



BlueCore2™-External

RS232 Cable Replacement

Example Design

June 2004



CSR

Cambridge Science Park
Milton Road
Cambridge
CB4 0WH
United Kingdom

Registered in England 3665875

Tel: +44 (0)1223 692000
Fax: +44 (0)1223 692001

www.csr.com

Contents

- 1 Introduction 4**
 - 1.1 Serial Port Profile..... 5
 - 1.2 On Chip Software 5
 - 1.3 A Solution for Both Ends of the Link 5
 - 1.4 Example Design Contents 6
- 2 Circuit Description 7**
 - 2.1 Power Supply 7
 - 2.1.1 Decoupling 7
 - 2.2 Flash Memory 8
 - 2.3 Crystal 8
 - 2.4 Impedance Matching and Balun 8
 - 2.5 Filter 9
 - 2.6 Antenna 9
 - 2.7 RF Test Point..... 9
 - 2.8 VCO 9
 - 2.8.1 RF Reference Design..... 9
 - 2.9 RS232 Level Converter 9
 - 2.10 Reset Circuit 10
 - 2.11 Switches 10
 - 2.12 PIO Lines..... 10
 - 2.13 LEDS 10
 - 2.14 Handshake Lines..... 10
 - 2.15 PCB 11
 - 2.15.1 Screening Can 11
 - 2.15.2 SPI Interface 11
- 3 Performance of the RS232 Cable Replacement Example Design 12**
 - 3.1 Achievable Data Rate 12
 - 3.2 Deep Sleep Implementation 12
- 4 Software Architecture 14**
 - 4.1 Block Diagram 14
- 5 Commissioning 15**
 - 5.1 Downloading Firmware 15
 - 5.2 Persistent Store Settings 17
 - 5.3 Steps for Pairing with Another Bluetooth RS232 Device 17
 - 5.4 More Information 18
- 6 Schematics 19**
- 7 Bill of Materials 21**
- 8 Document References 22**
- 9 Acronyms and Definitions..... 23**
- 10 Record of Changes 24**

List of Figures

| | |
|--|----|
| Figure 1.1: BlueCore Serial Port System Architecture | 4 |
| Figure 1.2: Configuration of Equipment for RS232 Link | 5 |
| Figure 2.1: Example RS232 Bluetooth Dongle | 7 |
| Figure 2.2: Equivalent Signal Connection of DTE to DTE | 10 |
| Figure 2.3: Equivalent Signal Connection of DTE to DCE | 11 |
| Figure 4.1: Serial Port Software Architecture | 14 |
| Figure 5.1: BlueFlash Dialog Box | 15 |
| Figure 5.2: Choose File to be Loaded | 16 |
| Figure 5.3: Program Flash Memory | 16 |
| Figure 6.1: BlueCore RS232 Cable Replacement Circuit Diagram (DCE) | 19 |
| Figure 6.2: BlueCore RS232 Cable Replacement Circuit (DTE) | 20 |

List of Tables

| | |
|--|----|
| Table 5.1: Persistent Store Settings | 17 |
| Table 5.2: Status LED Description | 17 |

1 Introduction

RS232 is a widely used standard for communications between industrial, medical and scientific apparatus. Despite growth in alternatives such as Ethernet and USB, RS232 remains the only communications standard which can offer the widespread interoperability and cost effectiveness required by manufacturers of this equipment. However the need for a physical, wired connection between devices can be inconvenient.

Many industrial, medical and scientific devices are used to collect data in the field. Results are then downloaded to a computer for analysis over a RS232 link. The user has to carry around an RS232 cable, or perhaps more frequently, search for one when they return to base.

Probes and measuring devices often have to be manipulated into difficult to access positions, a process made even more difficult when attached to a data cable. A wireless alternative would make these devices easier to use.

Bluetooth® offers a solution to these problems. Originally conceived as a cable replacement technology, Bluetooth is ideal and can replace a RS232 connection with a convenient wireless link, improving mobility and convenience.

Compared to other wireless technologies Bluetooth is low cost and consumes very little power. Several users can easily connect and disconnect to a wireless Bluetooth RS232 connection at distances of at least 20m – far longer than a RS232 cable. Built in security with an inclusion of strong (128-bit) encryption, means that data will always be protected from unauthorised eavesdropping.

CSR's **BlueCore™2-External** device is ideal for these applications. It combines radio frequency (RF) and baseband circuitry onto a single chip, reducing the external component count to a minimum. Radio output power and sensitivity exceed the Bluetooth specification by a wide margin, allowing inexpensive printed RF components to be used without significantly reducing range. The internal reduced instruction set computer (RISC) processor is user programmable and powerful enough to run RS232 port emulation software without an external host.

The BlueCore RS232 Cable Replacement Example Design⁽¹⁾ describes in detail how BlueCore2-External can be used to completely replace an RS232 cable, the system architecture is shown in Figure 1.1: BlueCore Serial Port System Architecture. No changes are needed to existing product architecture – the Bluetooth device behaves exactly like a standard RS232 cable.

The final details of the schematic and printed circuit board (PCB) layout are deliberately left to the customer to allow maximum flexibility in form factor. When designing the RF layout CSR strongly advises customers to refer to the BlueCore2-External Class 2 reference designs available separately from CSR. Further information is available in the Bluetooth Serial Cable Replacement Specification and the BlueCore2-External Data Book.

Note:

⁽¹⁾ The BlueCore RS232 Cable Replacement Example Design is intended to be a design that accompanies the purchase of a Casira™ development kit.

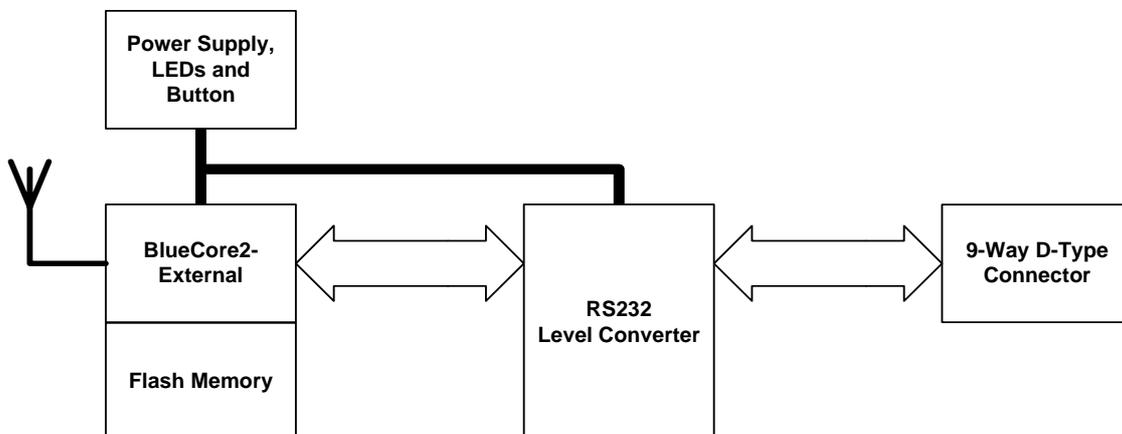


Figure 1.1: BlueCore Serial Port System Architecture

1.1 Serial Port Profile

CSR's RS232 Cable Replacement Software is an implementation of the Bluetooth Serial Port Profile (section K5 in the Bluetooth profiles specification v1.1). This defines the protocols and procedures that shall be used by devices using Bluetooth for RS232 (or similar) serial cable emulation.

