Car Audio Interface

David McCallum

Dec 12, 2002
Car Audio Interface

Introductory Description

Toys. No matter what the age everyone loves a toy. When we are young our toys are simple: blocks, legos, or tinker toys. As we mature the price and complexity of our toys increase. One such toy is a car audio system. With its blinking lights, pulsating bass, and an infinite number of possible combinations of the two it is easy to see how millions of dollars are spent each year on car stereo equipment. The goal of this project is to develop and implement a Graphical User Interface for a car audio system. The scope of this project extends only to the Panasonic head unit, ignoring for now any peripheral audio components. The target user group for this product would be people with several after-market audio components and the desire to integrate them into a handheld device for easy component control. The user interface will allow users to access information about their audio system and allow them the ability to change the settings of individual components.

General Description

The Car Audio Interface or CAI will be implemented using a Palm m505 handheld. The device will display the available button options for the Panasonic CQDF800u head unit. Figure 1.1 depicts the major system hardware for the CAI.

Designed for use in the car, the Palm m505 will be self powered, running off of its internal battery powered supply. The interfacing circuitry will be powered using the car’s 12-
volt battery. The enclosure for the interface hardware will be no larger than 5 inches tall by 3 inches wide by 2 inches deep. The enclosure will be isolated from vibration and be waterproof for the final design. Found in Figure 1.1 the user interface will consist solely of the Palm graphical user interface. Functions of the buttons will include preprogrammed radio control, volume control, track selection, volume display, mode selection, random play, power, and repeat play. Time permitting two optional features will be added including current display, and a RF transceiver for wireless operation.

Figure 1.1 System Hardware
The car audio interfacing system is designed for the car audio enthusiast who wants something more user-friendly and customizable than the standard LCD screen presented on the Panasonic head unit. For this reason buttons on the Palm will be arranged to allow the most functionality while maintaining an easy to understand layout. The user interface will incorporate nearly all the buttons of the original deck face and add a few more to allow for easier access. Figure 1.2 shows a graphical depiction of the user interface to be used on the Palm. The buttons shown can be activated by simply tapping on them with a stylus. The CAI program sends a signal indirectly to the deck by way of a serial interface. This serial interface connects directly with the microprocessor. The microprocessor then decodes the signal and the appropriate analog switch will be activated. The activation of this switch will then produce the desired effect consistent with a button push by the user.
Figure 1.2 Sketch of User Interface
Functional Description of Hardware

The hardware diagram for the CAI is shown in Figure 1.3. The main parts of the CAI are the Palm m505, M68HC12B32 micro-controller, decoder, analog switches, and power supply. This project incorporates Panasonic hardware and hinges on the ability to interface with it. Therefore a brief description of connections and means of interfacing is required.

![Figure 1.3 Functional Hardware Diagram](image-url)
Palm m 505

Palm pilot was originally created as a Personal Digital Assistant (PDA) to help organize and schedule meetings, hold phone numbers and e-mail, and take notes. Palm devices incorporate several features that make them very popular for program creation and hardware interfacing. Some of the more important features for this project are asynchronous serial port, color display, memory expandability, Graphical User Interface (GUI) software, diverse, event driven, operating system, and small size. Many of the newer Palm devices, including the Palm m505, use the new Motorola Dragonball VZ microprocessor, which is capable of processing information at a rate equal to a 1980’s desktop computer. The Palm’s compact size, color display, and power make it the perfect platform for creating a graphic display based interface.

The Palm will be used to determine which functions the user wishes to be activated and sends the appropriate command to the Motorola M68HC12B32 micro-controller (MCU) via serial link. The Palm will also determine any hold down time for a key press and send a stop code when the key press is to be stopped.

Panasonic CQDF800u

The Panasonic CD player was chosen by convenience. It is a generic CD player/tuner that is currently available for experimentation. To interface with the user, the deck uses a set of push buttons located on the face. Figure 1.4 shows the faceplate of the CQDF800u including the layout of the buttons. The Liquid Crystal Display (LCD) driver micro-controller uses a polling routine to determine the button that has been selected by the user. Some buttons, including the preset radio frequency selector and CD track selector buttons, use the hold down time of a button press to determine the desired function. Button presses for the Panasonic head unit will be
simulated using analog switches. These switches will act exactly as a user pressing a key on the faceplate.

