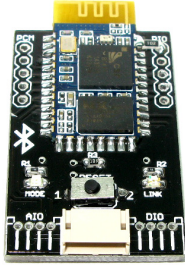


EBLUE *BLUETOOTH*[®] WIRELESS SERIAL BRIDGER



KEY FEATURES

- Based on CSR (Cambridge Silicon Radio) BlueCore4 Chipset; BC417143B-IRN-E4
- *Bluetooth*[®] Spec v2.0 + EDR Compliant
- Enhanced Data Rate (EDR) Compliant with V2.0.E.2 of specification for both 2Mbps and 3Mbps modulation modes
- Class 2
- Low Power Consumption
- 3.3V Operation
- Master/Slave Operation
- Pre-paired
- **Drop-in replacement for wired serial connections**
- Guaranteed 10 meters transmission range
- RoHS Compliant
- Fixed Baud Rate Settings: 115200bps, 8bit Data, 1 Stop Bit, No Parity, No Flow Control

SPECIFICATIONS

Operating Frequency Band	2.4Ghz – 2.48Ghz Unlicensed ISM band
Bluetooth Specification	V2.0 + EDR
Output Power Class	Class 2
Operating Voltage	3.3V
Host Interface	USB 1.1/2.0 or UART
Audio Interface	PCM and Analog Interface
Flash Memory Size	8Mbit
Dimension	25 mm (W) * 46 mm (L)

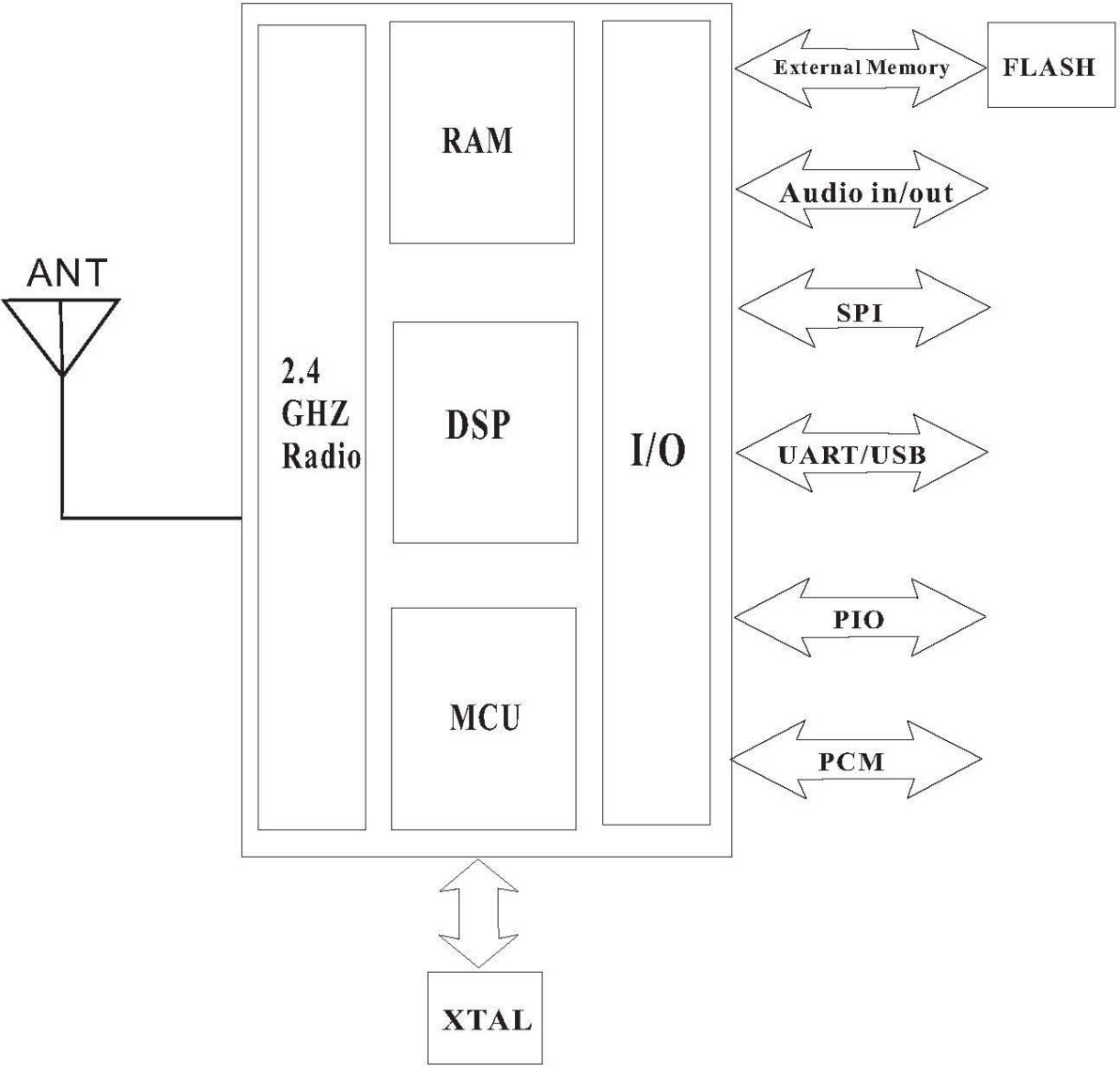
ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings		
Rating	Min	Max
Storage Temperature	-40 °C	+150 °C
Supply Voltage: VBAT	-0.4V	5.6V
Other Terminal Voltages	VSS -0.4V	VDD + 0.4V

