Homework Assignment #4

Due: **April 13, 2021, before midnight**

1. Consider a PIC24 microcontroller running at **16 MHz**.
   a. Assume that you need to generate a PWM signal to OC1 port with Timer 2. The PWM signal period and duty cycle should be 1 and 0.3 milliseconds, respectively. Find the value of PR2, OC1R(S).

PWM Period = $(PR2 + 1) * T_{cy} * \overset{\text{seconds}}{PRE\text{scalar}}$

$T_{cy} = \frac{1\text{sec}}{16 \times 10^6} = 62.5\text{ns}$

max PWM Period @ 1:1

$= 65536 * 62.5 \cdot 10^{-9} * 1 = 4.096 \times 10^{-3}$ s so we know prescalar can be 1:1 for 1ms

$1 \times 10^{-3} = (PR2+1) * (62.5 \times 10^{-9}\text{s}) * 1$

$\left(\frac{1 \times 10^{-3}}{62.5 \times 10^{-9}}\right) - 1 = \boxed{PR2 = 15999}$

Duty Cycle = $OC1RS * T_{cy} * \overset{\text{seconds}}{PRE\text{scalar}}$

$.3 \times 10^{-3} = OC1RS * (62.5 \times 10^{-9}\text{s}) * 1$

$$4800 = \boxed{\begin{array}{l} OC1RS \\ = 4800 \\ = 0001\ 0010\ 1100\ 0000 \end{array}}$$

   b. Find the value of PR2 and OC1R(S) if the PWM signal period and duty cycle should be 10 and 3 milliseconds, respectively.

max PWM Period @ PRE 1:1

$= 65536 * 62.5 \times 10^{-9} * 1 = 4.096 \times 10^{-3}$ which is less than 10×10^{-3}, thus we need a PRE > 1:1

PRE calculation: $\frac{10 \times 10^{-3}}{4.096 \times 10^{-3}} = 2.44 \longrightarrow$ Need a PRE $\geq 2.44 \longrightarrow$ PRE = 1:8 (smallest available)

PWM Period = $(PR2 + 1) * T_{cy} * \overset{\text{seconds}}{PRE\text{scalar}}$

$10 \times 10^{-3}\text{s} = (PR2+1) * (62.5 \times 10^{-9}) * (8)$

$PR2 = \frac{10 \times 10^{-3}}{(62.5 \times 10^{-9})(8)} - 1$

$\boxed{PR2 = 19999}$

Duty Cycle = $OC1RS * T_{cy} * \overset{\text{seconds}}{PRE\text{scalar}}$

$3 \times 10^{-3} = OC1RS * (62.5 \times 10^{-9}) * (8)$

$OC1RS = \frac{3 \times 10^{-3}}{(62.5 \times 10^{-9})(8)} = 6000$

$$\boxed{\begin{array}{l} OC1RS = 6000 \\ = 0001\ 0111\ 0111\ 0000 \end{array}}$$

2. Assume that there is an external sensor device. The device has a UART interface to send its sensor data. The device expects 1200 Baud Rate.
   a. You are supposed to use your PIC24 microcontroller to communicate with the sensor device. Find the correct U1BRG value and the Baud rate error (in %). Assume that the microcontroller runs at 16 MHz and BRGH = 0.

$$\text{Baud Rate} = \frac{F_{cy}}{16 \times (U_xBRG+1)}$$

$$U_xBRG = \frac{F_{cy}}{16 \times (\text{Baud Rate})} - 1$$

$F_{cy} = 16 \times 10^6$ Hz BRGH = Ø → so equations to left are true

$$\boxed{\text{U1BRG}} = \frac{16{,}000{,}000}{(16)(1200)} - 1 = 832.3 \rightarrow \boxed{832} =$$

$$\text{Calculated Baud Rate} = \frac{f_{cy}}{16 \times (U1BRG+1)} = \frac{16{,}000{,}000}{16 \times (832+1)} = 1200.48$$

$$\boxed{\text{Error}} = \left(\frac{\text{Calc baudrate} - \text{Desired baudrate}}{\text{Desired Baud Rate}}\right) = \left(\frac{1200.48 - 1200}{1200}\right) = \boxed{.040\ \%\ \text{Error}}$$

   b. Find the correct U1BRG value and the Baud rate error (in %) if your microcontroller runs at **12 MHz** and BRGH = 0.

