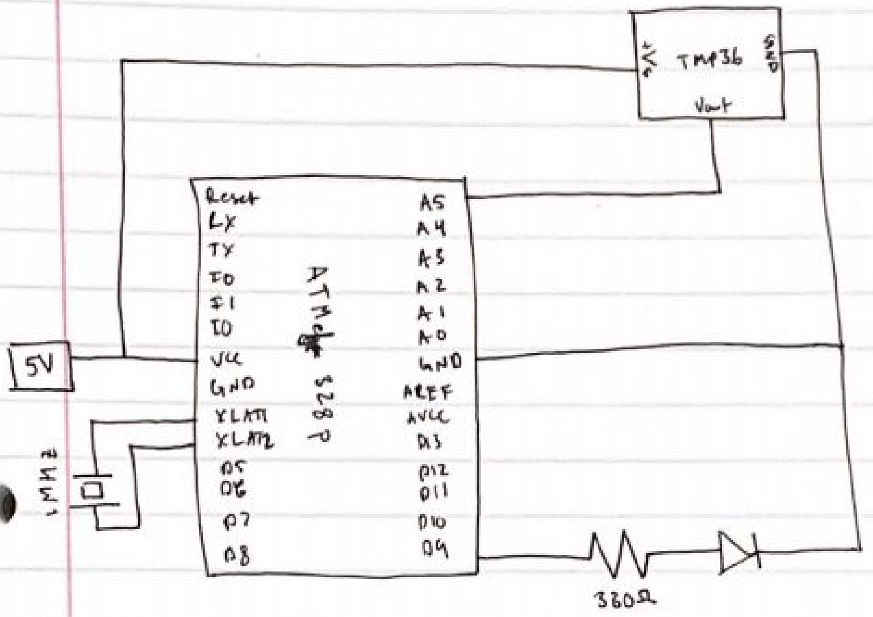
Kiersten Page
Professor Campisi
Exam One

Question One

① a)



```
b) ADMUX= 0x45; //ADC5 is channel input
   ADCSRA= 0xA4; // 16 prescaler
   ADCSRB= 0x00; // clears and assigns B to free-running mode
   DIDR0|= (1<<5); // enable digital input disable for ADC5
   ADCSRA|= (1<<6); // start conversion
c) TCCR0A= 0x 23; // Fast PWM, clear on compare match
   TCCR0B= 0x0D; // Fast PWM mode
   dc= 50;
d) temp= (((ADCH<<2) + ((ADCL & 0xC0) >>6)) -500)/10
   if (temp <0){ OCR0A = 488;
        OCR0B = (OCR0A × dc/100); }
   elif (temp >50) { OCR0A = 1;
        OCR0B = (OCR0A × dc/100); }
   else { OCR0A = map(temp, 0, 0x3FF, 1, 488);
        OCR0B = (OCR0A ×dc/100); }
```

Question Two

```
#define PINB unsigned char * 0x23
#define PINC unsigned char * 0x26
#define PIND unsigned char * 0x29
#define PORTB unsigned char * 0x25
#define PORTC unsigned char * 0x28
#define PORTD unsigned char * 0x2B
#define DDRB unsigned char * 0x24
#define DDRC unsigned char * 0x27
#define DDRD unsigned char * 0x2A

WriteGPIOhigh(int port, unsigned char mask){
//port = 0 means PortB, 1 = PortC, 2 = Port D
//mask = 8 bit mask indicating which pins to write high. Ex) 0001 0010 = Write bit 1 and 4 high
  if(port == 0){
    *PINB |= mask;
  }
  else if(port == 1){
    *PINC |= mask;
  }

  else if(port == 2){
    *PIND |= mask;
  }
}

WriteGPIOlow(int port, unsigned char mask){
//port = 0 means PortB, 1 = PortC, 2 = Port D
//mask = 8 bit mask indicating which pins to write low. Ex) 0010 0001 = Write bit 0 and 5 low
  if(port == 0){
    *PINB &= ~mask;
  }
  else if(port == 1){
    *PINC &= ~mask;
  }
  else if(port == 2){
    *PIND &= ~mask;
  }
}

EnableGPIOpullup(int port, unsigned char mask){
  //port = 0 means PortB, 1 = PortC, 2 = Port D
  //mask = 8 bit mask indicating which pins to enable pullup
  if(port == 0){
    *PORTB |= mask;
  }
  else if(port == 1){
    *PORTC |= mask;
  }
  else if(port == 2){
    *PORTD |= mask;
  }
}

unsigned char readGPIOport(int port){
  //port = 0 means PortB, 1 = PortC, 2 = Port D
  if(port == 0){
    return (unsigned char)*PORTB
  }
  else if(port == 1){
    return (unsigned char)*PORTC
  }

  else if(port == 2){
    return (unsigned char)*PORTD
  }
}

/* The above is not enough to completely implement GPIO functionality. In order to do so,
* a function must be created in order to indicate which pins are inputs and outputs.
* As of right now, the functions can only write a pin high or low, return the pins of a port
* and indicate which pins to pullup.
*/
```