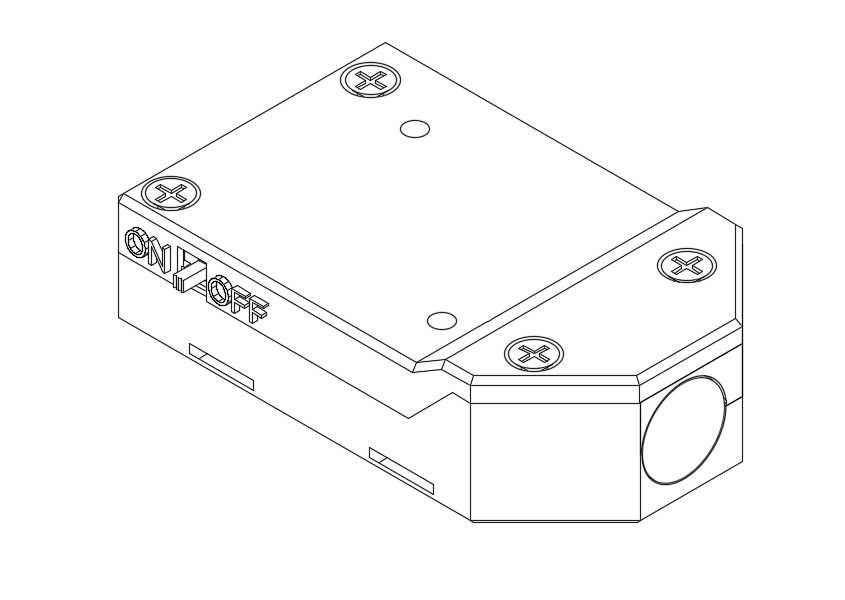
Dispozitiv de asistență pentru parapantă

(Glidar)

**Documentație **

1. **Abstract**

Glidar este un dispozitiv pentru asistarea zborului cu parapanta, care poate furniza următoarele informații utilizatorului: distanța față de sol, viteza, orientarea, temperatura și presiunea, având o durată lungă de viață și fiind compact. Dezvoltarea dispozitivului s-a făcut cu feedback constant din partea parapantiștilor, care au revizuit structura aplicației noastre și au dat feedback-ul pe care l-am folosit pentru a îmbunătăți experiența utilizatorului. Un senzor ultrasonic este folosit pentru a măsura distanța față de sol, deoarece oferă necesarul de distanță maximă detectabilă de 8m și o precizie ridicată de 1cm. Am folosit un IMU cu 6 axe pentru a calcula viteza și orientarea parapantei și am măsurat și temperatura și presiunea. Pentru a face dispozitivul mai compact și mai fiabil, am proiectat un PCB personalizată pentru el. Am folosit componente SMD mici astfel încât acestea să poată fi asamblate automat cu o mașină de tip pick-and-place și un cuptor de reflow în producție sau prin utilizarea unui șablon de aluminiu și a unei plite de reflow acasă. Înainte de a face placa de circuit reală, a trebuit să proiectăm o schemă electrică. Am separat-o în module pentru a fi mai ușor de depanat. De asemenea, am dorit o carcasă electronică compactă, netedă și ușor de transportat. Am folosit fabricația aditivă în avantajul nostru și am iterat rapid între mai multe versiuni până când am avut una cu care suntem mulțumiți. Am dorit să facem acest produs cât mai aproape de un produs real, așa că am făcut o cutie cu spumă pentru a ține dispozitivul în loc și am inclus și un mic manual de instrucțiuni. În ceea ce privește software-ul, am programat o aplicație în flutter datorită capacității sale de a exporta atât pe android, cât și la ios cu un singur program și pentru că folosește designul material de la Google pentru un aspect mai minimalist. Programul preia datele trimise de unitatea principală prin Wi-Fi și folosește datele de accelerație și viteză unghiulară pentru a calcula orientarea în X, Y și Z și o afișează ca un model 3D. Afișează, de asemenea, viteza calculată din vectorul de accelerație și presiunea și temperatura sub o formă ușor de înțeles de către utilizator.

1. **Hardware**
   1. **Rezolvarea problemelor de hardware**

Pentru a documenta toate problemele de hardware, am realizat următorul tabel.

