

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

modelGate

posuvná brána s elektrickým pohonem



Autor: Jiří Ryba
Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování
Třída: IT4
Školní rok: 2023/24

Poděkování

Rád bych poděkoval učitelům Mgr. Marcelu Godovskému a Mgr. Marku Lučnému za jejich pomoc s projektem, jelikož mi poskytovali součástky a cenné rady.

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 1. 1. 2024

.....
Podpis autora

Abstrakt

Projekt se zaměřuje na kompletní realizaci posuvné brány s elektrickým pohonem, zahrnující návrh, modelování, sestavení a programování. Je zde využit mikrokontrolér Arduino Nano, který umožňuje integraci s dálkovým ovládáním, koncovými spínači a senzorem pohybu. Výsledkem projektu je plně funkční 3D model posuvné brány, schopný reagovat na vstupy a poskytující efektivní automatizaci otevírání a zavírání.

Klíčová slova

Arduino Nano, modelování, programování, sestavení ...

Abstract

The project focuses on the complete implementation of an electrically operated sliding gate, including design, modelling, assembly and programming. An Arduino Nano microcontroller is used, which allows integration with remote control, limit switches and motion sensor. The project results in a fully functional 3D model of a sliding gate, capable of responding to inputs and providing efficient automation of opening and closing.

Key words

Arduino Nano, modelling, programming, assembling ...

Obsah

| | |
|--|-----------|
| Úvod | 5 |
| 1 Tvorba modelu | 6 |
| 1.1 3D modelování | 6 |
| 1.1.1 Výběr softwaru | 6 |
| 1.1.2 Výhody a nevýhody Autodesk Inventoru | 6 |
| 1.1.3 Modelování konstrukce | 7 |
| 1.1.4 Modelování brány | 7 |
| 1.1.5 Modelování ozubeného kolečka a ozubené lišty | 8 |
| 1.1.6 Modelování destiček a čepiček | 9 |
| 1.2 3D tisk | 10 |
| 1.3 Sestavování modelu | 10 |
| 2 Hardware | 11 |
| 2.1 Seznam použitého hardwaru | 11 |
| 2.2 Schéma hardwaru | 12 |
| 2.3 Arduino | 12 |
| 2.4 Problémy s hardwarem | 12 |
| 3 Funkce modelu brány | 13 |
| 3.1 Otevření, zavření a zastavení | 13 |
| 3.2 Senzor na pohyb | 15 |
| 3.3 LED indikátory a motor | 16 |
| 3.4 Optimalizace kódu | 17 |
| 3.5 Ovládání brány | 17 |
| Závěr | 18 |

ÚVOD

Cílem tohoto projektu bylo proměnit myšlenku v realitu. Jednoho dne, když jsem řešil stavbu skutečné posuvné brány na rodinném pozemku, mě napadlo, že bych podobný koncept mohl zkusit vytvořit jako menší projekt. Po schválení nápadu od všech učitelů jsem se do toho pustil.

Pro samotné provedení modelu jsem vytvořil 3D modely, které mi vytiskl pan učitel Mgr. Marcel Godovský na 3D tiskárně. Vytisknuté modely poskytly fyzickou podobu brány a umožnily snadné sestavení. Programování mikrokontroléru Arduino Nano bylo klíčovým krokem k dosažení ovládání brány. Vytvořil jsem řídicí algoritmus, který umožňuje plynulé otevírání a zavírání brány v reakci na externí podněty, například signály ze snímačů nebo dálkového ovladače.

Dokumentace obsahuje použité komponenty, zapojení a způsob programování mikrokontroléru. Kromě toho jsou zahrnuty i možné vylepšení projektu. Tato dokumentace slouží jako průvodce pro každého, kdo má zájem o pochopení a reprodukci podobného projektu posuvné brány s využitím Arduino technologie.

