

# **Návrh a konstrukce počítačem spínané zásuvky**

**Závěrečná práce**

Vedoucí práce:  
Mgr. Marek Blaha

Marek Nespěchal

Pozořice 2019

Rád bych poděkoval zejména panu profesorovi Mgr. Marku Blahovi za odbornou pomoc při vypracovávání mé závěrečné práce, kterou jsem díky jeho korekcím dovedl do zdárného konce. Dále bych chtěl poděkovat svému otci za pomoc a rady při samotné realizaci tohoto zařízení.

Prohlašuji, že jsem tuto závěrečnou práci vyřešil samostatně s použitím literatury, kterou uvádím v seznamu.

V Pozořicích dne 15. března 2019

.....

## **Abstract**

Nespěchal, M. Designing and constructing computer controlled outlet. Theses. Pozořice, 2019.

This text is about designing and constructing an outlet which is turned on and off together with computer. The text describes electronic circuit based on microcontroller and its programming. The text also covers process of designing and making own printed circuit board (PCB).

## **Abstrakt**

Nespěchal, M. Návrh a konstrukce počítačem spínané zásuvky. Závěrečná práce. Pozořice, 2019.

Práce se zabývá návrhem a konstrukcí zásuvky, která se společně s počítačem zapíná a vypíná. Práce popisuje obvod postavený na mikrokontroléru a také jeho programování. V práci je také zahrnut návrh a výroba vlastní desky plošného spoje (DPS).

## Obsah

1	Úvod a cíl práce.....	8
1.1	Úvod do problematiky .....	8
1.2	Cíl práce .....	8
2	Popis principu fungování .....	9
3	Návrh obvodu.....	10
3.1	Mikroprocesor.....	10
3.2	Relé .....	10
3.3	Indikační LED.....	10
3.4	Přepínač a vstup signálu .....	11
3.5	Schéma.....	11
4	Program mikroprocesoru.....	12
4.1	Funkce prepínac .....	12
4.2	Funkce blikání.....	13
4.3	Funkce setup .....	13
4.4	Funkce loop.....	14
5	Nahrávání programu.....	15
5.1	Knihovna pro Attiny85 .....	15
5.2	ArduinoISP .....	15
5.3	Propojení.....	15
5.4	Bootloader.....	15
5.5	Program.....	16
6	Návrh DPS.....	17
7	Výroba DPS.....	18
7.1	Tisk masky .....	18
7.2	Příprava desky.....	18
7.3	Přenos.....	18
7.4	Leptání .....	18
7.5	Dokončení.....	19
8	Osazení DPS.....	20
9	Konstrukce zařízení.....	22
9.1	Otvory .....	22
9.2	Propojení.....	22

9.3	Relé a zásuvka .....	22
9.4	Koncovky kabelů .....	23
9.5	Testování.....	23
10	Závěr.....	25
11	Literatura .....	26

Obrázek 1: Schéma obvodu.....	11
Obrázek 2: Funkce prepínac() .....	12
Obrázek 3: Funkce blikani() .....	13
Obrázek 4: Funkce setup().....	13
Obrázek 5: Funkce loop() .....	14
Obrázek 6: Plošný spoj .....	17
Obrázek 7: Hotová DPS .....	21
Obrázek 8: Zařízení bez kabelů .....	24

# **1 Úvod a cíl práce**

## **1.1 Úvod do problematiky**

Závěrečnou práci na toto téma jsem si zvolil proto, že se již od malička zajímám o počítače a elektroniku. Nápad na tento projekt – spínanou zásuvku, jsem dostal už před dlouhou dobou. Začal jsem pracovat na prototypu, který bez problému fungoval, ale neměl jsem odvalu se pouštět ke spínání 230 V. Toto nedokončené zařízení jsem tedy odložil a až do nedávna na něj zapomněl. Nyní jsem se k němu vrátil a pustil se do výroby finální verze.

Na svém počítačovém stole mám prodlužovací kabel s vypínačem, do kterého je zapojený monitor a reproduktory. Po spuštění počítače je potřeba zapnout právě i onu prodlužku. Jsem člověk líný, a tak mě každodenní mačkání vypínače omrzelo. Přišlo mi jako elegantní řešení si vyrobit zásuvku, která by se sama zapínala a vypínala podle toho, jestli je počítač v provozu, nebo ne. Podobnou funkci si pamatuji u počítače z 90. let, který měl ve svém zdroji i spínaný výstup, do kterého bylo vhodné zapojit monitor, jenž se tímto zapínal společně s počítačem. U současných zdrojů jsem tuto funkci již neviděl, což mi přijde jako velká škoda.

## **1.2 Cíl práce**

Ve své práci popisuji jednotlivé kroky výroby elektronického zařízení řízeného mikroprocesorem. Na internetu je spousta návodů na projekty využívající vývojovou desku Arduino, ale pokud chceme permanentní řešení, je lepší a ekonomičtější nahradit celou desku pouze mikroprocesorem.

Cílem mé práce je tedy navržení obvodu pro spínanou zásuvku, tvorba programu a samotné programování mikroprocesoru, návrh tištěného spoje, jeho výroba, osazení a uzavření do krabice, což vyústí v bezpečné a spolehlivé zařízení.



## 2 Popis principu fungování

Snažíme se navrhnout zařízení, které bude spínat síťové napětí. Jako signální vstup využijeme jeden z molex konektorů počítačového zdroje. Tento konektor má 4 piny: +5 V, +12 V a každý má svou zem. Po zapnutí počítače jsou na těchto pinech tato napětí. My využijeme větve s 5 V, protože 12 V nemůžeme připojit přímo k mikroprocesoru, neboť jeho I/O piny jsou dimenzovány pouze na 5 V.

Samotný mikroprocesor budeme napájet také 5 V, ale ty budou pocházet z obyčejné nabíječky k telefonu, která bude k zařízení připojena přes micro USB kabel. Zároveň do zařízení musí být přivedeno síťové napětí, které se snažíme držet co nejdále od nízkonapěťového obvodu z důvodu bezpečnosti. Toto napětí je přivedeno na relé, které je spínáno mikroprocesorem.

Pro indikaci sepnutí zásuvky použijeme červenou LED, která je umístěna na krabici. Zařízení má tři režimy fungování, které lze vybírat 3–polohovým přepínačem – stále vypnuto, automatický režim a stále zapnuto. Ve vypnutém režimu je přepínač v dolní poloze, indikační LED nesvítí a zásuvka je vypnutá. Pokud je páčka přepínače v horní poloze, indikační LED bliká a zásuvka je zapnutá.

