Networked Home Theater
Preliminary Proposal

By
Jeff Patrick

Advisor: Alexander Malinowski

Project Summary:

The Networked Home Theater is a hardware interface and software to connect any Sony CD player to a network. The hardware interface will consist of a microprocessor to control communications between the CD player and a host PC. The software will be two separate programs, a Java server and an Applet.

The server will communicate with the microprocessor and the Applet to moderate communications. It will also connect to CDDB, an Internet CD information database, to retrieve information about the current CD.

The applet is the front end of the project, and will be what the end user uses. The applet will display status pertaining to the current CD, information retrieved from CDDB, and controls for the CD player.

Overview:

Most Sony home theater components have a built in communications protocol. The various components connect to each other through a S-Link port. S-Link is a two-wire serial data bus that provides two way communications. Sony's intent for this bus was to allow their components to interact and allow an easier and more fun home theater experience.

The Networked Home Theater project connects the S-Link bus to a PC (Personal Computer). The PC has the opportunities for advanced automation of the components. The PC will be able to control the CD player as well as read information from the player. A Java server on the PC will allow Java applets on the Internet to connect and remotely control the player. The server will also download the CD information from CDDB.com.

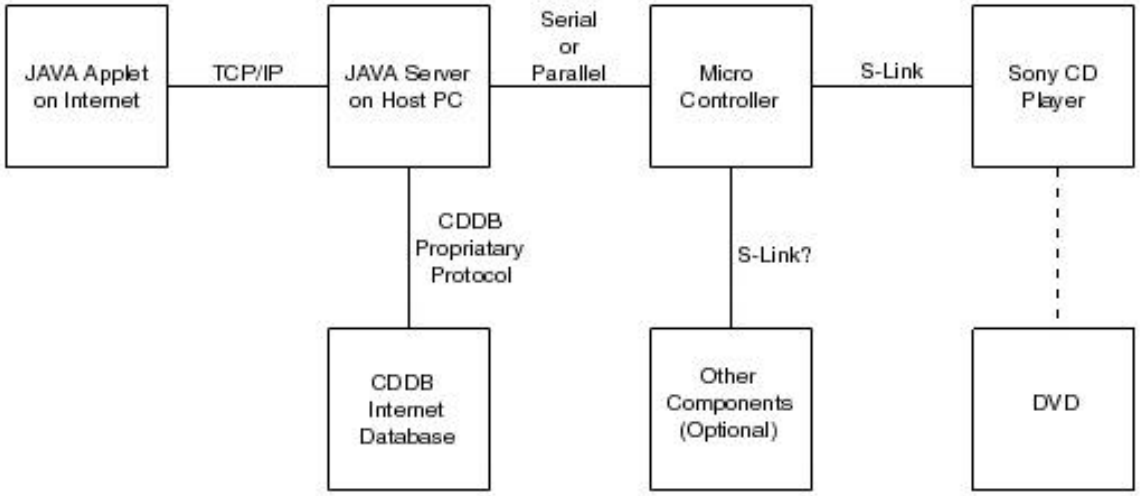

Figure 1. The block diagram of the networked Sony system connected to a personal computer connected to the home Intranet or to the Internet.

Figure 1 shows the block diagram of the entire system. The Sony CD player will connect directly to a microcontroller through the S-Link connection. The microcontroller will control communications directly to the CD player.

The microcontroller will then be connected to a PC running a Java server. The connection will be made through either a serial or parallel port. The microcontroller and PC will communicate to each other through a proprietary protocol that is not yet developed.

The Java server will manage several tasks. First, it will connect to the CDDB online database and retrieve the information about the currently playing CD (title and track contents.) Access to the CDDB database is available through a proprietary library available in C language for Windows platform. The Java language allows C functions to be called within the Java code. The server will also communicate with Java applets that will connect either locally or over the Internet. The applets will be the front end of the Networked Home Theater system. The end user will be able to send the server commands, which in turn will send the commands to the microcontroller, and ultimately, the CD player.

The Other Components Block could be other Sony products, such as DVD players or receivers. It could also be another brand of control signal if it exists.

Modes of Operation:

Idle

The system is in an idle state while it is waiting for a client to connect to the server and control the unit (Remote Control Mode) and listens to activity on the S-Link bus. In the case of playing a new CD, the mode switches to Query mode.

Remote Control

The system is being controlled by a remote user. The user sends input through the system, and the system sends back status information to the user. When a user plays a CD the system temporarily switches to the Query mode so that the CD information may be retrieved. The current command set is listed in Table 1 below:

Play:	Will play the selected Disc and track.
Stop:	Will stop all activities of the CD player.
Pause:	Will toggle pause mode while in play mode.
Next Track:	Switch to the next track on the disc.
Previous Track:	Switch to the previous track on the disc.
Power:	Will toggle the CD player power if digitally controlled.
Disc Check:	Will report on all of the discs in the player.
Eject:	Eject the CD tray.
Exchange:	Exchange other CD's while playing the current disc.
Next Disc:	Switch to the next disc.
Previous Disc:	Switch to the previous disc.
Disc<Disc #>	Switch to the first song of <Disc #>.
Disc<Disc #><Track #>	Switch to song <Track #> on disc <Disc #>
Track<Track #>	Switch to track <Track #> on the current disc.

