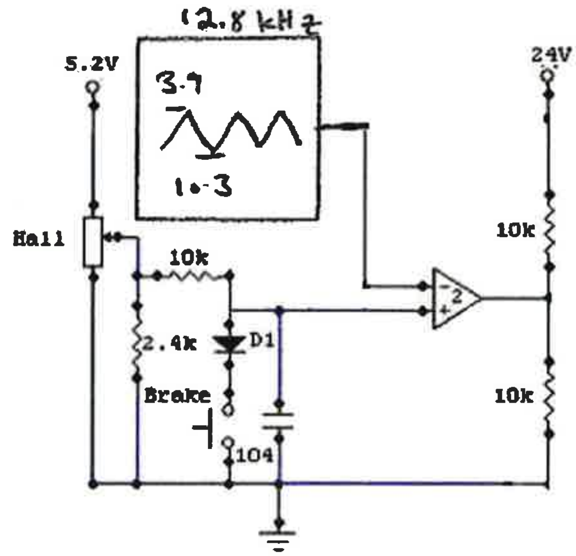
WEEK 6

Problem C, Solution

This is a PWM.



i) If the switch is open, the input to the non-inverting terminal of the comparator is just equal to the output of the Hall Effect sensor.

a) if the hall effect output is 1 V, the v+ is always less than v- so the output of the comparator is always 0.

b) if the hall effect output is 2 V , then v+ is higher than v- for (2-1.3)/(3.9-1.3)=27% of the time. So the output will vary between 0V (comparator output is short circuit to ground) and 12V (comparator output is an open circuit) with a duty cycle of 27%.

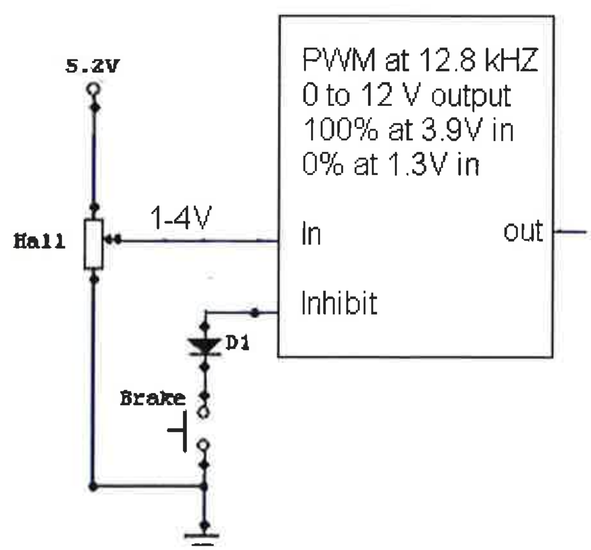
c) the duty cycle is now (3-1.3)/(3.9-1.3)=65%.

d) v+ is always greater than v-, so the duty cycle is 100% (the output is always high (12V)).

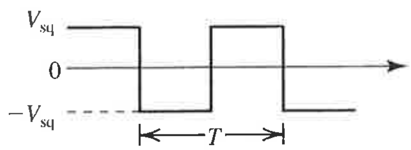
ii) In all cases v+ is 0.7 volts (using constant drop model of the diode), so v+ is always less than v-, and the output is always low (0V).

We can replace this part of the circuit with a PWM with an "inhibit" input. If we pull the inhibit line low, the output of the PWM is always low.

Note: the PWM output is low impedance for low output, but with a 5k impedance when the output is high.



1.33 If the amplitude of the square wave is V_{sq} then the power delivered by the square wave to a resistance R will be V_{sq}^2 / R. If this power is to equal that delivered by a sine wave of peak amlitude $\hat{V}$ then



$$\frac{V_{sq}^2}{R} = \frac{(\hat{V} / \sqrt{2})^2}{R}$$

Thus, $V_{sq} = \hat{V} / \sqrt{2}$. This result is independant of frequency.

