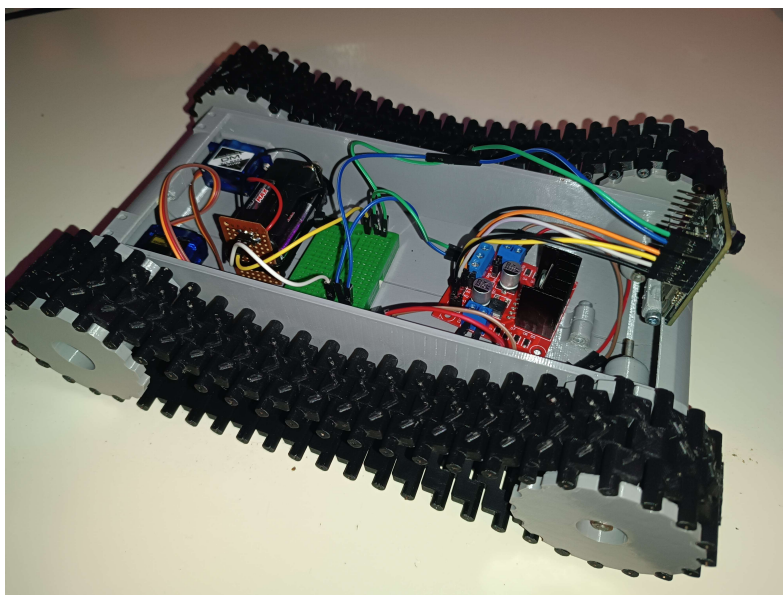


# ZÁVĚREČNÁ STUDIJNÍ PRÁCE

## dokumentace

### Tank na dálkové ovládání TM-1



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

## **Poděkování**

Chtěl bych poděkovat panu učiteli Mgr. Marcelu Godovskému za poskytnutí 3D tisku a užitečné rady při vytváření projektu.

## **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**

Tato práce se zaměřuje na konstrukci dálkově ovládaného tanku, jehož většina součástí byla vyrobena technologií 3D tisku. Základní řídicí jednotkou je vývojová deska ESP32 s vestavěnou kamerou a WiFi modulem, která umožňuje živé ovládání tanku prostřednictvím webového serveru postaveného na knihovně ESPAsyncWebServer. Webové rozhraní dává uživatelům možnost sledovat tank v reálném čase, což značně usnadňuje jeho ovladatelnost. Součástí projektu je také sestavení vlastního ovladače jako možnost alternativního ovládání.

## **Klíčová slova**

ESP32, Tank, závěrečná práce, dokumentace, . . .

## **Abstrakt v angličtině**

This work focuses on the construction of a remotely controlled tank, the majority of whose components were manufactured using 3D printing technology. The core control unit is an ESP32 development board with a built-in camera and WiFi module, enabling live control of the tank through a web server built on the ESPAsyncWebServer library. The web interface provides users with the ability to monitor the tank in real-time, significantly enhancing its controllability. As part of the project, the development of a custom controller is also included as an alternative control option.

# Obsah

<b>Úvod</b>	<b>5</b>
<b>1 Využité technologie</b>	<b>6</b>
1.1 Technologie 3D tisku . . . . .	6
1.2 Hardware . . . . .	6
1.2.1 Seznam součástek . . . . .	7
1.2.2 Vývojová deska ESP32-CAM . . . . .	7
1.2.3 H-můstek L298N . . . . .	8
1.2.4 Servo MG90S 180° . . . . .	9
1.2.5 Napájení . . . . .	9
1.2.6 Nepájivé pole . . . . .	10
1.3 Software . . . . .	11
1.3.1 Programovací jazyk Arduino . . . . .	11
1.3.2 Vývojové prostředí PlatformIO . . . . .	11
<b>2 Způsoby řešení a použité postupy</b>	<b>12</b>
2.1 Sestavení modelu . . . . .	12
2.2 Zapojení a testování součástek . . . . .	14
2.3 Zprovoznění serveru . . . . .	14
2.3.1 Připojení k WiFi . . . . .	16
2.4 Kód pro ovládání tanku . . . . .	17
2.5 Webová aplikace . . . . .	19
<b>3 Výsledky řešení</b>	<b>20</b>
3.1 Podoba hardwarového zařízení . . . . .	20
3.2 Podoba webové aplikace . . . . .	20

# ÚVOD

Mou hlavní motivací pro tento projekt byla má dlouhodobá záliba ve vojenské technice, jenž se se mnou táhne již od dětství a také zájem v oblasti hardware. Rozhodl jsem se tedy tyto své dvě záliby zkombinovat a vytvořit tak tento projekt.