Table 1. Current instructions to be made available to a remote user of the system.

Query Mode

The CD player reports the CD identification number to the PC. The PC then retrieves the CD information from the CDDB database. The information of each CD will be stored in a local database to allow selection of any CD in the player and to prevent the necessity of multiple inquiries about the same CD in the future. After obtaining the information the system returns to the mode previously active.

Inputs/Outputs:

CD Player

Port: S-Link 1	The CD player will send status information to the microcontroller. It will also receive instructions from the microcontroller on the same port.
Port: S-Link 2	May be used to connect multiple Sony components to the Networked Home Theater
Port: RCA Out	The CD player will send analog music signals out from the RCA ports. The music signals will be sent to an unspecified receiver to be processed.
Port: Digital out	Will most likely not be used.

MicroController

Port: RS232	This port will be connected to the PC to send status signals and receive instruction signals.
Port: S-Link	This port will be connected to the S-Link bus to control the CD player and receive status information.

PC

Port: Serial/Parallel	This port will be connected to the microcontroller to send and receive instructions or status information.
C API for CDDB:	The proprietary SDK from CDDB.com will be used to connect to the CDDB database and obtain information about the CD contents.
TCP/IP Socket	A TCP/IP socket will be used for to connect to the client applet and send/receive control information and CD information.

Applet

TCP/IP Socket	The TCP/IP port will be used to connect to the Java server and obtain information about the CD playing as well as any play-lists compiled. It will also be used to send the user input back to the Java server.
GUI	The GUI (Graphical User Interface) will be the front end of the project. It will display pertinent information about the CD playing as well as other CD's in the player.

Hardware Interface

In order to connect the CD player to the PC, several factors must be taken into consideration. The CD player outputs TTL (0-5) voltages, while the serial port on the PC outputs RS232 (-15 – 15) voltages. Some voltage conversion must be done in order to connect the two.

Second, the CD player communicates at a 600-microsecond time base. The Java server has a minimum wait time of 1 millisecond. In order to communicate at the speed required, the communications functions must be written in a separate language. In addition, the server PC will be slowed down considerably by the need to monitor a port at high speeds. This is a very large waste of processor power since the S-Link bus is unused 99% of the time.

To reduce strain on the PC processor, and reduce the complexity of the server software, a microprocessor will interface the PC to the CD player. The microprocessor has the ability to communicate to the CD player at a 600 microsecond base frequency. It can then talk to the PC at a slower speed. The microprocessor will also take care of the voltage conversions between TTL and RS232. Figure 1 shows the block diagram of the entire system with the microprocessor interface. In addition to controlling just Sony devices, the microprocessor will have the capability to talk to different bus's if the need arises.

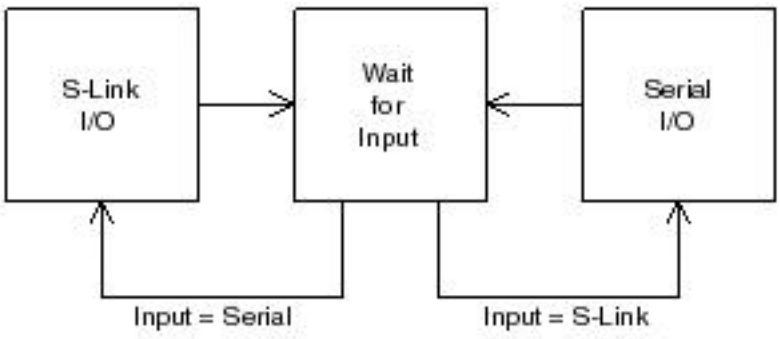

Figure 2. Microprocessor Code Flowchart

Microprocessor Code

The microprocessor code is very simple. Figure 2 shows the entire microprocessor code flowchart. The majority of the time the processor will be in an idle state. While in the idle state it will listen to both the Serial and the S-Link bus. When it detects information on either bus, it will convert it into the format of the other bus and send it. If the ability to interface multiple buses is implemented, the microprocessor will also analyze the incoming signal and route it accordingly.

