**MATURITNÍ PRÁCE**

Obohacené zapínací tlačítko se zabezpečovací funkcionalitou

Studijní obor: IT

Třída: C4

Školní rok: 2022/2023

Jméno a příjmení: Vít Bezouška

Vedoucí práce: Radek Lampíř

**Abstrakt**

Popis zařízení zlepšující požitek ze zapínání stolního počítače. Nabízí základní zabezpečovací funkce a možnosti přizpůsobení.

**Abstract**

Describing a device used to enhance the user experience of turning on one’s PC. Some basic security and customization features are included.

**Klíčová slova**

Arduino, hardware, periferie, C++, elektronika

Prohlašuji, že jsem tuto práci vypracoval(a) samostatně a použil(a) jsem literární prameny a informace, které cituji a uvádím v seznamu použité literatury a zdrojů informací.

V Praze dne …………. ……..……………………

jméno a příjmení

Obsah

[Úvod 9](#_Toc128942336)

[Účel zařízení 10](#_Toc128942337)

[Ochrana zapnutí počítače 10](#_Toc128942338)

[Ochrana před fyzickými keyloggery 10](#_Toc128942339)

[Způsob, jak zapnout počítač bez sehnutí 11](#_Toc128942340)

[Zpříjemnění začátku pracovní doby 11](#_Toc128942341)

[Zobrazení času nebo data 11](#_Toc128942342)

[Limitace zařízení 12](#_Toc128942343)

[Zapínání počítače nelze zabezpečit 12](#_Toc128942344)

[Heslo na RFID čipu lze přepsat 12](#_Toc128942345)

[Přeprogramovatelnost Arduin 12](#_Toc128942346)

[Použité materiály 13](#_Toc128942347)

[Popis jednotlivých součástek 14](#_Toc128942348)

[Arduino Nano 14](#_Toc128942349)

[Co je COM port? 15](#_Toc128942350)

[Arduino Leonardo 15](#_Toc128942351)

[Breadboardy 16](#_Toc128942352)

[Jednopinové spoje 16](#_Toc128942353)

[LCD displej 16](#_Toc128942354)

[Reproduktor 17](#_Toc128942355)

[Červené tlačítko s krytkou 17](#_Toc128942356)

[Páčkový přepínač 17](#_Toc128942357)

[Čtečka micro SD karet a karta samotná 17](#_Toc128942358)

[PNP tranzistor 18](#_Toc128942359)

[680KΩ rezistor 18](#_Toc128942360)

[Drát do počítače a zpět 18](#_Toc128942361)

[Spínač na zámek 18](#_Toc128942362)

[LED diody 18](#_Toc128942363)

[RFID-RC522 a MIFARE čip 19](#_Toc128942364)

[Dvojitý terminál 19](#_Toc128942365)

[Malé šroubky 19](#_Toc128942366)

[Použitý software 20](#_Toc128942367)

[Word 20](#_Toc128942368)

[Arduino IDE 20](#_Toc128942369)

[PyCharm Professional 24](#_Toc128942370)

[GIMP 24](#_Toc128942371)

[image2cpp 24](#_Toc128942372)

[Audacity 25](#_Toc128942373)

[GitHub 25](#_Toc128942374)

[FontConvert 25](#_Toc128942375)

[Tinkercad 26](#_Toc128942376)

[Použité knihovny 27](#_Toc128942377)

[Adafruit GFX a SSD1306 27](#_Toc128942378)

[TMRpcm 27](#_Toc128942379)

[SpaniakosAES 27](#_Toc128942380)

[Implementace base64 27](#_Toc128942381)

[CzechKeyboard 27](#_Toc128942382)

[Easy MFRC522 27](#_Toc128942383)

[Popis jednotlivých modulů 28](#_Toc128942384)

[Obrazovkový modul 28](#_Toc128942385)

[Popis fungování 28](#_Toc128942386)

[setup() 28](#_Toc128942387)

[loop() 29](#_Toc128942388)

[Uživatel nemačká tlačítko 29](#_Toc128942389)

[Uživatel mačká tlačítko 29](#_Toc128942390)

[Audio modul je připraven 30](#_Toc128942391)

[Audio modul není připraven 30](#_Toc128942392)

[Audio modul 30](#_Toc128942393)

[Popis fungování 30](#_Toc128942394)

[setup() 30](#_Toc128942395)

[loop() 31](#_Toc128942396)

[RFID moduly 31](#_Toc128942397)

[Zapisovací (šifrovací) modul 31](#_Toc128942398)

[Popis fungování 32](#_Toc128942399)

[setup() 32](#_Toc128942400)

[loop() 32](#_Toc128942401)

[Čtecí (dešifrovací) modul 33](#_Toc128942402)

[Popis fungování 33](#_Toc128942403)

[setup() 33](#_Toc128942404)

[loop() 34](#_Toc128942405)

[Šasí 35](#_Toc128942406)

[Základ krabičky 35](#_Toc128942407)

[Stropní panel 36](#_Toc128942408)

[Boční panel 36](#_Toc128942409)

[Závěr 37](#_Toc128942410)

# Úvod

Tato práce popisuje mnou navržené a vytvořené zabezpečovací zařízení pro stolní počítač a jeho vlastnosti.

Funkcionalita je rozdělená mezi tři mikrokontrolery Arduino, konkrétně dva modely Nano a jeden model Leonardo. Jedno Nano ovládá displej, zapíná počítač a komunikuje s pythonovým scriptem. Toto Nano a k němu připojené periferie budu nazývat displejovým nebo obrazovkovým modulem. Druhé z Nan na povel displejového modulu přečte z micro SD karty audio soubor a přehraje ho. Nazývejme jej zvukovým modulem. Úkolem Arduina Leonardo je přečíst zašifrované heslo z přiloženého RFID čipu nebo karty, dešifrovat ho, a následně ho poslat jako klávesnicový vstup do počítače. To je modul čipový.

Zabezpečení samotné je zajišťováno třemi funkcemi. Zařízení zaprvé omezuje přístup k napájecímu tlačítku počítače nepovolaným osobám tak, aby nemohly počítač vypnout ani zapnout. Zadruhé pracuje jako password manager hesel uložených na RFID čipu či kartě. Poslední funkcí je možnost nastavení automatického odhlášení z uživatelského profilu v systému Windows po odejmutí čipu či karty ze čtecího senzoru.

Součástí práce bylo navrhování a sestavování hardwarových obvodů, navrhování 3D tištěného šasí, programování mikrokontrollerů Arduino v C++ a vytváření pythonového scriptu, který zprostředkovává komunikaci mezi počítačem a jedním z Arduin.

1 První vizualizace toho, jak by mohla vypadat finální podoba zařízení

# Účel zařízení

Hlavním cílem tohoto modulu je pomoct i laickému uživateli zabezpečit jeho počítač před těmi nejobecnějšími kyberhrozbami a navíc trochu zpříjemnit jeho používání.

## Ochrana zapnutí počítače

Spínač na klíč funguje jako prevence neoprávněného zapnutí. Tento způsob ochrany je sice velmi jednoduchý, ale nelze ho obejít, aniž by byly zanechány známky vniknutí. Stačí na šroubky na krabičce a od počítače nalepit bezpečnostní nálepky. Když se uživatel vrátí k počítači a uvidí, že se do něj někdo pokoušel dostat, může si okamžitě změnit hesla a zabránit tak neoprávněnému přístupu do svých účtů.

2 Příklad použití bezpečnostní nálepky k zabezpečení krabičky čokoládových bonbónů

## Ochrana před fyzickými keyloggery

Keylogger je zařízení, které si zapamatovává uživatelské vstupy aniž by je blokovalo a je schopno je poté poskytnout útočníkovi, který z nich může vyčíst citlivé údaje. Proti digitálním keyloggerům se lze bránit pomocí aktuálního antivirového programu. Existují ale také fyzické keyloggery, malá zařízení se dvěma USB porty. Do jednoho se připojí zařízení, které bude odposloucháváno a ten druhý se připojí do počítače. Zařízení si všechny vstupy ukládá do interního úložiště, nebo je přeposílá dál přes wifi.[[1]](#footnote-1)

Můj modul má v sobě integrovanou čtečku MIFARE RFID karet, ze kterých přečte zašifrované heslo, sám si ho dešifruje a výsledek pošle do počítače jako klávesnicový vstup. Je tedy také zranitelný vůči tomuto útoku. Když se ale zkombinují tyto dva způsoby zadávání hesla, může to útočníkovi ztížit získávání hesla. Například když klávesnicí uživatel zadá *hes*, mým modulem *lo*, a klávesnicí ještě *123*, útočníkovi se zobrazí *hes123* a *lo* odděleně a bude pro něj těžší získat tu správnou kombinaci.