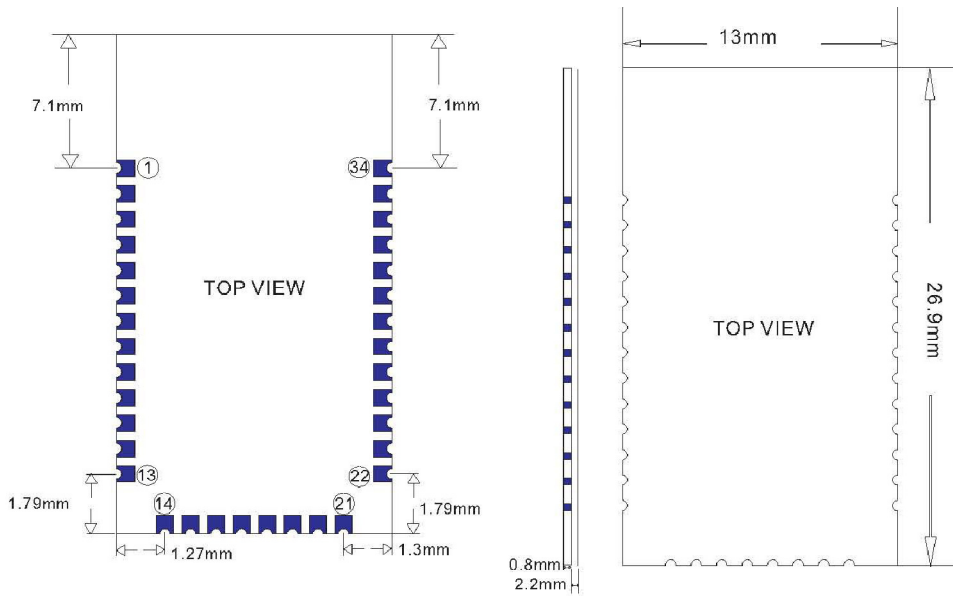
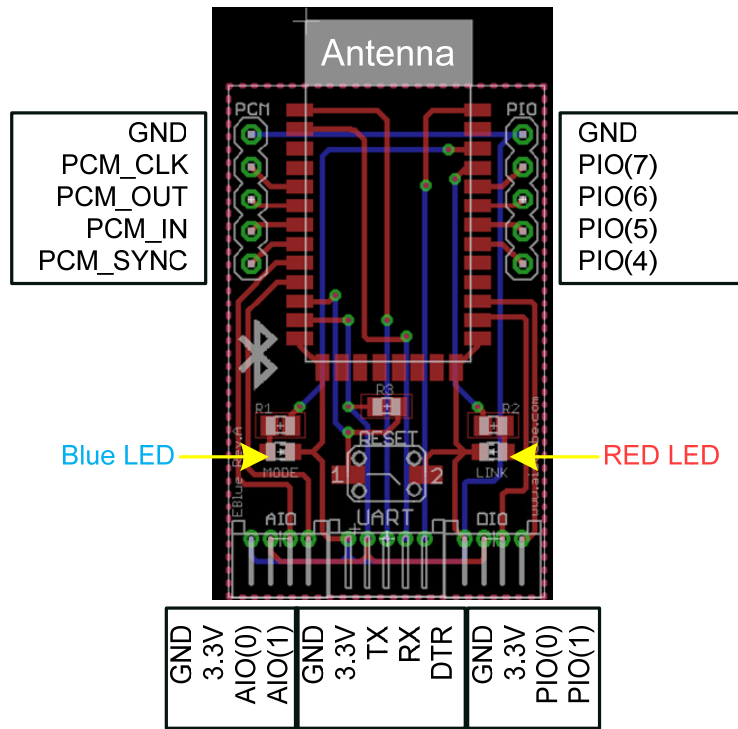
Recommended Operating Conditions		
Operating Conditions	Min	Max
Operating Temperature Range	-40 °C	+150 °C
Guaranteed RF Performance Range	-0.4V	5.6V
Supply Voltage: VBAT	2.2V	4.2V

