

Cuprins

Introducere	2
Tema proiectului	2
Capitolul 1 Analiza critică	4
Capitolul 2 Proiectarea sistemului de monitorizare	14
Capitolul 3 Implementarea sistemului de monitorizare	18
3.1 Implementarea hardware	18
3.2 Componente hardware	19
3.2.1 Placa de dezvoltare Heltec Stick Lite.....	19
3.2.2 Senzor de temperatura DS18B20.....	22
3.2.3 Senzor de măsurare cu ultrasunete JSN-SR04T	23
3.2.4 Senzor de temperatură și umiditate DHT22	25
3.2.5 Tranzistor AO3401	26
3.2.6 Convertor analog-numeric(ADC).....	29
3.2.7 Dioda 1N5819.....	31
3.2.8 Panoul solar.....	31
3.3 Realizarea montajului	32
3.4 Blockchainul Helium si configurarea consolei Helium	39
3.4.1 Crearea contului pentru consola.....	41
3.4.2 Adăugare dispozitiv	41
3.4.3 Creare funcție decodare.....	41
3.4.4 Aplicația de vizualizare.....	43
3.5 Configurarea si programarea microcontrolerului	44
3.6 Configurare TagoIO	52
3.6.1 Creare cont.....	52
3.6.2 Adăugare dispozitiv	53
3.6.3 Creare Dashboard	54
3.6.4 Creare alerte	57
Concluzii	58
Bibliografie	59
Summary	61

Introducere

De mai mulți ani exista pe piață comercianți ce au o varietate de sisteme folosite pentru monitorizare și gestionare a stocurilor de produse existente în parcuri de rezervoare. Principalul domeniu ce folosește aceste tipuri de sisteme fiind cel petrolier, ca de exemplu parcurile de rezervoare ale rafinăriilor de petrol, ale terminalelor de expediție/recepție produse petrolier, ale schelelor petroliere, etc.[11]

Toate aceste sisteme putând fi modularizate, fiind alcătuite din echipamente și/sau programe în funcție de nevoile și funcțiile specifice necesare utilizatorului. Odată cu evoluția tehnologiei, aceste sisteme au devenit accesibile din punct de vedere financiar, astfel acestea se regăsesc din ce în ce mai des la tot mai multe companii și persoane fizice. Toate aceste sisteme de monitorizare făcând parte astăzi din IOT (internetul lucrurilor).

În zilele noastre comunicarea fără fir este una din cele mai dorite metode de transmisie a datelor între două sau mai multe dispozitive. Prin această metoda, comunicarea datelor se efectuează și se transmite prin aer prin unde electromagnetice, cum ar fi mai exact undele radio, dar mai există posibilitatea folosirii și de unde infraroșu sau satelit. Toate acestea în detrimentul folosirii cablurilor și firelor.

Comunicațiile wireless (fără fir) funcționează pe diferite zone particulare din spectrul electromagnetic de la 3 Hz la 3000 GHz, numite unde radio. Aceste unde includ o varietate largă de utilizări și folosințe în diverse aplicații de comunicații, ca de exemplu: la dispozitivele celulare (3G, 4G, 5G), rețele WIFI, NFC, cuptorul cu microunde, sisteme aeronautice, maritime cât și alt servicii radio comerciale și private.

Tema proiectului

Proiectul intitulat “ Proiectarea și simularea unui sistem de monitorizare a parcurilor de rezervoare” are rolul de a îmbină atât partea hardware cât și cea software a unui sistem de achiziții date folosind tehnologia Lora.

Un astfel de sistem, monitorizează și contorizează informațiile recepționate, prin tehnologia de transmisie radio, de la mai mulți senzori: de temperatură, de distanță, de

umiditate si de măsurare tensiune. Toate aceste informații sunt trimise într-o bază de date putând fi accesate de la distanță.

Am ales folosirea tehnologiei de transmisie radio deoarece este una dintre cele mai dorite moduri de conectivitate între două sau mai multe dispozitive. Datele vor fi transmise cu ajutorul unei plăci de dezvoltare Heltec Wireless Stick Lite ce permite conexiuni LoRa si încărcarea datelor în consola Helium. De acolo datele vor fi redirecționate către platforma Tago.io unde vor fi memorate si vizualizate.

Capitolul 1 Analiza critică

În prezent echipamentele de automatizare sunt aproape în exclusivitate numerice. Acestea pot fi utilizate pentru realizarea celor patru funcții relevante asociate automatizării proceselor și anume:

- **monitorizarea automata**/cunoașterea stării;
- reglarea automata;
- protecția automata;
- optimizarea automata.

În contextul de obiect al automatizării **procesul** reprezintă o succesiune de transformări asociate evoluției spațio – temporale a diverselor obiecte. Procesele fizice sunt de regula însoțite de transferuri masice, energetice sau de impuls. Procesele care se desfășoară în instalații industriale sunt cunoscute sub denumirea de procese tehnologice.[1]

Unui proces îi sunt de regulă asociate trei categorii de obiective și anume:

- obiective de calitate
 - obiective de securitate
 - obiective de eficiență
-
- ❖ Monitorizarea automată este implicată în realizarea tuturor celor trei obiective ale unui proces.
 - ❖ Reglarea automată susține în primul rând realizarea obiectivului de calitate al unui proces.
 - ❖ Protecția automată asigură realizarea obiectivului de securitate al unui proces.
 - ❖ Optimizarea automată asigură realizarea obiectivului de eficiență al unui proces.

Pentru a fi utilizate în domeniul conducerii proceselor, calculatoarele numerice trebuie să răspundă unor cerințe între care de o importanță aparte sunt considerate următoarele:

- 1) siguranță în funcționare;
- 2) procesarea informației în timp real;
- 3) posibilitatea conectării la perifericele de proces;
- 4) posibilitatea dialogului cu personalul de operare.

1) Siguranța în funcționare unui echipament numeric se caracterizează prin intermediul fiabilității, mentenabilității, disponibilității și al indicatorilor specifici.

- Fiabilitatea este reprezentată de timpul mediu dintre defecțiuni.
- Mentenanța se definește din punct de vedere calitativ ca fiind ansamblul tuturor acțiunilor legate de menținerea și restabilirea funcțiilor unui produs înapoi la starea normală de funcționare.
- Disponibilitatea unui produs, reprezintă aptitudinea unui produs de a-și îndeplini funcțiile specificate sub aspectul combinat al fiabilității și mentenabilității la un moment dat sau un timp specificat.

2) Comportarea în timp real presupune un sincronism care trebuie să existe între operațiile interne de calcul și evenimentele lumii exterioare. Un sistem de conducere are comportare în timp real dacă viteza de reacție la stimulii din proces este în concordanță cu inerția acestuia. Se vorbește de comportarea în timp real la preluarea datelor din proces și la transmiterea comenzilor către acesta.[1]

3) Utilizarea echipamentelor numerice de calcul în conducerea proceselor presupune un permanent schimb de informație între cele două entități. Informația privind starea procesului se obține prin intermediul traductoarelor iar transpunerea în proces a comenzilor generate de către echipamentul numeric este realizată de către elementele de execuție.[1]

4) Conducerea instalațiilor tehnologice se realizează din camere sau puncte de comandă. În aceste locuri trebuie să existe mijloace capabile să ofere operatorului de proces posibilități care să-i permită atât informarea privind starea procesului cât și intervenții ocazionale de anumite evenimente apărute în evoluția acestuia.

Conceptul de monitorizare este legat nemijlocit de cunoașterea stării momentane și a evoluției parametrilor aferenți unui proces.[2]

Monitorizarea este implicată în realizarea tuturor obiectivelor specifice unui proces.

În cadrul sistemelor de măsurat la distanță, date privind valorile parametrilor se pot obține prin măsurare și/sau prin calcul.

Aplicațiile de monitorizare trebuie sincronizate cu timpul, respectiv executate cu o anumită periodicitate.

Sistemul reprezintă o colecție de obiecte aflate în interacțiune căreia îi sunt specifice o anumită organizare și un anumit scop. Caracterizarea unui sistem se poate face în manieră structurală sau informațională[1].

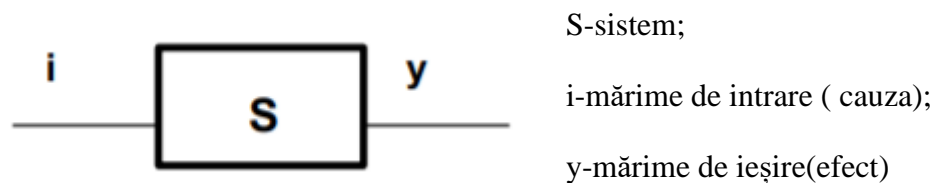


Fig. 1.1 Caracterizarea informațională a unui sistem [1]

Un **sistem de măsurat la distanță** include următoarele elemente:

- Traductor (senzor)
- Mediu(linie) de transmisie
- Instrument de vizualizare

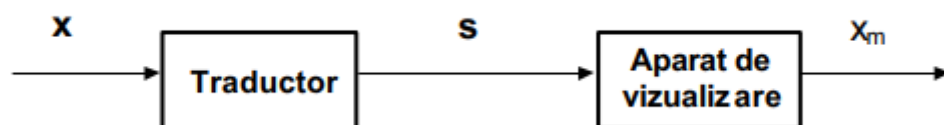


Fig. 1.2 Sistem de măsurat: x – mărime care se măsoară; s – semnal; x_m – rezultat măsurare[1]

Traductoarele sunt elemente de automatizare care sesizează și transformă mărimea de măsurat într-un semnal mai convenabil, de obicei de altă natură fizică.[10]

Traductorul mai poate fi definit ca un element cu ajutorul căruia valoarea unei variabile se transpune pe un semnal purtător de informație. Acest semnal poate fi utilizat în sisteme de măsurare, în dispozitive de calcul sau în sisteme de semnalizare și protecție[3].

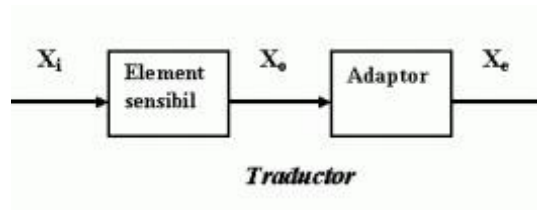


Fig. 1.3 Traductor[3]

Senzorii sunt acele elemente prezente în sistem care pot înlocui cu succes simțurile umane (auditiv, vizual, tactil), dar aceștia pot detecta și acele efecte fizice sau chimice greu detectabile de către oameni (ultrasunete, infraroșu, magnetic, electrostatic, etc). Un senzor cuprinde traductoare cu rolul de a converti mărimea de intrare într-un semnal util. Senzorii sunt principalele dispozitive ce deosebesc calitățile unui sistem. Rolul senzorilor fiind acela de a crește performanțele sistemelor prin adăugarea de noi proprietăți în acesta.

Mulți dintre senzori seamănă (cel puțin principal) cu elementele sensibile din aparatele de măsură, cu o diferență esențială: de regulă senzorul doar semnalează apariția (ori schimbarea) unei stări, pe când instrumentul de măsură o evaluează cantitativ (valoric).

Din altă perspectivă vom observa că senzorul este destinat să stea de veghe și să comunice unei instalații automate[5].

Linii de transmisie pentru semnale analogice sunt realizate în exclusivitate din conductoare metalice. În cazul traductoarelor numerice (smart transducers), acestea sunt conectate de regulă în rețele în care transmisia poate fi cu sau fără fir (wireless).

În zilele noastre comunicarea fără fir este una din cele mai dorite metode de transmisie a datelor între două sau mai multe dispozitive. Prin această metoda,

comunicarea datelor se efectuează și se transmite prin aer prin unde electromagnetice, cum ar fi mai exact undele radio, totul în detrimentul cablurilor.

Canalul de transmisie este un circuit fizic de tip electric sau electromagnetic, cu un efect atenuator asupra semnalului și cu o bandă de propagare limitată. În canalul de comunicație se mai adaugă și zgomote aleatoare care degradează semnalul-mesajul inițial.

Mesajul reprezintă o mulțime de caractere și simboluri destinate transferului de informație de la sursa (emițător) la destinație (receptor). De regulă semnalele sunt alcătuite conform anumitor reguli cunoscute atât de emițător cât și de receptor.[1]

Semnalul poate fi definit ca o mărime fizică de o anumită natură, luând valori într-un domeniu dat, utilizată într-un domeniu aplicativ.

Semnalul poate fi util sau perturbator. La transmiterea prin canalul de comunicație poate să apară o degradare a semnalului datorată perturbațiilor datorate mediului ambiant sau distorsiunilor provocate de tehnica de transmisie.[9]

Instrumentele de vizualizare pot fi reale sau virtuale. Instrumentele reale se prezintă sub forma indicatoarelor și înregistratoarelor digitale. Instrumentele virtuale prezintă practic utilizatorului o interfață grafică pe monitorul unui calculator, unde valorile parametrilor sunt prezentate pe diferite tipuri de indicatoare programate.

De regulă un sistem de măsurare la distanță poate avea ca utilizatori regulatorul sau operatorul uman. Regulatorului îi este suficient semnalul purtător de informație, în timp ce pentru utilizatorul uman este absolut necesară prezenta aparatului de vizualizare.

Informația poate fi definită ca știre, veste, în strânsă legătură cu conceptul de comunicație și cu modul de propagare a energiei asociate semnalului intermediar, precum și cu modalitățile de stocare a informației.[9]

În domeniul tehnic, termenul de informație a fost inițial introdus de Shannon pentru a desemna incertitudinea înlăturată prin realizarea unui eveniment dintr-un set de evenimente posibile. Ulterior semnificația termenului de informație s-a extins la cunoaștere în general, respectiv la apariția unui element nou, necunoscut anterior fie pentru om fie pentru un sistem de calcul, asupra realității înconjurătoare.

Măsurarea reprezintă un proces experimental de comparare a mărimii care se măsoară, x cu o alta mărime de aceeași natura, um, numita unitate de măsură.

Orice proces de măsurare este însoțit de erori, între care semnificative sunt erorile absolute si relativa. Eroarea absoluta este definita ca diferența între valoarea măsurata si cea reala respectiv.

$$e_{abs} = x_m - x_r$$

În figura de mai jos se prezinta o schema generala a unui sistem de monitorizare a unui parc de rezervoare:

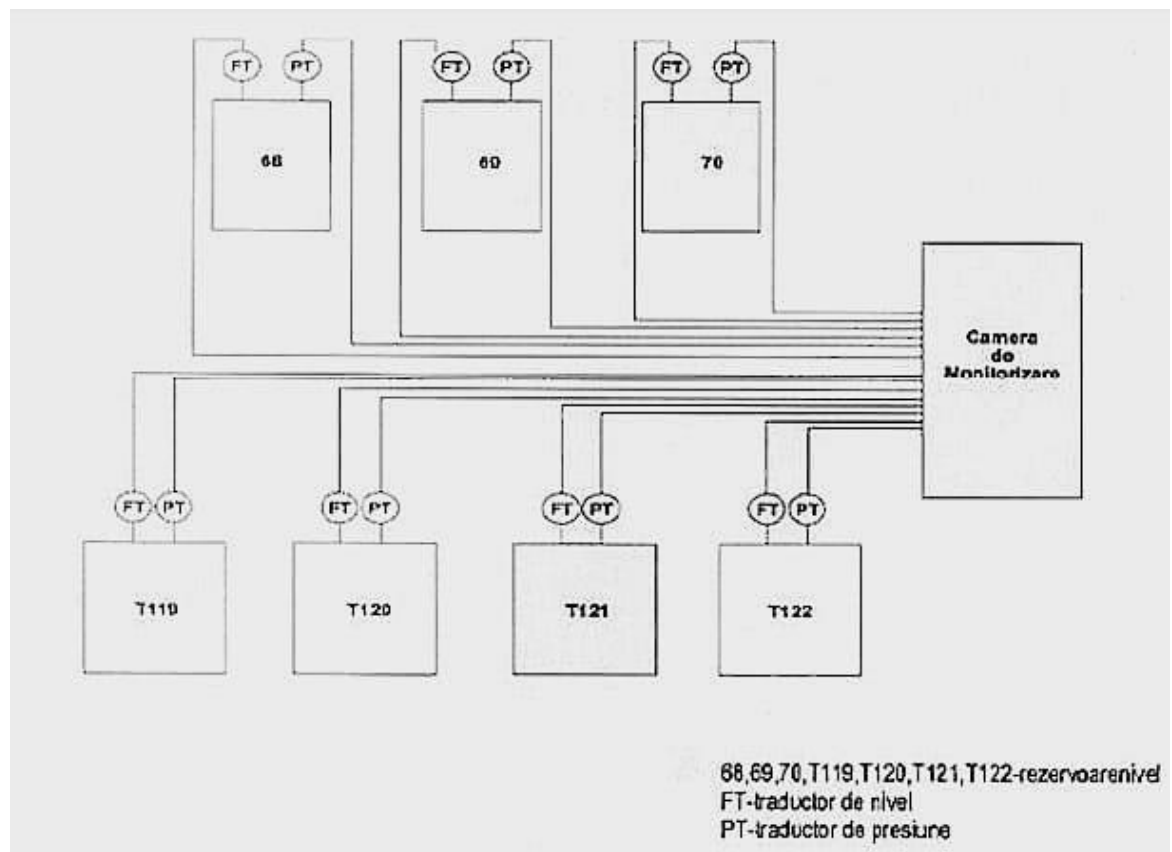


Fig. 1.4 Schema de automatizare a unui parc de rezervoare[11]

În continuare se prezintă un sistem complet de monitorizare existent de piață, produs de firma Saab-Rosemont, cu echipamente și programe TankMaster:

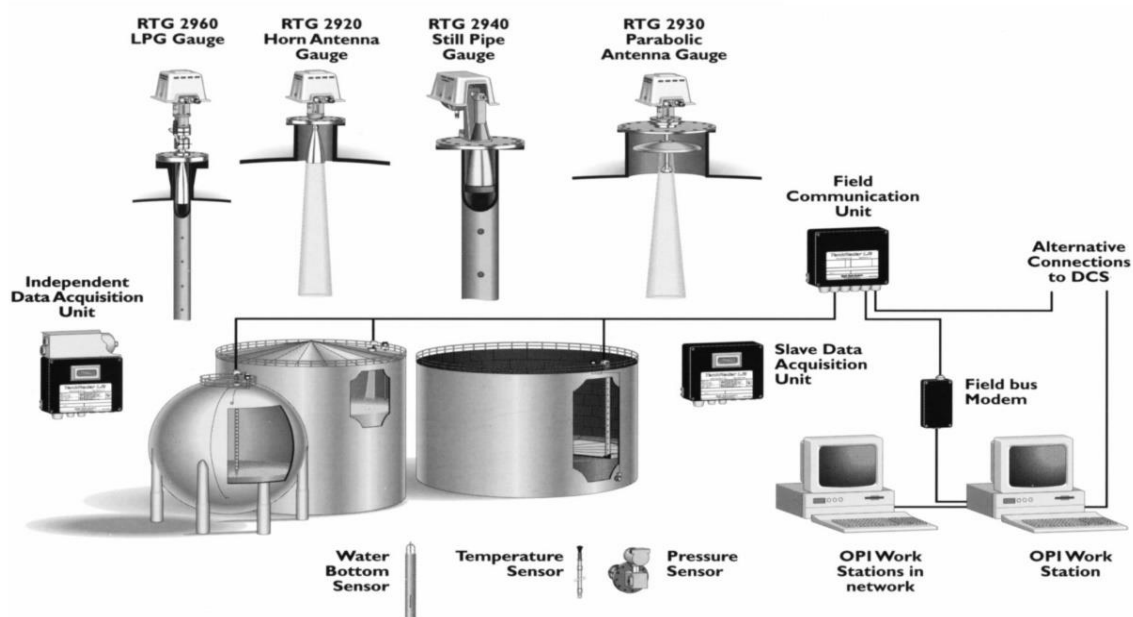


Fig. 1.5. Sistemul Saab TRL/2[11]

Principalele componente ale acestui sistem sunt:

- Traductoare de nivel
- Traductoare de temperatura
- Unitatea de comunicație
- Stațiile de lucru
- Interfața de operare OPI

Traductorul de nivel RTG-2940 este un dispozitiv de măsurare a nivelului. Acesta este montat de o țevă de liniștire cu diametrul între 6" și 12". Montajul trebuie să fie realizat într-un rezervor, în partea superioară a acestuia.

