

Homework 2

CSE 3442/5442 – Spring 2018
Embedded Systems 1

Due Date

- Tuesday 4/10/2018 by 10pm (submit single .C file to BlackBoard)
- Multiple submissions are allowed before due date (last one is the graded one)
- Neatly structured and thoroughly commented
- This is individual work; no pairs or groups
- Include your name and 1000-number commented at the top of .C file
- Name the file “**HW2_LastName.c**”

Demo

- A demo is not required but if I cannot run your code’s simulation fully on my machine, I will contact you about demoing your code’s functionality in person

Notes

- Choose **PIC18F452** as your microcontroller and **XC8** as your compiler
- Remember to choose **Simulator** as your Hardware Tool/Programmer/Debugger
- You may use any .C instructions available
- Your submitted .C file must be executable exactly how it is submitted (don’t just include the “meat” of your code; submit the full file including #include, CONFIGs, etc.)
- You may use “**HW2_StarterCode.c**” as a starting point (on classpage and blackboard)
 - It has valid configuration bits, delay setup, and printf() functionality.

Assignment

- In MPLAB’s simulator, write C code to utilize the CCP1 module’s **CAPTURE MODE**
 - Your program must use **capture mode and interrupts** to determine the frequency of a signal (square wave) coming in on pin RC2 (CCP1)
 - Continuously display the calculated frequency using printf()
 - Generate the signal using Window → Simulator → Stimulus → Clock Stimulus on pin RC2 (calculate how Low/High Cycles translates to Frequency)
 - Your program only needs to work for frequencies between **50Hz to 500Hz**
- The simulator’s default Instruction Frequency is **1MHz** (you may leave this unchanged)
 - This is **Instruction Frequency** (F_{cyc}), where F_{osc} isn’t available in Simulator
 - You can essentially think of the simulator as having a F_{osc} of **4MHz**
 - File → Project Properties → Simulator → Option Categories: Oscillator Options (shows the simulator’s Instruction Freq.)
- For printf() capabilities, the starter code already has the setup code but you need to make some settings in MPLAB to see print output in a window
 - File → Project Properties → Simulator → Option Categories → Uart1 IO Options → Enable Uart IO (check the box)
 - Keep your printf’s short and minimal since they can bog down the simulator’s speed sometimes (especially if they are in the ISR)
- The Stopwatch tool is useful for timing/debugging: Window → Debugging → Stopwatch