Pro automatický režim musí být páčka přepínače v prostřední poloze a stav zařízení se vyhodnocuje podle toho, jestli je na signálním pinu 5 V, nebo ne. Pokud ano, indikační LED svítí a zásuvka je zapnutá, pokud ne, indikační LED nesvítí a zásuvka je vypnutá.

### 3 Návrh obvodu

Zásuvka je řízena mikroprocesorem Atmel Attiny85. Jedná se o osmipinový čip s šesti vstupními/výstupními porty. Každý z těchto pinů zvládne maximální proud 40 mA. Tři z nich využijeme jako vstupní – pro indikaci stavu počítače, a pro dvě polohy přepínače. Další dva budou výstupní – jeden pro indikační LED a druhý pro řízení relé.

#### 3.1 Mikroprocesor

Kromě 6 I/O pinů má náš čip další 3 vývody VCC, GND a RESET, který zároveň slouží jako I/O pin, pokud by nám předchozích pět nestačilo.

Zapojení VCC a GND je velmi intuitivní, stačí je opravdu připojit na 5 V a zem. Pokud bychom měli nestabilní napájecí napětí, bylo by vhodné zapojit mezi tyto dva piny kondenzátor, který by se staral o stabilizaci napětí. V našem případě to není nutné, protože nabíječka k telefonu má napětí dostatečně regulované.

Jako poslední nám zbývá zapojit pin RESET, který by měl být připojený přes rezistor na 5 V. Na hodnotě rezistoru v podstatě nezáleží, většinou se používá 10 k $\Omega$ . Tímto zajistíme, že se nám mikroprocesor neresetuje sám od sebe během provozu, což by mohlo nastat, pokud bychom nechali pin nezapojený.

#### 3.2 Relé

Narazíme na problém právě s pinem pro řízení relé, protože mnou zvolené na 5 V má odpor 68  $\Omega$ . Po dosazení hodnot do Ohmova zákona  $I = U / R$  získáme spínací proud cca 73 mA. Tento problém lze vyřešit použitím tranzistoru.

Bázi tranzistoru připojíme přes 1 k $\Omega$  rezistor k výstupnímu pinu mikroprocesoru. Tímto omezíme tekoucí proud na 5 mA, který už pin zvládne. Emitor připojíme na zem, jeden pin relé na 5 V a druhý na kolektor. Tranzistor, který mám po ruce, má minimální proudové zesílení 30, takže proud 5 mA přes bázi by vyvolal 150 mA přes kolektor. Takový proud ale přes kolektor nemůže fyzicky téct z důvodu odporu relé, a tak se tranzistor dostane do stavu saturace, kdy funguje jako spínač – přesně tak, jak potřebujeme.

#### 3.3 Indikační LED

Jako indikaci jsem použil standardní 5mm červenou diodu, která je připojena přes 330  $\Omega$  rezistor z důvodu omezení proudu. Tato klasická LED má maximální proud 20 mA, ale intenzita svícení se při proudu blížícího se maximu moc nemění, proto je cca 15 mA s naším rezistorem naprosto dostatečné.

