

**STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
ZOCHOVA 9, BRATISLAVA**



SONAR – ultrasonické mapovanie dna a 3D interpretácia

projekt z praktickej časti odbornej zložky maturitnej skúšky

Meno kandidáta: Kristián Solovic

Trieda: IV.B

Vedúci práce: Mgr. Jakub Krcho

Šk. rok: 2023/2024

**STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
ZOCHOVA 9, BRATISLAVA**

SONAR – ultrasonické mapovanie dna a 3D interpretácia

Meno kandidáta: Kristián Solovic

Trieda: IV.B

Vedúci práce: Mgr. Jakub Krcho

Rozsah: 25 strán, 7 obrázkov, 0 tabuliek, 1 príloha

Odovzdané:

Podpis vedúceho práce:

STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ ZOCHOVA 9, 811 03 BRATISLAVA



Študijný odbor: 2573 M 00 - Programovanie digitálnych technológií
Meno: Kristián Solovic
Trieda: IV.B
Školský rok: 2023/2024

Zadanie praktickej časti odbornej zložky maturitnej skúšky formou obhajoby vlastného projektu

Názov práce: Sonar - ultrasonické mapovanie dna a 3D interpretácia

Úloha:

Navrhnete, zostrojíte a naprogramujete zariadenie na zaznamenávanie hĺbky jazera a korešpondujúcej GPS lokácie. Hĺbka a lokácia sa bude ukladať na SD kartu. Napíšete skript na interpretáciu získaných dát a zostrojíte 3d hĺbkového grafu jazera.

Technické parametre:

- programovanie C++
- programovanie Python
- skenovanie hĺbky pomocou senzoru JSN-SR04T
- lokácia pomocou modulu NEO-7M

Písomná časť:

Odovzdať vo viazanej forme (hrebeňová, termoväzba, lepená...) formátu A4 s počtom strán min. 20, bude obsahovať:

- zadanie praktickej maturitnej práce - originál
- čestné prehlásenie
- poďakovanie (nie je povinné)
- zoznam obrázkov, tabuliek
- obsah
- úvod
- popis použitých technológií (max. na 5 strán) - teoretická časť
- popis technického riešenia - praktická časť
- použitá literatúra podľa normy ISO 690
- záver
- prílohy

V Bratislave, dňa 23.02.2024

Stredná priemyselná škola
elektrotechnická
811 03 Bratislava, Zochova 9


Ing. Marián Beniak, PhD.
riaditeľ

Čestné prehlásenie

Čestne prehlasujem, že problematiku týkajúcu sa náplne zadaného projektu v rámci praktickej časti odbornej zložky maturitnej skúšky formou obhajoby som spracoval sám za pomoci uvedenej použitej literatúry. Inú, ako uvedenú literatúru, som nepoužil.

V Bratislave dňa:

Kristián Solovic

OBSAH

Úvod.....	1
Teoretická časť	2
1 O technológiách	2
1.1 GPS	2
1.2 Sonar	3
2 Hardvér	6
2.1 Komponenty.....	6
3 Softvér.....	8
3.1 Python	8
3.2 Knižnice	8
Praktická časť	10
4 Zapojenie.....	10
5 Komunikácia	12
6 USB štruktúra.....	13
7 Sonar	14
8 Graf	17
8.1 Validácia dát	18
8.2 Testovanie	18
8.3 Vykresľovanie grafu.....	19
8.4 Analýza grafu	21
9 Alternatívne aplikácie	22
10 Problémy	23
Záver	25

ZOZNAM OBRÁZKOV

Obrázok 1 Technický náčrt sonarových vln.....	4
Obrázok 2 Hardvér sonaru	7
Obrázok 3 Diagram zapojenia modulov	10
Obrázok 4 Označenie pinov RPI modelu 2B	11
Obrázok 5 funkciou scatter rozmiestnené body	19
Obrázok 6 na body je aplikovaná farebná vrstva.....	20
Obrázok 7 Graf vykreslený z testovacích dát	21

ZOZNAM SKRATIEK

Značka	Veličina	Jednotka
v	Rýchlosť	m/s
l	Dĺžka	m
mAh	Kapacita batérie	mAh

Skratka	Popis
MP	Maturitná práca
GPS	Global Positioning System
IP	Internet Protocol
SCP	Secure Copy Protocol
RPI	Raspberry pi
GE	Google Earth
LED	Light Emitting Diode
PIP	Package Installer for Python
NMEA	National Marine Electronics Association

Úvod

Ako príležitostného rybára ma už od ranného detstva zaujímalo čo sa deje pod vodnou hladinou. Pri prvom pohľade vidíme jednoliaty, trblietajúci sa povrch, netrvá ale dlho a začneme uvažovať čo všetko sa tam dole môže skrývať. Táto fascinácia s neznámom, ktorá nepochybne naplňa nespočetné množstvo mladých ľudí po celom svete, ma prenasledovala až sem. Na svetových mapách sú biele miesta už dávno vyplnené, kontinenty sú objavené a dopodrobna zmapované.

Pomocou Google Earth máme možnosť ocitnúť sa na takmer ktoromkoľvek mieste na našej planéte. Takmer. Na mapách sa ale stále, dokonca aj v dnešnej dobe technologického pokroku, nachádzajú „modré miesta“ a na tie sa budem v tejto maturitnej práci sústrediť my.

Cieľom mojej maturitnej práce (MP) je teda vytvoriť zariadenie, ktoré zabezpečí ak nie úplné odhalenie, tak aspoň priblíženie tohto podvodného sveta. Vo svojom vlastnom poetickom zmysle sa to v niektorých stránkach podobá objavovaniu novej krajiny. Po pár hodinách jazdenia po vodnej ploche na člne bude možné si pomocou nášho zariadenia a programu vykresliť podobu dna, ktoré sa ľudským očiam až do teraz úspešne skrývalo, či už pod klenbou mútnej vody alebo priveľkej hĺbky. A my budeme mať možnosť pozrieť sa ako to tam dole približne vyzerá.

Projekt sa skladá z dvoch častí: prenosného zariadenia, ktoré zobrazuje aktuálnu hĺbku vo vodnom stĺpci a zaznamená ju spolu s presnými súradnicami, a z programu, ktorý z týchto dát vykreslí 3D graf.

V rámci dokumentácie čitateľa oboznámime s použitými technológiami a ich limitáciami. Predstavíme spôsob, ktorým je všetko poskladané dokopy a ako to všetko funguje.

Teoretická časť

V tejto časti si predstavíme technológie použité pri skladaní a programovaní tohto projektu, ako aj ich limitácie a dopad na okolie.

1 O technológiách

Nasledujúce technológie zohrali kľúčovú rolu pri konštruovaní tejto MP. Predstavíme si ich fundamentálny spôsob fungovania a krátku históriu.

