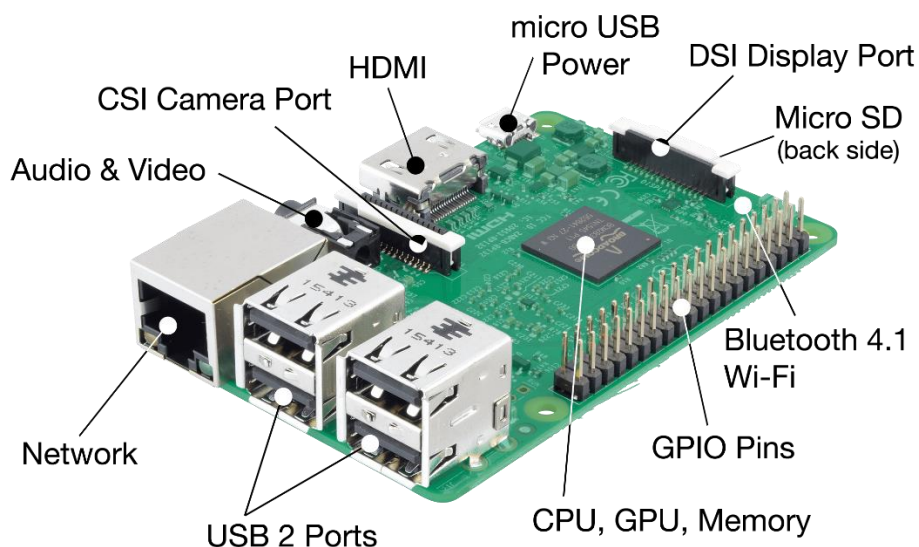


Introdução ao Raspberry Pi

Este documento contém uma série de exercícios a realizar utilizando o Raspberry Pi (<https://www.raspberrypi.org/products/>). A figura seguinte mostra as principais portas do dispositivo.



Links para tutoriais de Raspberry Pi:

- <https://pihw.wordpress.com/guides/direct-network-connection/>
- <http://www.circuitbasics.com/how-to-connect-to-a-raspberry-pi-directly-with-an-ethernet-cable/>
- <https://projects.raspberrypi.org/en/>
- https://elinux.org/RPi_Projects
- <https://www.element14.com/community/welcome>
- <https://ronnyvdbr.github.io/>

Alimentação:

- O Raspberry Pi necessita de alimentação de 5V (geralmente por micro USB)
- Também se pode fornecer alimentação através dos pinos GPIO (a partir de outra placa)
 - Cuidado ao fornecer alimentação a partir do arduino, uma vez que o Raspberry Pi não se protege contra sobre-voltagem. É necessário um conversor de 5V para 3,3V (se não o Raspberry Pi pode ficar danificado)
- Os pinos GPIO também podem ser usados para fornecer alimentação a outras placas (desde que a tensão seja 5V)

geralmente, um output de 3,3V é suficiente. Pode ser necessário um conversor

Configuração Inicial

1. **Setup do cartão de memória:** O Raspberry Pi necessita de um cartão de memória corretamente formatado (bootloader + sistema operativo) para funcionar. Os dois principais sistemas operativos utilizados com o Raspberry Pi são o *Raspbian* (baseado em Debian) e o *Pidora* (Baseado em Fedora).
 - a. Fazer o download do Raspberry Pi Imager¹
 - b. Inserir um cartão de memória com pelo menos 4GB num computador com leitor de cartões de memória
 - c. Formatar o cartão de memória utilizando o Raspberry Pi Imager
2. **Ligações:**
 - a. Introduzir o cartão de memória formatado no Raspberry Pi
 - b. Ligar o cabo HDMI entre o Raspberry Pi e o monitor, o cabo de rede entre o Raspberry Pi e o router (ou conectar por Wi-Fi) e o rato e o teclado nas portas USB
 - c. Ligar a alimentação no micro-USB
3. **Primeiro boot:**
 - a. Instalar o Raspbian Full e efetuar as configurações pedidas: Country / Language / Timezone; Password; Screen; Wi-Fi
 - b. Deixar o configurador atualizar o software e reiniciar o Raspberry Pi
4. **Configurar ligações remotas:** Na consola, introduzir o comando `sudo raspi-config`
 - a. Ativar o SSH: opção 3 - Interface Options | opção I2 - SSH | Ativar. Testar: no computador, instalar o Putty². Criar uma ligação para o IP³ do Raspberry Pi⁴ (Hostname)
 - b. Ativar o VNC: opção 5 - Interface Options | opção I3 - VNC | Ativar. Testar: no computador, fazer instalar o VNC Viewer⁵. Criar uma ligação para o IP³ do Raspberry Pi⁴
 - c. Configurar o Hostname: opção 1 - System options | opção S4 - Hostname | Definir um hostname para o Raspberry Pi.

