CoE 115 Lab 5: Pulse Width Modulation

March 2019

Topics/Objectives:

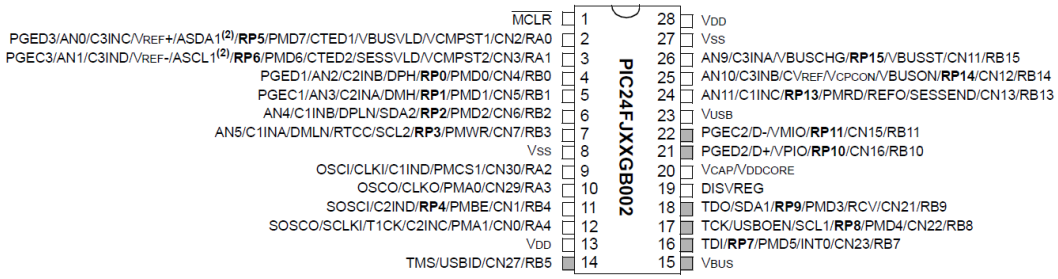
- To learn how to configure the remappable pins in the microcontroller
- To use the Output Compare module to generate PWM signals with controllable period and duty cycle

Peripheral Pin Select

Older versions of microcontrollers have the disadvantage of pins being limited or restricted to specific peripherals. In some cases of complex applications, multiple chips were required in order to attain full functionality of a complex system, simply because using one type of peripheral renders another one peripheral inaccessible because these peripherals share the same pin.

One of the significant improvements of recent microcontrollers is the increase in flexibility of pin functions. Now, most peripherals can be accessed through different pins. However, access to those peripherals requires additional configuration using software.

The list of peripherals accessible from physical remappable pins (RPx) should be indicated in the datasheet (see Tables 10-2 and 10-3 of the datasheet for the PIC24FJ64GB002). A separate set of SFRs are used for remappable input pins (RPINRx) and remappable output pins (RPORx). Predefined "codes" must be held at specific RPINRx and RPORx bits to assign the peripherals to the remappable pins.



A peripheral can have input/s, output/s, or both. For PIC24FJ64GB002, the mapping of a peripheral input and output to a pin follows a different approach:

- **Peripheral inputs are assigned to single pin.** For example, the T2CK input (Timer2 external clock) of the Timer2 module can be assigned to RP1 using:

```
_T2CKR = 1; // or RPINR3bits.T2CKR = 1;
```

- **Remappable pins are assigned to a single peripheral output.** For example, the pin RP12 can be assigned to the U2TX output of the UART2 peripheral using:

```
_RP12R = 5; // 5 is the "code" for U2TX
```

  This is because a peripheral output is allowed to drive multiple output pins. However, only one peripheral must drive an output pin.

There are three basic steps in configuring a remappable pin:

1. Unlock the IO registers
2. Configure the input functions or the output function.
3. Lock the IO registers

Below is an example of configuring the peripheral pins, which uses the RPxR names directly. You may also refer to the Example 10-3 in the datasheet. The example shown sets the OC1 (Output Compare 1) output to the remappable pin RP13.

```
// Make sure that the GPIO pin shared to RP13 is configured as an output
// This is because OC1 is an output of a peripheral.
_TRISB13 = 0;

// Unlock the RPINRx and RPORx registers.
// Performs sequence described in 10.4.4.1 of the datasheet.
__builtin_write_OSCCONL(OSCCON 0xBF); // clears IOLOCK bit


// Example: Assign OC1 output to RP13.
// Refer to Table 10-3 of the datasheet for the interpretation of values
_RP13R = 18;

// Lock the RPINRx and RPORx registers
// Performs sequence described in 10.4.4.1 of the datasheet.
__builtin_write_OSCCONL(OSCCON | 0x40); // sets IOLOCK bit
```

Note: The __builtin_write_OSCCONL() function is a built-in function in the MPLAB XC16 C compiler and does not need additional libraries to be used.

Output Compare Module

The Output Compare (OC) module is a peripheral that must be accessed through the remappable pins. This module can be used for generating pulses for a variety of applications which include Pulse Width Modulation or PWM.

The timing of the PWM signal from the OC module may be derived from the system clock or a Timer.

In edge-aligned mode, the period of the OC module is set to self-synchronize, meaning, the OC module triggers itself to restart the counting. The value at OCxRS and OCxR can be used to change the period and duty cycles, respectively, based on the clock source used. For simplicity, we can just use the system clock.

Also, in PWM mode the OCxR and OCxRS registers are double buffered. This means that the new values for OCxR and OCxRS would only be reflected after a timer rollover from 0xFFFF or after a Sync event occurs. As a result, any change on these registers during operation occurs only with the next pulse.

For more information on edge-aligned PWM mode (and other modes) of the PIC24F Output Compare module, read the documentation in this link:
http://ww1.microchip.com/downloads/cn/DeviceDoc/cn534503.pdf

Exercise

Part 1: Constant frequency and pulse width [60 pts]

Taking the first two digits of your student number (XXXX-**AB**YYY), configure the OC2 module in edge-aligned PWM to create a square wave with frequency **AB** kHz and 50% duty cycle.

- If the first digit of your SN is 0, use 2 instead (A = 2).
- Use the internal clock oscillator as the clock source.
- The output of the OC2 module should be mapped to the RP5.
- No PWM faults and interrupts.

Part 2: Constant frequency and variable pulse width [20 pts]

Use a potentiometer and the ADC module to control the duty cycle of the signal from 10% to 90% with the period set at **AB** kHz.

Part 3: Variable frequency and constant pulse width [20 pts]

Use a potentiometer and the ADC module to control the frequency of the PWM from 1kHz to **AB** kHz with the duty cycle set at 50%.

Bonus: Variable frequency and pulse width [10 pts]

Use two potentiometers and 2 ADC modules to control the PWM frequency from 1kHz to **AB** kHz and duty cycle from 10% to 90%.