1.1 GPS

Akronym pre Global Positioning System, GPS je základom moderných navigačných služieb. Skladá sa z konštelácie satelitov obiehajúcich našu planétu po presne stanovených trasách v približne 20 000 kilometrovej výške. Každý z týchto satelitov nepretržite vysiela signály obsahujúce ich lokáciu a presný čas z atómových hodín na palube. Ak sme schopní prijímať signály z troch alebo viacerých satelitov súčasne, môžeme pomocou komplexných rovníc vypočítať našu polohu s presnosťou niekoľkých metrov.

GPS sa stala nenahraditeľnou súčasťou mnohých odvetví, vrátane dopravy, logistiky alebo geodézie. Uľahčuje každodennú navigáciu v mobile ale zároveň sa využíva aj na navigovanie vesmírnych lodí.

Napriek svojej rozšírenosti a spoľahlivosti nie je bez obmedzení, rôzne prekážky ako sú vysoké budovy či stromy môžu signál oslabovať alebo narúšať. V tejto aplikácii by ale malo byť rušenie minimálne keďže sa budeme väčšinu času nachádzať v strede vodnej plochy. [2]

1.1.1 Vplyv GPS na životné prostredie

GPS satelity vysielajú slabé elektromagnetické žiarenie. Nie je vedecky dokázané že by škodili alebo narúšali procesy v prírode. Ak by aj tieto vlny nejaké negatívne efekty spôsobovali, náš projekt by k nim neprispieval, pretože použitý modul tieto vlny iba zachytáva, nevysiela nič.

1.2 Sonar

Ultrazvukové meranie vzdialenosti a sonarové technológie revolučne ovplyvnili rôzne oblasti, vrátane navigácie, robotiky a prieskumu pod vodou.

Počiatok ultrazvukového merania vzdialenosti možno vystopovať do začiatku 20. storočia. V roku 1912 objavil francúzsky fyzik Paul Langevin spolu so svojimi kolegami piezoelektrický efekt, pri ktorom určité materiály generujú elektrický náboj, keď sú podrobené mechanickému stresu. Tento objav položil základy pre vývoj ultrazvukových meničov, zariadení schopných meniť elektrickú energiu na ultrazvukové vlny.

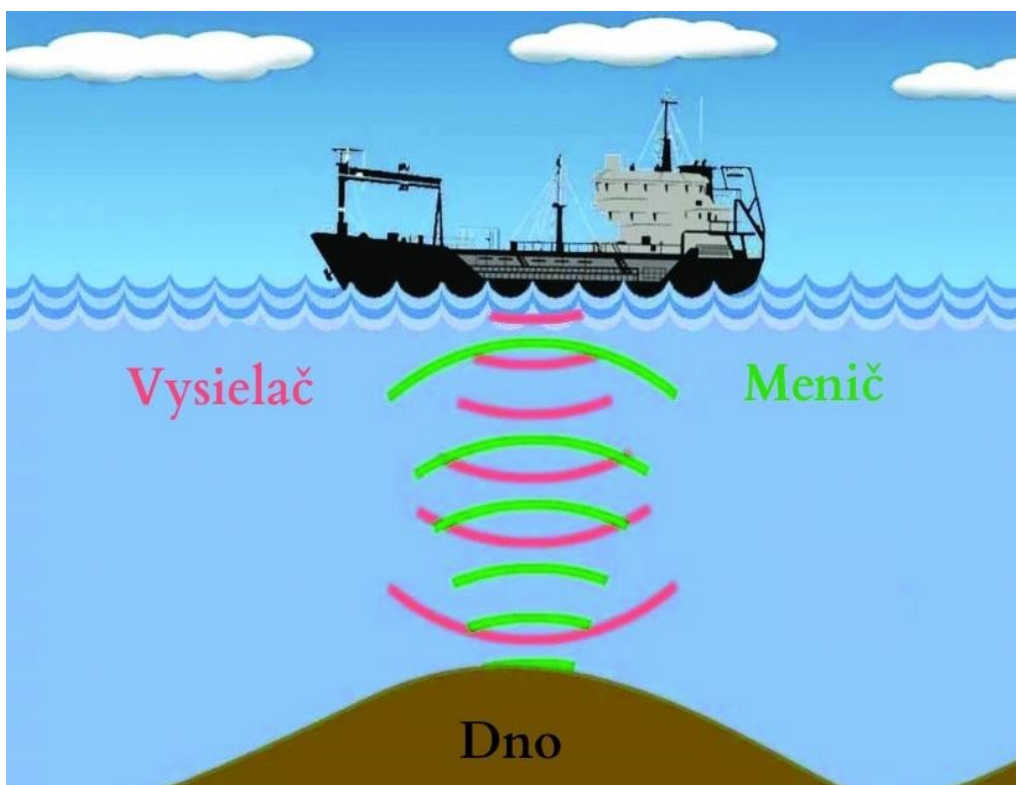
V 40. rokoch výskumníci ako napríklad americkí fyzici Robert H. Bolt a Floyd Dunn ďalej posunuli technológiu ultrazvuku. Skúmali aplikácie ultrazvukových vln na meranie vzdialenosti a zobrazovanie nameraných dát, čím vyšliapali cestu pre použitie ultrazvukových senzorov v rôznych odvetviach.

Sonar, akronym pre Sound Navigation and Ranging, sa stal kľúčovou technológiou počas prvej svetovej vojny. Potreba detegovať ponorky vyvolala rozsiahly výskum podmorskej akustiky. Britský fyzik a námorný dôstojník Lewis Fry Richardson je často označovaný za autora konceptu využívania zvukových vln na podmorskú detekciu už v roku 1916. Avšak až počas druhej svetovej vojny dosiahla sonarová technológia významný pokrok.

Kanadský inžinier Reginald A. Fessenden už v roku 1915 patentoval systém echolokácie na detekciu ľadovcov, čím preukázal potenciál využitia zvukových vln na podmorskú navigáciu. V nasledujúcich desaťročiach prispeli vedci a inžinieri, k zdokonaľovaniu sonarových systémov pre vojenské aj civilné aplikácie.

Po druhej svetovej vojne dochádzalo k rýchlemu rozšíreniu ultrazvukových a sonarových technológií do civilných a komerčných oblastí. Ultrazvukové senzory našli uplatnenie v priemyselnej automatizácii, medicínskom zobrazovaní a automobilových systémoch. Neustále inovácie v technológii ultrazvukových meničov, spracovaní signálu a zmenšovaní veľkosti viedli k vytvoreniu vysokej presnosti a kompaktných ultrazvukových meradiel vzdialenosti. [9]

Dnes sú ultrazvukové senzory neoddeliteľnou súčasťou robotiky, parkovacích asistenčných systémov v automobiloch a rôznych spotrebiteľských elektronických zariadení. Sonarová technológia sa stále využíva hlavne pri prieskume podvodných oblastí, správe rybníkov a v oceánografickom výskume.



Obrázok 1 Technický náčrt sonarových vln [5]

Základným princípom zostáva stále meranie času za ktorý sa vlna z vysielača odrazí od prekážky a je zaznamenaná meničom. V moderných echolokačných zariadeniach sa navrátená vlna ďalej analyzuje a podľa sily rušenia sa dá zistiť detailná podoba dna. V tomto projekte ale pracujeme s najjednoduchšou formou tohto zariadenia a preto ho budeme využívať iba na meranie jedného bodu a návratnú vlnu po zistení vzdialenosti ďalej analyzovať nebudeme.

