

Maturitný projekt

Internet vecí

Zber dát cez zariadenie internetu vecí

Maturitný projekt

Internet vecí

Zber dát cez zariadenie internetu vecí

Čestné prehlásenie

Dolu podpísaní, Marek Ratzenböck, žiak IV.C triedy čestne prehlasuje, že pracoval samostatne s konzultáciami s Ing. Romanom Kissom a že všetku použitú literatúru uvádza na príslušnom mieste.

Dňa: _____

Podpis: _____

PodĎakovanie

Na úvod sa chcem poďakovať Ing. Romanovi Kissovi, Ing. Jozefovi Dragulovi a Ing. Viktorovi Bagínovi za všetky odborné konzultácie a cenné rady k projektu.

Obsah

Obsah	1
1 Zoznam obrázkov	2
2 Zoznam tabuliek a grafov.....	2
3 Úvod a cieľ práce	3
4 Vysvetlenie pojmov.....	4
4.1 Firmvér	4
4.2 Kompilátor.....	4
4.3 Debugging	5
4.4 Algoritmus.....	5
4.5 Mikrokontrolérová jednotka.....	5
4.6 Embedded.....	5
4.7 Telemetry.....	6
4.8 Git	6
4.9 Modem	6
4.10 LoRa	6
5 Realizácia	7
5.1 Inštalácia softvérových nástrojov	8
5.2 Trieda Blink	8
5.3 Hlavný Program.cs.....	10
5.4 Senzor teploty a vlhkosti SHT40.....	10
5.5 Modul GPS	11
5.6 Snímač vzdialenosti ToF	11
5.7 Mikrokontrolér AtomS3 Lite	12

5.8	LoRa modem RYLR998	13
5.9	LoRa modem E22-900T22D	15
5.10	Prijímacia stanica E22-900T22U a COM port reader	16
5.11	Kompletizácia	17
1	Záver	18
6	Zoznam zdrojov	1
7	Zoznam príloh	1

1 Zoznam obrázkov

Obrázok 1	AtomS3Lite s rozsvietenou LED	9
Obrázok 2	senzor ENV IV	11
Obrázok 3	GPS modem	11
Obrázok 4	AtomS3Lite - pohľad na zadnú stranu	12
Obrázok 5	medem RYLR998	13
Obrázok 6	Prostredie Powershell ISE	14
Obrázok 7	Grafická vizualizácia GPS súradníc	16
Obrázok 8	Finálne zariadenie	17

2 Zoznam tabuliek a grafov

Tabuľka 1	Parametre funkcie Blinks	9
Tabuľka 2	vlastnosti AtomS3Lite	12

3 Úvod a cieľ práce

Internet vecí je technológia umožňujúca prepojenie veľkého množstva fyzických zariadení s internetom. Pre internet vecí sa zaužíva skratka IoT. Najčastejšie sa prepojenie rieši cez bezdrôtové technológie ako napríklad WiFi, Bluetooth, LoRa a podobné. IoT je uplatniteľné takmer v každej oblasti od priemyslu cez domácnosti až po zdravotníctvo alebo poľnohospodárstvo. Vďaka väčšiemu množstvu dát, ktoré sú zbierané prostredníctvom snímačov na diaľku má správca systému väčší prehľad o jeho stave a vie sa v čas vyvarovať chyby, zistiť stav opotrebenia stroja, alebo vykonať iný zásah. IoT kombinuje prvky z technológií ako telemetria, embedded (vstavaný alebo vnorený systém), informačné siete, prenos rádiovými a inými. Výzvou pre oblasť IoT je zvládnutie zabezpečenia pred neoprávneným vniknutím do tejto sústavy. Riešením by malo byť použitie kryptografie a jej odvetví ako je biometria.

Prototyp je schopný komunikovať bezdrôtovo, prenášať namerané údaje zo senzorov ako hodnoty vlhkosti, teploty, tlaku ovzdušia... Prístroj pracuje šetrne s energiou, aby bol schopný byť poháňaný z batérie relatívne dlhý čas. Celý projekt je navrhnutý tak, aby bol škálovateľný a ľahko rozšíriteľný inými podradenými modulmi, alebo aby bol sám použiteľný ako modul - súčasť väčšej štruktúry. Uprednostňoval som štandardné komunikačné protokoly. Na rady mojich konzultantov som softvérovú časť priebežne nahrával do verejne prístupného úložiska s podporou dohľadu nad verziami.

