



Experimento 5 – Comunicação Serial

OBJETIVO:

Este roteiro tem como objetivo revisar conceitos de comunicação serial e treinar a configuração do módulo dedicado do MSP430. Em especial, iremos estudar o funcionamento de um dispositivo receptor e transmissor UART.

INTRODUÇÃO:

Comunicação serial é o processo de envio de dados de maneira sequencial, um bit após o outro através de uma linha de dados. Em sistemas embarcados, dispositivos com funcionalidade específica geralmente implementam algum tipo de comunicação serial para enviar ou receber dados e instruções. Dispositivos como acelerômetros, cartões de memória, GPS, sensores de temperatura, umidade e pressão, todos possuem internamente um microcontrolador que trata o dado bruto e o envia por uma linha serial. Diante disso, é de suma importância compreender as diferentes formas de comunicação serial para interfacear o microcontrolador com outros dispositivos.

A comunicação serial assíncrona é provavelmente a forma serial mais simples de todas. Sua popularidade vem justamente da sua simplicidade e facilidade de implementar usando lógica digital ou microcontroladores.

Para cada linha de transmissão, a comunicação está ociosa quando em nível alto. Para iniciar uma transmissão, é necessário puxar a linha para nível baixo e transmitir bit a bit sequencialmente, começando pelo bit menos significativo. A taxa de transferência é um acordo prévio, ou seja, ambas as partes devem estar configuradas para se comunicar na mesma velocidade.

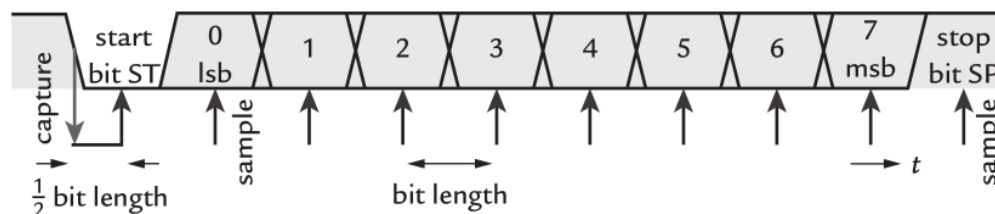


Figura 1: Comunicação serial assíncrona

Um dispositivo UART não permite comunicação bidirecional através de um único fio. São necessários pelo menos dois fios para uma comunicação full-duplex. Uma particularidade desta forma de comunicação é a ausência de sincronia entre o transmissor e receptor. Isso pode levar a erros de comunicação caso haja diferença significativa entre os clocks internos de cada dispositivo.

O MSP430 possui um módulo de comunicação serial dedicado. Isso torna o processo de comunicação tão simples quanto escrever num registro e esperar que o byte seja enviado pela linha serial. Isso nem sempre é o caso de microcontroladores mais simples. Nesses casos o protocolo deve ser implementado manualmente através de escritas nas portas do dispositivo. Esse tipo de implementação é chamado de *bit-banging* e exige cuidado extra com a temporização para que a linha não perca a sincronia.

Neste roteiro, iremos treinar o uso do módulo UART do MSP430 e implementaremos também o protocolo manualmente utilizando a técnica de *bit-banging*.



Programa 15:

Utilize a interface serial USCI-A0 do MSP430 para controlar os LEDs de outro microcontrolador. A ligação entre dois microcontroladores deve ser feita conforme a figura 2. Configure essa interface para se comunicar em 1200 de baud rate. Use as interrupções dos botões (portas P2.1 e P1.1) para enviar dados e as interrupções da USCI para receber dados e controlar os LEDs. A interface serial está roteada nos pinos P3.3 (TX) e P3.4 (RX). Não se esqueça de configurar esses pinos para sua funcionalidade dedicada.

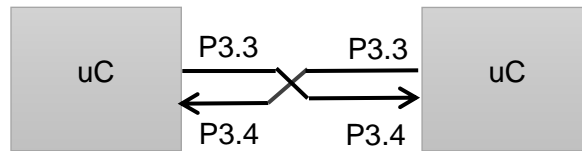


Figura 2 – Comunicação UART usando a interface USCI-A0

O receptor responder às seguintes palavras:

- 0x01 – Liga o LED **vermelho**
- 0x0A – **Apaga** o LED **vermelho**
- 0x10 – Liga o LED **verde**
- 0xA0 – **Apaga** o LED **verde**

O transmissor deve enviar apenas uma palavra a cada pressionar de botão:

- S1 – Envia **0x01** e **0x0A** alternadamente
- S2 – Envia **0x10** e **0xA0** alternadamente

Exemplo: Ao pressionar uma vez o botão S1 o transmissor deve enviar a palavra **0x01**. Ao pressionar uma segunda vez, o transmissor envia **0x0A**.

Programa 16:

Iremos repetir o exercício anterior usando a técnica de *bit-banging*. Este exercício é importante pois nem todos os microcontroladores disponíveis no mercado possuem interfaces de comunicação serial dedicada. Você pode usar os mesmos pinos do exercício anterior, porém em modo GPIO. Configure essa interface para se comunicar em 1200 de baud rate. A recepção dos dados deve ser feita no centro de cada bit, conforme a figura 1 para evitar erros de recepção caso os dispositivos estejam ligeiramente fora de sincronia. Implemente essa comunicação usando apenas escritas e leituras das portas de entrada e saída.

SUGESTÃO: Durante a fase de desenvolvimento do programa, você pode fechar um curto entre os pinos P3.3 e P3.4 e usar a mesma placa para transmitir e receber os comandos.

RELATÓRIO

O relatório é individual, e deve ser entregue impresso (ou feito à mão). Em hipótese alguma será admitida a entrega do relatório de forma eletrônica. Cada questão vale 2 pontos.

Questão 1: Quais são as limitações da comunicação com um UART? Explique como que o desvio de frequência entre o transmissor e receptor pode afetar a comunicação.

Questão 2: Explique como funciona a recepção de um sinal serial utilizando voto de maioria.

Questão 3: É possível conectar mais de dois dispositivos num barramento de comunicação serial assíncrona? Como?

Questão 4: Qual é a diferença de sinalização idle-line e address-bit?

Questão 5: Supondo que seja configurado o SMCLK com frequência numericamente igual aos últimos 6 dígitos de seu número de matrícula (substituindo todos os 0's por 1's) mostre os cálculos se operar a porta serial em 9600 bps. Indique o valor para os seguintes parâmetros: UCOS16, UCBRx, UCBRFx e UCBRSx