

## **MATURITNÍ PRÁCE**

Zabezpečení počítače pomocí mikrokontroleru/ů Arduino

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	8
Účel zařízení	9
Ochrana zapnutí počítače	9
Ochrana před fyzickými keyloggery	9
Automatické uzamčení počítače při odchodu	10
Způsob, jak zapnout počítač bez sehnutí	10
Zpríjemnění začátku pracovní doby	10
Zobrazení času nebo data	10
Limitace zařízení	11
Zapínání počítače nelze zabezpečit	11
Heslo na RFID čipu lze přepsat	11
Přeprogramovatelnost Arduin	11
Použité materiály	12
Popis jednotlivých součástek	13
Arduino Nano	13
Jak se s Arduinem pracuje?	13
Co je COM port?	13
Arduino Leonardo	14
Breadboardy	15
Jednopinové spoje	15
LCD displej	15
Reprodukтор	16
Červené tlačítko s krytkou	16
Páčkový přepínač	16

Čtečka micro SD karet a karta samotná	16
PNP tranzistor	17
680KΩ rezistor	17
Kabel do počítače a zpět	17
Spínač na zámek	17
LED diody	17
RFID-RC522 a MIFARE čip	18
Dvojitý terminál	18
Malé šroubky	18
Použitý software	19
Word	19
Arduino IDE	19
Programování Arduin v Arduino IDE	19
PyCharm Professional	23
GIMP	23
image2cpp	23
Audacity	24
GitHub	24
FontConvert	24
Tinkercad	25
Draw.io	25
Použité knihovny	26
Adafruit GFX a SSD1306	26
TMRpcm	26
SpaniakosAES	26

Implementace base64	26
CzechKeyboard	26
Easy MFRC522	26
Popis jednotlivých modulů	27
Obrazovkový modul	27
Popis fungování	27
setup()	27
loop()	28
Uživatel nemačká tlačítko	28
Uživatel mačká tlačítko	28
Audio modul je připraven	29
Audio modul není připraven	29
Audio modul	29
Popis fungování	29
setup()	29
loop()	30
RFID moduly	30
Zapisovací (šifrovací) modul	30
Popis fungování	31
setup()	31
loop()	31
Čtecí (dešifrovací) modul	32
Popis fungování	32
setup()	32
loop()	32

Šasí	34
Základ krabičky	34
Stropní panel	35
Boční panel	35
Závěr	36
Seznam použitých zdrojů	37
Slovník pojmu	38

## Úvod

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

Funkcionalita zařízení 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 povl 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 mikrokontrolerů 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 pomoci 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 šrouby 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.<sup>[1]</sup>

Přestože průměrný keylogger stojí okolo 1500Kč, začínají se s nimi setkávat i učitelé na základních školách. Dají se totiž objednat na dobírku přes internet. Děti jim pak získávají přihlašovací údaje učitelů, které následně zneužívají.<sup>[2]</sup> Správné použití mého zařízení tohle může přinejmenším velmi ztížit.

Používá totiž 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.

### **Automatické uzamčení počítače při odchodu**

Na čipovou kartu, na které je uložené zašifrované heslo, lze také nahrát informaci o tom, zda má RFID modul po odebrání karty ze čtečky zamknout počítač. Kdyby se pak čipová karta používala i třeba k odemknutí dveří toalety, musel by si ji uživatel vzít s sebou, a jeho počítač by pak nezůstal odemčený a zneužitelný.

### **Způsob, jak zapnout počítač bez sehnutí**

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

### **Zpríjemnění 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 kvůli letnímu času.

# Limitace zařízení

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

## 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. Při změně hesla nebo šifrovacího klíče by pak ale bylo nutné vydat uživateli nový čip.

## 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.

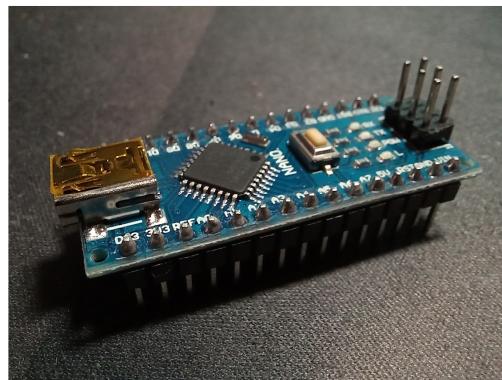
Počet	Součástka
<b>2</b>	Arduino Nano
<b>1</b>	Arduino Leonardo
<b>1</b>	Breadboard malý
<b>2</b>	Breadboard velký
<b>&gt;50</b>	Jednopinový spoj
<b>1</b>	IIC LCD displej 128x32px, jednobarevný
<b>1</b>	Reprodukтор 2W 8Ω
<b>1</b>	Velké červené tlačítko
<b>1</b>	Odklápací kryt na velké červené tlačítko
<b>3</b>	Páčkový přepínač
<b>1</b>	Micro SD modul
<b>1</b>	Micro SD karta
<b>4</b>	PNP tranzistor
<b>1</b>	Malý 680kΩ rezistor
<b>2</b>	Kabel, který dosáhne od krabičky k PW pinům
<b>1</b>	Spínač na zámek (s klíčkem)
<b>4</b>	Libovolná LED dioda
<b>1</b>	RFID modul RC-522
<b>1</b>	MIFARE Classic čip
<b>1</b>	Šroubovací terminál na dva drátky
<b>12</b>	Šroubků 2x10mm

# Popis jednotlivých součástek

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

## 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. Arduino, 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, a protože se prodejce nesnažil tvrdit, že by šlo o oficiální výrobek, je toto Arduino naprosto legální.<sup>[3]</sup>



3 Arduino Nano

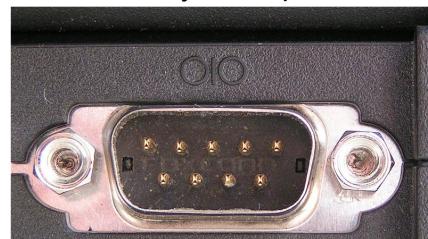
Arduino Nano v sobě má mikrokontroler ATmega328 o frekvenci 16 MHz a samo o sobě spotřebovává zhruba 20 mA při 5V DC. Tato spotřeba se samozřejmě zvyšuje, když k němu připojíme nějakou periferii. Každý z jeho 22 I/O pinů totiž dokáže poskytnout až 40 mA. Arduino ale nemusíme napájet přes USB port. Někdy 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. 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. Počítač 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,<sup>[4]</sup> 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

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 zdírek.

