Středoškolská technika 2024

Setkání a prezentace prací středoškolských studentů na ČVUT

HydraStation

Michal Spišak

Střední škola informatiky a finančních služeb, Plzeň,

Klatovská 200 G, 301 00 Plzeň

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Plzni dne 20. listopadu 2023 ……

Michal Spišak

Poděkování

Tímto chci poděkovat mému konzultantovi Janu Boháčovi, který mi pomohl s tvorbou podkladů pro mojí SOČ a zároveň mi byl k dispozici i mimo svojí pracovní dobu.

Anotace

Ve své práci jsem se zabýval vytvořením aplikace pro sběr, zpracování a vizualizaci telemetrických dat ze závodního autíčka na dálkové ovládání s vodíkovým pohonem. Realizace projektu zahrnuje nejen desktopovou aplikaci, ale i vysílač a přijímač. Cílem je vytvořit program, který umožní vizualizaci přijatých dat a usnadní jejich analýzu. Zároveň je kladen důraz na abstrakci pro možnost využití desktopové aplikace v jakémkoliv prostředí.

Klíčová slova

grafické rozhraní; ImGui; LoRa; vizualizace dat; telemetrie

Annotation

In my project I worked on application for collecting, processing and visualizing telemetry data from hydrogen powered remote controlled racing car. Project realization includes not only the desktop application, but also transmitter and receiver. The goal is to create a program, which allows visualization of received data and make the analysis easier. At the same time, emphasis is placed on abstraction for the possibility of using the desktop application in any environment.

Keywords

graphical interface; ImGui; LoRa; data visualization; telemetry

Obsah

Úvod

S příchodem klimatické krize se čím dál více mluví o ekologicky šetrnějších alternativách, jako jsou například elektroauta či vodík jakožto zdroj čisté energie. Zároveň vznikají vzdělávací programy, které rozšiřují osvětu ohledně ekologie. Jedním z nich je i Horizon se svým závodem H2GP, ve kterém týmy závodí se svými vodíkem poháněnými autíčky na dálkové ovládání. Je tedy ideální mít co nejúspornější vozidlo. Běžné řídící jednotky pracují pouze s pevně nastavenými hodnotami, čímž se snižuje účinnost a vodíkem je plýtváno.

Z tohoto důvodu jsme přišli s vlastním řešením, systémem HydraStation. Jedná se o desktopovou aplikaci pro sběr, zpracovávání a analýzu telemetrických dat. Hloupá závodní deska je nahrazena chytrou deskou s rádiem, která v pravidelném intervalu odesílá telemetrická data. Díky tomu jsme schopni vidět stav autíčka, aniž bychom ho museli mít fyzicky k dispozici. Zároveň jsme z těchto dat schopni odvodit anomálie. Tím se autíčko stává úspornějším a pro provoz bezpečnějším, neboť je schopné ujet více závodních kol a v případě anomálie lze provést opravy ještě před vznikem havárie.

Práce je rozdělena do dvou částí, teoretické a praktické. V teoretické části se zaměříme na využité technologie a samotnou problematiku přijímání, zpracovávání a vizualizaci velkého množství telemetrických dat. V praktické části se naopak zaměříme na realizaci, vývoj firmwaru a desktopové aplikace. Cílem je vytvořit program, který umožní efektivní analýzu telemetrických dat v reálném čase. Uživatelsky přívětivé prostředí je prioritou, neboť čtení dat probíhá v reálném čase. Při vývoji je kladen důraz abstrakci desktopové aplikace.

1 TEORETICKÁ ČÁST

1.1 Pojmy

1.1.1 GUI

GUI (Graphical User Interface) je rozhraní, které umožňuje uživateli interaktivně pracovat s programem za pomocí grafických prvků.

1.1.2 UX

UX (User Experience) je obor, který se zabývá usnadněním používání rozhraní webů, aplikací, ale i strojů a dalšího.[1]

1.1.3 Telemetrie

Telemetrie je technologie, která umožňuje dálkové přenášení měřených dat.[2]

1.1.4 Mikrokontroler

Monolitický integrovaný obvod obsahující kompletní mikropočítač, který se vyznačuje velkou spolehlivostí a kompaktností.[3]

1.1.5 Firmware

Firmware neboli mikroprogramové vybavení je označení pro software, který nejčastěji slouží k ovládání vestavěného systému.[4]

1.1.6 Sériová komunikace

Druh přenosu dat, při kterém jsou data odesílány postupně po jednotlivých bitech.

