



Středoškolská technika 2017

Setkání a prezentace prací středoškolských studentů na ČVUT

ŠKOLNÍ PROPAGAČNÍ ROBOT

Matěj Vasilevski

**Střední průmyslová škola elektrotechnická
V Úžlabině 320, Praha 10**

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil jsem literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

V Praze dne

.....

podpis autora (jméno a příjmení)

Anotace

Tato práce se zabývá popisem zhotovení školního propagačního robota založeného na mikropočítači Raspberry Pi. V práci je popsána mechanická konstrukce robota a jsou rozebrány jednotlivé softwarové části práce (web server, řízení displeje atd.).

Annotation

This work describes construction of school promotional robot that is based on the Raspberry Pi platform. The work contains description of mechanical construction of the robot and covers software topics (web server, controlling of the matrix display etc.).

Obsah

Úvod	5
1 Použité prvky	6
1.1 Raspberry Pi.....	6
1.2 Modul s PCA9685.....	6
1.3 Vlastní destičky.....	7
1.4 Arduino	7
1.5 LED maticový displej	7
1.6 Servo motory.....	8
1.7 PIR sensor	8
1.8 IR přijímač + ovladač.....	8
1.9 Mikrofon	9
2 Konstrukce, prvotní spuštění	9
2.1 Tělo robota	9
2.2 Čelo	9
2.3 Nohy.....	10
2.4 Vnitřek skříně.....	10
2.5 Čepice.....	10
2.6 Prvotní spuštění.....	10
3 Webové rozhraní	11
3.1 Webserver	11
3.2 Webová stránka.....	11
4 IR přijímač.....	12
4.1 Instalace LIRCu	12
4.2 Vytvoření konfiguračního souboru ovladače.....	13
5 Komunikace s Arduinem.....	13
6 Ovládání displeje.....	14
Závěr.....	15
Seznam literatury a ostatních zdrojů	16
Seznam příloh	18
Příloha č. 1: Návrh destičky pro napájení servo motorů	19

Úvod

Při volbě tématu mé dlouhodobé maturitní práce jsem se rozhodl pro vylepšení dosavadního školního robota. Toto téma jsem si vybral, protože začátkem školního roku jsem byl ještě čerstvým vlastníkem jednoho Raspberry a zajímalo mě, čeho všeho se s ním dá dosáhnout a kde jsou jeho limity.

1 Použité prvky

1.1 Raspberry Pi

Raspberry Pi je mikropočítač o velikosti kreditní karty, který lze používat jako plnohodnotný počítač. Cílem tvůrců bylo vyrobit levný počítač, který byl vhodný pro výuku programování a „oživení“ výuky technických předmětů. V této práci je použita verze 3 B, která disponuje 4 jádrovým procesorem architektury ARM a 1 GB paměti RAM. Mikropočítač je vybaven 4 USB porty, 1 micro USB konektorem určeným pro napájení, 1 Ethernetovým portem. Má slot na paměťovou kartu (na které je umístěn operační systém), HDMI a 3,5 mm JACK konektor. Od běžných počítačů ho odlišuje přítomnost 40 GPIO (General Purpose Input Output) pinů, tzv. univerzálních vývodů, které jsou programovatelné (nastavitelné jako vstup nebo výstup). S jejich pomocí lze komunikovat po sběrnici I2C, SPI a přes rozhraní UART. Tyto piny jsou napojeny přímo na procesor, je tedy nutné dávat pozor, aby se na ně nedostalo napětí větší než 3,3 V, které by procesor mohlo poškodit.

1.2 Modul s PCA9685

PCA9685 je čip určený pro řízení LED diod a ovládání servo motorů. Též je označován jako PWM expandér. Má 12-bitové rozlišení a může ovládat až 16 výstupů. Komunikuje po sběrnici I2C. Frekvence výstupního PWM signálu je nastavitelná v rozsahu cca 20 Hz až 1,5 kHz. Na modulu se nachází svorky určené k přivedení 5 V napájení, avšak jejich provedení (miniaturní šrouby) naznačuje, že jsou vhodné spíše pro napájení více LED diod než na několik servo motorů.

