

UART to RS232 or USB adapters

By BOJAN KOVAČ

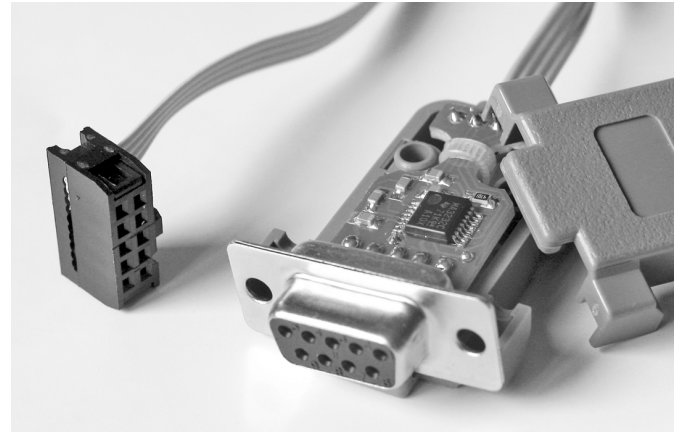
The abbreviation **UART** stands for **Universal Asynchronous Receiver/Transmitter**. It is a serial data interface with a defined speed. Regardless of the actual voltage level of the RS232 signals, these RS232/RS485 devices present signals to the microcontroller as a logical 0 or 1.

This protocol sends data to the microcontroller serially, which means the data stream must be converted back to parallel (which is the function of the UART block in the microcontroller). A block diagram of such a data transfer is shown in Figure 1.

Data transfer via the USB bus is similar to RS232 with a some differences: USB is a faster transfer method, and to interface it with a microcontroller, we can use a USB to UART bridge, such as the FT232RL made by Future Technology Devices International LTD (FTDI) [1].

There are a variety of different asynchronous serial adapters available on the market. Bascom-AVR communicates with all of these with one statement meant to send data to the AVR's UART- the *Print* statement.

Probably there is no other microcontroller language with as simple a statement to control the UART, as Bascom/



AVR. There are numerous UART parameters that must be configured, but many of them are changed very infrequently, so we can often just use the default settings.

When designing a new device, I always reserve pins for serial communications. That allows me to easily change program parameters, not to mention being able to externally store data from the microcontroller.

Logic levels amongst digital devices vary. Usually a logical 1 is either 5.0 V or 3.3 V (and more recently, 1.8 Volts). Generally the logic levels depend upon the microcontroller's power supply voltage rating. This can present a problem, as not all microcontrollers and peripheral devices work at the same voltage. So, for example, a simple connection between your mobile phone and a USB or RS232 port may be problematic. The solution to this is to use level-shifting devices (usually ICs) that present suitable logic levels to both sides of the communications link.

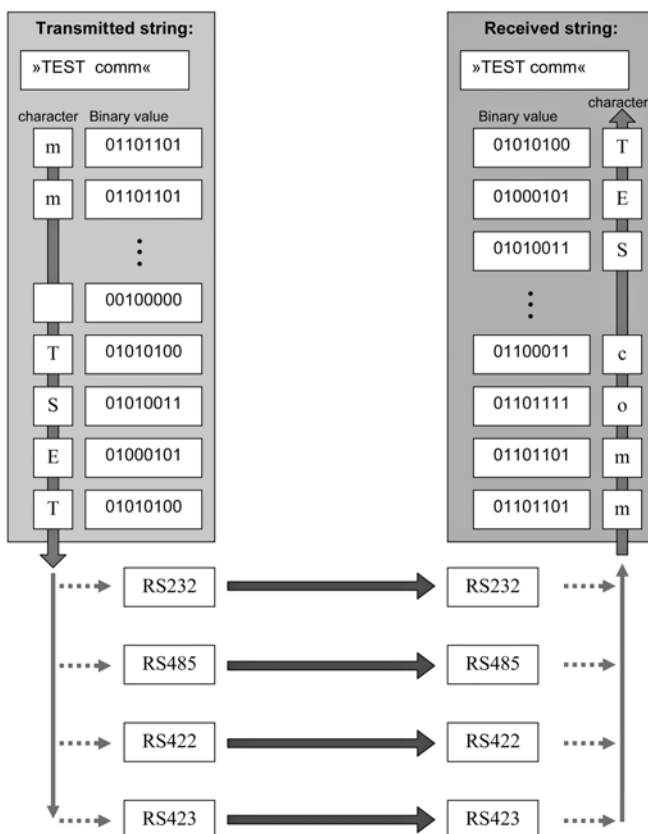


Figure 1: UART data



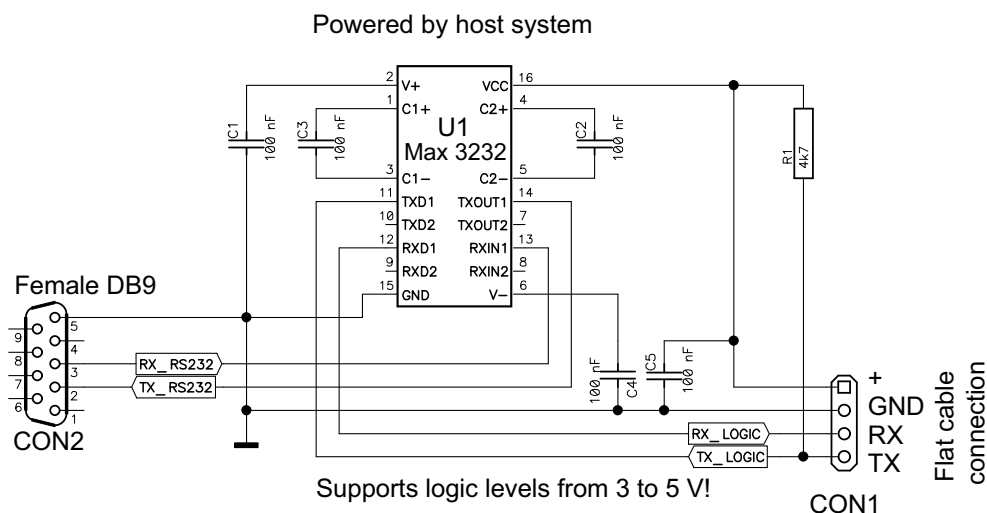


Figure 2: Schematic of an RS232 to UART adapter

For serial communications we use two signals: Rx and Tx, both of which are referenced to GND.

within the DB9 female connector that is used to mate with the PC's RS232 port.