1.1.7 Wi-Fi

Technologie umožňující připojení k síti prostřednictvím rádiových vln.

1.1.8 OTA (over-the-air) update

Způsob aktualizace systému za využití bezdrátových technologií.

1.1.9 Proxy

Prostředník mezi dvěma stranami, které spolu komunikují

1.2 Technologie

1.2.1 Java

Java je staticky typovaný programovací jazyk, který pro svůj běh využívá tzv. Virtuální stroj, což znamená že program se první převede do tzv. **bytového kódu**, který je následně spuštěn pomocí **JVM (Java Virtual Machine)**. To usnadňuje běh programu v jakémkoliv prostředí, kde se nachází JVM.

1.2.2 C++

C++ je nízko úrovňový multiparadigmatický programovací jazyk, který je před spuštěním nutné zkompilovat. Tento jazyk je vhodný pro psaní **firmwaru vestavěných systémů**.

1.2.3 ImGui

ImGui je knihovna napsaná v jazyce **C++**, se kterou lze vytvářet jednoduché a uživatelsky přívětivé grafické rozhraní. Existují knihovny i pro mnohé další programovací jazyky, které umožňují využití této knihovny.

1.2.4 ESP32

ESP32 je malý dvoujádrový programovatelný mikrokontroler, nejčastěji využívaný ve světě IoT. Jeho výhodou je **zabudovaná Wi-Fi** a **nízká spotřeba energie**.

1.2.5 LoRa

LoRa (low range) je proprietární radiokomunikační technologie, která umožňuje bezdrátový přenos dat, jejíž výhodou je **nízká spotřeba energie**.[5]

1.2.6 PlatformIO

Platforma usnadňující vývoj firmware pro mikrokontrolery, jako jsou **Arduino** či **ESP32**.

1.2.7 Castle Link Live

Proprietární protokol vytvořený společností **Castle Technologies** pro odesílání telemetrických dat z ESC regulátoru. Data jsou odesílány regulátorem v podobě rozdílů v napětí.

2 PRAKTICKÁ ČÁST

Celý systém **HydraStation** se skládá z řídící jednotky **HydraBoard**, komunikační brány **HydraGateway** a desktopové aplikace **HydraStation**. Data jsou nejprve sesbírány ze senzorů. Následně dojde k prvotnímu zpracování uvnitř samotné řídící jednotky. Data jsou sbírána do doby, než dojde k jejich odeslání. Těsně před odesláním se vypočítá průměrná hodnota ze všech uložených hodnot, čímž se **zbavíme odchylek či anomálií**, které při měření vznikly. Následně dojde k odeslání finálních dat skrze **LoRa transceiver**, načež jsou přijaty **LoRa transceiverem** v komunikační bráně. Zde dojde k převedení dat do **ASCII hexadecimálního textu**. Tyto data poté skrze USB sériovou komunikaci putují do desktopové aplikace, kde dochází k jejich zpracování a vizualizaci.

2.1 Části systému HydraStation

2.1.1 HydraBoard

HydraBoard je řídící jednotkou samotného autíčka. Stará se o sběr telemetrických dat ze senzorů a jejich následné odesílání. Srdcem jednotky je mikrokontroler **ESP32**. Firmware pro řídící jednotku je napsán v jazyce **C++**, pro správu projektu je využíván nástroj **PlatformIO**. Sběr dat probíhá neustále, při odesílání dochází k vytvoření průměrné hodnoty dat od posledního odeslání, čímž se data stávají stabilnější a odolnější vůči odchylkám v měření. Řídící jednotka využívá obou svých jader, jedno se stará o čtení hodnot z ESC regulátoru pomocí protokolu **Castle Link Live** a druhé o odesílání nasbíraných telemetrických dat skrze mikrokontroler **LoRa**. Řídící jednotka umožňuje **OTA** aktualizace za využití technologie **Wi-Fi**.

