EE281

Embedded System
Design Laboratory

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

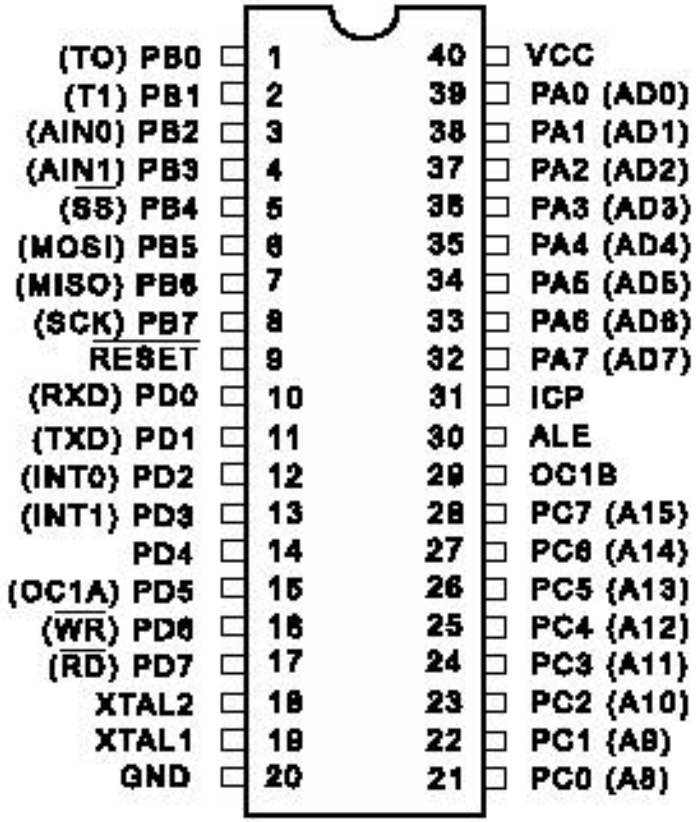
AVR Processor Resources

Embedded System
Design Laboratory

- Interrupts
- Timers
- UART (Universal Asynchronous Receiver/Transmitter)
- SPI (Serial Peripheral Interface)
- A/D Converters (Analog to Digital)
- Analog Comparator

AVR AT90S8515 Pinout

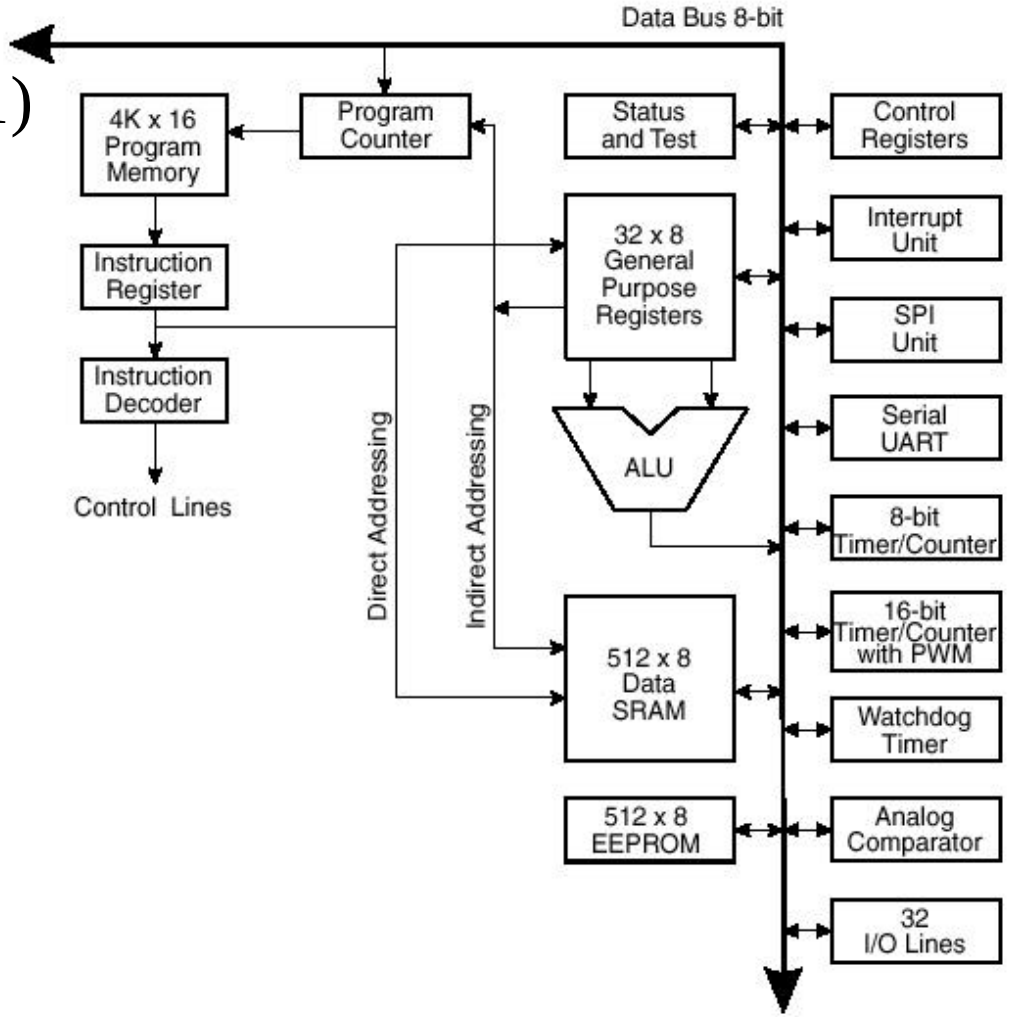
- General Purpose Ports
  - PORTA
  - PORTB
  - PORTC
  - PORTD
  - (Special Functions)
- Special Purpose Pins
  - Crystal (XTAL1/XTAL2)
  - RESET
  - ICP, OLE, OC1B
- Power (VCC/GND)



