



UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
Departamento de Engenharia Informática

Exame de Tecnologia da Informática
LEI - Época Recurso
2019-2020
5-02-2020
2ª parte

Nome: _____ Número: _____

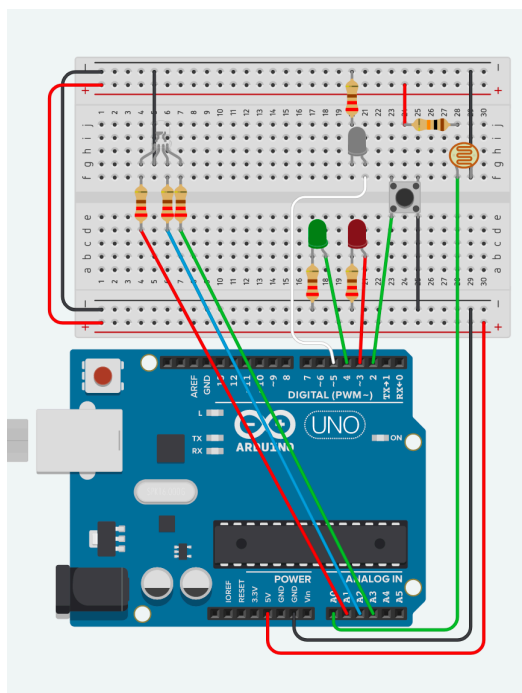
Qualquer tentativa de fraude leva a anulação da prova tanto do facilitador como do prevaricador. Não pode usar dispositivos eletrónicos, incluindo o telemóvel.

(pergunta A removida)

B2. Uma autarquia pretende implementar semáforos inteligentes para controlar o tráfego de veículos e para melhorar as condições de segurança dos peões. Os semáforos de tráfego funcionam com sensores de velocidade, atuando segundo a lógica definida na tabela abaixo. Os semáforos dos peões têm sensores de luz que acendem a iluminação das passadeiras quando o nível de luz é baixo (inferior a 512) e possuem um botão para ativar a luz verde para os peões e vermelha para o semáforo de tráfego.

Velocidade /Evento	Semáforo Tráfego	Semáforo peões
0 – 40 km/h	Verde	Vermelho
41 – 50 km/h	Amarelo	Vermelho
> 51 km/h	Vermelho	Verde
Botão peões premido	Vermelho	Verde
Botão não premido	(consoante velocidade)	(consoante velocidade)

Foi construído um protótipo recorrendo a um LDR para sensor de luz, a um led RGB para simular o semáforo, a dois LEDs para simular o semáforo dos peões (verde e vermelho) e um terceiro LED para a luz das passadeiras. O botão de pressão para além de ligar o semáforo dos peões tem associado um contador (*count_pedestrian*) para contar o número de vezes que o semáforo foi acendido a pedido dos peões (pressão do botão superior a 2s).



```
1  #define GREEN 0
2  #define RED 1
3  #define YELLOW 2
4  byte val_sem_traffic=GREEN, val_sem_pedestrian=RED;
5  boolean btn_ped =false, buttonStatus=false;
6  unsigned long timestamp;
7  int count_pedestrian=0;
8
9  void setup()
10 {
11     pinMode(A0, INPUT);
12     pinMode(A1, OUTPUT);
13     pinMode(A2, OUTPUT);
14     pinMode(A3, OUTPUT);
15     pinMode(2, INPUT_PULLUP);
16     for (int i=3; i<6;i++)
17         pinMode(i, OUTPUT);
18     timestamp = millis();
19     randomSeed(analogRead(A5));
20 }
21
22 void loop()
23 {
24     int val_LDR=read_LDR_pedestrian();
25     int speed=read_velocidade();
26
27     btn_ped = !digitalRead(2);
28     if (btn_ped && (millis()-timestamp >= 2000) ){
```

```

29     if (!buttonStatus) count_pedestrian++;
30 }
31 if (btn_ped != buttonStatus){
32     buttonStatus= btn_ped;
33     timestamp=millis();
34 }
35 //TODO:Implement logic colours
36 semaphore_traffic(val_sem_traffic);
37 semaphore_pedestrian(val_sem_pedestrian);
38 Serial.println(val_LDR);
39 lights_pedestrian(val_LDR);
40 delay(1000);
41 }

```

(cont B2) Implemente a lógica para atribuir as cores aos semáforos tendo presente as regras apresentadas na tabela. Tenha presente o código já existente e que o seu código deve ser projetado para funcionar ao ser introduzido na **linha 36**. A lógica a implementar deve controlar a cor dos semáforos de tráfego (**val_sem_traffic**) e a cor dos semáforos dos peões (**val_sem_pedestrian**). Assuma que as funções utilizadas no código (ex: **read_velocidade()**) estão implementadas ou serão implementadas.

B3. Tendo presente o exercício anterior, implemente as funções **semaphore_traffic(...)** e **semaphore_pedestrian(...)** que devem acender as luzes dos respetivos semáforos consoante a lógica definida na alínea anterior.

Nome: _____	Número: _____
-------------	---------------

B4. Tendo presente os exercícios anteriores, implemente as funções ***read_LDR_pedestrian(...)*** e ***lights_pedestrian(...)***. Esta última deve implementar a lógica de acender o led de iluminação a partir do valor lido do LDR. Deve ser assumido que é para acender a luz da passadeira para valores de luminosidade inferiores a 512.

B5. Como não teve acesso a um sensor de velocidade aquando do protótipo apresente o código que implementaria na função ***read_velocidade(...)*** de forma a simular a velocidade dos carros. Assuma como valor mínimo os 0km/h e como valor máximo 90km/h.