UART to RS232

This type of adapter will work when we want to establish serial communication between a microcontroller device (using logic level signals of 3.3 or 5.0 volts) and a PC with an RS232 port. This adapter gets its power from the microcontroller side, which generally uses a power supply of either 3.3 or 5.0 Volts. It converts the logic signals into those that are compatible with the RS232 protocol. This type of adapter can be totally enclosed

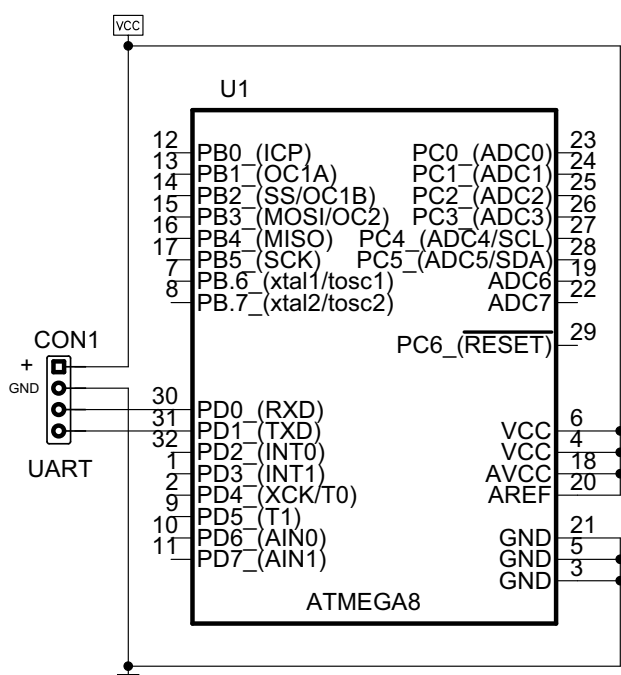
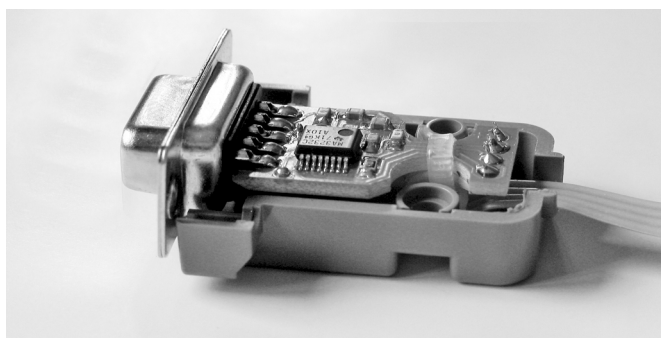


Figure 3: Example of an adapter connection to the microcontroller

Circuit details

This circuit contains a MAX3232 made by Maxim [2]. This device works well with logic levels in the 3.0 to 5.5 Volt range. To generate suitable RS232 voltage levels, we must add four capacitors (for its internal charge-pumps) plus one additional capacitor for filtering. The device is connected to the host using a 4-wire cable. Users of the MiniPin II or MegaPin boards will notice that both boards have a pin-compatible IDC connector mounted, for this purpose. On the RS232 side we have only the Rx, the Tx and GND signals. In most instances, these communications signals are sufficient (i.e. when no flow-control or *hand-shaking* is necessary).

Usage

These adapters can be used in various ways. For a start, we can save PCB space by not using an RS232 chip. Instead, we just put a 4-pin connector on-board that will connect to our adapter, when needed. An example of such a connection is shown in Figure 3.

Another very interesting use is to connect our cell-phone to a PC via this adapter. Using AT commands, we can access all cell phone functions, and there are a lot of



UART to USB

Circuit details

The FT232RL has an input pin labelled V_{CC} IO. This power



Table 1: Setting V_{IO} voltage with jumpers J1 and J2

Usage

If the target circuit board contains a suitable power supply, we will connect all four pins of the IDC connector



and short pins 1 and 2 on jumper J2. In this mode, the power supply voltage goes directly to V_{ccIO} pin. If the target circuit can't supply power, then we have to measure it and try to get as close to it as possible, with jumper J1. On jumper J2, we select the internal voltage source for V_{ccIO} , by shorting pins 2 and 3. All jumper combinations are shown in Table 1, and a picture showing the location of the jumpers is shown in Figure 5.

Conclusion

To do two-way communications, you don't need to design a special circuit, if this communication is only done infrequently. With the adapters described, you can establish communication only when needed. You just have to pick a suitable adapter. The typical use of these adapters is for debugging, when data from the microcontroller is sent to the PC using Bascom's *Print* statement. When you design your device, all you have to add is a four pin connector, wired to $+V_{cc}$, GND, Rx and Tx. It doesn't matter whether or not you use them: they will be there when we need them.

Both adapters operate at various voltages: the USB to UART board also supports logic voltages below 1.0 Volt. The IDC connector on both adapters is compatible with both the MiniPin II and MegaPin development boards.

Notes:

1. www.ftdichip.com
2. www.maxim-ic.com

