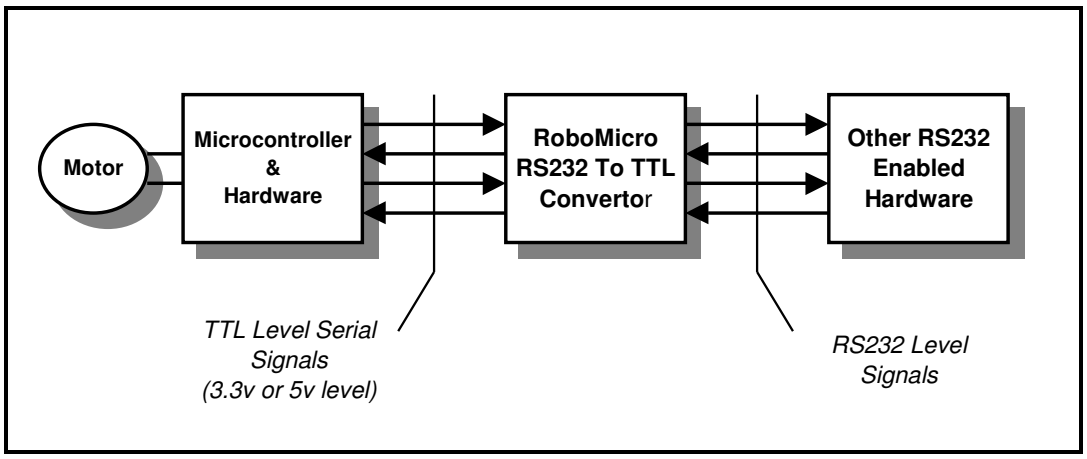
AP22 - RS232 Converter: Application Note

Introduction

This application note demonstrates how to connect a RoboMicro RS232 Converter module and use this module to allow communications between a low-voltage (TTL) based system and an RS232 enabled piece of equipment.

FIGURE 1: BLOCK DIAGRAM



A common feature that is found on many Micro-controllers is a peripheral commonly called a “Serial Port” (or UART). There are two ways a Micro-controller can implement a “Serial Port”.

> ***Hardware UART***: Where all the logic, Baud rate generator, interrupts, flags, control and I/O toggling is implemented in hardware, with the software providing minimal control.
>
> ***Software UART:*** Where the application software, running in the micro-controller, will implement the logic, baud rate generator and I/O line toggling

In either case, the resultant Serial-Stream will be used to communicate with another Micro-controller or other device. However, it is usually the case that this “other-device” will have an “RS232 Interface” hardware. The “Serial Stream” being generated by the Micro-controller UART is of low-voltage (typically 3.3v or 5v), and cannot directly be connected to this RS232 interface. Damage to the sensitive Micro-controller I/O would result if such a direct connection is made.

