Using Sample Code to Develop Embedded Applications Saves Time and Keeps Bugs at a Minimum

Dr. CF Lin, Director of Software Development
cf.lin@moxa.com

It is often the case that software developers build application programs based on sample code or code from previous projects. Building on your own experience or the experience of others is common practice in the software industry, since doing so allows developers to deliver products faster, and also helps guarantee that the source code is relatively bug-free. In particular, when developers are faced with the challenge of writing code for applications new to the programmer, relying on sample code to get started goes a long way to keeping frustrations at a minimum.

Sample code is a great educational tool for young software developers who are just getting started with embedded application programming. One of the biggest benefits of using sample code is that programmers can quickly grasp the full functionality of their applications, and in this way gain the confidence needed to complete their project.

Sample code is a good reference for developers who are already familiar with application programming. For a particular application, the code can be used to demonstrate different angles of implementation, and this kind of experience sharing provides a good focal point for mutual learning.

Sample code also delivers a more precise means of sharing technology between vendors and applications developers. In

Released on May 1, 2008

How to contact Moxa
Tel: 1-714-528-6777
Fax: 1-714-528-6778
Web: www.moxa.com
Email: info@moxa.com

This document was produced by the Moxa Technical Writing Center (TWC). Please send your comments or suggestions about this or other Moxa documents to twc@moxa.com.

particular, vendors often create proprietary software interfaces for accessing special functionalities provided by hardware components. Since these types of functionalities are not in the public domain, providing relevant sample code may be the only way for application developers to understand the functions, after which they can proceed to write their own code.

Serial Port Programming Example

Perhaps the most widely used sample code provided by Moxa is for serial communication applications. The code provides an interface for accessing Moxa's advanced ASIC chip used for serial communication. This chipset follows the UART 16550 standard, but outperforms the standard in terms of extensibility and data throughput. The chipset also supports a wide range of baudrates up to 921.6 Kbps, including those not used in standard Windows and Linux programming. To access these special baudrates, a Linux user can utilize the following code segment.

```
int mserial_port_set_special_baud_rate( int fd, unsigned int speed)
{
        struct termios termios;
        tcgetattr(fd, &termios);
        termios.c_cflag &= ~(CBAUD | CBAUDEX);
        /* a rate not one of them defined in the table */
        termios.c_cflag |= B4000000;
        return ioctl(fd, UC_SET_SPECIAL_BAUD_RATE, &speed)? -1:0;
}
```

Another useful function in the serial chipset arsenal is selecting which serial interface to use for a 3-in-1 serial port. Selecting either the RS-232, RS-422, or RS-485 interface can be done using the following sample program.

```
Int mserial_port_set_interface(int fd, int iface)
{
        if (iface != RS232_MODE && iface != RS485_2WIRE_MODE &&
            iface != RS422_MODE && iface != RS485_4WIRE_MODE )
{
        printf("not a supported interface\n");
        return -1;
        }
        if (ioctl(fd, MOXA_SET_OP_MODE, &iface) != 0)
```

```
        {
        printf("fail to set serial port interface\n");
        return -2;
        }
        return 0;
}
```

The Complexities of Socket Programming

Socket programming may seem easy enough for developers who design client-server applications, but sophisticated programming skill is required to create a truly stable client-server application. Unfortunately, many programmers start with a simple sample program, but quickly run into trouble as their client-server code increases in complexity. Socket programs for embedded applications require a very high degree of stability and can be quite complicated. Such programs not only involve reading and writing data to and from socket ports, but also require maintaining a well-designed structure for managing multiple clients, handling client connection and reconnection, taking care of non-blocking read and write operations, and other tasks.

The degree of reliability required for a client-server system can be extremely high, and inexperienced developers often have trouble creating such systems. In fact, based on the author's own experience, new programmers often invest a great deal of effort before fully understanding where the problems lie. Proceeding in this manner can take a long time, and also increase your development costs to unacceptable levels.

Well-structured sample code helps application developers minimize the difficulty of creating a stable system, and provides a good way for programmers to develop a better understanding of socket programming. The bottom line is that you can reduce your development costs by using reliable sample code.

Creating a Watchdog for Embedded Applications

Since embedded system applications are often left unattended for long periods of time, most embedded applications implement a watchdog function to ensure that the computer system can reboot automatically when the system hangs unexpectedly. Implementing such a function for RISC-based computers is usually system-dependent, and for this reason, using sample code to demonstrate how to use such a function can be extremely valuable.

Summary

Software developers should always be on the lookout for ways to ensure that products are developed quickly, but at the same time are also reliable and cost-effective. A simple yet very effective way of doing this is to rely on sample code when getting started with a project. Using sample code not only speeds up product development, but also helps ensure that code is efficient and bug-free.

Embedded applications that can benefit from using sample code include client-server programs, watchdogs, LCDs, and DIOs, to name but a few.

NOTE: Moxa provides sample code for a wide range of embedded applications. For details, please contact the author, Dr. CF Lin, at support@moxa.com.