# EPICS Record Processing and Device Support Usage in 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

An 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 record processing in EPICS and usage of device support in Raspberry Pi. EPICS record processing was done using VDCT (Visual Database Configuration Tool) and GPIO access of Raspberry Pi is also shown using device support.

**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]. Figure 1 show the EPICS network block diagram. It may have many servers and many client.

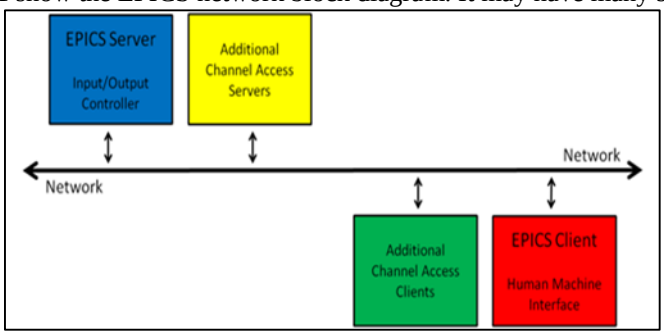



Fig. 1: EPICS network block diagram
Image source: National Instruments

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 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.

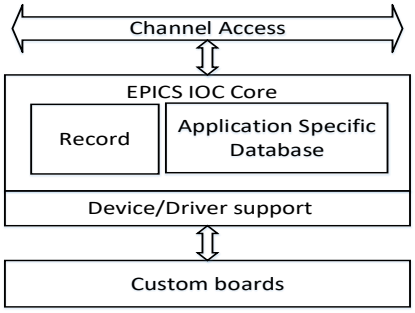



Fig. 2: EPICS functional block diagram

## II. VDCT IN RASPBERRY PI

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) [2].VDCT needs java development kit (JDK) to operate [3]. 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 [3].
java –jar VisualDCT.jar
This terminal command will open the VDCT opening window.

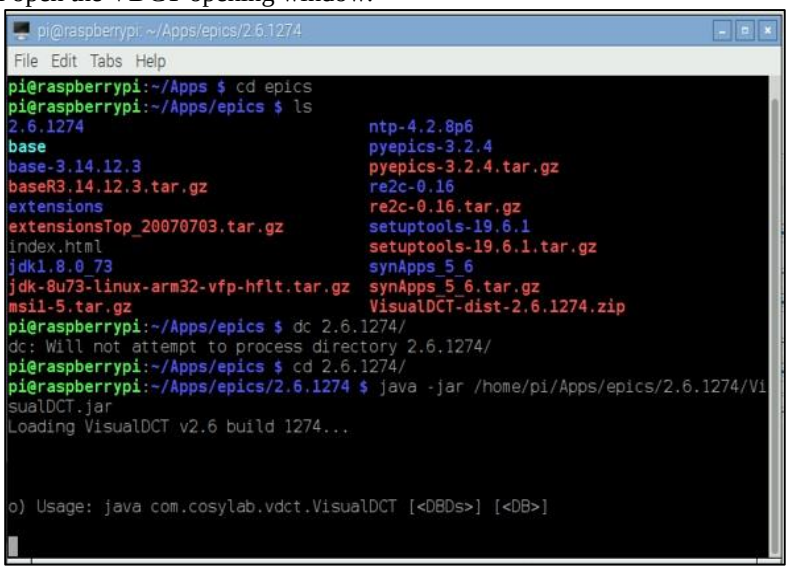



Fig. 3: Terminal window during VDCT initialization

Figure 3 shows VDCT initialization from the terminal of the Raspberry Pi. After the VDCT initialization, VDCT need a dbd file. It is shown in figure 4. This dbd file is created with the application script and is shown in next section.