EBLUE BLUETOOTH[®] CHIPSET BLOCK DIAGRAM

BLOCK DIAGRAM



PINOUT

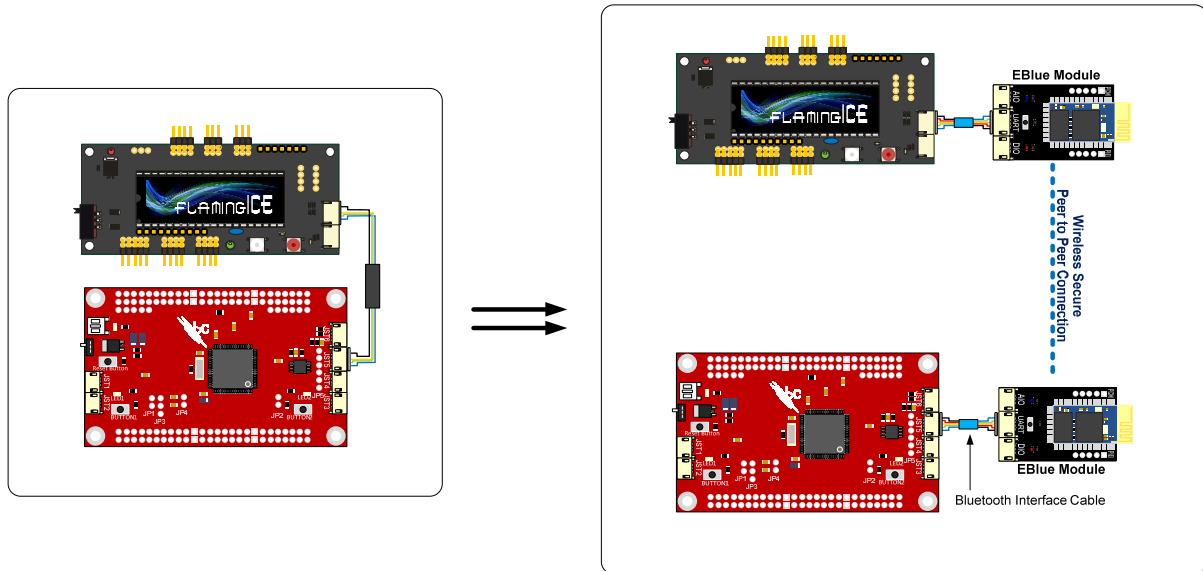


PIN CONFIGURATIONS

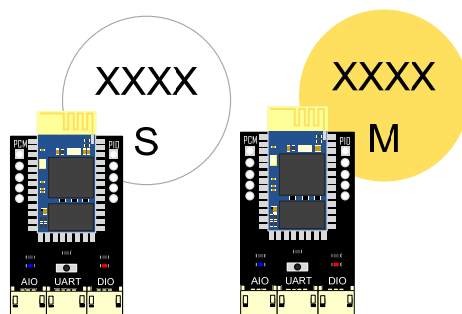
PIN NO.	NAME	TYPE	FUNCTION	RE-MARK
1	UART-TX	CMOS Output	UART Data Output	
2	UART-RX	CMOS Input	UART Data Input	
3	UART-CTS	CMOS Input	UART Clear To Send Active Low	
4	UART-RTS	CMOS Output	UART Request To Send Active Low	
5	PCM-CLK	Bi-directional	Synchronous Data Clock	
6	PCM-OUT	CMOS Output	Synchronous Data Output	
7	PCM-IN	CMOS Input	Synchronous Data Input	
8	PCM-SYNC	Bi-directional	Synchronous Data Sync	
9	AIO(0)	Bi-directional	Programmable Input/Output Line	
10	AIO(1)	Bi-directional	Programmable Input/Output Line	
11	RESETB	CMOS Input	Reset if low. Input debounced so must be low for $\geq 5ms$ to cause a reset	
12	3.3V	POWER	+3.3V Supply	For 3.3V Version
13	GND	GND	Ground	
14	GND	GND	Ground	
15	USB D-	Bi-directional	USB Data Minus	
16	SPI-CSB	CMOS Input	Chip Select For Synchronous Serial Interface	
17	SPI-MOSI	CMOS Input	Serial Peripheral Interface Data Input	
18	SPI-MISO	CMOS Output	Serial Peripheral Interface Data Output	
19	SPI-CLK	CMOS Input	Serial Peripheral Interface Clock	
20	USB D+	Bi-directional	USB Data Plus with selectable internal 1.5K Ω	
21	GND	GND	Ground	
22	GND	GND	Ground	
23	PIO(0)	Bi-directional with programmable strength	Control output for external LNA (if fitted)	