## Způsob, jak zapnout počítač bez sehnutí

Umístí-li si uživatel modul na stůl, nebude se muset shýbat k zemi, aby si zapnul počítač.

## Zpříjemnění začátku pracovní doby

Modul po spuštění počítače přehraje náhodný zvuk z připojené micro SD karty. Uživatel si na ní může nahrát své oblíbené písničky a zlepšit si náladu ještě než se pustí do práce.

## Zobrazení času nebo data

Podle toho, co si uživatel zvolí, se na displeji může při nečinnosti zobrazovat buď aktuální datum, nebo čas. Nastavení data a času se aktualizuje na systémové hodnoty při každém spuštění počítače, není tedy třeba modul přenastavovat každých půl roku.

# Limitace zařízení

Toto zařízení není perfektní a má jisté chyby. Považujme ho za proof of concept, ne jako finální produkt. Mnoho těchto chyb je způsobeno limitacemi, které není realistické řešit v mém socioekonomickém postavení.

## Zapínání počítače nelze zabezpečit

Dokud se budou počítače zapínat pomocí zkratování dvou pinů, nebude možné jej efektivně ochránit před neoprávněnými osobami. Vždy totiž bude možné nějakým způsobem obnažit kontakty a zkratovat je.

Tohle by se dalo vyřešit spoluprací s výrobci základních desek. Kdyby totiž základní deska reagovala pouze na zaslání nějakého relativně unikátního kódu, šlo by moje zařízení vybavit zabezpečovacím čipem, který by se základní deskou komunikoval a takovýto kód poskytl.

Mohlo by se jednat o hardwarový bezpečnostní token, který by si se zapnutým počítačem vyměňoval seedy ke generování klíčů. Vypnutý počítač by pak požadoval kód, který by hardwarový token vygeneroval třeba z několika posledních seedů a aktuálního času. Útočník by potom neměl jak počítač zapnout.

## Heslo na RFID čipu lze přepsat

Jelikož jsou data zapsaná na čipy, užívané v tomto projektu, přepsatelná, útočník by je teoreticky mohl změnit anebo smazat. Ideálně by se používaly čipy s pamětí WORM[[2]](#footnote-2).

## Přeprogramovatelnost Arduin

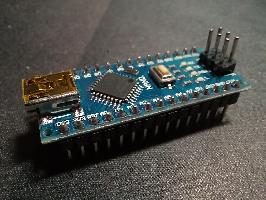
Mikrokontrolery, které používám jsou jednoduše přeprogramovatelné. Stačí je připojit k počítači a nahrát do nich jiný kód. Tento nedostatek by se dal vyřešit využitím vlastních designů mikrokontrolerů s WORM pamětí.

# Použité materiály

Na každý projekt je potřeba trocha hardwaru. Zde je seznam toho, co bylo použito na tento projekt.

|  |  |
| --- | --- |
| Počet | Součástka |
| 2 | Arduino Nano |
| 1 | Arduino Leonardo |
| 1 | Breadboard malý |
| 2 | Breadboard velký |
| >50 | Jednopinový spoj |
| 1 | IIC LCD displej 128x32px, jednobarevný |
| 1 | Reproduktor 2W 8Ω |
| 1 | Velké červené tlačítko |
| 1 | Odklápěcí kryt na velké červené tlačítko |
| 3 | Páčkový přepínač |
| 1 | Micro SD modul |
| 1 | Micro SD karta |
| 4 | PNP tranzistor |
| 1 | Malý 10KΩ rezistor |
| 2 | Kabel, který dosáhne od krabičky k PW pinům |
| 1 | Spínač na zámek (s klíčkem) |
| 4 | Libovolná LED dioda |
| 1 | RFID modul RC-522 |
| 1 | MIFARE Classic čip |
| 1 | Šroubovací terminál na dva drátky |
| 12 | Šroubků 2x10mm |

# Popis jednotlivých součástek

Důležité je vědět nejen co použít, ale i jak to použít.

3 Arduino Nano

## Arduino Nano

Nejstarší model Arduina, který se dnes běžně používá. Arduino je vlastně jen veřejně dostupná technická specifikace, podle které si kdokoliv může nechat vytvořit své vlastní Arduino. Arduina, která já používám v tomto projektu nejsou originální, jde o verzi z Číny za zlomek ceny. Pořád ale mají naprosto identickou funkcionalitu jako kdybych si koupil originální Genuino,[[3]](#footnote-3) a protože se prodejce nesnažil tvrdit, že by šlo o oficiální výrobek, je toto Arduino naprosto legální.[[4]](#footnote-4)

Arduino Nano v sobě má mikrokontroler ATmega328 o frekvenci 16 MHz a samo o sobě spotřebovává zhruba 20 mA při 5V DC. Tato spotřeba se samozřejmě zvyšuje, když k němu připojíme nějakou periferii. Každý z jeho 22 I/O[[5]](#footnote-5) pinů totiž dokáže poskytnout až 40 mA. Arduino ale nemusíme napájet přes USB port. Někdy může být vhodnější mít samostatnou baterii, ke které Arduino rovnou připojíme. V takovém případě by baterie měla mít napětí mezi 7 a 12V DC. Co se týče úložiště, má Nano k dispozici zhruba 30KB flash paměti z celkových 32, ty 2KB zabírá bootloader[[6]](#footnote-6). Operační paměť Arduina má kapacitu 2KB.Jak se s Arduinem pracuje?

Arduino Nano se k počítači připojuje do USB portu počítače pomocí mini USB kabelu. Počítači se ohlásí jako zařízení připojené přes emulovaný COM port.

### Co je COM port?

COM porty byly sériové porty, které se dříve používaly k připojení tiskáren, obrazovek a dalších periferií v dobách před USB.

4 Konektor DE-9 přes který původně probíhala komunikace COM (volné dílo)

Dnes už na počítačích tyto porty fyzicky nenajdeme, ale některé periferie si i tak vyžádají, aby jim byl přidělen virtuální COM port, protože jinak komunikovat neumí. Operační systém tedy „vytvoří“ nový COM port, který těmto zařízením přiřadí. Ten nový port sice fyzicky neexistuje, ale programy, ani připojená zařízení rozdíl nepoznají.

Proč se ale tato stará technologie pořád používá? Hlavním důvodem je, že implementace komunikace přes COM je relativně nenáročná a může probíhat i se zařízeními připojenými přes běžné USB.

Nevýhodou tohoto způsobu komunikace je velmi nízká propustnost dat, ta se pohybuje v jednotkách až desítkách kilobitů za sekundu,[[7]](#footnote-7) ale zařízení, která COM využívají, nepotřebují, aby komunikace byla *rychlá*. Naprosto jim stačí, že *nějaká vůbec je*, protože by vyšší rychlosti buď nestíhala, anebo neposílají tak velké objemy dat, aby vyšší propustnost měla nějaký znatelný dopad.

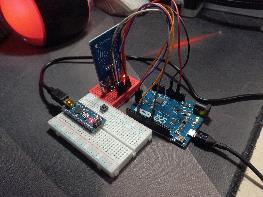
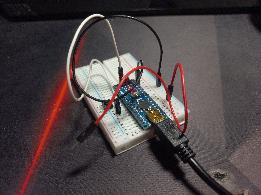
## Arduino Leonardo

5 Arduino Leonardo

Arduino Leonardo si lze představit jako větší a lepší Arduino Nano. Má o 500B větší operační paměť, pohání ho mikrokontroler ATmega32u4 a napájet ho lze 6 až 20V DC. Navíc nepotřebujeme breadboard, abychom do něj něco zapojili, drátky můžeme rovnou strkat do jeho zdířek.

