



# Laboratório de 8051

16/10/2018

---

Aluno: Tarlison Sander Lima Brito

Matrícula: 2017013008

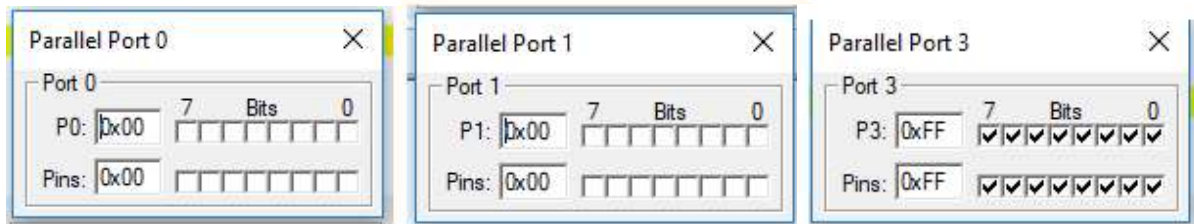
## Questão 1 - Máquina de chá e café

### 1.1 - Código do Programa:

```
main.c
1#include<at89x52.h>
2#define Tea P0_1
3#define Coffee P0_2
4#define Glass P1
5const LIGADO = 1;
6const DESLIGADO = 0;
7
8void delay(unsigned int time_ms){
9    unsigned int i, j;
10    for (i = 0; i < time_ms; i++){
11        for (j = 0; j < 12750; j++){
12        }
13    }
14}
15void coinInterrupt() interrupt 0 { // Séra chamada quando a moeda for colocado, ou seja a porta da interrupção P3_2;
16    if((Tea == LIGADO) && (Coffee == DESLIGADO)){
17        delay(800);           // pequeno tempo para a máquina identificar o que será feito;
18        Glass = LIGADO;       // O copo é ligado para encher;
19        delay(4000);          //tempo em que o copo enche;
20        Glass = DESLIGADO;    //Copo cheio;
21        Tea = DESLIGADO;      //Desliga o pino do Chá;
22        P3_2 = 1;             //Volta ao estado inicial para esperar outra moeda;
23    }
24    else if((Coffee == LIGADO) && (Tea == DESLIGADO)){
25        delay(800);           // pequeno tempo para a máquina identificar o que será feito;
26        Glass = LIGADO;       // O copo é ligado para encher;
27        delay(4000);          //tempo em que o copo enche;
28        Glass = DESLIGADO;    //Copo cheio;
29        Coffee = DESLIGADO;   //Desliga o pino do café;
30        P3_2 = 1;             //Volta ao estado inicial para esperar outra moeda;
31    }
32}
33
34int main(){
35    EA = 1; //habilita a chave geral das interrupções;
36    EX0 = 1; //habilita a interrupção 0;
37    P0 = DESLIGADO;
38    P1 = DESLIGADO;
39    while(1){
40        //Esperando moeda;
41    }
42}
```

## 1.2 Portas utilizadas e testes feitos:

Essas são as portas assim que começar a execução do programa(até chegar na linha do while(1) ).



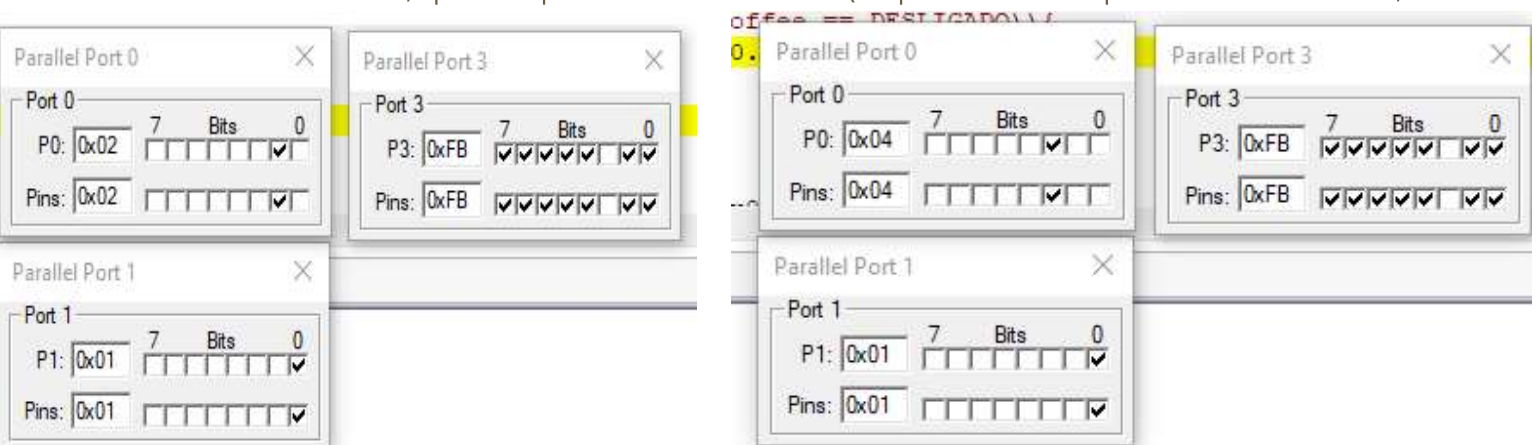
Portas: P0\_1 será ativada quando chá for selecionado;

