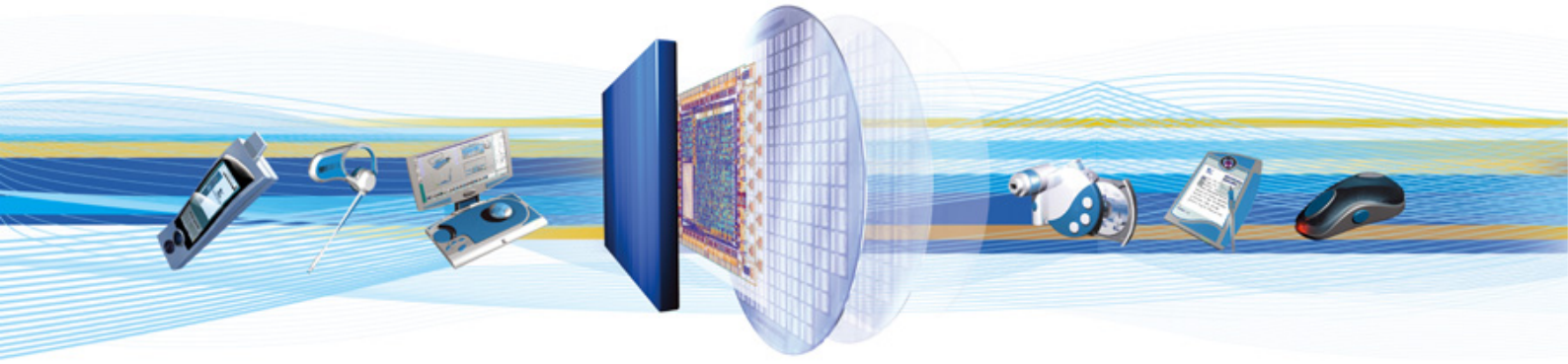
CSR Synergy Bluetooth 18.2.0

TPT Target Platform Tester

Demo Description

November 2011



Cambridge Silicon Radio Limited

Churchill House
Cambridge Business Park
Cowley Road
Cambridge CB4 0WZ
United Kingdom

Registered in England and Wales 3665875

Tel: +44 (0)1223 692000
Fax: +44 (0)1223 692001
www.csr.com

Contents

List of Figures

List of Tables

1 Target Platform Tester

1.1 Generally

The CSR Synergy Bluetooth Target Platform Tester (TPT) demo application can be used to determine the performance of a target platform before porting the full CSR Synergy Bluetooth package to it. The TPT demo application is designed to run on a standard Windows PC, which is interfacing to a CSR Bluetooth module, e.g. a Casira Bluetooth development kit. To use the TPT demo application, the TPT application must be running on the target platform (see **Figure 1**).

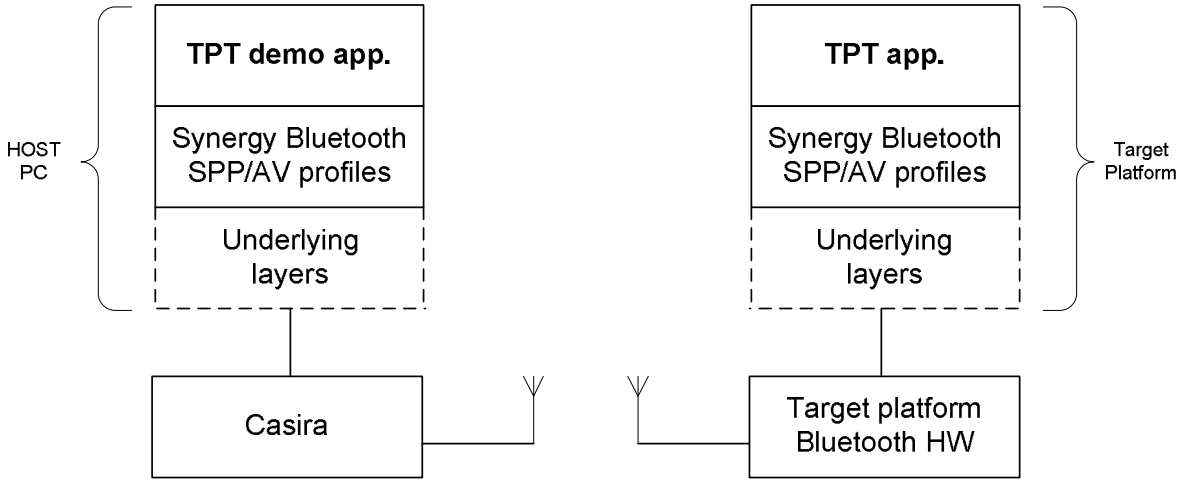


Figure 1: Physical setup

The figure above shows the physical setup of the demo application scenario. The host PC runs the host stack and the TPT demo application, while only the TPT application should be running on the target platform. The TPT application must be compiled to run on the appropriate target platform.