Cílem této práce bylo tedy vytvořit tank na dálkové ovládání s možností ovládání přes webový server nebo přes vlastnoručně vyrobený ovladač. Při vytváření mého projektu jsem se nejprve musel rozhodnout, které součástky budou nejvhodnější, a zároveň jsem pečlivě promýšlel, zda si vybrat předem sestavený model nebo se pustit do výroby vlastních komponentů a jejich následného 3D tisku.

Původně jsem zamýšlel všechny díly sám vymodelovat, ale vzhledem k tomu, že toto řešení by bylo docela dost časově náročné, rozhodl jsem se najít si na internetu vhodný model podvozku i s pásy a zbytek domodelovat sám. Všechny díly jsem nakonec vytiskl na 3D tiskárně a smontoval dohromady.

Jako jádro projektu jsem si zvolil právě vývojovou desku ESP32, vybavenou integrovaným Wi-Fi modulem a kamerou, a jako první jsem chtěl vytvořit jednoduchý webový server pomocí knihovny ESPAsyncWebServer, který by umožňoval připojenému uživateli tank ovládat pomocí živého obrazu z kamery na serveru. Další problematikou by bylo sestrojení vlastního ovladače jako alternativní možnost pro ovládání tanku.

V této dokumentaci se tedy snažím podrobněji popsat svůj postup při řešení projektu a také problémy se kterými jsem se setkal a jejich řešení.

# **1 VYUŽITÉ TECHNOLOGIE**

## **1.1 TECHNOLOGIE 3D TISKU**

Technologie 3D tisku představuje revoluční způsob výroby, který umožňuje vrstvení materiálů a vytváření trojrozměrných objektů podle digitálních modelů. Jednou z hlavních výhod této technologie je schopnost tvořit vrstvením, což znamená postupné nanášení materiálu podle přesného digitálního modelu. To přináší nejen efektivní výrobu, ale také možnost přizpůsobit design součástek dle konkrétních požadavků.

Proces využívá 3D modelovací software pro vytváření designů a slicer software k přípravě dat pro tisk. Tiskárna postupně nanáší a vrství materiál, často termoplastický filament, a vytváří tak fyzický objekt. Tato technologie nachází široké uplatnění v průmyslu, od výroby prototypů přes modelářství až po vytváření funkčních součástek pro stroje a zařízení.

Já se taktéž rozhodl využít tuto technologii ve svém projektu. Nejprve jsem chtěl všechny součástky sám vymodelovat, ale nakonec jsem přistoupil na kompromis vlastní výroby a již předem vymodelovaného projektu, který jsem si našel na internetu.

## **1.2 HARDWARE**

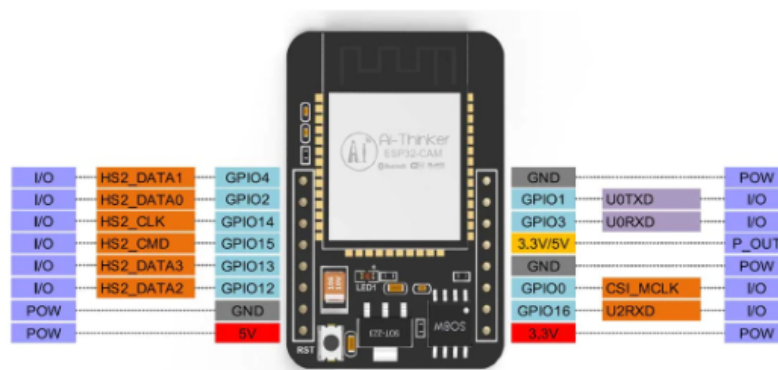
V této kapitole se budu podrobně věnovat představení klíčových součástek, které jsem ve svém projektu využil.

### 1.2.1 Seznam součástek

- Vývojová deska ESP32-CAM, WiFi + Bluetooth
- H-můstek L298N, dvoumotorový modul
- 2x Plastové micro servo SG90 9g, 180°
- Součástky vytvořené pomocí 3D tisku
- Napájecí modul
- Nepájivé pole

### 1.2.2 Vývojová deska ESP32-CAM

V mém projektu hraje klíčovou roli vývojová deska ESP32-CAM, která se stala nezbytným jádrem celého systému, jelikož na ní běží webový server, bez kterého by nebylo možné tank ovládat a také program pro pohyb tanku. Tato kompaktní vývojová deska od společnosti Espressif Systems kombinuje WiFi a Bluetooth komunikaci s kamerovým modulem OV2640 schopným snímání obrázků s rozlišením až 2 megapixely. Obsahuje celkem 10 digitálních GPIO pinů, a pro program je k dispozici paměť o celkové velikosti 128 KB. Deska taktéž pracuje s napětím 5 V, což zajišťuje bezproblémovou integraci s dalšími součástkami projektu.

