EPICS Installation Procedures for Raspberry Pi

Vivek Singh
Student
Department of Electrical Engineering
Jabalpur Engineering College, Jabalpur, Madhya Pradesh

Dr. Shailja Shukla
Professor and Head of the Department
Department of Computer Science and Engineering
Jabalpur Engineering College, Jabalpur, Madhya Pradesh

Abstract

One of the open source software tool kit is EPICS (Experimental Physics and Industrial Control System) developed and maintained by Argonne National laboratory, US and is in use worldwide. It maintains the server client distributed control system. Communication between server and client are done through a named piece of data called Process Variable (PV). There may be many client and many servers. [1].This paper presents the installation of EPICS in Raspberry Pi. EPICS installation in Raspberry Pi has been done in a stepwise process that includes some application specific modules.

Keywords- EPICS, Raspberry Pi, Linux

I. INTRODUCTION

EPICS is a set of Open Source software tools, libraries and applications developed collaboratively and used worldwide to create distributed soft real-time control systems for scientific instruments such as a particle accelerators, telescopes and other large scientific experiments[1] EPICS IOC is a program that resides in a system. It stands for Input Output Controller. EPICS IOC can be created in any platform. Here IOC is created in Linux. EPICS IOC is also called IOC core. It mainly comprises of all records created by user, Device/driver support for hardware devices and channel access settings. Figure 1 shows the functional representation of EPICS IOC. EPICS IOC core consists of records and application specific data base. Record can be created by using either a programming language or a software. Application specific data base is prepared after the creation of application. To add or remove modules "./st.cmd" file should be modified. To establish communication between EPICS and custom boards, device/driver support is needed. This device/driver support should be linked with records. Channel access represents the adjustment and measurement of process variable.

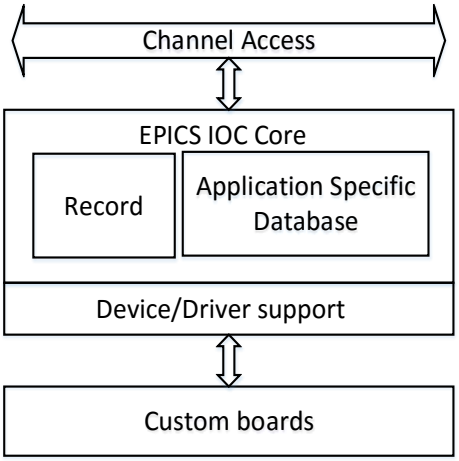

Fig. 1: EPICS functional block diagram

II. INSTALLATION OF EPICS IN RASPBERRY PI

EPICS is a distributed control system toolkit which can run on different operating system. Here Raspberry Pi is loaded with the Raspbian Jessie OS which is a distribution of Linux. Before building the EPICS, here a directory has been created which will contain all the packages to be built [2].

```
cd ~
mkdir -p ~/apps/epics
sudo su
cd /usr/local
ln -s /home/pi/Apps/epics
exit
cd ~/Apps/epics
```

An epics directory has been built under Apps directory, by this we can modify our epics packages without needing higher privileges.

Software's to be built are listed below

1) EPICS base

EPICS base contain many library and modules, which are generally needed.

2) SynApps

Tools generally needed for beam line control and data acquisition system.

(This document is regarding EPICS base 3.14 therefore the building of SynApps requires msi therefore SynApps building needs msi building, If EPICS base 3.15 is building then msi is included in that. EPICS sequencer is also needed by some module of SynApps therefore re2c sequencer package is also needed to build)

3) VDCT

A visual database configuration tool for EPICS, useful for creating dbd file.