1 TVORBA MODELU

1.1 3D MODELOVÁNÍ

1.1.1 Výběr softwaru

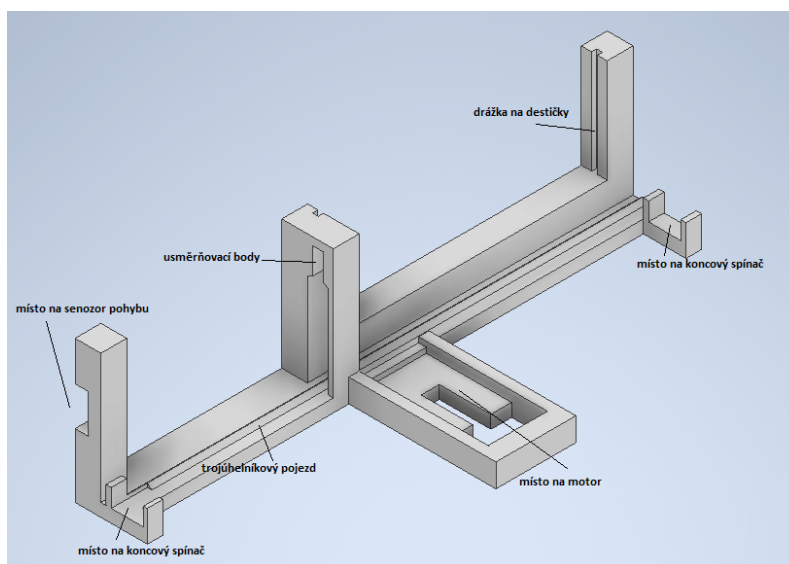
Při výběru modelovacího softwaru pro můj projekt jsem měl na výběr mezi několika možnostmi, mezi nimiž byly klíčovými kandidáty 3DS Max, Blender a Autodesk Inventor. Po důkladném zvážení všech faktorů jsem se rozhodl pro Autodesk Inventor.

1.1.2 Výhody a nevýhody Autodesk Inventoru

Rozhodl jsem se využít Autodesk Inventor, díky mé předchozí zkušenosti s tímto programem. Tato volba mi umožnila efektivní modelování bez nutnosti učení se nového softwaru. Díky studentské licenci poskytnuté školou jsem nemusel platit za software. Autodesk Inventor, specializovaný na inženýrství a konstrukci, odpovídá potřebám mého projektu s parametrickým modelováním a vhodnými nástroji. I když může být v uměleckém modelování omezený ve srovnání s některými uměleckými softwary, pro mé konkrétní potřeby a s ohledem na rychlý a efektivní postup modelování s minimální časovou investicí byla tato volba rychlá a optimální.

1.1.3 Modelování konstrukce

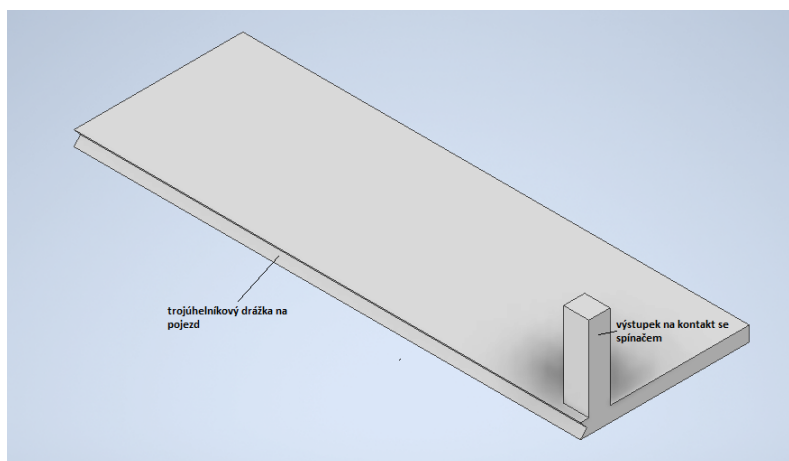
V případě běžné brány se brána obvykle pohybuje po kolečku a je usměrňována válečky. S těmito dvěma problémy jsem se musel vypořádat. První výzvu jsem řešil tím, že jsem namísto kolečka vytvořil trojúhelníkovou drážku, po které brána jezdí. Díky nízké váze brány není tření téměř žádné, což je ideální. Druhý problém jsem eliminoval tím, že jsem nahradil válečky pevnými body v konstrukci. Nakonec jsem vytvořil místa pro motor, koncové spínače, senzor pohybu a drážky pro destičky.