¹ <https://www.raspberrypi.com/software/>

² **Download Putty:** <https://www.putty.org>

³ Para saber qual é o IP atribuído ao Raspberry Pi, na consola, executar o comando `ifconfig`

⁴ Username: "pi", password: a definida durante a configuração

⁵ **Download VNC viewer:** <https://www.realvnc.com/pt/connect/download/viewer/>

5. Configurar IDE NetBeans para fazer deploy via rede:

- Instalar o NetBeans IDE (download: <https://netbeans.apache.org/download/index.html>)
- No NetBeans ir ao menu “Tools” > “Java Platforms” clicar em “Add Platform” e escolher “Remote Java Standard Edition”
- Preencher:

Platform Name	Raspberry Pi
Host	IP do Raspberry Pi
Username	Username do Raspberry Pi (pi)
Password	Password do Raspberry Pi (idc2022)
Remote JRE Path	em princípio será /usr/lib/jvm/java-11-openjdk-armhf

- Testar a ligação:
 - No NetBeans, criar um projeto (ex: RaspberryPi_teste0) que apenas imprima “Consegui!”.
 - Configurar o NetBeans para que este projeto corra no Raspberry Pi
 - Clicar em “File” e depois em “Project Properties (RaspberryPi_teste0)”
 - Clicar em “Run” à esquerda e depois, em “Runtime Platform”,
 - Escolher o definido no passo anterior (Raspberry Pi) e introduzir um nome para a configuração (ex: Raspberry Pi)
 - Clicar em OK em todas as janelas
 - Executar o projeto
 - Verificar, no Raspberry Pi, se na pasta /home/pi/NetBeansProjects/ apareceu uma pasta com o nome do projeto criado no NetBeans
 - Entrar em /home/pi/NetBeansProjects/ RaspberryPi_teste0/dist e executar o jar (java -jar RaspberryPi_teste0.jar)
 - Deve ser imprimido “Consegui”

6. Configurar IDE NetBeans para user o pi4j:

- Fazer download e extrair o ZIP do pi4j para o computador: <https://pi4j.com/download/pi4j-1.2.zip>
- Criar um projeto no Netbeans (ex: RaspberryPi_teste1). Garantir que “Create Main Class” está selecionado e que a classe se chama ControlGpioExample
- Na árvore do projeto, clicar com o botão direito do rato em “Libraries” e de seguida em “Add Library”
- Criar uma biblioteca usando o nome “Pi4J-Core”
- Em cada Tab, usar “Add JAR/folder” para adicionar os jars extraídos:

Tab	Jar
Classpath	pi4j-core.jar
Sources	pi4j-core-sources.jar
Javadoc	pi4j-core-javadoc.jar

- Na classe ControlGpioExample inserir o código pretendido (referido no “Primeiro projeto Raspberry pi / gpio / java)
- Configurar para correr no raspberry (como na configuração anterior)
- Executar

7. **Instalar pi4j no Raspberry Pi.** No terminal, executar o comando:

```
curl -sSL https://pi4j.com/install | sudo bash
```

8. **Instalar o WiringPi no Raspberry Pi.** No, terminal executar os comandos:

- `sudo apt-get update`
- `sudo apt-get dist-upgrade`
- `git clone https://github.com/WiringPi/WiringPi.git`
- `cd WiringPi`
- `./build`

9. **Instalar MySQL (MariaDB) no Raspberry Pi:**

a. Executar os comandos

- `sudo apt install mariadb-server`
- `sudo mysql_secure_installation`

b. Na configuração, executar os passos:

- Enter
- n + enter
- y + enter
- introduzir password + enter
- voltar a introduzir a password + enter
- n + enter
- n + enter
- n + enter
- n + enter

c. Abrir o mysql- Executar o comando `mysql -u root -p` (introduzir a password definida no passo anterior)

d. Criar uma base de dados com apenas uma tabela (a usar no exercício 6). No mysql, executar o código:

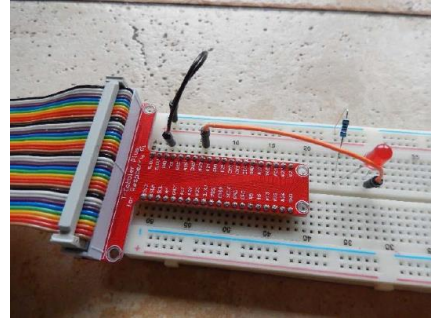
```
create database rpi_teste;
use rpi_teste;
CREATE TABLE registo(
    datahora VARCHAR(50),
    pino VARCHAR(50),
    estado VARCHAR(50));
```

Exercícios

1. Primeiro projeto raspberry pi / gpio / python

Material necessário:

- 1 Raspberry Pi
- 1 Pi Port extender
- 1 breadboard
- 2 conectores (fios)
- 1 LED
- 1 resistência (ex: 10Ω)



Procedimento:

1. Ligar a port extender entre a breadboard e o Raspberry Pi
2. Ligar o GND do port extender ao negativo da placa (fio preto na figura)
3. Ligar a resistência do negativo a outra parte da placa (ver figura)
4. Ligar o negativo do LED (perna mais curta) à resistência (ver figura)
5. Ligar o positivo do LED (perna mais comprida) à porta 23 do port extender (fio laranja na figura)
6. No Raspberry Pi, criar o ficheiro led.py com o seguinte conteúdo:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(23,GPIO.OUT)
print("LED on")
GPIO.output(23,GPIO.HIGH)
time.sleep(1)
print("LED off")
GPIO.output(23,GPIO.LOW)
```

7. Executar o código com o comando `sudo python led.py`

2. Piscar 3 vezes por um segundo com python

Criar o ficheiro led3.py que, em vez de piscar o LED apenas uma vez por um segundo, pisca 3 vezes por um segundo (o tempo para o led estar aceso/apagado é de 1 segundo).

3. Piscar X vezes por Y segundos com python

Criar o ficheiro led_arg.py que, em vez de piscar o LED apenas uma vez por um segundo, ou pisca 3 vezes por um segundo, recebe dois argumentos pela linha de comandos. O primeiro argumento define quantas vezes tem de piscar. O segundo argumento define o tempo que deve estar aceso/apagado. Por exemplo, `sudo python led_arg 5 1` deve piscar o LED 5 vezes por 1 segundo de cada vez.

4. Primeiro projeto Raspberry pi / gpio / java

Com a mesma configuração de hardware utilizada para o Exercício 1 com Python.

Procedimento:

1. No seguimento do ponto 6.f) de “6. Configurar IDE NetBeans para user o pi4j”, inserir na classe ControlGpioExample o código existente no ficheiro codigo1.txt (anexo) (neste caso o LED está ligado ao pino 29 porque, usando o WiringPi/Pi4j, a numeração é diferente, de acordo com a imagem existente na última página do documento).
2. Executar. Verificar que o LED pisca conforme indicado no código.

5. Mostrar na consola o estado de um botão

Ligar um botão com um dos pinos ligado aos 3.3 volts (Pi extender 1) e outro ligado ao GPIO 2 (Pi extender 13).

Procedimento:

1. Num novo projeto semelhante ao anterior, inserir na classe onde se encontra o main o código do ficheiro codigo2.txt (anexo).
2. Executar. Ao carregar no botão, deve aparecer a mudança de estado na consola.

6. Guardar o estado de um botão numa base de dados

Com a mesma configuração de hardware utilizada para o Exercício 1 com Python.

Procedimento:

No projeto anterior, adaptar o código para que, em vez de a informação ser mostrada para a consola, ser guardada numa base de dados

1. Fazer download do ficheiro jar do conetor mysql/java⁶
2. Adicionar o conetor ao projeto:
 - a. Clicar com o botão direito do rato no nome do projeto > Properties
 - b. Do lado esquerdo, seleccionar Libraries
 - c. Clicar em Add JAR/Folder e seleccionar o ficheiro jar obtido
3. Substituir o código da classe main pelo existente no ficheiro codigo3.txt (anexo)
4. Executar. Verificar que, ao pressionar o botão, os registos são inseridos na tabela da base de dados (usar o comando `SELECT * FROM registo;` no mysql)

⁶ <https://static.javatpoint.com/src/jdbc/mysql-connector.jar>

Numeração dos pinos no WiringPi

Raspberry Pi Model B+ (J8 Header)					
GPIO#	NAME			NAME	GPIO#
	3.3 VDC Power	1		5.0 VDC Power	2
8	GPIO 8 SDA1 (I2C)	3		5.0 VDC Power	4
9	GPIO 9 SCL1 (I2C)	5		Ground	6
7	GPIO 7 GPCLK0	7		GPIO 15 TxD (UART)	15
	Ground	9		GPIO 16 RxD (UART)	16
0	GPIO 0	11		GPIO 1 PCM_CLK/PWM0	1
2	GPIO 2	13		Ground	14
3	GPIO 3	15		GPIO 4	4
	3.3 VDC Power	17		GPIO 5	5
12	GPIO 12 MOSI (SPI)	19		Ground	20
13	GPIO 13 MISO (SPI)	21		GPIO 6	6
14	GPIO 14 SCLK (SPI)	23		GPIO 10 CE0 (SPI)	10
	Ground	25		GPIO 11 CE1 (SPI)	11
30	SDA0 (I2C ID EEPROM)	27		SCL0 (I2C ID EEPROM)	31
21	GPIO 21 GPCLK1	29		Ground	30
22	GPIO 22 GPCLK2	31		GPIO 26 PWM0	26
23	GPIO 23 PWM1	33		Ground	34
24	GPIO 24 PCM_FS/PWM1	35		GPIO 27	27
25	GPIO 25	37		GPIO 28 PCM_DIN	28
	Ground	39		GPIO 29 PCM_DOUT	29

Attention! The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>