EE281 '8515 Functional Architecture

Embedded System
Design Laboratory

- 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



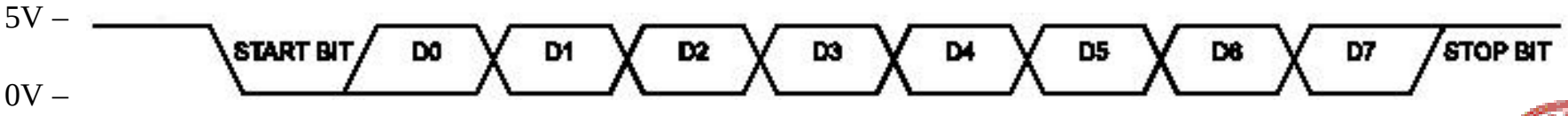
UART

- 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

EE281 Serial Specification (RS-232C)

Embedded System
Design Laboratory

- 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)



UART Registers

- 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

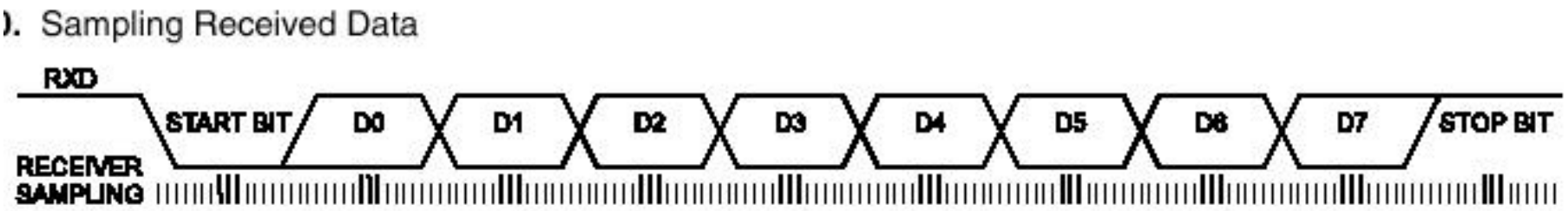
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	page 43
$0C ($2C)	UDR	UART I/O Data Register								page 47
$0B ($2B)	USR	RXC	TXC	UDRE	FE	OR	-	-	-	page 48
$0A ($2A)	UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	page 48
$09 ($29)	UBRR	UART Baud Rate Register								page 50