1.2.1 Vplyv sonaru na životné prostredie a vodné živočíchy.

Je pre mňa veľmi dôležité aby náš projekt nemal žiadne negatívne následky na faunu a flóru meraných vôd. Intenzívne zvukové vlny môžu znemožňovať komunikáciu, poškodzovať sluchové orgány alebo odpudzovať zvieratá od miesta kde sa používajú.

Keďže to ale platí skôr pri extrémnych prípadoch a náš vysielateľ je jeden z najslabších aké sa dajú získať, následky na životné prostredie sú zanedbateľné, až neexistujúce. Na základe vlastného testovania sme nenašli žiaden náznak toho že by vodným živočíchom vadili zvukové vlny vysielané sonarom, dokonca ani z bezprostrednej blízkosti. Rybky v akváriu to jednoducho ignorovali.

2 Hardvér

Všetky fyzicky hmatateľné časti tejto MP. Vysvetlenie komponentov a ich úlohy v tomto projekte.

2.1 Komponenty

Na realizáciu tohto projektu boli použité nasledujúce komponenty.

2.1.1 Raspberry Pi 2B

Jednodoskový mikropočítač, v tomto prípade mozog nášho meradla. Má na starosti ovládanie všetkých komponentov zariadenia ako aj zapisovanie hodnôt.

Operačným systémom tohto mikropočítača je Raspberry pi OS. Tento operačný systém je odvodený od Debianu, slobodnej distribúcie Linuxu. Síce je možné pracovať aj v GUI rozhraní, je na to ale potrebné RPI pripojiť k monitoru a zapojiť doňho klávesnicu a myš. My v tomto projekte komunikáciu vyriešime kompletne cez cli príkazový riadok Linuxu pomocou SSH protokolu. [1]

2.1.2 JSN-SR04T

Modul pomocou ktorého sme schopný merať hĺbku. Vysiela ultrazvukové vlny a počíta za aký čas sa mu odrazené vlny vrátia. Výrobca udáva maximálnu vzdialenosť merania 650 cm a minimálnu 20 cm.

Tento modul je síce presný na centimetre, to však ale z veľkej časti záleží na podmienkach vo vode. Zvuk sa v mútnej vode šíri pomalšie ako v krištáľovo čistej. V našej aplikácii zariadenia je absolútne presná hĺbka nepodstatná, dôležitejšia je proporcia, pri najhorších podmienkach bude celá mapa posunutá o pár decimetrov hore alebo dole. [4]

2.1.3 GPS NEO-7M

Je zodpovedný za zachytávanie signálu zo satelitov ktorých trianguláciou je schopný získať svoju polohu. Modul má udávanú presnosť do 2 m. To je v tomto prípade postačujúce, po preškáľovaní dát na veľkosť obrazovky malé nepresnosti nebude vidno. [8]

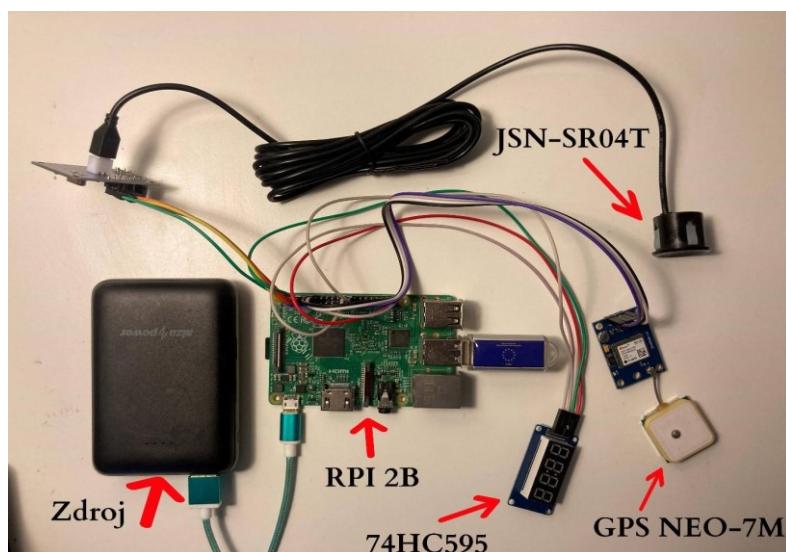
2.1.4 74HC595 - Segmentovaný displej

Dáta ako aktuálna hĺbka a status GPS signálu, v priebehu programu zobrazujeme pomocou tohto modulu.

Displej pozostáva z mnohých segmentov vyrobených z LED svietidiel, ktoré sa pomocou ovládača TM1637 rozsvietia v takej konštelácii aby vytvárali písmeno alebo číslo.

2.1.5 Zdroj

Elektrina prichádzajúca do RPI odkiaľ ďalej prúdi do komponentov, ktoré to práve potrebujú, pochádza z powerbanky s kapacitou 20 000mAh. Toto množstvo energie nám bohato postačí na niekoľko hodín merania.



Obrázok 2 Hardvér sonaru

3 Softvér

Všetky fyzicky nehmatateľné časti tejto MP. V tejto kapitole čitateľa oboznámime s použitým programovacím jazykom a jeho knižnicami.

3.1 Python

Python je objektovo orientovaný, komplexný programovací jazyk. Bol vyvinutý Guidom van Rossumom v Holandsku v roku 1991 ako hobby projekt aby sa Guido cez Vianoce nenudil.

Meno Python nepochádza od hada, ale z britskej show Monty Python's Flying Circus.

Jeho jednoduchá syntax a intuitívna integrácia z neho robia jeden z najpopulárnejších programovacích jazykov. Vďaka jeho popularite vzniklo mnoho knižníc, ktoré značne uľahčujú programovanie. Keďže pythonové knižnice existujú takmer na všetky účely, developerom často stačí iba ich správne pospájať. Všetky metódy sa dajú jednoducho dohľadať na internete, alebo priamo v stiahnutej knižnici. [3]

Poslúži nám ako hlavný programovací jazyk. Celý algoritmus je napísaný práve v ňom. Vďaka programovaciemu jazyku dokážeme udeliť zariadeniu presné inštrukcie na prácu s modulmi a zabezpečíme správny výpis nameraných dát. V spolupráci s knižnicami dáta spracuje do grafu.

Na inštalovanie knižníc používame PIP. PIP je správca balíčkov pre programovacie moduly pythonu. Jeho úlohou je inštalácia oficiálnych knižníc z Python Package Indexu alebo neoficiálnych z iných zdrojov.

Napríklad knižnicu Matplotlib nainštalujeme v príkazovom riadku pomocou príkazu:

```
pip install matplotlib
```

3.2 Knižnice

Nasledujúce knižnice sme využili pri programovaní tohto projektu.

