

Přehrávač MP3

V tomto projektu vám ukážu, jak vytvořit docela jednoduchý, avšak plně funkční audiopřehrávač MP3. Popíšu také, jak použít vývody GPIO počítače RasPi. Umožní používat v přehrávači hardwarová tlačítka k výběru skladeb MP3, které se mají přehrát.

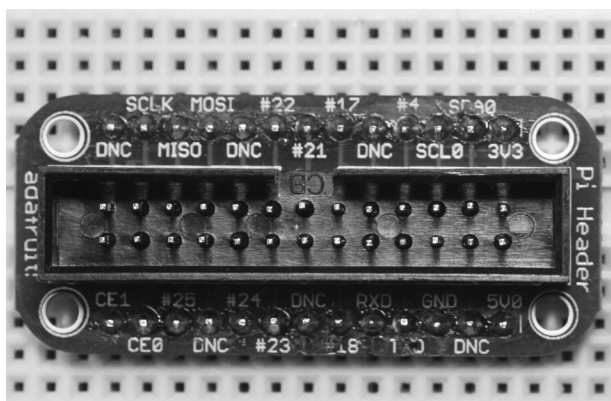
Podíváme se také na softwarový balíček ALSA (zkratka pro *Advanced Linux Sound Architecture*), který v Linuxu vyvádí zvuk a poskytuje některé funkce, které potřebujeme pro uskutečnění tohoto projektu.

V této kapitole:

- Prototypový konektor
- Přenosný stereo-reproduktor
- ALSA
- Formáty zvukových souborů
- Vstupy hardwarových spínačů
- Mějte na mysli možnosti RasPi
- Požadavky projektu MP3

Prototypový konektor

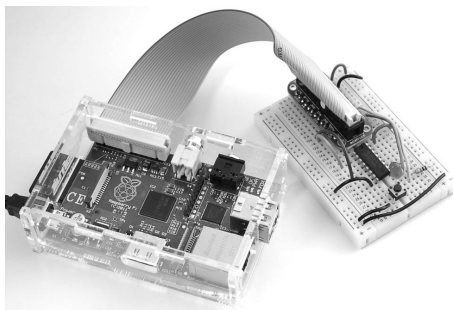
Zaměříme se nejprve na hardwarový prostředek, který budu používat k propojení hardwarových spínačů s vývody GPIO počítače RasPi. Tato prototypová pomůcka se liší od prototypové desky Pi Plate popsané v předchozí kapitole. Konektor se nazývá Pi Cobbler a je na obrázku 3.1 zobrazen bez konektoru plochého propojovacího kabelu, aby bylo dobře vidět jeho vývody.



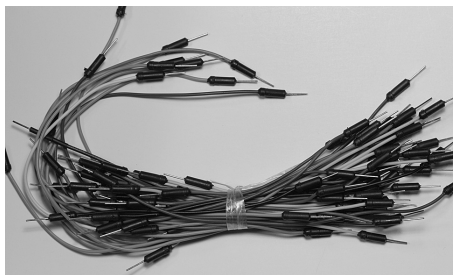
Obrázek 3.1 Prototypový konektor Pi Cobbler

Pi Cobbler je vlastně přímým prodloužením konektoru GPIO umožňujícím také jeho odpojení. Jeho vývody je možno připojovat na propojovací prototypovou desku (nazývanou breadboard čili „kuchyňské prkénko“) bez pájení. Takovou propojovací desku nabízí jako sadu řada výrobců, například firma Adafruit na webové adrese www.adafruit.com/products/914. Kompletně smontovaný Pi Cobbler nainstalovaný na propojovací desce a propojený s počítačem RasPi je zobrazen na obrázku 3.2.

Použití konektoru Pi Cobbler a propojovací desky umožňuje rychlé sestavení projektu a snadné provádění modifikací u již existujícího projektu. Bude užitečné, když budete mít připraveny různé, předem odizolované dráty vhodné pro použití na propojovací desce. Můžete si je vyrobit sami z pevného drátu velikosti 22 nebo si je koupit jako sadu u svého dodavatele elektronických součástek. Obrázek 3.3 ukazuje sadu, kterou jsem zakoupil pro projekty v této knize. Tyto vodiče jsou tvořeny vícežilovým lankem, aby byly ohebné, a jsou zakončeny pevným, izolovaným spojovacím kolíkem. Zakoupit takovou sadu propojovacích vodičů se opravdu vyplatí.



Obrázek 3.2 Konektor Pi Cobbler na propojovací desce připojený k počítači Raspberry Pi



Obrázek 3.3 Kvalitní propojovací vodiče pro spoje propojovací desky

Přenosný stereoreproduktor

Abyste mohli poslouchat skladby MP3, potřebujete nějaké zařízení pro přehrávání. Já používám velmi malý a nenákladný stereoreproduktor Veho360 (www.veho-uk.com), zobrazený na obrázku 3.4.

