

Raspberry Pi aplicado a projetos do mundo real

O Raspberry Pi, pouco maior que um cartão de crédito, torna mais fácil adquirir experiência com sistemas embarcados Linux. Apresentamos alguns exemplos práticos de como usá-lo em ambientes comuns do dia a dia.

por Werner Hein

O engenheiro britânico Eben Upton e uma equipe de hackers de hardware começaram o projeto *Raspberry Pi* como um meio de fornecer tecnologia de computador acessível para jovens interessados no assunto [1]. O objetivo era desenvolver e comercializar um computador de placa única, do tamanho aproximado de um cartão de crédito e compatível com o orçamento apertado do público alvo (quadro 1).

Se o usuário estiver lembrado dos primeiros computadores domésticos, não se sentirá completamente por fora do assunto: a intenção fundamental dos criadores era reocupar porões, garagens e salas de aula com o espírito de uma geração que cresceu com o Atari 400/800, ZX80/81 ou VC20/C64.

Quase um ano depois de surgir o Raspberry Pi, é apropriado olharmos para trás e analisarmos o que aconteceu entre a primeira série de cerca de 10 mil peças e a situação atual de aproximadamente 1 milhão de unidades entregues: qual teria sido a razão de sucesso do projeto? Quais capacidades o hardware oferece? O que é possível e o que não é (ainda) possível fazer com o sistema? Este artigo explora o Raspberry Pi através de três projetos do mundo real:

- ◆ Um multimídia, de sistema de reprodução de vídeo,
- ◆ Um ponto de acesso sem fio, e
- ◆ Uma unidade embarcada de controle de temperatura.

O primeiro projeto é um exemplo do sistema Raspberry Pi servindo como um substituto para um desktop padrão.

Os dois últimos exemplos mostram o Raspberry Pi como um sistema embarcado e que exigirão um pouco de conhecimento de programação, mas, também, ressaltam a versatilidade deste poderoso minicomputador. Talvez todos os três exemplos ajudem a mentalizar outros projetos “faça você mesmo” (de “do it yourself” ou DIY) que podemos criar com um Raspberry Pi próprio.

Lista de compras

Originalmente, os designers haviam previsto um modelo A e B, que diferem principalmente no número de portas USB (uma ou duas), na disponibilidade de uma porta Ethernet, e, evidentemente, no preço (de 25 a 35 dólares, nos Estados Unidos). A esta altura, apenas o modelo B está disponível. Veja a **tabela 1** para um resumo

Quadro 1: História

Em 2006, Eben Upton e sua equipe desenvolveram os primeiros conceitos para o Raspberry Pi, baseados no Atmel ATmega. A ideia de atrair jovens interessados no minicomputador já fazia parte do programa. Em 2009, os membros oficialmente estabeleceram a *Raspberry Pi Foundation*. Em agosto de 2011, a série alpha de aproximadamente 50 placas deixou a produção. Elas serviram principalmente como uma plataforma para desenvolvedores, para depuração e para fins de demonstração. Em dezembro de 2011, foi a vez da série de 25 placas beta, já baseada no layout de produção. Os desenvolvedores eliminaram as últimas falhas. Em um leilão online em janeiro de 2012, 10 placas desta série beta foram vendidas por um total de 16.336 libras no Reino Unido. Em 29 de fevereiro de 2012, às 7 horas (horário da Europa Central), os servidores web da fundação e duas distribuidoras foram sobrecarregadas em poucos minutos pela tempestade de pedidos. Muitos clientes esperaram durante horas para submeter uma pré-encomenda.

dos componentes do Raspberry Pi. Para obter o Raspberry Pi (ou RasPi, abreviadamente) instalado e funcionando, precisamos fazer uma lista dos componentes básicos. A Internet possui ainda mais dicas sobre operações avançadas [2] [3]. Atualmente, apenas dois distribuidores globais têm a placa em catálogo [4] [5]. Embora a entrega de alguns meses não fosse incomum no primeiro trimestre de 2012, agora o usuário geralmente só precisa aguardar algumas semanas. Além da placa, o usuário precisará dos seguintes componentes:

- ♦ Uma fonte de alimentação USB com nada menos do que 700mA. Se desejar um hardware com fonte adicional através de USB, deve planejar pelo menos 1 amp.

