Raspberry Pi

Manoel Neto

Raspberry Pi

- lançada em fevereiro de 2012
- De 2012 até hoje, muitas placas foram lançadas.
- PC de baixo custo para desenvolvimento de software de forma simples, do tamanho de um cartão de crédito e feita para fins educacionais
- Já vendeu mais de 3 milhões de placas!

Configurações

- O que usamos aqui é o Raspberry Pi 3 Model B
- CPU quad-core de 1,2 GHz (ARMv8 64-bits)
- Wi-Fi 02.11n e Bluetooth 4.0 Low Energy integrados.
- 4 portas USB, HDMI, Ethernet, microSD, entrada de fone de ouvido de 3,5 mm, interface para câmera e vídeo e 40 pinos GPIO.
- US$ 35 (Lá fora).

SOs

- Diversos sistemas operacionais rodam nessa placa: Debian GNU/Linux, Fedora, FreeBSD, Raspbian OS, Slackware Linux, entre outros.
- E como programar?
- Usamos o SO O Raspbian: uma distribuição linux baseada em Debian
- Ela já vem com Python configurado, a linguagem "oficial" da Raspberry Pi, e também com o IDLE 3, um ambiente de desenvolvimento integrado com a linguagem Python.

SOs

- É possível escrever programas que realizam comunicação de baixo nível com o sistema operacional, podendo acessar os periféricos do microprocessador (ver link a seguir)
- http://elinux.org/RPi_Low-level_peripherals

- Além de Python suporta Java, Perl, C, etc.

. https://youtu.be/t-qtqKYBniQ (palestra do autor livro Getting Started with Raspberry Pi)

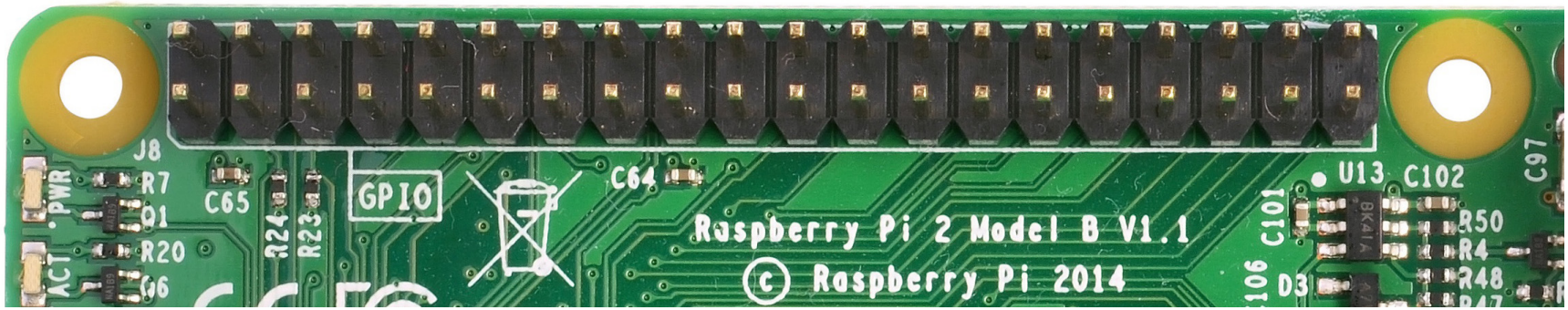
Linguagem Python

- Python possui um de recursos para os sistemas embarcados como.
- Por exemplo, interagir com um GPIO (General Purpose Input Output) da placa Raspberry PI.
- Para quem não conhece Python, um bom tutorial é oferecido em https://osantana.me

