

ARDUINO MI POMÁHÁ NA DIVADLE: VSTUPOJEME DO SVĚTA SVĚTELNÉHO DESIGNU

PAVEL STRÍŽ (CZ)

Abstrakt. V článku navazuji na zkušenosti z dřívějších let kolem technické pomoci u divadelních ochotníků v Bučovicích (Česká republika). Zaměřuji se na začlenění Arduina do světelného designu, ale nahlédnu na dřívější problém spuštění dávkových souborů z PDF v Linuxu i se pokusím popsat současné plány.

Klíčová slova. Arduino, divadlo, ochotníci, světelný design, ovládání elektrospotřebičů, relé, SRD, SSR, bash, stty, T_EX, hyperref, Foxit Reader, xpdf.

ARDUINO HELPS ME IN THE THEATRE PRODUCTION: ENTERING THE WORLD OF LIGHT DESIGN

Abstract. The article summarizes my new experience as a technician in the local theatre in Bučovice (Czech Republic). It focuses on implementing Arduino to the light design, but I also look back to the previous problem of running bash scripts from the PDF file under Linux as well as I am describing my current plans.

Keywords. Arduino, theater production, amateurs, light design, handling electrical devices, relay board, SRD, SSR, bash, stty, T_EX, hyperref, Foxit Reader, xpdf.

⚡ Upozornění autora ⚡

Během bastlení v tomto článku se pracuje s malým, ale hlavně s otevřeným nízkým napětím, které může být smrtelné (230 V~). Pokud nemáte příslušné vzdělání či zkušeného odborníka k ruce, bastlení se raději vyhněte! Pokud vám tento článek nedává spát, nezapojujte potenciálně smrtelné napětí (přívodní kábl ke spotřebičům do sítě), i tak můžete sledovat LEDky u relátek, kde je maximálně 5 V=. Pokud si musíte zapojit vše, nechte si zapojení alespoň zkontrolovat, protože pustit si omylem 230 V~ do notebooku jej může naprosto zlikvidovat. Nezbytné varování vyřčeno, pojďme na to.

Představení problému

S Arduino světem jsem se setkal na konferenci OSSConf v roce 2015 při neformální diskuzi. Poradil mi to Remigiusz Olejnik, díky mu. Měl jsem v hlavě „něco“ co by mi ovládalo světla přes notebook, ale měl jsem jen mlhavou představu o Raspberry Pi. Jinak totiž musím vše ovládat přes vypínače, to je únavné a při velkém množství spotřebičů nerealizovatelné. Nakoupit DMX mixážní pult by asi šlo, ale nejlevnější stojí cca 18 tisíc korun (cca 650 eur) a není zajištěna

Existují relátka, která udrží mnoho desítek ampér, ty by se možná hodily na velké reflektory, ale ty zatím nezvažuji připojovat. Ty, které na divadle máme z 60. let (1000 i víc wattů), půjdou do šrotu, neb nedávají adekvátní svit.

Pracuji s Arduino Uno či Arduino Mega, dle počtu spotřebičů (to závisí na studované inscenaci). Přivádím 230 V~ přes přívodní kábl (ze starých káblů k počítači jsem si udělal flexa odstřihnutím koncovky), které rozvádím do zásuvek a fázi ještě do relátek. To má výhodu, že nemusím do přívodních káblů u spotřebičů zasahovat či se nějak vrtat ve vidlicích. Spotřebič zůstává beze změny.

Z WAGO spojek/svorkovnic to mám vedené do relátek a dál do zásuvek. U některých YouTube videí si napojují relátka přímo mezi sebou, ale do toho jsem nešel. Naopak nulák a zem mezi zásuvkami napojené mám.

Považoval jsem za povinnost u svého prvního projektu připojit jistič, použil jsem Noark Ex9BN 1P B25. Funguje dobře, ale když se mi podařil zkrat, nezachytil vše. Tohle musím domyslet, aby mi výboj nešel mimo pokoj. Do (vrchem) a z jističe (spodem) mi jde fáze.