2.1.2 HydraGateway

HydraGateway je bránou v komunikaci mezi řídící jednotkou a aplikací a plní zde roli **proxy**. Srdcem brány je mikrokontroler **ESP32**. I v tomto případě je firmware napsán v jazyce **C++**, pro správu projektu je využíván nástroj **PlatformIO**. Pro komunikaci s řídící jednotkou je využíván mikrokontroler **LoRa**, pro komunikaci s aplikací naopak **sériová komunikace** skrze **USB**. Při komunikaci mezi bránou a aplikací data putují v čitelném ASCII, přesněji v hexadecimálním formátu, konec dat je definován znakem *\n* neboli koncem řádků. To umožňuje jednodušší zpracování dat či jejich čtení i mimo desktopovou aplikaci.

2.1.3 HydraStation

HydraStation je desktopová aplikace napsaná v **Javě**, která zpracovává a vizualizuje telemetrická data. Grafické rozhraní je napsáno pomocí knihovny **ImGui**. Data jsou přijímány sériovou komunikací a ukládány v paměti **RAM**. Data jsou následně vizualizovány v grafech. Pro ukládání nasbíraných dat aplikace využívá vlastní souborový formát **Hydra Station File**, který je vhodný pro uchovávání časosběrných dat.

2.2 Realizace

2.2.1 Čtení telemetrických dat ze senzorů

Prvním krokem je nejprve dostat samotná data ze senzorů. Jelikož je k dispozici řídící jednotka s mikrokontrolerem **ESP32** *(viz obr. 1)*, nejedná se o nijak závažný problém. Na autíčku se několik zdrojů cenných telemetrických dat – napětí na baterii, teplota palivového článku, stav tlaku vodíku, proud vycházející z palivového článku či velké množství telemetrických dat, které lze získat z **ESC regulátoru značky Castle**. Některá data lze číst jednodušeji, čtení jiných dat může být naopak náročnější.

Přečíst napětí na baterii je s naší řídící jednotkou velmi jednoduché. Na jednom z pinů se již předem nachází jednoduchý mechanismus pro čtení napětí, který je skrze analogový pin připojen ze zbytku systému elektrické energie v autíčku. Mikrokontroler ESP32 navíc umí skrze **ADC** *(analog to digital converter)* převést analogová data na digitální. To samé platí u senzoru teploty či proudu palivového článku, kde měření probíhá stejným způsobem.

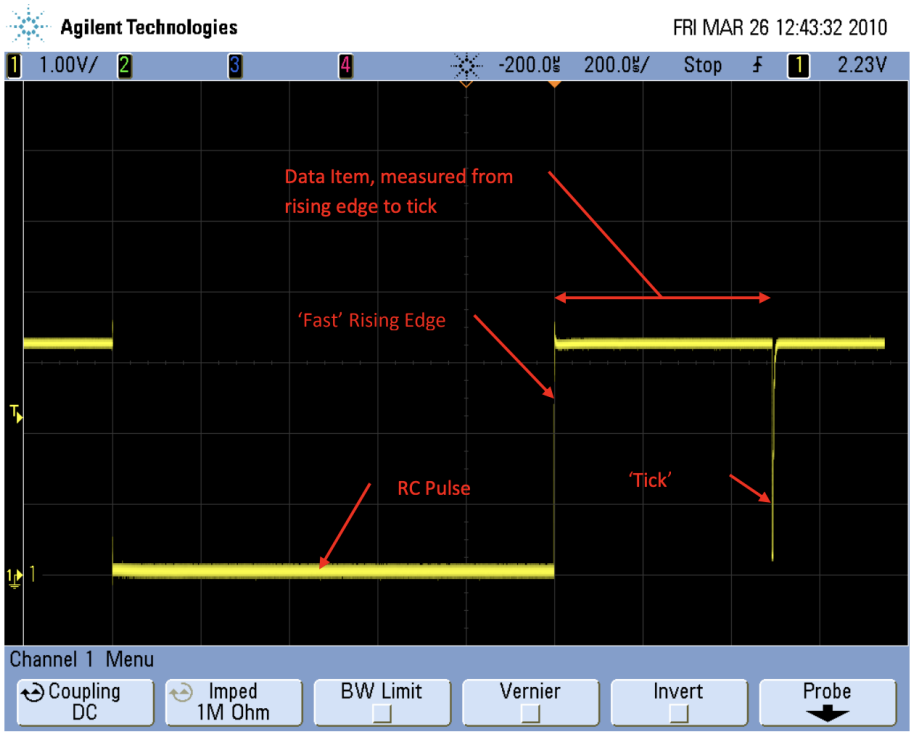
Čtení stavu tlaku vodíků probíhá skrze jednoduchý tlakový senzor. Jelikož regulátor vodíku reguluje vstupující vodík vždy na stejný tlak, nelze ho změřit v pascalech. To nám ale nebrání si skrze senzor změřit, zda vůbec nějaký tlak existuje. Senzor tlaku na autíčku se tedy sepne vždy, kdy je dostatečný tlak, načež dojde k vytvoření jednoduchého digitálního signálu.