![Figure 1.4 Button Layout of the Panasonic CQDF800u](image)

**M68HC12B32**

The micro-controller executes the interfacing of the Palm m505 and the Panasonic CQDF800u head unit. The tasks involved in this process will be described in the User Interface section of this description. The optimum choice for the micro-controller was an Atmel AT91FR4081-33CI. The Atmel MCU was superior to the HC12 MCU due to several considerations. The four main selling points for the Atmel chip are superior temperature range, higher operation frequency, 85 I/O pins, and 1MB flash. These advantages over the Motorola chip lend superior expandability and durability to the design. However, after evaluating the difficulty in learning a new MCU instruction set and hardware configuration, the decision was made to use the Motorola chip. Other resources would also have to be supplied to support the Atmel chip as the electronics department at WWU does not support the resources.

After the Palm m505 determines the appropriate button push and time duration a coded signal will be sent via serial link to the HC12. The HC12 will be operating in single chip mode and will receive the data through the SCI port. The data from the SCI port is then processed by
the MCU and acted upon according to the user interface software. Visually depicted in figure 1.3 are the allocated resources of the HC12. PortP (PP0-3) will provide the data lines necessary for the 4 to 16 bit decoder. PortP (PP4) will be used for the chip enable on the analog switches. A chip enable is necessary due to delay times in the changes of PP0-3. Having a chip enable ensures that only valid button pushes are seen as legal to the analog switches and that no glitches occur on the switch outputs. The SCI on pins PS0-1 will provide the asynchronous serial interface with the Palm.

Memory allocations for the CAI will be provided by memory present on the HC12. The main program will be stored in the 32 k bytes of Flash EEPROM. The 1 k byte of RAM will be used to run the time slice routine and to store any variables needed in the program.

Power will be supplied to the interface hardware from the car battery. This connection will be made indirectly through the remote power on activated by the key to the car being turned on. The remote is capable of sourcing up to 4 amps at 12 volts. A DC-to-CD converter IC will regulate the 12 volt input voltage. This regulator will supply the 5 volts for the interface hardware. Maximum power dissipation for the CAI hardware is as follows.
**Software Description**

Two programming languages will be used to program the Palm and the HC12. The primary language for programming the PDA will be C. There will be six modules associated with the user interface on the PDA. These modules contain tasks, which are related to the module. The software modules include MAIN, START_UP, UTILS, MAIN_FORM, BUTTONS, and SERIAL. Descriptions of the PDA modules are as follows:

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAIN</td>
<td>Checks for proper start-up of the program and minimum PalmOS version. Calls PilotMain and contains the main event loop for the program. Also calls any start up routines and handles shut down of the program. The module tasks include:</td>
</tr>
<tr>
<td></td>
<td>• AppEventLoop Determines what type of event is happening and what task needs to service the event.</td>
</tr>
<tr>
<td></td>
<td>• PilotMain Checks for normal system startup. Also calls functions for startup and shut down of the program.</td>
</tr>
<tr>
<td>UTILS</td>
<td>Contains routines related to checking proper system start-up and minimum OS version.</td>
</tr>
<tr>
<td>MAIN_FORM</td>
<td>Handles events that happen within the main form. Contains algorithms for button pushes and calls routines in BUTTONS to handle each legal button press.</td>
</tr>
<tr>
<td>BUTTONS</td>
<td>Contains the routines required by the buttons to transmit the proper button press codes, timing, and stop code over the SCI.</td>
</tr>
</tbody>
</table>
SERIAL  Has the routines to initialize the SCI port and transmit data to the HC12.

The second programming language chosen to program the HC12 is assembly, or machine code. Assembly was chosen because the simplicity of the program does not require a more powerful programming language. There will be no need for external modules to the main program. The main program will contain only one algorithm designed to read the serial port and determine the appropriate sequence of outputs for the ports. Descriptions of the basic interface software is as follows:

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td>Contains any startup initialization and the time slice based event loop.</td>
</tr>
<tr>
<td>WaitSlice</td>
<td>Blocks further instructions from being executed until the SliceTimeFlag is set by the ISR.</td>
</tr>
<tr>
<td>GetPalmCode</td>
<td>Retrieves the input character from the serial port and places the input character into a global variable.</td>
</tr>
<tr>
<td>MakeMask</td>
<td>Determines the correct output port to make high and places a coded value into a global variable mask.</td>
</tr>
<tr>
<td>ActivatePort</td>
<td>Uses the mask made in the previous step to activate the appropriate port.</td>
</tr>
</tbody>
</table>
User Interface Description

The user interface will be completely supported by the Palm m505. The buttons on the display of the Palm will be volume, mute, mode, track up, track down, repeat, shuffle, and preprogrammed channel/track. Figure 1.2 delineates the layout of buttons on the screen of the Palm. Upon initialization of the program calibration will take place. This calibration will consist of the mute button being electronically pressed and the volume being raised to setting 15 on the LCD. After calibration the display shown in Figure 1.2 appears. Each of the graphic buttons represents a function that can be executed. The functions and how they work are as follows.

