Microprocessor Principles and Applications Final (Hands-on Test)

Name: _____ Fall 2021

ID: _____

Note: We may change testcases when you demonstrate your programs to us.

1. Please implement a UART by using the keyboard to create two modes mentioned below.
   (a) (20%)Design a "mode1" mode. Type "mode1" to enter this mode. In "mode1", first input value n(1<=n<=9). Value of n means that it needs **2n** steps(size of each step is same) to change servomotor's degree from -90 to 90 and then go back to -90. The string you type and value n must be printed on screen. After setting n, when you click button(RB0) one time, servomotor will change its degree a step. Servormotor's degree will be 90 when you click button **n** times, and degree will be -90 when you click **2n** times. Program will exit mode1 after **2n** steps.

   Ex. n=2,

   Degree is -90->type 2->First click: degree is 0->Second click: degree is 90 ->Third click: degree is 0->Fourth click: degree is -90->exit mode1

   (b) (20%)Design a "mode2" mode. Type "mode2" to enter this mode. In "mode2", first input value n(1<=n<=9). Value of n means that it needs **2n** steps(size of each step is same) to change servomotor's degree from -90 to 90 and then go back to -90. The string you type and value n must be printed on screen. After setting n, when you click button(RB0), servomotor will start rotating **automatically**. The interval between each step is 1s. You need to use **timer** to create 1s delay. Program will exit mode2 after **2n** steps.

   Ex. n=2,

   Degree is -90->type 2 ->click button-> Wait 1s-> Degree is 0-> Wait 1s -> Degree is 90 -> Wait 1s-> Degree is 0->Wait 1s-> Degree is -90->exit mode2

   Hint:

TABLE 18-1: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-Bit/Asynchronous	Fosc/[64 (n + 1)]
0	0	1	8-Bit/Asynchronous	Fosc/[16 (n + 1)]
0	1	0	16-Bit/Asynchronous	
0	1	1	16-Bit/Asynchronous	Fosc/[4 (n + 1)]
1	0	x	8-Bit/Synchronous	
1	1	x	16-Bit/Synchronous	

Legend: x = Don't care, n = value of SPBRGH:SPBRG register pair

EXAMPLE 18-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

Desired Baud Rate	=	FOSC/(64 ([SPBRGH:SPBRG] + 1))
Solving for SPBRGH:SPBRG:		
X	=	((FOSC/Desired Baud Rate)/64) – 1
	=	((16000000/9600)/64) – 1
	=	[25.042] = 25
Calculated Baud Rate	=	16000000/(64 (25 + 1))
	=	9615
Error	=	(Calculated Baud Rate – Desired Baud Rate)/Desired Baud Rate
	=	(9615 – 9600)/9600 = 0.16%

2.

(a) (10%) Connect a push-button to **RB0 (INT0)** with a pull-up or pull-down resistor. Connect a LED to **RD0**. Press the button to **toggle** the LED. **You need to use an interrupt to handle the button event.**

(b) (30%) Use UART to implement Timer (the value is a two-digit integer, in units of seconds). You should follow the following three steps.

1. Show the string "Timer: 00" on the screen. The last two numbers mean the value of the Timer. When the button is pressed, turn on the LED, and go to step 2.
2. Use **TIMER1 to trigger an interrupt** every second. When the interrupt is triggered, you should modify the string on the screen to meet the current Timer value.
   Ex. Timer: 00 -> Timer: 01 and so on.
   (You should replace 0 with 1 when time = 1, and replace 1 with 2 when time = 2, …)
   When the button is pressed, go to step 3.
3. Turn off the LED, add a newline ("\r\n" on Windows, CLRF, 13 10 in ASCII), and go back to step 1.

Hint:

1. You can check the different sources of interrupt flags to handle multiple interrupts.
2. You can use the character '\b' (backspace, 8 in ASCII) to modify the string on the screen.

3.

(a) (10%) Connect the adjustable resistor to RA0, five light bulbs from RD0 to RD4 and a servomotor on CCP1. Rotate the resistor and servomotor should rotate from -90 degree to +90 degree. Meanwhile, the light bulbs should be sequentially turned on and turned off when the servomotor is rotating. Shown as the table below.

degree	lights
$-90 \leq \theta < -45$	RD0
$-45 \leq \theta < 0$	RD0 RD1
$0 \leq \theta < +45$	RD0 RD1 RD2
$+45 \leq \theta < +90$	RD0 RD1 RD2 RD3
$+90$	RD0 RD1 RD2 RD3 RD4

(b) (30%) Use UART to implement a system to control the servomotor system in (a). You have to implement two modes on UART. Use a button to change the mode by interrupt(Initial: nothing is printed on screen. First click: show "mode1" on screen and enter mode1).

[1] Mode 1: Print "mode1" on the screen. As the description in (a), rotate the adjustable resistor so that the servomotor rotates and light bulbs should be turned on or turned off. Meanwhile, show the degree on the screen via UART and update every 0.5s.

Example Output:

mode1 -20 -35 -41 -45 -46

[2] Mode 2: Print "mode2" on the screen. Type the degree and press Enter to send this message via UART to PIC18. The servomotor should turn to this degree and the light bulbs should be turned on as the table shown in (a).

Ex. type -22 and press Enter. The servomotor rotates to -22 degree and RD0,RD1 should be turned on.

