

PROJETO FINAL DE SISTEMAS MICROCONTROLADOS

Contexto:

Utilizar programação em linguagem Assembly para configurar periféricos integrados de um microcontrolador e implementar interface com o usuário para uma aplicação específica.

Objetivo:

Escrever um programa em linguagem Assembly para o kit EK-TM4C1294XL que implemente uma calculadora capaz de realizar as quatro operações aritméticas básicas (adição, subtração, multiplicação e divisão) com números inteiros de até três dígitos. A entrada e saída de dados para o usuário se dará por meio de comunicação pela porta serial do kit (COM virtual sobre USB) usando caracteres ASCII, de forma que um aplicativo emulador de terminal possa ser utilizado como interface, conforme diagrama em blocos ilustrado na Figura 1.

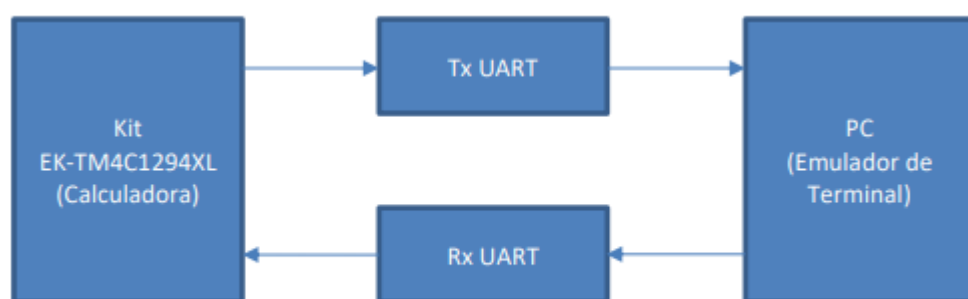


Figura 1: Diagrama em blocos do sistema a ser implementado.

Requisitos funcionais do sistema:

RF1: O sistema deverá receber o primeiro operando pela UART numa sequência de até três caracteres ASCII correspondentes aos dígitos decimais de 0 a 9.

RF1.1: O sistema deverá ecoar os caracteres recebidos, desde que sejam válidos, retransmitindo-os de volta pela UART para realimentação do usuário.

RF1.2: O sistema deverá converter os caracteres ASCII recebidos em um valor numérico equivalente para a realização da operação aritmética a ser selecionada em seguida.

RF2: O sistema deverá receber um caractere ASCII para selecionar a operação aritmética.

RF2.1: O caractere '+' deverá corresponder à operação aritmética de adição inteira.

RF2.2: O caractere '-' deverá corresponder à operação aritmética de subtração inteira.

RF2.3: O caractere '*' deverá corresponder à operação aritmética de multiplicação inteira.

RF2.4: O caractere '/' deverá corresponder à operação aritmética de divisão inteira.

RF2.5: A recepção de qualquer dos caracteres descritos nos requisitos RF2.1 a RF2.4 deve encerrar a recepção do primeiro operando caso o limite máximo de 4 dígitos não tenha sido atingido.

RF2.6: O sistema deverá ecoar o caractere recebido, desde que seja válido, retransmitindo-o de volta pela UART para realimentação do usuário.

RF3: O sistema deverá receber o segundo operando pela UART numa sequência de até três caracteres ASCII correspondentes aos dígitos decimais de 0 a 9.

RF3.1: O sistema deverá ecoar os caracteres recebidos, desde que sejam válidos, retransmitindo-os de volta pela UART para realimentação do usuário.

RF3.2: O sistema deverá converter os caracteres ASCII recebidos em um valor numérico equivalente para a realização da operação aritmética selecionada.

RF4: O sistema deverá receber o caractere '=' para efetuar a operação aritmética.

RF4.1: A recepção do caractere '=' deve encerrar a recepção do segundo operando caso o limite máximo de 3 dígitos não tenha sido atingido.

RF4.2: O sistema deverá ecoar o caractere recebido, desde que seja válido, retransmitindo-o de volta pela UART para realimentação do usuário.

RF5: O sistema deverá efetuar a operação aritmética selecionada e transmitir o resultado pela UART numa sequência de caracteres ASCII.

RF5.1: O sistema deverá converter o valor numérico do resultado da operação aritmética em uma sequência de caracteres que o represente.

RF5.2: Caso o resultado da operação seja negativo, o sistema deve transmitir o caractere '-' antes da sequência de caracteres que o representa.

RF5.3: A sequência de caracteres transmitidos deverá ser encerrada com os caracteres '\r' e '\n' (CR+LF).

Requisitos não-funcionais do sistema:

RNF1: A comunicação serial entre o kit de desenvolvimento e o PC hospedeiro deverá ocorrer em 9600 bps, com 8 bits de dados, paridade ímpar e 1 bit de parada.

RNF2: A estratégia de sincronização CPU-UART deverá ser por sondagem (polling)

Funcionalidades extras que podem ser implementadas:

- Indicar erro quando ocorrer uma divisão por zero.
- Permitir operações com números inteiros sinalizados.
- Utilizar os LEDs do kit para indicar estados da calculadora.