Lecture #3 Outline

• Announcements

• AVR Processor Resources
  – UART (Universal Asynchronous Receiver/Transmitter)
  – SPI (Serial Peripheral Interface)
Announcements

- Have you finished Lab #1?
  - Due date is: **Monday, (5pm?)**

- Lab#0
  - Research your ideas
  - Project Idea Discussion (arrange to meet)
  - Do you need a group

- In-Circuit Emulator (AVR-ICE)
  - Works like simulator but on real hardware
  - One station available, second one coming soon
• Interrupts
• Timers
• UART (Universal Asynchronous Receiver/Transmitter)
• SPI (Serial Peripheral Interface)
• A/D Converters (Analog to Digital)
• Analog Comparator
• General Purpose Ports
  – PORTA
  – PORTB
  – PORTC
  – PORTD
  – (Special Functions)
• Special Purpose Pins
  – Crystal (XTAL1/XTAL2)
  – RESET
  – ICP, OLE, OC1B
• Power (VCC/GND)
• 32 Registers (R0-R31)
• 4K Prog ROM
• 512 bytes RAM
• 512 bytes EEPROM
• 32 I/O lines
• 13 Interrupts
• Lots of fun built-in peripherals
• The UART, or Universal Asynchronous Receiver Transmitter, provides hardware support for a serial port on AVR processors
  – Signaling is compatible with PC/Mac/Unix serial (RS-232C)
• The UART provides:
  – Parallel-to-Serial and Serial-to-Parallel conversion
  – Start and Stop Bit framing
  – Parity Generation
  – Baud-Rate Generation (2400-115.2kbps at 3.686 or 7.37MHz)
  – Interrupts
    • Transmit Complete
    • Transmit Data Register Empty
    • Receive Complete
Serial Specification (RS-232C)

- **Data**
  - Start bit
  - 6, 7, 8, 9 data bits
  - Parity bit optional (E, O, M, S, N)
  - Stop bit

- **Voltages**
  - Processor outputs 0/5V logic-level signal
  - RS-232C uses +12V/-12V signal
  - Level-converter IC provided on STK500 (MAX202)
- **UDR (UART Data Register)**
  - Write bytes to transmit
  - Read received bytes

- **USR (UART Status Register)**
  - Rx/Tx complete signal bits
  - Framing error, overflow signal bits

- **UCR (UART Control Register)**
  - Interrupt enable bits
  - Rx/Tx enable bits
  - Data format control bits

- **UBRR (UART Baud Rate Register)**
  - Baud rate generator division ratio
UART Transmitting

- Send a byte by writing to UDR register
  - TXC bit in USR is set when the final bit has finished transmitting
  - Tx Complete interrupt triggered if enabled in the UCR
  - Must wait for current byte to finish transmitting before sending the next one
• How do I know a byte has arrived?
  – Watch the RXC bit in USR
  – Use the Rx Complete interrupt and write an ISR
• Read received bytes from the UDR
  – UDR is double-buffered, but be sure to read it in time
• Set by UBRR
• Varies with $f_{CK}$

![Table](image)

### BAUD Rate Formulas

$$BAUD = \frac{f_{CK}}{16(UBRR + 1)}$$

- BAUD = Baud rate
- $f_{CK}$ = Crystal Clock frequency
- UBRR = Contents of the UART Baud Rate register, UBRR (0 - 255)
UART Example Code

- echo1.asm
  - Echos characters sent over serial port back to the sender
  - Shows setup of UART
  - Shows non-interrupt use of both serial receive and transmit

- echo2.asm
  - Echos only printable characters
  - Uses interrupt-driven receive
  - Implements a 50-byte receive buffer
  - Can be used as starter code for Lab#2

- Available on the course website
STK500 UART Connection

Figure 3-10. Schematic of UART Pin Connections
Serial Peripheral Interface

- Allows transfer of information, 8 bits at a time, between microcontroller and any number of peripherals
- Read and Write operations happen simultaneously
- Using chip selects allows lots of peripherals to be connected to a single SPI bus at the same time
- Has no start/stop bit overhead
- High data rates: 250Kbit to 2Mbit/sec
- Common uses:
  - Inter-Processor Network
  - Sending MP3 data to MP3 decoders
  - Interfacing to external serial RAM/EEPROM/FLASH
  - Interfacing to serial graphic LCDs
  - Compatible with thousands of chips with SPI, Microwire, I2S, and other serial interfaces
SPI Pins and Registers

- SPI Pins
  - MOSI (master out, slave in)
  - MISO (master in, slave out)
  - SCK (serial clock)
  - SS (slave select, optional)

- SPI Registers
  - SPDR (transferred data read/write register)
  - SPCR (control register)
  - SPSR (status register)
  - SPI Transfer Complete interrupt