Obrázek 1.1: Model konstrukce

1.1.4 Modelování brány

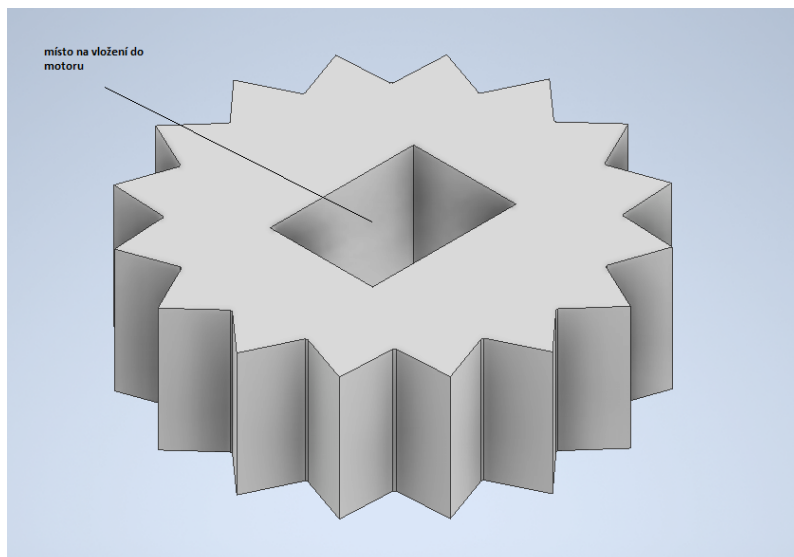
V procesu modelování brány jsem vytvořil samotnou bránu, vybavil ji trojúhelníkovou drážkou pro pojezd a výstupkem pro kontakt s koncovým spínačem.



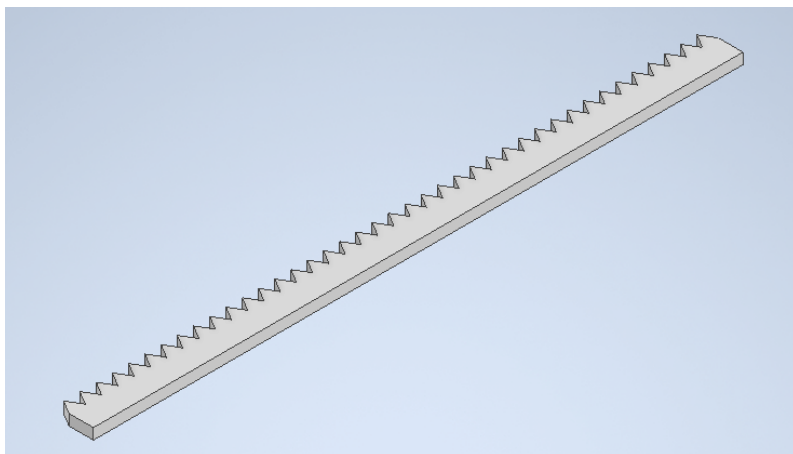
Obrázek 1.2: Model brány

1.1.5 Modelování ozubeného kolečka a ozubené lišty

Pro ozubené kolečko jsem vytvořil obdélníkovou díru pro umístění motoru a kolem obvodu kola jsem vytvořil zuby. Ozubená lišta má stejnou délku jako brána a je opatřena zuby shodnými s těmi na ozubeném kolečku.