24	PIO(1)	Bi-directional with programmable strength	Control output for external PA (if fitted)	
25	PIO(2)	Bi-directional	Programmable Input/Output Line	
26	PIO(3)	Bi-directional	Programmable Input/Output Line	
27	PIO(4)	Bi-directional with programmable strength	Programmable Input/Output Line or optional BT_Priority/CH_Clk output for co-	
28	PIO(5)	Bi-directional with programmable strength	Programmable Input/Output Line or optional BT_Active output for co-existence	
29	PIO(6)	Bi-directional with programmable strength	Programmable Input/Output Line or optional WLAN_Active/Ch_Data input for co-	
30	PIO(7)	Bi-directional	Programmable Input/Output Line	
31	PIO(8)	Bi-directional	Programmable Input/Output Line	
32	PIO(9)	Bi-directional	Programmable Input/Output Line	
33	PIO(10)	Bi-directional	Programmable Input/Output Line	
34	PIO(11)	Bi-directional	Programmable Input/Output Line	

EBLUE BLUETOOTH[®] USAGE EXPLANATION

USAGE EXPLANATION



These modules are meant to be used as drop-in replacements for wired serial connections. To make your wired connections wireless, have your devices/microcontrollers connect to these modules instead of each other through cables. And that's it! No changes in software necessary.*
*Provided the baud rate settings over at the devices are the same as these modules.



