****

**Projeto de Engenharia**

Contador Crescente e Decrescente utilizando Arduino

Felipe Castro C. do Sacramento

Flavio Rafael da Silva Travassos

Otávio Augusto Alves Silva

Priscila Sayuri T. Lameira

**Contador Crescente e Decrescente**

**1.Requisitos**

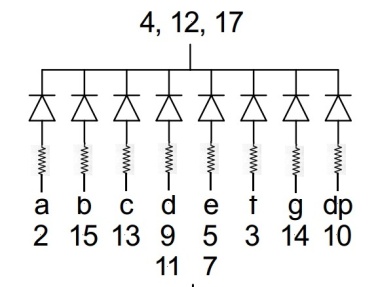
* Deve possuir um display de 7 segmentos.
* Contando crescente ou decrescentemente a partir do momento que for apertado o botão.

**2. Descrição de funcionamento do Hardware e Material utilizado**

* 1 Arduino UNO
* 1 Protoboard
* 1 Cabo USB
* 21 Jumper's
* Push Button
* 9 Resistores de 330
* 1 Display de LED de 7 segmentos (CTK D368K) - Catodo

Utilizaremos todos os hardwares padrões como nos projetos anteriores, o arduino, a protoboard, jumpers, push button e resistor, a única diferença é o display de sete segmentos.

O display de sete segmentos, é um tipo de [*display*](http://pt.wikipedia.org/wiki/Display) (mostrador) barato usado como alternativa a [displays de matriz de pontos](http://pt.wikipedia.org/w/index.php?title=Display_de_matriz_de_pontos&action=edit&redlink=1) mais complexos e dispendiosos,  displays de sete segmentos são comumente usados em [eletrônica](http://pt.wikipedia.org/wiki/Eletr%C3%B4nica) como forma de exibir uma informação alfanumérica ([binário](http://pt.wikipedia.org/wiki/Bin%C3%A1ria), [octadecimal](http://pt.wikipedia.org/wiki/Octadecimal" \o "Octadecimal), [decimal](http://pt.wikipedia.org/wiki/Decimal) ou [hexadecimal](http://pt.wikipedia.org/wiki/Hexadecimal)).



**Circuito Elétrico**

**3. Descrição do funcionamento do Software**

// Criação das variáveis e matriz identificando cada numero ou letra

int ponto=9;

int botao=13;

int st=0;

int pins[16][8]= {

{ 1,1,1,1,1,1,0,1 }, // = 0

{ 0,1,1,0,0,0,0,1 }, // = 1

{ 1,1,0,1,1,0,1,1 }, // = 2

{ 1,1,1,1,0,0,1,1 }, // = 3

{ 0,1,1,0,0,1,1,1 }, // = 4

{ 1,0,1,1,0,1,1,1 }, // = 5

{ 1,0,1,1,1,1,1,1 }, // = 6

{ 1,1,1,0,0,0,0,1 }, // = 7

{ 1,1,1,1,1,1,1,1 }, // = 8

{ 1,1,1,0,0,1,1,1 }, // = 9

{ 1,1,1,0,1,1,1,0 }, // = A

{ 0,0,1,1,1,1,1,0 }, // = B

{ 1,0,0,1,1,1,0,0 }, // = C

{ 0,1,1,1,1,0,1,0 }, // = D

{ 1,0,0,1,1,1,1,0 }, // = E

{ 1,0,0,0,1,1,1,0 }, // = F

};

// Configuração das portas digitais

void setup()

{

pinMode(2, OUTPUT);

pinMode(3, OUTPUT);

pinMode(4, OUTPUT);

pinMode(5, OUTPUT);

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

pinMode(ponto, OUTPUT);

pinMode(botao, INPUT);

}

void numeros(int num)

{

int pin=2;

for (int ct=0; ct<7; ct++)

{

digitalWrite(pin, pins[num][ct]);

pin++;

}

}

// Função de repetição Loop

void loop()

{

contador();

}

// Função de contagem da matriz

void contador()

{

crescente(st);

}

// Função crescente de 0 a F

int crescente(int cresc)

{

for(byte count = cresc; count < 16; count++)

{

numeros(count);

delay(1000);

int state = digitalRead(botao);

if(state == HIGH)

{

st=count;

break;

}

if(count==15)

{

count=-1;

}

}

decrescente(st);

}

// Função decrescente de F a 0

int decrescente(int decresc)

{

for(byte count = decresc; count > 0; count--)

{

numeros(count - 1);

delay(1000);

int state = digitalRead(botao);

if(state == HIGH)

{

st = count;

break;

}

if(count==1)

{

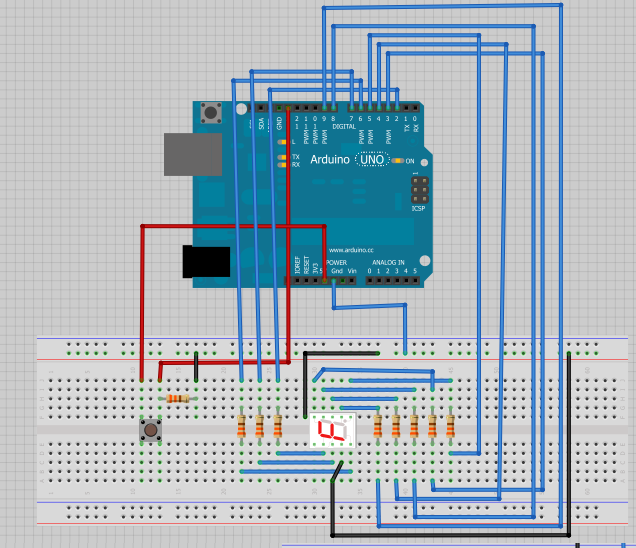
count=17;

}

}

crescente(st);

}

**Esquema final do PROJETO -**  **Contador Crescente e Decrescente**

**4. Memória de Calculo**

Formula para calcular o resistor necessário:

Onde:

R = Resistor

Vs = Voltagem Fornecida

Vl = Voltagem do LED

I = Corrente do LED

Dados fornecidos:

Vs = 5v

Vl = 2v

I = 15 mA

Logo :

**

*Ω*

Sabendo a resistência necessária, utilizamos sempre uma resistência com valor mais próximo maior do que o valor necessário. Utilizamos resistores de 330 *Ω.*