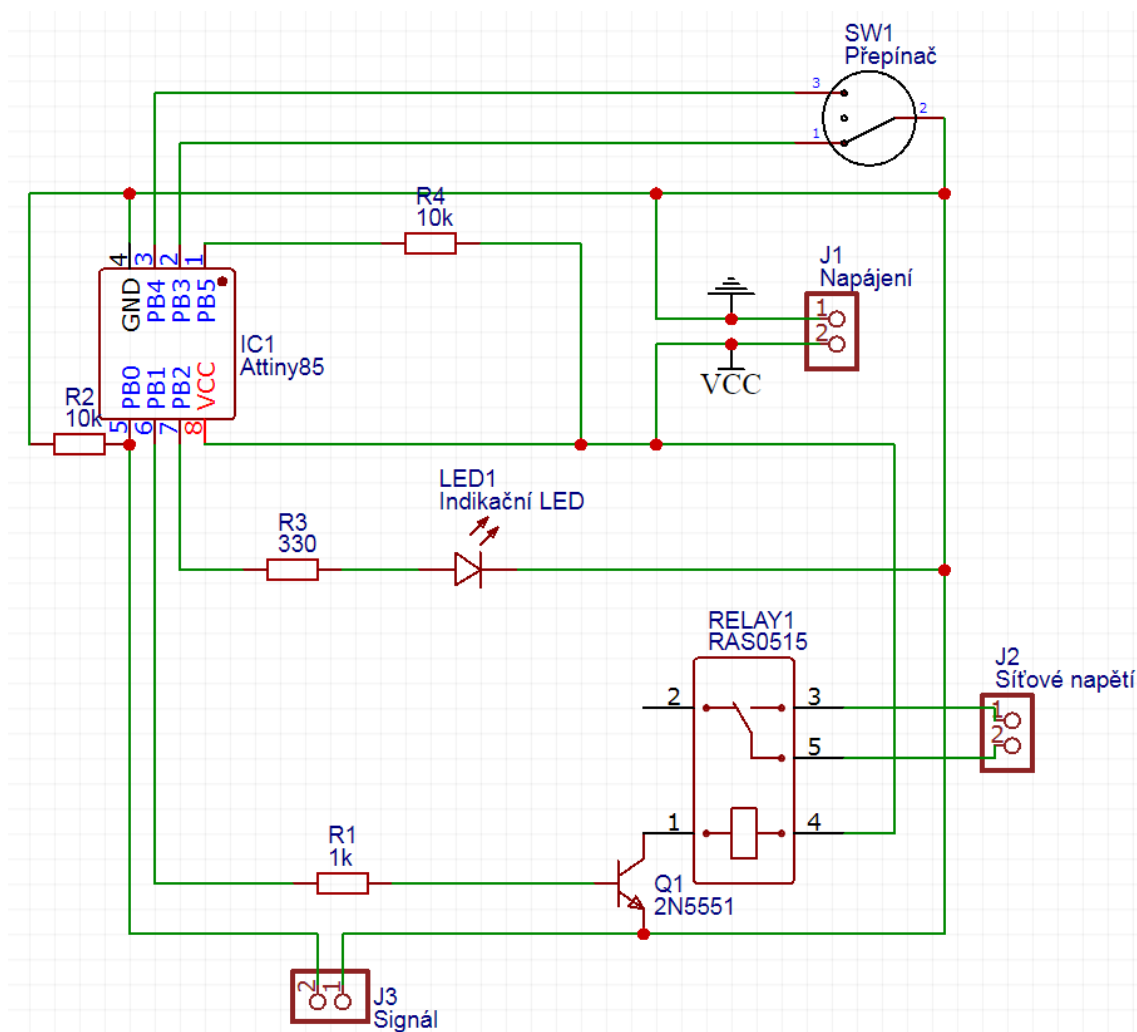
### 3.4 Přepínač a vstup signálu

Spínacím prvkem je obyčejný 1pólový, 3polohový přepínač (ON-OFF-ON). Jeho společný vývod připojíme na zem a krajní vývody ke dvěma vstupním pinům mikroprocesoru. Zde nemusíme použít žádný odpor limitující proud, protože náš čip má uvedený vstupní odpor I/O pinů 20–50 kΩ dle technické dokumentace. Poslední vstupní pin nám vyhodnocuje stav počítače, připojíme ho tedy na 5V větev molex konektoru. Abychom měli jistotu, že nám nebude pin vyhodnocovat stav při vypnutí počítače chybně, připojíme ho ještě přes 10 kΩ rezistor na zem.

### 3.5 Schéma

Pro tvorbu schématu jsou použity program EasyEDA, který je zadarmo a lze ho používat i v internetovém prohlížeči. Snadno se používá, má velkou knihovnu součástek a umožňuje konverzi schématu na desku plošného spoje.

Hotové schéma vypadá takto:



Obrázek 1: Schéma obvodu

## 4 Program mikroprocesoru

Nyní potřebujeme vymyslet kód, kterým se bude mikroprocesor řídit. Jako vývojové prostředí využijeme ARDUINO IDE, které je sice určeno k programování vývojových desek, ale při použití správné knihovny lze programovat i mikroprocesor.

### 4.1 Funkce prepínac

Naprogramujeme si funkci, která bude vracet celé číslo – 0, 1, 2, podle toho, v jaké poloze je páčka přepínače. K tomuto využijeme funkci *digitalRead(pin)*, kde parametr pin je číslo pinu, u kterého zjišťujeme stav. Funkce vrací LOW, nebo HIGH.

Prvně se zeptáme, jestli je pin 3 LOW – to znamená, že je přepínač v jedné krajní poloze. Pokud tomu tak je, funkce vrátí 1 a ukončí se. Pokud ne, to znamená pin 3 je HIGH, musíme se zeptat další podmínkou.

Nyní se ptáme, jestli je pin 4 LOW, pokud ano, přepínač je ve druhé krajní poloze, funkce vrátí 0 a opět se ukončí. Pokud ani tato podmínka není splněna, nemusíme se již na nic dalšího ptát a víme, že přepínač je v prostřední pozici, funkce vrátí 2 a ukončí se.

Naše funkce *prepínac()* tedy vypadá takto:

```
int prepínac() {  
    if(digitalRead(3) == LOW) {  
        return 1;  
    }  
    else{  
        if(digitalRead(4) == LOW) {  
            return 0;  
        }  
        else {  
            return 2;  
        }  
    }  
}
```

Obrázek 2: Funkce prepínac()

## 4.2 Funkce blikani

Tato funkce je velmi prostá a dělá pouze to, že zapne a vypne pin, na kterém je připojena indikační LED.

Využijeme dvě funkce:

- *digitalWrite(pin, stav)* – funkce, která mění u konkrétního pinu stav, parametr *pin* je číslo pinu, u kterého chceme měnit stav, parametr *stav* může být LOW nebo HIGH
- *delay(cas)* – funkce, která čeká konkrétní dobu, než je umožněno v programu pokračovat dál, parametr *cas* určuje dobu v ms, po jakou bude program čekat

Naše funkce *blikani()* tedy vypadá takto:

```
void blikani() {  
    digitalWrite(2, HIGH);  
    delay(500);  
    digitalWrite(2, LOW);  
    delay(500);  
}
```

Obrázek 3: Funkce blikani()

## 4.3 Funkce setup

V setup funkci se nachází příkazy, které se provedou pouze jednou při spuštění zařízení. V našem případě se v ní nachází pouze příkazy pro nastavení I/O pinů jako vstup, nebo výstup.

K tomuto slouží funkce *pinMode(pin, režim)*. Prvním parametrem je číslo pinu, který chceme nastavit. Druhý je režim fungování, který má tři možné hodnoty:

- OUTPUT – pin je nastavený jako výstupní
- INPUT – pin je nastavený jako vstupní
- INPUT\_PULLUP – pin je nastavený jako vstupní a pokud na něm není žádné napětí, pin je HIGH (odtud pullup – přitáhnout nahoru)

Naše funkce *setup()* tedy vypadá takto:

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(0, INPUT);  
    pinMode(3, INPUT_PULLUP);  
    pinMode(4, INPUT_PULLUP);  
    pinMode(1, OUTPUT);  
    pinMode(2, OUTPUT);  
}
```

Obrázek 4: Funkce setup()

## 4.4 Funkce loop

Tato funkce se provádí stále dokola, odtud název loop (smyčka). Zde se nachází *switch*, který řídí naše funkce *prepinac()*, která vrací hodnoty 0, 1, nebo 2.

V případě, že nám funkce *prepinac()* vrátí 0, řekneme, že je přepínač v dolní poloze, a tak chceme, aby bylo zařízení stále vypnuté. Vypneme indikační LED a také pin, který řídí relé.

V případě, že nám funkce vrátí 1, přepínač je v opačné poloze a chceme, aby bylo zařízení stále zapnuté. Abychom nezapomněli, že tomu tak je, chceme, aby nás na tuto skutečnost upozornila indikační LED blikáním. Proto jsme si připravili naši funkci *blikani()*, kterou zavoláme. Dále samozřejmě chceme zapnout pin, který řídí relé.

V případě, že nám funkce vrátí 2, víme, že je přepínač v prostřední poloze a musíme se dále ptát, jestli je počítač zapnutý, nebo ne. Zeptáme se, jestli je náš signální pin HIGH. Pokud ano, zapneme indikační LED a také pin, který řídí relé. Pokud tomu tak není, chceme relé vypnout stejně jako indikační LED.

Naše funkce *loop()* tedy vypadá takto:

```
void loop() {  
    // put your main code here, to run repeatedly:  
    switch(prepinac()) {  
        case 0:  
            digitalWrite(1, LOW);  
            digitalWrite(2, LOW);  
            break;  
        case 1:  
            blikani();  
            digitalWrite(1, HIGH);  
            break;  
        case 2:  
            if(digitalRead(0) == HIGH) {  
                digitalWrite(2, HIGH);  
                digitalWrite(1, HIGH);  
            }  
            else {  
                digitalWrite(1, LOW);  
                digitalWrite(2, LOW);  
            }  
            break;  
    }  
}
```

Obrázek 5: Funkce loop()

## 5 Nahrávání programu

### 5.1 Knihovna pro Attiny85

Abychom mohli pracovat s naším mikroprocesorem, je potřeba ho přidat jako vývojovou desku do programu Arduina. K tomuto slouží knihovny, které jsou dostupné zdarma na internetu. Já jsem zvolil tuto:

<https://github.com/SpenceKonde/ATTinyCore>

Tento balíček nám přidá podporu nepřeberného množství mikrokontrolérů, mezi kterými se nachází i naše Attiny. Postup při instalaci je srozumitelně popsán na hlavních stránkách knihovny (odkaz nahoře).

### 5.2 ArduinoISP

Pro programování dalších vývojových desek Arduinem na něj musíme nahrát program, který se nachází v příkladových projektech. Vybereme tedy z nabídky projekt ArduinoISP a nahrajeme ho na naši vývojovou desku. Slouží jako zprostředkovatel komunikace mezi Arduinem a mikrokontrolérem – díky němu jsme schopni s mikroprocesorem pracovat bez převodníku sériové komunikace (jako převodník funguje právě naše Arduino s nahraným projektem).

### 5.3 Propojení

Našemu mikrokontroléru musíme poskytnout napájení – to znamená 5 V a GND, které získáme z Arduina. Dále je potřeba propojit piny MISO, MOSI a SCK obou zařízení. Jako poslední propojíme resetovací pin Attiny s pinem SS Arduina. Kde se jaký pin nachází, vyčteme z technických dokumentací. Doporučuje se ještě přes 10 $\mu$ F kondenzátor připojit resetovací pin Arduina na GND, aby nemohlo dojít k samovolnému resetování. S tím jsem nikdy problém neměl, a tak to nedělám.

Schéma pro propojení Arduina UNA a Attiny85 lze nalézt na této adrese:

[https://hackster.imgix.net/uploads/image/file/50921/Programming%20ATTiny85%20with%20Arduino%20Uno\\_bb.png?auto=compress%2Cformat&w=1280&h=960&fit=max](https://hackster.imgix.net/uploads/image/file/50921/Programming%20ATTiny85%20with%20Arduino%20Uno_bb.png?auto=compress%2Cformat&w=1280&h=960&fit=max)

### 5.4 Bootloader

Před nahráním prvního programu mikroprocesoru na něj musíme vypálit takzvaný bootloader. Musíme si vybrat, jestli chceme použít interní, nebo externí oscilátor a jeho frekvenci; a napětí, při kterém se mikrokontrolér dostane do stavu resetu, pokud ho nebude dosaženo (B.O.D.). Naše Attiny může pracovat na vyšší frekvenci, než je frekvence oscilátoru – to má využití při pracování s PWM o vysokém kmitočtu (Timer 1 Clock). Navíc se musíme rozhodnout, zda chceme využívat funkce pro zmenšení velikosti našeho projektu (LTO).

V našem případě používáme interní oscilátor s frekvencí 8 MHz, kterou už dále nezvyšujeme, funkci B. O. D. ani LTO nevyužíváme. V programu Arduina tedy zvolíme pod položkou nástroje tyto parametry:

- Vývojová deska: ATtiny 25/45/85
- Timer 1 Clock: CPU
- Chip: ATtiny85
- Clock: 8 MHz (internal)
- LTO: Disabled
- B.O.D. Level: Disabled

Vybereme sériový port, na kterém se naše Arduino nachází a jako programátor použijeme Arduino as ISP (ATtinyCore). Jakmile naše nastavení zkontrolujeme, podíváme se, jestli máme správně propojené Attiny s Arduinem a můžeme vypálit zavaděč. Pokud jsme vše udělali správně, indikační LED Arduina by se nám měly rozblikat a po nějakém čase by nám měl program zahlásit úspěch.