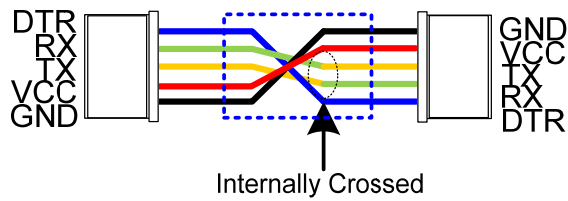
EBlue Modules are shipped in pairs of 1 Master (M) & 1 Slave (S). The Master EBlue will seek out EBlue Slaves to pair with, and attempt hand-shaking for automated pairing process, provided the pin-codes are the same.

Please refer to the circular stickers at the back of the Modules for Pin Codes and Role (Master/Slave) Configurations.

Modules are pre-paired with different 4-digit pin-code/passwords (XXXX) that will be indicated on the sticker. This ensures that in the situation where more than one pair of EBlue Modules is operating at any one-time, no cross-pairing will occur and the devices will stick to its pre-assigned pairing.

EBLUE BLUETOOTH[®] USAGE EXPLANATION

The modules come with an interfacing cable which is internally connected as such;



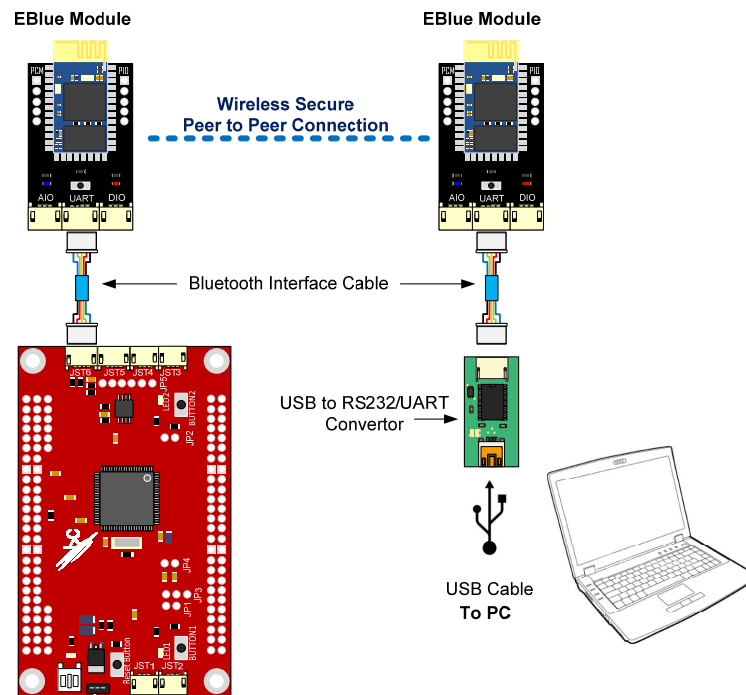
Steps for Use (Illustrated in the Application Examples in the pages that follow)

1. Connect the interfacing cable to the EBlue Module by plugging either side of the cable into the UART Port (In the middle).
2. Connect the other side of the cable to the UART port of your Device/Microcontroller.

Note: The Interfacing cable is designed for ease of use with our BlazingCore and FlamingICE Microcontroller Boards. Should you be using other microcontroller platforms, we highly recommend that you get the "JST 5 Way Cables and Connectors" available separately from our online store to use these modules with your own microcontrollers (eg. Arduino, PIC, AVR, etc)

APPLICATION EXAMPLE 1

Wireless Control of BCore100 Microcontroller Board from the PC



Connecting to the PC:

Using the same USB to RS232 Converter that is used to download programs into the BCore100, remove the JST cable with the black tube and use the Bluetooth Interface Cable with the Module instead.

Open up hyperterminal or similar, configure the Baud Rate settings (using the COM Port Number that was assigned to your Download Key) and open up the COM Port once you're done. You may now begin to transmit and receive data.