- ♦ Um cartão SD (HC) com pelo menos 2GB de capacidade; no entanto, 4GB seria melhor. Dependendo do que o usuário pretende fazer, ainda assim pode não ser suficiente. Como o sistema está também alojada no cartão SD, é aconselhável escolher um cartão com boa leitura e velocidade de escrita.

- ♦ Para acesso sem fio, um adaptador WLAN na forma de um pen-drive (stick USB) seria prático. Mas, na prática, um adaptador WLAN USB é muitas vezes um fator significativo no consumo de energia. Se estiver pensando em usar um adaptador USB, verifique se o driver

é suportado e escolha um design “enxuto” (slim) que não bloqueie a segunda porta USB.

- ♦ Tendo em vista o baixo número de portas USB, investir em um teclado e em um mouse sem fio valeria a pena também para poupar a carga desnecessária sobre os recursos limitados do adaptador de energia USB.

Além destes itens, o usuário também precisará de cabos diversos, tais como HDMI ou Ethernet, dependendo do uso planejado para o dispositivo. No que se refere ao cabo HDMI, quanto menor, melhor. O usuário tem a opção de aumentar a saída do driver, mas isso também aumenta o consumo de energia. Se estiver pensando em ouvir música ou assistir a vídeos, também irá precisar de fones de ouvido ou alto-falantes. O terceiro projeto (um controlador de temperatura embarcado) usa uma placa de expansão que requer um hardware adicional [6].

Primeiros passos

Uma distribuição Linux baseada em Debian conhecida como Raspbian é adaptada para o serviço como um sistema operacional on-board para Raspberry Pi. A Raspberry Pi Foundation recomenda o Raspbian “Wheezy” [7], última versão lançada, utilizada neste artigo.

Comece fazendo o download do arquivo de imagem Raspbian “Wheezy”, que é uma imagem completa do

sistema consistindo de uma partição de boot de 60MB e um sistema de arquivos root em torno de 2GB. Descompacte o arquivo e mova-o para um cartão SD com o seguinte comando:

```
$ sudo dd bs=1M
if=2012-08-16-wheezy-raspbian.img
of=SD_card_device_filename
```

Agora, conecte o cartão SD, teclado, mouse e monitor para o sistema Raspberry Pi e inicialize, ligando a fonte de alimentação USB. O menu de configuração aparece, e se o usuário não estiver usando um teclado padrão em inglês, seria melhor primeiro alterar as configurações nos itens do menu `configure-keyboard` e `change_locale` para o teclado e idioma preferidos; caso contrário, irá encontrar problemas assim que for solicitado o login. O nome de usuário padrão é `pi`; a senha é `raspberrypi`.

Outros itens de menu úteis incluem `change_timezone`, para definir o fuso horário correto; `ssh` para permitir o acesso seguro ao terminal via rede; e `expand_rootfs`, para alterar o tamanho do sistema de arquivos root de 2GB para o tamanho real do cartão SD. Finalmente, devemos atualizar o sistema no modo Debian típico:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Podemos iniciar a ferramenta de configuração a qualquer momento usando `sudo raspi-config`.

Projeto 1: reprodução de vídeo

Entrar com `startx` levará para o desktop gráfico LXDE, o que é otimizado para sistemas enxutos. Reproduzir um vídeo em HD é certamente o último teste para o hardware. Os players de vídeo comuns, no entanto, são de nenhuma ajuda porque a GPU da placa é otimizada para decodificação de áudio e dados de vídeo. Assim, precisaremos de um player que utiliza

Componente:	Tipo:
SoC	Broadcom BCM2835 (system on chip)
CPU	ARM1176JZF-S, 700 MHz
GPU	Broadcom VideoCore IV (OpenGL ES 2.0, OpenVG, 1080p30, H.264 alto nível de codificação/decodificação)
RAM	512MB of RAM
Ports	
USB	2 x USB 2.0
Ethernet	1 x 10/100 Mbps
SD/MMC	1 x slot
Audio	1 x stereo analog out
Video	1 x composite out, 1 x HDMI out

Tabela 1 Dados técnicos do Raspberry Pi.

a própria biblioteca do programa da Broadcom (encontrado em `/opt/vc`).

