# CoE 115: Laboratory 5

March 6-10, 2017

## Topics/Objectives:

- Using remappable pin functions through the microcontroller's peripheral pin select module
- Using the Output Compare module to generate PWM signals

## Introduction:

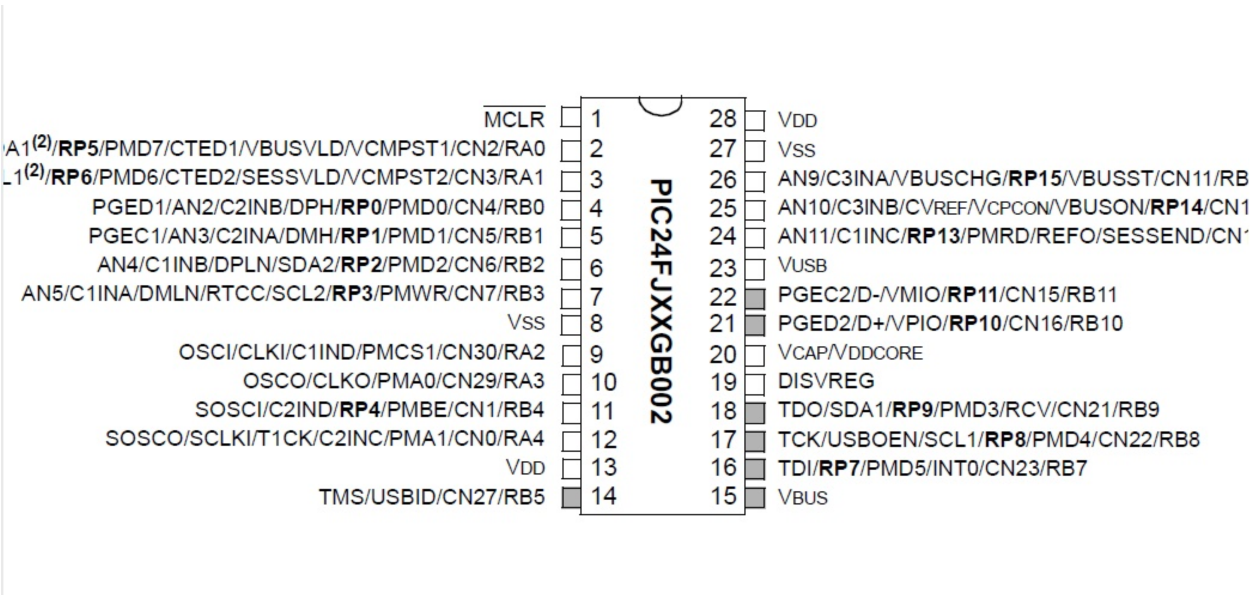
### Peripheral Pin Select

Older versions of the PIC microcontrollers had 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 up one peripheral renders another one inaccessible.

One of the significant improvements of recent microcontrollers is the increase in flexibility of pin functions. Now, most peripherals have the option to be accessed from any of most available pins. However, gaining access to those peripherals requires additional configuration in software.

The list of peripherals accessible from physical remappable pins is available in the corresponding datasheet of the microcontroller model you are using. There are assigned function "codes" which should be specifically configured to the corresponding input/output remappable pin.

The assignment of remappable pins (RP) can be seen in the image below. The numbered RPs are indicated by bold face characters.




Consider the following example in using a certain peripheral through RPs.
Let us say we want to use pin 24 as the receiver for incoming data from a PC terminal via serial communication. This means we must use the UART module of the microcontroller and assign U1RX function to pin 24 (RP 13). The corresponding register which contains the input functionality of UART1 is shown in Table 10.2 of the datasheet. Therefore, the following code must be applied.

```
void Setup(void){
   1  __builtin_write_OSCCONL(OSCCON & (~(1<<6))); // Release IOLOCK
   2  RPINR18bits.U1RXR = 0x000E;      // Assign U1RX to RP13
   3  __builtin_write_OSCCONL(OSCCON|(1<<6)); // Engage IOLOCK
...
}
```

*Note that the corresponding TRIS register bit for pin 24 must still be set separately.*
Lines 1 and 3 are necessary when assigning RPns to certain functions. This is an additional level of security since the microcontroller pins are more vulnerable to misuse/mistaken configuration, as a trade-off to their flexibility.