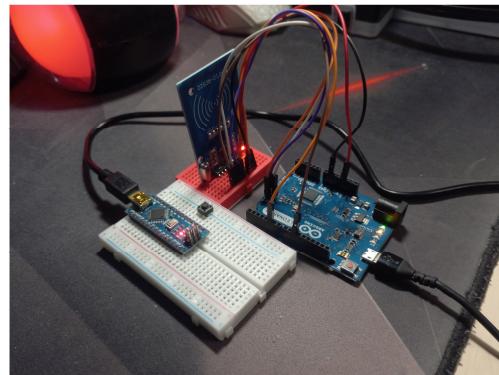


5 Arduino Leonardo

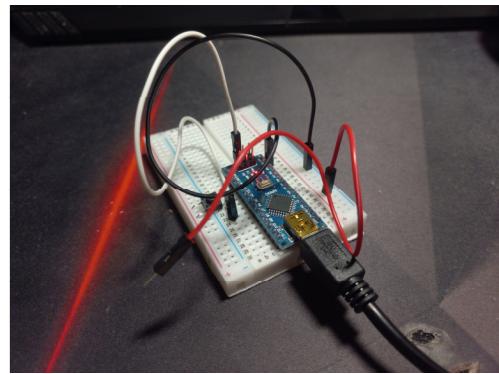
Jeho hlavní výhodou je ale schopnost posílat do počítače klávesnicové vstupy a to bez použití jiných kabelů, než micro USB, kterým ho napájíme. Můžeme pomocí něj tedy ulehčit uživateli práci a například za něj napsat heslo, nebo pomocí klávesové zkratky Win + L zamknout jeho počítač, pokud detekuje jeho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



7 Příklad použití jednopinových spojů



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

## Jednopinové spoje

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

## 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.

## Reprodukтор

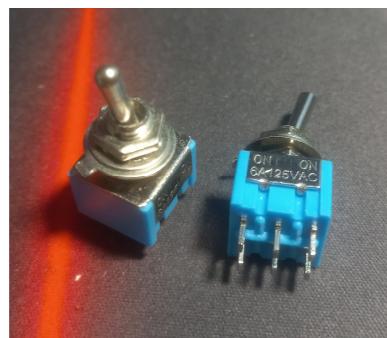
Reprodukтор, který používám má výkon 2W a impedanci 8Ω. Výborně reprodukuje vysoké frekvence, u těch nižších jsou tišší a reprodukovaný zvuk je méně kvalitní. Kvalita přehrávaného audia je ale také limitována přehrávacími schopnostmi Arduina, které dokáže z SD karty přehrát audio maximálně o 8 bitové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



9 Reproduktor s připájenými konektory



10 Tlačítko a kryt.



11 Dva přepínače



12 Čtečka micro SD karet a 128MB micro SD karta

## Červené tlačítko s krytkou

Tlačítko se bez matice vloží do krytky a matice se našroubuje na závit tlačítka. Krytka pak pevně drží a chrání tlačítko před nechtěným použitím.

## Páčkový přepínač

Běžný přepínač s nevšedním vzhledem. Jeden dokáže přepínat dva oddělené obvody zároveň. Upevňovací mechanismus se silně podobá tomu u červeného tlačítka.

## Čtečka micro SD karet a karta samotná

Jeden z mála modulů, který je napájen 3.3V místo 5V. To ale problém není, Arduina mají i třívoltový výstup. Co se týče kapacity micro SD karty, úplně by stačilo 64MB. Audio soubory, které jsou na ní uložené, jsou totiž velmi jednoduchého formátu, aby je Arduino stíhalo přehrát. Datový tok těchto souborů nepřevyšuje 8KB/s a jejich velikost je tedy velmi malá.

### PNP tranzistor

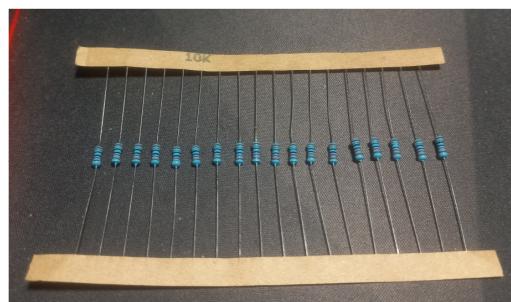
PNP tranzistory jsou něco jako velmi jednoduchý programovací *if*. Propouští proud z emitoru do kolektoru jenom když je napětí v bázi zhruba o půl voltu nižší než v emitoru.



17 Několik tranzistorů

### 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.



16 Pás rezistorů. Jeden chybí, ten už byl použit.

### Kabel do počítače a zpět

Protože napětí i proud, který kabelem bude procházet, je nízký, na volbě kabelu skoro vůbec nezáleží. Ten, který jsem použil já, byl červený CYA 1, nejlevnější druh prodávaný po metrech, vhodný víceméně na všechny nízkonapěťové účely.



15 Zhruba 25m červeného kabelu

### Spínač na zámek

Stejně jako většinu visacích zámků lze i tento jednoduše obejít, ale šlo mi spíš o estetickou hodnotu. Je to další vychytávka, která zaujmeme hlavně kutily.



14 Spínač s dvěma klíčky

### LED diody

Elektroluminiscenční diody jsou jeden z nejfektivnějších způsobů konverze elektrické energie na světlo. Navíc jsou levné a jednoduché k použití.



13 Několik svítících diod

## RFID-RC522 a MIFARE čip

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



18 RFID modul i s čipem a čipovou kartou

## Dvojitý terminál

Takovýto šroubovací terminál je nejjednodušší a nejspolehlivější způsob, jak připojit něco k Arduinu kabelem. Šroubek se vyšroubuje a do slotu se vloží 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 terminálu je volný, v druhém slotu pevně připojený kabel

## Malé šroubky

Těmito šroubkami jsem přišrouboval k sobě části 3D tištěné krabičky na finální produkt. Akorát se také vejdu do zdírek v Arduinu Leonardo, takže je relativně jednoduché ho připevnit ke stropu krabičky.



20 Krabička šroubků

# Použitý software

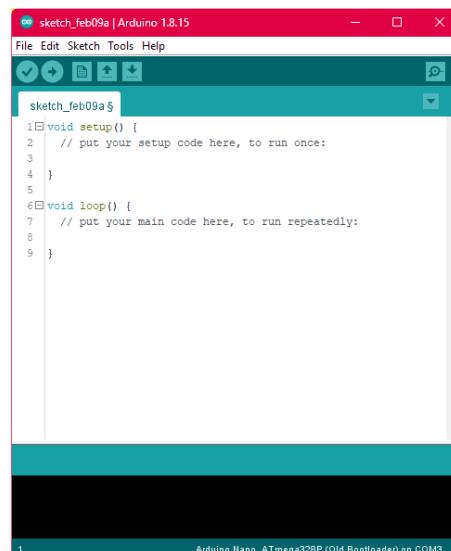
## Word

K psaní této práce byl použit Microsoft Word, konkrétně Word 2019.

## Arduino IDE

Integrované vývojové prostředí, neboli IDE je program určený 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.

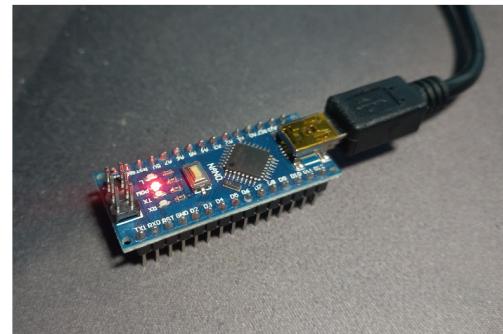
Jedno takové vyvinuli sami tvůrci Arduina. Nenabízí sice našeptávání či jiné pokročilé funkce, ale jednoduše se přes něj nahrávají programy na Arduino. Stáhnout se dá na <https://www.arduino.cc/en/software>.



