

## Guia da Placa de desenvolvimento PD Mega16 N1

Este tutorial consiste mostrar detalhes de hardware da placa de desenvolvimento PD Mega16 N1, necessário para iniciar a programação e testes. Boa aprendizagem ...

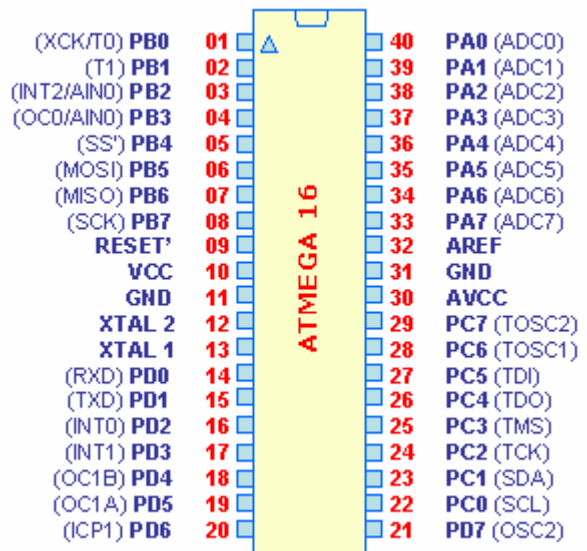
### O Atmega16

É um microcontrolador AVR de 8 bits com 16K bytes e programação flash In-System.

#### Principais características:

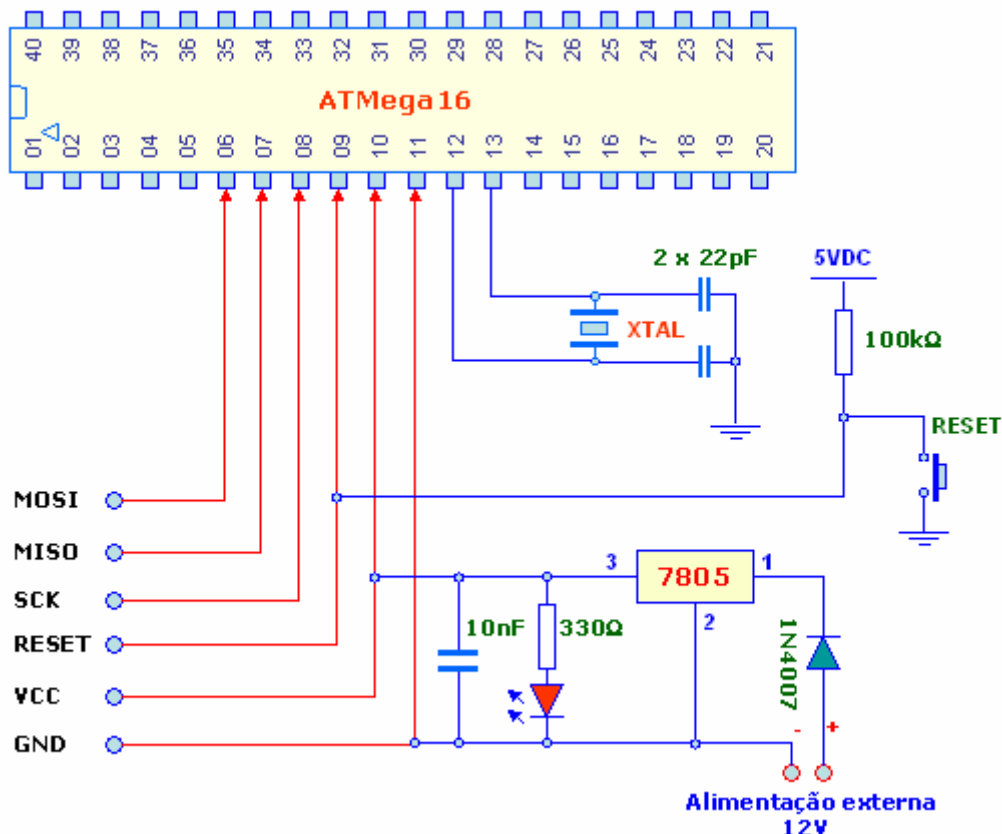
Microcontrolador de 40 pinos;  
Possui 32 portas de entrada e saída (I/O);  
Memória de programa Flash com 16Kb;  
Memória EEPROM com 512 bytes;  
Memória de Dados com 1024 bytes;  
Dois temporizadores de 8 bits;  
Um temporizador de 16 bits;  
Comunicação UART;  
8 canais AD de 10 bits;  
4 canais PWM  
etc...

