



FIAP

Engenharia de Software

EDGE COMPUTING & COMPUTER SYSTEMS

04 – Interactive Traffic, Chase Effect, Interactive Chase Effect



Prof. Airton Y. C. Toyofuku



profairton.toyofuku@fiap.com.br

Tipos de variáveis

Tipo	Tamanho na RAM	Intervalo
Void	Não ocupa espaço	Não aplicável
Bool	1 byte	0 ou 1 (False ou True)
Char	1 byte	-128 a 127
Unsigned Char	1 byte	0 a 255
Int	2 bytes	-32.768 a 32.767
Unsigned Int	2 bytes	0 a 65.535
Long	4 bytes	-2.147.483.648 a 2.147.483.647
Unsigned Long	4 bytes	0 a 4.294.967.295
Float	4 bytes	$3,4 \times 10^{-38}$ a $3,4 \times 10^{38}$
Double	8 bytes	$1,7 \times 10^{-308}$ a $1,7 \times 10^{308}$
String	1 byte + n	Sequência de caracteres
Array	1 byte + n	Coleção de variáveis

Tipos de variáveis

```
#include <stdio.h>
#include <stdbool.h>

int main() {

    printf("tamanho do bool:    %ld \n", sizeof(bool));
    printf("tamanho do char:    %ld \n", sizeof(char));
    printf("tamanho do int:       %ld \n", sizeof(int));
    printf("tamanho do long:        %ld \n", sizeof(long));
    printf("tamanho do float:       %ld \n", sizeof(float));
    printf("tamanho do double: %ld \n", sizeof(double));
    return 0;
}
```

- ❖ Dependendo do compilador, também podemos definir variáveis inteiras de tamanhos diferentes:

Tipo	Descrição
int8_t	Inteiro com sinal de 1 byte
int16_t	Inteiro com sinal de 2 bytes
int32_t	Inteiro com sinal de 4 bytes
int64_t	Inteiro com sinal de 8 bytes
uint8_t	Inteiro sem sinal de 1 byte
uint16_t	Inteiro sem sinal de 2 bytes
uint32_t	Inteiro sem sinal de 4 bytes
uint64_t	Inteiro sem sinal de 8 bytes

Tipos de variáveis

```
#include <stdio.h>
#include <stdint.h>

int main() {

    printf("tamanho do bool:    %ld \n", sizeof(uint8_t));
    printf("tamanho do char:    %ld \n", sizeof(uint16_t));
    printf("tamanho do int:        %ld \n", sizeof(uint32_t));
    printf("tamanho do long:        %ld \n", sizeof(uint64_t));
    return 0;
}
```

- ❖ São estruturas de dados;
- ❖ Permitem armazenar uma coleção de valores, mas sempre do mesmo tipo;
- ❖ A declaração é feita definindo o tipo de variáveis, o nome do array seguido de colchetes [], com o tamanho do array;
- ❖ Para declarar a coleção, os dados deve estar entre { };
- ❖ O acesso aos elementos é feito através de seu índice;
- ❖ O primeiro elemento tem índice 0, o segundo tem índice 1 e assim por diante, até n-1;
- ❖ Pode se alterar o valor de um elemento acessando seu índice e atribuindo um novo valor;

Arrays

```
#include <stdio.h>

int main() {
    int a [10];
    char b [] = {'a', 'b', 'c', 'd', 'e'};
    int c [] = {12, 13, 14, 15, 16};

    printf("b0: %c \n", b[0]);
    printf("b3: %c \n", b[3]);

    printf("a0: %d \n", a[0]);
    printf("a3: %d \n", a[3]);

    a[0] = c[0] + c[1];
    a[3] = 2 * a[0];

    printf("a0: %d \n", a[0]);
    printf("a3: %d \n", a[3]);

    return 0;
}
```


- ❖ São estruturas de dados;
- ❖ Permitem armazenar um texto;
- ❖ A declaração é feita como **char** nomeDaString [tamanho] = “Isso é uma string”;
- ❖ Para declarar a coleção, os dados deve estar entre “ ”;
- ❖ O acesso aos elementos é feito através de seu índice;
- ❖ O primeiro elemento tem índice 0, o segundo tem índice 1 e assim por diante, até n-1;
- ❖ O ultimo elemento deve ser “\0”, também conhecido como NULL, para indicar o fim da string;
- ❖ Existe uma biblioteca própria para manipular Strings: **string.h**

```
int main() {  
    char frase[6] = "frase";  
    char outraFrase[] = "outraFrase";  
  
    printf("%s \n\n", frase);  
    printf("%s \n\n", outraFrase);  
    printf("%c \n\n", outraFrase[6]);  
  
    outraFrase[6] = 'R';  
    printf("%s \n\n", outraFrase);  
    return 0;  
}
```

Strings

```
#include <stdio.h>
#include <string.h>

int main() {
    char frase[] = "O rato roeu a roupa do rei de roma \n";

    printf("O tamanho da frase é de %ld caracteres \n", strlen(frase));
    printf("O tamanho da estrutura de dados é de %ld bytes \n", sizeof(frase));

    char outraFrase[50];
    strcpy(outraFrase, frase);
    printf("frase copiada: %s", outraFrase);

    strcat(frase, outraFrase);
    printf("\n");
    printf("frase 1: %s", outraFrase);
    printf("\n");
    printf("frase 2: %s", frase);

    return 0;
}
```