21 Prázdný projekt v Arduino IDE

## Programování Arduin v Arduino IDE

Po instalaci IDE je nutné 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++. 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 navolíme správné nastavení, můžeme kliknout na ikonku šipky vlevo nahoře. Tím dáme IDE povel zkompilovat a poslat program do Arduina.

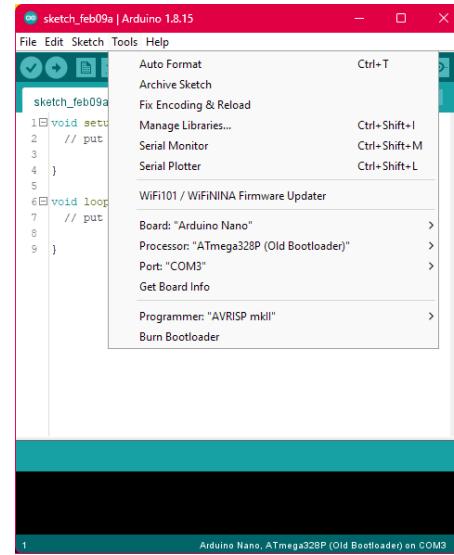
Pokud jsme zvolili správná nastavení, mělo by se Arduino rozblížit a pak by mělo rozsvítit dvě ze svých diod. Jedna indikuje zapnutí a tu druhou jsme zapnuli programem.

Vyskočila-li chybová hláška, většinou je problém v nastavení v *Tools*. Zkontrolujeme volbu Arduina a procesoru a zkusíme to znova.

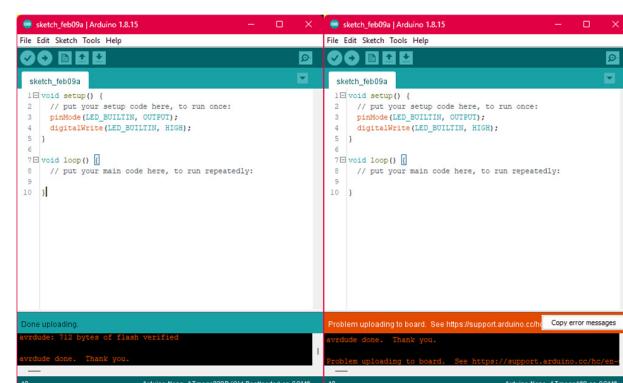
Kdybysme chtěli, aby dioda po zmáčknutí tlačítka zhasla, můžeme do metody setup() přidat ještě:

```
pinMode(3, INPUT_PULLUP);
```

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



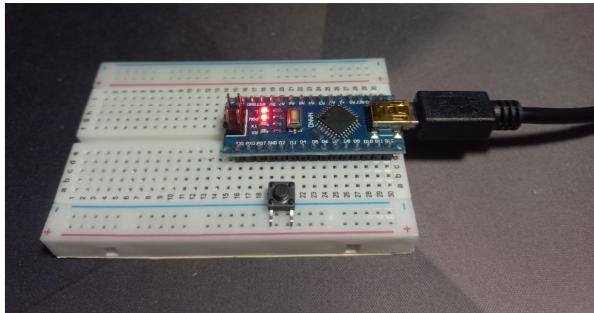
23 Rozbalené menu *Tools*



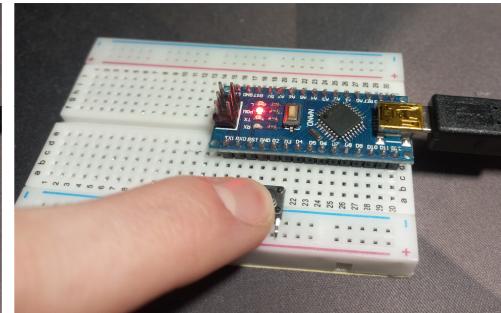
24 Výsledek úspěšného a neúspěšného pokusu o nahráni programu

```
if (digitalRead(3) == LOW) {  
    digitalWrite(LED_BUILTIN, LOW);  
}
```

Potom jeden pin tlačítka spojíme s D3 a druhý s GND. Dokud nezmáčkneme tlačítko, dioda bude svítit, ale jakmile ho zmáčkneme, dioda zhasne.



