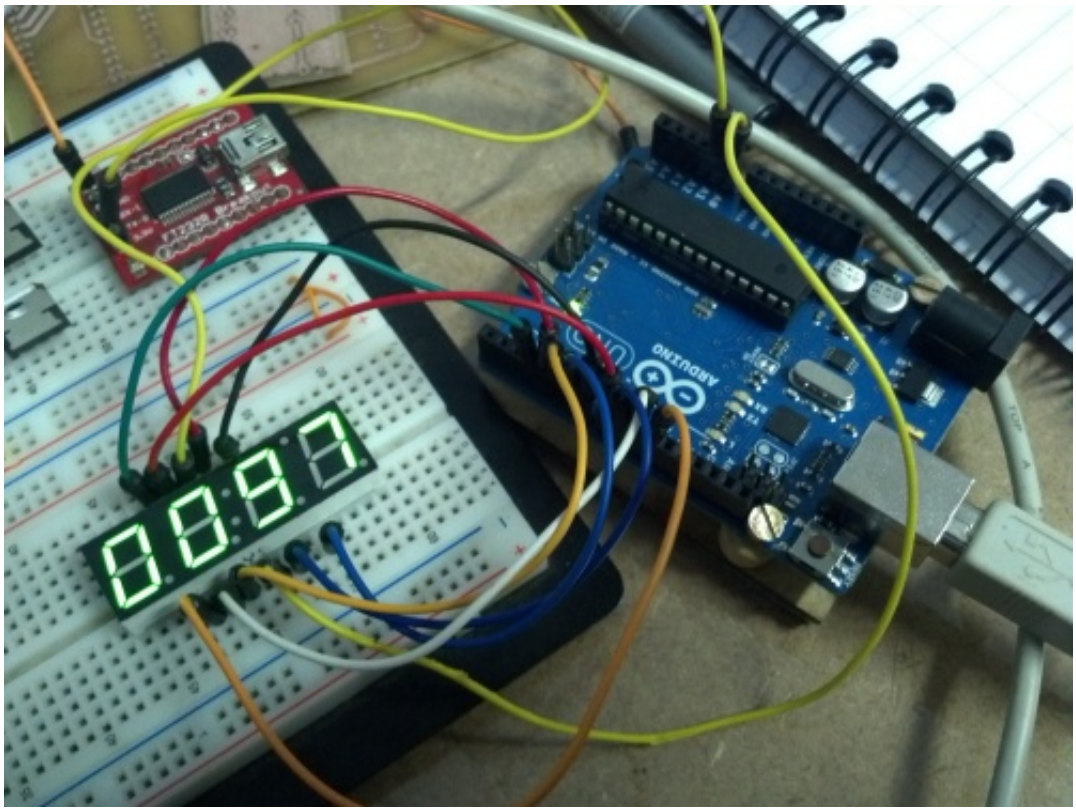
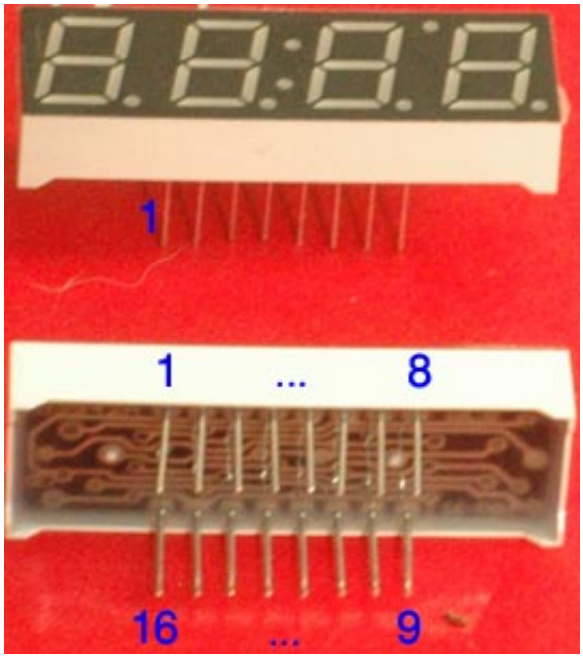
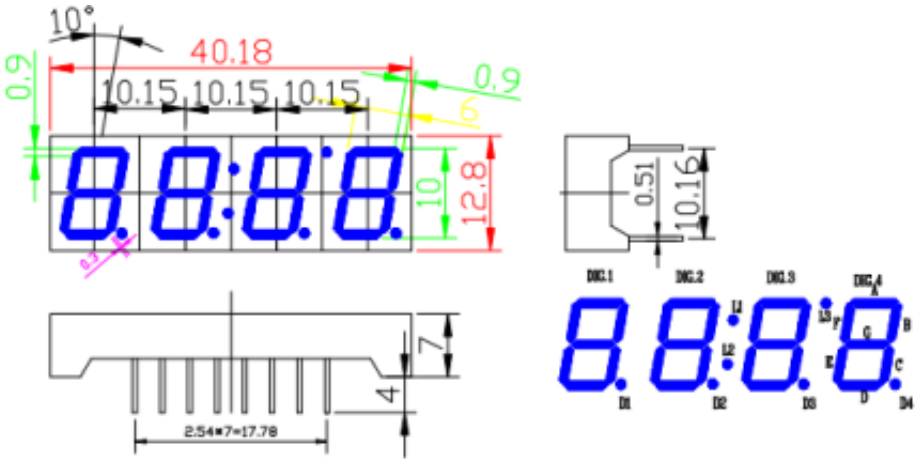
Projeto: Utilizando o display de 7 segmentos de 4 dígitos com Arduino para saber caracteres pela tabela ASCII