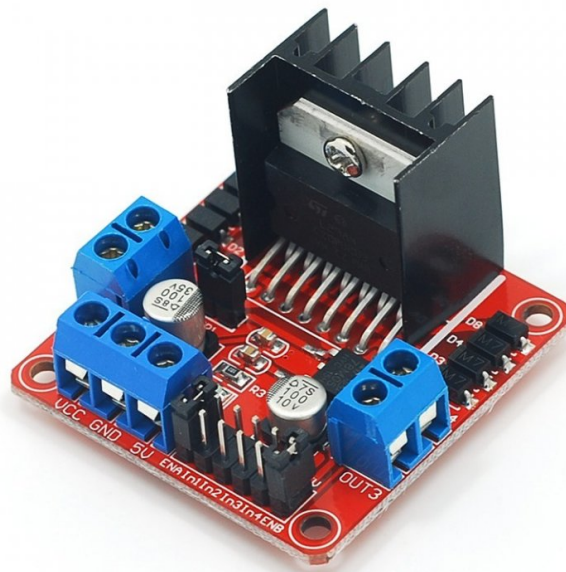


Obrázek 1.1: Vývojová deska ESP32-CAM [1].

### 1.2.3 H-můstek L298N

H-můstek L298N je integrovaný obvod, který slouží jako řadič pro motory. Je široce využíván v elektronických projektech, zejména v robotice a modelářství. Tento můstek umožňuje kontrolu směru otáčení a rychlosti stejnosměrných motorů. Modul využívá výkonný čip ST L298N a zaručuje taktéž ochranu proti zkratu a přehřátí. Jeho konstrukce obsahuje dva H-můstky, což znamená, že může řídit až dva motory nezávisle na sobě. Tato komponenta poskytuje efektivní řešení pro pohonný systém v různých elektronických aplikacích. Jeho vestavěný 5 V stabilizátor umožňuje kompatibilitu s dalšími komponenty mého projektu.

V mém projektu jsem tento H-můstek využil k ovládání motorů. Jeho úlohou je převádět vstupní signály, které přicházejí ze serveru, na konkrétní směr otáčení motorů.



Obrázek 1.2: VH-můstek L298N [2].



### 1.2.4 Servo MG90S 180°

Motor Servo MG90S 180° je kompaktní, ale výkonný servomotor široce používaný v různých aplikacích, zejména v oblasti modelování, robotiky a automatizace. Tento motor je schopen otáčet se o 180 stupňů, díky čemuž je neocenitelný ve scénářích vyžadujících přesnou manipulaci s úhly. Já pro svůj projekt ale potřeboval motor s možností otáčení o 360 stupňů, a proto jsem s pomocí pana učitele Godovského provedl úpravy na tomto motoru, aby byl schopen poskytnout požadované otáčení.



Obrázek 1.3: Plastové micro servo SG90 9g, 180° [3].

### 1.2.5 Napájení

Pro napájení jsem se rozhodl využít čtyři tužkové baterie umístěné v držáku, což mi poskytuje celkové napětí 7 V. Zdroj napájí nepájivé pole odkud pomocí kabelů přivádím proud do vývojové desky a také do H-můstku.

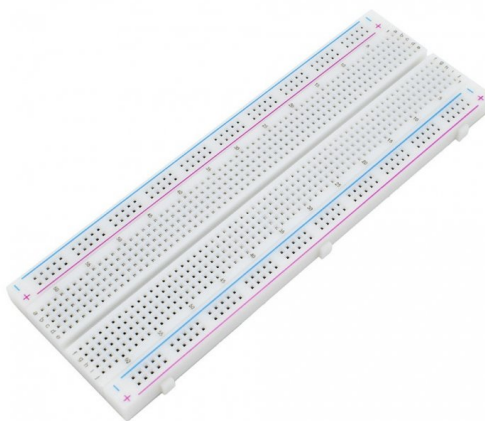


Obrázek 1.4: Napájení pro projekt.

### 1.2.6 Nepájivé pole

Nepájivé pole, často nazývané také experimentální nebo prototypovací deska, představuje klíčový nástroj v elektronickém vývoji, a já jsem ho aktivně využíval ve fázi testování mého projektu. Tato deska poskytuje strukturované pole propojovacích otvorů, které umožňují snadné a dočasné připojení elektronických součástek bez nutnosti pájení.