K ovládání tohoto modulu byla použita příslušná knihovna pro Python od společnosti Adafruit (dostupná na adrese https://github.com/adafruit/Adafruit_Python_PCA9685). Jediný nedostatek byl v tom, že byl problém při volání funkce provádějící reset modulu (tj. vynulování všech registrů). Autory byla tato funkce definována odděleně od objektu představujícího modul PCA9685, nyní je tato metoda doplněna přímo k objektu. Metoda byla doplněna ve zdrojovém kódu knihovny ještě před její instalací, aby se pak zbytečně nevytvářela nová třída dědicí od původní nebo se nepoužívaly neobvyklé okliky.

1.3 Vlastní destičky

Vzhledem k pochybnému zajištění napájení bylo vhodné vyrobit si vlastní desku, která by zajistila slušný přívod elektřiny k motorům. Na nové desce jsou stejně jako na modulu 3 řady pinů. Napětí se přivádí přes 2 molex konektory. Signály řídící motory jsou z modulu vyvedeny na destičku 40 pinovým plochým kabelem (klasický IDE kabel). Deska byla frézována ve škole.

Druhá destička propojuje přívody maticového displeje s GPIO vývody Raspberry. Na Raspberry je nasazen prostřední konektor IDE kabelu s tím, že kratší část je připevněna ke skříni, aby GPIO bylo přístupné i zvenčí, a delší část vede do destičky. Kromě propojení s displejem jsou vyvedeny nepoužité GPIO piny pro další použití.

Třetí destička nese zesilovač k elektretovému mikrofonu. Má vývody pro napájení (5 V a zem), připojení mikrofonu (kladný pól a zem) a výstup zesilovače (signál a zem), který povede do Arduina.

1.4 Arduino

Arduino je návrhářská deska postavená na mikrokontroléru ATmega. Programuje se v jazyce vycházejícím z jazyka C. Zatímco Raspberry je plnohodnotný počítač, Arduino je pouze vývojová deska. Má mikrokontrolér doplněný o přídavné obvody (napájení, ochranné diody atd.), jehož vstupy a výstupy jsou vyvedeny v podobě lehce přístupných pinů. V této práci je využit, protože na rozdíl od Raspberry disponuje převodníkem z analogového na digitální signál.

1.5 LED maticový displej

Použitý maticový displej má 16 řádků a 32 sloupců, tzn. 512 RGB LED diod. Avšak na 1 desce je pouze 12 čipů (posuvných registrů), každý po 16 čípech, tzn. 192 výstupů. Tudiž je displej rozdělen na 8 sekcí, které se postupně vykreslují. 1. sekce se skládá z 1. a 9. řádku, 2. sekce z 2. a 10. řádku atd. Řádky jsou voleny pomocí vodičů A, B, C (viz obrázek). Což odpovídá, neboť tři vodiče poskytují $2^3 = 8$ kombinací. Na dvou řádcích je $32 + 32 = 64$ RGB LED diod, z toho plyne $64 \cdot 3$ (barvy) = 192 diod. Což odpovídá 192 výstupům z řídicích čipů. Barvy prvních osmi řádků určují vodiče R1, B1, G1, barvy

druhých osmi řádků (9., 10., 11. atd.). Tím je zajištěno barev Avšak takto jednoduše řízený displej by měl málo barev, protože výstupy čipů mají jen dva stavy – vypnuto a zapnuto. Pro získání různých barevných odstínů je potřeba vykreslovat celý displej velice často. Raspberry ve verzi 3 (bez běžícího grafického rozhraní) zvládá vykreslit celý displej přibližně 450x za sekundu.

1.6 Servo motory

Servo motory jsou motory, u kterých lze nastavit polohu natočení osy. Řídí se PWM signálem (šířka pulzu určuje natočení), napájí se malým stejnosměrným napětím (přibližně 5 V). Mají široké uplatnění v různých situacích, např. u modelářů do RC modelů.

