

Programming the Serial Interface

Two ways:

- Polled I/O
 - Interrupts are not enabled. All the work is done by checking flags in flag registers or checking for the presence of data. Repetitive periodic checking of devices.
- Interrupt Driven I/O
 - Device generates a voltage to signal that it needs something. Real life example: teacher responding to students when their hands are raised.

Registers

BAUD

Set the rate of bit-wise communication on the serial wire.

The number of signal changes per second

Typical rate: 9600 BAUD

Other standard rates: 1200, 4800

The upper nibble should be a 0

The lower nibble should be 0b0011

SCCR1, SCCR2

SCSR – Status Register

RDRF - Bits are set to 1 to specify that a character has arrived

SCDR – Serial Communication Data Register

Contains 1 byte of buffer space. It will not receive more than 1 character before telling you something has happened. A flag is then set in the SCSR to tell you a character is waiting and no additional characters will be stored until you fetch it out of the buffer and clear the flag.

LDAA of the SCDR automatically clears the RDRF flag.

Similarly,

Check the TC flag before writing another value.

STAA to the SCDR automatically sets the TC flag.

PD1 – Serial Transmit

PD0 – Serial Receive

Max 232 chip – converts 0/5 volts to the +/- voltages of RS232

Device drivers initialize their device and then they wait.

Init BAUD

Init SCCR

TST RDRF

Is RDRF = 1?

 If not, back to RDRF

 If it is:

 Move SCDR character

 Was the character equal to CR?

Configuration Information:

Set BAUD to 0x03

Ignore SCCR1

Set SCCR2 to 0b00001100 = 0x0C