Embedded Real-Time Systems (AME 3623)
Homework 3 Solutions

May 2, 2011

Question 1

Assume that *student_ID* is the number that corresponds to your student ID number.

1. (2pts) What is *student_ID % 4*? Call this *key1*

   Possible answers are: 0, 1, 2, 3

2. (2pts) What is *student_ID % 5*? Call this *key2*

   Possible answers are: 0, 1, 2, 3, 4

Question 2

Assume the timer/counter equal to your key1.

Assume a prescaler of 1 (if key2 == 0), 8 (key2 == 1), 64 (key2 == 2), 256 (key2 == 3) or 1024 (key2 == 4).

1. (5 pts) What is the frequency of counting of the timer/counter?

   $t\ (tocks/sec) = \frac{16{,}000{,}000\ ticks/sec}{p\ ticks/tock}$

   So:

key2	p	t
0	1	16 MHz
1	8	2 MHz
2	64	250 KHz
3	256	62.5 KHz
4	1024	15.625 KHz

2. (5 pts) Assume that we have the overflow interrupt enabled. What is the period between overflow interrupts?

   $f\ (sec/int) = \frac{p\ ticks/tock \times x\ tocks/int}{16{,}000{,}000\ ticks/sec}$

   So:

key1	key2	x	p	f
0,2	0	256	1	16 μs
0,2	1	256	8	128 μs
0,2	2	256	64	1024 μs
0,2	3	256	256	4096 μs
0,2	4	256	1024	16.384 ms
1,3,4	0	256^2	1	4.096 ms
1,3,4	1	256^2	8	32.768 ms
1,3,4	2	256^2	64	262.144 ms
1,3,4	3	256^2	256	1.0486 s
1,3,4	4	256^2	1024	4.1943 s

Question 3

Suppose that we want to produce an overflow interrupt frequency of $488Hz$. Assume that we are using a 16 MHz crystal for our clock.

1. (5 pts) Which timer should we use?

   Timer 2.

2. (5 pts) Which prescaler should we use?

   Prescaler: 128

Question 4

1. (15pts) Suppose that we want a function – called *control*() – to be executed approximately once every second, and another function – called *sense*() – to be executed approximately once every 5 minutes. We will use the timer1 overflow interrupt to call both of these. Assume a system clock of $16MHz$. What is the timer1 prescaler configuration and the code for the interrupt routine (the code does not need to be syntactically correct)? Also - show the code in your main function that configures the timer.

   We will use a prescaler of 256. This gets us down to an interrupt every $1.0486s$*, which is close to the desired control frequency (within* 5%*). We then need an interrupt routine with an additional counter that expires at 286. So, we are left with an interrupt interval of:* $286 * 256 * 256 * 256/16000000 = 299.8927s \approx 5min$*.*

```
ISR(TIMER1_OVF_vect) {
    static uint16_t counter = 0;    // Must be 16 bits

    // Execute the control function each time
    control();

    ++counter;
    if(counter == 286) {
      // Execute this function only once out of every 286 interru
      sense();
      counter = 0;
    };
};

Somewhere in the main program:
// Interrupt occurs every
//      (256*256*256)/16000000 = 1.0486 sec
timer1_config(TIMER1_PRE_256);
// Enable the timer interrupt
timer1_enable();
// Enable global interrupts
sei();
```

Question 5

Consider the following code:

```
ISR(TIMER1_OVF_vect) {
    static uint8_t counter = 0;
    static uint8_t phase = 0;

    if(counter == 0) {
        switch(phase) {
            case 0:
                PORTC = PORTC & 0xFC | 1;
                counter = 75;
                phase = 1;
                break;
            case 1:
                PORTC = PORTC & 0xFC | 2;
                counter = 100;
                phase = 2;
                break;
            case 2:
                PORTC = PORTC & 0xFC;
                counter = 25;
                phase = 0;
                break;
        }
    }
    --counter;
};

Somewhere in the main program:

// Initialization
timer1_config(TIMER1_PRE_64);
// Enable the timer interrupt
timer1_enable();
// Enable global interrupts
sei();

DDRC = 0x3;
PORTC = 0;

while(1)
{
}
```

1. (15 pts) Explain in detail what the program does. You are welcome to provide a picture.

   This program produces two PWM signals at a frequency of $0.0191 Hz$ *on pins C0 and C1.*

   $$f(cycle/sec) = \frac{16{,}000{,}000\ ticks/sec}{64 ticks/tock \times 256^2 tock/int \times (75+100+25)\ int/cycle}$$

   C0 is high for the first $19.66s$ *(a duty cycle of* 37.5%*). After C0 is turned off, C1 is high for* $26.21s$ *(a duty cycle of* 50%*). Then, both are off for* $6.55s$