The sequential volume buttons will allow for the increase or decrease of the present volume level. A single tap of the stylus on the up or down button will result in a corresponding increase or decrease in the volume level. It will only be possible to increase or decrease the volume by one decibel per button press.

![Volume Buttons](image)

Figure 1.5 Sequential Volume Buttons.

The sequential track selection buttons serve a two-fold task within the CAI. A visual example of the track selection buttons can be seen in Figure 1.7. The primary function of the track up and down buttons are to scroll through the available tracks of a CD. The button can, however, also be used to scan through the radio frequency spectrum for available channels. This will not be the primary use of the buttons because one must use the original display from the
head unit to read the radio frequency. For the purpose of radio listening, preprogrammed buttons will be discussed later.

![Track selection buttons](image)

**Figure 1.6 Sequential Track Selection Buttons**

Preprogrammed radio frequency/track selection buttons are located along the left side of the graphical display. These buttons serve two purposes. The first use for the buttons is for track selection. If button 10 is pressed track 10 is played. This is not true when the head unit is in radio mode. For a press of buttons one through six the radio will be tuned to the corresponding frequency which is preprogrammed on the head unit. For a button press of buttons seven through 12 buttons one through six will be programmed with the current radio station. For example, if button seven is pressed button one will be reprogrammed with the current radio stations’ frequency.

The mode button selects the input to be listened to. There are three input options to be selected from: CD, radio, and auxiliary input. The auxiliary input is not used in this program. When selecting the source to listen to, all three options will have to be cycled through to select the appropriate input.

The repeat and shuffle buttons will only work when playing a CD. The repeat button selects the option to play a single song repeatedly. This option is deselected by pressing the button a second time or selecting another song using the sequential track selection buttons or the preprogrammed track selection buttons. The same deselecting process is used for the shuffle
option. The shuffle option randomly chooses songs from the CD until another track is specifically selected or the shuffle button is activated a second time.

The power button, when activated, will turn off the Panasonic head unit but will leave the interface hardware active. This will allow the user to turn the head unit back on using the CAI. Pressing the home key on the Palm m505 will deactivate the CAI program. This action will take the Palm back to the main program selection directory.

**Development Plan**

Project development thus far has contained mostly research and planning. Programming software has been purchased for the Palm. A Palm m505 has also been purchased. Preliminary programs have already been created to familiarize myself with the difficulties involved in creating a Palm program. Multiple successful test programs have been created and tested using the Palm emulation software. I would like to have hardware in hand by the end of fall quarter. By the end of Christmas break a prototype software program for the Palm device should be created. I would also like to have the leads for the analog switches soldered to the faceplate of the head unit and attached to a connector by the end of break. The success of the project revolves around three major aspects: hardware interfacing between the HC12 and the head unit, the HC12 interface software, and a Palm based graphical user interface. The GUI software and interface hardware are going to be the key points of this project. I plan on constructing and testing the hardware early on to allow time to overcome any technical difficulties.

I plan on taking the 454 embedded systems class, which utilizes an evaluation board for all of the projects. My plan is to create the software for the HC12 during winter quarter to take full advantage of the EVB. Much of the code for the stand alone time slice program has already
been written in a previous embedded systems class. This will allow for reduced programming time.

Fall Quarter 2002

Week 10: Order parts for prototype construction.

Week 11: Finals week. Study for finals. No work will be done on project.

Xmas Break 2002

Week 1: Begin GUI programming. Focus on basic visual depiction of GUI.

Week 2: Write serial port subroutines. Solder leads to faceplate and attach micro connector.

Week 3: Write delay subroutines. Begin fabricating prototype program.

Week 4: Finish prototype program and test.

Winter Quarter 2003

Week 1: Begin construction of prototype interface hardware.

Week 2: Open week in case of delays.

Week 3: Test interface hardware.

Week 4: Design and create PCB for interface hardware.

Week 5: Open week in case of delays.

Week 6: Begin writing interface software.

Week 7: Finish writing interface software. Test HC12 software with interface hardware.

Week 8: Test Palm GUI with EVB HC12 and interface hardware connected to Panasonic head unit.
Week 9: Open week in case of delays.
Week 10: Open week in case of delays.
Week 11: Open week in case of delays.

