Design Report: Clarion Bluetooth Car Kit

30 Sep 2002

Document Name F1T005

Version 02

Distribution:

Executive Summary

The final prototypes for the Clarion Integrated Bluetooth Headset have been built and tested. This document reviews the current design status following Bluetooth Qualification and CE testing.

Revision History

Version	Date	Description	Author	Approved by
01	18/04/2002	First Issue		
02	27/09/2002	Second issue following approvals		

Version 02	Approved by	MTJH

Contents

1 Circuit Description

The schematics for the Bluetooth module [3] can be found sheet by sheet in files F1P1S04_sheet1.PDF to F1P1S04_sheet6.PDF. The following sub-sections describe each sheet of the schematic.

1.1 Top Level, Sheet 1

This top level sheet shows the main hierarchical blocks of the integrated Bluetooth module. The design consists of four main blocks:

- BlueCore 2 Module (Sheets 2 and 3)
- Echo Cancellation (Sheet 4)
- Codec (Sheet 5)
- PSU (Sheet 6)

1.2 BlueCore 2 Module, Sheet 2

This sheet shows little more than the connections to the main Bluetooth circuitry and the printed antenna.

The symbol J5 represents the hierarchical block of Sheet 3, Baseband and RF, which is itself based closely on CSR's BC02 reference module and layout. J5 is used as place holder for that module including its footprint, test points, and screening can.

J2 is a footprint for a SMA connector that may be fitted to the board in order to perform conducted RF tests during development and Bluetooth Qualification. If J2 is used C30 should be removed to isolate the printed antenna and C46 fitted instead.

U7, Q5, R57 and C12 form a power on reset circuit. The BC02 would normally start executing code when its core voltage reaches 1.8V. However at this time the supply to the flash memory may not have reached its operating voltage causing erroneous operation. The power on reset circuit holds the BC02 in reset until the flash supply has reached its operating voltage. This threshold, depending on the tolerance of U7 will be between 2.70 and 2.97 Volts. U7 will re-assert reset should the supply fall below a corresponding threshold in the range 2.65 to 2.75 volts.

1.3 Baseband and RF, Sheet 3

This circuitry is based on CSR’s BC02 reference design and layout. It sits entirely within the footprint of J5 on Sheet 2.

The main difference between this sheet and CSR’s reference is that additional I/O lines PIO8 – PIO11 have been connected. Also on the layout the gap between the flash memory, U2 and the BC02, U1, has been increased to ease the fitting of the BGA devices during production.

U4 is a low noise regulator that is used to generate a 1.8V rail for used by the BC02’s analogue, radio and core circuits.

The rail, VDD_MEM can be powered either from the 1.8V of 3.0V BB (default) rails depending on the voltage supply range of the flash memory device fitted.

1.4 Echo Cancellation, Sheet 4

Echo cancellation for the hands-free system is provided by the Mitel MT93L16 IC, U10. This is inserted into the PCM data signals from the BC02 to the Codec.

The echo canceller requires a 20MHz clock signal to operate. This clock is based on a 20MHz ceramic resonator, CR1, and a dual 2-input NAND gate IC, U3. Alternatively a Micro-oscillator part, OSC1 can be used instead, but this is slightly more expensive and so the schematic shows this part of the circuit marked DO NOT FIT.

Signals EC_SCLK, EC_CSn and EC_DO are used to control the echo canceller. They are directly driven by software via the I/O lines of the BC02. These signals are primarily used to reset the echo canceller and change the received volume in response to Volume Up and Down key presses.

Should echo cancellation not be required, all components on this sheet can be removed and 0Ω links fitted for R91 and R94. However no volume control would be possible on the system.

1.5 Codec, Sheet 5

This sheet shows the CODEC, U17 and the analogue circuitry. Either a linear MC145483 or a companding MC145481 device may be fitted. However the BC02 is not compatible with both the echo canceller and the linear Codec. If the linear device is used the echo canceller must be removed. The companded Codec can only be used in its A-Law mode as pin 16 is hardwired to ground.

