Brushed DC Motor Handout

Lab 1: Software PWM

<u>Lecture Example   Slides 38 – 43</u>

<u>Knowns:</u>

F_{OSC} = 8MHz   Pulse Width = 100µs   Period = 500 µs
F_{PWM} = 2KHz   Duty Cycle = 20%   Instruction Clock = 2MHz
Instruction Time = 0.5µs (1 instruction every 0.5 µs)

TMR0 Int. = 50 µs (Thus, the TMR0 interrupt routine will execute every 50 µs.)
Knowing that TMR0 is an 8 bit register, therefore it can only do 256 instructions before resetting back to zero.

Calculate TMR0 value at which it will start counting instructions.

TMR0 value = 256 - (number of instructions TMR0 can execute in a 50 µs.)
= 256 -100
TMR0 value = 156
TMR0 value in Hex. = 09CH

Now, we know that the signal period for this example is 500 µs and the TMR0 interrupt routine will occur every 50 µs, what number value do we put into the software PWM code to represent this?
500 µs / 50 µs = **10 = PERIOD**

This means that in our software PWM code the variable 'PERIOD' will equal 10. Therefore, this variable will count down 10 times, thus executing the TMR0 interrupt routine 10 time in one signal period(500 µs).

So, now we need the register value of the variable DUTY_CYCLE in our software PWM code.

DUTY_CYCLE value = 20% of the PERIOD value.
= (20/100) ● 10
DUTY_CYCLE value = 2
This means that in our code the motor is on 20% of the signal period.

Lab 1: Software PWM   Slides 48 – 50   Task 1

Modify the Lab 1 assembly code to create a 1.5 kHz and 30% duty cycle software PWM.

Knowns:

F_{OSC} = 8MHz   Pulse Width = 100µs F_{PWM} = 1.5KHz
Duty Cycle = 30%   Instruction Clock = 2MHz

$Period_{SIGNAL} = 1 / F_{PWM}$

Solve for $Period_{SIGNAL}$:_________

Solve for TMR0 value:_________

Solve for PERIOD value:_________

Solve for DUTY_CYCLE value:___________

Lab 1: Software PWM Slides 48 – 50 Task 1

Solution

Modify the Lab 1 assembly code to create a 1.5 kHz and 30% duty cycle software PWM.

Knowns:
F_{OSC} = 8MHz Pulse Width = 100µs F_{PWM} = 1.5KHz
Duty Cycle = 30% Instruction Clock = 2MHz

$Period_{SIGNAL}$ = 1 / F_{PWM} = 1 / (1.5KHz)
= 1/ (1.5 ● 1000)
$Period_{SIGNAL}$ = 667 µs

Instruction Time = 0.5µs (1 instruction every 0.5 µs)

Select a new value for TMR0 Int.
TMR0 Int. = 66.5 µs (Thus, the TMR0 interrupt routine will execute every 66.5 µs.) (This value was selected through trial/error and estimation).
Knowing that TMR0 is an 8 bit register, therefore it can only do 256 instructions before resetting back to zero.

Calculate TMR0 value at which it will start counting instructions.
TMR0 value = 256 - (number of instructions TMR0 can execute in a 66.5 µs.)
= 256 -100
TMR0 value = 156
TMR0 value in Hex. = 09CH

Now, we know that the signal period for this example is 667 µs and the TMR0 interrupt routine will occur every 66.5 µs, what number value do we put into the software PWM code to represent this?
667 µs / 66.5 µs = **10 = PERIOD value.**

This means that in our software PWM code the variable 'PERIOD' will equal 10. Therefore, this variable will count down 10 times, thus executing the TMR0 interrupt routine 10 time in one signal period(667 μs).

So, now we need the register value of the variable DUTY_CYCLE in our software PWM code.
DUTY_CYCLE value = 30% of the PERIOD value.
= (30/100) ● 10
DUTY_CYCLE value = 3

This means that in our code the motor is on 30% of the signal period.