Zvláštní část mých experimentů mi tvořily dálkově ovládané zásuvky (RF 433,92 MHz), o tom snad jindy. Vyhovoval mi polský výrobce Kemot s produktem URZ3143 se statickým kódováním na ovladači. To má výhodu, že nedochází k fyzickému styku s Arduinem, udrží 2300 či až 3600 W u jiných výrobců, ale ne se všemi ovladači se dá rozumně domluvit a musí být zajištěno čisté RF vysílání i přijímání. Zůstal jsem u káblíků, protože něco vést do výšky dvou metrů znamená, že mi tam některý z herců projde, vést něco nad hlavami herců znamená, že to musím dobře uchytit, aby se nic nepohnulo, nedejbuh nespadlo.

Na obrázku 1 je schéma pro spotřebiče, na fotce (obrázek 2) můžete vidět zapojení pro 12 spotřebičů (tzv. instantní krabicové řešení). Pokud nepoužijeme digitální pin 13 (problikává na začátku, díky za tip Slavko Fedorikovi), tak nám stačí Arduino Uno a digitální piny. Analogové piny jsem nezkoušel. Jiný typ desky by neměl být problém. Raspberry Pi 2, model B, s 3,3 V na digitálních pinech by mělo být dostatečné, viz toto YouTube video. Napočítal jsem jich 17 dostupných. Projekt v akci, resp. otevřenou krabicí, bylo možné vidět na OSSConf 2016.



Obrázek 2. Řešení se 12 zásuvkami a 8 dálkově ovládanými zásuvkami (RF, firma Kemot)

2. Strana Arduina

Z Arduina se stává prostředník mezi notebookem a spotřebiči. Přijímá komunikaci přes sériovou linku, každý bajt/příkaz má svůj účel (má konvence: velká písmena zapínají spotřebiče, A je první spotřebič, B druhý atd.; malá vypínají), v mém případě jen zapnutí či vypnutí spotřebiče. U prvních testů jsem používal i cifry, ale těch je jen 10, spotřebičů mohu mít víc. Vícebajtové sekvence lze uvážit, ale snažil jsem se tomu vyhnout. Nebyl velký problém si zásuvky tužkou popsat. Testy funkčnosti kódu mohu dělat v Arduino IDE/Studiu pomocí monitoru (Tools → Serial Monitor), znaky mohu zasílat z tohoto místa. V produkci sekvenci píše v $\text{T}_{\text{E}}\text{X}$ ovém souboru. Malou prodlevu mezi spotřebiči obvykle nastavuji na úrovni kódu Arduina, aby se spotřebiče nevypínaly či nezapínaly v jednom okamžiku.

```
// Abych nemusel kroutit káblíky, mám to otočené zde.
// Digitální pin 13 spíš nepoužívám, podobně se snažím šetřit 0 a 1.
int jadro[] = {0, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2};
int citac; // cyklus přes digitální piny
int velikost = sizeof(jadro) / sizeof(int); // počet pinů
int h; // načtený znak přes sériovou linku

void setup() {
  Serial.begin(9600);
  for (citac = 1; citac < velikost; citac = citac + 1) {
    pinMode(jadro[citac], OUTPUT);
  } // for()
} // setup()

void loop() {
  if (Serial.available() > 0) { // Dorazil příkaz/bajt?
    h = int(char(Serial.read())); // Ano, ukládám si.
    if (h < 'a') { // Je to velké písmeno?
      digitalWrite(jadro[h - 'A' + 1], HIGH); // Ano, zapni spotřebič!
    } else { // Ne, není, je to malé písmeno.
      digitalWrite(jadro[h - 'a' + 1], LOW); // Není, vypínám
      spotřebič!
    } // if(), zapni/vypni spotřebič
    delay(50); // Chvilí přibrzdi, nezapínej/nevypínej vše co
    nejrychleji.
  } // if(), příkaz
} // loop()
```

