


Relatório Microcontroladores

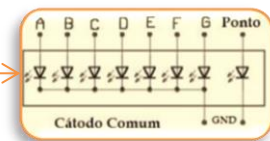
ETEPAM

Mecatrônica (Manhã) 02/12/2014

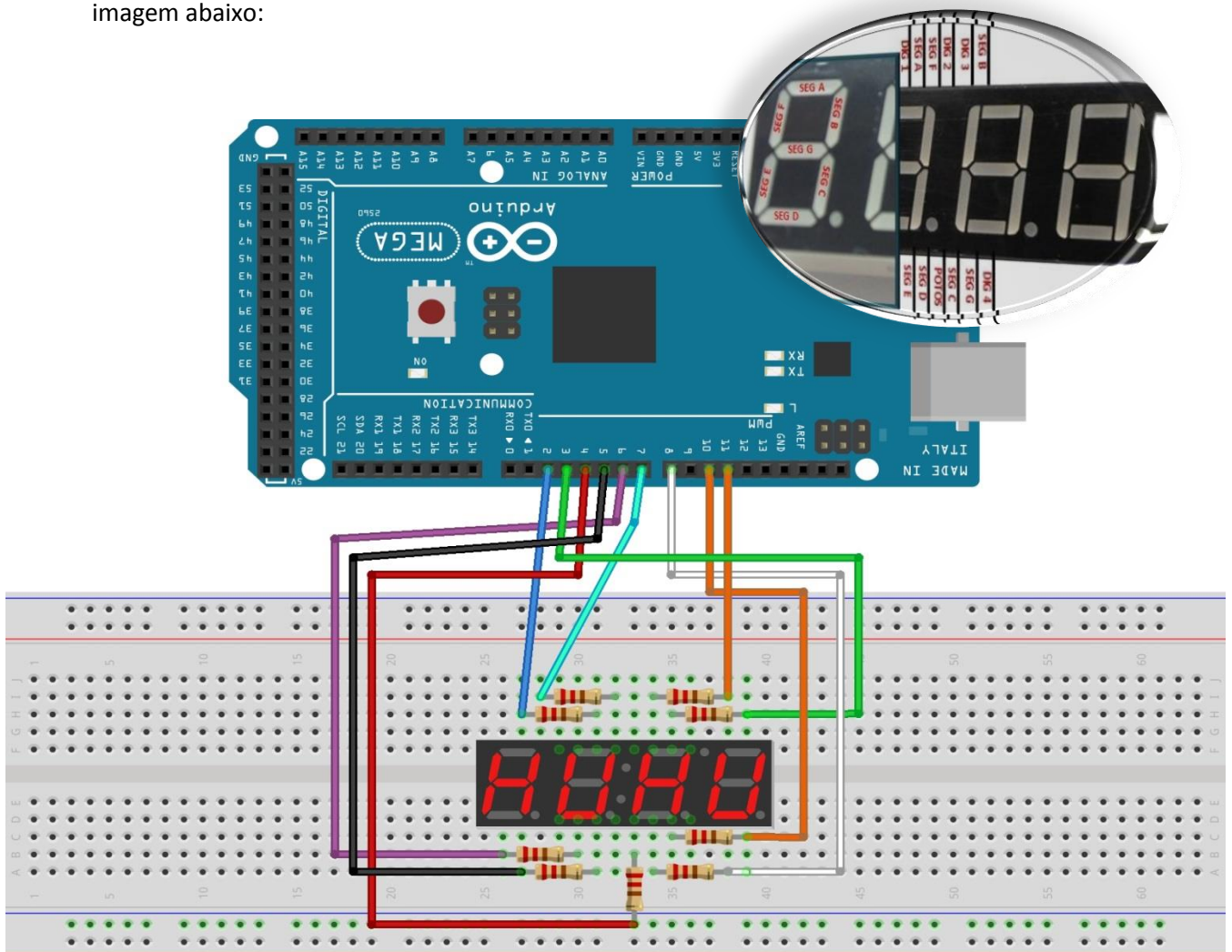
Danilo Martins de Brito Fialho

Foi feita a proposta de um projeto para criação de um contador decimal de repetição infinita de 0 a 99 com displays multiplexados. Foram utilizados os seguintes materiais na execução do projeto:

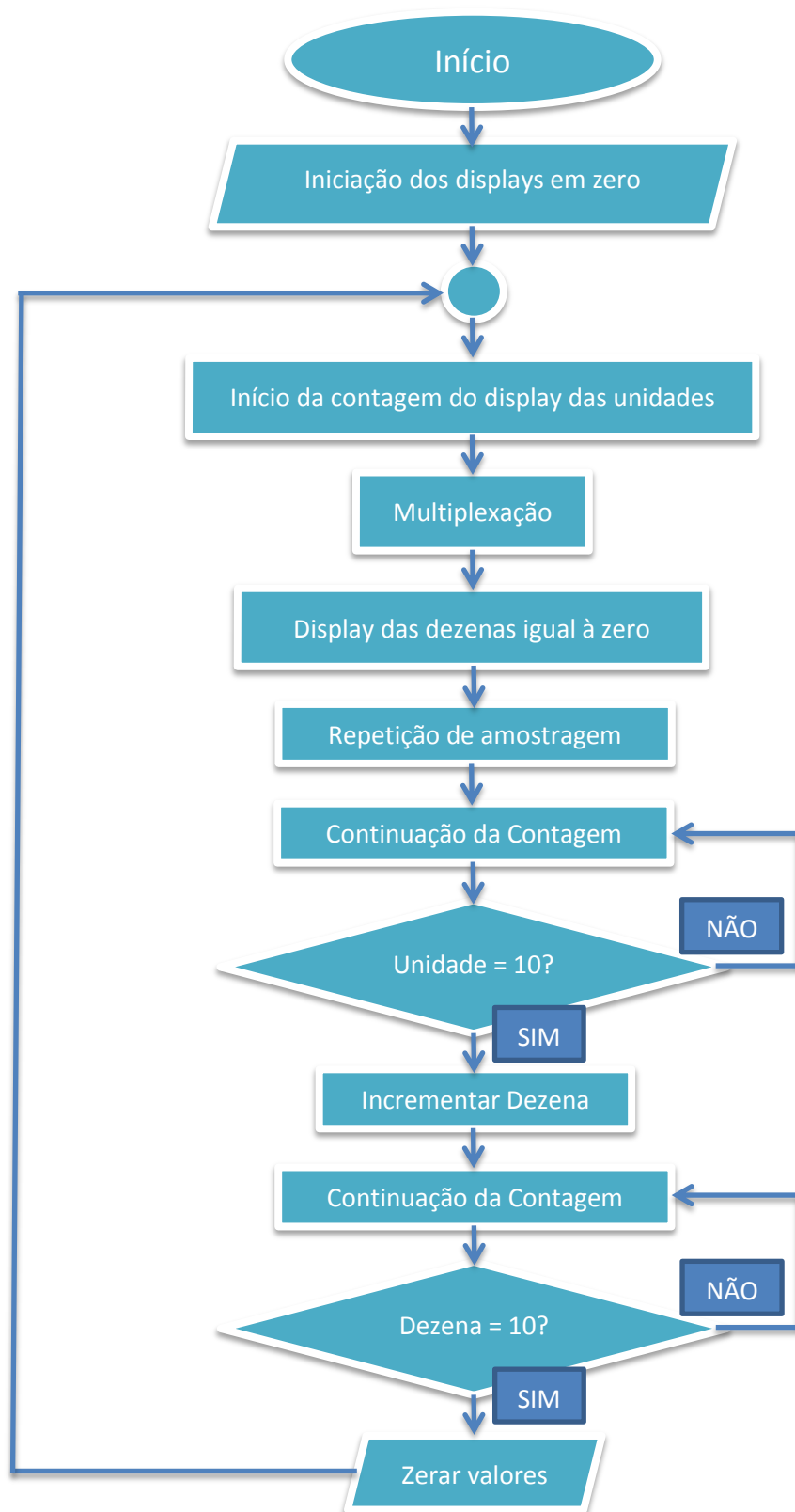
- 01 - Display 4 dígitos de 7 seguimentos (Cátodo Comum);
- 09 - Cabos de encaixe;
- 09 - Resistores 220Ω; 
- 01 - Arduino Mega 2560;
- 01 - Protoboard;



O esquema representando a devida localização de cada componente utilizado neste projeto foi feito pelo aplicativo Fritzing, para identificar e auxiliar no entendimento do projeto conforme a imagem abaixo:



A parte da programação do projeto foi completamente desenvolvida pela própria IDE do Arduino, e com intuito de criar uma programação simples o entendimento também será explanado conforme o esquema de fluxograma a seguir:



O fluxograma anterior foi desenvolvido para um melhor entendimento do programa, sendo necessário explicar o funcionamento da multiplexação executada. Este processo consiste em compartilhar os segmentos dos displays, onde um pino individualmente ativará um display de cada vez. Repetindo este processo com uma frequência mais alta, tem-se a impressão que ambos estão ligados simultaneamente.

O programa se inicia com a declaração das variáveis para os displays, sendo ele com variáveis normais (a,b,c...g) e em seguida são criadas duas variáveis: “un” que será responsável pela contagem das unidades (Dígito 4), e “dz” que é o valor do da contagem das dezenas (3 dígito).

Em seguida são criadas algumas funções para executar ações específicas, que são elas:

- num_display - Sequenciamento do barramento e amostragem do número no display;
- multiplex - Valores para ativar os pinos da multiplexação;
- numero - Determinação do barramento dos dígitos;

Logo em seguida é iniciada a função “setup”, que indica os pinos como saída, os valores dos displays em zero, e trava o display determinando nível lógico alto no seu ponto comum para que ele não seja ativado.

