

A.8 Chapter 8

1. The `vInterpretCommandTask` task must have some way of knowing when each command is over. The easiest way is probably to have `vGetCommandCharacter` write the carriage returns into the buffer.
2. No answer provided.

A.9 Chapter 9

Chapter 9 contains no problems.

A.10 Chapter 10

1. The first thing to try might be to connect to the chip enable pin on the UART and see if it ever goes low, enabling the microprocessor to access the UART. If not, that would indicate that the hardware is not working or that your software doesn't know the address range at which the UART appears in the address space. If the chip enable pin does go low, then the next thing to try might be to connect to the write (WR) pin on the UART and see if it ever goes low at the same time as the chip enable pin. You might be able to manage this with an oscilloscope, triggering it on the falling edge of the chip enable signal, but it will be easier to do with a logic analyzer (also triggering it on the falling edge of the chip enable signal). If the chip enable signal and the write signal both are low at the same time, then the next thing to do would be to hook up the eight data signals and the two address signals into the UART—now you will definitely need a logic analyzer—and capture the data that you are writing to the UART. You can use your logic analyzer in state mode and clock it on the rising edge of the write signal.
2. It would seem in this case that the interrupt signal from the UART is probably worth testing. If it is never asserted, it indicates that you have not set up the UART correctly to interrupt. If it is asserted, then the likely problem is that your software has not set the microprocessor up to jump to your interrupt routine.

A.11 Chapter 11

No answers provided for the problems in Chapter 11.