**Farol de pedestre e Carros.**

Funções abordadas: **if, else, = =,**

**If;** executa comandos se a condição for verdadeira.

**EX: If (FS = = 1 ) { comandos}** se caso a condição da variável FS, for igual a 1. (Verdadeira) então se executa os comandos dentro das chaves.

**= = : siguinifica validação, de dados. Fazer comparação entre um dados e outro.**

**EX: EX: If (FS = = 1 ) { comandos}** se caso a condição da variável FS, for igual a 1. (Verdadeira) então se executa os comandos dentro das chaves.

**else;** comando se não, se a sentença “if”, for falsa, então se executa a sentença “else” (“se não”)

**EX:**

**if (SB = = HIGH) {** // Se a variável **SB** for igual a HIGH, então se executa as linhas de comando abaixo.

// início das linhas de comando **If (SB = = HIGH)**

**if (FS <4) {** // Se a variável FB for menor que 4, então se executa as linhas de comando abaixo.

// inicio das linhas de comando **if (FS <4)**

**FS = FS + 1;** // comando para igualar o valor contido em FS e somar + 1. (Dando continuidade na sequência; 1,2,3 até 4, chegando em 4, a sentença é falsa e não se executa)

// Fim das linhas de comando **if (FS <4)**

**}  
else {** // comando se não, se a sentença “**if”,** for falsa, então se executa a sentença “**else” (“se não”)**

// início das linhas de comando **else**

**FS = 1;** // FS será igual a 1.

// FIm das linhas de comando **else  
}**

// Fim das linhas de comando **If (SB = = HIGH)**

**}**

**&& :** Teste simultânea de verdade, onde as condições apontadas tem que ser VERDADEIRAS para poder rodar a sentença.

**EX: if ((SB = = HIGH) && ( SAB = = LOW ) ){** // Se a variável **SB** for igual a HIGH & a variavel SAB for igual a LOW, então se executa as linhas de comando abaixo.

EXEMPLO DE CÓDIGO: FAROL DE PEDESTRE E CARROS.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// área também programável, usada para definir ou atribuir variáveis globais, aqueles que existirá em todos os campos de programação do Arduino.

**Int PV = 10 ;** //variável global, usada pra definir a variável PV (abv: PinLED VERMELHO) como a porta 10.

**Int PA = 9;** // variável global, usada pra definir a variável PA (abv: PinLED AMARELO) como sendo a porta 9.

**Int PVD = 8;** // variável global, usada pra definir a variável PVD (abv: PinLED VERDE) como sendo a porta 8.

**Int PB = 7;** // variável global, usada para definir a variável PB (abv: PinBotao) como sendo a porta 7.

**Int PPV = 6;** // variável global, usada para definir a variável PPV (abv: pinpedrestreverde) como sendo a porta 6.

**Int PPVD = 5;** // variável global, usada para definir a variável PPVD (abv: PinPedestreVermelho) como sendo a porta 75

**Int SAB;** // variável global, usada para controlar o estado anterior do botão.

**Int FS;** // variável global, usada para definir quais são os leds, do semáforo, que estarão ligados em cada fase. Vermelho ligado, ou deligado, ou verde ligado, ou deligado, assim por diante. Essa variável será acrescida de números, para alternar entre as fases. (Fase 1, fase 2, fase 3...)

**Int SB;** // variável global, usada para verificar o estado botão

**Int SPL;** // variável global, usada para verificar o estado do pisca led

**Int TPL;** // variável global, usada para verificar o estado do tempo do pisca led.

**void setup () {** // início da área programável de “void setup”

// Put your setup code here, to run once:

**pinMode (PV, OUTPUT);** // Atribuímos o parâmetro de SAIDA na porta PV(Porta 10).

**pinMode (PA, OUTPUT);** // Atribuímos o parâmetro de SAIDA na porta PA(Porta 9).

**pinMode (PVD, OUTPUT);** // Atribuímos o parâmetro de SAIDA na porta PVD(Porta 8).

**pinMode (PB, INPUT);** // Atribuímos o parâmetro de ENTRADA na porta PB(Porta 7). Neste caso,

**pinMode (PPV, OUTPUT);** // Atribuímos o parâmetro de SAIDA na porta PPV(Porta 6).

**pinMode (PPVD, OUTPUT);** // Atribuímos o parâmetro de SAIDA na porta PPVD(Porta 5). Neste caso, precisará de um resistor **PullDown.** (Por não ter usado o comando **INPUT\_PULLUP**.