Principiul de funcționare al acestui traductor de nivel este de emisie și recepție al unui impuls către suprafața lichidului, iar timpul parcurs de impuls dus-întors este proporțional cu distanța parcursă.

Traductoarele de nivel bazate pe emisia unui impuls trebuie să fie montate la o distanță suficient de mare față de peretele vasului astfel încât impulsul să nu atingă peretele vasului și să nu fie montat în drept orificiului de intrare a fluidului în vas. [4]



Fig. 1.6. Traductorul de nivel RTG 2940[11]

Traductor de temperatură Guenther de tip termocuplu este format dintr-o pereche de termoelectrozi din aliaje diferite, sudați împreună la unul din capete. Prin încălzire apare prin diferență de temperatură a ambelor metale o tensiune electrică măsurabilă care permite concluzii exacte și normat despre temperatura la care s-a ajuns.

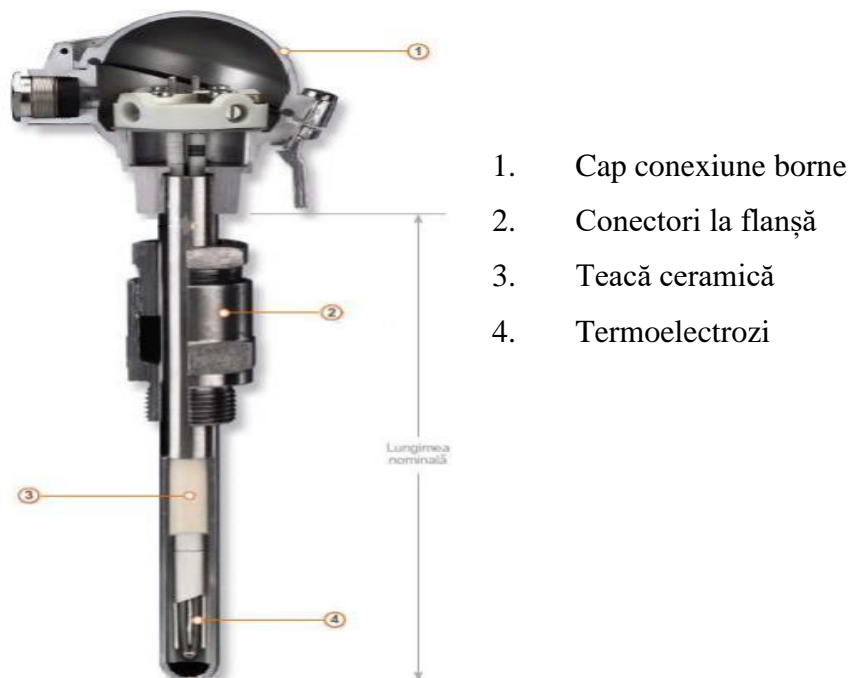


Fig. 1.7. Traductor Guenther de temperatura[4]

Montarea traductoarelor de temperatura in rezervoare pot fi amplasate in partea de sus a vasului sau in pereții laterali.[4]

Unitatea de comunicație FCU2160 este un concentrator de date care preia datele de la sistemele de măsurare existente la rezervoare și le stochează într-o memorie tampon. Ori de câte ori există o cerere de date, unitatea de comunicație va trimite imediat date de la un grup de rezervoare din memoria tampon actualizată.

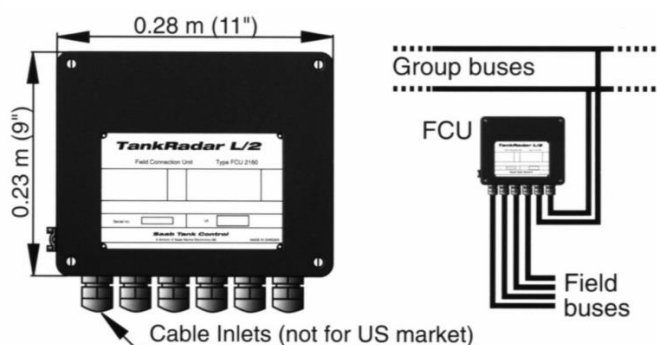


Fig. 1.8 Unitatea de comunicație FCU2160[11]

Interfața de operare OPI

Software-ul OPI este instalat pe calculatoarele de tip PC conectate la sistemul de monitorizare rezervoare Saab TankRadar L. Principalul avantaj al acestui soft este că ușurează manipularea informațiilor despre temperatura și nivel a produselor din rezervoare, dar totodată facilitează accesul la diferite opțiuni ale sistemului de măsurare. De exemplu: se pot vizualiza datele pentru fiecare rezervor, cu parametrii despre temperatura, nivel, etc; de altfel se pot grupa rezervoarele pe criterii (exemplu în funcție de locație sau de tipul de produs).

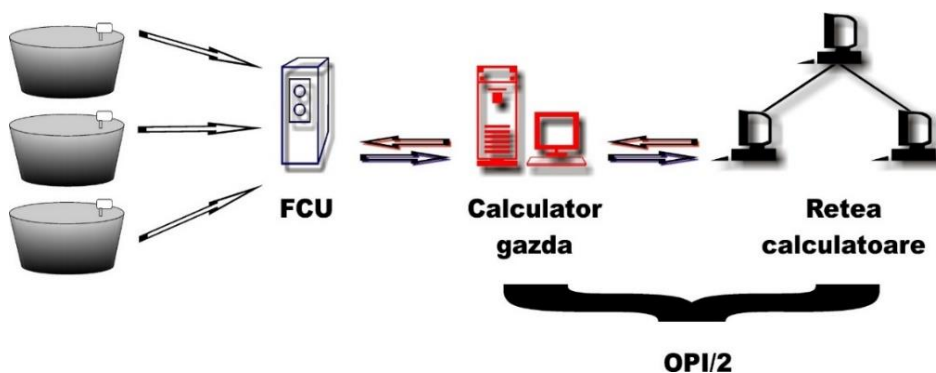


Fig. 1.9 Interfața de operare[11]

O alta caracteristica importanta a softului este aceea de creare alerte si liste de evenimente. Programul este ușor de folosit ,având diferite ferestre de asistentă si algoritmi de detecție ce permit configurarea de alarme de scurgeri de produse, dar are si opțiuni de vizualizare a mișcărilor fluidelor in rezervoare.

Capitolul 2 Proiectarea sistemului de monitorizare

Sistemul de monitorizare este structurat în doua mari componente, formate atât din părți hardware cât și software. Aceste doua componente sunt:

I) Interfața cu utilizatorul

- Componente hardware:
 - dispozitiv vizualizare: Pc, Smartphone
- Componente software:
 - Aplicația Tago
 - Consola Helium

II) Sistemul de monitorizare

- Componente hardware:
 - Placa de dezvoltare Heltec Wireless Stick Lite
 - Senzor temperatura DS18B20
 - Senzor ultrasonic JSN-SR04T
 - Senzor temperatura si umiditate DHT22
 - Acumulator 18650
 - Panou solar 6V
 - Diodă Schottky

Proiectul are rolul de a îmbină atât partea hardware cât și cea software a unui sistem de achiziții date folosind tehnologia Lora. Un astfel de sistem, monitorizează informațiile recepționate , prin tehnologia de transmisie radio, de la mai mulți senzori: de temperatură, de distanță, de umiditate și de măsurare tensiune. Toate aceste informații sunt trimise într-o bază de date tip cloud ce pot fi accesate de la distanță.

În acest proiect mi-am propus să utilizez tehnologia LoRa pentru transmiterea unor date. Pentru realizarea acestui proiect am folosit un modul Heltec Stick Lite , un senzor ultrasonic JSN-SR04T, un senzor de temperatura DS18B20, un senzor de temperatură și umiditate DHT22 pentru a demonstra cum se realizează transferul de date prin LoRa (868 MHz).

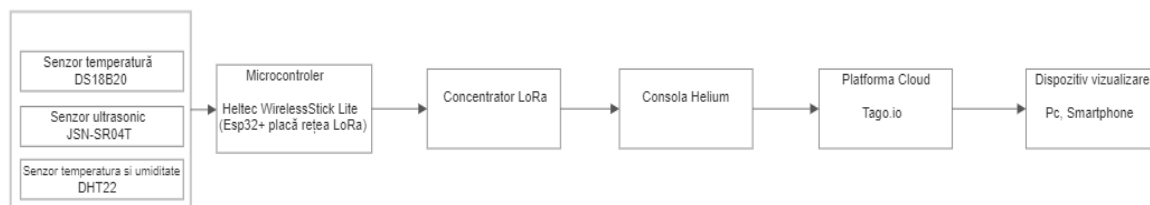


Fig. 2.1 Schema bloc a sistemului de monitorizare

Partea cea mai importantă, modulul Heltec Stick Lite, este o placă de dezvoltare alcătuită din microcontrolerul ESP32 și o placă de rețea LoRa, ce oferă posibilitatea conectării mai multor dispozitive datorită pinilor GPIO, de altfel acesta are deja implementat un chip pentru managementul încărcării și descărcării unui acumulator, și un tranzistor pentru controlul alimentării senzorilor. Aceasta se poate programa prin IDE-ul Arduino, Visual Studio, etc.

Am ales folosirea tehnologiei de transmisie radio deoarece este una dintre cele mai dorite moduri de conectivitate. Datele vor fi transmise cu ajutorul unei plăci de dezvoltare Heltec Wireless Stick Lite ce permite conexiuni LoRa și încărcarea datelor în consola Helium. De acolo datele vor fi redirecționate către platforma Tago.io unde vor fi memorate și vizualizate.

Modelul de utilizare de tip prototip pentru LoRaWAN este un dispozitiv cu baterie care transmite câțiva octeți de date la intervale de câteva minute până la o ora, sau la întâmplarea unui eveniment. Distanța de transmisie poate atinge cu ușurință peste 10 km, iar în condiții perfecte se poate depăși chiar și 100 km, care este mult mai mult decât distanța oferită de Bluetooth, WI-FI și unde milimetrice 4G sau 5G.

În funcție de aplicație, uneori datele mobile vor fi potrivite și alteori LoRaWAN sunt mai potrivite. Datele mobile au probleme cu propagarea semnalului, astfel pătrunderea undelor în structuri fizice precum clădirile constituie o reală problemă, problema ce va fi continuu agravată o dată cu trecerea la frecvențe mai înalte care vin cu tehnologia 5G.[13]

LoRaWAN deservește la o utilizare tehnică bine definită, care poate fi utilizat la mai multe aplicații. Unele din cele mai utilizate aplicații sunt contorizarea gazului, apei, rezervoarelor și parcării inteligente, unde dispozitivele trebuie să funcționeze pe teren

mai mulți ani , cu costuri reduse si să funcționeze pe baterii. Astfel instalarea dispozitivelor cu baterii în interiorul unei clădiri are costuri mai reduse decât instalarea de dispozitive cu fir sau folosirea unei tehnologii wireless.

Comparativ cu tehnicile de transmisiune radio obișnuite, tehnologia LoRa este utilizată în sistemele de senzori fără fir ,IoT și constituirea rețelelor de aparate inteligente, având aptitudinea în comunicarea datelor la distanțe mari .[14]

LoRa aparține stratului fizic și oferă comunicații de rază lungă , acesta ocupă nivelul 1 în modelul ISO OSI. LoRaWAN este un protocol de comunicație, ocupă nivelul 2 și 3 al modelului ISO OSI. LoRaWAN utilizează stratul fizic LoRa pentru a oferi comunicații fără fir în două sensuri.[6]

LoRa este o tehnologie bazată pe comunicații radio care a fost creată pentru a permite comunicarea cu rată de date scăzută să fie realizată pe distanțe lungi , potrivită pentru Internet of Things. Utilizează spectru radio nelicențiat în benzile industriale, științifice și medicale (ISM) pentru a permite o comunicare cu o suprafață largă de putere redusă între senzorii de la distanță și dispozitivele conectate la rețea.

LoRa transmite prin benzi de frecvență radio sub-1GHz fără licență, ca de exemplu 169 MHz, 433 MHz, 868 MHz (Europa) și 915 MHz (America de Nord). LoRa acceptă transmisii pe distanțe foarte lungi (mai mult de 10 km în zonele rurale) cu un consum redus de energie. Este cea mai predominantă alegere tehnologică pentru construirea rețelelor IoT la nivel Mondial. [15]

Spectrul de răspândire al rețelei LoRa este ideal pentru aplicațiile care necesită consum redus de energie și care necesită rate de date relativ mici (1 Mbit / s sau mai puțin).

Rețeaua LoRaWAN are la bază următoarele :

1. Noduri

Au rolul de a supraveghea și controla echipamentele date precum senzorii utilizați în parcuri ,aparatele folosite pentru a gestiona sistemele de iluminat și contoarele inteligente. În general sunt plasate la distanță.

2. Stația de baza(poarta / Gateway/ Concentrator) LoRa

Reprezintă dispozitivul care acceptă informații de la noduri prin rețeaua LoRa, pe care le transportă la serverul de rețea prin conexiune la internet. O stație de bază se poate conecta la server utilizând conexiunea IP. Datele trimise de senzori folosesc o structura comună, care se poate conecta la orice sistem de telecomunicații , chiar dacă este privat sau public.

3. Serverul rețelei

Rețeaua este administrată de către serverul de rețea LoRa. Acesta activează pentru a îndepărta pachetele replică, programele de identificare și a verifica viteza de trimitere a pachetelor.

4. Serverul aplicației

Datele primite de la noduri sunt redirecționate prin serverul rețelei la serverul aplicație. Acesta le expune într-un mod ce pune la dispoziție cele mai pertinente informații pentru client.

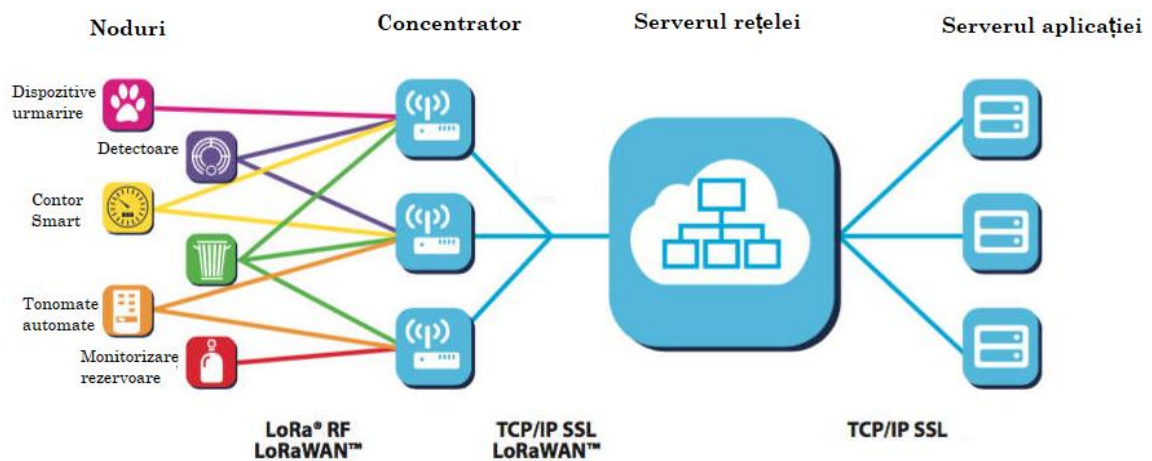


Fig. 2.2 Exemplu rețea LoRaWAN[15]

Capitolul 3. Implementarea sistemului de monitorizare

3.1 Implementarea hardware

Preluarea datelor de la senzori se realizează de către placa de dezvoltare pe care am numit-o Nod sau Transmițător cu ajutorul unor librării (lib NewPing.h DS18B20.h si lib DHT.h) . Așadar prin apelarea claselor și a funcțiilor din aceste librării, se preiau datele prin protocoale iar apoi sunt convertite folosind funcția CayenneLPP (sau mai bine spus împachetare). Astfel datele sunt pregătite de transmisie. Pentru a transmite datele prin modulul LoRa, acesta trebuie inițializat. Modulul se setează pentru a folosi frecvența de 868 MHz și puterea de transmisie de 20 Dbm, după care datele pot fi transmise către modulul LoRa.

