Jose Franco Baquera

April 14, 2018

CS 273

Homework #10 (FINAL VERSION)

1. **If the input clock source operates at 16 MHz, what is the TCNT1 value needed to generate a time delay of 0.5 millisecond in normal mode?**

Solution:

T = 0.5 ms = 500 µs
1/16MHz = 0.0625 µs
n = 500 µs / 0.0625 µs
n = 8000

Timer 1 is a 16-bit timer so use 2^{16}
2^{16} – 8000 = 65536 – 8000 = 57,536
The TCNT1 value needed is: **TCNT1 = 57,536**

2. **If the clock operates at 10 MHz, what is the OCR0 value needed to generate time delay of 4 microseconds in CTC mode without any prescaler.**

Solution:

(NOTE: We must subtract 1 from the final answer.)

T = 1/10MHz = 0.1 µs
OCR0 = (4 µs/ 0.1 µs) – 1 = 40 – 1 = 39 clock cycles
The OCR0 value that we need to generate a time delay of 4 microseconds is: **OCR0 = 39**

3. **Write an assembly program for Timer1 to generate a square wave of 16000 Hz to output on PB3 using a clock frequency of 16 MHz.**

NOTE: There are two versions of this answer. The first version does not use CTC mode while the second version (located on the third page) uses the CTC mode version.

****

First Solution (NOT USING CTC MODE):

For a square wave of 16000 Hz = 0.016MHz:

Square wave period: T = 1/f = 1/0.016 = 62.5 µs
62.5/2 = 31.25 (we divide by 2, square wave has 31.25 µs high and 31.25 µs low)

clock frequency = 16MHz
clock period T = 1/16 = 0.0625

n = 31.25/0.0625 = 500 clock cycles

TCNT1 = 65,536 – 500 = 65,036 = 0xFE0C
TCNT1H = 0xFE
TCNT1L = 0x0C

```
.set PORTB, 0x05
.set DDRB, 0x04
.set TIFR1, 0x16
.set TOV1, 0
.set TCCR1A, 0x80
.set TCCR1B, 0x81
.set TCNT1H, 0x85
.set TCNT1L, 0x84

.data
# No data segment for this program
.text
.global generateSquareWave

generateSquareWave:
      LDI R16, 0x08
      SBI DDRB, 3  ;set PB3 for output
      LDI R17, 0
      OUT PORTB, R17
BEGIN:
      LDI  R20, 0xFE
      OUT  TCNT1H, R20      ;load timer 1 high byte
      LDI  R20, 0x0C
      OUT  TCNT1L, R20      ;load timer 1 low byte
      LDI  R20, 0x00
      OUT  TCCR1A, R20      ;normal mode
      LDI  R20, 0x01
      OUT  TCCR1B, R20      ;internal clock for normal mode
AGAIN:
      IN  R20, TIFR1        ;read TIFR1
```

```
      SBRS  R20, TOV1       ;if TOV1 is set skip next instruction
      RJMP  AGAIN
      LDI R20, 0x00
      OUT  TCCR1B, R20      ;stop the timer
      LDI R20, 1 ≪ TOV1
      OUT TIFR1, R20 ; clear TOV1 flag
      EOR R17, R16   ; Toggle bit 3
      OUT PORTB, R17 ; Toggle PB3
      RJMP BEGIN
```

****

Second Solution (USING CTC mode):

```
.data
# No data segment for this program
.text
.global generateSquareWave

generateSquareWave:
      LDI R16, 0x08
      SBI DDRB, 3  ;set PB3 for output
      LDI R17, 0
      OUT PORTB, R17
      LDI R20, HI8(500-1)
      LDI R19, LO8(500-1)
      OUT OCR1AH, R20 ;load timer1 high byte
      OUT OCR1AL, R19 ;load timer1 low byte

BEGIN:
      LDI  R20, 0x01
      OUT  TCCR1A, R20      ;CTC mode
      LDI  R20, 0x01
      OUT  TCCR1B, R20      ;CTC mode, int clk

AGAIN:
      IN  R20, TIFR1        ;read TIFR1
      SBRS  R20, OCF1A ; If OCF1A is set skip next step
      RJMP  AGAIN
      LDI R20, 0x00
```

```
OUT  TCCR1B, R20      ;stop the timer
LDI R20, 1 ≪ OCF1A
OUT TIFR1, R20  ; clear OCF1A flag
EOR R17, R16   ; Toggle bit 3
OUT PORTB, R17 ; Toggle PB3
RJMP BEGIN
```

****

4. **<u>What is the advantage of using falling/rising edges instead of high/low voltages to detect events?</u>**

<u>Solution:</u>

The advantage of using falling and rising edges instead of low and high voltages to detect events is that we can send out an impulse all while keeping the voltage high or low for a specific amount of time when detecting events. In contrast, high and low voltage shifts happen quickly, meaning that we might not be able to detect events in time. Because of this, it is better to use rising and falling edges since we ensure that the computer will have enough time to both detect and act upon events. In addition, the advantage of using falling/rising edges is that we can detect every change in an event.

5. **<u>What is the signal the following code generates at PORTB pin 5? If it is a periodical signal, what is its period? The clock frequency is 16MHz.</u>**

(NOTE: The answer for this question is found on the next page.)

```
LDI  R16,0x20
  SBI  DDRB,5
  LDI  R17,0
  OUT  PORTB,R17
  LDI  R20,0x00
  OUT  TCCR0A,R20
BEGIN:  LDI  R20,0x3B
  OUT  TCNT0,R20
  LDI  R20,0x01
  OUT  TCCR0B,R20
AGAIN:  IN  R20,TIFR0
  SBRS  R20,0
```

```
RJMP  AGAIN
LDI  R20,0x0
OUT  TCCR0B,R20
LDI  R20,0x01
OUT  TIFR0,R20
EOR  R17,R16
OUT  PORTB,R17
RJMP   BEGIN
```

Solution:

0x3B = 59
Therefore, since this is Timer/Counter 0: 256 – 59 = 197 cycles

Clock cycle = 1/16MHz = 0.0625 µs
197 x 0.0625 = 12.3125 µs

12.3125 µs represents the high part of the wave, so we need another 12.3125 µs for the low part of the wave. Therefore, square wave 2 * 12.3125 = 24.625 µs.

Assume T = period and f = frequency. Therefore:
T = 1/f => f = 1/T => 1/24.625 µs = 0.04060914

The previous code generates a signal with a frequency of **0.04060914 MHz** (or 40609.14 Hz) and with a period of **24.625 µs.**