1.34

Decimal	Binary
0	0
5	101
8	1000
25	11001
57	111001

1.35

b_3	b_2	b_1	b_0	Value Represented
0	0	0	0	+0
0	0	0	1	+1
0	0	1	0	+2
0	0	1	1	+3
0	1	0	0	+4
0	1	0	1	+5
0	1	1	0	+6
0	1	1	1	+7
1	0	0	0	−0
1	0	0	1	−1
1	0	1	0	−2
1	0	1	1	−3
1	1	0	0	−4
1	1	0	1	−5
1	1	1	0	−6
1	1	1	1	−7

Note that there are two possible representation of zero: 0000 and 1000. For a 0.5-V step size, analog signals in the range ± 3.5 V can be represented

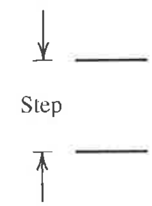
Input	Steps	Code
+2.5 V	+5	0101
−3.0 V	−6	1110
+2.7	+5	0101
−2.8	−6	1110

1.36 (a) For N bits there will be 2^N possible levels, from 0 to V_{FS}. Thus there will be $(2^N - 1)$ discrete steps from 0 to V_{FS} with the step size given by

$$\text{Step size} = \frac{V_{FS}}{2^N - 1}$$

This is the analog change corresponding to a change in the LSB. It is the value of the resolution of the ADC.

(b) The maximum error in conversion occurs when the analog signal value is at the middle of a step. Thus the maximum error is



$$\frac{1}{2} \times \text{step size} = \frac{1}{2} \frac{V_{FS}}{2^N - 1}$$

This is known as the quantization error.

(c) $\dfrac{10\ V}{2^N - 1} \leq 5$ mV

$2^N - 1 \geqslant 2000$

$2^N \geq 2001 \Rightarrow N = 11$,

For $N = 11$

$$\text{Resolution} = \frac{10}{2^{11} - 1} = 4.9 \text{ mV}$$

$$\text{Quantization error} = \frac{4.9}{2} = 2.4 \text{ mV}$$

1.37 When $b_i = 1$, the ith switch is in position 1 and a current $(V_{ref}/2^i R)$ flows to the output. Thus i_O will be the sum of all the currents corresponding to "1" bits, i.e.,

$$i_O = \frac{V_{ref}}{R}\left(\frac{b_1}{2^1} + \frac{b_2}{2^2} + \dots + \frac{b_N}{2^N}\right)$$

(b) b_N is the LSB

b_1 is the MSB

RESOLUTION OF ADC

$3.3V/1024 = 3.22mV$

TEMPERATURE RESOLUTION

VOLTAGE RESOLUTION $= 1.5/1024 = 1.46mV$

$V_{TEMP} = .00355(TEMP_c) + .986$

$TEMP_c = 281.69\, V_{TEMP} + 277.7$ IF $\Delta V = 1.46mV$

$\Delta T = 281.69 \cdot \Delta V = 0.4°$

---

pbFlag is volatile because, to compiler, it looks like it doesn't change, so it could be optimized away.

```c
#include <msp430.h>

#define P1LED BIT0
#define P1SW  BIT3
/* This program sets up P1.0 as an output (connected
 * to an LED, and P1.3 as in input with a pull up
 * resistor, connected to a switch.
 * On the falling edge of P1.3 and interrupt is generated
 * that raises a flag.
 * In the main loop, if the flag is raised, the LED is toggled.
 */
volatile int pbFlag=0;
#define LED BIT0

void main(void) {
  WDTCTL = WDTPW + WDTHOLD; // Stop watchdog

  P1OUT &= ~P1LED;          // Turn LED off.
  P1DIR |= P1LED;           // Make LED an output

  P1REN |= P1SW;            // Add resistor to switch
  P1OUT |= P1SW;            // Make it a pull-up resistor.
  P1IES |= P1SW;            // Make interrupt falling edge.
  P1IE  |= P1SW;            // Enable interrupt on switch.
  P1IFG &= ~P1SW;            // Clear interrupt flag.

__bis_SR_register(GIE);     // enable interrupts globally

  while (1){
     if (pbFlag) {
         pbFlag = 0;
         P1OUT ^= LED;
     }
  }
}

// Port 1 interrupt service routine
#pragma vector=PORT1_VECTOR
__interrupt void Port_1(void) {
    pbFlag = 1;                // Set flag to signal interrupt.
    P1IFG &= ~P1SW;            // Clear interrupt flag.
}
```