Reproduktor má konektor typu jack 3,5 mm, který lze zastrčit do audiovýstupu RasPi. Má také nabíjecí baterii, kterou lze dobít zasunutím do standardního konektoru USB. Nedoporučuji však používat k tomu konektory USB RasPi kvůli proudovým omezením, o kterých jsme hovořili v kapitole 1. Ovladač hlasitosti



Obrázek 3.4 Reproductor Veho360

na reproduktoru umožňuje nastavovat hlasitost pouze ve dvou krocích. Celkem vzato je tento napájený reproduktor docela dobrý pro reprodukci zvuku, ale svého domácího kina se kvůli němu zbavovat nebudu.

ALSA

ALSA (Advanced Linux Sound Architecture) je základní ovladač zvuku používaný ve většině distribucí Linuxu pro RasPi. Jedná se o velmi stabilní software široce používaný v různých distribucích Linuxu.

Je to zcela otevřený zdroj a je silně podporován na www.alsa-project.org. Osm softwarových balíčků, které v současné době tvoří ovladač ALSA, je i se stručným popisem uvedeno v tabulce 3.1.

Tabulka 3.1 Softwarové balíčky ALSA

Název balíčku	Popis
alsa-driver	Ovladače nízké úrovně propojující hardware s jádrem systému Linux
alsa-lib	API pro ovladače v jazyce C
alsa-utils	Sada velmi užitečných utilit (programových nástrojů)
alsa-tools	Pokročilá sada nástrojů používaná s určitými zvukovými kartami (ne s RasPi)
alsa-firmware	Firmware, který je třeba nainstalovat na určitých zvukových kartách (ne na RasPi)
alsa-plugins	Používány audioknihovny a zvukovými servery
alsa-oss	Otevřený zvukový systém, starší verze poskytovaná z důvodu kompatibility
pyalsa	Modul jazyka Python pro ovladač ALSA používaný se systémem pro videoúpravy

Moduly ALSA by měly být součástí distribuce Linuxu, ale měli byste se přesvědčit, zda jsou opravdu na místě, zadáním následujícího příkazu v terminálovém okně:

```
sudo modprobe snd-bcm2835 ↵
```

Aplikace modprobe načítá moduly LKM (*Loadable Kernel Modules*), což je v tomto případě zvukový ovladač snd-bcm2835 vhodný pro ALSA. Měli byste také zkontrolovat, zda soubor s názvem *modules* umístěný v adresáři */etc* obsahuje položku snd-bcm2835. Při zavádění systému se kontroluje, zda tento soubor obsahuje všechny LKM, které je třeba načíst.

Potřebujete nainstalovat také sadu alsa-utils, protože obsahuje několik aplikací potřebných pro funkci našeho projektu. Pro instalaci se používá příkaz apt-get, kterým jsme se zabývali v kapitole 1. Zadejte v terminálovém okně následující příkaz:

```
sudo apt-get install alsa utils ↵
```

Abychom se přesvědčili, že se na audiokonektoru vytváří zvukový signál, musíme provést ještě jeden krok. Zadejte v terminálovém okně následující příkaz:

```
sudo amixer cset numid=3 1 ↵
```

Aplikace *amixer* je součástí sady *alsa-utils* umožňující vybrat požadovaný konektor audiovýstupu. Po prvním zavedení systému do RasPi je výchozím audiovýstupem HDMI. Audiovýstup lze přeměrovat pomocí výše uvedeného příkazu. Číslo na konci příkazového řádku odpovídá následujícím výstupům:

- 0 – auto
- 1 – analogový
- 2 – HDMI

Testování analogového audiovýstupu

Nyní nastal čas pro otestování audiovýstupu RasPi. Připojte aktivní reproduktor ke konektoru typu jack 3,5 mm a zadejte v terminálovém okně následující příkaz:

```
sudo speaker-test -c 2
```

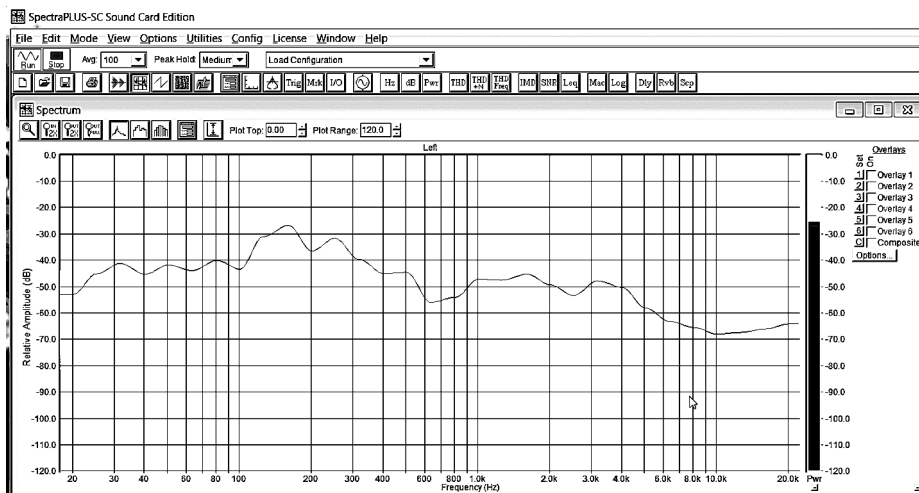
Měli byste z reproduktoru slyšet *hučivý* zvuk. Je to tzv. růžový šum vytvářený aplikací pro testování reproduktorů, která je součástí sady ALSA. (Podrobnější vysvětlení růžového šumu najdete v následujícím oddíle Frekvenční závislost analogového audiovýstupu.) Pokud tento zvuk neslyšíte, ověřte výše uvedené příkazy a ujistěte se, že byly zadány správně. Aplikace pro testování reproduktoru má řadu dalších možností, než jen vyrábět šum. Další informace najdete na <http://manpages.ubuntu.com/manpages/natty/man1/speaker-test.1.html>.