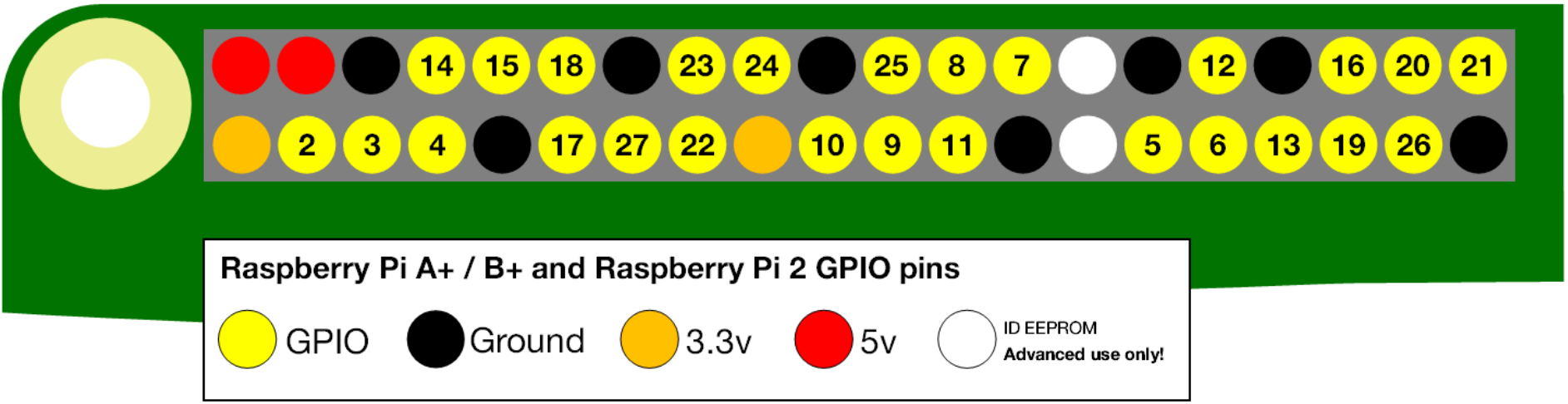
GPIO

- General Purpose Input/Output, um conjunto de pinos responsável por fazer a comunicação de entrada e saída de sinais digitais.
- Cada placa tem um determinado numero de pinos. Veja em: https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/
- Com eles é possível acionar LEDs, Motores, Relês, fazer leitura de sensores e botões, entre outros

Mapa dos Pinos



Mapa dos Pinos



Mapa de Pinos

- A placa suporta portas seriais, que utilizam o protocolo RS-232 para o envio e recebimento de sinal digital.
- I2C (Circuito Inter-integrado). Para quem não conhece, este é um protocolo criado pela Philips em 2006, para fazer conexões entre periféricos de baixa velocidade. No caso da Raspberry, utiliza-se um barramento entre dois fio, sendo um de dados e outro de clock, para comunicação serial entre circuitos integrados montados em uma mesma placa.

Mapa de Pinos

- Pinos com comunicação serial Full Duplex síncrono, que permite o processador do Raspberry comunicar com algum periférico externo de forma bidirecional. Mas essa comunicação só acontece, se e somente se o protocolo for implementado
- portas do ID EEPROM (Electrically-Erasable Programmable Read-Only Memory). Este é um tipo de memória que pode ser programado e apagado várias vezes, através de uma tensão elétrica interna ou externa.