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Problemă** | **Soluție** |
| 1 | Am avut nevoie de un dispozitiv foarte mic, care să aibă o durată minimă de viață de 2 ore pentru a facilita zboruri lungi. | Am folosit componente SMD care pot fi asamblate automat folosind o mașină de tip pick-and-place sau manual folosind un pistol de căldură sau un cuptor de reflow și pastă de lipit (pe care am folosit-o eu). În ceea ce privește bateria, am folosit o baterie li-po de 1600mah pentru o durată de viață a bateriei calculate de aproximativ 2,5 ore. |
| 2 | Avem nevoie să măsurăm distanța de la sol până la 8 metri cu o precizie de ± 10 cm pentru a avertiza utilizatorul când să înceapă aterizarea. | Am folosit senzorul de distanță ultrasonic URM13 pentru capacitatea sa de a măsura până la 9m, având o precizie de ±1cm și o amprentă mică de 20x22mm. |
| 3 | Am avut nevoie de măsurători precise ale temperaturii și presiunii pentru a fi afișate utilizatorului. | Am folosit senzorul bmp280 datorită consumului său redus de curent (3,4 μA la 1 Hz), preciziei sale de ±1°C și ±1 hPa și a amprentei sale mici datorită pachetului LGA (2x2,5mm). |
| 4 | Am avut nevoie de o modalitate mică și precisă de a măsura viteza și orientarea utilizatorului: | Am folosit IMU bmi088 cu 6-axe pentru dimensiunea sa foarte mică (3x4,5mm) și pentru că avea o documentație foarte bună și a fost proiectat pentru aplicații de performanță ridicată în medii dure. |
| 5 | Am avut nevoie de o modalitate de a încărca bateria li-po care va alimenta dispozitivul. | Am folosit un circuit integrat de încărcare a bateriei tp4056, deoarece are un curent de încărcare ușor de reglat și este foarte bine testat. |
| 6 | Am avut nevoie de o modalitate de a regla tensiunea bateriei la o tensiune constantă de 3,3v pentru a o utiliza cu restul componentelor. | Am folosit un regulator de tensiune prin comutație TPS631000 pentru curentul maxim de ieșire de 1,5A la 3,3v, zgomot redus și capacitatea de a schimba tensiunea de ieșire cu doar un divizor rezistiv. |
| 7 | Aveam nevoie de un microcontroler cu capacitate WiFi, care să poată rula un server și să primească date de la senzori folosind i2c. | Am folosit microcontrolerul ESP32 deoarece eram familiarizați cu el și suportă WiFi și are mai multe interfețe i2c, precum și destule GPIO-uri pentru alte componente. |
| 8 | Aveam nevoie de o modalitate de a notifica utilizatorul când dispozitivul este pornit și despre orice probleme cu acesta. | Am integrat un LED RGB controlat de ESP32 și care poate fi folosit pentru semnalizarea erorilor cu culori diferite. |
| 9 | Aveam nevoie de un software de proiectare PCB care să fie ușor de utilizat, în timp ce are toate caracteristicile standard precum DRC. | Am folosit Kicad deoarece este gratuit și open-source și are toate caracteristicile de care aveam nevoie. |
| 10 | Aveam nevoie de un program CAD care să aibă capacitatea de a modifica toleranțe în mod facil și care să poată exporta designul ca o schemă desenată sau explodată. De asemenea, doream să facem o cutie pentru Glidar, așadar aveam nevoie de un software care să aibă capacitatea de a imprima obiecte la scară și care să fie mai potrivit pentru aplicații 2D. | Am folosit Fusion360 pentru designul carcasei deoarece funcția de timeline face ușor ștergerea funcțiilor nedorite sau modificarea anumitor dimensiuni, iar schema explodată automată a fost o funcție plăcută. Pentru designul cutiei, am folosit Autocad deoarece a fost ușor să imprimăm la scară 1:1 cu acesta. |
| 11 | Aveam nevoie de o modalitate de a fabrica carcasă pentru electronice care să poată produce forme complicate și să putem itera pe aceasta de mai multe ori pe zi. | Am folosit fabricația aditivă și am făcut carcasa să fie adecvată pentru imprimarea 3D astfel încât să putem face mai multe iterații pe zi. |

| componentă | Consum(3.3v) |
| --- | --- |
| ESP32 | 230 ma |
| BMP280 | 0.18 ma la 5Hz max |
| BMI088 | 5ma cand masoară accelerația  si viteza unghiulară |
| URM13 | 350ma max |
| Consumul total: 585.18ma; Putere totală: 1.93 W | |