Frekvenční závislost analogového audiovýstupu

Máme-li k dispozici výstup růžového šumu, nabízí se příležitost změřit frekvenční závislost analogového audiosystému RasPi. Pracoval jsem mnoho let jako akustický technik, proto je pro mě přirozené využít této příležitosti k zhodnocení akustických vlastností určitého pod-systému. Nejprve se trochu zabývejme základními informacemi týkajícími se růžového šumu a důvody, proč je užitečný pro zjišťování frekvenční závislosti. Šum je ve své podstatě náhodný, jeho energie je rovnoměrně rozdělena v celém frekvenčním rozsahu čili šířce pásma. Tomuto typu šumu se také říká *bílý šum*, čímž se vyjadřuje rovnoměrnost rozdělení energie.

Při provádění frekvenční analýzy akustických systémů se obvykle používá řada filtrů, jimž se říká filtry třetinových oktáv, které do jisté míry simulují citlivost lidského ucha. To znamená, že filtry jsou užší při nízkých frekvencích a rozšiřují se při rostoucí frekvenci. Přivedení bílého šumu na takový filtr má za následek rostoucí průběh. Bílý šum se předem vyfiltruje (nebo předem zkreslí, chcete-li) tak, že upravený signál šumu, který vychází z filtrů třetinové oktávy, má rovný (konstantní) průběh. Jakákoli odchylka od tohoto průběhu je tedy způsobena testovaným systémem, nikoli přivedeným signálem šumu. Tento předem upravený signál šumu se nazývá *růžový šum*.

Obrázek 3.5 je snímek obrazovky programu akustické analýzy měřícího analogový audiovýstup vyrábějící růžový šum z aplikace pro testování reproduktorů.



Obrazek 3.5 Akustická analýza počítače Raspberry Pi (třetinové oktávové měření).

Průběh audiovýstupu zdaleka není rovný, jak vidíte na nepravidelné křivce. Dobrá zpráva je, že nepravidelnost není tak velká, aby audiovýstup nějak významně zkreslila. Ti, kteří si potrpí na dokonalý zvuk, pravděpodobně použijí ke kompenzaci odchylek průběhu vícepásmový ekvalizér.

Formáty zvukových souborů

Stručný popis formátů zvukových souborů vám pomůže porozumět kladům a záporům jednotlivých formátů při vytváření audioaplikací. Existují tři základní kategorie formátů zvukových souborů:

- Nekomprimované – nekomprimovaný soubor využívá maximum paměti.
- Bezeztrátová komprese – přeskakuje tiché úseky, ukládá audioobsah.
- Ztrátová komprese – komprimuje vše; zavádí určité zkreslení.

V tabulce 3.2 je uveden seznam reprezentativních formátů každé z těchto kategorií.

Tabulka 3.2 Formáty audiosouborů

Název	Typ komprese	Kompresní poměr	Poznámka
WAV	nekomprimovaný	1:1	Též nazývaný PCM
AIFF	nekomprimovaný	1:1	
AU	nekomprimovaný	1:1	
FLAC	bezeztrátová komprese	1:1	
WMA Lossless	bezeztrátová komprese	2:1	Windows Media Audio
M4A	bezeztrátová komprese	2:1	Apple – bezeztrátový formát

Název	Typ komprese	Kompresní poměr	Poznámka
MP3	ztrátová komprese	10:1 až 20:1	
Vorbis	ztrátová komprese	10:1 až 20:1	
ACC	ztrátová komprese	10:1 až 20:1	
WMA Lossy	ztrátová komprese	10:1 až 20:1	

Porovnání formátů WAV a MP3

Budeme se nyní zabývat těmito formáty, protože náš projekt používá formát MP3. Zahrnul jsem však do výkladu i jednoduchou ukázkou RasPi přehrávajícího formát WAV (Waveform Audio File Format). Formát WAV poskytuje nejvěrnější reprodukci původního zvukového obsahu, jaká je možná při použití digitální techniky. Zdrojový zvukový obsah se vzorkuje s vysokou vzorkovací frekvencí s použitím dostatečně velkého počtu bitů pro přesný záznam amplitudy digitálního vzorku. Vysoká vzorkovací frekvence a vysoká přesnost záznamu znamenají velké nároky na paměť. Například třiminutový záznam hudby symfonického orchestru může zabrat 30 MB paměti. Naproti tomu stejný obsah ve formátu MP3 může vystačit se 2 MB paměti, což znamená kompresní poměr 1:15. Pro většinu lidí je formát MP3 přijatelný, zvláště když se přehrává na zařízení, jako je chytrý telefon nebo tablet. Ani reprodukce zvuku pomocí RasPi není nejdokonalější, jak jsme viděli výše.

