STARS .NET INTERFACE FOR WINDOWS CE

Takashi Kosuge, KEK, Ibaraki, Japan

Abstract

Simple Transmission and Retrieval System (STARS) [1] [2] [3], a message transferring software for smallscale control systems using TCP/IP sockets, is installed in various control systems (beamline control system, access control system of experimental halls, key handling systems, etc.) at the Photon Factory (PF). Herein, we have developed a new STARS .NET interface library for Windows CE by remodelling the original STARS .NET interface. This library enables easy development of STARS GUI clients or I/O clients. Recently, we tested the library on HP iPAQ 112 with the help of a simple GUI program (written in C#), and it gave satisfactory results. We will describe in detail the STARS .NET interface for Windows CE and provide examples of its applications.

OVERVIEW OF STARS AND ITS RECENT STATUS

Simple transmission and retrieval system (STARS) consists of a server program (STARS server) and client programs (STAR clients). Each client is connected to the server via a TCP/IP socket (Fig. 1), and it communicates via text based messages. A STARS server is written in Perl and can run on various operating systems.



Figure 1: STARS server and clients.

Messages on STARS

Each client program has its own unique node name and sends text-based massages to the server. A message on STARS consists of three parts and is terminated by "LF" (0x0a). The first part comprises the node name of the source (optional); the second part comprises the destination node; the last part is used for a command, reply, or an event. The STARS server delivers the message to the corresponding client on the basis of the destination node of the message.

Classical Topics

Certification on STARS

STARS has a simple certification procedure which is used at connection time. It has three steps as follows:

- Host name certification
- Node name and keyword certification
- Node name and host name certification (optional)

These three steps prevent the occurrence of problems such as invalid connection.

Installation Status of STARS

STARS has been installed in various control systems at the Photon Factory (PF), while its installation in beamline control systems is still in progress. The installation status (installed) is as follows:

- PF Beamline Control Systems: BL-1A, BL-3A, BL-4B, BL-5A, BL-6A, BL-6C, BL-8B, BL-9A, BL-9C, BL-16A, BL-17A, BL-18B, BL-20A, NW-2, NW-10, NW-12, NW-14, slow positron beamline
- Other Systems: Beamline interlock system, access control system for experimental halls, and key handling system

STARS INTERFACES

STARS client programmers are required to use TCP/IP sockets and handle text-based messages. At present, a programmer can choose from a variety of programming languages that support TCP/IP sockets. Developing STARS clients is very easy for programmers with prior knowledge of TCP/IP socket programming, though beginners may find it slightly difficult. STARS is equipped with interface libraries for some programming languages. Every STARS interface library supports keyword certification, and hence, programmers need not worry about TCP/IP socket programming. Developing STARS clients with the help of STARS interface libraries is extremely simple.

STARS .NET INTERFACE

STARS .NET interface is one of the interface libraries that work on .NET Windows and Mono. The interface consists of two types of DLL files- "StarsInterface.dll," which supports the main part (connection, message handling, etc.) of the interface, and "StarsInterfaceWinForm.dll," helps which in the development of Windows Form GUI. We have ported this STARS .NET interface to Windows CE.

DEVELOPMENT OF CLIENTS WITH STARS .NET INTERFACE FOR WINDOWS CE

Preparation

The professional edition of Microsoft Visual Studio is equipped with a default Pocket PC environment. Here, we will describe the development of STARS client for Windows CE (Pocket PC 2003) with the help of an example using Visual C# in Microsoft Visual Studio 2005 Professional Edition as an example.

Project templates for "Pocket PC 2003" are displayed on selection in the following order "File," "New," "Project," "Visual C#," "Smart Device," "Pocket PC 2003," and programming is started by choosing the corresponding template (Fig. 2).