Jeho hlavní výhodou je ale schopnost posílat do počítače klávesnicové vstupy a to bez použití jiných kabelů, než micro USB, kterým ho napájíme. Můžeme pomocí něj tedy ulehčit uživatelovi práci a například za něj napsat heslo, nebo pomocí klávesové zkratky Win + L zamknout jeho počítač, pokud detekuje jeho nepřítomnost.

## Breadboardy

Breadboard je destička, pomocí které se dají navrhovat a testovat obvody bez pájení. Sloty v breadboardu jsou spojené do řádků, kraje velkých breadboardů pak do sloupců. Dovolují vývojářům a kutilům zastrčit do nich součástky, vyzkoušet funkcionalitu, a pak součástky zase vyndat a použít znova. Breadboardy mívají na stranách zářezy anebo výstupky, aby se daly spojit pro jednodušší práci.

6 Příklad použití breadboardů

## Jednopinové spoje

Drátky kompatibilní s breadboardy a jinými součástkami. Dají se pořídit v jakékoliv kombinaci konektorových pohlaví.

7 Přiklad použití jednopinových spojů

## LCD displej

LCD displej, který používám má rozlišení 128x32 pixelů. Umí zobrazovat jen jednu barvu, a to světle modrou. Komunikuje s Arduinem pomocí protokolu IIC. Lze na něm zobrazovat jednotlivé pixely, čáry, obrazce, anebo rovnou obrázky, či text s vlastním fontem. Víc o zobrazování obrázků anebo používání vlastních fontů najdete v sekci použitý software.

8 Příklad použití displeje k zobrazení mého pseudonymu při startu obrazovkového modulu

## Reproduktor

9 K reproduktoru bylo potřeba připájet konektory

Reproduktor, který používám má výkon 2W a impedancí 8Ω. Výborně reprodukují vysoké frekvence, u těch nižších jsou tišší a reprodukovaný zvuk je méně kvalitní. Kvalita přehrávaného audia je ale také limitována přehrávacími schopnostmi Arduina, které dokáže z SD karty přehrát audio maximálně o 8 bitové hloubce a vzorkovací frekvenci 16KHz.

## Červené tlačítko s krytkou

Je vůbec potřeba důvod mít červené tlačítko s krytkou? To je snad sen každého kutila, někam takové tlačítko zakomponovat! Tlačítko se bez matice vloží do krytky a matice se pak nandá na závit tlačítka. Krytka pak pevně drží a chrání před nechtěným odpálením.

## Páčkový přepínač

10 "Pane prezidente, rozhodnutí je na Vás."

Běžný přepínač, který vypadá, jako z kokpitu letadla. Jeden dokáže přepínat dva oddělené obvody zároveň. Přichycovací mechanismus se silně podobá uchycovacímu mechanismu u červeného tlačítka.

## Čtečka micro SD karet a karta samotná

11 Běžná páčka, běžný přepínač.

Jeden z mála modulů, který je napájen 3.3V místo 5. Co se týče kapacity micro SD karty, úplně by stačilo 64MB. Arduino dokáže přehrát jen velmi jednoduché soubory, které nemají šanci zabrat moc místa, a to i při uložení ve formátu bez komprese, jako je .wav.

12 I dnes může několik desítek MB bohatě stačit

## PNP tranzistor

13 Kupička silikonu na nožičkách

PNP tranzistory jsou něco jako velmi jednoduchý programovací if. Propouští proud z emitteru do kolektoru jenom když je napětí v bázi zhruba o půl voltu nižší než v emitteru. Tranzistory fungují jen při nízkých napětích a nízkých proudech. Při vyšších zatíženích se používají relé.

## 680KΩ rezistor

Bez pulldown odporu na lince signálu z obrazovkového do zvukového modulu se stávalo, že zvukový modul začal sám o sobě hrát. U jiných modulů se tohle zatím nestalo.

## Drát do počítače a zpět

14 Zhruba 25m kabelu. Sestříhat a použít.

15 Jeden chybí, ten už byl použit

Je docela jedno, co za drát použijete, ale musí být dost dlouhý. Drát, který jsem použil já, byl červený CYA 1, nejlevnější druh prodávaný po metrech, vhodný víceméně na všechny nízkonapěťové účely.

## Spínač na zámek

16 Záložní klíček se vždycky hodí

Stejně jako většina visacích zámků lze i tento být jednoduše obejit, ale šlo mi spíš o estetickou hodnotu. Je to další vychytávka, která zaujme kutily.

## LED diody

17 Osvětlení jako v nightclubu

Jak jinak indikovat zapnutí, provoz, nebo funkcionalitu? Žárovky by potřebovaly víc proudu než by Arduino bylo schopno poskytnout a doutnavka běží na AC.

## RFID-RC522 a MIFARE čip

Pomocí tohoto modulu lze zapisovat i číst z kompatibilních čipů a čipových karet MIFARE CLASSIC.

## Dvojitý terminál

18 RFID modul i s čipem a čipovou kartou

Takovýto šroubovací terminál je nejjednodušší a nejspolehlivější způsob, jak připojit něco k Arduinu kabelem. Šroubek se vyšroubuje a do slotu se nacpe obnažený kabel nebo drát. Šroubek se zase zašroubuje a kabel se přiskřípne mezi konektory. I přes mírná škubání a popotahování je pak kabel dobře připojen.

19 Jeden slot volný, v druhém slotu pevně připojený kabel

## Malé šroubky

20 Netuším, co je to za kov. Možná mosaz?

Těmito šroubky jsem přišrouboval k sobě části 3D tištěné krabičky na finální produkt. Krásně se také vejdou do zdířek v Arduinu Leonardo, takže se dá nádherně připevnit ke stropu krabičky.

# Použitý software

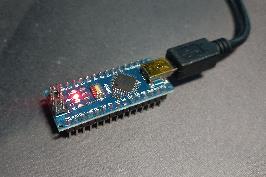
## Word

Myslím si, že nikoho nepřekvapí, že k psaní této práce byl použit Microsoft Word, konkrétně Word 2019.

## Arduino IDE

Integrované vývojové prostředí, neboli IDE je program určení k psaní dalších programů. IDE většinou podporují jen některé programovací jazyky, nebo jen některé platformy. K vývoji programů na Arduino potřebujeme takové IDE, které by nám dovolilo nejen napsat program v C++, ale i ho bezbolestně nahrát do Arduina.

Prázdný projekt v Arduino IDE

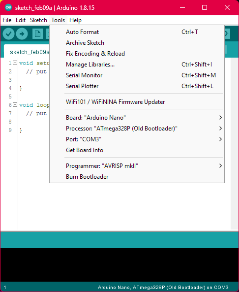
Jedno takové vyvinuli sami tvůrci Arduina. Nenabízí sice našeptávání, či jiné pokročilé funkce, ale jednoduše se přes něj nahrávají programy na Arduina. Stáhnout se dá na <https://www.arduino.cc/en/software>.

Když si IDE nainstalujeme, musíme ještě připojit samotné Arduino k počítači. Každý model má nějaký USB port, přes který ho lze jednoduše připojit. Starší modely mívají mini USB, novější mají micro USB a ty nejnovější podporují USB C.

22 Připojené Arduino Nano

Spusťme Arduino IDE. Objeví se nám šablona nového projektu s dvěma metodami, setup() a loop(). Jak napovídají komentáře, kód v metodě setup() se spustí jednou při startu Arduina a metoda loop() se bude volat donekonečna.

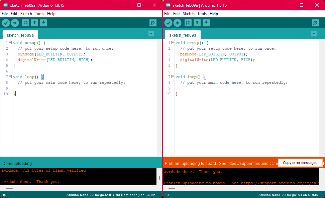
Arduino se programuje v C++. Můžeme se tedy rovnou pustit do programování. Chceme-li například, aby Arduino po zapnutí rozsvítilo LED diodu, kterou má v sobě zakomponovanou, vložíme do metody setup() následující kód:

pinMode(LED\_BUILTIN, OUTPUT);  
digitalWrite(LED\_BUILTIN, HIGH);

Tím Arduinu řekneme, že pin LED\_BUILTIN bude výstupní a že do něj chceme zapsat hodnotu HIGH, čili pustit do něj proud.