Întreg sistemul are următoarea structură:

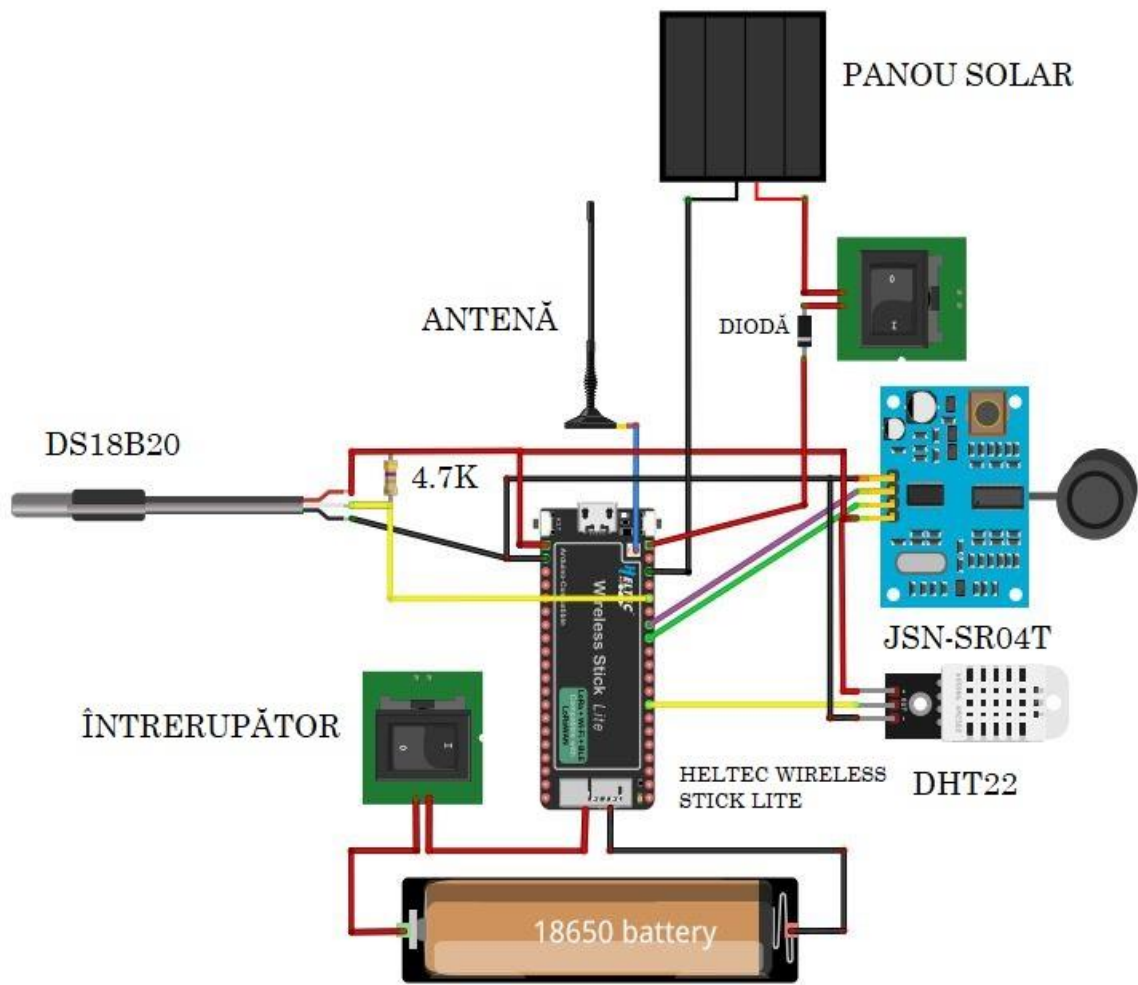


Fig.3.1.1 Schema de conexiuni electrice

3.2 Componente hardware

3.2.1 Placa de dezvoltare Heltec Stick Lite

Placa de dezvoltare Heltec Wireless Stick Lite are la bază un circuit incorporat microcontroller ESP32-pico și un circuit incorporat SX1276 utilizat pentru comunicarea LoRa. Se permite programarea cu medii de dezvoltare simplu de folosit, ca de exemplu Arduino IDE, Platform IO.

Microcontroller-ul are unul sau mai multe procesoare , are blocuri de memorie, intrări/ieșiri digitale, intrări/ieșiri analogice dar si alte periferice de bază. Microcontroller-ul este utilizat in mod obișnuit pentru control de baza, cum ar fi aplicații si sisteme incorporate.

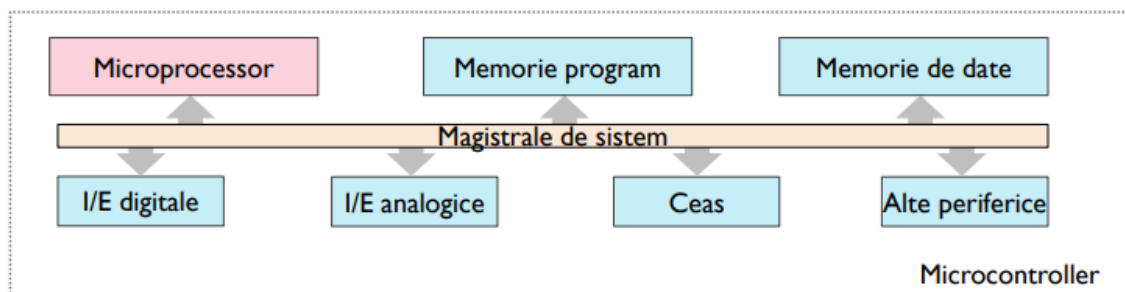


Fig. 3.2.1 Microcontroller[7]

Avantajele sistemelor incorporate prin folosirea microcontrollerlor sunt in primul rând cele financiare, deoarece costurile de producție, exploatare si întreținere sunt reduse. Un al avantaj este acela ca pot fi utilizate componente mai puțin costisitoare, iar software-ul face posibila asigurarea unui control complex.

Microprocesorul (MPU- Micro Processing Unit) este definit ca un nucleu de procesor unic, care accepta cel puțin preluarea, decodificarea si execuția instrucțiunilor. Procesorul acestora este realizat pe un singur circuit integrat pe scara foarte larga (VLSI- Very Large Scale Integrated)

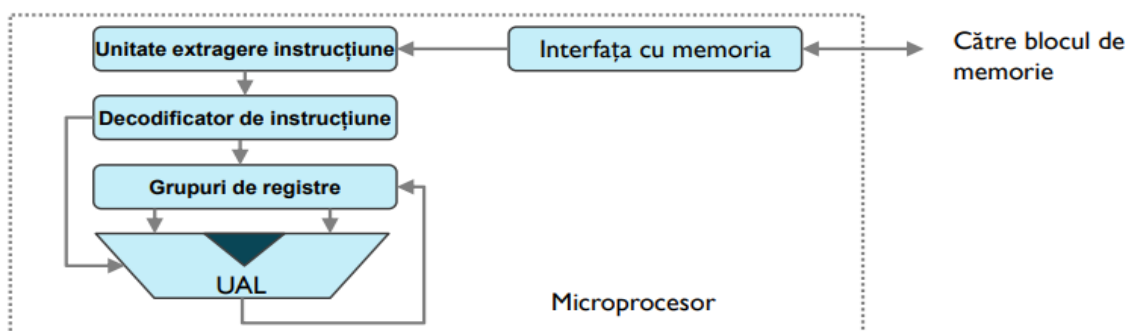


Fig. 3.2.2 Microprocesor[7]

Heltec Stick Lite este o placă de dezvoltare ce pune la dispoziție legături LoRa, Bluetooth, Wifi și pini GPIO, fiind perfectă pentru proiectul meu.



Fig 3.2.3 Placă de dezvoltare Heltec Stick Lite[20]

Specificații

✚ CPU

- Microprocesor Tensilica LX6 dual core cu 32 biți, 600 DMIPS
- Frecvență maximă 240Mhz
- Un coprocesor ULP suplimentar care monitorizează GPIO și canalele ADC
- Controlează cea mai mare parte a sistemului intern, consumând 25uA

✚ Memorie

- Ram 52KB
- Flash 4MB

✚ WIFI

- 802.11b/g/n 150mbps

✚ Bluetooth

- Versiunea 4.2 BR/EDR
- Tip energie scăzută și clasică

✚ RTC

- Rulează la 150KHz

✚ LoRa

- Putere maximă de emisie 20dB
- Mod comunicare nod la nod sau LoRaWAN
- Lăţime de bandă 125-500 kHz
- Sensibilitate până la 137 dBm

Securitate

- Permite SSL/TLS
- WPA

Criptare

- SHA
- AES
- DES
- MD5

Pini

ESP32 dispune de remaparea completă a funcţionalităţii pinului. Aceasta permite cartografierea perifericelor pe aproape orice pin GPIO disponibil. Imaginea de mai jos arată doar poziţiile implicite. De exemplu, maparea implicită are pinii SPI şi I2C care se suprapun, ceea ce înseamnă că ambele nu pot fi utilizate simultan.

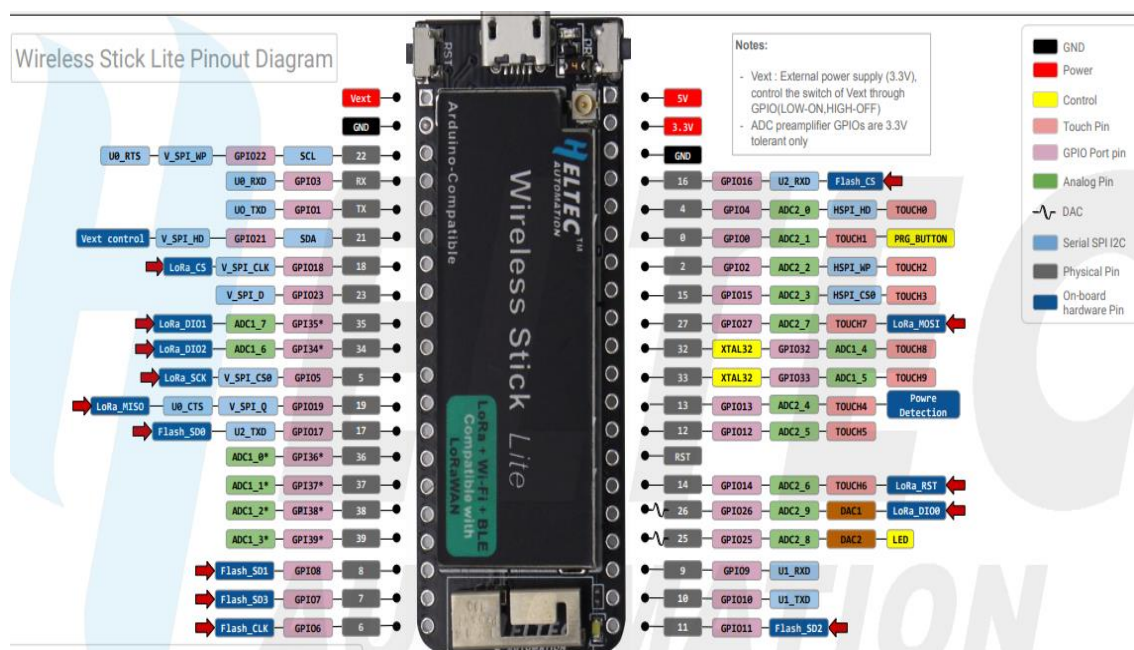


Fig. 3.2.4 Configurare pini Heltec Stick Lite[20]

Astfel microcontrolerul dispune de urmatorii tipi de pini:

- 3 x UART
- 2 x SPI
- 2 x I2C
- 1 x I2S
- 18 x 12 biti ADC input
- 2 x 8 biti DAC output
- 28 x GPIO
- 6 x GPI

Caracteristici electrice

- Tensiunea de intrare a alimentării : 3.3- 6 V
- Curent de ieșire al alimentării: maxim 500mA
- Tensiunea de iesire a alimentării: 3.3V
- Temperatura de operare: -40 85 °C

3.2.2 Senzor de temperatura DS18B20

Senzorul conține un termistor folosit pentru măsurarea temperaturii si un microcontroler pe 12 biți pentru a transmite valorile temperaturii ca date seriale. Astfel acest senzorul este unul de tip digital, trimițând date digitale prin pinul său de date.

Senzorul DS18B20 comunică printr-o magistrală cu 1 fir care necesită un singur fir de date (și masă) pentru comunicarea cu un microprocesor central.

Fiecare senzor DS18B20 are un cod serial unic pe 64 de biți, care permite mai multor senzori să funcționeze pe aceeași magistrală de 1 fir de date. Astfel, este ușor de folosit un microprocesor pentru a controla mai mulți senzori identici distribuiți pe o zonă largă.

Senzorul poate măsura temperatura de la -55 ° C la 125 ° C cu o exactitate de ± 0.5 ° C.

Senzorul de temperatură DS18B20 este unul de înaltă performanță ce poate da rezultatul cu o precizie de până la 12 biți. Puterea necesară este foarte mică, având nevoie de 1.5 mA maxim. Avantajul acestui model de senzor este că rezistă la apă, putând să măsoare temperatura apei sau să fie pus în exterior.



Fig. 3.2.5 Senzor de temperatura DS18B20[18]

Senzorul de temperatură DS18B20 are 3 conexiuni :

1. VCC/Firul roșu => Alimentarea cu tensiune de 3.3V până la 6V
2. DATA/ Firul galben => Rezultă temperatura prin intermediul datelor seriale
3. GND/Firul negru => Conectat la masa circuitului

3.2.3 Senzor de măsurare cu ultrasunete JSN-SR04T

Senzorii ultrasonici emit pulsații acustice scurte, de înaltă frecvență, la intervale de timp regulate. Acestea se propaga prin aer cu viteza sunetului. Dacă lovesc un obiect, acestea sunt reflectate înapoi ca semnale ecou la senzor, care calculează distanța până la obiect pe baza intervalului de timp dintre emiterea semnalului și receptarea ecoului.[17]

Senzorul ultrasonic JSN-SR04T prezintă câteva avantaje constructive importante față de senzorii convenționali, cel mai important lucru îl reprezintă faptul că este rezistent la apă. Un alt avantaj este că vine în două părți separate unul fiind transductorul care este elementul de detectare și celălalt fiind tabloul de comandă. Astfel că se poate pune elementul de detectare departe de toate circuitele de comandă.



Fig. 3.2.6. Senzor ultrasonic JSN-SR04T[16]

Acest senzor emite un sunet ultrasonic care are o frecvență de aproximativ 40 kHz, fiind o frecvență înaltă pe care oamenii nu o pot auzi. Senzorul are două părți active principale: un traductor care creează un sunet ultrasonic și altul care ascultă ecoul acestuia; care în cazul acestui senzor sunt într-un obiect comun.

Curentul consumat de senzor în modul de lucru este de 30 mA, iar în modul static este de 5mA.

Senzorul ultrasonic JSN-SR04T are patru pini :

1. VCC => Alimentarea cu tensiune de 3.3V până la 6V
2. Trig => Pinul de declanșare

3. Echo => Pinul de ecou

4. GND => Conectat la masa circuitului

Pentru utilizarea senzorului trebuie să conectăm pinii VCC și GND la o sursă de alimentare. Iar pinul de declanșare (Trig) și pinul ecou (Echo) trebuie legați la o ieșire/intrare digitală a microcontrolerului. Se aduce pinul de declanșare (Trig) în starea HIGH timp de cel puțin 10 μ s (microsecunde) și apoi se așteaptă până când pinul ecou (Echo) intră în starea HIGH. Perioada de timp până când pinul Echo intră în starea HIGH corespunde distanței pe care a parcurs-o sunetul ultrasonic. Cu cât răspunsul este mai rapid, cu atât un obiect este mai aproape de senzor.

Sunetul circulă cu aproximativ 340 de metri pe secundă. Asta înseamnă aproximativ 29,412 μ s (microsecunde) pe centimetru. Pentru a măsura distanța pe care a parcurs-o sunetul, folosim formula:

$$\text{Distanța} = (\text{Timp} \times \text{Viteza sunetului}) / 2$$

2- este în formulă deoarece sunetul trebuie să călătorească până la obiect dar și înapoi la senzor.

Modul simplu de a citi distanța în centimetri este să utilizăm formula:

$$\text{Distanță(în centimetri)} = ((\text{Timp(în microsecunde)} / 2) / 29).$$

3.2.4 SENZOR DE TEMPERATURĂ ȘI UMIDITATE DHT22

Senzorul conține un termistor folosit pentru măsurarea temperaturii, un senzor capacitiv pentru măsurarea umidității și un microcontroler pe 8 biți pentru a transmite valorile temperaturii și umidității ca date seriale. Astfel acest senzorul este unul de tip digital, trimițând date digitale prin pinul său de date.

Acesta are un cadru de transmitere pe 40 biți, de aici rezultând și rezoluția crescută a senzorului.

Senzorul este calibrat din fabrică și este ușor de interconectat cu alte microcontrolere. Senzorul poate măsura temperatura de la -40 ° C la 80 ° C și umiditate în intervalul 0% - 100% cu o exactitate de ± 1 ° C și $\pm 1\%$.

Senzorul DHT22 poate fi achiziționat fie ca senzor, fie ca modul. În orice caz, performanța senzorului este aceeași. Senzorul conține 4 pini din care doar 3 pini vor fi utilizați, în timp ce modulul conține direct doar 3 pini.

Alimentarea senzorului se face cu o tensiune de minim 3.3 V până la maxim 6V de curent continuu. Pentru a filtra undele se adaugă un condensator de 100nF între Vcc și GND.

Curentul consumat de senzor în momentul conversiei este de 2.5 mA.