A. Building the EPICS base

This is the main core of EPICS, comprising the build system and tools, common and OS-interface libraries, Channel Access client and server libraries, static and run-time database access routines, the database processing code, and standard record, device and driver support [3]. EPICS base 3.14 package has been used. Copy it in Apps/epics directory and then

```
cd ~/Apps/epics
tar xzf baseR3.14.12.3.tar.gz
ln -s ./base-3.14.12.3
```

EPICS has been built for different types of OS and there is way to determine whether EPICS is built for our architecture or not by running the following lines in the terminal,

```
export EPICS HOST_ARCH=`/usr/local/epics/base 3.14.12.3/startup/EpicsHostArch`
export EPICS HOST_ARCH=$(/usr/local/epics/base-3.14.12.3/startup/EpicsHostArch)
```

Now by using echo command, it can be determined whether EPICS is built for our system architecture or not echo $EPICS_HOST_ARCH

Terminal should return Linux-arm which ensures that EPICS can be built in Raspberry Pi, different system will return their system architecture.

Finally, run make command

```
cd ~/Apps/epics/base-3.14.12.3
make
```

The building process will take few minute for completion. If terminal returns no error, it may indicate that EPICS base has successfully built. Further to ensure whether epics base has built or not there are many other criteria also in terminal

```
cd ~/Apps/epics/base-3.14.12.3
./bin/linux-arm/softIoc
```

EPICS prompt will appear, in this prompt write

```
iocInit
```

Terminal will return the version and date when the EPICS has built in the system and this ensures that EPICS base has been successfully built..EPICS base has been built and one needs to set environment variable under ~/.bashrc or ~/.bash_aliases file, To do this open ~/.bash_aliases file in nano editor

```
sudo nano ~/.bash_aliases
```

This will open the bash aliases file, under this bash aliases file writes the following to declare environment variable

```
export EPICS_HOST_ARCH=`/usr/local/epics/base-3.14.12.3/startup/EpicsHostArch`
export EPICS_HOST_ARCH=$(/usr/local/epics/base-3.14.12.3/startup/EpicsHostArch)
export EPICS_HOST_ARCH=${EPICS_BASE}/startup/EpicsHostArch
export EPICS_BASE_BIN=${EPICS_BASE}/bin/${EPICS_HOST_ARCH}
export EPICS_BASE_LIB=${EPICS_BASE}/lib/${EPICS_HOST_ARCH}
if [ "" = "${LD_LIBRARY_PATH}" ]; then
export LD_LIBRARY_PATH=${EPICS_BASE_LIB}
else
export LD_LIBRARY_PATH=${EPICS_BASE_LIB}:${LD_LIBRARY PATH}
fi
export PATH=${PATH}:${EPICS_BASE_BIN}
"export PATH=$PATH:/home/pi/Apps/epics/base-3.14.12.3/bin/linux-arm" >> ~/.bashrc
```

Save it with CTRL_X and then Y

To check how much space EPICS base has taken, go to terminal and write

```
cd ~/Apps/epics
du -sc base-3.14.12.3
```

The terminal will show the occupied space by EPICS base.

B. Building the SynApps

SynApps is a collection of software tools that help to create a control system for beamlines. It contains beamline-control and data-acquisition components for an EPICS based control system. SynApps is distributed under the EPICS Open license [4]. Download the SynApps package from www.aps.anl.gov and put it in under ~/Apps/epics directory

```
cd ~/Apps/epics
tar xzf synApps_5_6.tar.gz
```

Before building the SynApps one needs to configure it for its own purpose, following are the steps to configure SynApps

Go to Apps/epics/synApps_5_6/support/configure/RELEASE

Open the file named RELEASE and modify the path of EPICS base and SUPPORT, here following changes has to be made. Here path support and EPICS path has been given and it depends where these are installed. In our case the path used are

```
SUPPORT=/usr/local/epics/synApps_5_6/support
EPICS_BASE=/usr/local/epics/base-3.14.12.3
```

To propagate these changes everywhere, one needs to make release under support

```
cd ~/Apps/epics/synApps_5_6/support
Make release
```

This release file contains support for various module so to remove modules from the build, delete or comment out the module name. Here all modules have been commented out except ASYN and CALC. After configuring the SynApps as per requirement, it needed an msi EPICS extension. Download the extension Top from www.aps.anl.gov and put it in ~/Apps/epics directory

```
cd ~/Apps/epics
tar xzf extensionsTop_20070703.tar.gz
```