### Output Compare Module

The Output Compare (OC) module is one example of a peripheral which must be accessed through RPs. This can be used for a variety of applications by configuring it to operate in any of its multiple modes, one of which is Pulse Width Modulation (PWM). Refer to Tables 10.2 and 10.3 from the datasheet for the Peripheral Pin Select options associated with said module.

The timing of the PWM signal may be derived from a Timer (2,3,4, or 5 if available) which is dependent on either an external clock source or the internal clock of the microcontroller. The signal frequency and on-time may be set by calculating the corresponding period, and assign the correct values to PRy (of Timer y), OCxR for the initial duty cycle, and OCxRS for the duty cycle. The OCxR register is only accessed during initialization, afterwhich, it becomes a read-only register in PWM mode once the OC module is enabled.

However, it is still possible to operate the signal with variable duty cycle by updating the value of OCxRS, which is still accessible to the user after initialization. Each time a period match occurs, the OC module will automatically load the current value of OCxRS into OCxR which serves as the compare value for duty cycle. This means, there will be at most 1 period latency of updating the duty cycle of the PWM signal.

## Quiz (85%)

You will be required to generate variable level voltage by controlling the drive signals of a buck converter module. In order to increase efficiency, the buck converter was designed to operate in synchronous mode. This means the bottom switch should be driven by a signal that is complementary to the drive signal of the top switch. An external mosfet driver IC can be driven by a single PWM input waveform, in order to generate these 2 separate complementary waveforms.

The module to be driven is configured in this way, meaning you are only required to generate one PWM signal to serve as the input to the external driver. The input signal will be replicated to the top switch drive, and complemented to the bottom switch drive. Since the output voltage level is dependent on how long the top switch is on, this means the duty cycle of the top drive will be proportional to the output voltage level.

The user input will be the wiper arm position of a single-turn potentiometer to control the output voltage level. At the wiper's minimum position, the output of the converter should be at around $200mV$, and at maximum position, the output should be around $9.2V$ for a $10V$ input.

The specifications below will guide you to the implementation:

- Implement a variable duty cycle, edge-aligned PWM signal using the Output Compare module of the microcontroller.
- Use Timer 2 as the time base reference, with the internal oscillator as the clock source.
- Use RB7 (Remappable Pin 7) as the OCx pin. Make sure to temporarily disable security lock of RPn before assigning the appropriate value, and enable it afterwards. The RPOR register to which RP7 is assigned can be found in the datasheet.
- Use Output Compare Module 1

- The PWM signal must operate at 50kHz. You may arbitrarily set the initial duty cycle to a low value (e.g. 5%)
- There is no need to use PWM Fault pins or interrupts.
- It is advisable to use Timer 2's interrupt due to the 1 period latency of updating the duty cycle, so that each value update will be implemented at least once. Set any necessary status flags that would enable this delay. In order to access Timer 2's corresponding interrupt service routine, you may call the following function.

```
void __attribute__ ((interrupt,no_auto_psv)) _T2Interrupt(void){
...
}
```

- The duty cycle should be scaled to an A/D converted value such that $0V$ corresponds to around 4% duty cycle, and $3.3V$ corresponds to around 93% duty cycle. The ADC input will be from the wiper arm of a potentiometer, being fed into RB15 (AN9), as was used in Lab 4.
- When calculating/scaling the ADC Value, keep in mind that the microcontroller only has 16-bit wide registers. Make sure you do not trigger any overflows during computation.
- The microcontroller's system clock is configured to run at 4MHz.

  You may refer to the Pre Lab Quiz for the code flow on how to configure the OC Module as well as assign OC1 to RP7.

  *Partial Points:* You may first try generating a fixed-duty 50kHz signal before incorporating the ADC scaling. This warrants 45 points out of the 85.

## Post Lab Comments

If applicable, certain PIC24FJ models only have one OC register that requires configuration – OCxCON. Most of the settings correspond to OCxCON1 of the more complex models. However, all provided microcontrollers are capable of operating to the level required by all succeeding tasks in this course.