

RS232 Board C & Assembly Strategy

Introduction

This document will suggest basic strategies for creating 'C' and Assembly code for the RS232 E-Block. Because this E-Block can be used with a variety of upstream boards (e.g. PICmicro multi-programmer board), this document will not provide all of the information required. See the "further reading" section for more complete reference information.

This E-Block has 4 lines - "tx", "rx", "cts" and "rts", which are "transmit", "receive", "clear-to-send" and "request to send" respectively. The latter 2 are optional and are only required if hardware flow control is required.

Implementing a strategy

The following strategy is specific for a PICmicro microcontroller, but should be adaptable to any upstream device. It will rely on a PICmicro's internal USART module and will not implement any flow control.

Remember that the correct jumper setting on the patch system of the E-Block must be selected so that the "tx" and "rx" lines are connected to the correct TX and RX lines on the PICmicro.

Initialisation

The following things need to be initialised to use the USART appropriately:

- 1) The BAUD rate must be configured appropriately
- 2) The USART needs to be enabled, including the transmit and receive sections
- 3) Optionally, the PICmicro can be configured to interrupt on successful transmission or reception

Data transmission

Here's a basic approach to implement data transmission:

- 1) Wait for any pending transmissions to complete
- 2) Load the data into the TXREG register

Data reception

Here's a basic approach to implement data reception:

- 1) Wait until data reception has been confirmed
- 2) Check for any reception errors and deal with them appropriately
- 3) Read the data from the RCREG register

Advanced strategies

Interrupts can be set so that your program is informed when data reception takes place. This will allow your program to continue normal operation and only deal with data when it has been received.

Basic hardware flow control can also be implemented by using the cts and rts pins.

Further reading

There is a vast amount of information regarding RS232 available which should be the first port of call for anyone interested in implementing this popular communication protocol.

Specific information on how to initialise the USART and send and receive serial data can be found in the relevant microcontroller / USART datasheet.