**FS = 1;**

**SAB = digitalRead (PB);**

**TPL = 0;**

**SPL = HIGH;**

// fim da área programável de “void setup”

**}**

**void loop() {** // início da área programável de “voip loop”

// Put your main code here, to run repeteadly:

**SB = digitalRead (PB) ;**

**if ((SB = = HIGH) && ( SAB = = LOW ) ){** // Se a variável **SB** for igual a HIGH & a variavel SAB for igual a LOW, então se executa as linhas de comando abaixo.

// início das linhas de comando **If (SB = = HIGH)**

**if (FS <4) {** // Se a variável FB for menor que 4, então se executa as linhas de comando abaixo.

// inicio das linhas de comando **if (FS <4)**

**FS = FS + 1;** // comando para igualar o valor contido em FS e somar + 1. (Dando continuidade na sequência; 1,2,3 até 4, chegando em 4, a sentença é falsa e não se executa)

// Fim das linhas de comando **if (FS <4)**

**}  
else {** // comando se não, se a sentença “**if”,** for falsa, então se executa a sentença “**else” (“se não”)**

// início das linhas de comando **else**

**FS = 1;** // FS será igual a 1.

// FIm das linhas de comando **else  
}**

**SAB = SB;** // revalidando o estado do botão como o estado atual.

// Fim das linhas de comando **If (SB = = HIGH)**

**}**

**If (FS = = 1) {** // se caso a condição for VERDADEIRA, as seguintes ações serão executadas.

//início da linha de comandos do campo de parâmetro **if (FS = = 1), (SERT COMO SINAL VERDE ABERTO)**

**digitalWrite (PVD, HIGH);** // Set LEDVERDE como Ligado.

**digitalWrite (PA, LOW);** // Set LEDAMARELO como Desligado.

**digitalWrite (PV, LOW);** // Set LEDVERMELHO como Desligado.

**digitalWrite (PPV, HIGH);** // Set LEDVERMELHO como Ligado.

**digitalWrite (PPVD, LOW);** // Set LEDVERDE como Desligado.

//Fim da linha de comandos do campo de parâmetro **if (FS = = 1).**

**}**

**If (FS = = 2) {** // se caso a condição for VERDADEIRA, as seguintes ações serão executadas.   
//início da linha de comandos do campo de parâmetro **if(FS = = 2),(SERT COMO SINAL AMARELO ABERTO)**

**digitalWrite (PVD, LOW);** // Set LEDVERDE como Desligado.

**digitalWrite (PA, HIGH);** // Set LEDAMARELO como Ligado.

**digitalWrite (PV, LOW);** // Set LEDVERMELHO como Desligado.

**digitalWrite (PPV, HIGH);** // Set LEDVERMELHO como Ligado.

**digitalWrite (PPVD, LOW);** // Set LEDVERDE como Desligado.

//Fim da linha de comandos do campo de parâmetro **if(FS = = 2).**

**}**

**If (FS = = 3) {** // se caso a condição for VERDADEIRA, as seguintes ações serão executadas.   
//início da linha de comandos do campo de parâmetro **If (FS = = 3), (SERT COMO SINAL VERMELHO ABERTO)**

**digitalWrite (PVD, LOW);** // Set LEDVERDE como Desligado.

**digitalWrite (PA, LOW);** // Set LEDAMARELO como Desligado.

**digitalWrite (PV, HIGH);** // Set LEDVERMELHO como Ligado.

**digitalWrite (PPV, LOW);** // Set LEDVERMELHO como Desligado.

**digitalWrite (PPVD, HIGH);** // Set LEDVERDE como Ligado.

//Fim da linha de comandos do campo de parâmetro **If (FS = = 3).**

**}**

**If (FS = = 4) {** // se caso a condição for VERDADEIRA, as seguintes ações serão executadas.   
//início da linha de comandos do campo de parâmetro **If (FS = = 4), (SERT COMO SINAL VERMELHO ABERTO)**

**digitalWrite (PVD, LOW);** // Set LEDVERDE como Desligado.

**digitalWrite (PA, LOW);** // Set LEDAMARELO como Desligado.

**digitalWrite (PV, HIGH);** // Set LEDVERMELHO como Ligado.

**digitalWrite (PPV, SPL);** // Set LEDVERMELHO como Desligado.

**digitalWrite (PPVD, LOW);** // Set LEDVERDE como Desligado.

**TPL = TPL + 1;**

**If (TPL == 400) {  
 STL = !SPL;  
 TPL = 0;  
}**

//Fim da linha de comandos do campo de parâmetro **If (FS = = 4).**

**}**

**delay (1);** //para dar tempo de soltar o botão.

// Fim da área programável de “voip loop” após isso irá voltar ao início e recomeçar o Loop, \*sempre devemos pensar de forma Cíclica\*

**}**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_