PO_p and PO_n form the differential audio output. The unit can be configured for single ended output by fitting R52 0Ω and removing shorting link R34. The gain of the audio output is set by the resistors R19 and R20. However as the CODEC only has a single 3.0V supply rail only a limited increase in gain can be made without causing the signal to clip. Gain increases are best made in the Clarion head unit.

MIC1 is the on board mounted microphone. The final choice for this part is a noise cancelling type from Bujeon. An external microphone can be plugged in J8 and used instead of the internal one. Microphone gain is controlled by resistors R17, R18, R21 and R22, please refer to the CODEC data sheet.

Side-tone can be added to the system by fitting R38 (10kΩ). Side-tone cannot be used if the echo canceller is in circuit.

The signal PDC is used to disable the audio circuitry and the 20MHz ceramic oscillator on sheet 4. This signal is not used at present and is currently held high in software. It is intended for use on a future system where power saving is required.

R48 (100Ω) is placed in series with the power supply to the codec to prevent the codec entering a high current state during ESD discharges.

1.6 PSU and I/O, Sheet 6

The module is supplied with 5V from connector J1. Two 3.0 Volt rails are regulated from the incoming 5V supply. One is used to power the BlueCore circuits the other for the echo canceller and Codec. Two regulators, U5 and U8, are used to try to isolate digital noise that may be generated by the echo canceller circuitry from the Bluetooth radio. In addition one regulator in this package could not supply both rails under all load and temperature conditions. U5 and U8 also provide over-current and thermal shutdown protection.

Component tolerances mean that the outputs of the two regulators will be in the range 3.0 to 3.3 Volts.

Q1 drives the open collector nIN_CALL signal to switch the Clarion head unit into phone mode. Q2, R13 and R12 provide protection to Q1 should nIN_CALL be shorted to 5V. This circuit has been tested and shows that the current limits at 40mA when nIN_CALL is shorted to 6V. Under normal operation the Low output voltage was measured at 0.3 V when sinking 16mA which meets the design constraint of being <0.4V at 10mA.

LED's D1 and D2 are driven by transistors Q3 and Q4 respectively. These are configured to give a constant current drive to the LED's independent of the incoming supply rail. Depending largely on the actual value of the 3.0V BB rail the current through the LED's will be in the range of 15 to 20mA.

The constant current LED drives lessens the loading on the regulators U5 and U8 and gives a degree of overvoltage protection to the circuitry. At ambient temperature conditions the system should survive incorrect connection to a 12V supply. However prolonged operation at voltages above the design limit of 5 – 6V will cause the regulators U5 and U8 to overheat, triggering their thermal shutdown. Overvoltage protection is not a design requirement and has not been tested.

Connector J3 allows the LED's, buttons and microphone to be remote mounted. If the LED's are remote mounted D1 and D2 should be removed from the board.

If the microphone is remote mounted only a single ended connection is possible. In this case R29 on sheet 5 should be replaced with 0Ω. A 0Ω link should also be fitted for either R41 or R42 depending on whether the external microphone socket, J8 is still fitted.

2 Testing

All Bluetooth Qualification and CE tests have now been completed. For detailed performance of a typical unit, please refer to the test results.

2.1 Radio

Radio performance is well within the Bluetooth specifications, achieving the transmit power between 0dBm and +4dBm and sensitivity below –80dBm. The low cost printed antenna allows Bluetooth operation over at least 10m in an open environment and adequate performance in a vehicle.

2.2 Audio

Audio tests have been undertaken to confirm the basic operation and to set transmitted and received loudness.

2.2.1 Received Audio

The gain of the received audio signal which drives the Head Unit has been set so that at maximum volume, the codec output is not clipping and the input to the Head Unit amplifier is not being overdriven. It is suggested that if required, Clarion optimise the gain of the Head Unit to give a similar loudness at the speaker whether the audio source is telephone or music.