#### Os pinos do Atmega16



#### A alimentação, o clock e o reset do atmega16

**Alimentação** – A alimentação do atmega16 é feita por uma fonte externa de 12V, nela está conectado um LED que indica se a fonte está energizada. O atmega16, assim como outros microcontroladores operam com tensão de 5V, daí a necessidade do regulador de tensão 7805, que regula a tensão em 5V, o capacitor tem como função eliminar possíveis ruídos. **(Um Led indicará se o circuito esta energizado)**



O diodo de uso geral 1N4007 evitará danos caso haja inversão de polaridade na alimentação.

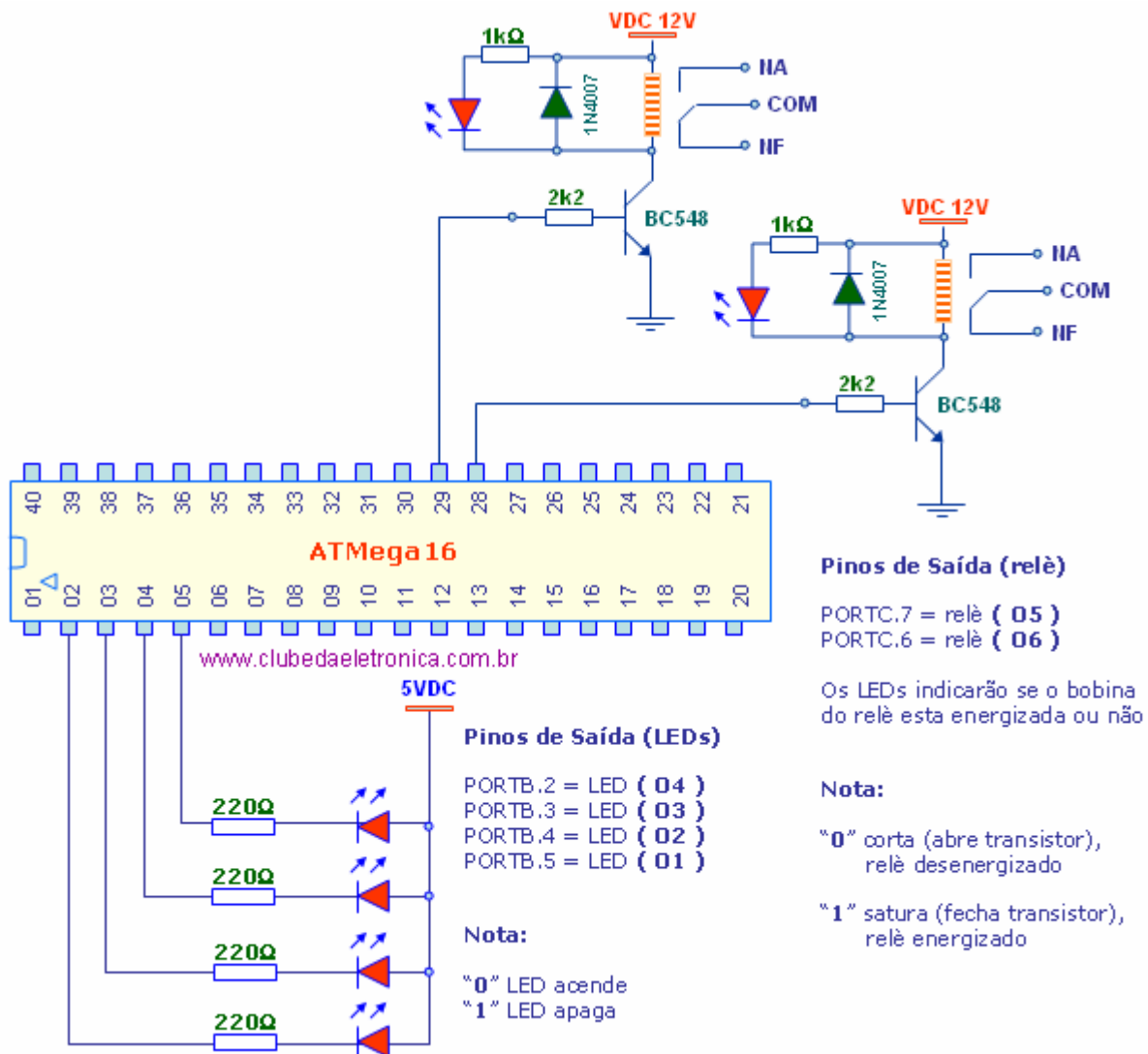
**O clock** – Estando conectado ao gravador e ao computador a placa **PD Mega16 N1** não necessita de clock externo, porém em muitas situações há necessidade de desconectá-la e mantê-la funcionando em campo, assim, o um cristal é utilizado para geração de clock, embora o atmega16 possua um clock interno, este não é utilizado neste material.