Do problematiky som sa dostal počas tretieho ročníka na predmete Programovania mikrokontrolérov. Ešte pred tým som o tejto technológii nevedel takmer nič a deje na pozadí práce IoT systémov mi boli záhadou. Po dokončení tejto práce, by som mal ovládať základy programovania, práce s mikrokontrolérmi a bezdrôtovom prenose LoRa. Praktický výstup z tejto práce použijem i pre vlastnú potrebu na riešenie problému. Otázkou zostáva, či moje riešenie bude lacnejšie, ako už dostupné zariadenia. Ak nie, či jeho škálovateľnosť a optimalizácia na konkrétny účel vykompenzujú vyššie náklady. V každom prípade si z tejto výzvy odnesiem skúsenosti, ktoré budú snáď využiteľné v budúcnosti.

4 Vysvetlenie pojmov

V tomto dokumente budem používať výrazy, ktoré pre lepšie porozumenie bude lepšie vysvetliť. Základné termíny vysvetlím veľmi stručne o niečo nižšie, špecifické pojmy rozvediem v samostatných podkapitolách.

Výpočtová technika slúži na spracovanie číselných operácií strojom. Počítač je nástroj, ktorý okrem svojej hlavnej úlohy – vypočítavať zvláda často tieto údaje i zobrazovať a ukladať. Digitálne technológie pracujú s binárnym systémom, základnou jednotkou údaju je bit a môže mať hodnotu buď jedna, alebo nula. Jadrom počítača je tzv. CPU – z ang.: central processing unit – hlavná výpočtová jednotka. Vstupné dáta spracováva podľa inštrukcií daných strojovým kódom. Strojový kód je najnižšia úroveň programového vybavenia. Programátor – človek zadávajúci počítaču úlohy – píše kód v programovacom jazyku, ktorý je vyššej úrovne a je pre ľudí čitateľnejší. Kód vyššej úrovne sa na ten strojový prekladá tzv. kompilátorom. Kódové, programové, vybavenie počítača sa nazýva softvér. Mechanické, elektronické a všeobecne hmatateľné komponenty sa volajú hardvér. Periférne zariadenia rozširujú schopnosti počítačov o možnosť snímať stavy z okolia, ale aj ovplyvňovať ho. Periférie sa preto delia na vstupné – senzorické a výstupné – aktuálne / akčné.

4.1 Firmvér

Firmvér (anglicky "firmware") je špeciálny typ softvéru, ktorý je trvalo uložený na hardvéri zariadenia. Je to program, ktorý ovláda a riadi hardvérové komponenty a zariadenia, ako sú napríklad počítače, smartfóny, tlačiarne, routery a ďalšie elektronické prístroje. Firmvér je zvyčajne uložený v pamäti typu ROM (Read-Only Memory) alebo flash pamäti.

4.2 Kompilátor

Kompilátor je špeciálny typ softvéru, ktorý prekladá zdrojový kód napísaný v programovacom jazyku (ako je C, C++, Java, a podobne) do strojového kódu, ktorý môže vykonávať počítač. Tento proces prekladu sa nazýva kompilácia.

4.3 Debugging

Debugovanie, debugging by sa dalo voľne preložiť ako „odvšivenie“ softvéru. Je to rada úkonov vedúcich k odstráneniu chýb v kóde. Debugovaniu predchádza testovanie, pri ktorom sa chyby snažíme odhaliť. Často sa chyby do programu dostanú pridaním veľkého množstva nových funkcií naraz.

4.4 Algoritmus

Algoritmus je poradie krokov vedúce k nejakému výsledku. Vlastnosti algoritmu sú jeho determinovanosť: rovnaké vstupy vedú k rovnakému výsledku (neplatí u náhodných generátorov a pod.), všeobecnosť: nerieši jeden konkrétny problém (koľko je $3 \cdot 3$), ale hromadu (súčin dvoch čísel).

4.5 Mikrokontrolérová jednotka

Mikrokontrolérová jednotka (MCU, Microcontroller Unit) je integrovaný obvod, ktorý obsahuje procesor, pamäť a vstupno-výstupné periférie v jednom puzdre. MCU je navrhnutá na ovládanie a riadenie hardvéru a je široko používaná v embedded systémoch, ktoré vykonávajú špecifické úlohy a nie sú určené na obecné výpočty ako počítače.

