



Středoškolská technika 2018

Setkání a prezentace prací středoškolských studentů na ČVUT

Výukové Shieldy pro Arduino

Miroslav Voborský

VOŠ a SPŠ Jičín
Pod Koželuhy 100, Jičín

Anotace

Cílem mé práce byl návrh a zhotovení tří shieldů na Arduino. První shield měl obsahovat 8x červenou LED diodu s průměrem 5mm, 1xfotočidlo, 4x mikropínač, 1x potenciometr. Druhý shield měl být vytvořený do křižovatky s periferiemi 4x trojice LED R,G,Orange, 4x mikropínač. Třetí shield měl mít 6x sedmsegmentový displej, 2x mikropínač, 1x senzor teploty a 1x bzučák. Všechny tři shieldy bylo nutné po zhotovení ještě naprogramovat pro demonstraci funkčnosti každého z nich.

Annotation

The aim of my work was to design and manufacture three shields on Arduino. The first shield should have 8x red LEDs with a diameter of 5mm, 1xphoto, 4x microswitch, 1x potentiometer. The second shield was to be created to junction with peripherals 4x triple LED R, G, Orange, 4x microswitch. The third shield was 6x seven-segment display, 2x microswitch, 1x temperature sensor and 1x buzzer. All three shields had to be programmed to make a demonstration of the functionality of each of them.

Obsah

Úvod	1
1 Návrh řešení	2
1.1 Návrh řešení 1. shieldu	2
1.2 Návrh řešení 2. shieldu	3
1.3 Návrh řešení 3. shieldu	3
2 Teoretická část	4
2.1 Arduino.....	4
2.1.1 Co je to Arduino ?	4
2.1.2 Platforma.....	4
2.1.3 Příklady nejznámějších Arduino Shieldů.....	5
2.1.4 Vývojové prostředí	6
2.1.5 Historie.....	7
2.2 EAGLE	7
3 Praktická část	8
3.1 Tvorba v programu EAGLE.....	8
3.1.1 Proč právě Eagle ?	8
3.1.2 Verze programu a jeho stažení	8
3.1.3 Vytvoření nového projektu	9
3.1.4 Základní funkce funkčních panelů.....	10
3.1.5 Knihovna.....	12
3.1.6 Tvorba schéma zapojení (schematic)	15
3.1.7 Tvorba desky (board)	15
3.2 Shield 1	20
3.3 Shield 3.....	26
4 Závěr	30
4.1 Zdroje.....	32

Úvod

Svou maturitní práci jsem si vybral z toho důvodu, že jsem si chtěl vyzkoušet práci s arduinem, které mě již dlouhou řadu let lákalo, a též mě motivoval k práci fakt, že vyrobené shieldy budou sloužit jako výukové materiály pro žáky nižších ročníků, protože vím, že já bych to jako žák velice ocenil. Smyslem výukových shieldů je ulehčení práce se zapojováním součástek a díky své přehlednosti

a jednoduchosti pomáhají pochopit základní funkce programování v Arduinu.

Návrhy a zapojení shieldů byly vytvořeny v programu Eagle 7.7 pod dohledem vedoucího individuální maturitní práce a vyrobeny firmou Printed.cz. Od každého typu shieldu bylo zakoupeno 10 kusů. Veškeré náklady na shieldy i potřebné součástky byly hrazeny školou.

1 Návrh řešení

K návrhu 3 shieldů se zadanými požadavky, které budou sloužit k výuce, jsem se rozhodl použít program Eagle. Učinil jsem tak, protože je Eagle dle mě přehledný, dlouhodobě se zabývá editací plošných spojů, což je přesně to, co potřebuji a je v základní verzi ke stažení zdarma. Každý shield bude o velikosti 68.6mm x 53 což je přesná velikost Arduina Uno. Pro úsporu místa a přehlednost na shieldu jsem přemýšlel o SMD součástkách a lepším řešení zapojení sedmsegmentovek v podobě kontroléru displeje MAX7219, který by umožňoval zapojení 6ti- segmentovek pomocí 3 pinů na ovládání, ale vzhledem k tomu, že shieldy mají sloužit pro výuku žáků a žáci jistě ocení, když uvidí co všechno vlastně shield obsahuje a s čím pracují, bude lepší použít normální součástky a případný přehled či nedostatek místa řešit dvouvrstvým plošným spojem a sedmsegmentovku zapojit pomocí multiplexu. Na doporučení vedoucího maturitní práce používám k jednotlivým návrhům pro funkce shieldů online Arduino simulátor od společnosti ThinkerCad, kde zapojuji a programuji simulace pro demonstraci a ověření funkčnosti shieldů při konečné prezentaci.

1.1 Návrh řešení 1. shieldu

Pro využití veškerých komponentů na shieldu, jsem například vymyslel, že by se dle fotočidla mohli rozsvěcet červené LED diody a to tak, že při tmě by se diody rozsvítily a při světle zhasnuly. Budou dány hranice vlnových délek, podle kterých budou LED diody buď ve stavu HIGH (rozsvícení) nebo LOW (zhasnutí). V případě rozsvícení by se intenzita svitu dala ještě regulovat potenciometrem. A každým mikrotlačítkem by se ovládalo vypnutí či zapnutí 1 dvojice LED diod. Jako fotočidlo navrhuji použít fototranzistor TEPT4400 s vlnovou délkou $\lambda=570$ nm, což je střed záře denního světla. Dále budou použity LED diody dle zadání (5mm) a potenciometr s vertikálním zapojováním pinů kvůli snazšímu zapojení, lepší přístupnosti a pohodlnému regulování diod. Potenciometr i fotočidlo budou zapojeny do analogu, kvůli čtení hodnot v SerialMonitoru a následnému pracování s nimi.

1.2 Návrh řešení 2. shieldu

Na 2. Shieldu bych znázornil křižovatku, u které by byly na každé straně 3 Diody a to v pořadí barev semaforu (R, O, G). Tudíž bych navrhoval naprogramování semaforů:

a) automatické svícení - Vždy by svítily na stejno dva semaforey na proti sobě a vzájemně se střídaly s dvěma zbylými jako je tomu na reálné křižovatce.

1. tlačítko na zapnutí 2. na vypnutí (2nevyužité).

b) individuální svícení - Vždy by se rozsvítil ten semafor, u kterého by sezmáčklo mikrospínač. Tlačítka by se každý semafor dal i vypnout, tudíž by se při správném pořadí zmáčknutí tlačítek dalo nasimulovat i předešlé řešení

1.3 Návrh řešení 3. shieldu

První 3 sedmissegmentové displeje budou zobrazovat teplotu a druhé 3 sedmissegmentové displeje odpor na teplotním čidle, tak bude vlastně dobře znázorněn princip, na kterém teplotní čidlo funguje (změny odporu-změny teploty). Bzučák se rozezní po překročení nějaké hodnoty teploty - 1. tlačítkem jej lze umlčet a druhým (pokud je teplota stále překročena) opět rozeznít. Zapojení sedmissegmentových displejů bude řešeno přes multiplex, tzn., že bude 7 pinů (V případě tečky 8) pro segmenty a 6 pinů na povely pro každou z nich. Tlačítka budou na analogu. Bzučák buď na 14. pinu, nebo též na analogu v případě zapojení tečky a určitě budu volit bzučák s generátorem, protože nebude potřeba řešit rozkmitání. Teplotní čidlo jsem zvolil TMP36, které půjde na analogový vstup.

2 Teoretická část

2.1 Arduino

2.1.1 Co je to Arduino ?

Arduino je název pro jednodeskový počítač, který funguje na bázi mikrokontrolerů ATmega od firmy Atmel. Slouží k podpoře informatiky ve školách a jeho podstatou je seznámit studenty s tím, jak jsou pomocí počítačů řízena různá zařízení (např. mikrovlnná trouba, automatická pračka a jiné stroje). Nejedná se tedy o počítač ve smyslu stolního počítače nebo chytrého telefonu. Nelze proto k němu snadno přímo připojit monitor ani klávesnici či myš, ale je připraven na připojení LED diod, displeje z tekutých krystalů, servomotorů, senzorů tepla či světla, osvětlení atd.

2.1.2 Platforma

Desky arduino obsahují 8bitové mikrokontrolery Atmel a množství dalších podpurných obvodů. Oficiální Arduina jsou vyráběny s čipy ATmega8, ATmega168, ATmega328, ATmega1280 a ATmega2560. Arduino Due obsahuje 32bitový ARM procesor Atmel SAM3X8E. Každá deska má většinu I/O pinů přístupných přes standardizované patice, do kterých se jednoduše připojují další obvody, kterým se ve světě Arduina říká Shildy. Na deskách bývá několik diod, resetovací tlačítko, konektory pro programování, napájecí konektor, oscilátor a obvod zprostředkovávající komunikaci po USB. Základní verze Arduina, Arduino Uno, poskytuje celkem 14 I/O digitálních pinů a 6 pinů analogových. Programy do Arduina se píšou v jazyce podobném C/C++. Ačkoliv je Arduino připojeno k počítači pomocí rozhraní USB, je softwarově simulována sériová komunikace přes linku RS-232.

2.1.3 Příklady nejznámějších Arduino Shieldů

Ethernet Shield

Tento shield umožňuje připojit Arduino k internetu. Samotný shield obsahuje integrovaný obvod W5100, konektor RJ-45 pro připojení síťového kabelu, slot pro paměťové karty typu microSD, resetovací tlačítko a indikační LED diody signalizující činnost LAN portu. Ethernet shield využívá pro komunikaci s Arduinem piny 4, 10, 11, 12 a 13, přičemž samotná komunikace probíhá pomocí komunikačního protokolu SPI.