**Importante: O cristal deve estar o mais próximo possível dos pinos 12 e 13 do microcontrolador.**

**O RESET** – Como o próprio nome diz tem a função de resetar o microcontrolador. Como se vê a pino de reset do microcontrolador está conectado diretamente na alimentação de 5V, isto o mantém operando, uma vez que ele for conectado à terra o atmega 16 será resetado, um resistor de 100kΩ evita um curto circuito no momento que a chave RESET for pressionada.

### As saídas LEDs e Relês

**LEDs** – Os LEDs com seus catodos conectados aos PORTs do Atmega 16 e o anodo à fonte de alimentação (5V), isso implica que para que o led acenda deve-se enviar “0” e para apagá-lo “1”.



**RELÊ** – Para o acionamento de cargas maiores, foram adicionados à placa dois transistores NPN e dois relês. Note que, paralelo com a bobina do relê temos um led que indicará se a bobina esta energizada ou não.

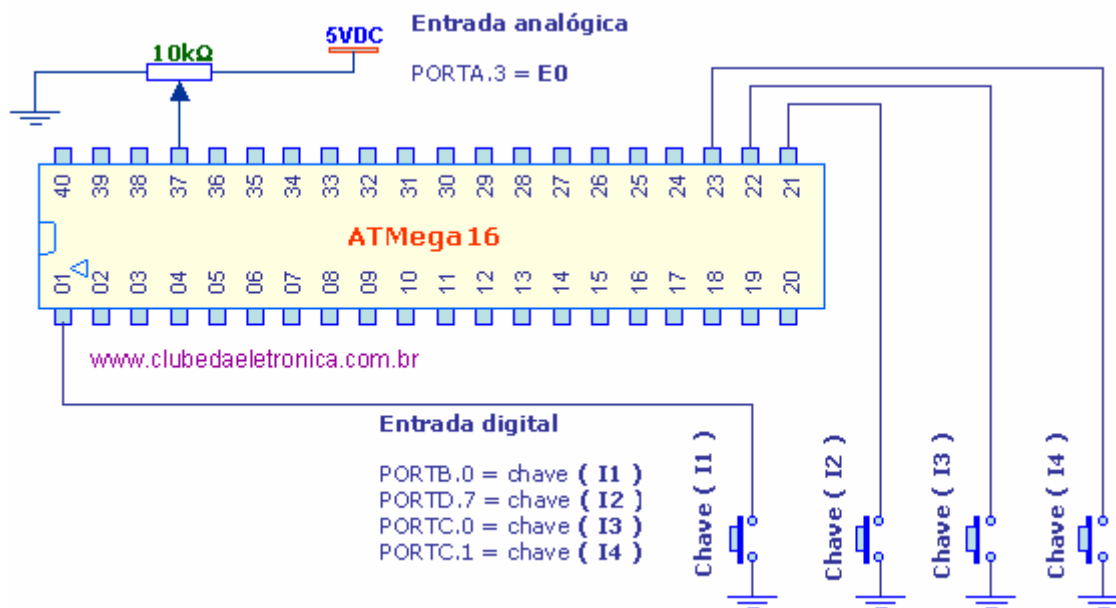
Ao atmega está conectado um transistor BC548 (NPN), ou seja, corta em “0” e satura em “1”.