Chceme-li do Arduina kód zapsat, musíme nejdříve IDE říct, co máme za Arduino a přes který COM port s ním má komunikovat. To nastavíme v dropdown menu *Tools* na horní liště okna. Jakmile si navolíte správné nastavení, můžete kliknout na ikonku šipky vlevo nahoře. Tím dáte IDE povel zkompilovat a poslat program do Arduina.

Rozbalené menu Tools

Pokud jste zvolili správná nastavení, mělo by se Arduino rozblikat a pak by mělo rozsvítit dvě ze svých diod. Jedna indikuje zapnutí a tu druhou jste zapnuli programem.

Pokud se vám v konzolovém výstupu IDE objevila chybová hláška, není třeba panikařit. Pravděpodobně jste špatně nastavili něco v *Tools.* Vraťte se tam a zkuste jiná nastavení.

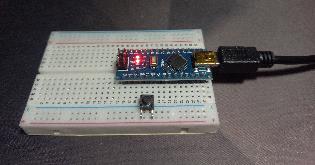
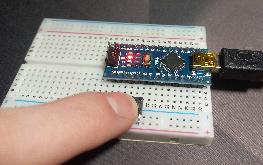
Kdybyste chtěli, aby dioda po zmáčknutí tlačítka zhasla, můžete do setup() přidat ještě:

24 Výsledek úspěšného a neúspěšného pokusu o nahrání programu

pinMode(3, INPUT\_PULLUP);

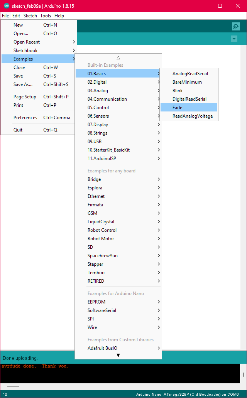
Čímž řeknete Arduinu, že pin D3 bude vstupní a dokud ho nespojíte s jedním z uzemňovacích GND pinů, tak z něj Arduino vyčte hodnotu HIGH. Do loop() pak přidejte ještě

if (digitalRead(3) == LOW){  
 digitalWrite(LED\_BUILTIN, LOW);  
}

Potom jeden pin tlačítka spojte s D3 a druhý s GND. Dokud nezmáčknete tlačítko, dioda bude svítit, ale jakmile ho zmáčknete, dioda zhasne.

25 Tlačítko zmáčknuto, dioda přestala svítit

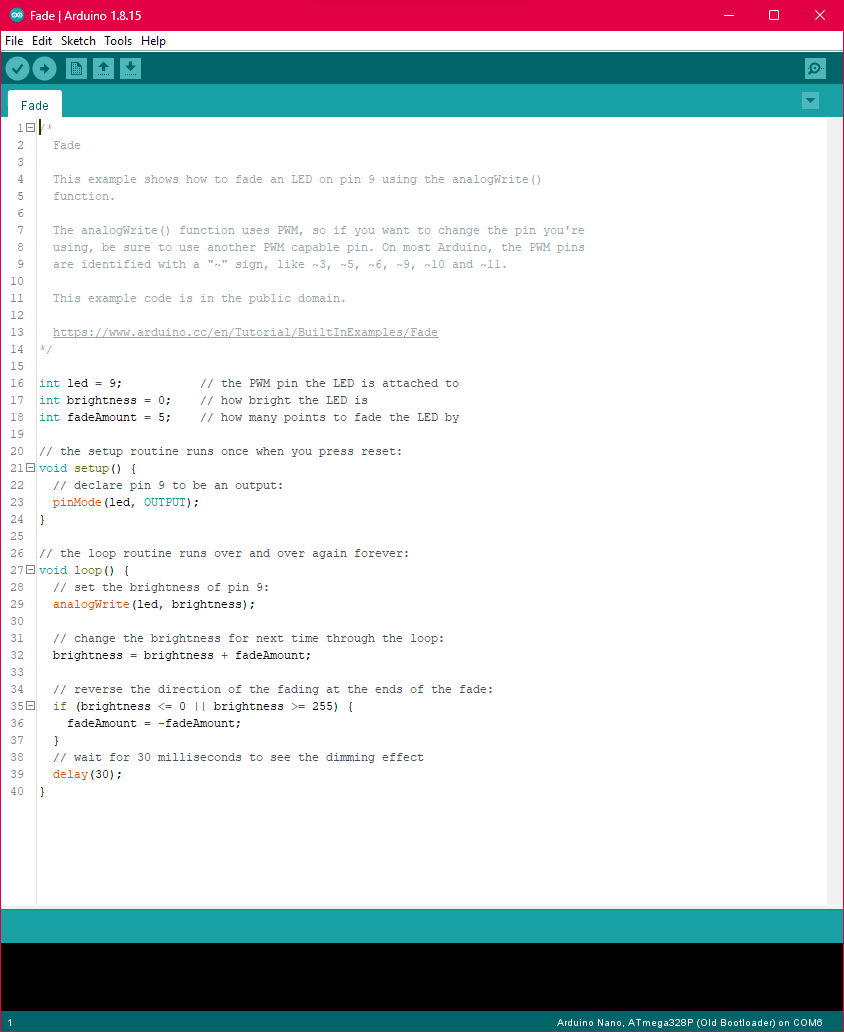
26 Tlačítko nezmáčknuto, dioda svítí

Nemusíme ale všechno programovat sami. Arduino IDE také obsahuje ukázky. V dropdown menu *File* v horní liště okna je můžeme najít pod volbou *Examples*. Ukázky mohou pocházet jak od tvůrců Arduina, tak od developerů různých knihoven.

Podívejme se třeba, jak se podle tvůrců Arduina dělá, aby LED dioda postupně zhasínala a zase se rozsvěcela. Najděte v *Examples* položku *01.Basics*, a v ní najděte *Fade*. Po rozkliknutí se vám načte už hotový projekt.

Tudy se k ukázce Fade dostaneme

Tyto projekty jsou většinou velmi dobře okomentované, aby se z nich dalo jednoduše vyčíst, která část co dělá.

 Podívejme se tedy na kód. Můžeme z něj vyčíst, že program bude do pinu 9 velmi rychle střídavě posílat a neposílat proud v poměru, který se každých 30ms zvýší o , dokud nebude na . Jakmile této hodnoty dosáhne, bude zase poměr postupně snižovat dokud nedosáhne nuly. Když k pinu 9 připojíme LED diodu, uvidíme ji postupně zhasínat a zase se rozsvěcet. Tím jsme měli možnost zjistit nejen jak funguje PWM, ale i jak ho v projektu použít.

Ukázka Fade v Arduino IDE

## PyCharm Professional

PyCharm je IDE vytvořené speciálně pro práci s Pythonem. Vyvinula jej česká společnost JetBrains, která je mimochodem také zodpovědná za vytvoření programovacího jazyka Kotlin, což je nadstavba na Javu.

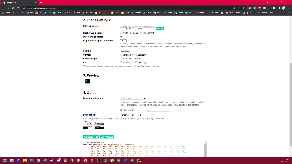
Práce v prostředí PyCharm

PyCharm jako takový má dvě varianty, Community Edition, která je zdarma a open-source, a Professional Edition, která je placená a proprietární. Liší se hlavně v pokročilých funkcích jako podpora databází a SQL, která chybí v bezplatné edici. Jako student ale mám k dispozici edici Professional zdarma.

## GIMP

Program k manipulaci s obrázky pod licencí GNU je freewarová imitace Photoshopu. Využil jsem jej hlavně k tvorbě pixel artu, který jsem pak zobrazoval na displeji obrazovkového modulu.

Práce s GIMPem

Abych redukoval množství informací, které se musí na Arduino vejít, vytvořil jsem vždy nejprve návrh jak bude nějaká scéna či animace vypadat a poté jsem ji rozdělil na jednotlivé komponenty a ty teprve nahrál na Arduino.