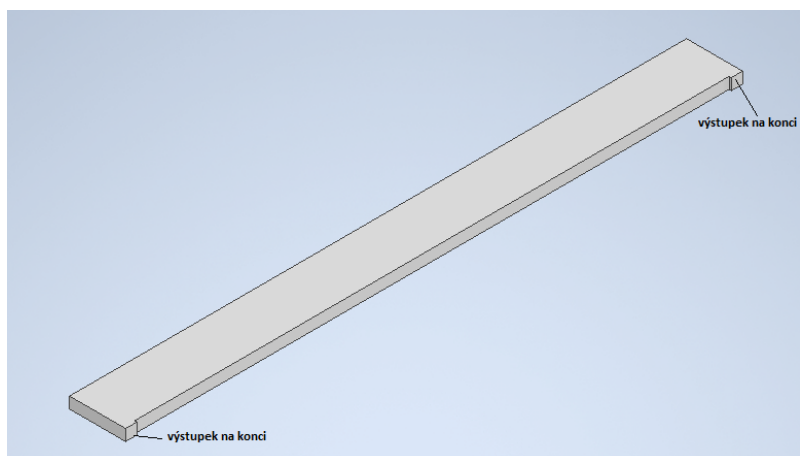
Obrázek 1.3: Model ozubeného kolečka



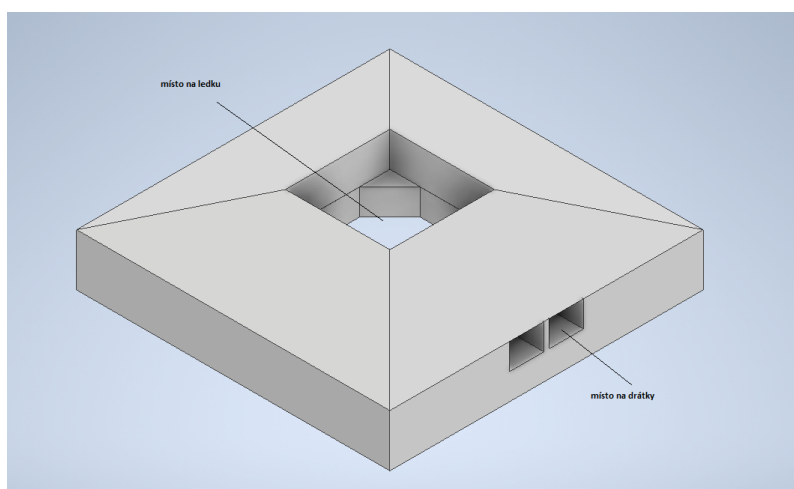
Obrázek 1.4: Model ozubené lišty

1.1.6 Modelování destiček a čepiček

Na destičky jsem přidal malé výčnělky na obou koncích, vytvářející malý rozestup mezi jednotlivými destičkami. V čepičkách je vyhrazeno místo pro LED diody a prostor pro drátky. Tyto drátky vedoucí k diodám jsou odděleny tak, aby nedošlo k dotyku mezi kladným a záporným pólem.



Obrázek 1.5: Model destičky



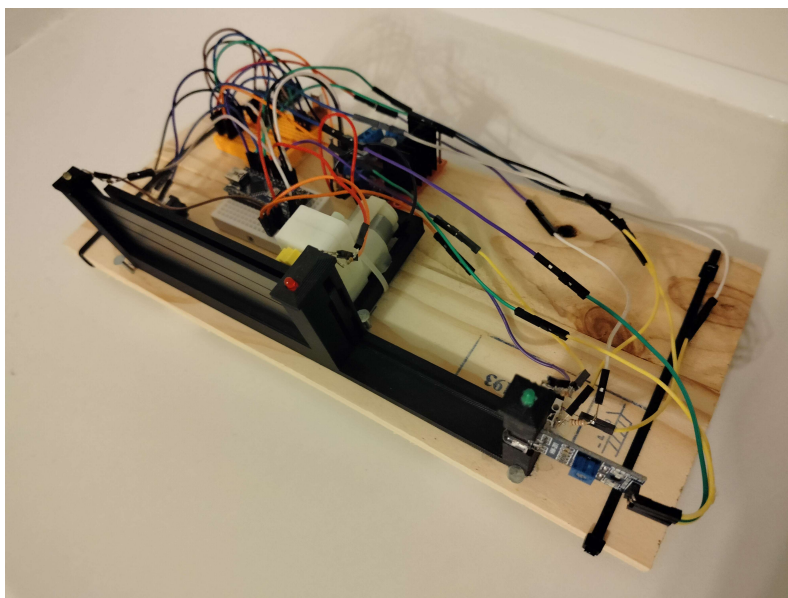
Obrázek 1.6: Model čepičky

1.2 3D TISK

3D tisk je inovativní výrobní proces, který umožňuje vytvářet třírozměrné objekty vrstvu po vrstvě na základě digitálního modelu. Tento postup eliminuje potřebu tradičního odstraňování materiálu a umožňuje vytvářet komplexní a přesné objekty s různorodými geometriemi. 3D tisk může využívat různé materiály, včetně plastů, kovů, keramiky a biologických tkání, což poskytuje široké možnosti aplikací od průmyslové výroby a prototypování po zdravotní péči a výzkum. Tento inovativní proces zrychluje vývojové cykly a umožňuje tvorbu unikátních a složitých designů s minimálním odpadem.