4.6 Embedded

Embedded znamená vstavaný. V technickom kontexte sa toto pomenovanie vzťahuje na stroje, predmety a nástroje do ktorých je integrovaný nejaký druh počítača. Asistuje, dopĺňa funkcionality vecí, do ktorej je vstavaný, alebo je podmienkou plnohodnotnej práce. Embedded prístroje pracujú zväčša s obmedzenými zdrojmi elektrickej energie i výpočtového výkonu. Príkladom embedded systémov môžu byť: palubné počítače v automobiloch, lietadlách, doplnkové funkcie ako ABS, samočinné stieranie okien, ovládacie jednotky spotrebičov ako kávovary, práčky, televízie...

4.7 Telemetria

Telemetria slúži na meranie a prenos údajov na diaľku. Pôvodne sa používala v oblasti telekomunikácií, ale dnes sa uplatňuje v rôznych oblastiach, ako je monitorovanie vozidiel, plynárenský priemysel a vesmírne technológie. Prenos sa môže realizovať rádiovými, infračervenými, optickými a elektrickými médiami. Digitalizácia zberu a prenosu je dnes štandardom.

4.8 Git

Git je moderný verziový systém, ktorý sa stal štandardom v oblasti správy zdrojového kódu a dokumentov. Jeho hlavnou úlohou je uchovávať a sledovať zmeny v textových súboroch, čo je obzvlášť užitočné pre programátorov a vývojárov, ktorí pracujú na komplexných projektoch.

4.9 Modem

Modem slúži na prevod digitálneho signálu na analógový a naopak. Umožňuje hlavne bezdrôtový prenos.

4.10 LoRa

Názov pre bezdrôtovú komunikačnú metódu vznikol z anglického slovného spojenia long range – dlhý dosah. Vyznačuje sa nízkou energetickou náročnosťou, odolnosťou voči rušeniu a veľkými prenosovými vzdialenosťami. Nevýhodou je relatívne nízka priepustnosť dát. Je založená na CSS modulácií.

5 Realizácia

V tejto kapitole sa zameriam na postup práce, prekážky, s ktorými som sa stretol, a poznatky, ktoré som získal počas konštruovania a programovania. Vytvorený kód som priebežne zaznamenával na: https://github.com/romankiss/R-IoT/tree/main/Projects/2024/R-IoT-Data_Hub

1. Postup práce

1. **Plánovanie projektu:** Na začiatku som si stanovil ciele.
2. **Výber komponentov:** Mikrokontrolér, senzory, konektory...
3. **Inštalácia softvérových nástrojov**
4. **Programovanie:** Jazyk C# v nanoFrameworku.
5. **Testovanie a ladenie:** Odstraňovanie chýb a vylepšovanie.

2. Prekážky a riešenia

- **Problémy s inštaláciou softvéru:** Mať počítač schopný spúšťať potrebný SW.
- **Komunikácia s perifériami:** Správna adresácia, pripojovanie, protokoly...

3. Tipy a triky

- **Dokumentácia:** Mať po ruke potrebné manuály a datasheety, vedieť hľadať.
- **Verziovanie kódu:** Nestrať prehľad, alebo celý SW, používať Git.
- **Testovanie v malých krokoch:** Vývoj po malých prehľadných krokoch.

4. Predvedenie funkčného prototypu

Na konci procesu som vytvoril funkčný prototyp, ktorý dokáže zbierať a prenášať údaje o teplote, vlhkosti a tlaku. Prototyp je schopný komunikovať s inými zariadeniami a spracovávať dáta v reálnom čase. Všetky komponenty sú umiestnené v ochrannom puzdre.

5.1 Inštalácia softvérových nástrojov

Pred samotným začatím práce s mikrokontrolérmi programovanými v C# nanoFrameworku treba stiahnuť a nainštalovať potrebné softvérové nástroje a aplikácie. Podrobný návod na ich inštaláciu je dostupný v prílohách, alebo online. Bude treba používať:

- vývojové prostredie Visual studio Community edition 2022
- Microsoft Visual studio Build tools
- .NET framework
- .NET nanoFramework
- nanoff firmware flasher – napáľovač firmwéru
- Serial monitor 2 – konzola pre sériové rozhrania
- Windows PowerShell – konzolu som použil na debugovanie
- COM port reader

Nástroje sú v tomto prípade platformovo obmedzené na operačný systém Microsoft Windows. Odporúčaný je Windows 11, kvôli blížiacemu sa koncu podpory firmy Microsoft pre operačný systém Windows 10.

5.2 Trieda Blink

LED disponujúca 3 farbami a ich odtieňmi je užitočná pre potreby signalizácie, či indikácie stavov. Priebeh, úspešnosť alebo neúspešnosť operácie zakódované do farebného signálu sprostredkovaného LED je spoľahlivou alternatívou v prípadoch, keď nie je dostupná komunikácia cez bezdrôtové technológie. Vďaka indikácií možno podať chybové hlásenia a vyžiadať si zásah obsluhy, ako napríklad reštart zariadenia.

