gridconnect®

GRIDBluFly

User Guide

Revision C May 1, 2023

Copyright and Trademark

Grid Connect Inc.

1630 W. Diehl Rd.
Naperville, IL 60563, USA
Phone: 630.245.1445

Technical Support

Phone: 630.245.1445
Fax: 630.245.1717
On-line: www.gridconnect.com

Disclaimer

Operation of this equipment in a residential area is likely to cause interference in which case the user, at his or her own expense, will be required to take whatever measures may be required to correct the interference.

> ***Attention: This product has been designed to comply with the limits for a Class B digital device pursuant to Part 15 of FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with this guide, may cause harmful interference to radio communications.***

Changes or modifications to this device not explicitly approved by Grid Connect will void the user's authority to operate this device.

The information in this guide may change without notice. The manufacturer assumes no responsibility for any errors that may appear in this guide.

Table of Contents

OVERVIEW

Introduction

The GRIDBluFly or also referred to in this guide as just BluFly is a Class 1 Bluetooth® radio modem with 2.4GHz RF PCB trace antenna. The modules are Bluetooth version 5.0 compliant (BR + EDR + Bluetooth® LE). Two BluFly devices can be configured and paired to make a wireless RS232 cable replacement or extender. A PC with Bluetooth capability can also pair with a single BluFly creating a virtual COM port for wireless serial communication.

The serial devices can be configured, commanded, and controlled with simple ASCII strings through the BluFly's hardware serial UART or over a remote Bluetooth RF connection. The BluFly has an RS232 DB9 Male or Female connector. The BluFly is a Class 1 Bluetooth device with mid-range power transceiver (300 feet). Actual range may vary due to environment or type of client device used to connect to BluFly.

Bluetooth wireless technology is a short-range radio technology. Bluetooth wireless technology makes it possible to transmit signals over short distances between computers and other devices and thereby simplify communication and synchronization between devices.

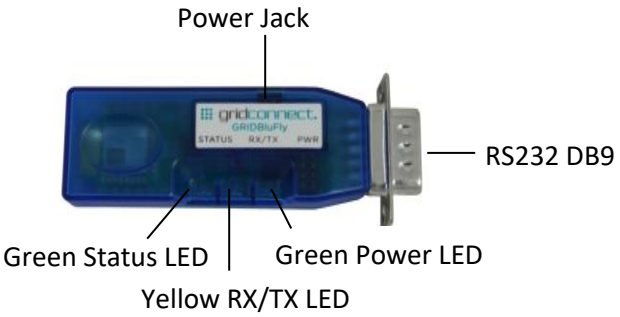
GRIDBluFly Features

- High speed RS232, with standard baud rates from 1200 to 921,600. Baud Rates other than 9600 or 115200 must be selected in command mode.
- Parity Even, Odd, or None, 7/8 Data Bits, 1 Stop Bit.
- Hardware flow control via RTS, CTS (local serial connection only).
- Modem control output (DTR) tied to flow control output (RTS) optional.
- Settings can be easily changed via local serial port connection.
- Mid-range Power (Class 1, 12dBm TX, 300', 90m) Bluetooth™ radio
- Serial Port Profile (SPP) used for direct connect to Classic BT clients.
- Custom GATT profiles for BLE clients (peripheral role only).
- Green (connection), Yellow (TX/RX) and Green (Power) LEDs.
- Low power operation requires only 5VDC at 40ma when connected.
- Powered by external 5VDC wall adapter. Input range is 5-16VDC. Can be powered through the DB9 connector.
- Smallest possible form factor.
- DB9 Male or DB9 Female connector standard.

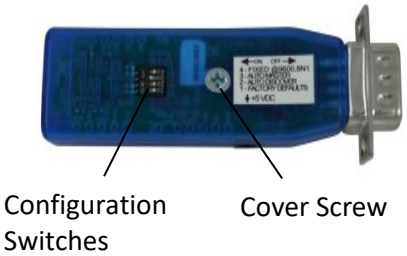
Hardware Description

The following pictures show the top and bottom views of the GRIDBluFly. Note that the standard DB9 Male connector is shown. An DB9 Female version is also available.

BluFly Top View



BluFly Bottom View



RS232 DB9 Pin Assignments

The straight DB9 pin numbers and descriptions (default) are shown in the following table. Please note that some pins have no connection (NC). See the NULL MODEM JUMPERS sections for all wiring options.

Pin	DTE Description (DB9 Male)	DCE Description (Female)
1	NC	NC
2	RXD (in)	TXD (out)
3	TXD (out)	RXD (in)
4	DTR (out) [option]	NC
5	GND	GND
6	NC	DSR (out) [option]
7	RTS (out)	CTS (in)
8	CTS (in)	RTS (out)
9	External Power 5-16 VDC	External Power 5-16 VDC

LEDs

The GRIDBluFly comes equipped with three LEDs. The green Status LED provides Bluetooth Device status, while the Yellow LED shows activity on the transmit/receive lines.

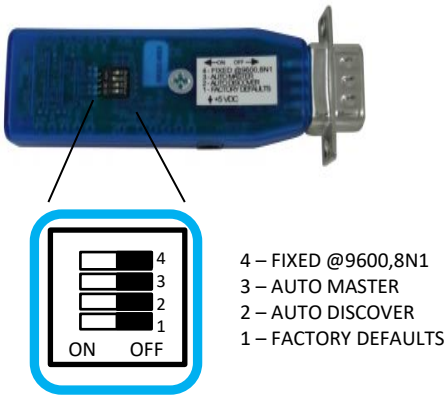
The GRIDBluFly Device status is indicated by the green Status LED as shown below.

GRIDBluFly Device Status	Status LED Blink Rate
Configuring/Command Mode	Fast, 10 times per second
Boot up, Remote Configurable, Discoverable, Idle	2 times per second
Discoverable, Idle	1 time per second
Connected	On Solid

The Yellow LED shows the physical state of the TX and RX data pins, pulse stretched for improved visibility. The Yellow LED blinks when data is **transmitted** or **received** on pins 2 and 3 of the serial port.

There is also a green Power LED that is always on solid with power applied.

GRIDBluFly Configuration Switches



The Configuration switches give you an easy method of setting the operating mode of the BluFly. Changes to the Configuration switches take effect on the next power on.