Another Solution

Knowns:
F_{OSC} = 8MHz   Pulse Width = 100μs F_{PWM} = 1.5KHz
Duty Cycle = 30%   Instruction Clock = 2MHz
Instruction Time = 0.5μs (1 instruction every 0.5 μs)

$Period_{SIGNAL}$ = 1 / F_{PWM} = 1 / (1.5KHz)
= 1/ (1.5 ● 1000)
$Period_{SIGNAL}$ = 667 μs

Leave this value the same:
TMR0 Int. = 50 μs (Thus, the TMR0 interrupt routine will execute every 50 μs.)
Knowing that TMR0 is an 8 bit register, therefore it can only do 256 instructions before resetting back to zero.

Calculate TMR0 value at which it will start counting instructions.

TMR0 value = 256 - (number of instructions TMR0 can execute in a 50 μs.)
= 256 -100
TMR0 value = 156
TMR0 value in Hex. = 09CH

Now, we know that the signal period for this example is 667 µs and the TMR0 interrupt routine will occur every 50 µs, what number value do we put into the software PWM code to represent this?
667 µs / 50 µs = 13.34 ≈ **13 = PERIOD value.**

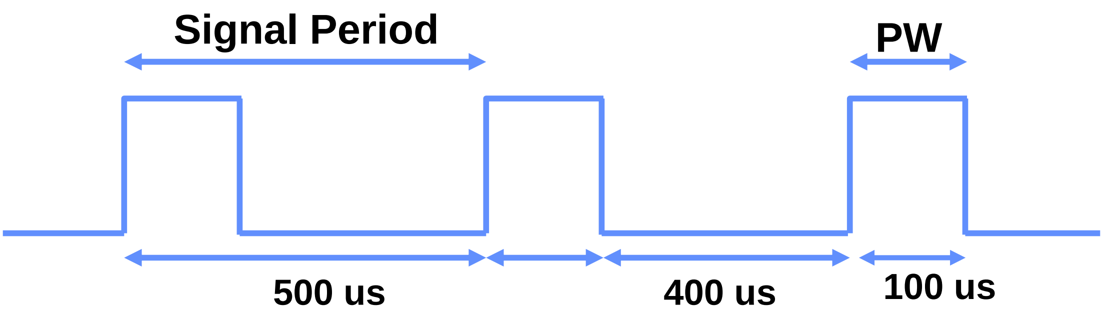
This means that in our software PWM code the variable 'PERIOD' will equal 13. Therefore, this variable will count down 13 times, thus executing the TMR0 interrupt routine 13 time in one signal period(667 µs).

So, now we need the register value of the variable DUTY_CYCLE in our software PWM code.

DUTY_CYCLE value = 30% of the PERIOD value.
= (30/100) ● 13
DUTY_CYCLE value = 4

This means that in our code the motor is on 30% of the signal period.

Lab 2: Hardware PWM



Equations:

$T_{OSC} = 1 / F_{OSC}$
$\text{Period}_{SIGNAL} = 1 / F_{PWM}$
$\text{Pulse Width (PW)} = \text{Period} \bullet \text{Duty Cycle}$

$\text{Period}_{PWM} \text{ (Seconds)} = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (\text{TMR2 Prescaler Value})$

$\text{Pulse Width}_{PWM} \text{ (PW)} = (CCPR1L:CCP1CON<5:4>) \bullet T_{OSC} \bullet (\text{TMR2 Prescaler Value})$

$$\text{Duty Cycle}_{PWM} \text{ (Ratio)} = \frac{(CCPR1L:CCP1CON<5:4>)}{[4 (PR2 + 1)]}$$

$$\text{Resolution}_{PWM} \text{ (Bits)} = \frac{\log [4 (PR2 + 1)]}{[\log (2)]}$$

Lab 2: Hardware PWM

Lecture Example   Slides 52 – 56

Determine the PR2 value for a F_{PWM} of 2 kHz and a prescaler of 16;

Knowns:

F_{OSC} = 8MHz F_{PWM} = 2KHz   Duty Cycle = 30%
T_{OSC} = 1.25ns   TMR2 Prescaler = 16

