Laboratory 2
This Laboratory should take an average student 6 hours (2 lab sessions) to complete.

Task 1: Serial I/O

Objectives:
In this experiment you will learn how to use the serial functionality of the 68HC11.
At the end of the session you should know how to send bytes and strings using the SCI serial port on the EVBPlus board, and the use of a terminal emulator on the PC, to verify the data being sent.

Before the Lab:

- Read the 68HC11 reference material and familiarise yourself with the registers associated with serial I/O (SCI in particular), making sure you understand their purpose.
- Read Chapter 10 and Section 4.4 of your textbook by Spasov.
- Find a reference and read about the ASCII character set. A web search turns up many references.

Lab Exercise Part 1: Serial Output

Write a program that takes the two strings “What does AC stand for in ACDave?” and “<an answer of your choice>” stored in memory, and send them as separate lines out of the serial port at the rate of one message every second. Use the ASCII carriage return and line feed characters to separate the statements (question: what do these characters do?).

Hint: Make sure the terminal emulator (use Hyperterm supplied with Windows) and your software are using the same configuration (baud rate, start and stop bits, parity etc.)
Hint: Check Jumper J8. It should be on the pins marked ‘232’ (short for RS232 – the electrical protocol used)
Hint: Make sure you configure the serial port before sending data out
Hint: Make sure you understand the assembler pseudo-op FCC
Hint: Try to make your code general enough to handle strings of any length

Now you can change the strings to whatever you like. Nothing that would offend the tutors please… (they’re very sensitive!)
Extension: Serial Input

Write a program that polls the serial port (continuously checks to see if data is available). When data is available, write each byte to the Port B LED’s.

Hint: Use the PC terminal emulator to send data to the evaluation board
Hint: The serial input/output routines are another useful way of debugging. Particularly in situations where a monitor is not available

Question: What problems can you see arising from the use of polling when dealing with data input?
Task 2: Timing and Interrupts

Objectives:

In this experiment you will learn how to use the timer subsystem and interrupt capabilities of the Motorola 68HC11

At the end of the session you should know how to use the basic functionality of the timer system, and how to generate and capture interrupts.

Before the Lab:

- Read the 68HC11 reference material and familiarise yourself with the registers associated with the timer subsystem, and the interrupt vector table.
- Read Chapter 3 and Chapter 11 of your textbook by Spasov.

Lab Exercise Part 1: Timer System

Write a program that generates a PWM (Pulse Width Modulated) signal. The signal should have a period of 68 milliseconds and be output on a pin of your choice (as long as you use Port A!). Use the dip switches to input the duty cycle. Your program should be interrupt driven.

Hint: Make sure you configure the timer subsystem before your main routine
Hint: Spasov Chapter 11 should give you some very useful tips
Hint: Do you remember how to use the oscilloscope? Time to find out! – use it to verify your results

Extension:

Write a program that generates a PWM signal, as in the previous exercise. In addition, the program should monitor the serial port for data sent from the PC. By sending strings from the PC (in any format you choose), you should be able to adjust both the period of the PWM signal, and the duty cycle. Use interrupt driven serial routines.
Marking Scheme

Minimum 45

• Meets Specification? 25
• Quality of Solution 20
  o Code documentation
  o Code/algorithm structure

Extension 25

• Meets Specification? 15
• Quality of Solution 10
  o Code documentation
  o Code/algorithm structure

Record Keeping 30

• Lab Notebook (documentation) 100
  o Completeness
  o Conciseness
  o Clarity/Layout