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

UART Receiving

- 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



UART Baud Rate

- Set by UBRR
- Varies with f_{CK}

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

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

Baud Rate	3.2768 MHz		%Error	3.6864 MHz		%Error	4 MHz		%Error
2400	UBRR=	**84**	**0.4**	UBRR=	**95**	**0.0**	UBRR=	**103**	**0.2**
4800	UBRR=	**42**	**0.8**	UBRR=	**47**	**0.0**	UBRR=	**51**	**0.2**
9600	UBRR=	**20**	**1.6**	UBRR=	**23**	**0.0**	UBRR=	**25**	**0.2**
14400	UBRR=	**13**	**1.6**	UBRR=	**15**	**0.0**	UBRR=	16	2.1
19200	UBRR=	10	3.1	UBRR=	**11**	**0.0**	UBRR=	**12**	**0.2**
28800	UBRR=	**6**	**1.6**	UBRR=	**7**	**0.0**	UBRR=	8	3.7
38400	UBRR=	4	6.3	UBRR=	**5**	**0.0**	UBRR=	6	7.5
57600	UBRR=	3	12.5	UBRR=	**3**	**0.0**	UBRR=	3	7.8
76800	UBRR=	2	12.5	UBRR=	**2**	**0.0**	UBRR=	2	7.8
115200	UBRR=	1	12.5	UBRR=	**1**	**0.0**	UBRR=	1	7.8

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

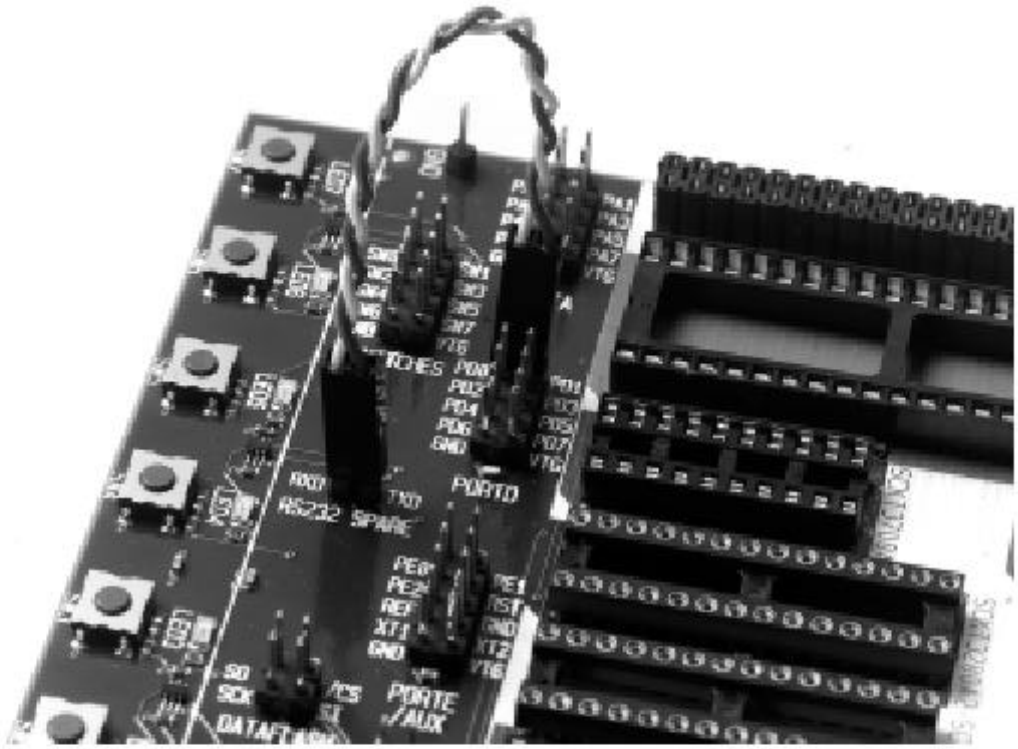
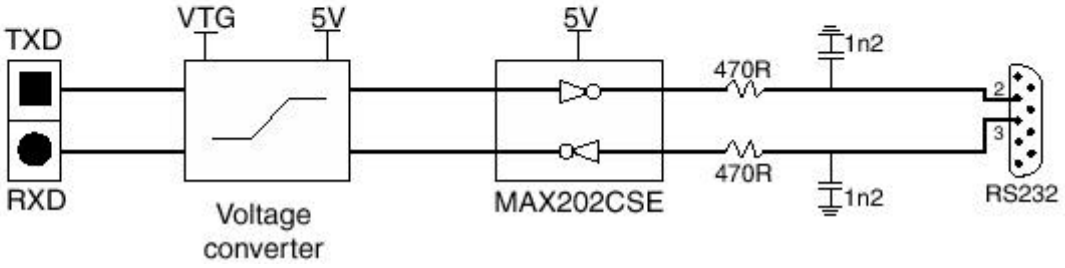


