

LABORATÓRIO DE INTRODUÇÃO A SISTEMAS EMBARCADOS

Débora Bianca Taveira de Moura

2018.2

QUESTÃO 1

MÁQUINA DE CHÁ E CAFÉ

1. CÓDIGO DO PROGRAMA

```
#include <at89x52.h>
#define tea P2_0
#define coffe P2_1
#define glass P1_0
#define fill P1_1

void delay(ms){ //função que define o delay entre ações
    int i;
    for (i=0;i<=ms;i++){
        ;
    }
}

void insereMoeda() interrupt 0{
    if((tea == 1)&&(coffe == 0)){ //cha e selecionado
        glass = 1; //xicara reconhecida
        delay(100000);
        fill = 1; //enche a xicara
        delay(100000);
        fill = 0; //para de encher a xicara
        glass = 0; //retira a xicara
        tea = 0; //maquina volta ao estado inicial
    } else if ((tea == 0)&&(coffe == 1)){
        glass = 1; //xicara reconhecida
        delay(100000);
        fill = 1; //enche a xicara
        delay(100000);
        fill = 0; //para de encher a xicara
        glass = 0; //retira a xicara
        coffe = 0; //maquina volta ao estado inicial
    }
}
```

```
int main(){
    P1=0;
    P2=0;
    EA=1;
    EX0=1;

    while(1){
        ;
    }
}
```

2. PORTAS UTILIZADAS

Parallel Port 1

Port 1

P1: 0x00

Pins: 0x00

Parallel Port 2

Port 2

P2: 0x00

Pins: 0x00

Parallel Port 3

Port 3

P3: 0xFB

Pins: 0xFB

PORTA P1

P1_0 é ativada enquanto a xícara estiver na máquina;

P1_1 é ativada enquanto a máquina enche a xícara;

PORTA P2

P2_0 é utilizada para representar a escolha do **chá**;

P2_1 é utilizada para representar a escolha do **café**;

PORTA P3

P3_2 é a interrupção externa;

As portas **P1** e **P2** iniciadas com o valor 0;

3. EXECUÇÃO

Parallel Port 1 ×

Port 1

P1: 0x03 7 Bits 0 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☒

Pins: 0x03 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☒

Parallel Port 2 ×

Port 2

P2: 0x01 7 Bits 0 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☐

Pins: 0x01 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☐

Parallel Port 3 ×

Port 3

P3: 0xFB 7 Bits 0 ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒

Pins: 0xFB ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒

```
void insereMoeda() interrupt 0{
  if((tea == 1)&&(coffe == 0)){ //cha e selecionado
    glass = 1; //xicara reconhecida
    delay(100000);
    fill = 1; //enche a xicara
    delay(100000);
    fill = 0; //para de encher a xicara
    glass = 0; //retira a xicara
    tea = 0; //maquina volta ao estado inicial
  } else if ((tea == 0)&&(coffe == 1)){
    glass = 1; //xicara reconhecida
    delay(100000);
    fill = 1; //enche a xicara
    delay(100000);
    fill = 0; //para de encher a xicara
    glass = 0; //retira a xicara
    coffe = 0; //maquina volta ao estado inicial
  }
}
```