3.2.1 Numpy

Je fundamentálnou knižnicou na prácu s číslami. Podporuje mnohodomenzionálne polia a matice spolu s širokou kolekciou funkcií na vykonávanie najrôznejších matematických operácií medzi nimi.

Rozdiel medzi numpy polom a pythonovým listom spočíva hlavne v konzistencii. Pythonový list obsahuje elementy rôznych typov zatiaľ čo pole Numpy umožňuje pracovať iba s jedným a je špecificky prispôsobená na pracovanie s numerickými hodnotami čo umožňuje efektívnejšie vykonávanie operácií.

Vďaka rovnomernejšej reprezentácii v pamäti sú numpy polia značne rýchlejšie, čo je veľkou výhodou pri načítavaní tisícok dátových bodov ktoré sú v priebehu merania zaznamenané. Zároveň sú na jej základe postavané mnohé ďalšie knižnice ako je Pandas alebo aj Matplotlib.

3.2.2 Matplotlib

Obsiahla knižnica určená na vytváranie statických, animovaných a interaktívnych vizualizácií.

Ponúka veľké množstvo nástrojov na generovanie rôznych typov grafov. Je jednoducho prispôsobiteľná, umožňuje úpravu všetkých aspektov grafu vrátane farby, ôs, názvov atď... Využíva sa najmä na vizualizáciu rôznorodých dát, v našom prípade interpretuje jednotlivé body hĺbky, zemepisnej dĺžky a zemepisnej šírky.
[6]

3.2.3 TM1637

Knižnica naprogramovaná špecificky na ovládanie segmentovaných displejov s ovládačom typu TM1637. Využívame ju na zobrazovanie aktuálnej hĺbky.

3.2.4 Serial

Knižnica serial je potrebná na monitorovanie portu na ktorý sú zdieľané NMEA vety prichádzajúce z GPS modulu.

3.2.5 Pynmea2

Formátuje NMEA vety do kódovania z ktorého sa dajú potrebné dáta ako napríklad zemepisná šírka/dĺžka spoľahlivo vytiahnuť.

3.2.6 RPL.GPIO

Pomocou tejto knižnice sú v pythone ovládané fyzické porty Raspberry pi.

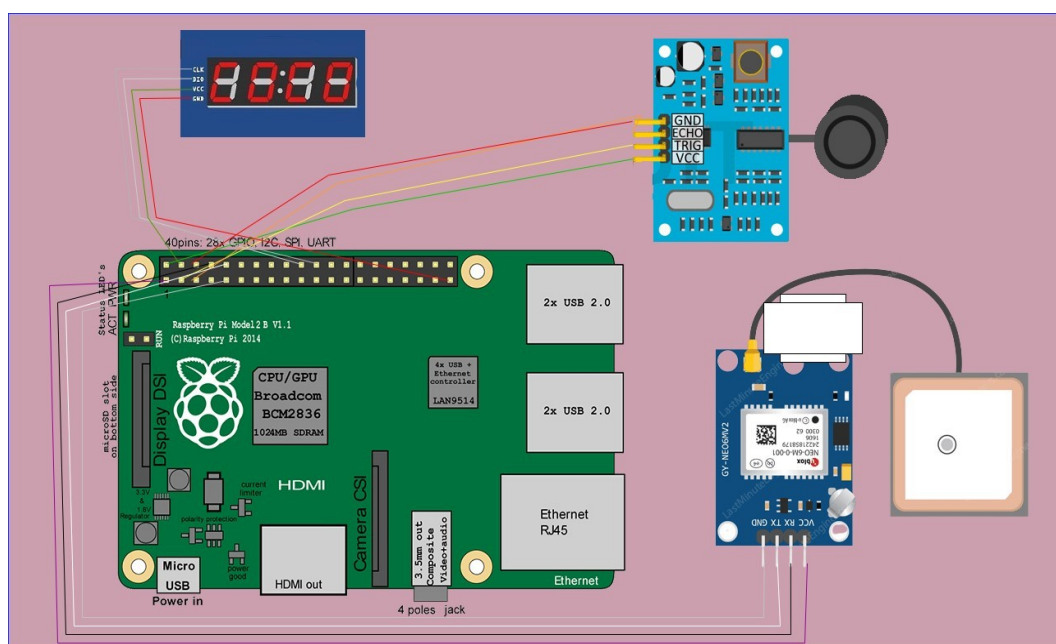
3.2.7 Time

Ako z názvu vyplýva, spravuje všetky potreby týkajúce sa času.

Praktická časť

V tejto kapitole si predstavíme akým spôsobom boli tieto technológie v našom projekte implementované. Oboznámime čitateľa s operáciami pre najpresnejšie meranie, pozrieme sa na najdôležitejšie kúsky kódu a spôsob zapojenia.

4 Zapojenie



Obrázok 3 Diagram zapojenia modulov

Všetky komponenty okrem jedného sú zapojené rovnakým spôsobom: pomocou GPIO pinov, z ktorých im prichádzajú signály buď true alebo false, (1, 0). Modul NEO-7M je pripojený do i2c pinov tak aby mohol namiesto prijímania, vysielat' súradnice.

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I2C)		DC Power 5v	04
05	GPIO03 (SCL1 , I2C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Obrázok 4 Označenie pinov RPI modelu 2B [7]

4.1 Pripevnenie celého prístroja

Upevnenie hlavice sonaru. Pri suchom testovaní, keď sonar počíta s rýchlosťou zvuku vo vzduchu je nevyhnutne potrebné mieriť presne kolmo na vec, ktorej vzdialenosť chceme odmerať. Vo vode sa ale podľa nášho testovanie správajú zvukové vlny zhovievavejšie, predpokladáme že je to do značnej miery zapríčinené menším množstvom prekážok (väčšina sa nachádza až na dne) a faktom že zvuková vlna je vystrelená do všetkých smerov. Preto nie je potrebné aby bola hlavica sonaru vždy namierená presne kolmo na dno, avšak pre najpresnejšie výsledky odporúčame aby bola prilepená o čln alebo iné plavidlo čo najpresnejšie.

Okrem hlavice je celý projekt bezpečne schovaný vo vlastnej krabíčke, ktorá môže byť uložená hocikde v lodi.

5 Komunikácia

SSH alebo Secure shell je kryptografický sieťový protokol na zabezpečenie bezpečnej komunikácie a prístupu do zariadení pripojených na sieť. Jedinými predpokladmi na zostrojenie takéhoto kanálu je v našom prípade povolené SSH na hlavnom počítači sonaru, ten zároveň musí byť na nejakej sieti pripojený a potom sme po zistení IP adresy a mena zariadenia schopný ho na diaľku ovládať.

Testovanie sa mi po zistení tohto spôsobu značne uľahčilo.

Zdieľanie výstupných súborov bolo pri prvej funkčnej verzii vyriešené pomocou protokolu SCP (Secure Copy Protocol), avšak neskôr sme to v rámci prehľadnejšej user interface vymysleli tak, aby sa výsledné dáta zapisovali priamo na USB kľúč.

