Intro to Microcontrollers

Class 5: Serial I/O and Interrupts

October 20, 2008

Outline

Review and Today's Setup

Serial and the AVR

Interrupts

Outline

Review and Today's Setup

Serial and the AVR

Interrupts

Review

Show and Tell

- Anyone make anything cool they want to show?

ADC

- Learned (a bit) about LDR and voltage divider
- Set up the pin for ADC input
- Aim the multiplexer at our pin
- Enable the ADC
- Read the values out of the ADC register (optionally waiting for a conversion to finish)
- Then do whatever with the values

Today

Serial Communication

- Basic setup and use
- Great for debugging
- But also can connect one device to another:
  Dual AVRs? AVR + GPS? AVR + anything?

Interrupts

- Just functions that get called (automatically) when a certain condition is true
- Pin-change interrupts, overflow interrupts, timer-driven interrupts (next class), etc
- Allow for a nice workflow for robot code:
  main loop & interrupt

Outline

Review and Today's Setup

Serial and the AVR

Interrupts

Serial is Simple

Built-in hardware stuffs

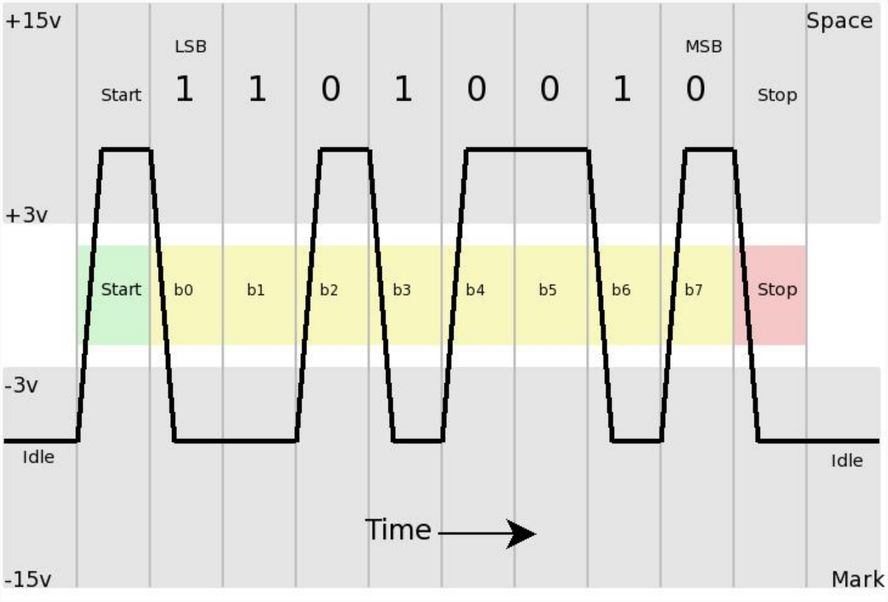
- AVR has built-in hardware serial
- Configure, then read and write bytes to a register
- Serial to USB is a bit trickier, but there's help
- Or if you're lucky, you have an old computer with a serial port – will need to do some level-shifting, but it's all good.

Serial Protocols

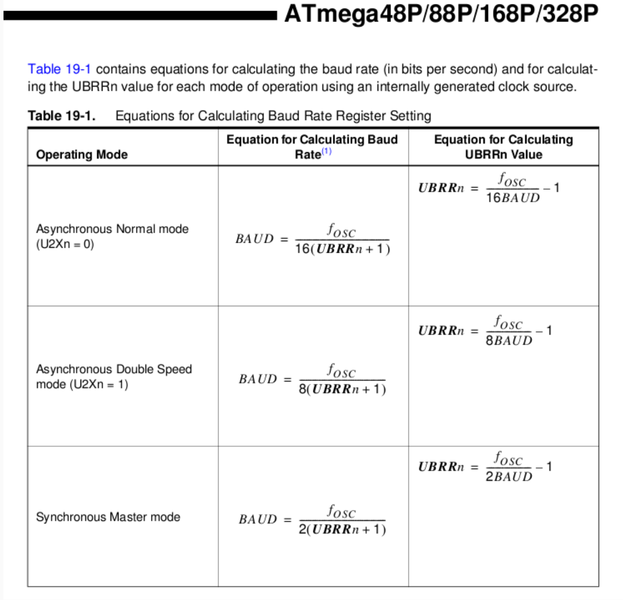
What's going on under the hood?

- Two main kinds: synchronous and asynchronous
- Async only uses data lines (no clock): USART, RS-232
- Changes in voltage on the data line tell the computer/AVR when to read the bit value...
- ...as long as the two agree on a speed of communication (baud rate)
- Bytes start with a "start bit" that signals the beginning of the transmission
- RS-232 "standard": -12v to 12v voltage, -12 is 1, 12 is 0, RX, TX, and a bunch of random other wires: DTR, DSR, CTS, DCD, RI.
- USART: 0v to 5v. 5v is 1. Can run on just GND, RX, TX.

Wikipedia on Serial



Setting Baud Rate Sucks



ATmega48P/88P/168P/328P

Table 19-1 contains equations for calculating the baud rate (in bits per second) and for calculating the UBRRn value for each mode of operation using an internally generated clock source.

Table 19-1. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRRn Value
Asynchronous Normal mode (U2Xn = 0)	$BAUD = \frac{f_{OSC}}{16(\mathbf{UBRR}n + 1)}$	$\mathbf{UBRR}n = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed mode (U2Xn = 1)	$BAUD = \frac{f_{OSC}}{8(\mathbf{UBRR}n + 1)}$	$\mathbf{UBRR}n = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master mode	$BAUD = \frac{f_{OSC}}{2(\mathbf{UBRR}n + 1)}$	$\mathbf{UBRR}n = \frac{f_{OSC}}{2BAUD} - 1$

Setup on AVR

Mildly annoying, here's a shortcut

- Need to calculate baud rate timing, two optional pre-scalers (8, 16)
- Routine on pages 179-183 of the datasheet. Mildly annoying.
- #include ”USART48.h”, set BAUDRATE macro variable
- void initUART (void)
  void transmitByte (uint8_t data)
  uint8_t receiveByte (void)
  void sayOK(void)

Hookup to Computer

Serial-USB Cable

- Max232 is a hardware serial-to-USB chip
- There's one built into that black cable
- Reads in our (0v to 5v) serial data, then transmits the result in USB
- (Additionally takes care of much USB overhead)
- Three wires: GND, RX, TX
- Here's my pet peeve: RX pin on the AVR to TX pin on the cable, vice-versa

Look at The Code

Sections

- Includes and defines (note BAUDRATE and "USART48.h")
- Main loop
- Init section
- Infinite while()

Outline

Review and Today's Setup

Serial and the AVR

Interrupts

Interrupts

What are they?

- General purpose interruptions in the program flow
- Use alone for pin-change interrupts
- Use with serial input to complete action on keypress
- Use in conjunction with timers (next class) for PWM, driving servo motors, bleeps and bloops, other time-critical code
- See page 57, 58 for all of 'em

Serial Buffer Interrupts

Setup

- See pages 195, 196 for details on the serial registers
- Turn on interrupt vector for receiving data
- Turn on global interrupts
- The read data out as soon as you press a key on laptop
- (Expand by testing which key?)

Pin-change Interrupts

Setup

- See p. 71 for setup: EICRA
- Tell AVR which interrupts you want enabled (p. 74)
- Write the function you want executed when the interrupt is called
- Think hard about whether you want the interrupt itself to be interruptable

The End

◂ Outline