Atualmente, o único que faz isso é o programa de linha de comando pré-instalado `omxplayer` [8]. Se possível, simplesmente monte o diretório que contém os arquivos na rede. O NFS oferece acesso mais rápido à coleção de vídeos, mas o SSHFS é menos complicado. Os pacotes apropriados estão instalados, como mostrado na **listagem 1**. O usuário, então, inicia a reprodução com o comando `omxplayer <File>`.

Depois de um curto soluço de imagem e som, o software fez o teste de vídeo em HD em formato 720p com áudio Dolby Digital 5.1 – claro, misturando até dois canais – sem qualquer problema. E isso é tudo do que se trata o espartano `Omxplayer`. O programa demonstra capacidades impressionantes para decodificação de áudio e vídeo, mas nada mais. Há também duas desvantagens extras: a biblioteca de vídeo mencionada

Listagem 1: Configuração de um player de vídeo

```
$ sudo apt-get sshfs
$ sudo adduser pi fuse
$ mkdir myVideos
$ sshfs username@host_name:/
directory_name myVideos
```

Listagem 2: Como modificar o arquivo de configuração

```
ctrl_interface=/var/run/wpa_
supplicant
eapol_version=1
ap_scan=1
network={
ssid="(E)SSID"
scan_ssid=1
proto=WPA (or RSN for WPA2)
key_mgmt=WPA-PSK
pairwise=TKIP (or CCMP)
group=TKIP (or CCMP)
psk="key"
}
```

Quadro 2: Ativação da interface serial

Equipar o Raspberry Pi com uma interface serial requer algumas medidas prévias: os sinais TXD e RXD do UART do SoC são enviados para os pinos 8 e 10 do cabeçalho P1 situado entre a saída de TV e o slot para cartão SD (**figura 1**). No entanto, os sinais ainda estão no nível I/O do SoC com 3.3 volts. Com um deslocador de nível (level shifter), podemos movê-los para a faixa serial padrão RS232. Para isso usamos, por exemplo, o MAX3232CP IC.

O `minicom` pode ser utilizado como um programa de terminal no PC. Com o comando `minicom -s`, é possível fazermos mudanças mínimas nas configurações necessárias. Por exemplo, a porta serial pode ser representado tanto pelo arquivo de dispositivo `/dev/ttySnumber` como `/dev/tty-USBnumber`.

A taxa de dados (115.200 bps) e o formato de dados (8N1) são predefinidos. Certifique-se de que nem o Hardware Flow Control (controle de fluxo de hardware) ou o Software Flow Control (software de controle de fluxo) sejam usados pelo software. O comando `minicom -o` impede que o programa execute as sequências para inicializar no startup e conecta o usuário diretamente ao console.

acima não é um software de código aberto, e é necessário adquirir licenças para os formatos MPEG-2 e VC-1.

O consumo de energia, por outro lado, é surpreendentemente baixo: a carga média durante a renderização do vídeo em HD era de apenas 3.8 watts, apesar da árdua tarefa de transmissão através da rede. Os programas `Omxplayer`, `SSH` e `SSHFS` criam a maior carga para CPU e memória, enquanto o player reivindica a maior parte para si. No entanto, se o usuário quiser reproduzir vídeo em HD com 1080i ou 1080p, vale a pena conectar o drive de rede via NFS.

Sistemas embarcados

O segundo e terceiro projetos empregam o Raspberry Pi como um sistema embarcado em uma rede – sem um teclado, mouse ou monitor, mas com acesso LAN/WLAN. Neste tipo de sistema, podemos ler as mensagens iniciadas com um terminal na porta serial. Se ocorrer um erro na configuração de rede, não será possível acessar o sistema através do SSH, logo, conectar-se diretamente através de um monitor e teclado é a única maneira de logar no sistema e corrigir o erro. Por esse motivo, o usuário faria bem em ativar a interface serial (**quadro 2**). O acesso com SSH através da rede deve ser sempre a primeira escolha; o acesso via interface serial deve ser

usado apenas como uma porta de fundo para emergências.