Čiže pre spustenie interpretačného programu je postačujúce zo sonaru vytiahnuť USB na ktorom sa už nachádzajú aj výsledné dáta aj skript na zobrazenie a zapojiť ho do počítača.

Týmto riešením sme teda značne uľahčili prepojenie medzi zberačom dát a konečným výsledkom.

6 USB štruktúra

Použitím USB zariadenia ako úložiska kde uchováваме všetky pre projekt dôležité súbory toho mnoho získame. Najdôležitejším benefitom tohto riešenia je zjednodušenie samotného používania. Zrazu, namiesto hľadania IP adresy RPI zariadenia, ktorá sa mení pri zmene siete, nehovoriac o tom že je to možné iba ak sa v nejakej sieti nachádzame, máme všetko hneď po ruke, pripravené na spustenie.

Je dôležité podotknúť, že naša verzia RPI nemá prístup k WIFI sieti a na vytvorenie komunikačného kanála je nutné ho do siete zapojiť pomocou ethernetového kábla.

Ďalší zbytočne zložitý úkon je výsledné dáta nakopírovať do správneho súboru tak aby ho program na zobrazenie vôbec našiel. Tým že sonar.py beží priamo z USB ktoré je jednoduché odpojiť z RPI a pripojiť do počítača, následne stačí spustiť graph.py a všetko je okamžite na správnom mieste, pripravené na použitie. Program je teraz spustiteľný aj priamo pri jazere, keďže nepotrebujeme žiadnu sieť.

Zároveň uľahčujeme prípadné opravy alebo vylepšenia, pretože zase stačí iba USB vytiahnuť a zapojiť do počítača. Opravovať kód v linuxovom nano editore je takmer nemožné a aj pri najmenších opravách veľmi zdĺhavé (tabulátory je ťažké zoradiť do správneho poradia bez toho aby nám program padol na chybe nesprávneho odsadenia).

Na zabezpečenie funkčnosti tohto riešenia bolo ale potrebné sa vysporiadať s niekoľkými komplikáciami. Najprv treba nájsť kde sa USB v linuxovom systéme nachádza a potom treba vytvoriť mount point, čo je vlastne počítačová cesta k súboru. Po vypnutí, ku ktorému dochádza náhlo v momente keď je zdroj od RPI odpojený OS nemá možnosť odstrániť tento mount point a po opätovnom zapnutí vytvorí nový. V tom prípade sa ale cesty na automatické spúšťanie nezhodujú a program padá, pretože súbory sa už nachádzajú inde. Dalo sa nám to ale pomerne ľahko vyriešiť tým, že ešte pred spustením nášho skriptu napíšeme operačnému systému aby všetky mount pointy odstránil, a následne pridáme ten správny.

7 Sonar

Sonar.py je skript napísaný v pythone a jeho úlohou je meranie, zobrazovanie a ukladanie dát. Vynásobením času, ktorý sme získali z JSN modulu koeficientom rýchlosti zvuku získame vzdialenosť, v našom prípade hĺbku. Koeficient rýchlosti zvuku vo vzduchu je približne 346m/s zatiaľ čo vo vode je to niečo okolo 1500m/s.

Tým že používame tento modul pod vodou dokážeme viac ako štvornásobne zväčšiť dosah na približne 25 metrov (zároveň tým ale zväčšujeme aj minimálnu vzdialenosť). Sonar vysieľa pulzy podľa nastavenia, zvuková vlna sa vyšle vždy keď dostaneme nový pulz s GPS súradnicami. Takto dokážeme získať veľké množstvo dát potrebné na detailné vykreslenie dna pretože GPS modul prijíma signály niekoľko krát za sekundu, takže v priebehu hodiny merania sme schopní získať viac ako 4000 bodov.

Na vytvorenie grafu teoreticky nepotrebujeme presné súradnice, úplne by nám stačilo vedieť o aký kus sme sa pohli a ktorým smerom. To je ale ľahšie povedať ako urobiť obzvlášť v plavidle ktoré podlieha rôznym prírodným silám od vetra až po prúd a je prakticky nemožné udržať ho na jednom mieste. Riešením tohto problému sú už vyššie spomenuté presné súradnice. Tým nielenže kompletne vyriešime problém s meraním vzdialenosti ale ani sa nemusíme pohybovať po meranej ploche v žiadnej zmysluplnej postupnosti. Snažíme sa získať iba najväčšie množstvo údajov a s nimi si už poradí program.

Z tisíc riadkového textového súboru, kde každý riadok reprezentuje práve 1 bod na grafe, sa po spojení všetkých bodov zobrazí mapa.

Najprv sú importované knižnice s ktorými bude počas behu program pracovať a zadefinované fyzické porty pripojených modulov.

```
import RPi.GPIO as g
import time
import serial
import pynmea2
trigPin =2
echoPin =3
```

Potom je nastavený port z ktorého sú ťahané GPS dáta. Vo vnútri nekonečného cyklu sa po GPS lokalizácii a overení dát, (ak sú náhodou prázdne, proces sa opakuje bez zapísania) odmeria hĺbka, zobrazí sa na displeji a nakoniec je všetko zapísané do textového súboru sonarOutput.txt vo formáte zemepisná šírka, zemepisná dĺžka, hĺbka.

```
port="/dev/ttyAMA0" ser=serial.Serial(port, baudrate=9600,
timeout=1) dataout = pynmea2.NMEAStreamReader()
```

Vzor súboru sonarOutput.txt

```
49.174395332,20.055687685,1940.1
49.174438532,20.055807809,1938.4
49.174605334,20.056107965,1940.7
49.174729892,20.056349755,1947.0
49.174814411,20.056655414,1943.5
```

Definícia tejto funkcie je zaujímavá, pretože sa na nej dá presne vidieť akým spôsobom funguje komunikácia medzi mikropočítačom a modulmi. Na začiatku sa do trigPinu vyšle jeden signál. Zapne sa časovač a nastaví sa timeout aby sme v prípade chyby nečakali na stratenú vlnu. Ďalej vo while cykle zisťujeme či už sa nejaká vlna vrátila alebo či už nemeríme dlhšie ako je nastavený timeout. Po tom ako dostaneme hodnotu 1 z echoPinu, čo znamená že sa vlna dostala späť k zariadeniu, tento čas vynásobíme rýchlosťou zvuku vo vode a vydelíme 2 (cesta tam a späť) a tak dostaneme vzdialenosť, ktorú funkcia vráti.

```
def getDepth():
    g.output(trigPin, True)
    time.sleep(TRIGGER_TIME)
    g.output(trigPin, False)
    start = time.time()
    timeout = start + MAX_TIME
    while g.input(echoPin) == 0 and start <= timeout:
        start = time.time()
    if(start > timeout):
        return -1
    stop = time.time()
    timeout = stop + MAX_TIME
    while g.input(echoPin) == 1 and stop <= timeout:
        stop = time.time()

    if(stop <= timeout):
        elapsed = stop-start
        distance = float(elapsed * speedS)/2.0
    else:
        return -1
    return distance
```


8 Graf

Program graph.py sa začína importovaním všetkých knižníc dôležitých pre náš projekt.(numpy a funkcia pyplot z matplotlib). Po inicializácii sú prázdne polia naplnené dátami z textového súboru sonarOutput.txt. Z x a y polí následne vytiahneme najmenšie hodnoty aby sme ich neskôr mohli odčítať od každého prvku v poli a tým získať dáta ktoré sú relatívne iba voči sebe navzájom a najmenšia možná hodnota sa stáva 0 namiesto celého čísla zemepisnej šírky alebo dĺžky daného bodu.