1.2 On Chip Software

BlueCore2-External contains a 16-bit RISC microcontroller which is used to run the Bluetooth protocol stack. CSR has implemented a virtual machine (VM) on this RISC microcontroller, the VM is an interpreter that allows applications to be run without adversely affecting the underlying Bluetooth communications stack. The architecture is much the same as that used for Java™.

CSR's BlueLab™ Software Development Kit (SDK) contains tools for compiling and debugging VM applications. It also contains source code for CSR's Serial Port Application. Customers can take this code and tailor it to their individual requirements.

The RS232 cable replacement VM application is layered on pre-qualified RFCOMM firmware. The firmware does not need to be re-qualified when qualifying the VM application. This greatly reduces the time and cost of qualifying a new Bluetooth product.

1.3 A Solution for Both Ends of the Link

The example design includes schematics for both ends of the RS232 link. One end connects to the Data Terminating Equipment (DTE) and the other end to the Data Communication Equipment (DCE).

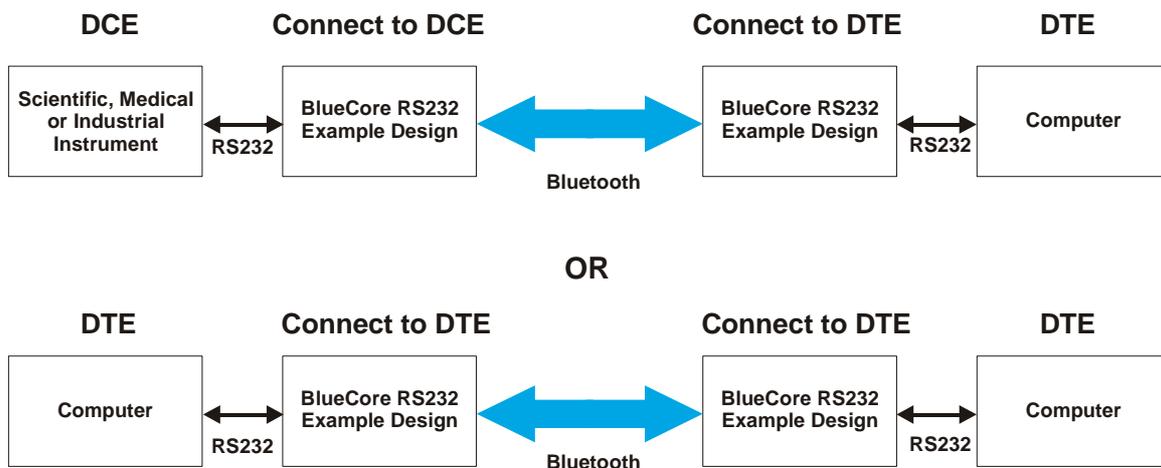


Figure 1.2: Configuration of Equipment for RS232 Link

Note:

Different on-chip software for each end of the link is available from CSR in the BlueLab Software Development Kit. Figure 1.2: Configuration of Equipment for RS232 Link shows how the Bluetooth link using the RS232 Cable Replacement Example Design can replace the traditional links between DTE to DTE e.g., PC to PC link and between DCE to DTE e.g., PC to another piece of equipment.

1.4 Example Design Contents

The example design contains all of the information required for an original equipment manufacturer (OEM) to add Bluetooth functionality to any electronic equipment which currently supports RS232. The following items are included in the package:

- Full product description (contained in this document)
- Circuit schematic (contained in this document)
- Bill of materials (contained in this document)
- RS232 Cable Replacement on chip software (available in the BlueLab Development Kit)

2 Circuit Description

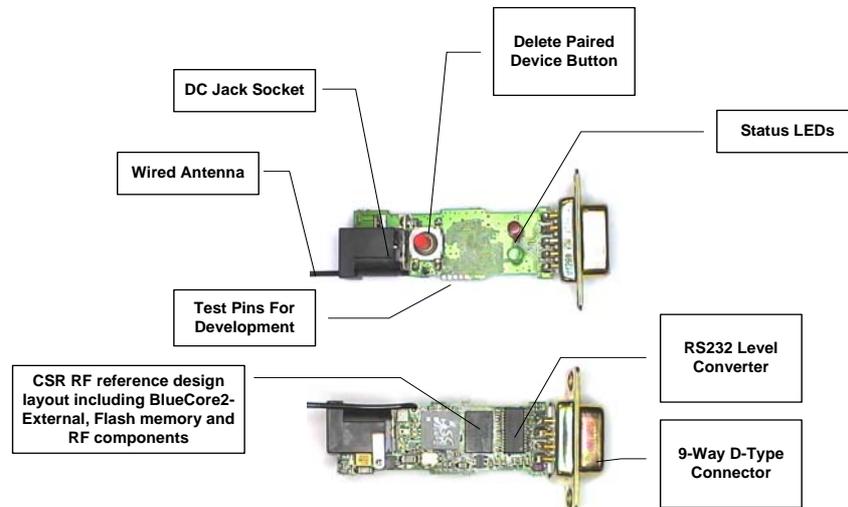


Figure 2.1: Example RS232 Bluetooth Dongle

Note:

Please refer to the schematics shown in Section 6 while reading the following sections.

Figure 2.1: Example RS232 Bluetooth Dongle shows how the example design can be used to make a Bluetooth dongle which plugs directly into a RS232 port to provide wireless functionality. The example design is equally suitable for applications where Bluetooth will be embedded directly onto an existing PCB, or on an internally mounted daughter board.

The circuit is based around U1, a BC212015DN BlueCore2-External from CSR. It contains a 2.4GHz radio and baseband providing voice and data communications in compliance with the Bluetooth Specification v1.1.

The firmware for U1 is contained in a 8Mbit Flash memory (U2), whilst RS232 functions are accomplished by a UART connection to a MAX3238 (U5).

2.1 Power Supply

The example design requires an external 5V - 12V D.C. power supply that is connected via a centre jack socket. The centre pin is positive. A XC6204B332MR (U6) is used to regulate the input voltage to 3.3V and XC6204B182MR (U4) is used to regulate the 3.3V to 1.8V.

2.1.1 Decoupling

The 1.8V supply should be routed on a separate, split power plane to the BlueCore2-External. Decoupling capacitors should be as close as is practical to the appropriate VDD pin. There are 3 types of decoupling used:

1. Bulk decouplers of 2.2 μ F, one per power rail. Their acceptable proximity to the destination of the power is roughly 10mm if tracked, or 50mm if on a power plane.
2. 10nF/47nF/100nF decouplers are placed close to the pins they are decoupling. These are:
 - C28 for VDD_ANA
 - C10 for VDD_PADS and VDD_PIO
 - C29 for VDD_MEM
 - C8 for VDD_CORE
3. RF decoupling of 47pF aimed to bypass 2.4GHz RF

The VDD_CORE pin has a resistor/capacitor (RC) filter on it, composed of R3 and C8. This prevents noise emanating from the digital circuits within U1 from coupling onto the 1.8V rail, where it could have an adverse effect on RF performance.

L5 and C9 filter 2.4GHz RF from the power entering the VDD_VCO pin. In some cases, L5 can be linked out and Bluetooth qualification still attained. The ground end of C9 should be kept clear of the ground near C1, C2, C3 and U3, else RF can be coupled onto VDD_VCO through C9 itself.

