Name ___________________________

Lab Section ______________________

PIC – Analog Voltage to PWM Duty Cycle

Lab 5

Introduction: In this lab you will convert an analog voltage into a pulse width modulation (PWM) duty cycle. The source of the analog voltage will be the trim pot voltage divider attached to RA2/ANA2 (pin 11) of your PIC Dev 14 board. The pulse width modulation output will drive the LED connected to RC5 (pin 5) so you can monitor the PWM duty cycle as the brightness of the LED.

Lab Requirements:

1. Demonstration of LED Dimmer Control using the analog to digital converter (ADC) and pulse width modulation (PWM).

Demo Check (JK)______

Analog to Digital Converter:
The PIC16F18324 has a 10-bit analog to digital converter that is multiplexed to 11 external pins as well as a number of internal voltages. To sample an external signal with the ADC you must tristate the pin using the TRIS register and specify the pin as an analog input by configuring the ANSEL register. To route a signal into the ADC module the CHS bits of the input MUX must be set to the corresponding channel. For some applications the full 10-bit conversion is not needed and 8-bits of resolution may be adequate and more efficient due to the microcontroller's 8-bit architecture. We will discuss using the ADC in both 8-bit and 10-bit modes in the lab. To store a 10-bit result requires two output registers ADRESH and ADRESL where the conversion result can be either left or right justified by setting the ADFM bit. Other settings that will need to be configured are the positive (ADPREF) and negative (ADNREF) voltage references and the ADC clock source. Take a look at Figure 1 on the next page to understand the basic structure ADC module.

A timer interrupt can be a convenient way to schedule an analog to digital conversion. You can use your code from last week's lab to configure TMR0 to provide a 10ms interrupt interval which will provide a sampling rate of around 100Hz. To start a conversion the GO/DONE bit is asserted and when the conversion has finished the GO/DONE bit will be automatically cleared by the module.