$Period_{SIGNAL} = 1 / F_{PWM} = 1 / 2000Hz. = 5e^{-4}s$

$Period_{PWM}$ (Seconds) = [(PR2) + 1] ● 4 ● T_{OSC}
● (TMR2 Prescaler Value)

$5e^{-4}s$ = [(PR2) + 1] ● 4 ● T_{OSC} ● (TMR2 Prescaler Value)
$5e^{-4}s$ = [(PR2) + 1] ● 4 ● 1.25ns ● (16)

Solve for PR2:

PR2 + 1 = ($5e^{-4}s$) / [4 ● 1.25ns ● (16)]
PR2 = {($5e^{-4}s$) / [4 ● 1.25ns ● (16)]} – 1
PR2 = 61.5 ≈ 62 = 03EH
PR2 ≈ 62 = 03EH

See the table below for characteristics of varying the prescaler and F_{PWM} values and their affects on PR2.

Prescaler setting	1kHz	2kHz	4kHz
1:16	124	62	31
1:4	499	249	124
1:1	1999	999	499

PR2 values
(All colored values will work in this example.)

As you can see from the table above, if you set the prescaler to 1:1 you will have to increase the F_{PWM} until the PR2 value is small enough to fit in an 8 bit register. But if you increase the F_{PWM} too much you will affect the switching efficiency of the MOSFETs. So in selecting a prescaler, you are limited by F_{PWM} and the size of the PR2 value, it must fit within an 8 bit register (thus, the PR2 value must be less than 256). So, as you can see, there are other values that will work as well. For example, the only difference between PR2 = 62 at 2kHz, Prescaler = 16 and PR2 = 249 at 2kHz, Prescaler = 4 is that the PR2 = 249 has a greater bit resolution. Is this necessarily better?, depends on your application.

Calculate the $Resolution_{PWM}$:

$$Resolution_{PWM}\ (\text{Bits}) = \frac{\log[4(PR2+1)]}{[\log(2)]}$$

$$= \frac{\log[4(62+1)]}{[\log(2)]}$$

$Resolution_{PWM}$ = 7.96 bits

So, knowing that the $Resolution_{PWM}$ is 7.96 bit, then How many discrete pulse widths to you have?

$2^{ResolutionPWM} = 2^{7.96} \approx 250$ discrete pulse widths.

Now you need to know what goes into the CCPR1L and CCP1CON <5:4> registers.
Thus, with a 30% duty cycle;

$$250 \bullet 30\% = X$$
$$250 \bullet (30 / 100) = X$$

$$X = 75$$

75 = B‘01001011’

So, **CCPR1L = B‘00010010’ and CCP1CON<5:4> = B‘00111100’**

Lab 2: Hardware PWM Slides 60 – 61 Task 1

Modify the Lab 2 assembly code to create a 3 kHz 40% duty cycle PWM using the CCP module.

Knowns:

F_{OSC} = 8MHz F_{PWM} = 3KHz Duty Cycle = 40%

T_{OSC} = 1.25ns TMR2 Prescaler = 16

$Period_{SIGNAL} = 1 / F_{PWM}$

Calculate $Period_{SIGNAL}$:___________

Calculate $Period_{PWM}$:_____________

Calculate PR2:_________________

Calculate $Resolution_{PWM}$:___________________

How many discrete pulse widths? _______________

What value goes into the CCPR1L and CCP1CON <5:4> registers?

CCPR1L (Binary):_______________________

CCP1CON <5:4> (Binary):______________________

Lab 2: Hardware PWM Slides 60 – 61 Task 1

Solution

Modify the Lab 2 assembly code to create a 3 kHz 40% duty cycle PWM using the CCP module.

