



Vyšší odborná škola
a Střední průmyslová škola
elektrotechnická,
Plzeň, Koterovská 85

ROČNÍKOVÁ PRÁCE S OBHAJOBOU

Téma: Inteligentní inventární systém

Autor práce: Lukáš Vágner
Třída: 3.M
Vedoucí práce: Ing. Pavel Jedlička
Dne: 11. dubna 2024

Hodnocení:



**Vyšší odborná škola a
Střední průmyslová škola elektrotechnická Plzeň,
Koterovská 85**

ZADÁNÍ ROČNÍKOVÉ PRÁCE	
Školní rok	2023/ 2024
Studijní obor	78-42-M/01 Technické lyceum
Jméno a příjmení	Lukáš Vágner
Třída	3.M
Předmět	Kybernetika
Hodnoceno v předmětu	Kybernetika
Téma	Inteligentní Inventární systémy
Obsah práce	<ul style="list-style-type: none">• Návrh velikosti a tvaru krabiček<ul style="list-style-type: none">○ Návrh designu stojanu na krabičky○ Výroba samostatného stojanu○ Výroba krabice na uložení mikropočítače a čtečky karet/čipu• Softwarová část<ul style="list-style-type: none">○ Vymyslet si schéma kódu co mají dělat jednotlivé části○ Naprogramovat a zprovoznit funkčnost celého systému zatím bez web-aplikaces• Uživatelské rozhraní<ul style="list-style-type: none">○ zavedení přihlašovacího systému pomocí kartami/čipy○ Vytvoření webového rozhraní pro sledování stavu inventárního systému○ vizualizace inventury
Zadávací učitel Příjmení, jméno	Ing. Jedlička Pavel
Podpis zadávajícího učitele	
Termín odevzdání	30. dubna 2024

Anotace

Tato ročníková práce se zaměřuje na návrh a implementaci inteligentního inventárního systému určeného pro školní dílny. Cílem systému je efektivní evidence a správa vypůjčených položek pomocí kombinace hardwarových a softwarových technologií. Text podrobně popisuje jednotlivé části systému, včetně návrhu, hardwaru a softwaru, a také proces realizace. Hardwarová část zahrnuje použití různých komponent, jako jsou NFC/RFID čtečky, senzory a LED a jejich integraci do policového systému. Na softwarové úrovni je popsáno vytvoření systému pro správu inventáře a jeho funkce, včetně automatického sledování položek a uživatelského rozhraní pro půjčování a vrácení položek. Text dále diskutuje o možných vylepšeních a úpravách systému, které by mohly zlepšit jeho uživatelskou přívětivost a efektivitu. Celkově přináší praktický návod a náhled na implementaci inteligentního inventárního systému pro školní prostředí.

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil(a) literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

V Plzni dne:

Podpis:

Obsah

1	Úvod	5
2	Návrh	5
2.1	3D design	6
3	Hardware	7
3.1	Vybrané komponenty	7
3.1.1	Raspberry Pi Zero WH	7
3.1.2	Pn532 - NFC/RFID čtečka karet	7
3.1.3	Pcf-8575 - 16bit I/O expander pinů	8
3.1.4	TCRT-5000 - Infračervený optický senzor	8
3.1.5	WS2812B - RGB LED	8
3.2	Pasivní součástky	9
4	Software	9
4.1	Main	9
4.2	Flask	10
4.3	Web	11
5	Realizace	12
5.1	Zadní část políček	12
5.2	Ovládací krabice	12
6	Závěr	14

1 Úvod

Tato ročníková práce se zaměřuje na návrh a implementaci inteligentního inventárního systému určeného pro školní dílny. Cílem systému je efektivní evidence a správa vypůjčovaných položek s využitím kombinace hardwarových a softwarových technologií.

Školní dílny jsou místem, kde studenti a učitelé často vyhledávají potřebné nástroje, součástky a další vybavení pro své projekty a experimenty. Bohužel, často dochází k nekontrolovanému vypůjčování a neúplnému navracení těchto položek, což vede k chaosu a ztrátám. Naše práce si klade za cíl tuto situaci zlepšit.

