

ME218C Final Project, 2016
Team 6
PWM6 Pseudocode

PWM

Module Variables: PWMFreq 0-3, PWMduty 0-7

InitializePWM8

Takes nothing, returns nothing

Enable Clock to PWM Module

Enable Clock to port B

Enable Clock to port D

Select the PWM clock as System Clock/32

Make sure that the PWM Module clock has gotten going

Disable the PWM while initializing

Program generator A to go to 1 at rising compare A, 0 on falling compare A -
Generator 0

Program generator B to go to 1 at rising compare B, 0 on falling compare B -
Generator 0

Program generator A to go to 1 at rising compare A, 0 on falling compare A -
Generator 1

Program generator B to go to 1 at rising compare B, 0 on falling compare B -
Generator 1

Program generator A to go to 1 at rising compare A, 0 on falling compare A -
Generator 3

Program generator B to go to 1 at rising compare B, 0 on falling compare B -
Generator 3

Calculate Period

Set the PWM period

Set duty cycles for each channel

Enable PWM Outputs 0-7

Configure Port B pin as PWM Outputs

Configure Port D pin as PWM Outputs

Map PWM to these pins

Enable Port B Pin 4,5,6,7 for digital IO

Make Port B Pin 4,5,6,7 an output

Set up/down count mode and enable PWM generator

End InitializePWM8

SetDuty6

Takes a channel and duty, returns nothing

Select a channel based on input argument

For that channel, handle 100% edge case

Handle 0% edge case

If neither of those cases, reset cmp actions, calculate the number of ticks and load it into the cmp register

Update module var

End SetDuty8

SetFreq6

Takes a channel and frequency, returns nothing

Select a channel pair based on input argument

Disable pwm

Update module variable

For that channel calculate period in ticks

Update load register

Update duty cycles for that pair

Reenable PWM

End SetFreq8

SetWidth6

Takes a channel and pulse width, returns nothing

Select a channel based on input argument

Update duty cycle based on pulse width

Update comparator value based on the given width

End Setwidth8