Zde představuji další funkční možnost, kdy jsem přes Arduino ovládal 4 dálkově ovládané zásuvky od polské firmy Kemot, produkt URZ3143. Jedním z mých pracovních cílů bylo spojit kódy, tedy ovládat 12 elektrospotřebičů přes káblíky a 4 či 8 elektrospotřebičů přes tyto zásuvky. Používám 433MHz RF link kit. Jak vidíte, spojení bude operativa na úrovni bastlení a programování.

```
#include <RCSwitch.h> // Užívaná knihovna.
RCSwitch mySwitch = RCSwitch();
int pocet = 3; // Vlastní počet pokusů.
String h; // Načítaný bajt ze sériové linky.
void mal(char* malsend) { // Příkaz odesílající znaky.
  for (int malcount = 0; malcount < pocet; malcount++) { // Počet
    pokusů.
      mySwitch.sendTriState(malsend); // Zašli sekvenci na zásuvky.
      delay(200); // Přestávka mezi pokusy. }
      delay(100); // Přestávka mezi příkazy. }

void setup() {
  Serial.begin(9600); // Zahájení komunikace.
  mySwitch.enableTransmit(2); // Aktuální nastavení na zásuvky.
  mySwitch.setPulseLength(164); // Délka vln přenosu, experimentálně
  nastaveno.
  mySwitch.setProtocol(1); // Sada zásuvek, nastavení dle příjmu v
  RCSwitch.
  //mySwitch.setRepeatTransmit(1); // Počet pokusů si nastavuji sám. }

void loop() {
  if (Serial.available() > 0) {
    h = String(char(Serial.read()));
    Serial.println(h); // Kontrolní výpis na terminál.
    // Zapnutí zásuvek, mám jich max. 8, zvolil jsem cifry
    if (h == "1") {mal("FF00F0FFFF01");}
    if (h == "2") {mal("FF00FF0FFF01");}
    if (h == "3") {mal("FF00FFF0FF01");}
    if (h == "4") {mal("FF00FFFF0F01");}
    // Vypnutí zásuvek, čísla s klávesou Shift na anglické klávesnici.
    if (h == "!") {mal("FF00F0FFFF10");}
    if (h == "@") {mal("FF00FF0FFF10");}
    if (h == "#") {mal("FF00FFF0FF10");}
    if (h == "$") {mal("FF00FFFF0F10");}
  } // Serial.available()
} // loop()
```

3. Odeslání dat

V produkci to mám tak, že před otevřením PDF se nahraje HEX soubor do Arduina (ten jsem si vytáhl z dočasných souborů, cesty jsou vidět při zapnutí logu v Arduino IDE). V Arduino IDE: **File** → **Preferences**, **Show verbose output during:**, zaškrtnám si obojí **compilation** a **upload**. Pracuji z PDF a po uzavření PDF se načte minimální HEX soubor do Arduina (vyčištění paměti mikrokontroleru na další práci), tedy:

```
void setup() {}
void loop() {}
```

V pozadí se po nakliknutí hypertextového odkazu spouští dávkové soubory, v naší ukázce jsou generované \TeX em, ale mohou být předpřipravené a vedle světel mohou zasahovat i do hudby a zvuků.

Název se dá zjistit rychle při srovnání výpisů `/dev` před a po zapojení Arduina. Například přes program `diff` nebo `colordiff`:

```
$ ls /dev >~/mount-before.txt
$ # zapojíme Arduino
$ ls /dev >~/mount-after.txt
$ diff ~/mount-before.txt ~/mount-after.txt
138a140 > ttyS5
```

Po aktivaci (Linux či CygWin pod Microsoft Windows) komunikace jsem mohl přejít na testy, na mém stroji mám Arduino na `ttyS5`, resp. `COM6`:

```
$ stty -F /dev/ttyS5 9600 cs8 cread clocal
```

Pod Microsoft Windows zašlu sérii bajtů přes dva kroky:

```
echo ABCDE >xfile.txt
print /D:COM6 xfile.txt
```

Pod CygWinem/Linuxem jedním krokem:

```
$ echo ABCDE >/dev/ttyS5
```

Pro zájemce o sériovou komunikaci mohu doporučit tuto stránku s diskuzí.