Tieto polia sa pretransformujú na numpy polia s ktorými vie pracovať pyplot a vytvorí sa graf s 3d projekciou. Dáta sa na 3d plátno najprv nahádzu iba ako malé bodky. A o riadok nižšie sú všetky spojené vrstvou ktorá ich použije ako záchytné body. Nakoniec sa graf vykreslí aj s legendou.

V nasledujúcom bloku kódu odčítame od všetkých súradníc minimálnu hodnotu v poli čím získame iba čisté vzdialenosti medzi bodmi. Potom prepočítame zemepisné stupne na metre.

A vypočítame proporciu pre výšku, tento krok je potrebný pretože chceme aby bol graf čitateľný a pri nespriemerovaných hodnotách by sa nám mohlo stať že by graf pôsobil iba dvojdimenzionálne, keďže vzdialenosti medzi okrajmi vodnej plochy môžu byť kilometrové ale hĺbka zriedkavo prekročí 25 metrov, a keby aj, tak to nedokážeme odmerať. Týmto spôsobom teda zaručíme že je graf vždy natiahnutý po celej osi Z a je jednoduchší na porozumenie.

```
gpstoMeter = 111319.49079327358 #1 stupen
xi = [(i - minx)*gpstoMeter for i in xi]
yi = [(i - miny)*gpstoMeter for i in yi]
proportion=(max(xi)/max(elevationi)
max(yi)/max(elevationi))/2
elevationi= (((i-mine)/proportion)*h for i in elevationi]
```

8.1 Validácia dát

Aby sme zabránili vodným živočíchom alebo iným objektom plávajúcim vo vodnom stĺpci pred kazením nášho merania, museli sme implementovať systém ktorý nám pomocou priemeru vyhodnotí či sa jedná o dno alebo o živočícha, ktorého v grafe mať nechceme.

Tento systém funguje tak, že pri pridávaní dát do numpy polí vždycky spriemerujeme niekoľko predchádzajúcich meraní a niekoľko nasledujúcich meraní, ktorými potom vydelíme to meranie, ktoré práve posudzujeme. Ak je táto hodnota v rozsahu, ktorý sme si na začiatku zdefinovali, je pridaná do poľa a neskôr vykreslená na grafe. Ak v nie, jednoducho sa preskočí.

Keďže sa táto operácia vykonáva až tesne pred bodom vykresľovania, namiesto pri zbere, problematické hodnoty nie sú vymazané, iba nie sú v tejto iterácii vykreslené. Takto si môžeme náš rozsah podľa potreby meniť takým spôsobom, aby sme čo najviac obmedzili počet vykreslených bodov, ktoré nie sú dno a zároveň nestratili veľa bodov, ktoré sú.

8.2 Testovanie

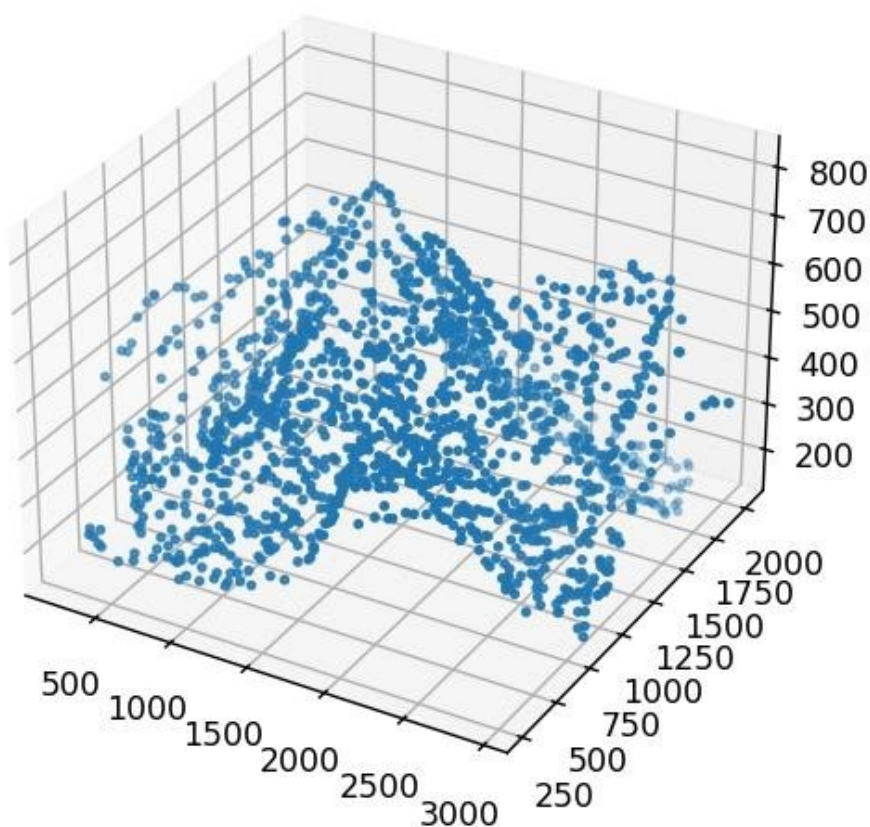
Program na vytváranie grafu bol testovaný na cvičných dátach, ktoré sme získali z Google Earth (GE). V aplikácii GE sme si vytvorili vlastnú trasu pozostávajúcu z množstva pospájaných bodov na mape. Tú sme exportovali ako kml súbor. Nakoniec pomocou webstránky gpsvisualizer.com sme ku každému z bodov priradili odpovedajúcu výšku a kml súbor konvertovali na textový. Potom už stačí len premiestniť súbor do adresára kde sa nachádza `graph.py` a po spustení sa v priebehu niekoľkých sekúnd zobrazí naše pohorie, dolina alebo mesto podľa tohto súboru.

8.3 Vykresľovanie grafu

Po načítaní dát do zodpovedajúcich polí a po vykonaní všetkých matematických operácii tak že máme k dispozícii dáta, ktoré chceme zobrazit', rozhádzeme všetky body po trojdimenzionálnom grafe pomocou funkcie scatter z knižnice Matplotlib.

```
scatter=ax.scatter(x,y,elevation,c=elevation,cmap=colormap, marker='.')
```

V tomto bode by sme na mape mali iba jednotlivé body, zdanlivo chaoticky rozhádzané a bolo by veľmi zložité niečo z grafu vyčítať.