## 5.5 Program

Jestliže jsme úspěšně vypálili bootloader, již nám nic nebrání nahrání programu na mikroprocesor. Propojení zůstává stejné jako při zavaděči. Pro ověření funkce je lepší zkusit prvně nahrát nějaký jednoduchý, například ukázkový program Blink, ve kterém pouze změníme výstupní pin, na který později připojíme přes rezistor limitující proud LED. Očíslování pinů pro Arduino opět nalezneme na hlavních stránkách knihovny. Pokud se nám po testovacím zapojení LED rozbliká, udělali jsme vše správně a po opětovném připojení Arduina můžeme stejným způsobem nahrát konečný program zařízení.



## 6 Návrh DPS

Pro tvorbu DPS (deska plošných spojů) jsem použil stejný program jako na tvorbu schématu – EasyEDA. Program umožňuje konvertovat právě ono schéma na DPS.

Součástky se nám přenesou mimo desku a je potřeba je umístit na nějaké logické místo. Vždy se snažíme dávat součástky, které jsou navzájem propojeny, co nejbližše k sobě, aby nebyly propojovací cesty zbytečně dlouhé, a zároveň šetříme místem na desce.

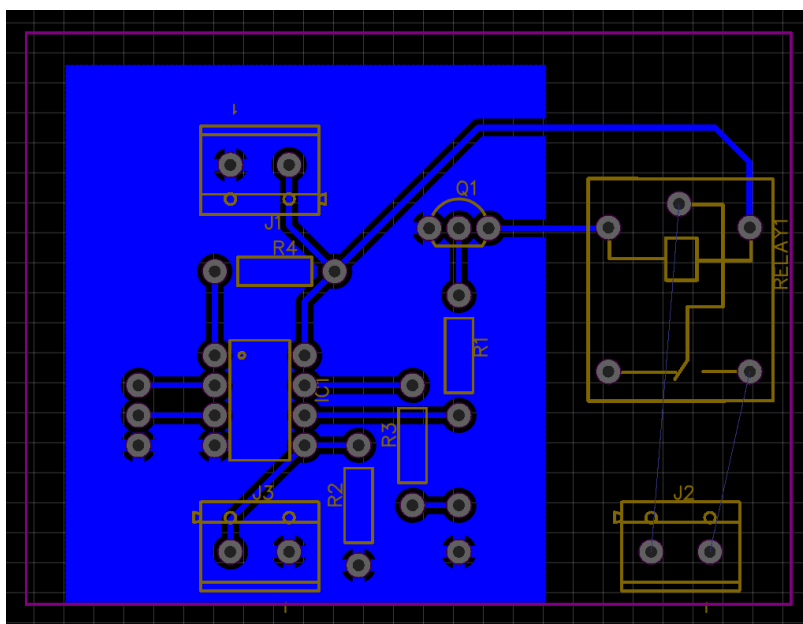
Součástky jsou propojeny čarami, které nám slouží k tomu, abychom věděli, jaké cesty je potřeba vytvořit. Tloušťku cest jsem zvolil 0,5 mm, průměr děr 1 mm a průměr plošek k pájení 2 mm.

Je důležité oddělit od sebe nízkonapětovou část (mikrokontrolér, přepínač, LED, ...) a část se síťovým napětím. Bezpečná vzdálenost je pro 230 V 2 mm, ale větší mezera rozhodně není na škodu. Samotné propojení relé a svorek bude provedeno měděným drátem s izolací, takže není potřeba řešit tloušťku cest na desce z důvodu větších proudů (relé je konstruované až na 10 A).

Při výrobě desky je možné vyleptat vše kromě propojujících cest, ale efektivnější řešení je vyleptávat prostor ohraničující jednotlivé cesty. Zbývá měděná plocha je GND. Z důvodu bezpečnosti ale pod relé a celou oblastí se síťovým napětím měděnou plochu nechceme, takže necháme vysokonapětovou část vyleptat celou.

Program nám umožňuje vyexportovat si navrženou desku do souboru PDF v reálné velikosti, takže si ji můžeme vytisknout a později přenést na DPS.

Hotový návrh desky vypadá takto:



Obrázek 6: Plošný spoj

## 7 Výroba DPS

### 7.1 Tisk masky

Vyexportovaný PDF soubor si vytiskneme pomocí laserové tiskárny na lesklý papír. Ideální je papír fotografický, ale lze použít i stránku z nějakého časopisu nebo letáku. Obraz není potřeba zrcadlově otáčet, protože se jedná o spodní stranu desky a při přenosu z papíru na podklad se otočí ještě jednou, takže se otočení vyruší. V nastavení tiskárny zvolíme co nejvyšší kvalitu tisku (DPI), aby se nám správně vytiskly i nejmenší detaily. Vytisknutou masku si vystříhneme nebo ideálně vyřízneme podle pravítka, abychom měli rovné hrany.

### 7.2 Příprava desky

Vystřižený papír si změříme a podle rozměrů uřízneme pilkou na železo cuprexitovou desku. Opět se snažíme dělat co nejrovnější řez. Měděný povrch na vyříznuté destičce je potřeba zbavit zoxidované povrchové vrstvy. To lze provést například drsnou stranou kuchyňské houbičky. Doporučuje se tento krok dělat namokro. Dále je potřeba zbavit desku mastnoty a jiných nečistot, aby se maska správně přenesla. Očištění provedeme technickým lihem. Nyní už při manipulaci s destičkou nesmíme sahat na měděnou vrstvu.

### 7.3 Přenos

Samotný přenos lze uskutečnit dvěma metodami – teplem, nebo chemickou cestou. První možnost spočívá v zahřátí papíru s tonerem, který je položený potištěnou stranou na měděnou vrstvu, například žehličkou. Úskalí této metody je nalezení správné teploty, protože při příliš nízké se toner nepřenese a při vysoké se nám začne papír pálit.

Dle mého názoru je lepším způsobem přenosu použití roztoku acetonu a technického lihu v poměru 3:8. Tento roztok lze nahradit i obyčejným odlakovačem na nehty. Na měděnou stranu desky přiložíme potištěnou stranu papíru a aplikujeme roztok. Prstem jezdíme po celé ploše a snažíme se vyvíjet tlak. Je důležité nevynechat žádné místo, protože se takováto část přenesla jen částečně. Pokud nám papír vyschl a my jsme nechali nějaké místo netknuté, lze papír polít znovu a nedodělky opravit.

Pokud jsme zpracovali celou plochu, vezmeme destičku a položíme ji do nádoby s vlažnou vodou. Papír se nesnažíme sundat, zhruba po 15 minutách ve vodě se odloupne sám nebo pouze s malou naší pomocí.

