Implementing RS-232 Flow Control on a Stellaris® Microcontroller

Application Note

AN01255-02

Copyright

Texas Instruments
108 Wild Basin, Suite 350
Austin, TX 78746
Main: +1-512-279-8800
Fax: +1-512-279-8879
http://www.luminarymicro.com

Table of Contents

Introduction

When connecting a Stellaris® microcontroller UART to a device, it may be necessary to implement serial flow control. Reasons for this might include:

- A legacy device requires flow control for proper operation.
- A data source from within the Stellaris device (for example, the Ethernet port in a Serial-to-Ethernet converter) may provide data faster than can be consumed by the external serial device.
- The serial device may provide a large burst of data faster than it can be consumed by another peripheral within the Stellaris device (for example, a low-speed I^2C device).

The Stellaris UART does not provide built-in flow control signals. However, by using the Stellaris GPIO module with its interrupt support, flow control can be easily implemented.

Flow Control Basics

RS-232 is a standard for serial binary data transfer defined by the Electronics Industries Alliance (EIA). This standard defines both the electrical and functional characteristics of the various serial interface circuits. In this application note, the focus is on the functional characteristics of the following circuits:

- Transmitted Data (TxD): Data sent from the Data Terminal Equipment (DTE) device to the Data Communications Equipment (DCE) device.
- Received Data (RxD): Data sent from the DCE device to the DTE device.
- Request-to-Send (RTS): Signal asserted by the DTE device to indicate that data is ready to be sent by the DTE device to the DCE device.
- Clear-to-Send (CTS): Signal asserted by the DCE device to indicate that the DTE device can send its data.

Note: Most computers with serial interfaces are configured as DTE devices. An example of a DCE device would be a modem.

In the original versions of the RS-232 specifications, the definition of RTS/CTS flow control was asymmetric. This type of flow control allowed for the following conditions:

- The DCE device would be able to disable the transmitter during idle time, only waking up the transmitter when the DTE device indicated that data was ready for transmission.
- Similarly, the DCE device could be operating in a bus topology, and leave the transmitter off when not actively driving the bus.
- The DTE device could source/sink data at a much higher rate than the DCE device. In the DCE-to-DTE direction, data loss would not be an issue. However, in the DTE-to-DCE direction, without this RTS/CTS flow control, data loss would become a problem.

As technology progressed, a symmetric variation on this RTS/CTS flow control was developed. Eventually, it became a part of the RS-232 standard in version E. In this new version of the standard, the CTS signal still carries the same basic meaning, but the RTS signal is assumed to be always asserted. This means that the DCE device will assert the CTS signal at any time it is capable of receiving data from the DTE device. The RTS signal now becomes an indication from the DTE device that it is ready to receive characters from the DCE device.

Flow Control Example on Stellaris

An example of RTS/CTS flow control can be found in the Serial-to-Ethernet RDK that is available from the www.luminarymicro.com web site. In this device, the more modern, symmetric RS-232E version of RTS/CTS flow control is implemented. The software used to implement this flow control comes with the RDK, and is also available for download from the web site.

In this code, the Serial-to-Ethernet converter is configured as a DCE device. However, because symmetric RTS/CTS flow control is used, the code will work for either DCE- or DTE-configured interfaces.

In this code, the GPIO port on which the RTS signal resides is configured as an input, and interrupts are enabled on both rising and falling edges. In the GPIO interrupt handler, if the flow control input is determined to be asserted, the UART transmitter is immediately disabled. This allows any character that is currently being transmitted by the hardware to complete, but will not send any additional characters on the serial interface. If the signal is determined to be deasserted, the transmitter will be reenabled, allowing the data in the UART FIFO to be sent, and normal transmit operations to resume.

In the reverse direction, the GPIO port on which the CTS signal resides is configured as an output. In this code, the assumption is made that we can respond to the UART interrupt fast enough to prevent overrun of the UART FIFO. With this assumption, the triggering of the CTS flow control signal is based on the level of a ring buffer that is used to buffer the data between the UART and the Ethernet port. If the amount of data exceeds the 75% threshold, the CTS signal is deasserted to stop transmission of data from the DTE device. When the threshold drops back down to 25%, CTS is reasserted, allowing the DTE device to resume transmission of data.

Conclusion

With the improved interrupt latency provided by the ARM Cortex-M3 processor and the interrupt support provided by the Stellaris GPIO modules, the addition of RTS/CTS flow control to your embedded serial application is fairly straightforward. The symmetric RS-232E version of RTS/CTS flow control is implemented in the RDK S2E software. The source code is provided allowing incorporation into your own applications.

References

The following documents and source code are available for download at www.luminarymicro.com. The source code for the module is also available as part of the Stellaris Peripheral Driver Library package.

- StellarisWare® Driver Library, Order number SW-DRL
- *StellarisWare® Driver Library User's Manual*, publication number SW-DRL-UG
- *Serial-to-Ethernet Converter for Stellaris® Microcontrollers Application Note*, Publication Number AN01266

The Serial-to-Ethernet hardware/software is also available as both a module and as a reference design kit (RDK) at the following locations:

- Stellaris® Serial-to-Ethernet module: www.luminarymicro.com/products/mdl-s2e.html
- Stellaris® Serial-to-Ethernet RDK: www.luminarymicro.com/products/rdk-s2e.html

Important Notice

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265