HW 2: Timer Functions, PWM, Stepper Motor Control

Souradeep Bhattacharya
EE128 Section: 001

November 13, 2017

Q1 Timer Functions

```
#define DELAY 3

unsigned long uptime_milli;
unsigned long uptime_micro;
unsigned long lastTime;
unsigned short overflows;

__interrupt void TC2_ISR(void)
{
    TC2 += DELAY;
    uptime_micro += 1;

    // We overflowed.
    if(lastTime > uptime_micro)
    {
        overflows += 1;
    }
    lastTime = uptime_micro;
}

typedef void (*near tIsrFunc)(void);
const tIsrFunc _vect @0x3E64 = TC2_ISR;

void timer_init(void)
{
    // Set all initial values.
    uptime_milli = 0;
    uptime_micro = 0;
    lastTime = 0;
    overflows = 0;

    TSCR1  = 0x90;          // Enable TCNT and fast timer flag clear
    TSCR2  = 0x03;          // Set Prescaler for 8;
    TIOS   |= (1 << 2);     // Enable OC2
    TCTL2  = 0x40;          // Select Toggle for output on pin 2
    TC2    = TCNT + DELAY;  // Set toggle time 1 micro in the future
    TIE    |= (1 << 2)      // Enable TC2 Interrupt
    EnableInterrupts;
}

unsigned long millis(void)
{
    uptime_milli = overflows * 4294937;
    uptime_milli += uptime_micro / 1000;

    return uptime_milli;
}

unsigned long micro()
{
    return uptime_micro;
}
```

Q2 PWM

```
void analogWrite(int pin, int value)
{
    PWMCLK = 0;       // Select normal clock for all.
    PWMPRCLK = 0x77; // Set clock prescaler for 128 for A and B
    PWMPOL = 0xFF;    // Active high for all PWM outputs.

    // With the prescaler, 192 on the period gives us the frequency of 976Hz

    switch(pin)
    {
        default:
        // Pin 0
        case 0:
            PWMPER0 = 192;
            PWMDTY0 = value;
            PWMCNT0 = 0;
        break;

        // Pin 1
        case 1:
            PWMPER1 = 192;
            PWMDTY1 = value;
            PWMCNT1 = 0;
        break;

        // Pin 2
        case 2:
            PWMPER2 = 192;
            PWMDTY2 = value;
            PWMCNT2 = 0;
        break;

        // Pin 3
        case 3:
            PWMPER3 = 192;
            PWMDTY3 = value;
            PWMCNT3 = 0;
        break;

        // Pin 4
        case 4:
            PWMPER4 = 192;
            PWMDTY4 = value;
            PWMCNT4 = 0;
        break;

        // Pin 5
        case 5:
            PWMPER5 = 192;
            PWMDTY5 = value;
            PWMCNT5 = 0;
        break;

        // Pin 6
        case 6:
            PWMPER6 = 192;
            PWMDTY6 = value;
            PWMCNT6 = 0;
        break;

        // Pin 7
        case 7:
            PWMPER7 = 192;
            PWMDTY7 = value;
            PWMCNT7 = 0;
        break;
    }

    // Enable the pin
    PWME != (1 << pin);

}
```

Part 2

We can reduce the frequency by half if we enable the center PWM. (CAE)

Q3 Stepper Motor Control

Part A

	Coil Points							
	A1	A2	B1	B2	C1	C2	D1	D2
Step 1	+	-	-	-	-	-	-	-
Step 2	-	-	-	-	-	-	+	-
Step 3	-	-	+	-	-	-	-	-
Step 4	-	-	-	-	+	-	-	-
Step 5	-	+	-	-	-	-	-	-
Step 6	-	-	-	-	-	-	-	+
Step 7	-	-	-	+	-	-	-	-
Step 8	-	-	-	-	-	+	-	-

Part B

	Coil Points							
	A1	A2	B1	B2	C1	C2	D1	D2
Step 1	+	-	-	-	-	-	+	-
Step 2	-	-	+	-	-	-	+	-
Step 3	-	-	+	-	+	-	-	-
Step 4	-	+	-	-	+	-	-	-
Step 5	-	+	-	-	-	-	-	+
Step 6	-	-	-	+	-	-	-	+
Step 7	-	-	-	+	-	+	-	-
Step 8	+	-	-	-	-	+	-	-

Part C

	Coil Points							
	A1	A2	B1	B2	C1	C2	D1	D2
Step 1	+	-	-	-	-	-	-	-
Step 2	+	-	-	-	-	-	+	-
Step 3	-	-	-	-	-	-	+	-
Step 4	-	-	+	-	-	-	+	-
Step 5	-	-	+	-	-	-	-	-
Step 6	-	-	+	-	+	-	-	-
Step 7	-	-	-	-	+	-	-	-
Step 8	-	+	-	-	+	-	-	-
Step 9	-	+	-	-	-	-	-	-
Step 10	-	+	-	-	-	-	-	+
Step 11	-	-	-	-	-	-	-	+
Step 12	-	-	-	+	-	-	-	+
Step 13	-	-	-	+	-	-	-	-
Step 14	-	-	-	+	-	+	-	-
Step 15	-	-	-	-	-	+	-	-
Step 16	+	-	-	-	-	+	-	-