Download msi from www.aps.anl.gov and put it under ~/Apps/epics/extensions/src

```
cd extension/src
tar xzf ../.. msi1-5.tar.gz
cd msi1-5
make
```

EPICS extension has been made therefore it is needed to set the environment variable in ~/.bash_aliases or ~/.bashrc file. Open it in nano editor

```
sudo nano  ~/.bash aliases
```

and add the following additional lines

```
export EPICS_EXT=${EPICS_ROOT}/extensions
export EPICS_EXT_BIN=${EPICS_EXT}/bin/${EPICS_HOST_ARCH}
export EPICS_EXT_LIB=${EPICS_EXT}/lib/${EPICS_HOST_ARCH}
if [ "" = "${LD_LIBRARY_PATH}" ]; then
export LD_"export PATH=$PATH:/home/pi/Apps/epics/base-3.14.12.3/bin/linux-arm" >> ~/.bashrc
LIBRARY_PATH=${EPICS_EXT_LIB}
else
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${EPICS_BASE_LIB}
fi
export PATH=${PATH}:${EPICS_EXT_BIN}
//bashrc//
"export PATH=$PATH:/home/pi/Apps/epics/base-3.14.12.3/bin/linux-arm" >> ~/.bashrc
```

Extension has been made, declaration has been done, now SynApps need sequencer so download re2c sequencer or either

```
sudo apt-get install re2c
```

(if sudo apt-get don't works then manually download the package and install it by reading the README File, also ensure that system clock is correct otherwise installation will give error, Sinee Raspberry Pi needs a setup for date and time following command may be used to set date and time)

```
sudo date –set="YYYY-MM_DD HR:MIN:SEC " && date -rfc-3339=ns
```

Everything that SynApps building process requires has been done, finally

```
cd ~/Apps/epics/synApps_5_6/support
make release
make rebuild
```

C. Installation of VDCT

VDCT (also known as VisualDCT) is an EPICS Visual Database Configuration Tool. The code is maintained by the EPICS community under a Free Software license (GPLv3) [5]. VDCT needs java development kit (JDK) to operate [6]. JDK for Linux arm 32-bit architecture was downloaded and installed on Raspberry Pi. after installation of JDK, VDCT was downloaded and installed successfully. VDCT is opened by navigating to the VDCT folder and using the following command [6].

```
java –jar VisualDCT.jar
```

This terminal command will open the VDCT opening window.

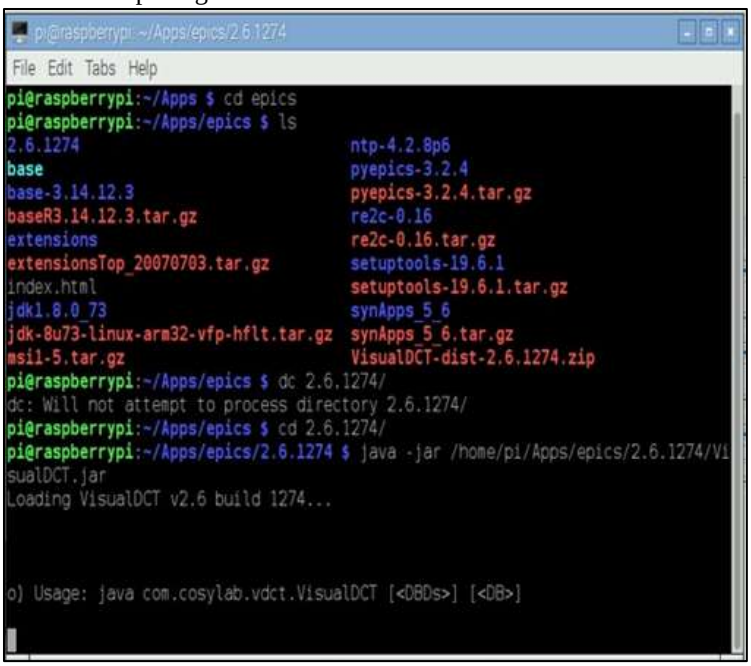

Fig. 2: Terminal window during VDCT initialization

Figure 2 shows VDCT initialization from the terminal of the Raspberry Pi. After the VDCT initialization, VDCT need a dbd file. It is shown in figure3. This dbd file is created with the application script and is shown in next section.