1.2 Use of the program hci_tpt_client_demo_app.exe

Start the program hci_tpt_client_demo_app.exe:

The program must be executed using certain parameters – one mandatory (-O) and two optional (-B and –C). This can be done using e.g. a 'command prompt'. The parameters are described below:

-B <baudrate>	specifies the baud rate for the COM port connected to the Casira. It should be set to the default baud rate needed by the BC chip. If no parameter is given, the default value of 115200 baud will be used.
-C <COM port>	specifies the COM port number connected to the Casira. If no parameter is specified, com1 will be used as default value.
-O < file name>	specifies which configuration file the program should use. Details on the configuration file syntax etc. are given in **Table 1**.

An example of the program execution running on com6 at baud rate 921600 using `tpt_config.txt` as configuration file is given below:

```
> hci_tpt_client_demo_app.exe –B 921600 –C com6 –O tpt_config.txt
```

Note: The following descriptions are based on a configuration where the TPT demo application is running BCSP over a serial interface. If a USB interface is desired, this can be obtained by using the `hci_tpt_client_demo_app_usb.exe` program. Note that the USB program does not use the –B and –C options. Otherwise, its behaviour is similar to the program described.

Using the program hci_tpt_client_demo_app.exe:

Once the application is started, it will try to read the configuration file specified by the –O option and determine which test(s) to run in which configurations. When that is done the program automatically tries to establish a Bluetooth connection to the module specified in the configuration file (see **Table 1**). Once the Bluetooth connection is established, the program will instruct the TPT application running on the target platform how it should behave during the next test and then start the test. The TPT application on the target platform will signal to the `hci_tpt_client_demo_app.exe` program when it considers the test to be finished.

After the `hci_tpt_client_demo_app.exe` program has received a finish-signal from the target platform and it has finished the test itself, it will ask the TPT application running on the target platform for test results if applicable. This is only applicable for tests where the target platform receives test data from the `hci_tpt_client_demo_app.exe` program. The test results will be saved in a Comma Separated Values (CSV) file. The format of this file is described in Section “Reading the result files” on page 6.

When all tests are completed, the `hci_tpt_client_demo_app.exe` program will calculate the final test results and summarize them in a number of CSV files. The format of these files is described in Section “Reading the result files” on page 6.

1.3 Use of the program hci_tpt_server_app.exe

Start the program hci_tpt_server_app.exe:

The program must be executed using two optional parameters (-B and –C). This can be done using e.g. a ‘command prompt’. The parameters are described below:

-B <baudrate>	specifies the baud rate for the COM port connected to the Casira. It should be set to the default baud rate needed by the BC chip. If no parameter is given, the default value of 115200 baud will be used.
-C <COM port>	specifies the COM port number connected to the Casira. If no parameter is specified, com1 will be used as default value.

An example of the program execution running on com6 at baud rate 921600 using `tpt_config.txt` as configuration file is given below:

```
> hci_tpt_server_app.exe –B 921600 –C com6
```

Note: The following descriptions are based on a configuration where the TPT demo application is running BCSP over a serial interface. If an USB interface is desired, this can be obtained by using the `hci_tpt_server_app_usb.exe` program. Note that the USB program does not use the –B and –C options. Otherwise, its behaviour is similar to the program described.

Using the program hci_tpt_server_app.exe:

When the `hci_tpt_server_app.exe` program is started, it should be noted, that it does not have any user interface, i.e. noting is printed on the screen etc. This is done to ease the porting of the program to different target platforms.

1.4 Use of the configuration file

The default configuration file released with the TPT demo application is named `tpt_config.txt` and it contains the following parameters:

Parameter	Description
SPP_SERVER_SEND_TEST	Set to (Y) to run a test, which measures the throughput when sending data from the target platform to the TPT demo application using the Serial Port Profile. Set to (N) otherwise.
SPP_CLIENT_SEND_TEST	Set to (Y) to run a test, which measures the throughput when sending data from the TPT demo application, to the target platform using the Serial Port Profile. Set to (N) otherwise.
AV_SERVER_SEND_TEST	Set to (Y) to run a test, which measures the throughput when sending data from the target platform to the TPT demo application using the Audio Visual Profile. Set to (N) otherwise.
AV_CLIENT_SEND_TEST	Set to (Y) to run a test, which measures the throughput when sending data from the TPT demo application, to the target platform using the Audio Visual Profile. Set to (N) otherwise.
RUN_SBC_ENCODE	Set to (Y) to run SBC encoding on the target platform while testing. Set to (N) otherwise. Running SCB encoding while testing will load the processor of the target platform.
RUN_INQUIRY	Set to (Y) to run Bluetooth inquiry on the target platform while testing. Set to (N) otherwise. Running Bluetooth inquiry while testing will load the radio of the target platform.
CSR_BT_MAX_OBEX_SIGNAL_LENGHT	Maximum packet size to use. If only one packet size is desired, both this parameter and MIN_PACKET_SIZE must be set to the desired value. Also PACKET_SIZE_INCREMENT should be set to zero in this case.
MIN_PACKET_SIZE	Minimum packet size to use. If only one packet size is desired, both this parameter and MAX_PACKET_SIZE must be set to the desired value. Also PACKET_SIZE_INCREMENT should be set to zero in this case.
PACKET_SIZE_INCREMENT	The “step” between the packet sizes to use. Note that MAX_PACKET_SIZE will always be included in the “packet size test list”. For example if MAX_PACKET_SIZE = 201, MIN_PACKET_SIZE = 100 and PACKET_SIZE_INCREMENT = 100, the packet sizes tested will be 100, 200 and 201. If only one packet size is desired, this parameter should be set to zero and both MIN_PACKET_SIZE and MAX_PACKET_SIZE must be set to the desired value.
MAX_NUMBER_TO_SEND	The number of packets to send during one test. The default value is 1000 packets, and it is not recommended to use a lower value as this will make the accuracy of different target platform timers influence the test results significantly. However if desired, it is possible to use a lower number.
SERVER_ADDRESS	The Bluetooth address of the target platform to connect to.
SPP_SERVER_SEND_FILE	Name of file in which the test results for the test where the target platform sends data to the TPT demo application using the Serial Port Profile are stored. A “.csv”-extension will be added to the filename.
SPP_CLIENT_SEND_FILE	Name of file in which the test results for the test where the TPT demo application sends data to the target platform using the Serial Port Profile are stored. A “.csv”-extension will be added to the filename.
AV_SERVER_SEND_FILE	Name of file in which the test results for the test where the target platform sends data to the TPT demo application using the Audio Visual Profile are stored. A “.csv”-extension will be added to the filename.

Parameter	Description
AV_CLIENT_SEND_FILE	Name of file in which the test results for the test where the TPT demo application sends data to the target platform using the Audio Visual Profile are stored. A ".csv"-extension will be added to the filename.
AV_SERVER_DELAY_FILE	Name of file in which the delay between the data packets in the test where the target platform sends data to the TPT demo application using the Audio Visual Profile are stored. A ".csv"-extension will be added to the filename.
AV_CLIENT_DELAY_FILE	Name of file in which the delay between the data packets in the test where the TPT demo application sends data to the target platform using the Audio Visual Profile are stored. A ".csv"-extension will be added to the filename.

Table 1: Configuration parameters

Note that all parameter values must be surrounded by round brackets to enable the TPT demo application to interpret the values correctly. Also it should be noted that the parameter names can not be changed without compromising the TPT demo application's ability to interpret the configuration correctly.

1.5 Reading the result files

When all tests are complete, three different types of results files will exist. These are:

- Intermediate result files
- Throughput result files and
- Delay result files (This type will exist only if one or more AV tests are performed)

The characteristics and properties of the different types are described below.

Intermediate result files

The names of the result files are constructed in the following way: <Result file name from configuration file>_<packet size>_<number of packets>`.csv`, i.e. an intermediate result file for a test where the result file name is specified to `spp_server_results` in the configuration file, the packet size is 300 and the number of packets is 1800, will be named `spp_server_results_300_1800.csv`.

As the intermediate result file name extension indicates, the file contains Comma Separated Values. The values will be written in the following order:

- Delay between the first packet and the second packet
- Sequence number of the first packet
- Delay between the second packet and the third packet
- Sequence number of the second packet
- Delay between the third packet and the fourth packet
- Sequence number of the third packet
- Etc…

Note: For SPP tests all sequence numbers will be zero.
For example, an intermediate result file containing the following data: `0,0,10000,1,60000,4,` it should be interpreted as this:

- The delay between the first and the second packet is 0µs (or less than a timer tick on the given platform)
- The sequence number of the first packet received is 0 (as expected)
- The delay between the second and the third packet received is 10,000µs.

- The sequence number of the second packet is 1 (as expected)
- The delay between the third and the fourth packet received is 60,000µs.
- The sequence number of the third packet received is 4. This means that two packets were lost as it was expected to receive a packet with sequence number 2.