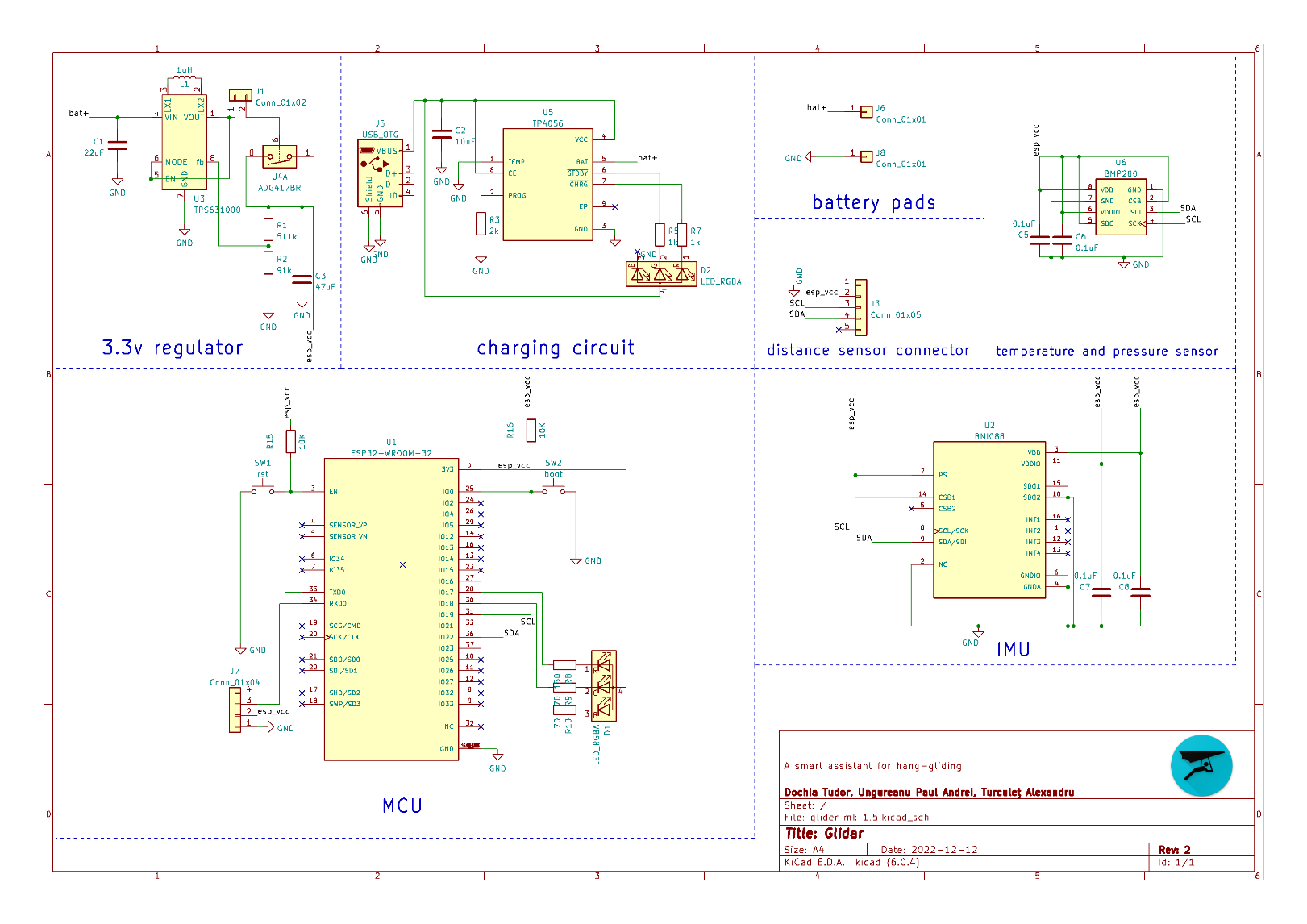
**Power budget folosit pentru a calcula durata de viață**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Descriere** | **Producător** |
| EG1213 | întrerupător | E-Switch |
| TPS631000 | Regulator prin comutare | Texas Instruments |
| TP4056 | Incărcător pentru baterii cu litiu | Kuongshun Electronic |
| BMP280 | Sensor de temperatură și presiune | Bosch Sensortech |
| BMI088 | IMU pe 6 axe cu accelerometru și giroscop | Bosch Sensortech |
| ESP32-WROOM-32 | microcontroller cu support pentru WiFi | Espressif |
| JST SH 0x05 connector | Conector utilizat pentru senzorul de distanță (același tip de conector utilizat în sistemul Qwiic) | JST |
| 434781025816 | 2 butoane pentru programarea ESP32-ului | Wurth Elektronik |
| AAA3528LSEKJ3ZGKQBKS | LED RGB | Kingbright |
| 105017-0001 | connector micro USB | Molex |
| M3x12mm screws | Înșurubarea carcasei | unknown |
| Threaded inserts-M3 standard | Facilitarea demontării carcasei | CNCKitchen |
| PLA NX2 Black | Filamentul utilizat pentru imprimarea carcasei | Extrudr |

