(2 pts) Name:

CS [45]163: Embedded Systems: Final Exam

May 10, 2010

- This examination booklet has 15 pages.
- Do not forget to write your name at the top of the page and to sign your name below.
- The exam is opened book and notes, but is closed electronic device.
- The exam is worth a total of 200 points (and 20% of your final grade). 4163 students may waive out of doing a total of 30 points. The 30 points must be clearly marked and must be used for entire questions (i.e., no partial credit games).
- Explain your answers clearly and be concise. Do not write long essays (even if there is a lot of open space on the page). A question worth 5 points is only worth an answer that is at most 2 sentences.
- You have 2 hours to complete the exam. Be a smart test taker: if you get stuck on one problem go on to the next. Don't waste your time giving details that the question does not request. Points will be taken off for answers containing extraneous information.
- Show your work. Partial credit is possible, but only if you show intermediate steps.

Problem	Topic	Max	Grade
0	Name	2	
1	Electronics	100	
2	Interrupt Service Routines and Digital I/O	35	
3	Finite State Machines and Control	25	
4	Communication	40	
Total		202	

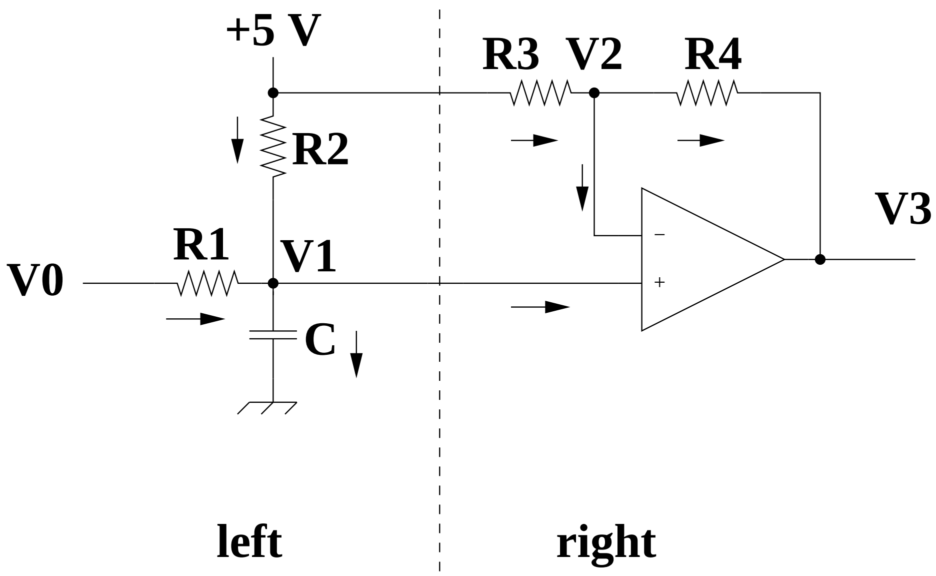
On my honor, I affirm that I have neither given nor received inappropriate aid in the completion of this exam.

Signature: ______________________________

Date: ______________________________

1. **Electronics** (100 pts)

Consider the following circuit:



(a) (5 pts) What are the fundamental equations for the left hand side of the circuit that 1) describe the current flowing along each component, and 2) describe the relationships between the different currents.

(b) (5 pts) Assume a quiescent state ($d\,V_1/d\,t = 0$). Given that $V_1 = 3V$, what is V_0? Show your derivation.

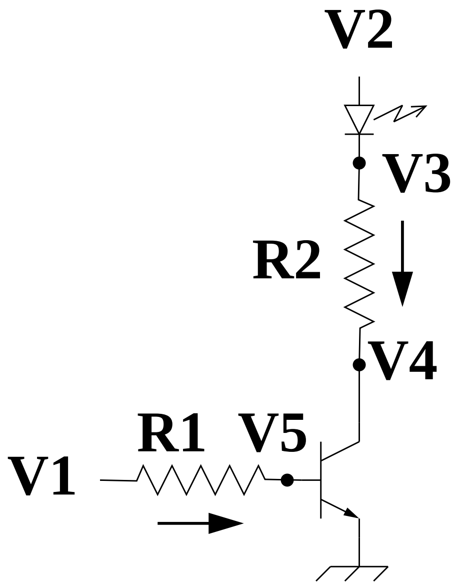
(c) (20 pts) Assume that V_0 is constant. Given a known initial state $V_1(0)$, derive the equation for $V_1(t)$ for $t > 0$. Hint: remember that the units of your time constant should still be $\Omega - F$ (which is equivalent to seconds).

(d) (5 pts) Assume $C = 1\mu F$ and $R_1 = R_2 = 100\Omega$. What is the time constant of this sub-circuit?

(e) (10 pts) What are the fundamental equations for the right hand side of the circuit?

(f) (20 pts) Derive an equation for V_3 in terms of V_1, R_3, and R_4, and assuming the ideal properties of the operational amplifier.

Consider the following circuit:



Assume that R_1, R_2, V_{fT}, V_{fD} and g are known.