1.3 SESTAVOVÁNÍ MODELU

Připevnil jsem ozubenou lištu na bránu. Motor s ozubeným kolečkem, destičky, čepičky, koncové spínače a senzor pohybu jsem přilepil na vytisknutý model pomocí tavící pistole. Nakonec jsem celý model přilepil na dřevěné desky, kde jsem rovněž přidal dvě kontaktní pole, h-můstek a následně vše zapojil.



Obrázek 1.7: Ukázka sestavení modelu

2 HARDWARE

Výběr hardware proběhl rychle a jednoduše, neboť jsem se rozhodl použít Arduino Nano, které jsme již využívali ve škole. Následně mi chyběly pouze některé komponenty, které mi vstřícně poskytl pan učitel Mgr. Marcel Godovský.

2.1 SEZNAM POUŽITÉHO HARDWARU

1x Arduino Nano Mikrokontrolér

2x DM00 (DM 03S) Koncové spínače

1x L298N H-Můstek, umožňujě měnit polarity motoru

1x 5V motor

3x LED

3x 330R Rezistor

3x 1k Rezistor

1x HW-201 Senzor pohybu

1x VS1838B IR přijímač

1x Vytisknutá konstrukce

1x Vytisknutá brána

1x Vytisknuté ozubené kolečko

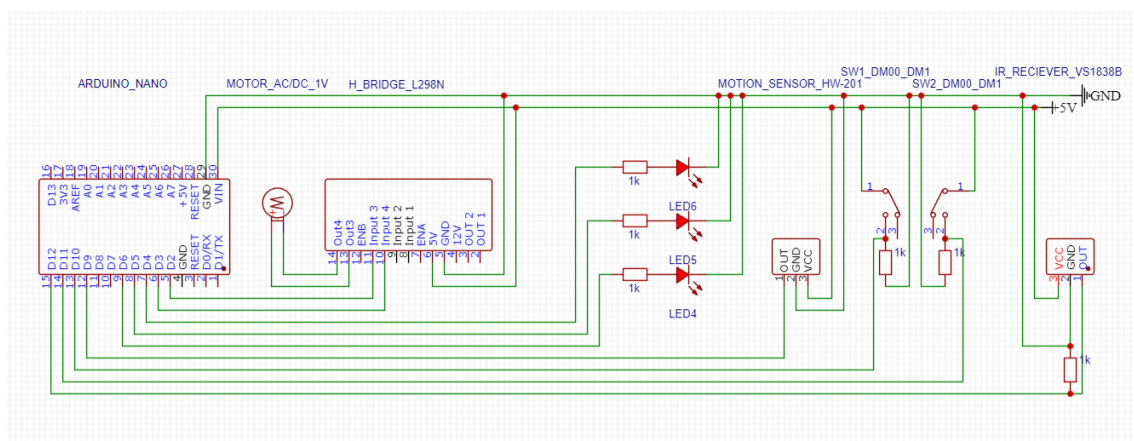
1x Vytisknutá ozubená tyč

4x Vytisknuté plotové destičky

3x Vytisknuté čepičky na plot

2.2 SCHÉMA HARDWARU

Schéma je koncipováno s cílem dosáhnout co největší přehlednosti, a proto se nemusí shodovat s reálným uspořádáním komponent v modelu.



Obrázek 2.1: Schéma zapojení

2.3 ARDUINO

Arduino je open-source projekt, který je založen na snadno použitelném hardwaru a softwaru. Arduino desky mají schopnost číst různé vstupy, jako například světlo na senzoru, stisknutí tlačítka nebo dokonce zprávy z Twitteru, a přeměňovat je na odpovídající výstupy. To může zahrnovat aktivaci motoru, rozsvícení LED nebo dokonce publikaci něčeho online. S Arduino deskou můžete komunikovat pomocí instrukcí, které posíláte do mikrokontroléru. K tomu využíváte programovací jazyk Arduino a speciální vývojové prostředí Arduino Software (IDE).

2.4 PROBLÉMY S HARDWAREM

Ovladač nečekaně přiděloval různé kódy na jedno tlačítko, což bylo trochu zvláštní. Toto se stávalo po určité době testování, a občas to fungovalo, zatímco jindy ne. Opravit tento problém se ukázalo být složitější, než jsem původně očekával, a tak jsem se rozhodl ho zatím neřešit. Celkově však ovladač pracuje správně přibližně 90 % času.