L1 and C1 decouple the power feed to the RF balun (T1). L1 can be increased to higher values (e.g., 15nH or 22nH) with some slight improvement in RF performance.

2.2 Flash Memory

U2 is a 8Mbit Flash memory used to store both the firmware and configuration parameters for the BlueCore2-External. The configuration parameters are stored in an area of the Flash memory that is called the Persistent Store. This can be manipulated using the PSTool Windows program available from CSR.

BlueCore2-External supports a number of Flash memory devices. As new Flash memory devices are released, support is added to the BlueCore2-External firmware and programming tools.

Before choosing a Flash memory device, please read document bcore-an-001Pb which can be obtained from the CSR secure web site.

The Flash memory operates from 3.3V. C13 is used as a decoupling capacitor and should be close to the Flash power pin and BlueCore2-External VDD_MEM pin. Ideally, route a small split power plane for VDD_MEM.

The circuit can be easily modified to use a 1.8V Flash memory chip. The following changes are required:

1. Remove the reset circuit (C16, R10 and R12). This is no longer necessary because the Flash will power up at the same time as BlueCore2-External, eliminating any race hazard.
2. Ground the RESET pin so that BlueCore2-External starts immediately on power up. The internal power on reset circuit within the BlueCore2-External will ensure correct power up operation.
3. Connect VDD_MEM to 1.8V.
4. Move C13 to the 1.8V rail (near the Flash memory).
5. Connect the Flash to the 1.8V rail.

2.3 Crystal

Crystal XT1 (16MHz) should have better than 20ppm long-term stability. The RF transmission and reception frequencies are derived from XT1. Ballast capacitors C11 and C12 determine the crystal oscillation amplitude, and can also be used to 'pull' the crystal operating frequency slightly. Exact trimming of the crystal frequency can be accomplished by setting a Persistent Store Key (PS Key) during manufacture. This in turn configures the capacitor type digital to analogue (DAC) within BlueCore2-External to pull the crystal frequency to the exact frequency required.

The crystal connection lines to XTAL_IN and XTAL_OUT should be kept short and clear of other noisy tracks. In particular, they should be routed away from the Flash memory lines, and away from the RF areas. Similarly, C11 and C12 should be connected with short tracks to a clean ground.

Consult the BlueCore2-External Data Book (bc212015lf-ds-001d) for the exact specification of the crystal.

2.4 Impedance Matching and Balun

L2, L3, C2 and C3 provide a balanced impedance matching network between the BlueCore2-External and T1. The balun is fed from a DC supply. L1 and C1 prevent RF from coupling onto the 1.8V power rail. In some cases (due to capacitive strays and differing PCB layout) C2 and C3 may need to be 2.2pF for optimal impedance matching.

2.5 Filter

Filter U3 is included for robustness of the link in the presence of high RF fields out of the Bluetooth band, such as from cellular telephones held in very close proximity (<20cm) to the Bluetooth transceiver. If the antenna provides adequate out-of-band rejection, U3 may be omitted in low cost applications.

The same filter, balun and matching components are used for both transmission and reception. No external RF switch is required.

2.6 Antenna

Unless extended range (>20m) is required, an expensive, high quality antenna is not required. Good results can be achieved with a simple flying wire antenna or planar inverted 'F' antenna (PIFA). Even accounting for the effects of proximity detuning, the performance of these simple, inexpensive antennae will be perfectly adequate.

2.7 RF Test Point

During manufacture some means of performing conducted RF measurements will be required. There are three means of doing this:

1. Via a permanently connected RF test point. Typically, this will be a pad on the rear of the PCB that can be probed using a bed-of-nails. Coaxial, spring-loaded probes are available that can provide a 50Ω connection to such a pad. Capacitor C6 should be fitted only when the RF tests are complete (this avoids testing with the antenna wired in parallel with the test point). Keep the test point on the RF track – do not lead a stub to it.
2. Via a switching RF connector, placed between the filter and the antenna. These 3 pin connectors have an integrated normally-closed switch that will pass RF through to the antenna. When a coaxial probe is inserted into the connector, the antenna is disconnected and RF diverted to the test harness. Although more expensive than a simple test point, it eliminates the need for a separate soldering step, where C6 is fitted after the RF testing.
3. Via the antenna, with the circuit board under test located in a test jig or TEM Cell. In this situation the isolation that can be achieved will determine whether multiple boards can be tested simultaneously, at least 90dB of isolation will be required between devices undergoing maximum output power.

Note:

In either of the above cases, some RF loss will be inevitable in the connection to the test harness. Given that this loss will be roughly consistent across tested PCBs (it is a systematic error), it may be calibrated out. It is important to account for this loss when setting the power table with BlueCore2-External; otherwise the maximum power allowed for Class-2 operation (+4dBm) can easily be exceeded.

2.8 VCO

R2 and C7 form part of the voltage controlled oscillator (VCO) that synthesises the transmission and reception frequency. In the same way as the crystal, the routing for R2 and C7 should be kept clear of other noisy PCB tracks (such as Flash memory lines and the RF components). C7 should be connected to a clean ground.

2.8.1 RF Reference Design

Detailed reference designs are available separately from CSR. These designs include layout design (Gerber files) for the critical RF section of the circuit. CSR strongly recommends that all customers refer to these designs when laying out the RF part of the circuit.

Note:

The BlueCore RS232 Cable Replacement Example Design is intended to be a design that accompanies the purchase of a Casira™ development kit. For more information, contact your local distributor or sales@csr.com.

2.9 RS232 Level Converter

The purpose of this block is to provide a connection from the RS232 levels to the 3.3V logic levels of BlueCore2-External and visa-versa. Converter chips are available from a number of suppliers, including Texas

Instruments and Maxim. The exact choice of level converter depends on whether the dongle is communicating to the DTE or DCE, and determines the number of drivers and receivers. For a dongle communicating to the DTE the MAX3238 can be used and for communication to the DCE the MAX3243 can be used.

2.10 Reset Circuit

C16 and R10 take the RESET pin on U1 high when the Bluetooth circuit is powered up. This ensures that U1 is held in reset whilst the 3.3V power rail reaches stability. Otherwise, U1 will attempt to fetch instructions from the Flash memory when the 1.8V rail reaches 1.6V. The supply to the Flash memory may not be ready at this point, and it will return invalid instructions to U1. This can lead to a boot-time failure of the firmware.

For further details on power-up sequencing, refer to CSR application note bc2x-an-001Pa available from <http://www.csr.support.com>.

At the moment the Bluetooth circuit is turned off, the midpoint of C16 and R10 will go negative. R12 limits the current drawn out of the RESET pin at this instance.

2.11 Switches

Switch SW1 pulls the PIO[6] pin low when pressed. Resistor R11 pulls the line high when the switch is released. Switch debouncing is performed in the software.

2.12 PIO Lines

The switches and Light Emitting Diodes (LEDs) may be moved to any of the PIO lines, although the ones shown match directly with CSR's example RS232 Cable Replacement source code. Indeed, additional switches or LEDs can be added if required. Unused PIO lines can be left unconnected as they have weak pull-down resistors internal to the BlueCore2-External Integrated Circuit (IC).

2.13 LEDs

LEDs D1 and D2 are driven directly from the PIO pins of U1. Typically these will pass around 2.5mA each with the given resistor values. This will brightly light a modern efficient LED.

