

Assignment 2B (50 marks) – Lab Week Nine – Approximately 2.5 – 3.0 hours to complete

Due Dates:

This Assignment has a
Pre-Lab Component**Part A** (25 marks) Assignment 2B Quiz Questions – due Friday, end of Week 10 by 11:30 P.M. on Brightspace**Part B** (10 marks) – Document Submission of completed “Questions for A2B Displaying Values of LEDs video” document questions – due Friday, end of Week 10 by 11:30 P.M. on Brightspace.[Submission Link for Lab Week
Nine - Assignment 2B - Material](#)**Part C** (8 marks) – Code Submission of **Flash_PB7.asm** due Friday, end of Week 10 by 11:30 P.M. using the submission link on Brightspace – **DO NOT submit any code in a compressed file.** (Marking Rubric on Brightspace)**Part D** (7 marks) – DEMO of **Flash_PB7.asm** on Dragon 12 Plus Trainer board – due Week 9 or Week 10 during your lab period – no late demos will be permitted.Notes:

1. Part A will be normalized to a mark out of 2.5 on Brightspace as “Assignment 2B – Quiz.”
2. Parts B, C and D marks will be combined on Brightspace as “Assignment 2B – Code Submission”, normalized to a mark out of 2.5.
3. **Demos/submissions that do not include your Name and Student Number will not be assessed.**
4. **Submissions in compressed files will not be assessed.**
5. **Late Flash_PB7.asm demos/submissions will not be accepted and will receive a mark of zero (0).**

More Assembly Programming – Using the Dragon 12 PLUS Trainer, Completing Code and Tracing a Program**PURPOSE OF LAB:***The purpose of Part A of this lab is to confirm your knowledge of past course activities:*

- Week 5 - Introduction to Assembly Language Programming
- Week 6 - Flowcharting
- Week 7 - Assembly Language Branches and Iteration
- Week 7 - Assembly Language Addressing Modes
- Lecture material taught since Week 5 inclusive.

*The purpose of the remainder of this lab is to gain experience in Assembly Language using **AsmIDE** using the **Dragon12 & Student Mode Simulator** and the **Dragon 12 PLUS Trainer** hardware board by creating software that will display values on Light Emitting Diodes (LEDs).***PRELIMINARY WORK**

To communicate with the Dragon12-Plus board, a USB to RS232 driver must be installed **PRIOR TO YOUR LAB PERIOD** in accordance with the instructions detailed in the document to the right. Then, as soon as possible during Week 9's Lab, you should complete the instructions in that document to configure your computer with the Dragon 12 Plus Board, ensuring that you demo (non-credit) your results. Then, be ready to demo **Flash_PB7.asm** no later than at the **start** of Week 10's lab period.

Dragon12 Plus Board Setup and Use Procedures.pdf**Part A** (25 marks) – Lab Week 9 - Assignment 2B Quiz Questions[Lab Week Nine - A2B Quiz Questions](#)**Questions 1 – 10** of this part of the assignment tests your understanding of tracing through an Assembly Language program and evaluating the contents of Registers and Memory Address locations.**Question 11 – 25** tests your understanding of simple Assembly Language instructions as we have discussed in various lectures.

A reminder to use “\$” and “%” signs in your answers, as applicable, signifying a HEX value or an 8-bit BINARY value as appropriate. Note that 8-bit HEX values take the form of “\$A3” and 16-bit HEX values take the form of “\$1300” as examples, and 8-bit BINARY values take the form of “%00000101”, without the quotes in all cases.

In the same Brightspace module used for this assignment, I have included a non-credit Self-Assessment Quiz that will re-emphasize the use of “\$” and “%” signs for the entry of required values. The correct answers for each question will be presented upon completion of the quiz, which has unlimited attempts.

[A2BExampleQuiz](#)

Structured Programming in Assembly Language

This lab exercise provides the essence of structured programming in Assembly Language where programs' structure leads to software re-use and an easier understanding of how to solve problems using Assembly Language. Using a structured programming methodology will greatly assist you in not only the assignments in this course, but in industry as well where you will likely be tasked with developing portions of a software program that must be integrated into a larger problem solution. To that extent, you will be provided with Library files that will greatly assist you in Configuring the hardware/simulator that is used in this assignment.

Part B (10 marks) – Completion of “Questions for A2B Displaying Values on LEDs video” document questions

Prior to commencing Parts C and D of this assignment, view the Hybrid Lecture “A2B Displaying Values on LEDs, answering all the questions contained on the last page of this document, and submit it on Brightspace by its deadline.

