

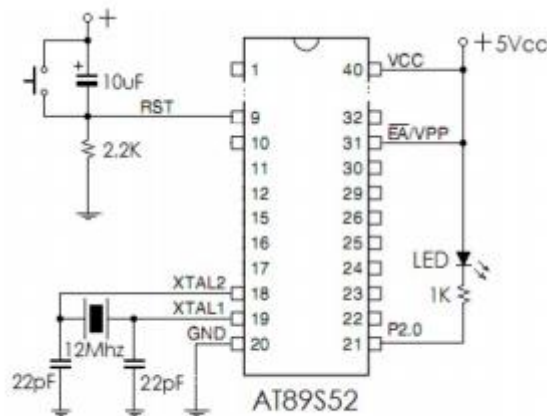
Universidade Federal de Roraima
Departamento de Ciência da Computação
Introdução a Sistemas Embarcados

Atividade - Aula 15

Prazo de Entrega: 05/04/2021

Aluno: André Leandro Schillreff dos Santos

[Questão – 01] Utilizando o esquema abaixo, escreva um programa para o microcontrolador 8051, tal que o Timer 0 seja programado para gerar interrupção a cada 5 segundos, onde o pino P2.0 terá seu nível lógico alterado (invertido). Assim, calcule o número de interrupções necessárias e apresente os seus resultados.



[Questão – 02] Utilizando como base o exemplo no slide da aula sobre comunicação serial, escreva um programa que tenha uma função denominada de putchar para imprimir (letra por letra) na porta serial o seu primeiro nome.

A solução dessa questão foi basicamente usar as funções que já estavam presentes no slide, com alterações nos caracteres que são passados na função putchar e também foi adicionado ao código uma função de delay.

O pino utilizado na solução foi o P3.0

```
1  #include <at89x52.h>
2  #include <serial232.h>
3
4  #define XTAL 12000000
5  #define T1_CLOCK XTAL/12
6  void init_sio_poll(unsigned int baud_rate){
7      TR1=0; //para Timer 1 para programá-lo.
8      ET1=0; //desabilita interrupção Timer1
9      TMOD &=0x0F; //Setup timer1 no modo 2
10     TMOD |=0x20; //8-bit auto-reload timer.
11     TH1=256-((T1_CLOCK/32)/ baud_rate); //valor da recarga
12     TR1=1; //timer1 no modo 'roda'
13     SCON=0x50; //Modo 1 com 8 bits
14     TI=1; //indicate TX buffer empty
15     //setado pelo microcontrolador com 0 na após
16     //transmissão
17 }
18
19 // função imprimir cchar na porta serial
20 void putchar(char outChar){
21     while(!TI);
22     SBUF = outChar;
23     TI = 0;
24 }
25
26 // função para esperar um determinado tempo
27 void delay(long int Vezes)
28 {
29     while (Vezes--);
30 }
31
```

Primeiro temos as funções que vão ser usadas no main, init_sio_poll, putchar , são as funções que vão ser responsáveis por imprimir os caracteres na porta serial, e a função delay é responsável por esperar um tempo para ficar melhor de poder ver a saída da porta serial.

```

32 void main(){
33     //sinaliza se a tecla foi pressionada e solta
34     bit foiPressionado = 0;
35     // inicializa com serial em 300bp/s
36     init_sio_poll(300);
37     // desliga todos os pinos P0
38     P3 = 0;
39
40     while(1){
41         // sinaliza que P3.0 foi pressionado
42         if(P3_0 == 1){
43             foiPressionado = 1;
44         }
45         // Se P3.0 foi solta, vai imprimir as letras na porta serial
46         if(P3_0 == 0 & foiPressionado == 1){
47             delay(1000000); // aguarda um tempo
48             putchar('L'); // transmite a letra 'L'
49             delay(1000000); // aguarda um tempo
50             putchar('E'); // transmite a letra 'E'
51             delay(1000000); // aguarda um tempo
52             putchar('A'); // transmite a letra 'A'
53             delay(1000000); // aguarda um tempo
54             putchar('N'); // transmite a letra 'N'
55             delay(1000000); // aguarda um tempo
56             putchar('D'); // transmite a letra 'D'
57             delay(1000000); // aguarda um tempo
58             putchar('R'); // transmite a letra 'R'
59             delay(1000000); // aguarda um tempo
60             putchar('O'); // transmite a letra 'O'
61             delay(1000000); // aguarda um tempo
62             putchar(' ');
63             foiPressionado = 0;
64         }
65     }
66 }