O próximo passo na construção do sistema embarcado é a configuração de uma conexão de rede; neste exemplo, vamos conectar através de uma WLAN. A instalação padrão do Raspbian já contém os pacotes de firmware dos sticks WLAN populares, bem como do programa cliente `wpa_supplicant`. Para ver se a stick WLAN está ativa, conecte o hardware e pesquise a reação correspondente do kernel no `syslog`.

Agora, a única coisa que resta a fazer é configurar o `wpa_supplicant` para um ponto de acesso. Em uma rede com um servidor DHCP no roteador, encontraremos os dados essenciais no relatório fornecido pelo comando `iwlist scanning`. O usuário modifica, então, a configuração do arquivo `/etc/wpa_supplicant/wpa_supplicant.conf`, como mostrado na **listagem 2**, e `/etc/network/interfaces`, como mostrado na **listagem 3**. Finalmente, inicie a conexão com `sudo ifup wlan0`.

Projeto 2: ponto de acesso

O Raspberry Pi também pode servir como um ponto de acesso (**figura 2**). Neste caso, ele simplesmente envia tudo o que o stick WLAN recebe diretamente à porta LAN, como uma espécie de ponte. É por isso que esta

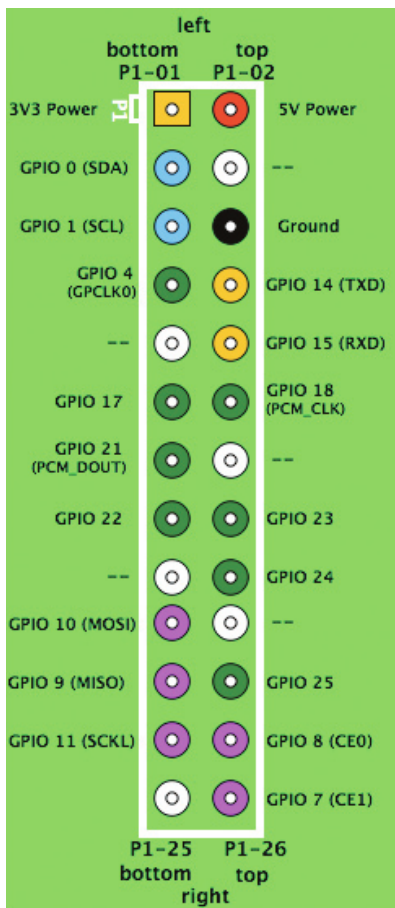


Figura 1 Cabeçalho para o qual a placa de circuito envia os sinais TXD e RXD de uma interface serial.

configuração é chamada de “modo bridge” (*bridged mode*). Desta forma, podemos aumentar o alcance de uma rede sem fio existente.

A configuração é realizada em duas etapas. Na primeira, devemos configurar o acesso direto à WLAN com o daemon `hostapd`. No entanto, note que o `hostapd` não é totalmente compatível com todas as WLAN sticks. Procure na Internet por uma lista de hardwares suportados [9]. Na segunda etapa, precisamos configurar

Listing 3: Como modificar `/etc/network/interfaces`

```
auto wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/
wpa_supplicant.conf
```

o encaminhamento de pacotes entre as duas interfaces.

Os pacotes necessários são instalados com:

```
$ sudo apt-get install
hostapd bridge-utils
```

Configure o ponto de acesso à WLAN em três etapas: primeiro, insira a seguinte linha no `/etc/default/hostapd`:

```
DAEMON_CONF=/etc/hostapd/hostapd.
conf
```

Em seguida, adicione os conteúdos da **listagem 4** para o arquivo `/etc/hostapd/hostapd.conf`, modificando-o para se ajustar ao ambiente local, se necessário. Um exemplo de configuração para o arquivo `/etc/network/interfaces` é encontrado na **listagem 5**. A **listagem 6** mostra como iniciar o ponto de acesso e a ponte de rede (*network bridge*).

O consumo de energia é menor com a operação LAN. Dependendo da distância entre as estações WLAN e do volume de dados transferidos, o consumo de energia para um cliente WLAN ou ponto de acesso flutua entre 3,3 e 3,8 watts, enquanto que em funcionamento de rede local (LAN),

Listagem 4: Adicionar ao `hostapd.conf`

```
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
driver=nl80211 # according to
driver support
macaddr_acl=0
auth_algs=3
ignore_broadcast_ssid=0
wpa=2 # for WPA2 with PSK
wpa_key_mgmt=WPA-PSK
rsn_preauth=1
rsn_preauth_interfaces=wlan0
rsn_pairwise=CCMP
wpa_passphrase=password
interface=wlan0
hw_mode=g
channel=11 # a free or weak
channel
ssid=(E)SSID
```

o consumo foi de cerca de 2,9 watts. Se ambas as interfaces fossem ativas, o valor estaria entre 3,5 e 4 watts.