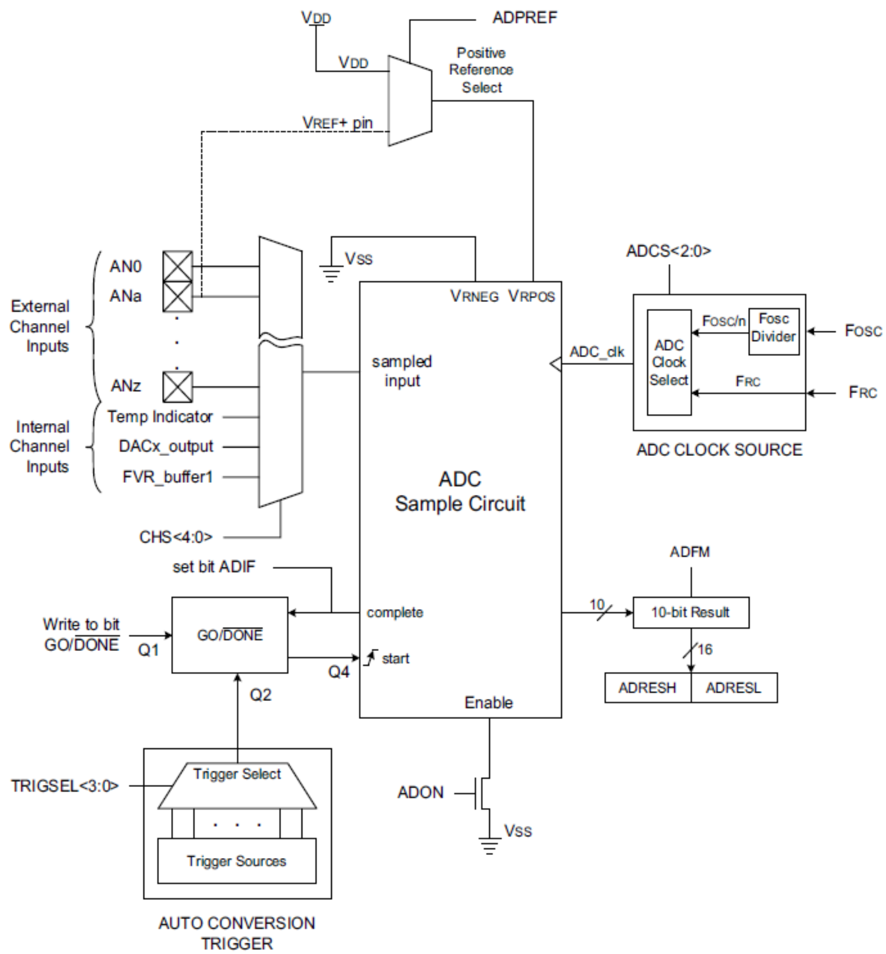


Figure 1 A2D Converter Module

The first step in using the A2D converter is to specify a pin as an analog input. This is typically done in the initialization sequence since it is unlikely that a pin would change from an analog input to digital functionality at runtime. Set the port pin direction as an input using the TRIS register and configure the pin as analog using the ANSEL register.

REGISTER 11-2: TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **TRISA<5:4>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

bit 3 **Unimplemented:** Read as ‘1’

bit 2-0 **TRISA<2:0>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

REGISTER 11-4: ANSELA: PORTA ANALOG SELECT REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **ANSA<5:4>**: Analog Select between Analog or Digital Function on pins RA<5:4>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **ANSA<2:0>**: Analog Select between Analog or Digital Function on pins RA<2:0>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

The configuration of the ADC converter in the PIC16F18324 is handled in two registers; ADCON0 and ADCON1. For this lab, both of these registers can be configured during initialization and the only bit you will need to assert at runtime is GO_DONE (ADGO).

REGISTER 21-1: ADCON0: ADC CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CHS<5:0>						GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-2 **CHS<5:0>:** Analog Channel Select bits
111111 = FVR (Fixed Voltage Reference)(2)
111110 = DAC1 output(1)
111101 = Temperature Indicator(3)
111100 = AVss (Analog Ground)
111011 = Reserved. No channel connected.
•
•
•
010101 = ANC5(4)
010100 = ANC4(4)
010011 = ANC3(4)
010010 = ANC2(4)
010001 = ANC1(4)
010000 = ANC0(4)
001111 = Reserved. No channel connected.
•
•
•
000101 = ANA5
000100 = ANA4
000011 = Reserved. No channel connected.
000010 = ANA2
000001 = ANA1
000000 = ANA0

bit 1 **GO/$\overline{\text{DONE}}$:** ADC Conversion Status bit
1 = ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle.
This bit is automatically cleared by hardware when the ADC conversion has completed.
0 = ADC conversion completed/not in progress

bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

The Analog Channel Select bits CHS <5:0> should be set to route the input from the port pin into the analog to digital converter. Since the potentiometer is connected to RA2/ANA2 (pin 11) the value should be "000010". The ADON bit should be set to turn the ADC on but the Go/nDONE bit should not be set at the same time that the converter is being switched on. The Go/nDONE bit will be asserted later to start a conversion.

ADCON0 = 0b00001001;

REGISTER 21-2: ADCON1: ADC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **ADFM:** ADC Result Format Select bit
1 = Right justified. Six Most Significant bits of ADRESH are set to '0' when the conversion result is loaded.
0 = Left justified. Six Least Significant bits of ADRESL are set to '0' when the conversion result is loaded.

bit 6-4 **ADCS<2:0>:** ADC Conversion Clock Select bits
111 = ADCRC (dedicated RC oscillator)
110 = Fosc/64
101 = Fosc/16
100 = Fosc/4
011 = ADCRC (dedicated RC oscillator)
010 = Fosc/32
001 = Fosc/8
000 = Fosc/2

bit 3 **Unimplemented:** Read as '0'

