

TD 2 – Sistema de Controlo de Semáforos

Eletrónica 2022/2023

1. Enquadramento e objetivos

O semáforo é um instrumento utilizado para controlar o tráfego de veículos em cidades em quase todo o mundo (Fig. 1).



Fig. 1 – Fotografia de três semáforos

Suponha que, numa cidade, existe um cruzamento no qual se pretendem instalar três semáforos por forma a controlar o fluxo do trânsito. O cruzamento é composto por duas vias: uma principal e uma secundária, como é possível constatar na Fig. 2.

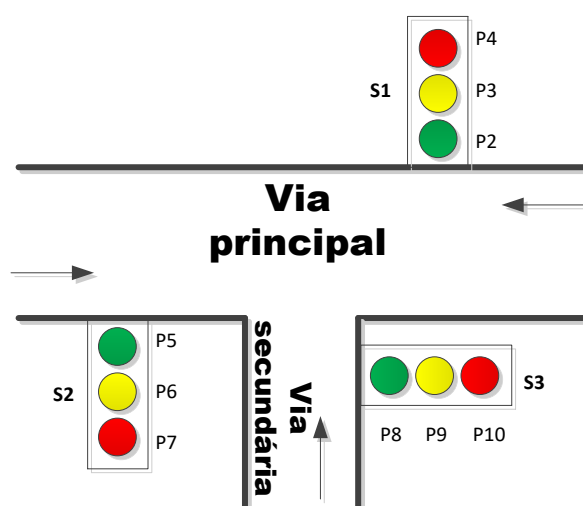


Fig. 2 – Cruzamento no qual se pretende instalar três semáforos para controlar o fluxo do trânsito

As viaturas podem seguir os trajetos identificados na Fig.3.

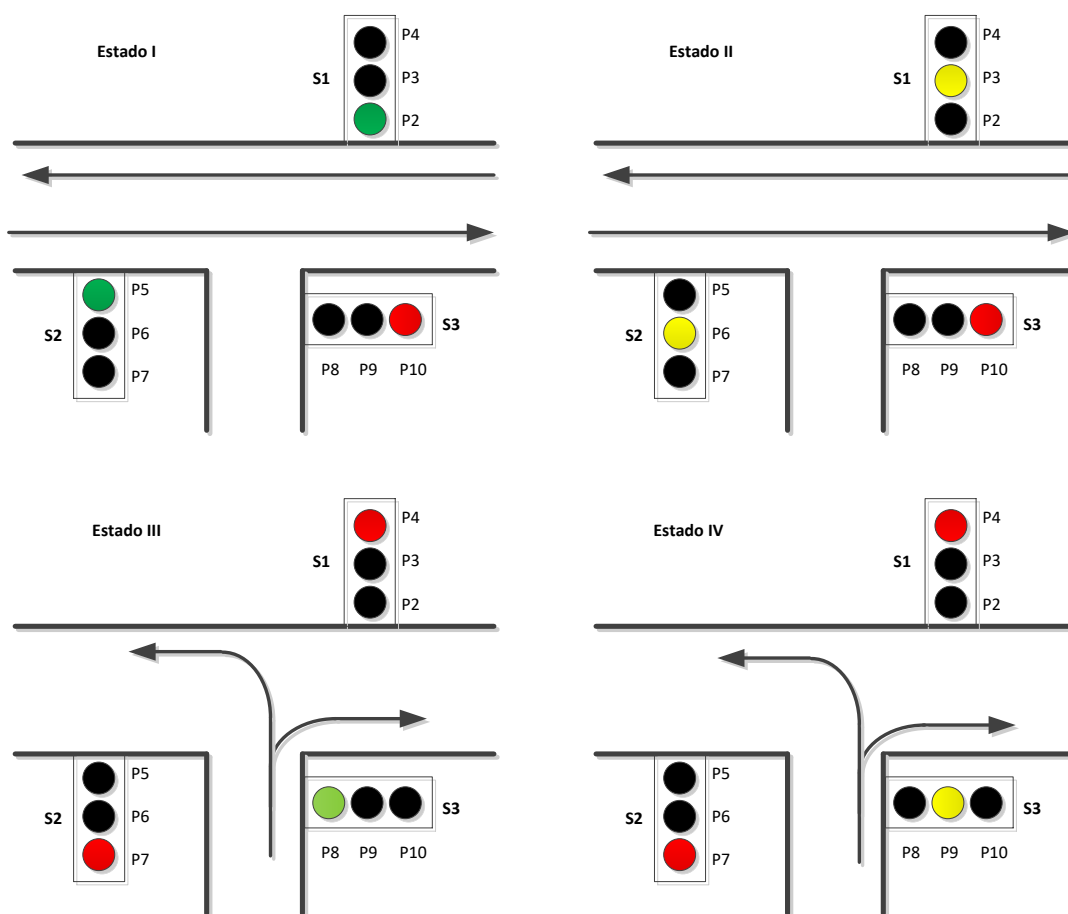


Fig. 3 – Trajetos que as viaturas podem seguir

Neste trabalho pretende-se desenvolver o sistema de controlo dos três semáforos. A figura anterior mostra a sequência de estados que o sistema deve assumir:

- No estado I (os veículos apenas podem circular na via principal):
 - o semáforo 1 (s1) e o semáforo 2 (s2) devem acender simultaneamente a luz verde para que os veículos na via principal possam circular em ambos os sentidos. Os veículos da via secundária devem esperar que o semáforo 3 (s3) acenda a luz verde para avançarem. Este estado tem um tempo de duração de 30 segundos.
- No estado II (estado de transição para o estágio III):
 - o semáforo 1 (s1) e o semáforo 2 (s2) devem acender simultaneamente a luz amarela para que os veículos na via principal saibam que a luz vermelha está prestes a acender. Os veículos da via secundária devem esperar que o semáforo 3 acenda a luz verde para avançarem. Este estado tem um tempo de duração de 5 segundos.
- No estado III (os veículos apenas podem circular da via secundária para a via principal):

- o semáforo 1 (s1) e o semáforo 2 (s2) devem acender simultaneamente a luz vermelha e o semáforo 3 (s3) a luz verde para que os veículos da via secundária possam seguir para a via principal sem que haja qualquer acidente. Este estado tem um tempo de duração de 15 segundos.
- No estado IV (estado de transição para o estado I):
 - o semáforo 1 (s1) e o semáforo 2 (s2) devem acender simultaneamente a luz vermelha e o semáforo 3 (s3) a luz amarela para que os veículos da via secundária se apercebem que o sinal s3 está prestes a ficar vermelho. Este estado tem um tempo de duração de 5 segundos.

Para o projeto deste circuito serão necessários nove *LEDs* (três vermelhos, três amarelos e três verdes), nove resistências, que têm como função limitar a corrente que atravessa os *LEDs*, e um microcontrolador, que irá controlar os três semáforos. A linguagem de programação utilizada para programar o microcontrolador é o *Python*.

Chama-se a atenção que este projeto poderia ser implementado exclusivamente em *Hardware* (sistemas digitais - método baseado em lógica discreta) como é possível constatar no manual: Acácio Amaral, “Eletrónica Digital: Fundamentos e projeto”, Edições Silabo, 2019.

2. Díodo Emissor de Luz – *Ligth Emitting Diode (LED)*

Um *LED (Ligth Emitting Diode)* é um díodo especial capaz de emitir luz. Quando um *LED* se encontra diretamente polarizado, os eletrões livres pertencentes ao semicondutor do tipo *N* atravessam a junção, eliminando as lacunas existentes no semicondutor do tipo *P*. Ao eliminarem as lacunas, descem da banda de condução para a banda de valência libertando energia. Num díodo convencional, a libertação de energia é manifestada sob a forma de calor, no caso dos *LEDs*, manifesta-se essencialmente pela emissão de luz ^[1].

O símbolo usado para representar o *LED* pode ser observado na figura seguinte.

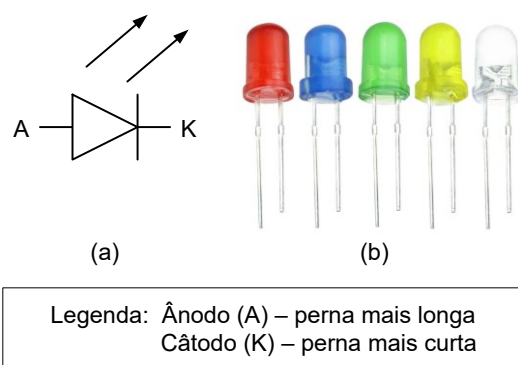


Figura 4 – LED (a) símbolo elétrico e (b) fotografia

O díodo emissor de luz apenas emite luz quando diretamente polarizado, sendo que a cor da luz emitida depende do semicondutor utilizado, assim como dos materiais dopantes. A curva

característica de um *LED* é semelhante à do díodo convencional, sendo que a sua tensão de arranque é aproximadamente 2 *Volts*. A corrente no *LED* deve ser mantida entre 10 *mA* e 50 *mA*, sendo que o seu brilho aumenta à medida que a corrente aumenta ^[1].

3. Sistema Embebido

Um microcontrolador (μC) é um pequeno computador de propósito específico construído num único circuito integrado (*CI*). O referido *CI* é composto por um microprocessador (μP), memórias e periféricos programáveis de entrada e saída ^[3].

Os μC s são utilizados em sistemas embebidos (*SE*), tipicamente desenvolvidos para aplicações específicas com o propósito de controlar dispositivos e processos, tais como eletrodomésticos, automóveis, impressoras, etc.... (Fig. 5).



Figura 5 – Exemplos de sistemas embebidos²

Os μP s utilizados em μC s são bastante mais simples do que os μP s utilizados em *PC*s. Importa referir que, nos *PC*, a ênfase principal é o desempenho, já nos *SE* é o baixo custo, reduzida dimensão e consumo ^[1].

4. Arduino Nano

Neste trabalho será utilizado o microcontrolador *ATmega328* que faz parte da placa de desenvolvimento *Arduino Nano*. A linguagem de programação utilizada para programar o referido μC tem origem em *Wiring*, que é essencialmente C/C++.

O microcontrolador *ATmega328*, que compõe o *Arduino Nano*, encontra-se pré-programado com um *boot loader*. Desta forma, é possível realizar o *upload* de novos programas, a partir do computador, sem que haja necessidade de hardware adicional ^[1].

A figura seguinte mostra o *pinout* da placa de desenvolvimento *Arduino Nano*.

^[1] Acácio Amaral, "Eletrónica Aplicada", Edições Silabo, 2021, Lisboa

² <https://www.utec.edu.pe/blog-de-carreras/ingenieria-electronica/sabias-que-los-sistemas-embebidos-estan-en-casi-todo-lo-que-usamos>

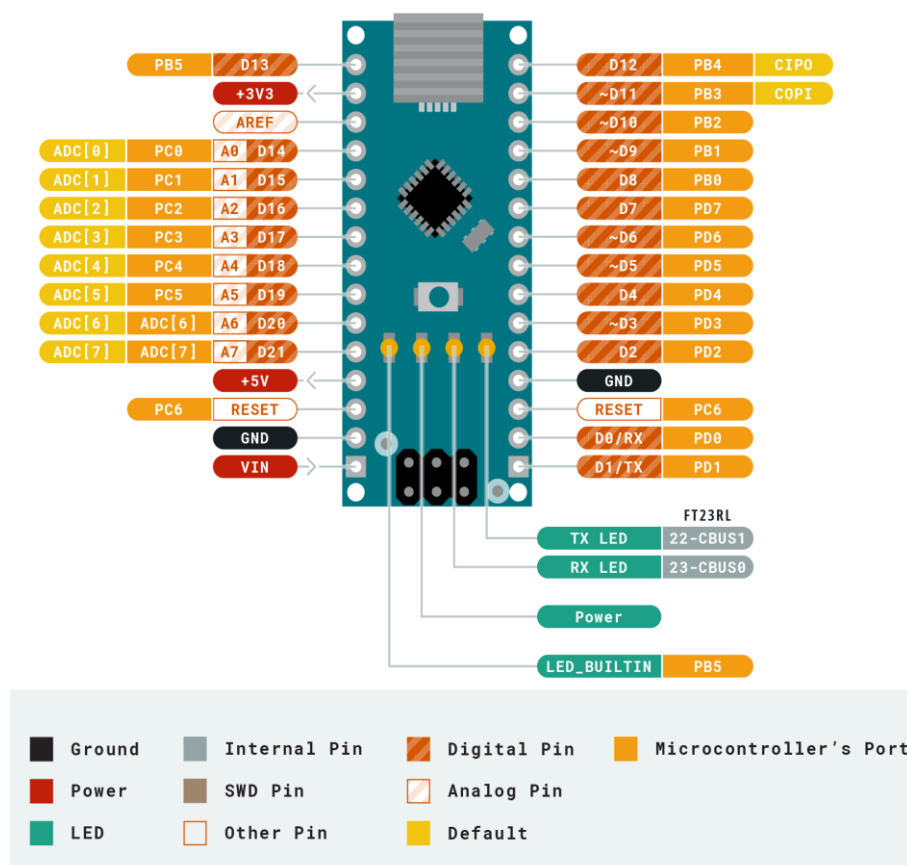


Figura 6 – Pinout Arduino Nano (top view)³

Em seguida apresentam-se os recursos da placa de desenvolvimento *Arduino Nano* que serão utilizados no âmbito deste trabalho ^[1]:

- 12 pinos digitais que começam no pino D_2 e terminam no pino D_{13} . Estes pinos podem funcionar como entrada⁴ ou como saída⁵, sendo essa função definida por *software*;
- Os pinos digitais possuem uma resistência de *pull-up* interna (20 kΩ), que se encontra desligada por defeito e suportam uma corrente máxima de 40 mA;
- Um pino 5 V – fornece uma tensão fixa de 5 V, a qual se associa à tecnologia *TTL*;
- Um pino GND – corresponde à terra, ou seja, os 0 V;
- Três tipos de memórias:
 - *Flash* – a memória *Flash* possui uma capacidade de 32 KB e é responsável por armazenar o código;
 - *SRAM* (2KB) - destinada ao armazenamento dos dados;
 - *EEPROM* (1KB) - destinada a armazenar dados de forma permanente. Este tipo de *ROM* pode ser apagada eletricamente.

³ <https://docs.arduino.cc/hardware/nano>.

⁴ Os pinos devem funcionar como entrada quando se pretende ler uma informação proveniente do exterior, por exemplo, quando se pretende ler um sensor.

⁵ Os pinos devem funcionar como saída quando se pretende enviar um comando para um atuador, por exemplo, quando se pretende acender/apagar um *LED* ou ligar/desligar um motor.

O *Arduino Nano* possui uma ligação *mini-B USB*, logo é possível ligar a placa de desenvolvimento a um *PC* por intermédio de um cabo *USB/A* para *mini USB/B*. O referido cabo pode ser utilizado para alimentar a placa, mas também para realizar o *upload* do programa do computador ^[1].

Como foi referido no início desta secção, a linguagem de programação utilizada para programar o microcontrolador *ATmega328*, que faz parte da placa de desenvolvimento *Arduino Nano*, é essencialmente *C/C++*. No entanto, neste trabalho iremos utilizar a linguagem de programação *Python*, que tem vindo a ser utilizada para simular alguns sistemas eletrónicos, nomeadamente, circuitos com díodos e *amp-ops*.

Para programar este microcontrolador com recurso à linguagem de programação *Python* podem-se utilizar diferentes estratégias. Uma solução passa por executar o programa principal no *PC* e utilizar a ligação série para comunicar com o *Arduino Nano* através do cabo *USB*. Para que o microcontrolador possa compreender a informação proveniente da linguagem de programação *Python* é necessário recorrer ao protocolo *Firmata*.

Assim, antes de escrever o programa em *Python*, que irá controlar o microcontrolador, é necessário fazer o *upload* do protocolo *Firmata* para o μC^6 . Para o efeito, deve utilizar um *sketch*⁷ designado por *StandardFirmata* e que está disponível nos exemplos do *Arduino IDE*.

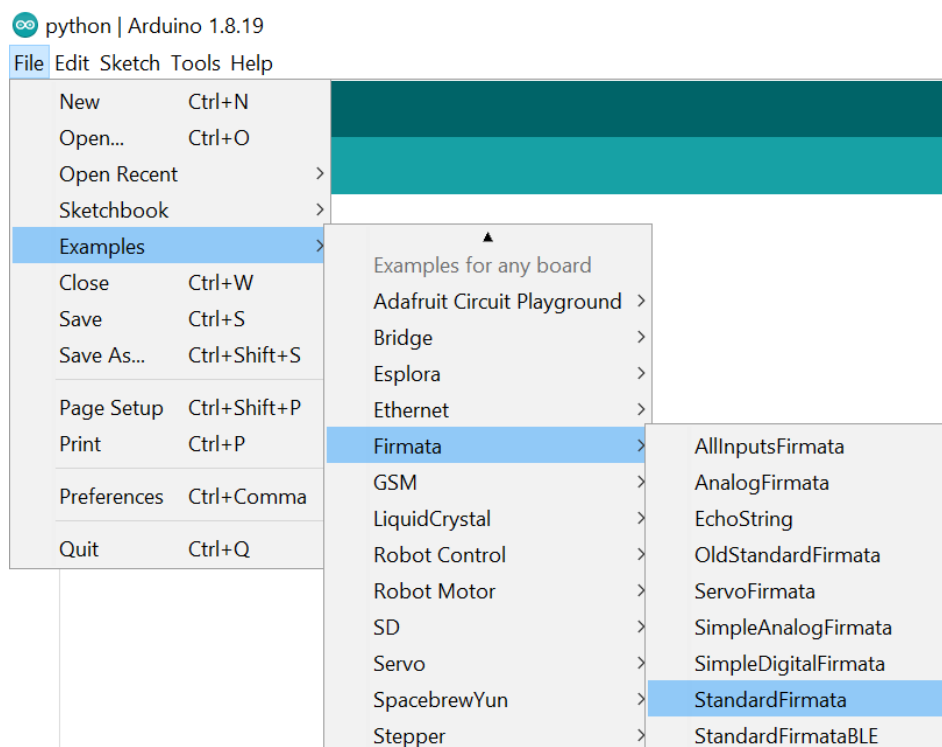


Figura 7 – Caminho para o sketch “Standard Firmata”

⁶ Deve ligar o cabo *USB* entre a placa de desenvolvimento (*Arduino Nano*) e o *PC*.

⁷ Programa desenvolvido no *IDE* do *Arduino*.

Seguidamente já poderá desenvolver o código *Python* no *IDLE*. Para o efeito, deve criar um ficheiro *.py.

5. Projeto – Sistema de controlo dos semáforos - Montagem

Antes de ligar o *Arduino* ao PC através do cabo USB deve efetuar as ligações entre a placa de desenvolvimento e os *LEDs* (Fig. 8). Importa recordar que cada um dos pinos digitais do *Arduino* não suporta mais do que 40 mA, motivo pelo qual é necessário impedir que a corrente que atravessa os *LEDs* ultrapasse esse valor. Para o efeito, utilizam-se resistências de 220 Ω , que são colocadas em série com os *LEDs*. **Após efetuar estas ligações deve solicitar ao docente responsável pelo laboratório que verifique as ligações.**

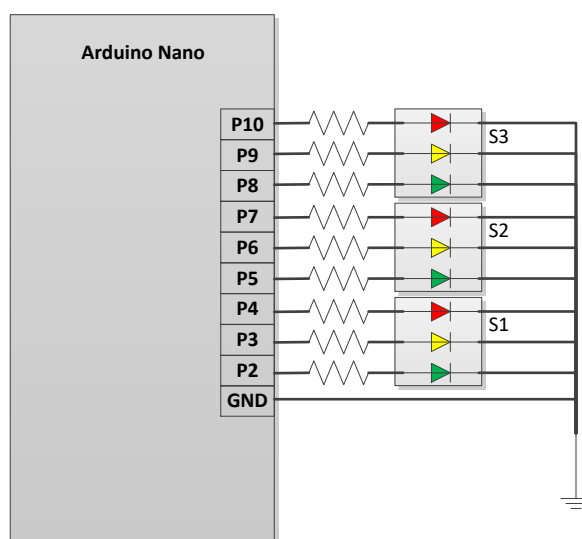


Figura 8 – Ligações entre o Arduino Nano e os LEDs

A ligação do Arduino ao PC só pode ser efetuada após o Professor responsável pelo laboratório verificar as ligações efetuadas (Fig. 8).