V produkci před a po tomto bloku nahrávám do Arduina předpřipravený HEX soubor a čistím jej dalším HEX souborem (Microsoft Windows), např.:

```
@rem Nahrání souboru do Arduina
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avrdude.exe"
-C"C:\Program Files
(x86)\Arduino\hardware\tools\avr\etc\avrdude.conf" -v -patmega2560
-cwiring -PCOM4 -b115200 -D
-Uflash:w:svetla\ovladani-svetel.cpp.hex:i
```

```
@rem Spuštění PDF
"C:\Program Files (x86)\Foxit Software\Foxit Reader\FoxitReader.exe"
  carodejky-ze-salemu.pdf
@rem Vyčištění Arduina pro další experimentální práci
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avrdude.exe"
  -C"C:\Program Files
  (x86)\Arduino\hardware\tools\avr\etc\avrdude.conf" -v -patmega2560
  -cwiring -PCOM4 -b115200 -D -Uflash:w:svetla\prazdny-soubor.cpp.hex:i
```

4. Strana PDF: generování

Z pohledu automatizace celého procesu je však výhodné si várky znaků připravit dopředu či necht' mi je generuje T_EXový soubor při přípravě PDF. Ukáží cestu, jak T_EX generuje dávkové soubory, které zařizují odeslání dat Arduinu.

Pracuji přímo s BAT/SH soubory a poněvadž je bezpečnost i na straně T_EXu, tak generuji dočasný soubor, který se na požadovaný název přejmenuje.

T_EX se spouští se zapnutým `--shell-escape` (T_EX Live), na to mám extra profil v editoru T_EXworks. To T_EXu dovoluje volat si příkazy z operačního systému. Šlo by to udělat bez toho, vygenerované soubory by se přejmenovaly po dokončení T_EXování (např. Bash, Makefile).

```
% !TeX program = LuaLaTeX (+shell) TeX encoding = UTF-8
\documentclass[12pt,a4paper]{article}
\usepackage{hyperref}
\newwrite\zapis\newcount\citac
\def\malcom{COM6}\def\maltty{ttyS5}
\def\malukladej{soubory}
% \immediate\write18{mkdir \malukladej} % TeX přímo nepustí.
\def\klik#1 {\advance\citac by 1%
  \def\malformat{\ifnum\citac<100 0\fi\ifnum\citac<10 0\fi\the\citac}%
  \immediate\openout\zapis=\malukladej/\malformat.tex%
  \immediate\write\zapis{echo #1 >xfile.txt}%
  \immediate\write\zapis{print /D:\malcom\space xfile.txt}%
  \immediate\closeout\zapis
  \href{run:\malukladej/\malformat.bat}{#1}%
  \immediate\write18{cd \malukladej\space && mv \malformat.tex
\malformat.bat}
  \immediate\openout\zapis=\malukladej/\malformat.sh%
  \immediate\write\zapis{echo #1 >/dev/\maltty}%
  \immediate\closeout\zapis }
\begin{document}
  vzorek \klik ABCDE nebo \klik FGHIJKL \par
  vzorek \klik abcde nebo \klik fghijkl \par
```

```

vzorek \klik ABCDEabcdeABCDEabcde nebo \klik fghijklFGHIJKLfgghijkl
\par
vsechno zapnout: \klik ABCDEFGHIJKL \par
vsechno vypnout: \klik abcdefghijkl
\end{document}

```

Po ručním vytvoření si adresáře soubory do něj \TeX ukládá dávkové soubory, postupně očíslované. Příkaz `\klik` si načte všechny znaky až po mezeru a zapíše je. Hypertextové odkazy zajišťuje balíček `hyperref`. Zde je ukázána cesta přes `run:`, o vylepšeném způsobu v Linuxu přímo pomocí odkazů za okamžik.