26 Tlačítko nezmáčknuto, dioda svítí

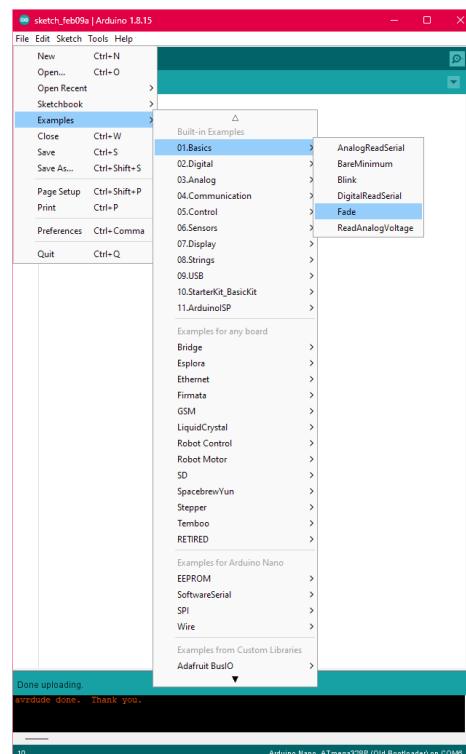


25 Tlačítko zmáčknuto, dioda přestala svítit

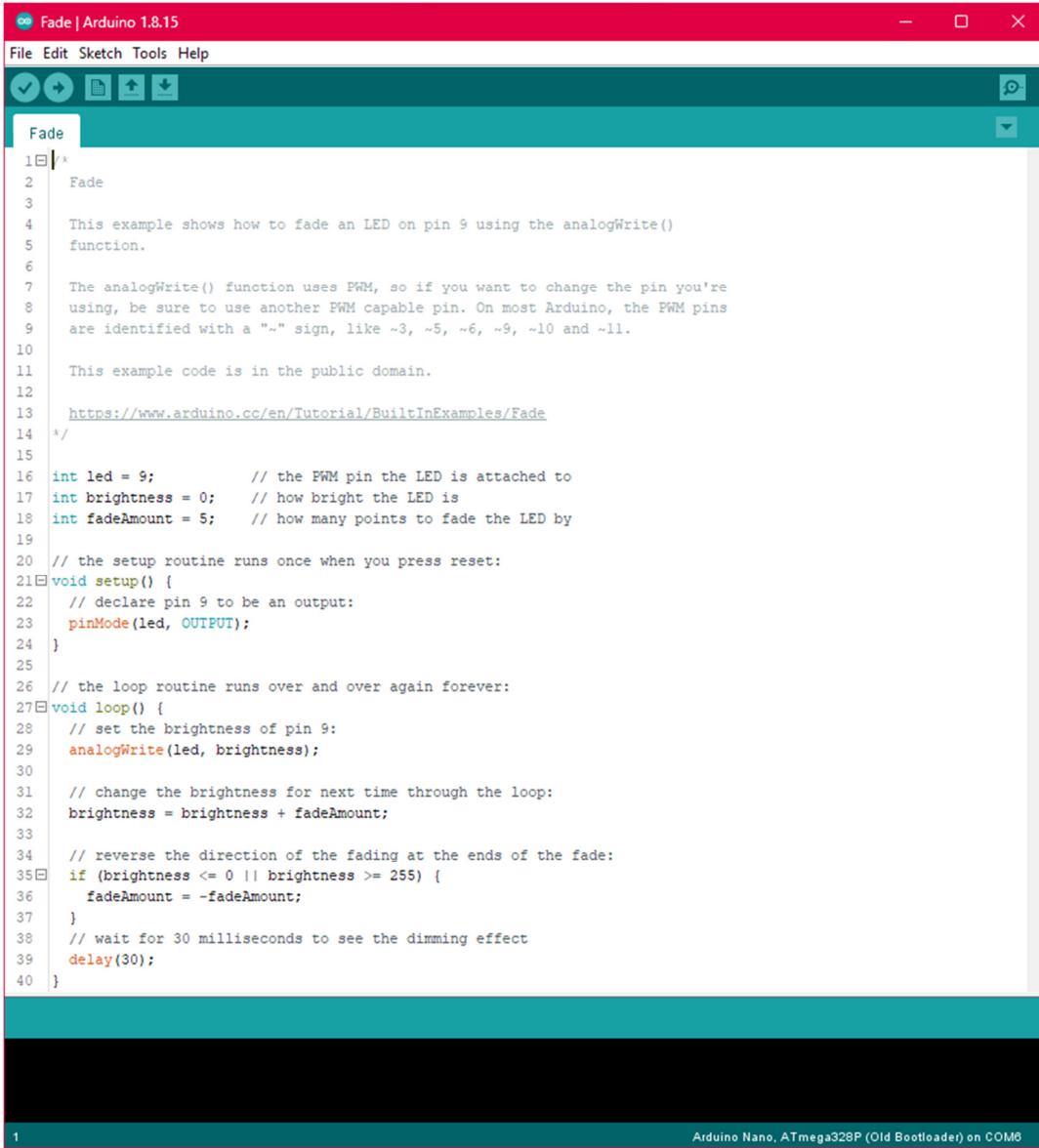
Nemusíme ale všechno programovat sami. Arduino IDE také obsahuje ukázky. V dropdown menu *File* v horní liště okna je můžeme najít pod volbou *Examples*. Ukázky mohou pocházet jak od tvůrců Arduina, tak od developerů různých knihoven.

Podívejme se třeba, jak se podle tvůrců Arduina dělá, aby LED dioda postupně zhasnala a zase se rozsvěcela. Klikneme-li v *Examples* na položku *01.Basics*, a v ní na *Fade*, načte se nám už hotový projekt.

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



27 Cesta nabídkami k ukázce *Fade*



The screenshot shows the Arduino IDE interface with the title bar "Fade | Arduino 1.8.15". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for Open, Save, Print, and others. The main window displays the "Fade" sketch code. The code is a classic example of PWM control using the analogWrite() function. It defines a pin for the LED (led = 9), initializes brightness (brightness = 0), and sets a fading amount (fadeAmount = 5). The setup() function initializes the pin mode. The loop() function repeatedly changes the brightness by the fadeAmount, checks for end points, and waits 30ms using delay(30). A status bar at the bottom indicates "Arduino Nano, ATmega328P (Old Bootloader) on COM6".

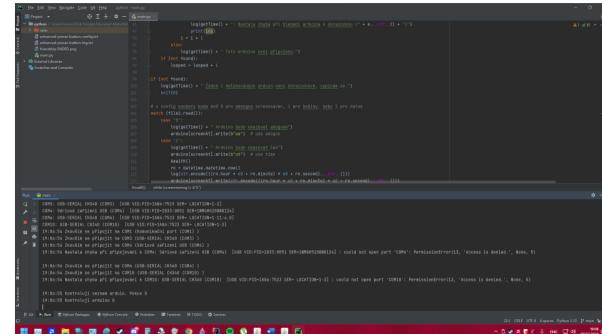
```
1 // Fades an LED attached to pin 9
2 // This example shows how to fade an LED on pin 9 using the analogWrite()
3 // function.
4
5 // The analogWrite() function uses PWM, so if you want to change the pin you're
6 // using, be sure to use another PWM capable pin. On most Arduino, the PWM pins
7 // are identified with a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.
8
9 // This example code is in the public domain.
10
11
12
13 https://www.arduino.cc/en/Tutorial/BuiltInExamples/Fade
14 */
15
16 int led = 9;          // the PWM pin the LED is attached to
17 int brightness = 0;    // how bright the LED is
18 int fadeAmount = 5;    // how many points to fade the LED by
19
20 // the setup routine runs once when you press reset:
21 void setup() {
22   // declare pin 9 to be an output:
23   pinMode(led, OUTPUT);
24 }
25
26 // the loop routine runs over and over again forever:
27 void loop() {
28   // set the brightness of pin 9:
29   analogWrite(led, brightness);
30
31   // change the brightness for next time through the loop:
32   brightness = brightness + fadeAmount;
33
34   // reverse the direction of the fading at the ends of the fade:
35   if (brightness <= 0 || brightness >= 255) {
36     fadeAmount = -fadeAmount;
37   }
38   // wait for 30 milliseconds to see the dimming effect
39   delay(30);
40 }
```

## 28 Ukázka Fade v Arduino IDE

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  $\frac{5}{255}$ , dokud nebude na  $\frac{255}{255}$ . 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 rozsvítit. Tím jsme měli možnost zjistit nejen jak funguje PWM, ale i jak ho v projektu použít.<sup>[5]</sup>

## PyCharm Professional

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

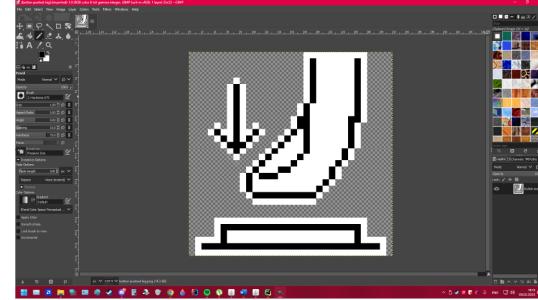


29 Práce v prostředí PyCharm

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 freewareová imitace Photoshopu. Využil jsem jej hlavně k tvorbě pixel artu, který jsem pak zobrazoval na displeji obrazovkového modulu.

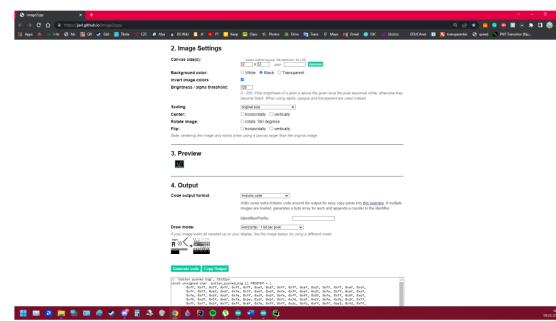


30 Práce s GIMPem

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 bytecode udělat obrázek.