Problém s motorem a proudem vznikl, když jsem původně plánoval napájet motor přes Arduino. Nicméně jsem si neuvědomil, že Arduino poskytuje pouze omezený proud (asi 200 mA), zatímco motor, který používám, vyžaduje alespoň 500 mA. To mě vedlo k hledání externího napájení, a nakonec mi pan učitel Mgr. Marcel Godovský poskytl 5V zdroj, což je výhodné, protože z něj mohu napájet i Arduino.

3 FUNKCE MODELU BRÁNY

Model brány má několik základních funkcí, které by měl obsahovat každý bránový systém. To zahrnuje otevírání, zavírání a zastavování na povel z dálkového ovládání. Dále umí zastavit na základě reakce na senzor pohybu. Kromě toho model disponuje třemi LED diodami, které signalizují aktuální stav brány.

3.1 OTEVŘENÍ, ZAVŘENÍ A ZASTAVENÍ

Pro aktivaci jedné z těchto funkcí je nezbytné stisknout příslušné tlačítko na dálkovém ovládání. Následně IR přijímač detekuje signál a přiřadí mu specifický kód. Na základě tohoto kódu tlačítka a dalších podmínek, jako jsou koncové spínače nebo senzor pohybu, rozhodne programový kód, zda se motor má roztočit či nikoli. V případě rozhodnutí pro roztočení je do motoru správně přivedeno napětí, aby se pohyboval ve správném směru.

Aby byl IR přijímač funkční, je třeba zahrnout knihovnu `IRremote.h`, nastavit pin připojený k přijímači (například `const int RECV_PIN = 12`), tento pin definovat pro `IRrecv` (`IRrecv irrecv(RECV_PIN)`), vytvořit proměnnou pro ukládání výsledků a aktivovat ho pomocí `irrecv.enableIRIn()`. Podmínka `if` uloží kód do `results`, pokud obdrží signál.

```
#include <IRremote.h> //knihovna k IR prijimaci
const int RECV_PIN = 12; //definovani cisla pinu
IRrecv irrecv(RECV_PIN); //definovani pinu pro IR prijimac
decode_results results; //definovani promenne results
//povoleni IR prijimace, pise se do setupu
irrecv.enableIRIn();
//podminka v loopu
if (irrecv.decode(&results)) {
    irrecv.resume();
}
```

Když máme kód tlačítka uložený, můžeme jej využít. Uložil jsem si všechny potřebné kódy do proměnných a následně je porovnával.

```
const unsigned long OPEN_CODE = 3328992116;
const unsigned long CLOSE_CODE = 404571636;
const unsigned long STOP_CODE = 204228740;
```

Pro další postup je nutné konfigurovat koncové spínače; o senzoru pohybu bude podrobněji pojednáno v následující části. Nastavení pinů je identické s tím u IR přijímače, s tím rozdílem, že v setupu nastavíme tyto dva piny jako `INPUT`.

```
const int Button_Levy_Pin = 10;
const int Button_Pravy_Pin = 11;

//povolení pinu, pise se do setupu
pinMode(Button_Levy_Pin, INPUT);
pinMode(Button_Pravy_Pin, INPUT);
```

Vytvoříme proměnné pro koncové spínače a senzor, abychom zabránili duplikaci kódu.

```
bool buttonLevyPressed = (digitalRead(Button_Levy_Pin) == HIGH);
bool buttonPravyPressed = (digitalRead(Button_Pravy_Pin) == HIGH);
bool irSensorHigh = (digitalRead(IR_Senzor_Pohybu_Pin) == HIGH);
```