Projeto 3: controlador de temperatura

Além de servir como um ponto de acesso flexível, o Raspberry Pi também é excelente para o controle de processos simples. O exemplo a seguir utiliza medições de tempe-

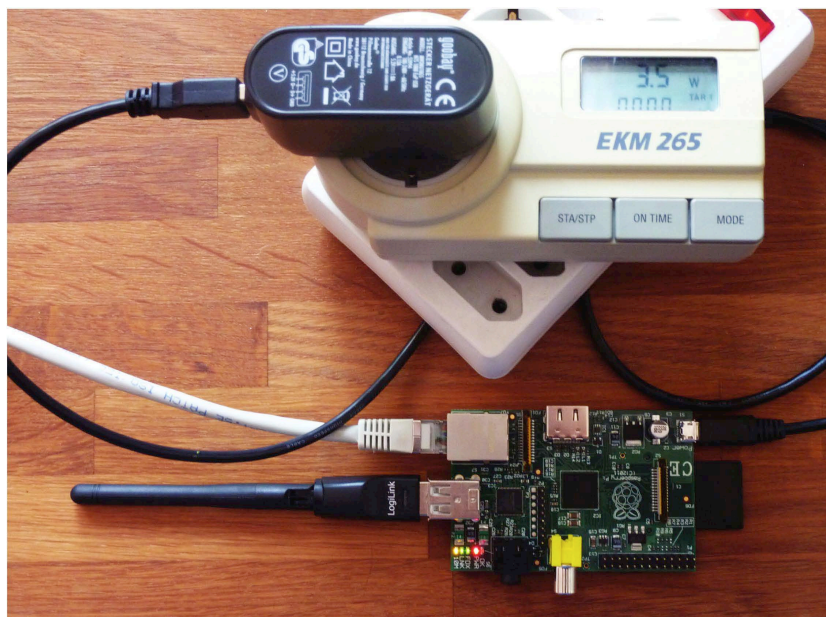


Figura 2 Raspberry Pi operando como um ponto de acesso - com um consumo de energia de apenas 3.5 Watts.

Listagem 5: Interfaces de rede

```
auto wlan0
iface wlan0 inet static
address 192.168.128.1
netmask 255.255.255.0
broadcast 192.168.128.255
```

Listagem 6: Iniciar ponto de acesso

```
$ sudo service hostapd start
$ sudo brctl addbr myBridge
$ sudo brctl addif myBridge eth0
$ sudo brctl addif myBridge wlan0
$ sudo ifconfig myBridge up
```

ratura cíclicas de um ambiente e, dependendo do valor, sinaliza os resultados visualmente com dois LEDs. Em temperaturas acima de 25 graus Celsius, o controlador aciona o LED vermelho; a menos de 22 graus Celsius, aciona o LED verde. Ambas as luzes são acionadas a valores entre estes dois extremos (figura 3).

Este projeto, é claro, destina-se a um experimento de prova de conceito. Em uma implementação do mundo real, o controlador pode conduzir a alguma outra ação, como iniciar um aquecedor ou fechar as cortinas, em vez de simplesmente acionar um LED. A unidade de temperatura do controlador requer um hardware adicional ocupando uma placa de expansão separada.