- ❖ Não existe uma documentação oficial sobre as bibliotecas disponíveis em C;
- ❖ Mas podemos encontrar manuais e documentos de forma muito fácil na internet;
- ❖ Um exemplo:
https://www.tutorialspoint.com/c_standard_library/string_h.htm

❖ Condicional if:

```
int main() {  
    int a = 10;  
    if(a > 9){  
        printf("a é maior que 9");  
    }  
}
```

❖ Condicional if – else:

```
int main() {  
    int a = 8;  
    if(a > 9){  
        printf("a é maior que 9");  
    }  
    else{  
        printf("a é menor que 9");  
    }  
    return 0;  
}
```

Project 4

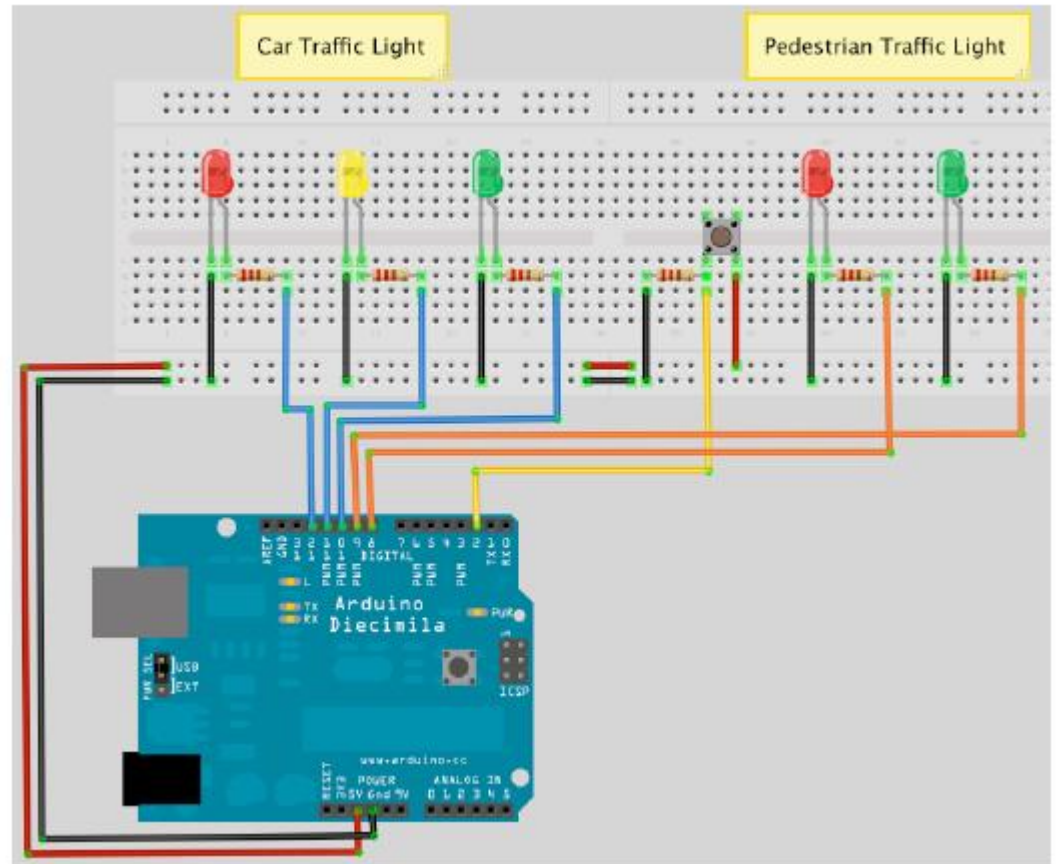
Interactive Traffic Lights

Interactive Traffic Lights

Vamos reproduzir um semáforo de trânsito e de pedestres

Material necessário:

- 1 Arduino;
- 6 Resistores de 220 ohms
- 2 Led Vermelho;
- 1 Led Amarelo;
- 2 Led Verde;
- 1 Protoboard;
- Jumpers cables.