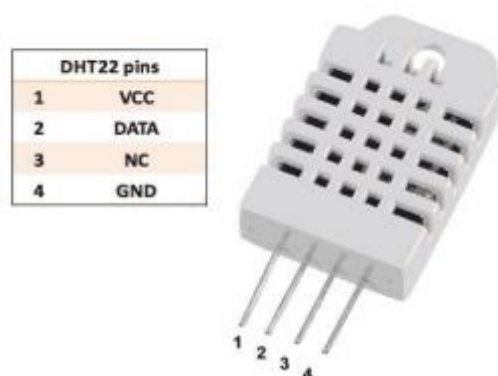


Fig. 3.2.7 Senzor de temperatura si umiditate DHT22 [21]

Senzorul de temperatură și umiditate DHT22 are patru pini :

1. VCC => Alimentarea cu tensiune de 3.3V pana la 6V
2. DATE => Rezultă temperatura și umiditatea prin intermediul datelor seriale
3. NC => Nu există conexiune , deci nu este utilizat
4. GND => Conectat la masa circuitului

3.2.5 Tranzistor AO3401

Tranzistorul este o componentă electronică a cărei rezistență electrică poate fi controlată cu ajutorul unui semnal electric numit semnal de comandă. Tranzistorul ne permite să controlăm un curent electric mare cu ajutorul unei cantități mici de energie electrică. Din acest motiv, una din principalele aplicații ale sale este cea de releu si amplificator.[12]

Tipul de tranzistor folosit este Mosfet cu canal tip P indus

Tensiunea de prag (V_{TH}) a acestui tranzistor este de aproximativ -0.9V.

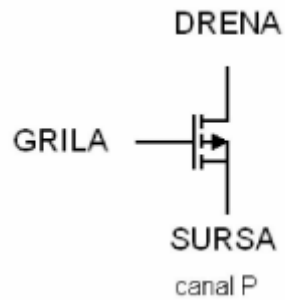


Fig.3.2.8 Tranzistor Mosfet canal P [12]

Condiția de funcționare în starea de blocare: $V_{gs} < V_{th}$ (în cazul nostru, când pinul 21 este în starea HIGH, $-3.3 < -0.6$)

$I_d = 0$ adică curentul tranzitoriu în regiunea de blocare este aproximativ 0

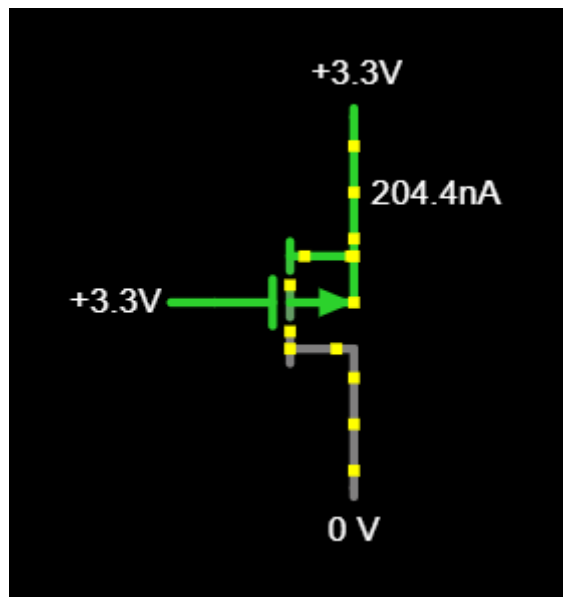


Fig. 3.2.9 Simulare funcționare tranzistor în regiunea de blocare

Condiția de funcționare în regimul activ: $V_{gs} > V_{th}$ și $V_{ds} < v_{gs} - V_{th}$ (în cazul nostru, când pinul 21 este în starea LOW, $3.3 > -0.9$ și $0 < 0 + 0.9$)

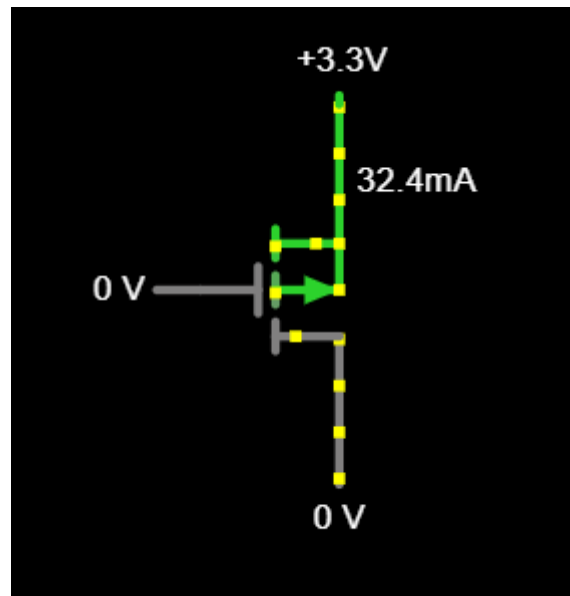


Fig. 3.2.10 Simulare funcționare tranzistor în regimul activ

Controlul pinului Vext

Pinul Vext este sursa de alimentare pentru toți senzorii externi. Acesta este controlat de GPIO21. Așadar:

- Dacă GPIO 21 este în starea logica LOW atunci Vext este ON
- Dacă GPIO 21 este în starea logica HIGH atunci Vext este OFF

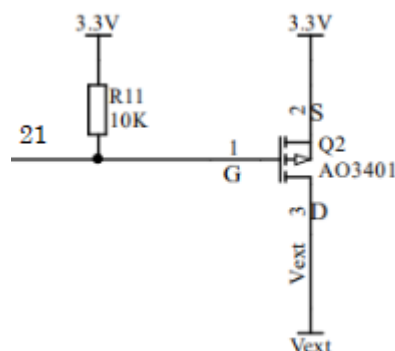


Fig. 3.2.11 Circuitul integrat Vext

3.2.6 Convertor analog-numeric(ADC)

În electronică, un convertor analog-digital este un sistem care convertește un semnal analog într-un semnal digital.

Microcontrolerul ESP32 are două ADC-uri (Convertoare analog numerice) de tip SAR(Registre de aproximări succesive) pe 12 biți și acceptă măsurători pe 18 pini analogici.

ADC-urile ESP32 au 12 biți de rezoluție, astfel încât intervalul total de citire a ADC-urilor merge de la 0 la 4095 ((2¹²)-1). Majoritatea ADC-urilor sunt asemănătoare, așa că putem crea următoarea ecuație:

$$\frac{\text{rezolutia ADC} - \text{ului}}{\text{Voltajul sistemului}} = \frac{\text{valoarea ADC} - \text{ului}}{\text{valoarea tensiunii citită}}$$

Conversiile analog-digitale depind de tensiunea de funcționare a sistemului, deoarece folosim un ESP32 cu tensiunea de sistem de 3.3V, rezulta ca putem măsura în mod direct tensiuni până la valoarea de 3.3 volți.

În cadrul proiectului se dorește măsurarea tensiunilor de alimentare al sistemului, adică tensiunea bateriei Li-Ion (ce ajunge până la 4.2V) dar si tensiunea prin cablul Usb (5V), tensiuni ce sunt mult mai mari decât tensiunea sistemului 3.3V, astfel va fi necesara utilizarea unui divizor de tensiune.

Un divizor de tensiune este un circuit liniar pasiv care produce o tensiune de ieșire (V_{out}), care este o fracțiune din tensiunea de intrare (V_{in}), folosind 2 rezistente electrice.

Microcontrolerul folosit de mine, Heltec Stick Lite, are deja implementat un divizor de tensiune, folosit pentru a măsura tensiunea electrica a alimentarii. Acesta este format din 2 rezistente, de 220k respectiv 100k Ohm, ca in figura de mai jos.

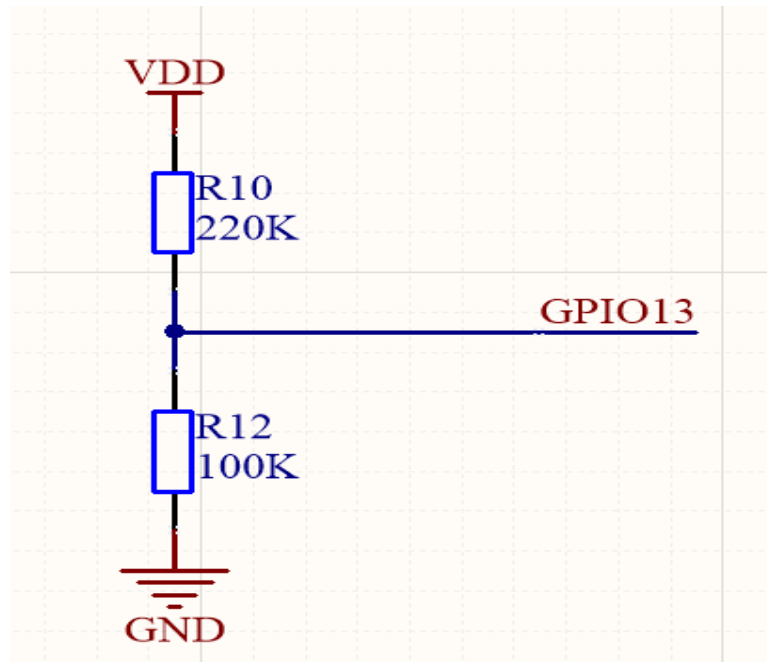


Fig. 3.2.12 Divizor de tensiune existent pe Heltec Stick Lite[26]

Astfel, teoretic:

$$V_{gpoi13} = \frac{R_{12}}{R_{10} + R_{12}} * VDD$$

VDD- tensiunea de alimentare

Vgpoi13-tensiunea la ieșirea din divizor aplicata de pinul 13

R10=220K

R12=100K

Domeniul de măsurare pentru ADC al ESP32 este între 0.15 și 3.3V, iar rezoluția acestuia fiind de 12 biți adică 4095, rezulta următoarea relație matematică între Tensiunea pe pinul 13 și valoarea returnată de ADC:[26]

$$V_{gpoi13} = \frac{ADC * (3.3 - 0.15)}{4095}$$

Combinând toate formulele rezultă:

$$\frac{100}{220 + 100} * VDD = 0.000769 * ADC$$

Astfel relația matematică dintre valoarea tensiunii de intrare și valoarea returnată de ADC este:

$$VDD = 0.00246 * ADC$$

3.2.7 Dioda 1N5819

Dioda semiconductoră este cel mai simplu dispozitiv semiconductor, aceasta are două terminale. Cele două terminale ale diodei poartă denumirea de anod, respectiv catod.[12]



Fig. 3.2.12 Dioda 1N4819 [32]

Cel mai important aspect al diodelor este reprezentat de diferența valorilor de deschidere între polarizarea directă și inversă. Aceste valori având denumirile de tensiune de prag și tensiune de străpungere.

Pentru această diodă, valoarea tensiunii de prag este 0.6 volți, iar valoarea tensiunii de străpungere este de 28 volți.[32]

Astfel, în timpul parcurgerii sarcinilor electrice de la anod la catod, dioda va conduce curentul electric dacă diferența de potențial este mai mare decât 0.6 volți. Iar când sarcinile electrice se deplasează de la catod la anod, dioda va conduce curentul electric dacă diferența de potențial este mai mare decât 28 volți.

Curentul maxim ce poate fi parcurs de această diodă este de 1 amper.

3.2.8 Panoul solar

Panoul solar este un generator de energie electrică, ce transformă energia luminoasă direct în energie electrică.



Fig. 3.2.13 Panou solar

Acesta funcționează prin efectul fotoelectric extern, în care fotonii (lumina) vor ceda energie electronilor periferici, acești electroni eliberați vor forma un curent electric, fiind extrași din material.[33]

Caracteristici:

- Tensiune 5V
- Intensitate 0.4A
- Putere maximă 2W

Dacă panoul nu debitează curent și este legat la un acumulator, polaritatea tensiunii la borne s-ar schimba și acesta s-ar defecta. Acest lucru este împiedicat de dioda de protecție care oprește curentul în acest caz.[27]

3.3 Realizarea montajului

Montajul este format din 2 părți: rezervorul și incinta microcontrolerului.

Ambele elemente au fost construite din placi transparente de plexiglas transparent, am ales acest element deoarece este ieftin și rezistent.



Fig. 3.3.1 Plexiglas

Incinta microcontrolerului

Am pornit de la ideea ca microcontrolerul, antena, acumulatorul, senzorul DHT22, întrerupătorul, rezistența, dioda, panoul solar, împreună cu toate conexiunile electrice de la senzori să fie înglobate într-o incintă fiind astfel ferite de praf și apă.

Circuitul electric l-am realizat folosind o placa de prototipare PCB, am ales varianta aceasta in schimbul unui breadbord pentru a mă asigura ca legăturile electrice rămân fixe iar astfel nu voi avea întreruperi.

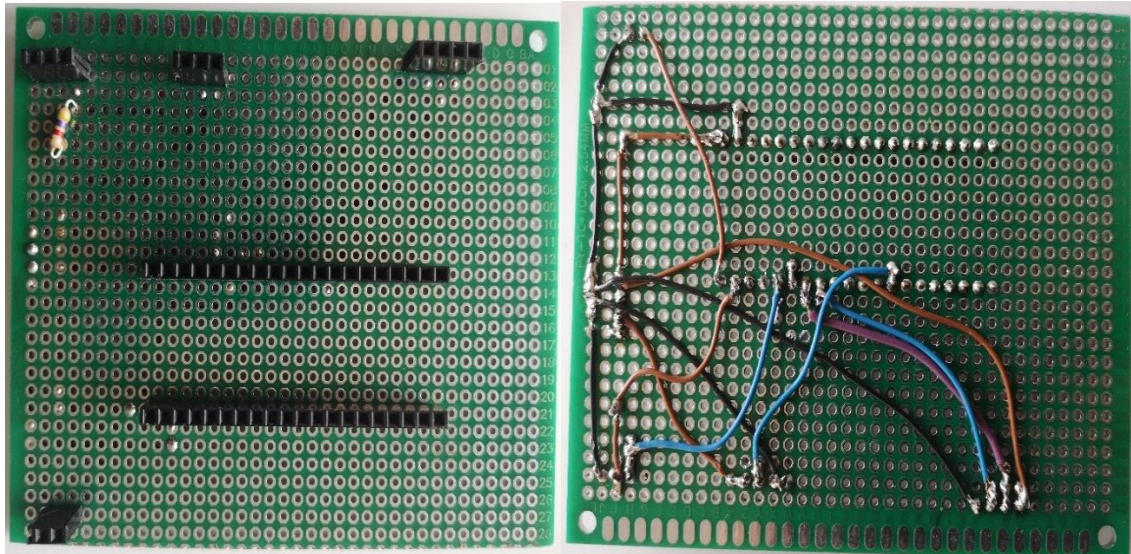


Fig. 3.3.2 Placa de protipare

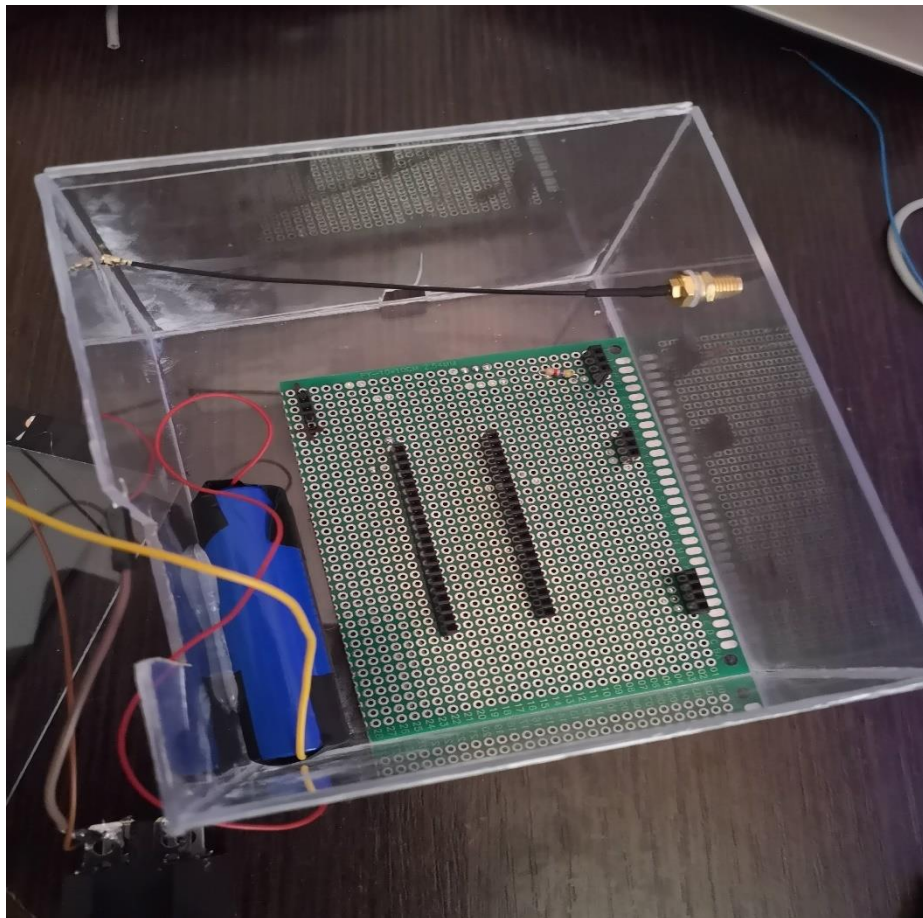


Fig. 3.3.3 Montarea componentelor

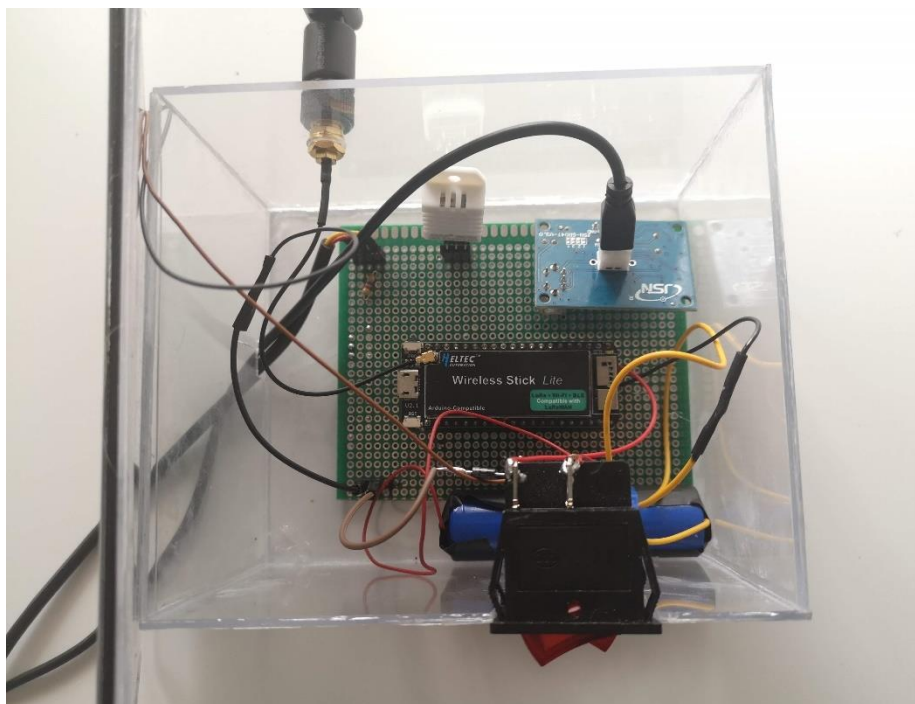


Fig. 3.3.4 Montarea componentelor

Iar montat complet incinta microcontrolerului arată așa:



Fig. 3.3.5 Incinta microcontrolerului



Fig. 3.3.6 Incinta microcontrolerului

Rezervorul

Rezervorul este construit din plăci de plexiglas transparente lipite între ele, pentru a fi ușor de distins nivelul lichidului din interior. Ca forma geometrica aceasta este o prisma patrulateră regulată. Dimensiunile acestuia sunt de aproximativ $L=25\text{ cm}$, $l=25\text{ cm}$, $H=50\text{ cm}$.