## image2cpp

I2CPP je jednoduchá webová aplikace, která je schopná konvertovat obrázek na bytecode, se kterým je Arduino schopno pracovat. Umí ale fungovat i obráceně, tzn. z bytecodu udělat obrázek.

Konverze malé bitmapy na kód.

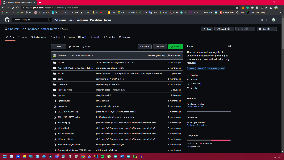
Bez tohoto programu bych musel buď všechny obrázky přepisovat na bytecode ručně, anebo se obejít bez nich.

## Audacity

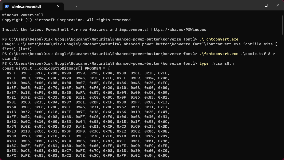
Audacity je freewarový prográmek, pomocí kterého jsem překodekovával zvukové soubory pro audio modul. Arduino totiž nemá dost rychlé CPU, aby zvládlo v reálném čase přečíst a dekomprimovat příliš náročný audio soubor.

STARTU46.WAV otevřený v Audacity

Bohužel se při zjednodušování audia pro Arduino výrazně zhorší kvalita zvuku. Snímkovací frekvence se musí snížit na 16 000 za sekundu a ukládat se mohou jen hodnoty od 0 do 255, takže nelze reproduktor ovládat tak jemně, jak by bylo potřeba.

Na druhou stranu reproduktor, který používám, není moc závratné kvality, takže i s vysokou kvalitou zdroje zvuku by to neznělo o moc lépe.

## GitHub

GitHub je společnost, u které si můžete vytvořit libovolný počet programovacích projektů, tzv. repozitářů, a poté do nich v balících posílat změny v kódu. Všechny iterace kódu jsou uloženy a lze se jednoduše vrátit k jakékoliv verzi.

Chesters-CZ/Enhanced-power-button je repozitář tohoto projektu

## FontConvert

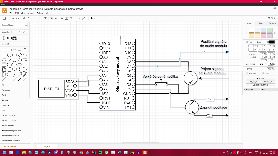
FontConvert je jednoduchý konzolový prográmek, který ze souboru s fonty přečte, jak jednotlivé znaky vypadají, a pak je konvertuje na bitmapy, které uloží ve formátu kompatibilním s grafickými knihovnami na Arduino.

Trochu návrat do éry MSDOSu

## Tinkercad

Další webová aplikace, ale tentokrát na design 3D tištěného šasí na všechny 3 moduly. Dovoluje pracovat v reálných rozměrech a následně exportovat do formátů, které podporuje většina slicerů.

Tinkercad dovoluje nejen modelovat, ale i komentovat.

Draw.io

Draw.io je bezplatná webová aplikace, která se dá stáhnout i jako lokální. Výborně se v ní dělají profesionálně vypadající schémata a flowcharty.

Rozpracované schéma obrazovkového modulu

# Použité knihovny

Tento projekt by nebylo možné reálně začít, kdyby nebylo vývojářských hrdinů, kteří naprogramovali knihovny, které jsem použil. Knihovny vytvořené vývojáři Arduin nebo programovacího jazyku, který používám zde nezmiňuji.

### Adafruit GFX a SSD1306

Tyto knihovny se starají o vykreslování grafiky a následnou komunikaci s displejem. Vyvinula je společnost Adafruit Industries, LLC, která designuje, vyrábí a prodává různé součástky k použití s Arduiny, Raspberry Pi a dalšími mikrokontrolery.

### TMRpcm

Knihovna, která si z připojené SD karty přečte audio soubor a pokusí se ho přehrát. Jméno jejího autora je neznámé, schovává se pod pseudonymem TMRh20.

### SpaniakosAES

Implementace šifrovacího standardu AES. Podporuje všechny tři verze, AES-128, AES-192 i AES-256. Jejím autorem je řecký programátor Georgios Spanos.

### Implementace base64

Base64 je způsob, jak konvertovat binární kód do textu pomocí písmen, čísel a pár znamínek. Jejím autorem je Densaugeo.

### CzechKeyboard

Upravená verze knihovny Keyboard od Arduina s upravenou konfigurací pro české keyboard layouty. Vytvořil ji Michal Altair Valášek.

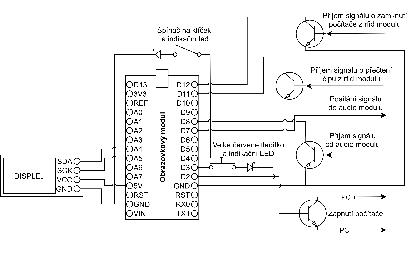
### Easy MFRC522

Wrapper knihovny MFRC522, který zjednodušuje její použití. Knihovnu MFRC522 vytvořil Miguel André Balboa. Wrapper je od Pabla Sampaio.

# Popis jednotlivých modulů

V této sekci najdete detailní popis fungování všech tří modulů i pythonového scriptu.

## Obrazovkový modul

Tento modul se stará o zapínání počítače a zobrazování informací na displeji. Komunikuje také s pythonovým scriptem na počítači, aby získal aktuální datum a čas.

37 Schéma obrazovkového modulu

Tento modul zahrnuje:

* Arduino Nano
* Displej
* 2 tranzistory
* Velké červené tlačítko

### Popis fungování

#### setup()

Arduino začne tím, že si inicializuje potřebné hodnoty a pak zobrazí hezkou animaci. Poté se pokusí přes sériovou komunikaci do počítače poslat zprávu a čeká, jestli mu přijde odpověď.

Pokud ne, znamená to, že spojení s pythonovým scriptem nebylo navázáno. Arduino o tom informuje uživatele a pokračuje metodou loop().

Pokud odpověď dostane, informuje o tom uživatele a z odpovědi si přečte jaký „spořič obrazovky“ bude používat. Tento spořič obrazovky ale doopravdy nic nešetří, jde spíš jen o využití displeje při nečinnosti. Jsou tři možnosti: odrážející se animace, která připomíná spořič obrazovky starých DVD přehrávačů, zobrazování času anebo zobrazování data.

Pokud si uživatel zvolil odrážející se animaci, Arduino nepotřebuje žádné další informace. Potvrdí přijetí zprávy a pokračuje metodou loop().

Pokud si uživatel zvolil datum nebo čas, Arduino potvrzením přijetí zprávy také zažádá počítač o informace o čase. Ty mu pythonový script pošle jako počet milisekund od půlnoci toho dne. Arduino si informaci uloží a pošle potvrzení přijetí.

Bude-li Arduino zobrazovat datum, dostane ještě informaci o datu ve formátu ISO, tzn. YYYY-MM-DD. Arduino si data zpracuje a potvrdí přijetí.

#### loop()

Arduino si nejdříve vyčistí display buffer, aby mu na displeji nezbývaly zbytky z předchozích zobrazování.

Poté zkontroluje, jestli uživatel právě mačká tlačítko.

##### Uživatel nemačká tlačítko

Arduino zkontroluje, zda od RFID modulu nepřichází nějaký signál. Pokud ano, zobrazí příslušnou hlášku spolu s ikonou. Potom pokračuje dále.

Pokud je tlačítko nezmáčknuté alespoň 100 čtecích pokusů za sebou, začne zobrazovat spořič obrazovky. Pokud to ještě nebylo 100 pokusů, zobrazí Arduino na displeji piktogram spícího Arduina, čímž uživatele informuje o blížícím se spuštění spořiče.

##### Uživatel mačká tlačítko

Arduino zkontroluje, je-li spínač na klíček odemknut. Pokud ne, zobrazí na displeji informační hlášku. Pokud ano, informuje pythonový script o statusu tlačítka, začne do počítače posílat vypínací/zapínací signál a na obrazovce ukáže piktogram prstu mačkajícího tlačítko, aby informoval uživatele o zpracování jeho akce.

Jakmile uživatel přestane tlačítko mačkat, Arduino přestane posílat do počítače signály a zkontroluje, jestli je audio modul připraven přehrát nějakou skladbu. To zjistí tak, že do něj audio modul posílá signál.

###### Audio modul je připraven

Arduino začne do audio modulu posílat signál a čeká, až to audio modul zjistí a přestane posílat signál zpátky. Do té doby zobrazuje na displeji načítací animaci.

Pokud audio modul neodpoví zhruba do tří sekund, Arduino o tom informuje uživatele přes chybovou hlášku na displeji a spustí loop() znovu.

Pokud audio modul odpoví, loop() se spustí bez chybové hlášky.

###### Audio modul není připraven

Pokud audio modul právě signál neposílá, znamená to buď že už nějakou znělku nahrává z SD karty, anebo že není zapnutý. Každopádně by povel ke spuštění písničky nezpracoval, takže Arduino žádný nepošle, ale jen oznámí uživateli proč nic nehraje.

## Audio modul

38 Schéma audio modulu

Audio modul na povel obrazovkového modulu načte náhodnou skladbu z SD karty a přehraje ji.

Tento modul zahrnuje:

* Arduino Nano
* Čtečku micro SD karet
* Reproduktor
* 2 tranzistory
* Rezistor

### Popis fungování

#### setup()

Po zapnutí si Arduino ověří, že je připojená SD karta a jestli je na ní aspoň jeden audio soubor k přehrání. Pokud ano, inicializuje stavy pinů a nastaví hlasitost přehrávaného zvuku na 5, což je nejvyšší hlasitost bez zkreslení. Pokud ne, oznámí kde se stala chyba pomocí Morseovy abecedy a dále již nepokračuje.

Nebyl-li detekován problém, Arduino si načte náhodný seed, který bude používat pro generování náhodných čísel. Nakonec už jen započne komunikaci s SD kartou a přehraje z ní soubor BOOT.WAV, čímž uživatele upozorní, že je připraven.

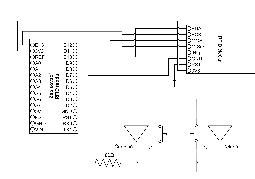
#### loop()

Na začátku loopu Arduino začne posílat do obrazovkového modulu signál. Obrazovkový modul tím ví, že je audio modul připraven zpracovat povel. Arduino potom zkontroluje, jestli od obrazovkového modulu povel už nepřišel. Pokud ne, spustí loop() znovu.

Pokud povel přišel, Arduino potvrdí obrazovkovému modulu přijetí tím, že do něj přestane posílat signál a přijetí povelu také oznámí počítači přes sériovou komunikaci. Následně pokud nějaký zvuk už hraje, zastaví jeho přehrávání a hodí si n-strannou kostkou, kde n je přednastavený počet skladeb na SD kartě. Toto číslo pak vycpe nulou, je-li třeba, a dosadí ho do STARTU\_\_.WAV. Tento soubor pak přehraje a jeho přehrání také oznámí počítači. Nutno poznamenat, že Arduino neví, jestli je reproduktor odpojen či připojen a signál do něj pouští vždy.

S dalším spuštěním metody loop() pak počká čtvrt sekundy po tom, co přestane od obrazovkového modulu dostávat signál.

## RFID moduly

Pro tento projekt je nutné buď využívat dva RFID moduly zároveň, nebo si pro nahrání dat na čipy dočasně přeprogramovat Arduino. Funkcionalita obou variant je rovnocenná, ale může být jednodušší mít k dispozici dedikovaný nahrávací modul.

39 Schéma šifrovacího modulu

### Zapisovací (šifrovací) modul

Tento modul zašifruje uložené heslo šifrovacím klíčem za použití inicializačního vektoru, zakóduje ho do base64, a zapíše na libovolný počet čipů spolu s informací, jestli po odebrání čipu zamknout počítač či ne.

Před nahráním programu na Arduino je nutné, aby uživatel upravil výchozí hodnoty v projektu na své. Šifrovaný text (heslo) může obsahovat alfanumerické znaky a symboly a musí mít délku menší než 48 znaků. Inicializační vektor může být libovolné celé číslo od 0 do zhruba 18.4 trilionů[[8]](#footnote-8)[[9]](#footnote-9) Šifrovací klíč se zadává jako text a může také obsahovat alfanumerické znaky a symboly. Jeho délka ale musí být přesně 16, 24 nebo 32 znaků. Délka šifrovacího klíče také určuje bezpečnost šifry, buďto AES128, AES192, nebo AES256.

Tento modul zahrnuje:

* Arduino Nano
* RFID modul RC522
* 2x tlačítko
* Červená LED dioda
* Zelená LED dioda

#### Popis fungování

##### setup()

Po zapnutí Arduino pouze naváže kontakt s počítačem, komunikuje-li, a inicializuje čtečku čipů.

##### loop()

Arduino inicializuje vstupní a výstupní piny a rozsvítí obě diody. Přes sériovou komunikaci pošle instrukce na používání; pokud chce uživatel, aby se jeho počítač po ztrátě spojení s čipem zamkl, má zmáčknout tlačítko u zelené diody, pokud ne, zmáčkne tlačítko u té červené.

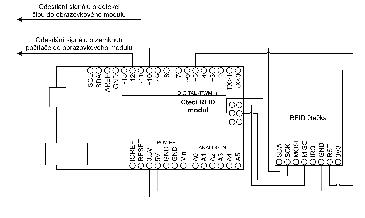
Když uživatel zvolí, Arduino zhasne diodu u nezvolené volby a zkontroluje délku šifrovacího klíče. Nabývá-li délka jedné ze správných hodnot, zašifruje heslo a výsledek zakóduje do base64, aby se dalo zapsat na čip. Pak si připraví ještě délku hesla zakódovaného v base64 a zdali se má počítač vypnout po ztrátě kontaktu s čipem, a jakmile uživatel přiloží kompatibilní čip ke čtečce, tyto informace na něj zapíše a znovu spustí loop().

Schéma dešifrovacího modulu

### Čtecí (dešifrovací) modul

Tento modul přečte zašifrované heslo z RFID čipu, pomocí přednastaveného šifrovacího klíče jej dešifruje a pošle do počítače jako klávesnicový vstup. Poté podle nastavení, které si také přečte z čipu, po odebrání čipu z dosahu čtečky buď zamkne počítač, nebo neudělá nic.

Jako u šifrovacího modulu je i tady nutné upravit program před nahráním na čip. Uživatel si ale tentokrát nemůže zvolit nový šifrovací klíč ani inicializační vektor, musí použít ty, které použil při šifrování, jinak Arduino nedešifruje heslo správně. Heslo samotné ale není třeba do tohoto modulu ukládat, Arduino ho zjistí z čipu.

Tento modul zahrnuje:

* Arduino Leonardo
* RFID modul RC522

#### Popis fungování

#### setup()

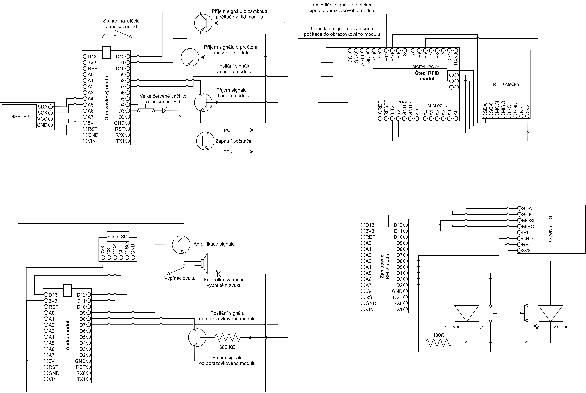
Po zapnutí se Arduino pokusí navázat sériovou komunikaci s počítačem a inicializuje RFID modul a klávesnicovou komunikaci.

#### loop()

Arduino zkontroluje délku šifrovacího klíče a podle ní se připraví na příslušnou úroveň šifry k dešifrování. Pak každých 50ms zkontroluje, jestli je v dosahu čtečky RFID čip. Jakmile je, počká ještě další desetinu sekundy, aby se mohl čip více přiblížit, a spojení bylo stabilnější. Poté si přečte všechny potřebné informace a dešifruje heslo, které pak pošle do počítače jako klávesnicový vstup.

Pokud bylo na čipu zapsáno, že se počítač má zamknout jakmile se čip odebere od čtečky, Arduino bude každých 100ms ukončovat a opět navazovat kontakt s čipem a bude ověřovat, že má pořád stejné ID.

