****

**Vyšší odborná škola**

**a Střední průmyslová škola elektrotechnická**

**Plzeň, Koterovská 85**

**ročníková práce s obhajobou**

**Téma: Displej vyhodnocující akciový trh**

**Autor práce: Matyáš Mlnařík**

**Třída: 2. P**

**Vedoucí práce: Miroslav SOUKUP**

**Dne: 7.6.2025**

**Hodnocení:**

**Vyšší odborná škola**

**a Střední průmyslová škola elektrotechnická**

**Plzeň, Koterovská 85**

**Zadání ročníkové práce**

**Žák:** Matyáš Mlnařík

**Třída:** 2.P

**Studijní obor:** 26-41-M/01 Elektrotechnika

**Zaměření:**Internet věcí

**Školní rok:** 2024–2025

**Téma práce: Displej vyhodnocující akciový trh**

**Pokyny k obsahu a rozsahu práce:**

Tato ročníková práce se zabývá návrhem a realizací displeje pro vyhodnocení akciového trhu, který zobrazuje aktuální ceny akcií získané prostřednictvím Finnhub.io API. Systém je navržen jako vzdělávací projekt v rámci 2. ročníku studia oboru Elektrotechnika se zaměřením na Internet věcí. Hlavním cílem je vytvořit funkční IoT zařízení, které kombinuje hardwarový návrh, programování a mechanickou konstrukci, a prohlubuje znalosti studenta v oblasti práce s mikrokontroléry, bezdrátovou komunikací a uživatelskými rozhraními.

Systém využívá mikrokontrolér PIC32CM5164JH01048, který byl zadaný školou, pro zpracování dat a komunikaci s 7palcovým HMI displejem Nextion NX8048P070-011 prostřednictvím UART rozhraní (piny PA22 a PA23). ESP32 modul zajišťuje připojení k internetu přes Wi-Fi a komunikaci s Finnhub.io API přes UART piny PB10 a PB11, aby získal aktuální ceny akcií (např. cena, změna v %, objem obchodů) každých 5 minut. Celé zařízení je napájeno z powerbanky (5V) přes VBUS, bez potřeby dalších stabilizátorů napětí. Grafické rozhraní na Nextion displeji je vytvořeno pomocí Nextion Editoru, který umožňuje zobrazení textových dat a případně grafů prostřednictvím widgetů. Mechanická konstrukce zahrnuje kryt navržený v programu Fusion 360 a vytištěný z PLA materiálu na 3D tiskárně, který chrání komponenty a zajišťuje přístup k portům a chlazení.

**Určení částí tématu zpracovávaných jednotlivými žáky:**

1. HW – Elektrotechnický návrh zařízení

Popis zapojení mikrokontroléru PIC32CM5164JH01048, ESP32 modulu a Nextion displeje, včetně výběru součástek, blokového schématu a schématu zapojení. Popis postupu při vytváření zapojení, včetně použití vývojových desek a propojovacích kabelů.

1. FW – Vývoj programu pro zařízení

Popis programu pro získávání dat z Finnhub.io API na ESP32, jejich přenos přes UART na PIC32CM a zobrazení na Nextion displeji.

Detailní popis programování v MPLAB X pro PIC32CM a Nextion Editoru pro uživatelské rozhraní, včetně omezení programu (např. závislost na internetu, rychlost odezvy API).

1. Návrh mechanických částí zařízení

Popis návrhu krytu v programu Fusion 360, vytištěného z PLA na 3D tiskárně, včetně rozměrů, otvorů pro kabely a chlazení.

Popis výběru materiálu a procesu 3D tisku pro zajištění odolnosti a estetického vzhledu.

***Požadavek na počet vyhotovení ročníkové práce:*** *1 výtisk*

Termín odevzdání: **8. června 2025**

Čas obhajoby: **5 minut + 5 Dotazy**

Vedoucí práce: **Miroslav Soukup**

**Anotace:** Tato ročníková práce se zabývá návrhem a realizací displeje pro vyhodnocení akciového trhu, který zobrazuje aktuální ceny akcií získané prostřednictvím Finnhub.io API. Systém je vytvořen v rámci 2. ročníku studia oboru Elektrotechnika se zaměřením na Internet věcí a využívá vývojovou desku PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit s mikrokontrolérem PIC32CM5164JH01048 pro zpracování dat a komunikaci s 7palcovým HMI displejem Nextion NX8048P070-011 přes UART sběrnici. ESP32 modul zajišťuje Wi-Fi připojení a komunikaci s API. Zařízení je napájeno z powerbanky (5V) a umístěno v krytu navrženém v programu Fusion 360, vytištěném z PLA na 3D tiskárně.

**Poděkování:** Rád bych poděkoval bratrům Soukupům za tři roky obohacování mých znalostí z oblasti Internetu věcí. Jejich cenné rady, podpora a odborné vedení mi významně pomohly při realizaci této ročníkové práce a prohloubení mých dovedností v programování a návrhu IoT zařízení.

**Čestné prohlášení:**

„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.“  
„Souhlasím s využitím mé práce učiteli VOŠ a SPŠE Plzeň k výuce.“

V Plzni dne: 2.6. 2025 Podpis: Mlnařík

Obsah

[1. Úvod 5](#_Toc130145254)

[2. HW – Elektrotechnický návrh zařízení 6](#_Toc130145255)

[2.1 Výběr vhodných součástek 6](#_Toc130145256)

[2.2 Blokové schéma 6](#_Toc130145257)

[2.3 Schéma zapojení 6](#_Toc130145258)

[2.4 Postup výroby plošného spoje 6](#_Toc130145259)

[3. FW – Vývoj programu pro zařízení 7](#_Toc130145260)

[3.1 Textový popis programu 7](#_Toc130145261)

[3.2 Vývojový diagram programu 7](#_Toc130145262)

[3.3 Postup vývoje programu 7](#_Toc130145263)

[4. Návrh mechanických částí zařízení 8](#_Toc130145264)

[4.1 Výběr / návrh mechanických částí zařízení 8](#_Toc130145265)

[5. Odzkoušení hotové ročníkové práce v praxi 9](#_Toc130145266)

[6. Závěr 10](#_Toc130145267)

[7. Seznam obrázků 11](#_Toc130145268)

[8. Reference 12](#_Toc130145269)

# Úvod

Tato ročníková práce se zabývá návrhem a realizací displeje pro vyhodnocení akciového trhu, který zobrazuje aktuální ceny akcií získané prostřednictvím Finnhub.io API. Systém je založen na vývojové desce PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit s mikrokontrolérem PIC32CM5164JH01048, zadané školou, který řídí komunikaci s 7palcovým HMI displejem Nextion NX8048P070-011 přes UART piny PA11 (TX) a PA25 (RX). PIC32CM posílá příkazy na ESP32 modul přes UART piny PB10 (TX) a PB11 (RX), který zajišťuje Wi-Fi připojení a komunikaci s API každých 5 minut. Zařízení je napájeno z powerbanky (5V) přes VBUS a umístěno v krytu navrženém v programu Fusion 360, vytištěném z PLA na 3D tiskárně.

Téma jsem si vybral z důvodu zájmu o finanční trhy a studiu oboru Internet věcí na Vyšší odborné škole a Střední průmyslové škole elektrotechnické v Plzni. Projekt mi umožnil propojit znalosti programování, hardwarového návrhu a bezdrátové komunikace, abych vytvořil funkční IoT zařízení pro zobrazení finančních dat v reálném čase. Jako součást 2. ročníku studia má práce za cíl prohloubit moje dovednosti v oblasti IoT technologií, práce s API, řízení periferií přes UART a návrhu uživatelského rozhraní, a přispět k pochopení praktického využití mikrokontrolérů v reálných aplikacích.

# HW – Elektrotechnický návrh zařízení

Tato ročníková práce se zabývá návrhem a realizací displeje pro vyhodnocení akciového trhu, který zobrazuje aktuální ceny akcií získané prostřednictvím Finnhub.io API. Systém je postaven na vývojové desce PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit s mikrokontrolérem PIC32CM5164JH01048, zadané školou, který zpracovává data a komunikuje s 7palcovým HMI displejem Nextion NX8048P070-011 přes UART piny PA11 (TX) a PA25 (RX). ESP32 modul zajišťuje Wi-Fi připojení a komunikaci s API přes UART piny PB10 (TX) a PB11 (RX) každých 5 minut. Zařízení je napájeno z powerbanky (5V) přes VBUS a umístěno v krytu navrženém v programu Fusion 360, vytištěném z PLA na 3D tiskárně.