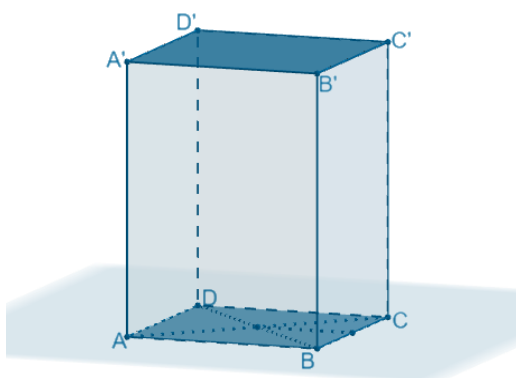


Fig. 3.3.8 Prisma patrulateră regulată

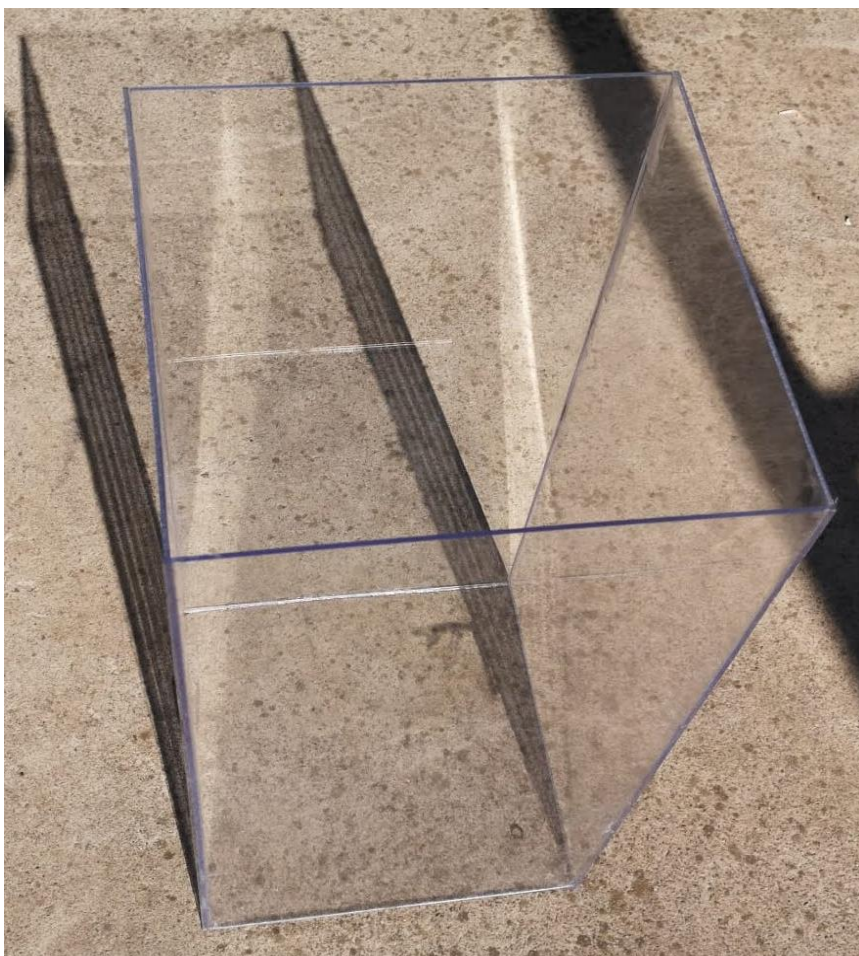


Fig.3.3.7 Rezervorul construit

Pentru a grada acest rezervor am folosit un recipient etalon de 500ml.



Fig. 3.3.9 Recipient etalon

Astfel am creat o scală gradată pe rezervor pentru măsurarea cu ușurință a volumului din acesta.

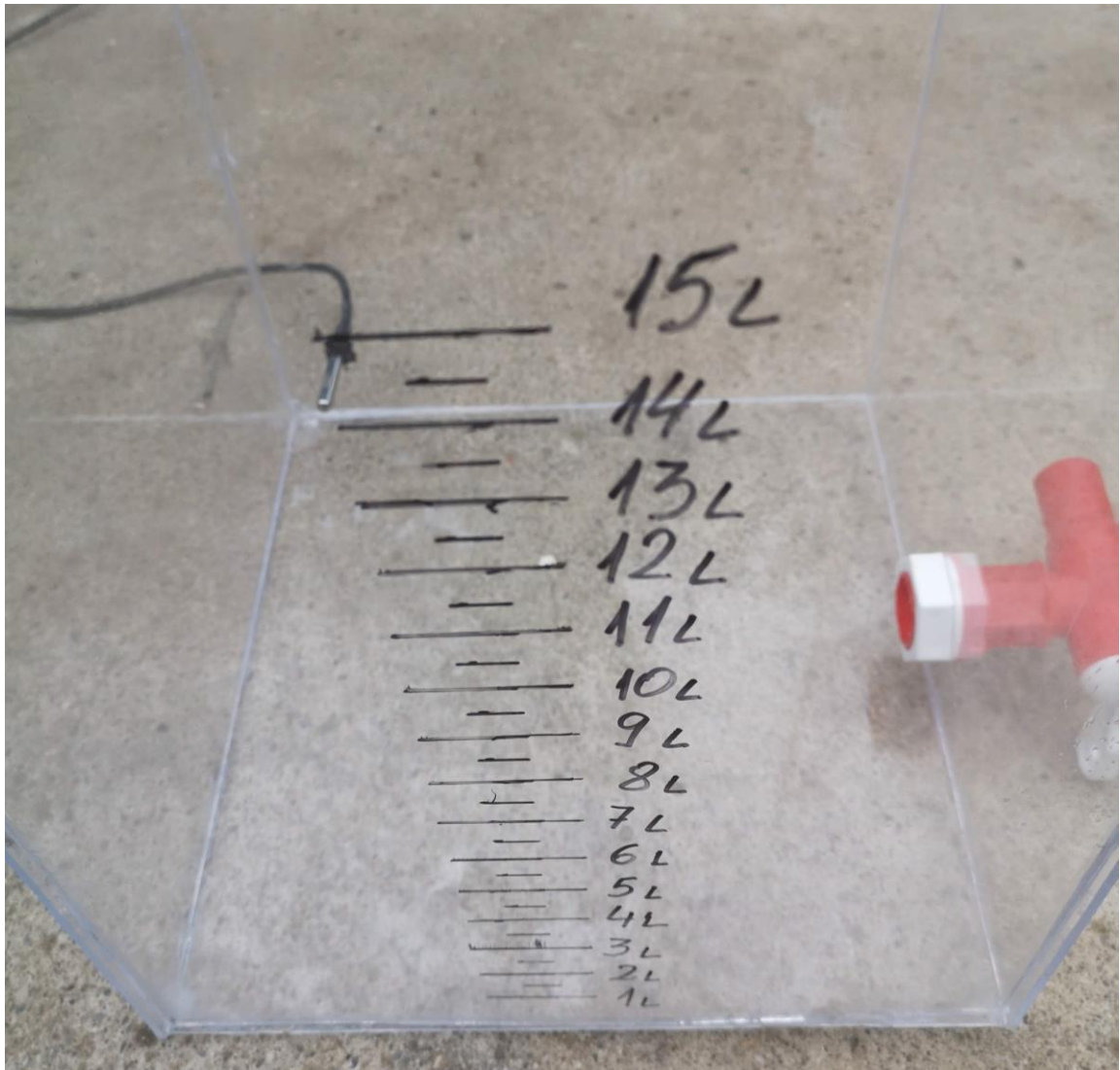


Fig. 3.3.10 Rezervor gradat

Odată cu gradarea rezervorului am memorat și valorile returnate de senzorul ultrasonic. Astfel am tras următoarele concluzii:

- când rezervorul era gol valoarea returnată de senzorul ultrasonic era de 49,3 cm.
- deci relația matematică de calcul pentru nivelul de apă din rezervor este:
$$\text{nivelul apei} = 49,3 - \text{distanța returnată de senzorul ultrasonic}$$

Deoarece rezervorul nu este lipit perfect drept și existau unele abateri, iar liniaritatea senzorului ultrasonic putea să nu fie perfectă, am optat pentru crearea unei

funcții monovariabile de tip polinom. Astfel a fost necesar să notez perechile de valori dintre cantitatea de apă și nivel, rezultând următorul tabel:

Nr citire	Volum(ml)	Nivel(cm)
1	0	0
2	1000	2.1
3	2000	3.76
4	3000	4.95
5	4000	7.13
6	5000	8.78
7	6000	10.43
8	7000	12.11
9	8000	14.39
10	9000	16.04
11	10000	17.65
12	11000	19.37
13	12000	21.4
14	13000	22.6
15	14000	24.25
16	15000	25.79

Coeficienții polinomului de regresie sunt calculați cu ajutorul metodei celor mai mici pătrate. Această metodă presupune minimizarea funcției obiectiv:

$$f_{ob}(A) = \sum_{j=1}^m \left[y_j - (a_0 + a_1 x_j + a_2 x_j^2 + \dots + a_n x_j^n) \right]^2$$

Folosind un algoritm de aproximare a funcției discrete prin regresie polinomială au rezultat următorii coeficienți:

```
*****

Gradul polinomului de regresie = 4

Abaterea standard = 1.0431536912254365E+002

*****

Coeficientii polinomului de regresie

a0 = -1.3333385E+002
a1 = 6.381823E+002
a2 = -9.39092E+000
a3 = 4.0166E-001
a4 = -4.561E-003|

*****
```

Fig.3.3.11 Rezultat rulare program

Deci confirm rezultatului rulării algoritmului, rezultă următorul polinom de gradul 4:

$$y = -0.00456x^4 + 0.40166x^3 - 9.39092x^2 + 638.1823x - 133.33385$$

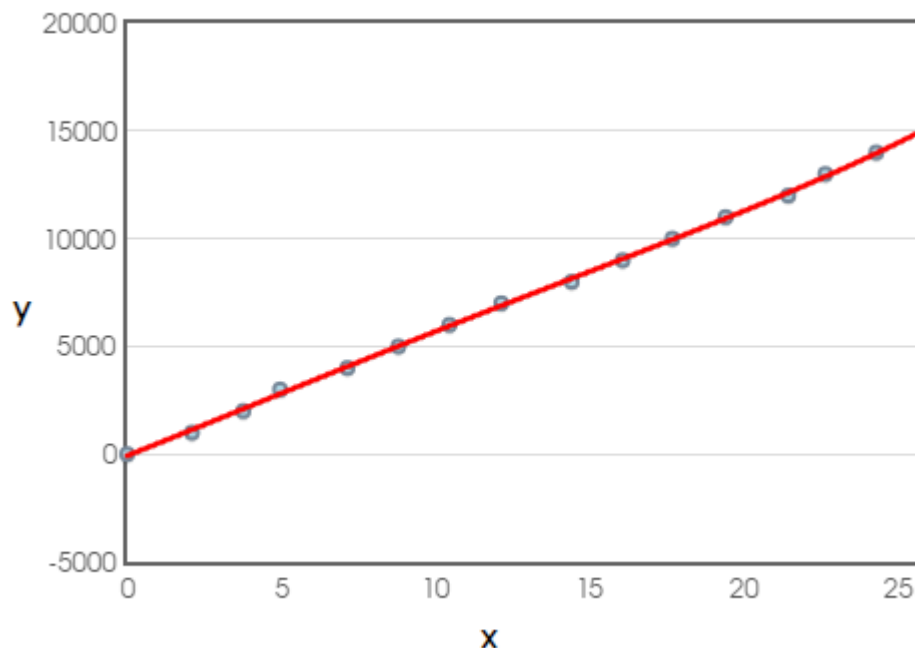


Fig.3.3.12 Reprezentare grafică polinom

3.4 Blockchainul Helium si configurarea consolei Helium

Helium a creat o rețea LoRaWAN la nivel global. Acesta este în continuă creștere și oferă zone mari de acoperire prin utilizarea a ceea ce ei numesc Hotspot-uri. În limbaj LoRa aceste Hotspoturi sunt de fapt concentratoare /gateway LoRa. Ele asigură transportul de date pentru senzorii noștri.

Presupunând că avem deja acoperire cu rețea LoRaWAN, putem spune că o mare parte din munca grea este eliminată, adică aceea de a construi o astfel de rețea LoRa. Dar totuși dacă nu avem acoperire în zona senzorilor, ne putem construi sau achiziționa propriul Hotspot ce poate fi adăugat în rețeaua Helium. Există un stimulent pentru a folosi aceste dispozitive, prin faptul că se va câștiga o sumă de bani pentru fiecare pachet de date ce este preluat de acesta.

Pentru zona Ploiești și împrejurimi, conform figurii de mai jos, se poate observa un număr substanțial de concentratoare LoRa conectate la rețeaua Helium. Astfel suntem aproape siguri ca dispunem de acoperire completă în întreaga zonă pentru proiectul nostru, dar și oricare alt dispozitiv LoRaWAN conectat prin rețeaua Helium.

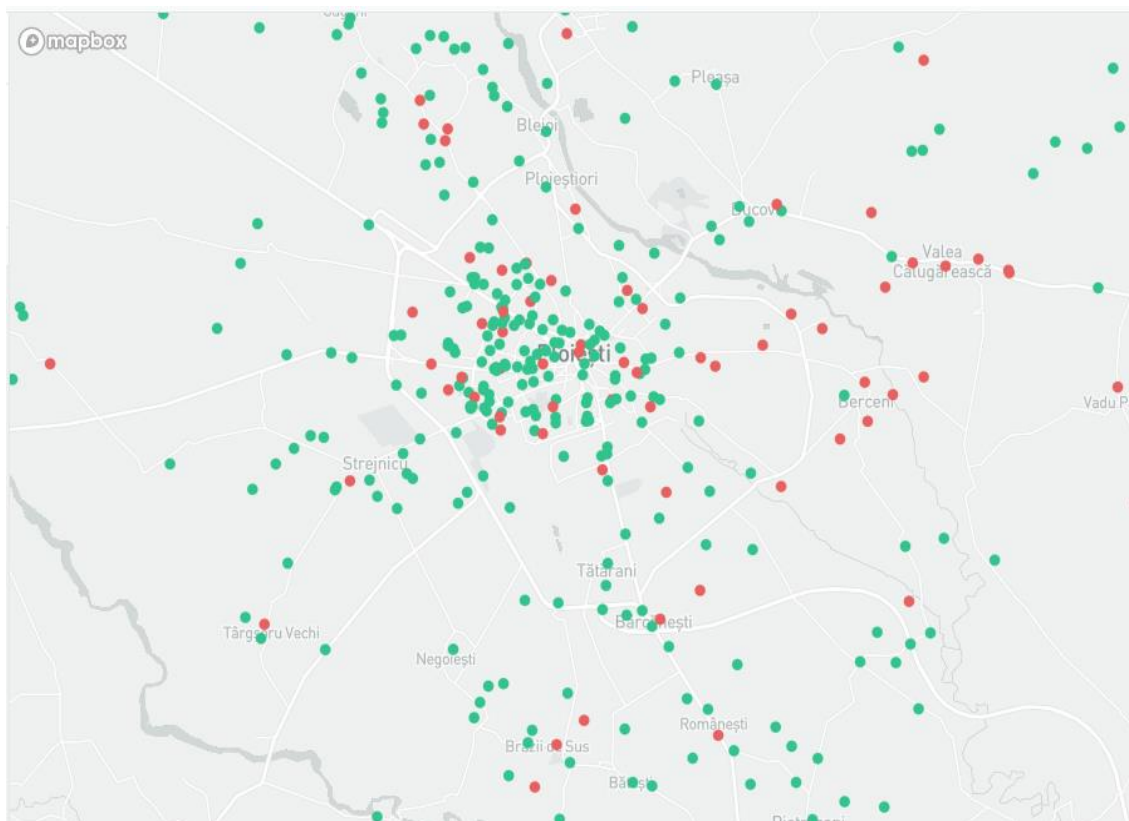


Fig.3.4.1 Locația concentratoarelor LoRa în zona Ploiești

DC-urile (data credit) sunt folosite pentru a plăti fiecare pachet trimis în rețeaua Helium, inclusiv Join-urile inițiale.

Fiecare 24 de octeți trimiși într-un pachet costă 1 DC = 0,00001 USD.