V rámci hardwarové části projektu jsme navrhli a realizovali jednoduchý policový systém, který umožňuje systematické uskladnění a identifikaci jednotlivých položek, které jsou uskladněny ve standardních krabičkách s čipem. Tento systém využívá kombinaci NFC/RFID čtečky k rozpoznání těchto čipů ale i identifikační karty uživatelů. Díky senzorům je schopen systém rozpoznat, zda je krabička na svém místě. Dále využívá inteligentní LED pro snadnou identifikaci a signalizaci stavu jednotlivých krabiček.

Na softwarové straně jsme vyvinuli jednoduchý systém pro správu inventáře, který umožňuje uživatelům jednoduché a intuitivní půjčování, vrácení a sledování položek přes lokální webové rozhraní. Díky tomuto systému mohou studenti a učitelé rychle a efektivně spravovat inventář nebo hledat kdo má jednotlivé krabičky vypůjčené, popřípadě kdo jejich obsah poškodil nebo nevrátil.

2 Návrh

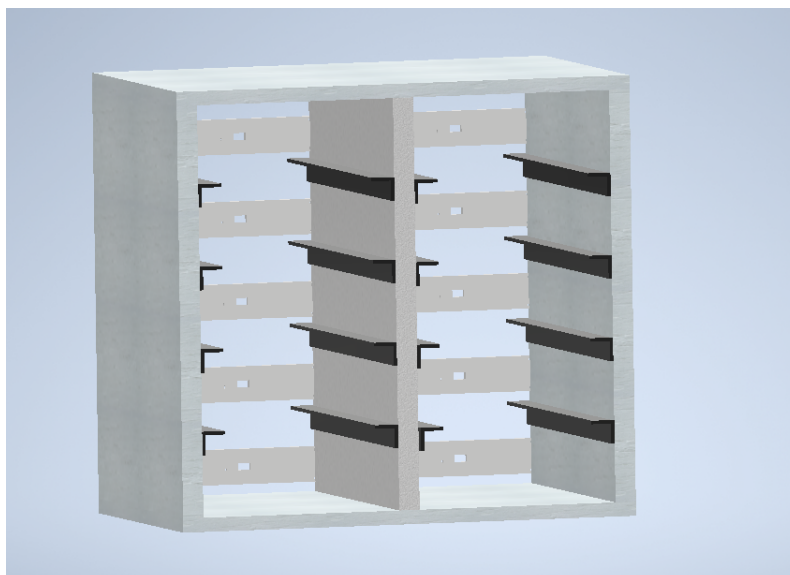
Původně jsme si představovali jednoduchý systém, který by usnadnil uskladnění krabiček a sledování jejich stavu pomocí senzorů. Vytvořili jsme několik návrhů, jak by mohla taková krabička vypadat. Celkový design policek prošel několika změnami. Nejprve jsme uvažovali o tištěných dílech, ale nakonec jsme z těchto plánů ustoupili kvůli časovým a ekonomickým omezením. Další variantou byla klasická skříňka s celodřevěnými policemi, avšak i tento návrh jsme nakonec opustili. Po důkladné úvaze jsme se shodli na nejlepším řešení: využití dřevěných desek ze všech čtyř stran a středové desky pro rozdělení boxu na dva sloupce po pěti plastových krabičkách.

Jako police jsme se rozhodli použít plastové úhelníky zakoupené v obchodě a pro zadní část jsme navrhli jednoduché tištěné díly s vestavěnými senzory a LED. Elektroniku jsme také navrhli s ohledem na jednoduchou výměnu v případě potřeby s důrazem na její cenovou dostupnost a spolehlivost.

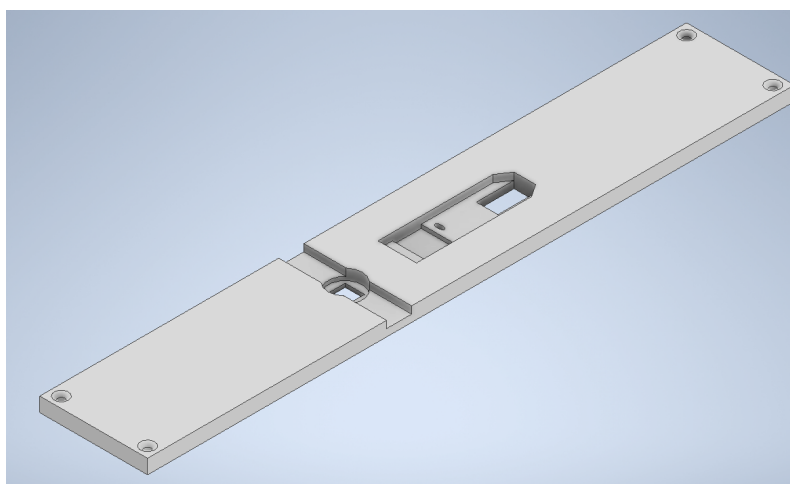
Tento proces návrhu prošel několika iteracemi, ale nakonec jsme se dostali k řešení, které odpovídalo našim požadavkům a omezením.