Je uložená v zvlášť súbore, oddelene od hlavného programu, kvôli lepšej prehľadnosti. Použitelnosť tejto triedy z hlavného programu zo súboru Blink.cs zabezpečuje rovnaký „menový priestor“, „namespace“.

Zdrojový kód možno nájsť na adrese: <https://github.com/romankiss/R-IoT/blob/main/Projects/2024/R-IoT-Data Hub/Solutions/Basic/Dummy LoRa Node/Blink.cs>

Obrázok 1 AtomS3Lite s rozsvietenou LED



Po vytvorení objektu a inicializácii možno vyvolať funkciu Blinks, ktorá má niekoľko parametrov. Ich význam, jednotky a dátové typy sú rozpísané v tabuľke nižšie.

Tabuľka 1 Parametre funkcie Blinks

Parameter	Jednotka	Dátový typ	Poznámka
r	Parameter červenej farby	byte	Nadobúda hodnoty od 0 do 255, vrátane
g	Parameter zelenej farby	byte	Nadobúda hodnoty od 0 do 255, vrátane
b	Parameter modrej farby	byte	Nadobúda hodnoty od 0 do 255, vrátane
periodIntMs	ms	int	Periódna blikania
duty	%/100	double	Podiel zapnutého stavu v perióde (0,5 = 50% periódy zapnúť LED)
times	-	int	Počet opakovaní

5.3 Hlavný Program.cs

Hlavný program mojej aplikácie je uložený na adrese https://github.com/romankiss/R-IoT/tree/main/Projects/2024/R-IoT-Data_Hub/Solutions/Basic/sht40_temp%26hum_sensor/E22modem_Implementation.

Na úplnom začiatku sa nachádzajú použité knižnice/nuggety. Nasleduje inicializácia objektov z tried, definícia dôležitých premenných, započatie komunikácie s LoRa modemom a senzorom. Na základe periódy odosielania sú zmerané potrebné údaje a odoslané cez LoRa. Program obsahuje funkciu „FireTelemetryData“, ktorá je pravidelne vyvolaná časovačom. Na správne fungovanie celého riešenia je potrebné do súborovej štruktúry projektu vložiť aj už vyššie spomenutú triedu Blink, E22Wrapper, súbor na interakciu s modemom a Extensions obsahujúci funkcie na prácu s dátami a výpočty. Tieto súbory sú dostupné na adrese: <https://github.com/romankiss/R-IoT/tree/main/Projects/2024/PatternsAndPractices/Lora/E22-900Txxx/Driver>. Program by mal fungovať v celku spoľahlivo, keďže obsahuje množstvo overovania, či sa predchádzajúce úkony vykonali správne. Treba dbať na správne nastavenie najmä pinov, adries, COM portov.

5.4 Senzor teploty a vlhkosti SHT40

Snímač teploty a vlhkosti od firmy SENSIRION je veľmi dobrou voľbou na zbieranie teploty i relatívnej vlhkosti. Technické špecifikácie a spôsob práce s týmto snímačom sú obsiahnuté v tomto dokumente: [SHT40.pdf](#) Snímač je v jednom púzdre spolu so snímačom atmosférického tlaku [BMP280](#). Toto poskytuje firma M5stack: https://shop.m5stack.com/products/env-iv-unit-with-temperature-humidity-air-pressure-sensor-sht40-bmp280?srltid=AfmBOoq7ihtFUtG8C6rnoG9gZm2vprdubArSbj4_wYeY384tutD48a2a

Komunikuje cez I2C zbernicu. Fyzicky sa dátovo pripája cez Grove konektor a uchytáva pomocou dierok zapadajúcich do LEGO stavebnicového systému.

Obrázok 2 senzor ENV IV



5.5 Modul GPS

Podporuje systémy GPS, Galileo, Glonass, BD. Na frekvencií 1Hz posiela informácie o polohe cez formát GNGGA. Pracuje na baudrate 115200 a medzi jeho nevýhody patrí dlhé/neúspešné hľadanie satelitného spojenia, najmä vo vnútorných priestoroch.

