KSTAR Widget Toolkit using Qt Library for the EPICS-based Control System

Sulhee Baek*, Sangil Lee, Mikyung Park , Hoonkyun Na and Myeun Kwon

National Fusion Research Institute (NFRI), Daejeon, Republic of Korea

ABSTRACT

The KSTAR Widget Toolkit (KWT) was developed as a development toolkit of channel access (CA) client application for the KSTAR commissioning. The KWT is based on Qt library and includes channel access interface to communicate with EPICS. In order to enhance development speed and increase aesthetic quality of application, 18 plug-in widgets were developed to enable for developers to create new panel using drag and drop method. Some of them use QWT as a plotting library and some widgets display alarm status with a specified color according to the EPICS alarm convention. The KWT has cross-platform development environment and feasibility of extending new widgets using Qt plug-in API with plenty of documents and tutorials. Around 120 panels and several applications such as multi-channel plotting tool, process variable searching tool, and logbook application were developed through the KWT and they proved functionality of the KWT being used for the integrated control and machine control during the KSTAR commissioning. The KWT is applicable to fast and easy development of operator interfaces and applications for the EPICS-based control system.

DEVELOPMENT STATUS

Items listed below show the requirements for the KSTAR Operator Interfaces(OPIs) or the development toolkit of them.

Requirements for KSTAR OPIs;

• Performance
• Stable EPICA CA communication
• Easy & fast development
• Maintenance
• Usability
• Consistency of appearance

Features of KWT Library

KWT inherits the features listed below from Qt library.

• Intuitive C++ class library
• Portability across desktop and embedded operating systems (uses Qt cross-platform development environment but Qt-CA interface was not fully tested on platforms except linux)
• Integrated development tools with cross-platform IDE (it's possible to use Qt designer, which is a Qt cross-platform integrated development environment (IDE), to make OPI panels because KWT widgets were designed as custom plug-in widgets for it.)
• High runtime performance and small footprint on embedded

Inheritance Hierarchy of Library Properties

KWT libraries depend on Qt-4.3.2 and some plotting widgets among them inherit properties from QWT libraries. In addition, to communicate with EPICS, Qt-CA library in KWT interfaces with EPICS base-3.4.18.2. Some classes use boost library as a Standard Template Library support package. The libraries listed below are prerequisite libraries for the KWT installation and inheritance hierarchy of the KWT is shown in Figure 1 in detail.

• Qt -4.3.2
• QWT-5.0.0rc1
• EPICS base-3.14.8.2
• Boost library

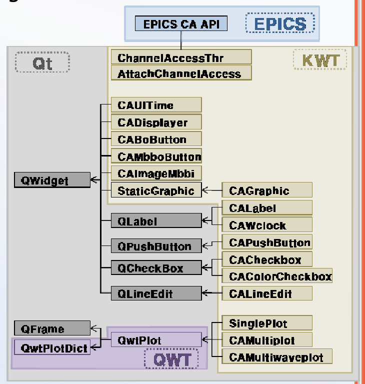


Figure 1: Inheritance hierarchy of the KWT

Principal of the Qt-CA Interface

The core libraries of KWT Qt-CA interface are AttachChannelAccess library and ChannelAccessThr library. Working procedure of AttachChannelAccess is drawn in Figure 2 briefly. Usually this library is used for a widget containing child widgets of a UI file created by Qt designer. After initialization, this instance acquires list of CAobjects from the QWidget or UI file. For every CAobjects it links event filter and work thread. If the CAobject has control property, it is registered as control object. The work thread created using ChannelAccessThr class updates all hash tables for all CAobjects. Hash table structure named by ChAccess which is updated by work thread is shown in Figure 3. It is updated at every pre-defined period or CA monitor event.

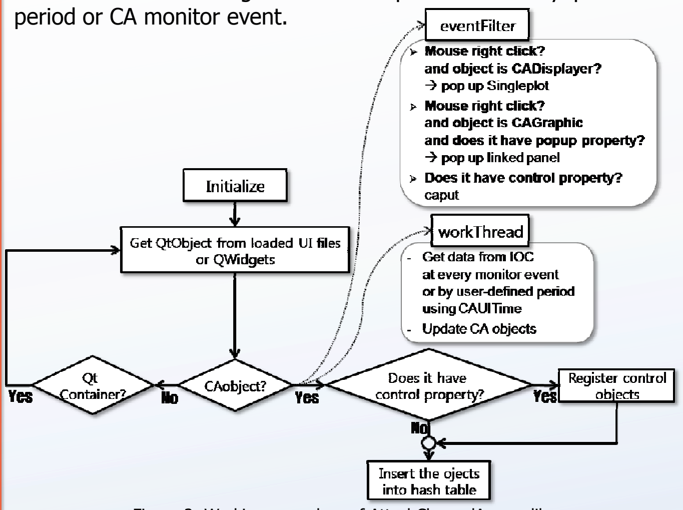

eventFilter
Mouse right click? and object is CADisplayer? → pop up Singleplot
Mouse right click? and object is CAGraphic and does it have popup property? → pop up linked panel
Does it have control property? caput
workThread
Get data from IOC at every monitor event or by user-defined period using CAUITime
Update CA objects
Initialize
Get QtObject from loaded UI files or QWidgets
Qt Container?
CAobject?
Does it have control property?
Register control objects
Insert the ojects into hash table
Yes
No
Yes
Yes
No

