

## System Info:

ADHamamatsu was built with EPICS version 7.07, synApps version 6\_1, asyn-R4-36, and areaDetector-R3-7. It was built on an Ubuntu 22.04 LTS (linux-x86\_64) system.

## Prerequisites:

### Compiler tools: make, g++

The above compiler tools must be installed. Use “make –version” and “g++ --version” to verify installation.

### Packages: libreadline8, libreadline6-dev, re2c, libtirpc-dev, libx11-dev, libxcomposite-dev

The above libraries may be necessary. If you receive an error while using “make” command, use “sudo apt-get install” to install missing package(s).

### EPICS Requirements

This document assumes EPICS-base, synApps module, Asyn module, and areaDetector are already set up. If these are not set up, instructions are found here:

[https://areadetector.github.io/master/install\\_guide.html](https://areadetector.github.io/master/install_guide.html)

## Implemented Features:

- DCAM Initialization
- Device detection/ selection
- Device description (part number, serial number, firmware etc...)
- Binning
- Subarray
- Pixel type
- Readout speed
- Exposure time
- Image capture (single, multiple, continuous)
- Input triggering (trigger source, trigger mode, trigger active, trigger polarity, trigger times, trigger delay)
- Sensor cooler temperature and status

## DCAM:

### Download DCAM-API Lite for Linux

1. Install DCAM-API Lite for Linux
2. Navigate to usr/lib folder

3. Run “sudo ldconfig”
4. Restart PC

### Download DCAM-SDK

*Note: Some of these files will be copied to other folders in subsequent steps*

## ADHamamatsu for Linux:

1. Download ADHamamatsu folder from our website.
  - a. Place it in the following directory ...EPICS/synApps\_6\_1/support/areaDetector-R3-7/
    - i. Your exact path may vary based on the where your EPICS folder is and the versions of synApps and areaDetector.

### Configure EPICS:

*Note: ../ADHamamatsu = EPICS/synApps\_6\_1/support/areaDetector-R3-7/ADHamamatsu*

1. Add the following commands to your **.bashrc** file (you may copy these commands from the **bashCommands.txt** file)
  - export EPICS\_BASE=\${HOME}/EPICS/epics-base
  - export EPICS\_HOST\_ARCH=\${EPICS\_BASE}/startup/EpicsHostArch
  - export  
PATH=\${EPICS\_BASE}/bin/\${EPICS\_HOST\_ARCH}:=\${EPICS\_BASE}/extensions/bin/\${EPICS\_HOST\_ARCH}:\${PATH}
  - export EPICS\_CA\_AUTO\_ADDR\_LIST=YES
  - export EPICS\_CA\_MAX\_ARRAY\_BYTES=20000000
  - export EPICS\_DISPLAY\_PATH=\${HOME}/EPICS/epics-base/adls
  - export LD\_LIBRARY\_PATH=\${EPICS\_BASE}/lib/\${EPICS\_HOST\_ARCH}

*Note: The **.bashrc** is a hidden file. It is a script file that's executed when a user logs in. The **.bashrc** file is located in your home directory. These variables might already be defined in your set up, use your definitions and modify as needed.*

2. restart pc
3. Go to your EPICS/synApps\_6\_1/support/areaDetector-R3-7/configure/RELEASE.local file and add  
ADHAMAMATSU=\$(AREA\_DETECTOR)/ADHamamatsu
4. Place hamamatsu.adl and hamamatsu\_BAK.adl with your other adl files. This is typically the location of EPICS\_DISPLAY\_PATH.
5. CD into ../ADHamamatsu
  - a. Run make clean uninstall
  - b. Run “make -j [# of processors]”

*Note: The make command above (5.b) will generate some errors. These errors will be addressed in the next section. Do **not** skip this step. The make command above (5.b) **must** be called **before** step 6 is executed. Leave the terminal window open as you will need to run “make” a second time.*

6. Copy and configure files from DCAM-SDK to EPICS
  - a. Copy and paste dcamapi4.h and dcamprop.h from “dcamsdk4/inc” folder to “../ADHamamatsu/include”
  - b. Copy and paste files common.h and console4.h from “dcamsdk4/samples/cpp/misc” folder to “../ADHamamatsu/include”
  - c. Copy and paste files common.cpp and qthread.cpp from “dcamsdk4/samples/cpp/misc” folder to “../ADHamamatsu/hamamatsuApp/src”
  - d. Edit console4.h and add “#include <time.h>”
  - e. Change line 55 in console4.h from #include “../../inc/dcamapi4.h” to #include “dcamapi4.h”
  - f. Change line 56 in console4.h from #include “../../inc/dcamprop.h” to #include “dcamprop.h”
  - g. Run “make -j [# of processors]”
7. Navigate to “../ADHamamatsu/iocs/hamamatsuIOC/iocBoot/iocHamamatsu”
  - a. Open the makefile and confirm the correct architecture (linux-x86\_64) is uncommented.
  - b. Open Terminal
  - c. Run “make”