- Postado por Laboratório de Garagem em 6 junho 2012 às 10:41
- Exibir blog

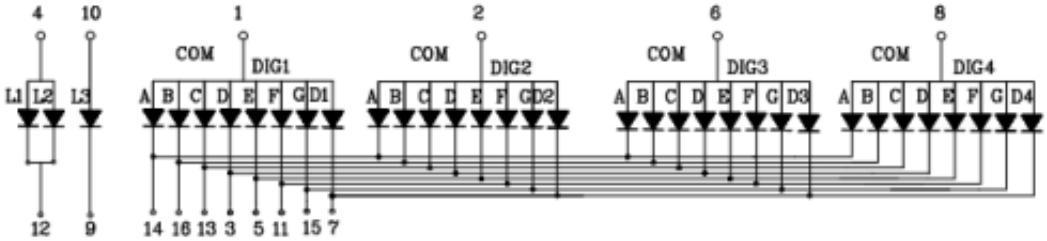


Neste projeto faremos com que o Arduino leia caractere do teclado e mostrar o número correspondente da tabela ASCII pelo display de 4 dígitos! O display de 7 segmentos de 4 dígitos contém 16 pinos com anodo comum. Abaixo está sua pinagem: (Caso queira ver o datasheet, clique aqui)



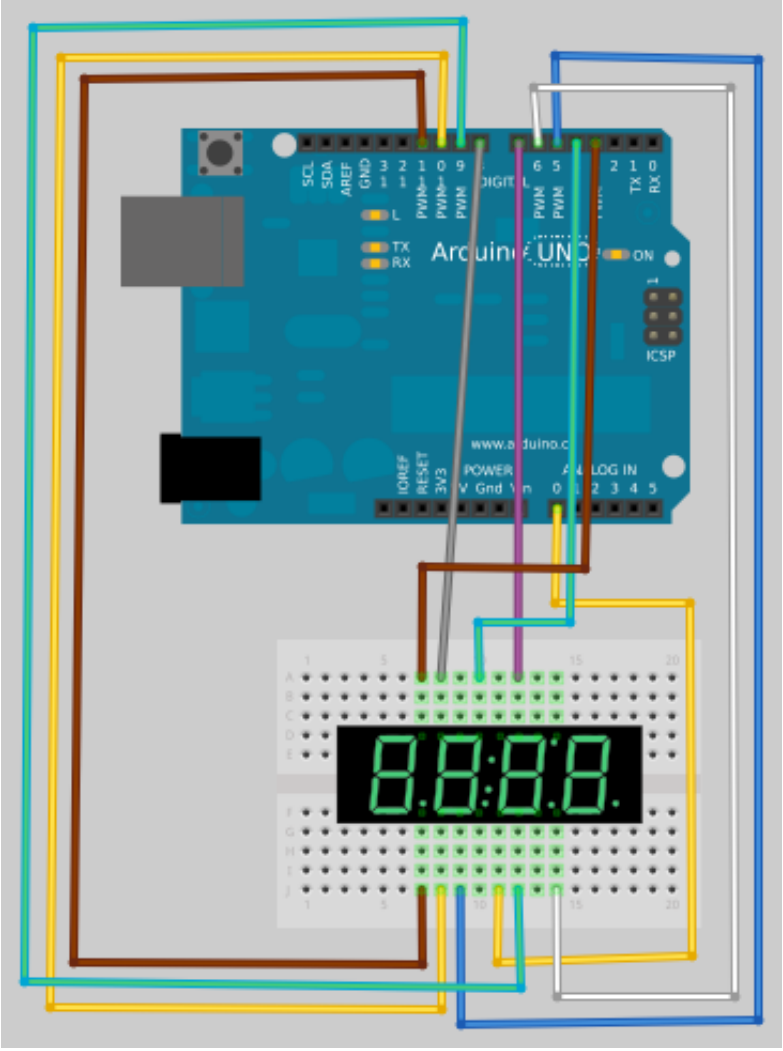


4. 电路图（Circuit Diagram）：



Podemos ver que os anodos são: 1, 2, 6, 8, 4 e 10. Em cada dígito tem seu ponto e os pinos 4 e 10 são os outros pontos que podemos utilizar também.

Agora, vamos para as ligações:



Feitas as ligações, abra a IDE do Arduino e passe a seguinte programação:

```
int digit1 = 11; //PWM Display pin 1
int digit2 = 10; //PWM Display pin 2
int digit3 = 9; //PWM Display pin 6
int digit4 = 6; //PWM Display pin 8


int segA = A1; //Display pin 14
int segB = 3; //Display pin 16
int segC = 4; //Display pin 13
int segD = 5; //Display pin 3
int segE = A0; //Display pin 5
int segF = 7; //Display pin 11
int segG = 8; //Display pin 15
int num=0;

void setup() {
pinMode(segA, OUTPUT);
pinMode(segB, OUTPUT);
pinMode(segC, OUTPUT);
pinMode(segD, OUTPUT);
pinMode(segE, OUTPUT);
pinMode(segF, OUTPUT);
pinMode(segG, OUTPUT);
```

```
pinMode(digit1, OUTPUT);
pinMode(digit2, OUTPUT);
pinMode(digit3, OUTPUT);
pinMode(digit4, OUTPUT);
Serial.begin(9600);
pinMode(13, OUTPUT);
}

void loop()
{

if(Serial.available()>0)
{
num = Serial.read();
}
dispNumber(num);
}

void dispNumber(byte toDisplay) {
#define DISPLAY_BRIGHTNESS 500

#define DIGIT_ON HIGH
#define DIGIT_OFF LOW

long beginTime = millis();

for(int digit = 4 ; digit > 0 ; digit--) {

//Turn on a digit for a short amount of time
switch(digit) {
case 1:
digitalWrite(digit1, DIGIT_ON);
break;
case 2:
digitalWrite(digit2, DIGIT_ON);
break;
case 3:
digitalWrite(digit3, DIGIT_ON);
break;
case 4:
digitalWrite(digit4, DIGIT_ON);
break;
}

//Turn on the right segments for this digit
ligNumber(toDisplay % 10);
toDisplay /= 10;

delayMicroseconds(DISPLAY_BRIGHTNESS); //Display this digit for a fraction of a second

//Turn off all segments
ligNumber(10);

//Turn off all digits
digitalWrite(digit1, DIGIT_OFF);
```

```
digitalWrite(digit2, DIGIT_OFF);
digitalWrite(digit3, DIGIT_OFF);
digitalWrite(digit4, DIGIT_OFF);
}

