST environment variables by navigating to CISST's build tree in a command prompt and typing "cisstvars.bat Release"
 - b. Run cisstGlobalComponentManager by typing cisstGlobalComponentManager in the same command prompt.
4. In a new command prompt, set cisst environment variables (same as in Step 3A).
5. Using that command prompt window, navigate to the folder where saw3Gear is built. You can run either program by typing in its name.
 - a. To connect saw3GearIce to a cisstGlobalComponentManager on a different machine, type "saw3GearIce.exe [ip address of machine with Global Component Manager]".

cisstToRos Installation Guide (Ubuntu 12.04, ROS Fuerte)

1. Install ICE by ZeroC: <http://www.zeroc.com/download.html>
 - a. Install the same version of ICE that you will be using on other machines (different versions of ICE are not cross-compatible).
 - b. For older versions of ICE, this is as easy as typing "sudo apt-get install zeroc-ice34" for ICE 3.4. <http://packages.ubuntu.com/precise/zeroc-ice34>
 - c. We installed ICE 3.5.0 in order to network with a Windows machine that could not use older versions of ICE. In order to do this, you must install all of ICE's dependencies and build ICE from source.
 - i. Install (using Ubuntu's package manager, i.e. sudo apt-get install): g++, Libbz2-dev, Libexpat1-dev, Libssl-dev.
 - ii. Download and unpack the "3rd party sources" on Ice's downloads page. Follow the instructions for installing for C++.
 1. For 64-bit systems, you need to either rename the BerkeleyDB lib folder or create a symbolic link: `sudo ln -s /usr/local/BerkeleyDB.5.3/lib /usr/local/BerkeleyDB.5.3/lib64`
 - iii. Download and unpack the ICE source code on Ice's downloads page. Follow those steps.
 1. In Make.rules, the location for BerkeleyDB should be specified (using the default location when installing from 3rd party sources, mine was /usr/local/BerkeleyDB.5.3)
 - iv. Do: `Export $LD_LIBRARY_PATH=/path/to/Ice-3.5.0/cpp/lib`
 - v. Finally, make a symbolic link at /path/to/Ice-3.5.0/lib that targets /path/to/Ice-3.5.0/cpp/lib (`ln -s /path/to/Ice-3.5.0/cpp/lib /path/to/Ice-3.5.0/lib`)
2. Install CISST according to the website: <https://trac.lcsr.jhu.edu/cisst/wiki/DownloadAndInstallationFAQ>
 - a. When configuring cmake (under advanced options), enable

- “CISST_MTS_HAS_ICE”, and make sure CISST_cisstMultiTask is on (and most likely turn CISST_cisstStereoVision off).
- b. These are the CMake options for ICE:
 - i. ICE_HOME: /path/to/Ice-3.5.0
 - ii. ICE_ICE_H_INCLUDE_DIR: /path/to/Ice-3.5.0/cpp/include (if installing Ice 3.5 from source; otherwise, it would just be /path/to/Ice/include)
 - iii. ICE_SLICE2CPP: /path/to/Ice-3.5.0/cpp/bin/slice2cpp (if installing Ice 3.5 from source; otherwise, it would just be /path/to/Ice/bin/slice2cpp)
3. Follow instructions on ROS’s website to install ROS Fuerte:
<http://www.ros.org/wiki/fuerte/Installation/Ubuntu>
- a. Create a ROS workspace (ros_ws init ~/fuerte_workspace /opt/ros/fuerte)
 - b. In the ROS workspace folder, create a ROS package: roscreate-pkg cisstRosBridge roscpp tf
4. To build cisstToRos:
- a. Put cisstToRos’s source files in the /path/to/cisstToRos folder (in the folder, svn co <https://svn.lcsr.jhu.edu/robotorium/trunk/apps/3gear/cisstRosBridge/cisstRosBridge/> .). You will need to first remove ROS’s default CMakeLists.txt.
 - b. Using cmake (type cmake . in this folder):
 - i. Specify the CISST directory (you might need to configure before this option comes up). CISST_DIR: /path/to/cisst/build/cisst
 - ii. Generate.
 - c. In /path/to/cisstToRos, type “rosmake”

cisstToRos User Guide (Ubuntu 12.04, ROS Fuerte)

1. Currently, cisstToRos only works with saw3Gear_ice. So in order to use cisstToRos, you will also need a Windows computer with saw3Gear installed. Future work with cisstToRos will generalize the program to work with more CISST Multitask interfaces.
 - a. Or, you simply need a program that uses the same provided interfaces and passes the same types of data as saw3Gear.
2. On the Ubuntu machine:
 - a. Set CISST environment variables by navigating to CISST’s build tree in a Terminal window and typing “. cisstvars.sh”
 - b. Run cisstGlobalComponentManager by typing cisstGlobalComponentManager in the same Terminal window (alternatively, the cisstGlobalComponentManager can be run on the Windows machine, or on any machine that can connect the saw3Gear_ice program and the cisstToRos program).
 - c. In a new Terminal window, run ROS Core by typing roscore.
3. On the Windows machine, start the saw3Gear_Ice program:
 - a. Start the 3Gear hand driver by running handdriver.bat in a command prompt window.
 - b. Set CISST environment variables by navigating to CISST’s build tree in a command prompt and typing “cisstvars.bat Release”

- c. Using that command prompt window, navigate to the folder where saw3Gear is built. and run saw3GearIce by typing "saw3GearIce.exe [IP address of Ubuntu machine]"
4. On the Ubuntu machine:
 - a. In a new Terminal window, set CISST environment variables by navigating to CISST's build tree in a Terminal window and typing ". cisstvars.sh"
 - b. Run cisstToRos by typing "roslaunch cisstRosBridge [or the name you gave the package] cisstToRos_ice".
 - i. You might get "mailbox full" messages. This is OK, as long as you get "running" messages shortly afterwards.
5. To visualize moving frames in ROS's rviz:
 - a. Run rviz by typing "roslaunch rviz rviz"
 - b. Set "Fixed Frame" to /world
 - c. Set "Target Frame" to /world
 - d. Click Add, and add a TF Display Type.
 - e. You should see moving frames corresponding to the left and right hands picked up by the 3Gear.