Spring Quarter 2003

Week 1: Design PCB for HC12.
Week 2: Send Eagle PCB design in to PCB express or other PCB manufacturer.
Week 3: Finalize the graphics of GUI. Put finishing touches on program.
Week 4: Open week in case of delays.
Week 5: Project Design Review.
Week 6: Construct PCB for HC12.
Week 7: Testing.
Week 8: Testing.
Week 9: Code review.
Week 10: Project Demonstration.
Development Hardware

Development of the CAI project will take place at my home and in Western Washington University’s ETEC lab, EET 340. The two locations contain all the tools required to construct the hardware for the CAI. The tools used most frequently from the university lab will include digital oscilloscope, PC, programmable power supply, and multimeter. System development hardware will include an evaluation board for the HC12 and the corresponding software. Software packages will include Metrowerks Codewarrior for the PalmOS, PalmOS Emulator, Constructor, MiniIDE, and D-Bug12. Additional software may be used as needed.

Demonstration Prototype

The demonstration CAI will be a stand-alone project. A generic project enclosure will be purchased to encase the HC12 and the interface hardware. The interface portion of the project will connect to the real world using a DB-9 connector for the serial port and a micro connector to connect to the Panasonic head unit.

Demonstration of a project that is designed to operate in a car presents certain logistical challenges when attempting a demonstration in an electronics lab. For the purpose of demonstration the head unit will be removed from the car. A pair of generic speakers will be connected and the programmable power supply in the lab will power the head unit at 12 volts. The remote line, which normally connects to a relay activated by the key being turned on in a vehicle, will be connected to 12 volts also. This will simulate the car being on. Users will then be invited to “tinker” with the CAI for hands on functional demonstration. Explanations of the CAI’s application will be made upon request. Informational materials about the Palm operating system and Panasonic head unit will be available for viewing.
**Electrical Specifications**

**Palm Requirements**

- Serial port data rate 9600 k-bits/sec
- Serial port universal Palm connector that is RS-232 compatible
- Palm OS 4.0 or greater
- 1Mbyte non-volatile memory for OS and program

**Interface Unit**

- Serial port data rate 9600 k-bits/sec
- Serial port female DB9 that is RS-232 compatible
- Time slice running at 1 ms per slice
- Input supply 12V +- 2.4V, max current 50 mA
- Panasonic CQDF800u

**Special Environmental Requirements**

- Shock Proof Housing (Final Product) for HC12
- Normal Operating Temperature 0° to 85° C
## Preliminary Parts List

<table>
<thead>
<tr>
<th>Description</th>
<th>Part #</th>
<th>Sources</th>
<th>Price ($)</th>
<th>Current (max)</th>
<th>Lead Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>16MHz Crystal</td>
<td>73-XT49U1600-20</td>
<td>Mouser</td>
<td>0.41</td>
<td></td>
<td>1 week</td>
</tr>
<tr>
<td>DC/DC converter</td>
<td>MAX710</td>
<td>Dallas Semiconductor</td>
<td>2.95</td>
<td>.2mA</td>
<td>3 week</td>
</tr>
<tr>
<td>MCU</td>
<td>MC68HC912B32</td>
<td>Motorola</td>
<td>11.31</td>
<td>45 mA</td>
<td>3 weeks</td>
</tr>
<tr>
<td>Reset IC</td>
<td>MAX6314</td>
<td>Dallas Semiconductor</td>
<td>0.99</td>
<td>12uA</td>
<td>3 weeks</td>
</tr>
<tr>
<td>PCB</td>
<td>Custom</td>
<td>PCB express</td>
<td>4.48</td>
<td></td>
<td>2 weeks</td>
</tr>
<tr>
<td>Analog Switch</td>
<td>MM74HC4316N</td>
<td>Mouser</td>
<td>0.42</td>
<td>160uA</td>
<td>1 week</td>
</tr>
<tr>
<td>Case</td>
<td>270-1801</td>
<td>Radio Shack</td>
<td>1.99</td>
<td></td>
<td>2 weeks</td>
</tr>
<tr>
<td>DB-9 connector</td>
<td>152-3309</td>
<td>Mouser</td>
<td>0.70</td>
<td></td>
<td>1 week</td>
</tr>
<tr>
<td>Capacitors</td>
<td>Various</td>
<td>Mouser</td>
<td>.12</td>
<td></td>
<td>1 week</td>
</tr>
<tr>
<td>Resistors</td>
<td>1%</td>
<td>Allied</td>
<td>.08</td>
<td></td>
<td>Parts Room</td>
</tr>
<tr>
<td>16 pin Palm universal connector to DB-9 converter</td>
<td>9MPUC</td>
<td>VIA</td>
<td>19.95</td>
<td></td>
<td>3 weeks</td>
</tr>
</tbody>
</table>

**TOTAL** 47.20 45.7 mA