while( (millis() - beginTime) < 10) ; //Wait for 20ms to pass before we paint the display again
}

void ligNumber(int numToDisplay) {

#define SEGMENT_ON LOW
#define SEGMENT_OFF HIGH

switch (numToDisplay){

case 0:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_ON);
digitalWrite(segE, SEGMENT_ON);
digitalWrite(segF, SEGMENT_ON);
digitalWrite(segG, SEGMENT_OFF);
break;

case 1:
digitalWrite(segA, SEGMENT_OFF);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_OFF);
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_OFF);
digitalWrite(segG, SEGMENT_OFF);
break;

case 2:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_OFF);
digitalWrite(segD, SEGMENT_ON);
digitalWrite(segE, SEGMENT_ON);
digitalWrite(segF, SEGMENT_OFF);
digitalWrite(segG, SEGMENT_ON);
break;

case 3:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_ON);
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_OFF);
digitalWrite(segG, SEGMENT_ON);
break;
```

```
case 4:
digitalWrite(segA, SEGMENT_OFF);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_OFF);
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_ON);
digitalWrite(segG, SEGMENT_ON);
break;

case 5:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_OFF);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_ON);
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_ON);
digitalWrite(segG, SEGMENT_ON);
break;

case 6:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_OFF);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_ON);
digitalWrite(segE, SEGMENT_ON);
digitalWrite(segF, SEGMENT_ON);
digitalWrite(segG, SEGMENT_ON);
break;

case 7:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_OFF);
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_OFF);
digitalWrite(segG, SEGMENT_OFF);
break;

case 8:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_ON);
digitalWrite(segE, SEGMENT_ON);
digitalWrite(segF, SEGMENT_ON);
digitalWrite(segG, SEGMENT_ON);
break;

case 9:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_ON);
```

```
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_ON);
digitalWrite(segG, SEGMENT_ON);
break;

case 10:
digitalWrite(segA, SEGMENT_OFF);
digitalWrite(segB, SEGMENT_OFF);
digitalWrite(segC, SEGMENT_OFF);
digitalWrite(segD, SEGMENT_OFF);
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_OFF);
digitalWrite(segG, SEGMENT_OFF);
break;
}
}
```

