

Verification Strategy:

For part 1:

Verification of transmission is viable with either the Saleae logic analyzer or the Tera Term dumb terminal emulator. To verify transmission, I will have the transmitter send a continuous string of A's, and verify that the Tera Term is receiving them all. To make sure that none are missed, I will either increase the delay so as to check each transmission or open it with the Saleae logic analyzer to look at the waveform generated and measure the accuracy of the bit times.

For part 2:

For verification of reading UART data, I will set up a watch on the data variable inside the read function. Then I will set up a breakpoint on the statement at the end of the read function so I can check the value being returned. The rest is trivial, but I can also verify that the global character variable is being set correctly using the watch window. By sending data from the Tera Term, I can verify correct operation.

For part 3:

Similar to part 2, except I will set a breakpoint inside the ISR as well to make sure it is being called.

For part 4:

The verification for part 4 will be simple. Using the Tera Term, I will be able to type each character of the alphabet and make sure that I receive its uppercase equivalent. For non-alphabetic characters, the behavior of the program is undefined, as I just have it subtract 0x20 from the lowercase character for capitalization.

For part 5:

Similar to part 4, the Tera Term will be used to verify that I receive the same line that I send. I am not checking for overflow, but if I set the terminal to only 80 characters that won't be an issue. I am not adding a LF character to the string I send, so it will erase the line above each time, but if I want to change that it would be trivial. I will basically just compare input data to output data for verification of correct operation.