2.14 Handshake Lines

The BlueLab SDK contains the two versions of software required, one version for the DTE and another version for the DCE. Figure 2.2: Equivalent Signal Connection of DTE to DTE and Figure 2.3: Equivalent Signal Connection of DTE to DCE are the diagrams depicting the equivalent cable and signal connections.

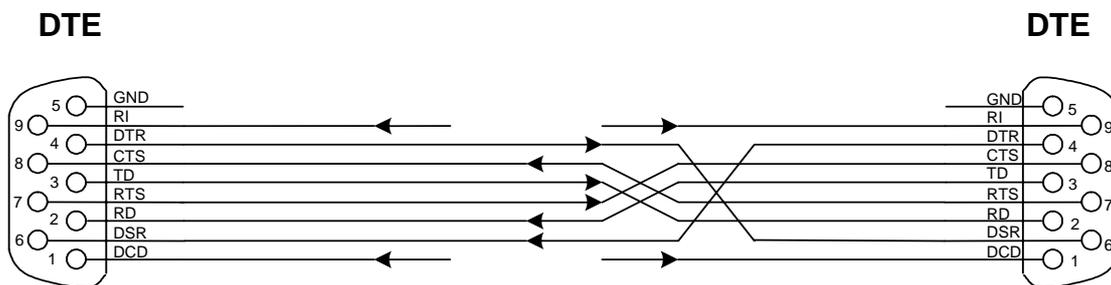


Figure 2.2: Equivalent Signal Connection of DTE to DTE

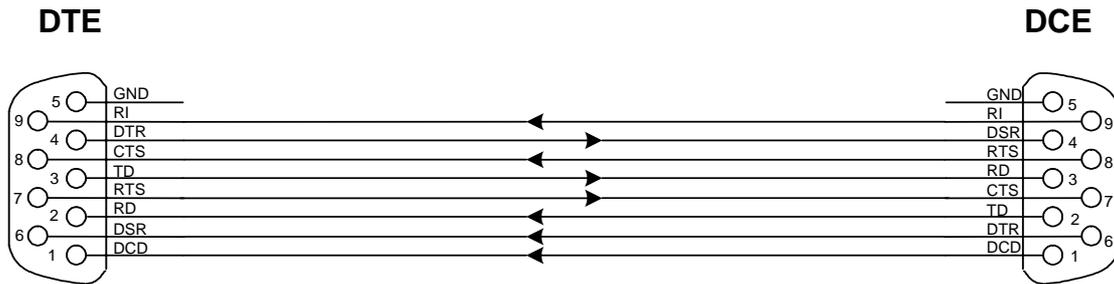


Figure 2.3: Equivalent Signal Connection of DTE to DCE

2.15 PCB

The design resides on a single PCB. The most complex routing is that of the Flash memory address and data lines. With care, the entire design can be routed on a four layer PCB. One of the four layers is devoted to ground, whilst another, forms a split power plane and carries some signal tracks.

2.15.1 Screening Can

A screening can should be placed over the BlueCore2-External device, Flash memory, decoupling capacitors, RF components, R3 and C8. This improves electromagnetic compatibility (EMC) and reduces the impact of the 2.4GHz RF field on the BlueCore2-External itself.

The screening can should be soldered down to a proper PCB ground plane at points around its perimeter not more than 6mm apart. Similarly, holes in the screening can should not be more than 3mm in diameter and should be as few in number as possible.

2.15.2 SPI Interface

The serial peripheral interface (SPI™) pins (SPI_MOSI, SPI_MISO, SPI_CS and SPI_CLK) should be brought out to probe points on the PCB. They are used to program, calibrate and interrogate the BlueCore2-External during development and production. Refer to CSR application note called BlueCore Interface SPI Signals to a PC (bcore-an-002Pa) for details on how the SPI interface is connected to a PC.

3 Performance of RS232 Cable Replacement

The RF section of this design is based on the BC2MOD2C module, and it has proven to have similar performance to this module. For details on RF performance, please refer to the BlueCore2-External Data Book available from <http://www.csrsupport.com>.

3.1 Achievable Data Rate

The maximum throughput of the example design is limited by the RS232 level converter (U5). The existing device is limited to 115Kbps, which allows an effective data rate of approximately 10KB/s (100Kbps). If necessary this can be updated to a faster, but more expensive device, e.g. Maxim MAX3245. In this way it is possible to achieve a throughput of up to 20KB/s (200Kbps). The baud rate used is determined by a PS Key setting, see document BlueCore2 Persistent Store Key Settings (bc02-an-005a).

3.2 Deep Sleep Implementation

In order to conserve power the user may want to consider using the deep sleep mode on the BlueCore2-External. The RS232 Cable Replacement Example Design does not implement deep sleep in the lower layer firmware. The issue is that to be woken up from deep sleep the BlueCore2-External would need to have data transmitted on the UART or include extra signals from the hardware to wake it up. Using data transmitted on the UART to wake the BlueCore2-External unfortunately causes the initial data used to be lost by the time the device is up and running at full speed. The implications this has on the firmware if deep sleep was to be implemented is as follows:

1. Waking the device up from deep sleep will cause loss of data, so the BlueCore2-External may need to be woken with a pre-amble. This has implications on data throughput speeds as a pre-amble would always need to be sent prefixed onto a group of data packets as it is not known if the BlueCore2-External device is asleep.
2. An alternative to the pre-amble scheme described above would be to implement a retransmission scheme, so that each packet of information sent from the host to the BlueCore2-External is acknowledged i.e. the data lost during the time BlueCore2-External is waking up is retransmitted until it is acknowledged. Again this has implications on data throughput speeds.
3. Another solution involves additional signalling from the attached device which this hardware design does not include, the RS232 pin out does not specify and the software does not allow for.

It is felt that in all of the above cases, these solutions to allow deep sleep to occur and wake-up the BlueCore2-External are outside of the scope of the lower layer firmware. They are part of the higher layer protocol and need to be implemented by the user. The user needs to add an error detection and correction scheme to their host software on top of the RS232 link. They need to see the link as essentially prone to noise i.e. that data may get lost or corrupted and be able to handle this, and then they will be able to take advantage of deep sleep.

Below is a list of items that need to be considered when trying to implement a deep sleep scheme. This is not necessarily a comprehensive list and further items may need to be taken into account:

1. Sniff and/or park mode needs to be enabled before the device will go into deep sleep
2. There are no default setting for sniff and park mode it is down to the user to experiment
3. Setting VmDeepSleepEnable(1) will permit the chip to go into deep sleep when using VM
4. Deep sleep will obey all the normal PS Keys such as PSKEY_DEEP_SLEEP_STATE
5. PSKEY_DEEP_SLEEP_CLEAR_RTS does not work with the user transport required for this application
6. The device will not go into deep sleep if the CTS is held high

The hardware for the line receiver circuit of the RS232 Cable Replacement Example Design has been designed so that the circuit can take advantage of lower power modes. The MAX3243 and the MAX3228 used for RS232 interface devices, which are U5 on Figure 6.1 and Figure 6.2 respectively, have an auto power down feature. The feature allows the device to shutdown and consumes approximately 1 μ A of power when there are no valid RS232 signals.