Obrázok 3 GPS modem



5.6 Snímač vzdialenosti ToF

Po konfigurácii komunikuje po I2C zbernici, verzia „D“ má dosah až 50m, žiaľ cena sa pohybuje až okolo 30€. [https://www.waveshare.com/wiki/TOF_Laser_Range_Sensor_\(D\)](https://www.waveshare.com/wiki/TOF_Laser_Range_Sensor_(D))

5.7 Mikrokontrolér AtomS3 Lite

Firma M5stack vo svojej rade Atom vydala množstvo mikrokontrolérov v rovnakom form-faktore, aby mohli byť vzájomne nahraditeľné. Vo svojej práci som využil AtomS3 Lite disponujúci väčším výpočtovým výkonom oproti Atom Lite s pomocou ktorého som sa zoznamoval s programovaním v C# .NET nanoFrameworku. Mikrokontrolér ESP32, ktorý je jadrom tohto zariadenia je veľmi populárny vo vzdelávacích i profesionálnych aplikáciách. Pracuje na frekvencii do 240MHz s dvojjadrovým procesorom Xtensa 32-bit LX7.

Tabuľka 1 vlastnosti AtomS3Lite

Vlastnosť	Hodnota
MCU	Xtensa 32-bit LX7
FLASH pamäť	8MB
RGB LED	WS2812C-2020
Prevádzková teplota	0°C ~ 40°C
Zdrojové napätie	5V
Rozmery	24mm × 24mm × 9.5mm
Hmotnosť	5.3g

Obrázok 4 AtomS3Lite - pohľad na zadnú stranu



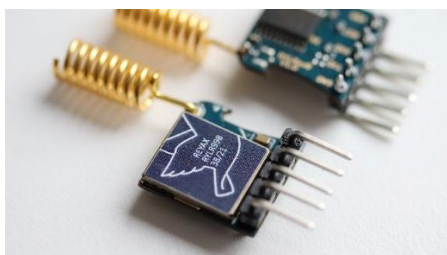
Veľmi prakticky bol vymyslený zadný štítok, na ktorom sa dá rýchlo zistiť číslo potrebného GPIO pinu. Táto drobnosť mi ušetrila čas tým, že som nemusel hľadať tieto podstatné údaje v dokumentácii k tomuto výrobku, ale stačilo mi sa pozrieť na jeho zadnú stranu.

5.8 LoRa modem RYLR998

Začiatočnícky prijateľná možnosť na LoRa prenos. Avšak kvôli potrebe dlhšieho dosahu som vo svojom riešení nakoniec použil iný modem. Postup práce je veľmi podobný, ako s modemom E22, až na pár odlišností, ako spôsob komunikácie s PC. V prípadoch, kde by dosah signálu do 1km bol postačujúci je RYLR998 asi najlepšia voľba. Medzi jeho hlavné prednosti patria napríklad:

- Malé rozmery
- Nízka hmotnosť
- Štandardná komunikácia cez AT príkazy

Obrázok 5 modem RYLR998



Doska plošných spojov má ikonický tvar obrysu vtáka, vďaka čomu si vyslúžil prezývku „birds“.

Tento výrobok možno konfigurovať a priebežne nastavovať. Tieto nastavenia si zariadenie ukladá do vnútornej pamäte a netreba ich tým pádom zadávať po každom reštarte. Opis príkazov je obsiahnutý v dokumente poskytovanom výrobcom na adrese: https://reyax.com/upload/products_download/download_file/LoRa_AT_Command_RYLR998_RYLR498_EN.pdf.

Odoslanie balíčku dát je možné až do veľkosti 240 bajtov. V príkaze musíme špecifikovať na akú adresu chceme daný balík poslať. Musíme zadať aj dĺžku dátového balíku v bajtoch, balíky presahujúce maximálnu povolenú veľkosť budú zahadzované a zariadenie odpovie chybovou hláškou. Samotné dáta sú v ASCII formáte. Treba zdôrazniť, že zariadenia, ktoré chcú spolu komunikovať musia byť nielen v rovnakej sieti, ale mať rovnaké aj iné dôležité parametre, ako šírka pásma, frekvencia, kódovací pomer... Ak nechceme poslať balík

len jednému zariadeniu, môžeme ho poslať na tzv. broadcast adresu, čiže 0 a obdržia ho všetky zariadenia v tej istej sieti a s rovnakými parametrami.

Ak chceme vyčítať zo zariadenia prijatý balík musíme vyčítať výpis v sériovom porte. Spôsoby vyčítavania zo sériového portu sa líšia od nadradeného zariadenia a jeho programového vybavenia (PC-Windows-PS, ESP32-nFramework-knižnica, smartfón-android-SerialMonitor...). Modul si uchováva v pamäti určitý počet balíkov, pri čítaní sa vyčíta ten najstarší a vyhodí sa.