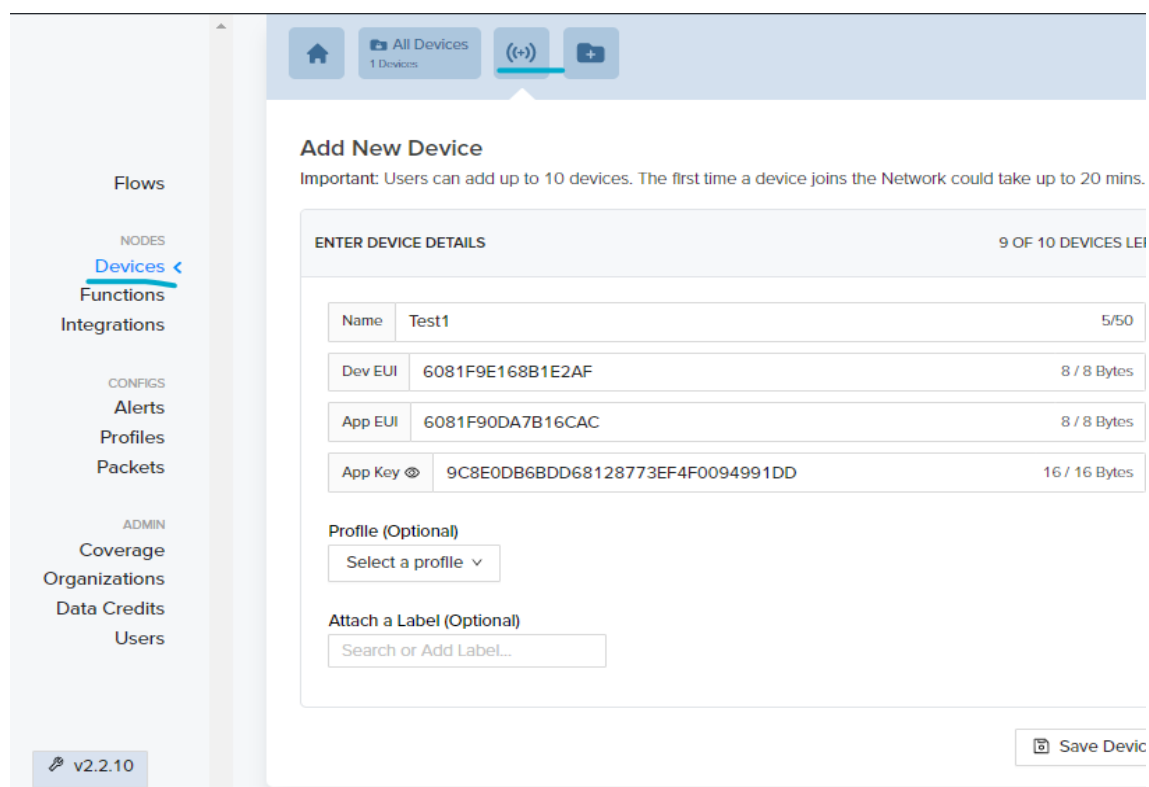
Spre deosebire de alte rețele tradiționale, rețeaua Helium nu are abonament lunar, astfel se plătesc doar datele pe care le utilizăm. Această tranzacție are loc între posesorul senzorului ce transferă date și posesorul concentratorului LoRa. Pașii pe scurt ar fi: Un concentrator LoRa primește un pachet și determină cui îi aparține, după se adresează serverul LoRaWAN Helium și i se transferă pachetul, iar după se face schimbul de credite.

3.4.1 Crearea contului pentru consola

Pe link-ul <https://console.helium.com/> va fi necesar sa creăm un nou cont de consola. Fiecare cont nou vine cu 10.000 de credite de date (DC credit) gratuit. Ca exemplu , dacă trimitem un pachet pe oră, timp de peste un an vom putea folosi aceste credite gratuite.

3.4.2 Adăugare dispozitiv

După crearea contului, trebuie sa apăsăm pe butonul “Devices” iar după pe “+” pentru a adăuga un nou dispozitiv. Introducem un nume pentru dispozitiv si notam valorile pentru “Dev EUI”, “App EUI” și pentru “App Key”. Iar după se apasă pe butonul de salvare, in plus trebuie sa nu facem publică cheia dispozitivului (App Key).



The screenshot shows the 'Add New Device' interface in the Helium console. The left sidebar contains navigation links: Flows, NODES (with 'Devices' selected), Functions, Integrations, CONFIGS (with Alerts, Profiles, and Packets), and ADMIN (with Coverage, Organizations, Data Credits, and Users). The main content area is titled 'Add New Device' and includes an important note: 'Important: Users can add up to 10 devices. The first time a device joins the Network could take up to 20 mins.' Below this is a form titled 'ENTER DEVICE DETAILS' with a progress indicator '9 OF 10 DEVICES LEFT'. The form contains four input fields: 'Name' (Test1, 5/50), 'Dev EUI' (6081F9E168B1E2AF, 8 / 8 Bytes), 'App EUI' (6081F90DA7B16CAC, 8 / 8 Bytes), and 'App Key' (9C8E0DB6BDD68128773EF4F0094991DD, 16 / 16 Bytes). There are also optional sections for 'Profile (Optional)' with a dropdown menu and 'Attach a Label (Optional)' with a search input. A 'Save Device' button is at the bottom right. The version 'v2.2.10' is shown in the bottom left corner.

Fig. 3.4.2 Adăugare dispozitiv

3.4.3 Creare funcție decodare

Programul pe care îl vom încărca pe microcontroler utilizează formatul de împachetare pachete de date numit CayenneLPP. Acesta este un format foarte comun pentru IOT și majoritatea platformelor îl au implementat .

Cei de la myDevices au creat Cayenne Low Power Payload (LPP) care oferă o modalitate convenabilă și ușoară de a trimite date prin rețele LoRaWAN. Cayenne LPP

respectă restricția privind dimensiunea sarcinii utile și permite dispozitivului să trimită mai multe date de la senzori simultan.

În plus, Cayenne LPP permite dispozitivului să trimită date diferite ale senzorului în cadre diferite. Pentru a face acest lucru, fiecare dată de senzor trebuie să conțină doi octeți de prefix:

- Canal de date: identifică în mod unic fiecare senzor din dispozitiv în cadre, de exemplu. „senzorul numărul 1”
- Tip de date: identifică tipul de date din cadru, de ex. "temperatura".

Structura format Cayenne LPP [31]

1 Bit	1 Bit	N biți	1 bit	1 bit	M biți	...
Canal de date 1	Tip de date 1	Date 1	Canal de date 2	Tip de date 2	Date 2	...

Exemplu de dispozitiv cu 2 senzori de temperatura

Sarcina utila(in hexazecimal)	03 67 01 10 05 67 00 FF	
Canal de date	Tip	Valoare
03 ⇒ 3	67 ⇒ Temperatură	0110 = 272 ⇒ 27.2°C
05 ⇒ 5	67 ⇒ Temperatură	00FF = 255 ⇒ 25.5°C

Așadar, în consola Helium, trebuie să facem clic pe butonul “Functions”, apoi pe „+” pentru a adăuga o nouă funcție. Trebuie să introducem un nume pentru funcția noastră, funcția este de tip decodor, iar formatul este CayenneLPP. După facem clic pe Salvare funcție. Totodată există posibilitatea de creare al unui decodor personalizat, pentru mărirea siguranței.

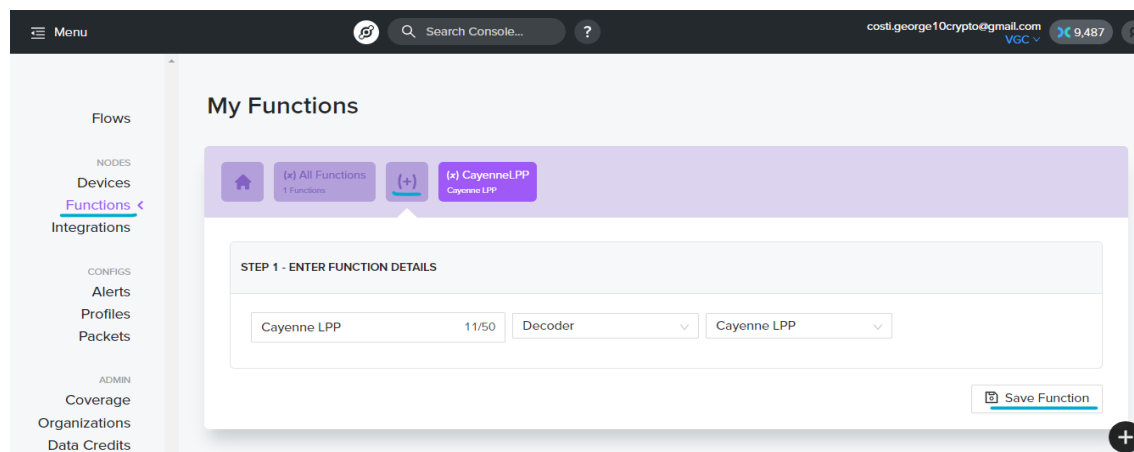


Fig. 3.4.3 Adăugare funcție

3.4.4 Aplicația de vizualizare

Există multe platforme/aplicații IOT diferite pentru a trimite și vizualiza datele.

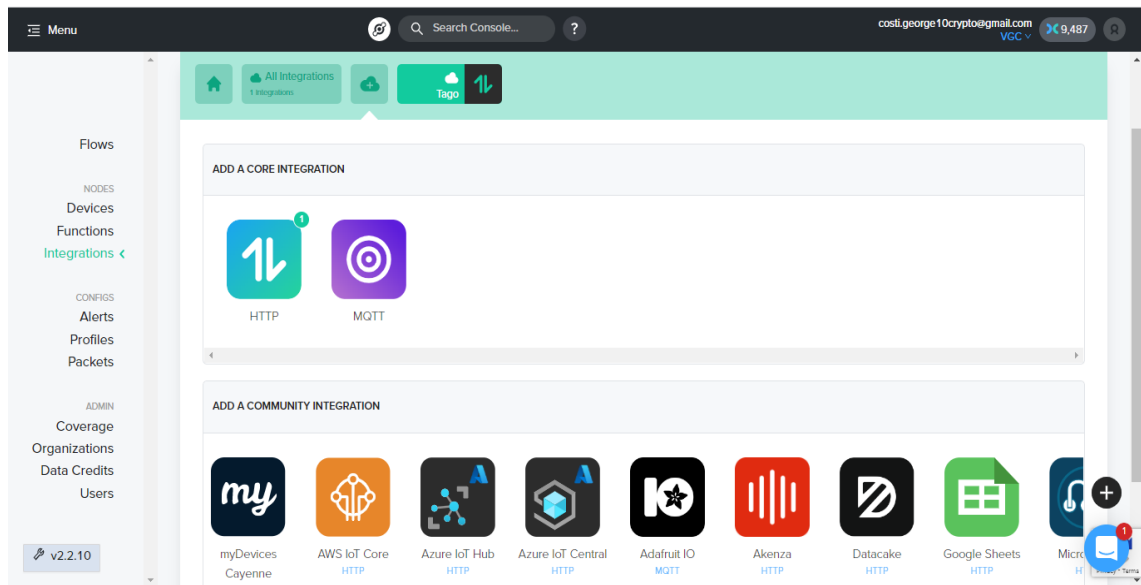


Fig. 3.4.4 Aplicații vizualizare

Cele mai folosite platforme sunt:

- Datacake: are interfața ușor de personalizat și multe opțiuni de configurare, dar pentru contul gratuit datele se stochează doar o săptămână
- AdafruitIO: cel mai ușor de configurat, dar pentru contul gratuit nu dispune de alerte prin email
- myDevices Cayenne: simplu de configurat dar și cu puține instrumente de vizualizare
- TagoIO: pentru contul gratuit datele sunt memorate timp de 30 zile, iar alertele prin email/sms sunt incluse

Se alege platforma dorită, în cadrul acestui proiect voi folosi TagoIO, deoarece mi se pare că este cea mai potrivită acestui tip de monitorizare, fiind necesare alertele. Așadar se apasă pe butonul “TagoIO” și se adaugă ca implementare.

Următorul pas este să conectăm fluxul de date. Se apasă pe butonul “Flows” iar după se adaugă dispozitivul, decodorul și aplicația de vizualizare. După acestea trebuie conectate, asemănător ca în figura de mai jos.

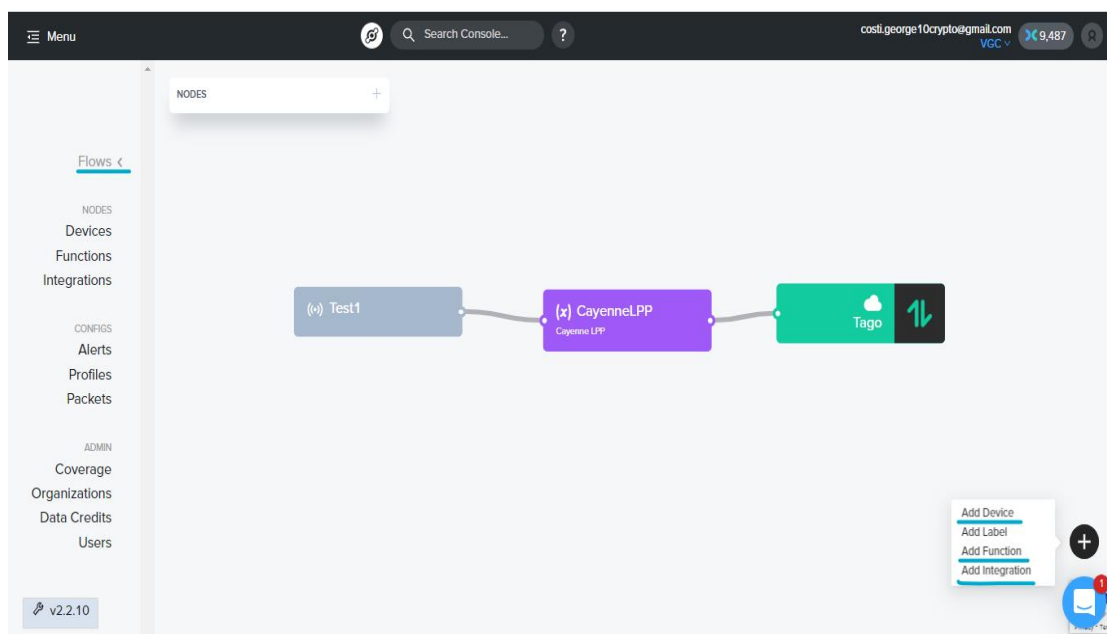


Fig. 3.4.5 Creare flux date

3.5 Configurarea si programarea microcontrolerului

Pentru a programa microcontrolerul am folosit mediul de dezvoltare PlatformIO.

Mediul de dezvoltare PlatformIO conține un editor de text pentru scrierea codului, o zonă de mesaje, o consolă de text, o bară de instrumente cu butoane pentru funcții comune și o serie de meniuri. Acesta se conectează la hardware-ul folosit pentru a încărca programe și a comunica cu acestea. Programele scrise în PlatformIO se numesc schițe. PlatformIO funcționează ca o extensie ce trebuie instalat în programul de baza, Visual Studio Code.[29]

Visual Studio Code, este un editor de cod sursă realizat de Microsoft pentru Windows, dar și alte platforme. Caracteristicile includ suport pentru depanare, completarea inteligentă a codului, evidențierea sintaxei, fragmente, etc.

Înainte de a încărca o schiță trebuie să ne asigurăm că am selectat tipul de microcontroler corect pe care vrem să încercăm schița, dar și portul la care este conectat acesta.

PlatformIO este folosit în locul Arduino IDE, deoarece este mai flexibil, mai puternic și suportă mai bine dezvoltarea multiplatformă.

Un program implicit conține două funcții:

- `setup()` – aceasta este apelată atunci când o schiță pornește rularea. Este folosită pentru inițializarea variabilelor, inițializarea pinilor și a bibliotecilor. Această funcție rulează o singură dată, după fiecare pornire sau resetare a plăcii.
- `loop()` - după crearea unei funcții `setup()` , funcția `loop()` rulează ca o buclă infinită permitând programului schimbul și răspunsul pentru a controla microcontrolerul.

Programul de plecare folosit de mine este acesta:

https://github.com/Chiumanfu/LMIC-node_Sensor-for-Helium-Network

A fost necesar să configurez programul , pentru tipul meu de microcontroler, tipul de placă de rețea , frecvența LoRa folosită , dar cel mai important , citirea datelor noilor senzori folosiți în cadrul proiectului , prin adăugarea de noi biblioteci și memorarea/prelucrarea lor în diferite variabile pentru a fi trimise prin rețeaua LoRaWAN.

Pentru selectarea plăcii:

În fișierul `platformio.ini` selectăm exact placa acceptată din lista disponibilă , în cazul nostru pentru Heltec Wireless Stick Lite variabila `default_envs` va fi:

```
[platformio]
default_envs =

    heltec_wireless_stick_lite
```

Fig.4.5.1 Secvență program selectare microcontroler

Selectarea plăcii de rețea și frecvența

Tot în fișierul `platformio.ini` selectăm tipul plăcii de rețea SX1276 , iar frecvența folosită pentru Europa este de 868Mhz, astfel liniile de cod vor fi ca în următoarea poză:

```

build_flags =
    -D ARDUINO_LMIC_PROJECT_CONFIG_H_SUPPRESS

    -D DISABLE_PING
    -D DISABLE_BEACONS

    -D CFG_sx1276_radio=1
    -D USE_ORIGINAL_AES

    -D CFG_eu868=1

```

Fig. 4.5.2 Secvență program selectare placă rețea

O altă setare importantă este **intervalul de timp la care dorim sa retransmitem datele**, eu am optat pentru 5 minute, adică 300 secunde. Astfel variabila “DO_WORK_INTERVAL_SECONDS” va avea valoarea 300, ca in figura de mai jos:

```

[common]
monitor_speed = 115200
build_flags =
    -D DO_WORK_INTERVAL_SECONDS=300

```

Fig. 4.5.3 Secvență program interval retransmitere date

Pentru a se face conexiunea cu consola Helium , este necesar sa introducem **cheile unice ale dispozitivului**, mai exact DevEui(identificator dispozitiv), AppEui(identificator aplicație),AppKey(cheia aplicației) , despre care am vorbit mai sus la etapa de configurare a Consolei Helium.