The circuit shown in Figure 6.1 and Figure 6.2 have the power down feature disabled this has been achieved by using a zero ohm resistor R6 to connect the FORCEON signal to the +3.3V supply. The PCB design is such that

there is an extra set of pads that allow R6 to be moved such that it connects FORCEON to 0V and as the FORCEOFF is connected to the +3.3V supply this will therefore enable the auto power down mode. For the RS232 devices to shutdown and consume low power there needs to be no valid RS232 level signals on the receiver side of the device. To indicate whether there is a valid signal the RS232 devices output the signal INVALID, which can be used by the BlueCore2-External device to wake it up from deep sleep. A high logic level on the INVALID output indicates the RS232 device has detected valid RS232 levels on its receive pins and has come out of low power mode.

The INVALID pin has been connected to PIO7 of the BlueCore2-External device and therefore can be used to wake-up the device from deep sleep as it is an active high signal for valid data and the BlueCore2-External requires a high level as its interrupt.

With respect to the BlueCore2-External caution also has to be taken with respect to the input levels on the PIO pins and being able to put the device into deep sleep. If a PIO is set as an input it needs to be at an active low state for BlueCore2-External to go into deep sleep, therefore receiver output pins that are connected from the RS232 interface device to the inputs on the PIOs of BlueCore2-External may need to have pull down resistors on the lines to ensure they are at a logic low when the output enables of the receiver outputs are tri-stated on the RS232 device.

To allow a PIO to be an interrupt it requires to be configured as an input and the PS Key called PSKEY_DEEP_SLEEP_PIO_WAKE needs to be set appropriately, if using the VM this needs to be set using VmDeepSleepEnable(1).

Note:

CSR does not support deep sleep in this application at present and implementation is very much the user's responsibility

4 Software Architecture

The software for the RS232 Cable Replacement example design has been engineered by CSR. A full Bluetooth software stack and application profile all execute from the external Flash connected to BlueCore2-External.

4.1 Block Diagram

Figure 4.1: Serial Port Software Architecture shows the layered architecture of the software.

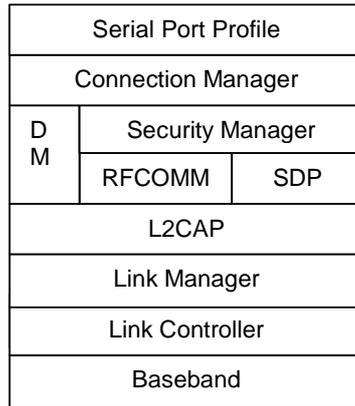


Figure 4.1: Serial Port Software Architecture

The link manager (LM), link controller (LC), L2CAP, SDP and RFCOMM are supplied as one binary file. The connection manger library and serial port applications are supplied as C source code and, once compiled in BlueLab, run in the VM.

CSR has developed the default behaviour of the serial port to be as intuitive as possible. BlueLab can be used to modify the standard behaviour. Possible modifications include:

- The way in which two devices are paired
- The types of interface to the device

5 Commissioning

The Bluetooth communications stack and RS232 Cable Replacement Application are all run from Flash memory. The Flash is also used to store module specific data (i.e., the Bluetooth address) in an area of memory known as the Persistent Store.

The software and default Persistent Store settings are programmed into the external Flash memory by programming a single image into the device via the SPI.

CSR supplies suitable software and tools to program the Flash in the BlueLab SDK. These tools download firmware to the Flash over an SPI cable. Refer to CSR documentation for details on how the SPI interface is connected to a PC.

5.1 Downloading Firmware

The BlueLab SDK ships with source code for the serial port firmware. This needs to be compiled into a binary file before it can be used to control the RS232 Cable Replacement Example Design. The BlueLab SDK documentation explains how this is done in detail.

BlueFlash™ is a flash programming utility supplied by CSR with BlueLab SDK. It requires a PC running Windows 98™, Windows Millennium™, Windows NT™ or Windows 2000™. It is used to download precompiled object code files to the RS232 Cable Replacement example design.

- These instructions assume that the BlueLab SDK is installed in:
C:\Program Files\BlueLab26
- Comments in **button type** are computer commands

To program the binary image into the external Flash memory the following steps should be taken:

1. Compile the RS232 Cable Replacement application source code following the instructions in:
C:\Program Files\BlueLab26\docs\spp.html
2. Connect the SPI interface cable between a parallel port on your PC and J1 (CON5) on the example RS232 Cable Replacement board.
3. Start BlueFlash from the **bin** directory of the BlueLab installation:
C:\Program Files\BlueLab26\bin\BlueFlash.exe
4. The dialog box in Figure 5.1: BlueFlash Dialog Box will be displayed

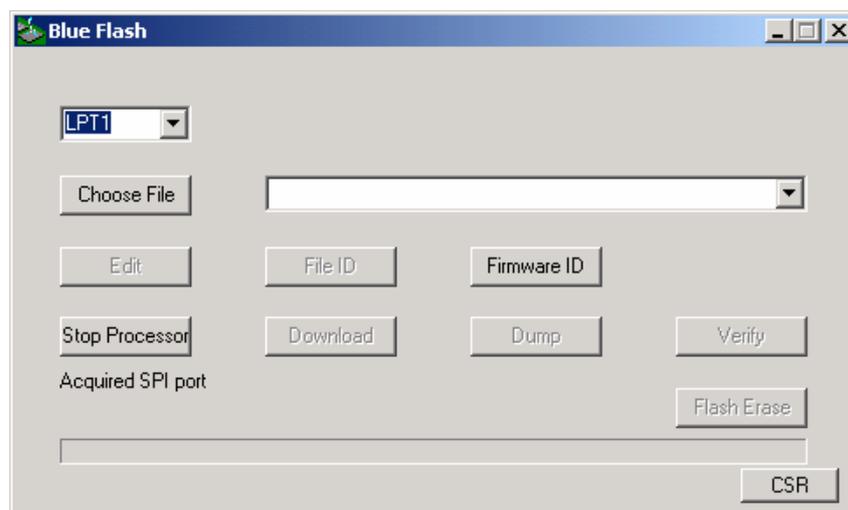


Figure 5.1: BlueFlash Dialog Box

5. Select the correct parallel port and press the button labelled **Stop Processor**. This will halt the processor on BlueCore, download a Flash programming utility into system RAM on the BlueCore and the dialog box will display the Flash device detected on the printed circuit board (PCB). Press the button labelled **Choose File** and browse to the location that contains the Flash binary image. Load the required Flash image, the default name of the RS232 Cable Replacement binary file is `spp_master_merge.xpv` or `spp_slave_merge.xpv` as appropriate.

Note:

Dongles need to be programmed in pairs, with one end running the `spp_master` application, while the other runs the `spp_slave` application.

