[BGScript]: uart_echo_packet - UART1/Alt1 loopback (local echo) with watermarking and packetization

Jeff Rowberg
posted this on November 8, 2013, 01:45

Share Tweet 0 Like 0

Project Name:	uart_echo_packet

Description:	Demonstrates simple UART RX/TX functionality using watermarking to detect incoming data . Note that this project does not use *any* BLE functionality; it will not advertise or accept remote connections. It is written only to demonstrate basic UART endpoint usage.
	
	Rather than blindly echoing back all received data, which is how the **simpler "uart_echo" project** works, this project behaves in much the same way that BGAPI "packet" mode does (see BLE SDK documentation for detail). Essentially, you must send a <length> byte to the module indicating how many data bytes you are about to send before you actually send that data. This <length> byte is then used to read the correct number of bytes in and only trigger the UART RX command when you have received all of the expected bytes.
	
	For example, if you wanted to send the ASCII string "ABCD", that would be four bytes:
	
	`0x41 0x42 0x43 0x44`
	
	Therefore, you would use **0x04** as the **<length>** byte, followed by the actual data:
	
	`0x04 0x41 0x42 0x43 0x44`
	
	The "0x04" value would be interpreted as the <length> value, and then ignored (not echoed back). The RX watermark threshold will then be set to 4 bytes, meaning no more UART reads will be done until you have at least 4 bytes of data in the buffer. Once this condition has been met, ONLY those 4 bytes will be read, even if there are more than 4 bytes available (to ensure that all data is correctly processed later). Those 4 bytes will be stored as a single block in the "in" byte array, and then echoed back all at once. (Typically, you will do something besides echoing the data back, but this is a proof of concept.)

Instructions:	1. Unzip **uart_echo_packet.zip**. 2. Compile and flash **project112.bgproj** or **project113.bgproj** with BLE Update utility. 3. Connect a 3.3v UART transceiver/adapter directly to the UART1/Alt1 pins (P0_2/3/4/5), or if you are using the DKBLE112 dev kit, plug a DB9 serial cable into the onboard connector and turn on the RS232 switch. 4. Open a terminal application such as **Realterm** and connect to the appropriate COM port at **115200, 8/N/1**. 5. Send data **AS DESCRIBED ABOVE WITH A PREPENDING <length> BYTE**. Each packet (without the length byte) will be echoed back.
Notable Functions:	• **Command**: system_endpoint_rx • **Command**: system_endpoint_tx • **Command**: system_endpoint_set_watermarks • **Event**: system_endpoint_watermark_rx
Safe to Flash:	BLE112, DKBLE112, BLE113, DKBLE113
Unsafe to Flash:	BLED112 (no ability to return to DFU mode)

uart_echo_packet.zip

One person found this useful. Me too!