Aceste numere trebuie convertite in format hexazecimal in tip Lsb sau Msb , exact cum este specificat in fișierul cu chei. Consola Helium ne vine in ajutor având integrată si partea de conversie, astfel aceste șiruri de numere se pot copia cu ușurință direct de pe site si adăugate in fișierul cu numele “lorawan-keys.h” ca in figura următoare:

```

// Keys required for OTAA activation:

// End-device Identifier (u1_t[8]) in lsb format
#define OTAA_DEVEUI 0xEF, 0x90, 0x7B, 0x1A, 0x74, 0xF9, 0x81, 0x60

// Application Identifier (u1_t[8]) in lsb format
#define OTAA_APPEUI 0xAC, 0x6C, 0xB1, 0xA7, 0x0D, 0xF9, 0x81, 0x60

// Application Key (u1_t[16]) in msb format
#define OTAA_APPKEY 0xAC, 0x91, 0x91, 0x57, 0xAD, 0xA7, 0x33, 0x0E, 0x2B, 0xA6, 0x2F, 0xF0, 0x8A, 0x38, 0: ),

```

Fig.4.5.4. Fisierul Lorawan-keys

Fișierul numit “LMIC-node.cpp” este programul principal , ce se ocupa de colectarea si prelucrarea datelor de la senzori si transmiterea acestora prin rețeaua LoRa către un concentrator.

Pentru a folosi senzorii sunt necesare librări speciale , dar si declararea pinilor unde aceștia sunt conectați. Astfel sunt necesare următoarele biblioteci: NewPing.h (pentru senzorul ultrasonic) , Ds18B20.h(pentru senzorul temperatura apa) , DHT.h(pentru senzorul DHT22) si CayenneLPP.h (pentru împachetarea datelor).

Următorul pas este de definire al pinilor, astfel pentru controlul tranzistorului, explicat la capitolul 3.2.5, îl vom asocia pinului 21, senzorul DS18B20 este conectat la pinul 4, senzorul DHT22 conectat la pinul 12, iar pentru senzorul ultrasonic : pinul declanșator conectat la pinul 2 iar cel de ecou conectat la pinul 15 al microcontrolerului

```
// Librăriile folosite
#include <CayenneLPP.h>
#include <DHT.h>
#include <DS18B20.h>
#include <NewPing.h>

#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"

// Definirea pinilor
#define VEXT_PIN 21
#define DHTPIN 12
#define DHTTYPE DHT22

#define TRIGGER_PIN 2 // Pinul declanșator pentru senzorul ultrasonic.
#define ECHO_PIN 15 // Pinul de ecou pentru senzorul ultrasonic.
#define MAX_DISTANCE 300 // Distanța maximă pe care dorim să o măsurăm (în centimetri).

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); //Funcția setup NewPing pentru pini și distanța maximă.

DS18B20 ds(4);

DHT dht(DHTPIN, DHTTYPE);

CayenneLPP lpp(30);
```

Fig. 3.5.5 Declarare biblioteci și pini

Cum am mai explicat în capitolul teoretic ,modelul de utilizare de tip prototip pentru LoRaWAN este un dispozitiv cu baterie care transmite câțiva octeți de date la intervale de câteva minute, asta înseamnă ca echipamentul trebuie să fie cât mai economic, astfel plin alimentarea senzorilor doar în momentul citirii valorilor se poate reduce cantitatea de energie consumată de aceștia.

Astfel, in subprogramul de citire si trimitere date, numit “processWork”, înainte de citirea valorilor returnate de senzori , trebuie sa ii alimentam. Transistorul funcționează ca o poarta NOT, astfel pentru alimentarea senzorilor, pinul de comanda al tranzistorului trebuie sa fie in starea logica LOW iar după am setat un delay pentru a ne asigura ca senzorii s-au alimentat si au pornit.

```
pinMode(VEXT_PIN, OUTPUT);  
digitalWrite(VEXT_PIN, LOW);  
delay(1000);
```

Fig. 3.5.6 Pinul Vext pornit

Iar dupa ce am citit valorile senzorilor oprim alimentarea acestora.

```
digitalWrite(VEXT_PIN, HIGH); //oprirea alimentarii senzorilor
```

Fig. 3.5.7 Pinul Vext oprit

Pentru senzorul DS18B20 valoarea temperaturii returnata de acesta va fi memorata in variabila de tip float “temp_apa”

```
// Citirea datelor de la senzorul DS18B20  
Serial.print(ds.getTempC());  
Serial.println(F(" Temperatura apă"));  
float temp_apa = ds.getTempC();
```

Fig. 3.5.8 Secvență program citire date de la DS18B20

Pentru senzorul DHT22 valoarea temperaturii si umidității returnată de acesta va fi salvată in variabilele de tip float “temp_aer” si “humi”.

```
// Citirea datelor de la senzorul DHT22  
Serial.print("DHT22 - \t");  
float temp_aer = dht.readTemperature();  
float humi = dht.readHumidity();
```

Fig. 3.5.9 Secvență program citire date de la DHT22

Pentru senzorul ultrasonic valoarea distanței returnată de acesta va fi salvata in variabila de tip float “dist”, dar înainte de asta a fost necesar sa fie prelucrată. A fost necesară prelucrarea datelor deoarece uneori senzorul returna valoarea 0. Dar si pentru a

crește precizia măsurării am optat pentru o medie aritmetică a 3 valori citite corect. Astfel a rezultat programul de mai jos:

```
// Citirea datelor de la senzorul ultrasonic
int ok=0;
float sumdist=0,dist=0;
for (size_t i = 0; i <= 30; i++)
{
    float x=sonar.ping_cm();
    Serial.print(x);
    Serial.println(F(" Distanța măsurată"));
    if (x!=0)
    {
        sumdist=sumdist+x;
        ok++;
    }
    if (ok>=3)
    {
        break;
    }
    delay(150);
}
if (ok>0)
{
    dist=sumdist/ok;
}
else
{
    dist=0;
}
```

Fig. 3.5.10 Secvență program citire și prelucrare date senzor ultrasonic

Pentru convertorul analog numeric valoarea returnată de acesta va fi salvată în variabila “voltaj”

```
// Citirea datelor de la convertorul analog-numeric pentru aflarea voltajului
double voltaj =ReadVoltage(37)*3.306617;
```

Fig. 3.5.11 Citire voltaj alimentare

Conform explicațiilor din capitolul 3.2.6, pinul 37 este conectat la un divizor de tensiune, folosind funcția “ReadVoltage” de mai jos, am realizat conversia din ADC-uri în volți.

```
double ReadVoltage(byte pin){
    double reading = analogRead(pin); // Voltajul de referința este 3.3v, deci valoarea maxima este pt 3.3v = 4095 in intervalul 0 to 4095
    Serial.print(reading);
    Serial.println(F(" Battery Voltage ADC"));
    return -0.0000000000000016 * pow(reading,4) + 0.000000000118171 * pow(reading,3)- 0.000000301211691 * pow(reading,2)+ 0.001109019271794 * reading + 0.03;
}
```

Fig. 3.5.12 Secvență program polinom calcul voltaj

Când rezervorul era gol valoarea returnată de senzorul ultrasonic era de 49,3 cm, astfel am folosit următoarea ecuație pentru a afla nivelul din rezervor. In plus am mai adăugat o condiție pentru a evita situațiile cu nivel negativ.

```
float nivel_apa=49.3-dist; //calculul nivelului apei din rezervor
if(dist>49.3)
    nivel_apa=0;
```

Fig. 3.5.13 Secvență program calcul nivel apă

Iar pentru a calcula volumul apei din rezervor am folosit un polinom de gradul 4 ce are ca valoare de intrare nivelul apei.

```
//calculul volumului folosind un polinom de gradul 4 cu valoarea nivelului ca data de intrare
float volum= (-0.00456 * pow(nivel_apa,4) + 0.40166 * pow(nivel_apa,3) - 9.39092 * pow(nivel_apa,2)+ 638.1823 * nivel_apa - 133.33385) * 0.001;
if(volum<0)
    volum=0;
```

Fig. 3.5.14 Secvență program polinom calcul volum

După ce s-au memorat noile valori, urmează etapa de împachetare și trimitere date, dar înainte de asta se vor trimite toate datele prin portul serial, lucru util pentru procesul de depanare al programului, pentru a verifica datele citite de microcontroler cu datele recepționate în baza de date.

```

// Afisarea datelor de la toți senzorii
Serial.print(temp_aer);
Serial.println(F(" °C temperatură aer DHT22"));
Serial.print(humi);
Serial.println(F(" percent umiditate DHT22"));
Serial.print(dist);
Serial.println(F(" Distanță"));
Serial.print(voltaj);
Serial.println(F(" Voltaj baterie"));
Serial.print(temp_apa);
Serial.println(F(" Temperatură apă"));
Serial.print(nivel_apa);
Serial.println(F(" Nivel apa"));
Serial.print(volum);
Serial.println(F(" Volum apa"));

```

Fig. 3.5.15 Secvență program afișare date

După urmează efectiv procesul de împachetare și trimitere date folosind standardul CayenneLPP.

```

//Crearea pachetului CayenneLPP
lpp.reset();
lpp.addAnalogInput(0, temp_aer);
lpp.addRelativeHumidity(1, humi);
lpp.addAnalogInput(2, dist);
lpp.addAnalogInput(3, voltaj);
lpp.addAnalogInput(4, temp_apa);
lpp.addAnalogInput(5, nivel_apa);
lpp.addAnalogInput(6, volum);
uint8_t payloadLength = (lpp.getSize());
uint8_t* payloadbuffer = (lpp.getBuffer());
scheduleUplink(fPort, payloadbuffer, payloadLength);

```

Fig. 3.5.16 Secvență program împachetare și trimitere date

Cam acestea au fost cele mai importante configurări realizate în program, acum vom trece la etapa de configurare a aplicației de vizualizare folosite, respectiv TagoIO.

3.6 Configurare TagoIO

3.6.1 Creare cont

Pe link-ul <https://tago.io/> va fi necesar sa creăm un nou cont pentru a ne putea importa datele. După crearea contului, trebuie sa apăsăm pe butonul din stânga “Devices” iar după pe “+” pentru a adăuga un nou dispozitiv.

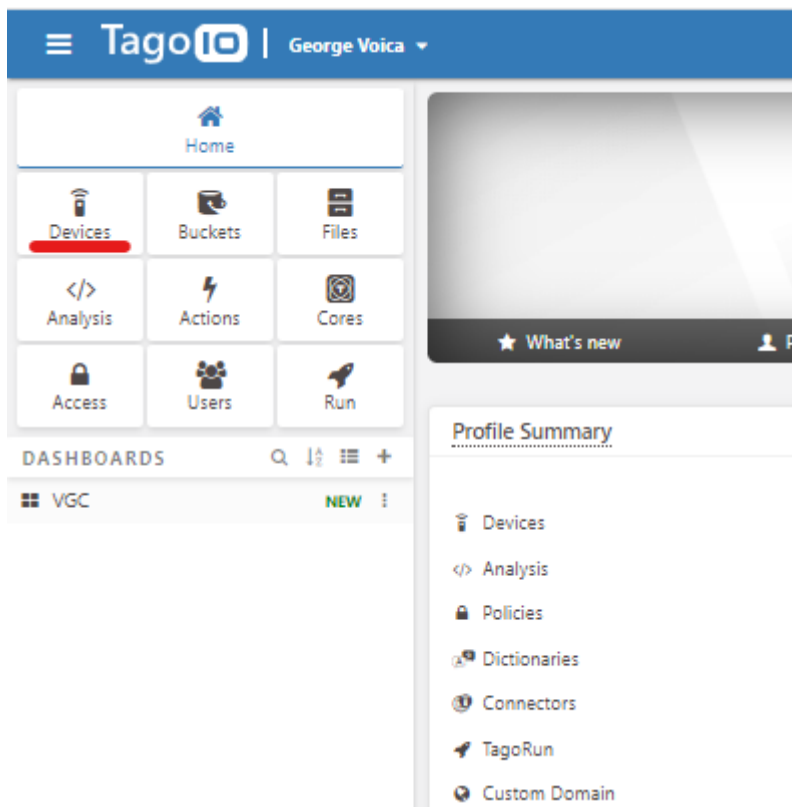


Fig. 3.6.1 Platforma Tago.io

După trebuie să generăm un cod de autorizare introducând orice nume în câmpul Nume și facem clic pe butonul de generare.

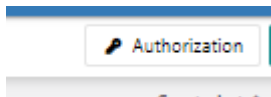


Fig. 3.6.2 Buton autorizare

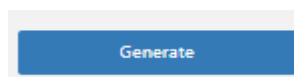


Fig. 3.6.3 Buton generare cod

Acum pur și simplu copiem codul de autorizare făcând clic pe pictograma de copiere:



Fig. 3.6.4 Buton copiere

Următoarea etapă este de a ne întoarce în Consola Helium, și a accesa meniul “Integrations” mai exact TagoIO, pentru a introduce codul de autorizare. Meniul trebuie să fie ca cel de mai jos, se vor introduce și restul datelor iar la sfârșit se apasă butonul “Add integration”.

Step 2 - Endpoint Details

Enter TagoIO Authorization Token:

atceb30f4fbad34253a6259c696c9a08a6

Fig.3.6.5 Adăugare cod autorizare

3.6.2 Adăugare dispozitiv

Facem clic pe butonul „Devices”, apoi pe “Add devices” în colțul din dreapta sus. Iar apoi căutăm dispozitivul numit „Custom Helium”.

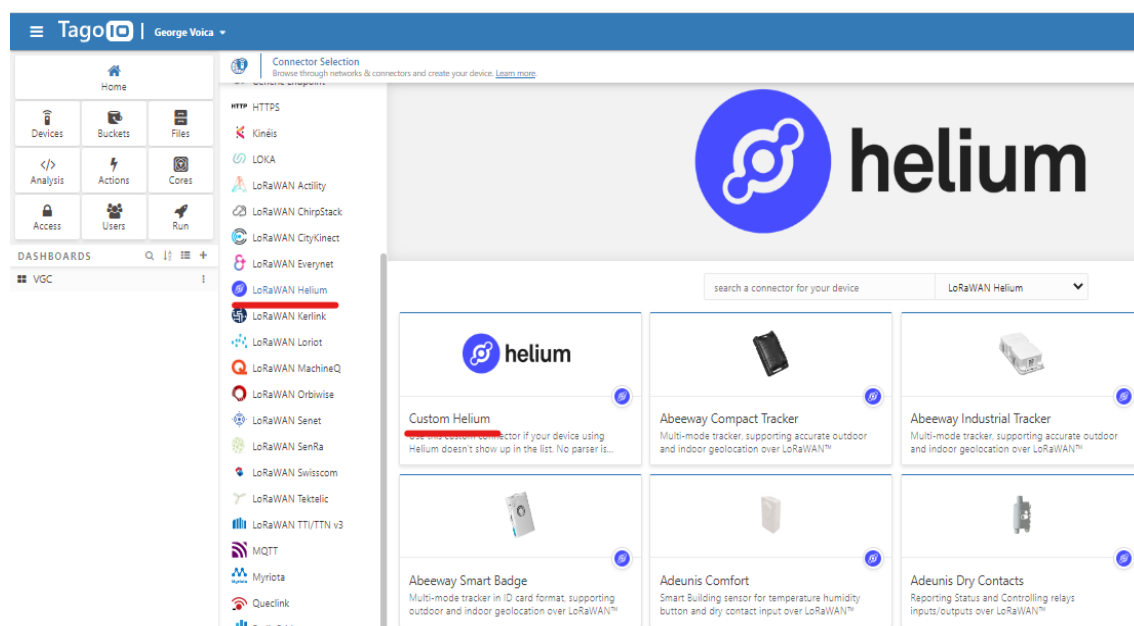


Fig. 3.6.6 Adăugare dispozitiv

Trebuie să denumim dispozitivul, apoi introducem DevEui (identificator dispozitiv), cel ce apare în Consola Helium și despre care am vorbit la capitolul 3.4.2. După facem clic pe “create device”.

3.6.3 Creare Dashboard

Creăm un nou Dashboard apăsând pe + , apoi adăugăm ce instrumente de vizualizare dorim.

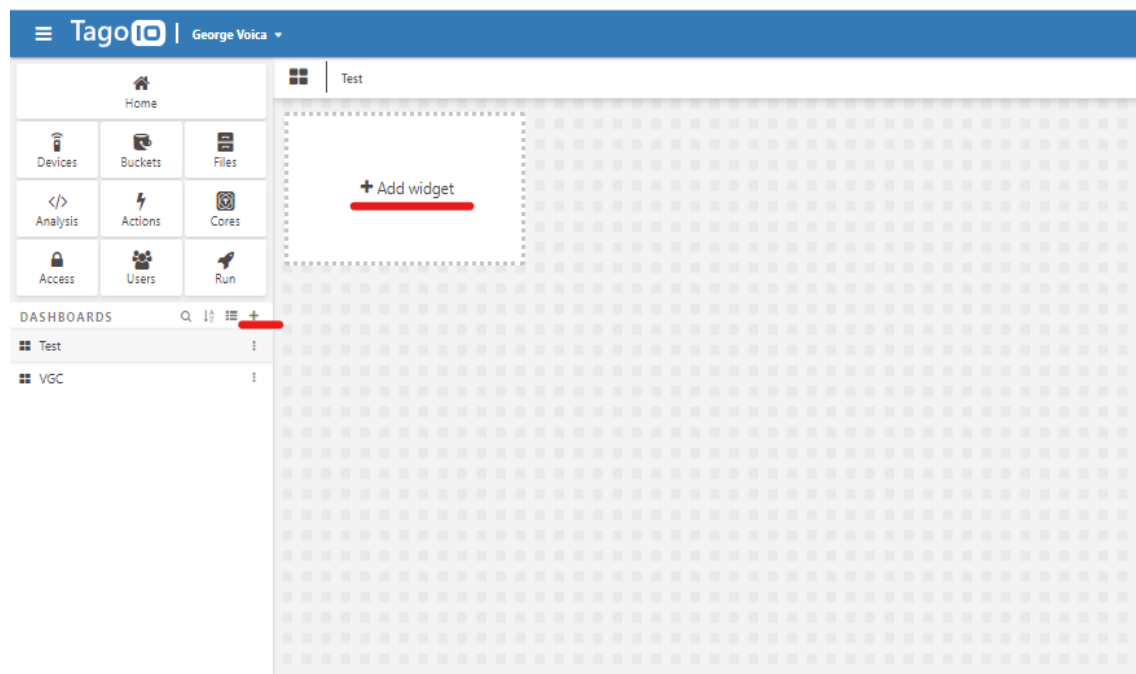


Fig. 3.6.7 Dashboard

Platforma ne pune la dispoziție o varietate largă de alegeri, ca de exemplu:afișaje, grafice, diagrame, linii, ceasuri, manometre, cilindrii, harți, imagini, videoclipuri; ce pot fi personalizate, colorate după bunul plac.