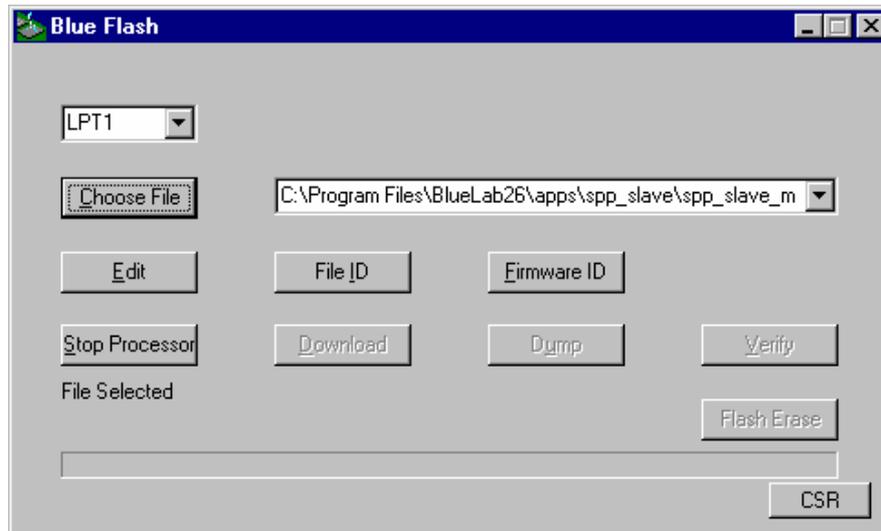


Figure 5.2: Choose File to be Loaded

6. Press the button labelled **Download** to program the external Flash memory with the object image. During programming the current status will be displayed. On completion the following dialog box will be displayed reporting that the image was successfully downloaded to the external Flash memory.

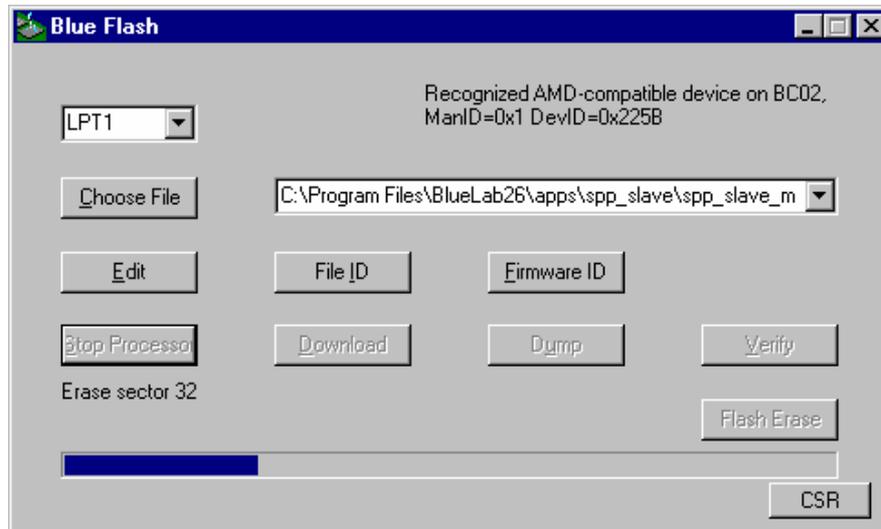


Figure 5.3: Program Flash Memory

5.2 Persistent Store Settings

The following Persistent Store settings shown in Table 5.1: Persistent Store Settings are essential for the RS232 Cable Replacement example design:

| Persistent Store Key | ID (Decimal) | Setting |
|-----------------------------|--------------|--|
| Host Interface | 505 | VM access to the UART |
| UART Configuration Bitfield | 516 | 168 (Decimal) |
| Access to PIO pins bitmap | 514 | 0 |
| TX and RX PIO control | 521 | PIO does not drive TX+RX, internal amp ramps power |
| UART baud rate | 516 | 115Kbps |
| Enable host query task | 508 | False |

Table 5.1: Persistent Store Settings

Note:

The PSTool application, included with the BlueLab SDK, is used to set Persistent Store settings. See the BlueLab SDK documentation for detailed PSTool instructions.

5.3 Steps for Pairing with Another Bluetooth RS232 Device

The following steps are required for pairing with another device acting as an RS232 Cable Replacement:

1. Power on the RS232 devices.
2. If the device has already been paired it will automatically try to establish a connection to its partner. The application stores the Bluetooth address of the opposite paired device.
3. If this fails, or if there is no address entry (i.e., if the device has never been paired) then the slave end will become discoverable waiting for incoming connections and the master end will begin scanning for suitable devices. If an inquiry finds a suitable device the application attempts to connect to it. Once a connection is successfully established the address of the other device is stored in the persistent store.

After successful pairing the paired devices will actively seek each other whenever powered up. To connect to another device the user must clear the Persistent Store by pressing SW1. This returns the device to unpaired state.

Table 5.2: Status LED Description defines the purpose of the status LEDs:

| LED | Description |
|-------------------------|---|
| RED – (D1) ON | Power – The device is powered and the internal code is running. |
| RED – (D1) Flash | No RS232 – The device is powered and the internal code is running but there are no valid RS232 levels on the input to the dongle. |
| GREEN – (D2) Slow Flash | (Master) Inquiry – Device is looking for equivalent devices to connect to. |
| GREEN – (D2) Fast Flash | (Slave) Discoverable – Device is waiting to be discovered and connected to. |
| GREEN – (D2) ON | Bluetooth Connection – Device has been paired and has established an ACL connection |

Table 5.2: Status LED Description

5.4 More Information

The BlueLab Software Development Kit comes with detailed documentation for the RS232 Cable Replacement application:

- For more detailed BlueLab SDK documentation see:
C:\Program Files\BlueLab26\docs\welcome.html

- For more information on setting up the Casira™ development system see:
C:\Program Files\BlueLab26\docs\gettingstarted.html

- For more information on the RS232 Cable Replacement application see:
C:\Program Files\BlueLab26\docs\SPP.html

- For an additional source of information check the CSR public newsgroups, for set up information see:
<http://www.csr.com/forum.htm>