$$\text{Baud Rate} = \frac{F_{cy}}{16 \times (U_xBRG+1)}$$

$$U_xBRG = \frac{F_{cy}}{16 \times (\text{Baud Rate})} - 1$$

$$\boxed{\text{U1BRG}} = \frac{12{,}000{,}000}{(16) \times (1200)} - 1 = 625 - 1 = \boxed{624}$$

$$\text{Calculated Baud Rate} = \frac{F_{cy}}{16 \times (U1BRG+1)} = \frac{12{,}000{,}000}{(16)(624+1)} = 1200$$

$$\boxed{\text{Error}} = \left(\frac{\text{Calc baudrate} - \text{Desired baudrate}}{\text{Desired Baud Rate}}\right) = \left(\frac{1200 - 1200}{1200}\right) = \boxed{Ø\ \%\ \text{Error}}$$

   c. Briefly discuss the Baud rate error of the aforementioned cases. Is the error critical for UART communication? Briefly explain why.

An error of .04% and Ø% will not be enough to distort UART communication. This is because the baud rates will synchronize between the devices before every new communication. Therefore as long as the devices agree with a .04% or Ø% error, then the communication is not enough to be critical for UART communication.

d. Now let's assume that you are supposed to use your PIC24 microcontroller running at **4 MHz** to communicate with the sensor device. Find the U1BRG value and BRGH bit that have the smallest Baud rate error (in %).

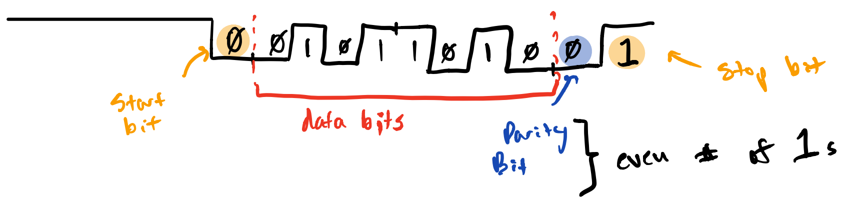
BRGH = ∅	BRGH = 1
$\text{Baud Rate} = \frac{F_{cy}}{16 \times (U_xBRG+1)}$	$\text{Baud Rate} = \frac{F_{cy}}{4 \times (U_xBRG+1)}$
$U_xBRG = \frac{F_{cy}}{16 \times (\text{Baud Rate})} - 1$	$U_xBRG = \frac{F_{cy}}{4 \times (\text{Baud Rate})} - 1$
$U1BRG = \frac{1000,000}{4 \times (1200)} - 1 = 207.\overline{33} \rightarrow 207$	$U1BRG = \frac{1000\,000}{1200} - 1 = 832.\overline{33} \rightarrow 832$
$\text{Calculated BR} = \frac{4\,000\,000}{16 \times (207)} = 1201.923$	$\text{Calculated BR} = \frac{4\,000\,000}{4 \times (833)} = 1200.48$
$\text{Error} = \frac{\text{Calc BR} - \text{actual BR}}{\text{actual BR}} = \frac{1201.92 - 1200}{1200}$	$\text{Error} = \frac{\text{Calc BR} - \text{actual BR}}{\text{actual BR}} = \frac{1200.48 - 1200}{1200}$
$= .16\ \%$	$= .040\ \%$

Since Error is least with BRGH = 1, $\boxed{U1BRG = 832 \ ; \ BRGH = 1}$

e. Assume that the sensor just sent one byte (0x5A) to your PIC24 microcontroller. If the data bit is 8-bit and even parity, **draw** the UART signal. Annotate the start bit, data bits, parity bit, and stop bit on your drawing.

0x5A

0101 1010



3. Caesar Cipher is a simple way to encrypt a sequence of characters. For details about Caesar Cipher, please read carefully here. Implement a C program using the UART of PIC24 microcontroller that decrypts the following encrypted text

```
QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD
```

to the original text

```
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
```

Reuse the code of Discussion 8 (`disc08_UART.c`) and use the provided UART input file (`hw4_uart_caesar_cipher_in.txt`). As we did in Discussion 8, you will need to simulate the UART stimulus and write to the UART 1 Output window.

   a. Submit your source code (filename: `hw4_uart_caesar_cipher_(your x500 id).c`) and the screen capture of the UART 1 Output window displaying the deciphered original text (filename: `hw4_uart_caesar_cipher_(your x500 id).[jpg | png]`).

   b. Briefly describe how your program decrypts the input.

1) I found how much the Cipher offset T from Q manually. I then defined a variable called "offset" to 3.

2) I used the UART function PGetch to read the encrypted text from the text file to the buffer

3) for each char x read, if x > A & x < Z - offset → PPutch x + 3
if x > Z - offset → PPutch x - (26 - 3)
Otherwise it's a space so just print the space