1- **FACTORY DEFAULTS**- Set this switch **ON**, power up the unit, and toggle the switch from ON to OFF 2 times to return the unit to factory settings. The Status LED will flash when the reset is successful.
2-**AUTO DISCOVER MODE** – In Slave mode (AUTO MASTER OFF), switching AUTO DISCOVER ON will set a special class of device which is used by a remote BluFly Master to auto connect (see AUTO MASTER MODE).
3- **AUTO MASTER MODE**- When switched ON the BluFly will act as master and auto-connect to a stored remote address. You must first set the BluFly address of the device to connect to manually using the SR command or have the address set automatically by also switching AUTO DISCOVER ON to have the BluFly discover another BluFly slave with just its AUTO DISCOVER ON. See the Quick Start for further information.
4- **FIXED @9600,8N1** – Set switch 4 to ON to force the BluFly to use the serial settings 9600 baud, 8 data bits, no parity and 1 stop bit on the next power on. This is useful for entering command mode when the BluFly configuration is unknown. When switched OFF the BluFly will operate with the configured serial settings on the

next initialization. The factory default baud rate is 115200. You can configure the baud rate and other serial settings in command mode using the SU command.

Power

The BluFly's Power Jack accepts 5VDC – 16VDC. The center pin of the power jack is for the input power (typically +5V), while the outer cylinder should be connected to ground. The requirements can be met with the supplied 5VDC, 300ma wall plug adapters.
Barrel connector: Outer diameter: 3.50mm, inner diameter: 1.35mm, length: 8 - 8.25mm

The BluFly can be powered alternatively through the DB9 connector on Pin 9 with ground on Pin 5.

Note: When using an external power supply, Pin 9 on the DB9 connector is connected to the external power supply. You can remove the power to the pin by removing a small resistor. Remove R11 for units with DB9 Male or remove R12 for units with DB9 Female.

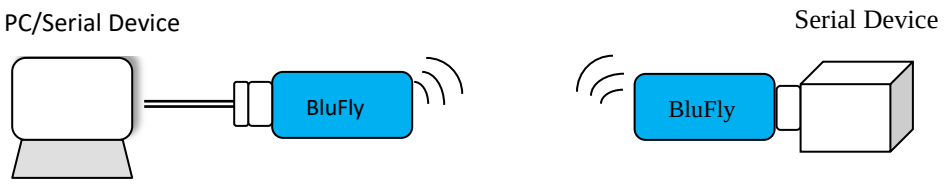
When actively transmitting or receiving data, the BluFly has a power draw of 30-50ma @5V, depending on the data rate.

QUICK START

Follow these instructions to get your device(s) up and running fast.

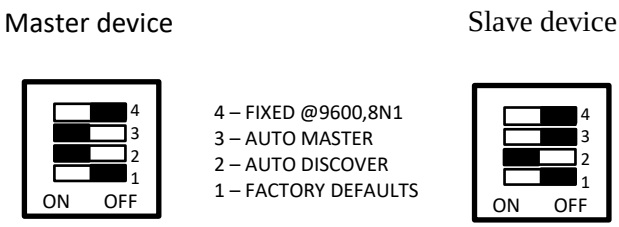
Pairing Two BluFly Serial Devices as a Cable Replacement

A pair of BluFly devices can be used as a serial cable replacement/extender. The serial data then travels over the wireless Bluetooth connection. The BluFly works best for applications where only RS232 3-wire cabling is required (TX, RX and GND). Note that only the serial data is transported over the connection and not the state of all the RS232 control signals.



Pairing Using Configuration Switches

The two BluFly devices first need to be paired so they will transfer serial data between them. To do this, first select one of the devices as the master (client) and the other as the slave (peripheral). Set the configuration switches as shown below.



With the configuration switches set, power on the two devices. The green Status LED will begin by blinking. The Master device will automatically discover the Slave device and store its address. The Master will then establish a connection to the slave and the green Status LED of both devices will be On solid.

Once the two devices have paired change option switch 2 (Auto Discover) to OFF for both devices. This will prevent the devices from inadvertently pairing with other devices on the next power on.

Configuration switch 4 can be used to select the baud rate. When the switch is ON it forces the baud rate to 9600. When the switch is OFF the stored configured baud rate will be used. The stored factory default serial settings are 115200 baud, 8 data bits, no parity and 1 stop bit. If your application requires different serial settings then you must use Command mode to configure the BluFly's baud rate.

Pairing Using Serial Command Mode

The BluFly Bluetooth adapter operates in two modes: data mode (default) and command mode. While in data mode, the module is essentially a data pipe passing serial data between the DB9 and the Bluetooth connection. Sending the BluFly a string of '**$$$**' will put it in Command mode and the BluFly will respond with '**CMD**'. While in command mode the BluFly will rapidly blink the Status LED. The BluFly will then accept ASCII commands to configure BluFly settings or perform other operations. Appropriate commands have been provided to make the module perform the two core actions of a Bluetooth device, which is make/break connections and Inquiry.

Pairing using Command mode requires some knowledge of the commands supported by the BluFly. The following commands give an example of how to Discover, pair and connect to another BluFly.

Example of a Master Discovery/Connection Sequence

From power up and no connection:

1) Perform an Inquiry to obtain remote ***BT_Address*** (unless it is already known).

Send : $$$ **<cr>** // Places Radio in Command Mode
Reply:CMD<cr>

Send : I,30**<cr>** // Looks for Bluetooth devices
Reply: <cr>**Inquiry, COD=0<cr>**001D4B060005,GRIDBluFly-0005,1F00**<cr>**Found 1<cr>Inquiry Done**<cr>**

2) Store the remote address just found.

Send : SR, 001D4B060005**<cr>**
Reply:<cr>AOK

3) Connect.

Send : C **<cr>** // Places Radio in Connect
Reply:TRYING**<cr>**

BluFly will attempt connection to remote slave.
Reply:CONNECT**<cr>** // this will be displayed once connection is made, if <text> string is defined in the stored parameters.

4) Send /Receive data.

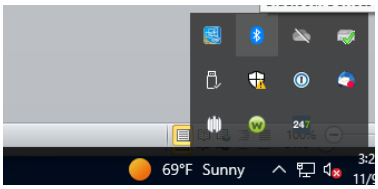
Creating a Bluetooth connection to the BluFly from your PC

A PC with Bluetooth capability can pair with the BluFly creating a virtual COM port for wireless serial communication. The PC is assumed to be the master and the BluFly is the slave. You connect to the BluFly using your computer's Bluetooth device manager, which varies depending on the operating system. Regardless of the operating system, the process is the same: discovery, pairing, and connecting.