```

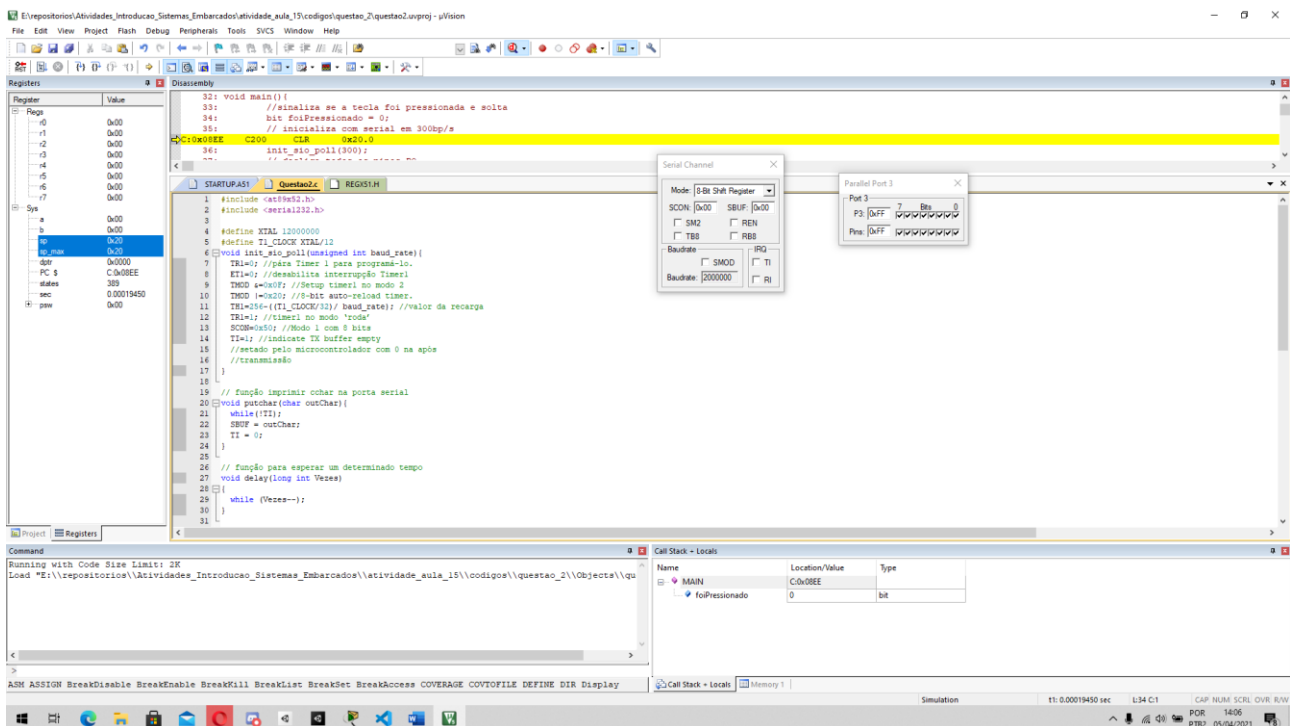
Em seguida temos a função principal, onde foi declarado um bit para sinalizar P3.0 foi pressionado, se P3.0 foi pressionado, foiPressionado vai receber valor 1, em seguida um se P3.0 foi solta e foiPressionado for igual a 1, vai imprimir na porta serial os caracteres passados na função putchar.

Para comprovar que o código está funcionando vamos comparar a saída da porta serial, com os valores em hexadecimal dos caracteres informados.