Em corte, o transistor se comporta como chave aberta e, portanto, a bobina do relè está sem energia. Saturado, o transistor comporta-se como chave fechada e, portanto, a bobina do relè está energizada.

Deve ser adicionados borner para conexão de lâmpadas, motores etc...

### As entradas digitais e analógicas

**CHAVES** – As chaves são Push Button NA, ou seja, uma vez pressionadas elas fecham, não pressionadas elas abrem. Agora, observem que elas estão conectadas aos PORTs do atmega e ao terra, assim pressionando a chave ela envia “0” ao microcontrolador e não pressionadas não há nem “0” e nem “1”, há um estado de alta impedância que deverá ser previsto na programação.



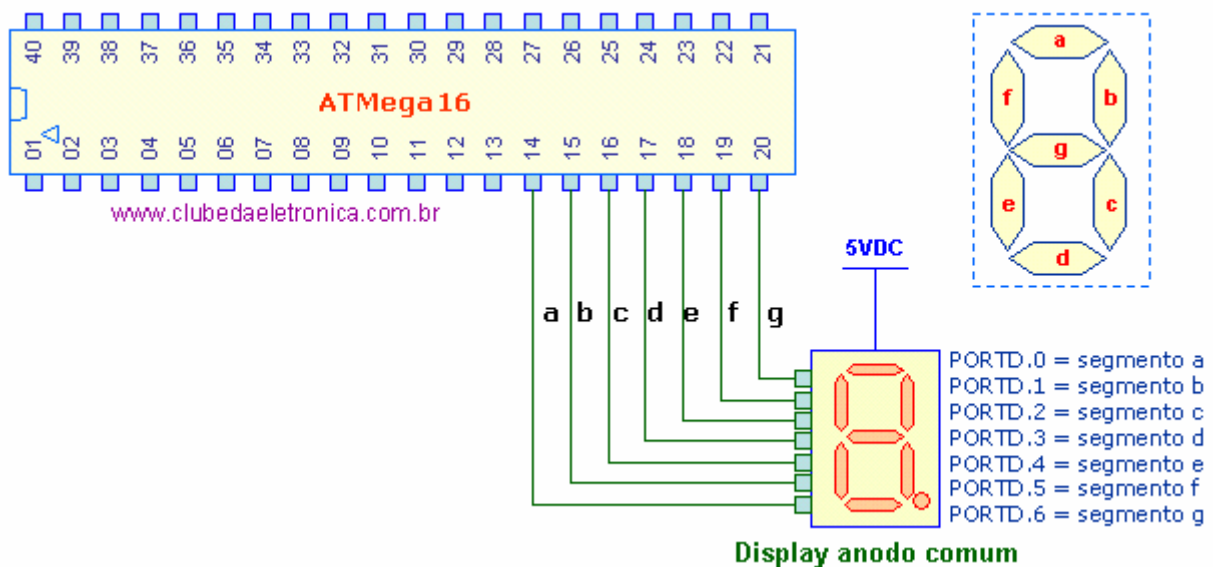
**Potenciômetro** – Os atmega 16 possui 8 conversores AD, nesta placa, disponibilizamos o PORT A.3 para receber o sinal analógico de 0 a 5V que será convertido em um sinal digital de 10 bits. Assim,

0V = 0000000000.