6. Projeto – Sistema de controlo dos semáforos - Código

Após efetuar as ligações anteriores, crie um ficheiro *.py, no *IDLE*, e desenvolva o código que permite controlar os semáforos.

1. Num primeiro momento, deve desenvolver o programa principal:

```
import pyfirmata
#biblioteca pyfirmata permite comunicar com Arduino
import time

board = pyfirmata.Arduino('COM4')
# COM4 - corresponde ao porto série onde se encontra ligado o Arduino
# Altere para o porto correcto

while True:
    estado1() # chama procedimento estado1()
    estado2() # chama procedimento estado2()
    estado3() # chama procedimento estado3()
    estado4() # chama procedimento estado4()
```

Figura 9 – Programa principal que permite controlar os semáforos

2. Seguidamente deve desenvolver o código relativo a cada um dos procedimentos, os quais devem ser colocados no topo do programa⁸.

- a. A Fig. 10 mostra o código relativo ao procedimento estado1:

```
def estado1():
    board.digital[2].write(1)
    #escreve no pino digital 2 o valor 1 lógico - TRUE
    board.digital[5].write(1)
    #escreve no pino digital 5 o valor 1 lógico - TRUE
    board.digital[10].write(1)
    #escreve no pino digital 10 o valor 1 lógico - TRUE
    time.sleep(30)
    #interrompe a execução do programa durante 30 segundos
    board.digital[2].write(0)
    #escreve no pino digital 2 o valor 0 lógico - FALSE
    board.digital[5].write(0)
    #escreve no pino digital 5 o valor 0 lógico - FALSE
```

Figura 10 – Código relativo ao procedimento estado 1

- b. Tendo por base o código apresentado na figura anterior, assim como os respetivos comentários⁹, desenvolva o código relativo aos restantes procedimentos: (estado2, estado3 e estado 4).
- c. Seguidamente, execute o código e verifique se o resultado obtido corresponde às especificações do projeto.
3. Suponha que substitui os procedimentos por uma única função (Fig. 11)

```
def estado(P2,P3,P4,P5,P6,P7,P8,P9,P10,tempo):
    board.digital[2].write(P2)
    board.digital[3].write(P3)
    board.digital[4].write(P4)
    board.digital[5].write(P5)
    board.digital[6].write(P6)
    board.digital[7].write(P7)
    board.digital[8].write(P8)
    board.digital[9].write(P9)
    board.digital[10].write(P10)
    time.sleep(tempo)
```

Figura 11 – Código correspondente a uma única função

Altere o programa principal de forma a executar adequadamente a função apresentada na Fig. 11.

4. Em alguns países, tais como Reino Unido, Alemanha, Polónia, Dinamarca, Groelândia e Israel, a sequência de cores inclui um terceiro estado entre a passagem do verde para o vermelho. Nestes casos temos quatro estados:
- a. Estado I – acende luz verde
- b. Estado II – acende luz amarela

⁸ Antes das instruções de *import*

⁹ Texto que surge à direita do carácter #

- c. Estado III – acende simultaneamente a luz vermelha e amarela
- d. Estado IV – acende luz vermelha.

Durante o estado III, todos os veículos estão impedidos de atravessar o cruzamento.

Implemente o sistema de controlo relativo ao problema apresentado. Considere que o estado III tem uma duração de 5 segundos.

7. Material a Utilizar

- 3 – *LEDs* verdes
- 3 – *LEDs* amarelos
- 3 – *LEDs* vermelhos
- 9 – Resistências $220\ \Omega$, $\frac{1}{2}\ W$, 10%
- 1 – Placa de montagem laboratorial
- 1 – Uma placa de desenvolvimento *Arduino Nano*
- 1 – Cabo *USB/A* para mini *USB/B*.

8. Bibliografia

[1] [Amaral, Acácio \(2021\), Eletrónica Aplicada, Edições Sílabo, Lisboa, Portugal.](#)