### 7.4 Leptání

Desku již bez papíru vezmeme, osušíme a zkontrolujeme, zdali se přenesl veškerý toner, případné nedostatky můžeme opravit například lihovou fixou zatřením defektních míst. Pro leptání se používá 40% roztok chloridu železitého ( $\text{FeCl}_3$ ). Na leptání se nesmí používat žádné kovové nástroje, takže je vhodné použít plastovou leptací misku, která by neměla být příliš velká, ale naše deska se do ní musí vejít. Měli bychom používat ochranné pomůcky, jako jsou rukavice a brýle. Podrobné bezpečnostní pokyny jsou napsány na nádobě s chloridem.

Do misky nalijeme roztok tak, aby byla deska ponořená. Do samotného leptání nemusíme zasahovat, ale pokud chceme proces urychlit, lze s deskou v lázni pohybovat a otáčet nekovovým nástrojem, například špejlí. Dobu leptání nelze přesně určit, takže je potřeba stav desky kontrolovat. Jakmile bude všechna nezakrytá měď vyleptána, desku z lázně vyjmeme a důkladně opláchneme vodou.

Roztok lze používat vícekrát, ale intenzita leptání se s každým dalším použitím snižuje. Obsah misky uschováme do uzavíratelné nádoby, kterou patřičně označíme. Roztok i s nádobou lze odevzdat ve sběrném místě nebezpečných odpadů.

## **7.5 Dokončení**

Vyleptanou desku musíme ještě zbavit krycího toneru na měděných cestách. To lze provést mechanicky, například drátěnkou; nebo technickým lihem. Desku opět opláchneme a osušíme. Nakonec zkontrolujeme kontinuitu všech cest pomocí multimetru. Pokud by byla nějaká cesta přerušena, je potřeba najít defektní místo a přemostit ho připájením kousku drátu, který bude spojovat nepropojené strany.

Pro vrtání desky je vhodné použít pákový stojan na vrtačku. Lze také vrtat frézku, která je často používaná například modeláři. V našem případě, kdy potřebujeme vyvrtat pouze pár děr, si vystačíme s ručním vrtáním. Já používám modelářský nožík, ze kterého jsem vyndal čepel a místo ní uchytil vrták.

Vrtáky do DPS jsou velmi tenké a tudíž křehké, takže je třeba dbát zvýšené opatrnosti, abychom se v případě zlomení vrtáku nezranili. Vrtáky jsou z různých materiálů, od kterých se odvíjí cena, křehkost a doba, za kterou se ztupí. S ohledem na to, že lze vrták snadno zlomit, je lepší si nakoupit zásobu levných vrtáků, které se sice rychle opotřebovávají, ale není problém je po 20 vyvrtaných dírách měnit.

## 8 Osazení DPS

Před započítím osazení je vhodné si připravit, proměřit a roztřídit všechny součástky, které budeme umisťovat. Osazovat se začínají malé a nakonec největší, protože větší součástka by mohla překážet při umisťování malé.

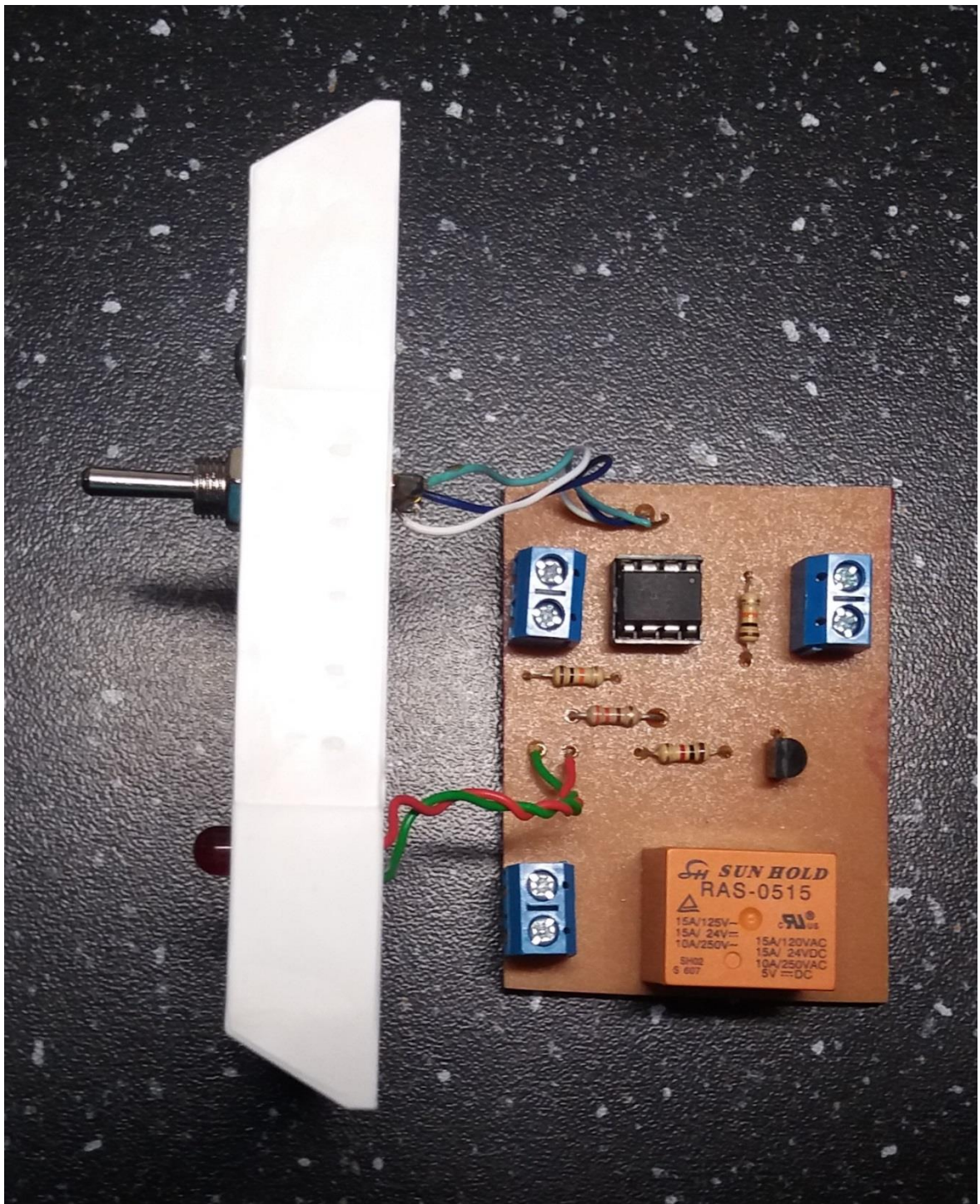
Někdo do desky umisťuje všechny součástky najednou, ohne jejich nožičky, aby z desky nevypadly, vše připájí a nakonec zkrátí přebytečně dlouhé nožičky. Takto umístěné součástky se ale velmi těžko v případě potřeby výměny demontují, protože zahnutá nožička v desce drží i po odstranění pájky, takže je potřeba při demontáži používat hrubou sílu.

Je lepší nožičky neohýbat, sice mají součástky tendenci z desky vypadnout, ale pokud pájíme menší počet součástek najednou, není to problém. Při umisťování součástky s velkým počtem kontaktů, např. patice pro náš mikroprocesor, se připájí dvě nejvzdálenější nožičky, aby součástka v desce držela, a nakonec se už bez problému pájí nožičky ostatní.

Jako poslední připájíme izolované měděné dráty, které propojují relé se svorkami pro kabel. Je potřeba myslet na velikost proudu, který tímto spojením poteče. Přes relé při 230 V může proudit až 10 A, takže tomu musíme přizpůsobit tloušťku drátu. V našem případě použijeme měděný drát o průměru aspoň 1,5 mm s izolací.

Po dokončení pájení vizuálně zkontrolujeme, zda jsou všechny kontakty propojeny tam, kde mají. Pokud narazíme na místo, které vypadá, že je propojeno jinak než má, je potřeba proměřit okolní spojení a případnou chybu opravit. Toto je velmi důležité, protože po zapojení desky, která by byla omylem při pájení propojena jinde než má, hrozí zničení součástek, kterých se to týká, nebo dokonce celé desky.