P0\_2 será ativada quando o café for selecionado;

P1\_0 será ativada quando o copo estiver enchendo;

P3\_2 será a moeda, funcionando como interrupção(interrupt 0);

Quando a interrupção (P3\_2) for ativada, o programa chamará a função de interrupção, e se P0\_1 for ativado, a máquina irá identificar que será chá para fazer e então a porta P1\_0 ficará ativada enquanto o copo começará a encher(P1\_0 Ficarà ativado para mostrar isso), depois que o copo encher tudo volta ao estado inicial esperando outra moeda(Ou seja, P0\_1, P1\_0, e P3\_2 voltam ao estado inicial da imagem acima). Assim também acontece com o café. Aqui embaixo temos a foto do que acontece quando está sendo feito o chá, em que o pino do chá fica ligado junto ao pino do copo para indicar, como dito antes, que o copo está sendo enchido(O que é o mesmo procedimento do café,



apenas muda as portas usadas). E Logo (imagens acima mostram a execução do programa) temos respectivamente as portas quando o chá é selecionado para ser feito e ao lado quando o café é selecionado(programa funcionando corretamente).

## Questão 2 - Sistema de Embalagem de Produtos

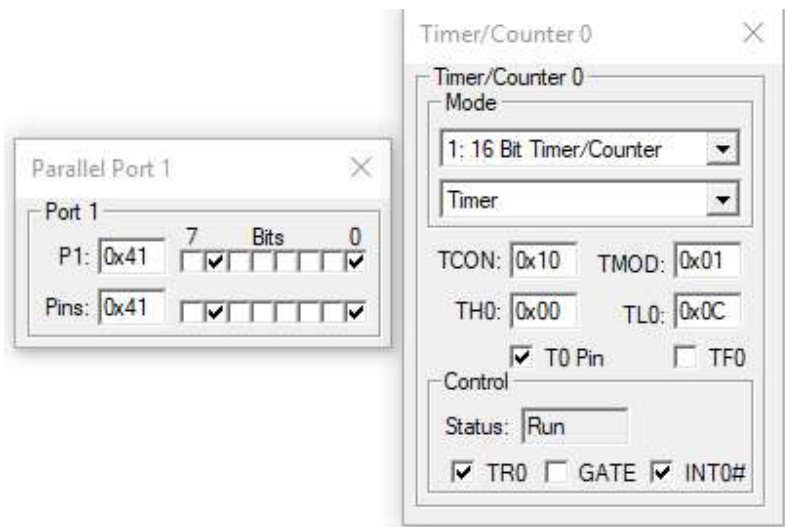
### 2.1 Código do Programa

```
main.c
1#include <at89x52.h>
2#define Esteira_produto P1_0
3#define Esteira_caixa P1_6
4#define AndaEsteira_caixa P1_7
5#define ParaEsteira_produto P1_1
6const LIGADO = 1;
7const DESLIGADO = 0;
8int contador = 0;
9
10void caixaCheiaInterrupt() interrupt 1{
11    TR0 = 0; //pausa timer para recarregar (Timer Stop)
12    TH0 = 0x3C; //recarrega o timer com o valor inicial de
13    TL0 = 0xAF; //15.535
14    TR0 = 1; //ativa timer para contar (Timer RUN)
15    contador++;
16    if(contador == 2500){ //Quando o contador for igual a 600, a caixa estará cheia então ocorre a parada da Esteira_Produto
17        ParaEsteira_produto = LIGADO; // Esteira do produto para, para a Esteira da caixa andar e a caixa trocar
18        AndaEsteira_caixa = LIGADO; // Esteira da caixa anda para trocar a caixa
19        Esteira_produto = DESLIGADO; //Pino sinalizador da esteira produto em movimento é desligado;
20        Esteira_caixa = DESLIGADO; //Pino sinalizador da esteira caixa parada é desligado;
21    }else if(contador == 5000){
22        ParaEsteira_produto = DESLIGADO; // Agora que a caixa já andou e a Esteira do produto pode andar, desligamos esse pino;
23        AndaEsteira_caixa = DESLIGADO; // Agora que a caixa já andou desligamos esse pino;
24        Esteira_produto = LIGADO; // Pino sinalizador de Esteira produto em movimento é ligado novamente;
25        Esteira_caixa = LIGADO; // Pino sinalizador de Esteira da caixa parada é ligado;
26        contador = 0; //zerando o contador para reiniciar a contagem de produtos na caixa;
27    }
28}
29
30int main(){
31    EA = 1;
32    ET0 = 1;
33    TMOD = 1;
34    TR0 = 1;
35    P1 = DESLIGADO; //Desligando todos os pinos da porta P1, para ficar visível a operação nas portas;
36    Esteira_produto = LIGADO; //Pino Sinalizador de esteira produto em movimento é ativado;
37    Esteira_caixa = LIGADO; //Pino Sinalizador de esteira caixa parada é ativado;
38    while(1){
39    }
40}
```