Obr. 1: Příklad 1ms RC pulzu, následovaný 0.5ms datovou položkou[6]

2.2.2 Čtení telemetrických dat z ESC regulátoru

Nejsložitější částí je čtení telemetrických dat z **ESC regulátoru**. Ten průběžně sbírá data ze svých senzorů, následně je ale potřeba přečtená data z regulátoru dostat do řídící jednotky. Regulátory značky **Castle** zvládnou odesílat telemetrická data skrze protokol **Castle Link Live**, který je založený na systému klesajícího či stoupajícího napětí a době mezi těmito rozdíly. Naštěstí je tento protokol pečlivě zdokumentovaný na jejich webových stránkách. K odesílání těchto dat dochází mezi regulátorem a přijímačem pro ovladače RC autíček. Řídící jednotka je v tomto případě mezi samotným regulátorem a přijímačem, tudíž je možné komunikaci číst uvnitř firmware.

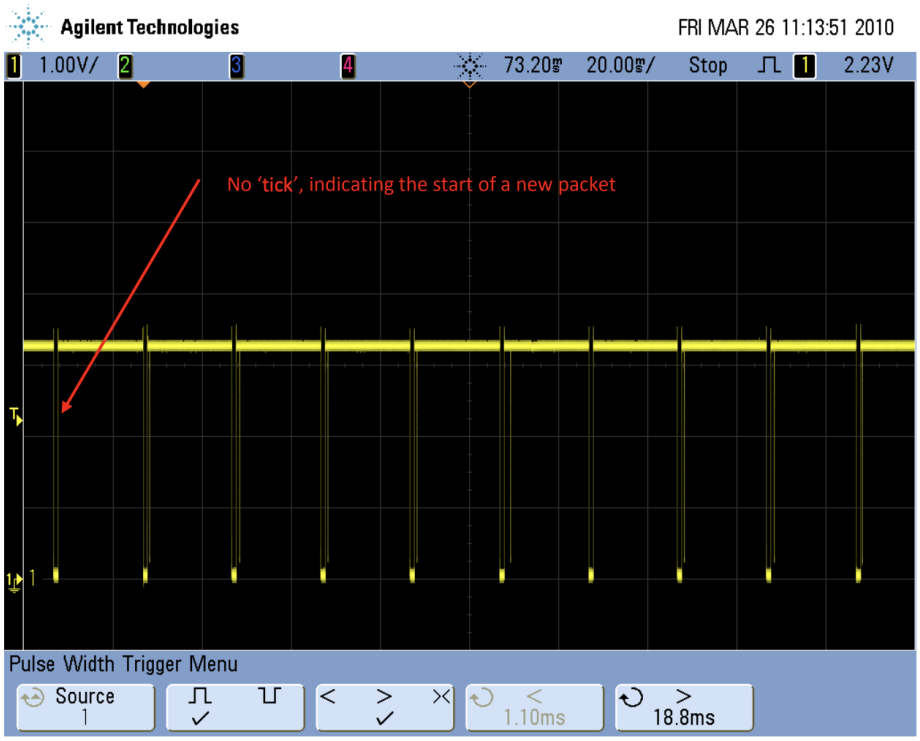
Jednotlivá data jsou odesílané sekvenčně za sebou. Jako první dojde k prudkému poklesu napětí, tzv. **falling edge**. *(viz obr. 2)*. Tím započne odesílání aktuálního stavu stisknutí plynu na ovladači. Tento signál je vytvářen přijímačem od ovladače RC autíček, tzv. *RC pulse*. Následně dojde k prudkému vzestupu, tzv. **rising edge** *(viz obr. 2)*. Následně se spočítá počet doba v nanosekundách, po které bylo napětí nízké. Jelikož je RC puls vždy delší než datové položky, je možné se dle něj synchronizovat se samotným regulátorem a aktuálním stavem protokolu. Poté, co dojde k prudkému nárustu v napětí, dojde k změření hodnoty datové položky, opět v nanosekundách. Ve finále dojde k tzv. **ticku** *(viz obr. 2)*, prudkému poklesu a okamžitému vzestupu napětí, což definuje ukončení datové položky. Následně podle tabulky uvedené v dokumentaci protokolu lze vypočítat samotnou hodnotu datové položky ve správných jednotkách.