Přehrávání souboru WAV

Přehrání souboru WAV je docela snadné, použijeme-li další aplikaci obsaženou v sadě nástrojů alsa. Tato aplikace se jmenuje aplay a nám stačí vyvolat tuto aplikaci s názvem souboru WAV, který se má přehrát. Na webovou stránku této knihy, www.mhprofessional.com/raspi, jsem umístil volně dostupné soubory WAV a MP3 nazvané class.wav a class.mp3. Stáhněte si je prosím do adresáře pi. Soubor class.mp3 budeme používat v následujícím oddíle.

Nyní otevřete terminálové okno a napište:

```
sudo aplay class.wav ↵
```

Ujistěte se, že váš aktivní reproduktor je zapojen do analogového audiokonektoru (jack) a že je tento konektor vybrán příkazovým řádkem `amixer`. Měli byste uslyšet nádherné tóny Bachovy fugy. Hudební ukáзка trvá asi 2,5 minuty. Lze ji kdykoli ukončit stisknutím `^c`.

Verze MP3 téže hudební ukázky má velikost asi 1,2 MB, na rozdíl od verze WAV, která zabírá více než 25 MB. Příležitost přehrát verzi MP3 budete mít v následujícím oddíle, až vás uvedu do softwaru přehrávače MP3. To vám poskytne skvělou příležitost porovnat kvalitu zvuku nekomprimovaného souboru WAV a souboru MP3 komprimovaného se ztrátou.

Přehrávání souboru MP3

Pro přehrávání souborů MP3 je k dispozici řada aplikací kompatibilních s Linuxem. Já jsem zvolil soubor `mpg123`, protože je velmi stabilní a dobře se ovládá z příkazového řádku, což

pro náš přehrávač MP3 pro počítač RasPi potřebujeme. Načtete tuto aplikaci následujícím příkazem `apt-get`:

```
sudo apt-get install mpg123 ↵
```



Poznámka: Na webu je k dispozici klon `mpg123` nazvaný `mpg321`. Jsem si jist, že funguje docela dobře, ale dávám raději přednost originálu.

Nyní stažený soubor MP3 přehrajte příkazem:

```
sudo mpg123 class.mp3 ↵
```

Soubor MP3 se bude přehrávat přesně stejně dlouhou dobu jako soubor WAV. Přehrávání lze také kdykoli zastavit stisknutím `^c`, stejně jako v případě aplikace `aplay`.

```

pi@raspberrypi: ~
File Edit Tabs Help
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
version 1.14.4; written and copyright by Michael Hipp and others
free software (LGPL/GPL) without any warranty but with best wishes
Decoder: ARM

Terminal control enabled, press 'h' for listing of keys and functions.
Playing MPEG stream 1 of 5: class1.mp3 ...

MPEG 2.0, Layer: III, Freq: 22050, mode: Joint-Stereo, modext: 2, BPF : 209
Channels: 2, copyright: No, original: Yes, CRC: No, emphasis: 0.
Bitrate: 64 kbit/s Extension value: 0
Title:  onclassical_demo_demicheli_geminiani_pieces_allegro-in-f-major_small-ve
rsion
Artist:
Comment:                                     Album:
Year:                                         Genre:  Unknown
Frame# 1040 [ 2969], Time: 00:27.16 [01:17.55], RVA:  off, Vol: 100(100)

.-= terminal control keys =-
[s] or [ ] interrupt/restart playback (i.e. '(un)pause')
{f} next track
{d} previous track
{b} back to beginning of track
{p} loop around current position (don't combine with output buffer)
[.] forward
[,] rewind
[:] fast forward
[;] fast rewind
[>] fine forward
[<] fine rewind
[+] volume up
[-] volume down
[r] RVA switch
[v] verbose switch
[l] list current playlist, indicating current track there
[t] display tag info (again)
[m] print MPEG header info (again)
[h] this help
[q] quit
[c] or [C] pitch up (small step, big step)
[x] or [X] pitch down (small step, big step)
[w] reset pitch to zero

Frame# 4008 [ 1], Time: 01:44.69 [00:00.02], RVA:  off, Vol: 114(114)
[1:44] Decoding of class1.mp3 finished.

Terminal control enabled, press 'h' for listing of keys and functions.
Playing MPEG stream 2 of 5: class.mp3 ...
Title:  onclassical_demo_latry_bach_fugue_in_d_major_bwv-532_live_cut-version
Artist:
Comment:                                     Album:
Year:                                         Genre:  Unknown
Frame# 5891 [ 1], Time: 02:33.88 [00:00.02], RVA:  off, Vol: 114(114)
[2:33] Decoding of class.mp3 finished.