## 2.2 Portas utilizadas e testes feitos:

Essas são as portas assim que começa a execução do programa(até chegar na linha do `while(1)`).



Portas: P1\_0 será a porta para indicar que a Esteira dos produtos estão andando;

P1\_1 será a porta para indicar que a Esteira dos produtos está parada;

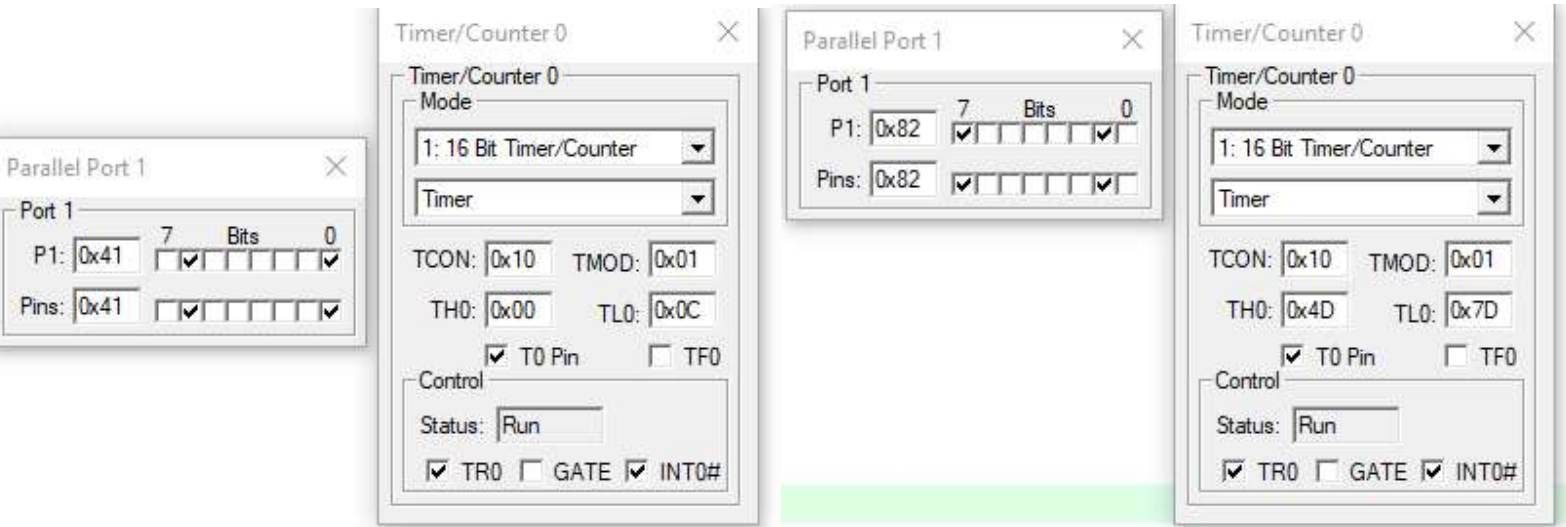
P1\_6 será a porta para indicar que a Esteira da caixa está parada, recebendo os produtos;

P1\_7 será a porta para indicar que a Esteira da caixa andou para que continuasse recebendo produtos, pois estava cheia;

E a janela que acompanha o timer também está presente pois foi usado a interrupção de tempo para o programa funcionar.