Obr. 2: Příklad 1ms RC pulzu, následovaný 0.5ms datovou položkou

Jelikož je dle dokumentace počet datových položek **11**, znamená to, že je potřeba změřit přesně tento počet. Konec dat se vyznačuje stavem, kdy datová položka není ukončena již zmíněným tickem *(viz obr. 3)*. Pokud tedy například k měření došlo při odeslání již osmé položky a při zaznamenání stavu bez ticku nedošlo k napočítání počtu jedenácti datových položek, je vhodné nekompletní data zahodit. Pokud následně dojde k přečtení potřebného počtu dat, jsou přijatá data brána jako finální a lze je uložit pro pozdější zpracování.

Poslední překážkou je plynulé čtení dat. Jelikož se protokol pohybuje v řádech nanosekund, i sebemenší zpomalení může výrazně zkreslit měřená data. Naštěstí je možnost **využití druhého jádra mikrokontroleru ESP32**, které se může starat čistě o pouhé čtení dat z ESC regulátoru. Výhodou je též podpora detekce prudkých vzestupů a poklesů napětí za pomocí **interruptů**.



Obr. 3: Příklad prvních deseti datových položek. Zajímavostí je stav bez ticku[6]

2.2.3 Odesílání telemetrických dat

Po tom, co dojde k přečtení všech potřebných dat, musí tyto data opustit řídící jednotku. Jelikož není možné při závodu být k řídící jednotce připojen kabelem, jsou telemetrická data odesílána rádiovým přenosem. Ten je zajišťován **LoRa transceiverem**. **LoRa** je v tomto případě vybrána z důvodu své jednoduchosti při využití v IoT zařízení a nízké spotřebě elektrické energie. Zároveň lze data posílat na velmi dlouhé vzdálenosti.

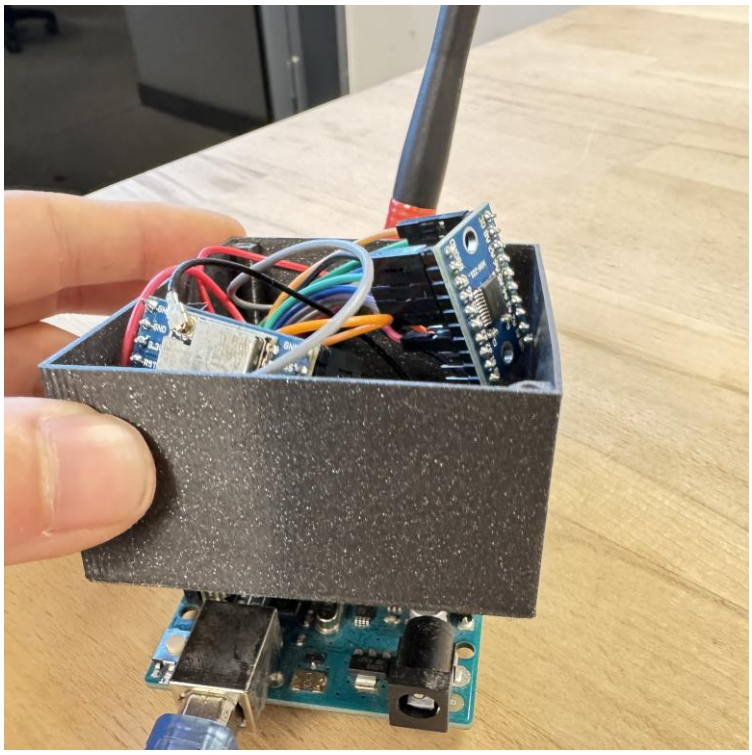
Řídící jednotka z důvodu úspory elektrické energie odesílá telemetrická data každou sekundu. Mezi tím dochází k průběžnému sběru dat, přičemž těsně před odesláním dojde k vytvoření průměrné hodnoty ze všech nasbíraných dat. Tímto procesem se odstraní všechny odchylky a anomálie, ke kterým při měření mohlo dojít. Cyklus tedy funguje tak, že opakovaně za sebou dochází ke čtení hodnot ze senzorů. Po určitém intervalu dojde k **vytvoření průměru z nasbíraných hodnot** a **odeslání skrze protokol LoRa**. To zaručuje vysokou rychlost, přesnost a stabilitu celého systému odesílání telemetrických dat.