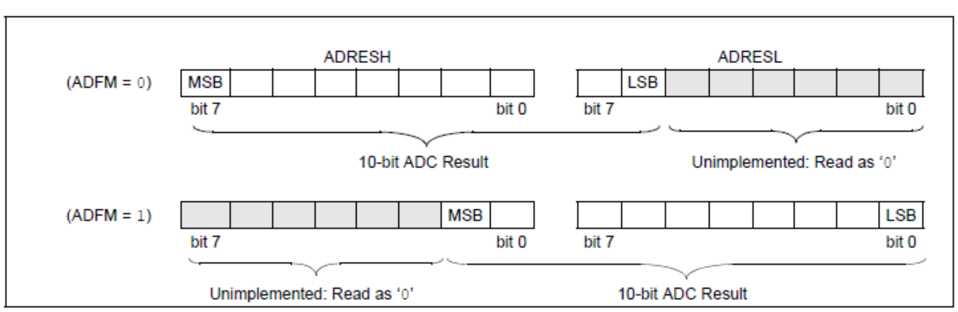
bit 2 **ADNREF:** A/D Negative Voltage Reference Configuration bit
When ADON = 0, all multiplexer inputs are disconnected.
0 = VREF- is connected to AVss
1 = VREF- is connected to external VREF-

bit 1-0 **ADPREF<1:0>:** ADC Positive Voltage Reference Configuration bits
11 = VREF+ is connected to internal Fixed Voltage Reference (FVR) module(1)
10 = VREF+ is connected to external VREF+ pin(1)
01 = Reserved
00 = VREF+ is connected to VDD

Note 1: When selecting the VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See Table 34-13 for details.

The ADCON1 register is used to set the output format, the conversion clock, and the ADC positive and negative reference voltages. The ADC produces a 10-bit result that is stored in two 8-bit registers. The justification of the result can be set with the ADFM bit as Illustrated in the figure below.

FIGURE 21-3: 10-BIT ADC CONVERSION RESULT FORMAT



The recommended ADC conversion times are from 1-4us per bit. When operating with a Fosc of 4MHz a suitable conversion clock (ADCS) would be either Fosc/4, Fosc/8 or Fosc/16. For this lab the ADC reference voltages can be V_{DD} and V_{SS}.

ADCON1 = 0b01010000;

To start a conversion set the Go/nDONE bit (***ADGO = 1;***). The conversion result will be ready when the Go/nDONE bit clears. You can wait for the conversion to finish by testing the status of the Go/nDONE bit like the code below:

```
bsf     ADCON0, GO_DONE        ; Start Conversion
btfsc   ADCON0, GO_DONE        ; Conversion Done?
goto    $-1                    ; No, Test Again
movf    ADRESH, W              ; Yes, Put A2D result into W
```

Alternatively, you can start the conversion at the end of one interrupt service routine and pickup the result at the start of the next. Using this method you will not need to test the Go/nDONE bit if you provide enough time to guaranty that the conversion is complete.

Pulse Width Modulation:

The PIC16f18324 microcontroller provides up to four dedicated 10-bit pulse width modulation modules. Two are located in the Compare/Capture/PWM modules (CCP1 and CCP2) and two are dedicated PWM modules (PWM5 and PWM6). These modules can each generate PWM signals of varying duty cycles but the frequency of modulation is fixed to one value because they share a common timer (TMR2). Just like with the A2D converter, sometimes it is sufficient to use the PWM module with only 8-bits, in which case you can take the ADC result (ADRESH) and place it into the PWM duty cycle register (PWMxDCH). We will discuss the consequences of using the PWM in 8-bit mode and 10-bit mode in the lab.

