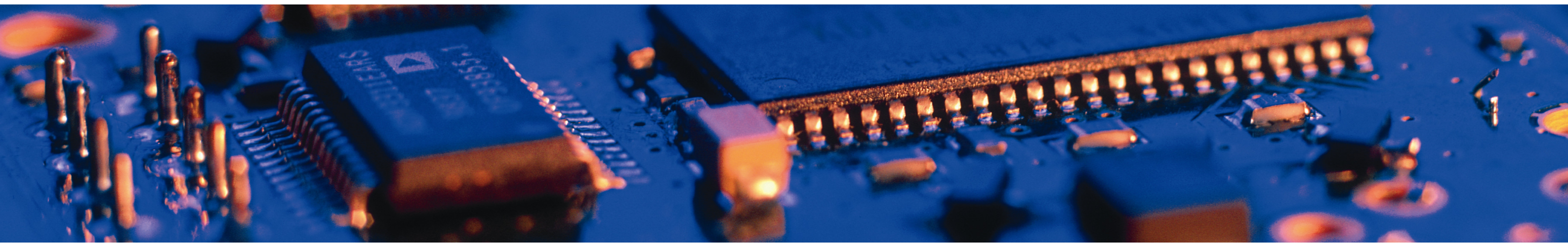
INDUSTRIAL BLUETOOTH™



Bluetooth™

FTP/OPP

User Manual

connectBlue

FTP/OPP

User Manual

The contents of this document can be changed by connectBlue AB without prior notice and do not constitute any binding undertakings from connectBlue AB. connectBlue AB is not responsible under any circumstances for direct, indirect, unexpected damage or consequent damage that is caused by this document.

Release: 200804

Document version: 1.01

Document number: cBProduct-0709-02 (4)

Printed in Sweden.

Trademarks
Registered trademarks from other companies are: Bluetooth is a trademark owned by the Bluetooth SIG, Inc. Microsoft™, Windows™, Windows NT™, Windows 2000™, Windows CE™, Windows ME™, Windows XP™ are registered trademarks from Microsoft Corporation.

Contents

1 Introduction

This manual describes how to use FTP and OPP packet with the connectBlue Serial Port Adapter.

Section 2 describes the different components of the delivery and Section 3 describes how to get started. The source code is briefly discussed in Section 4 and Section 5 explains how to port the code to another platform.

1.1 Related Documents

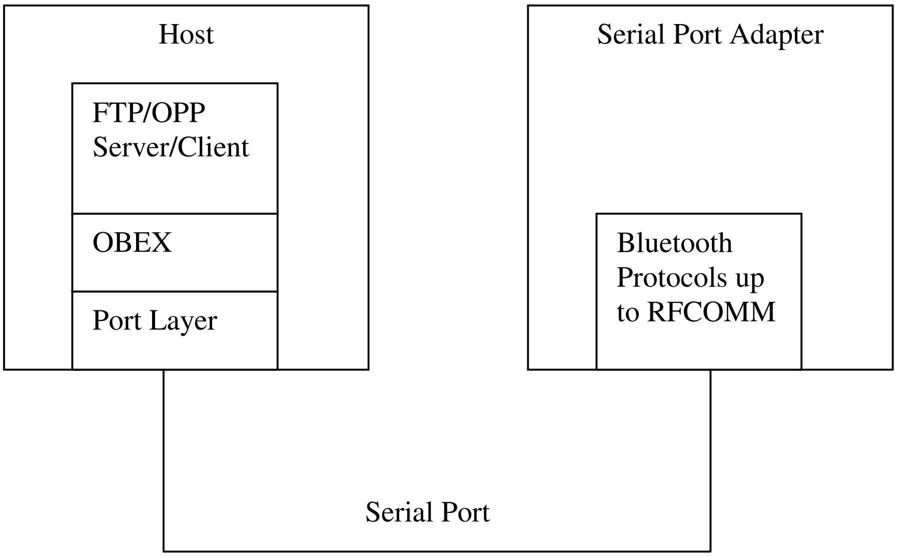
- The **Serial Port Adapter AT Commands** document, this document, contains a description of the AT commands supported in the Serial Port Adapter. It also contains information on how to use the AT commands to create Bluetooth applications.
- The **OEM Serial Port Adapter Electrical & Mechanical Datasheet** contains important information about the OEM Serial Port Adapter. Read this document if you are using the OEM Serial Port Adapter.