6 Schematics

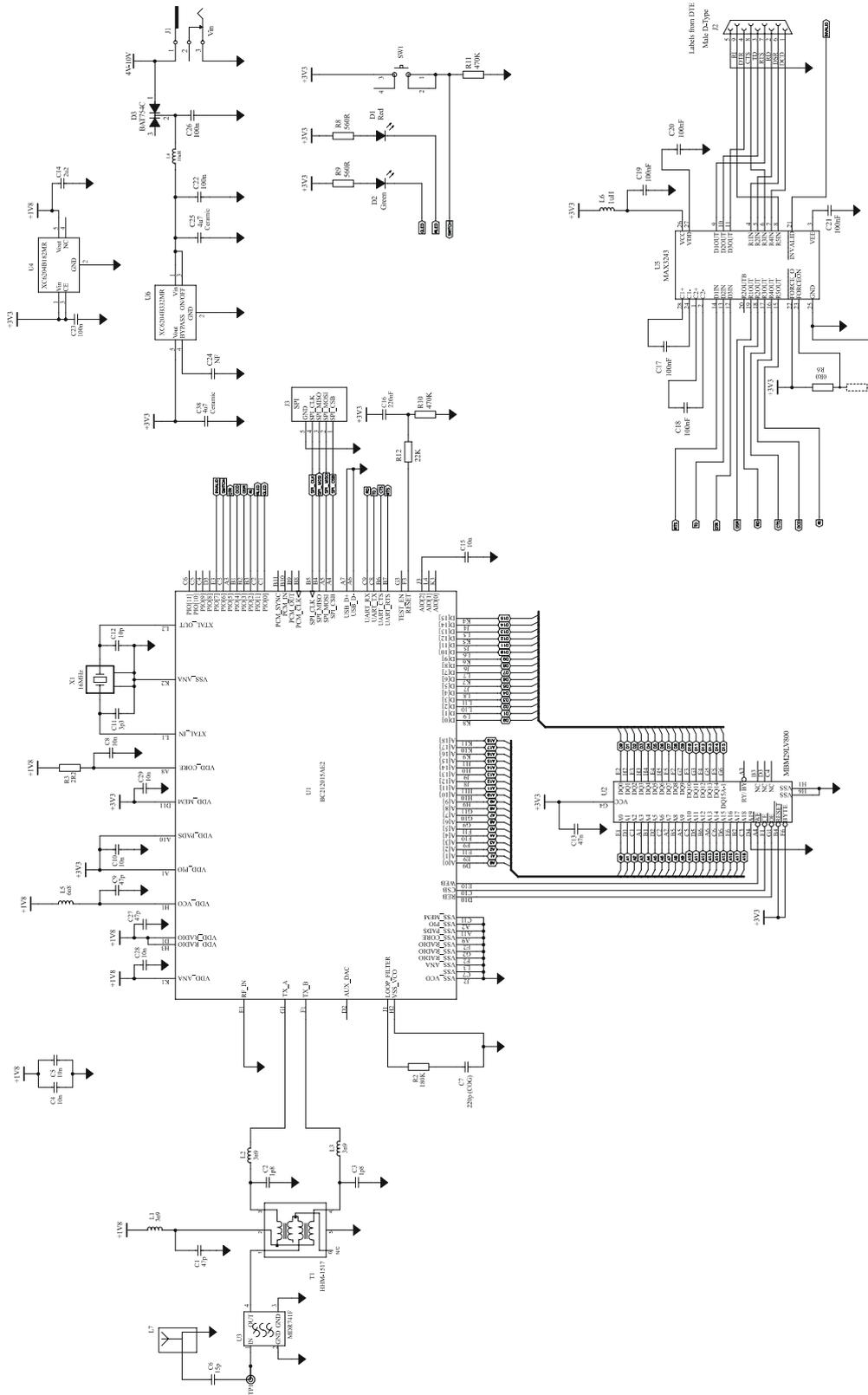
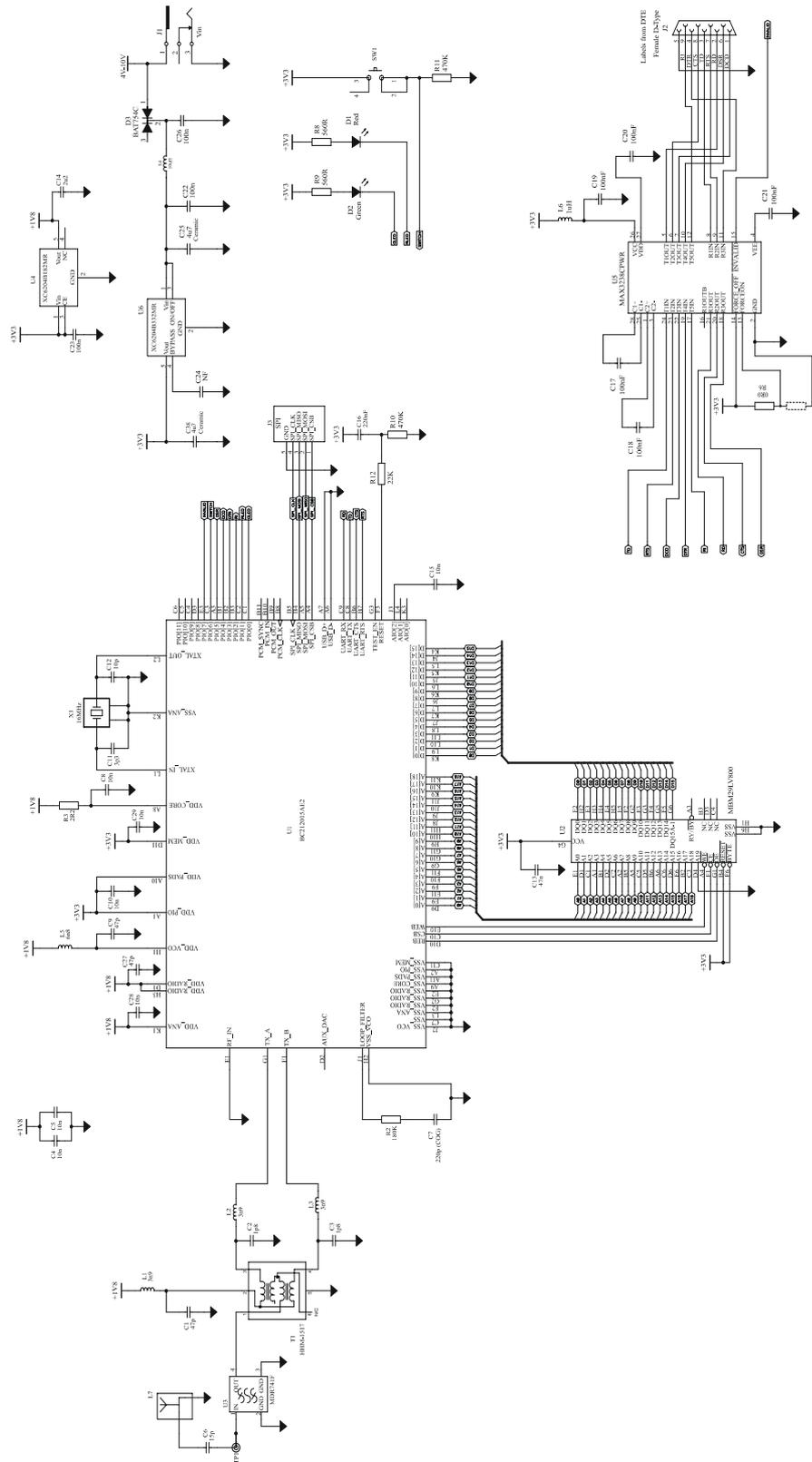


Figure 6.1: BlueCore RS232 Cable Replacement Circuit Diagram (DCE)



BlueCore™ RS232 Cable Replacement Example Design

Figure 6.2: BlueCore RS232 Cable Replacement Circuit (DTE)

7 Bill of Materials

For the Device Terminal Equipment (DTE) side.