Vďaka svojej komunikačnej metóde cez UART rozhranie a AT príkazy je možné používať tento modem aj v priamom spojení s napríklad notebookom. Komunikáciu medzi notebookom a modemom som riadil cez skripty v programe PowerShell ISE. Jednoduchý kus kódu je spustiteľný cez funkciu „Spustiť aktuálny výber“ – „Run current selection“. Zdrojový kód možno nájsť na adrese: <https://github.com/romankiss/R-IoT/blob/main/Projects/2024/PatternsAndPractices/Lora/Sender.ps1>.

Obrázok 6 Prostredie Powershell ISE

```

Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
LoRa_Receiver.ps1 X Run Selection (F8)

7 $port.ReadTimeout = 5000
8 $port.Open()
9 $port.WriteLine("AT" + "`r`n")
10 $port.ReadLine()
11 #Attention! moving-disconnecting the contact on the pins will force the device to reset - can be seen when there are no commands send, but the blue led on the adapter blinks. After e
12 $port.WriteLine("AT+RESET" + "`r`n")
13 #after the reset command, there must be 2 reads, to get the full 2 responses, else later in the program the read will be reading the second newest, not the newest line from the buff
14 $port.ReadLine()
15 $port.ReadLine()
16 # $port.WriteLine("AT+BAND=868500000,W" + "`r`n")
17 # $port.ReadLine()
18 # $port.WriteLine("AT+ADDRESS=2000" + "`r`n")
19 # $port.ReadLine()
20
21 do {
22     while ($port.IsOpen)
23     {
24         $port.Close()
25         $port.Dispose()
26     }
27 }
28
29 $port.WriteLine("AT+SEND=0,8,loopback" + "`r`n")
30 $port.ReadLine()
31
32 $port.WriteLine("AT+SEND=0,5,HELLO" + "`r`n")
33 $port.ReadLine()
34
35 $port.WriteLine("AT+SEND=0,20,{"Temp":21,"Hum":50} + "`r`n")
36 $port.ReadLine()
37
38 # $port.ReadLine()
39 # $port.WriteLine("AT+ADDRESS=2000" + "`r`n")
40 # $port.ReadLine()
41
42 Device should be connected at port: COM6
43 7+OK
44 +RESET
45 +READY
46
47 PS C:\Users\vicko> $port.WriteLine("AT+SEND=0,5,HELLO" + "`r`n")
48 $port.ReadLine()
49 +OK
50
51 PS C:\Users\vicko>
52
Completed Ln 27 Col 20 90%

```

Treba však dbať na dodržanie povolených napäťových rozsahov, aby sa predišlo zničeniu zariadenia. Použil som preto USB adaptér na vývody so zabudovaným meničom napätia z 5V používaných v USB notebooku na 3,3V povolených pre modem..

5.9 LoRa modem E22-900T22D

Tento modem som napokon použil vo svojej aplikácii, pre jeho väčší dosah. Pochopenie správneho postupu práce so zariadením mi zabralo relatívne najviac času. Oproti RYLR998 má komunikáciu s nadradenou jednotkou riešenú cez bajtové kódy posielané cez UART. So sprevádzkovaním bolo veľa problémov a preto som sa spolupodieľal na písaní akéhosi návodu, ktorý je prílohou s obdobným názvom. Konfiguráciu si možno uľahčiť využitím GUI aplikácie – odkaz na jej stiahnutie je v prílohe. Keďže ide tiež o LoRa modem, nutnosť konfigurácie oboch staníc na rovnaké parametre atď. je podobný s RYLR998, avšak treba podotknúť, že tieto dva modemy od rôznych značiek nie sú schopné spolu komunikovať – príjemca a odosielateľ musia byť spolu kompatibilný.