**Lista finală a componentelor**

**(fără componente complementare cum ar fi rezistori sau condensatori)**

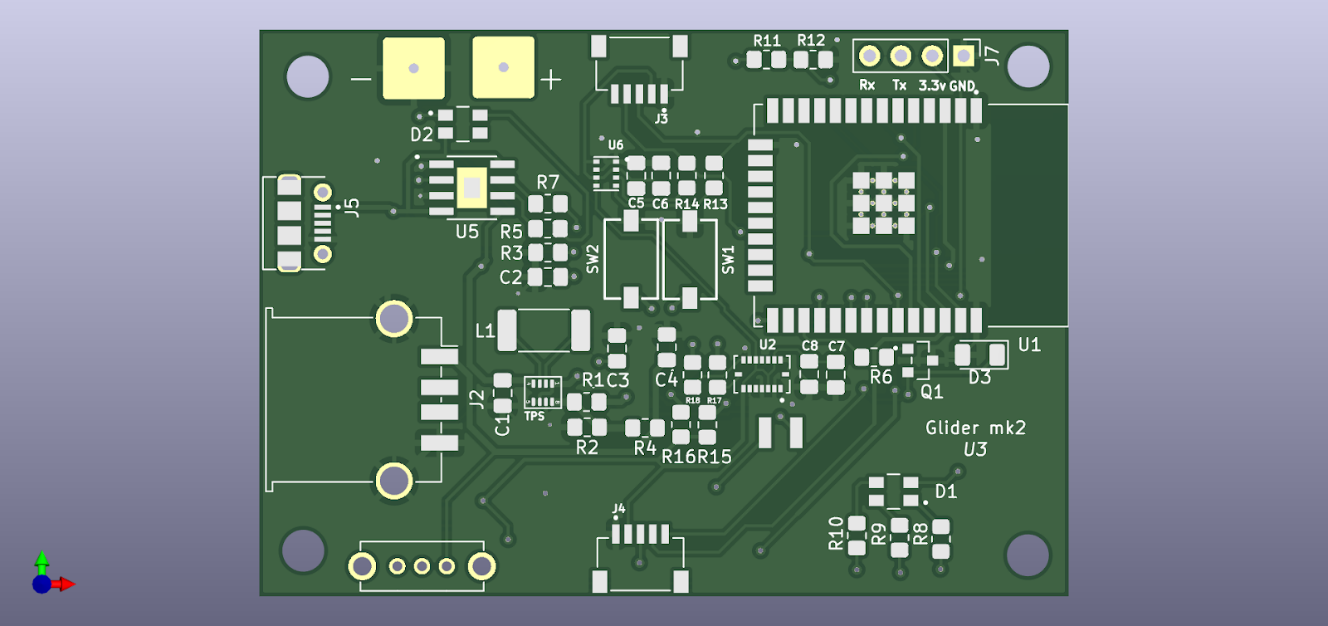
* 1. **Procesul de design hardware**
     1. **Designul schemei electrice**

După alegerea tuturor senzorilor și componentelor, am început lucrul la schema electrică prin citirea tuturor fișelor tehnice și căutarea considerațiilor de proiectare implicate de producător, cum ar fi valorile condensatoarelor de filtrare și începerea plasării tuturor componentelor specificate într-un mod logic pentru lizibilitate, separându-le în secțiuni delimitate cu linii punctate și folosind etichete cu denumiri clare în loc de conexiuni reale. În această fază, am decis ce conectori să folosim și cum să programăm ESP32-ul, hotărând să folosim 2 butoane conectate la piniul enable și GPIO0 și connector temporar pentru UART.