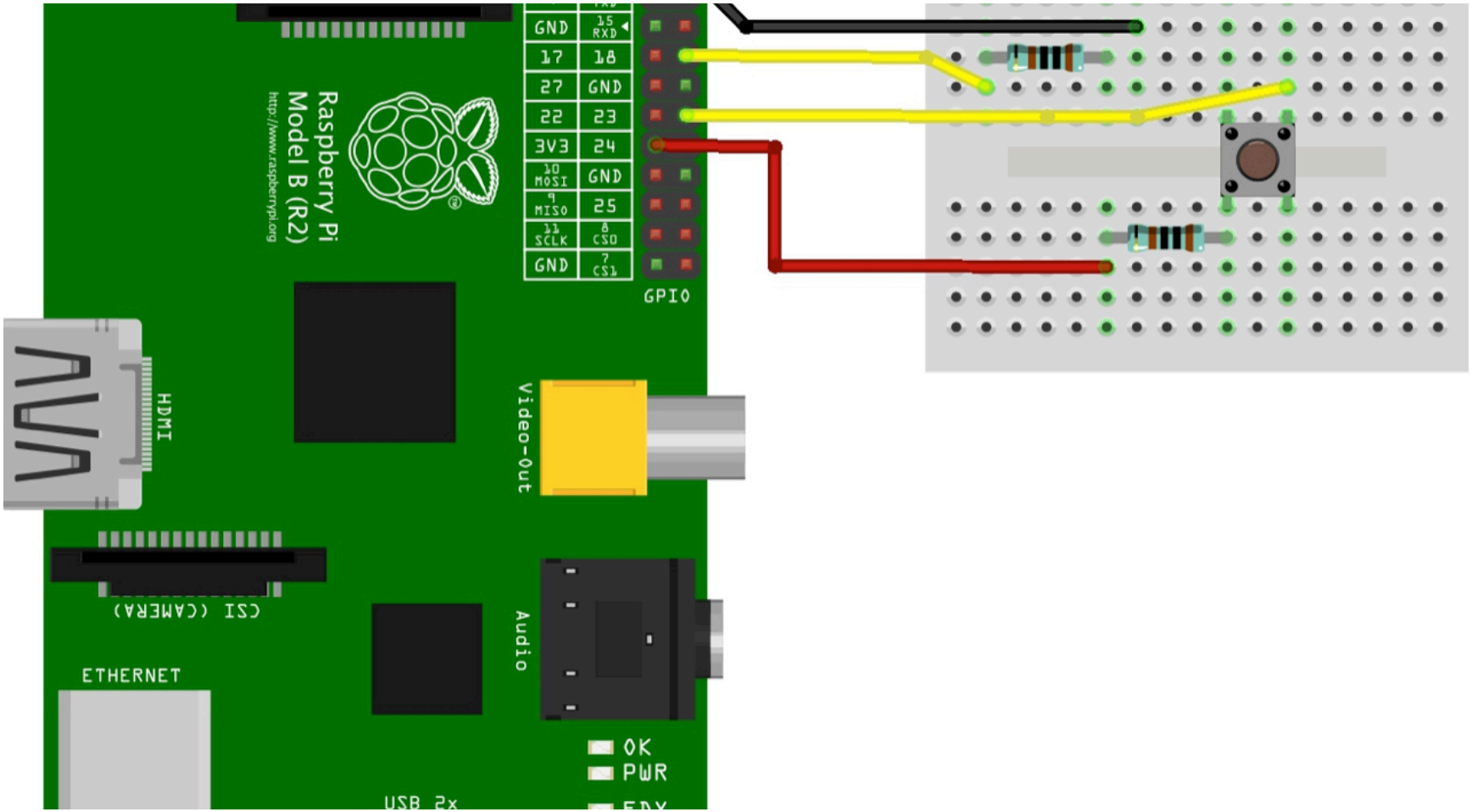
RPI.GPIO

- Vamos preparar o Raspbian para utilizar este módulo, utilizando pip (Gerenciador de Pacotes do Python). Não sabe se possui o pip? Execute o comando abaixo:
  - pip –version
- Se for exibido um erro durante a etapa acima, execute o procedimento abaixo para instalar
- apt-get update && apt-get -f -y install python-pip

RPI.GPIO

- Vamos verificar instalar RPi.GPIO
  - pip install RPi.GPIO
- Caso durante a instalação do RPi.GPIO um erro como “fatal error: Python.h: No such fileor directory” surgir, não se assuste, é ausência dos headers, algumas bibliotecas necessárias e ferramentas do Python para a construção do pacotes, e que é facilmente resolvido instalando o python-dev.
  - apt-get install python-dev

Exemplo Básico



Exemplo Basico

- Use o IDLE 3 ou qualquer editor /IDE python instalado no Raspian
- Crie um novo arquivo e (se seu circuito estiver montado digite o codigo a seguir).

Exemplo Basico

```
#Definindo da biblioteca GPIO
import RPi.GPIO as GPIO

#Aqui definimos que vamos usar o numero de ordem do pino, e não o numero que refere a porta
#Para usar o numero da porta, é preciso trocar a definição "GPIO.BOARD (ex. Pino 12)" para "GPIO.BCM (ex.GPIO 18)"
GPIO.setmode(GPIO.BOARD)
# Setando as portas de entrada e saída
GPIO.setup(12, GPIO.OUT)
GPIO.setup(16, GPIO.IN)
#Loop principal (Laço indefinido)
While(True):
    if GPIO.input == False:
        GPIO.output(12,1)
    if GPIO.output == True:
        GPIO.output(12,0)
```

Exemplo Basico

- O código é bastante simples e auto-explicativo
- inicia importando a biblioteca GPIO
- Define as portas de saída e entrada de sinais digitais
- Depois disso, um laço que é executado indefinidamente para verificar o status do botão.
- Então,se o valor recebido pelo pushbutton for “True” (nível lógico máximo) o LED será desligado. Caso contrário, o mesmo irá permanecer aceso.