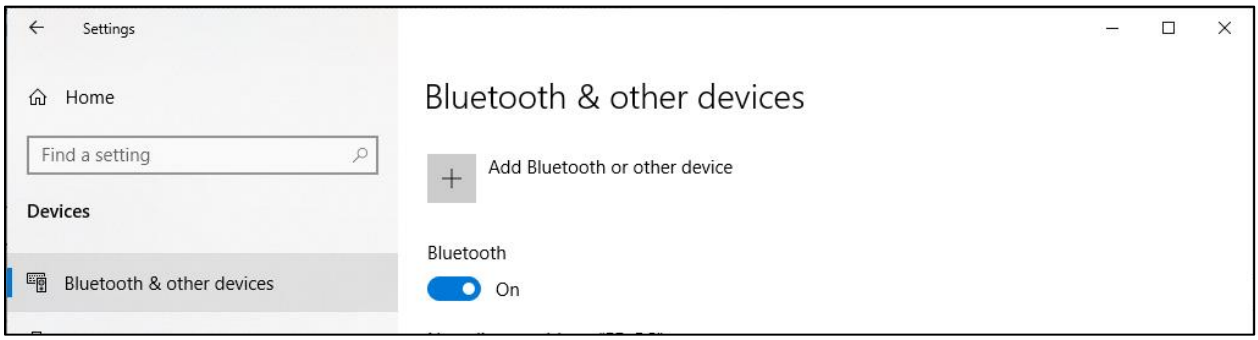
Discovery

When you turn on the Bluetooth adapter, the green Status LED should blink and the adapter should be

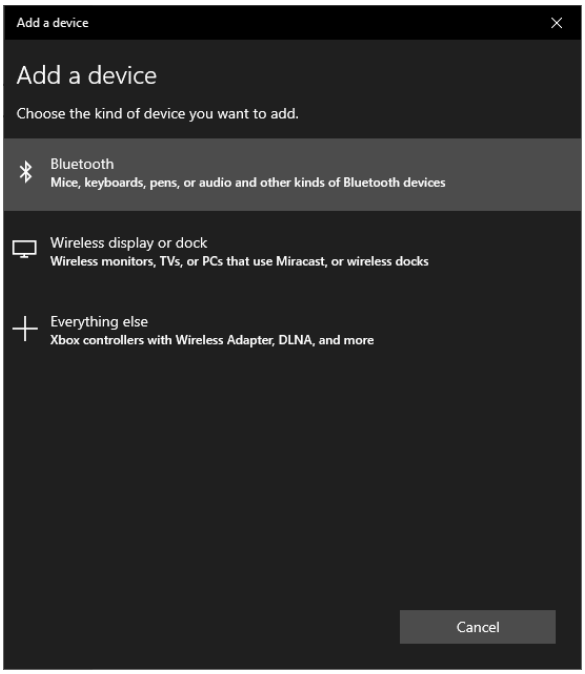
discoverable. Open your PC's Bluetooth device manager and choose to add a new device. The Bluetooth device manager's icon is normally located in the bottom right corner of your screen in the Windows System Tray.

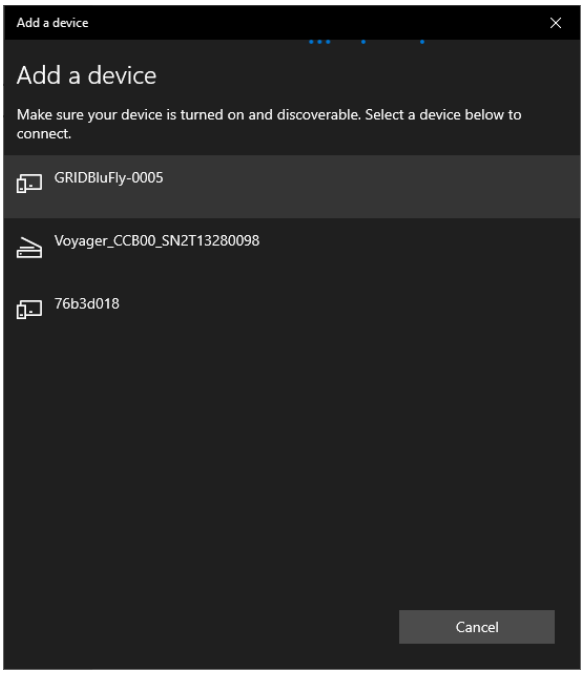


Or click the Windows icon -> Settings -> Devices.



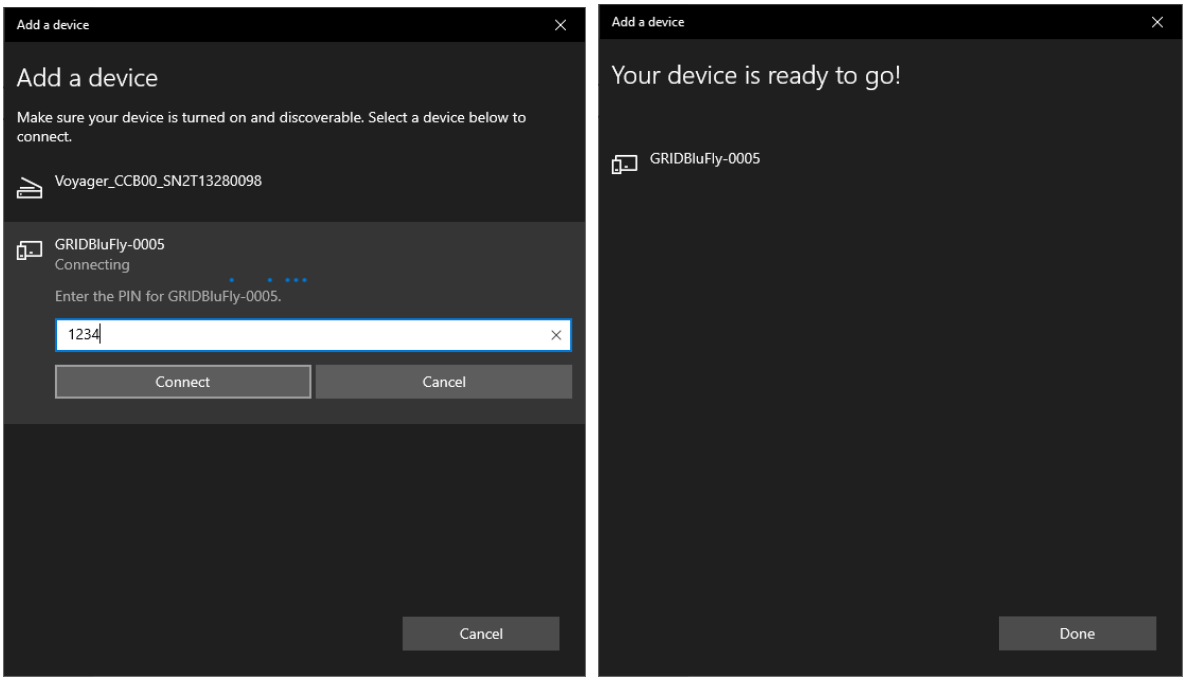
After choosing "Add Bluetooth or other device" the Bluetooth device manager displays a list of discoverable Bluetooth devices. The BluFly adapter displays as GRIDBluFly-XXXX where XXXX is the last 4 digits of the adapter's MAC address.





Pairing

To pair with the adapter, click the BluFly's name (i.e., BluFly -*XXXX*) in the list. You may be asked to enter the PIN for the BluFly then enter the default pin code, 1234, and click Connect. When the Bluetooth device manager completes pairing, it issues a message that the Bluetooth device is ready to go!



Go to the Windows Device Manager and view the available Ports. You should now have new COM ports named "Standard Serial over Bluetooth link (COMx)". In most cases two COM ports will be created, one for outgoing connections and one for incoming connections. Only the outgoing COM port (normally the higher numbered port) should be used for connecting to the BluFly.

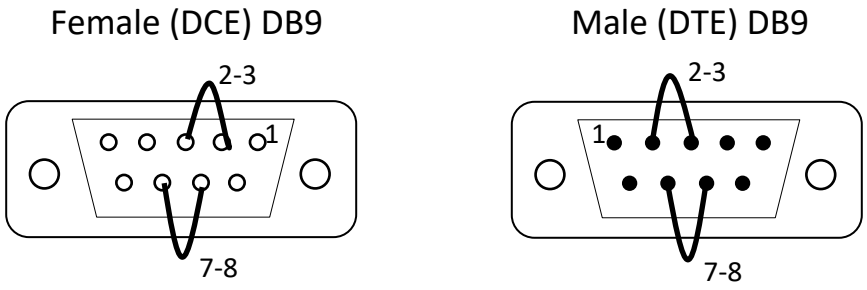
Connecting

To establish a Bluetooth connection, open the outgoing COM port associated with the BluFly from your application or a terminal emulator. When the COM port is open the BluFly's Status LED changes from blinking to solid on. The device remains connected until you close the COM port or turn off the BluFly adapter.

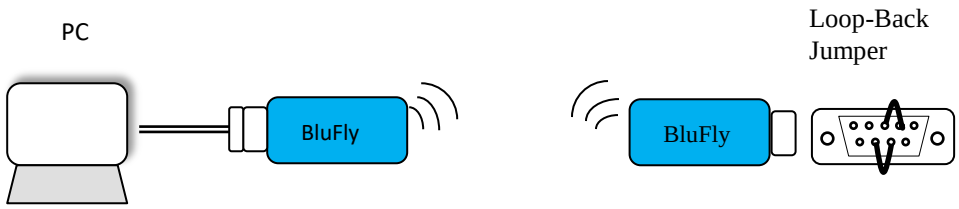
Link Test

Once you have a Bluetooth connection you can verify communications by making a Loop-back jumper on the remote BluFly as shown in the drawing below. The loop-back jumper connects the Transmit and Receive lines together so that any transmitted data will be looped back to receive. It's easiest to make a loop-back by using a DB9 Female connector and adding a wire between pins 2 and 3. Even a bent paper clip will do. If you have configured the BluFly for hardware flow control with RTS and CTS signals, also add a wire from pin 7 to 8.

DB9 LOOP-BACK JUMPER



With the loop-back jumper in place on the remote BluFly you can use the terminal emulator to send characters over the Bluetooth connection and the characters should be echoed back. Each character typed will have traveled round trip over the Bluetooth connection.



CONFIGURATION

This section provides the setup and operation of the BluFly modules and introduces the *command mode* protocol used to control and configure BluFly Serial Modules.

The BluFly Bluetooth adapter operates in two modes: data mode (default) and command mode. While in data mode, the device transparently transfers serial data between the RS232 DB9 and the Bluetooth connection.

Upon power up the device will be in data mode. To enter command mode, send the characters “$$$” through the serial port or from the remote Bluetooth connection.

There are two ways to configure the BluFly Bluetooth module:

1. Local configuration using your computer’s serial port
2. Remote configuration using a Bluetooth connection

You need a terminal emulator to complete the setup. We suggest using the Tera Term terminal emulator.

Local Configuration Using a Serial Port

Connect the BluFly Bluetooth adapter to your computer’s serial port. This may require a null-modem cable/adapter (i.e. with DB9 pins 2 and 3 swapped). If your computer does not have a serial port, you can use

a serial RS232 to USB adapter such as the GC-ATC-810 to connect the BluFly to your computer.

With the BluFly connected and powered on, run a terminal emulator and open the COM port to which the serial interface or serial USB is connected.

The terminal emulator's communication settings should be set to match the default serial port settings of the BluFly: 115,200 baud, 8 data bits, no parity, 1 stop bit, no flow control.
Note: you can also set the BluFly configuration switch 4 to ON to force the BluFly to use 9600 baud instead.

In the terminal emulator type "**$$$**" to send these characters to the BluFly. The BluFly will not echo the typed characters so they will not appear on the screen. After sending "**$$$**" you should see "**CMD**" returned to you and the BluFly Status LED will blink rapidly. This will verify that your cable and serial settings are correct. Valid commands will return an "**AOK**" response or text information, and invalid ones will return "**ERR** ". Commands that are not recognized will return a "**?**".

Another quick check to see if you are in command mode is to type the "D" and "E" commands after entering command mode. This will show the parameters, such as the *Bluetooth* Name, Class of Device and Serial settings. See the "COMMAND REFERENCE" section for the full list of possible commands.

To exit command mode, type "**---**"<cr>. (three minus signs). The device will respond with "**END**".

Local configuration works at any time when the device does NOT have a Bluetooth connection. Once a connection is made, you can only enter command mode if the config timer has not expired. If the device is in configuration mode and a connection occurs, the device will exit configuration mode, and data will pass back and forth from the remote device.

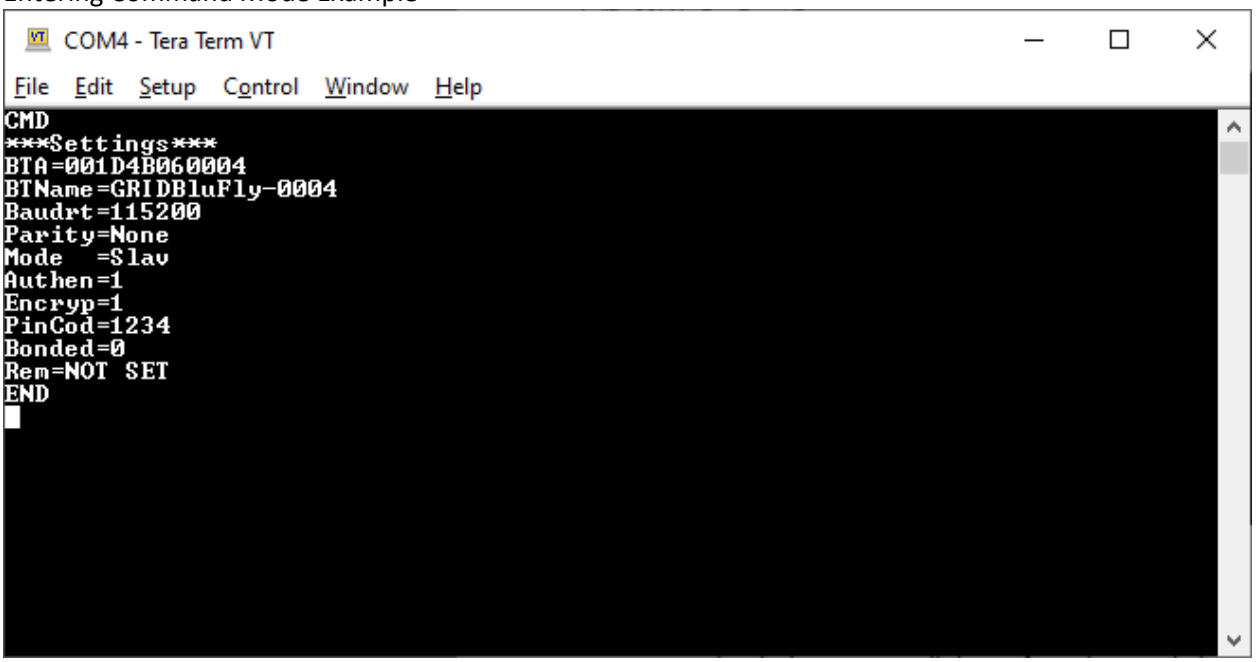
Remote Configuration Using Bluetooth

It is often useful to be able to perform configuration remotely over a Bluetooth connection. Before performing remote configuration using Bluetooth, you must first pair the Bluetooth device with your computer. See the QUICK START section on "Creating a Bluetooth connection to the BluFly from your PC".

Power on the BluFly to restart it's config timer. Run the terminal emulator and connect to the COM port created by the Bluetooth connection. Then perform the same steps as you would for local configuration above. Send "**$$$**" to bring the BluFly into Command mode. The device will respond with "CMD". When finished configuring, be sure to either reset the device, or Send the "---" command, which will exit configuration mode and allow data to pass normally.

NOTE: You can only configure remotely if the config timer (default 60 seconds) has not expired. This is indicated by the Status LED still blinking twice a second. Also, if the device is in Auto Master mode 3, you will NOT be able to enter command mode when connected over Bluetooth.

Entering Command Mode Example



Minimum Configuration

At minimum the BluFly must normally be configured to match the serial parameters of whatever serial device you need to communicate with over Bluetooth. Only if you are lucky then your serial device requires 9600 baud or 115200 baud which can be selected with the configuration switches. This works as long as the other parameters are 8 data bits and No parity. The following example shows the most common commands needed to configure the BluFly to match your serial device.

Example Commands for Setting Serial Parameters

Send: $$$ // Enter Command mode
Reply: CMD

Send: SU,19 // Set the baud rate to 19200 (replaces the 115200 factory default baud rate)
Reply: AOK

Send: SL,E // Set the parity
Reply: AOK

Send: S7,1 // Only for setting 7 data bits
Reply: AOK

Send: R,1 // Reboot with new settings
Reply: Reboot!

MODES OF OPERATION

Classic Bluetooth operational modes can be set using the **SM** command.

Slave Mode (SM,0)

This is the default mode, whereby other Bluetooth devices can discover and connect to the device. Outbound connections can also be made in this mode.

Master Mode (SM,1)

This mode is useful when the BluFly wants to initiate connections (not receive them). In this mode the BluFly will NOT be discoverable or connectable.

Trigger Master Mode (SM,2)

In this mode, the device will automatically connect to the pre-configured remote slave address when a character or characters are received on the local UART. Connection will remain open until a configurable idle timer (1 to 255 seconds) expires with no data being received, or a configurable BREAK character is seen.

Auto-Connect Master Mode (SM,3)

This mode can be set by command, or by setting the external dip switch 3 before power up of the BluFly. If this mode is set, the device will initiate a connection to the pre-stored remote address immediately upon power up. If no address is stored, an inquiry process will be attempted and the first device found that matches the COD will be stored. In this mode, data is passed without being interpreted by the BluFly (FAST Data), hence the connection cannot be broken via command. If disconnect occurs, the device will attempt to re-connect until successful.

BLE Server

In addition to the classic Bluetooth operational modes the BluFly makes use of a few custom GATT profiles for implementing a BLE interface to the serial RS232. The BluFly works in the peripheral/server role only and a BLE client device must be used to make the connection.

COMMAND REFERENCE

This section describes the *command mode* protocol used to control and configure BluFly Serial Modules. The protocol is similar to the industry standard Hayes AT protocol used in telephone modems due to the fact that both types of devices are connection oriented. Appropriate commands have been provided to make the module perform the two core actions of a Bluetooth device, which is make/break connections and Inquiry. Additional commands are also provided to perform ancillary functions.

All commands are either one or two characters and can be upper or lower case. Arguments for commands are delimited by a comma. Commands take decimal input except where noted. Text data, such as Bluetooth name, and pin code, are case sensitive. Commands fall into five general categories:

SET COMMANDS	store information to flash, changes take effect after power cycle or reboot
GET COMMANDS	retrieve and display the stored information
CHANGE COMMANDS	temporarily change the value of serial baudrate, parity, etc.
ACTION COMMANDS	perform action such as inquiry, connect, etc.

SET Commands

All set commands do not take effect until after the module has been rebooted.

S7,<1,0>

7 bit data mode. 1 to enable, 0 to disable. (setting can be seen with the "d" command).
Default: 0

SA,<value>

Authentication. Sets the type of authentication used when a remote device attempts to connect, where <value> is one of the values shown in the following table.

SET AUTHENTICATION VALUES

Value	Description
0-3	SSP "just works" mode for simple pairing without MITM protection for BLE and Classic Bluetooth.
4	Pin code. Legacy pin code mode authentication (Classic Bluetooth only), which requires the host device to enter a pin code that matches the stored pin code.

SBLEAT,<num>

Set BLE advertising timer. This sets how long to allow remote BLE discovery after power up. Set to 0 to keep BLE advertising forever.
Default: 60
Example : *"SBLEAT,70"* // Set BluFly BLE advertising time to 70 secs.

SC,<hex word>

This command sets the service class field in the class of device (COD). The service class consists of the most significant 11 bits in the COD. This command sets the MSW to create the 24-bit device class number. The inquiring device interprets the service class to determine the service. A complete listing of available Bluetooth service classes is referenced on the Bluetooth SIG web site.
Default: 0000
Example : *"SC,0002"*

SD,<hex word>

This command sets the class of device (COD) LSW. The COD is a 24-bit number that is made up of the device class with major 8 bit and minor in a 16-bit word. This command is used with the service class command.
Default: 1F00
Example : *"SD,8040"*

To set the Class of Device (COD) to 0x1F0123 use the commands
SC,001F
SD,0123

SF,1

Set Factory Defaults.

SG,<0,1,2>

Set the Bluetooth to serial interface mode (0 = dual mode, 1 = BLE/GATT server, 2 = BT Classic/SPP).
Default: 0
Example : *"SG,2"* sets the mode to Bluetooth Classic SPP profile only.

SHWF,<0,1>

Set UART Hardware Flow control (0 = disabled, 1 = enabled).
Default: 0
Example : *"SHWF,1"* sets the HW flow control to enabled.

SL,<E,O,N>

Set UART parity. Can be any of, Even, Odd, or None. Only the first character is needed and must be capital.
Default: N
Example : *"SL,E"* sets the parity to Even.

SM,<5,4,3,2,1,0>

Mode (0=slave, 1=master,2=trigger, 3=auto, 4=DTR)
Default: 0
Example : *"SM,1"* sets the mode to Master

SN,<name>

Name of the device, 20 characters maximum.
Default: BluFly-xxxx
Example: *"SN,MyDevice"*

S-,<name>

Serialized Friendly Name of the device, 15 characters maximum. This command will automatically append the last 2 bytes of the BT MAC address to the name. Useful for generating a custom name with unique numbering.
Example: *S-,MyDevice* will set the name to "MyDevice-ABCD"

SO,<text>

Extended Status String, 8 character maximum. Setting this string to from 1 to 8 characters will enable status messages to be sent to the local serial port. Two status messages are sent:

- when a Bluetooth connection is established, the string "<text>CONNECT" will be sent.
- When disconnecting, the string <text>DISCONNECT will be sent.

This parameter is useful, for example, when connected to a printer, the printer can examine an escape sequence, if the <text> is set to ESC%, the printer can parse the ESC%CONNECT and ESC%DISCONNECT messages without interfering with normal print jobs. In Trigger or Master modes, the first character of this string is used as the BREAK connection character.
Default: NULL
Example: *SO,ESC%*
SO,*<space>* // Disables the extended status string

SP,<text>

This command sets the security pin code, where *<string>* is up to 16 alphanumeric characters. Each time the device pairs successfully, it saves the Bluetooth address. The device can store up to four addresses on a first in first out basis. Using this command also erases all stored pairings. You can use the same value that is already set.
You cannot erase the pin code, however, you can overwrite the default pin code.
Default: 1234
Example: *SP,secretcode* sets pin code to "secretcode"

SQ,<num>

Special configuration commands, num is a decimal number with the following interpretation.

num	Description
0	Disable all special commands
128	This option causes the device to reboot after disconnect.
256	Set 2 stop bit mode on the UART.

Default: 0
Example: `SQ,128` // Reboot after disconnect

SR,<*hex value*>

This command stores the remote address, where *<hex value>* is 12 hexadecimal digits (6 bytes) with no spaces or characters between digits. Additionally, this command takes two special characters for the address parameter:

• SR,Z erases any stored addresses.
• SR,I writes the last address observed using the inquiry command. This command can be helpful when you only have one other device in range.
Default: Not Set

ST,<number>

Configuration timer, number of seconds (range= 0 to 255 decimal) to allow remote configuration over Bluetooth after power up in Slave Mode. In all Master modes, the remote configuration timer is set to 0 (no remote configuration). In Trigger Master Mode, the configuration timer is used as an idle timer to break the connection after time expires with no characters being received.
Default: 60
Examples:
ST,0 disables remote configuration
ST,60 sets remote configuration to 60 seconds (default value)
ST,255 enables remote configuration forever

SU,<rate> Baudrate, {1200, 2400, 4800, 9600, 19.2, 28.8, 38.4, 57.6, 115K, 230K, 460K, 921K }, only the first 2 characters are needed.
Default: 115K
Example: *SU,57* sets the baudrate to 57600 baud.

SY,<*hex value*> This command sets the module's transmit power for BLE, where <*hex value*> represents the desired power setting.

Hex	Power (dBM)
0008	8
0004	4
0000	0
FFFC	-4
FFF8	-8
FFF4	-12

Example: SY,0008 // Set the power to 8 dBM

SZ,<*value*> You use this command to specify non-standard raw baud rates, where <*value*> is a decimal number for the desired baud rate. This setting takes effect after rebooting.
Example: SZ,9000 // Set the baud rate to 9,000

**S$,<*char*>** This command sets the configuration detect character string, where <*char*> is a single character. This setting allows you to change the default string to go into command mode ($$$) to some other character string. Restoring the factory defaults returns the device to using $$$.
Default: $
Example S$,# // Set ### as string to go into command mode

GET Commands

The get commands retrieve and display the device's stored information. These commands do not have a keyword or character and do not take any parameters, except as noted.

D Display basic settings. Address, Name, Uart Settings, Security, Pin code, Bonding, Remote Address.

E Display extended settings: device class, configuration Timer.

O Display other settings. Configuration character, I/O port values, debug mode.

G<X> Display stored settings for command X. These commands correspond to the SET commands above.
Example: *G7* will return 1 or 0 depending on the value of 7 bit data mode.

In addition to the above, there are a few other useful commands available**.**

GB	Returns the Bluetooth Address of the device.
GF	Returns the Bluetooth Address of the currently connected device.
GK	Returns the current connection status: 1=connected, 0 = not connected.
V	Return the firmware version
&	Returns the value of the DIP switches.

Action Commands

Action commands perform actions such as inquiries, connecting, and entering/exiting command mode.

$$$	Enter command mode Characters are passed as data until this exact sequence is seen. If any bytes are seen before or after the $$$ characters in a 1 second window, command mode will not be entered and these bytes will be passed on to other side. **NOTE:** The device will only enter command mode over bluetooth if it is within the configuration timer window (60 seconds from power up by default). In master mode the configuration timer is set to zero. The character string to enter command mode is configurable using the *S$* command
---	This command causes the device to exit command mode, displaying END.
+	This command toggles the local echo on and off. If you send the + command in command mode, all typed characters are echoed to the output. Typing + a second time turns local echo off.
C	This command causes the device to attempt to connect to the stored remote address.
C,<*address*>	This command causes the device to connect to a remote address, where <*address*> is specified in hex format. The address is also stored as the remote address. Example: C,00A053112233 // Connect to the Bluetooth address 00A053112233
CF,<address>	Connect and immediately go into FAST data mode. NOTE: you will not be able to re-enter command mode while connected. Example: CF,00A053112233 // Connect to 00A053112233 in fast data mode
CFI	Connect and immediately go into FAST data mode using the last address found from the Inquiry command. NOTE: you will not be able to re-enter command mode while connected.
CFR	Connect and immediately go into FAST data mode using the stored remote address.

Similar to the C command but bypasses the configuration timer.
NOTE: you will not be able to re-enter command mode while connected.

F,1 This command ends configuration immediately and puts the device into fast data mode.
Example: F,1 // Leave command mode and enter fast data mode

H The help command displays a list of commands and their basic syntax.

I,<*time*>,<*cod*> This command performs an inquiry scan, where <*time*> is the scan time in seconds and <*cod*> is the optional COD of the device class for which you are scanning. The default time is 10 seconds, and the maximum is 48. If <*cod*> is unused or set to 0, the device looks for all device classes. When entering a COD, you must provide all six characters, e.g., you would enter 0040F0 for COD 0x40F0. The scan returns a maximum of 9 devices. As devices are found, they are displayed in the format:

<*Bluetooth address*>,< *Bluetooth name*>,<*COD*>00A053000123,MySerial-Port,72010C

Example: I,20,0040F0 // Scan for 20 seconds using the COD 0x40F0

IN,<*time*>,<cod> This command is similar to the I command, but it does not return the Bluetooth name, where <*time*> is the scan time in seconds and <*cod*> is the optional COD of the device class for which you are scanning. Therefore, the device returns the scan result much faster because the device does not have to perform a remote lookup for each device found.
Example: IN,10,001F00 // Scan for 10 seconds using the COD 0x1F00

IS,<*time*> This command performs an inquiry scan with a COD of 0x001F00, which is the default COD for BluFly devices, where <*time*> is the scan time in seconds.
Example: IS,10 // Scan for Roving Networks devices for 10 seconds

IR,<*time*> This command performs an inquiry scan with a COD of 0x0055AA, where <*time*> is the scan time in seconds. BluFly devices use this COD for instant cable replacement.
Example: IR,10 // Scan for instant cable replacement devices for 10 seconds

J This command hides the current 4-digit pin code (or pairing code) used for legacy pairing mode or default mode. When the pin code is hidden, the GP, D, and X commands do NOT display the currently assigned pin code.
To disable the pin code hiding:
• Use the SP, command to set a new pin code
or
• Restore the factory defaults using the commands
SF,1
R,1

K The kill command disconnects the device the current connection**.** The characters KILL<cr><lf> are echoed to the local UART once the connection is broken.
Example: K // Disconnect the current connection.

P,<*char*>	This command passes through any <*char*> up to a carriage return or line feed while in command mode.
Q	Causes device to be non-discoverable and non-connectable (temporarily). Does not survive a power cycle or reset**. Use the "W" command to re-enable.** This command will return **"Quiet"** as a response**.**
R,1	This command forces a complete device reboot (similar to a power cycle).
U,<*rate*>,<parity>	This command causes a temporary UART change, where <*rate*> is the baud rate and <*parity*> is the parity. This command changes the serial parameters immediately, but does not store them to flash memory. The device returns AOK at the current settings, then automatically exits command mode and switches to the new baud rate. The baud rate, <*rate*>, must be EXACTLY 4 characters: 1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115K, 230K, 460K, or 921K. The parity, <*parity*>, is E, O, or N (must be capital letters). Example: U,9600,E // Set the baud rate to 9,600 with even parity
W	This command enables discovery and connection after it has been disabled with the "**Q"** command. The device returns "**Wake"** as a response.

BLE Server OPERATION

The BluFly makes use of a few custom GATT profiles defined by Cypress Semiconductor for implementing a BLE interface to the serial RS232. The service UUIDs, characteristic UUIDs, special permissions, and overall structure are outlined here for quick reference. For detailed reference materials, visit the Cypress website.

CYSPP Profile

The Cypress Serial Port Profile (CYSPP) provides bidirectional serial data transfer between a BLE Client and a remote BluFly BLE Server which passes data through a single hardware serial interface. It supports both acknowledged transfers and unacknowledged transfers, and provides a mechanism for virtual flow control in both the RX and TX direction.

The profile contains a single service ("CYSPP"), which contains three characteristics for data transfer and flow control ("Acknowledged Data", "Unacknowledged Data", and "RX Flow"). The structural outline of this profile is as follows:

- CYSPP Service: UUID 65333333-A115-11E2-9E9A-0800200CA100
  - **Acknowledged Data** Characteristic: UUID **65333333-A115-11E2-9E9A-0800200CA101**
    (Write, Indicate)

    The Acknowledged Data Characteristic is used to send and receive data in an acknowledged fashion. The BluFly firmware can fully track every transfer in both directions. This characteristic has a variable length, supporting transfers in each direction of up to 20 bytes per packet.

    - Configuration Descriptor: UUID **0x2902**
  - **Unacknowledged Data** Characteristic: UUID **65333333-A115-11E2-9E9A-0800200CA102**
    (Write without response, Notify)

    The Unacknowledged Data Characteristic is used to send and receive data in an unacknowledged fashion. The BluFly firmware cannot track transfers using this mode once they have been accepted by the BLE stack. This provides less control, but the lack of acknowledgements also allows for greater maximum throughput. This characteristic has a variable length, supporting transfers in each direction of up to 20 bytes per packet.

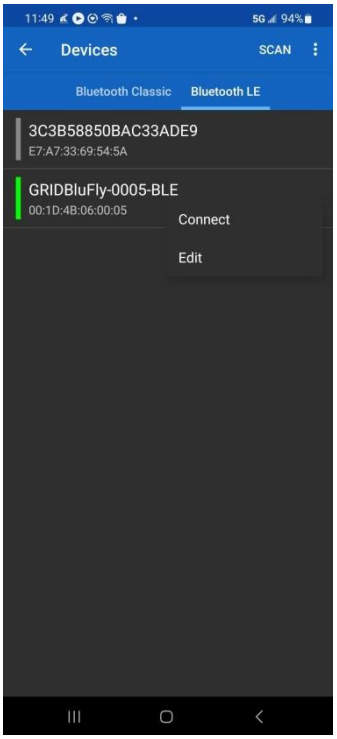
    - Configuration Descriptor: UUID **0x2902**
  - **RX Flow** Characteristic: UUID **65333333-A115-11E2-9E9A-0800200CA103**
    (Indicate)

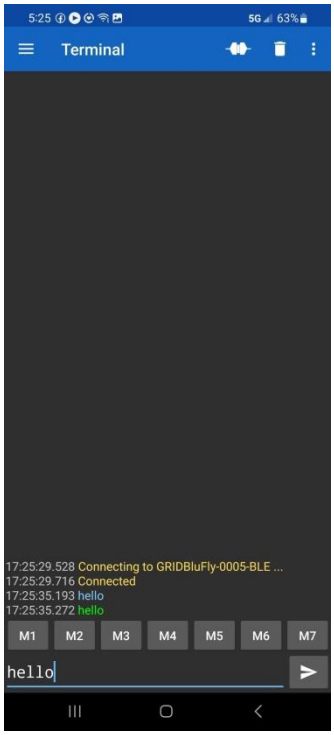
    The RX Flow Characteristic is used to indicate to the client that the server can no longer safely receive new data. If the client subscribes to indications from this characteristic, the server will assume that the client obeys flow control signals. This characteristic is one byte in length. An indicated value of "0" means that it is safe for the client to send data, while a value of "1" means that the client must refrain from sending data.

- Configuration Descriptor: UUID **0x2902**

Example BLE Client connection from a cell phone App

1. Power cycle the BluFly to restart the BLE Discovery timer
2. Select Devices from the menu
3. SCAN for devices
4. Long click on the GRIDBluFly-xxxx-BLE device and select Edit
5. Choose Custom and the Service UUID, Read Characteristic UUID, and Write Characteristic UUID
6. Connect
7. Send data (screenshots show data sent with Loop back jumpers on BluFly)



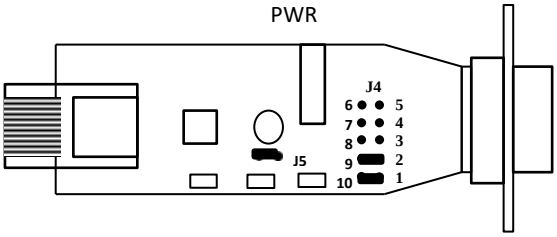


NULL MODEM JUMPERS

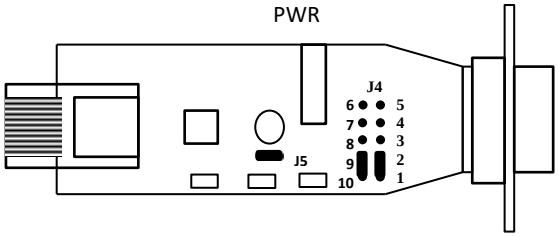
A common issue in RS232 connections is a mismatch between TX and RX pins 2 and 3. In some instances you may need to swap TX and RX and then also RTS and CTS if using these control signals. This can be accomplished using a null modem cable or adapter.

You can configure the BluFly's serial interface to enable null modem wiring where the TX and RX signals are swapped. You access the J4 jumper block by removing the cover from the BluFly. Remove the screw from the back of the case and lift off the top cover. See the following diagrams for common configuration examples. The jumper at J5 must always be installed for correct operation.

Male (DTE) DB9

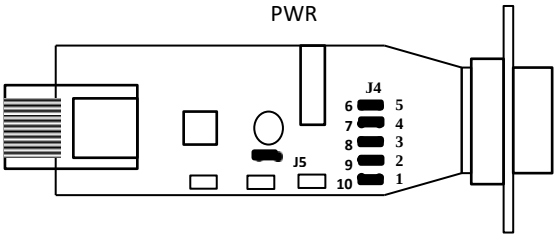


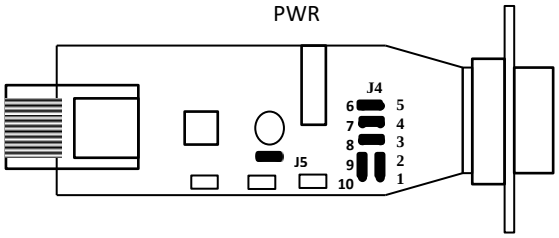
Female (DCE) DB9



3 wire straight DTE/DCE pinout (TX, RX and GND)

Pin	Straight DTE (DB9 Male)	Straight DCE (Female)
1	NC	NC
2	RXD (in)	TXD (out)
3	TXD (out)	RXD (in)
4	DTR (out) [option]	NC
5	GND	GND
6	NC	NC
7	NC	NC
8	NC	NC
9	External Power 5-16 VDC	External Power 5-16 VDC

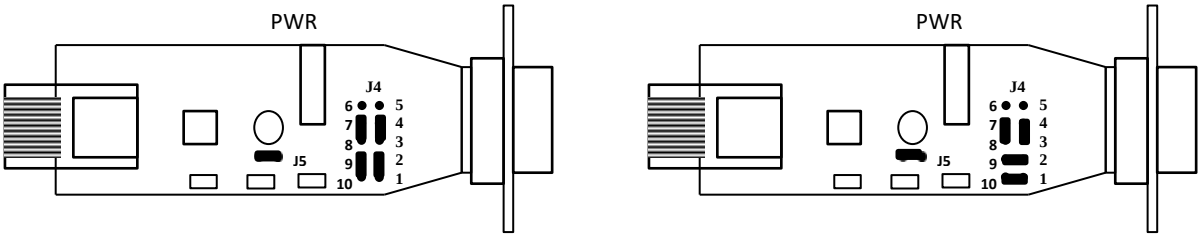




All signals straight DTE/DCE pinout (TX, RX, RTS, CTS, DTR, GND)
DTR output is tied to the RTS output by jumper 6-5.

Pin	Straight DTE (DB9 Male)	Straight DCE (Female)
1	NC	NC
2	RXD (in)	TXD (out)
3	TXD (out)	RXD (in)
4	DTR (out) [option]	NC
5	GND	GND
6	NC	DSR (out) [option]
7	RTS (out)	CTS (in)
8	CTS (in)	RTS (out)
9	External Power 5-16 VDC	External Power 5-16 VDC

Note: Simply connecting the RTS and CTS signals over RS232 does not enable hardware flow control. You must set UART Hardware Flow control using the command **SHWF** while in Command Mode.



Null modem pinout (TX, RX, RTS, CTS, GND, No DTR)
TX-RX and RTS-CTS pins reversed, DTR is not reversible

Pin	NULL Modem (DB9 Male)	Null Modem (Female)
1	NC	NC
2	TXD (out)	RXD (in)
3	RXD (in)	TXD (out)
4	NC	NC
5	GND	GND
6	NC	NC
7	CTS (in)	RTS (out)
8	RTS (out)	CTS (in)
9	External Power 5-16 VDC	External Power 5-16 VDC

RS232

RS232 is a telecommunication standard for point-to-point serial communication. The physical interface uses asynchronous serial with voltage levels in the +/- 5 volt range. The most common RS232 interface connector is a DB9, either DTE (male) or DCE (female). A 3 wire interface is the minimum required for two-way communication: Transmit data, Receive data and Ground.

The standard configuration for RS232 connections with straight DB9 to DB9 connectors is shown below. It is

very important to note the direction of the data flow, as indicated by the arrows. The names of the signals are often confusing so follow the direction of the data flow.

Note: Some of the signals have no connection in the BluFly. Only RD, TD, RTS, CTS and DTR are supported.

Straight Cable DB9 to DB9

DTE Device (Computer)		Signal Direction	DCE Device (Modem)	
Pin	**Signal Names**		**Pin**	**Signal Names**
#1	Carrier Detect CD	←	#1	Carrier Detect CD
#2	Receive Data RD	←	#2	Receive Data RD
#3	Transmit Data TD	→	#3	Transmit Data TD
#4	Data Terminal Ready DTR	→	#4	Data Terminal Ready DTR
#5	Signal Ground GND	—	#5	Signal Ground GND
#6	Data Set Ready DSR	←	#6	Data Set Ready DSR
#7	Request to Send RTS	→	#7	Request to Send RTS
#8	Clear to Send CTS	←	#8	Clear to Send CTS
#9	Ring Indicator RI	←	#9	Ring Indicator RI