Vamos precisar das seguintes peças: o LM75 IC como sensor de temperatura e o MAX-3232CPE IC, incluindo quatro capacitores eletrolíticos 0.47µF, como transceptores para a interface serial (UART), juntamente com um conector de nove pinos D-sub ou um soquete para montagem PCB e um cabo de interface. Além disso, precisaremos também de dois LEDs, vermelho e verde, cada um com um resistor em série de 22Ω e um transistor NPN (por exemplo, o

Listagem 7: Pacotes Python e bibliotecas

```
$ sudo apt-get install
git python3 python-pip python-
virtualenv
$ git clone
https://github.com/quick2wire/quick2wire-gpio-admin.git
The red and green
$ git clone
https://github.com/quick2wire/quick2wire-python-api.git
$ cd quick2wire-gpio-admin
$ make
$ sudo make install
$ sudo adduser pi gpio
$ virtualenv TEMPCONTROL
$ source TEMPCONTROL/bin/activate
$ cd quick2wire-python-api
$ sudo python3 setup.py install
```

BC237) como driver, bem como duas resistências de 10kΩ para o barramento I2C. Materiais diversos, como uma placa de montagem (*breadboard*), filtro de linha, cabo de fita plana (*flat-ribbon cable*) e capacitores de desacoplamento servirão como base.

O popular módulo LM75 serve como sensor de temperatura e é conectado ao barramento I2C. A máquina controla os dois LEDs através de dois pinos GPIO e uma fase de transistor. A fiação é simples – podemos usar um modelo que encontrarmos na Internet.

Na verdade, o SoC possui duas interfaces mestres no barramento I2C, cuja saída é também encontrada no syslog, mas somente uma das duas está disponível no cabeçalho de linha dupla (P1-03 e P1-05) – a que possui o barramento ID 0 (bus-ID 0). Podemos ativar o barramento I2C instalando os dois drivers *i2c-bcm2708* e *i2c-dev*. Para fazer isso, remova o driver *i2c-bcm2708* da lista de drivers que não devem ser carregados em */etc/modprobe.d/raspi-blacklist.conf*. Podemos também especificar o driver *i2c-dev*, que cria a interface do dispositivo no diretório */dev*, em */etc/modules*.

Listagem 8: Programa do controlador

```
01 #!/usr/bin/env python3
02 import quick2wire.i2c as i2c
03 import time
04 from quick2wire.gpio import
Pin, exported
05
06 busaddr = 0x48
07 regaddr = 0x00
08 redLED = Pin(16, Pin.Out)
09 greenLED = Pin(18, Pin.Out)
10
11 with exported(Pin(16, Pin.
Out)) as redLED, \
12 exported(Pin(18, Pin.Out))
as greenLED, \
13 i2c.I2CMaster() as bus:
14 while True:
15 temp_hib, temp_lob = bus.
transaction(
i2c.writing_bytes(busaddr,
regaddr),
i2c.reading(busaddress, 2))[0]
16 temp = (temp_hib << 1 |
temp_lob >> 7) / 2.
17 if temp > 25:
18 redLED.value = 1
19 greenLED.value = 0
20 elif temp < 22:
21 redLED.value = 0
22 greenLED.value = 1
23 else:
24 redLED.value = 1
25 greenLED.value = 1
26 print ("%02.01f" % temp)
27 redLED.value = 0
28 greenLED.value = 0
29 time.sleep(5)
```

O barramento estará ativo uma vez que o sistema for reiniciado. Para ativá-lo sem reiniciar, execute o comando `modprobe i2c-bcm2708 i2c-dev`.

Para realizar operações no barramento I2C como um usuário normal, primeiro precisamos instalar algumas ferramentas correspondentes e definir as permissões de grupo:

```
$ sudo apt-get install i2c-tools
$ sudo addgroup pi i2c
```

Os endereços onde os módulos se encontram é revelado pelos valores encontrados na saída retornada por