## Výběr vhodných součástek

* **PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit**: Zadaná školou, obsahuje mikrokontrolér PIC32CM5164JH01048 s podporou UART a dotykového ovládání, vhodný pro zpracování dat a komunikaci.
* **ESP32**: Zvolen pro spolehlivé Wi-Fi připojení, podporu knihoven pro API (např. HTTPClient)
* **Nextion NX8048P070-011**: 7palcový HMI displej (800x480) pro jednoduché programování přes Nextion Editor a zobrazení textů i grafiky.
* **Powerbanka (5V)**

## Blokové schéma

Obsah obrázku text, diagram, řada/pruh, Paralelní

Obsah generovaný pomocí AI může být nesprávný.

## Schéma zapojení

Obsah obrázku diagram, text, skica, Plán

Obsah generovaný pomocí AI může být nesprávný.

Obrázek 1 blokove schema

## Postup výroby plošného spoje

Obsah obrázku nářadí, kabel, interiér, Elektrické vedení

Obsah generovaný pomocí AI může být nesprávný.

Obrázek 2 Pajka

Obsah obrázku text, elektronika, Obvodoví součástka, Elektronická součástka

Obsah generovaný pomocí AI může být nesprávný.

Obrázek 3 pohled zevnitř

# FW – Vývoj programu pro zařízení

Firmware je rozdělen na program pro PIC32CM (v C, MPLAB X), který řídí systém a posílá příkazy na ESP32, který zpracovává příkazy a komunikuje s Finnhub.io API.

Další je použito rozhraní pro Nextion (Nextion Editor). PIC32CM zpracovává přijatá data a odesílá je na Nextion pro zobrazen

## Textový popis programu

Firmware zařízení pro displej vyhodnocující akciový trh zajišťuje spolupráci mezi vývojovou deskou **PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit** s mikrokontrolérem **PIC32CM5164JH01048**, **ESP32** modulem a 7palcovým HMI displejem **Nextion NX8048P070-011**. Program je rozdělen do 2 hlavních částí: firmware pro PIC32CM (vytvořený v MPLAB X v jazyce C), včetně ESP32 a uživatelské rozhraní pro Nextion (navržené v Nextion Editoru). Hlavním úkolem firmwaru je pravidelné získávání dat o akciích z **Finnhub.io API**, jejich zpracování a zobrazení na displeji v reálném čase každých 5 minut.

Porgram se po připojení k wifi a ověření pomocí api klíče zavolá o .json soubor, který následnedně parsuje a již rozdělené hodnoty zobrazí na displeji.

## Přesný popis programu

### 1. Funkce programu na PIC32CM

PIC32CM5164JH01048 je centrální řídicí jednotkou systému a zodpovídá za koordinaci komunikace mezi ESP32 a Nextion displejem. Firmware je napsán v jazyce C v prostředí MPLAB X s využitím knihovny **MPLAB Harmony** pro konfiguraci periferií.