2 Overview

2.1 Architecture

The connectBlue Serial Port Adapter now supports both the File Transfer Profile (FTP) and the Object Push Profile (OPP).

The solution is intended for host-based applications where the host executes the actual FTP or OPP server/client and the Serial Port Adapter executes all Bluetooth functionality up to the RFCOMM protocol. Since most of the Bluetooth stack is executed in the Serial Port Adapter the solution is ideal for hosts with a limited amount of memory.



Typically, to use the FTP or OPP code in a new host the following must be done:

- Modify port layer to new environment. This is very easy.

- Update/develop FTP/OPP server/client according to the specific customer requirements.

All development and testing (except port layer) can be done on a PC with a Serial Port Adapter connected to it.

There are currently both FTP and OPP servers available for a Windows platform to use for testing or as a starting point for development.

2.2 Qualification

Please note that the FTP and OPP profile support has not been Bluetooth qualified. This means that the customer is responsible to Bluetooth qualify the final product according to the specific customer configuration and requirements.

For SIG members such as connectBlue, a tool can be downloaded for free, and all PRD v2.0 profile tests can be done according to the customer specific configuration. As before the SIG listing fee is still mandatory. It is possible that there is some code missing to pass the PRD 2.0 profile testing.

2.3 OBEX

The OBEX protocol is needed to implement both the File Transfer Profile (FTP) and the Object Push Profile (OPP). Both profiles mainly describe how to use the OBEX protocol to comply with the Bluetooth specification.

The OBEX protocol that executes in the host is developed by the OpenOBEX project, which is an open source project to implement the OBEX protocol. It was originally implemented to run in a Linux environment. The source code is licensed under LGPL. For more information see *www.openobex.org*.

connectBlue has modified the code to use a COM port as the transport media and a connectBlue Serial Port Adapter connected to the COM port. The Serial Port Adapter is configured using AT commands. The Bluetooth connection is handled by configuring the Serial Port Adapter, which means that all parts of the OpenOBEX code concerning setting up the transport media is not needed.

2.4 MD5

The MD5 Message-Digest Algorithm is used for authentication. The algorithm is developed by RSA Data Security, Inc. License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

There are several free MD5 algorithms implemented in C and it should be no problem to change if the used one is not acceptable.

3 Getting Started

The following section describes how to set up the Serial Port Adapter, the FTP Server and the OPP Server.

3.1 Setting up the Serial Port Adapter

Firmware is downloaded to the module using Blueberry Loader 1.5. Settings are modified with the Serial Port Adapter Toolbox. These applications are available from http://www.connectblue.se/support/. (For Blueberry Loader download the zip-file called Flash Loader).

1. Connect the Serial Port Adapter to a COM port on a PC.
2. Follow the instructions in the *readme.txt* file for the Blueberry Loader, but change the firmware file from the default to the one in the firmware directory in this delivery. Since the paths are relative you also need to update the paths of the other files after adjusting the firmware file.
3. To set up for FTP the default server mode needs to be changed. Open the Toolbox and connect. Switch to AT Mode when connected to the Serial Port Adapter. Issue the command at*addsp=X,1 with X replaced with an appropriate number depending on which profiles to use. See the following table for values for X and their corresponding profiles. Restart the Serial Port Adapter after changing the server mode.

X=4	FTP
X=5	OPP
X=6	FTP,SPP
X=7	OPP,SPP

The Serial Port Adapter is now set up for use with either the FTP or OPP Server.