1.7 PIR sensor

PIR senzor se používá jako spolehlivý detektor pohybu. Zkratka PIR je z anglického „passive infrared detector“, volně přeloženo jako pasivní infračervený detektor. Tyto senzory využívají pyroelektrický jev. Pyroelektrický jev je možné definovat jako vykazování elektrického potenciálu při změně teploty. Ten jev je vlastností materiálů nazývaných pyroelektrické materiály (které jsou podmnožinou piezoelektrických materiálů). Změny teploty způsobuje například průchod člověka nebo zvířete. Nestane se tak, že by senzor sepnul, když se kolem bude kutálet míč. Díky tomu se tento typ senzorů stal velmi oblíbený, především v zabezpečovací technice.

Na senzoru jsou dva trimry. Prvním se nastavuje citlivost snímání okolí. Druhý udává dobu, po kterou výstup senzoru po detekci pohybu zůstává sepnutý. Čidlo se napájí 5 V.

1.8 IR přijímač + ovladač

Pro možnost ovládání na dálku byla využita sada dálkového ovladače a infračerveného přijímače HX1838. Tato sada je oblíbená, na obchodním serveru ebay.com se prodává od cca 30 Kč. Součástí sady je i modul s rezistory a indikační diodou, který zprostředkovává pracovní bod pro IR přijímač.

1.9 Mikrofon

Mikrofon byl zvolen elektretový. Má rovnou frekvenční charakteristiku, je malý a levný. Elektretový mikrofon je typ kondenzátorového mikrofону, který nepotřebuje externí zdroj polarizačního napětí, nýbrž toto napětí je vytvářeno tzv. elektretem. Elektret je keramický, elektricky polarizovaný materiál. Polarizace se dosáhne vypálením materiálu v silném elektrostatickém poli (působením vysokých teplot a silného elektrostatického pole). Elektret si náboj udržuje po dobu cca 5 let s tím, že se dá obnovit. Elektret nemůže být zdrojem energie (náboje nejsou volné), ale poskytuje elektrostatické pole potřebné k činnosti mikrofónu.

2 Konstrukce, prvotní spuštění

2.1 Tělo robota

Stejně jako v předchozí verzi, na tělo robota byla použita počítačová skříň. Po nalezení optimální skříně (dostatečná šířka, bez koroze) jsem se pustil do přestavby. Ze skříně byly vymontovány přebytné prvky: koš na pevné disky a optické mechaniky a malý signální reproduktor. Po demontáži nadbytečných prvků byly do kovového čela skříně vyvrtány díry pro uchycení maticového displeje. Dále bylo zapotřebí vyvrtat do bočnic díry pro přišroubování rukou. Kromě těchto děr byly udělány i díry pro přívodní kabely od servo motorů (na obou stranách po 1 díře). Díry pro kabely jsou umístěné tak, aby byly co nejméně „na očích“.

2.2 Čelo

Plastové čelo skříně vyžadovalo upravit jen pár drobností. Bylo zapotřebí uříznout spodní třetinu čela kvůli displeji. Toho bylo dosaženo použitím ruční přímočaré pily. Dále byly s pomocí vrtačky a přímočaré pily udělány otvory na přichycení PIR senzorů.

2.3 Nohy

Nohy byly potřebné ze dvou důvodů. Jednak ruce byly dlouhé a často narážely na zem, jednak celkovému vzhledu robota něco chybělo. Na nohy jsem využil kovových nábytkových nohou o délce 200 mm a průměru 30 mm. Kvůli rozmístění nýtů na spodku skříně bylo nutné vyvrtat nové díry a vybrousit místo pro nýty.

2.4 Vnitřek skříně

Za účelem dosažení alespoň minimálního pořádku byl uvnitř skříně zhotoven „rozvaděč“. Jedná se o laminátovou desku (stejná, jako se ve škole používá na plošné spoje, akorát zbavená měděné vodivé vrstvy), která je našroubována na místo obvyklé pro základní desky běžných počítačů. Teprve na tuto desku jsou přichyceny zbylé prvky (mikropočítač Raspberry, modul s PWM expandérem PCA9685 a další). Tento přístup přináší řadu výhod. Například to, že s laminátem se lépe pracuje (vrtají díry apod.) a konektory na Raspberry jsou pohodlněji přístupné.