* **Inicializace**:  
  Program inicializuje UART moduly na pinech **PA11 (TX)** a **PA25 (RX)** pro komunikaci s Nextionem a **PB10 (TX)** a **PB11 (RX)** pro komunikaci s ESP32. Používá se funkce SERCOM\_UART\_Init() z knihovny MPLAB Harmony pro nastavení baud rate (např. 115200 bps) a formátu přenosu (8 bitů, bez parity, 1 stop bit). Dále je inicializován časovač (TC\_Start()) pro generování přerušení každých 5 minut, které spouští odeslání příkazu na ESP32.
* **Odesílání příkazů**:  
  Každých 5 minut PIC32CM odesílá textový příkaz na ESP32 přes UART pomocí funkce SERCOM\_UART\_Write(). Příkaz má formát jednoduchého textového řetězce, např. FETCH\_AAPL\n, který instruuje ESP32, aby získal data o akciích společnosti Apple z Finnhub.io API. Příkazy jsou jednoduché, aby minimalizovaly složitost parsování na straně ESP32.
* **Příjem a zpracování dat**:  
  Po obdržení dat od ESP32 (např. řetězec „AAPL:150.25:+1.2\n“) přes UART (pomocí SERCOM\_UART\_Read() v přerušení), PIC32CM parsuje data pomocí vlastní funkce parse\_stock\_data(), která extrahuje cenu a procentuální změnu. Tato data jsou poté formátována do příkazů pro Nextion (např. „t0.txt=150.25“ pro textové pole ceny) a odeslána přes UART na Nextion pomocí SERCOM\_UART\_Write().
* **Použité funkce**:
  + SERCOM\_UART\_Init(): Inicializace UART periferií.
  + SERCOM\_UART\_Write(): Odeslání dat přes UART (příkazy na ESP32, data na Nextion).
  + SERCOM\_UART\_Read(): Čtení dat z UART (odpovědi od ESP32).
  + TC\_Start(): Spuštění časovače pro pravidelné přerušení (5 minut).
  + parse\_stock\_data(): Vlastní funkce pro parsování přijatých dat (např. rozdělení řetězce na cenu a změnu).
  + NVIC\_EnableIRQ(): Aktivace přerušení pro asynchronní zpracování UART.
* **Proč takto?**  
  PIC32CM je použit jako hlavní řídicí jednotka, protože má dostatečný výkon pro koordinaci komunikace a zpracování dat. Použití přerušení (NVIC\_EnableIRQ()) místo funkce delay() zajišťuje asynchronní zpracování, což zabraňuje blokování programu při čekání na data od ESP32 nebo Nextionu. Funkce delay() nebyla použita, protože by způsobila zpoždění v reakci systému a mohla by vést k ztrátě dat při souběžné UART komunikaci. Asynchronní přístup umožňuje rychlé zpracování příkazů a odpovědí, což je kritické pro reálný čas zobrazení dat.

### 3. Funkce programu na Nextion

Uživatelské rozhraní na Nextion NX8048P070-011 je navrženo v Nextion Editoru a zahrnuje textová pole a případně grafické widgety pro zobrazení dat.

* **Zobrazení dat**:  
  Nextion přijímá data od PIC32CM přes UART (PA22 jako RX, PA23 jako TX) ve formátu příkazů Nextion protokolu. Například příkaz „t0.txt=150.25“ nastaví textové pole t0 na hodnotu 150.25 (cena akcie). Pro zobrazení procentuální změny je použit příkaz jako „t1.txt=+1.2“. Pokud jsou zobrazeny grafy, např. křivka ceny, používá se widget waveform s příkazy jako „add 1,0,150“ pro přidání hodnoty do křivky.
* **Použité funkce**:  
  Nextion Editor nevyžaduje programování v tradičním smyslu, ale definuje příkazy pro ovládání widgetů:
  + t0.txt: Nastavení textového pole pro cenu.
  + t1.txt: Nastavení textového pole pro procentuální změnu.
  + add: Přidání dat do grafického widgetu (např. křivky ceny).
  + ref: Obnovení zobrazení widgetu po aktualizaci.
* **Proč takto?**  
  Nextion protokol je jednoduchý a efektivní pro zobrazení dat na HMI displeji. Použití textových příkazů minimalizuje zátěž na PIC32CM a umožňuje rychlé aktualizace. Funkce delay() není relevantní pro Nextion, protože zpracování probíhá na straně PIC32CM, který řídí časování příkazů.

## Proč některé funkce nebyly použity

Funkce delay() nebyla použita v žádném modulu (PIC32CM ani ESP32), protože by způsobila blokování programu, což by vedlo k:

* **Zpoždění v komunikaci**: Čekání na data od ESP32 nebo Nextionu by způsobilo ztrátu synchronizace při UART přenosu.
* **Zmeškané odpovědi**: Při použití delay() by mohla být zmeškána odpověď z API nebo příkaz od PIC32CM.
* **Neefektivní provoz**: Systém by nemohl souběžně zpracovávat více úkolů (např. příjem dat a odesílání na displej).  
  Místo toho jsou použity přerušení (NVIC\_EnableIRQ() na PIC32CM, události na ESP32) a asynchronní zpracování, které zajišťují plynulý chod a rychlou odezvu systému.

## Omezení programu

* **Závislost na internetu**: Funkčnost závisí na stabilním Wi-Fi připojení a odezvě Finnhub.io API.
* **Paměť Nextionu**: Omezená paměť displeje omezuje složitost grafů (např. počet bodů v křivce).
* **UART synchronizace**: Nesprávné časování může vést k chybám v přenosu, což bylo vyřešeno přerušeními.
* **API limity**: Finnhub.io API má omezení na počet požadavků (např. 60 za minutu), proto je aktualizace nastavena na 5 minut.