Obrázek 1 Ethermet Shield

WiFi Shield

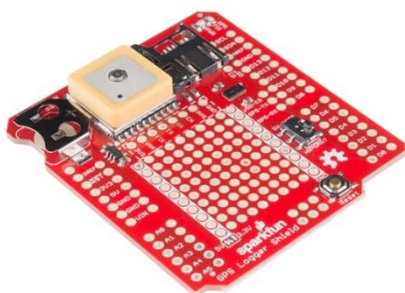
Tento shield umožňuje připojit Arduino k internetu pomocí bezdrátové sítě WiFi se zabezpečením WEP a WPA2. Samotný shield obsahuje slot pro microSD kartu, miniUSB port primárně určený pro aktualizaci firmwaru shieldu, několik signalizačních LED diod a resetovací tlačítko. Wi-fi shield využívá, stejně jako Ethernet shield, komunikační protokol SPI.



Obrázek 2 WiFi Shield

GPS Shield

Umožňuje rozšířit Arduino o GPS modul, kterým lze zjistit aktuální polohu, nadmořskou výšku nebo rychlost pohybu. Velmi snadno se používá, neboť stačí jen přečíst data ze sériové linky a získáme aktuální GPS data. Dále tento shield obsahuje slot pro připojení microSD karty, na kterou si můžeme uložit například získaná GPS data.



Obrázek 3 GPS Shield

2.1.4 Vývojové prostředí

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0.5". The main editor area displays the code for the "Blink" example. The code includes comments explaining the function and the setup/loop routines. The status bar at the bottom indicates "1" and "Arduino Uno on /dev/tty.usbmodem641".

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

Obrázek 4 Vývojové prostředí Arduino

2.1.5 Historie

Projekt vznikl v roce 2005 v Itálii ve městě Ivrea. Jeho cílem bylo vytvořit jednoduchou prototypovací platformu pro studenty, která umožní rychlý vývoj a jednoduché používání. Projekt zaznamenal velký úspěch a později začaly vznikat jeho další, novější verze. Do února 2010 se údajně prodalo více jak 120 tisíc kusů. Projekt Arduino získal ocenění v kategorii digitálních komunit na Prix Ars Electronica 2006. V roce 2010 vyšla, zatím poslední, verze Arduino desky s označením *Uno*, která nahradila svého předchůdce Duemilanove. Od roku 2012 jsou v prodeji desky Due (s procesorem ARM) a Leonardo a také nová revize Una R3.

2.2 EAGLE

EAGLE je aplikace pro automatizaci elektronických návrhů, která umožňuje skriptování, schématické snímání s rozvržení desky s plošnými spoji. Představuje snadno použitelný editor grafického rozhraní a byl vyvinut společností CadSoft Computer GmbH. V roce 2016 ho získala společnost Autodesk Inc. Obsahuje schématický editor pro návrh schémat obvodu. Schémata jsou uložena v souborech s příponou SCH. Součástky jsou definovány v knihovnách zařízení s příponou LBR. Board (deska) je uložena v souboru s příponou BRD. Zhotovené soubory ukládá pod názvem Gerber a PostScript, stejně jako výcvikové soubory Excellon a Sieb & Meyer EAGLE poskytuje grafické uživatelské rozhraní s několika okny a systém menu pro editaci, správu projektů, přizpůsobení rozhraní a návrhovými parametry K sehnání je buď verze zdarma, která je omezena na určitou velikost vytvářeného plošného spoje, nebo placená verze, která vyjde zhruba na 3300kč za rok, nebo 450 Kč na měsíc.

3 Praktická část

3.1 Tvorba v programu EAGLE

3.1.1 Proč právě Eagle ?

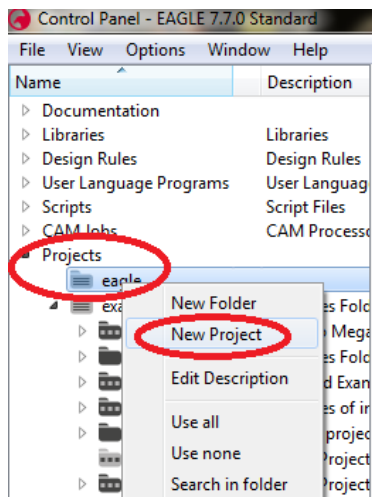
K výrobě výukových Arduino shieldů jsem používal program EAGLE, ve kterém jsem navrhl veškerá schémata i desku plošných spojů. Tento program mi vyhovoval, udělal jsem v něm vše, co jsem potřeboval a nenarazil jsem na žádný jeho nedostatek, nebo funkci, kterou bych mu mohl jakkoliv vytknout. Další jeho velká výhoda je celosvětová známost, což se značně odráží na širokém výběru knihoven a na nejrůznějších schématech i pro méně známé součástky, které jsou vytvořeny a zdarma na internetu ke stažení. Proto bych vám ho rád představil a popsal, jak v něm něco takového vytvořit. Program je pro začátečníka z počátku složitý, avšak nezalekněte se! Po zjištění většiny funkcí se jedná o vsukutku propracovaný, přehledný a dobře optimalizovaný program, který se pohodlně ovládá. A i já jako samouk, jsem to za nějaký čas zvládl, proto se není čeho bát, pusťme se do toho.

3.1.2 Verze programu a jeho stažení

Zvolil jsem verzi EAGLE 7.7, která je sice starší, ale pozor, starší neznámá horší. Jsou zde všechny funkce, které budeme potřebovat a naopak nejsou zde ty, které bychom nevyužili, což dělá celé prostředí volnější a pro uživatele přehlednější, tudíž jasná volba. Nám bude stačit základní verze programu, která je ke stažení zdarma na oficiálních stránkách eaglu: <http://www.eagle.cz/>. Předpokládám, že instalaci programu netřeba zmiňovat a pusťme se do dalšího kroku.

3.1.3 Vytvoření nového projektu

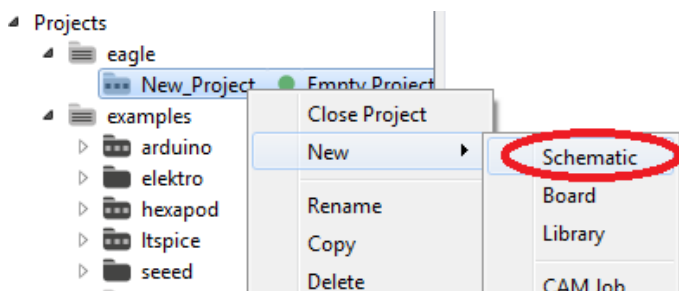
Pro vytvoření nového projektu otevřeme záložku "Projects" a pravým tlačítkem klikneme na "eagle", poté zvolíme 2. možnost "New Project".



Obrázek 5 EAGLE - Nový projekt

Náš nový projekt si libovolně pojmenujeme. Napravo od nově vytvořeného projektu nám bude svítit zelená signalizace, která značí stav našeho projektu, pokud je zelená, je to v pořádku a znamená to, že je aktivní, zabrání to například tomu, aby byl projekt smazán a jasně tím dáváme programu najevo, že právě s tímto projektem pracujeme.

Poté si pravým kliknutím na náš projekt vytvoříme prostředí pro návrh schémata.



Obrázek 6 EAGLE - Nové schéma

Nyní se nám otevře nové okno, kde budeme navrhovat pomocí součástek z knihoven náš obvod. POZOR! V žádném případě nesmí dojít k zavření počátečního Control Panelu, nebo se vypnou všechny zapnuté části Eaglu a může tak dojít ke ztrátě dat.

3.1.4 Základní funkce funkčních panelů.

Nyní si ukážeme pár základních funkcí, které se nám budou hodit a o kterých je dobré vědět. Funkce, které si nyní budeme popisovat, se nachází v panelu nalevo.

Začneme postupně.

Každou funkci si zvolíme levém kliknutím myši a poté opět levém aplikujeme na vybranou součástku.



Info zobrazí veškeré informace o součástce.



Show slouží k orientaci ve schématu, dvojklikem zobrazí vybraný spoj světle zelenou barvou, užitečné při hustém a nepřehledném zapojení, či při pouhé kontrole zapojení.



Move přesouvá součástky a spoje. Při přesouvání součástek je potřeba kliknout na její křížek $+$ a táhnout.



Copy okopíruje součástku nebo vybranou skupinu a automaticky jim přidělí číselné označení, které následují v pořadí.



Mirror dokáže zrcadlově převrátit součástku.



Rotate otáčí do 4 směrů vybraný objekt.



Group seskupí větší počet částí schémata dohromady a lze je poté upravovat jinými funkcemi.



Paste vloží zkopírované části obvodu.



Odstraní označenou část.



Add přidá vybranou součástku z knihovny.



Pinswap dokáže vyměnit dva zvolené piny, výměna proběhne i v případě již zapojeného pinu.




Replace v případě vizuálně a funkčně podobné součástky ji vymění za původní. Užitečné pro opravu například špatné rozteče odporu.




Name pojmenuje součástku, například pojmenuje odpor R2.




Value přidá hodnotu součástce, například bude mít odpor 120 Ω .


 Invoke vhodná funkce pro použití zalomení či šikovné posunování spoje.

 Wire vytvoří spojení čarou mezi dvěma součástkami (znázorňuje drát).


T Text slouží pro popsání čehokoliv v obvodu.

 Junction vytvoří spoj mezi dvěma vedenými dráty.

  Errors je výpis veškerých nalezených chyb v obvodu.

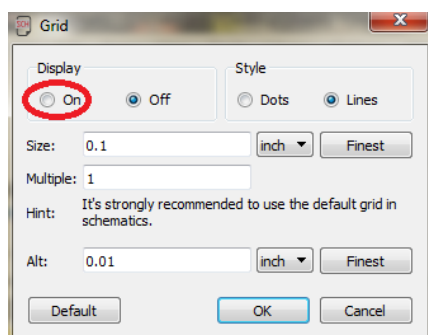
 Generate / switch to board vytvoří z našeho navrženého obvodu návrh na desku plošných spojů, opět se otevře nové okno.

