Relatório 7 - EA871

Mônica Aoki Faria RA:156787

Leonardo Rodrigues Marques RA:178610

1. Setamos o bit 10 de SIM\_SCGC4 para nível alto para habilitar o clock de UART0:

SET\_BIT(SIM\_SCGC4, 10);

Setamos o bit 9 de SIM\_SCGC5 para nível alto para habilitar o clock de PORTA. Fazemos isso pois o Rx e Tx da UART0 está associada ao PTA1 e PTA2 do PORTA.

SET\_BIT(SIM\_SCGC5, 9);

1. Configuramos o MUX do PORTA\_PCR1 e PORTA\_PCR2 para 010 pois, ao observar a tabela das configurações dos pinos na Seção 10.3.1 do manual do KL25, verificamos que a configuração como Tx e Rx é dada na Alternativa 2. é necessário escrever nestes registradores para inicializar a UART0 para associar o UART0 com PTA1 e PTA2 funcionando como Tx Rx.

SET\_BIT(PORTA\_PCR1, 9);

SET\_BIT(PORTA\_PCR2, 9);

1. É necessário escrever nesse registrador para selecionar como fonte do clock para a UART0 o sinal MCGFLLCLK que opera na frequência de 20.97 MHz.

SET\_BIT(SIM\_SOPT2, 26);

1. Para iniciar a UART0 para operar com uma taxa de 19200 bps:

Atribuímos um valor de OSR+1=16

baud rate = baud clock / ((OSR+1) × BR)

19200 = (20,97 × 10^6) / (16 × BR)

BR = 68

Atribuímos ao registrador UART0\_BDL o valor 1000100 (68 em binário) e ao SBR de UART0\_BDH zero.

1. Pooling é o conceito de esperar para transmitir e testar a UART para receber, isto é, você fica verificando quando o evento ocorre em um laço de espera.
2. Para verificar se o data buffer de transmissão está cheio ou vazio. Se ele estiver cheio, deve-se esperar até que ele fique vazio para escrever.
3. O TDRE indica se o data buffer de transmissão está cheio ou vazio enquanto o TC indica se a transmissão está sendo efetuada ou se ela já foi concluída.
4. Quando você implementa “while( !(UART0\_S1 & UART0\_S1\_TDRE\_MASK)) ;”, o código não faz nada além de verificar constantemente essa condição até que ela seja verdadeira. Se implementarmos “ while( !(UART0\_S1 & UART0\_S1\_TDRE\_MASK)) ” há a possibilidade do programa ficar executando a instrução seguinte até que a condição seja satisfeita.
5. A *flag* RDRF do registrador UART0\_S1 mostra se o registrador que recebe data está cheio ou vazio. Se ele estiver cheio, quer dizer que existem dados a serem lidos no registrador.
6. Se não testarmos a flag RDDF o programa poderá ficar tentando ler um registrador que está vazio e computar valores errados.