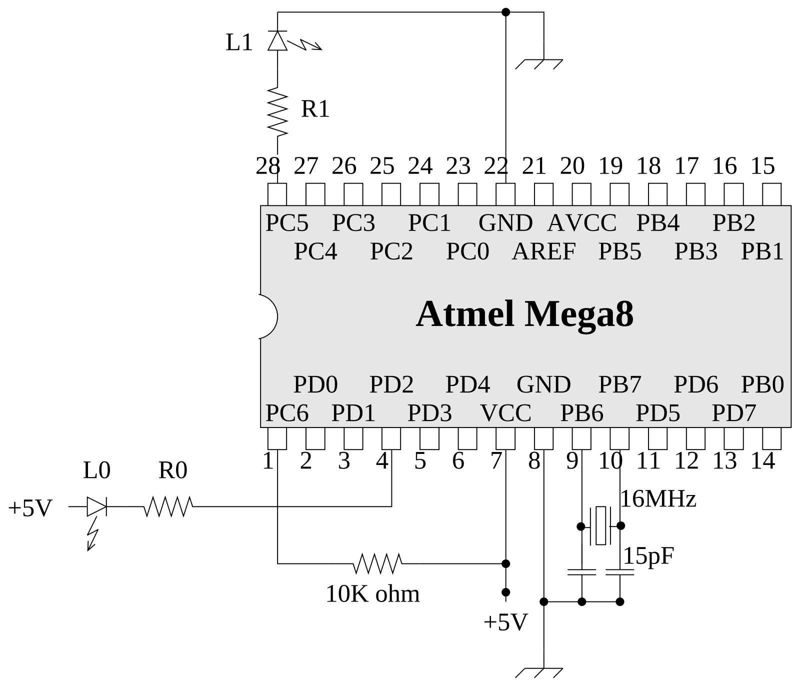
(g) (5 pts) As V_1 increases from $-1V$, at what point does the transistor begin to conduct current from base to emitter?

(h) (10 pts) What is the range of possible values for V_4 assuming that $I_{CE} > 0$?

(i) (20 pts) As V_1 increases from the point at which current begins to conduct, I_{CE} also increases. At what V_1 does I_{CE} reach a maximum? What is the maximum?

2. **Interrupt Service Routines and Digital I/O** (35 pts)

Carefully consider the following circuit:



Our goal is to write an ISR and supporting code that will do the following:

(a) Pulse LED $L0$ with a pulse-width-modulated signal with a variable period and a fixed duty cycle of 25%.

(b) Pulse LED $L1$ with a pulse-width-modulated signal with twice the period of $L0$ and with a fixed duty cycle of 50%.

(c) When the period is changed (by the main program), the ISR should never produce a period that is longer than the max of the old and new periods (we call this behavior "glitch free").

(d) The ISR may assume that $period > 0$.

The following is a first attempt at implementing this code:

```
volatile uint16_t period = 0;

ISR(TIMER2_OVF_vect) {
  static uint16_t counter = 0;

  if(counter == period)
     counter = 0;

  if(counter == 0) {
     PORTD |= 0x04;
     PORTC |= 0x40;
  }

  if(counter == (period>>3)) {
     PORTD &= ~0x04;
  }

  ++counter;
}

void set_period(uint16_t per) {
  period = per;
}

int main(void) {
  DDRD = 0x04;
  DDRC = 0x40;
  timer2_config(TIMER0_PRE_8);
  timer2_enable();
  sei();

  while(1) {
     set_period(100);
     delay_ms(2000);
     set_period(1000);
     delay_ms(2000);
  }
}
```

(a) (5 pts) Assuming a system clock of $16MHz$, at what frequency is the timer 2 counter incrementing?

(b) (5 pts) At what frequency is the timer 2 overflow interrupt being generated? (yes, work out the long division by hand)

(c) (5 pts) With $period = 100$, what is the intended flashing frequency of $L0$?

(d) (20 pts) There are several bugs in the code implementation, correct these (do this on the previous page).

3. **Finite State Machines and Control** (25 pts)

Consider our problem of the robot dribbling a ball down the field until it reaches the far end (the goal). We will design a FSM that performs the high level control for this robot.

The actions are as follows:

- Search ball (Search). This is a generic action that takes a variety of strategies to find the ball (don't worry about which)
- Move to location behind ball (Behind)
- Approach ball (Approach) (as in the *Motion I* milestone)
- Kick
- Celebrate

The robot sensor system has the following information available:

- Absolute position of the ball on the field
- Pan and tilt orientations of the neck
- Absolute orientation of the base (the CREATE)
- Size of the ball "blob" in the image (which could be zero, indicating "not found")

Other notes:

- At any time, the robot may lose visual track of the ball. The FSM must take this into account by initiating a new search.

(a) (10 pts) List the discrete events that you will need for your FSM. For each, describe in detail the relationship between the event and the available sensory data.

(b) (15 pts) Draw the FSM

4. **Communication** (40 pts)

(a) (10 pts) List the similarities of the I^{2}C and SPI protocols.

(b) (10 pts) The SPI and RS232-C protocols share one common feature that is not shared with the I^{2}C protocol. What is it?

(c) (10 pts) True or false and explain why or why not: the SPI protocol requires a *start bit* at the beginning of a data frame.

(d) (10 pts) Explain the utility of using *mixed transactions* in I^2C over simple sequences of *read* and *write* transactions.