O programa ao ser iniciado liga os sinalizadores de que a esteira do produto está em movimento e preenchendo a caixa (indicado por P1\_0 ativado), que em paralelo possui a esteira das caixas que está parada recebendo os produtos que é indicado por P1\_6 ativado. Quando a caixa atingem a quantidade máxima de produtos, os sinalizadores de que a esteira dos produtos está em movimento e que a esteira das caixas está parada são desligados(P1\_0 e P1\_6 são desligados) e P1\_1 e P1\_7 são ativados pois indicam, respectivamente, que a Esteira de Produtos está parada e esperando pela troca de caixa e, P1\_7, que a Esteira de Caixas andou pro lado para trocar a caixa que estava cheia e colocar uma vazia para receber os outros produtos. E após isso P1\_1 e P1\_7 são desligados e P1\_0

e P1\_6 são ligados novamente pois agora a Esteira de produtos está enchendo a caixa normalmente como antes, até que encha de novo e o processo falado se repita.



Nesta duas imagens acima temos a execução do programa. A esquerda temos quando a esteira, de produtos, está preenchendo a caixa com os produtos e a direita temos quando a caixa enche e é preciso ser trocada e a esteira de produtos para e a esteira das caixas anda.

## Questão 3 - Multiplexar a gravação em VCR

### 3.1 Código do programa:

main.c\*

```

1 #include <at89x52.h>
2 #define camera1 P1_0
3 #define camera2 P1_1
4 #define camera3 P1_2
5 #define camera4 P1_3
6 #define sala1 P0_0
7 #define sala2 P0_1
8 #define sala3 P0_2
9 #define sala4 P0_3
10 const LIGADO = 1;
11 const DESLIGADO = 0;
12 int contador = 0;
13 void alarme() interrupt 0{
14     if(sala1 == 1 || sala2 == 1 || sala3 == 1 || sala4 == 1 ){
15         P3_1 = ~P3_1;
16         P3_3 = ~P3_3;
17         P3_3 = ~P3_4;
18         P3_5 = ~P3_5;
19         P3_6 = ~P3_6;
20         P3_7 = ~P3_7;
21     }
22 }
23 void pauseVCR(){
24     while(sala1 == 0 && sala2 == 0 && sala3 == 0 && sala4 == 0){ //verifica se existe alguém em alguma das alas;
25         camera1 = DESLIGADO;
26         camera2 = DESLIGADO;
27         camera3 = DESLIGADO;
28         camera4 = DESLIGADO;
29         contador = 0;
30         if(sala1 == 1 && sala2 == 0 && sala3 == 0 && sala4 == 0){
31             contador = 60; //alguém foi detectado na camera 1, pule para la;
32             break;
33         }
34         else if(sala1 == 0 && sala2 == 1 && sala3 == 0 && sala4 == 0){
35             contador = 120; //alguém foi detectado na camera 2, pule para la;
36             break;
37         }
38         else if(sala1 == 0 && sala2 == 0 && sala3 == 1 && sala4 == 0){
39             contador = 180; //alguém foi detectado na camera 3, pule para la;
40             break;
41         }
42         else if(sala1 == 0 && sala2 == 0 && sala3 == 0 && sala4 == 1){
43             contador = 220; //alguém foi detectado na camera 4, pule para la;
44             break;
45         }
46     }
47 }
48 void CamarasInterrupt() interrupt 1{
49     TR0 = 0; //pausa timer para recarregar (Timer Stop)
50     TH0 = 0x3C; //recarrega o timer com o valor inicial de
51     TL0 = 0xAF; //15.535
52     TR0 = 1; //ativa timer para contar (Timer RUN)
53     if(contador == 60){ //se passou 50ms x 60 = 3 segundos
54         camera1 = LIGADO; //Liga a camera um e desliga as outras

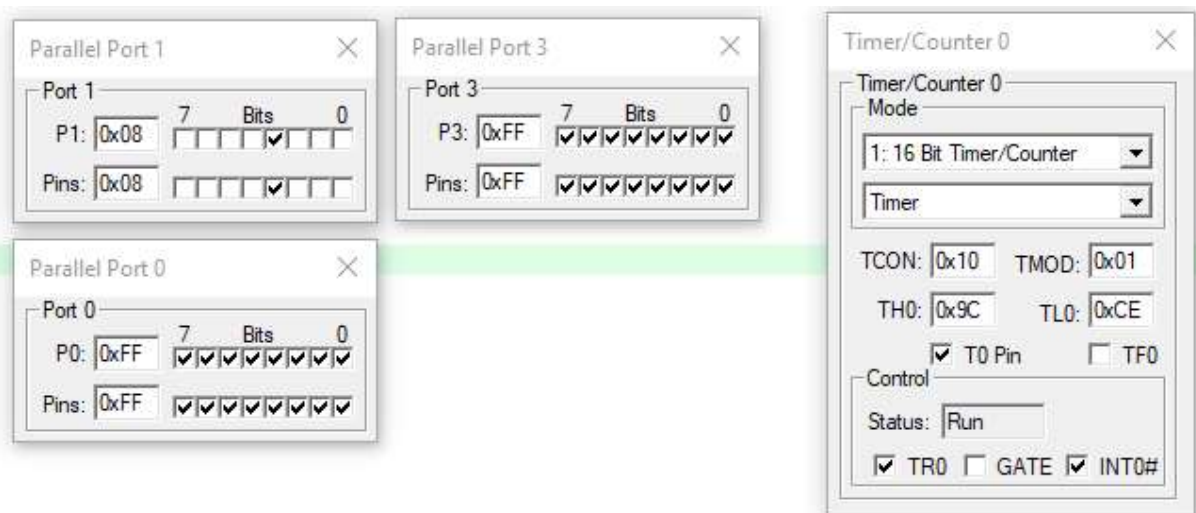
```

```

54  if(contador == 60){ //se passou 50ms x 60 = 3 segundos
55      camera1 = LIGADO; //Liga a camera um e desliga as outras
56      camera2 = DESLIGADO;
57      camera3 = DESLIGADO;
58      camera4 = DESLIGADO;
59
60  }else if(contador == 120){ //se passou 50ms x 120 = 6 segundos
61      camera2 = LIGADO; //liga camera dois e desliga as outras
62      camera1 = DESLIGADO;
63      camera3 = DESLIGADO;
64      camera4 = DESLIGADO;
65  } else if(contador == 180){ //se passou 50ms x 180 = 9 segundos
66      camera3 = LIGADO; //liga a camera tres e desliga as outras
67      camera1 = DESLIGADO;
68      camera2 = DESLIGADO;
69      camera4 = DESLIGADO;
70  }else if(contador == 240){ //se passou 50ms x 240 = 12 segundos
71      camera4 = LIGADO; //liga a camera quatro e desliga as outras
72      camera1 = DESLIGADO;
73      camera2 = DESLIGADO;
74      camera3 = DESLIGADO;
75      contador = 0;
76  }
77  contador++;
78  //verificação se há alguém ou não na sala
79  pauseVCR();
80  }
81  int main(){
82      EA = 1;
83      EX0 = 1;
84      ET0 = 1;
85      TMOD = 1;
86      TR0 = 1;
87      P1 = DESLIGADO;
88      while(1){
89      }
90
91  }