## Vývojový diagram programu

Obsah obrázku diagram, skica, kresba, Technický výkres

Obsah generovaný pomocí AI může být nesprávný.

Obrázek 4 diagram

## Postup vývoje programu

Vývoj firmwaru pro zařízení displeje vyhodnocujícího akciový trh probíhal ve více fázích, aby byla zajištěna funkčnost systému založeného na vývojové desce PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit s mikrokontrolérem PIC32CM5164JH01048, ESP32 modulu a 7palcovém HMI displeji Nextion NX8048P070-011. Firmware byl vyvíjen ve dvou prostředích: MPLAB X pro PIC32CM (v jazyce C), Nextion Editor pro návrh uživatelského rozhraní. Proces zahrnoval implementaci příkazové komunikace, získávání dat z Finnhub.io API, jejich přenos a zobrazení, s důrazem na synchronizaci a stabilitu systému.

**Fáze vývoje**

1. **Inicializace a testování jednotlivých komponent**  
   Vývoj začal konfigurací UART komunikace na PIC32CM (piny PA11 jako TX a PA25 jako RX pro Nextion, PB10 jako TX a PB11 jako RX pro ESP32). V MPLAB X byl vytvořen základní kód pro inicializaci mikrokontroléru a testování přenosu jednoduchých zpráv na Nextion displej. Nextion Editoru byl navržen základní návrh rozhraní s textovými poli pro zobrazení ceny akcií a změny v procentech.
2. **Implementace příkazové komunikace mezi PIC32CM a ESP32**  
   Druhá fáze se zaměřila na vytvoření protokolu pro komunikaci mezi PIC32CM a ESP32. PIC32CM posílal jednoduché textové příkazy (např. „FETCH\_AAPL“ pro získání dat o akciích společnosti Apple) na ESP32 přes UART. ESP32 zpracovával příkazy, prováděl HTTP požadavky na Finnhub.io API a vracel data (např. JSON formát s cenou a změnou v %) zpět na PIC32CM. Tato fáze vyžadovala pečlivé ladění, protože nesprávná synchronizace UART vedla k chybám v přenosu.
3. **Získávání a zpracování dat z Finnhub.io API**  
   Na ESP32 byl implementován kód pro pravidelné volání Finnhub.io API (každých 5 minut) na základě příkazů od PIC32CM. Data byla parsována pro přesné odsílání.

# Návrh mechanických částí zařízení

Hlavním mechanickým prvkem je kryt, který spojuje vývojovou desku PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit s mikrokontrolérem PIC32CM5164JH01048, ESP32 modul a 7palcový HMI displej Nextion NX8048P070-011. Kryt byl navržen v programu Autodesk Fusion 360 a vytištěn z PLA materiálu na 3D tiskárně, aby splňoval požadavky na odolnost, funkčnost a přístupnost komponent.

## Výběr / návrh mechanických částí zařízení

Mechanická konstrukce zařízení zahrnuje kryt, který chrání vývojovou desku PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit s mikrokontrolérem PIC32CM5164JH01048, ESP32 modul a 7palcový HMI displej Nextion NX8048P070-011. Kryt byl navržen v programu Autodesk Fusion 360 a vytištěn z PLA materiálu na 3D tiskárně, aby zajistil ochranu komponent, estetický vzhled a funkční přístup k portům a chlazení.

**Výběr materiálu**  
Pro výrobu krytu byl zvolen PLA (polylaktidový) materiál díky jeho dostupnosti, nízké ceně, snadné tisknutelnosti a dostatečné mechanické odolnosti. PLA je ekologický, lehký a vhodný pro 3D tisk dílů, které nevyžadují extrémní teplotní nebo mechanickou odolnost. Tloušťka stěn krytu byla navržena na 2 mm, aby zajistila pevnost při zachování nízké hmotnosti. Žádné hotové mechanické díly nebyly použity, protože kryt byl plně navržen a vytištěn na míru potřebám zařízení.