| Item # | Used | Designator | Part Type | Tolerance/Rating | Material | Footprint | Description | Manufacturer and Part Number |
|--------|------|--|--------------------|------------------|----------|-------------------------|-------------------------|------------------------------|
| 2 | 1 | D3 | BAT 754C | | | SOT23 | Schottky Dual diode | Philips BAT754C |
| 3 | 1 | U1 | BlueCore2-External | | | BGA, 0.65mm pitch | BC02 Bluetooth | CSR BC212015DN |
| 4 | 2 | C2, C3 | 1p8 | 5% | COG | 0402 | Non-polarised capacitor | Murata GRP1555C1H1R8JZ01B |
| 6 | 1 | C11 | 3p3 | 5% | COG | 0402 | Non-polarised capacitor | Murata GRP1555C1H3R3JZ01B |
| 7 | 8 | C4, C5, C8, C10, C15, C16, C28, C29 | 3p4 | 10% | X7R | 0402 | Non-polarised capacitor | Murata GRP155R71A103KA01B |
| | 1 | C16 | 220n | 10% | X7R | 0603 | Non-polarised capacitor | Phycomp 220024606654 |
| 15 | 1 | C12 | 3p5 | 5% | COG | 0402 | Non-polarised capacitor | Murata GRP1555C1H100JZ01B |
| 16 | 1 | C6 | 3p6 | 5% | COG | 0402 | Non-polarised capacitor | Murata GRP1555C1H150JZ01B |
| 17 | 1 | C13 | 3p7 | 10% | X7R | 0402 | Non-polarised capacitor | Murata GRP155R71A473KA01B |
| 18 | 3 | C1, C9, C27 | 3p8 | 5% | COG | 0402 | Non-polarised capacitor | Murata GRP1555C1H470JZ01B |
| 22 | 8 | C17, C18, C19, C20, C21, C22, C23, C26 | 3p9 | 10% | X7R | 0603 | Non-polarised capacitor | Phycomp 222278655649 |
| 29 | 1 | C7 | 3p10 | 5% | COG | 0603 | Non-polarised capacitor | Phycomp 222286775221 |
| 30 | 1 | C24 | 3p11 | | | 603 | | |
| 31 | 1 | C14 | 3p12 | 10% | X7R | 0805 | Non-polarised capacitor | Phycomp 222224013667 |
| 32 | 2 | C25, C38 | 3p13 | | | 1206 | Ceramic | CAP-1206 |
| 34 | 1 | X1 | 3p15 | | | TSX10 | 16 MHz crystal | Toyocom TN4-25820 |
| 35 | 1 | J2 | | | | D9_FEMALE_SOLDER_BUCKET | | McMURDO SDE9S |
| 36 | 1 | T1 | HHM-1517 | | | HHM1517 | Balun 2.4 GHz | TDK HHM-1517 |
| 37 | 3 | L1, L2, L3 | 3n9 | | | 0402 | Inductor | Murata LPQ10A3N9B00 |
| 40 | 1 | L5 | 6n8 | | | 0402 | Inductor | Murata LPQ10A6N8B00 |
| 41 | 1 | L6 | 1uH | | | 805 | | L0805 |
| 42 | 1 | L4 | 10uH | | | 1210 | | L1210 |
| 43 | 1 | D2 | Green LED | | | LED-3MMR | Green LED | Kingbright L-424GDT |
| 44 | 1 | D1 | Red LED | | | LED-3MMR | Red LED | Kingbright L-424HDT |
| 45 | 1 | U5 | MAX2328 | | | SSOP28-6.6MM | RS232 level convertor | TI MAX3238CPWR |
| 46 | 1 | U2 | MBM29LV800 | | | BGA, 0.8mm pitch | 8MBit Flash Memory | Fujitsu MBM29LV800 |
| 47 | 1 | U3 | 741FILTER | | | 741FILTER | Antenna filter | Soshin MDR741 |
| 48 | 1 | J1 | DC Power Jack | | | POWER_JACK_SPECIAL | DC Power Jack | |
| 49 | 1 | R3 | 2R2 | 5% | | 0402 | Resistor | Phycomp 232270570228 |
| 50 | 1 | R2 | 180K | 5% | | 0402 | Resistor | Phycomp 232270587184 |
| 51 | 2 | R10, R11 | 470K | 5% | | 0402 | Resistor | Phycomp 232270587474 |
| 52 | 1 | R12 | 22K | 5% | | 0402 | Resistor | Phycomp 232270587223 |
| 54 | 2 | R8, R9 | 560R | 5% | | 0402 | Resistor | Phycomp 232270587561 |
| 56 | 1 | R6 | 0R0 | | | 0402 | Resistor | Phycomp 232270587000 |
| 58 | 1 | SW1 | Switch | | | SW-PTH-SMD | Push to make switch | |
| 60 | 1 | U4 | XC6204B182MR | | | SOT235 | LDO regulator 1.8V | Torex XC6204B182MR |
| 61 | 1 | U6 | XC6204B332MR | | | SOT235 | LDO regulator 3.3V | Torex XC6204B332MR |

For the Device Communications (DCE) side the following component changes are necessary.

| | | | | | | | | |
|----|---|----|---------|--|--|-----------------------|-----------------------|----------------|
| 35 | 1 | J2 | | | | D9_MALE_SOLDER_BUCKET | | McMURDO SDE9P |
| 45 | 1 | U5 | MAX3243 | | | SSOP28-6.6MM | RS232 level convertor | TI MAX3243CPWR |

Document References

| Document | Reference |
|---|---|
| Specification of the Bluetooth System | v1.1, Profiles, 1 December 2000 |
| BlueCore2-External Data Book | bc212015lf-ds-001d, Revision d, December 2002 |
| Selection of Flash Memory for Use with BlueCore | bc02-an-001Pb |
| BlueCore2-External Power Up Sequence | bc2x-an-001Pa |
| BlueCore2 Persistent Store Key Settings | bc02-an-005a |

Acronyms and Definitions

| | |
|------------|--|
| BlueCore™ | Group term for CSR's range of Bluetooth chips |
| Bluetooth® | Set of technologies providing audio and data transfer over short range radio connections |
| ACL | Asynchronous ConnectionLess |
| ADC | Analogue-to-Digital Converter |
| CSR | Cambridge Silicon Radio |
| DAC | Digital to Analogue Converter |
| DC | Direct Current |
| DCE | Data Communication Equipment |
| DTE | Data Terminating Equipment |
| DIL | Dual-In-Line |
| EMC | Electromagnetic Compatibility |
| IC | Integrated Circuit |
| LC | Link Controller |
| LED | Light Emitting Diode |
| LM | Link Manager |
| OEM | Original Equipment Manufacturer |
| PCB | Printed Circuit Board |
| PIFA | Planar Inverted 'F' antenna |
| PIO | Parallel Input / Output (or Programmable Input / Output) |
| POR | Power On Reset |
| PS Key | Persistent Store Key |
| RC | Resistor/Capacitor |
| RFCLI | RFCOMM Command Line Interface |
| RFCOMM | Protocol layer providing serial port emulation over L2CAP |
| RISC | Reduced Instruction Set Computer |
| SDK | Software Development Kit |
| SDP | Service Discovery Protocol |
| SIG | (Bluetooth) Special Interest Group |
| SPI | Serial Peripheral Interface |
| UART | Universal Asynchronous Receiver Transmitter |
| VCO | Voltage Controlled Oscillator |
| VM | Virtual Machine |

Record of Changes

| Date | Revision | Reason for Change: |
|-----------|----------|--|
| 10 OCT 02 | a | Original publication of this document |
| 12 OCT 02 | b | 4Mbit Flash has been changed to 8Mbit Flash |
| 01 MAY 03 | c | Correction of Torex part numbering on BOM and on IC U4 on Figure 6.1: BlueCore RS232 Cable Replacement Circuit Diagram (DCE) and Figure 6.2: BlueCore RS232 Cable Replacement Circuit (DTE), as well part references in Section 2.1.1 and minor style changes. Figure 6.1 and Figure 6.2 updated so that signal "SWITCH" is in correct state if user wants to implement a deep sleep scheme. |
| 03 JUN 04 | d | Removed reference bcore-an-002Pa |

BlueCore RS232 Cable Replacement

RS232CR-an-001Pd

June 2004

Bluetooth® and the Bluetooth logos are trademarks owned by Bluetooth SIG Inc, USA and licensed to CSR.

BlueCore™ is a trademark of CSR.

All other product, service and company names are trademarks, registered trademarks or service marks of their respective owners.

CSR's products are not authorised for use in life-support or safety-critical applications.