Pokud jsme si jistí, že je vše jak má být, zbývá desku zapojit a otestovat funkce na nečisto. Jestliže vše funguje, můžeme desku s klidným svědomím zavřít do finální krabice.



Obrázek 7: Hotová DPS

## 9 Konstrukce zařízení

Základem zařízení je dvojité elektroinstalační krabice na lištu. V horní části je obyčejné víčko, do kterého je umístěn přepínač a indikační dioda. Ve spodní je mělká dvojbáseň. Plošný spoj je v krabici upevněn jedním šroubem, který je od obvodu oddělen izolační průchodkou. Aby se deska nemohla pohnout, je zapuštěna do drážek v krabici.

### 9.1 Otvory

Do krabice je nutné vyvrtat několik otvorů – v nízkonapěťové části pro napájení a vstup signálu; v blízkosti zásuvky je potřeba připravit otvor pro kabel síťového napětí. Pro přepínač a indikační LED je potřeba vyvrtat díry do víčka.

Vrtání do plastu lze provést vrtáky na železo. Je potřeba si dobře změřit průměry objektů, které budeme chtít otvory protáhnout. Příliš velká díra již nelze opravit, je tedy vhodné začít s malými vrtáky a postupně otvory zvětšovat. Užitím tohoto způsobu se nemusíme bát vyvrtání příliš velké díry, pokud budeme po každém zvětšení otvoru zkoušet umisťovat naše součástky. Také si tímto nepoškodíme plastovou krabici, která je vcelku měkká.

### 9.2 Propojení

Po umístění přepínače a LED do víčka je potřeba přiletovat krátké dráty propojující součástky s tištěným spojem. Je vhodné nedělat propojení příliš dlouhé, aby se nám zbytečně dlouhé dráty v krabici nepletly. Po nalezení vhodné délky drátů přiletujeme jejich volné konce na správná místa plošného spoje a zajistíme ho šroubem.

Do svorek na DPS zasuneme a zašroubujeme patřičné kabely nízkého napětí. Je potřeba si dávat pozor, abychom měli odizolovanou dostatečně dlouhou část drátu, protože nesmíme dotahovat šroubek svorky proti izolaci kabelu. Takto upevněný drát totiž nemusí ve svorce dostatečně držet a také nemusí tvořit správný kontakt se svorkou.

Snad ještě důležitější je zkontrolovat, zdali jsme připojili správné kabely do správných svorek. Pokud bychom později připojili síťové napětí na svorku pro 5V napájení, v nejlepším případě si pouze zničíme celou naši práci, v tom horším může dojít k úrazu elektrickým proudem!

### 9.3 Relé a zásuvka

Jako poslední propojíme zásuvku s relé. Vodič fáze připojíme do zdířky svorky na DPS, která propojuje normálně otevřený kontakt relé (v našem případě pravá zdířka). Do spínané zdířky připojíme kabel, který má druhý konec zašroubovaný v levé zdířce zásuvky. Neutrální vodič se zapojí do zdířky pravé na zásuvce. Nesmíme také zapomenout na zemnicí vodič, který je připojený na kolík zásuvky. Zkontrolujeme správnost zapojení a celé zařízení uzavřeme a zašroubujeme.

## 9.4 Koncovky kabelů

Zbývá nám pouze připojit volné konce vodičů do správných koncovek. Potřebujeme propojit molex z počítačového zdroje s naším vodičem signálu. Musíme si tedy sehnat samici tohoto konektoru. Černé kabely jsou GND a nás zajímá ještě kabel červený, na něm se nachází 5 V. Před zapojením je vhodné si ověřit multimetrem, zda tomu tak opravdu je. Jeden z černých kabelů tedy propojíme s vodičem, který je zapojený do GND zdířky na našem obvodu a červený s vodičem signálním. Pokud jsme odhalili holé vodiče, je potřeba zajistit, že nedojde k jejich spojení zaizolováním.

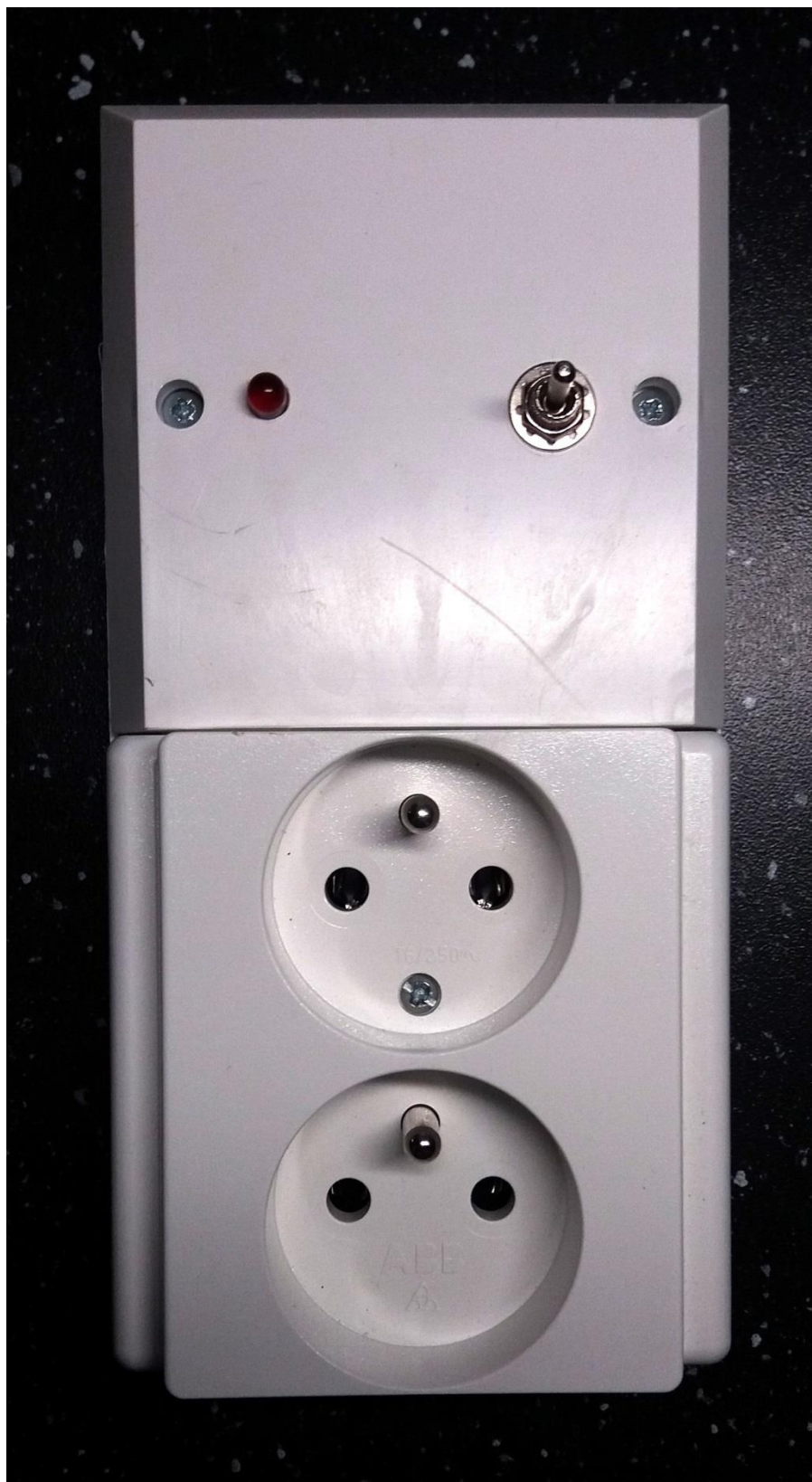
Potřebujeme zajistit napájení pro náš obvod, které musí mít konstantní napětí 5 V. Je vhodné použít standardní nabíječku na telefon, která má výstupní napětí dostatečně regulované. Opět si musíme sehnat samčí konektor USB A. Krajiní piny jsou napájecí, ale jejich polarita se z obrázků špatně určuje. Je lepší si konektor zapojit a potom proměřit, který z krajních pin je kladný a patřičným způsobem si ho označit. Po určení polarity přiletujeme napájecí vodiče našeho zařízení.