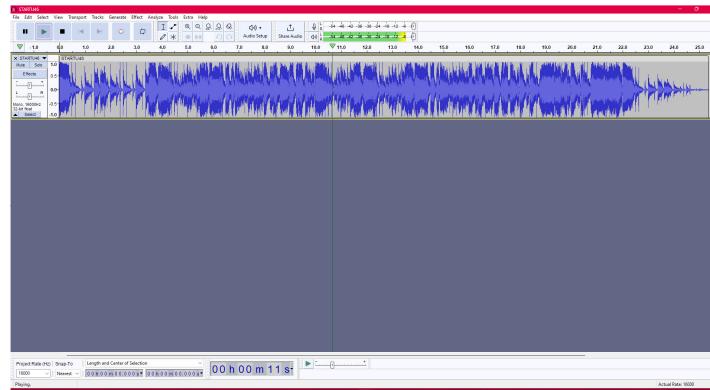


31 Konverze malé bitmapy na kód.

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

## Audacity

Audacity je freewarový program, 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.



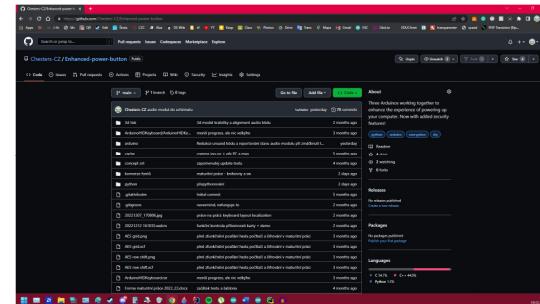
32 STARTU46.WAV otevřený v Audacity

Bohužel se při zjednodušování audia pro Arduino výrazně zhorší kvalita zvuku. Snímkovací frekvence se musí snížit na 8kHz a ukládat se mohou jen osmibitové hodnoty. To znamená, že poloha membrány reproduktoru se může změnit maximálně 8 000x za sekundu, a to na jednu z 256 poloh, takže nelze reproduktor ovládat tak jemně, jak by bylo potřeba. Například zvuk ukládaný na CD má standardně snímkovací frekvenci 44.1kHz a 16bitovou hloubku.

Na druhou stranu reproduktor, který používám, sám zvuk zkresluje, takže i s vysokou kvalitou zdrojového audia by to neznělo o moc lépe.

## GitHub

GitHub je společnost, u které si lze 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 jakékoli verzi.



33 Repozitář tohoto projektu na GitHubu

## FontConvert

FontConvert je jednoduchý konzolový program, který ze souboru s fonty přečte, jak jednotlivé znaky vypadají, a pak je

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements: https://aka.ms/PSWindows

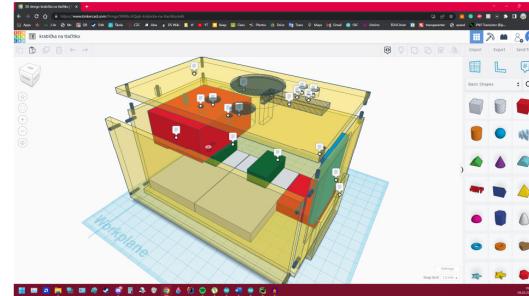
Usage: C:\Users\bezcuz\Disk Google\educanet\Material\Enhanced-power-button\konverze font> .\fontconvert.exe Fontfile size [File]
PS C:\Users\bezcuz\Disk Google\educanet\Material\Enhanced-power-button\konverze font> type ..\comic.ttf 8 > csans8h
consol: 0x00 .._comics8h700italic[1] PHONEM = 1
0xFF, 0x00, 0xF0, 0x00, 0x99, 0x99, 0x90, 0x00, 0x3B, 0x01, 0xE1, 0x0E,
0x00, 0x00,
0x00, 0x03, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

34 Práce s FontConvertem v Terminálu Windows

konvertuje na bitmapy, které uloží ve formátu kompatibilním s grafickými knihovnami na Arduino.

### Tinkercad

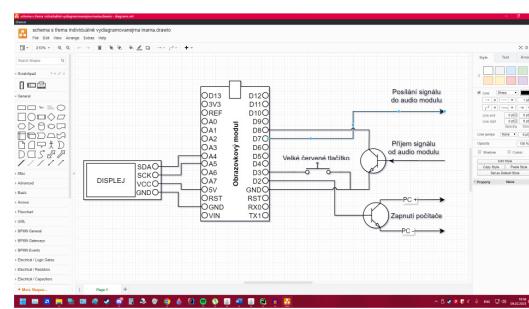
Další webová aplikace, ale tentokrát na design 3D tištěného šasí na všechny 3 moduly. Dovoluje pracovat v reálných rozměrech a následně exportovat do formátů, které podporuje většina slicerů.



35 Model včetně komentářů a pomocných bloků v prostředí Tinkercad

### 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.



36 Rozpracované schéma obrazovkového modulu

## Použité knihovny

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 Arduinou, 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 znamének. 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

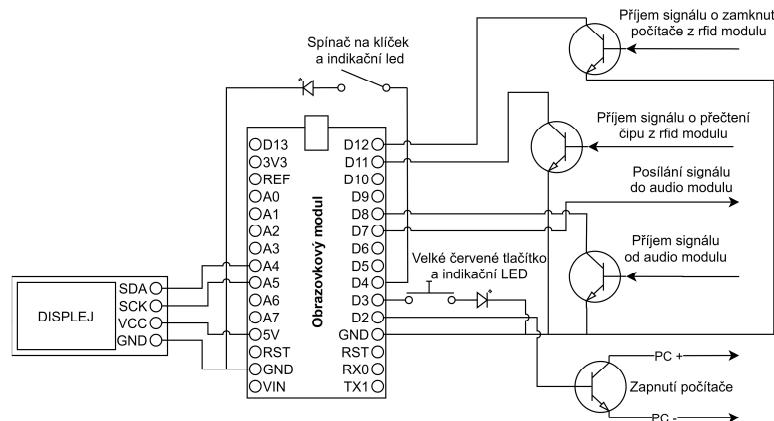
V této sekci najdete detailní popis fungování všech tří modulů i pythonového scriptu.

## 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.

Tento modul zahrnuje:

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



37 Schéma obrazovkového modulu

## 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, zda 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, co má zobrazovat na displeji při nečinnosti. 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 zkонтroluje, jestli uživatel právě mačká tlačítko.

Uživatel nemačká tlačítko

Arduino zkonzroluje, 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 zkonzroluje, 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 zkonzroluje, 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*

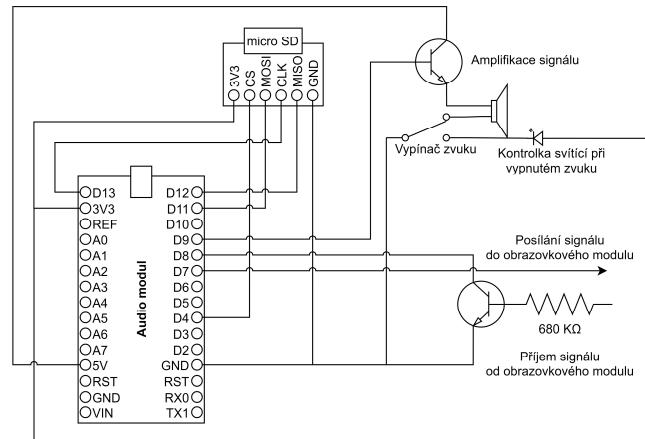
Pokud audio modul právě signál neposílá, znamená to buď že už nějakou znělku nahrává z SD karty, anebo že není zapnutý. Každopádně by povel ke spuštění písničky nezpracoval, takže Arduino žádný nepošle, ale jen oznámí uživateli proč nic nehraje.

