

CC256x MSP430 TI's Bluetooth Stack Basic SPPDemo APP Improving throughput v14

Contents

Throughput measurement on the v1.4 release of TI's Bluetooth Stack on MSP430

Sending and Receiving data

Demo Setup

Application overview

MSP430 Server

Data Transfer from Sample Application to MSP430

Data Transfer from MSP430 to Sample Application(MSP430 Send)

MSP430 Client

Throughput measurement on the v1.4 release of TI's Bluetooth Stack on MSP430

This document describes the setup of the SPP demo and configuring the various parameters to achieve maximum throughput. We use the BR/EDR portion of the SPPL Demo but the SPPDemo can be used as well.

Note: This document is for the TI's Bluetooth Stack release V1.4 only. For the TI's Bluetooth Stack release v1.3 refer here (http://processors.wiki.ti.com/index.php/CC256x_MSP430_TI's_Bluetooth_Stack_Basic_SPPDemo_APP_Improving_throughput).

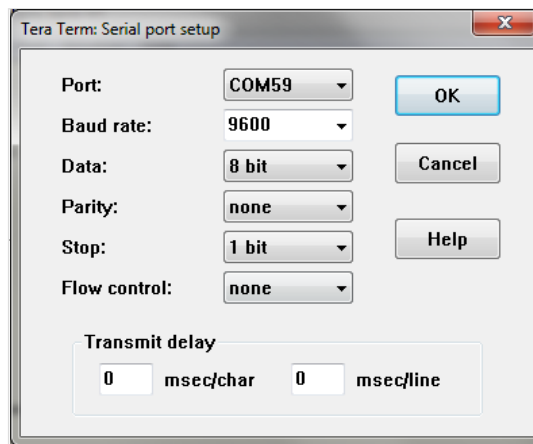
Sending and Receiving data

To achieve maximum throughput using TI's Bluetooth Stack while receiving data, read the data as soon as the application receives a data indication.

While sending data, write data using SPP_Data_Write as fast as possible. If SPP_Data_Write returns a value less than what you are trying to write, that means the outgoing path is full and the application should wait to write until more buffers are available. The stack will indicate that more space is available using the etPort_Buffer_Empty_Indication event in the SPP callback. Once you receive this event, you can send more data. The user will have to take care to send any unsend bytes in the previous call by using the return code of SPP_Data_Write. For reference the look at the CBSend command on the SPPL Demo (Send on the SPP Demo).

Demo Setup

Refer to the SPP Demo setup wiki for detailed instructions on how to get SPPDemo running. Use IAR or CCS to flash the SPPDemo on to the MSP430. Once the code is flashed use the following settings in your terminal application.



This should setup the MSP430 for use for our demo.

Application overview

MSP430 Server

We setup the MSP430 as a server and a client in two separate cases. We use a Windows SPP application to test against the MSP430(referred to as the Sample application). In our first scenario we use the MSP430 as the Server and the Sample application as the Client. We open a server on the MSP430 using Open 1 and connect to it from the Sample application by doing a scan and connecting to its BD-Address.

```

COM17:9600baud - Tera Term VT
File Edit Setup Control Window Help
SPP*LE>server
*****
* Command Options General: Help, GetLocalAddress, SetBaudRate
* Quit,
* Command Options BR/EDR: Inquiry, DisplayInquiryList, Pair,
* EndPairing, PINCodeResponse,
* PassKeyResponse,
* UserConfirmationResponse,
* SetDiscoverabilityMode,
* SetConnectabilityMode,
* SetPairabilityMode,
* ChangeSimplePairingParameters,
* GetLocalName, SetLocalName,
* GetClassOfDevice, SetClassOfDevice,
* GetRemoteName, SniffMode,
* ExitSniffMode, Open, Close, Read,
* Write, GetConfigParams,
* SetConfigParams, GetQueueParams,
* SetQueueParams, Loopback,
* DisplayRawModeData, AutomaticReadMode,
* CBSend.
* Command Options GAP/LE: SetDiscoverabilityMode,
* SetConnectabilityMode,
* SetPairabilityMode,
* ChangePairingParameters,
* AdvertiseLE, StartScanning,
* StopScanning, ConnectLE,
* DisconnectLE, PairLE,
* LEPasskeyResponse,
* QueryEncryptionMode, SetPasskey,
* DiscoverGAPS, GetLocalName,
* SetLocalName, GetLERemoteName,
* SetLocalAppearance,
* GetLocalAppearance,
* GetRemoteAppearance,
* Command Options SP/LE: DiscoverSP/LE, RegisterSP/LE, LERSend,
* ConfigureSP/LE, LERRead, Loopback,
* DisplayRawModeData, AutomaticReadMode
*****
SPP*LE>

