

TP 5 – Sistema de controlo de uma fechadura eletrónica

Eletrónica 2022/2023

1. Enquadramento e objetivos

As fechaduras eletrónicas (digitais) permitem eliminar um conjunto de problemas associados às fechaduras tradicionais (manuais), nomeadamente, a necessidade do uso de uma chave física. A utilização de uma chave física introduz alguns inconvenientes, já experienciados por todos nós, tais como a necessidade de arrombar a porta em caso de esquecimento da chave no interior do edifício, problemas mecânicos associados ao funcionamento do canhão da fechadura e a perda de tempo associada à abertura da porta.

A introdução de fechaduras eletrónicas permite aumentar a segurança do utilizador, pois:

- Tipicamente, após cinco tentativas consecutivas de abertura incorreta, a própria fechadura ativa um sinal de alarme, identificando uma tentativa de intrusão;
- Possuem um sistema de fecho automático da porta, evitando que a porta fique fechada apenas no trinco;
- Em caso de incêndio destrancam automaticamente. Estes sistemas possuem um sensor de temperatura que, quando deteta um valor superior 65° C, destranca automaticamente a porta.

O sistema de controlo de uma fechadura eletrónica pode utilizar diferentes tecnologias, tais como:

- Abertura da porta baseado num teclado alfanumérico com 12 dígitos;
- Abertura da porta baseada na proximidade de um cartão *RFID* (*radio frequency identification*¹).

Neste trabalho pretende-se desenvolver o sistema de controlo de uma fechadura eletrónica, baseado num teclado alfanumérico. A implementação deste trabalho prático será subdividida em dois pequenos projetos:

- No primeiro projeto, pretende-se que os alunos desenvolvam um sistema que permita identificar qual foi a tecla (teclado matricial) pressionada pelo utilizador;
- No segundo projeto, os alunos devem desenvolver um sistema de controlo que permita a abertura da porta quando o utilizador introduz uma sequência correta de 9 caracteres (*código de acesso*).

¹ Identificação por radio frequência – tecnologia que utiliza a frequência de radio para identificar um objeto ou uma pessoa.

2. Teclado matricial de membrana

O teclado matricial de membrana 4X3 possui 12 teclas, onde 10 teclas representam os 10 algarismos decimais e 2 teclas representam os caracteres (* e #).

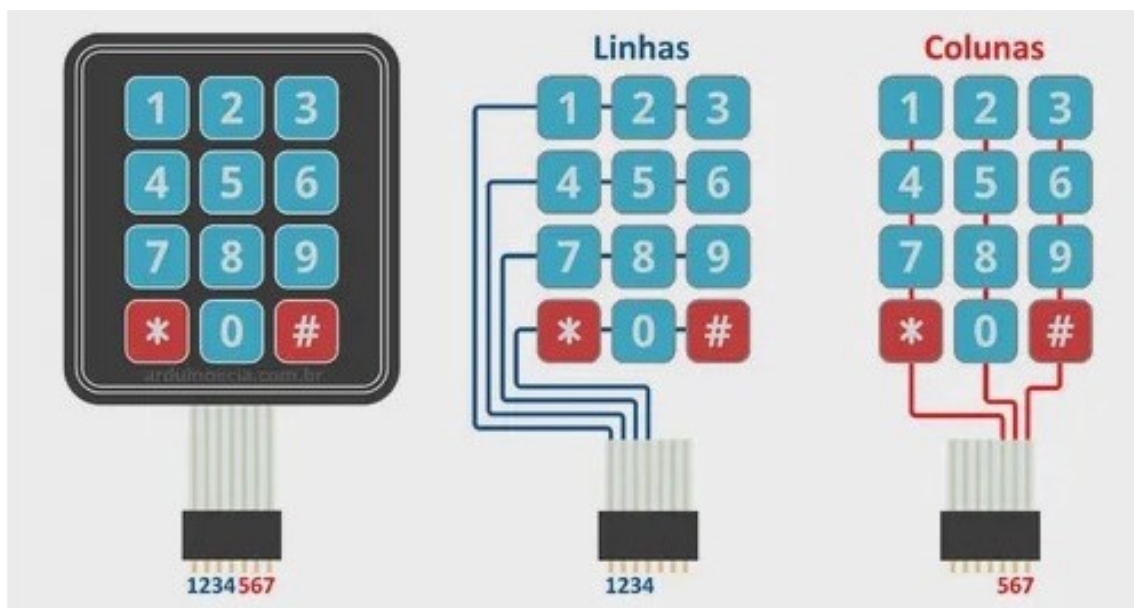


Figura 1 – Teclado matricial de 12 teclas

A figura anterior mostra que as 12 teclas estão dispostas em 4 linhas e 3 colunas e o teclado possui uma ligação de 7 pinos, dos quais

- 4 pinos (da esquerda **1,2,3,4**) permitem identificar as **linhas**;
- 3 pinos (da direita **5,6,7**) permitem identificar as **colunas**.

Suponha que pressiona a tecla 1, nesse caso, o pino 1 e o pino 5 serão ligados entre si, sendo que os restantes pinos ficam em circuito aberto.

Para capturar a tecla pressionada será utilizada a biblioteca *Keypad*, que já se encontra instalada no IDE² do *Arduino*.

3. 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 ^[3].

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

² *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado.

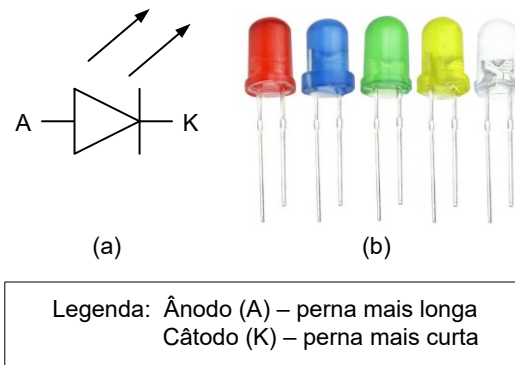


Figura 2 – LED (a) símbolo elétrico e (b) fotografia

O díodo emissor de luz apenas emite luz quando diretamente polarizado, sendo que a cor da luz emitida depende do semiconductor 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 [3].

4. 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. 3).



Figura 3 – Exemplos de sistemas embebidos⁴

^[3] Acácio Amaral, "Eletrónica Aplicada", Edições Silabo, 2021, Lisboa

⁴ <https://www.utec.edu.pe/blog-de-carreras/ingenieria-electronica/sabias-que-los-sistemas-embebidos-estan-en-casi-todo-lo-que-usamos>

Os μ Ps utilizados em μ Cs são bastante mais simples do que os μ Ps utilizados em PCs. Importa referir que, nos PC, a ênfase principal é o desempenho, já nos SE é o baixo custo, reduzida dimensão e consumo [3].

5. 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 μ C 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 [3].

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

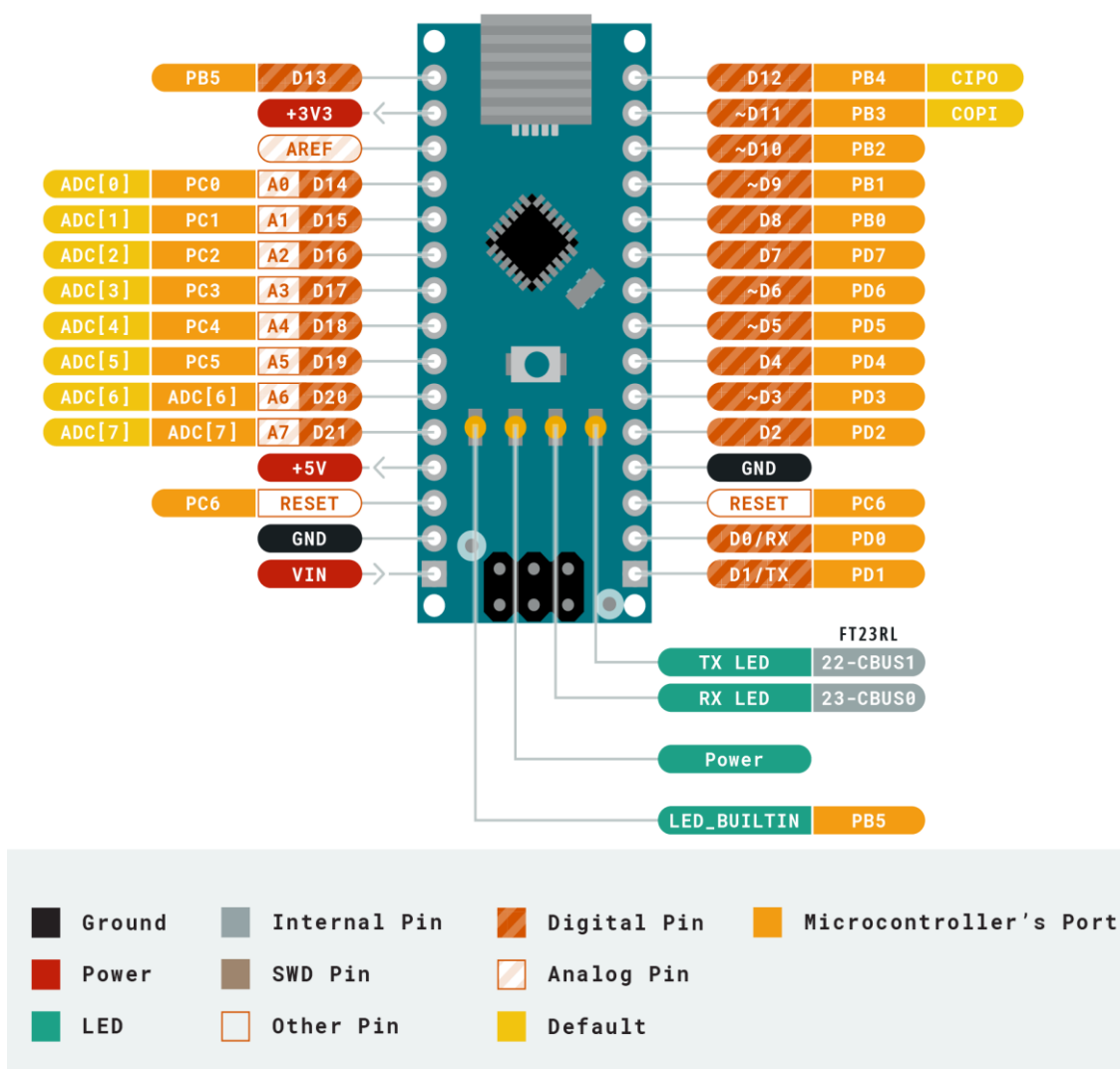


Figura 4 – Pinout Arduino Nano (top view) ⁵

Em seguida apresentam-se os recursos da placa de desenvolvimento *Arduino Nano* que serão utilizados no âmbito deste trabalho [3]:

- 12 pinos digitais que começam no pino D_2 e terminam no pino D_{13} . Estes pinos podem funcionar como entrada⁶ ou como saída⁷, sendo essa função definida por *software*;
- Os pinos digitais possuem uma resistência de *pull-up* interna (20 k Ω), 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.

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 ^[3].

Qualquer programa desenvolvido para esta placa é composto por duas funções pré-definidas:

- *void setup()* – nesta função devem ser definidas as configurações do programa como, por exemplo, iniciar variáveis, definir se um determinado pino irá operar como entrada ou como saída, definir o ritmo de transmissão série, etc... Esta função é apenas executada uma vez ^[3];
- *void loop()* – o código principal deve ser escrito nesta função⁸. A função *loop* repete-se indefinidamente ^[3].

Este trabalho não se foca na linguagem de programação do *Arduino*, motivo pelo qual será disponibilizada a maioria do código necessário à implementação do projeto. O referido código encontra-se comentado e através destes comentários os alunos deverão ser capazes de explicar a sua utilidade e acrescentar pequenas funcionalidades ao sistema.

6. Projeto1 – Identificação da tecla pressionada pelo utilizador

A tecla pressionada pelo utilizador será exibida no ecrã do computador, sendo para o efeito utilizado o programa *Putty*, que já se encontra instalado no *PC*. O *Putty* é um *software* de emulação de terminal grátis e de código livre que permite ler a informação disponível no porto série.

Num primeiro momento, deve efetuar a ligação de 7 fios ao teclado matricial, como é possível observar na figura seguinte e o *Arduino Nano* ao *PC* através do cabo *USB* adequado.

⁵ <https://docs.arduino.cc/hardware/nano>.

⁶ Os pinos devem funcionar como entrada quando se pretende ler uma informação proveniente do exterior, por exemplo, quando se pretende ler um sensor.

⁷ 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.

⁸ Por exemplo ler (*input*) ou escrever (*output*) um valor num pino.

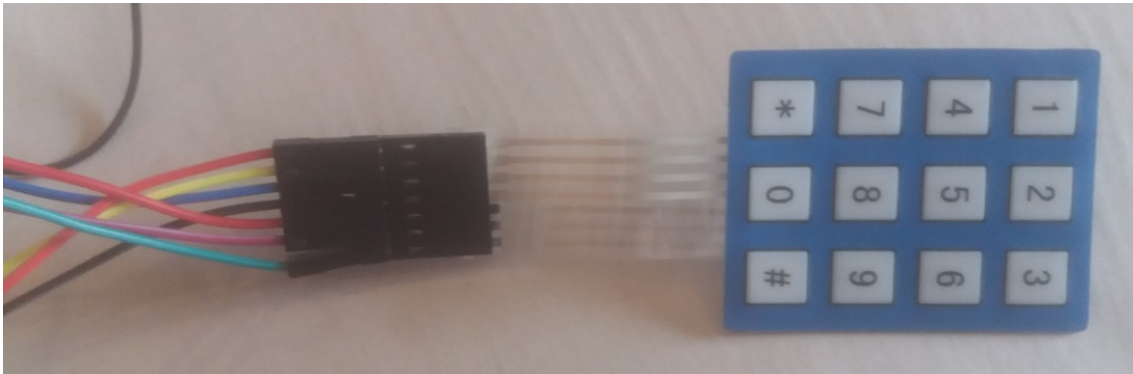


Figura 5 – Ligação de 7 fios ao teclado matricial

O pino 1 do teclado matricial corresponde ao pino que se encontra na posição superior (Fig. 5), sendo que o pino 7 corresponde ao pino que se encontra na posição inferior (Fig. 5).

Seguidamente, deve ligar os pinos ao *Arduino Nano* da seguinte forma:

- O pino 1 do teclado matricial deve ligar ao pino digital 9 do *Arduino Nano* (D9);
- O pino 2 do teclado matricial deve ligar ao pino digital 8 do *Arduino Nano* (D8);
- O pino 3 do teclado matricial deve ligar ao pino digital 7 do *Arduino Nano* (D7);
- O pino 4 do teclado matricial deve ligar ao pino digital 6 do *Arduino Nano* (D6);
- O pino 5 do teclado matricial deve ligar ao pino digital 5 do *Arduino Nano* (D5);
- O pino 6 do teclado matricial deve ligar ao pino digital 4 do *Arduino Nano* (D4);
- O pino 7 do teclado matricial deve ligar ao pino digital 3 do *Arduino Nano* (D3).

No passo seguinte deve abrir o *IDE* do *Arduino* e selecionar a placa de desenvolvimento que será utilizada neste trabalho (Fig. 6).

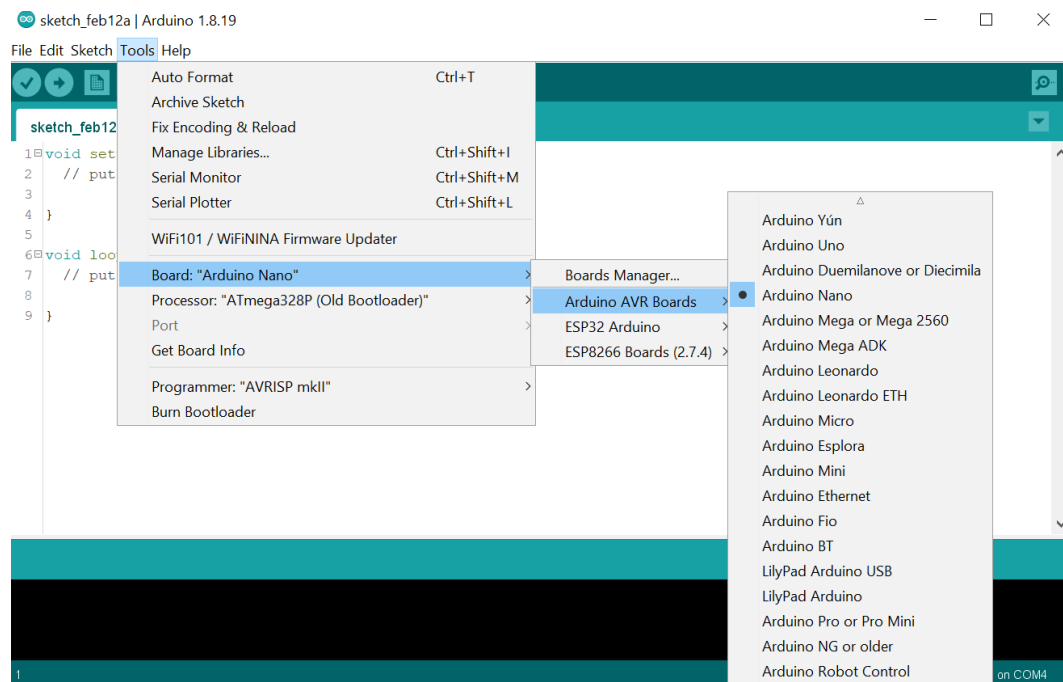


Figura 6 – Seleção da placa de desenvolvimento

Em seguida deve criar um *Sketch* (*Ctrl+N*) e digitar o seguinte código. Os comentários, texto à direita do *//*, não devem ser escritos.

```
#include <Keypad.h> // biblioteca que fornece todos os objetos e métodos necessários para capturar a tecla pressionada
const byte numero_linhas=4; // define o numero de linhas do teclado matricial
const byte numero_colunas=3; // define o numero de colunas do teclado matricial
char valor_teclas[numero_linhas][numero_colunas]={{'1','2','3'},{'4','5','6'},{'7','8','9'},{'*','0','#'}};
// cria array bidimensional com os caracteres presentes no teclado matricial
byte mapear_linhas_pinos[numero_linhas]={9,8,7,6};
// cria um array que permite mapear as linhas do teclado matricial com os pinos do Arduino Nano
byte mapear_colunas_pinos[numero_colunas]={5,4,3};
// cria um array que permite mapear as colunas do teclado matricial com os pinos do Arduino Nano
Keypad teclado=Keypad(makeKeymap(valor_teclas),mapear_linhas_pinos,mapear_colunas_pinos,numero_linhas,numero_colunas);
// cria o objeto teclado com todas as definições anteriores
#define TAM 10

void setup()
{
  Serial.begin(9600);
  // inicio da ligação série
  //Habitualmente utiliza-se uma taxa de transmissão (baud rate) de 9600
}

void loop()
{
  char tecla_pressionada;
  int i;
  Serial.println(""); // introduz \n
  Serial.println("Digite um conjunto de 10 caracteres no teclado matricial"); // escreve a frase e em seguida introduz \n
  Serial.print("> "); // escreve a frase
  for (i=0;i<10;)
  {
    tecla_pressionada=teclado.getKey(); // captura a tecla pressionada
    if (tecla_pressionada!=NO_KEY) // se tecla for pressionada
    {
      Serial.print(tecla_pressionada); // mostra tecla pressionada
      i++;
    }
  }
  Serial.println("");
  Serial.print("Espere 10 segundos: ");
  for (int i=10;i>=0;i--)
  {
    Serial.print(i);
    Serial.print(", ");
    delay(1000); // introduz um atraso de 1s
  }
}
```

Figura 7 – Código relativo ao primeiro projeto

Tipicamente o porto onde se encontra ligado o *Arduino* é automaticamente identificado. Se não for identificado de forma automática deve solicitar ajuda ao Professor.

Para efetuar o *upload* do programa para o *Arduino*, deve digitar os comandos *Ctrl+U*.

Antes de abrir o aplicativo *Putty*, deve identificar qual o porto série onde se encontra ligado o *Arduino*. Tipicamente, essa informação encontra-se disponível no canto inferior direito do *IDE* do *Arduino*. Em alternativa pode selecionar as opções: tools->Port:?

Após efetuar o *upload* do programa para o *Arduino Nano*, deve abrir o aplicativo *Putty* e selecionar as seguintes opções (Fig. 5).

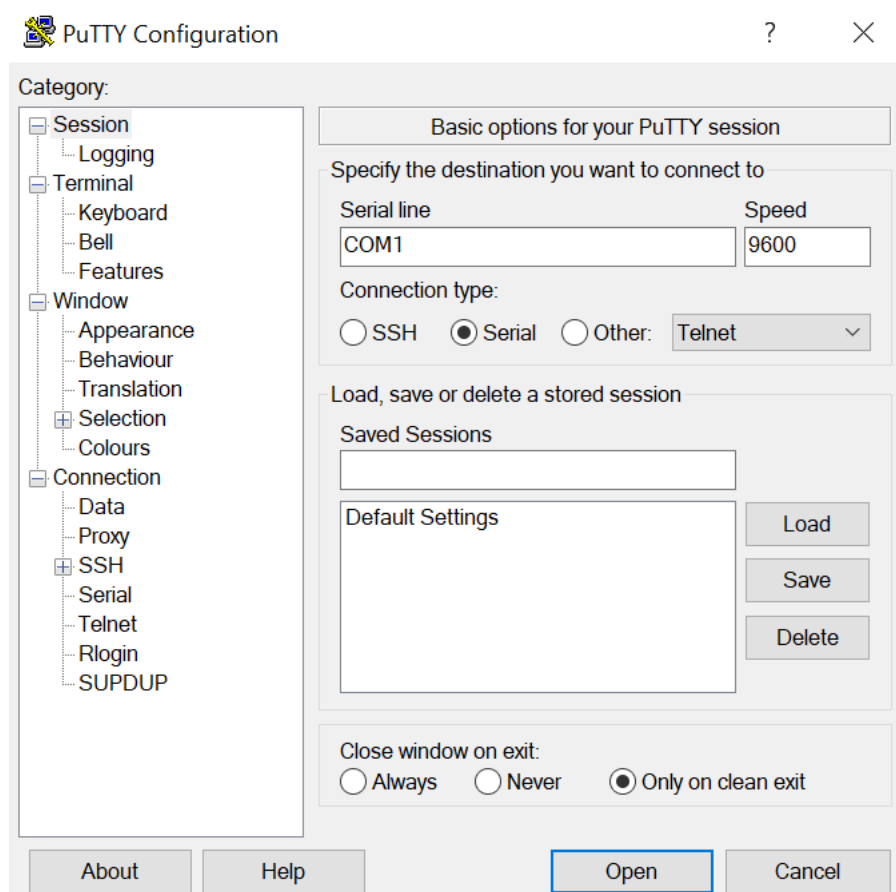


Figura 8 – Opções de configuração do programa *Putty*

Na opção *Serial line* deve colocar a porta série onde se encontra ligado o *Arduino Nano*.

6.1 Projeto 1 - Leitura da tecla pressionada.

Nesta secção serão propostas algumas alterações ao código.

- Suponha que se pretendem ler apenas 5 caracteres consecutivos. Represente a linhas ou linhas de código que deverão ser modificadas. Justifique.
- Algumas fechaduras eletrónicas possuem alguns sistemas de segurança extra, como por exemplo, permitem apenas a leitura de uma certa sequência de teclas pressionadas. Nesse sentido, suponha que se pretendem ler apenas 5 caracteres consecutivos e de forma alternada. Efetue as alterações ao código que lhe permitem ler apenas a seguinte sequência de teclas pressionadas: a primeira, a terceira, a quinta, a sétima e a nona. Represente a linhas ou linhas de código que deverão ser modificadas. Justifique.
- Uma outra solução que poderia ser implementada para reduzir a possibilidade de intrusão seria inverter a sequência de teclas introduzidas. Suponha que o código de acesso possui 10 caracteres. Represente a linhas ou linhas de código que deverão ser modificadas. Justifique.

6.2 Projeto 2 - Sistema de controlo que permita a abertura da porta

Nesta secção pretende-se que os alunos implementem o sistema de controlo de abertura de uma porta. Para simular abertura e fecho de uma porta serão utilizados dois *LEDs*.

- Um *LED* vermelho que representa a porta fechada;
- Um *LED* verde que representa a porta aberta.

O *Arduino* ficará responsável por ligar ou desligar os *LEDs*, motivo pelo qual serão ligados aos pinos digitais do *Arduino*:

- O *LED* vermelho será ligado através de uma resistência 200 Ω ao pino D10. O cátodo do *LED* deve ser ligado ao *GND* do *Arduino* e o ânodo à resistência que, por sua vez, deve ser ligada ao pino D10;
- O *LED* verde será ligado através de uma resistência 200 Ω ao pino D11. O cátodo do *LED* deve ser ligado ao *GND* do *Arduino* e o ânodo à resistência que, por sua vez, deve ser ligada ao pino D11.

A função da resistência é limitar o valor da corrente fornecida pelo *Arduino*.

- a) Considere que a tensão de arranque do *LED* é aproximadamente igual a 2 V, que a corrente máxima suportada pelo *LED* é de 50 mA, e que a corrente máxima que os pinos digitais suportam é 40 mA. Determine qual o valor mínimo que a resistência, colocada em série com *LED*, deve assumir.
- b) Faça as alterações ao código original, tendo em atenção o *Sketch* que se apresenta na Fig. 9.

- i. Indique qual a função das seguintes linhas de código.

```
for (i=0;i<10;)
{
    tecla_preSSIONada=teclado.getKey();
    if (tecla_preSSIONada!=NO_KEY)
    {
        Serial.print(tecla_preSSIONada);
        if (tecla_preSSIONada!=codigo[i])
            abrir=0;
        i++;
    }
}
```

- ii. Indique qual a função das seguintes linhas de código.

```
if (abrir==0)
{
    digitalWrite(10,HIGH);
    digitalWrite(11,LOW);
}
else
{
    digitalWrite(10,LOW);
    digitalWrite(11,HIGH);
}
```

- c) Represente, na forma de um esquema elétrico, as diferentes ligações entre os diferentes componentes.
- d) Suponha que se pretende introduzir um terceiro *LED* amarelo (ligar pino 12) que informe o utilizador sobre o período temporal em que pode introduzir o código. Apresente as alterações que são necessárias introduzir do ponto de vista do *software* e do ponto de vista do *Hardware*. Justifique.

```
#include <Keypad.h>
const byte numero_linhas=4;
const byte numero_colunas=3;
char valor_teclas[numero_linhas][numero_colunas]={{'1','2','3'},{'4','5','6'},{'7','8','9'},{'*','0','#'}};
byte mapear_linhas_pinos[numero_linhas]={9,8,7,6};
byte mapear_colunas_pinos[numero_colunas]={5,4,3};
Keypad teclado=Keypad(makeKeymap(valor_teclas),mapear_linhas_pinos,mapear_colunas_pinos,numero_linhas,numero_colunas);
#define TAM 11

void setup()
{
  Serial.begin(9600);
  pinMode(10,OUTPUT); // Programa o pino digital 10 como saída.
  pinMode(11,OUTPUT); // Programa o pino digital 11 como saída.
  digitalWrite(10,HIGH); // Coloca o pino digital 10 a '1'
  digitalWrite(11,LOW); // Coloca o pino digital 11 a '0'
}

void loop()
{
  char tecla_preSSIONada;
  char codigo[TAM]="16#04#1874";
  int abrir;
  int i;
  Serial.println("");
  Serial.println("Digite um conjunto de 10 caracteres no teclado matricial");
  Serial.print("> ");
  abrir=1;
  for (i=0;i<10;)
  {
    tecla_preSSIONada=teclado.getKey();
    if (tecla_preSSIONada!=NO_KEY)
    {
      Serial.print(tecla_preSSIONada);
      if (tecla_preSSIONada!=codigo[i])
        abrir=0;
      i++;
    }
  }

  if (abrir==0)
  {
    digitalWrite(10,HIGH);
    digitalWrite(11,LOW);
  }
  else
  {
    digitalWrite(10,LOW);
    digitalWrite(11,HIGH);
  }

  Serial.println("");
  Serial.print("Espere 10 segundos: ");
  for (int i=10;i>=0;i--)
  {
    Serial.print(i);
    Serial.print(", ");
    delay(1000);
  }
  digitalWrite(10,HIGH);
  digitalWrite(11,LOW);
}
```

Figura 9 – Código relativo ao segundo projeto

7. Avaliação

A avaliação deste trabalho será baseada no desempenho individual dos alunos nas aulas laboratoriais, complementada com a classificação da folha de resultados laboratoriais.

8. Material a Utilizar

- 1 – Um *LED* verde;
- 1 – Um *LED* amarelo;
- 1 – Um *LED* vermelho;
- 3 – Resistência 220 Ω , $\frac{1}{2}$ W, 10%;
- 1 – Placa de montagem laboratorial;
- 1 – Uma placa de desenvolvimento Arduino Nano;
- 1 – Cabo USB/A para mini *USB/B*.

9. Bibliografia

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

TP 5 – Sistema de controlo de uma fechadura eletrónica

Eletrónica 2022/2023

6.1 a)

Suponha que se pretendem ler apenas 5 caracteres consecutivos. Represente a linhas ou linhas de código que deverão ser modificadas. Justifique.

6.1 b)

Algumas fechaduras eletrónicas possuem alguns sistemas de segurança extra, como por exemplo, permitem apenas a leitura de uma certa sequência de teclas pressionadas. Nesse sentido, suponha que se pretendem ler apenas 5 caracteres consecutivos e de forma alternada. Efetue as alterações ao código que lhe permitem ler apenas a seguinte sequência de teclas pressionadas: a primeira, a terceira, a quinta, a sétima e a nona. Represente a linhas ou linhas de código que deverão ser modificadas. Justifique.

6.1 c)

Uma outra solução que poderia ser implementada para reduzir a possibilidade de intrusão seria inverter a sequência de teclas introduzidas. Suponha que o código de acesso possui 10 caracteres. Represente a linhas ou linhas de código que deverão ser modificadas. Justifique.

6.2 a)

Considere que a tensão de arranque do *LED* é aproximadamente igual a 2 V, que a corrente máxima suportada pelo *LED* é de 50 mA, e que a corrente máxima que os pinos digitais suportam é 40 mA. Determine qual o valor mínimo que a resistência, colocada em série com *LED*, deve assumir.

6.2 bi)

Indique qual a função das seguintes linhas de código.

```
for (i=0;i<10;)
{
    tecla_preSSIONada=teclado.getKey();
    if (tecla_preSSIONada!=NO_KEY)
    {
        Serial.print(tecla_preSSIONada);
        if (tecla_preSSIONada!=codigo[i])
            abrir=0;
        i++;
    }
}
```

6.2 bii)

Indique qual a função das seguintes linhas de código.

```
if (abrir==0)
{
    digitalWrite(10,HIGH);
    digitalWrite(11,LOW);
}
else
{
    digitalWrite(10,LOW);
    digitalWrite(11,HIGH);
}
```

6.2 c)

Represente na forma de um esquema elétrico as diferentes ligações entre os diferentes componentes.

6.3 d)

Suponha que se pretende introduzir um terceiro *LED* amarelo (ligar pino 12) que informe o utilizador sobre o período temporal em que pode introduzir o código. Apresente as alterações que são necessárias introduzir do ponto de vista do *software* e do ponto de vista do *Hardware*. Justifique.

Laboratório: _____

Data: ____/____/____

Elementos do Grupo:

Nome: _____

Nº Aluno: _____

Nome: _____

Nº Aluno: _____

Nome: _____

Nº Aluno: _____

Nome: _____

Nº Aluno: _____