2.1 3D design

Návrhy všech dílů jsme dělali v Softwaru 3D CAD Inventor®[®], ve kterém jsme si z těchto dílů složili celou sestavu, tak jak by celá polička měla vypadat. Jednotlivé díly jsme vyrobili a vytiskli na tiskárně nebo nařezali pilou. Zbylé díly jsme zakoupili v obchodě, a nakonec složili podle sestavy.



Obrázek 1: Vizualizace sestavy poliček



Obrázek 2: Vizualizace zadního bracketu

3 Hardware

Při výběru hardwarových komponentů jsme se zaměřili především na hotové moduly a pasivní součástky, které byly vhodné pro náš projekt.

3.1 Vybrané komponenty

Všechny komponenty byly pečlivě zvoleny s ohledem na jejich schopnost vzájemné komunikace a kompatibilitou s napájecími systémy. Komponenty jsou schopny pracovat s napájením buď 3.3V nebo 5V, což umožňuje jejich snadnou integraci do systému. Jako hlavní komunikační protokol jsme se rozhodli pro I2C, což zajišťuje rychlou a spolehlivou komunikaci mezi jednotlivými komponenty.

3.1.1 Raspberry Pi Zero WH

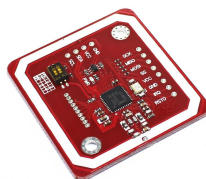
Raspberry Pi Zero WH je kompaktní jednodeskový počítač s vestavěnou anténou pro Wifi a Bluetooth (**W**), což usnadňuje připojení k internetu. Jeho GPIO header umožňuje snadné připojení dalších zařízení nebo rozšíření funkcí (**H**). (*Raspberry Pi Zero Dokumentace 2024*)



Obrázek 3: Raspberry Pi Zero WH

3.1.2 Pn532 - NFC/RFID čtečka karet

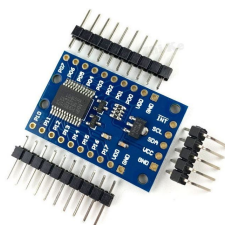
Pn532 je kompatibilní s širokou škálou NFC nebo RFID karet a čipů, což z ní činí vhodnou volbu pro naši aplikaci. Připojení pomocí I2C umožňuje snadnou komunikaci s ostatními zařízeními. (*Pn532 návod 2024*)



Obrázek 4: NFC/RFID čtečka Pn532

3.1.3 Pcf-8575 - 16bit I/O expander pinů

Pcf-8575 rozšiřuje možnosti vstupů a výstupů Raspberry Pi, což je užitečné pro projekty s více senzory. Jeho I2C rozhraní a možnost nastavení různých adres umožňuje připojení více těchto expanderů k jednomu zařízení.



Obrázek 5: 16-bit expander Pcf-8575

3.1.4 TCRT-5000 - Infračervený optický senzor

TCRT-5000 je spolehlivý infračervený senzor vhodný pro detekci objektů nebo překážek. V našem případě jednotlivých krabiček.



Obrázek 6: Optický senzor TCRT-5000

3.1.5 WS2812B - RGB LED

WS2812B LED poskytuje jednoduchou možnost signalizace různých stavů pomocí barev. Jedná se o inteligentní LED, která umožňuje nastavení různých barevných efektů a jasů. Jejich sériové zapojení umožňuje snadné ovládání a propojení s ostatními zařízeními pouze jedním datovým vstupem.