5. Strana PDF: spouštění odkazů pod Microsoft Windows

Před několika lety (OSSConf 2013, Jak se \TeX ista mezi... , str. 131–138) jsem řešil problém spuštění dávkových souborů přes PDF, abych mohl spouštět hudbu a zvuky u místních ochotníků. Vyřešil jsem to pod Microsoft Windows, v programu Foxit Reader a pomocí programu MPlayer. Používám to na zkoušky i divadelní produkci, jen jsem přešel na MPV, odnož s možností konfigurovatelnosti přes jazyk Lua, který v \TeX ovém světě frčí.

6. Strana PDF: spouštění odkazů v Linuxu

Co mě však trápilo, bylo, že jsem neměl řešení pro svět Linuxu. V `xpdf` sice dávkový soubor spustíte (testy roku 2013 i 2016), ale vyskočí vám dialogové okno, což je nepříjemné zdržení na zkoušky či divadelní produkci.

Během psaní článku mě něco osvítilo (procházení manuálu `xpdf`), že bych se měl na dávkové soubory dívat jako na odkazy (po kliknutí se otevírá webový prohlížeč) a jen to trochu předefinovat. Po nainstalování `xpdf` (testováno v Ubuntu)

```
$ sudo apt install xpdf
```

si v domečku stačí vytvořit konfigurační soubor `.xpdfrc` a přidat jeden řádek:

```
urlCommand "sh '%s'"
```

V \TeX ovém světě (balíček `hyperref`) je pak každý odkaz (nepíšeme ani `run:`) zvažován jako dávkový soubor. Je to jádro, se kterým by se dalo dál pracovat, pro divadelní produkci je to řešení nejkratší. Jednoduché uvozovky zajišťují práci se soubory, kde je v názvu mezera, což by nebyl můj případ, takže by se to dalo ještě o dva znaky zjednodušit. Co jedno kliknutí v PDF, tak jedno spuštění.

Praktické by bylo užít `urlCommand "sh rozcestnik.sh '%s'"`, kde by `rozcestnik` otestoval, zda-li je parametr spustitelný soubor (postoupil by `sh`) nebo jiný soubor (postoupil by například v `xpdf` předdefinovanému chování `netscape-remote 'openURL(%s)'`). Nechtějte to pro zájemce jako domácí úkol.

Tím se uspokojivě řeší oba světy (Mac OS X už trochu považuji za linuxový svět a dál to netestuji). Shodou okolností jsem našel ještě jiný přístup. Obejít bezpečnost přes vlastní příponu, které se nastaví příslušná mime pravidla. Pro zájemce viz <https://tug.org/pipermail/texhax/2010-September/015705.html>. Bylo by poutavé zkoumat nastavitelnost dalších programů (Evince, Adobe Reader, KPDF ap.) v tomto duchu, ale to je nad mé možnosti a rozsah článku. Problém je to zajímavý i u webových stránek (řeší např. PHP, JavaScript ad.).