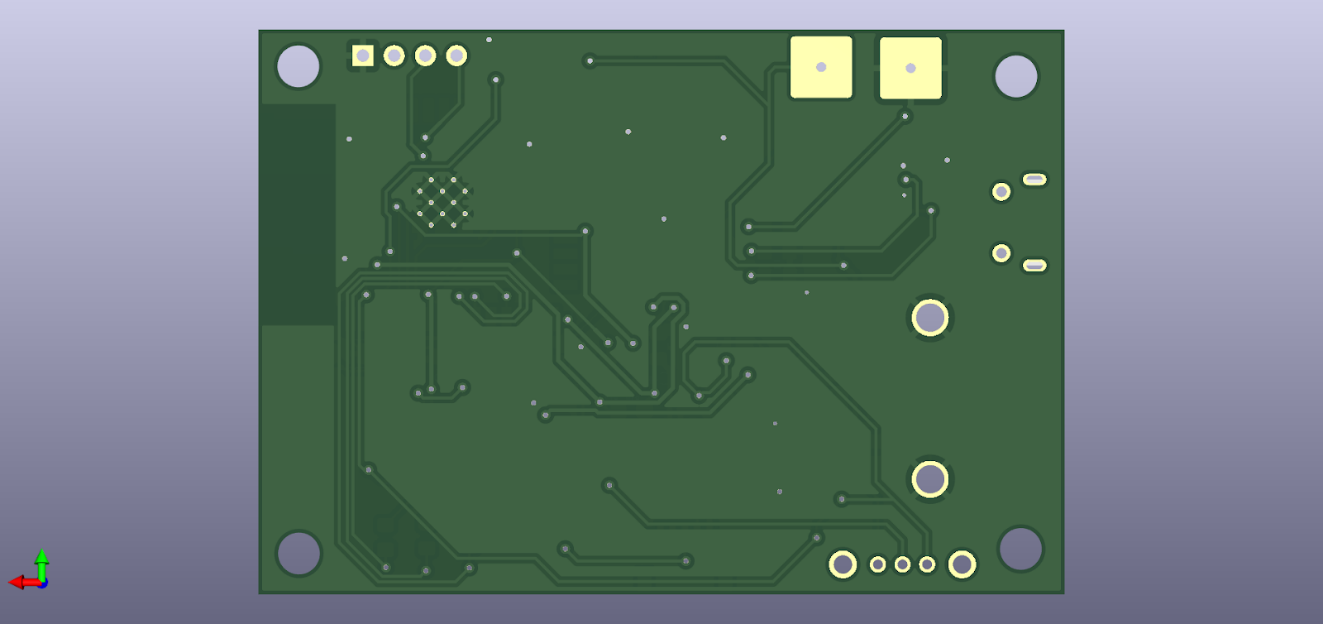
**Schema electrică finală**

* + 1. **Design-ul PCB-ului**

După realizarea schemei electrice, am început să lucrăm la PCB, plasând mai întâi toate componentele dintr-o secțiune a schemei aproape unul de celălalt și aranjându-le fie folosind specificațiile de plasare PCB din fișele tehnice, fie convențional, folosind reguli generale, cum ar fi condensatorii de filtrare să fi aproape de liniile de alimentare. Apoi, am rutat toate traseele logic, respectând reguli generale, cum ar fi evitarea unghiurilor de 90 de grade și folosirea celor 2 straturi în avantajul nostru. După aceea, am folosit funcția DRC din Kicad și am introdus toleranțele de la producătorul nostru de PCB-uri și am corectat orice greșeli pentru a nu fi nevoie să facem mai multe revizii. Unul dintre cerințele noastre a fost ca conectorul microUSB să aibă suporturi prin gaură, deoarece am avut experiențe foarte neplăcute cu conectorii pur SMD care se rupeau după doar câteva utilizări.