5V = 1111111111

## O display de 7 segmentos

O **display** – É do tipo anodo comum, ou seja, está conectado na alimentação de 5V e para acender o segmento deve-se enviar “0”.



## Comunicação serial

O Atmega 16 possui um canal de comunicação serial que não está disponível na placa PD mega16 N1, porém, retirando o jumper J1 os pinos TX e RX ficarão disponíveis bastando adicionar uma placa externa para comunicação, neste caso, o display de 7 segmentos deixa de funcionar.

### O módulo de comunicação serial (não disponível na placa PD mega16 N1)

## O padrão RS232

RS-232 é um padrão para troca serial dados entre um DTE (terminal de dados, de *Data Terminal equipment*) e um DCE (comunicador de dados, de *Data Communication equipment*).

### Níveis de tensão (padrão RS232)

- ❑ "0" tem um nível de tensão entre -15V e -5V
- ❑ "1" tem um nível entre +5 V e 15 V.

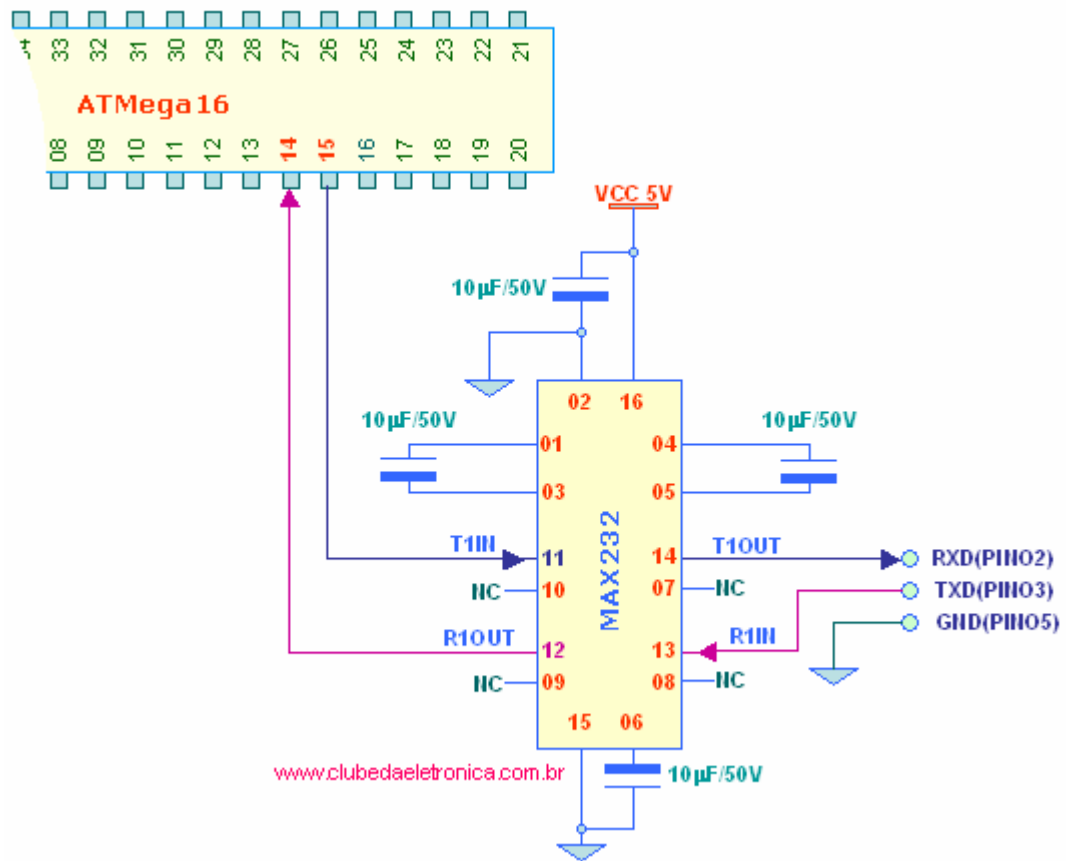
Definitivamente, isto não nos ajuda muito porque os microcontroladores AVR utilizam lógica TTL, ou seja:

- ❑ "0" tem um nível de tensão de 0 a 1,8V
- ❑ "1" tem um nível de tensão de 3,5 a 5V

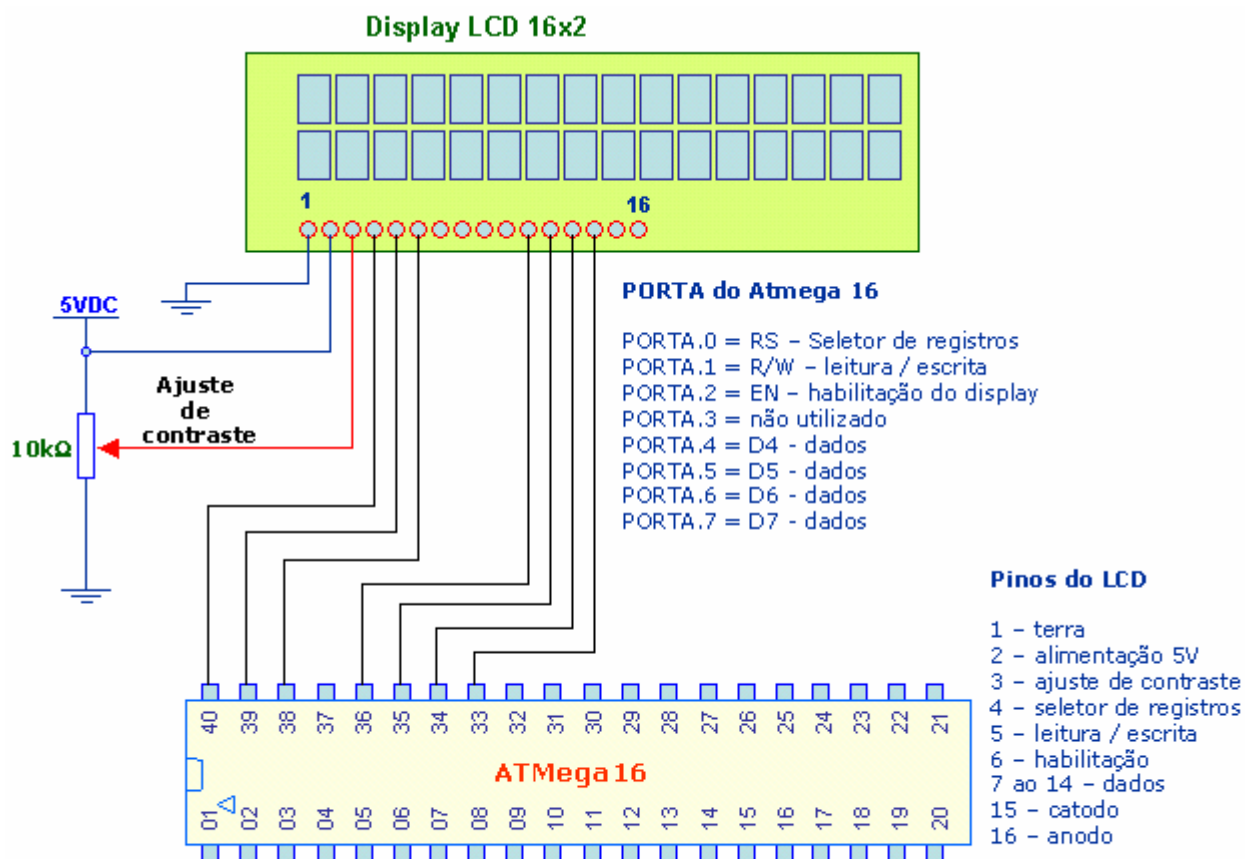
Então, para que possamos traçar informações entre PC e microcontrolador, devemos converter os sinais do padrão RS232 para TTL e vice versa.