1. Quando a moeda é inserida (P3_2), você seleciona a bebida: café (P2_1) ou chá (P2_0);

2. A xícara é inserida (P1_0);

3. A máquina enche a xícara com a bebida (P1_1) selecionada durante um delay;

4. Em seguida ela zera as portas P1_0, P1_1 e P2, assim, finalizando a execução e aguarda uma outra moeda.

QUESTÃO 2

ESTEIRA DE PRODUTOS

1. CÓDIGO DO PROGRAMA

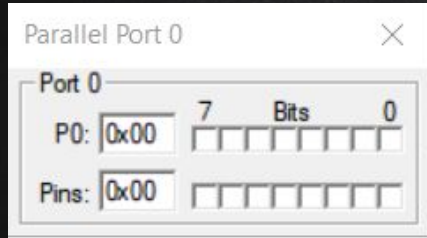
```
#include <at89x52.h>
#define esteira_caixa P0_0
#define esteira_prod P0_1
int prod, caixa, contador = 0;

void Addproduto() interrupt 1{
    TR0=0; //pausa timer para recarregar (Timer Stop)
    TH0=0x3C; //recarrega o timer com o valor inicial de
    TL0=0xAF; //15.535
    TR0=1; //ativa timer para contar (Timer RUN)
    prod++;
    if(prod==120){ //se passou 50ms x20= 1 segundo
        esteira_prod = 0;
        esteira_caixa = 1;
        contador++;
        prod = 0;
    }
}
```

```
void Addcaixa() interrupt 3 {
    TR1=0; //pausa timer para recarregar (Timer Stop)
    TH1=0x3C; //recarrega o timer com o valor inicial de
    TL1=0xAF; //15.535
    TR1=1; //ativa timer para contar (Timer RUN)
    caixa++;
    if(caixa==20){ //se passou 50ms x20= 1 segundo
        esteira_prod = 1;
        esteira_caixa = 0;
        caixa = 0;
    }
}

void main(){ //declara inicio do programa
    P0 = 0;
    EA=1; //habilita chave geral das interrupções
    ET0=1; //habilita a interrupção por estouro de Timer0
    ET1=1; //habilita a interrupção por estouro de Timer1
    TMOD=1; //habilita Timer0 no modo 16 bits
    TR0=1; //Ativa Timer0
    while(1){
        if (contador == 10){
            contador = 0;
            TR1=1;
        }
    }
}
```

2. PORTAS UTILIZADAS



PORTA P0

P0_0 é ativada enquanto a esteira de caixas está sendo movimentada

P0_1 é ativada enquanto a esteira de produtos está sendo movimentada

3. EXECUÇÃO

```
void Addproduto() interrupt 1{
    TR0=0; //pausa timer para recarregar (Timer Stop)
    TH0=0x3C; //recarrega o timer com o valor inicial de
    TL0=0xAF; //15.535
    TR0=1; //ativa timer para contar (Timer RUN)
    prod++;
    if(prod==120){ //se passou 50ms x20= 1 segundo
        esteira_prod = 0;
        esteira_caixa = 1;
        contador++;
        prod = 0;
    }
}

void Addcaixa() interrupt 3 {
    TR1=0; //pausa timer para recarregar (Timer Stop)
    TH1=0x3C; //recarrega o timer com o valor inicial de
    TL1=0xAF; //15.535
    TR1=1; //ativa timer para contar (Timer RUN)
    caixa++;
    if(caixa==20){ //se passou 50ms x20= 1 segundo
        esteira_prod = 1;
        esteira_caixa = 0;
        caixa = 0;
    }
}
```

1. Inicialmente a esteira de produtos é movimentada até que a caixa esteja cheia de produtos;
2. Quando cheia, a esteira de produtos é desativada, e a esteira de caixas é ativada.
3. Quando uma nova caixa é posicionada, a esteira dos produtos é ativada, e a esteira de caixas é desativada.

QUESTÃO 3

GRAVAÇÃO VCR

1. CÓDIGO DO PROGRAMA

```
#include <at89x52.h>
#define cameras P2
#define cam1 P2_0
#define cam2 P2_1
#define cam3 P2_2
#define cam4 P2_3

#define sensors P1
#define sensor1 P1_0
#define sensor2 P1_1
#define sensor3 P1_2
#define sensor4 P1_3

#define alarm P0_1

int count;
void alarme() interrupt 1{
    alarm =!alarm;
}

int main(){
    P0=0;
    sensors=0;
    cameras=0;
    EA=1;
    EX0=1;
    ET0=1;
    TMOD=1;

    while(1){
        TR0 = 1;
    }
}
```

```
void sensor() interrupt 0{
    TR0=0; //pausa timer para recarregar (Timer Stop)
    TH0=0x3C; //recarrega o timer com o valor inicial de
    TL0=0xAF; //15.535
    TR0=1; //ativa timer para contar (Timer RUN)
    count++;
    if((count >= 0) && (count <= 60)){
        cam4 = 0;
        if (sensor1 ==1){
            cam1 = 1;
        }
    }
    if ((count >= 60)&& (count <= 120)){
        cam1 = 0;
        if (sensor2 ==1){
            cam2 = 1;
        }
    }
    if ((count >= 120)&& (count <= 180)){
        cam2 = 0;
        if (sensor3 ==1){
            cam3 = 1;
        }
    }
    if ((count >= 240)&& (count <= 300)){
        cam3 = 0;
        if (sensor4 ==1){
            cam4 = 1;
        }
    }
    if (count >= 300){
        count = 0;
    }
}
```