2.5 Čepice

Vrchní čepice se skládá ze 4 dílů: vrchního kusu kuželu (ve kterém je zabudován mikrofon), spodního kusu kuželu (který je kvůli minimalizaci objemu tělesa dutý) a dvou prvků umožňujících propojení vršku se spodkem a připevnění servo motoru uvnitř spodního kuželu. Všechny 4 díly byly navrženy v AutoCADu.

2.6 Prvotní spuštění

V novějších verzích operačního systému Raspbian (linuxová distribuce navržená pro mikropočítač Raspberry, vychází z Debianu) je ve výchozím stavu z bezpečnostních důvodů zakázán SSH server (protože mnoho uživatelů si po instalaci nezmění heslo, je pro případného útočníka velice snadné se do takového zařízení dostat). SSH server lze jednoduše povolit použitím konfiguračního nástroje `raspi-config`, který je určen pro provedení běžných a častých konfiguračních zásahů. Nástroj se vyvolá příkazem **sudo raspi-config** (sudo kvůli potřebným právům administrátora). V zobrazené nabídce zvolíme položku **č. 5 – Interfacing Options**, kde povolíme **SSH** a **I2C** (kvůli modulu

PCA9685 pro řízení servo motorů). V položce č. 4 – **Localisation Options** je vhodné nastavit rozložení klávesnice a časové pásmo. Nakonec v položce č. 2 – **Hostname** (hostname je jméno identifikující počítač v síti) je nastaveno jméno na **robot123**. Následuje potvrzení a restart.

3 Webové rozhraní

3.1 Webserver

Jako webserver je nasazen Lighttpd. Lighttpd je dle slov autorů rychlý, spolehlivý a hodí se tam, kde není potřeba komplexní řešení. Pro tyto své vlastnosti byl také zvolen. Instaluje se příkazem **sudo apt-get install lighttpd**.

Po instalaci je třeba povolit cgi skripty. Na konec souboru **lighttpd.conf** ve složce **/etc/lighttpd/** je potřeba přidat následující:

```
$HTTP["url"] =~ "^/cgi-bin/" {  
    cgi.assign = ( ".py" => "/usr/bin/python" )  
}
```

a do seznamu modulů (*server.modules*) je nutné připsat:

```
"mod_cgi"
```

Skript, který zpracovává data z formuláře na webové stránce, patří do složky **cgi-bin**. Tu vytvoříme příkazem **sudo mkdir /var/www/html/cgi-bin** (/var/www/html je kořenový adresář pro stránky na serveru). Následně se příkazem **sudo chown www-data:www-data /var/www/html/cgi-bin** změní vlastník složky **cgi-bin** na uživatele a skupinu **www-data** (skupina i uživatel mají stejné jméno). Nakonec se skript udělá spustitelným pomocí příkazu **sudo chmod +x /var/www/html/cgi-bin/název_skriptu.py**.

3.2 Webová stránka

Webová stránka je napsána v jazyce HTML. Stránka je psána s důrazem na funkčnost a upozaduje okrasné prvky. Na stránce je seznam s možností volby chování, textové pole pro zadání zobrazovaného textu, dvě barevné palety (přístupné po kliknutí na tlačítko) a

tlačítko pro odeslání formuláře. Po odeslání formuláře se zobrazí výstup skriptu zpracovávajícího odeslaná data.

Barevná paleta je zprostředkována knihovnou jscolor. Tato knihovna je dostupná na adrese <http://www.jscolor.com>. Po stažení archivu s knihovnou a jeho následném rozbalení se nakopíruje do adresáře, ve kterém je umístěna webová stránka. Ve zdrojovém kódu stránky se knihovna přidá značkou `<script src="jscolor.js"></script>`.

4 IR přijímač

4.1 Instalace LIRCu

LIRC (Linux Infrared Remote Control) je program/sada programů umožňující přijímání a odesílání infračervených signálů. LIRC byl zvolen z důvodu, že ve spojení s Raspberry ve funkci multimediálního centra se často používá, a tak lze běžné problémy při zapojování dohledat na diskuzních fórech. Instaluje se příkazem **sudo apt-get install lirc**. Instalace knihovny pro Python je příkazem **sudo apt-get install python-lirc**.

