IP-based Software for Sensory Control and Data Acquisition

William Emmanuel S. Yu

Abstract—**Sensory Control and Data Acquisition (SCADA) Systems play a vital role in ensuring that distributed systems are maintained well. However, most of these system are specialized and have specialized or proprietary interfaces. The Sensory Control and Data Acquisition - Total Environmental Management System (SCADA-TEMS) uses an IP-based interface. This allow multiple networks types to have access to the SCADA-TEMS. The SCADA-TEMS uses the Simple Access and Control Protocol (SACP) that provides an interface for the different clients to access the system. Some of these clients are a Java client, web-based PHP client and a SMS-to-SACP gateway. The aim of this SACP is to provide a simple prototype protocol for controlling different external devices. The SCADA-TEMS also has a flexible and extensible pluggable module interface for adding new hardware.**

Keywords— **Sensory Control and Data Acquisition, Computer Interfacing**

I. Introduction

"A good products is the perfect balance between hardware and software."

This is a common saying in integrated hardware-software development houses. An implementation in hardware tends to have better performance and speed. An implementation in software tends to be more flexible and cheaper. Transmeta, a California-based microprocessor firm, is really taking to heart this saying with their new Crusoe microprocessors. The Crusoe is an Intel 386 (i386) compatible microprocessor that has a number of feature that give it the edge over other i386 processors. Some of these features include low power consumption, low heat dissipation and smaller and lighter form factor. How was Transmeta able to do this? These engineers built a hardware-software microprocessor instead of the traditional hardware only microprocessors.

The SCADA-TEMS team also took this saying into serious consideration too. Where do we draw the line between hardware and software? Certain aspects of the systems would really benefit from being implemented in software. The entire accessibility aspect, which includes the SACP implementation, was done with software. Aspects such as some aspects of data manipulation is easier and cheaply done with software. The SCADA-TEMS team took extra care in ensuring that the perfect mix of software and hardware went into the SCADA-TEMS. This paper aims to discuss one side of the coin-the Software.

II. Features

The SCADA-TEMS is composed of many different elements. The main software element is the SCADA-TEMS server. This server implements a good number of feature. The more prominent of these features will be discussed below.

Department of Electronics, Computer and Communications Engineering, Ateneo de Manila University

A. Dynamically Loadable Modules

Code for hardware interfacing is placed in this portion. The software includes code for the proper conversion of data to the proper units of measure. It also includes the code for probing system status and levels. Each software module is composed of three main methods. The first methods is the status retrieval method which is called to return the status of a sensors or a device. The next method is the control invocation method that is called to change the status or output of a device. The last method is the system control method. This method should contain code to manipulating the sensor or device data. For sensors, the system control method is typically empty unless the sensors need some auto recalibration mechanism.

For example, if the system control method of the air conditioning unit should call the status retrieval method of the temperature sensor and send the appropriate output to the air conditioning unit by calling it's own control invocation method.

In order to add support for new hardware, it would sufficient to write the appropriate methods for the new hardware and compile them into shared objects. The configuration file should then be modified to include the new object. This gives hardware developers and system intergrators a means to add new hardware without having to recompile the entire SCADA-TEMS server source code.

B. Network Daemon Implementing SACP

Main access to the system is done by opening a socket connection to the SCADA-TEMS server in port 6543 (this port number can be changed when necessary). The network daemon code allows SACP clients to interact with the SCADA-TEMS server. The SACP clients can pass commands and retrieve status from the SCADA-TEMS server. The network daemon code also provides the proper system checks for invalid commands and improper invocation.

C. User Authentication

Security is a main issue with distributed and open access system. In security circles, an issue is that increased accessibility also increases the likelihood of security breaches. The SCADA-TEMS server, being an open access system, is also prone to these security problems. However, a user authentication scheme is implemented to reduce these problems.

The SCADA-TEMS server does two types user authentication. The first type is standard UNIX passwd+crypt authentication. This is programmed for systems that only need a simple user authentication mechanism. For more restricted security requirements, the second type should be used. The second type is Pluggable Authentication Modules (PAM). PAM was introduced by SUN Microsystems in their Solaris platform to allow systems administrators to choose their user authentication mechanism without having to change the running applications that need these mechanisms. Simple PAM authentication can be performed with a regular password+DES mechanism or a more complicated system using crypto cards, bar code readers and other gadgets. The prototype SCADA-TEMS system uses PAM with a password+DES+MD5 mechanism.

D. MPEG2-Layer3 Decoder, Audio Mixer and Playback Library

The SCADA-TEMS server comes with a built-in MPEG2-Layer3 audio manipulation and playback library. The library contains a number of components. The first component is the MPEG2-Layer3 (MP3) decoder. This portion is based on the code of mpg123 which is a open-sourced MP3 player. The audio playback uses the esound libraries. These are open-sourced sound packages to providing cross platform audio playback. Esound was chosen to provide greater cross platform compatibility. The next portion is the mixer adjustment code which allows the user to change certain mixer settings such as volume, bass, treble and others. All these portions were taken together and compiled into the libmp3 library which can be called from any C program in a supported platform. The libmp3 library can be separated from the SCADA-TEMS code and used in another program.