Po tomto kroku budou otevřena 3 okna (základní control, schéma a deska)

 Library zobrazí nebo přidá seznam knihoven.


 Grid je možnost zobrazení pomocných čar v celém schématu.

Hustotu a velikost čar lze libovolně měnit. Tuto funkci doporučuji zapnout, práce je pak přehlednější a člověk vidí, kam se vedený drát chytne, a když je potřeba umístit drát jinak, hustotu zjemní. Zapnout Grid lze po stisknutí ikony a převedení stavu z "off" na "on".



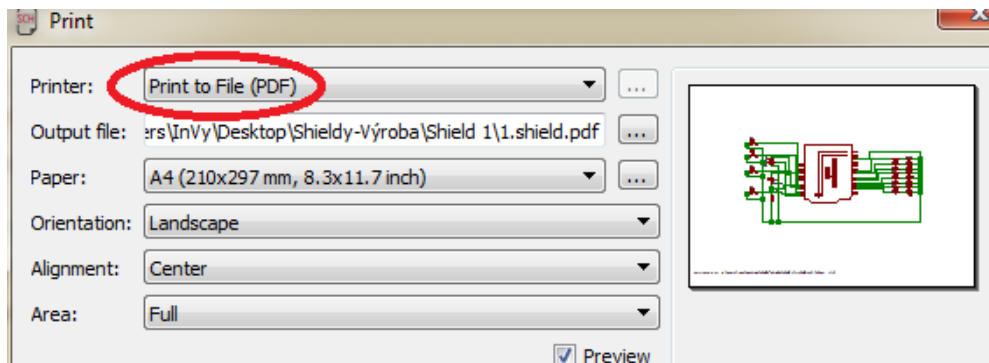
Obrázek 7 EAGLE - nastavení mřížek

Velikost (size) si nastaví každý, jak mu to vyhovuje, doporučuji ideální rozměr 0.1 v palcích (inch). Poté námi vybrané nastavení potvrdíme.

 Save uloží aktuální stránku, doporučuji ukládat vše průběžně.

 Print slouží k vytisknutí souboru do fyzické či virtuální podoby.

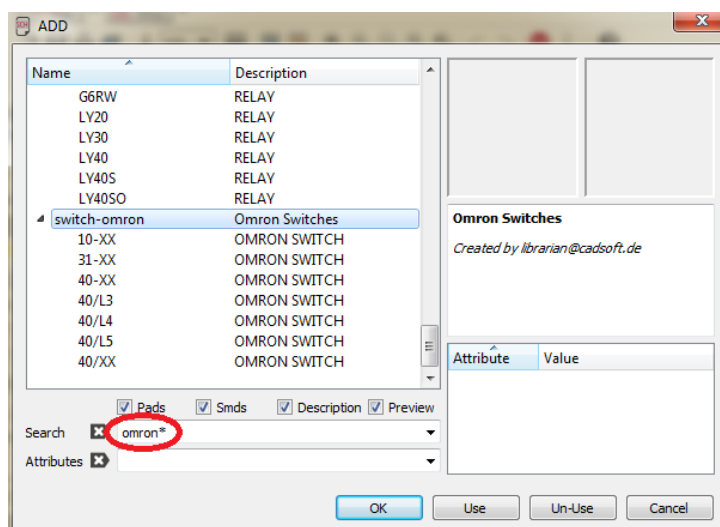
V případě tisku do virtuální podoby je potřeba vybrat "Print to file" v poli "Printer", soubor se vytvoří ve formátu PDF. Poté se uloží do vybraného adresáře.



Obrázek 8 EAGLE - tisk

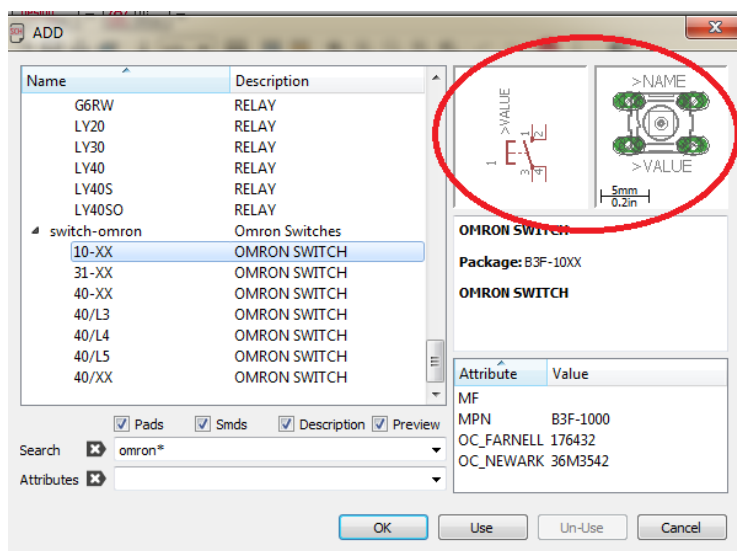
3.1.5 Knihovna

Knihovnu v programu EAGLE si můžeme představit jako soubor plný součástek s kompatibilními schématickými značkami a zobrazením v prostředí desky. Proto už teď je jasné, že se bez toho neobejdeme. Po otevření záložky Add vybíráme součástky právě z knihoven. V případě že k vyhledání součástek či knihoven, použijeme vyhledávač (search) a my nevíme, jak zní přesný název, napíšeme část slova, kterým jsme si jisti a zbytek stačí doplnit znakem " * ", poté dojde k vyhledání.



Obrázek 9 EAGLE Knihovna - vyhledání

Nyní nám už nic nebrání k tomu si součástku vybrat a libovolně ji umístit do našeho obvodu. Při rozkliknutí knihovny a zvolení konkrétní součástky se nám na pravé straně zobrazí její parametry a náhled schématické značky a reálné zobrazení na desce na které je potřeba se zvýšenou pozorností dbát.



Obrázek 10 EAGLE knihovna - součástky

Jakmile si EAGLE stáhneme, je v něm umístěno nespočet základních knihoven v angličtině a některé z nich je výhodné použít. V následující tabulce č. 1 si ukážeme, které z nich to jsou a co v nich najdeme.

Tabulka 1 Seznam užitečných knihoven

Knihovna	Součástky
Rcl	Odpor, Kondenzátor, Cívka
switch-omron	mikrospínač
transistor-npn	Tranzistor NPN
transistor-pnp	Tranzistor PNP
speaker	bzučák
Piher	potenciometr
Led	LED dioda

Co ale dělat v případě, že nemůžeme nějakou z potřebných součástek najít, nebo potřebujeme konkrétní typ, který víme, že se v základním seznamu knihoven nenachází? Přidáme knihovnu.

Kde sehnat knihovny?

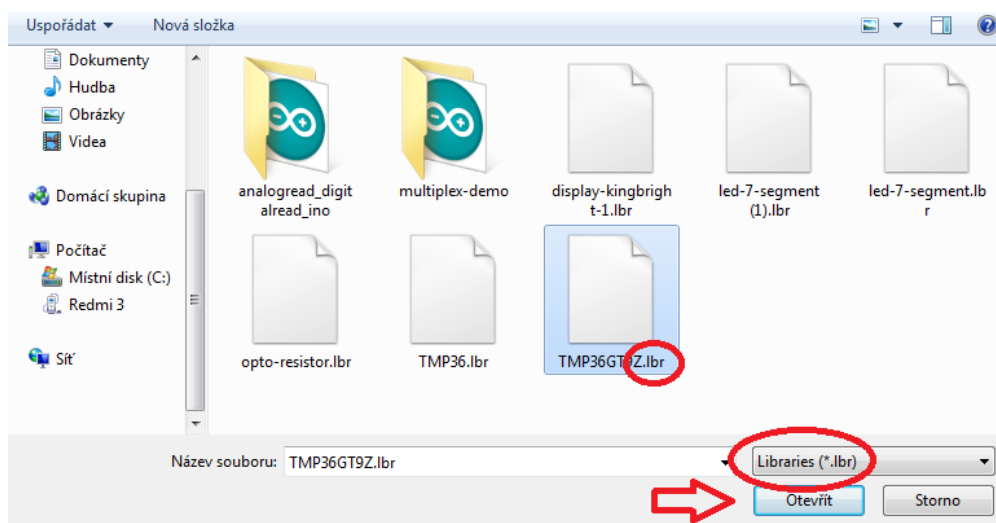
Každou knihovnu, kterou chceme do EAGLU přidat, musíme nejprve stáhnout a pro úspěšné přidání musí být ve formátu library, tedy s koncovkou LBR. Mně osobně se osvědčila knihovna Adafruit, kterou je možné stáhnout z odkazu <https://github.com/adafruit/Adafruit-Eagle-Library>, která obsahuje mimo jiného i Arduino shieldy. Dále bych doporučil PaJa knihovny, které obsahují veškeré potřebné základní elektro součástky a jejich názvy jsou v češtině. Tyto knihovny je možné stáhnout z odkazu <http://paja-trb.cz/eagle/index.html>. Dále existují webové stránky na výhledání jednotlivých komponent jako například: DIYModules.org, SnapEda a github. Veškeré stahování knihoven je zdarma s občas nutnou registrací.

Jak přidat knihovnu ?

Pokud máme již staženou knihovnu, lze ji přidat kliknutím na ikonu



"Use library" v horním panelu a otevře se nám adresář. Nyní najdeme umístění stažené knihovny a zvolíme ji.

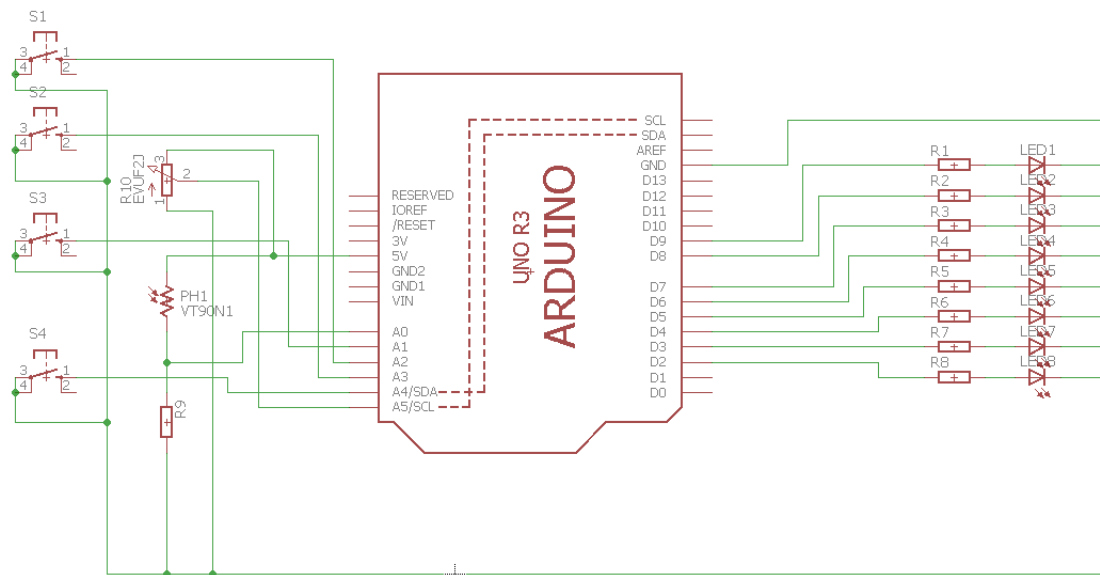


Obrázek 11 Knihovna - přidání

Teď máme knihovnu přidanou a zobrazí se nám mezi ostatními v adresáři Add.


3.1.6 Tvorba schématu zapojení (schematic)

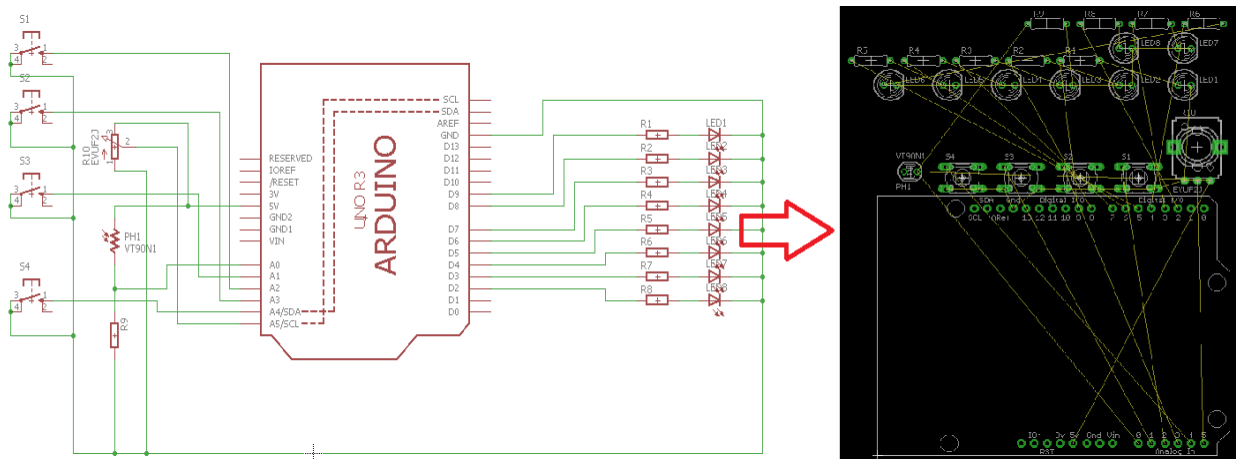
Nyní jsme si řekli vše potřebné k vytvoření našeho obvodu v předchozích krocích. A nic nám nebrání v tom zhotovit naše dílo. Mé první navržené schéma vypadalo následovně:



Obrázek 12 EAGLE - Tvorba schématu

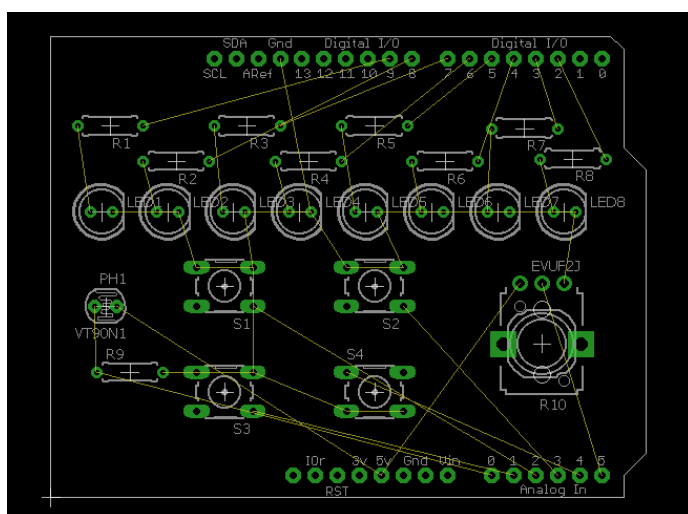
3.1.7 Tvorba desky (board)

Při dokončení schéma zapojení se přesouváme k dalšímu kroku. Tím je vytvoření desky plošného spoje z našeho vytvořeného schéma zapojení. Kliknutím na ikonu  generate / switch to board v horním panelu, vyskočí dotaz, zda chceme k tvorbě desky použít naše schéma zapojení, ten potvrdíme. Poté se nám otevře nové okno se součástkami, které máme ve schéma zapojení a jsou žlutými čarami propojeny tak, jak jsme je zapojili ve schématu.



Obrázek 13 EAGLE - Tvorba desky


Nyní si srovnáme součástky na navrhnutou desku tak, jak nám to vyhovuje a zároveň dbáme na to, aby každá součástka byla co nejbližší ke svému pinu a nám se pak snáze dělali cesty spojů. Princip přesunů součástek zůstává stejný jako v předchozím případě schéma zapojení, kde jsme zapojovali obvod, avšak je potřeba myslet na to, že každá součástka musí být uchycena za křížek. Piny se nesmí překrývat a veškerý skládaný obsah musí být v poli, které je vyznačeno čtvercem. Výsledek může vypadat zhruba takto:

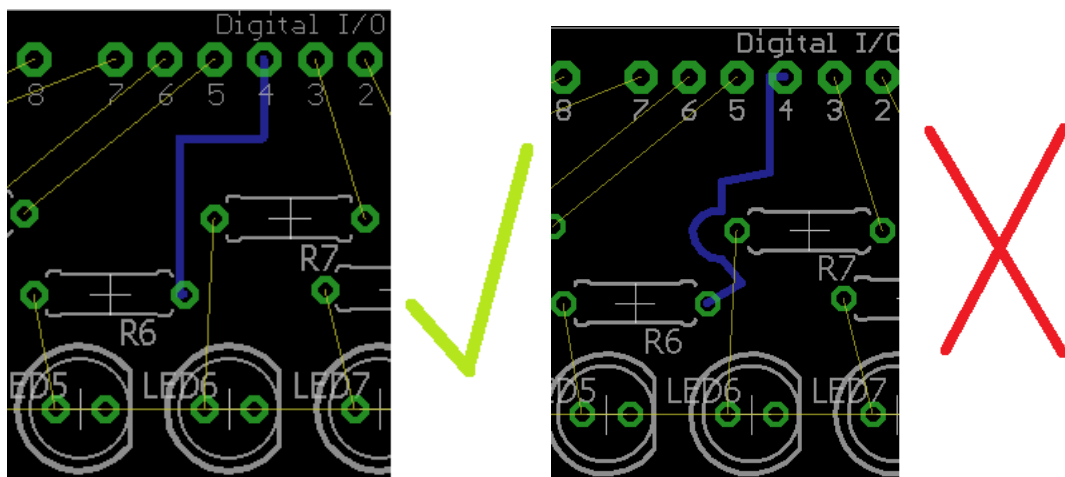


Obrázek 14 EAGLE - Umístění součástek na desce

Abychom mohli pokračovat, opět si vysvětlíme pár základních funkcí, které nám zde v levém panelu přibyly a bez kterých se v žádném případě neobejdeme. To hlavní o co nám teď půjde, bude vytvoření cest na desce a proto tím začneme.

Vedení cest na desce

Pokud chceme vytvořit novou cestu, klikneme na ikonu  Route a poté ji vedeme od počátku do konce, dle žluté pomocné čáry. Jak ji povedeme je na nás, ale je dobrým zvykem vést spoje pravými úhly.





Obrázek 15 EAGLE - Vedení cest na desce

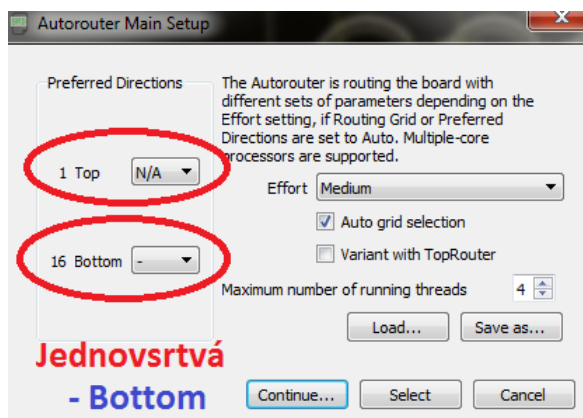
Při vedení cest je potřeba si pečlivě rozmyslet, zda ji chceme vést pouze na spodní straně desky (=Bottom), která je z pravidla značena modře, nebo chceme dvouvrstvou desku a cestu budeme vést i na horní straně (=Top), která je značenou červenou barvou. Mezi vrstvami, do kterých cestu vkládáme, si volíme v levém horním rohu po zvolení funkce Route:





Pro změnu velikosti a tvaru vedené cesty, využijeme horní panel, který se zobrazí v případě, že máme vybranou funkci Route.




V případě, že děláme dvouvrstvou desku a není jiná možnost, než udělat propoj (vodivá díra, která propojuje bot s topem) použijeme funkci Follow me  umístěnou v horní liště. Poté vedeme spoj, kam chceme a propoje se automaticky udělají samy. Další užitečnou funkcí, která mnohdy ušetří hodně práce a času, je funkce Autorouter  dokáže sama navrhnout veškeré cesty na desce. Při použití Autorouteru je třeba nastavit typ vedené cesty, pokud má být jednovrstvá, bude ve vrstvě Top napsáno "N/A" (vypnuto), pokud dvouvrstvá bude u každé z nich pomlčka " - " .



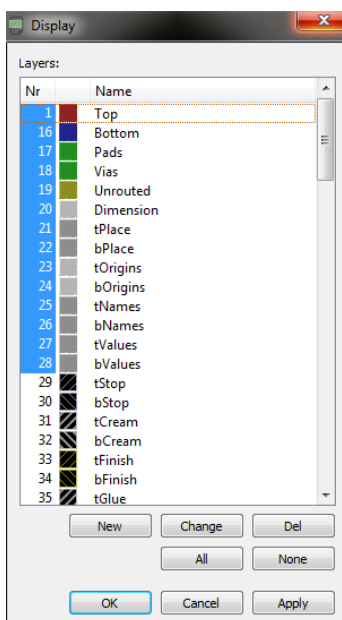
Obrázek 16 EAGLE - Nastavení Autorouteru

Pokud výběr potvrdíme a odstartujeme autorouterování, navrhne nám to různé možnosti vedení cest, zvolíme si tu, která nám přijde nejvhodnější, a výběr dokončíme. Na první pohled se zdá tato funkce ideální, bohužel při složitějších schématech se mi neosvědčila a zvolil jsem raději ruční způsob Routování. Pokud zjistíme, že nám vedená cesta nevyhovuje a potřebujeme ji z nějakého důvodu upravit nebo ji vést jinudy, využijeme funkci Ripup . Ta dokáže vedenou cestu zrušit a opět tam zůstane jen pomocná žlutá čára. První klik slouží k označení cesty. Druhý klik smaže vybranou část a třetím kliknutím smažeme celou větev té cesty a vrátíme jí tak do původního stavu. Pro kontrolu a optimalizaci vedených cest na desce využijeme funkci Ratsnest  do levého dolního rohu vypíše zbývající počet žlutých čar a smaže je na místech, kde již nejsou potřeba. Zde v editoru desky, je obzvlášť potřeba dát si pozor na výpis chyb z DRC, protože program odhalí fatální chyby, jako jsou například zkratky, se kterými by obvod nefungoval.

Vrstvy

Další důležitou součástí desky jsou vrstvy, ve kterých je zapsán různý text, označení, hodnoty, obrysy schémat a podobně. Vrstvy se dají vypnout, zapnout a podle toho zjistíme, co do jaké patří. Nastavení vrstev, které na desce máme, spustíme ikonou  Layers settings.

Poté se nám otevře přehled, kde jednotlivé vrstvy vidíme. Modré zbarvení před názvem vrstvy znamená, že je zobrazení zapnuté a my ji na desce vidíme. V případě bílé barvy je tomu naopak.



Obrázek 17 EAGLE - Vrstvy

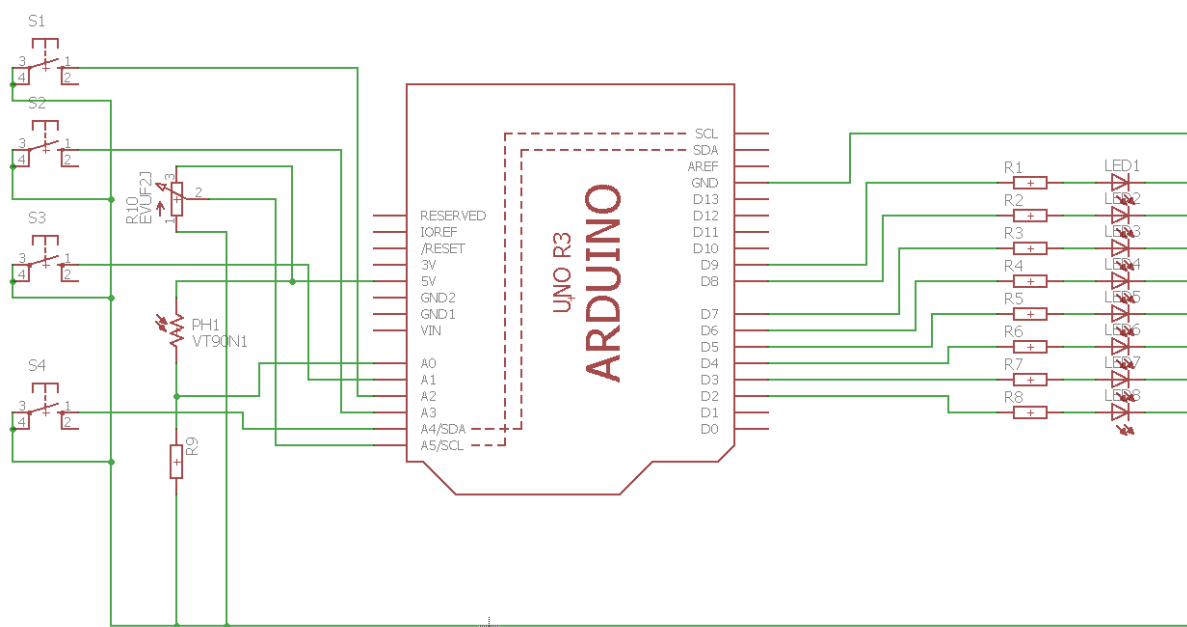
Další důležitou věcí, na kterou nesmíme před finálním řešením zapomenout, je vepsání jakéhokoliv textu (například název školy) do vrstvy bot popřípadě i top, aby výrobci podle zrcadlového otočení textu dokázali určit, o jakou stranu jde.

Výroba

Při výrobě je potřeba si zvolit firmu, která nám to vyrobí. Já jsem si zvolil firmu Printed.cz a mým hlavním důvodem pro tuto volbu byla možnost výroby prokovených děr na desce. Při odesílání materiálů pro firmu, je potřeba odeslat soubor s deskou, tedy s příponou BRD.

3.2 Shield 1

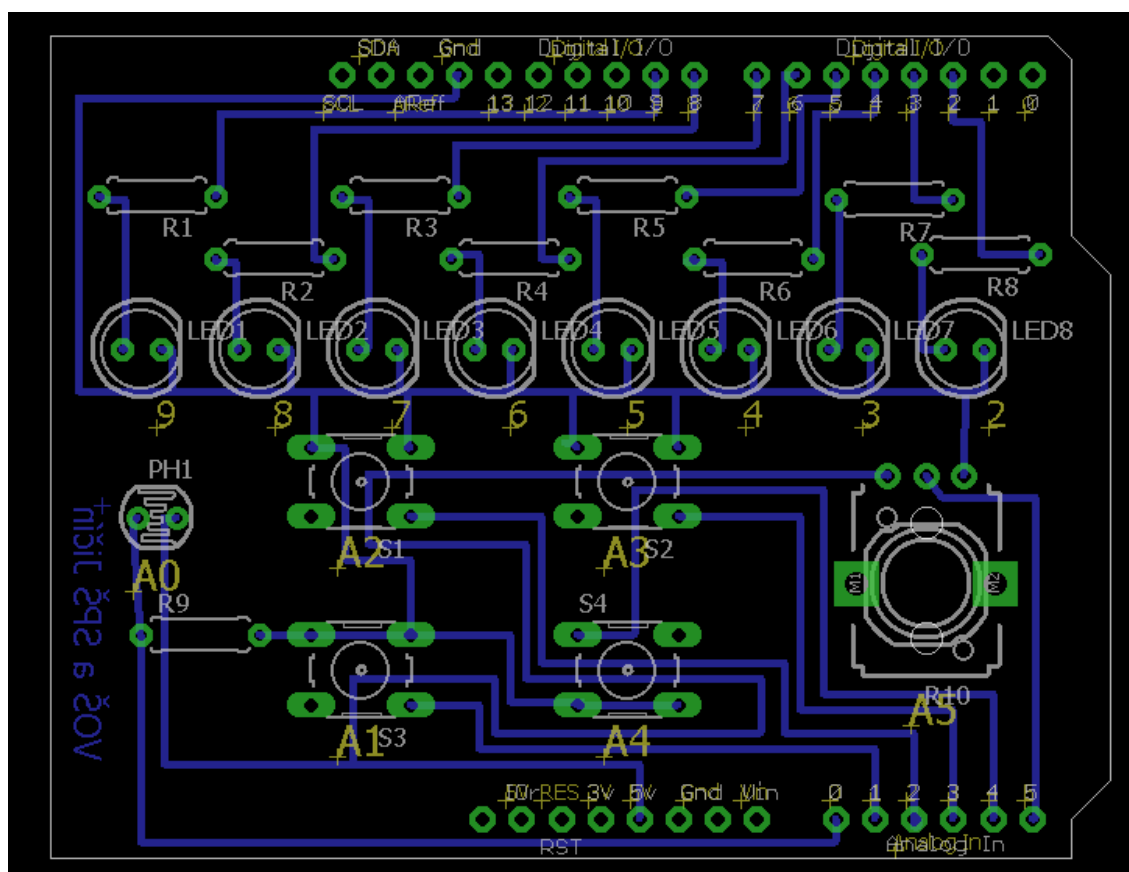
Při tvorbě schéma zapojení prvního shieldu, jsem nejprve našel knihovnu, která obsahovala schéma arduino shieldu a ke kterému jsem to následovně všechno zapojil. Začal jsem zapojením 8x LED diody o průměru 5mm na digitální piny. Ke každé jsem dal odpor bez zadané velikosti, protože odpory jsem chtěl doladit až v realitě dle svítivosti zvolených diod. Každý odpor jsem zvolil se 7mm roztečí, což je běžná velikost odporů. Dále jsem do schémata přidal 4 mikrotlačítka, ke kterým jsem odpor nedával, kvůli úspoře místa na shieldu a rozhodl jsem se použít vnitřní odpor arduina, který se použije přes příkaz INPUT_PULLUP. Všechna tlačítka jsem zapojil na analogový vstup. Poté následovalo zapojení vertikálního potenciometru, který má 3 piny. První pin slouží k na napájení, druhý na ovládání, proto je připojen na analogový pin A5 a poslední pin je zem. Jako poslední byl do obvodu přidán světelný senzor, který jsem zapojil na analogový vstup 0 a při jeho pinu na zem přidal 3kΩ odpor. Veškeré součástky jsem připojil na napětí 5 V a připojil na společnou zem, bylo to výhodné kvůli následnému tvoření cest na desce plošného spoje. Konečné schéma zapojení vypadalo následovně:



Obrázek 18 Shield1 - schéma

Desku na první shield jsem tvořil z navrhnutého schéma zapojení a po domluvě s vedoucím maturitní práce, jsem tuto desku udělal jednovrstvou.

Zpočátku to byl kvůli místu docela problém, ale po srovnání součástek se mi povedlo udělat cesty tak, aby nedocházelo k žádnému zkratu a byla možná i výroba. Do vrstvy tfinish jsem dopsal označení pinů, pro snadnou orientaci programátorů shieldu.



Obrázek 19 Shield1 - Deska

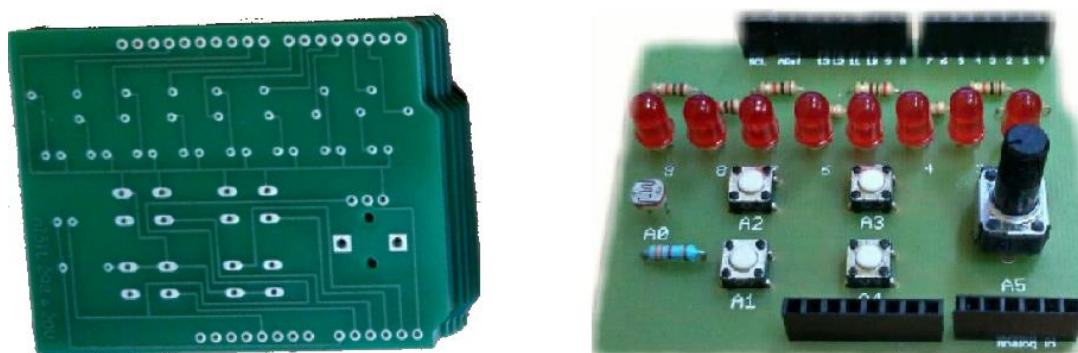
V konečné fázi, kdy jsem osazoval shield součástkami, jsem zvolil červenou barvu všech osmi diod a na každou diodu jsem umístil 1k Ω rezistor, protože já osobně mám radši, když to při práci s tím svítí trochu méně a nepálí to do očí. Pouze k diodě D5, která je určena pro regulaci svítivosti potenciometrem (A5), jsem dal menší 200 Ω odpor, regulování je pak lépe vidět. Potenciometr se otáčí o 180 ° a přesně v půlce má při otáčení cítitelnou zarážku, takže přesně víte, kdy jste v půlce, což považuji kvůli programování za velkou výhodu. Pro osazení shieldu jsem vybral kompatibilní součástky s mým schématem a osadil je následujícími komponenty:

Tabulka 2 Shield1 - součástky

Druh	Typ	Počet kusů
Mikrospínač	B3F-1000	4
Potenciometr	RK09K1130081	1
Fotorezistor	PGM5506	1
LED červená	TLLR5401	8
Odpor 200 Ω	CF1/4w-200R	1
Odpor 1k Ω	CF1/4w-1k	7
Piny k arduinu	Po 8pinech extra prodloužené	4

Tyto komponenty byly objednány 10x na každý ze shieldů tohoto typu. Pro demonstraci funkčnosti prvního shieldu, jsem udělal program, který dokazuje funkčnost a to tím způsobem, že: Každá z prvních čtyř diod, reaguje na jedno ze čtyř tlačítek, rozsvícením a zhasnutím. Pátá dioda je regulovaná potenciometrem A5. A poslední 3 diody reagují na světelný senzor a to tak že čím větší světlo je, tím víc jich svítí a naopak, pokud je úplná tma, nesvítí žádná. Celý program přikládám do přílohy.

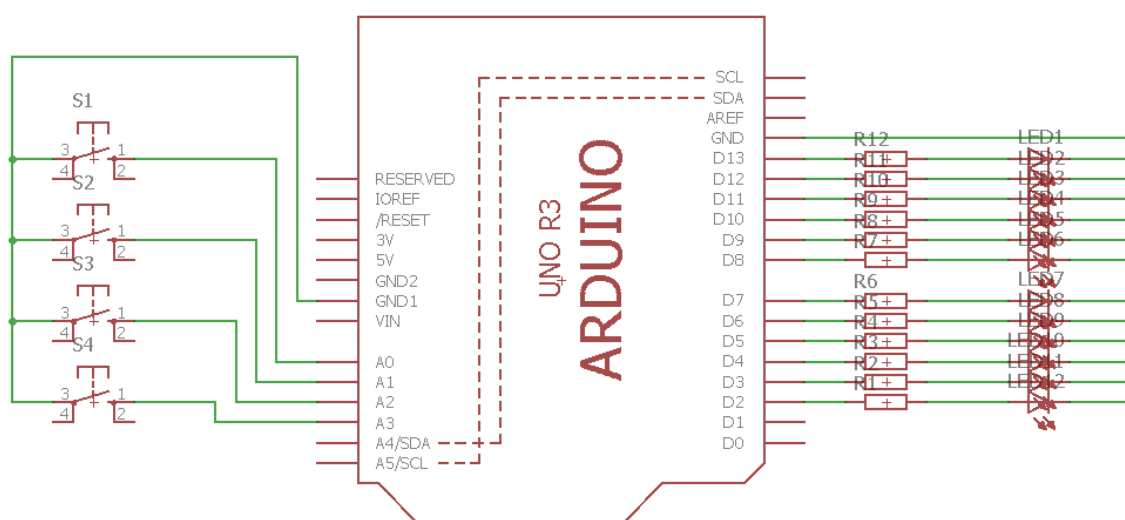
[\[Příloha č.1\]](#)



Obrázek 20 Shield1 - Deska plošného spoje, Osazený Shield

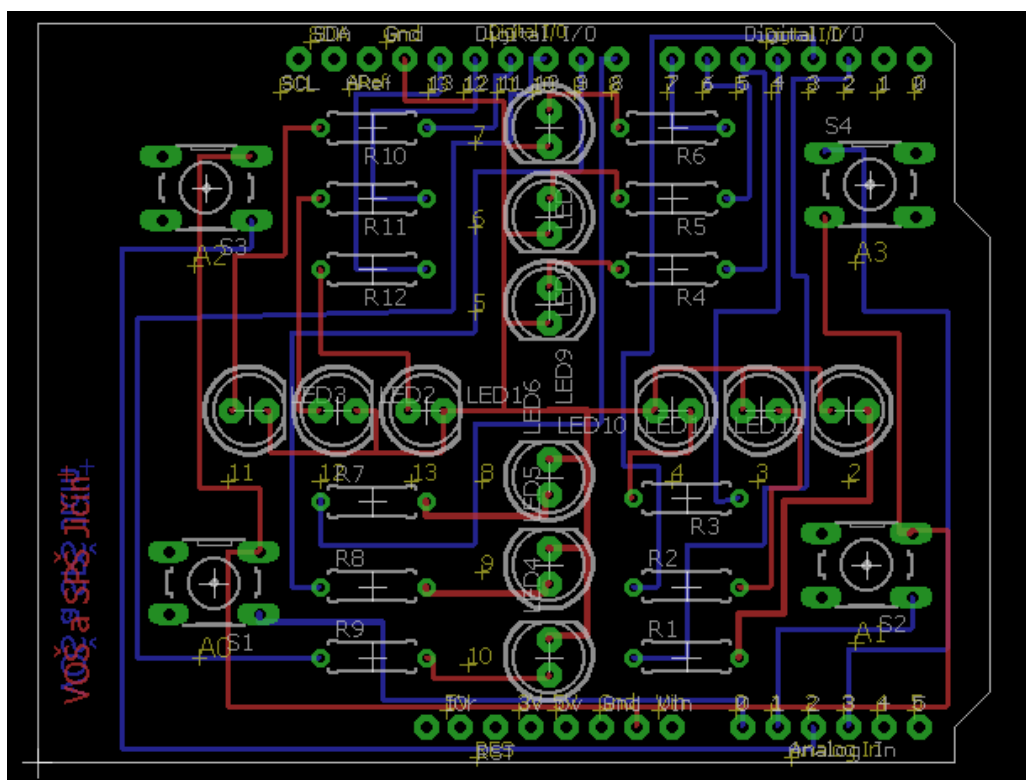
Shield 2

Při návrhu 2. schémata jsem opět jako první věc vložil na pracovní plochu arduino shield z knihovny Afraduit a na to jsem později připojoval veškeré součástky. Postup byl obdobný, nejprve jsem na digitální vstupy zapojil LED diody o průměru 5mm s rezistory, které měly rozteč 7mm. Poté jsem přidal tlačítka na analogové vstupy, jejichž odpor je také řešen Arduino funkcí INPUT_PULLUP. Obvod je připojen na 5V a společnou zem mají diody a mikrosvínače zvlášť.



Obrázek 21 Shield2 - Schéma

O kolik bylo zde lehčí schéma obvodu, o tolik těžší bylo vytvoření cest na desce. Dle zadání jsem se snažil dodržet pomyslnou křížovatku, kterou jsem vytvořil z diod a díky tomu bylo pro mě opravdu složité srovnání komponent a vedení cest na shieldu Arduina. Odporů i tlačítek jsou srovnány tak, aby bylo vidět, k jaké větvi diod patří. Umístění součástek jsem dělal souměrné. Nakonec jsem se bohužel nevyhnul dvouvrstvému řešení desky a routování jsem dělal ručně, protože autorouter tvořil nevhodné a vizuálně nehezské řešení. Také jsem se rozhodl, že zapojení LED diod bude ve směru od středu, v pořadí zelená, žlutá, červená, aby bylo možné odsimulovat žádaný semafor.



Obrázek 22 Shield2 - Deska

I zde jsou u součástek pro lepší přehlednost popsány piny, na které jsou součástky připojeny. Kvůli dvouvrstvé desce je text, ve kterém je uveden název školy popsán ve vrstvě Top i Bottom. Při vybírání vhodných součástek na osazení mého shieldu, jsem objednal všechny barvy diod stejného typu, aby mohl být všude stejný odpor a svítily stejně. Opět jsem zvolil velikost 1 k Ω kvůli mému pracovnímu pohodlí při nižším osvětlení diod.

Tabulka 3 - Shield2 - Součástky

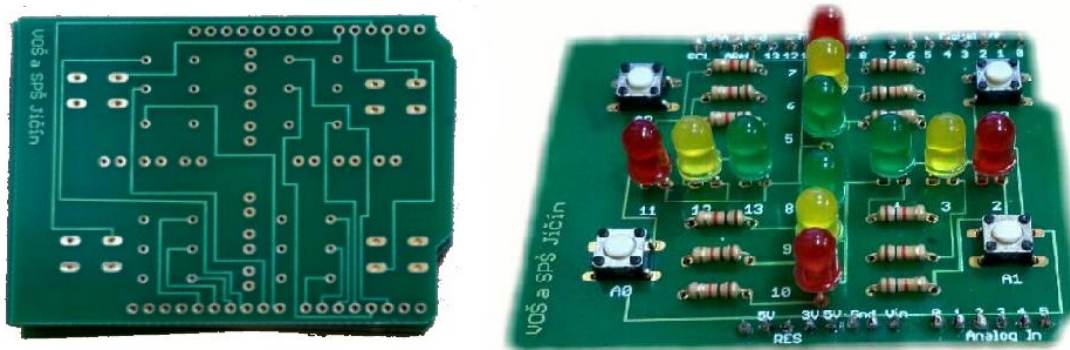
Druh	Typ	Počet kusů
Mikrospínač	B3F-1000	4
LED červená	TLLR5401	4
LED žlutá	TLLY5401	4
LED zelená	TLLG5401	4
Odpor 1k Ω	CF1/4w-1k	12
Piny k arduinu	Po 8 pinech	4

Celý seznam byl objednáno 10x pro každý shield tohoto typu.

Při programování bylo mým úkolem dokázat, že veškeré součástky, které jsou umístěny na 2.shieldu, fungují. Proto jako první věc, kterou jsem zde musel nutně, už kvůli vzhledu naprogramovat, byla simulace semaforů. Tato simulace se spustí po stisku mikrospínače A0. Nejprve se rozsvítí zelené diody, které jsou vertikálně naproti sobě, zároveň s červenými diodami, které jsou proti sobě horizontálně. Poté se na každé větvi rozsvítí žlutá dioda, předešlý stav zhasne. Následuje zhasnutí i oranžových diod a rozsvícení zelených s červenými na vedlejších větvích, což znamená, že budou svítit červené diody na vertikálních větvích a zelené diody na horizontálních větvích. To je poslední stav této simulace na tlačítku A0, poté se opět opakuje. Délka zobrazení červených a zelených diod je nastavena na 2,5 sekundy. Délka zobrazení oranžových diod je 1sekunda. Snažil jsem se co nejvíce napodobit reálnou simulaci, z toho důvodu je stav svícení oranžové diody kratší. Pro vypnutí tohoto cyklu je třeba podržet tlačítko A0 v době, kdy probíhá přeměna z červené diody 11 a 2, na červenou diodu 10 a 7. Poté program dokončí svou začatou scénu a diody vypne. Abych využil a dokázal, že mi fungují i zbylá tři tlačítka A1,A2 a A3, vytvořil jsem následující program, ve kterém má každé tlačítko 3 módy a ovládá jednu barvu diod. Číslo módu je rovno s počtem stisknutí tlačítka.

1. mód diody rozsvítí.
2. mód diody rozbliká s periodou 400 ms.
3. mód diody zhasne.

[\[Příloha č.2\]](#)

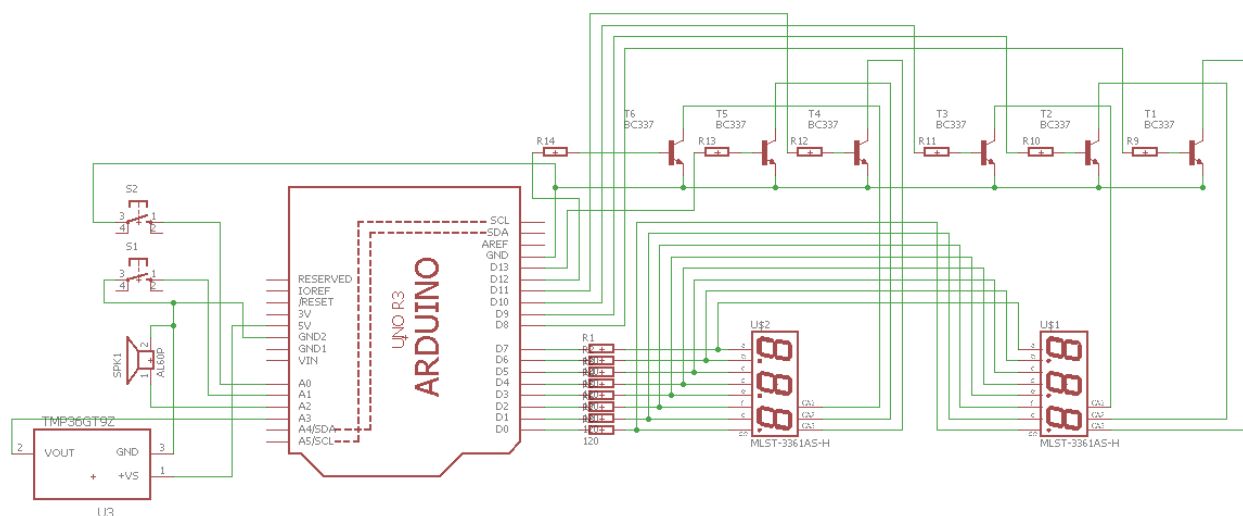


Obrázek 23 Shield2 - Deska plošného spoje, Osazený Shield

3.3 Shield 3

Mým posledním a zároveň na návrh nejtěžším shieldem, byl 3.shield, který měl obsahovat 6 sedmissegmentových displejů, 4 mikrospínače, teplotní čidlo a bzučák. Vedoucím maturitní práce mi byly poskytnuty netradiční 2 troj-sedmissegmentové displeje s označením 3361AS se společnou katodou pro které byl problém sehnat knihovny, nakonec je bylo nutné stáhnout z neoficiálních zdrojů a bylo nutné si ověřit, zda schéma jak velikostně tak pinově sedí. Pro displeje jsem použil zapojení s NPN tranzistory a odpory, tedy multiplexu. Dle testování svítivosti displejů při zobrazování hodnot na displeji, jsem zvolil odpor 200Ω o rozteči 7mm pro každou sedmissegmentovku. Smyslem multiplexového zapojení je následná funkčnost, která se projevuje při programování více displejů. Používá se v případě, že jsou v zapojení víc jak 2 sedmissegmentové displeje, kvůli nedostatku místa pro piny na Arduinu. Proto jsou jednotlivé segmenty propojeny a tranzistorem pomocí programu se přepíná mezi jednotlivými displeji a tím pádem svítí vždy pouze jeden. Ale přepínání probíhá tak rychle, že si toho lidské oko nevšimne a působí to tak, jakoby svítily všechny. Já osobně tam mám pro plynulé zobrazení nastavené 2 ms. Schéma desky shieldu, o rozměrech Arduina jsem použil z knihovny Adafruit. Celkem bylo tedy využito 14pinů, 6 na přepínání mezi displeji a 8 na ovládání jednotlivých segmentů. Vše jsem připojil na digitální výstupy. Dále jsem do schémata zapojil 4 mikrospínače, které byli opět přivedeny na analogový vstup s funkcí INPUT_PULLUP pro ušetření místa na desce. Mnou zvolené teplotní čidlo má 3 piny, proto jsem jeden zapojil na zem, druhý na 5 V a prostřední na vstupní analogový port A3, který dokáže promítat naměřené hodnoty například do sériového monitoru. Zapojení bzučáku nebylo složité, stačilo ho přivést na zem a napojit také na 5V zdroj. Jeho rozteč jsem zvolil 5mm, podle předem vybraného typu. Při objednávce bzučáku jsem dbal na to, aby byl s generátorem, kvůli usnadněnému programování.