To se ukázalo jako neocenitelné zejména v raných fázích projektu, kdy byly potřeba časté úpravy a testování funkčnosti komponentů. Díky nepájivému poli jsem mohl flexibilně experimentovat s různými propojeními, připojovat a odpojovat komponenty a rychle reagovat na změny v návrhu. To nejen zrychlilo proces vývoje, ale také mi umožnilo systematicky ověřit funkčnost jednotlivých součástek.



Obrázek 1.5: Nepájivé pole [4].

## **1.3 SOFTWARE**

V této části bych se rád věnoval programovacímu jazyku Arduino, ve kterém jsem napsal kód pro svůj projekt. Zároveň představím vývojové prostředí PlatformIO, které jsem využil pro snadný vývoj, správu knihoven a ladění aplikace a také věnuji pár slov programu 3DS Max, ve kterém jsem modeloval některé součástky.

### **1.3.1 Programovací jazyk Arduino**

Arduino je open-source programovací jazyk přímo vytvořen k programování integrovaných obvodů. Postavený na jazyce C++ s rozšířeními pro mikrokontrolery, umožňuje rychlé prototypování a snadnou implementaci elektronických zařízení. Kódy, nazývané "sketchy", obsahují funkce `setup()`, sloužící k inicializaci a `loop()`, umožňující opakované provádění kódu. Arduino vyniká svou jednoduchou syntaxí a bohatou knihovnou funkcí. Díky aktivní komunitě je snadné sdílet kódy a projekty, což přispívá k jeho popularitě.

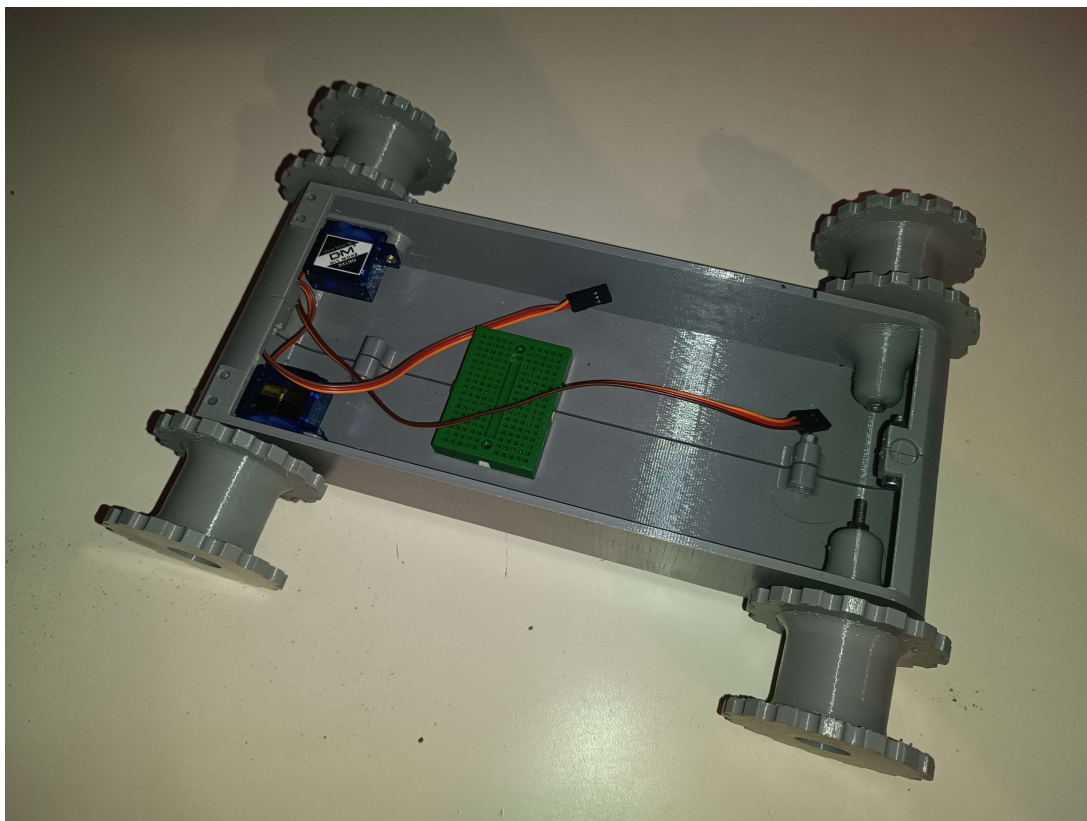
### **1.3.2 Vývojové prostředí PlatformIO**

PlatformIO je integrované vývojové prostředí (IDE) pro vývoj mikrokontrolerových aplikací, včetně těch založených na platformě Arduino. Jeho hlavní výhody zahrnují jednoduchou správu knihoven, snadný přístup k různým platformám a jednotné rozhraní pro vývoj s různými mikrokontrolery. Toto prostředí usnadňuje vytváření, ladění a správu projektů, což z něj činí efektivní nástroj pro rozsáhlý vývoj embedded systémů.