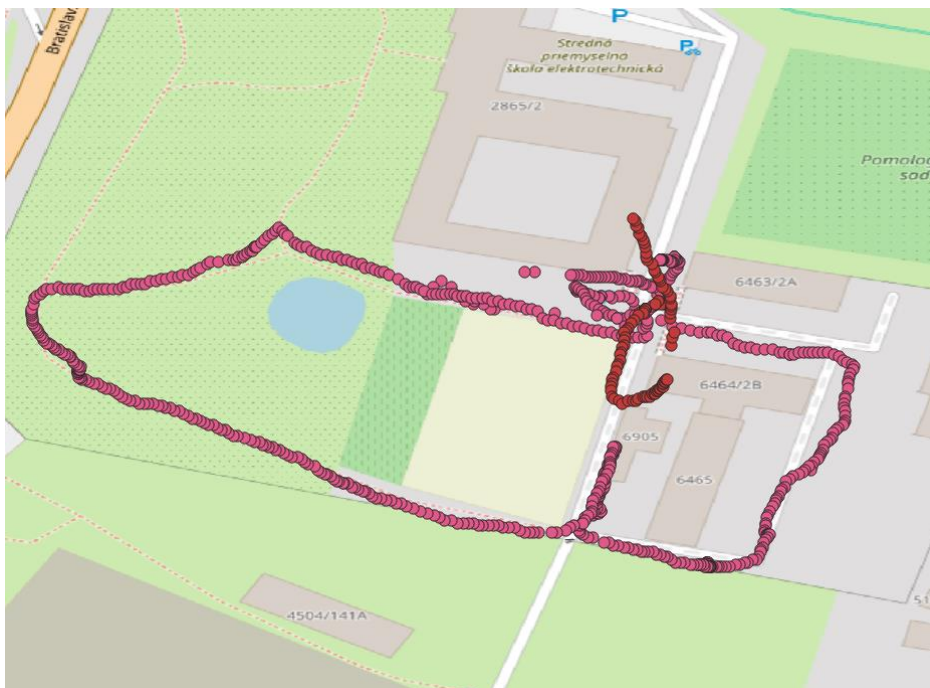
Doska plošných spojov má vyvedená konektor na anténu, na ktorý je potreba pripojiť podľa nastavení minimálne 22dBm anténu s požadovaným frekvenčným rozsahom. Vyskúšal som viaceré antény a zostal u tej s najmenšími rozmermi, čo môže mať za následok zmenšenie dosahu. Modemu treba určiť, či má pracovať v konfiguračnom, alebo prenosovom režime privedením požadovaných napäťových hodnôt na piny M0 a M1. Toto je vo finálnom prototypu riešené drobným spínačom. Jednou z možností bolo priletovať modem na prototypovaciu platformu <https://shop.m5stack.com/products/mini-proto-board-unit> disponujúcu konektorom typu Grove cez ktorý sú vyvedené piny Rx a Tx a samozrejme napájanie. Na prototypovanie je pripojenie cez LEGO diery unifikované aj so senzorom najvhodnejšie.

Dosah bol overený testom pri letisku Piešťany, kde sú ideálne podmienky – holá lúka bez tlmivých faktorov, ako je napr. zástavba.

5.10 Prijímacia stanica E22-900T22U a COM port reader

Na príjem dát zo zariadenia do počítača som využil kompaktný modem s jednoduchým pripojením USB a režimovým tlačidlom. Jedná sa v podstate o rovnaký modem, ako E22-900T22D, len v šikovnom obale s tlačidlom. Z počiatku som prijaté dáta vyčítaval pomocou PowerShell skriptov. Pri skúšaní dosahu som prišiel na to, aké je to nepraktické a rozhodol sa vyvinúť grafické rozhranie so stabilnou funkcionalitou. Musím priznať, že som vo veľkej miere využil umelú inteligenciu DeepSeek, ktorá mi zadanú úlohu splnila ukážkovo. Špecifické detaily som potom už doplnil manuálne. Rozhranie zobrazené nižšie ponúka rýchli výber portu, baudratu, a formátu prezentácie dát – hexadecimálne bajty, text, alebo ak sa jedná o dáta teploty a vlhkosti v stanovenom formáte, tak aj živé vykresľovanie grafu.