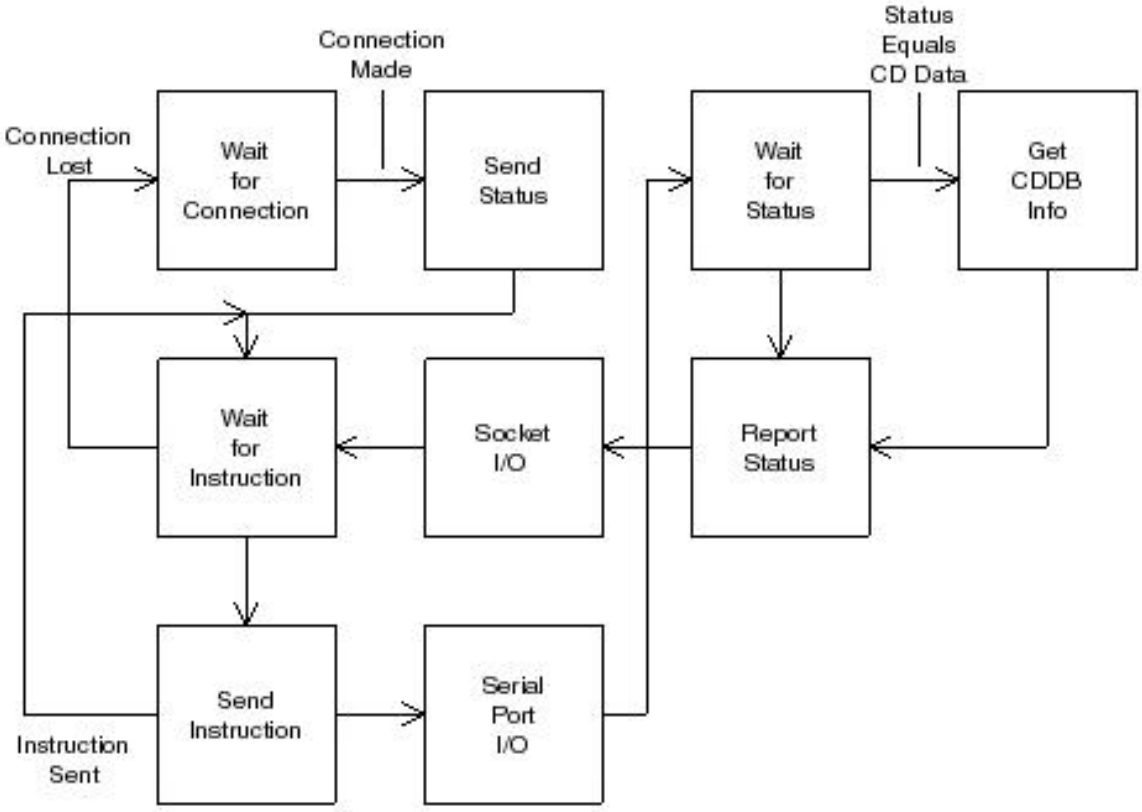

Figure 3. Server Code Flow Chart

Server Code

The Java server has several functions, which are shown in figure 3. The first operation of the server is to wait for a connection from an applet. Once a client is

connected, the server will transmit status information about the system. After the status information is sent the server will wait for an instruction from the client. When an instruction is received, the server will pass the command to the microprocessor through the Serial Port I/O. Finally, the server will return to the Wait for Instruction mode. The server will also be running a separate thread to detect if any information is being sent from the microprocessor. When status information is received, the server will analyze weather the information is a CD identification number. If so, the server will connect to CDDB using proprietary libraries from Gracenote/CDDB. The data retrieved from CDDB will be sent to the applet by using the Socket I/O.

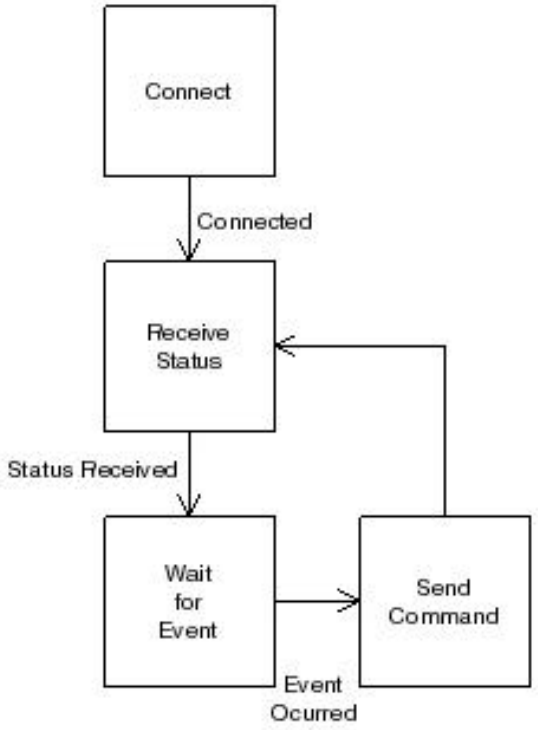

Figure 4. Applet Code Block Diagram

Applet Code

Figure 4 is the flowchart for the Applet. The applet will connect to the server and retrieve information about the discs in the player. If the client is the first connection to the server, it will be able to send commands to the system. Otherwise, it can just monitor the status of the system. After a command is sent to the system, the applet will receive an updated status before it waits for the next command.