EE 2361 - Homework 4

Saturday, April 11, 2020 10:11 PM

Homework Assignment #4

Due: **April 14, 2020, 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).

$PWM = (PR2 + 1) \cdot T_{cy} \cdot PRE$

$PR2 = \frac{PWM}{T_{cy} \cdot PRE} - 1$

$= \frac{1\,mS}{62.5nS \cdot 1} - 1$

PR2 = 15,999

$Duty\ Cycle = OC1R \cdot T_{cy} \cdot PRE$

$OC1R = \frac{Duty\ Cycle}{T_{cy} \cdot PRE}$

$= \frac{0.3\,mS}{62.5nS \cdot 1}$

OC1R = 4800

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

$PR2 = \frac{PWM}{62.5nS \cdot PRE} - 1$

PRE = 8

$PR2 = \frac{10\,mS}{62.5nS \cdot 8} - 1$

PR2 = 19,999

$OC1R = \frac{Duty\ Cycle}{T_{cy} \cdot PRE}$

$= \frac{3mS}{62.5nS \cdot 8}$

OC1R = 6000

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.

$U1BRG = \frac{F_{cy}}{16 \cdot (Baud\ Rate)} - 1$

$= \frac{16\,MHz}{16(1200)} - 1 = [832.33]$

U1BRG = 832

$Baud\ Rate = \frac{F_{cy}}{16(U1BRG+1)}$

$= 1200.48$

$Error = \frac{Calculated\ BR - Desired\ BR}{Desired\ BR} \times 100$

Error = 0.04%

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

$U1BRG = \frac{12\,MHz}{16(1200)} - 1$

U1BRG = 624

$BR = \frac{12\,MHz}{16(624+1)}$

BR = 1200

Error = 0%

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

No-- The UART receiver resets its timer every time it receives a start bit, then samples the following bits. As long as the number of bits it receives after the start is close to the average number it has been receiving, it shouldn't cause any issues.

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.

The function void **PPUTCH**(const unsigned char c) shifts the character by 3 and takes the modulus by 26, as per the given Caesar Cypher formula. We must also subtract by 65 before the modulus to ensure c is a valid ASCII character, then add 65 after to set the letter 'A' as position 0.