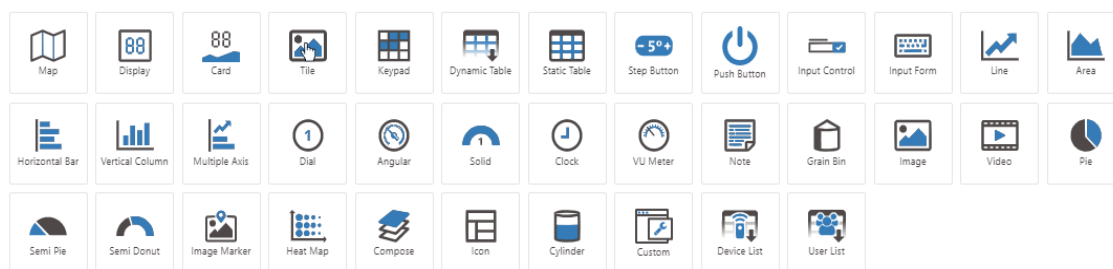


Fig.3.6.8 Listă instrumente vizualizare

După ce alegem un instrument, trebuie să îl asociem unei variabile de date. Deschidem meniul de date iar la “Device” selectăm dispozitivul nostru, care în mod implicit are numele “Device #1”, iar la caseta “variable” selectăm variabila pe care dorim să o vizualizăm.

In cadrul programului folosit de mine, datele sunt in aceeași ordine ca la împachetarea datelor, respectiv:

- Variabila 0: temperatură aer
- Variabila 1: umiditate
- Variabila 2: distanță
- Variabila 3: voltaj
- Variabila 4: temperatură apă
- Variabila 5: nivel apa
- Variabila 6: volum apa

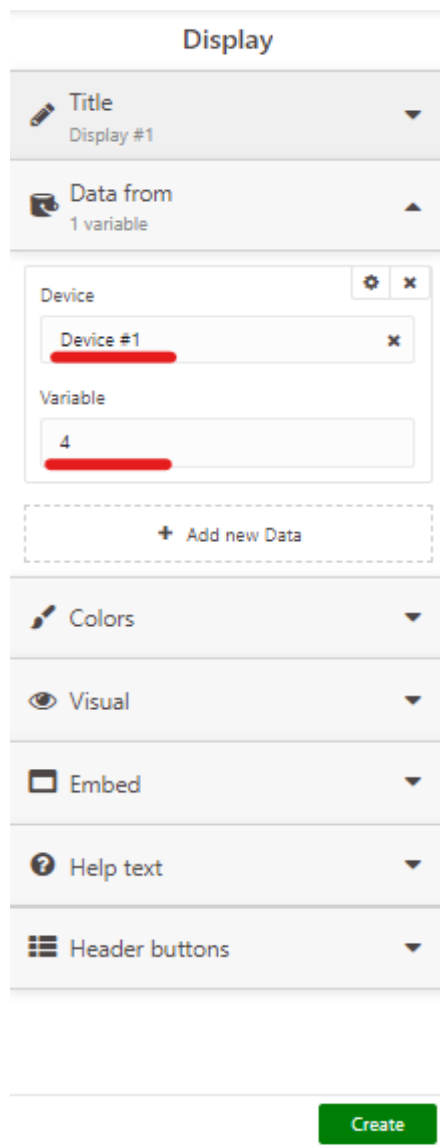


Fig. 3.6.9 Date dispozitiv

După apăsarea butonului de creare ar trebui să vedem că datele apar pe afișaj, ca in figura de mai jos:

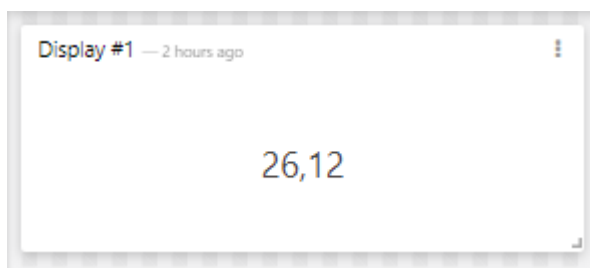
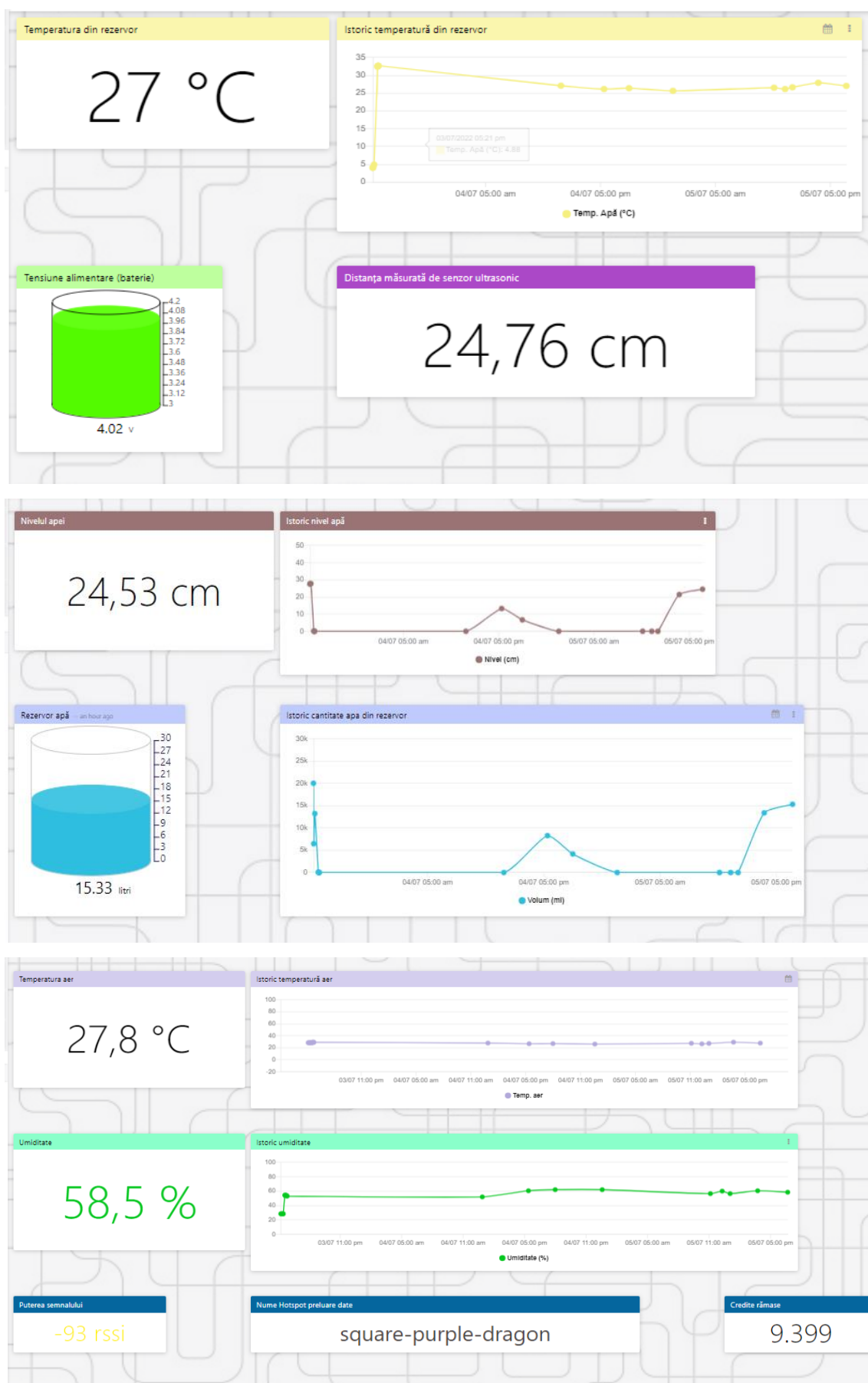


Fig. 3.6.10 Afișaj temperatură

După terminarea configurărilor, platforma mea arată așa:



Acesta este link-ul public pentru sistemul meu de monitorizare:

<https://admin.tago.io/public/dashboard/622b44f98defcf001801d5b2/2fa4552a-f055-4703-b73b-1321f48081c3>

3.6.4 Creare alerte

O alta caracteristică importantă a unei platforme de monitorizare este aceea de configurare alerte și alarme. Platforma Tago.io ne pune la dispoziție în mod gratuit alerte prin email și sms, dar într-un număr lunar limitat ce pot fi extinse prin migrarea la un abonament plătit.

Astfel am un exemplu de configurare a unei alerte prin email pentru avertizarea nivelului scăzut de lichid din rezervor.

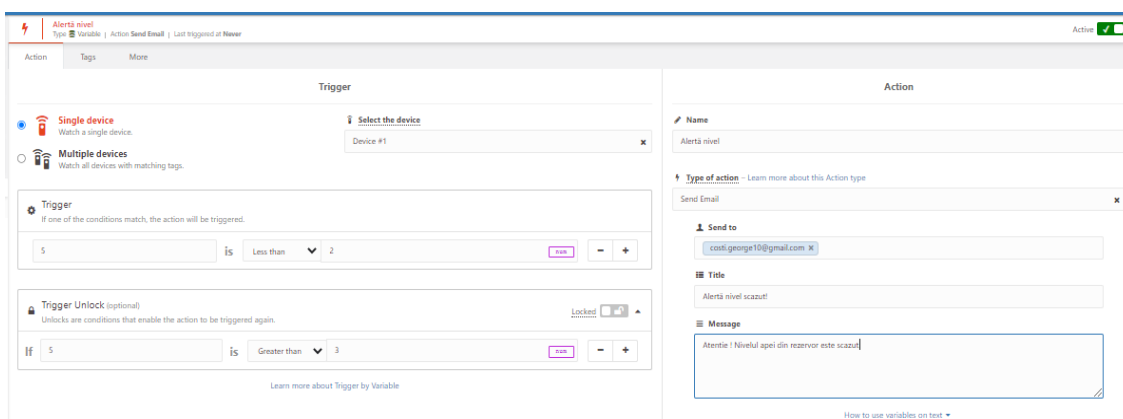


Fig.3.6.11 Configurare alertă

Se alege emailul destinație, titlul și descrierea, dar și condiția de declanșare a alertei, care în cadrul acestui exemplu este ca nivelul să fie mai mic de 2 cm. O altă setare importantă este și condiția de deblocare a alarmei, astfel se evită situațiile de retransmisie continuă a alertei.

Iar exemplu de execuție a unei alerte condiționată de umiditatea din incinta microcontrolerului:

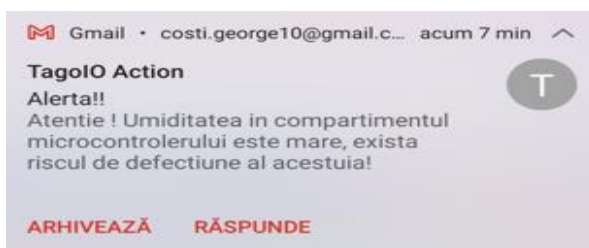


Fig. 3.6.12 Alertă email

Concluzii

Scopul principal al proiectului este de a cerceta și însuși diferite aspecte teoretice privind monitorizarea automată dar și implementarea fizică a unui sistem de monitorizare și transmisii de date bazat pe comunicațiile radio.

Prin prezentul proiect am reușit să realizez un sistem ce oferă transmisii de date prin tehnologia LoRa. Sistemul se bazează pe comunicații radio, banda de frecvență utilizată în acest sistem este 868 MHz. Am decis să utilizez această tehnologie deoarece comunicarea fără fir și pe distanțe mari este una dintre cele mai dorite moduri de comunicare între două sau mai multe dispozitive.

Avantajele acestui sistem sunt :

- transmisii pe distanțe mari cu un consum mic de energie
- costul total anual pentru transmiterea datelor este mai mic de 10 lei.
- aplicabilitate extraordinară pentru rețele de senzori wireless , IoT și crearea de rețele de dispozitive inteligente.
- permite transmiterea datelor în afara ariei de acoperire GSM sau WiFi prin adăugarea cu ușurință în zona dorită de concentratoare LoRa cu un cost relativ redus.

Consider că sistemul realizat îndeplinește cu succes obiectivele pe care mi le-am propus și anume: asigură monitorizarea automată de la distanță a volumului și temperaturii lichidului dintr-un rezervor, a temperaturii și umidității aerului și a tensiunii de alimentare, prin memorarea datelor într-o baza de date de tip cloud ,totodată notifică și personalul prin email la apariția unei alerte, toate acestea cu un preț mic de implementare și funcționare.

O altă caracteristică importantă este aceea că am reușit să creez un sistem independent, ce nu necesita mentenanță periodică pentru reîncărcarea acumulatorului, deoarece sistemul construit beneficiază de un panou solar ce va alimenta sistemul reîncărcând acumulatorul în perioadele însorite, perioadele de mentenanță fiind doar în situațiile cauzate de defecțiuni sau îmbunătățiri.

Bibliografie

1. Paraschiv N., Introducere în automatică și calculatoare. Editura Universității Petrol – Gaze din Ploiești, 2017
2. Paraschiv N., Programarea aplicațiilor în timp real, curs, platforma elearning UPG Ploiești, 2021.
3. Bucur, G., Senzori, traductoare, măsurare, Editura UPG Ploiesti, 2011.
4. Popa, C. Proiectarea asistată în automatizări, curs, platforma elearning UPG Ploiești, 2022.
5. Mihalache S. Ingineria reglării automate, curs, platforma elearning UPG Ploiești, 2021.
6. Pricop E., Rețele de calculatoare, curs, platforma elearning UPG Ploiești, 2022.
7. Zamfir F., Sisteme cu microprocesoare, curs, platforma elearning UPG Ploiești, 2021.
8. Moise A. Automate si microprogramare, curs, platforma elearning UPG Ploiești, 2021.
9. Cangea O. Transmisia datelor, curs, platforma elearning UPG Ploiești, 2021.
10. Baiesu, A. Tehnica reglării automate, Editura Matrix Rom, București 2012
11. Bucur, G., Sisteme inteligente de masurare. Structuri de baza si aplicatii, Editura UPG Ploiești, 2018, ISBN 978-973-719-751-1
12. Bala, S. Circuite electrice, curs , platforma elearning Piazza , 2020
13. Rahabul Islam, Wahidur Rahman, Rahmina Rubaiat, Mahfuz Reza. LoRa and server-based home automation using the internet of things (IoT), Computer and Information , 7 ianuarie 2021
14. Laure Moiroux-Arvis, Christophe Cariou, Jean Chanet, Evaluation of LoRa technology in 433-MHz and 868-MHz for underground to aboveground data transmission, Computers and Electronics in Agriculture, 7 Februarie 2022
15. Cosmin C. Utilizarea tehnologiei LoRa pentru transmiterea datelor în afara ariei de acoperire GSM, 2020
16. <https://cleste.ro/senzor-ultrasonic-rezistent-la-apa.html>
17. <https://www.senzori-ultrasonici.ro/principiul-ultrasonic>
18. <https://cleste.ro/senzor-temperatura-rezistent-la-apa-ds18b20.html>
19. <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
20. <https://heltec.org/project/wireless-stick-lite/>

21. <https://wifimag.ro/piese/dht22>
22. <https://github.com/lnlp/LMIC-node>
23. <https://www.hackster.io/uFire/helium-water-quality-monitor-075e6d?>
24. <https://docs.helium.com/use-the-network/console/>
25. https://github.com/Chiumanfu/LMIC-node_Sensor-for-Helium-Network
26. <http://community.heltec.cn/t/heltec-wifi-lora-v2-battery-management>
27. <http://community.heltec.cn/t/wifi-lora-32-v2-and-solar-panel-management>
28. <https://docs.helium.com/use-the-network/console/integrations/tago/>
29. <https://github.com/platformio/platformio-core>
30. <https://docs.helium.com/use-the-network/console/data-credits/>
31. <https://docs.mydevices.com/docs/lorawan/cayenne-lpp>
32. <https://www.diodes.com/assets/Datasheets/ds23001.pdf>
33. <http://free-energy-monitor.com/index.php/energy/fotovoltaice>

Summary

The main goal of the project is to research various theoretical aspects of automatic monitoring and physical implementation of a data transmission system based on radio communications.

Through this project we managed to create a system that provides data transmission through LoRa technology. The system is based on radio communications, the frequency band used in this system is 868 MHz. I decided to use this technology because wireless and long distance communication is one of the most desirable ways to communicate between two or more devices.

The advantages of this system are:

- long-distance transmissions with low energy consumption
- the total annual tariff for data transmission is less than 10 Ron.
- extraordinary applicability for wireless sensor networks, IoT and network of smart devices.
- allows data transmission outside the GSM or WiFi coverage area by easily adding LoRa Gateways to the desired area at a relatively low cost.

I consider that the realized system successfully fulfills the objectives I set myself, namely: it ensures the automatic remote monitoring of the volume and temperature of the liquid in a tank, of the air temperature and humidity and of the supply voltage, by storing the data in a cloud database, also notifies staff by email when an alert occurs, all with a low cost of implementation and operation.

Another important feature is that we managed to create an independent system that does not require periodic maintenance to recharge the battery, because the built system benefits from a solar panel that will power the system by recharging the battery in sunny periods, maintenance periods being only in situations with defects or improvements.