## 2 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

### 2.1 SESTAVENÍ MODELU

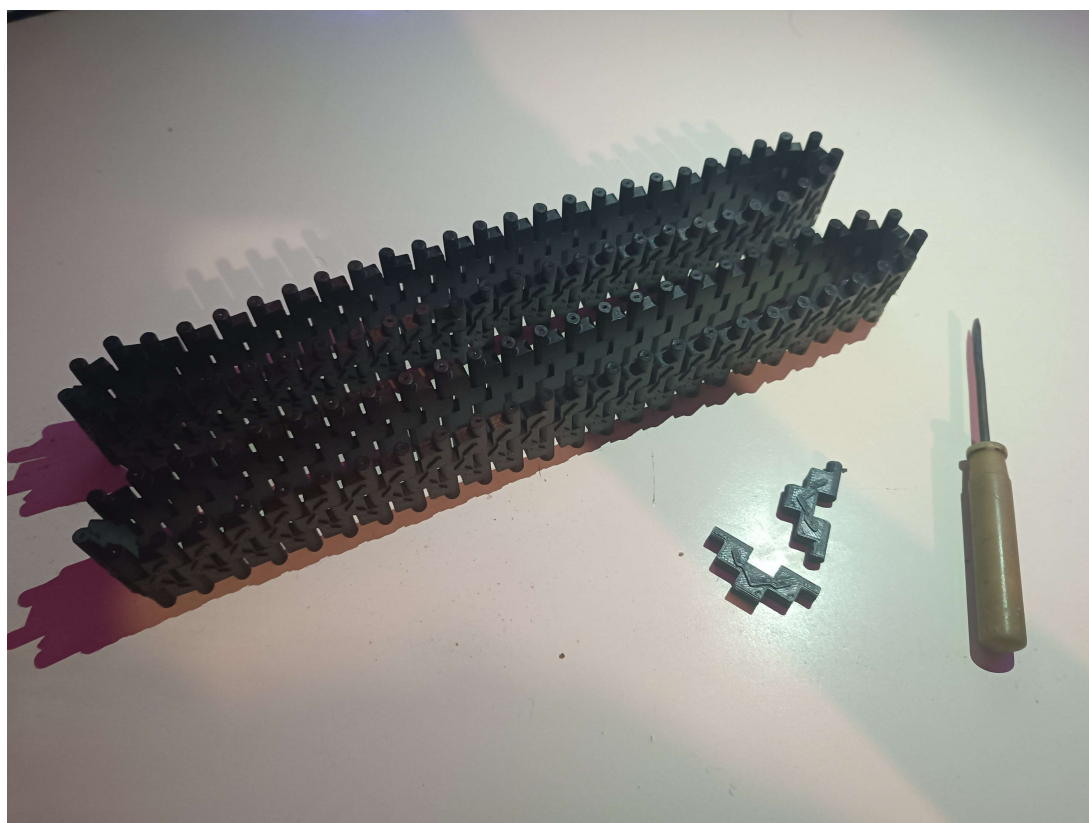
Proces sestavování mého tankového modelu byl poměrně namáhavý. Začal jsem tím, že jsem obdržel vymodelované součástky. Jako první jsem se pustil do skládání předních kol, kde jsem pečlivě propojil jednotlivé části a zajistil, aby byly pevně a správně umístěny. Dále jsem spojil dvě hlavní desky podvozku vozidla, čímž mi vznikl základ pro další konstrukci. Dalším klíčovým krokem bylo zapojení dvou servo motorů do zadní části korby. Tyto motory umožňují pohyb a ovládání tanku, a tak jsem k nim následně připojil hnací kola, čímž jsem vytvořil pohonný mechanismus vozidla.



Obrázek 2.1: Sestavený podvozek tanku

Jako další jsem se věnoval vytváření pásů tanku. Každý z nich se skládal z padesáti jednotlivých dílků, které jsem sestavil a spojil pomocí drobného drátku, který jsem pečlivě provlékl mezi jednotlivými články. Tento proces byl sice detailní, a zabral poměrně dost času ale vytvořil jsem díky tomu opravdu realistický model pásu, bez kterého by se tank neobešel.

