

Projeto 10:

RGB serial controlled LEDs.

(Beginning Arduino

Copyright © 2010 by Michael McRoberts)

Aula usa referencia acima.

Para o projeto 10 usamos o hardware da figura (projeto 8). Usando o “Serial Monitor” na IDE, enviaremos comandos para o ARDUINO. Desse modo controlamos qual LED e com que intensidade deverão acender.

Vamos manipular textos (string characters).

// Project 10 - Serial controlled LEDs

```
char buffer[18];
```

```
int red, green, blue;
```

```
int RedPin = 11;
```

```
int GreenPin = 10;
```

```
int BluePin = 9;
```

```
void setup()
```

```
{
```

```
Serial.begin(9600);
```

```
// determina o numero de pulsos/s comunic.
```

```
Serial.flush();
```

```
pinMode(RedPin, OUTPUT);
```

```
pinMode(GreenPin, OUTPUT);
```

```
pinMode(BluePin, OUTPUT);
```

```
}
```

```
void loop() {
```

```
  if (Serial.available() > 0) {
```

```
    int index=0;
```

```
    delay(100); // let the buffer fill up
```

```
    int numChar = Serial.available();
```

```
    if (numChar>15) {
```

```
      numChar=15;
```

```
    }
```

```
    while (numChar-->0) {
```

```
      buffer[index++] = Serial.read();
```

```
    }
```

```
    splitString(buffer);
```

```
  }
```

```
}
```

```
void splitString(char* data) {
```

```
  Serial.print("Data entered: ");
```

```
  Serial.println(data);
```

```
  char* parameter;
```

```
  parameter = strtok (data, ",");
```

```
  while (parameter != NULL) {
```

```
    setLED(parameter);
```

```
    parameter = strtok (NULL, ",");
```

```
  }
```

```
  // Clear the text and serial buffers
```

```
  for (int x=0; x<16; x++) {
```

```
    buffer[x]='\0';
```

```
  }
```

```
Serial.flush();
```

```
}
```

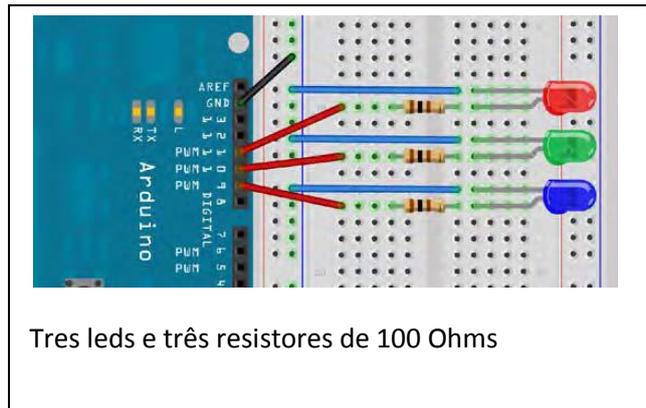
```
void setLED(char* data) {
```

```
  if ((data[0] == 'r') || (data[0] == 'R')) {
```

```
    int Ans = strtol(data+1, NULL, 10);
```

```
    Ans = constrain(Ans,0,255);
```

```
    analogWrite(RedPin, Ans);
```



Tres leds e três resistores de 100 Ohms


```

void splitString(char* data) {
  Serial.print("Data entered: ");
  Serial.println(data);
  char* parameter;
  parameter = strtok (data, " ,");
  while (parameter != NULL) {
    setLED(parameter);
    parameter = strtok (NULL, " ,");
  }
  // Clear the text and serial buffers
  for (int x=0; x<16; x++) {
    buffer[x]='\0';
  }
  Serial.flush();
}

```

Um apontador é caracterizado por um * antes da variavel apontadora: *data. É uma variável que aponta para outra variavel. **splitString(buffer)**; passamos à função o conteúdo apontado por buffer: a matriz de caracteres.

Em seguida:

```
Serial.print("Data entered: ");
```

Escreve Data entered:

```
Serial.println(data);
```

Pula uma linha e imprime a matriz que escrevemos.

```
R255 G127 B56
```

```
Char* parameter;
```

```
parameter = strtok (data, " ,");
```

Divide a linha até o primeiro branco ou vírgula.

Em nosso caso

```
R255
```

Agora esse parameter é enviado à função **setLED(parameter)**;

enquanto

```
while (parameter != NULL) {
```

O comando **parameter = strtok (NULL, " ,");** informa para continuar a partir de onde parou.

```
char* parameter;
```

```
parameter = strtok (data, " ,");
```

```
while (parameter != NULL) {
```

```
  setLED(parameter);
```

```
  parameter = strtok (NULL, " ,");
```

```
}
```

```
// Clear the text and serial buffers
```

```
for (int x=0; x<16; x++) {
```

```
  buffer[x]='\0';
```

```
}
```

```
Serial.flush();
```

A função **setLED()** pega cada parametro da linha de texto, separa a cor e a intensidade e executa.

```
void setLED(char* data) {
```

```
if ((data[0] == 'r') || (data[0] == 'R')) {
```

```
int Ans = strtol(data+1, NULL, 10);
```

```
Ans = constrain(Ans,0,255);
```

```
analogWrite(RedPin, Ans);
```

```
Serial.print("Red is set to: ");
```

```
Serial.println(Ans);
```