```

Obrázek 3.6 Seznam terminálových ovládacích kláves aplikace `mpg123`

Já osobně jsem nebyl schopen sluchem zjistit jakýkoli znatelný rozdíl mezi těmito dvěma formáty, vy jej však možná zjistíte. Může být také docela zábavné pozvat přátele a členy rodiny a vyzkoušet, zda někdo z nich zjistí rozdíl mezi oběma audioformáty.

Aplikaci mpg123 se někdy říká „konzolový přehrávač“, protože ji můžete ovlivňovat pomocí klávesnice a pozorovat výstup aplikace v terminálovém okně. To je přesně to, co potřebujeme k vytvoření přehrávače MP3 ovládaného počítačem RasPi. Je důležité vědět, jaké terminálové ovládací prvky jsou k dispozici a jaké jsou jejich funkce. Toho lze snadno docílit stisknutím klávesy h při chodu aplikace. Na obrázku 3.6 je vidět výsledek stisknutí této klávesy při přehrávání MP3.

V tabulce 3.3 je uveden snadno čitelný seznam terminálových ovládacích kláves s doplňkovým vysvětlením.

Mimochodem, je velmi snadné přehrát celý adresář souborů MP3 následujícím příkazem:

```
sudo -vC *.mp3 ↵
```

Znak v v příkazu zavede režim s komentářem (verbose) a znak C uvede aplikaci do režimu úplné kontroly, kdy jsou funkční všechny ovládací klávesy. Znak * v příkazovém řádku je zástupný znak příkazující aplikaci přehrát všechny soubory MP3, které nalezne v aktuálním adresáři. Je-li všechno řečeno a provedeno, je to velmi výkonná, ale přesto jednoduchá aplikace přehrávače MP3, která velmi dobře poslouží našim účelům. Nyní již potřebujeme jen nějaké spínače pro výběr skladeb a trochu programového kódu v jazyce Python, abychom to vše zautomatizovali.

Tabulka 3.3 Terminálové ovládací klávesy mpg123

Terminálová ovládací klávesa	Akce
s	Přepínač přestávka (stisknutím se zastaví přehrávání, dalším stisknutím se obnoví)
f	Následující skladba
d	Předchozí skladba
b	Zpět na začátek skladby
p	Smyčka kolem aktuální polohy
.	Vpřed (Forward)
,	Převinout zpět (Rewind)
:	Rychle vpřed (Fast forward)
;	Rychle převinout zpět (Fast rewind)
+	Zvětšit hlasitost (Volume up)
-	Zmenšit hlasitost (Volume down)
r	Spínač Relativní nastavení hlasitosti (RVA) – používá příznak MPEG ID3V2 k nastavení hlasitosti alba
v	Spínač režimu s komentářem (verbose)
l	Vypsát aktuální seznam stop (playlist)
t	Vypsát informaci příznaku (tag)

Terminálová ovládací klávesa	Akce
m	Vytisknout záhlaví MPEG
h	Seznam nápovědy (help list)
q	Ukončit (quit)
c	Malá změna k vyšším tónům
C	Velká změna k vyšším tónům
x	Malá změna k hlubším tónům
X	Velká změna k hlubším tónům
w	Zrušit tónovou clonu

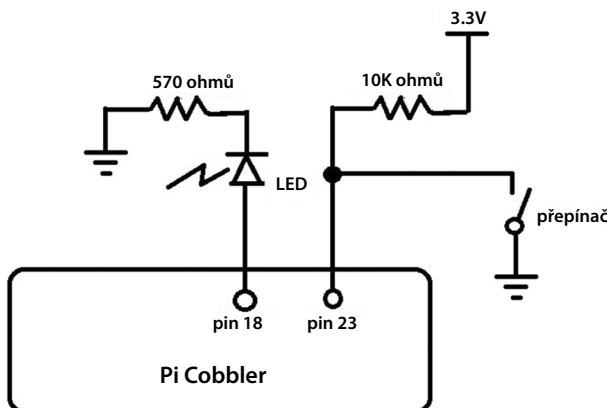
Vstupy hardwarových spínačů

Odpočínme si teď od softwarových záležitostí a zaměřme se na to, jak k počítači RasPi připojit hardwarové spínače a jak reagovat na stisknutí tlačítka uživatelem. Vzpomeňte si na kapitolu 2, kde jsme si řekli, že všechny vývody konektoru GPIO je možno nastavit buď na vstupní režim, nebo na výstupní režim. Pro blikáč LED jsme vývod 18 nastavili na výstup. V tomto oddíle nastavíme vývod 23 na vstup a vývod 18 ponecháme jako výstup. Cílem je, aby se LED připojená na výstup 18 rozsvítila, když se stiskne tlačítko připojené na vstup 23. Ke spouštění vývodu 23 budeme používat napájecí napětí 3,3 V. Pro jistotu zopakují varování uvedené v kapitole 1:



Upozornění: Všechny vývody GPIO snesou napětí 3,3 V, nikoli však 5 V. V RasPi není zabudována žádná přepětová ochrana. Jestliže neúmyslně připojíte vývod GPIO na 5 V, dojde ke zničení RasPi.