```

### 3.2 Portas utilizadas e testes feitos:





Obs.: O print foi tirado do programa em execução logo, só é possível perceber a movimentação de um pino no print, mas as portas P1\_0, P1\_1, P1\_2, P1\_3 estão piscando pois está indicando as câmeras gravando suas respectivas salas.

Portas: P1\_0, P1\_1, P1\_2, P1\_3 estão sendo utilizadas para mostrar as câmeras e quando estão ligadas ou desligadas;

P0\_0, P0\_1, P0\_2, P0\_3 estão sendo utilizadas como os sensores que irão mostrar se há alguém na sala ou não;

Pinos da porta P3 são utilizados para o alarme, o pino P3\_2 para a interrupção que indicará que o alarme foi ativado e as outras portas piscarão caso o alarme seja acionado;

E a janela que acompanha o timer também está presente pois foi usado a interrupção de tempo para o programa funcionar.

O programa assim que iniciado às câmeras fazem uma verificação de todas as salas e após isso verificam se ainda possui alguém na sala ou não, e se detectarem que não a função `pauseVCR()` fará com que todas as câmeras desliguem e assim o sistema fica em "alerta" verificando todas as salas e quando for detectado alguma presença o `pauseVCR()` irá setar o valor da variável global 'contador' para que assim que entrar no timer interrupt a câmera que possui o valor do contador X irá ligar, por exemplo alguém entra na sala um, a função do `pauseVCR()` detecta e coloca o contador = 60 e sai da função do `pauseVCR()` e assim a partir do momento que entrar na função de `CamerasInterrupt()` ele irá ligar a sua respectiva câmara no caso a câmera um. E assim que alguém sair da loja e todos os sensores não estiverem detectando ninguém o pino P3\_2 é ativado e a interrupção também, sinalizando que o alarme está ativado e se for detectado alguém na sala os pinos da porta P3 irão piscar sinalizando que o alarme foi ativado, o alarme é desativado quando a interrupção é desativada. Para melhor visualização é recomendável a execução do código já que o fato de estarem sempre piscando impossibilita o print.