3.2 Running the FTP Server on Windows

1. Open the openObex FTP project in the build folder in Microsoft Developer Studio 6.0.
2. In file cp_port.h in the lib folder there is a #define called ROOT_DIR. This constant designates the root directory used by the server. Change it to an appropriate value. The default value is "C:\\Temp\\Ftp".
3. Adjust the COM port, which is hard coded in the application. The port is the first argument to the function CreateFile called in com_open in file cb_port.c. Default value is COM1, 57.6 kbits/s, 8N1, HW flow Ctrl.
4. Add "#define FORCE_FTP_AUTHENTICATION" in the cb_port.h file to enforce the use of authentication to setup an OBEX connection. As default it is commented out.

5. Compile the project. It compiles with 15 warnings on warning level 3.
6. Run the application. From a remote FTP client it should now be possible to access the files in the folder to which ROOT_DIR is set.

3.3 Running the OPP Server on Windows

1. Open the openObex OPP project in the build folder in Microsoft Developer Studio 6.0.
2. Change the #define VCARD_DIR in file cb_port,.h to an appropriate value. It designates where the owner's business card is stored and also where to put received objects (typically business cards). The default is "C:\\Temp\\Vcards"
3. Change the #define BUSINESS_CARD to the name of the file containing the owner's business card, which will be sent to clients on request. The default is bluetooth.vcf and it should be located in the VCARD_DIR above.
4. Adjust the COM port in the same way as for the FTP Server.
5. Compile the project. It compiles with 15 warnings.
6. Run the application. Connect with an OPP client (this could be a cell phone for example) to send and receive business cards.

4 Source Code

The host software is divided into four parts: Port layer, OBEX, MD5 and FTP/OPP Server. The FTP and OPP servers are implemented for reference.

Port layer
Files cb_port.c and cb_port.h contain the port layer to the Windows environment. It mainly consists of functions to handle COM ports and accessing the file system of the host.

Of the functions in the port layer only the FTP/OPP Server uses the following:

com_open, file_open, file_close, file_read, file_write, file_set_path.

The FTP Server additionally uses these functions:

file_directory, file_create_dir, file_delete, md5_getSeed, md5_getPassword

Only the OBEX layer uses following functions: com_is_data_available, com_write, com_read, port_ntohs, port_htons, port_ntohl, port_htonl.

OBEX
OpenOBEX implements the OBEX layer. The OBEX API is in the file obex.h. The files implementing OBEX are databuffer.c, databuffer.h, usbobex.h and all files whose names start with obex.

MD5
Used for authentication. Implementation is in files md5.c and md5.h. Not used in the OPP Server.

FTP Server
The FTP server is implemented in file main_ftp_server.c.

OPP Server
The OPP Server is implemented in file main_opp_server.c.

The server first opens the COM port and initializes OpenOBEX to receive a handle. Then the main (infinite) loop is entered which calls OBEX_HandleInput. This function repeatedly checks for data on the COM port. When there is incoming data the server part of OpenOBEX will be called. Events are delivered to the server via a callback function called obex_event. The function performs a switch-command on the received event to execute the correct function. Commands may be received on the events.

For example, when the server receives OBEX_EV_REQ (indicating an incoming request) it will perform a switch statement on the possible commands, see function server_request in file main_ftp_server.c.

5 Porting To Another Platform

When porting the FTP/OPP Server and OpenOBEX to a host other than Windows there are a few issues to consider. Most importantly, the functions in the files cb_port.c and cb_port.h must be rewritten for the new platform. It might also be necessary to make some changes in OpenOBEX itself, described in the following paragraphs.

The OpenOBEX implementation assumes that structures (defined by the keyword "struct" in C) are packed. That is, they are not padded to a particular size. On the GCC compiler for ARM this can be achieved by defining the symbol PACKED as follows in file obex_header.h.

#define PACKED __attribute__((__packed__))

It may also be necessary to change the definitions in file common_types.h to match the sizes of the data types on the new platform.

5.1 Memory usage

The FTP Server uses approximately 40kb of ROM. When running the OPP Server the ROM usage is about 34kb.