Table 1: TTL & RS232 Voltage Levels

Logic	TTL	RS232
Logical 0	0v	+5v to +15v
Logical 1	+3.3v to +5v	-5v to –15v

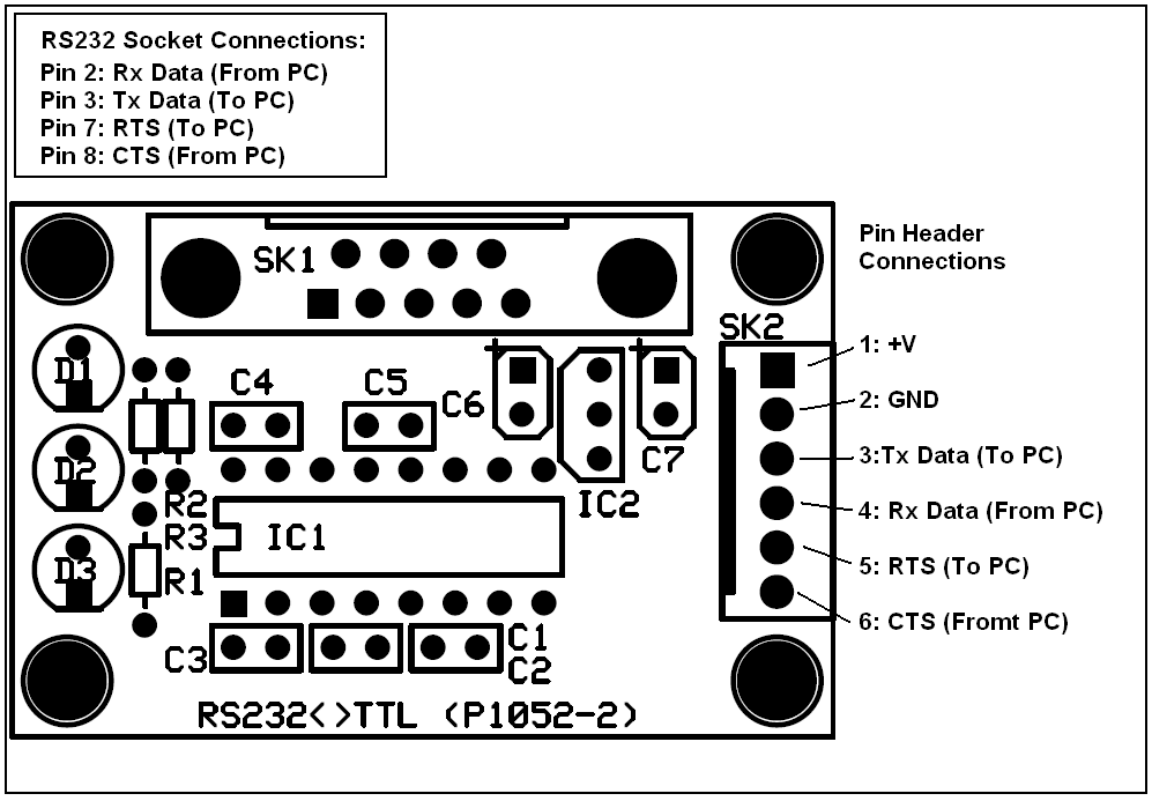
As can be seen in Table 1, what is needed is a way to connect the Micro-controller UART Serial Stream to an RS232 Interface. This is where the RoboMicro RS232 converter module fits in.

This module converts signals between RS232 voltage levels and TTL voltage levels. It is a two-way process, where TTL signals are converted to RS232 levels and RS232 levels are converted to TTL levels.

The RoboMicro RS232 converter carries out two functions
1. Provides a voltage conversion from TTL to RS232 and back to TTL
2. Provides the required logical inversion of 1 to 0 and 0 to 1.

RoboMicro Converter

FIGURE 2: RS232 Converter



Low Voltage (TTL Connections)

Pin1 : +V
This pin is the PSU connection needed by the RS232 converter. This voltage can be unregulated and be a value between +8 v and +25v (+5v to +20v for 3.3v version)

Pin 2: GND
This pin is the PSU & signal ground pin. This pin is usually connected to the Vss (0v) rail of the application hardware.

Pin 3: Tx Data (UART to PC)
This pin is usually connected to the Tx pin of the UART in the Micro-controller. This is an input pin and is expecting the "Serial Stream" from the Micro-controller.

Pin 4: Rx data (PC to UART)
This pin is usually connected to the Rx pin of the UART in the Micro-controller. This is an output pin and will be outputting data received from the RS232 side of the converter.

Pin 5: RTS (UART to PC)
This pin is part of the RS232 Flow-Control System. This signal is used by the UART to flag to the PC when the UART is ready to receive data. However, due to the various "standards" of RS232, the exact way this signal will be used will change from system to system.

Pin 6: CTS (PC to UART)
This pin is part of the RS232 Flow-Control System. This signal is used by the PC to flag when the micro-controller can send data. However, due to the various "standards" of RS232, the exact way this signal will be used will change from system to system.