Na konektoru Pi Cobbler je přítomno napětí 5 V a není vůbec těžké připojit toto napájecí napětí místo 3,3 V. Proto buďte mimořádně opatrní a všechny spoje si dvakrát zkontrolujte, než zapnete napájení.



Obrázek 3.7 Testovací obvod pro vstup spínače (tlačítka)

Sestavte testovací obvod a připojte LED, tlačítko a příslušné odporníky podle schématu na obrázku 3.7. Použijte propojovací prototypovou desku bez pájení, na niž snadno umístíte konektor Pi Cobbler i ostatní součástky.

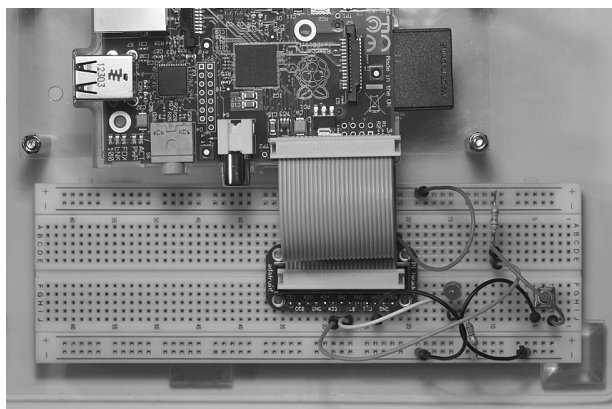
Níže je uveden programový kód v jazyce Python, který tento malý obvod spustí. Program je dostupný na webu této knížky jako soubor `Test_Switch.py`.



Poznámka: Pro příkaz porovnání se musí napsat dva znaky rovnítka za sebou, tj. `GPIO.input(23) == False`. Kdybyste jedno rovnítko vynechali, stal by se z toho příkaz přiřazení, který může a nemusí být vyhodnocen jako logický stav. Chybný příkaz `GPIO.input(23) = False` by se pravděpodobně vyhodnotil jako `False`, protože vstup nelze programově nastavit. V každém případě by program běžel, ale nefungoval by tak, jak očekáváme. Říká se tomu logická chyba a začínajícím programátorům se stává poměrně často.

`Test_Switch.py`

```
#Import knihovny GPIO pro přístup na vývody.
import RPi.GPIO as GPIO
#Použít číslování vývodů BCM.
GPIO.setmode(GPIO.BCM)
#Nastavit vývod 18 na výstup.
GPIO.setup(18, GPIO.OUT)
#Nastavit vývod 23 na vstup
GPIO.setup(23, GPIO.IN)
#Spustit "nekonečnou" smyčku.
while True:
    #Kontrola, zda je tlačítko stisknuto.
    if(GPIO.input(23) == False):
        #Nastavit vývod na výstup HIGH
        GPIO.output(18, GPIO.HIGH)
    #Pokud není stisknuto, nastavit výstup na LOW.
    else:
        GPIO.output(18, GPIO.LOW)
```



Obrázek 3.8 Skutečné provedení obvodu testování tlačítka

V tomto programu jsem použil podmíněný příkaz `if/else`, který kontroluje, je-li vstup 23 na úrovni LOW čili má logickou hodnotu `False`, a pokud je, následující příkaz nastaví výstupní vývod 18 na GPIO.HIGH. V opačném případě přejde podmíněný příkaz ihned na část `else` a vývod 18 nastaví na úroveň GPIO.LOW.

Spusťte program a pozorujte, že při každém stisknutí tlačítka se LED rozsvítí a zůstane svítit, dokud je tlačítko stisknuto. Obrázek 3.8 ukazuje skutečný obvod na propojovací desce s RasPi připojeným plochým kabelem na konektor Pi Cobbler.

Mějte na mysli možnosti RasPi

Při vytváření nového programu musíte vždy myslet na pracovní rychlost RasPi. Smyčka `while` v programu se opakuje více než milionkrát za sekundu, což znamená, že všechny příkazy ve smyčce se trvale opakují. To je důvod, proč jsem v programu použil podmíněný příkaz `if/else` k opakovanému nastavování hodnoty na výstupu na úroveň HIGH, aby tato úroveň souhlasila s dobou stisknutí tlačítka. Kdybych část `else` vynechal, LED by se při prvním stisknutí tlačítka rozsvítila, ale po uvolnění tlačítka by svítila dál. Zkuste část `else` v programu vynechat a ověřte, zda LED zůstane svítit. Ale proč zůstane rozsvícena?