Exemplo Basico

- Para executar o código e ver todo circuito funcionar, basta apertar o botão F5 do teclado (Usando o IDLE3)

Principais Funcoes RPI.GPIO

- Iremos ver agora as principais funções do RPi.GPIO
  - RPi.GPIO.setmode()=>Modo de acesso ao GPIO, BOARD (Posição física dos pinos) ou BCM (Numero após GPIO, deve-se tomar cuidado com a revisão da placa pois esta informação pode mudar)
  - RPi.GPIO.setup()=>Configura o pino: (pino e direção [IN ou OUT], exemplo:
    - RPi.GPIO.setup(12, RPi.GPIO.OUT)

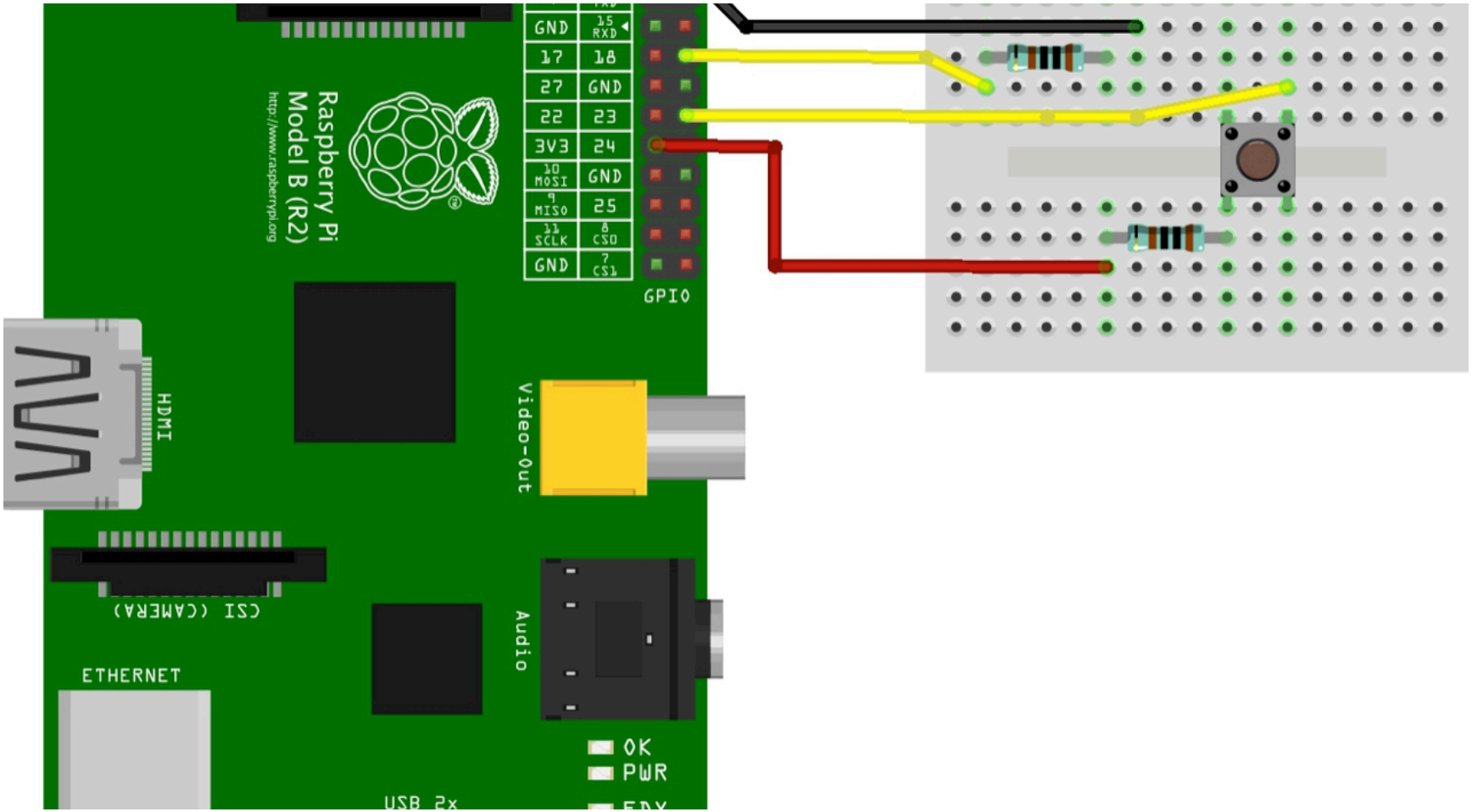
Principais Funcoes RPI.GPIO

- RPi.GPIO.output()=>Configurar como saide: (pino e valor [HIGH ou LOW]), exemplo:
  - RPi.GPIO.output(12, RPi.GPIO.HIGH)
- RPi.GPIO.input()=>Configurar como entrada: (pino) e possui retorno, exemplo:
  - valor_pino = RPi.GPIO.input(12)
- RPi.GPIO.cleanup()=>Restaura para o estado anterior as portas que foram modificadas no programa, deve ser a ultima linha antes de finalizar o programa.