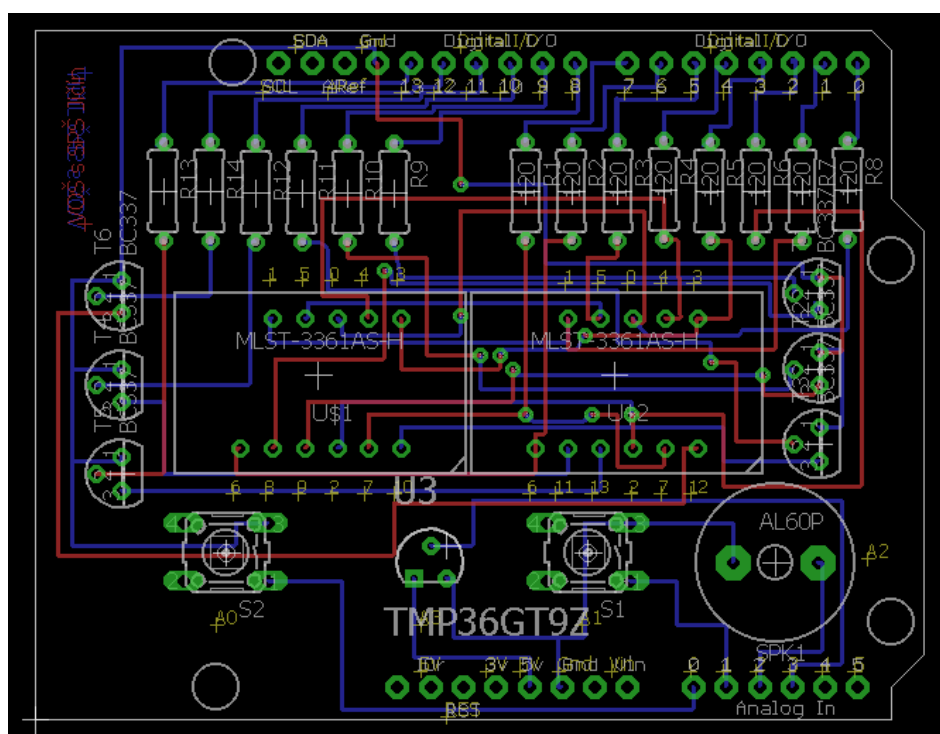
Výsledné schéma zapojení 3. shieldu v programu EAGLE.



Obrázek 24 Shield3 - Schéma

Z tohoto schémata jsem vygeneroval prostředí pro tvorbu desky plošného spoje. Zde jsem byl nucen kvůli husté síti cest použít dvouvrstvé řešení desky. Troj-segmentové displeje jsem dal na těsno k sobě, aby bylo možné zobrazovat plynule za sebou jakýkoliv text, či číselný údaj.

Po strategickém uspořádání součástek jsem se pustil do ručního routování, kde jsem se nevyhnul nutným propojům obou stran. Piny jak analogové tak digitální jsem pro přehlednost opět popsal.



Obrázek 25 Shield3 - Deska

Tabulka 4 Shield3 - Součástky

Druh	Typ	Počet kusů
Troj-sedmisegmentový displej	3361AS	2
Odpor 200 Ω	CF1/4w-200R	14
NPN Tranzistor	BC546B	6
Teplotní čidlo	TMP36GT9Z	1
Bzučák	AL60P	1
Mikrospínače	B3F-1000	2
Pin k Arduino	Po 8 kusech	4

Kvůli aktuálně chybějícím kusům na skladě obchodu, jsem byl nucen použít náhrady. Teplotní čidlo TMP36 bylo nahrazeno čidlem MCP9700. Bzučák AL60P byl nahrazen bzučákem BPT-14X. Oba dva kusy jsou shodné a v obvodu fungují. Programování 3.shieldu bylo díky multiplexu propojeným s teplotním čidlem a bzučákem jednoznačně nejobtížnější.

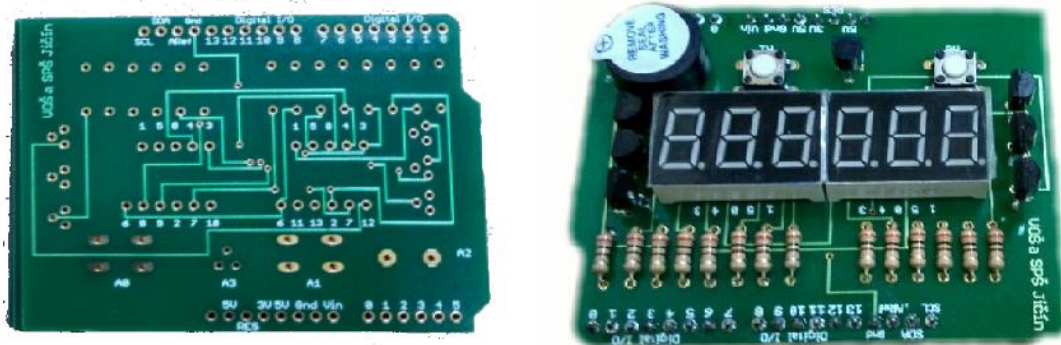
Sedmisegmentové displeje zobrazují teplotu s aktualizací každých 500ms. Pro přehledné zobrazení teploty je první a pátá sedmisegmentovka vypnutá. U první je proto, aby byla hodnota umístěna ve středu, pátá slouží jako oddělení hodnoty od jednotek, v tomto případě Celsia (C).



Obrázek 26 Shield3 - Zobrazení teploty

Teplota je zobrazena s přesností na jedno desetinné číslo. Při překročení teploty 25°C se spustí zvukový alarm v podobě přerušované signalizace o frekvenci 1000 Hz. Při stisku jednoho ze dvou mikrospínačů se spustí tón, při kombinování stisků obou je možné nasimulovat zvuk sirény.

[\[Příloha č.3 - Zdrojový kód\]](#)



Obrázek 27 Shield3 - Deska plošného spoje, Osazený Shield

4 Závěr

Během počáteční tvorby shieldů, jsem se potýkal s různými problémy, které plynuly z neznalosti programu EAGLE i prostředí Arduina. Přestože z toho maturuji, jsem se s těmito programy nikdy dříve nesetkal a tak jsem nemálo času investoval právě do studia o těchto dvou programech. Teprve když jsem se naučil základy, jsem se mohl pustit alespoň do základů s chybami, které spočívaly v různých často vizuálních nedostacích, funkčních zapojení či ve zbytečně komplikovaných řešeních. Po vyřešení těchto chyb, jsem dosáhl výsledku v podobě tří fungujících Arduino shieldů. Nyní se znalostmi, které mám a které jsem díky této práci získal, bych se určitě vyvaroval chybám, kterých jsem se při výrobě shieldů dopustil. A také bych zvolil větší rozměr plošného spoje pro snazší umístění součástek, vedení cest a následného letování.

Nyní už jen doufám, že budou shieldy využity k výuce a pomohou žákům pochopit a naučit se programovat Arduino.

Na závěr bych chtěl mnohokrát poděkovat panu učiteli Vladimíru Vikovi, který byl mým vedoucím maturitní práce a který mi po celou dobu dával užitečné rady, pohotově komunikoval a vždy mi se vším ochotně pomohl.

Obrázek 1 Ethermet Shield	5
Obrázek 2 WiFi Shield.....	5
Obrázek 3 GPS Shield	6
Obrázek 4 Vývojové prostředí Arduina	6
Obrázek 5 EAGLE - Nový projekt.....	9
Obrázek 6 EAGLE - Nové schéma	9
Obrázek 7 EAGLE - nastavení mřížek	11
Obrázek 8 EAGLE - tisk	12
Obrázek 9 EAGLE Knihovna - vyhledání.....	12
Obrázek 10 EAGLE knihovna - součástky.....	13
Obrázek 11 Knihovna - přidání.....	14
Obrázek 12 EAGLE - Tvorba schématu	15
Obrázek 13 EAGLE - Tvorba desky	15
Obrázek 14 EAGLE - Umístění součástek na desce.....	16
Obrázek 15 EAGLE - Vedení cest na desce	17
Obrázek 16 EAGLE - Nastavení Autorouteru	18
Obrázek 17 EAGLE - Vrstvy.....	19
Obrázek 18 Shield1 - schéma	20
Obrázek 19 Shield1 - Deska.....	21
Obrázek 20 Shield1 - Deska plošného spoje, Osazený Shield.....	22
Obrázek 21 Shield2 - Schéma.....	23
Obrázek 22 Shield2 - Deska.....	24
Obrázek 23 Shield2 - Deska plošného spoje,Osazený Shield.....	25
Obrázek 24 Shield3 - Schéma.....	27
Obrázek 25 Shield3 - Deska.....	27
Obrázek 26 Shield3 - Zobrazení teploty	29
Obrázek 27 Shield3 - Deska plošného spoje,Osazený Shield.....	29
Tabulka 1 Seznam užitečných knihoven	13
Tabulka 2 Shield1 - součástky.....	22
Tabulka 3 - Shield2 - Součástky.....	24
Tabulka 4 Shield3 - Součástky	28

4.1 Zdroje

Citace:

Arduino: popis. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-02-28]. Dostupné z: <https://cs.wikipedia.org/wiki/Arduino>

EAGLE: informace. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-02-28]. Dostupné z: [https://en.wikipedia.org/wiki/EAGLE_\(program\)](https://en.wikipedia.org/wiki/EAGLE_(program))

Hlavní zdroje pro výpomoc při tvorbě zdrojových kódů:

Arduino - StateChangeDetection . Arduino - Home [online]. Dostupné z: <https://www.arduino.cc/en/Tutorial/StateChangeDetection>

Use tone() with Arduino for an Easy Way to Make Noise - Programming Electronics Academy. 301 Moved Permanently [online]. Dostupné z: <https://programmingelectronics.com/an-easy-way-to-make-noise-with-arduino-using-tone/>

How to measure temperature with Arduino and MCP9700. Starter Kit - Learning Arduino [online]. Dostupné z: <http://starter-kit.nettigo.eu/2010/how-to-measure-temperature-with-arduino-and-mcp9700/>

itnetwork.cz - Ajtácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další. [online]. Dostupné z: <https://www.itnetwork.cz/hardware-pc/arduino/arduino-7-segmentovy->

Seznam příloh

Příloha č.1 - zdrojový kód k shieldu 1

Příloha č.2 - zdrojový kód k shieldu 2

Příloha č.3 - zdrojový kód k shieldu 3