Figure 3-10. Schematic of UART Pin Connections



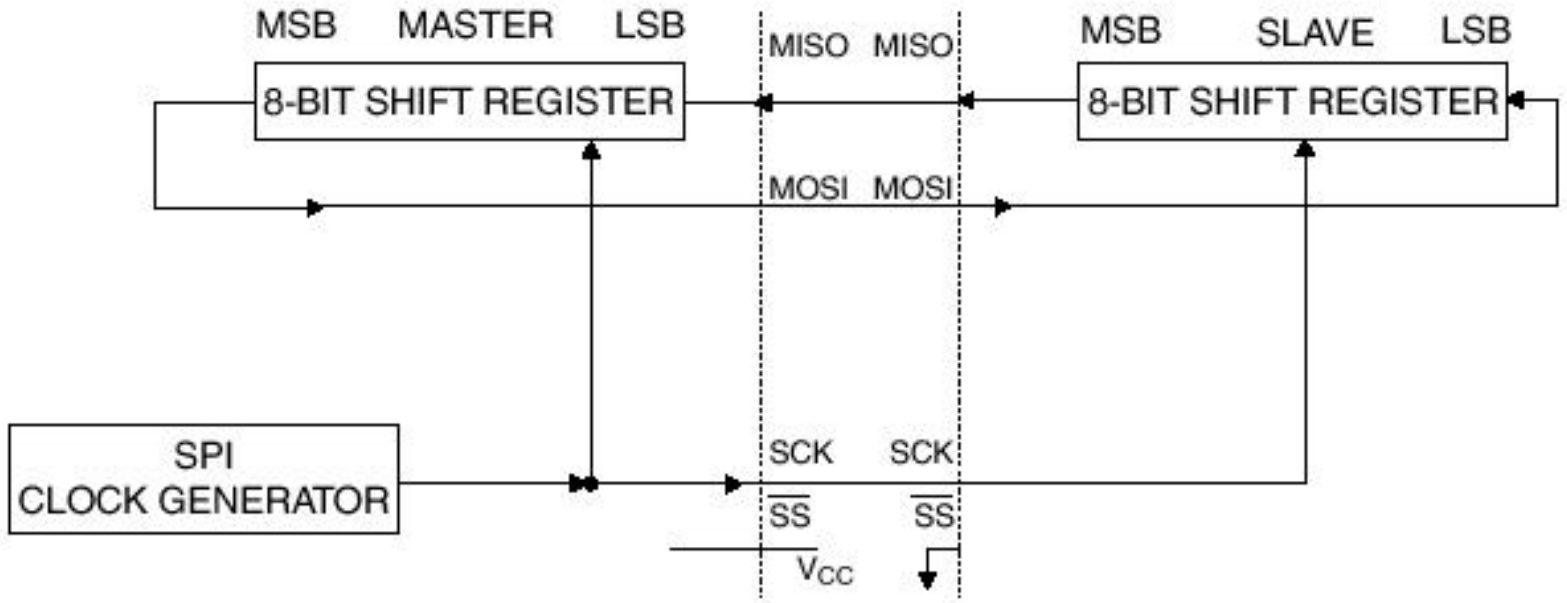
Serial Peripheral Interface

- 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 Connections

AVR Microcontroller ← → PERIPHERAL

SPI Master-slave Interconnection



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