Obrázek 7: WS2812B

3.2 Pasivní součástky

Kromě hotových modulů jsme také využili několik pasivních součástek pro naši aplikaci:

- **LED** : Použili jsme 2 LED červenou a zelenou pro vizuální signalizaci různých stavů.
- **Buzzer**: Pro zvukovou signalizaci jsme zahrnuli buzzer, který je schopen generovat různé zvukové signály podle aktuálního stavu systému. Buzzer přispívá k celkové interaktivitě a uživatelskému povědomí o událostech v systému.

4 Software

Hlavní kód je psán v Pythonu. V kódu můžeme najít několik funkcí a řešení různých případů, které mohou nastat. Program běží pomocí knihovny **threads** ve dvou samostatných vláknech, jedno pro samotné přihlašování a vypůjčování / vracení krabiček a druhé pro lokální webový server, který běží pomocí knihovny **Flask**.

4.1 Main

V hlavním (Main) vlákne najdeme tyto funkce:

- **Check-card** neustálé sledování, zda byla přiložena identifikační karta nebo NFC čip nalepený na boxu. Když se tak stane, zavolá jiné funkce. Jestliže byl přiložen čip nějaké z krabiček zavolá funkci **Return-box** (viz níže). Pokud bude přiložena jakákoliv jiná karta dočasně uloží její unikátní identifikační číslo (UID).
- **Check-change** neustále hlídá, zda nějaký ze senzorů překážky nezaznamenal změnu stavu (1- krabička se zde nachází, 0 - krabička se zde nenachází). Pokud nastane změna stavu od minulého čtení, tak funkce postupně projíždí všechny možnosti, které mohou nastat. Nejdříve jestli byla krabička vypůjčena nebo vrácena.

Jestli že byla krabička odstraněna, ověřujeme, zda byl přihlášen uživatel. Pokud ano, zapíše se čas výpůjčky a identifikační číslo uživatele a vše je v pořádku. Pokud uživatel není přihlášen začne blikat podsvícení krabičky a přihlašovací zařízení. Nastane 10 vteřinová prodleva, kdy má uživatel možnost se přihlásit. Za předpokladu, že se přihlásí vše se zopakuje jako kdyby již byl přihlášen. Pokud ne, je krabička označena za ukradenou a podsvícené políčko svítí červeně, dokud se nevrátí nebo tento stav není zresetován z webu.

Druhá možnost: Krabička byla vrácena. V tomto případě systém ověří, zda předtím

byl načten čip správné krabičky, pokud ano vše je v pořádku a zapíše se čas vrácení. Pokud předtím nebyl načten správný čip, podsvícení políčka se rozsvítí červeně a označí se jako "krabička vrácená na špatné místo".

- **Return-box** funkce je vyvolána ve chvíli, kdy byl načten čip krabičky. V tu chvíli se najde v databázi, kam samotná krabička patří a rozbliká se místo kam se má krabička vrátit. Po vrácení je krabička označena za vrácenou a podsvícení se změní na zelenou.
- **Log** je zapisování jakékoliv změny do textového dokumentu například: výpůjčka, vrácení, přihlášení, ukradnutá krabička, čas načtení systému po odpojení z napájení atd...
- **Save** ukládá po každé změně aktuální podobu listu **DATA** pro případ vypnutí systému, aby se data znovu načetla a vše se obnovilo, tak jak to bylo před vypnutím.
- **Box-colors** se stará o různé podsvícení jednotlivých políček podle jejich aktuálního stavu, a k tomu použít a vypíná buzzer.

Nakonec jsou data vypsána i do konzole pythonu v podobě tabulky.

```
Data saved successfully.
```

pin	status	uid	time	Returned correctly	name
0	False		Sat Apr 6 18:27:31 2024	True	Arduina
1	False		Sat Apr 6 20:27:29 2024	True	Kablíky
2	False			True	Ledky
3	False			True	Breadboardy
4	False		Sat Apr 6 18:26:20 2024	True	Bzučáky
5	False		Sat Apr 6 18:25:22 2024	True	Demo
6	False		Sat Apr 6 18:29:01 2024	True	Demo
7	False			True	Demo
8	False			True	Demo
9	False	59717d9d	Sat Apr 6 18:27:27 2024	False	Demo