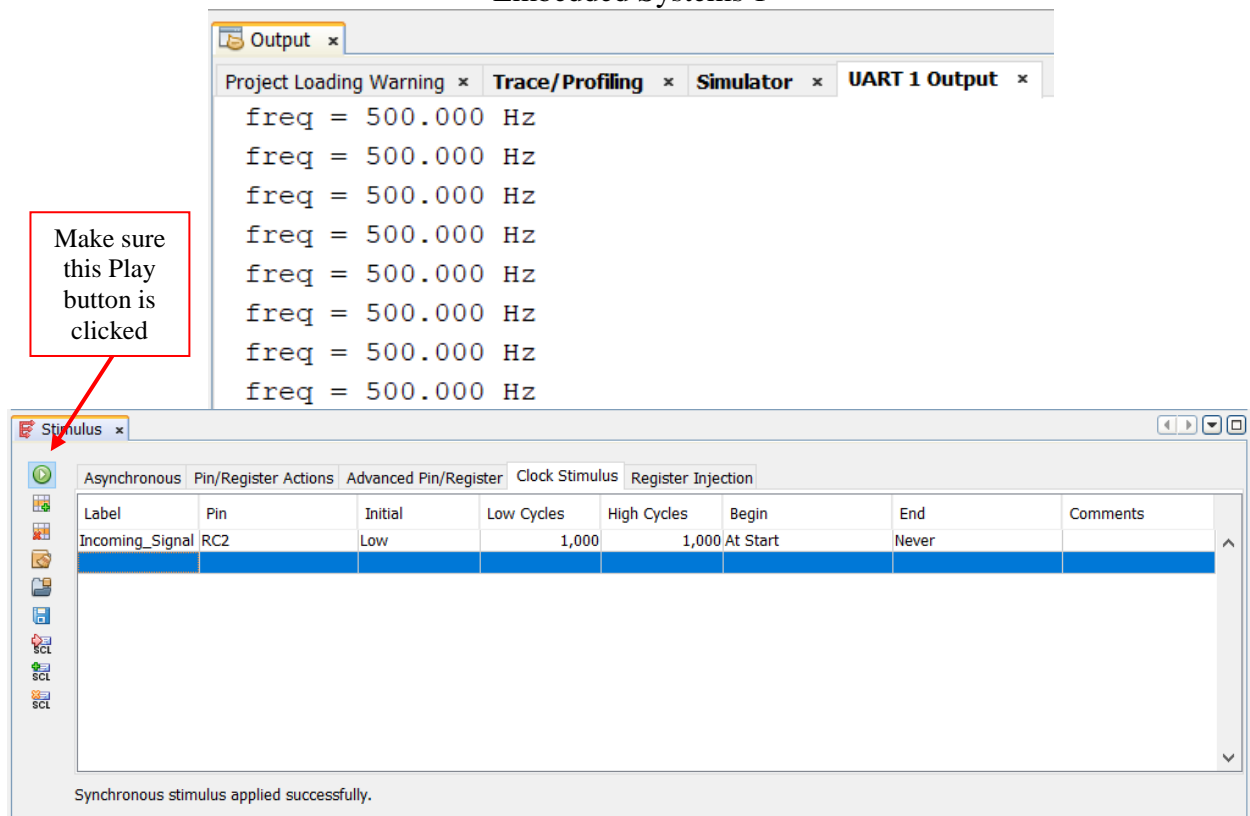
Some sample screen shots on next page

Homework 2

CSE 3442/5442 – Spring 2018

Embedded Systems 1

Make sure this Play button is clicked



Output

Project Loading Warning x Trace/Profiling x Simulator x UART 1 Output x

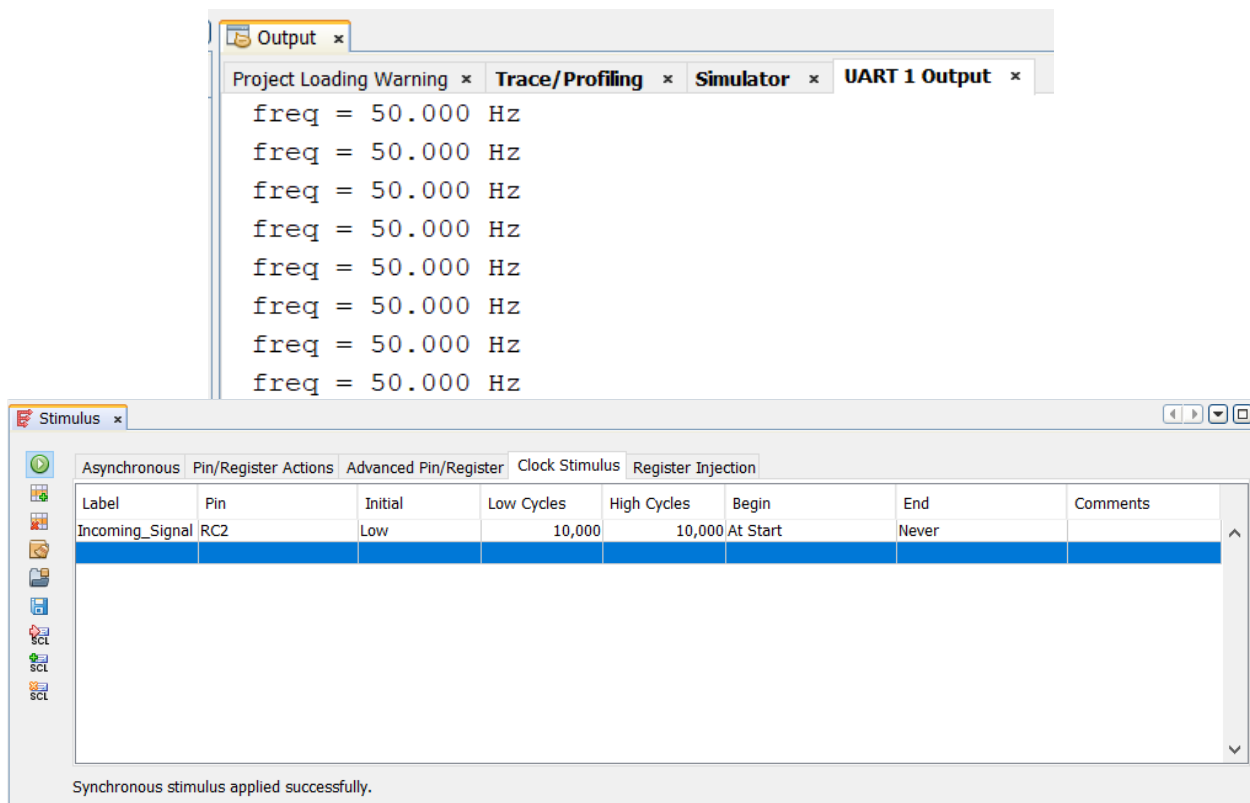
freq = 500.000 Hz
freq = 500.000 Hz
freq = 500.000 Hz
freq = 500.000 Hz
freq = 500.000 Hz
freq = 500.000 Hz
freq = 500.000 Hz
freq = 500.000 Hz

Stimulus

Asynchronous Pin/Register Actions Advanced Pin/Register Clock Stimulus Register Injection

Label	Pin	Initial	Low Cycles	High Cycles	Begin	End	Comments
Incoming_Signal	RC2	Low	1,000	1,000	At Start	Never	

Synchronous stimulus applied successfully.



Output

Project Loading Warning x Trace/Profiling x Simulator x UART 1 Output x

freq = 50.000 Hz
freq = 50.000 Hz
freq = 50.000 Hz
freq = 50.000 Hz
freq = 50.000 Hz
freq = 50.000 Hz
freq = 50.000 Hz
freq = 50.000 Hz

Stimulus

Asynchronous Pin/Register Actions Advanced Pin/Register Clock Stimulus Register Injection

Label	Pin	Initial	Low Cycles	High Cycles	Begin	End	Comments
Incoming_Signal	RC2	Low	10,000	10,000	At Start	Never	

Synchronous stimulus applied successfully.