CSE 466 Exam I **10 February 2006**

Name: ________________________________ ID#: ______________________________

Please read through the *entire* examination first! This exam was designed to be completed in 50 minutes. There are 3 problems for a total of 100 points. The point value of each problem is indicated in the table below. There will be partial credit so make sure to get to every problem.

Each problem is on a separate sheet of paper. Write you answer *neatly* in the space provided. Do not use any other paper.

Good luck.

Problem	Max Score	Score
1	30	
2	40	
3	30	
TOTAL	100	

1. Microcontrollers (30 points)

Define the following terms related to features of your Atmega16 microcontroller and provide an example of when each feature would be used. Relate your examples to the project you worked on for Labs 3 and 4, if at all possible. If not, explain why not.

a) Pulse-width modulation

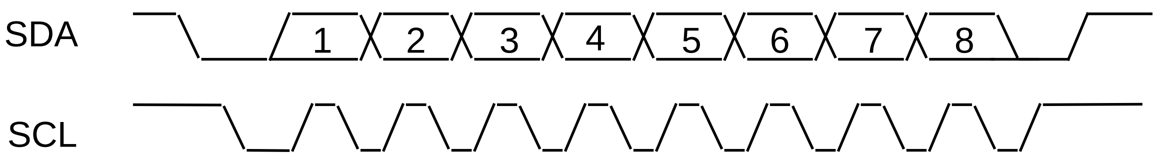
b) General-purpose I/O pins

c) SPI slave interrupt

d) Auto trigger for A-to-D conversion

2. Decoding (40 points)

Below is an example of an 8-bit packet of encoded data similar to an I2C (or TWI) serial interface. Recall that this interface has two open collector wires. Either the transmitter or receiver can hold them low.



Note that the transmitter starts the byte with a "start" falling transition on SDA while the clock (SCL) is high. This is then followed by 8 pulses on SCL corresponding to the 8 bits of data. Assume the receiver will sample the signal on the rising edge of the clock. Finally, after the 8^{th} bit, the transmitter signals a "stop" bit by holding SCL high while placing a rising edge on SDA. The receiver can slow down the transmitter by holding SCL low, thus preventing a rising transition on SCL to signal a new bit.

Show how you would write the code to receive and decode such a signal using the features of the ATmega16 microcontroller. Do not be concerned with syntax. In fact, you can use English (e.g., "read timer value"). However, you must label interrupt routines so that it is clear what condition will cause them to be executed. Also, make sure to clearly show how timers are set. Make sure to describe the timers/input-capture/output-compare/GPIO-pins/port-configurations or other devices you would use internally to the microcontroller.

a) Describe how a receiver would detect the "start" of a packet.

b) Describe how a receiver would decode each bit. What would a receiver have to do if it was very slow – and could not receive the bits as fast as the transmitter would like to send them? Your routines should call a function called "got_bit(bit_value, bit_type)". The parameters are the value of the bit just decoded and a bit type which is one of START, DATA, or STOP. For a START the value is assumed to be 0, for a STOP the value is assumed to be one, for DATA it is the value of the SDA line when the SCL line went high.

c) Describe how a receiver would detect the “stop” at the end of a packet.

d) If you assume the receiver has a clock that runs at 10MHz, what is the fastest a packet can be properly received? Explain your conclusion and clearly state your assumptions about how long each routine may take.

3. Measuring time (30 points)

You are building a sonar range finder. It will make use of a compass and an ultrasound transceiver. The compass is only an input to the microcontroller and has 4 wires, one for each of the cardinal directions. Up to two wires can be high at one time so that the compass can specify any of N, NE, E, SE, S, SW, W, NW, and W. The ultrasound range finder uses just two wires. One is an output from the microcontroller that tells the range finder to emit a sound pulse. The other is an input to the microcontroller that will be high when an echo is detected. Your device is to take a range reading every time the compass changes direction. At the completion of the reading it should call a function `new_range` with arguments of direction and time until the echo returns in timer units (don't worry about conversions), e.g., `new_range(direction, return_time)` where `direction` is the state of the compass when the distance measurement was started and `return_time` is the timer's counter value when the echo returned. Assume that sound travels at 343m/sec and that you have a 10MHz clock for your microcontroller.

a) Write pseudo-code for how you would implement this system. You should have a routine for detecting a change in the compass heading and starting an ultrasound pulse and a routine for determining the time from starting the ultrasound pulse until the sensor hears its echo.

b) How fast can a user move the compass (i.e., minimum time between changes in heading)? How does it affect the possible range of distance measurements?