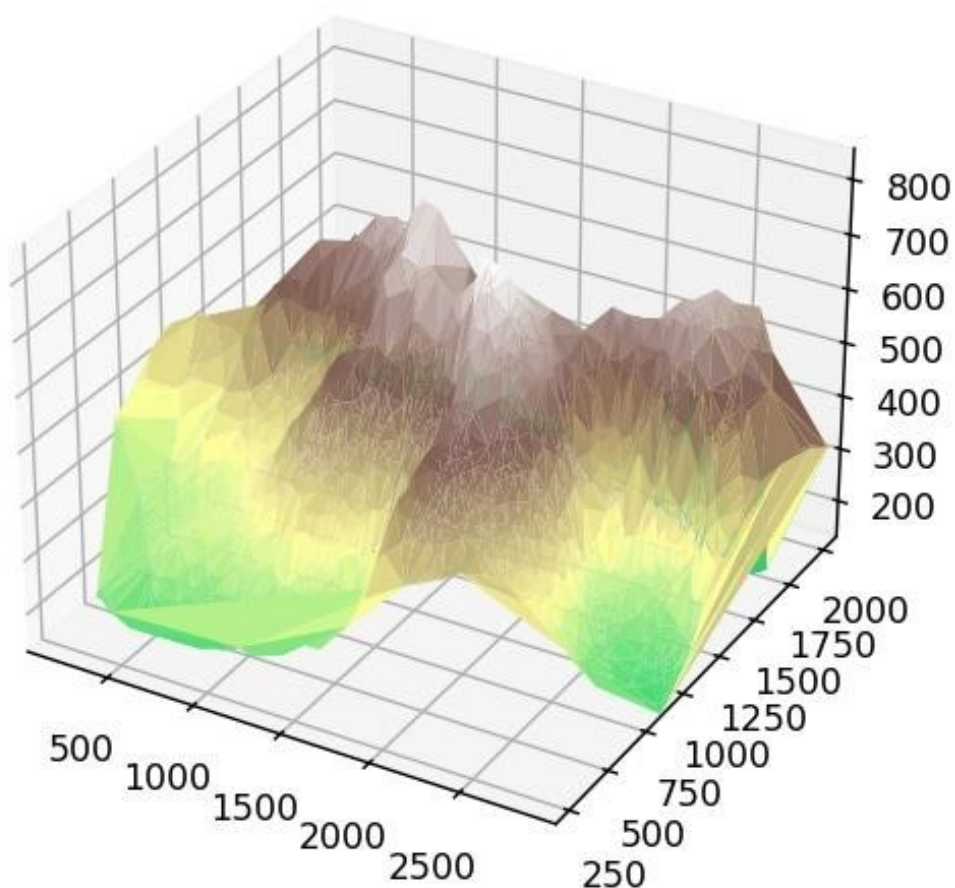


Obrázok 5 funkciou scatter rozmiestnené body

Z dát ale jednoducho, pomocou funkcie `plot_trisurf` vytvoríme čitateľný graf.

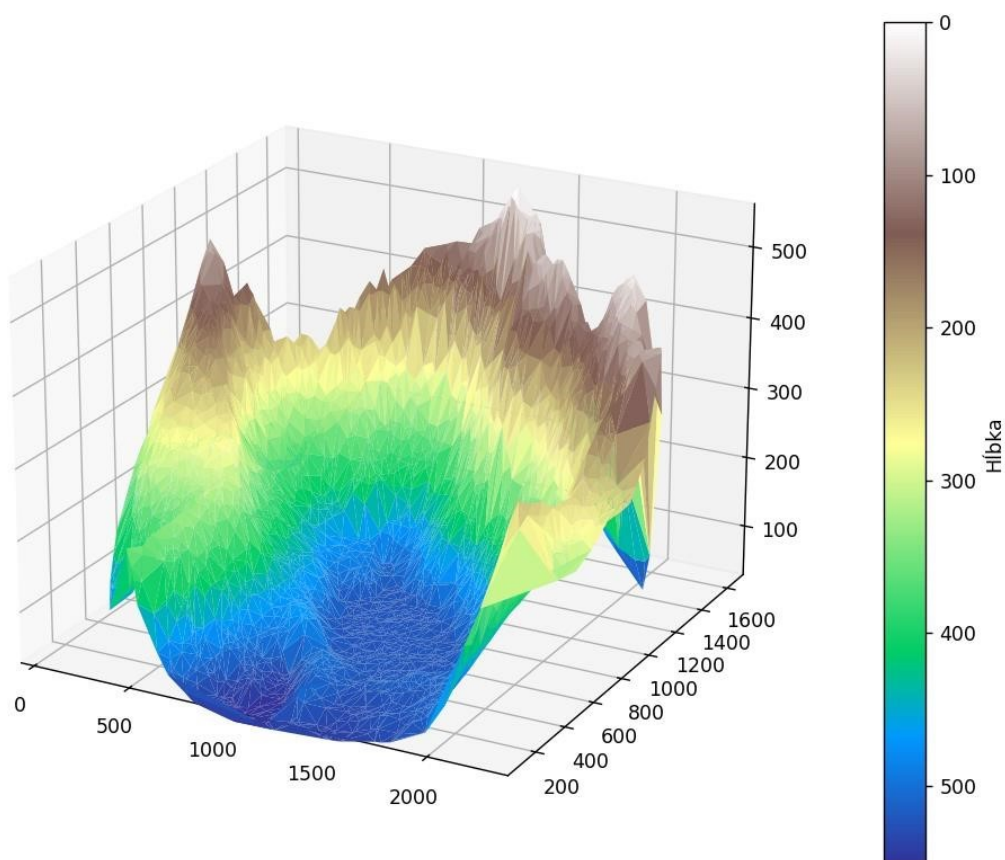
```
ax.plot_trisurf(x, y, elevation, cmap=colormap, edgecolor='none')
```

Táto funkcia cez body prehodí vrstvu ako by sme ich prekryli plachtou s podmienkou že každý bod sa jej musí dotýkať a na všetkých miestach je úplne vyrovnaná. Na túto vrstvu je potom aplikovaný colormap, ktorý má na starosti farebne diferencovať na základe hĺbky.



Obrázok 6 na body je aplikovaná farebná vrstva

8.4 Analýza grafu



Obrázok 7 Graf vykreslený z testovacích dát

Na pravej strane máme legendu, modrá – hlboko a hnedá – plytko. To je v skratke všetko dôležité, čo pred čítaním potrebujeme vedieť.

Podľa všetkých rybárskych poznatkov, ktoré som bol schopný získať od ďalších rybárov môžeme určiť najpravdepodobnejšie miesto výskytu rýb podľa druhu.

Napríklad o zubáčoch je všeobecne známe že sa nachádzajú na hranách. Z tohto by sme mohli usúdiť, že najviac ich bude na súradniciach [1000,400] tesne nad dnom.

9 Alternatívne aplikácie

Fyzickú časť projektu - sonar - je možné využiť iba pre potreby mapovania dna vodných plôch. S malými úpravami by ale bolo možné celý prístroj prilepiť na dron a mapovať ako pod vodou, tak aj na súši. Limitujúcim faktorom by v tomto prípade bol dosah senzoru JSN, ktorý je na súši iba niečo okolo 6 metrov.