2.2.4 Komunikace mezi řídící jednotkou a desktopovou aplikací

Telemetrická data jsou po odeslání řídící jednotkou přijaté modulem **HydraGateway**. Jedná se o jednoduchý transceiver založený na **jednodeskovém počítači Arduino Uno**. Účelem tohoto transceiveru je zajištění komunikace mezi řídící jednotkou HydraBoard a desktopovou aplikací HydraStation, plní zde tedy roli tzv. **proxy**. Z tohoto důvodu je k Arduinu též připojen LoRa transceiver s podporou detekce přijetí dat skrze **interrupt**. Tento transceiver je zároveň naprogramován tak, aby umožňoval obousměrnou aplikaci. Díky tomu je tedy možné odesílat příkazy z desktopové aplikace do řídící jednotky.

Komunikace mezi transceiverem a desktopovou aplikací probíhá skrze **USB sériový port**. Pro výměnu dat se zde využívá jednoduchý **ASCII protokol** založený na **hexadecimálně kódovaných zprávách**, které jsou od sebe oddělné novým řádkem. Výhodou je také možnost v případě nouze či z důvodu testování data jednoduše číst a zapisovat skrze terminál. Aplikace je zároveň schopná rozeznat, zda je zdrojem přijatých dat řídící jednotka či transceiver, neboť transceiver veškerá data směrovaná desktopové aplikaci označuje identifikátorem o velikosti jednoho bytu. To umožňuje aplikaci detekovat případné závady transceiveru.

HydraGateway je z důvodu bezpečnosti elektroniky uzavřen v **jednoduché 3D vytisknuté krabičce** *(viz obr. 4)*, ze které vystupuje pouze USB port a rádiová anténa.