Po instalaci se LIRC musí nakonfigurovat. Do souboru `/etc/modules` se přidá:

```
lirc_dev
```

```
lirc_rpi gpio_in_pin=26
```

v souboru `/etc/lirc/hardware.conf` se provedou následující změny:

```
LIRCD_ARGS="--uinput"
```

```
DRIVER="default"
```

```
DEVICE="/dev/lirc0"
```

```
MODULES="lirc_rpi"
```

a do souboru `/boot/config.txt` se přidá:

```
dtoverlay=lirc-rpi,gpio_in_pin=26 (pozor, před gpio_in_pin nesmí být mezera).
```

Ověřit funkčnost lze pomocí příkazů **sudo /etc/init.d/lirc stop** (dočasné vypnutí LIRCu na pozadí) a **mode2 -d /dev/lirc0**. Druhý jmenovaný příkaz zapne program mode2, který vypisuje strukturu přijatého infračerveného signálu (vypisuje šířky pulzů a mezer, z jejichž posloupnosti se infračervený signál sestává).

4.2 Vytvoření konfiguračního souboru ovladače

Pro tvorbu konfiguračního programu je určen program `irrecord` (pozor, potřebuje běžící LIRC – příkaz `sudo /etc/init.d/lirc start`). Příkazem **`irrecord -d /dev/lirc0 -f konfigurace.conf`** (přepínač `-f` vynucuje ukládání v RAW módu, používá se, pokud `irrecord` nemůže rozpoznat protokol (strukturu), který ovladač používá) se spustí průvodce pro vytvoření konfiguračního souboru. Na začátku procesu požaduje mačkání tlačítek (aby v signálu našel určité vzory – délku celého signálu, hlavičku...), které signalizuje postupným psaním teček. Poté následuje ukládání jednotlivých kláves: zadání názvu klávesy (např. „KEY_0“) a stisknutí korespondující klávesy.

Také je vhodné vytvořit soubor **`lircrc`** kvůli pozdější práci s Pythonem. Jeho struktura je následující:

begin

button = KEY_0

prog = ridiciProgram

config = 0

end

begin

button = KEY_1

prog = ridiciProgram

config = 1

end

begin

...

Button je název stisknutého tlačítka, prog je jméno programu a config je to, co se mu má předat při stisku daného tlačítka na ovladači.

5 Komunikace s Arduinem

Raspberry a Arduino spolu komunikují po sběrnici I2C. Ačkoliv u Raspberry je hodnota logické 1 3,3 V a Arduina 5 V, není zapotřebí převodníku logické úrovně. Protože

Raspberry je master (a master je ten, který udává taktovací kmitočet CLK) a Arduino slave (může jen přizemňovat vedení), nemůže se na vodiči objevit 5 V z Arduina, které by byly pro Raspberry nebezpečné.

V Arduinu je nahrán jednoduchý program, který využívá knihovny Wire. V programu je nastavena I2C adresa 0x09 a nastavení funkce, která se má volat při žádosti o data od Raspberry. Arduino čeká v nekonečné smyčce na žádost o data. Na žádost odpoví odesláním dat naměřených na analogovém vstupu (mikrofonní napětí). Data naměřená v 10-bitovém převodníku jsou vydělena 4, aby se vešly do jednoho bytu a komunikace po I2C byla jednodušší.

6 Ovládání displeje

Za účelem ovládání maticového displeje byl z Githubu zkopírován repozitář (<https://github.com/hzeller/rpi-rgb-led-matrix>) s knihovnou pro ovládání tohoto displeje. V tomto repozitáři je také aplikace, která na displej vykresluje obrázek, který jí byl zadán jako vstup. V práci je využita tato aplikace napsaná v jazyce C, neboť je násobně rychlejší než knihovna pro Python.

Závěr

Práci se nepodařilo dokončit, neboť jsem velice hrubě podcenil čas potřebný ke kompletaci výrobku. Program na ovládání motorů není vůbec v provozu, program na ovládání zobrazovaného textu by potřeboval několik úprav (vyzkoušet změnu pozadí, pravděpodobně chybí adekvátní řešení při změně textu). Program na komunikaci s Arduinem funguje, ale není na nic navázán. Je tedy prozatím zbytečný. To samé platí i pro LIRC – příjem infračervených signálů funguje, avšak není k ničemu navázán. Jedině web server zvládá dělat to, co má. Akorát skript zpracovávající formulář nezvládá odeslat žádné údaje.