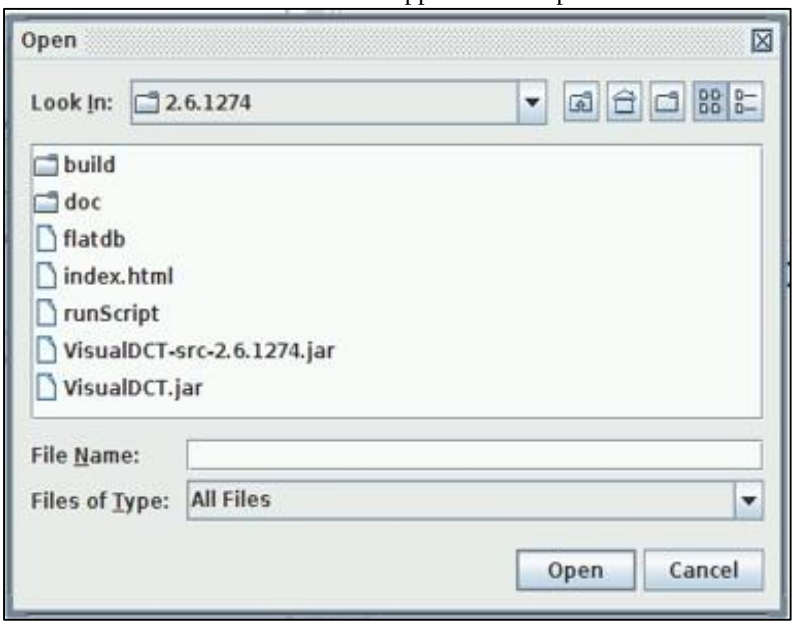



Fig. 4: VDCT window during opening

".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. USAGE OF EPICS IN RASPBERRY PI

### *A. Record Creation and its Processing in VDCT*

#### *1) Record*

As name suggest record contains the informational detail of process variable [4]. Type of record is defined by user. A record may give input to other records or accept input from other records. Record consists of various fields. Records can communicate with hardware with the help of device support. Device support for various hardware is available for Raspberry Pi. One can also write customized device support. Record can be created by specific software or can be created by a programming language. For sake of simplicity here we used a record creation software toolkit called visual database management tool (VisualDCT).

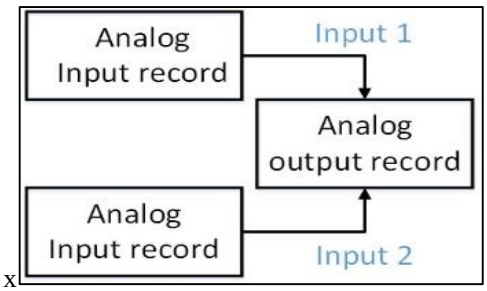
#### *2) Procedure for using VDCT in EPICS*

VDCT Requires dbd file for operation, this dbd file is created by application and the application directory is created by makeBaseApp.pl script.
Open a terminal and create a directory by using following commands.
cd vivek
su
cd vivek

Then enter the following script in the same directory to create an application. In the following commands addition is the name of application that is being created. Name of application should denote its purpose.
makeBaseApp.pl –t example addition
makeBaseApp.pl –i –t example addition
make

After the make command, a directory called addition will gets created. Inside that directory there is a sub-directory called iocBoot will be created. Move inside that directory with following command.
cd iocBoot
Finally, run ./st.cmd command to create IOC
./st.cmd
Giving the permissions to application with following commands
chmod 777 st.cmd
./st.cmd
It will create IOC and execute it.

Open another terminal and run VDCT. VDCT will ask to load dbd file, Load the dbd file by going to addition/dbd/adding.dbd. Make two records ai (analog input) record and one calc (calculation) record. Figure 5 shows the conceptual view of record processing in VDCT.




x

Fig. 5: Flow diagram of record processing

In ai records fill the required field. DTYP field of each record should “soft channel”, because no hardware is used now
DTYP=soft Channel
VAL=any numeric value
In calc record enter the following in the calc field

CALC=A+B

Where, A and B are the name of variable that is being added. SCAN field of each record was “1” as its represents the timing interval after which each record process.
SCAN 1 Second
ai record need a connection to calc record. Right click in calc record then INLINK then INPA, calc record will become reddish which indicates that calc record is asking for its input A, then click on first record, similarly do for second ai record save it with name add.db inside addition/addingApp/Db/add.db