No “loop” é criada inicialmente uma estrutura de repetição “for”, que muda o valor da variável “un” de 0 a 9 (somando um a cada repetição). Dentro desta estrutura de repetição encontra-se outro “for”, que desta vez varia o valor de “t” de 0 a 50 somando um a cada repetição. Isso significa o segundo “for” vai repetir 50 vezes, e o “delay” entre a multiplexação é de 10ms, formando um “delay” de 1000ms $[(10+10)*50]$, ou 1s. No fim do segundo “for” será somado um a variável “dz”, e então o laço do primeiro for vai se romper iniciando novamente a função loop, mas com a variável “dz +1” (valor anterior+1).

Repetindo esse ciclo, quando “dz” for igual a 10, então todos os valores serão zerados e o “loop” se iniciará novamente, formando um “loop infinito”, assim a contagem só atinge o valor desejado (99).

Imagens da programação a seguir:

```
//DANILO MARTINS DE BRITO FIALHO;
//ETEPAM - MECATRONICA MANHÃ 2014;
//PROJETO DE UM CONTADOR (0 - 99) MULTIPLEXADO COM LOGICA SIMPLES;
//OBS.: PROGRAMAÇÃO PARA DISPLAY 4 DÍGITOS CATODO COMUM (GND COMUM);
```

```
Contagem_0_99_Simples_Multiplexado | Arduino 1.0.5
File Edit Sketch Tools Help

Contagem_0_99_Simples_Multiplexado

//PINAGEM DOS DISPLAYS, E DA MULTIPLEXAÇÃO;
int a=2,b=3,c=4,d=5,e=6,f=7,g=8,d1=10,d2=11;
//TEMPO EM MILLISEGUNDOS; VALOR NUMÉRICO DAS UNIDADES E DEZENAS DOS
int t=0,un=0,dz=0;

//FUNÇÃO PARA MOSTRAR O NÚMERO NO DISPLAY;
void num_display(int va, int vb, int vc, int vd, int ve, int vf, int vg) {
  digitalWrite(a,va);digitalWrite(b,vb);digitalWrite(c,vc);digitalWrite(d,vd);digitalWrite(e,ve);digitalWrite(f,vf);digitalWrite(g,vg);
}

//DEFINIÇÃO DOS PINOS DE MULTIPLEXAÇÃO;
void multiplex(int vd1, int vd2){
  digitalWrite(d1, vd1); digitalWrite (d2, vd2);
}
```

```
void setup(){
  //DEFINIÇÃO DAS PORTAS DO DISPLAY COMO SAÍDAS;
  pinMode(a, OUTPUT); pinMode(b, OUTPUT); pinMode(c, OUTPUT); pinMode(d, OUTPUT);
  pinMode(e, OUTPUT); pinMode(f, OUTPUT); pinMode(g, OUTPUT);
  //DEFINIÇÃO DAS PORTAS DA MULTIPLEXAÇÃO COMO SAÍDAS;
  pinMode(d1, OUTPUT);pinMode(d2, OUTPUT);

  //INICIANDO OS DISPLAYS EM ZERO;MULTIPLEXAÇÃO PARA "TRAVAR" OS DISPLAYS
  numero(0); multiplex(1,1);
}
```

```
//FUNÇÃO PARA DEFINIR O NÚMERO NO DISPLAY;
void numero(int num){
  if (num == 0){num_display(1,1,1,1,1,1,0);}
  if (num == 1){num_display(0,1,1,0,0,0,0);}
  if (num == 2){num_display(1,1,0,1,1,0,1);}
  if (num == 3){num_display(1,1,1,1,0,0,1);}
  if (num == 4){num_display(0,1,1,0,0,1,1);}
  if (num == 5){num_display(1,0,1,1,0,1,1);}
  if (num == 6){num_display(1,0,1,1,1,1,1);}
  if (num == 7){num_display(1,1,1,0,0,0,0);}
  if (num == 8){num_display(1,1,1,1,1,1,1);}
  if (num == 9){num_display(1,1,1,1,0,1,1);}
}
```

```
void loop(){
  //ESTRUTURA DE REPETIÇÃO PARA EXECUTAR A CONTAGEM NOS DISPLAYS;
  for (un = 0; un < 10; un++){
    for (t=0; t<50; t++){
      numero(un);
      multiplex(0,1);delay(10);
      numero(dz);
      multiplex(1,0);delay(10);
    }
  }
  //ADICIONAR UM A VARIÁVEL DZ (DEZENA);
  dz++;
  //QUANDO A VARIÁVEL DZ FOR IGUAL A 10 ENTÃO OS VALORES SÃO ZERADOS
  if (dz == 10){dz=0;un=0;multiplex(1,1);}
}
```