Configure a IDE do Arduino de acordo com sua placa (UNO, Duemilanove, etc) e a porta referente a porta USB que o Arduino está conectado. Faça o UPLOAD e abra o Serial Monitor. Configure o Serial Monitor para 9600 baud e "No ending line". Digite um caractere do seu teclado e aperte ENTER. No display irá aparecer o valor em ASCII correspondente ao caractere que você digitou!

É isso!! Esperamos que tenham gostado! Caso tenham dúvidas, poste sua dúvida aqui mesmo neste blog! Caso tenha sugestões de tutoriais, poste aqui! Caso queira ver outros tutoriais e projetos desenvolvidos pela equipe LdG e pelos garagistas da rede, clique aqui e aqui respectivamente.

Referências:

http://www.labdegaragem.org/loja/index.php/display-de-4-digitos-ver...

http://www.sparkfun.com/products/10931

http://www.pavius.net/2010/02/rotary-encoder-based-cooking-timer/

http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Components/LED/ATA34...

Entrar em Laboratorio de Garagem

Bem-vindo a
Laboratorio de Garagem

Registre-se
ou acesse

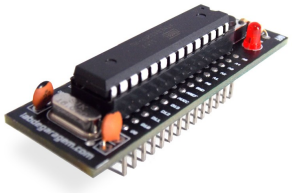
Or sign in with:

-
-
-
-

Publicidade

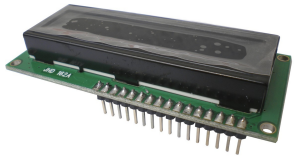
Convide um amigo para o Lab!

Loja Lab de Garagem



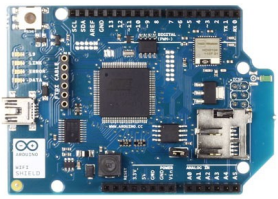
Garagino Proto

7x R$5,10 ou R$33,00 à vista



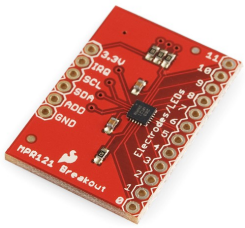
LCD 16x2

6x R$5,18 ou R$29,00 à vista



Arduino WiFi Shield

18x de R$25,92 ou R$389,00 à vista



Breakout Sensor de Toque Capacitativo MPR121

7x R$5,56 ou R$36,00 à vista