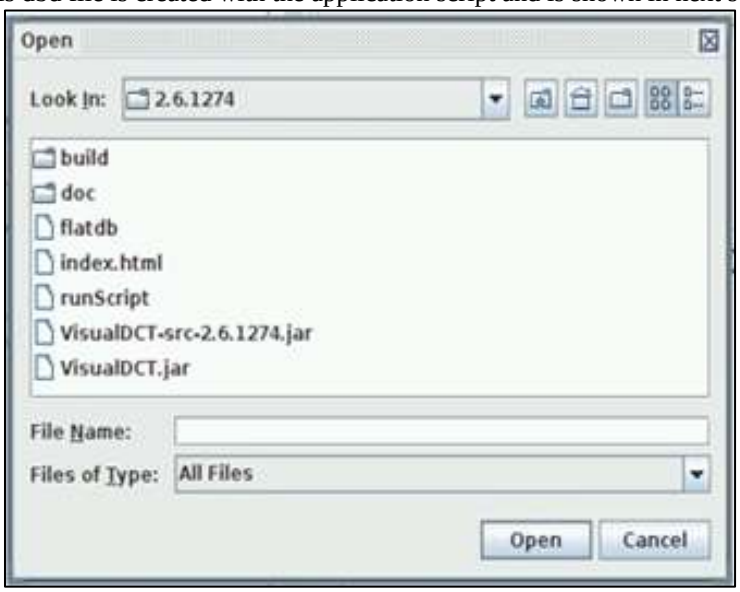

Fig. 3: VDCT window during opening

".dbd" file was created by creating an application as following
makeBaseApp.pl –t example test
maakeBaseApp.pl –I –t example test

In above command "test" is the application created. Here application should be created first and then VDCT should open so that a ".dbd" file should be available to be loaded into VDCT.

III. CONCLUSION AND FUTURE SCOPE

Installation of EPICS base, SynApps and VDCT in Raspberry Pi has been successfully done. It has been found that Raspberry Pi supports EPICS and which further can be used for creation of different types of records and establishing a communication between them. To install EPICS in different boards like beagle bone [10], orange pi [11], banana pi [12] similar approach should be used.

REFERENCES

[1] Description of Experimental Physics and Industrial Control System
http://www.aps.anl.gov/epics/
[2] EPICS in Raspberry Pi
https://prjemian.github.io/epicspi/
[3] Description of EPICS base
http://www.aps.anl.gov/epics/base/.
[4] Description of SynApps
https://www1.aps.anl.gov/bcda/synapps
[5] Description of VDCT
http://www.aps.anl.gov/epics/extensions/vdct/
[6] VDCT user manual page
http://www.aps.anl.gov/epics/extensions/vdct/2.6.1274/MAN-VisualDCT_Users_Manual.html
[7] EPICS record reference manual
http://www.aps.anl.gov/epics/EpicsDocumentation/AppDevManuals/RecordRef/Recordref-1.html
[8] Device support for the GPIO in raspberry pi
https://github.com/ffeldbauer/
[9] Availability of device support for I2C
http://www.aps.anl.gov/epics/tech-talk/2016/msg00433.php/
[10] Beagle bone products
http://beagleboard.org/bone
[11] Orange pi products
http://www.orangepi.org/
[12] Banana pi products
http://www.banana-pi.org/product.html
[13] Writing EPICS driver and device support
http://www.aps.anl.gov/epics/meetings/2009-07/talks/em_WritingEPICSDrivers.ppt
[14] Richard Peterson, The complete reference 6E, Tata McgrawHill Edition
[15] Raj Kamal, Embedded system Architecture, programming and design 3E, McGrawHill Education.