It is essential that the received audio is not subjected subsequent to distortion or clipping by the Head Unit or speakers. Should such distortion occur the echo canceller IC U10 will not be able to correctly eliminate acoustic echo.

2.2.2 Transmitted Audio

The microphone gain has been set-up to operate with an external uni-directional microphone and in this configuration the transmitted audio is of good quality. Adequate performance can be obtained with the internal microphone if the unit is placed near the user's mouth and the microphone gain is increased by adjusting resistors R17, R18, R21 and R22. It is not possible to obtain a more sensitive but similar internal microphone from Ole Wolff.

2.2.3 Echo Cancellation

Following field trials by Clarion, a short echo was noticed when the Bluetooth user began to speak. Following investigation and assistance from Zarlink, this has been greatly improved by adjusting two of the default registers in the echo canceller chip. These settings are now standard and part of the CSR firmware.

2.3 Bed of Nails Tester

A bed of nails tester was procured and wired-up at Plextek using level shifter circuitry originally included as a snap-off section on the first prototype of the PCB. As well as easy access to all the test points, the tester is required for access to the SPI bus which is used to program the flash memory. The schematic for the test jig is in file F1P2S01.PDF.

2.4 Caseworks & Connectors

The caseworks have been designed by Clarion to fit the PCB. Unfortunately, advice given in April 2002 to improve the performance of the caseworks for immunity to ESD was not heeded. Therefore, an additional iteration of the PCB was required from 03 to 04 to add components to compensate for the failings of the caseworks. The performance was greatly improved such that the ESD tests for CE marking were passed with serial number F1050. However, in order for this to take place, the rear of the cable connector required sealing to prevent ESD discharges directly to the connector. It is assumed that for production, a sealed connector at the end of the lead will be used.

In addition the design of the caseworks around the microphone is not as recommended by Plextek and the cavity may be prone to resonance at certain audio frequencies.

The additional connectors for external UART control and remote buttons and indicators are not fitted to the PCB because the existing caseworks do not have apertures for them. For future versions of the product such as a firmware upgrade to Handsfree Profile, the additional connectors will need to be fitted. In such circumstances, it will be paramount that the mechanical design is optimised to prevent ESD discharges from reaching the CSR chip if the existing PCB design is to be used.

3 Software

The software used in the CM HS1 is based on CSR's Bluelab SDK Headset application. It maintains the same overall functions but the MMI is greatly modified.

3.1 Source Files

The main application consists of the following modules:

checkbuttons.c	Responsible for scanning for and debouncing key presses. This module replicates code that is now performed in a CSR library libbutton.a. However this CSR library does not support simultaneous pressing of multiple buttons and so cannot be used directly for this application. In addition libbutton.a has a habit of evolving with each release of Bluelab with the new version incompatible with applications written with the old.
hal.c	A handful of low level routines that map functions onto the actual hardware. Most notable of which is *HALsetVolume*() which has compiler flags that allow volume setting commands to be mapped between the echo canceller or the codec.
volume.c	Handles higher level volume commands. Largely unmodified from the CSR original.
main.c	Contains *main*(). This is responsible for configuring the system and handling messages from the headset framework library.
open.c	Supplies the *openReq*() and *openCfm*() message handlers. *openReq*() is called from *main* and is used to configure the and headset framework and rfcomm layers. Once configured *openCfm*() is called with the results indicating whether the device is currently paired.
reset.c	The reset function is used to initialise key parameters in the persistent store. In the CSR original code *resetReq*() was called in response to a particular key press combination. In this version it is only called directly from a UART command. The function *checkReset*() however is a Plextek addition and is called from *main* every time the system starts. It compares some key persistent store values and forces a call to *resetReq*() if they are absent.
pair.c	Implements the *pairButton*() handler for the *virtual* pair button. There is no longer a real pair button in this application. Instead it is a specific key combination as detected by checkButtons.c (or a UART command).
cmd.c	Single function module as written by CSR. The function in this module is called if ever an unknown AT command is received from the Audio Gateway.
connect.c	Implements the *idleButton*() and *talkButton*() handlers. The idle button is a *virtual* button and is a Plextek addition to the application. It puts the system into an idle state as if powered down. This module contains the main MMI state machine and controls the connection process between the unit and the audio gateway.

