Design of Virtual Instrument Interface Based on Embedded Linux+QT

Yan-jun DI\textsuperscript{1}, Ni JIN\textsuperscript{2,*} and Li-guo WANG\textsuperscript{1}

\textsuperscript{1}Shenyang EWAY Co., Ltd. Shenyang China
\textsuperscript{2}Shenyang institute of Automation Chinese Academy of Sciences, Shenyang China
*Corresponding author

Keywords: Linux, QT, QML.

Abstract. This paper mainly summarizes the technical scheme of replacing traditional pointer automotive instrument with full liquid crystal by choosing hardware platform and software platform. Emphasis is laid on the application of QT graphical development environment in embedded Linux operating system. The full liquid crystal can solve the display and instruction of various monitoring information on automotive instrument. Qt Quick tool is used in the dynamic pointer design of full liquid crystal instrument, and how to use QML (descriptive scripting language) and C++ language mixed programming technology. A practical and mature instrument system can be designed for users, and a set of reasonable construction ideas and methods can be provided.

Introduction

Compared with the traditional automobile instrument, the contents of the full liquid crystal instrument are more abundant, flexible and beautiful. Full liquid crystal instrument can display what according to vehicle driving state and user driving habits, it focus on what drivers are most concerned about. As the center of human-machine interaction, it is far more than traditional vehicle instrument. Therefore, the emergence of full liquid crystal instrument is a revolutionary progress in the history of automotive, it will become an important development direction of automotive instrument.

This paper mainly introduces a full liquid crystal instrument developed with embedded Linux operating system and QT graphical software platform.

Full Liquid Crystal Instrument Solution

System interface composition as shown in Figure 1.

![Figure 1. System block diagram.](image-url)
Controller

Main MCU Processor. The main MCU processor uses NXP Cortex-A9 four core i.MX6Q processor. It’s main frequency is 1GHz, and it has 2G DDR3 memory, 16GB EMMC storage, I.MX6 development platform. The main MCU processor has the advantages of high quality and strong stability which can develop the product more flexibly and design more competitive products quickly. The operating system supports Android6.0, Android4.4.2, Linux4.1.15+QT5.6, Linux3.0.35+QT4.8.5 and so on.

Slave CPU Processor. The CPU uses STM32F105 processor, which is the Cortex-M3 core. The highest CPU speed is 72 MHz. STM32 processor is widely used in the market. It can be developed in IAR and KEIL environments. It can be applied to applications that require connection functions and real-time performance, such as, industrial control, security application control panel, UPS and home audio. STM32F05 has 64-256KB on-chip Flash memory, 64KB SRAM and 14 communication interfaces, two CAN buses, which can meet the requirements of automobile interface control and monitoring.

Display Screen

At present, the 12.3 inch screen is the mainstream full liquid crystal instrument. liquid crystal panel manufacturers mainly include Sharp, Samsung, LG, Philips, Youda Optoelectronics, QiMei Electronics, etc. There are many choices on the market. Liquid crystal screen can be selected reasonably according to the specific demands of automobile manufacturers. The full liquid crystal instrument mainly changes the instruction mode of the traditional instrument information monitoring. According to the user's driving habits, it can display information reasonably, such as engine speed, speed, water temperature, oil temperature, oil pressure and so on.

Embedded Linux + QT Software Platform

Among all operating systems, Linux is the fastest growing and most widely used operating system. The nature of Linux itself makes it the first choice for embedded development. Embedded Linux is widely used in communications, information, digital home, industrial control and other fields. Its open source code can greatly reduce the cost of development and improve the development efficiency of products.

Qt is a cross-platform C++ graphical user interface application development framework developed by Qt Company. It can be used to develop GUI programs as well as console tools and servers. Qt is an object-oriented framework that uses special code generation extensions and some macros. Qt is easy to extend and allows for true component programming.

Qt is support for X11, X86, ARM, and Symbian is the first opportunity for developers to develop the same code base for both platforms. The ability to share code across platforms means that applications developed by developers will be faster to market and will have a broader user group.

This project mainly uses the embedded Linux + QT software platform, which is also making full use of its own advantages and features.

QWidget and QML Mixed Interface Design

The QWidget class is the base class for all user interface objects. The QMainWindow class provides a main application window with a menu bar, a dock window (such as a toolbar), and a status bar which Inherited from QWidget. The QDialog class is the base class for dialog windows which Inherited from QWidget. The QFrame class is the base class for framed widgets and inherits from QWidget.