**Návrh krytu**  
Kryt byl navržen v programu Fusion 360 s ohledem na rozměry hlavní komponenty – Nextion displeje (218.1 mm x 150 mm x 22.5 mm), který určuje celkové rozměry zařízení. Rozměry krytu byly stanoveny na přibližně 230 mm x 160 mm x 40 mm, aby poskytoval dostatek prostoru pro vývojové desky (PIC32CM JH01 Curiosity Nano+ a ESP32) a propojovací kabely. Kryt obsahuje následující prvky:

* **Otvory pro displej**: Přední část krytu má výřez pro displej Nextion (přibližně 180 mm x 110 mm), aby byl zajištěn plný přístup k obrazovce a případnému dotykovému ovládání.
* **Větrací otvory**: Zadní část krytu obsahuje mřížku s otvory (10 mm x 20 mm,) pro pasivní chlazení, aby se zabránilo přehřívání ESP32 a PIC32CM při dlouhodobém provozu.

**Proces návrhu a výroby**  
Návrh krytu začal vytvořením 3D modelu v Fusion 360, kde byly importovány rozměry komponent z jejich technických listů. Po vytvoření základního tvaru krytu byly přidány funkční prvky (otvory, držáky) a model byl optimalizován pro 3D tisk (např. minimalizací převisů a podpůrných struktur). Model byl exportován jako STL soubor a připraven pro tisk v slicovacím softwaru. Tisk proběhl na 3D tiskárně s PLA filamentem o průměru 3 mm, při teplotě trysky 200 °C a teplotě podložky 60 °C. Doba tisku byla přibližně 6 hodin při vrstvě 0.2 mm. Po tisku byl kryt očištěn od drobných nedokonalostí a otestován z hlediska přesnosti uložení komponent a funkčnosti otvorů.

# Odzkoušení hotové ročníkové práce v praxi

Systém, založený na vývojové desce PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit s mikrokontrolérem PIC32CM5164JH01048, ESP32 modulu pro komunikaci s Finnhub.io API a 7palcovém HMI displeji Nextion NX8048P070-011, byl testován z hlediska spolehlivosti komunikace, přesnosti zobrazení dat a stability napájení z powerbanky (5V).

Během vývoje a testování se objevily dlouhodobé problémy s funkčností zařízení, zejména v oblasti UART komunikace mezi PIC32CM, ESP32 a Nextion displejem, a také při zpracování dat z Finnhub.io API. Tyto problémy zahrnovaly nepravidelný přenos dat mezi moduly a občasné selhání zobrazení na Nextion displeji kvůli omezené paměti nebo nesprávnému formátování dat. Po ladění firmwaru v MPLAB X pro PIC32CM a optimalizaci příkazové komunikace na ESP se tyto problémy podařilo vyřešit.

# Závěr

Tato ročníková práce se zabývala návrhem a realizací IoT zařízení – displeje pro vyhodnocení akciového trhu, který zobrazuje aktuální ceny akcií získané prostřednictvím Finnhub.io API. Projekt byl vytvořen v rámci 2. ročníku studia oboru Elektrotechnika se zaměřením na Internet věcí na Vyšší odborné škole a Střední průmyslové škole elektrotechnické v Plzni. Systém využívá vývojovou desku PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit s mikrokontrolérem PIC32CM5164JH01048, který řídí komunikaci s 7palcovým HMI displejem Nextion NX8048P070-011 přes UART piny PA11 a PA25 a posílá příkazy na ESP32 modul přes UART piny PB10 a PB11 pro získání dat z API. ESP32 zajišťuje Wi-Fi připojení a komunikaci s Finnhub.io API. Zařízení je napájeno z powerbanky (5V) a umístěno v krytu navrženém v programu Fusion 360, vytištěném z PLA na 3D tiskárně. Práce zahrnovala hardwarový návrh, vývoj firmwaru v MPLAB X, a Nextion Editoru a mechanickou konstrukci krytu.

Během realizace projektu jsem narazil na několik výzev. Největší obtíže představovala synchronizace UART komunikace mezi PIC32CM, ESP32 a Nextionem, zejména kvůli správnému časování příkazů a zpracování dat. Dalším problémem bylo ladění grafického rozhraní na Nextion displeji, kde omezená paměť displeje ztěžovala zobrazení složitějších grafů. Tyto překážky se podařilo překonat pečlivým testováním a optimalizací kódu, což však vyžadovalo značné úsilí a trpělivost.