Aby nedošlo ke zbytečnému vyrušení uživatele kvůli pár špatným pokusů o přečtení, povel zamknout se se počítači pošle až s desátým neúspěšným pokusem.

Povel samotný se pošle jako klávesová zkratka Win + L, která, alespoň na počítačích se systémem Windows 2000 a novějším,[[10]](#footnote-10) „zamkne“ počítač, tzn. odhlásí právě přihlášeného uživatele aniž by ukončila běžící programy.

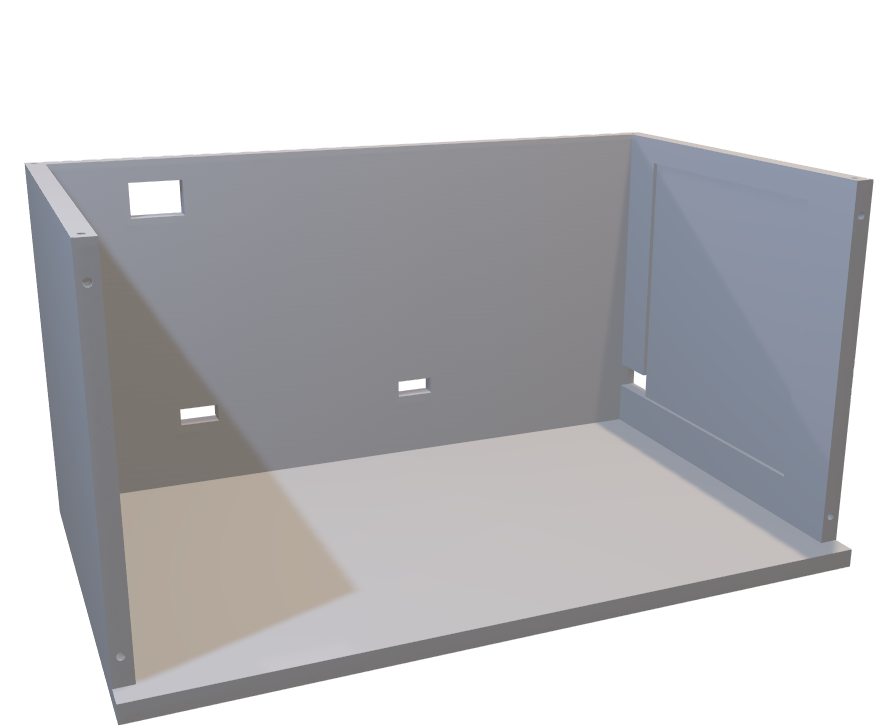
41 Schéma propojených modulů

# Šasí

Tento projekt by byl relativně úplný i bez obalu, jen jako několik součástek propojených drátky. Na druhou stranu, průměrný člověk pracující v sedavém zaměstnání by si pravděpodobně na stůl nedal něco, z čeho koukají obnažené kontakty.

Navrhl jsem tedy jednoduchý 3D model krabičky, do které jsem postupně přidával díry na různé periferie a kabeláž. Byl to můj vůbec první pokus o práci s 3D tiskárnou, a tak se mnohé nepovedlo. Některé sloty byly moc blízko u sebe, jiné moc daleko a pár jich dokonce bylo moc malých.

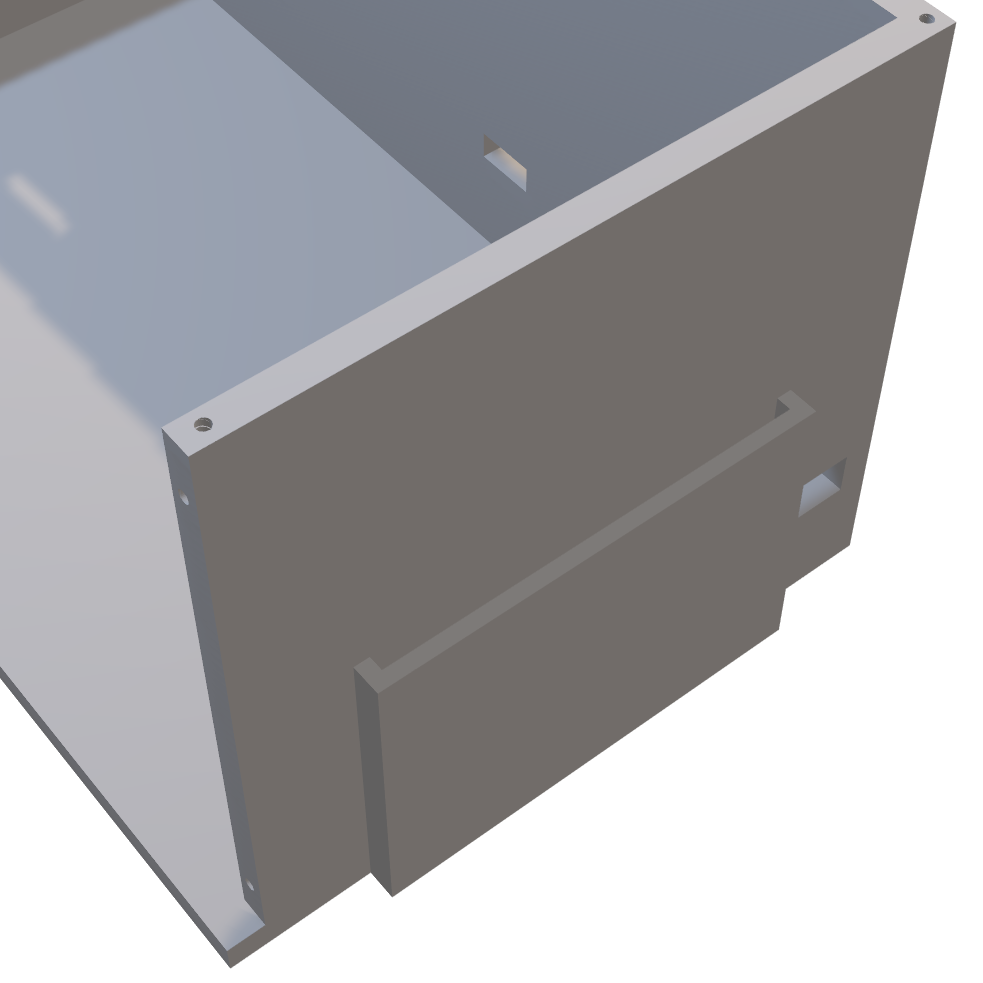
Všechny mouchy se mi podařilo vychytat bez nutnosti cokoli tisknout znovu. Díry, co chyběly jsem provrtal, ty, které byly moc malé jsem zvětšil a ty, které byly na špatném místě jsem buď nahradil, anebo jsem se obešel bez nich.

Možná jsem se mohl pokusit model vylepšit, aby byl perfektní bez jakýchkoli modifikací hrubou silou, ale vzhledem k tomu, že 3D tisk mi zajišťoval vedoucí práce z jeho dobré vůle pomocí jeho osobní 3D tiskárny a jeho filamentu, nechtěl jsem zbytečně zneužívat jeho pohostinnost.