Odpověď je jednoduchá: Bez části `else` by v programu nebyl žádný příkaz, který by LED vypnul. Vývod zůstává v tom stavu, do kterého byl příkazem uveden. Právě proto je třeba mít na mysli možnosti RasPi. Zjistil jsem, že pro nové (a zčásti i ne zcela nové) programátory mikroprocesorů může být aspekt reálného času při programování málo srozumitelný. Ovládání skutečného hardwaru může být náročné. Budu se snažit upozorňovat na tyto *náročnější* části programů, abych vám pomohl vyhnout se jejich nepochopení.

Požadavky projektu MP3

Prvním krokem každého projektu je specifikace požadavků, které musí být splněny, aby uspokojily uživatele projektu. V tomto případě jsou požadavky projektu dosti jednoduché. Jsou uvedeny zde:

- Použít aplikaci `mpg123` jako přehrávač MP3
- Přehrát vybrané soubory MP3 uložené v předem nastaveném adresáři
- Použít počítač ke sledování dat přehrávaného souboru
- Použít analogový audiokonektor (jack) 3,5 mm pro výstup
- Použít klávesnici k zadávání příkazů k ovládání `mpg123` při přehrávání skladeb
- Použít jedno ze tří tlačítek k přehrání předem vybrané skladby

Postup uskutečnění tohoto projektu je jednoduchý, protože mnoho k tomu potřebných součástí a softwaru jsme již probrali.

Psaní programu

Klíčovou záležitostí je spustit aplikaci mpg123 z programu v jazyce Python se všemi potřebnými argumenty. Python naštěstí poskytuje funkci nazvanou `system()`, která to umožňuje provést docela snadno. A je součástí sady OS. Následující program v jazyce Python, který jsem vytvořil, je pojmenován `MP3_Player.py` a je k dispozici ke stažení z webové stránky této knihy.

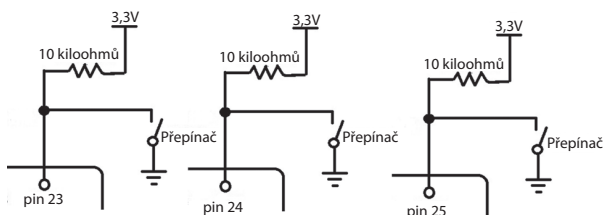
MP3_Player.py

```
#Import sady os, protože obsahuje volání funkce system().
import os
#Import knihovny GPIO pro přístup k vývodům.
import Rpi.GPIO as GPIO
#Použit číslování vývodů BCM.
GPIO.setmode(GPIO.BCM)
#Nastavit vývod 23 jako vstup.
GPIO.setup(23, GPIO.IN)
#Nastavit vývod 24 jako vstup.
GPIO.setup(24, GPIO.IN)
#Nastavit vývod 25 jako vstup.
GPIO.setup(25, GPIO.IN)
#Ověřit, že je vybrán analogový výstup.
os.system('amixer cset numid=3 1')
while True:
    #Zkontrolovat, zda je stisknuto tlačítko připojené na vývod 23.
    if(GPIO.input(23) == False):
        #Spustit aplikaci mpg123 s úplným ovládáním klávesnicí a výstupem
        #na obrazovku při přehrávání skladby class.mp3.
        os.system('mpg123 -vC class.mp3')
    #Zkontrolovat, zda je stisknuto tlačítko připojené na vývod 24.
    if(GPIO.input(24) == False):
        #Spustit aplikaci mpg123 s úplným ovládáním klávesnicí a výstupem
        #s komentářem při přehrávání skladby class1.mp3.
        os.system('mpg123 -vC class1.mp3')
    #Zkontrolovat, zda je stisknuto tlačítko připojené na vývod 25.
    if(GPIO.input(25) == False):
        #Spustit aplikaci mpg123 s úplným ovládáním klávesnicí a výstupem
        #s komentářem při přehrávání skladby Maid.mp3.
        os.system('mpg123 -vC Maid.mp3')
```

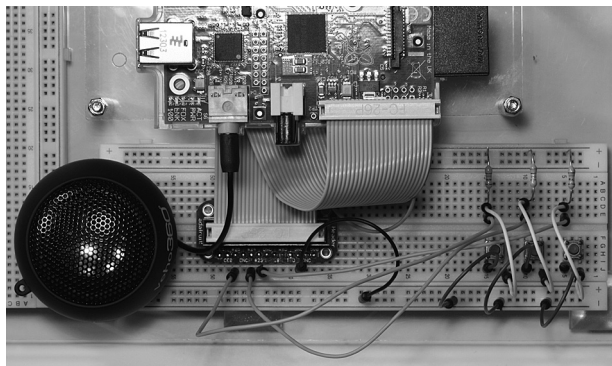
Schéma přehrávače MP3

Všechna propojení vývodů a tlačítek ukazuje obrázek 3.9. K existujícímu obvodu sestavenému podle obrázku 3.7 můžete jednoduše přidat další dvě tlačítka a odporníky. Přejete-li si, můžete ponechat zapojenou i LED, i když do tohoto obvodu nebude zahrnuta.

Obrázek 3.10 ukazuje skutečný obvod na propojovací desce s počítačem RasPi připojeným plochým kabelem ke konektoru Pi Cobbler a s reproduktorem Veho360 připojeným na analogový audiokonektor. Tři tlačítka vybírající jednu ze tří skladeb MP3 jsou umístěna na pravé straně propojovací desky.