E. ECP Control Library

This is some routines for manipulating the Extended Communications Port (ECP). The code is a library that was extended from the parpin another open-sourced parallel port manipulation library. The extensions include some data conversion extensions, control line manipulation and an internal software latch for maintaining output settings. The library is called libparport. Similar to libmp3, this code can be used outside the SCADA-TEMS server.

These are the major features of the SCADA-TEMS server. There some other features implemented in the SCADA-TEMS server. A complete list can be found in the TODO file in the SCADA-TEMS server source code distribution.

III. Simple Access and Control Protocol

The Simple Access and Control Protocol (SACP) is an IP-based protocol that is similar to an FTP server's responses. The responses are 1xx for in progress, 2xx for success, 3xx for more information needed, 4xx for temporary failure, and 5xx for permanent or critical failure. Below is a simple SACP session.

```
S: 100 SCADAD v1.0\r\n
E: user <login>\r\n
S: 300 Enter Password\r\n
E: pass <password>\r\n
S: 500 Authenticated.\r\n
E: quit\r\n
S: 200 Bye.\r\n
```

Some simple SACP commands are:
⋆ set ¡device¿ ¡status¿ ¡level¿ - this command is used to set the status and output level of the devices. Current devices supported are lights, aircon and audio. For lights and aircon, the status is either 0 for off or 1 for on. A light status of 100 means blinking. The level parameter for lights and aircon is brightness and temperature respectively. The audio device is control by the command set play ¡playlist number¿. The playlists and their numbers are defined in the configuration file. A special device called mood can also be used to set the lights, temperature and playlist settings. The moods are also defined in the configuration file. The mood devices is not an actual hardware device.
⋆ show ¡device¿ - this command shows the current status and level of the device if applicable. There is a special show device called system which is not a hardware device. This device is used to display a summary of all devices available.
⋆ stop ¡device¿ - this is synonymous to the command set ¡device¿ ¡0¿ ¡0¿.
⋆ help - displays a help message

This simple protocol is only meant to be an presentation layer protocol. Applications should be written to enable users to be able to access the system via this protocol. Some client applications are included in this distribution in the clients folder. There is a java client and a web-based PHP client. These are just simple reference implementation clients only and were not meant to be used in the production environments. SACP being a simple protocol is not intended to be implemented as is, but is meant to be extended.

IV. SACP Clients

The prototype SCADA-TEMS server implementation comes with three SACP clients. These clients are reference implementation clients and are not meant for production environments.

A. Java SACP Client

This is the first SACP client to be written. A Java client was written to provide the developers with cross platform SACP client for systems testing. Java is a programming language developed at SUN Microsystems to create programs that compile once and run anywhere.

B. Web-based PHP SACP Client

Perl Hypertext Preprocessor (PHP) SACP Client was written to provide the system with a web-based interface. PHP was chosen because it is an embedded scripting language that is able to make socket connections which are necessary to interacting with the SCADA-TEMS server. The PHP client can also be modified to serve Wireless

Markup Language (WML) instead of Hypertext Markup Language (HTML) pages. WML pages can be viewed by Wireless Access Protocol (WAP) enabled mobile phones.

C. SMS-to-SACP Gateway

Being the SMS messaging capital of the world, it was imperative that the group provided a SMS-to-SACP gateway for controlling the SCADA-TEMS server. The SMS-to-SACP gateway utilizes the Gnokii libraries, yet another piece of open source software, to access a GSM mobile phone and retrieve its SMS messages. The SMS-to-SACP gateway was written in Perl.

SACP clients in other programming languages can also be implemented. Currently, SACP only contain a very limited amount of commands and it is not very difficult to implement a SACP client.

V. Conclusion

Software can be a powerful tool for building integrated hardware and software solutions. The key benefit of software is flexibility. The SCADA-TEMS server code was designed to be as flexible and extensible as possible. Great care was also taken to ensure that the hardware costs would be greatly reduced by utilizing software measure when applicable. The group used the "hardware only when necessary" development strategy. This is especially true when the necessary hardware components are not available or are too costly.

The SCADA-TEMS server serves as a tool for controlling Distributed SCADA systems. The SCADA-TEMS, in particular, is an example of one such system. The SCADA-TEMS server provided the necessary interface to allowing user interaction via a number of means that include but are not limited to: Java and web-based clients and a SMS-to-SACP gateway. The SCADA-TEMS server also uses a flexible dynamically loadable modules based architecture that allows the adding of support for new hardware devices simpler.