// Semáforos dos Carros - Semáforo 1

const int car1Green = 2;

const int car1Yellow = 3;

const int car1Red   = 4;

// Semáforos dos Carros - Semáforo 2

const int car2Green = 8;

const int car2Yellow = 9;

const int car2Red   = 10;

// Semáforo do Pedestre

const int pedGreen = 5;

const int pedRed   = 6;

// Botão do Pedestre

const int buttonPin = 7;

bool pedestrianRequest = false;

void setup() {

  // Configura os pinos dos semáforos dos carros como saída

  pinMode(car1Green, OUTPUT);

  pinMode(car1Yellow, OUTPUT);

  pinMode(car1Red, OUTPUT);

  pinMode(car2Green, OUTPUT);

  pinMode(car2Yellow, OUTPUT);

  pinMode(car2Red, OUTPUT);

  // Configura os pinos do semáforo do pedestre como saída

  pinMode(pedGreen, OUTPUT);

  pinMode(pedRed, OUTPUT);

  // Configura o botão como entrada

  pinMode(buttonPin, INPUT);

  // Estado inicial: Carro 1 verde, Carro 2 vermelho, pedestre vermelho

  digitalWrite(car1Green, HIGH);

  digitalWrite(car1Yellow, LOW);

  digitalWrite(car1Red, LOW);

  digitalWrite(car2Green, LOW);

  digitalWrite(car2Yellow, LOW);

  digitalWrite(car2Red, HIGH);

  digitalWrite(pedGreen, LOW);

  digitalWrite(pedRed, HIGH);

}

void loop() {

  // Carro 1 verde, Carro 2 vermelho

  controlPhase(car1Green, car1Yellow, car1Red, car2Green, car2Yellow, car2Red, 10000);

  // Carro 2 verde, Carro 1 vermelho

  controlPhase(car2Green, car2Yellow, car2Red, car1Green, car1Yellow, car1Red, 10000);

}

// Função para controlar a fase dos semáforos dos carros e verificar o botão

void controlPhase(int green1, int yellow1, int red1, int green2, int yellow2, int red2, int duration) {

  // Ligar o semáforo ativo

  digitalWrite(green1, HIGH);

  digitalWrite(red2, HIGH);

  digitalWrite(red1, LOW);

  digitalWrite(green2, LOW);

  unsigned long startTime = millis();

  while (millis() - startTime < duration) {

    if (digitalRead(buttonPin) == HIGH) {

      pedestrianOverride(); // Se o botão for pressionado, executar travessia

      return; // Retorna para reiniciar o ciclo

    }

    delay(100); // Verifica a cada 100ms

  }

  // Amarelo antes de trocar

  digitalWrite(green1, LOW);

  digitalWrite(yellow1, HIGH);

  delay(2000);

  digitalWrite(yellow1, LOW);

  digitalWrite(red1, HIGH);

}

// Função que interrompe para travessia de pedestre

void pedestrianOverride() {

  // Coloca ambos os semáforos de carros em vermelho

  digitalWrite(car1Green, LOW);

  digitalWrite(car1Yellow, LOW);

  digitalWrite(car1Red, HIGH);

  digitalWrite(car2Green, LOW);

  digitalWrite(car2Yellow, LOW);

  digitalWrite(car2Red, HIGH);

  // Pedestre verde, liberar travessia

  digitalWrite(pedRed, LOW);

  digitalWrite(pedGreen, HIGH);

  delay(5000); // 5 segundos para travessia

  digitalWrite(pedGreen, LOW);

  digitalWrite(pedRed, HIGH);

  delay(1000); // Pequeno delay para retorno seguro

}