Můj výkon při práci na projektu hodnotím jako Dobrý. Podařilo se mi splnit hlavní cíle – vytvořit funkční zařízení, které zobrazuje aktuální finanční data, a propojit hardwarové, softwarové a mechanické části do jednoho celku. Zároveň jsem si uvědomil, že některé části, jako například návrh robustnějšího firmwaru nebo detailnější dokumentace, by mohly být vylepšeny při větším časovém prostoru. Práce na projektu mě naučila několik klíčových dovedností: práci s mikrokontroléry a UART komunikací, integraci API do IoT zařízení, programování v různých prostředích (MPLAB X, Nextion Editor) a základy 3D modelování v Fusion 360. Získal jsem také lepší pochopení propojení hardwaru a softwaru v reálných aplikacích a důležitosti testování a ladění.

Do budoucna vidím potenciál pro další rozvoj projektu. Plánuji přidat funkce, jako je podpora více akciových trhů, interaktivní ovládání přes dotykový displej Nextion nebo integrace webového rozhraní pro vzdálený přístup k datům. Rád bych také vylepšil kryt zařízení, aby byl odolnější a estetičtější, a optimalizoval firmware pro rychlejší odezvu. I když v nejbližší době neplánuji intenzivní práci na projektu kvůli studijním povinnostem, rád bych se k němu vrátil a využilvzískané zkušenosti v dalších IoT aplikacích.

Celkově mi ročníková práce přinesla cenné zkušenosti a posunula moje technické dovednosti na novou úroveň. Jsem rád/a, že jsem mohl propojit svůj zájem o finanční trhy s technologiemi IoT, a těším se na další projekty, které mi umožní tyto znalosti dále rozvíjet.

# Seznam obrázků

[Obrázek 1 blokove schema 8](#_Toc200236625)

[Obrázek 2 Pajka 9](#_Toc200236626)

[Obrázek 3 pohled zevnitř 9](#_Toc200236627)

[Obrázek 4 diagram 13](#_Toc200236628)

# Reference

1. Microchip Technology. (2023). *PIC32CM JH01 Curiosity Nano+ Touch Evaluation Kit User Guide*. Dostupné z: https://www.microchip.com/en-us/development-tool/EV87M29A  
   (Dokumentace k vývojové desce PIC32CM JH01 Curiosity Nano+ Touch, použité pro návrh zapojení a programování mikrokontroléru PIC32CM5164JH01048.)
2. Espressif Systems. (2024). *ESP32 Technical Reference Manual*. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32\_technical\_reference\_manual\_en.pdf  
   (Technická dokumentace k ESP32 modulu, využitá pro nastavení Wi-Fi komunikace a UART příkazů.)
3. Nextion. (2024). *Nextion NX8048P070-011 User Manual*. Dostupné z: https://nextion.tech/datasheets/nx8048p070-011/  
   (Dokumentace k HMI displeji Nextion NX8048P070-011, použitá pro návrh uživatelského rozhraní v Nextion Editoru.)
4. Finnhub. (2024). *Finnhub API Documentation*. Dostupné z: https://finnhub.io/docs/api  
   (Oficiální dokumentace Finnhub.io API, použitá pro implementaci volání dat o akciích.)
5. Autodesk. (2024). *Fusion 360 Help – 3D Printing Guidelines*. Dostupné z: https://help.autodesk.com/view/fusion360/ENU/?guid=GUID-3D-PRINTING  
   (Průvodce pro návrh 3D modelů v Fusion 360, využitý pro návrh krytu zařízení.)
6. Microchip Technology. (2024). *MPLAB X IDE User Guide*. Dostupné z: https://www.microchip.com/en-us/development-tools-tools-and-software/mplab-x-ide  
   (Dokumentace k MPLAB X, použitá pro programování PIC32CM v jazyce C.)
7. Espressif Systems. (2024). *ESP-IDF Programming Guide*. Dostupné z: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/  
   (Průvodce pro vývoj firmwaru na ESP32 v prostředí ESP-IDF)
8. Nextion. (2024). *Nextion Editor User Guide*. Dostupné z: https://nextion.tech/nextion-editor/  
   (Průvodce pro Nextion Editor, využitý pro vytvoření grafického rozhraní na displeji.)