ring.c	This module handles the RING command received from the audio gateway. It is responsible for enabling the IN CALL signal and inserting the ring tone into the audio path. The original CSR implementation of ring tone generation only works with a linear codec. As we are using an A-Law codec a table of A-law values had to be obtained and embedded in the code as if they were 16-bit sampled values, (see files e7alaw.h, f7alaw.h and c7alaw.h). Generating these files is a tedious process. Firstly a source of 16 bit linear data with the correct 8kHz data rate is required. This can then be passed through a conversion application to produce the equivalent A-Law values and finally these values need to be converted to hexadecimal text for inclusion in the header files.
sco.c	Handles the completion and deletion of the SCO audio link.
battery.c	Stub code that handles reading from some battery monitoring system. Not used in this application.
HostIf.c	Plextek addition to the Headset application. This module defines and implements a simply UART protocol running over BCSP channel 13. Refer to document F1T00301.doc.
VMSpyString.c	The CSR compiler system is inherently 16 bits. chars and bytes are mapped to 16 bit quantities. So the 'c' string constant "ABCD" is encoded as 0x0041 0x0042 0x0043 0x0044 0x0000. However when using VMSpy to send strings "ABCD" is packaged as 0x4241 0x4443. This module contains some string functions to handle these differences.
EchoCanceller.c	This module implements the interface to the echo canceller IC. The echo canceller is controlled using a synchronous serial bus. This bus is simulated in the module by direct manipulation of port pins.
Leds.c	This module makes use of the CSR sequencing utilities to drive the LEDs and the IN CALL signal.

3.2 Libraries

The source files listed in the previous section represent the application layer of the CM HS1. They make use of various utilities and protocol layers that are included in CSR supplied Bluelab libraries. These in turn are then linked with CSR's RFCOMM protocol stack version 15 for the BC02. In implementing the application it was necessary to gain access to some hardware features that were not directly supported by these CSR libraries. Consequently, libraries libcm_rfcomm.a and libframework.a were replaced with modified versions libplxcm_rfcomm.lib and libplxframework.lib. The following sections describe the changes made to these libraries, for more information on the libraries themselves please refer to the CSR documentation.

3.2.1 Headset framework library (libplxframework.a)

framework_pair.c

HFSpairCfm()	Return values of PsStore checked so that a debug message can be printed on error. This code was amended whilst tracking down a persistent store problem that was actually due to power supply and reset issues on the BC02.

framework_reset.c

HFSresetCM()	This function sets the default value for the PIN number and was modified to make the default 0000.

framework_main.c	Changes in this file occur in two places both of which are delimited with Plextek comments.
DECALRE_TASK1()	This contains the main message handling switch statement for the library. Additional case statements were added to support new commands for reading and writing voice settings[1]. Since making these changes CSR have added a persistent store key PSKEY_PCM_FORMAT that so these are no longer needed.
HFSVoiceSettingCfm() HFSVoiceSettingWrCfm() HFSReadVoiceSetting() HFSWriteVoiceSetting()	Additional helper functions added by Plextek.
framework_private.h	Prototypes for new functions declared here in a section delimited by Plextek comments.
plxframework.h	based on framework.h new structures and constants are defined here to support the new connection handler messages.