FIGURE 18-1: PWM OUTPUT

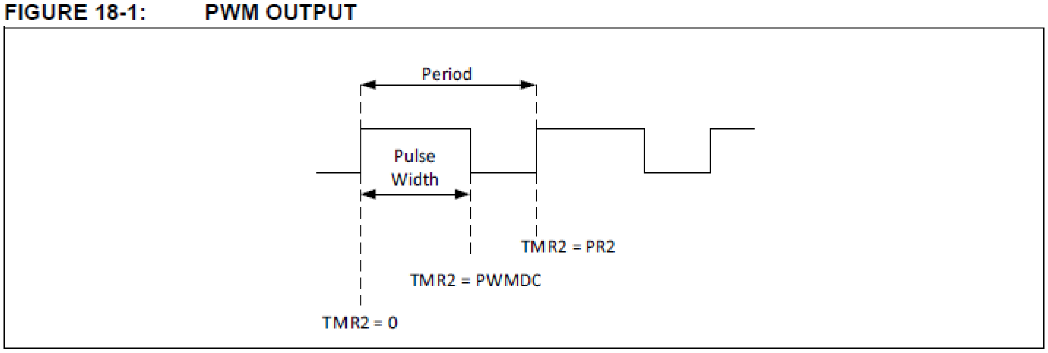
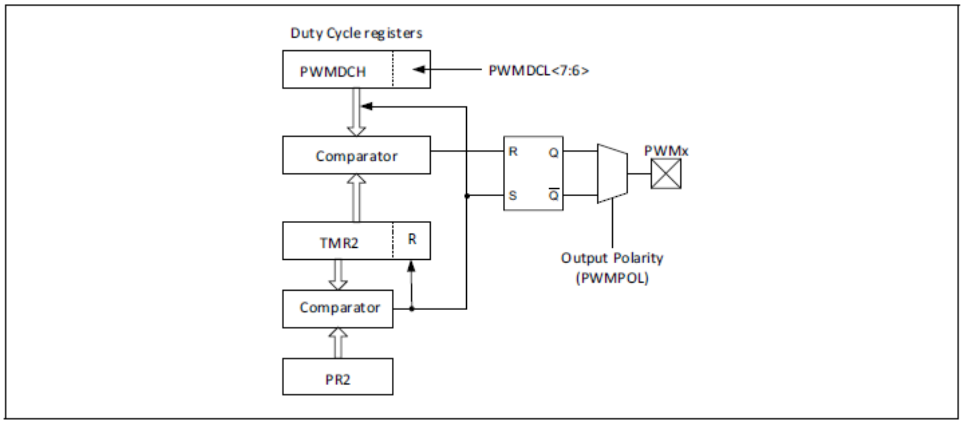


FIGURE 18-2: SIMPLIFIED PWM BLOCK DIAGRAM



To initialize the PWM module, you will need to configure several registers.

- T2CON – Timer2 Control Register
- PR2 – Timer2 Period Register
- PWMxCON – PWM Control Register
- PWMxDCH – PWM Duty Cycle High Bits
- PWMxDCL – PWM Duty Cycle Low Bits

Timer 2 is the clock source for the PWM module and can be configured to set the frequency of modulation by setting the prescaler and match register PR2. For today's lab turn on timer 2 and load the match register with 0xFF.

PR2 = 0xff;
TMR2ON = 1;

The PWMxCON Control Register (PWM5CON) will need to be configured to turn the PWM on and set the output polarity.

REGISTER 18-1: PWMxCON: PWM CONTROL REGISTER

R/W-0/0	U-0	R-0	R/W-0/0	U-0	U-0	U-0	U-0
PWMxEN	—	PWMxOUT	PWMxPOL	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **PWMxEN:** PWM Module Enable bit
1 = PWM module is enabled
0 = PWM module is disabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **PWMxOUT:** PWM module output level when bit is read.

bit 4 **PWMxPOL:** PWMx Output Polarity Select bit
1 = PWM output is active-low
0 = PWM output is active-high

bit 3-0 **Unimplemented:** Read as '0'

For active high PWM operation the PWM5CON can be configured as follows:

PWM5CON = 0b10000000;

To set the PWM duty cycle you will write to the PWM5DCH and PWM5DCL registers.

REGISTER 18-2: PWMxDCH: PWM DUTY CYCLE HIGH BITS

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PWMxDC<9:2>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **PWMxDC<9:2>:** PWM Duty Cycle Most Significant bits
These bits are the MSbs of the PWM duty cycle. The two LSbs are found in the PWMxDCL register.

REGISTER 18-3: PWMxDCL: PWM DUTY CYCLE LOW BITS

R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	U-0	U-0
PWMxDC<1:0>		—	—	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