	Project_	Ctrl+Shift+N	*		- 2	- 💎 🖗	
Open	Meb Site_			• ×	Solution Explorer	+ # ×	
Close	Eile	Ctrl+N		^	6		
Close Solution	Project From	Existing Code					
Save Selected Iter Ctrl+	s pluuio 2	005					
Save Selected Items As							
Save All Ctrl+Shift+		MSDN: Visual C#					
Export Template	w Project					2	
Page Setup	Project types:		Templates:			11 1	
	P Visual C#	B Visual C#		Visual Studio installed templates			
Open: Project. Create: Project. Getting Started	V V V V V V V V V V V V V V V V V V V	Booter HP 2000 Smartphone 2003 Window (2 6 3) Window (2 6 3) Database Other Languages Cother Project Types		20 Control Lbay 第Control Application 目目時から和epication 10 同じたい、利用したいのでは、 同じたいのでは、 日日のででは、 日日のでは、 日日のでは、 日日のででは			
What's new in C# 2005 Create Your First Appli Use a Starter Kit How Do 1? Learn Visual C#	catic						
Connect With the Comr Download Additional Co	A project for creat	A project for creating a .NET Compact Framework 2.0 forms application for Pocket PC 2003 and later					
	Name:	Name: Device Application1 Location: C-WDocuments and S					
				SettingsWicosugeVMy DocumentsVVisual Studio 2005WProjects		Browse.	
	Location	C-FEOCOMENIAS and I					

Figure 2: Creating a new pocket PC project.

The next step involves copying DLLs from the STARS distribution package into the selected project template and adding a reference in the "Solution Explorer" (Fig.3). The "key file" of STARS must be manually modified in "bin\Debug" and "bin\Release."



Figure 3: Adding a reference using solution explorer.

Methods

The STARS .NET Interface for Windows CE was ported from the original STARS .NET Interface. The

Classical Topics

same methods as in the original were used in the ported interface as well.

Before employing these methods, the object needs to be defined as follows:

using STARS;

using STARS.WinForm; //If you use Windows Form

//Definition of object
 StarsInterface stars
 = new StarsInterface(myNodeName, starsServerName, kevFileName, starsPort);

In the above example, "STARS" is used as the object name. "myNodeName," "starsServerName," and "keyFileName" are string values. "starsPort" is an integer with a default value of "6057."

Connect

The "Connect" method is used for establishing connection. If a programmer calls the "stars.Connect();" method, connection to STARS is automatically established through the keyword checking process. Furthermore, this method throws an exception when it fails to establish a connection.

Send

The "Send" method is used to send messages to the STARS server in the following manner:

//Send "GetValue" command to a node named "Dev1". stars.Send("Dev1 GetValue");

The message will be delivered to client "Dev1" by the STARS server.

Receive

The messages received by the client program can be read from the receive buffer by using the "Receive" function as follows:

StarsMessage rcvMesg = stars.Receive();

"StarsMessage" has the following features:

- "allMessage" stores all messages (from, to, messages).
- "from" is the message source node name.
- "to" is the destination node name.
- "command" retains the command part of the message.
- "parameter" keeps some of the parameters used in the message.

Callback Functions

Windows CE also supports a callback function on the STARS .NET interface. When a message arrives from the STARS server, the function set by the programmer is

called by using "StarsCbHandler." "StarsCbWinForm" is also used for Windows Form program. An example is shown below.

private void Form1_Load(object sender, EventArgs e)
{

. // Add the stars handler named 'handler' which is called // when arriving stars messages. StarsCbHandler cb = new StarsCbHandler(handler);

// For windows form.
StarsCbWinForm cbStars = new StarsCbWinForm(this,
stars);

// Callback mode will be started with this method. cbStars.StartCbHandler(cb);

```
:
}
```

{

2

//The method is called by callback function when a //message arrives.

private void handler(object sender, STARS.StarsCbArgs ev)

// Receiving Stars message by callback mode. Show
// reply at messagebox.
MessageBox.Show(ev.allMessage);
return;

```
CONTROL PANEL WITH POCKET PC
```

We have developed a multipurpose GUI control panel for Pocket PC 2003 with the STARS .NET interface for Windows CE. Fig. 4 shows a snapshot of the GUI control panel programming using Visual Studio 2005.



Figure 4: Programming of the GUI control panel.

Users can remotely control pulse motor controllers etc. through the GUI control panel via wireless LAN. Fig. 5 shows a control panel that runs on HP iPAQ 112.



Figure 5: Control Panel with iPAQ 112.

CONCLUSION

We have developed a STARS .NET interface for Windows CE and sufficiently validated its efficiency. With this novel interface, the Pocket PC is expected to emerge as a powerful STARS tool.

REFERENCES

- [1] http://stars.kek.jp/
- [2] T. Kosuge, et al., "Recent Progress of STARS", PCaPAC2005, Hayama, 2005.
- [3] T. Kosuge, et al., "The Interconnection of TINE and STARS", PCaPAC2006, Newport News, 2006.