Obrázek 8: Tabulka aktuálních dat v terminálu

4.2 Flask

V tomto vlákne najdeme spuštění lokálního web-serveru a obnovování informací o aktuálním stavu sestavy. Dále funkce pro vyřizování web requestů jako třeba změna jména krabičky, vyresetování stavu a označení krabičky z webu. Ukazuje i aktuální kartu, pokud je aktivní. (*Flask Dokumentace* 2024)

4.3 Web

Web je postaven pomocí jednoduchého HTML s adaptivní tabulkou, která se nastaví podle počtu používaných senzorů automaticky. Dále jsou zde Akce: Nastavení jména krabičky, resetování stavu a prokliknutí krabičky pro jednodušší nalezení. Je zde využito i CSS k samotnému designu stránky. Nakonec už je zde využitý jen templating engine **JinJa**, který je schopný nastavovat tabulku podle potřeb. Označit řádek s problémem pomocí červené barvy. Tento engine je používán především s Python knihovnou **Flask**, kterou využívám. (*JinJa Dokumentace 2024*)

pin	status	uid	time	Returned correctly	name	Actions
0	False		Sat Apr 6 18:27:31 2024	True	Arduina	<input type="button" value="Reset Status"/> <input type="button" value="Ping Box"/> <input type="text" value="Enter new name"/> <input type="button" value="Update Name"/>
1	False	STOLEN	Wed Apr 10 22:42:00 2024	True	Kabličky	<input type="button" value="Reset Status"/> <input type="button" value="Ping Box"/> <input type="text" value="Enter new name"/> <input type="button" value="Update Name"/>
2	False			True	Ledky	<input type="button" value="Reset Status"/> <input type="button" value="Ping Box"/> <input type="text" value="Enter new name"/> <input type="button" value="Update Name"/>
3	False	Status reseted from web	Wed Apr 10 22:41:14 2024	True	Breadboardy	<input type="button" value="Reset Status"/> <input type="button" value="Ping Box"/> <input type="text" value="Enter new name"/> <input type="button" value="Update Name"/>
4	False		Sat Apr 6 18:26:20 2024	True	Bzučáky	<input type="button" value="Reset Status"/> <input type="button" value="Ping Box"/> <input type="text" value="Enter new name"/> <input type="button" value="Update Name"/>
5	False		Sat Apr 6 18:25:22 2024	True	Demo	<input type="button" value="Reset Status"/> <input type="button" value="Ping Box"/> <input type="text" value="Enter new name"/> <input type="button" value="Update Name"/>
6	False		Sat Apr 6 18:29:01 2024	True	Demo	<input type="button" value="Reset Status"/> <input type="button" value="Ping Box"/> <input type="text" value="Enter new name"/> <input type="button" value="Update Name"/>
7	False			True	Demo	<input type="button" value="Reset Status"/> <input type="button" value="Ping Box"/> <input type="text" value="Enter new name"/> <input type="button" value="Update Name"/>
8	False			True	Demo	<input type="button" value="Reset Status"/> <input type="button" value="Ping Box"/> <input type="text" value="Enter new name"/> <input type="button" value="Update Name"/>
9	False	59717d9d	Sat Apr 6 18:27:27 2024	False	Demo	<input type="button" value="Reset Status"/> <input type="button" value="Ping Box"/> <input type="text" value="Enter new name"/> <input type="button" value="Update Name"/>

Obrázek 9: Web

5 Realizace

5.1 Zadní část poliček

V zadní části poliček najdeme 10 senzorových bracketů, které slouží i jako zarážka pro jednotlivé krabičky. Dále je zde shromažďovací bit expander ze kterého všechna data jdou do ovládací krabičky.