Ačkoliv práce na mechanické konstrukci robota výrazně posunula mé znalosti z této problematiky a rozšířila mi obzory, při zpětném ohlédnutí lituji, že jsem se do toho pouštěl. Kromě toho, že strojírenství není můj obor, jsem se ani k nějakému programování nedostal. Jednak kvůli nedostatku času, jednak proto, že jsem příliš mnoho času strávil „lepením“ technologií (výběr správného/pokus o svůj vlastní web server, meziprocsová komunikace – zde jsem se snažil využít roury, webové stránky – framework/základní HTML + javascript), které důvěrně neznám.

Smrtelná mi byla honba za přílišnou dokonalostí. Rozdělení několika skříní při hledání té optimální (dostatečná šířka a přitom malá velikost, použitelný stav bez pokročilé koroze atd.), několikanásobné předělání softwarového řešení (web server: Apache, základní server v Pythonu, pokus o vlastní server v Pythonu, Lighttpd; skripty v PHP + bash vs Python; ovládání displeje zkompilevaným programem z Githubu vs používání Pythoní knihovny). Příliš mnoho času jsem strávil při hledání optima, abych nakonec použil celé Arduino jako A/D převodník.

Seznam literatury a ostatních zdrojů

INSTALLING OPERATING SYSTEM IMAGES USING WINDOWS [online]. [cit. 2017-03-20]. Dostupné z: <https://www.raspberrypi.org/documentation/installation/installing-images/windows.md>

SSH [online]. [cit. 2017-03-20]. Dostupné z: <https://www.raspberrypi.org/documentation/remote-access/ssh/>

ŠTRAUCH, Adam. *Lighttpd: lehký webserver* [online]. [cit. 2017-03-20]. Dostupné z: <https://www.root.cz/clanky/lighttpd-lehky-webserver/>

Lighttpd does not find my Python web page [online]. [cit. 2017-03-20]. Dostupné z: <https://www.raspberrypi.org/forums/viewtopic.php?f=32&t=90703>

KUBÁLEK, Jan. *PWM expandér PCA9685* [online]. [cit. 2017-03-20]. Dostupné z: <http://robodoupe.cz/2016/pwm-expander-pca9685/>

VODA, Zbyšek. *Programujeme arduino* [online]. [cit. 2017-03-20]. Dostupné z: <https://arduino.cz/programujeme-arduino/>

BURGESS, Phillip. *How the matrix works* [online]. [cit. 2017-03-20]. Dostupné z: <https://learn.adafruit.com/32x16-32x32-rgb-led-matrix?view=all#how-the-matrix-works>

MICHALEC, Libor. *PIR detektor: skvělý sluha, ale zlý pán* [online]. [cit. 2017-03-20]. Dostupné z: <http://vyvoj.hw.cz/automatizace/pir-cidlo-skvely-sluha-ale-zly-pan.html>

FEJT, Radek. *Pasivní infračervený senzor* [online]. [cit. 2017-03-20]. Dostupné z: <http://senzory.aspone.cz/pir.htm>

JEDLIČKOVÁ, Markéta. *JAK HACKNOUT SERVO MOTOR* [online]. [cit. 2017-03-20]. Dostupné z: <https://arduino.cz/jak-hacknout-servo-motor/>

FUKÁTKO, Jaroslav. *Akustika* [online]. [cit. 2017-03-20]. Dostupné z: https://skola.uzlabina.cz/==UCITELE_MATERIALY==/Fukatko/vyuka_15_16/ZE_2/Akustika.pdf

Elektrety [online]. [cit. 2017-03-20]. Dostupné z: <http://dielektrika.kvalitne.cz/elektrety.html>

BAIN, Alex. *Setting Up LIRC on the RaspberryPi* [online]. [cit. 2017-03-20]. Dostupné z: <http://alexba.in/blog/2013/01/06/setting-up-lirc-on-the-raspberrypi/>

Seznam příloh

Příloha č. 1: Návrh destičky pro napájení servo motorů.

Příloha č. 1: Návrh destičky pro napájení servo motorů