### Interface para transmissão (MAX232)

É o circuito integrado responsável pela conversão RS232/TTL, ou seja, deve ser utilizado para fazer a conexão entre o microcontrolador e a porta serial do computador.



## O display de LCD (Pinagem compatível com Hitachi - HD44780)



### Notas:

- 1 - Os pinos 15 e 16 do LCD somente serão utilizados se no LCD houver back light (luz de fundo)
- 2 - A transmissão de dados para o LCD é feita 4 bits.

Procurando otimizar a placa não foi colocado o potenciômetro para ajuste de contraste e sim um divisor de tensão que tem o mesmo efeito.

Importante: Cada LCD possui suas próprias características, ou seja, são diferentes uns dos outros, portanto, devemos ajustar o contraste desejado com um potenciômetro, medir as resistências, conseguir resistores aproximados e aí sim solda-lo na placa.

## Configurando os pinos para placa PD mega16 N1

```
#include <mega16.h>           // chama biblioteca do microcontrolador utilizado.
#include <delay.h>           // chama biblioteca delay
                             // outras bibliotecas

void main (void)

{
  // *****INICIA CONFIGURAÇÕES DAS PORTAS*****

  // configurações dos LEDs como saída e inicia apagado

  DDRB.1=1;
  PORTB.1=1;

  DDRB.2=1;
  PORTB.2=1;

  DDRB.3=1;
  PORTB.3=1;

  DDRB.4=1;
  PORTB.4=1;

  // configurações das chaves como entrada e inicia não pressionada

  DDRB.0=0;
  PORTB.0=1;

  DDRC.0=0;
  PORTC.0=1;

  DDRC.1=0;
  PORTC.1=1;

  DDRD.7=0;
  PORTD.7=1;

  // configurações dos relés como saída e inicia em corte

  DDRC.7=1;
  PORTC.7=0;

  DDRC.6=1;
  PORTC.6=0;

  // configurações 7 segmentos como saída e inicia desligado

  DDRD.0=1;
  PORTD.0=1;

  DDRD.1=1;
  PORTD.1=1;

  DDRD.2=1;
  PORTD.2=1;

  DDRD.3=1;
  PORTD.3=1;
```

```
DDRD.4=1;
PORTD.4=1;
```

```
DDRD.5=1;
PORTD.5=1;
```

```
DDRD.6=1;
PORTD.6=1;
```

## // INICIA DEFINIÇÕES

// definindo portas de entrada

```
#define I1 PINB.0
#define I2 PIND.7
#define I3 PINC.0
#define I4 PINC.1
```

// definindo portas de saída

```
#define O4 PORTB.1 //LED
#define O3 PORTB.2
#define O2 PORTB.3
#define O1 PORTB.4
```

```
#define O5 PORTC.6 //reles
#define O6 PORTC.7
```

// saídas para display 7 seg (anodo comum)

```
#define a PORTD.0
#define b PORTD.1
#define c PORTD.2
#define d PORTD.3
#define e PORTD.4
#define f PORTD.5
#define g PORTD.6
```

## //FINAL DAS CONFIGURAÇÕES E DEFINIÇÕES

```
while(1)
```

```
{
```

```
    // coloque aqui seu programa
```

```
}
```

```
}
```