This plugin allows one to control the numeric value of an EPICS variable from ImageJ. This allows macro code to control EPICS Process Variables.

To install this, one first needs to install JCA (Java channel access from APS). The easiest way to do this is to install the EPICS Area Detector viewer prebuilt binary.

http://cars9.uchicago.edu/software/epics/areaDetector.html

After downloading the proper version and unpacking you will find a folder located at :

areaDetectorPrebuilt\_R1-9-1\_windows-x64\areaDetectorPrebuilt\_R1-9-1\_windows-x64\areaDetector-1-9-1\Viewers\ImageJ\EPICS\_areaDetector

this entire folder can be copied inside the ImageJ plugins folder to install the EPICS Area Detector viewer. This will also install the necessary JCA libraries needed for using CAPut inside Imagej.

Next copy and paste the EPICSIJ\_.java and EPICSIJ\_.class inside the plugins folder as well.

Restart Imagej.

You can now run the plugin in a macro, and then access its functions with Ext.Read() and Ext.Write.

First you must use a command to initialize the plugin in a macro-

run(“EPICSIJ “);

The space after EPICSIJ is necessary.

Ext.Read() takes one string argument, which is a process variable name and returns the value as a string.

Ext.Write() takes one string argument followed by one numeric argument. It writes the numeric value to the process variable name given as the first argument.

For example:

run("EPICSIJ ");

//Write an epics variabe

Ext.write("KOZ:m1", 1);

//Read an epics variable, returns a string

x= Ext.read("KOZ:m1.RBV");

//Print the string to an imagej console

print(x);