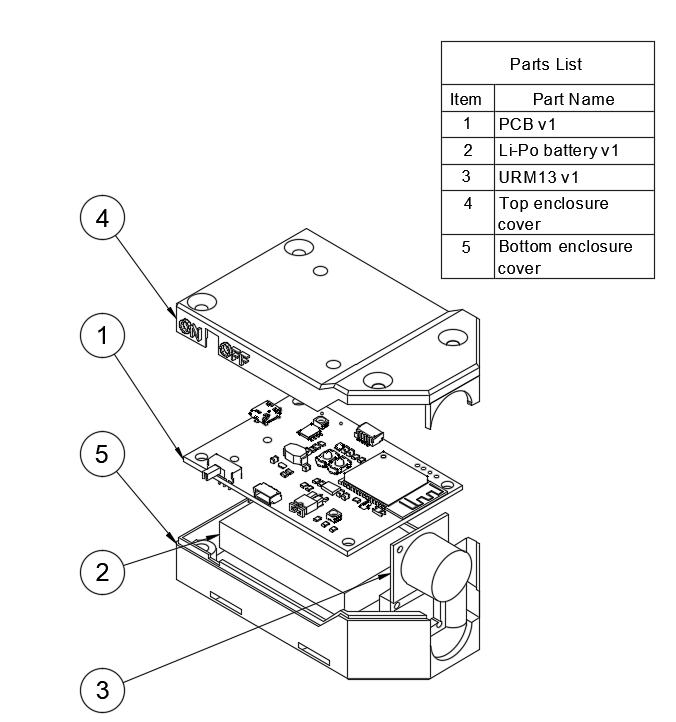
**Partea de sus**



**Partea de jos**

* + 1. **Design-ul carcasei și al cutiei**

După ce PCB-ul a fost finalizat, am exportat fișierul ca .step și am început să lucrăm la carcasă. Am utilizat inserturi cu încălzire pentru majoritatea șuruburilor, deoarece a făcut mai ușor dezamblarea și depanarea ulterioară. De asemenea, ne-am asigurat să ținem cont de toleranțele imprimantei noastre 3D, pe care le-am măsurat anterior pentru atât caracteristicile noastre interne, cât și externe. După mai multe revizii, am imprimat versiunea finală folosind un compozit de PLA numit PLA NX2 de la Extrudr, care are o finisare mată plăcută și proprietăți mecanice și termice îmbunătățite. După ce am finalizat carcasa, am trecut la proiectarea ambalajului său, deoarece doream să arate ca un produs finit. Am folosit o cutie de la o placă de bază a unei imprimante 3D și am realizat un sticker personalizat la dimensiune în AutoCAD, prezentând un desen tehnic al Glidar-ului. Apoi am tăiat spuma în forma dispozitivului astfel încât să fie protejat în interior și am inclus un mic manual de instrucțiuni.



**Schema explodată Dispozitivul fizic și cutia**

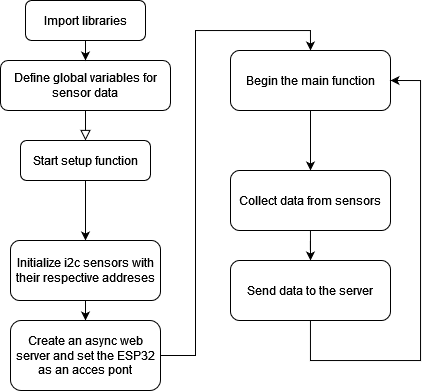
1. **Software**
   1. **Rezolvarea problemelor de software**

Pentru a documenta problemele de software am realizat acest tabel

|  |  |
| --- | --- |
| **Problemă** | **Soluție** |
| Aveam nevoie de o modalitate de a programa ușor esp32-ul pentru a comunica cu senzorii și a trimite datele prin WiFi. | Am folosit framework-ul Arduino pentru a programa microcontroller-ul, deoarece se bazează pe C++, un limbaj cu care eram familiarizat și are o comunitate extinsă |
| Aveam nevoie de un limbaj de programare pentru a crea aplicația de telefon care poate primi date prin intermediul unui server web și care folosește designul material Google pentru o interfață minimalistă. | Am folosit limbajul de programare flutter, deoarece se baza pe dart, un limbaj similar cu C++, și avea capacitatea de a exporta atât pentru dispozitivele Android, cât și către cele iOS dintr-un singur cod sursă. |