Nakonec zbývalo už jsem nasadit pásy na svá místa, to se ale neobešlo bez problémů. Hnací kola jsou k servo motoru přilepena lepidlem a tudíž jsou docela křehká. Pásy byly udělané tak aby byly mezi koly napnuté a nestalo se, aby nějaký pás vypadl z drážek, to ale zapříčinilo, že se hnací kola pod nátlakem několikrát zkřivila, nebo rovnou odpadla. Rozhodl jsem se tedy pásy o článek rozšířit, což může způsobit, že pásy při jízdě mohou vypadnout.



Obrázek 2.2: Pásy tanku

## 2.2 ZAPOJENÍ A TESTOVÁNÍ SOUČÁSTEK

Po sestavení modelu jsem se zaměřil na zapojování součástek do nepájivého pole. Prvním krokem bylo důkladné otestování každé součástky, abych zjistil, zda pracuje tak, jak má. Tento průběžný testovací proces mi umožnil identifikovat případné problémy a zajistit, že každá část mého projektu funguje spolehlivě a bezchybně. Pro nahrávání kódu do vývojové desky ESP32 jsem použil MB programmer, což je štít s USB portem. Tento štít se zapojí přímo do pinů vývojové desky, což usnadňuje konfiguraci a programování. Následně jsem propojil servo motory s H-můstkem, abych ověřil jejich funkčnost. Během této fáze jsem však narazil na problém, když se H-můstek ukázal jako nedostatečně výkonný pro kontrolu obou motorů. V důsledku toho jsem musel nahradit H-můstek silnějším modelem, což vyřešilo problém a umožnilo plné využití servo motorů v mém projektu.

## 2.3 ZPROVOZNĚNÍ SERVERU

Pro zprovoznění webového serveru jsem se rozhodl použít knihovny AsyncTCP a ESPAsyncWebServer. Nejdříve jsem integroval knihovnu AsyncTCP, což je asynchronní TCP/IP knihovna, umožňující rychlou a efektivní komunikaci. Tato knihovna poskytuje neblokující komunikaci, což je zásadní pro zachování rychlosti a plynulosti webového serveru. Dále jsem implementoval knihovnu ESPAsyncWebServer, což je moderní a flexibilní HTTP serverová knihovna pro ESP32. Tato knihovna umožňuje snadné vytváření webových stránek, obsluhu HTTP požadavků a vytváření uživatelsky příjemného rozhraní. Při vývoji serverových funkcí jsem mohl využít asynchronní přístup, což výrazně zvyšuje výkon serveru.

Nakonec jsem konfiguroval webový server tak, aby poskytoval potřebné funkce pro mé zařízení. To zahrnovalo zpracování HTTP požadavků, ovládání výstupních zařízení, a vytváření dynamických webových stránek. Díky kombinaci těchto knihoven jsem mohl vytvořit responzivní a efektivní webové rozhraní, které je klíčové pro interaktivitu a vzdálené ovládání mého projektu. Celkově tyto knihovny poskytují robustní infrastrukturu pro vytváření webových serverů na platformě ESP32 s důrazem na rychlost, efektivitu a asynchronní komunikaci.

```
1 AsyncWebServer server(80);
2
3 void handleRoot(AsyncWebServerRequest *request)
4 {
5     request->send_P(200, "text/html", htmlHomePage);
6 }
7
8 void handleNotFound(AsyncWebServerRequest *request)
9 {
10     request->send(404, "text/plain", "File Not Found");
11 }
12
13 void setup(void)
14 {
15     Serial.begin(115200);
16
17     server.on("/", HTTP_GET, handleRoot);
18     server.onNotFound(handleNotFound);
19
20     server.begin();
21     Serial.println("HTTP server started");
22
23 }
```

### 2.3.1 Připojení k WiFi

Pro dosažení bezdrátové konektivity a propojení s Wi-Fi jsem v projektu využil knihovnu WiFi.h. Tato knihovna poskytuje nezbytné nástroje pro konfiguraci a správu bezdrátového připojení na platformě ESP32. Po integrování knihovny jsem mohl v kódu definovat přístupová jména a hesla k Wi-Fi sítím, ke kterým jsem chtěl, aby se zařízení připojilo. Dále jsem implementoval kód pro inicializaci a připojení k Wi-Fi síti. Tímto způsobem jsem umožnil zařízení komunikovat s okolním prostředím a využívat výhod bezdrátového připojení.

```
1  const char *ssid = "MyWiFiCar";
2  const char *password = "12345678";
3
4  void setup(void)
5  {
6      Serial.begin(115200);
7
8      WiFi.softAP(ssid, password);
9      IPAddress IP = WiFi.softAPIP();
10     Serial.print("AP IP address: ");
11     Serial.println(IP);
12
13     server.on("/", HTTP_GET, handleRoot);
14     server.onNotFound(handleNotFound);
15
16     server.begin();
17     Serial.println("HTTP server started");
18
19 }
```