Poslední koncovkou je obyčejná vidlice síťového napětí, kterou zapojíme standardním způsobem – levý kolík fáze, pravý neutrální vodič a zdířka je pro vodič zemnicí.

## 9.5 Testování

Tímto jsme zařízení dokončili a zbývá ho jen otestovat. Na první zapojení nebudeme připojovat zařízení do síťového napětí, pouze vyzkoušíme, jestli nám relé spíná tak, jak má. Pokud je vše v pořádku, zapojíme do zásuvky testovací zátěž – např. lampičku, a připojíme naše zařízení do sítě. Opět otestujeme všechny funkce. Když se vyskytne nějaký problém, zařízení musíme co nejrychleji odpojit od síťového napětí a nezapojovat ho do té doby, než nalezneme a opravíme chybu. Nikdy nenecháváme bez dozoru zapojené zařízení, u kterého si nejsme jistí funkčností!





Obrázek 8: Zařízení bez kabelů



## 10 Závěr

Tento dokument se zabývá návrhem a výrobou spínané zásuvky, která se zapíná a vypíná společně s počítačem. Cílem bylo vytvořit zařízení, které bude automaticky zapínat a vypínat počítačovou periferii (monitor, reproduktory) zároveň s počítačem, což bylo úspěšně splněno.

Během realizace projektu jsem narazil na několik problémů. Při navrhování DPS u pájecích ploch na společné GND cestě se tyto plochy můžou se společnou cestou spojit dvěma způsoby – přímo, takže není poznat, kde pájecí plocha končí; nebo paprskovitým způsobem, a tak je jednoznačně vidět ohraničení pájecí plošky. Jako výchozí nastavení je právě paprskovitý způsob, který se mi ale vzhledově nelíbil. Bohužel až při pájení desky jsem zjistil, že způsob přímého spojení sice vypadá vizuálně lépe, ale je velký problém nanést pájku na místo, kde má být. Protože kontaktní ploška není od zbytku společné cesty fyzicky oddělená, pájka se rozlévá všude jinde, než je potřeba. Proto jsem ve finálním návrhu, prezentovaném v této práci, změnil spojení na paprskovité.

Další problém nastal při prvním testování osazené desky. Po zapojení napájení a signálu z počítače zařízení fungovalo jinak, než jsem předpokládal. Chyba spočívala v prohození signálu a napájení. I když jsem o riziku prohozeného zapojení věděl, spletl jsem se. Díky této události jsem si uvědomil, jaké katastrofální následky by mohlo mít zaměnění např. svorky pro 5V napájení s napájením síťovým. Toto riziko jsem proto v příslušné kapitole zdůraznil.

Posledním problémem bylo nedokonalé přenesení a tudíž i vyleptání jedné cesty. Po prvním testu kontinuity se zdály všechny cesty v pořádku, ale během pájení se vlivem změny teploty cesta roztáhla a v nejslabším místě se přerušila. Díky této zkušenosti jsem v práci zmínil způsob, jak tento problém vyřešit.

Kromě těchto menších nedostatků se realizace projektu obešla bez dalších problémů a zásuvka je plně funkční.

Význam tohoto zařízení spočívá nejen v usnadnění zapínání počítače a jeho periferie jedním tlačítkem, ale také spoří energii, protože při přechodu PC do úsporného režimu zůstávají reproduktory ve stále stejném zapnutém stavu, protože na rozdíl od dnešních monitorů nemají žádný stand-by režim.

## 11 Literatura

ATMEL CORPORATION. Atmel ATtiny25, ATtiny45, ATtiny85 Datasheet. Microchip Technology [online]. [cit. 2019-03-03]. Dostupné z:

[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf)

SEMICONDUCTOR COMPONENTS INDUSTRIES. 2N5550. ON Semiconductor [online]. [cit. 2019-03-03]. Dostupné z: <https://www.onsemi.com/pub/Collateral/2N5550-D.PDF>

GANESAN, Arjun. Programming ATtiny85 with Arduino Uno - Arduino Project Hub. Arduino Project Hub [online]. [cit. 2019-03-03]. Dostupné z:

<https://create.arduino.cc/projecthub/arjun/programming-attiny85-with-arduino-uno-afb829>

KONDE, Spence. GitHub - SpenceKonde/ATTinyCore: Arduino core for ATtiny 1634, 828, x313, x4, x41, x5, x61, x7 and x8. GitHub [online]. [cit. 2019-03-03]. Dostupné z:

<https://github.com/SpenceKonde/ATTinyCore>

SIMPLETRONIC. Heatless (cold) Toner Transfer for PCB Making: 10 Steps (with Pictures). Instructables [online]. [cit. 2019-03-03]. Dostupné z:

<https://www.instructables.com/id/Heatless-cold-Toner-Transfer-for-PCB-Making/>