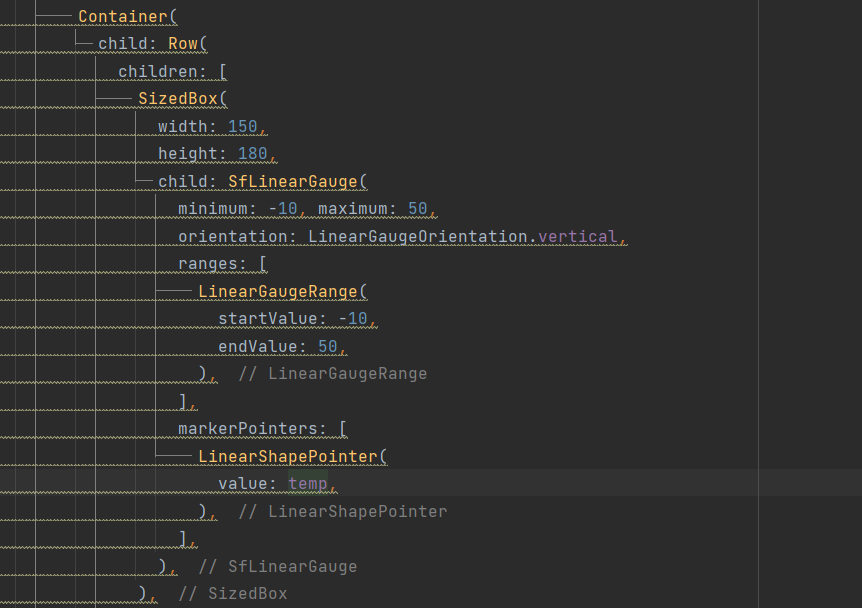
* 1. **Programarea microcontroller-ului**

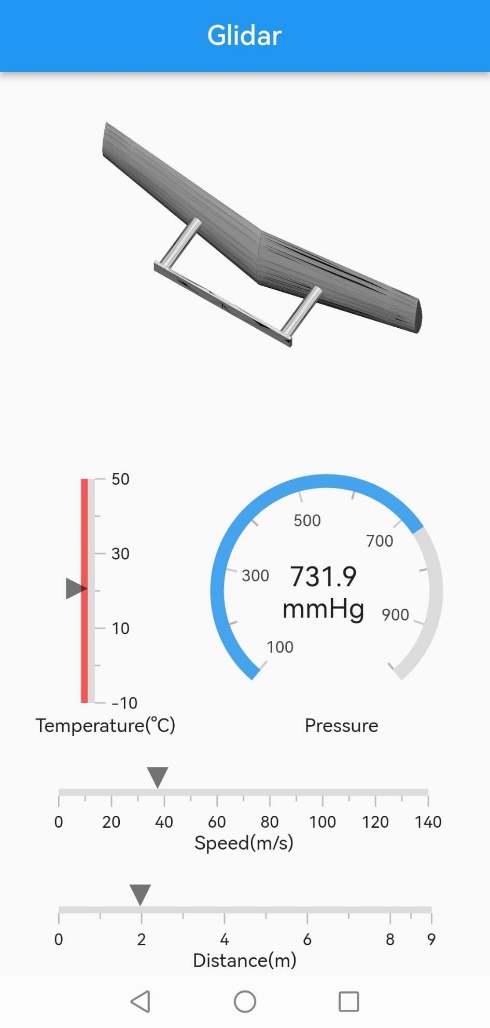
Codul ESP32 inițializează mai întâi toți senzorii I2C utilizând adresele lor specifice, apoi inițializează ESP32 ca un punct de acces și pornește un server web asincron, apoi rulează o funcție pentru a obține datele de temperatură, presiune, accelerație, viteză unghiulară și distanță de la senzori și le trimite ca un șir de caractere la serverul web. Un avantaj al acestei metode este că utilizatorul poate folosi în continuare dispozitive Bluetooth în timp ce utilizează Glidar.