Connecting to the BCore100 Microcontroller Board/Device:

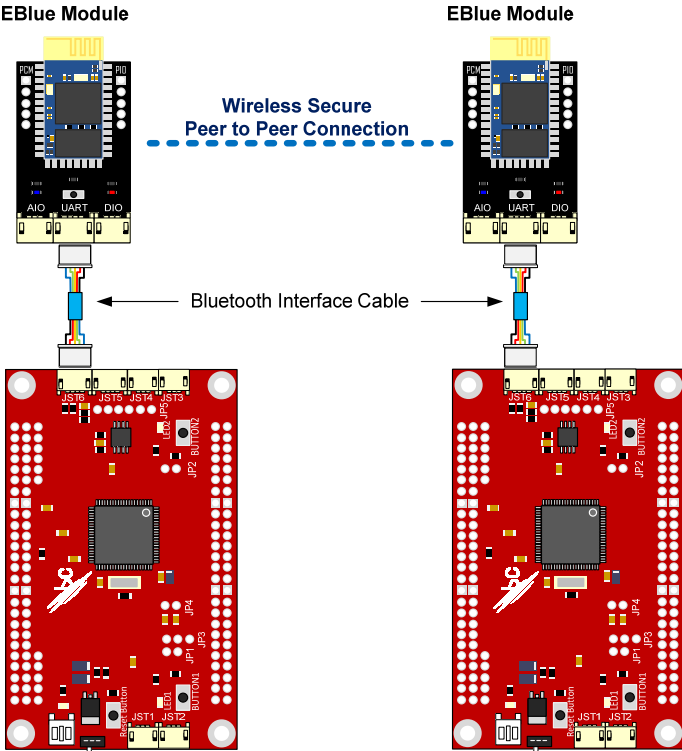
Connect the EBlue Module to either COMM1/2 using the Bluetooth Interface Cable.

Monitor the respective port for Data transmitted by the PC side, and use that same port to transmit any data for the EBlue Module to send over to the PC.

If you already have a project that is working with the JST Serial Cable (Black Tube), no code changes are necessary. All you need to do is to replace the Cable with the EBlue Modules.

APPLICATION EXAMPLE 2

Wireless communication (exchange of Info) between 2 BCore100 Microcontrollers



Connection Diagram

LATEST DOCUMENTATION

All of our documentations are constantly updated to provide accurate and/or new information that we feel would help you with developing with our products.

The latest documentation may be obtained from our website: <http://www.aiscube.com/main/downloads.html>

HOW YOU CAN HELP

You can help us to improve our documentations by emailing to us or posting a thread in our forum, reporting any mistakes/typos or errata that you might spot while reading our documentation.

Email: TechSupport@aiscube.com

Forum: <http://forum.aiscube.com/index.php>

DISCLAIMER

All information in this documentation is provided 'as-is' without any warranty of any kind.

The products produced by AIS Cube are meant for rapid prototyping and experimental usage; they are not intended nor designed for implementation in environments that constitute high risk activities.

AIS Cube shall assume no responsibility or liability for any indirect, specific, incidental or consequential damages arising out of the use of this documentation or product.

COPYRIGHT© 2009 - 2010 AIS CUBE. ALL RIGHTS RESERVED.

ALL PRODUCT AND CORPORATE NAMES APPEARING IN THIS DOCUMENTATION MAY OR MAY NOT BE REGISTERED TRADEMARKS OR COPYRIGHTS OF THEIR RESPECTIVE COMPANIES. AND ARE ONLY USED FOR IDENTIFICATION OR EXPLANATION FOR THE OWNER'S BENEFIT. WITH NO INTENT TO INFRINGE.

THE BLAZINGCORE(BCORE) AND BCORE IDE ARE TRADEMARKS OF AIS CUBE IN SINGAPORE AND/OR OTHER COUNTRIES. ALL IMAGES DEPICTING THE BLAZINGCORE OR ANY PART OF IT IS COPYRIGHTED.

ALL OTHER TRADEMARKS OR REGISTERED TRADEMARKS ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS.