


[HOME](#) [PRODUCTS](#) [SUPPORT CENTER](#) [WIRELESS RESOURCES](#) [SUBSCRIBE](#) [CONTACT US](#)

Search



WANT TO RECEIVE THE LATEST BLOG POSTS FROM LAIRD? SUBSCRIBE BELOW:

SUBSCRIBE

CATEGORIES

[Bluetooth](#)
[CPBU News](#)
[Industrial](#)
[Industry News](#)
[IoT](#)
[Medical](#)
[RAMP](#)
[Uncategorized](#)
[Wi-Fi](#)
[Wireless Technology](#)

RECENT POSTS

[New 60 Series SOM & Summit Software Stack Deliver Superior Enterprise-Class Wi-Fi Connectivity](#)
[New BL651 Module Joins Laird's Growing Nordic Family of BLE 5 Solutions](#)
[What You Need to Know About Bluetooth SIG Deprecation of BT v4.1 and Older Specs](#)
[Get a First Look at Laird's BL654 Bluetooth 5 Module](#)
[Webinar Recap: BLE Mesh – A Practical Guide to Simplify Product Development Time](#)

ARCHIVES

Select Month ▼

MARCH 2020

M	T	W	T	F	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15

« Nov

TBT: White Paper on BLE

New BLE Breakout Board Series for Any Hardware Budget

Is UART Flow Control (RTS/CTS) Necessary for Proper Operation in Wireless Modules?

on AUGUST 27, 2015 · [LEAVE A COMMENT](#)

Share

By Mark Duncombe, Field Applications Engineer III

Some serial implementations link CTS and RTS to remove the need for handshaking. Laird strongly recommends that developers utilize RTS / CTS for flow control in their applications. Continue reading for technical scenarios that developers should consider before foregoing the correct use of CTS / RTS.

When referring to RTS/ CTS:

- Module UART CTS is an input Module
- RTS is an output
- Asserted is 0V or low
- Deasserted is 3.3V or high

Module UART CTS is an input and is connected to the host's RTS, which consequently is an output.

The host is able to tell the module it is available to accept data over the UART by controlling its RTS output which signals to the module via the module CTS input. So the module will not send data unless CTS is asserted. Basically, an asserted RTS output tells the peer that it is safe for the peer to send data.

The host can stop the module sending it data by taking its RTS high, which in turn takes the module's CTS high. Likewise the module can stop the host sending data by taking its RTS high which takes the host's CTS high.

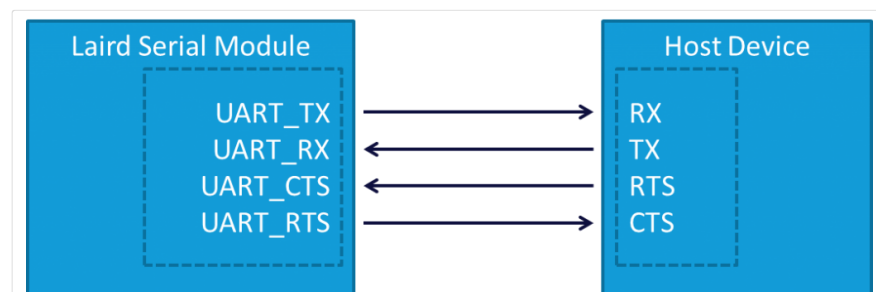


Figure 1: Recommended minimum UART wiring with correct use of CTS/RTS (recommended).

Disabling Flow Control

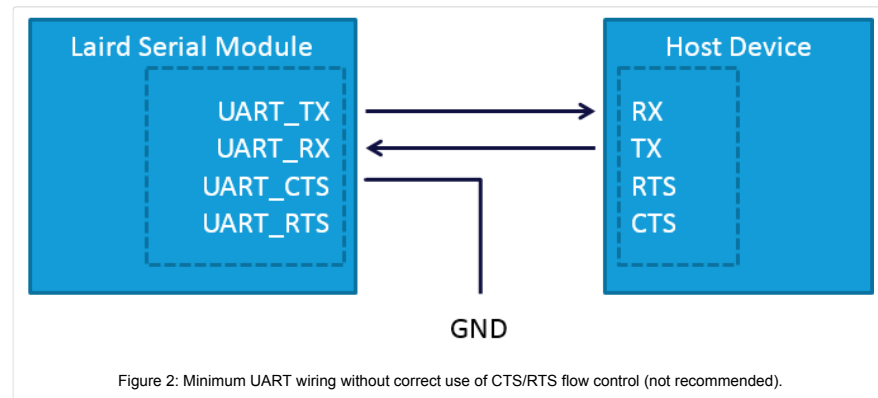
M	T	W	T	F	S	S
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

« Nov

RSS

Search

However if the application demands are for an on-air throughput that is lower than the throughput of the UART (baud rate, parity, and stop bits) setting or the risk of potential data loss or module reset is accepted by the developer, then the CTS input line can be pulled to 0V/Gnd and RTS output line allowed to float as per Figure 2.



Before Foregoing the Correct Use of CTS / RTS Please Consider the Following

Issue #1: Module not responding to commands

Module UART_CTS is an input and it's not good practice to leave it floating. If it was to inadvertently sit at 3.3V due to the circuit design then the module will assume the host is not ready to receive data. The host sends it a command such as "at" but gets no response back as the module senses it can't send because the host is not ready to receive data from the module. This is a common problem for developers that skip using a dev kit and wire directly to a module using just UART TX, RX and GND.

Solutions

1. Make proper use of flow control (highly recommended), see Figure 1.
2. Tie module UART CTS to 0V so that the module will always assume the host is ready to receive data (not recommended), see Figure 2.

Issue #2: Module overflow causing data loss

Unlike a wired connection, a radio connection is more readily subject to degradation caused by increasing range and interference. At some point the over-the-air throughput may fall below the UART throughput (note maximum throughput when there is no parity and one stop bit is configured is 80% of the baud rate), at which point the module will throw its arms in the air and say, "hang on, stop sending data so I can catch up". It does this by taking its RTS (output) high. The host sees its CTS (input) follow suit and stops sending data. Let's consider what happens when CTS/RTS is not used correctly and Module UART_CTS is tied to Gnd or 0V. When this happens, the module is overwhelmed with data from the host, throws its arms in the air and says, "no more" but Module UART_RTS does not go anywhere, so the host keeps on sending, oblivious to the module's plight. At some point an overflow may occur, which may result in data loss or possibly even a module crash.

Solutions

1. Use proper CTS/RTS flow control (highly recommended, see Figure 1).
2. Accept that you might lose data.

Issue #3: But what about the host?

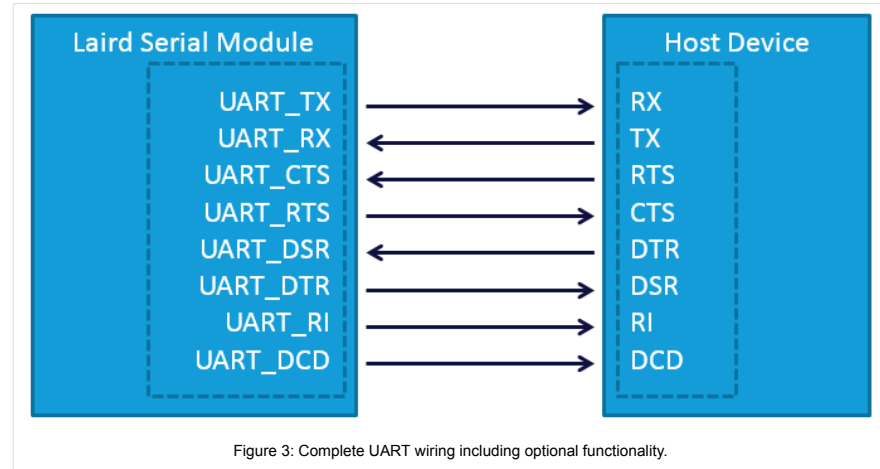
If the host is also expecting proper use of CTS/RTS it too might have issues when it comes to sending data to the module as it may not be told that the module is ready to accept its data. So you might also have to take the host CTS to 0V before it will communicate. However, taking host CTS to 0V or tying host CTS to RTS you have the same potential issues as the module sees in the issues above.

Solutions

1. Tie the host CTS to 0V.
2. Make proper use of CTS/RTS flow control (highly recommended), see Figure 1.

Additional UART Lines

The additional UART lines of DSR and DTR are optional and RI and DCD are not necessary, but the developer can create the functionality using any spare GPIO that is available and using [smartBASIC](#) / AT commands to provide behaviour as necessary, please refer to the relevant module documentation for further information.



Have an opinion on this post? Comment below with your thoughts and don't forget to [subscribe](#)!

Learn more about Laird's Classic Bluetooth and Bluetooth Low Energy (BLE) solutions, [here](#).

Tagged with: [BLE](#) • [Bluetooth](#) • [CTS](#) • [Flow Control](#) • [handshaking](#) • [how to](#) • [RTS](#) • [UART Wiring](#)

If you enjoyed this article, please consider sharing it!



Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

POST COMMENT