Microprocessor Principles and Applications Final (Hands-on Test)

Name: _____ Fall 2021

ID: _____

Note: We may change testcases when you demonstrate your programs to us.

1. Please implement a UART by using the keyboard to create two modes mentioned below.
   (a) (20%)Design a "mode1" mode. Type "mode1" to enter this mode. In "mode1", first input value n(1<=n<=9). Value of n means that it needs **2n** steps(size of each step is same) to change servomotor's degree from -90 to 90 and then go back to -90. The string you type and value n must be printed on screen. After setting n, when you click button(RB0) one time, servomotor will change its degree a step. Servormotor's degree will be 90 when you click button **n** times, and degree will be -90 when you click **2n** times. Program will exit mode1 after **2n** steps.

   Ex. n=2,

   Degree is -90->type 2->First click: degree is 0->Second click: degree is 90 ->Third click: degree is 0->Fourth click: degree is -90->exit mode1

   (b) (20%)Design a "mode2" mode. Type "mode2" to enter this mode. In "mode2", first input value n(1<=n<=9). Value of n means that it needs **2n** steps(size of each step is same) to change servomotor's degree from -90 to 90 and then go back to -90. The string you type and value n must be printed on screen. After setting n, when you click button(RB0), servomotor will start rotating **automatically**. The interval between each step is 1s. You need to use **timer** to create 1s delay. Program will exit mode2 after **2n** steps.

   Ex. n=2,

   Degree is -90->type 2 ->click button-> Wait 1s-> Degree is 0-> Wait 1s -> Degree is 90 -> Wait 1s-> Degree is 0->Wait 1s-> Degree is -90->exit mode2

Hint:

TABLE 18-1: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-Bit/Asynchronous	Fosc/[64 (n + 1)]
0	0	1	8-Bit/Asynchronous	Fosc/[16 (n + 1)]
0	1	0	16-Bit/Asynchronous	
0	1	1	16-Bit/Asynchronous	Fosc/[4 (n + 1)]
1	0	x	8-Bit/Synchronous	
1	1	x	16-Bit/Synchronous	

Legend: x = Don't care, n = value of SPBRGH:SPBRG register pair

EXAMPLE 18-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

Desired Baud Rate = FOSC/(64 ([SPBRGH:SPBRG] + 1))

Solving for SPBRGH:SPBRG:

X = ((FOSC/Desired Baud Rate)/64) – 1
= ((16000000/9600)/64) – 1
= [25.042] = 25

Calculated Baud Rate = 16000000/(64 (25 + 1))
= 9615

Error = (Calculated Baud Rate – Desired Baud Rate)/Desired Baud Rate
= (9615 – 9600)/9600 = 0.16%

2.

(a) (10%) Connect a push-button to **RB0 (INT0)** with a pull-up or pull-down resistor. Connect a LED to **RD0**. Press the button to **toggle** the LED. **You need to use an interrupt to handle the button event**.

(b) (30%) Use UART to implement Timer (the value is a two-digit integer, in units of seconds). You should follow the following three steps.

1. Show the string "Timer: 00" on the screen. The last two numbers mean the value of the Timer. When the button is pressed, turn on the LED, and go to step 2.
2. Use **TIMER1 to trigger an interrupt** every second. When the interrupt is triggered, you should modify the string on the screen to meet the current Timer value.
   Ex. Timer: 00 -> Timer: 01 and so on.
   (You should replace 0 with 1 when time = 1, and replace 1 with 2 when time = 2, …)
   When the button is pressed, go to step 3.
3. Turn off the LED, add a newline ("\r\n" on Windows, CLRF, 13 10 in ASCII), and go back to step 1.

Hint:

1. You can check the different sources of interrupt flags to handle multiple interrupts.
2. You can use the character '\b' (backspace, 8 in ASCII) to modify the string on the screen.

3.

(a) *(10%)* Connect the adjustable resistor to RA0, five light bulbs from RD0 to RD4 and a servomotor on CCP1. Rotate the resistor and servomotor should rotate from -90 degree to +90 degree. Meanwhile, the light bulbs should be sequentially turned on and turned off when the servomotor is rotating. Shown as the table below.

degree	**lights**
$-90 \le \theta < -45$	RD0
$-45 \le \theta < 0$	RD0 RD1
$0 \le \theta < +45$	RD0 RD1 RD2
$+45 \le \theta < +90$	RD0 RD1 RD2 RD3
$+90$	RD0 RD1 RD2 RD3 RD4

(b) (30%) Use UART to implement a system to control the servomotor system in (a). You have to implement two modes on UART. Use a button to change the mode by interrupt(Initial: nothing is printed on screen. First click: show "mode1" on screen and enter mode1).

[1] Mode 1: Print "mode1" on the screen. As the description in (a), rotate the adjustable resistor so that the servomotor rotates and light bulbs should be turned on or turned off. Meanwhile, show the degree on the screen via UART and update every 0.5s.

Example Output:

mode1 -20 -35 -41 -45 -46

[2] Mode 2: Print "mode2" on the screen. Type the degree and press Enter to send this message via UART to PIC18. The servomotor should turn to this degree and the light bulbs should be turned on as the table shown in (a).

Ex. type -22 and press Enter. The servomotor rotates to -22 degree and RD0,RD1 should be turned on.