RS232 D-type Connections

Pin 2: Rx Data
This pin receives RS233 data sent from the other device. This pin is usually connected to Pin 3 of the other devices RS232 D-Type socket (9-way)

Pin 3: Tx Data
This pin outputs the converted RS232 serial stream sent from the device connected to the converter. This pin usually connects to Pin 2 of the other devices RS232 D-type socket (9-way)

Pin 5: RS232 Ground
This is the common/ground connection between the two RS232 devices

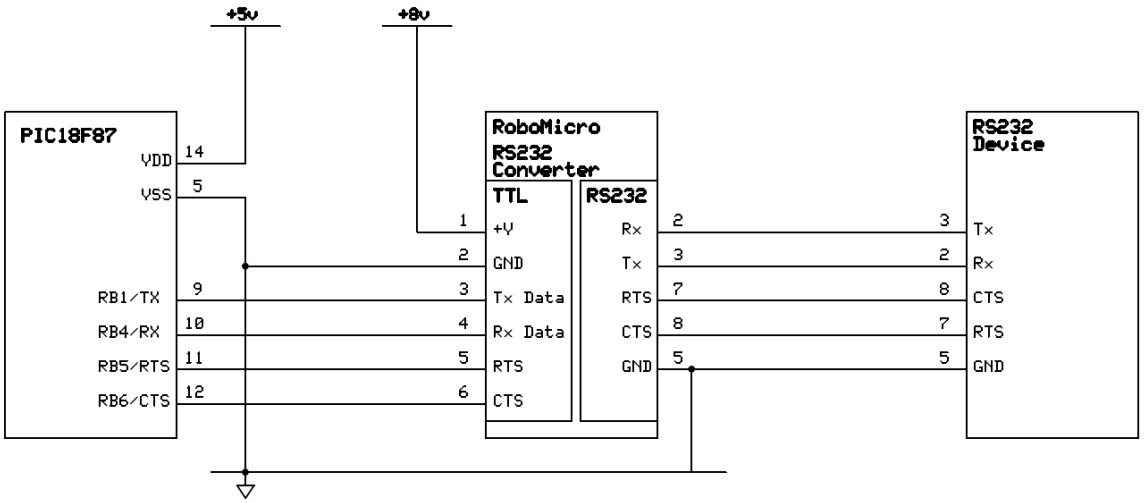
Pin 7: RTS (Ready to Send)
This signal is controlled by the RS232 Converter. It is used to indicate to the other RS232 device when it can send data via the Rx data Line (Pin 3)

Pin 8: CTS (Clear To Send)
This signal is controlled by the other RS232 device. It is used to allow the other RS232 device to indicate when it is ready to receive serial data via the Tx Data Line (Pin2)

(NB! All other RS232 Pins are unconnected)

Application Example (Automatic Boot loader)

This application note will describe how the RoboMicro RS232 converter can be used to implement an Automatic Boot loader Application.



Boot loader Operation

Most boot loaders operate in a similar manor;
Reset the Application Micro-controller, Establish Communications Link with Boot loader Firmware, and Download New Application to Micro-controller Memory and then restart the Micro-controller once download is complete.

Using the RoboMicro RS232 Converter – this Boot loader & download process can be made fully automatic, via the use of the RTS and CTS control lines.

- *The CTS Line is going to be used to RESET the Micro-controller*
- *The RTS Line to going to be used to indicate when the Boot loader code is active in the Micro-controller.*

Boot loader Control & Download Sequence.

1. The Application is started in the PC. The PC application checks & configures the RS232 interface to the correct speed for the bootloader, etc.

2. The PC Application then toggles the CTS Line. This causes the Micro-controller to reset and start to run the Boot loader Firmware. This firmware sets logical 1 on the RTS Line. (Normal Application operation will set logical 0 on this RTS line)

3. The PC application waits for 10msecs and then checks the logical level of the RTS Line. If the RTS Line is Logical 1, then the Boot loader code is running in the Micro-controller. If the RTS line is Logical 0, then the

Application is running in the Micro-controller and the PC application will toggle the CTS Line again (repeating step 2)

4. The PC Application then starts communication (Using the Tx & Rx lines) with the Boot loader, and downloads the new application.

5. Once download is complete, the PC Application then toggles the CTS Line again to reset the Micro-controller. It then waits until it sees the RTS Line go from logical 0 to Logical 1 and back to logical 0 again. This then indicates that the download was successful and can stop the download process.