## Audio modul

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



38 Schéma audio modulu

## 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 zkонтroluje, 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 pomyslnou n-strannou kostkou, kde n je detekovaný 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 přereprogramovat Arduino. Funkcionalita obou variant je rovnocenná, ale může být jednodušší mít k dispozici dedikovaný nahrávací modul.

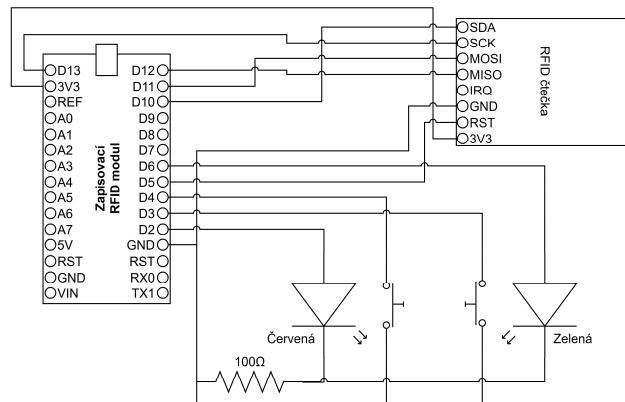
## Zapisovací (šifrovací) modul

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 trillionů<sup>1[6]</sup>. 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()

39 Schéma šifrovacího modulu

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

<sup>1</sup> Jde o 64 bitový kladný integer. Jeho maximální hodnota je  $2^{64}$ , což je 18 446 744 073 709 551 615.

s čipem, a jakmile uživatel přiloží kompatibilní čip ke čtečce, tyto informace na něj zapíše a znova spustí loop().

### Čtecí (dešifrovací) modul

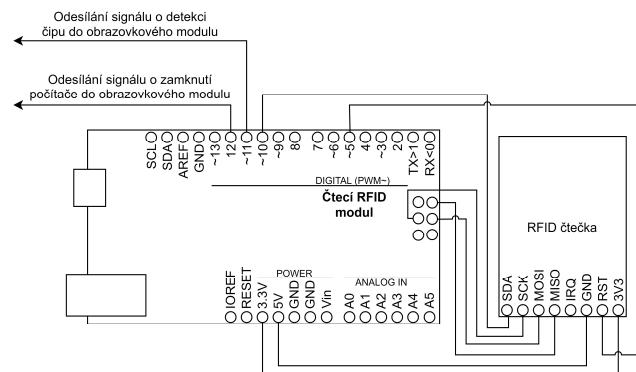
Tento modul přečte zašifrované heslo z RFID čipu, pomocí přednastaveného šifrovacího klíče jej dešifruje a pošle do počítače jako klávesnicový vstup. Poté podle nastavení, které si také přečte z čipu, po odebrání čipu z dosahu čtečky buď zamkne počítač, nebo neudělá nic.

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

40 Schéma dešifrovacího modulu

setup()

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

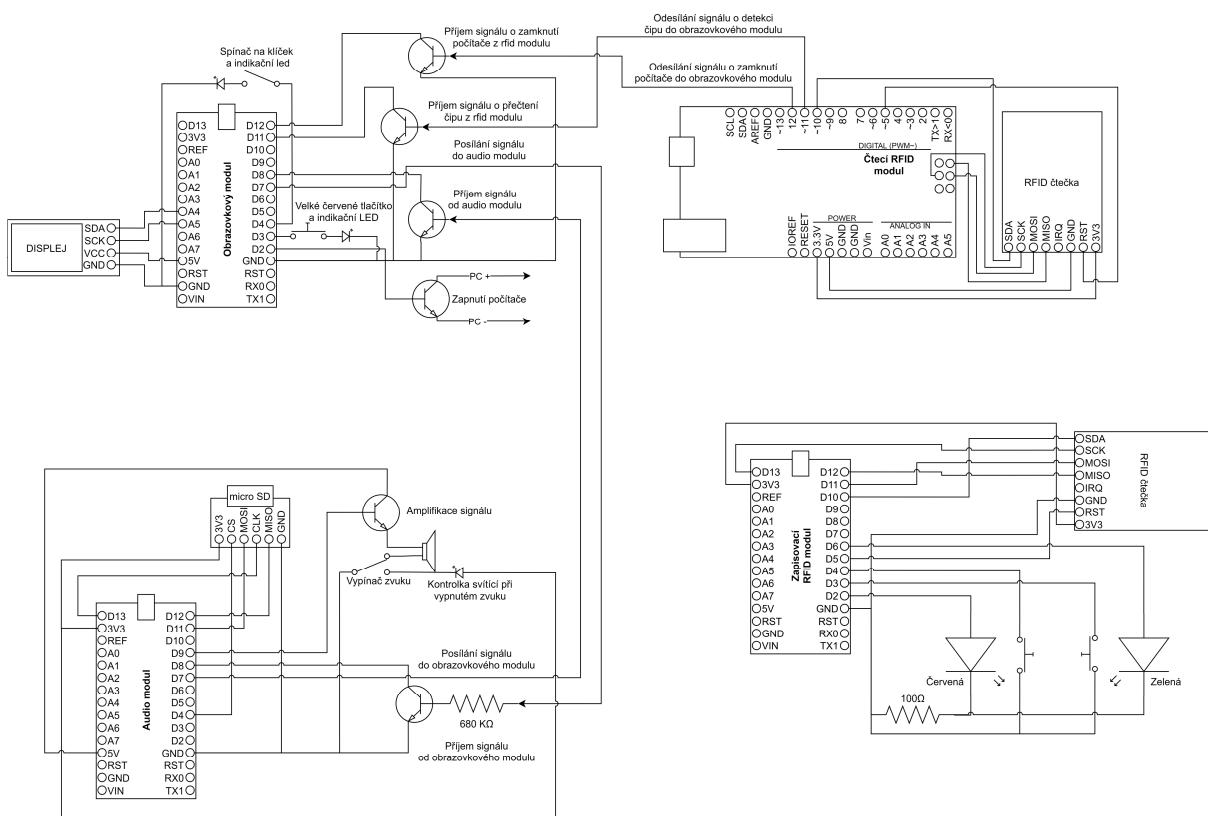
loop()

Arduino zkонтroluje délku šifrovacího klíče a podle ní se připraví na příslušnou úroveň šifry k dešifrování. Pak každých 50ms zkонтroluje, 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 stejně ID.

Aby nedošlo ke zbytečnému vyrušení uživatele kvůli pár špatným pokusům 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,<sup>[7]</sup> „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.