Throughput result files

The throughput result file names are given in the configuration file parameters SPP_SERVER_SEND_FILE, SPP_CLIENT_SEND_FILE, AV_SERVER_SEND_FILE and AV_CLIENT_SEND_FILE. A ".csv" extension will be added to these parameters to indicate that the files contain Comma Separated Values. The values will be written in the following order:

- Number of packets in current test
- Packet size in bytes
- Total send time in µs
- Throughput in kbytes/s
- Total number of packet errors

Every line in a throughput result file represents one test. For example, a throughput result file containing the data below:

```
1000,100,2684000,37,0
1000,300,4987000,60,0
1000,500,7561000,66,0
1000,672,10736000,62,4
```

should be interpreted as this:

- When sending 1000 packets using a packet size of 100 bytes, the total data transfer time is 2,684,000µs, which means that the resulting throughput is 37kbytes/s. The total number of errors for this test was zero
- When sending 1000 packets using a packet size of 300 bytes, the total data transfer time is 4,987,000µs, which means that the resulting throughput is 60kbytes/s. The total number of errors for this test was zero
- When sending 1000 packets using a packet size of 500 bytes, the total data transfer time is 7,561,000µs, which means that the resulting throughput is 66kbytes/s. The total number of errors for this test was zero
- When sending 1000 packets using a packet size of 672 bytes, the total data transfer time is 10,736,000µs, which means that the resulting throughput is 62kbytes/s. The total number of errors for this test was four.

Note: If the total test time is lower than 1 second (1,000,000µs), the throughput will be set to 4,294,967,295 (0xFFFFFFFF) to indicate than the test is invalid. The reason to deem the test invalid in this case, is that one second is too short a period to regard the timer accuracy of the different platforms to be insignificant.

Delay result files

The Delay result files are only created for AV tests and the names are given in the configuration file parameters AV_SERVER_DELAY_FILE and AV_CLIENT_DELAY_FILE. A ".csv" extension will be added to these parameters to indicate that the files contain Comma Separated Values. The values will be written in the following order:

- Number of packets in current test
- Packet size in bytes

- Time delay between first- and second packet received in µs.
- Time delay between second- and third packet received in µs.
- Time delay between third- and fourth packet received in µs.
- Etc…

Every line in a Delay result file represents one test. For example, a delay result file containing the data below:

```
1000,100,0,10000,0,0,
1000,300,30000,0,0,10000,
1000,500,0,0,66000,0,
1000,672,10000,12000,20000,15000,
```

should be interpreted as this:

- When sending 1000 packets using a packet size of 100 bytes, the delay between second and third packet was 10,000µs, and the delay between the rests of the packets was zero (or less than a timer tick on the given platform).
- When sending 1000 packets using a packet size of 300 bytes, the delay between first and second packet was 30,000µs, and the delay between the fourth and fifth packet was 10,000µs. The delay between the rests of the packets was zero (or less than a timer tick on the given platform)
- When sending 1000 packets using a packet size of 500 bytes, the delay between third and fourth packet was 66,000µs, and the delay between the rests of the packets was zero (or less than a timer tick on the given platform)
- When sending 1000 packets using a packet size of 672 bytes, the delay between the packets were 10,000µs, 12,000µs, 20,000µs and 15,000µs respectively

Terms and Definitions

BlueCore®	Group term for CSR's range of Bluetooth wireless technology chips
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CSR	Cambridge Silicon Radio
UniFi™	Group term for CSR's range of chips designed to meet IEEE 802.11 standards

Document History

Revision	Date	History
1	5 MAR 11	Ready for release 18.1.0
2	26 SEP 11	Ready for release 18.2.0

TradeMarks, Patents and Licences

Unless otherwise stated, words and logos marked with ™ or ® are trademarks registered or owned by CSR plc or its affiliates. Bluetooth® and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc. and licensed to CSR. Other products, services and names used in this document may have been trademarked by their respective owners.

The publication of this information does not imply that any licence is granted under any patent or other rights owned by CSR plc.

CSR reserves the right to make technical changes to its products as part of its development programme.

While every care has been taken to ensure the accuracy of the contents of this document, CSR cannot accept responsibility for any errors.

No statements or representations in this document are to be construed as advertising, marketing, or offering for sale in the United States imported covered products subject to the Cease and Desist Order issued by the U.S. International Trade Commission in its Investigation No. 337-TA-602. Such products include SiRFstarIII™ chips that operate with SiRF software that supports SiRFInstantFix™, and/or SiRFLoc® servers, or contains SyncFreeNav functionality.

Life Support Policy and Use in Safety-critical Compliance

CSR's products are not authorised for use in life-support or safety-critical applications. Use in such applications is done at the sole discretion of the customer. CSR will not warrant the use of its devices in such applications.

Performance and Conformance

Refer to www.csrsupport.com for compliance and conformance to standards information.