*Note: If output is “Nothing to be done for all”, Run “make clean” then “make”*

## MEDM:

Download source code and follow instruction in this link: <https://github.com/epics-extensions/medm/tree/master>

Open <https://epics.anl.gov/extensions/medm/index.php> in your browser.

1. Navigate to the Home directory and open a **new** terminal window
2. Using the first set of three Ubuntu installation commands in the Required Software sections, install the listed packages and fonts. Some installations like x11proto-print-dev, and libxp-dev may fail.

*Note: Install packages individually. If packages are installed as a group, one failed installation will cause the rest to be ignored.*

3. Make sure medmfonts.ali.txt found in the link above is in the home directory.
4. Run the last two Ubuntu commands in the Required Software section.

5. Restart PC

## ImageJ pvAccess Viewer:

The following link has instructions on how to install ImageJ and set up the pvAccess Viewer:

[https://areadetector.github.io/master/ADViewers/ImageJ\\_EPICS\\_NTNDViewer.html](https://areadetector.github.io/master/ADViewers/ImageJ_EPICS_NTNDViewer.html)

## Running EPICS

### Run EPICS IOC:

1. Navigate to “../ADHamamatsu/iocs/hamamatsuIOC/iocBoot/iocHamamatsu” and open a **new** terminal window.
2. Run the following command
  - a. `../bin/linux-x86_64/hamamatsuApp st.cmd`

*Note: This command starts the EPICS IOC. Ignore all the connection failed warnings. They are not important. It also mentions a few DCAM functions failed. This is fine too. When the menus/buttons are initialized, some of them have to call DCAM functions.*

*Note: Control + C in terminal will stop running EPICS.*

### Run medm:

1. Navigate to the Home directory and open a **new** terminal window
2. Run the following command
  - a. `medm -x -macro "P=13HAMA1; R=cam1:" hamamatsu.adl &`

*Note: This runs medm using hamamatsu.adl. The “-x” is needed to run MEDM in execute mode. If the “-x” is not there, MEDM defaults to edit mode. To close MEDM, close the windows and type Control + C in terminal.*

*Note: EPICS has an autosave feature. The values from the previous session of MEDM will auto-populate the fields when a new session is opened.*

3. In the plugins section, click all. Enable the second plugin “PVA1”. This allows images to be transferred to ImageJ when using the EPICS NTND viewer. At this time, you can ignore all other plugins.

### Run ImageJ:

1. Navigate to the location of your ImageJ folder (../EPICS/ij153-linux64-java8/ImageJ) and open a **new** terminal window
2. Type “./ImageJ” to run the application

3. In the Plugins Menu, Select “EPICS NTNDA Viewer”

*Note: A green bar under the channel name in the Viewer indicates EPICS IOC is active. If the bar is red, Repeat step: **Run EPICS IOC**.*

*Note: Verify channel name is 13HAMA:Pva1:Image*

4. The **Start** button creates the image viewer. The display will not appear until an image has been acquired. See **Using the Camera** below.
5. The **Snap** button makes a copy of the current frame into a new window.
6. The **Stop** button will disable the updating of the display.

*Note: The Viewer automatically resets the contrast and brightness when creating a new window, resulting a white image. When this occurs, use the Image →Adjust →Brightness/Contrast... menu in ImageJ to update the LUT.*

## Using the Camera

1. Click **Init** button to initialize DCAM
2. Click **Open** button to open the camera.
3. Change camera/capture settings by using the buttons and edit boxes provided.
4. Use the **Start** and **Stop** buttons to control image capture

*Notes:*

- *When modifying the contents of edit boxes, you will need to press the Enter Key to submit a new value. The blue text next to the box will update if the value was accepted. **Keep the mouse cursor over the edit box when changing/entering values. Failing to do so will result in your edits being ignored***
- *Subarray, binning, and exposure time are validated when start is pressed*
- *To change binning, use the BinX edit box. BinY will automatically update to the same value when acquisition starts.*
- *Not all buttons are functional.*