3. EXECUÇÃO

Parallel Port 1 ×

Port 1

P1: 0x03 7 Bits 0 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☒

Pins: 0x03 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☒

Parallel Port 2 ×

Port 2

P2: 0x01 7 Bits 0 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☐

Pins: 0x01 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☐

Parallel Port 3 ×

Port 3

P3: 0xFB 7 Bits 0 ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒

Pins: 0xFB ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒

```
void insereMoeda() interrupt 0{
  if((tea == 1)&&(coffe == 0)){ //cha e selecionado
    glass = 1; //xicara reconhecida
    delay(100000);
    fill = 1; //enche a xicara
    delay(100000);
    fill = 0; //para de encher a xicara
    glass = 0; //retira a xicara
    tea = 0; //maquina volta ao estado inicial
  } else if ((tea == 0)&&(coffe == 1)){
    glass = 1; //xicara reconhecida
    delay(100000);
    fill = 1; //enche a xicara
    delay(100000);
    fill = 0; //para de encher a xicara
    glass = 0; //retira a xicara
    coffe = 0; //maquina volta ao estado inicial
  }
}
```

1. Quando a moeda é inserida (P3_2), você seleciona a bebida: café (P2_1) ou chá (P2_0);

2. A xícara é inserida (P1_0);

3. A máquina enche a xícara com a bebida (P1_1) selecionada durante um delay;

4. Em seguida ela zera as portas P1_0, P1_1 e P2, assim, finalizando a execução e aguarda uma outra moeda.

3. PORTAS UTILIZADAS

Parallel Port 0 ×

Port 0

P0: 0x00 7 Bits 0

Pins: 0x00

Parallel Port 1 ×

Port 1

P1: 0x00 7 Bits 0

Pins: 0x00

Parallel Port 2 ×

Port 2

P2: 0x00 7 Bits 0

Pins: 0x00

Parallel Port 3 ×

Port 3

P3: 0xFB 7 Bits 0

Pins: 0xFB

```
#define cameras P2
#define cam1 P2_0
#define cam2 P2_1
#define cam3 P2_2
#define cam4 P2_3

#define sensors P1
#define sensor1 P1_0
#define sensor2 P1_1
#define sensor3 P1_2
#define sensor4 P1_3

#define alarm P0_1
```


3. EXECUÇÃO

```
void sensor() interrupt 0{
  TR0=0; //pausa timer para recarregar (Timer Stop)
  TH0=0x3C; //recarrega o timer com o valor inicial de
  TL0=0xAF; //15.535
  TR0=1; //ativa timer para contar (Timer RUN)
  count++;
  if((count >= 0) && (count <= 60)){
    cam4 = 0;
    if (sensor1 ==1){
      cam1 = 1;
    }
  }
  if ((count >= 60)&& (count <= 120)){
    cam1 = 0;
    if (sensor2 ==1){
      cam2 = 1;
    }
  }
  if ((count >= 120)&& (count <= 180)){
    cam2 = 0;
    if (sensor3 ==1){
      cam3 = 1;
    }
  }
  if ((count >= 240)&& (count <= 300)){
    cam3 = 0;
    if (sensor4 ==1){
      cam4 = 1;
    }
  }
  if (count >= 300){
    count = 0;
  }
}
```

1. Quando os sensores de presença estão acionados, significa que há alguém na sala.
2. Se eles forem acionados, a câmera é acionada.

3. EXECUÇÃO

```
int count;  
void alarme() interrupt 1{  
    alarm =!alarm;  
}
```

1. Se houver uma interrupção externa (alguém entrar na sala sem autorização), então o alarme é acionado.