## 2.4 KÓD PRO OVLÁDÁNÍ TANKU

Při navrhování kódu pro ovládání mého tanku jsem systematicky postupoval, abych zajistil přehlednost a efektivitu programu. Začal jsem tím, že jsem přiřadil piny pro každý motor vozidla. Pro každý motor jsem stanovil dva piny, jeden pro jízdu dopředu a druhý pro jízdu dozadu. Abych zvýšil čitelnost kódu a usnadnil případné úpravy, využil jsem globální konstanty pro definici směrů pohybu a stavů motorů. Tyto konstanty poskytují srozumitelný a jednotný zápis v celém kódu. Dále jsem vytvořil funkci `rotateMotor`, která přijímá číslo motoru a směr otáčení. Tato funkce nastavuje příslušné piny motoru tak, abych dosáhl požadovaného směru pohybu. Pokud je směr `FORWARD`, motor se nastaví pro jízdu dopředu, pokud je směr `BACKWARD`, motor se nastaví pro jízdu dozadu, a v ostatních případech jsou oba piny vypnuty, což zastavuje motor.

```
1 void rotateMotor(int motorNumber, int motorDirection)
2 {
3     if (motorDirection == FORWARD)
4     {
5         digitalWrite(motorPins[motorNumber].pinIN1, HIGH);
6         digitalWrite(motorPins[motorNumber].pinIN2, LOW);
7     }
8     else if (motorDirection == BACKWARD)
9     {
10        digitalWrite(motorPins[motorNumber].pinIN1, LOW);
11        digitalWrite(motorPins[motorNumber].pinIN2, HIGH);
12    }
13    else
14    {
15        digitalWrite(motorPins[motorNumber].pinIN1, LOW);
16        digitalWrite(motorPins[motorNumber].pinIN2, LOW);
17    }
18 }
19 }
```

Jako další bylo třeba vytvořit funkci, pomocí které bych definoval samotný pohyb tanku. Vytvořil jsem tedy funkci `moveTank`, jenž přebírá vstupní hodnotu, která odpovídá pohybu tanku. S využitím předem definovaných konstant pro směry pohybu, jako jsou `UP`, `DOWN`, `LEFT` a `RIGHT`, tato funkce zajišťuje různé typy pohybů vozidla. V případě, že vstupní hodnota indikuje pohyb nahoru (`UP`), funkce aktivuje otáčení obou motorů směrem dopředu. Podobně pro pohyb dolů (`DOWN`) se tank otáčí dozadu. Pro pohyb vpravo (`RIGHT`) a vlevo (`LEFT`) dochází ke kombinaci otáčení motorů v odpovídajícím směru. Během návrhu této části kódu jsem dbal na jednoduchost a srozumitelnost, abych usnadnil budoucí úpravy nebo přidání nových funkcionalit.

```
1 void moveTank(int inputValue)
2 {
3     switch (inputValue)
4     {
5         case UP:
6             rotateMotor(RIGHT_MOTOR, FORWARD);
7             rotateMotor(LEFT_MOTOR, FORWARD);
8             break;
9
10        case DOWN:
11            rotateMotor(RIGHT_MOTOR, BACKWARD);
12            rotateMotor(LEFT_MOTOR, BACKWARD);
13            break;
14
15        case LEFT:
16            rotateMotor(RIGHT_MOTOR, FORWARD);
17            rotateMotor(LEFT_MOTOR, BACKWARD);
18            break;
19
20        case RIGHT:
21            rotateMotor(RIGHT_MOTOR, BACKWARD);
22            rotateMotor(LEFT_MOTOR, FORWARD);
23            break;
```

```

24
25     default:
26         rotateMotor(RIGHT_MOTOR, STOP);
27         rotateMotor(LEFT_MOTOR, STOP);
28         break;
29     }
30 }

```

## 2.5 WEBOVÁ APLIKACE

Během vytváření webové aplikace bylo mým cílem vytvořit jednoduché webové rozhraní, které by umožnilo nejen ovládání pohybu tanku pomocí tlačítek na webové stránce, ale také sledování živého videa z kamerového modulu. Přestože jsem původně plánoval ukládat HTML kód do samostatného souboru, narazil jsem na omezení mé desky, která neumožňovala pracovat s více soubory. Nakonec jsem se rozhodl uložit veškerý kód pro webový server společně s kódem pro ovládání vozidla do jednoho souboru. Toto omezení mě však motivovalo k pečlivé organizaci a strukturování kódu, aby byl stále přehledný a snadno udržovatelný.