Obr. 4: Otevřená HydraGateway se svým 3D vytisknutým obalem

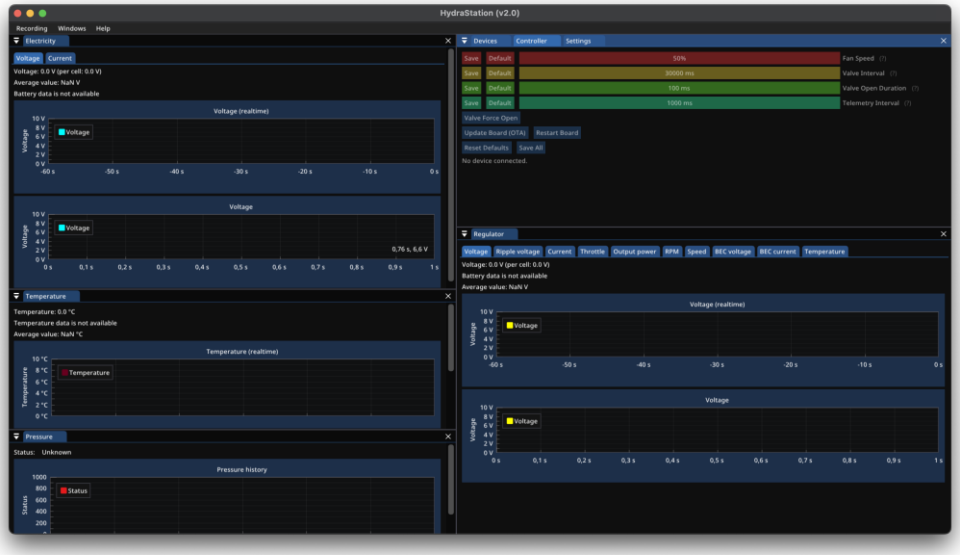
2.2.5 Struktura desktopové aplikace

Pro realizaci desktopové aplikace byl **zvolen jazyk Java**, a to z důvodu jeho jednoduchosti, přenositelnosti a stabilitě. Cílem je vytvořit aplikaci, která bude přijímat, zpracovávat a vizualizovat telemetrická data. Zároveň se v aplikaci nachází několik dalších užitečných funkcí, jako je posílání příkazů řídící jednotce, časovač, nastavení aplikace či živé zobrazování žebříčku probíhajícího závodu. Grafické rozhraní je vytvořeno pomocí knihovny **ImGui**, která umožňuje vytváření pokročilých grafických rozhraní knihovny **ImPlot**, díky které lze vykreslovat grafy v ImGui grafickém rozhraní.

Při vývoji byl kladen důraz na optimalizace, neboť při závodech je často zakázáno využívání elektrické energie v prostorách závodů. Je tedy nutné, aby zařízení s běžící desktopovou aplikací vydrželo fungovat co nejdéle na svůj akumulátor. Zároveň s přibývajícími daty narůstá zátěž na samotnou aplikaci, neboť dochází k vykreslování čím dál většího množství dat.

2.2.6 UX desktopové aplikace

UX desktopové aplikace *(viz obr. 5)* je stavěno tak, aby bylo možné co nejefektivněji vyhodnocovat telemetrická data. Je zde kladen důraz na přizpůsobivost. V rámci aplikace lze podokna různě přesouvat, spojovat, překrývat, ukotvovat, skrývat, zvětšovat či zmenšovat. Aplikace si zároveň nastavení ukládá pro příští otevření. Vykreslované grafy lze zvětšovat, zmenšovat, přibližovat a zpětně prohlížet. Každá metrika má dva grafy – pohyblivý s nejčerstvějšími daty a dlouhodobý pro pokročilejší analýzu dat. Okna lze zobrazovat a skrývat přímo z navigace. V přípravě je též podpora možnosti úprav vzhledu celé aplikace.



Obr. 5: Pohled na aplikaci HydraStation po otevření

2.2.7 Ukládání záznamů o proběhlých závodech

Ukládání telemetrických dat probíhá průběžně po zahájení ukládání telemetrických dat. Pro skladování těchto dat byl vytvořen jednoduchý souborový formát **Hydra Station File**, který je přizpůsobený pro ukládání časosběrných dat. Zároveň lze do formátu uložit i další typy informací, jako jsou například události.

Vlastní formát byl zvolen z důvodu efektivity při ukládání časosběrných telemetrických dat. Zároveň je kladen důraz na co nejmenší velikost finálního souboru.

Explanation

Info

- File extension: .hsd
- Byte order: Big Endian
- If version is older than HydraStation supported version, file must be converted.

Header

File Signature (5 bytes, string, must be **HYDRA** in ASCII)
File Version (2 bytes, int16, must be same as HydraStation Data File supported version)
File Timestamp (4 bytes, int32, timestamp from epoch in seconds)
File Title (2 + [lenght] bytes, string, length + string encoded as UTF8)
File Description (2 + [lenght] bytes, string, length + string encoded as UTF8)

Data

Block Timestamp (4 bytes, int32, timestamp from epoch in seconds)
Block Type (2 bytes, int16, defines block type)
Block Length (4 bytes, int32, length of block content)
Block Content ([Block Length] bytes, byte array, must be parsed by type)
Block Content CRC (4 bytes, int32, CRC32 for detecting broken content)