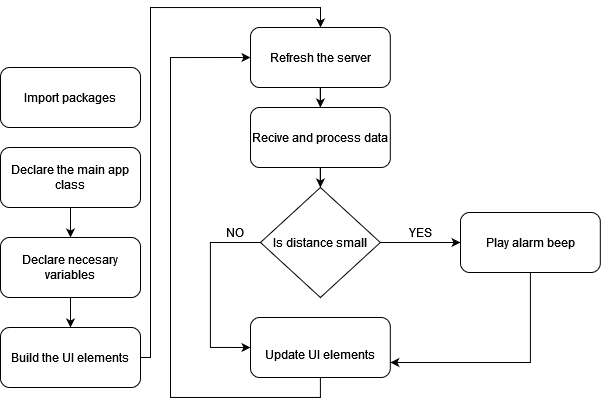
  **codul pentru microcontroller Flowchart-ul pentru codul pentru ESP32**

* 1. **Programarea aplicației pentru telefon**

Aplicația de pe telefon actualizează serverul web și obține datele la fiecare 200 ms, convertește datele în valori zecimale individuale și le afișează într-o formă ușor de înțeles pentru utilizator. Temperatura și presiunea sunt afișate sub forma de indicatoare radiale, în timp ce viteza calculată din vectorul de accelerație împreună cu distanța față de sol măsurată cu senzorul ultrasonic sunt afișate sub formă de bare lineare. De asemenea, afișăm orientarea utilizatorului calculată pe baza datelor IMU sub formă de model 3D rotativ. Când distanța față de sol este prea mică, se redă un sunet de avertizare pentru a alerta utilizatorul. Acest sunet este redat chiar dacă utilizatorul a părăsit aplicația pentru a preveni orice accidente.



****

 **Programul pentru aplicația în flutter**

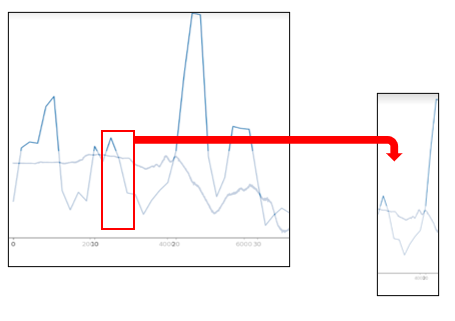
**Interfața aplicației**

**Flowchart-ul programului în flutter**

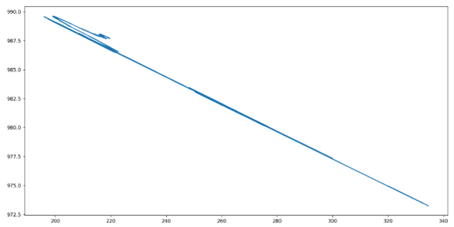
1. **Analiza datelor**

Pentru a înțelege mai bine principiile de funcționare, am colectat date prin testarea echipamentului la sol și am concluzionat că atunci când accelerația crește, presiunea scade chiar dacă nu se schimbă altitudinea. După ce am determinat că acest fenomen apare de mai multe ori pe parcursul traseului, am căutat o explicație pentru această afirmație pentru a înțelege mai bine cauza, ajungând în cele din urmă la legea lui Bernoulli care spune că suma presiunii statice, presiunii dinamice și a presiunii hidrostatice este constantă (). Această lege explică perfect fenomenul deoarece atunci când apare o accelerație, presiunea dinamică crește și, deoarece altitudinea nu se schimbă, presiunea hidrostatică rămâne aceeași, dar pentru ca suma celor 3 presiuni să fie constantă, presiunea statică trebuie să scadă.

Prin calibrare, valoarea vitezei parapantei aproape de sol pot fi determinate prin valoarea presiunii statice. Am făcut acest lucru, de data aceasta vertical, și am obținut graficul de mai jos.



**Această porțiune a graficelor presiunii (hPa/timp) și vitezei (m/s/timp) arată cel mai bine principiul lui Bernoulli. Când accelerația crește, crescând astfel presiunea dinamică, presiunea statică scade.**



**Graficul presiune(hPa)/altitudine(m)**

Am avut și profesioniști care au verificat design-ul aplicației noastre și ne-au dat feedback, pe care l-am folosit pentru a îmbunătăți experiența utilizatorului. De exemplu, am făcut astfel încât să nu fie nevoie să deruleze cu degetul pentru a vedea toate datele.

1. **Project conclusion**

În concluzie, am reușit să facem un dispozitiv inteligent care crește nivelul de siguranță într-un sport periculos, utilizând senzori moderni și l-am făcut să semene cât mai mult posibil cu un produs finit cu resursele disponibile, cu speranța de a-l comercializa.