

**ATIVIDADE DE LABORATÓRIO 6 – COMUNICAÇÃO SERIAL**

**Sistemas Microprocessados II**

**Caíque Vendemiatti 122270487**

**José Miguel Montevani Óthon 1110241**

## **SOROCABA**

## **24/OUTUBRO/ 2014**

**ATIVIDADE DE LABORATÓRIO 6 – COMUNICAÇÃO SERIAL**

Relatório apresentado a Professora Dra. Marilza Antunes de Lemos, da disciplina de Sistemas Microprocessados II, da turma XI, do curso de Engenharia de Controle e Automação.

**UNESP**

**Sorocaba – 24/10/2014**

1. **Programa 1:**

O primeiro programa implementado e comentado tem como objetivo funcionar como um código de transmissão em assembly para 8051 para se transmitir informações de modo síncrono, com uma taxa de transmissão (ou baud rate) igual ao valor do clock sobre doze, por exemplo, no caso de um cristal mais comum de 12 MHz a taxa de transmissão seria de 1 MHz.

Para o funcionamento do código se foi utilizado o modo 0 do SCON, que permite a transmissão síncrona, juntamente de uma contagem progressiva linear no registrador A. O código também possui com uma sub-rotina de atraso e um teste que espera o bit TI receber o valor 1. O bit TI recebe este valor assim que o valor armazenado é transmitido pelo microprocessador. Logo em seguida o bit é novamente colocado em 0, para que esteja pronto para o próximo loop.

ORG 0H

LJMP INICIO

ORG 30H

INICIO: MOV A, #0H ; Inicia A com 0, que será futuramente usado em contagem que será enviada

MOV SCON, #00H ; Configura as portas seriais como modo 0

LOOP: MOV SBUF, A ; Armazena o valor de A antes de ser enviado

JNB TI, $ ; Espera até transmissão ser realizada

CLR TI ; Limpa a flag de transmissão

INC A ; Aumenta o valor de A em um, para prosseguir com a contagem

LCALL ATRASO

SJMP LOOP

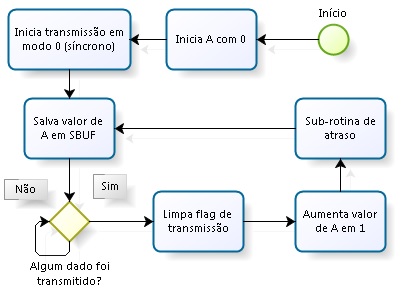
ATRASO:MOV R1, #10H ; Sub-rotina de atraso

ATR: MOV R2, #03H

DJNZ R2, $

DJNZ R1, ATR

RET



1. **Programa 2:**

O segundo programa implementado, assim como o primeiro, é um código em assembly feito para 8051, porém, diferente do programa anterior, este tem como objetivo funcionar como um receptor funcionando no modo 2 e não no modo 0. Sua taxa de transmissão (ou baud rate), considerando um cristal de 12 MHz, é de 1 MHz, sendo que poderia ser dobrada caso o bit SMOD fosse igual a 1. O cálculo da taxa de transmissão foi feito da mesma maneira que no primeiro programa, se dividindo o clock por 12.

Diferente do primeiro programa que utilizava da estratégia de pooling para funcionar, este trabalha utilizando interrupção serial para prosseguir com a recepção. O programa é iniciado habilitando recepções seriais em modo 2 através do SCON, em seguida também se é configurado o timer 1 em modo 2 e os bits necessários em IE para se habilitar interrupções de transferência serial, que serão ativadas sempre que uma informação for recebida. Quando a interrupção é ativada, o programa prossegue resetando a flag que indica que aconteceu uma recepção e transfere os dados recebidos em SBUF para a P1. Após a transferência a interrupção retorna para o programa principal, onde espera por outra transferência.

ORG 00H

LJMP INICIO

ORG 23H

CLR RI ; Limpa flag indicadora de interrupção

MOV P1, SBUF ; Coloca dado recebido e armazenado no registrador de dados seriais na P1

RETI ; Retorna ao programa principal

ORG 50H

INICIO: MOV SCON, #90H ; Habilita recepção de sinal serial no modo 2

MOV TMOD, #20H ; Habilita timer 1 no modo 2,

MOV TL1, #0FDH ; Seta valor inicial de timer

MOV TH1, #0FDH ; Seta valor de recarga do timer

MOV IE, #90H ; Habilita interrupção da transferência serial

SETB TR1 ; Dispara timer para geração de clock

SJMP $ ; Aguarda a ocorrência de interrupção serial, a qual ocorrera ao final da chegada do dado

END

