

Planejamento

Há duas maneiras de se resolver o problema, por polling e por interrupção. Foi escolhida a segunda opção para implementar uma melhor estrutura de tratamento de interrupção daquela já feita no laboratório 2.

A ideia principal é ter o tratamento de interrupção armazenar cada byte recebido, limpar a interrupção e indicar ao programa principal, que ficará responsável pelo tratamento do caractere (transformar letras maiúsculas em minúsculas) e envio do mesmo pela UART.

Estudo e solução

A UART que será utilizada é a UART0 que está integrada com o USB e se utilizando dos pinos 1 e 0 da GPIO A.

As interrupções da UART em recebimento podem ser em níveis de FIFO, sendo que o mínimo é 1/8 que corresponde a 2 bytes de dados, mas se desabilitada a FIFO, uma interrupção pode ser gerada a cada byte recebido, que é o objetivo. Então, apesar de ser bem improvável que se perca bytes pela falta de FIFO e o recebimento de bytes enquanto o caractere recebido não foi tratado, em um programa mais complexo, tal pode ocorrer. Então pode ser implementada um buffer que armazena poucos membros.

Com relação ao tratamento de caractere, se consultada a tabela ASCII, as letras maiúsculas são representadas por valores inteiros entre 65 e 90, e as minúsculas entre 97 e 122. Então na solução basta verificar se tal está em um conjunto de valores e somar 32 .

Testes

Houveram apenas dois problemas nos testes

- 1)Ao apertar a tecla de "Enter" é enviado o caractere carriage return (CR ou "\r") ao invés do Line Feed (LF ou "\n").
- 2)A chamada de UARTDisable(...), dentro de UARTConfigSetExpClk gerava um loop infinito por causa da tentativa de esvaziar a FIFO. Após desconectar e reconectar a placa ao USB, tal parou de ocorrer.