

```

#include<89c51rx2.inc>
CS EQU P2.6
CLK EQU P2.3
DIN EQU P2.4
DOUT EQU P2.5

CSEG AT 0H
JMP MAIN

//interrupcao externa 1
CSEG AT 013H
JMP P_Logicas

//interrupção externa 0
CSEG AT 03H
JMP EX0_CONTAR
CSEG AT 30H
//configurar a porta RS 232 com um baud Rate de 9600
RS_232:
    MOV SCON,#50H
    MOV T2CON,#34H
    MOV RCAP2H,#0FFH
    MOV RCAP2L,#0D9H
    RET

//configurar a interrupcao externa 1 VERIFICAR MAQUINA
CONFIG_EX1:
    SETB IT1
    SETB EX1
    SETB P3.3
    RET

//configurar a interrupção externa 0 usada para contar o Toff do timer
CONFIG_EX0:
    SETB IT0
    SETB EX0
    SETB P3.2
    RET

MAIN:
    SETB EA                ;activar as interrupções externas
;    SETB ES
    SETB P2.0             ;PINO DE ROUBO
    CALL RS_232            ;chama a função RS 232 para configurar a porta
    CALL CONFIG_EX0        ;chama a função CONFIG_EX1 para configurar a interrupção externa 0 para
                            ;ser usada para ler os valores de temperatura
    CALL CONFIG_EX1        ;chama a função CONFIG_EX1 para configurar a interrupção externa
1
    MOV 60H,#00            ;ID da maquina
    JMP $                  ;ciclo infinito

```

EXO_CONTAR:

MOV R0,#61H ;coloca R0 apontar a a posição 60H para guardar os valores dos
timers

MOV R2,#03H ;move para o R2 o valor 2

MOV TMOD,#01H ;configura o timer 0 no modo de 16 bits

MOV TL0,#00H ;colocar 0 no timer para começar a contagem do timer off

MOV TH0,#00H ;

SETB TR0 ;arranca a contagem do timer 0

CLR TF0 ;limpa a flag de transbordo do timer 0

JNB P3.2,\$;fica num ciclo infinito enquanto que o porto P3.3 estiver a 0

CLR TR0 ;para a contagem do timer 0

MOV @R0,TH0 ;guarda o valor do TH0 na posição apontada por R0

INC R0 ;incrementa R0

MOV @R0,TL0 ;guarda o valor do TL0 na posição apontada por R0

INC R0 ;incrementa R0

SETB CS ;ACTIV O PORTO DO CS

MOV A,#0FH ;MOVE PARA O ACUMULADOR O VALOR F QUE SERVIRA PARA

ACTIVAR O ADC

MOV R3,#4H

CLR CS ;ACTIVA OA ADC

ARRANQUE:

SETB CLK ;ACTIVAR O CLOCK

NOP

NOP

CLR CLK ;LIMPA O CLOCK PARA ENVIAR O PRIMEIRO BIT PARA O DIN

RRC A ;RODA O BIT MENOS SIGNIFICATIVO PARA O CARRY

MOV DIN,C ;ENVIA O BIT DO CARRY PARA O PORTO DIN

DJNZ R3,ARRANQUE ;DECREMENTA R3 E REPETE O CICLO

MOV R3,#09H

CLR C

CLR A

RECEBE:

SETB CLK ;ACTIVA O CLOCK

NOP

NOP

MOV C,DOUT ;COLOCA O BIT ENVIADO PELO ADC NO CARRY

CLR CLK ;LIMPA O CLOCK PARA RECEBER O PRIMEIRO BIT DO ADC

RRC A ;RODA O BIT DO CARRY PARA PARA O

ACUMULADOR

DJNZ R3,RECEBE ;DECREMENTA R3 E REPETE O CICLO PARA RECEBER NOVO

BIT

MOV 64H,A ;GUARDA O BYTE RECEBIDO NA POSIÇÃO DE

MEMORIA 64H

MOV R3,#04

//ESTA FUNÇÃO SERVE PARA RECEBER MAIS 4 BITS

RECEBE_2:

```
    SETB CLK
    NOP
    NOP
    CLR CLK
    MOV C,DOUT
    RRC A
    DJNZ R3,RECEBE_2
    SWAP A
    SETB CS
    MOV @R0,A
    MOV P1,#0
    MOV R3,#6H
    MOV 65H,#01H
    MOV R0,#60H
    JMP ENVIAR                ;salta para enviar
```

ENVIAR:

```
    MOV A,@R0                ;move o valor que esta a ser apontado por R0 para o acumulador
    INC R0                    ;decrementa R0
    MOV SBUF,A                ;move o valor do acumulador para o sbuf para ser enviado pela porta serie
    JNB TI,$                  ;fica num ciclo infinito ate que ti tenha enviado o que estiver em SBUF
    CLR TI                     ;limpa ti
    DJNZ R3,ENVIAR            ;decrementa R2 e salta para enviar para enviar o TH0
;                               MOV R2,#20H                ;move para R2 o valor 14H para gerar uma espera de 2
segundos
    CALL PAUSE                ;salta para a função pause
    CLR IE0
    RETI
```

P_Logicas:

```
    CALL PORTA_ABERTA1
    CALL FORCADA1
    CALL MOEDEIRO_CHEIO1
    CLR IE1
    RETI
```

PORTA_ABERTA1:

```
    JB P2.0,PORTA_ABERTA    ;SE O PORTO P2.0 ESTIVER A SALTA PARA PORTA ABERTA
    RET
```

FORCADA1:

```
    JB P2.1,FORCADA
    RET
```

MOEDEIRO_CHEIO1:

```
    JB P2.2,MOEDEIRO_CHEIO
    RET
```

PORTA_ABERTA:

```
    MOV A,#57H           ;move para o acumulador o valor P
    CLR TI
    MOV SBUF,A           ;COLOCA O VALOR DO ACUMULADOR NO SBUF PARA SER
EVIADO
    JNB TI,$             ;FICA NUM CICLO ATE TI FOR A 1 QUE SIGNIFICA QUE FOI ENVIADO

    RET
```

FORCADA:

```
    MOV A,#52H           ;move para o acumulador o valor R
    CLR TI
    MOV SBUF,A
    JNB TI,$
    RET
```

MOEDEIRO_CHEIO:

```
    MOV A,#42H           ;move para o acumulador o valor Z
    CLR TI
    MOV SBUF,A
    JNB TI,$
    RET
```

PAUSE:

```
    MOV TMOD,#01H        ;configura o timer 0 no modo de 16 bits
    MOV TH0,HIGH(-50000)
    MOV TL0,LOW(-50000)
    SETB TR0             ;arranca a contagem do timer 0
    CLR TF0              ;limpa TF0
    JNB TF0,$            ;fica num ciclo infinito enquanto que TF0
estiver a 0
    DJNZ R2,PAUSE         ;decrementa R2 e salta para pause para gerar uma
espera de 2s
    RET                  ;fim da interrupção externa
```

END