BORD ou BCM?

- Ao configurar o setmode() como BOARD ou BCM, a diferença é o número do pino,
- por exemplo, no esquema informado na Figura do slide a seguir estamos utilizando o Led1(Vermelho) no GPIO18 e o Led2(Verde) no GPIO17

BORD ou BCM?


GND
15 RXD
17
18
27
GND
22
23
3V3
24
10 MOSI
GND
9 MISO
25
11 SCLK
8 CSO
GND
7 CS1
GPIO
Raspberry Pi
Model B (R2)
http://www.raspberrypi.org
HDMI
Video-Out
CSI (CAMERA)
Audio
ETHERNET
OK
PWR
USB 2x

BORD ou BCM?

- Baseado na Figura seguinte temos:
  - BOARD: BCM
  - 11: GPIO17
  - 12: GPIO18

BORD ou BCM?

3.3V	1	2	5V
I2C1 SDA	3	4	5V
I2C1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
GROUND		10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
SP10 MOSI	19	20	GROUND
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
GROUND	25	26	SP10 CE1 N

BORD ou BCM?

- Se o setmode() for setado como BOARD, devemos usar o número 11 e 12 em setup().
- Se for setado como BCM, devemos usar 17 e 18 em setup().
- É importante se atentar neste detalhe para não ter maiores problemas.

Exemplos

- O o código a seguir faz com que um led verde pisque usando Python no modo BOARD, vamos chamar o código abaixo de led1.py.

```
import RPi.GPIO as gpio
import time

# Configurando como BOARD, Pinos Fisicos
gpio.setmode(gpio.BOARD)

# Configurando a direcao do Pino
gpio.setup(11, gpio.OUT) # Usei 11 pois meu setmode é BOARD, se estive usando BCM seria 17
while True:
gpio.output(11, gpio.HIGH)
time.sleep(2)
gpio.output(11, gpio.LOW)
time.sleep(2)

# Desfazendo as modificações do GPIO
gpio.cleanup()
```

Exemplos

- Para executar o codigo, rode:
  - sudo python led1.py

Exemplos

- Novamente o mesmo código, alterando apenas o modo para BCM e trocando para o outro led, no led2.py.

```
import RPi.GPIO as gpio
import time

# Configurando como BCM, Numeracao do GPIO
gpio.setmode(gpio.BCM)

# Configurando a direcao do Pino
gpio.setup(18, gpio.OUT) # Usei 18 pois meu setmode é BCM, se estive usando BOARD seria 12
while True:
gpio.output(18, gpio.HIGH)
time.sleep(2)
gpio.output(18, gpio.LOW)
time.sleep(2)

# Desfazendo as modificações do GPIO
gpio.cleanup()
```

Exemplos

- Para encerrar, vamos interagir com os dois leds, utilizando modo BOARD, no código led1_2.py.

```
import RPi.GPIO as gpio
import time

# Configurando como BOARD, identificacao fisica dos pinos
gpio.setmode(gpio.BOARD)
print " Configurando o modo de acesso ao GPIO - BOARD"
print

# Configurando a direcao do Pino
gpio.setup(11, gpio.OUT)
print "Setando modo OUTPUT no PINO11 GPIO17 [LED VERDE]"

gpio.setup(12, gpio.OUT)
print "Setando modo OUTPUT no PINO12 GPIO18 [LED VERMELHO]"
print

gpio.output(11, gpio.HIGH)
print "Led Verde aceso!"
time.sleep(2)

gpio.output(11, gpio.LOW)
print "Led Verde apagado!"
time.sleep(2)

print

gpio.output(12, gpio.HIGH)
print "Led Vermelho aceso!"
time.sleep(2)

gpio.output(12, gpio.LOW)
print "Led Vermelho apagado!"
time.sleep(2)

# Desfazendo as modificações do GPIO
gpio.cleanup()

print
print "Tchau..."
```

Desafio

- http://raspi.tv/2013/8-x-8-led-array-driven-by-max7219-on-the-raspberry-pi-via-python

- Tente!!!!