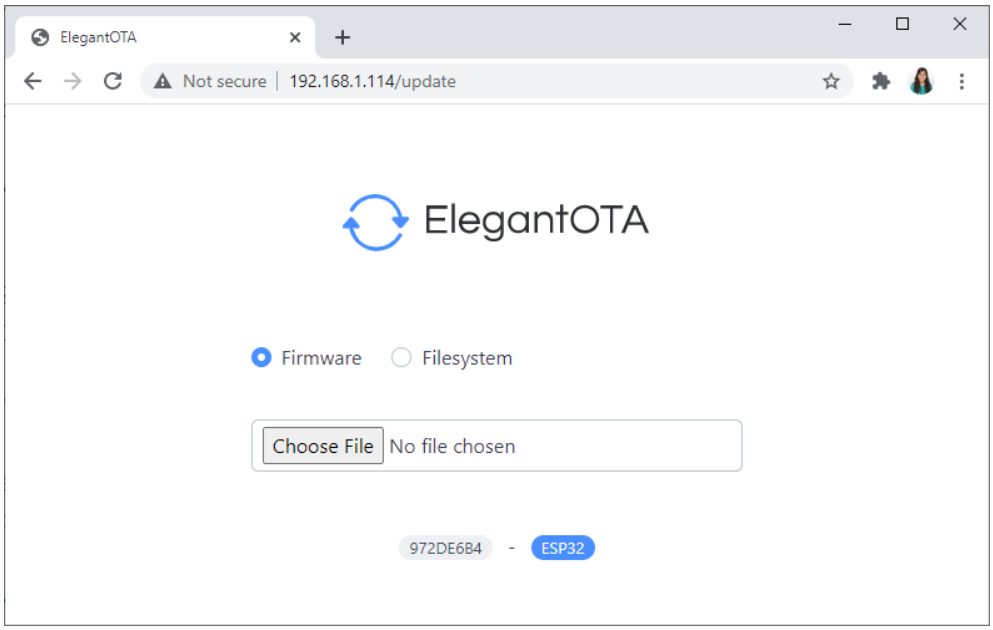
Obr. 6: Jednoduchá dokumentace formátu Hydra Station File

2.2.8 Ovládání řídící jednotky

Desktopová aplikace zároveň podporuje odesílání příkazů řídící jednotce. Díky tomu lze ovládat jednotlivé komponenty autíčka v závislosti na získaných telemetrických datech. Aktuálně lze ovládat **rychlost větráčků palivového článku, dobu a interval vstřikování vodíku a interval odesílání telemetrických dat**. Zároveň lze skrze příkazy na dálku desku restartovat, **vynutit její aktualizaci či vynutit otevření ventilu pro vstřiknutí vodíku do palivového článku**.

2.2.9 Aktualizace řídící jednotky na dálku (OTA)

V případě potřeby lze aktualizovat řídící jednotku na dálku pomocí knihovny AsyncElegantOTA tzv. metodou **OTA**. Díky tomu lze usnadnit aktualizace nejen při vývoji, ale v případě chyby v kódu lze aktualizovat řídící jednotku bez nutnosti návštěvy depa. Aktualizaci na dálku lze vynutit vysláním příkazu z desktopové aplikace HydraStation. Při zahájení dojde ke **spuštění WiFi přístupového bodu** a jednoduché webové stránky. Po připojení k WiFi přístupovému bodu a otevření webové stránky lze skrze formulář *(viz obr. 7)* nahrát binární soubor obsahující nový firmware pro řídící jednotku. Po nahrání dojde k automatické aktualizaci a restartování mikrokontroleru.



Obr. 7: Příklad stránky pro aktualizace na dálku

ZÁVĚR

V rámci práce byl navržen a zrealizován systém pro sběr telemetrických dat. Projekt se skládá z řídící jednotky, brány pro komunikaci a desktopové aplikace. V současné době je tento systém nasazen při závodech. Zároveň tento systém obdržel při závodě **H2GP 2022 cenu za inovace**, která se dává za největší technologický pokrok mezi týmy.

POUŽITÉ ZDROJE

[1] UX – ITNetwork *[cit.11. 1. 2024] [online]*
dostupné z https://www.itnetwork.cz/html-css/user-experience/uvod-do-ux.

[2] Telemetrie – Wikipedie *[cit. 21. 11. 2023] [online]*
dostupné z *https://cs.wikipedia.org/wiki/Telemetrie*.

[3] Jednočipový počítač – Wikipedie *[cit. 21. 11. 2023] [online]*
dostupné z *https://cs.wikipedia.org/wiki/Jednočipový_počítač*.

[4] Firmware – Wikipedie *[cit. 21. 11. 2023] [online]*
dostupné z *https://cs.wikipedia.org/wiki/Firmware*.

[5] LoRa – Wikipedie *[cit. 21. 11. 2023] [online]*
dostupné z *https://cs.wikipedia.org/wiki/LoRa*.

[6] Castle Link Live *[cit. 21. 11. 2023] [online]*
dostupné z *https://www.castlecreations.com/castle-link-live*.