Figure 2: Working procedure of AttachChannelAccess library

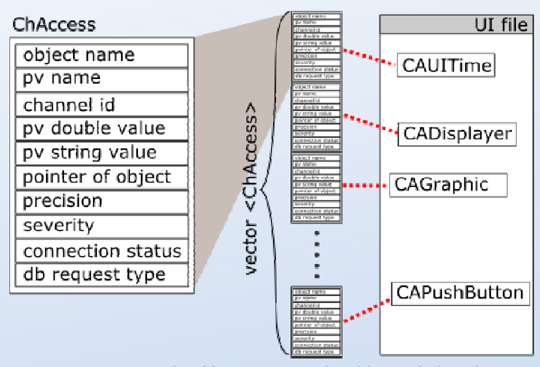


Figure 3: Hash table structure updated by work thread

KWT Widgets

The KWT including 18 widget classes are shown in Table 1 and brief explanations about each widgets are as follows.

Table 1: Brief explanations about the KWT widgets

Widget name	Data type	Appearance	Feature
AttachChannelAccess	N/A	Hidden	• Qt-CA attach library
ChannelAccessThr	N/A	Hidden	• Qt-CA thread library
CAUITime	N/A	Hidden	• Define screen rate (Periodic or Event-driven) •Master PV locks control widgets
CADisplayer	EPICS AI	2.12e-05	• Displays numeric data with the corresponding alarm information with a specific color. • Pops-up SinglePlot with right clik
CABoButton	EPICS BO	CLOSE OPEN	• A pair of QPushButtons with a false button and a true button
CAMbboButton	EPICS MBBO	이상 정지 준비 운전	• A collection of multiple CAPushButtons for EPICS mbbo record
CAImageMbbi	EPICS MBBI	Vac. Pump Cooldown Charge-up	• A collection of multiple QLables for EPICS mbbi record
StaticGraphic	N/A		• A symbol library including vacuum devices, arrows, ellipse, rectangle, etc.
CAGrahpic	EPICS BI		• Inherits symbols from StaticGraphic lib. and displays alarm status
CALabel	EPICS BI, AI, String, MBBI	NORMAL	• Displays text label corresponding to the value
CAWclock	EPICS timeStamp	2007/08/01 18:01:49	• Displays time information as text label
CAPushButton	BO	OPEN	• A QPushButton which sends operator's command to the PV
CAColorCheckbox	PVname and plot color	✓ VMS_CR_TPG262C0_P0 ✓ VMS_CR_TPG262C1_P0	• A part of CAMultiplot or CAMultiwaveplot widget which displays color information
CALineEdit	EPICS AO	0.0	• A QLineEdit which sends entered numeric data to the PV
SinglePlot	EPICS AI		• A single channel plotting class using QWTPlot library
CAMultiplot	10 EPICS AI		• A multi-channel plotting class using QWTPlot library for ai record
CAMultiwaveplot	10 EPICS Waveform		• A multi-channel plotting class using QWTPlot library for waveform record

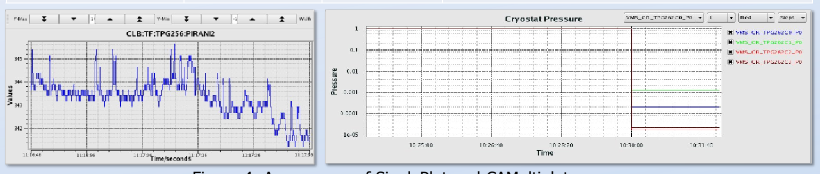


Figure 4: Appearance of SinglePlot and CAMultiplot

OPI Development Example with the KWT

Widgets in the KWT were designed as plug-in widgets to be inserted using Qt designer. Below procedures show a simple example using the KWT.

•Create a Widget using Qt designer.
•Drag a CADisplayer instance from KSTAR widget box and drop it to the proper position.
•Configure it with a valid PV name. Figure 5 shows screenshot of Qt designer inserting KSTAR widget into a UI file. PV name property is shown in the property editor which is located in right lower part.