3.2.2 RFCOMM Connection manager library (libplxcm_rfcomm.a)

dm_handler.c	Changes in this file occur in two places both of which are delimited with Plextek comments.
dmhandler()	Handles messages returned from the Bluestack virtual machine. This switch statement has been extended to include messages for the reading and writing of voice settings.
dmReadVoiceSettings() dmWriteVoiceSettings()	New helper functions to support receiving replies to the HCI voice settings commands.
dmWriteLocalName()	New handler that allows the user friendly name of the Bluetooth device to be set from the main application.
cm,_main.c	Changes in this file occur in two places both of which are delimited with Plextek comments.
DECALRE_TASK0()	This contains the main message handling switch statement for the library. Additional case statements were added to support new commands for reading and writing voice settings and to set new user friendly name.
cmVoiceSettingRead() cmVoiceSettingsWriteCfm()	Additional helper functions added by Plextek.
cm_plxrfcomm.h	based on cm_rfcomm.h new structures and constants defined here to support the new connection handler messages.

[1] Voice settings are HCI commands that control the PCM audio format. (Linear, μ-Law, A-Law etc).

3.3 Building the Application

The application is built from within a "Cygwin" shell on a Windows 2000 or NT machine. Please refer to the CSR documentation for further details.

Building is controlled by two files:

makefile. This file lists the object modules and the libraries required to build the application. It also sets the output filename.

makefile.inc Included into makefile this is a CSR supplied file that contains the non project specific settings necessary to compile and build the project. It has been modified slightly to allow the correct version of the RFCOMM protocol stack to be used in building the final executables.

When building the application ensure that the replacement Plextek libraries are up to date. libplxcm_rfcomm should be build before libplx_framework. Go to the appropriate directories within the cygwin shell and type:

make install

This will build the library modules and copy the libraries and header files into the appropriate directories for use with the main application.

To build the application go to the appropriate directory within the cygwin shell and type

make bc02rfc15

This will build the application and will link it with the correct version of the BC02 RFCOMM stack.

3.4 Persistent Store

The CM HS1 application make use of three additional user location in the persistent store over those already used by the CSR headset application.

PSKEY_USR16 This contains the 11 character user friendly name string (one character per 16-bit location). Default "CM HS1". If the name required is less than 11 characters then the remaining locations need to be padded with zeros.

PSKEY_USR17 16 bit flags register.

If the top bit (0x8000) is set then debug "printf" statements in the program will become active. In this case the application will only run if connected to VmSpy via the UART interface. VmSpy if need to display the printf text and if it is not connected the application will stall.

If the bottom bit (0x0001) is set the operation of the idle and pair key press combinations are swapped.

PSKEY_USR17 bit-0	5 sec Press on CALL	Simultaneous press of Vol+ and Vol-
0	idle	Pair
1	pair	idle

PSKEY_USR18 Echo canceller initialisation string. This contains a sequence of commands that are sent to the echo canceller every time an audio channel is established. The string consists of up to 8 16-bit command values. If less than 8 commands are required then the remaining locations must be padded with zeroes. Each 16 bit command value contains an 8-bit echo canceller address in the MSB and an 8-bit data value in the LSB. The default value for this string is:

0x0101 0x3210 0x1210 0x0000 0x0000 0x0000 0x0000 0x0000.

Please refer to the echo canceller data sheet for more information.

4 Conclusion

The Clarion Bluetooth Car Kit development is now complete and all relevant Bluetooth and CE tests have been completed and passed. In general, the development has gone well, although for future projects, it is suggested that the effect of ESD on the Caseworks and PCB are evaluated and taken into consideration by both Plextek and Clarion at an earlier stage in the design.

5 Abbreviations and Definitions

BC02	BlueCore 2 Bluetooth IC
BGA	Ball Grid Array
CSR	Cambridge Silicon Radio
ECR	Engineering Change Request
HS	Headset, used to denote the Clarion Bluetooth module
PCM	Pulse Code Modulation
PSU	Power Supply Unit

6 Associated Documents

[1] Functional Specification for Clarion Integrated Bluetooth Headset, F1T001 05

[2] Manual Test Specification, F1T004 05

[3] Module Schematics, F1P1S04 sheets 1 to 6

[4] Bed of Nails schematic, F1P2S01