Namerané dáta sa dajú využiť síce iba na vyplňanie už spomenutých modrých miest na mapách, avšak takýchto máp je nesmierne množstvo a viem si predstaviť že napríklad aplikácia fisharea, ktorá vyslovene slúži iba na zaznamenanie rybárskych revírov, by tieto dáta vedela jednoducho do mapy integrovať.

Program na vytvorenie grafu sme schopní spustiť s najrôznejšími dátami. Napríklad z Google Earth si vieme exportovať náš vlastný set bodov, ku ktorým si pomocou ďalšieho programu vieme priradiť prevýšenie a máme teda možnosť si vizualizovať akékoľvek miesto na našej planéte, či už Lomnický štít, alebo horu Kilimanjaro.

10 Problémy

Pri konštruovaní tohto projektu sme museli čeliť viacerým problémom, ktoré zabrali viac času ako by bolo ideálne a po eventuálnom vyriešení sa javili ako banálne, až hlúpe.

10.1 GPS modul nič nezobrazuje

Sériový port ktorý má podľa našej konfigurácie na starosti zobrazovanie NMAE viet z GPS nič nezobrazuje. Keďže podľa všetkých návodov by malo červené svetielko na module prinajmenšom blikať na potvrdenie signálu a moje iba svietilo, začali sme mať podozrenie že je GPS modul nefunkčný. Po veľa hodinách nastavovania všetkých možných konfigurácii a preštudovania všetkých existujúcich zdrojov k tomuto modelu v troch rôznych jazykoch sme zistili že na pozadí beží skript na polievanie mojej letnej farmy na cvrčkov.

Samozrejme po odstránení riadku `sudo python Cvrcky.py` z rc.local sa problém okamžite vyriešil.

10.2 Nezastaviteľný skript

Po spustení programu sa nedá vypnúť. Toto bol zase veľmi náročný problém na zvládnutie. Otestoval ale skôr naše psychické hranice než znalosť počítačov. Keďže skript sa automaticky spúšťal pri spustení a nedal sa vypnúť inak ako cez terminál a zároveň štandardné `ctrl+c` (skratka na vypnutie bežiaceho procesu) nefungovalo pretože na spracovanie NMEA viet, ktoré občasne vynechali kvôli čomu následne program spadol, bolo do kódu potrebné pridať `try-except` blok. Pri ktorom sme ale nevedeli čo by mala tá `except` časť robiť lebo sme vlastne potrebovali iba aby to skúsil znovu čiže tam padol `pass`.

Týmto nešťastným riešením sa síce jeden problém krásne vyriešil ale pri ďalšom testovaní sme zistili že je skript možné vypnúť iba cez terminál pomocou príkazu

```
sudo kill <pid> #pid je id programu
```

A pri ďalšom testovaní sme zistili že do tohto terminálu sa cez SSH ale nedokážeme dostať, pretože kanál pre SSH už využíva niektorá časť programu. Takže sa pri každom zapnutí spustil program, ktorý sa nedal vypnúť, čo bolo veľmi frustrujúce.

Nakoniec po hodinách trápenia to vyriešila už spomenutá USB štruktúra. Tým že sa skript spúšťa z USB kľúča ho na zastavenie stačí vytiahnuť a RPI reštartovať. Ďalej sme to neriešili a v konečnom dôsledku nie je potrebné finálny skript ani vypínať, takže nevádi že sa to nedá.

Záver

Zadanie mojej maturitnej práce sa nám podarilo splniť. Po spojení oboch komponentov sme schopní vyraziť na meraciu misiu a po tom ako sonarom nazbierame dostatočné množstvo dát, môžeme pomocou skriptu `graph.py` vykresliť podrobnú mapu štruktúry dna.

V dokumentácii sme rozobrali rôzne problematiky týkajúce sa technológií v našom projekte, programovací jazyk spolu s použitými knižnicami ako aj schémy zapojenia a využitých modulov. Oboznámili sme čitateľa s niektorými časťami kódu a ich dôležitosťou aj s chybami, ktoré v procese programovania vznikli aj akým spôsobom zase zanikli.

Aj keď je projekt hotový, má svoje limitácie. Technológia GPS prijímačov má stále priestor na zlepšenie a ak by bolo nutné, presnosť nášho grafu by mohla byť vylepšená novším prijímačom alebo silnejšou anténou.

V budúcnosti, v rámci vylepšovania, by sme si ešte vedeli predstaviť veľmi jednoduché používateľské rozhranie na zadávanie kritérií pre validáciu dát. Toto je pre nás v momentálnom štádiu úplne bezúčelové nakoľko je veľmi jednoduché ich zmeniť priamo v programe, pre bežného používateľa by to ale mohlo byť veľkým prínosom.

Použité technológie zostanú aktuálne a relevantné ešte dlhú dobu a práve preto som rád že sme sa podujali práve na tento projekt a s týmito technológiami sa oboznámili, verím že sa nám to ešte niekedy zide.

Zároveň treba spomenúť že aj keď cieľom mojej maturitnej práce bolo iba tento systém zostrojiť a sprevádzkovať, teším sa keď budem mať príležitosť ho generálne otestovať. Snáď sa mi podarí aspoň zopár modrých miest z mojej mapy vymazať.

ZOZNAM BIBLIOGRAFICKÝCH ODKAZOV

[1] Gus. The Raspberry Pi 2. Cit. 20.1.2024.

Dostupné na:

<https://pimylifeup.com/raspberry-pi-2/>

[2] Geog862. The GPS signal. Cit 20.1.2024.

Dostupné na:

<https://www.e-education.psu.edu/geog862/book/export/html/1407>

[3] Aws. What is Python. Cit 20.1.2024.

Dostupné na:

<https://aws.amazon.com/what-is/python/>

[4] JSN-SR04T-2.0. Cit 21.1.2024

Dostupné na:

<https://media.s-bol.com/RLLgqDkqO7EY/original.pdf>

[5] Research gate. Figure 2. Cit 24.1. 2024.

Dostupné na:

https://www.researchgate.net/figure/Measuring-of-depth-of-ocean-bySONAR_fig2_351817295

[6] Matplotlib. Custom hillshading in a 3D surface plot. Cit. 24.1. 2024.

Dostupné na:

https://matplotlib.org/stable/gallery/mplot3d/custom_shaded_3d_surface.html

[7] The Pi4j Project. Pin Numbering - Raspberry Pi 2 Model B. Cit 24.1. 2024.

Dostupné na:

<https://pi4j.com/1.2/pins/model-2b-rev1.html>

[8] Ublox. NEO-7 series. Cit 28.1.2024

Dostupné na:

<https://www.u-blox.com/en/product/neo-7-series>

[9] ETHW. Milestones:Invention of Sonar, Milestones:Invention of Sonar. Cit 28.1.2024

Dostupné na:

https://ethw.org/Milestones:Invention_of_Sonar,_1915-1918

Prílohy

1. USB