In the design of the full LCD instrument, information such as sensor values and warning lights and other data (as shown in Figure 2) need to be displayed on the interface. The traditional instrument uses the pointer method to indicate the speed of the car, the speed, the water temperature and the amount of oil, which is deeply loved by the user. It is necessary to display the pointer on the full liquid crystal. The problem is that it is slightly more difficult to use the controls designed by the QWidget class to make dynamic pointers on the disk. Fortunately, QT supports the Quick tool, which is a good solution to this problem.
In the industry standard, the traditional instrument regulations that when the signal changes smoothly, the pointer movement should be stable, there should be no stuck phenomenon. In order to achieve a standard that is not lower than the traditional instrument, the acquisition speed is increased to make the pointer rotate smoothly.

![Dashboard display interface.](image)

In the two large dials in Figure 2, the information display of speed of the vehicle, speed, water temperature and oil quantity is displayed by the Qt Quick tool. Qt Quick is a script language (similar to css), which is a combination of QML and C++. It is more suitable for interface classes. The QML development interface actually uses the QGraphicsView architecture. The advantage of using QML to develop the interface is that the development cycle is short and the portability is strong. The advantage of QWidget is that it is very convenient to interact with other non-GUI modules, and you don't need to use the classes in the Declarative module.

**Quick Qml Programming Case**

**Configuration in Engineering Pro.** This is the identifier added by the Qt module:

In the pro file as follows:

```cpp
QT    += core gui            // Gui is a graphical interface library
QT    += quick qml
# indicates that the quick and qml modules have been added, and then you can reference the quick component in your cpp.
```

**Mixed Programming of QML and C++ Classes.** If there is not any graphics components in InterfaceOne.qml, or if you need to avoid using QDeclarativeView for other reasons, you can create QDeclarativeEngine directly. In this case, InterfaceOne.qml will be loaded as a QDeclarativeComponent instance instead of being displayed in a view.

```cpp
#include <QDeclarativeEngine>
#include <QDeclarativeContext>
#include <QDeclarativeComponent>

MeanForm oMeanForm;          // QmainWindow class
CollectThread m_CollectThread;   // Collection thread class

//If there is a user interface based on a graphical view framework, then you can use this method to directly integrate QML components into QMainWindow. For example, you can integrate QML component into QMainWindow.
QDeclarativeContext *context= oMeanForm.quickwidget->rootContext();
//Register C++ objects to qml
context->setContextProperty("m_CollectThread",&m_CollectThread);
QObject *rootObject = dynamic_cast<QObject*>(oMeanForm.quickwidget->rootObject());
QObject::connect(&m_CollectThread,SIGNAL(MeterDisParaSig(QVariant,QVariant,QVariant,QVariant)), rootObject,SLOT(meterDisParaSlot(QVariant,QVariant,QVariant,QVariant)));
```

369
It is mainly use the signal slot mechanism of QT to define the signal in CollectThread.

class CollectThread : public QThread
{
    ........
    signals:
    void MeterDisParaSig(QVariant, QVariant, QVariant, QVariant);
    ........
}

// Define a slot in InterfaceOne.qml to receive the speed of the vehicle, rotating speed, oil, and coolant temperature values in the acquisition thread.

function meterDisParaSlot(engineS, vehicleS, oilP, coolantT)
{
    ........
}

Drawing a Dial Pointer

Image {
    id: needle       // Pointer ID
    x: 173; y: 180    // Pointer coordinate value
    source: "Meter/Pointer-01.png"  // Pointer picture position
    transform: Rotation  // Pointer rotation
    {
        id: needleRotation
        Math.min(Math.max(0 , rootA.vehicles), 240)   // Indicates that the rotation range is between 0 and 240 degrees.
        Behavior on angle
        {
            SpringAnimation  // Flexible animation
            {
                spring: 1       // Elastic trait
                damping: 0.2     // Damping degree
            }
        }
    }
}

Summary

This paper mainly introduces the software and hardware components of the full LCD instrument, let the reader understand the key technologies related to it and how to turn the technology into products. The technical platform of the full LCD instrument solves the core technology related to the automobile instrument well, and its system composition and design ideas are also applicable to other terminal designs with liquid crystal screens.

Acknowledgement

This work is supported by the Shenyang science and technology plan project of China under contract F16-073-2-00.

References