```

There are three parameters that affect the throughput of data being sent: 1) ConfigParams, 2) BaudRate, 3) UART Buffer Sizes. The UART Buffer sizes are set using the DEFAULT_INPUT_BUFFER_SIZE and the DEFAULT_OUTPUT_BUFFER_SIZE in HCITrans.c before flashing while the ConfigParams and Baud Rate are set using the application. The configparams are actually 3 different parameters, MaximumFrameSize, TransmitBufferSize and ReceiveBufferSize that are set using the SetConfigParams. The current values can be got by using the GetConfigParams.

The section below summarizes the different combinations of ConfigParams, BaudRate and UART Buffer Sizes. For our first set of tests we keep the UART Buffer size at the default value. Using the default config params of 58, 58, 58 respectively for MaximumFrameSize, TransmitBufferSize and ReceiveBufferSize, we use two baudrates of 460800 and 921600. We then increase the Transmit&Receive BufferSizes and test at the same two baud rates. Finally we increase all the config params and test at the same two baud rates. We repeat the same tests after changing the UART Buffer sizes. The results are shown in the table below.

Note: Changing these values may cause the MSP430 to run out of memory. You can increase the amount of memory allocated by changing the BTPS_MEMORY_BUFFER_SIZE. This option can be found in CCS 5 by going to Project->Properties->Advanced Options->Predefined Symbols. Double click on BTPS_MEMORY_BUFFER_SIZE in the pre-defined names to change it. If you are using IAR, Project->Options->C/C++ Compiler->Preprocessor(Press the arrow button if you can't see it initially). We set it from 3600 to 9600 and were able to run all of the changes mentioned here.

```

COM17:9600baud - Tera Term VT
File Edit Setup Control Window Help
SPP*LE>SetBaudRate 921600
HCI_Reconfigure_Driver(921600): Success.
SPP*LE>GetConfigParams
SPP_Get_Configuration_Parameters(): Success
MaximumFrameSize : 58 (0x3A)
TransmitBufferSize : 58 (0x3A)
ReceiveBufferSize : 58 (0x3A)
SPP*LE>SetConfigParams 329 987 2303
SPP_Set_Configuration_Parameters(): Success
MaximumFrameSize : 329 (0x149)
TransmitBufferSize : 987 (0x3DB)
ReceiveBufferSize : 2303 (0x8FF)
SPP*LE>automaticreadmode 1
Current Automatic Read Mode set to: ACTIVE.
SPP*LE>open 1
Server Opened: Server Port 1, Serial Port ID 1.
Server Port Context Stored.
SPP*LE>
SPP Open Indication, ID: 0x0001, Board: 0x00027232591C.

```

```

HCITRANS.c
64 #define DEFAULT_INPUT_BUFFER_SIZE 128
65 #define DEFAULT_OUTPUT_BUFFER_SIZE 64
66 #define XOFF_LIMIT 8
67 #define XON_LIMIT 16
68

```

Data Transfer from Sample Application to MSP430

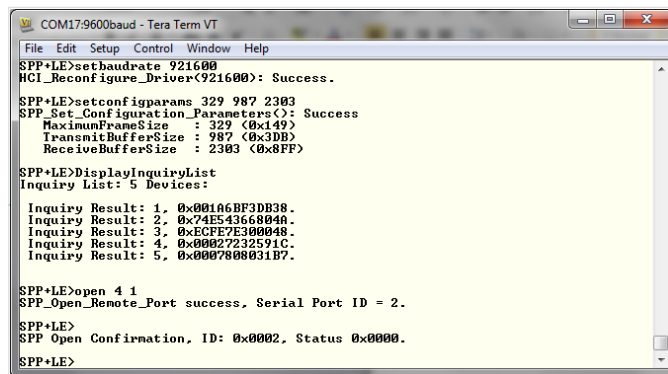
ConfigParams	BaudRate	UART Buffer Sizes	Throughput
58 58 58	460800	128,64	60Kbps
58 58 58	921600	128,64	65Kbps
58 348 348	460800	128,64	231Kbps
58 348 348	921600	128,64	249Kbps
329 987 2303	460800	128,64	335Kbps
329 987 2303	921600	128,64	620Kbps
58 58 58	460800	1042, 384	58Kbps
58 58 58	921600	1042, 384	64Kbps
58 348 348	460800	1042, 384	224Kbps
58 348 348	921600	1042, 384	227Kbps
329 987 2303	460800	1042, 384	337Kbps
329 987 2303	921600	1042, 384	675Kbps