7. Možná vylepšení a posuny kupředu

- Rád bych našel stabilní a bezdrátovou komunikaci mezi notebookem a Arduinem, ale teoreticky i mezi několika Arduiny, případně notebookem a více Arduiny (zvažuji RF i WiFi: XBee, ESP8266 či Slavko Fedorik radí HC-11, možná zkusit přímo Arduino Uno WiFi či Tiān, WeMos D1, MKR1000, nRF24L01+ či RS485). To by mi pomohlo s praktickým zasítováním jeviště. Na druhý konec bych si dal Arduino a světla bych nemusel síťovat každé zvlášť ze svého technického koutku, kde mám zatím jediné Arduino. WiFi zní jako přípustné řešení, ale bude to chtít asi vlastní WiFi router, protože místní WiFi se nezdá stabilní. Zvlášť je ve spodním patře pod jevištěm mikrovlnná trouba jako součást kuchyně a zvuková aparatura jako součást restaurace.
- Testy se spuštěním animací (AV) za sebou nemám. To zatím řeším dalším notebookem, který má na starosti jen to. Spustit VLC či MPV někde bokem by mělo jít. AV výstup používám na LCD monitor jako nápovědka s texty.
- Podobně jsem nepostoupil s testy u Vixen Lights (www.vixenlights.com), programu, který by měl usnadnit přiřazení hudby a světel. Na YouTube je to oblíbené téma u vánočních světýlek (viz Scott Shaver).
- Jistou možnost dává program PD (www.puredata.info, Pure Data), ale to chce hlubší studium, viz například kniha Jan Kavan: Pure Data: platforma pro tvorbu interaktivního díla, ISBN 978-80-7460-033-3.
- Největší plány mám u světel. Mám za sebou první úspěšné pokusy u stmívačů (IGBT-N i BT136/BT139; 60 i 200W žárovky), u toho si chci vyzkoušet přípravu plošných desek s následným osazením. Cesta přes IGBT bude o něco lepší skrz programování (nedohledávání přechodu přes nulu) a možná s menším stupněm rušení. V hledáčku mám HGTG30N60A4D, ale musí se to koupit přes AliExpress či eBay, u nás je to drahé. To bude chtít inovovat jednobajtové sekvence jako příkazy, protože potřebujeme zaslat nejen kanál, ale i intenzitu (0 až 255, tedy dva bajty). Je dost možné, že bude víc Arduin, tak se bude muset i upřesnit, které Arduino příkaz vykonává.
- Rád bych zahrnul spotřebiče ovládané přes IR dálkové ovladače. Například RGB LED reflektor, který je takto řízen. Signály z ovladače dokáží číst, ale nedaří se mi je zpátky vyslat.

- Rád bych vyzkoušel RGB LEDkové pásky. Každý barevný kanál se řídí zvlášť, takže se na to jde trochu jinak než u obyčejných žárovek. Dává to však naději, že divadlo bude konečně trochu řádně osvětlené (tlumené světlo, barva, podsvěcování užších ploch).
- Komunikace a řízení přes protokol DMX512 by mělo usnadnit práci s RGBW kanály. Převodníky mám vyhlídlé dva (OpenDmx a místní firmu SOH.cz).
- Můj největší plán je vyzkoušet si práci s RGB LED čipy, například s WS2812B či dražšími APA102. Doplnkem by měly být silné XML T6, SMD či COB LEDky, ze kterých bych rád udělal mobilní reflektory ve stylu baterek či čelovek. To by snad pomohlo i se zajištěním barevných filtrů. Velikost by dala možnost napojit krokové motorčky či servomotorčky a rotaci takového reflektoru, to je aktuálně hudba možné budoucnosti.

Závěrem

Mohu říci, že jsem vlastně spokojen, během roku jsem projekt dostal do fáze, že jej mohu používat a celý proces není pro mne černá skříňka. Respekt z nízkého (ale potenciálně smrtelného napětí) je velký, ale během bastlení mám vše vypnuté a proud zapínám až jsem hotov. Testy komunikace jsem prováděl a sledoval LEDky u relátek, to stačilo. To se obvykle dělá tak, že vedle produkčního Arduina máte další bez zapojených spotřebičů. Já to dělal tak, že jsem si zapojil úplně vše, ale přívodní kábl zásobující spotřebiče jsem neměl v zásuvce, tedy celou soustavou probíhalo maximálně 5 V=.

Velice zajímavá situace by nastala, kdyby některé z relátek odešlo, to by se dalo řešit paralelním napojováním více relé, nebo na T_EXové úrovni předefinováním výstupu. Tak hluboko jsem se v projektu nedostal a prakticky bych asi přehodil žílu na desce s relátky z nefunkčního na volné relátko.

Zklamání je z těch malých 20cm káblů u Arduina. Na druhou desku s relátký od Arduina to mám vedené nějak přes prodlužky, ale není to ono. Jakmile se pustím do stmívačů a dalších projektů, logistika káblíků bude taky důležitá.

Jiří Rybička mě potěšil, že by měl rád nasvícená kolejiště. To by mohl být zajímavý projekt pro nejbližší období. Budiž světlo!

Kontaktní adresa

Ing. Pavel Stríž, Ph.D., Nakladatelství Martin Stríž, U Škol 940, 685 01 Bučovice, Česká republika,

E-mailová adresa: pavel@striz.cz