Obrázek 10: Senzorové brackety

5.2 Ovládací krabička

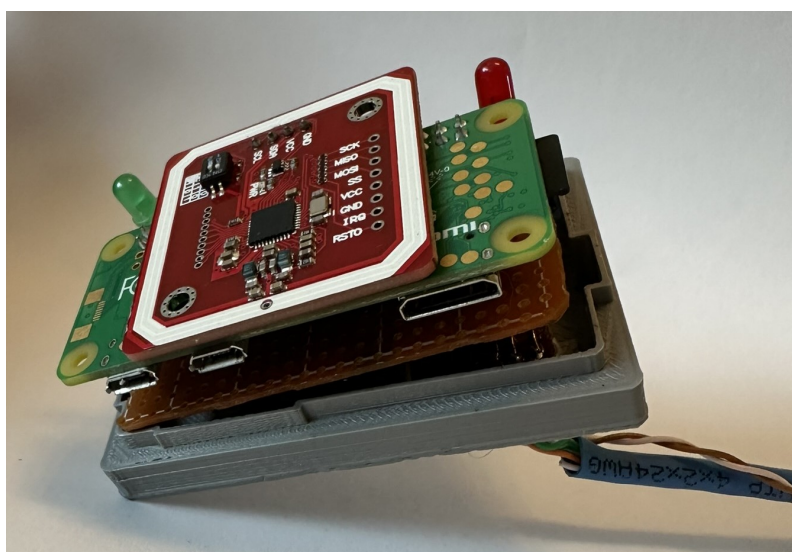
Samotné zařízení je uzavřeno ve speciálním 3D tištěném boxu, který zajišťuje ochranu jeho vnitřních komponentů. Na zadní straně boxu se nachází konektor sloužící k propojení se senzory zapojených do shromažďovače.

Vnitřek zařízení obsahuje NFC čtečku umístěnou v dolní části a připojenou k Raspberry Pi prostřednictvím pájivého pole. Samotné Raspberry Pi, také připojené na pájivém poli, funguje jako hlavní řídicí jednotky. Ta zpracovává informace z NFC čtečky a řídí celý systém.

Dvě LEDky, jedna červená a druhá zelená, jsou umístěny na vnější straně zařízení a slouží k vizuální signalizaci stavu. Bzučák pak poskytuje zvukový signál jako součást signalizace problému.



Obrázek 11: Ovládací krabička



Obrázek 12: Sestava v ovládací krabičce

6 Závěr

Naším cíle bylo vyrobit funkční inventární systém, který by řešil zapůjčování a evidenci elektronických součástek ve školních dílnách a umožňoval zobrazení aktuálního stavu na webovém rozhraní.

Náš systém obsahuje řadu funkcí, které řeší výzvy spojené se správou krabiček ve školních dílnách. Webový server poskytuje uživatelům přehledný pohled na aktuální obsah každého políčka, což usnadňuje lokalizaci potřebných součástek. Možnost označení hledaného políčka z webu, využitím barevného podsvícení při vizuální kontrole umožňuje rychlé a efektivní hledání. Kromě toho ukládání informací o posledních výpůjčkách poskytuje užitečné údaje pro identifikaci a správu vypůjčených položek pro případ poškození.

Před uvedením do provozu však bude nutné provést několik úprav. Například je zapotřebí upravit funkce pro změnu jména krabičky a resetování statusu tak, aby byli dostupné pouze po přihlášení, aby nedocházelo ke zneužití neoprávněnými lidmi.

Další vylepšení by mohlo spočívat v propojení systému se školní databází jmen a identifikačních karet. Tím by se zajistilo, že každá karta je reálně přiřazena konkrétnímu žákovi či učiteli a k údajům poté přiřadit jejich jméno.

Celkově si myslím, že náš systém je uživatelsky přívětivý a je snadný na pochopení. To je dáno zejména signalizací, která jasně ukazuje, kam má být položka vrácena, a výstražným signálem v případě, že položka není umístěna na správném místě. Taková jednoduchá a intuitivní funkcionality by mohla zvýšit efektivitu používání systému a minimalizovat chyby uživatelů.

Dalším krokem bude tento systém zprovoznit ve školních dílnách a přizpůsobit jej jak nejlépe bude možné pro všechny uživatele.



Obrázek 13: Finální podoba naší políčky

Seznam použité literatury a zdrojů informací

Flask Dokumentace (2024). URL: <https://flask.palletsprojects.com/en/3.0.x/> (cit. 10.04.2024).

Jinja Dokumentace (2024). URL: <https://jinja.palletsprojects.com/en/3.1.x/> (cit. 10.04.2024).

Pn532 návod (2024). URL: https://www.elechouse.com/elechouse/images/product/PN532_module_V3/PN532_%20Manual_V3.pdf (cit. 10.04.2024).

Raspberry Pi Zero Dokumentace (2024). URL: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#raspberry-pi-zero-w> (cit. 10.04.2024).