Data Transfer from MSP430 to Sample Application(MSP430 Send)

ConfigParams	BaudRate	UART Buffer Sizes	Throughput
58 58 58	460800	128,64	142Kbps
58 58 58	921600	128,64	126Kbps
58 348 348	460800	128,64	139Kbps
58 348 348	921600	128,64	148Kbps
329 987 2303	460800	128,64	323Kbps
329 987 2303	921600	128,64	522Kbps
58 58 58	460800	1042, 384	135Kbps
58 58 58	921600	1042, 384	140Kbps
58 348 348	460800	1042, 384	133Kbps
58 348 348	921600	1042, 384	134Kbps
329 987 2303	460800	1042, 384	345Kbps
329 987 2303	921600	1042, 384	672Kbps

MSP430 Client

We can repeat the same tests with the two devices switching roles, i.e. the MSP430 as Client and the Sample application as Server.



<p>1. switchcategory:MultiCore=</p> <ul style="list-style-type: none"> For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum For questions related to the BIOS MultiCore SDK 	<p>Keystone=</p> <ul style="list-style-type: none"> For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum 	<p>C2000=For technical support on the C2000</p> <ul style="list-style-type: none"> please post your questions on The C2000 	<p>DaVinci=For technical support on the DaVinci</p> <ul style="list-style-type: none"> Forum. Please post only 	<p>MSP430=For technical support on MSP430</p> <ul style="list-style-type: none"> please post your questions on The MSP430 Forum. 	<p>OMAP35x=For technical support on OMAP please</p> <ul style="list-style-type: none"> post your questions on The OMAP Forum. Please post only 	<p>OMAPL1=For technical support on OMAP please</p> <ul style="list-style-type: none"> post your questions on The OMAP Forum. 	<p>MAVRK=For technical support on MAVRK</p> <ul style="list-style-type: none"> please post your questions on The MAVRK
--	--	---	---	---	---	---	---

(MCSDK), please use the BIOS Forum

Please post only comments related to the article **CC256x MSP430 TI's Bluetooth Stack Basic SPPDemo APP Improving throughput v14** here.

- For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **CC256x MSP430 TI's Bluetooth Stack Basic SPPDemo APP Improving throughput v14** here.

Forum.

Please post only comments about the article

CC256x MSP430 TI's Bluetooth Stack Basic SPPDemo APP Improving throughput v14 here.

comments about the article **CC256x MSP430 TI's Bluetooth Stack Basic SPPDemo APP Improving throughput v14** here.

Please post only comments about the article **CC256x MSP430 TI's Bluetooth Stack Basic SPPDemo APP Improving throughput v14** here.

comments about the article **CC256x MSP430 TI's Bluetooth Stack Basic SPPDemo APP Improving throughput v14** here.

only comments about the article **CC256x MSP430 TI's Bluetooth Stack Basic SPPDemo APP Improving throughput v14** here.

[Toolbox Forum.](#)
Please post only comments about the article **CC256x MSP430 TI's Bluetooth Stack Basic SPPDemo APP Improving throughput v14** here.

B
A
}}
h

Links



[Amplifiers & Linear Audio](#)
[Broadband RF/IF & Digital Radio](#)
[Clocks & Timers](#)
[Data Converters](#)

[DLP & MEMS High-Reliability Interface](#)
[Logic](#)
[Power Management](#)

[Processors](#)

- [ARM Processors](#)
- [Digital Signal Processors \(DSP\)](#)
- [Microcontrollers \(MCU\)](#)
- [OMAP Applications Processors](#)

[Switches & Multiplexers](#)
[Temperature Sensors & Control ICs](#)
[Wireless Connectivity](#)

Retrieved from "https://processors.wiki.ti.com/index.php?title=CC256x_MSP430_TI%27s_Bluetooth_Stack_Basic_SPPDemo_APP_Improving_throughput_v14&oldid=201709"

This page was last edited on 12 June 2015, at 13:17.

Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.