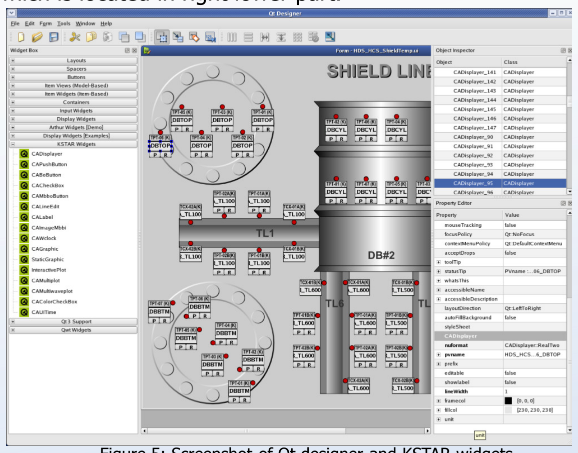


Figure 5: Screenshot of Qt designer and KSTAR widgets

•Save the widget as uifilename.ui.
•Develop a simple main application(main.cpp) referring to Table 2.

Table 2: A simple example of main application

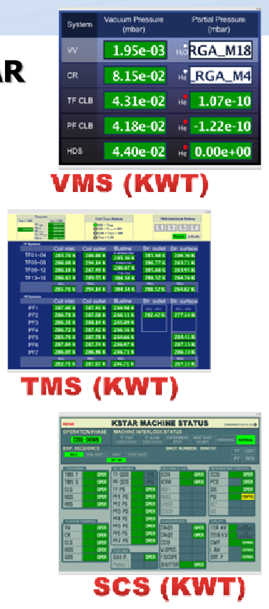
```
#include <QtUiTools>
#include <QtGui>
#include "qtchaccesslib.h"
int main (int argc, char *argv[])
{
QApplication app(argc, argv);
 AttachChannelAccess attach("uifilename.ui, 1);
attach.SetUiCurIndex(1);
QWidget *pwidget = attach.GetWidget();
Pwidget->show();
Return app.exec();
}
```

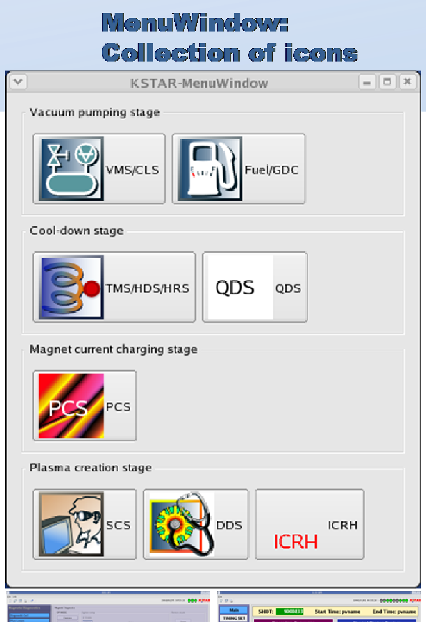
•Compile the main application.
•Execute it.

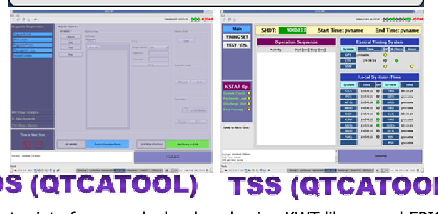
APPLICATIONS OF KWT TO KSTAR

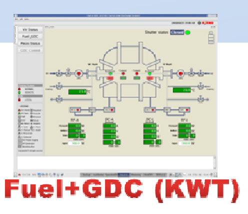
KSTAR OPI Panels

Over than 120 OPI panels were developed using KWT and about 50 panels were developed using other EPICS extensions such as QtCAtool, MEDM, EDM. During the KSTAR commissioning, they were used for remote operation without any serious problems. The OPI panels with KWT were well accepted by the operators because of the simple panel switching and consistent appearance. Besides OPI panels, some applications such as multi-channel plotting tool, process variable searching tool, and logbook application were developed using KWT library.









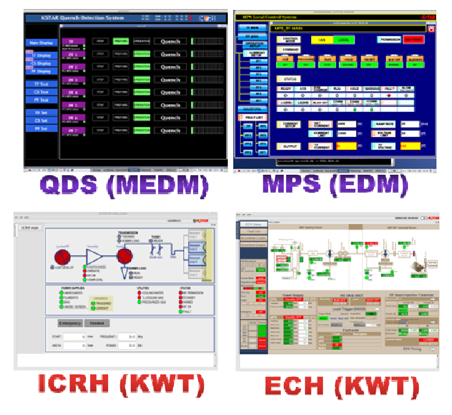


Figure 6: Operator interface panels developed using KWT library and EPICS extensions

LICENCE POLICY AND S/W RELEASE

KWT is available as free software under the GNU General Public License (GPL). The source code for the KWT will be released on SourceForge in the CVS repository of the KWT project.

CONCLUSION

Even if it took more time and effort to develop not only OPI panels but also development toolkit, KSTAR OPIs developed using KWT library satisfied almost requirements concerning performance, easy & fast development, nice maintenance, usability, and consistency of appearance. However, abnormal stop occurred intermittently at some CA server down should be fixed as soon as possible to enhance stability of CA communication.