Knowns:

F_{OSC} = 8MHz F_{PWM} = 3KHz Duty Cycle = 40%

T_{OSC} = 1.25ns TMR2 Prescaler = 16

$Period_{SIGNAL} = 1 / F_{PWM} = 1 / 3000Hz. = 3.33e^{-4}s$

$Period_{PWM}$ (Seconds) = [(PR2) + 1] ● 4 ● T_{OSC}
● (TMR2 Prescaler Value)

$3.33e^{-4}s$ = [(PR2) + 1] ● 4 ● T_{OSC} ● (TMR2 Prescaler Value)
$3.33e^{-4}s$ = [(PR2) + 1] ● 4 ● 1.25ns ● (16)

Solve for PR2:

PR2 + 1 = ($3.33e^{-4}s$) / [4 ● 1.25ns ● (16)]
PR2 = {($3.33e^{-4}s$) / [4 ● 1.25ns ● (16)]} – 1
PR2 = 40.7 ≈ 41 = 029H
PR2 ≈ 41 = 029H

Calculate the $Resolution_{PWM}$:

$$Resolution_{PWM} (Bits) = \frac{\log [4 (PR2 + 1)]}{[\log (2)]} = \frac{\log [4 (41 + 1)]}{[\log (2)]}$$

$Resolution_{PWM}$ = 7.39 bits

So, knowing that the $Resolution_{PWM}$ is 7.39 bit, then How many discrete pulse widths to you have?

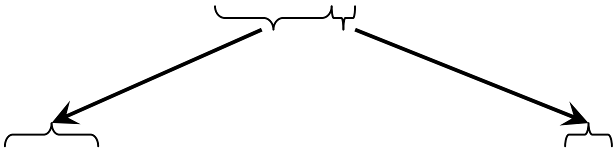
$2^{ResolutionPWM} = 2^{7.39}$ ≈ 168 discrete pulse widths.

Now you need to know what goes into the CCPR1L and CCP1CON <5:4> registers.
Thus, with a 30% duty cycle;

$$168 \bullet 40\% = X$$
$$168 \bullet (40 / 100) = X$$

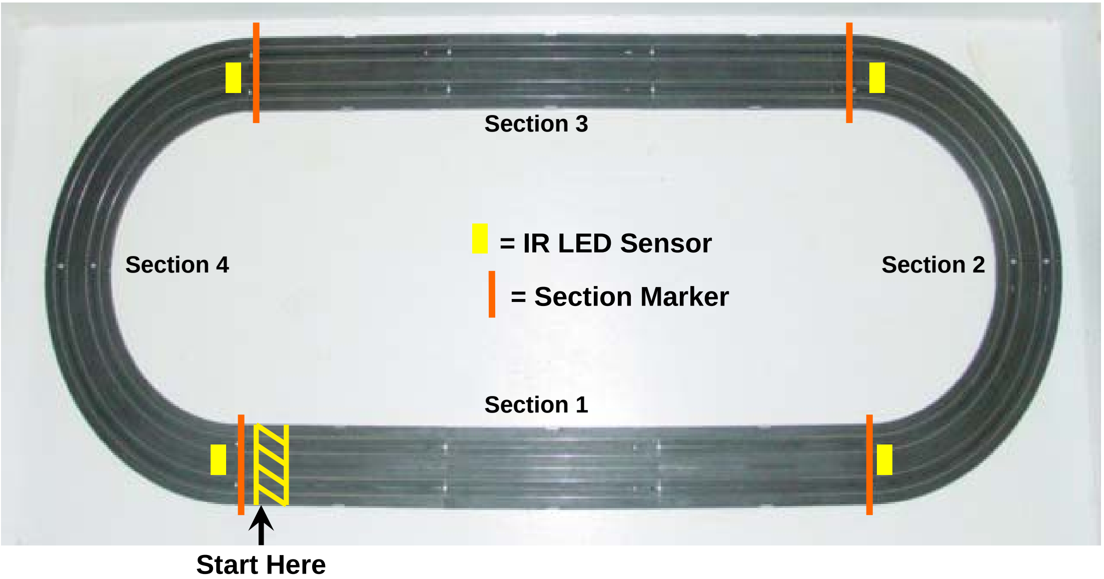
$$X = 67$$

67 = B‘01000011’



So, **CCPR1L = B‘00010000’ and CCP1CON<5:4> = B‘00111100’**

Oval Track Layout



Complex Track Layout