Vytvoříme podmínky pro spouštění a zastavování motoru, kde proměnná `n` určuje stav motoru. První podmínka se ptá, zda je senzor aktivní či ne. Druhá podmínka kontroluje, zda je aktivní `STOP_CODE`. Třetí podmínka zkoumá, zda není aktivní ani `OPEN_CODE`, ani `CLOSE_CODE`. S každým průchodem smyčkou se kód resetuje, a tím pádem tato podmínka začne platit, což umožní koncovým spínačům zastavit motor při jejich aktivaci. Čtvrtá podmínka se ptá, zda je vždy jeden spínač sepnutý a druhý nikoli. Toto zajistí zastavení motoru, když se jeden ze spínačů sepne. Pátá podmínka ověřuje, zda je aspoň jeden nebo žádný spínač sepnutý. Toto je implementováno z důvodu, že brána je obvykle na konci a má spínač sepnutý, nebo je uprostřed cesty a spínače nejsou sepnuté.

```
if (irSensorHigh) {
    if (results.value != STOP_CODE) {
        if (results.value != OPEN_CODE && results.value != CLOSE_CODE) {
            if ((buttonPravyPressed && !buttonLevyPressed) ||
                (!buttonPravyPressed && buttonLevyPressed)) {
                n = 2;
            }
        } else if (!buttonLevyPressed || !buttonPravyPressed) {
            n = 1;
        }
    } else {
        n = 2;
    }
} else {
    n = 2;
    results.value = STOP_CODE;
}
```

Točení motoru ve správném směru vyřešíme porovnáním, které tlačítko na ovladači bylo stisknuto, a přiřadíme odpovídající směr do proměnné `direction`. Při tomto procesu také resetujeme proměnnou `results`, abychom zajistili správnou funkčnost programového kódu, jak je zmíněno v posledním odstavci.

```
if (results.value == OPEN_CODE) {
    direction = LEFT;
    results.value = 0;
} else if (results.value == CLOSE_CODE) {
    direction = RIGHT;
    results.value = 0;
}
```

Ještě musíme definovat proměnnou `direction`.

```
enum Direction { LEFT = 1, RIGHT = 2 };
int direction;
```

3.2 SENZOR NA POHYB

U senzoru pohybu musíme definovat pin, ke kterému je připojen, a v setupu povolit tento pin jako `INPUT`.

```
const int IR_Senzor_Pohybu_Pin = 9;
//povoleni pinu, pise se do setupu
pinMode(IR_Senzor_Pohybu_Pin, INPUT);
```

Kontrolujeme stav senzoru a sledujeme, zda je ve stavu `HIGH` nebo `LOW`. Vzhledem k tomu, že stav `LOW` signalizuje aktivitu, když je senzor v tomto stavu, motor se nespustí a nastavíme kód tlačítka na `STOP_CODE`. Tímto způsobem zajistíme, že se motor okamžitě zastaví a nebude znovu spouštěn. Tato opatření jsou implementována tak, aby se předešlo situaci, kdy by brána reagovala na krátkodobou aktivaci senzoru, například při průchodu dítěte, a brána by se zastavila a ihned znovu spustila. Tímto způsobem je zajištěno, že brána zůstane zastavená po detekci aktivity.

```
if (irSensorHigh) {
    /* chybejici kod */
} else {
    n = 2;
    results.value = STOP_CODE;
}
```

3.3 LED INDIKÁTORY A MOTOR

Zapnutí motoru a LED na základě proměnných `n` a `direction` začíná tím, že znovu definujeme a nastavujeme piny. Tyto piny jsou poté povoleny v setupu, přičemž všechny jsou nastaveny jako `OUTPUT`.

```
const int LED_Uprostred_Pin = 6;
const int LED_Vlevo_Pin = 4;
const int LED_Vpravo_Pin = 5;
const int Motor_Pravy_Pin = 2;
const int Motor_Levy_Pin = 3;

//povolení pinu, piseme do setupu
pinMode(LED_Uprostred_Pin, OUTPUT);
pinMode(LED_Vlevo_Pin, OUTPUT);
pinMode(LED_Vpravo_Pin, OUTPUT);
pinMode(Motor_Pravy_Pin, OUTPUT);
pinMode(Motor_Levy_Pin, OUTPUT);
```

Pro aktivaci motoru potřebujeme nastavit jeden pin na `HIGH` a druhý na `LOW`. Tuto konfiguraci řídí proměnná `n`, která určuje, zda se motor má točit, a proměnná `direction`, která určuje směr otáčení. Pro aktivaci LED diod postupujeme obdobně. Nakonec přidáváme `delay(180)` pro správnou funkčnost koncových spínačů, které reagují příliš rychle. Toto zpoždění bylo zavedeno k úpravě problémů s přílišnou citlivostí spínačů, které vedly k nežádoucím krátkým pohybům brány.

```
digitalWrite(LED_Uprostred_Pin, n == 1);
digitalWrite(LED_Vlevo_Pin, n == 1 && direction == LEFT);
digitalWrite(LED_Vpravo_Pin, n == 1 && direction == RIGHT);
digitalWrite(Motor_Pravy_Pin, n == 1 && direction == LEFT);
digitalWrite(Motor_Levy_Pin, n == 1 && direction == RIGHT);

if (n == 1) {
    delay(180);
}
```