Samotný webový server obsahuje minimalistické uživatelské rozhraní upravené s pomocí CSS stylů, video stream z kamery, čtyři tlačítka pro ovládání pohybu tanku a také posuvný slider, který umožňuje plynule nastavit intenzitu světla LED, jež je navržena k osvětlení prostředí v podmínkách nižší osvětlenosti.. Při stisku tlačítka je do funkce pro pohyb vozidla odeslána odpovídající konstanta, která určuje směr pohybu. To umožňuje intuitivní a plynulé ovládání tanku prostřednictvím webového rozhraní.

## **3 VÝSLEDKY ŘEŠENÍ**

### **3.1 PODOBA HARDWAROVÉHO ZAŘÍZENÍ**

Úplně nakonec jsem se zaměřil na vytvoření designu, který by skryl veškerou elektroniku uvnitř modelu tanku. Pro uzavření jsem vytvořil model horní části korby, poskytující funkční obal pro komponenty. Zdroj energie je umístěn v zadní části korby a prostřednictvím kabelů poskytuje napájení nepájivému poli. Zde je nainstalován H-můstek pro řízení motorů, a přes další soubor kabelů je propojena i deska ESP32 s kamerou. Ta je pevně připevněna ke přední části korby, umožňující snímání obrazu z kamery během pohybu tanku.

I když jsem spokojen s celkovým vzhledem a umístěním elektroniky, jsou věci které bych v budoucnu chtěl určitě změnit. Z časových důvodů jsem se bohužel nedostal k vymodelování a vytisknutí věže tanku, což bych chtěl ještě ovšem dořešit. Chtěl bych také efektivněji a přehledněji řešit zapojení všech elektrotechnických prvků, jelikož aktuální propojení všech komponentů je poměrně nepřehledné a neestetické.

### **3.2 PODOBA WEBOVÉ APLIKACE**

Webová aplikace spolehlivě splňuje svůj účel, uživatelé se připojí a mohou tank bez problému ovládat pomocí 4 směrových tlačítek a sledovat při tom živý obraz z kamery tanku. Aplikace je také plně responzivní, lze ji proto používat i na mobilních telefonech nebo tabletech. Do budoucna bych si chtěl s celkovým vzhledem aplikace ještě pohrát a doladit některé nedostatky.

## ZÁVĚR

Cílem projektu bylo vytvořit funkční tank na dálkové ovládání, který by byl ovládán buď pomocí webové aplikace nebo vlastního ovladače. Většina cílů byla splněna a tank je nyní možné ovládat pomocí webové aplikace. Tvorbu vlastního ovladače, který by sloužil tanku jako alternativní možnost ovládání, jsem bohužel z časových důvodů již nestihl zrealizovat. Tuto myšlenku avšak nemám v plánu zahodit a do budoucna bych na ní chtěl určitě ještě zapracovat. Při vývoji mě nicméně napadl nespočet dalších možných změn, které by projekt vylepšily, jako například vymodelování věže a její následné zprovoznění, nebo také menší úpravy vzhledu webové aplikace. Na tato vylepšení bych se chtěl v budoucnu určitě také zaměřit. Celkově lze tedy říci, že tento projekt byl pro mě nejen technickou výzvou, ale i příležitostí k neustálému zdokonalování a rozšiřování mých znalostí.

## LITERATURA

- [1] *Vývojová deska ESP32* [online]. [cit. 2023-12-29]. Dostupné z: <https://bit.ly/vyvojova-deska-esp32-drateg>
- [2] *H-můstek L298N* [online]. [cit. 2023-12-29]. Dostupné z: <https://www.laskakit.cz/h-mustek-pro-krokovy-motor-l298n--dualni-motorovy-modul/>
- [3] *Plastové micro servo SG90 9g, 180°* [online]. [cit. 2023-12-29]. Dostupné z: <https://www.hadex.cz/1759-servo-plastove-sg90-180-9g-prevody-z-nylonu/>
- [4] *Nepájivé pole* [online]. [cit. 2023-12-29]. Dostupné z: <https://www.laskakit.cz/nepajive-kontaktni-pole-830-pinu--bile/>

## Seznam obrázků

1.1	Vývojová deska ESP32-CAM [1]. . . . .	7
1.2	VH-můstek L298N [2]. . . . .	8
1.3	Plastové micro servo SG90 9g, 180° [3]. . . . .	9
1.4	Napájení pro projekt. . . . .	9
1.5	Nepájivé pole [4]. . . . .	10
2.1	Sestavený podvozek tanku . . . . .	12
2.2	Pásky tanku . . . . .	13