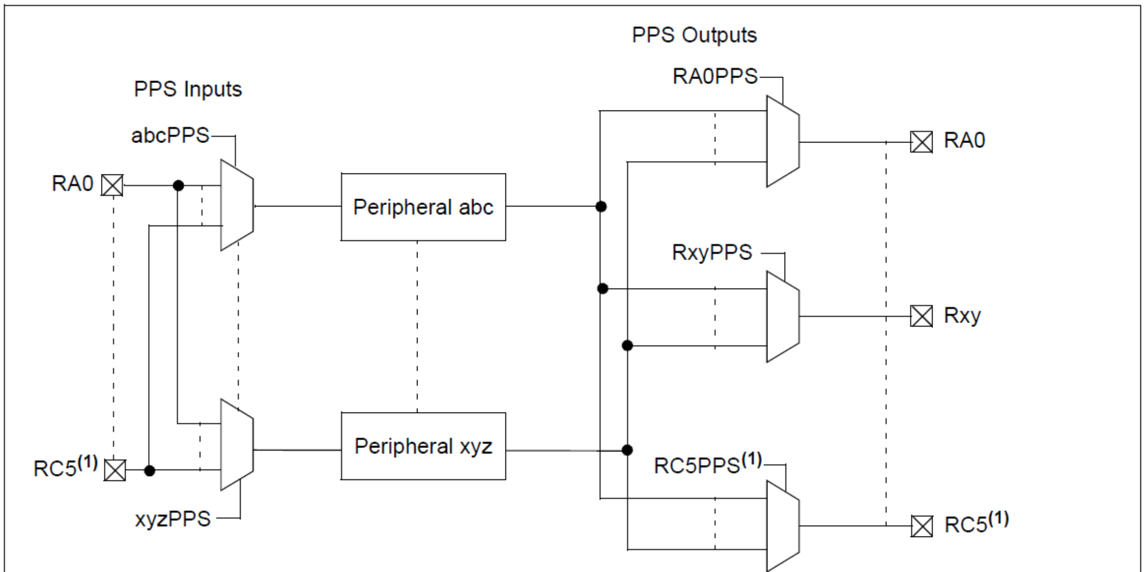
bit 7-6 **PWMxDC<1:0>:** PWM Duty Cycle Least Significant bits
These bits are the LSbs of the PWM duty cycle. The MSbs are found in the PWMxDCH register.

bit 5-0 **Unimplemented:** Read as '0'

Peripheral Pin Select:

The PIC16f18324 microcontroller contains a peripheral pin select (PPS) module which allows you to connect digital peripherals to the chips I/O pins. This is a very useful feature because it allows you to take advantage of the devices wide variety of peripherals in low pin count parts.

FIGURE 12-1: SIMPLIFIED PPS BLOCK DIAGRAM



Note 1: RC[y] are available on PIC16(L)F18323 only.

Inputs are configured using the xxxPPS registers where xxx refers to the peripheral name. Outputs are configured using the RxyPPS registers where xy refers to the pin name.

REGISTER 12-2: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	RxyPPS<4:0>				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RxyPPS<4:0>:** Pin Rxy Output Source Selection bits

11111 = Rxy source is DSM
11110 = Rxy source is CLKR
11101 = Rxy source is NCO
11100 = Rxy source is TMR0
11011 = Reserved
11010 = Reserved
11001 = Rxy source is SDO/SDA(1)
11000 = Rxy source is SCK/SCL(1)
10111 = Rxy source is C2OUT(2)
10110 = Rxy source is C1OUT
10101 = Rxy source is DT(1)
10100 = Rxy source is TX/CK(1)
...
01101 = Rxy source is CCP2
01100 = Rxy source is CCP1
01011 = Rxy source is CWG1D(1)
01010 = Rxy source is CWG1C(1)
01001 = Rxy source is CWG1B(1)
01000 = Rxy source is CWG1A(1)
...
00111 = Reserved
00110 = Reserved
00101 = Rxy source is CLC2OUT
00100 = Rxy source is CLC1OUT
00011 = Rxy source is PWM6
00010 = Rxy source is PWM5
00001 = Reserved
00000 = Rxy source is LATxy

Note 1: TRIS control is overridden by the peripheral as required.
2: PIC16(L)F18323 only.

It’s a good idea to lock PPS once setup so you can’t accidentally make changes after initialization. The datasheet recommends disabling the output drivers before configuration so to configure RC5 for PWM5 you might do something like the code below:

```
TRISC = 0xff;           // Disable Output Drivers
RC5PPS = 0b00010;       // PWM5 on RC5
PPSLOCK = 1;
```