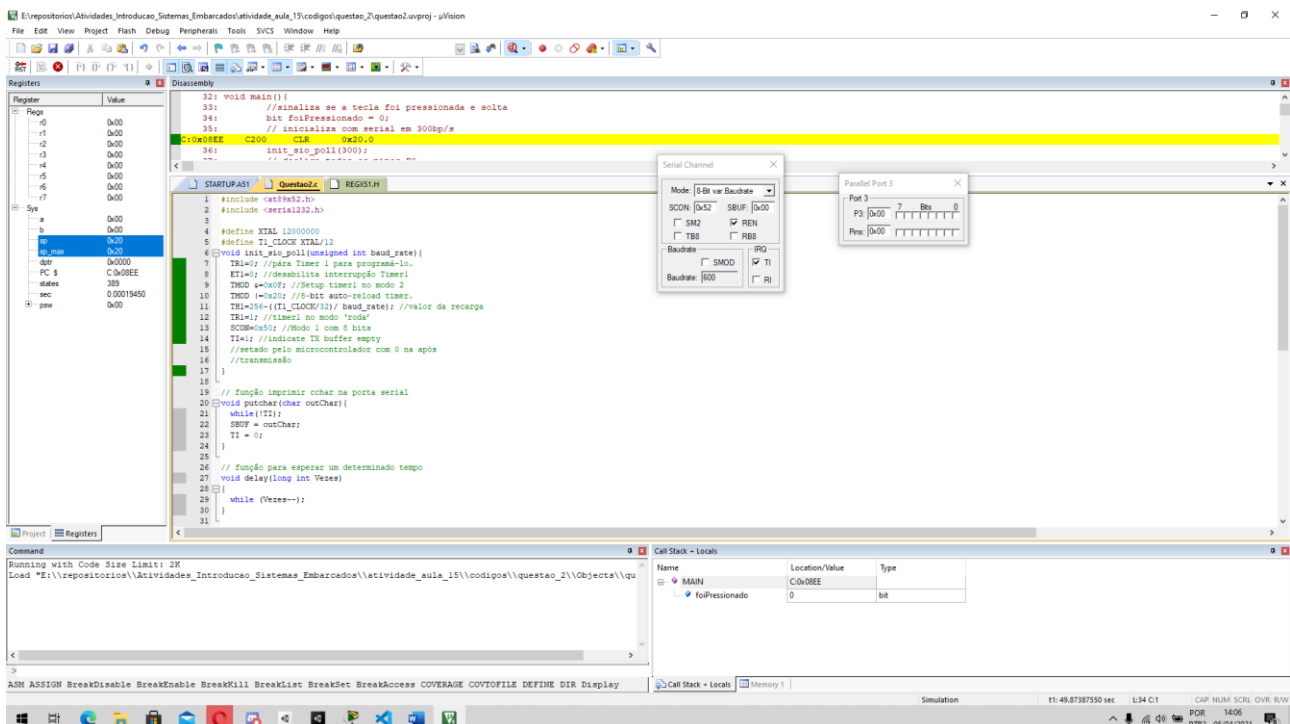
Valor em hexadecimal da tabela ASCII, das letras que formam o nome “LEANDRO”:

L = 0x4c
 E = 0x45
 A = 0x41
 N = 0x4e
 D = 0x44
 R = 0x52

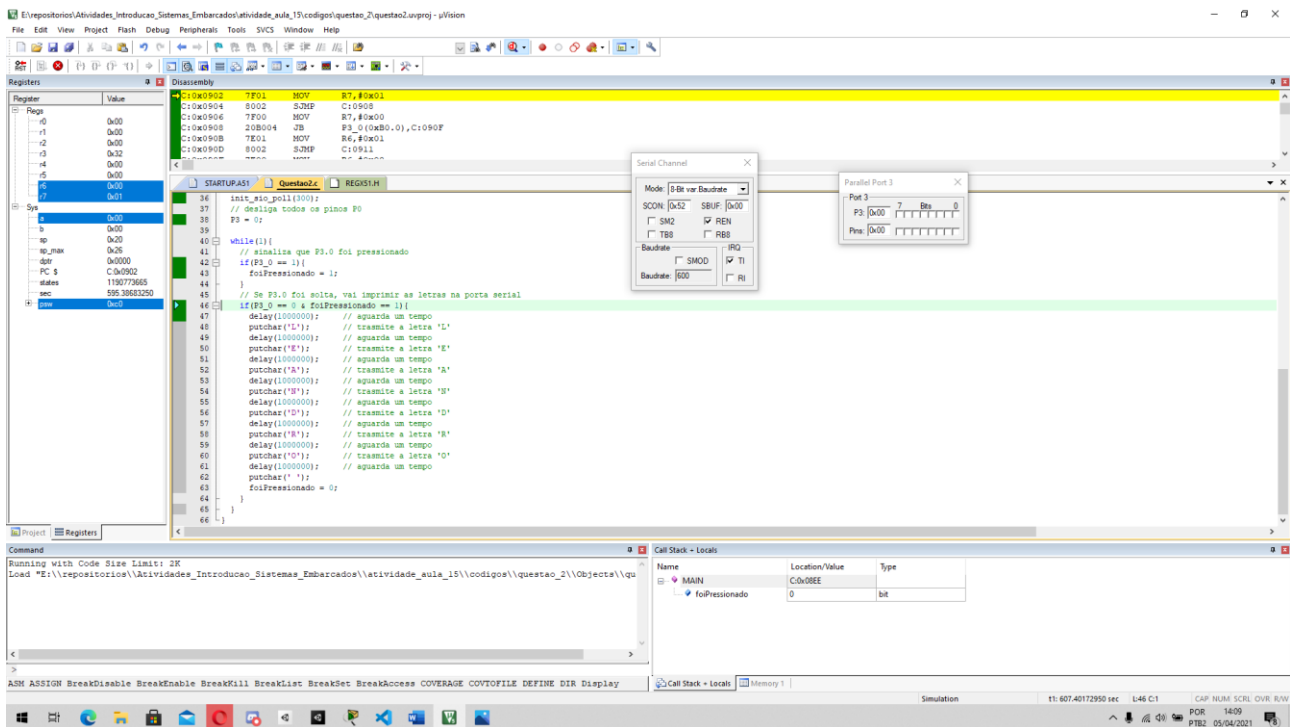
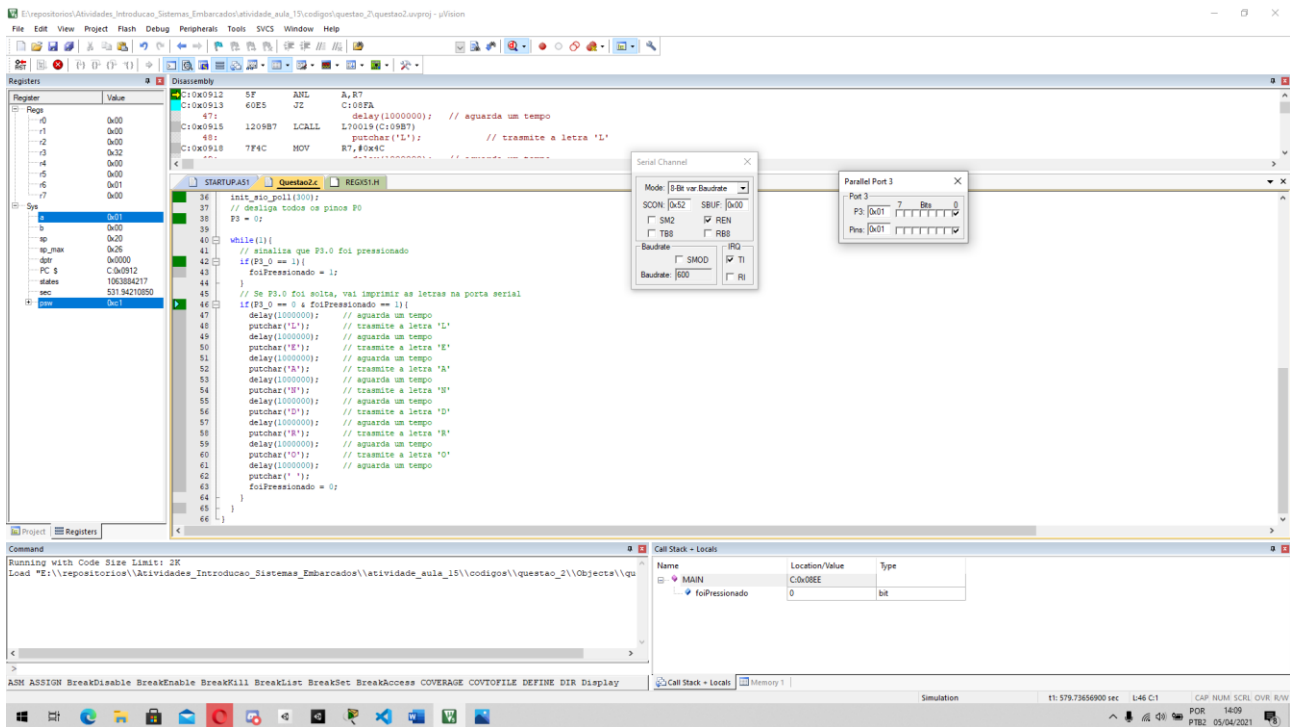
O = 0x4f

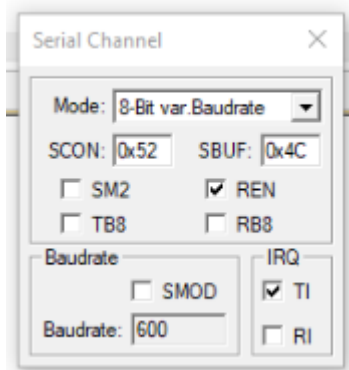


Primeiro temos a tela inicial da execução do programa



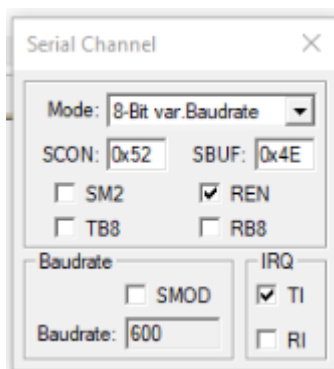
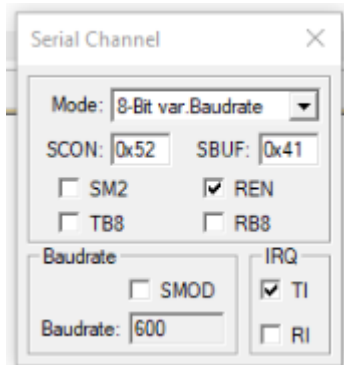
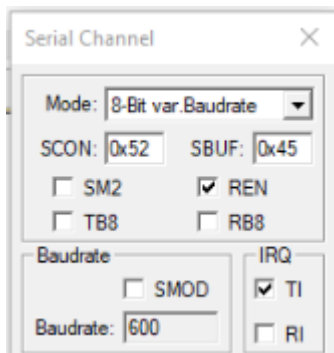
Em seguida podemos ver que quando o programa entra em execução os pinos P3 são todos zerados.





Podemos ver que a primeira saída no SBUF é o valor em hexadecimal da letra L.

Em seguida após aguardar um tempo da função delay são imprimidas os outros caracteres.



Serial Channel

Mode: 8-Bit var.Baudrate

SCON: 0x52 SBUF: 0x44

☐ SM2 ☒ REN

☐ TB8 ☐ RB8

Baudrate

☐ SMOD

Baudrate: 600

IRQ

☒ TI

☐ RI

Serial Channel

Mode: 8-Bit var.Baudrate

SCON: 0x52 SBUF: 0x52

☐ SM2 ☒ REN

☐ TB8 ☐ RB8

Baudrate

☐ SMOD

Baudrate: 600

IRQ

☒ TI

☐ RI

Serial Channel

Mode: 8-Bit var.Baudrate

SCON: 0x52 SBUF: 0x4F

☐ SM2 ☒ REN

☐ TB8 ☐ RB8

Baudrate

☐ SMOD

Baudrate: 600

IRQ

☒ TI

☐ RI

Serial Channel

Mode: 8-Bit var.Baudrate

SCON: 0x52 SBUF: 0x20

☐ SM2 ☒ REN

☐ TB8 ☐ RB8

Baudrate

☐ SMOD

Baudrate: 600

IRQ

☒ TI

☐ RI