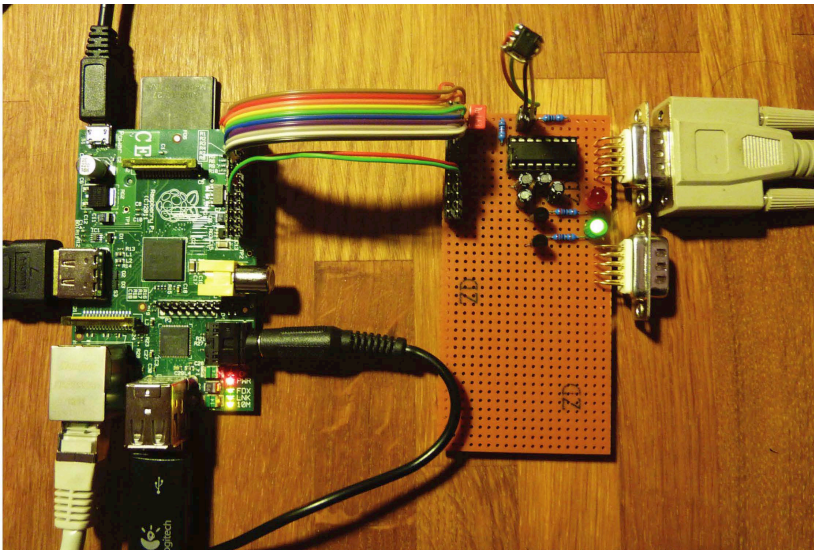


Figura 3 O Raspberry Pi com uma placa de expansão como um monitor de temperatura. A luz verde na placa de expansão confirma que a temperatura está abaixo de 22 graus Celsius.

```
i2cdetect -y 0
```

Dependendo de como foram cabeados os três pinos do LM75, a saída irá conter exatamente um valor na faixa hexadecimal de 0x48 a 0x4F. O usuário, então, irá ler a palavra de temperatura de 2 bytes do LM75 no registro de endereço 0 com o comando:

```
$ i2cget -y 0 I2C bus_address  
register_address w
```

A ficha de dados do módulo será uma ajuda para a conversão do valor críptico retornado (exemplo: 0x8015) para a temperatura correspondente. Python é um ambiente de software adequado para o controlador, desde que as bibliotecas de classe já existam para o Raspberry Pi para acessar o barramento I2C e os pinos GPIO. Este exemplo usou duas bibliotecas [10] [11]. A **listagem 7** mostra a instalação dos pacotes e bibliotecas. O programa em si é muito simples (**listagem 8**): ele lê o valor do sensor (**linha 15**) em um ciclo de cinco segundos e converte a leitura do sensor para o valor da temperatura (**linha 16**). O programa então aciona os dois LEDs para ligado ou desligado, de acordo com a intenção do usuário.

As diferentes designações para os pinos GPIO são um pouco confu-

sas. Os LEDs vermelho e verde são GPIO23/GPIO24, de acordo com o cabeçalho, mas também estão conectados ao P1-16/P1-18. A primeira denominação corresponde à documentação BCM2835 e aos valores do kernel. No entanto, as bibliotecas Python seguem a segunda nomenclatura.

O usuário terá que fornecer o script com as permissões necessárias para execução. Após a inicialização, ele irá continuamente dispor os valores medidos no console a cada atualização dos LEDs.

Conclusão

O Raspberry Pi não é, obviamente, um sistema desktop. Se utilizar o terminal parecer algo muito estranho e enigmático para o usuário, comece no desktop e trabalhe o caminho para a linha de comando de forma gradual. Desde o princípio, o RasPi foi concebido como um PC com eficiência energética para aplicativos multimídia, e é claramente destinado para uso como um sistema embarcado. ■

Gostou do artigo?

Queremos ouvir sua opinião.
Fale conosco em
cartas@linuxmagazine.com.br
Este artigo no nosso site:
<http://lnm.com.br/article/8388>

Mais informações

- [1] Raspberry Pi: <http://www.raspberrypi.org/>
- [2] Guia de compra de hardware: http://elinux.org/RPi_Hardware_Basic_Setup_#Typical_Hardware_You_Will_Need
- [3] Mais ajuda ao escolher hardware: http://elinux.org/RPi_VerifiedPeripherals
- [4] Farnell element14 (Europa) / Newark element14 (América do Norte): <http://uk.farnell.com> <http://www.newark.com>
- [5] RS Components (Inglaterra) / Allied Electronics (América do Norte): <http://uk.rs-online.com> <http://www.alliedelec.com>
- [6] Para respostas a questões: <http://elinux.org/RaspberryPiBoard>
- [7] Raspbian "Wheezy": <http://www.raspbian.org/>
- [8] Reprodutor de áudio/vídeo Omxplayer: <https://github.com/huceke/omxplayer/>
- [9] Hostapd: <http://hostap.epitest.fi/hostapd/>
- [10] API Quick2Wire Python: <https://github.com/quick2wire/quick2wire-python-api>
- [11] GPIO Admin: <https://github.com/quick2wire/quick2wire-gpio-admin>