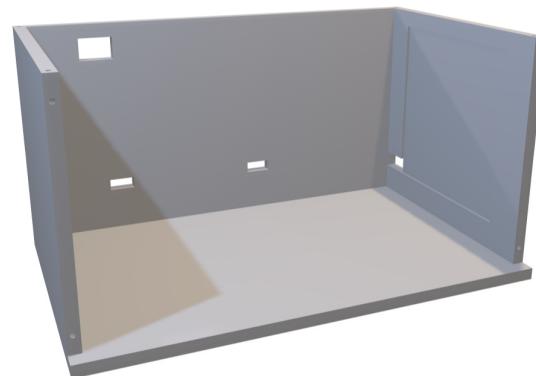
Navrhl jsem tedy jednoduchý 3D model krabičky, do které jsem postupně přidával díry na různé periferie a kabeláž. Byl to můj vůbec první pokus o práci s 3D tiskárnou, a tak se mnohé nepovedlo. Některé sloty byly moc blízko u sebe, jiné moc daleko a pár jich dokonce bylo moc malých.

Všechny mouchy se mi podařilo vychytat bez nutnosti cokoli tisknout znovu. Díry, které 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.

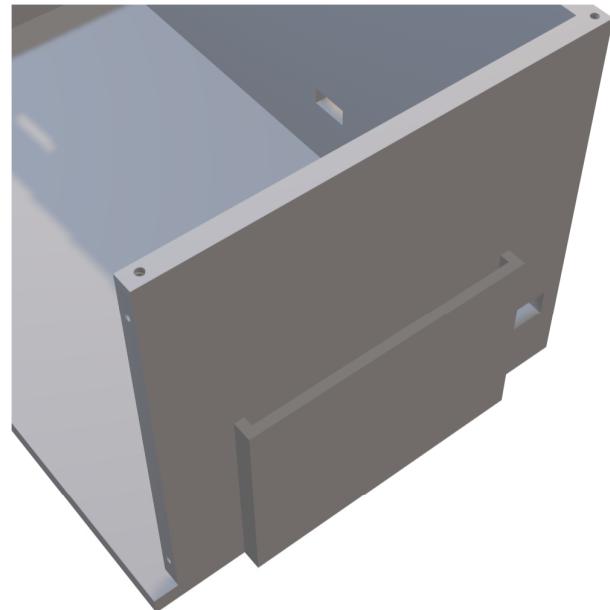


42 Šikmý pohled na základ krabičky zepředu

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.

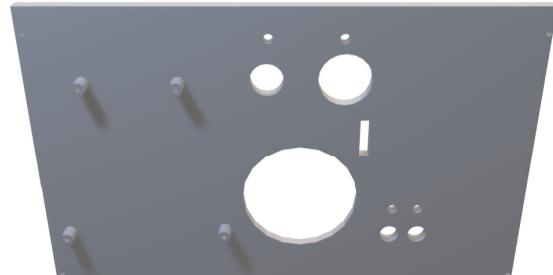
V rozích jsou také umístěny malé otvory na šrouby. Tyto otvory se tiskárně ale pravděpodobně nepodaří vytisknout přesněji než malý důlek. To ale stačí na to, aby se tam dala opřít špička šroubku. Zbytek se dá našroubovat hrubou silou.



43 Pohled zblízka na prostor pro čipovou kartu



44 Pohled shora na stropní panel



45 Pohled na stropní panel zespodu



46 Boční panel

### Boční panel

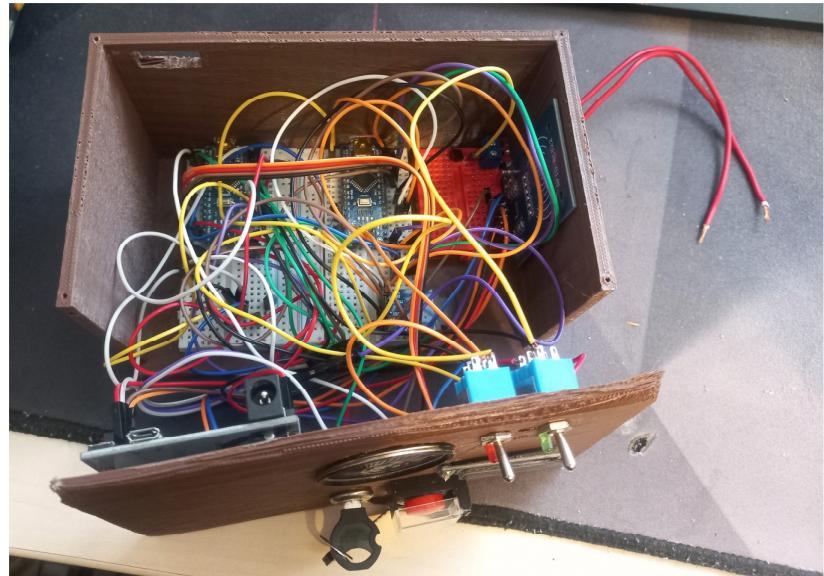
Tento panel už jen zavírá krabičku a nemá na sobě žádné speciální výstupky.

## Závěr

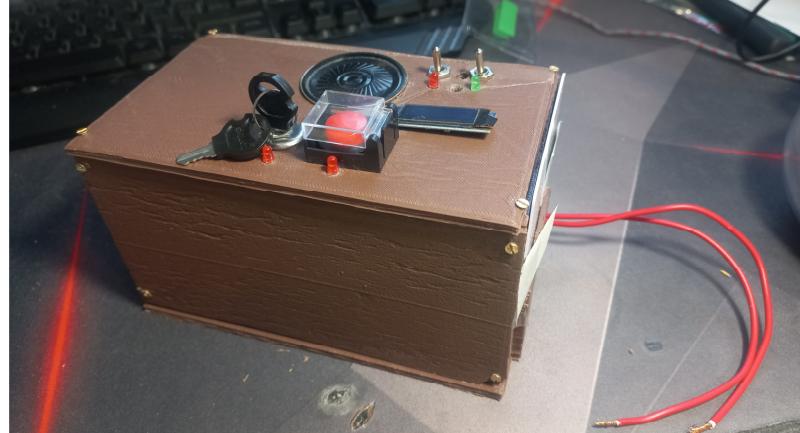
Tento projekt byl jako vztah. Když jsem na něm začal pracovat, bylo to něco, co jsem dělal jen tak pro radost. Sem tam jsem na něm zapracoval, ale žádné velké pokroky jsem neudělal, dokud jsem se nedozvěděl, že to s ním mohu někam dotáhnout. Jakmile mi byl uznán jako maturitní, začal jsem na něm pracovat 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.

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.