## Základ krabičky

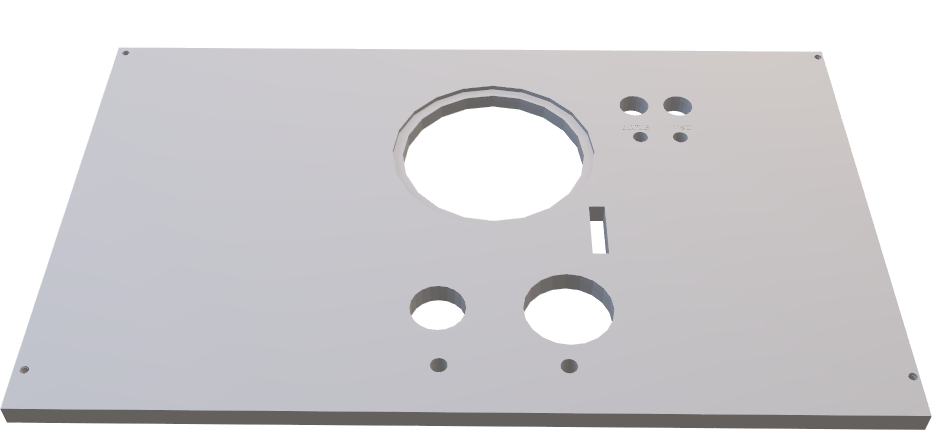
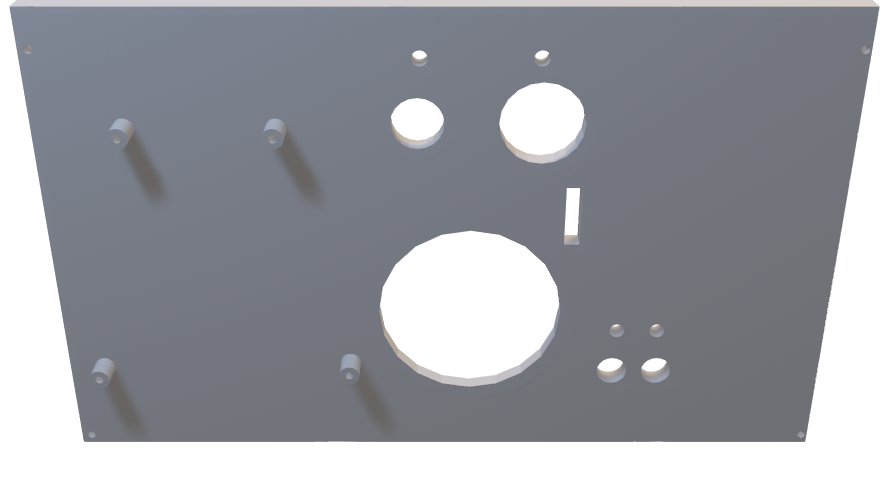
Základem krabičky je kvádr bez horní a přední strany. Aby se mi projekt lépe sestavoval do krabičky, rozhodl jsem se ponechat Arduina v breadboardech a ty pouze přilepit k podlaze krabičky. Abych ušetřil trochu místa, Arduino z RFID modulu jsem připevnil na strop krabičky. Proto je také jedna z dírek tak vysoko. Tyto dírky bylo potřeba zvětšit, protože jsem při jejich návrhu nepočítal s tloušťkou celého kabelu, nýbrž jen konektorové části portu.

Šikmý pohled na základ krabičky zepředu

Tyto dírky jsou potřeba, aby bylo možné Arduina zapojit do počítače. Celkem je potřeba mít v počítači 3 volné USB-A porty, anebo si pořídit USB hub. Ta čtvrtá osamocená dírka na pravé straně je na drátky z počítače.

Vedle této dírky si také můžete všimnout ztenčené stěny. Tam jsem se snažil o lepší propustnost elektromagnetických vln. Když se totiž podíváme na tuto stranu zvenčí, najdeme tam prostor na ponechání čipové karty.

Pohled zblízka na prostor pro čipovou kartu

V rozích jsou také umístěny malé otvory na šroubky. Tyto otvory se tiskárně ale pravděpodobně nepodaří vytisknout přesněji než malý ďůlek. To ale stačí na to, aby se tam dala opřít špička šroubku. Zbytek se tam dá našroubovat hrubou silou.

Pohled shora na stropní panel

45 Pohled na stropní panel zespodu

## Stropní panel

Hlavní vlastností tohoto panelu je několikero kulatých otvorů na reproduktor, tlačítko, spínač na klíček, displej, spínače a LED kontrolky. Zespoda jsou na panelu výstrupky na připevnění Arduina Leonardo z RFID modulu. Panel sám o sobě totiž není dost tlustý na to, aby šroubek neprošel skrz.

## Boční panel

Tento panel už jen zavírá krabičku a nemá na sobě žádné speciální výstupky.

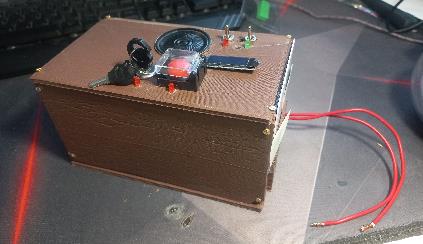
Boční panel

# Závěr

Tento projekt byl jako vztah. Když jsem na něm začal pracovat, bylo to něco, co jsem dělal jen tak pro radost. Sem tam jsem na něm zapracoval, ale žádné velké pokroky jsem neudělal, dokud jsem se nedozvěděl, že to s ním mohu někam dotáhnout. Jakmile mi byl uznán jako maturitní, začal jsem na něm pracovat dennodenně. Trávil jsem na něm čas a utrácel jsem kvůli němu peníze. Byly i týdny, kdy jsem na něm vůbec nepracoval, ale nakonec se mi povedlo to dotáhnout do šťastného konce.

47 Vnitřek krabičky. Cable management nebyl součástí tohoto projektu.

Cítím, že jsem odvedl perfektní práci na něčem, co mě opravdu bavilo. Nedovedu si představit jak bych vůbec mohl tento projekt nějak zlepšit. Ani teď, ani v průběhu práce na něm.

Odnáším si z něj mnohé. Naučil jsem se programovat v C++, vytvářet modely k 3D tisku, porozumět jednoduché elektronice a jejím principům, naučil jsem se pájet a sestavovat obvody a ještě jsem mnoho dalších dovedností oprášil. Jsem toho názoru, že jsem tím dokázal nejen, že umím použít znalosti, které jsem se naučil ve škole, ale i že jsem schopen si informace dohledávat a co je potřeba i sám sebe naučit.

48 Finální podoba tlačítka

1. Keelog. *Hardware Keylogger: AirDrive & KeyGrabber Keylogger - C64 PSU Power Supply.* USB keylogger hardware solutions - KeyGrabber - USB Keylogger, Wireless Keylogger, DIY Keylogger, Wi-Fi Keylogger, C64 PSU, C64 Power Supply, Commodore 64 [internet]. [Wroclaw], [2004-02-05], 2023/01/31 [31.1.2023]. Veřejně přístupné na keelog.com. [↑](#footnote-ref-1)
2. Write once, read many. Druh paměti, která nedovoluje měnit data, která jsou na ni zapsána. [↑](#footnote-ref-2)
3. Globální trademark, označující Arduina prodávaná společností Arduino LLC, která Arduina vyvíjí. [↑](#footnote-ref-3)
4. Arduino LLC. How to spot a counterfeit Arduino – Arduino Help Center. *Arduino*. [internet]. [3.2.2023]. Veřejně dostupné na <https://support.arduino.cc/hc/en-us/articles/360020652100-How-to-spot-a-counterfeit-Arduino> [↑](#footnote-ref-4)
5. Input/Output – Vstup/Výstup. Jak budou tyto piny použity záleží na naprogramování. [↑](#footnote-ref-5)
6. Program, který se stará o správný start systému a načtení nahraného programu. [↑](#footnote-ref-6)
7. Wikimedia Foundation, Inc. Serial port – Wikipedia. *Wikipedia: The Free Encyclopedia.* [online]. 26 October 2005, 1 December 2022 [31.1.2023] Veřejně dostupné z <https://en.wikipedia.org/wiki/Serial_port> [↑](#footnote-ref-7)
8. Jde o 64 bitový kladný integer. Jeho maximální hodnota je 264, což je 18 446 744 073 709 551 615. [↑](#footnote-ref-8)
9. UtkarshPandey6. Maximum value of unsigned long long int in C++. In: Sandeep Jain. *GeeksforGeeks.* [internet]. A-143, 9th Floor, Sovereign Corporate Tower,

   Sector-136, Noida, Uttar Pradesh – 201305: GeeksforGeeks, 03 Dec, 2020 [11.2.2023]. Veřejně dostupné na <https://www.geeksforgeeks.org/maximum-value-of-unsigned-long-long-int-in-c/> [↑](#footnote-ref-9)
10. Wikimedia Foundation, Inc. Windows key – Wikipedia. *Wikipedia: The Free Encyclopedia.* [online]. 25 April 2006, 5 January 2023 [11.2.2023] Veřejně dostupné z <https://en.wikipedia.org/wiki/Windows_key> [↑](#footnote-ref-10)