Project 5

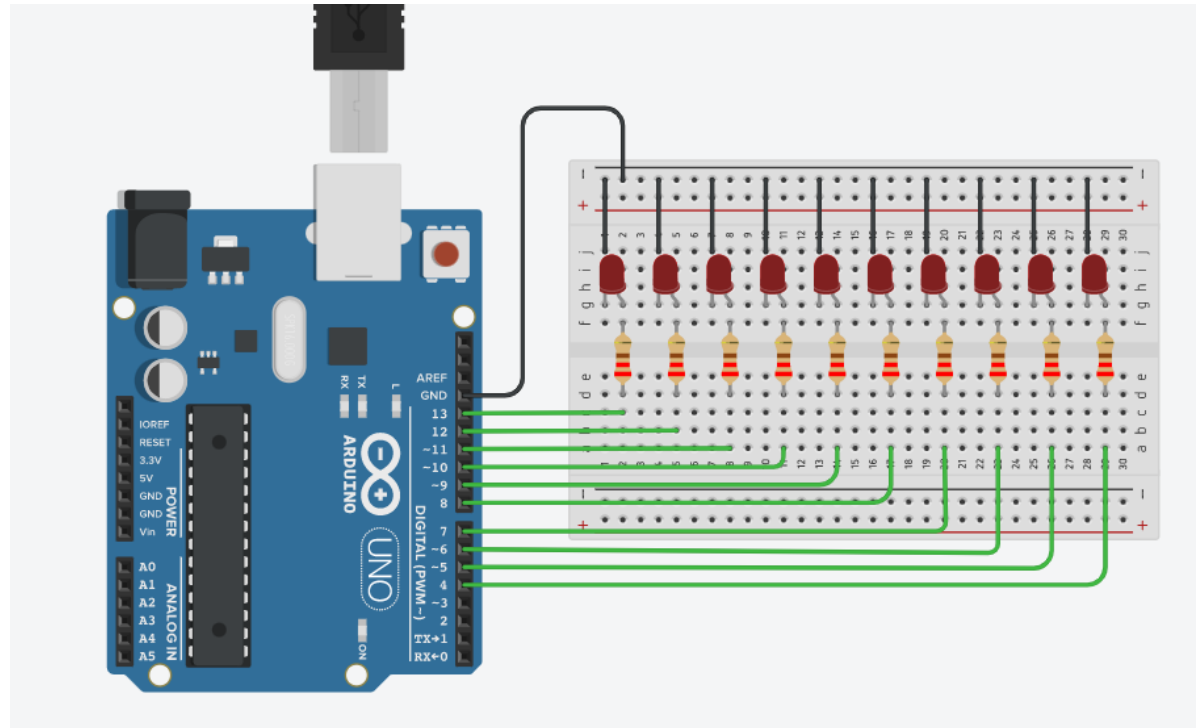
LED Chase Effect

Chase Effect

Vamos simular a KITT do Super Maquina :D

Material necessário:

- 1 Arduino;
- 10 Resistores de 220 ohms
- 10 Leds Vermelho;
- 1 Protoboard;
- Jumpers cables.



Project 6

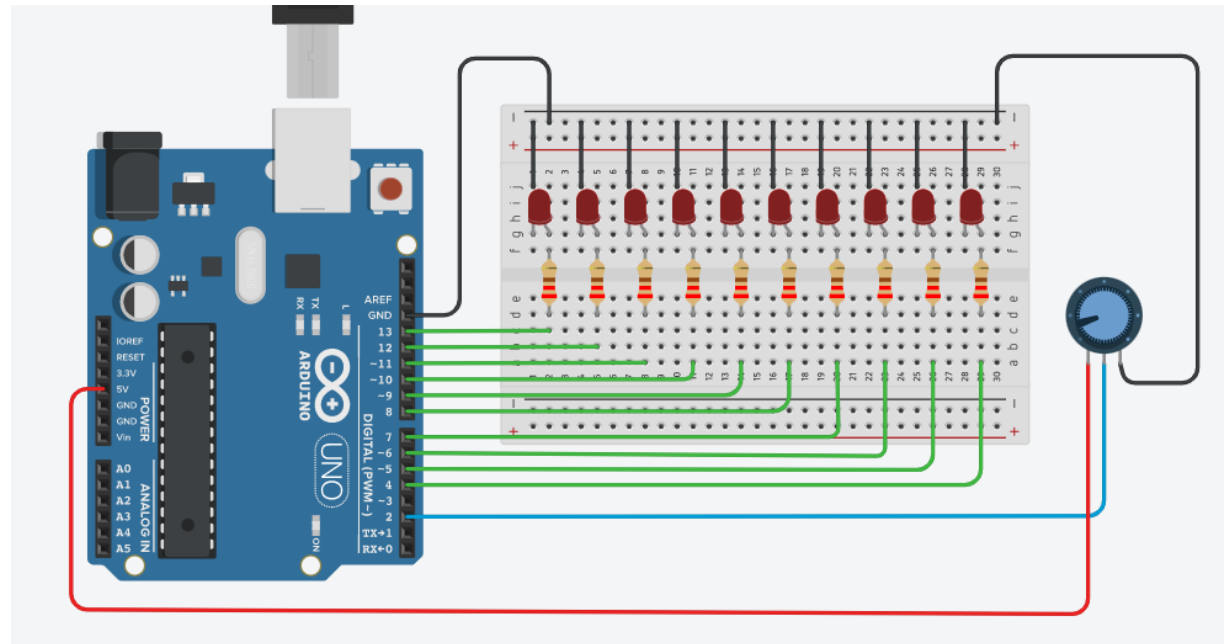
Interactive LED Chase Effect

Interactive Chase Effect

Vamos controlar a KITT do Super Maquina

Material necessário:

- 1 Arduino;
- 10 Resistores de 220 ohms;
- 1 Potenciometro;
- 10 Leds Vermelho;
- 1 Protoboard;
- Jumpers cables.



Copyright © 2023 Prof. Airton Y. C. Toyofuku

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).