A2B Displaying Values on LEDs

Video

Part C and D Programming Task One (Code: 8 marks, Demo 7 marks) – A Simple Hardware Programming Exercise: Flash_PB7.asm

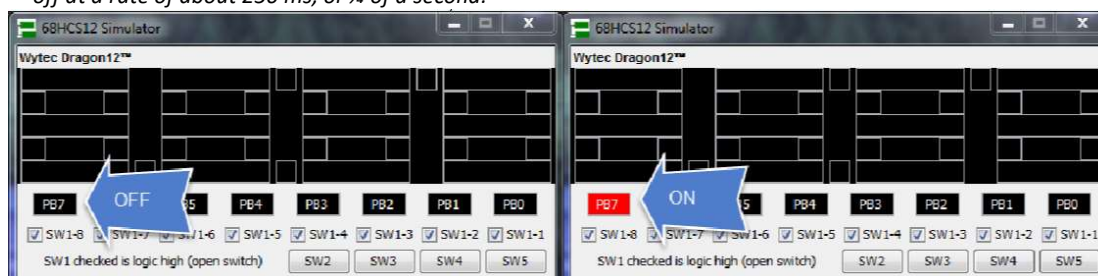
The purpose of the supplied assembly language program is to flash an LED on and off every 250 ms. First, we will use the simulator to observe the correct program run and then download the assembled program to the Dragon 12 Plus Trainer board.

- Download the compressed Library Files package (Library_Files.zip). Create a new folder called Lib in your C:\68HCS12 folder (this was the folder where you originally installed the assembly language software). Unzip Library_Files.zip into C:\68HCS12\Lib. The figure at the right illustrates what you should have after unzipping the files. **Ensure that you did not create a subfolder inside the Lib folder when you unzipped the files.**
- Download the corrupted Flash_PB7.asm into your Week Nine source directory (e.g. CST8216\Lab9) and correct the assembly language code source code listing so that the labels, opcodes, operands and comments are in the correct columns. You will also have to add the two missing lines of code that will include the two library files found in C:\68HCS12\Lib (read those files for instructions on how to use them – ensure that their "include" statements are placed just before the "end" statement in your code). Note that if you receive the following error: Fatal error -- Can't open #include file C:\68HCS12\registers.inc, then you incorrectly installed the software package and that you must de-install the software and then reinstall it into the correct path: C:\68HCS12.
- Once you have the previous steps complete, assemble Flash_PB7.asm ensuring that there are no errors or warnings.
- Load the .s19 file into the simulator and take the following action to run it:
 - Ensure “Tracing” is unchecked; otherwise, the LED will flash on and off only after several minutes of program run:
 - On the main menu, click on View than Parallel Ports
 - For your convenience, I have included a short video presentation on the expected behaviour of this program. Observe that the simulator and parallel ports are identical to the ones in that video. If they are not, then you are using the wrong simulator, and you will have to select the correct one before proceeding any further
 - Next, click the “Go” button on the simulator and observe that the Port B LED (PB7) on the simulator Parallel Port flashes on and off at a rate of about 250 ms, or ¼ of a second:



Flash_PB7 - Simulator

Video



Once the program run is correct in the simulator, demonstrate it on the Dragon 12 Plus Trainer board. Note that your timing may be slightly different from that illustrated in the video.

Flash PB7 - Dragon12 Plus Trainer Board

Video

Questions for A2B Displaying Values on LEDs Video (Hand-in sheet via Brightspace)

Name: _____

Student Number: _____

After including your Name and Student Number above, complete the following questions **in your own handwriting** (please print) and either scan your answers or submit a legible picture(s) of your solution within a MSWord or Adobe document. If your name and Student Number are not included, or your handwriting is illegible or your answers are typed, your submission will not be assessed.

Each question is worth one mark.

Questions	Answers
1. There are three ways to display data on the Dragon12 Plus hardware board. Which way is not supported on the Dragon12 Simulator?	
2. Which port on the Dragon12 Plus hardware board is used to display data on the LEDs.	
3. What is the name of the file that contains the Dragon12 Plus hardware memory map (Provide the complete path name where the file is located)?	
4. What logic level (0 or 1) must be applied to a Port Pin to turn on its LED, and what voltage is associated with that level?	
5. Why are the labels PB7ON and PB7OFF capitalized?	
6. What is the line of code that causes the program flow to execute the code in subroutine Config_SWs_and_LEDs?	
7. What instruction is used at the end of every subroutine to return program control to the calling file (e.g. the code that called the subroutine)?	
8. What is the line of code that causes the program flow to execute the code in subroutine Delay_ms?	
9. What register holds the delay value for subroutine Delay_ms?	
10. If we wanted a delay of 750 ms after displaying a value on portb, how many lines of code would be required to accomplish this task? Write those lines of code below, but do not use iteration in your solution.	