Communication between records is done by using get and put command, get command was used to find the value of process variable while put command is used to change the value of process variable. Open a terminal and do the following as

dicussed previously as the creation of application. Following commands are used to enter in that application directory and running the startup command.
cd addition
make
cd iocboot/iocaddition
./st.cmd

To know the value of process variable and to set the value of process variable, caget and caput commands is used. CA stands for channel access. Open a terminal. Use following commands to get the value of process variable. Here a process variable is sum which stores the value of the addition of A and B, val1 is the process variable that stores the value of A and val2 is the process variable that stores the value of B
caget sum
caget val1
caget val2
The value of variable A and B can be change directly from the terminal by using following commands.
caput val1 3
The above command is setting the value A as numeric 3.

*3) Linking Device Support of GPIO with IOC and Record*

GPIO pins of Raspberry Pi are the medium to connect it with the external world. To access the GPIO pins of Raspberry Pi using EPICS, a device support is required. Download the devgpio device support from github ffeldbaur [5] and change the path of base in Configure/release file and then make
To make an application use following commands
mkdir ioc
cd ioc
makeBaseApp.pl -t example test
makeBaseApp.pl -i -t example test
Then go to configure file of application and add path where devgpio is made
DEVGPIO=/PATH/WHERE/devgpio/IS /MADE
Then goto testApp/src/makefile and add the following lines
test_DBD += devgpio.dbd
test_LIBS += devgpio
Save it and run make in the top of the application file to compile it. Open VDCT and load dbd file from the application. Create record bo (binary output)
name:val
DTYP field of record now should be devgpio because GPIO needs to be access through it.
DTYP=devgpio (devgpio should automatically come otherwise devgpio is not properly configured)
The following command is use to set pin number 20 as output and making its voltage level HIGH.
OUT =@20 H
goto save as and save it as suppose 20high.db under db directory of application file.
goto st.cmd of application and add the following lines under #load Record instances
dbLoadRecords”db/20high.db”(copy just above line of it and replace with 20high.db).
Save it and run it as ./st.cmd
After initialization of IOC run the command
dbpf val 1 (to set the pin high)
dbpf val 0 (to set the pin low)
The GPIO pin voltage status can be determined by using following command in the terminal of Raspberry Pi
gpio readall

## IV. CONCLUSION

Record creation using VDCT has been done. Record processing has been done by creating an application in EPICS. This application has been used to get and set the value of process variable. GPIO pins have been accessed using device support. It has found that Raspberry Pi supports EPICS. Florian Feld Bauer had written a device support to access the GPIO pins of Raspberry Pi through EPICS [5]. Device support for I2C is also available [6].

## REFERENCES

[1] Description of Experimental Physics and Industrial Control System
http://www.aps.anl.gov/epics/
[2] Description of VDCT

http://www.aps.anl.gov/epics/extensions/vdct/
[3] VDCT user manual page
http://www.aps.anl.gov/epics/extensions/vdct/2.6.1274/MAN-VisualDCT_Users_Manual.html
[4] EPICS record reference manual
http://www.aps.anl.gov/epics/EpicsDocumentation/AppDevManuals/RecordRef/Recordref-1.html
[5] Devgpio device support from github ffeldbaur
https://github.com/ffeldbauer/epics-devgpio
[6] Availability of device support for I2C
http://www.aps.anl.gov/epics/tech-talk/2016/msg00433.php/
[7] Writing EPICS driver and device support
http://www.aps.anl.gov/epics/meetings/2009-07/talks/em_WritingEPICSDrivers.ppt
[8] Richard Peterson, The complete reference 6E, Tata McgrawHill Edition
[9] Raj Kamal, Embedded system Architecture, programming and design 3E, McGrawHill Education
[10] Installation procedure in Raspberry Pi
https://prjemian.github.io/epicspi/
[11] Description of EPICS base
http://www.aps.anl.gov/epics/base/.
[12] Description of synApps
https://www1.aps.anl.gov/bcda/synapps.