Obrázek 3.9 Zapojení přehrávače MP3



Obrázek 3.10 Skutečný přehrávač MP3

Testování projektu

Načtete program MP3_Player do adresáře pi. Potom otevřete terminálové okno a napíšete:

```
sudo python MP3_Player.py
```

```
pi@raspberrypi ~ $ sudo python MP3_Player.py
numid=3,iface=MIXER,name='PCM Playback Route'
; type=INTEGER,access=rw-----,values=1,min=0,max=2,step=0
: values=1
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
version 1.14.4; written and copyright by Michael Hipp and others
free software (LGPL/GPL) without any warranty but with best wishes
Decoder: ARM

Terminal control enabled, press 'h' for listing of keys and functions.

Playing MPEG stream 1 of 1: class.mp3 ...

MPEG 2.0, Layer: III, Freq: 22050, mode: Joint-Stereo, modext: 2, BPF : 209
Channels: 2, copyright: No, original: Yes, CRC: No, emphasis: 0.
Bitrate: 64 kbit/s Extension value: 0
Title: onclassical_demo_latry_bach_fugue_in_d_major_bbw-532_live_cut-version
Artist:
Comment: Album:
Year: Genre: Unknown
Frame# 680 [ 5212], Time: 00:17.76 [02:16.15], RVA: off, Vol: 100(100)
```

Obrázek 3.11 Přehrávání MP3

Zjistil jsem, že pokoušet se spustit tento program v prostředí Python IDLE shell může způsobovat nespolehlivou funkci, protože aplikace mpg123 vyžaduje přímý vstup z klávesnice, který prostředí shell může přerušit. Jakmile se program spustí, jsou k dispozici všechny příkazy uvedené v tabulce 3.3. Můžete experimentovat s různými příkazy a ocenit hloubku funkčnosti aplikace mpg123.

Obrázek 3.11 ukazuje přehrávač MP3 v provozu. Na posledním řádku obrázku jsou uvedeny informace ze záhlaví MPEG (*Moving Picture Experts Group*) a údaje o běhu.

Rozšíření projektu

Předvedl jsem vám klasický přehrávač MP3, který může být mnoha způsoby rozšířen tak, aby splňoval budoucí požadavky. Zobrazení je možno změnit ze standardního monitoru na víceřádkový sériový displej LCD. Prozatím nám stačí soustředit se na jednodušší aspekty použití RasPi pro ovládání projektu.

Jiný aspekt, kterému jste se možná podívovali, je použití samostatného tlačítka určeného pro specifickou funkci, například skok na následující skladbu. Zabudování jednotného tlačítka by znamenalo spoustu softwarové práce, protože aplikace mpg123 očekává vstup ovládacích kláves ze standardního vstupního zařízení (stdin), tj. z klávesnice. Změnit nebo zřídit paralelní vstupní zařízení není snadné, takže se domnívám, že by nebylo vhodné zahrnovat to do knížky s projekty pro začátečníky.

Přidání dalších skladeb MP3 pro přehrávání je velmi snadné. Stačí je zkopírovat do výchozího adresáře pomocí USB portu RasPi. Program se musí změnit, aby reflektoval názvy nových souborů. Program je však možno modifikovat tak, aby vypsal všechny skladby v adresáři, takže je možno si vybírat ty, které chcete přehrát, což je podstata seznamu k přehrávání.

Počítač RasPi můžete samozřejmě napájet z baterie, takže bude zcela přenosný, ovšem za předpokladu, že pro interakci s uživatelem použijete displej LCD.

Shrnutí

Tuto kapitolu jsem začal výkladem o prototypové pomůcce, konektoru Pi Cobbler, což je nepostradatelný nástroj umožňující nám experimentovat s většinou vývodů GPIO dostupných na RasPi.

Ve výkladu o systému ALSA Linuxu jsem vám ukázal, jak operační systém počítače RasPi zpracovává zvuk. Systém ALSA je nutno nainstalovat, abyste mohli používat aplikaci mpg123, která je základnou pro projekt přehrávače MP3. Představil jsem vám také základní formáty zvukových souborů včetně formátu MP3, který se používá pro záznam hudby přehrávané v projektu. Následovala krátká zájímavá zastávka zabývající se frekvenčním průběhem zvukového systému RasPi naměřeným na základě růžového šumu.

Potom jsem vám ukázal, jak lze používat tlačítka k vytváření vstupních signálů pro RasPi prostřednictvím vývodů konektoru GPIO. Tato tlačítka byla použita i jako součást projektu přehrávače MP3.

Následoval podrobný výklad aplikace mpg123 MP3 obsahující podrobný seznam všech ovládacích prvků dostupných pro použití v této mnohostranné aplikaci.

Kapitolu jsem ukončil stručným výkladem o rozšíření přehrávače MP3 tak, aby se více podobal moderním přenosným přehrávačům.