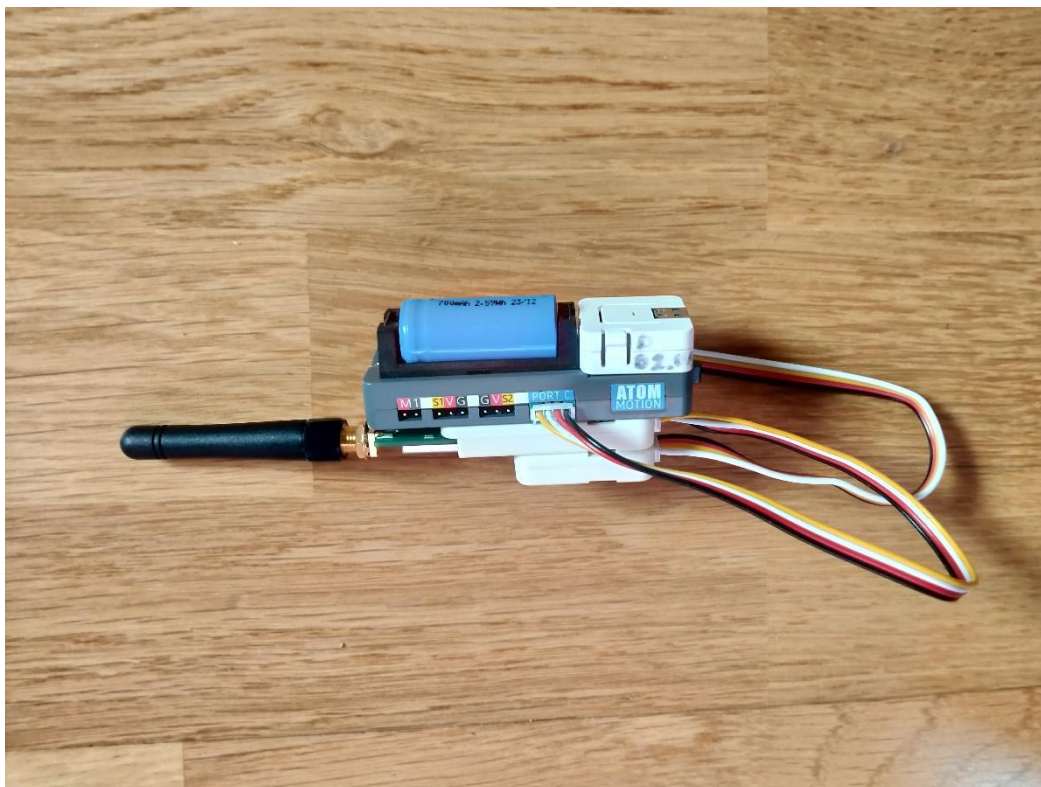
Obrázok 7 Grafická vizualizácia GPS súradníc



5.11 Kompletizácia

Ako zdroj záložného napájania z batérie som použil základňu Atomic Motion Base, taktiež od M5stack: <https://shop.m5stack.com/products/atomic-motion-base-stm32f030?variant=44393917219073>. Zariadenie tým pádom dokáže pracovať nejakú dobu aj bez stáleho napájania zo siete. Základňa disponuje aj dvoma Grove konektormi, ktoré sa dajú využiť na pripájanie senzorov a modemu. Zvažoval som, či celé zariadenie umiestniť do elektroinštalačnej krabice. Tento variant by zvýšil odolnosť, pred poveternostnými vplyvmi a mechanickým poškodením, avšak by výrazne zväčšil rozmery. Nakoniec som sa rozhodol pre jednoduché a kompaktné spojenie do tzv. „sandwicha“ a komponenty naskladal na seba s pomocou LEGO spojok ako vidno nižšie:

Obrázok 8 Finálne zariadenie



1 Záver

Podľa tohto návrhu bol vytvorený funkčný prototyp zariadenia internetu vecí. Počas práce na tomto projekte som si obohatil vedomosti a nadobudol mnohé skúsenosti z programovania a riešenia problémov spojených s vývojom aplikácie a prototypu.

Nezanedbateľnú pomoc s výberom hardvéru a cenné rady a nápovedi pri písaní a používaní softvéru mi poskytol pán Ing. Roman Kišš. Pán učiteľ Ing. Jozef Dragula sprostredkoval stretnutia, koordinoval výučbu, smeroval moju prácu k úspešnému záveru.

Oblasť Internetu vecí je pre mňa velice zaujímavá a perspektívne sa v nej budem ďalej zlepšovať. Či prinesie sľubované benefity v dostatočnej miere ukáže až čas. Je to technológia, ktorá sa zo dňa na deň vyvíja. To čo v nej platí dnes, nemusí platiť zajtra. Ktokoľvek by sa chcel udržať ako-tak v obraze musí si neustále aktualizovať prehľad a vedomosti.

6 Zoznam zdrojov

1. <https://shop.m5stack.com/>
2. <https://docs.m5stack.com/en/unit/ENV%E2%85%A3%20Unit>
3. <https://shop.m5stack.com/products/atomic-motion-base-stm32f030?variant=44393917219073>
4. <https://www.cdebyte.com/products/E22-900T22D>
5. <https://github.com/romankiss/R-IoT/tree/main/Projects/>
6. <https://www.itnetwork.sk/windows/powershell/powershell-pre-zaciatocnikov-lekcie-1>

7 Zoznam príloh

1. E22-900T22D_UserManual_EN_v1.3.pdf
2. E22 návod – dokumentácia.docx
3. LoRa_AT_Command_RYLR998_RYLR498_EN.pdf
4. SHT40.pdf
5. visualstudio-vs-2022.pdf