47 Vnitřek krabičky. Cable management nebyl součástí tohoto projektu.



48 Finální podoba tlačítka

## Seznam použitých zdrojů

- [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 [online]. [Wroclaw], [2004-02-05], 2023/01/31 [31.1.2023]. Veřejně přístupné na <https://keelog.com>
- [2]. MAFRA, a.s. Děti v roli hackerů. Školy řeší případy, kdy se jim žáci nabourají do systémů - iDNES.cz. iDNES.cz – s námi víte víc [online]. Karolína Novotná (knn). MAFRA, a. s., 28. prosince 2022, [20.3.2023]. Veřejně dostupné na [https://www.idnes.cz/zpravy/domaci/skolstvi-kyberneticky-utok-it.A221216\\_102917\\_domaci\\_knn](https://www.idnes.cz/zpravy/domaci/skolstvi-kyberneticky-utok-it.A221216_102917_domaci_knn)
- [3]. 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>
- [4]. 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](https://en.wikipedia.org/wiki/Serial_port)
- [5]. Arduino LLC. Arduino Reference - Arduino Reference. Arduino. [online]. [3.2.2023]. Veřejně dostupné na <https://reference.arduino.cc/reference/en/>
- [6]. UtkarshPandey6. Maximum value of unsigned long long int in C++. In: Sandeep Jain. GeeksforGeeks. [online]. 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/>
- [7]. 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](https://en.wikipedia.org/wiki/Windows_key)

## Slovník pojmu

Termín	Význam
AES	Advanced Encryption Standard / Pokročilý šifrovací standard. Velmi rozšířený a velmi silný standard symetrického šifrování.
Alfanumerické znaky	Číslice a písmena
Arduino	Značka programovatelných mikrokontrolerů
Arduino Leonardo	Model mikrokontroleru Arduino se schopností posílat do počítače klávesnicové vstupy
Arduino Nano	Jeden z nejzákladnějších mikrokontrolerů Arduino
base64	Způsob zapsání bytecodu jako písmena, číslice a některé znaky
báze tranzistoru	Prostřední pin tranzistoru. Pokud pustíme do báze PNP tranzistoru proud, začne téct i z kolektoru do emitoru
bezpečnostní nálepka	Nálepka, která po odloupnutí zanechá lepidlo a část potisku na povrchu na kterém byla přilepena.
Bezpečnostní token	Fyzické nebo digitální zařízení, použitelné jako náhrada hesla, nebo jako součást dvoufaktorové autentifikace. V práci je jím méněno zařízení, které ze seedu a informaci o čase vygeneruje každou minutu unikátní kód, kterým ověří svou totožnost počítači.
Bootloader	Program, který se stará o nastartování zařízení a načtení dalších instrukcí.
Buffer	Mezipaměť, do které se ukládají data, která jsou, nebo brzy budou přenášena.
Bytecode	Data, zapsaná v bytech, která jsou zpracována na počítači srozumitelný formát.
C++	Programovací jazyk
COM port	Port přes který se připojují zařízení, která potřebují jednoduchou sériovou komunikaci.
Dropdown menu	Nabídka, která se zobrazí jako seznam možností po stisknutí tlačítka.
Emitor tranzistoru	Výstupní pin tranzistoru.
Emulace	Digitální simulace nějakého prostředí či funkce za účelem vytvoření vhodných podmínek pro spuštění programu, který by bez nich neběžel správně.
Filament	Tiskový materiál do 3D tiskárny.
Flowchart	Vizualizace postupu, diagram.
Freeware	Software dostupný bezplatně.
Genuino	Globální trademark, označující Arduina prodávaná společností Arduino LLC, která Arduina vyvíjí.
GND, ground	Záporný pól, uzemnění
I/O pin	Pin, který může být použit buď jako vstupní (input) nebo jako výstupní (output)
IDE	Program, určený k psaní kódu a vývoji dalších programů.
IIC, I2C, I <sup>2</sup> C	Způsob připojení jednoduchých lowendových zařízení k sobě.
Inicializace	Prvotní nastavení hodnot.

Termín	Význam
ISO	International Organization for Standardization / Mezinárodní organizace pro standardizaci. Organizace, která vytváří a propaguje jednotné standardy.
Java	Programovací jazyk
keylogger	Zařízení nebo program, které si zapamatovává uživatelské vstupy.
klávesa Win	Klávesa s ikonou OS Windows, lidově windowsítko. Obecně nazývané jako meta tlačítka.
knihovna v programování	Kus softwaru, určený k použití jako součást jiného programu. Většinou jde o implementaci nějaké funkce, kterou by si vývojář jinak musel naprogramovat sám.
kolektor tranzistoru	Vstupní pin tranzistoru.
Kotlin	Programovací jazyk
kyberhrozba	Digitální hrozba
lokální aplikace	Aplikace, která je uložena na počítači na kterém běží.
metoda loop()	Kód, který se spouští pořád dokola po doběhnutí kódu v metodě setup()
metoda setup()	Kód, který se spustí jednou po zapnutí Arduina
micro SD karta	Malá paměťová karta
micro USB	Malý USB konektor. Předchůdce USB-C a nástupce mini USB. Používal se zejména ve starších zařízeních s Androidem.
MIFARE	Standard čipových karet a čteček, komunikujících přes RFID.
mini USB	Nejstarší malý USB konektor. Předchůdce micro USB. Používal se ve starých fotoaparátech, PDA a dalších periferiích.
open source	S veřejně dostupným zdrojovým kódem. Na tento kód se pořád mohou vztahovat autorská práva.
pin	Kontakt. Často jako kovový výstupek.
PNP tranzistor	Tranzistor, který pouští proud z kolektoru do emitoru, pokud teče proud i z báze do emitoru.
pohlaví konektorů	Princip většiny konektorů spočívá v tom, že se něco do něčeho zastrčí. Predominantně konkávní konektor je označován za samce a predominantně konvexní konektor je označován za samici. Původ této terminologie je v principu vnitřního oplodňování.
port	Samice konektoru, zdířka. Někdy přenesením významu také typ konektoru.
programovací if statement	Kus kódu, jehož spuštění je podmíněno.
proof of concept	Důkaz existence konceptu. Něco, co není finální produkt, ale demonstreuje vlastnosti, které by mohly být v nějakém produktu využity.
PWM	Pulse Width Modulation / Pulzně šířková modulace. Způsob, jak ovládat jednoduchý komponent tím, že do něj v různých intervalech budeme střídavě pouštět a nepouštět proud.
Python	Programovací jazyk
RFID	Radio Frequency IDentification / Identifikace na radiové frekvenci. Způsob bezdrátové komunikace na velmi krátké vzdálenosti.
script	Program, který se nemusí kompilovat, aby se spustil.

Termín	Význam
seed	Číslo, pomocí kterého se dá generovat nekonečná řada pseudonáhodných čísel. Generátor náhodných čísel inicializovaný stejným seedem bude pokaždé vracet stejnou řadu čísel.
slicer	Program, pomocí kterého se konvertuje 3D model na tisková data pro 3D tiskárnu.
SQL	Jazyk, kterým se ovládají některé databázové systémy.
USB	Specifikace univerzálních konektorů a protokolů.
USB-A	První USB konektor. Nejčastěji používaný pro myši, klávesnice, flashdisky a jiné periferie běžně připojované k počítačům.
USB-C	Nejnovější USB konektor. Je menší, oboustranný a určen k použití jak u periferií tak u stolních počítačů. Lze jím nahradit mnoho jiných konektorů, ale tyto funkce musí být podporovány oběma zařízeními a propojovacím kabelem.
vycpání čísla nulou	Přidání nul na začátek čísla, aby bylo dosaženo specifikované délky ve znacích. Například chci-li vypsat 16 nebo 7 na trojciferné číslo, zapíšu je jako 016 a 007.
webová aplikace	Aplikace, běžící na webové stránce.
WORM paměť	Write once, read many. Druh paměti, která nedovoluje měnit data, která jsou na ni zapsána.
zkratování	Spojení dvou kontaktů. Nemusí nutně jít o nebezpečnou situaci.
zkreslení zvuku	Nelibá úprava audiosignálu.