3.4 OPTIMALIZACE KÓDU

Pro optimalizaci kódu začneme definováním proměnných úplně nahoře (kromě knihoven). Následně pokračujeme funkcí `setup()`, kde povolujeme piny a definujeme akce. Dále máme vlastní funkci `handleLEDsAndMotors()`, která se stará o zapínání pinů pro LED diody a motor. Následuje funkce `checkIRSensorAndButtons()`, kde se nachází rozsáhlá sada podmínek. Dále je tu funkce `processIRResults()`, ve které určujeme směr. Na závěr máme funkci `loop()`, kde jsou volány všechny vlastní funkce, a kód ovladače je uložen do proměnné `results`.

```
void loop() {  
  if (irrecv.decode(&results)) {  
    irrecv.resume();  
  }  
  checkIRSensorAndButtons();  
  processIRResults();  
  handleLEDsAndMotors();  
}
```

3.5 OVLÁDÁNÍ BRÁNY

Pomocí ovladače odesíláme signál, který programový kód vyhodnotí a přiřadí kód tlačítka do proměnné `results` (`OPEN_CODE`, `STOP_CODE`, `CLOSE_CODE`). Pokud je kód tlačítka roven `OPEN_CODE` a splňují se všechny podmínky, motor se zapne a brána se otevře. V případě, že kód tlačítka odpovídá `CLOSE_CODE` a všechny podmínky jsou splněny, motor se zapne a brána se zavře. Při shodě s kódem `STOP_CODE` se nic nestane, motor zůstane vypnutý. Pokud je motor aktivní a senzor pohybu se aktivuje, motor se zastaví a už se nespustí.

ZÁVĚR

Cílem projektu bylo vytvořit model posuvné brány s elektrickým pohonem, kterou lze ovládat pomocí dálkového ovládání. Model byl vymodelován v programu Autodesk Inventor, následně vytisknut na 3D tiskárně a spolu s hardwarem sestaven. Pro realizaci byl využit mikrokontrolér Arduino Nano, který byl naprogramoval.

Podle mého názoru jsem cílů projektu dosáhl. Během procesu práce jsem získal dovednosti efektivního modelování v programu Autodesk Inventor a naučil se vhodně vytvářet modely pro snadné 3D tisknutí. Dále jsem zdokonalil své programátorské dovednosti a získal zkušenosti s Arduinem. Pro budoucí vylepšení plánuji přidat kameru, která bude snímat SPZ a na základě těchto údajů otevírat bránu. Také mám v úmyslu vytvořit mobilní aplikaci, která bude fungovat jako dálkový ovladač.

SEZNAM POUŽITÝCH ZDROJŮ

- [1] *Citace PRO* [online]. Citace.com, 2020 [cit. 2020-08-31]. Dostupné z: <https://www.citacepro.com>
- [2] *Arduino* [online]. Arduino.cc, 2024 [cit. 2024-1-8]. Dostupné z: <https://www.arduino.cc>
- [3] *EasyEDA* [online]. easyeda.com, 2024 [cit. 2024-1-8]. Dostupné z: <https://easyeda.com/>
- [4] *Autodesk Inventor* [online]. autodesk.cz/products/inventor/, 2024 [cit. 2024-1-8]. Dostupné z: <https://www.autodesk.cz/products/inventor/>
- [5] *Canva* [online]. canva.com, 2024 [cit. 2024-1-9]. Dostupné z: <https://www.canva.com>

Seznam obrázků

| | | |
|-----|-----------------------------------|----|
| 1.1 | Model konstrukce | 7 |
| 1.2 | Model brány | 7 |
| 1.3 | Model ozubeného kolečka | 8 |
| 1.4 | Model ozubené lišty | 8 |
| 1.5 | Model destičky | 9 |
| 1.6 | Model čepičky | 9 |
| 1.7 | Ukázka sestavení modelu | 10 |
| 2.1 | Schéma zapojení | 12 |