ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
ELEKTRONIKAI ÉS KOMMUNIKÁCIÓS RENDSZEREK INTÉZET
MŰSZERTECHNIKAI ÉS AUTOMATIZÁLÁSI TANSZÉK

SZAKDOLGOZAT

OE-KVK
2023.

Bálint Dániel
T003671/FI12904/K

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Műszertechnikai és Automatizálási Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Bálint Dániel
Szakdolgozat száma: SZD2309040923399441[redacted]
Törzskönyvi száma: T003671/FI12904/K **Neptun kódja:** [redacted]

Szak: villamosmérnöki (magyar nyelven)
Specializáció: Műszer-automatika specializáció

A dolgozat címe: Pulzoximéteres betegfelügyelő rendszer
A dolgozat címe angolul: Patient Monitoring System with Pulse Oximeter
A feladat részletezése: Készítsen egy mikrokontrolleren alapuló mérési és megfigyelő rendszert, melyet a felhasználó folyamatosan hord, és ami probléma esetén jelzést képes adni.

Intézményi konzulens neve: Sándor Tamás

A kiadott téma elévülési határideje:2025. 12. 31.
Beadási határidő: 2023. 12. 15.

A szakdolgozat: Nem titkos.

Kiadva: Budapest, 2023. 10. 24. P.H

Óbudai Egyetem Kandó Kálmán Villamosmérnöki Kar Elektronikai és Kommunikációs Rendszerek Intézet

.....
Intézetigazgató

A dolgozatot beadásra alkalmasnak találom:

.....
belső konzulens

.....
külső konzulens

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY
KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
ELEKTRONIKAI ÉS KOMMUNIKÁCIÓS RENDSZEREK INTÉZETE
MŰSZERTECHNIKAI ÉS AUTOMATIZÁLÁSI TANSZÉK

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Budapest, 2023.12.06.

Bálint Oáriel

hallgató aláírása

TARTALOMJEGYZÉK

1. BEVEZETÉS

Bizonyos betegségben szenvedők, vagy idősek esetén szükség lehet az adott személy pulzusának és/vagy véroxigén szintjének folyamatos figyelésére. Az általános pulzoximéterek többsége egy ujjra csíptethető szerkezet, mely egyszeri mérésre használatos, nem pedig folyamatos megfigyelésre. Ezen pulzoximéterek használata egyszerű, de folyamatos használat esetén a mozgásban korlátozó és zavaró tényező.

Az általam elképzelt műszer ennek a feladatnak megfelel, valamint egyéb kiegészítő funkciókkal is rendelkezik a további biztonság érdekében.

Lehetőséget adhat a felhasználó jelenlegi állapotának távolról való megtekintésére, valamint képes jelzést leadni, ha a felhasználó esést szenved el, vagy akár abban az esetben is, ha sokáig tétlen.

Lehetőséget ad a riasztások és a különböző események idő szerinti feljegyzésére, valamint azon események bekövetkezési helyének jelzésére.

Képes a mért adatok rögzítésére, mely lehetővé teszi a későbbi visszatekintést, és grafikonok rajzolását, ami segít a gyors áttekintésében.

Ezen műszer segítségével lehetséges az adott felhasználó állapotának egész napon át tartó figyelése, és probléma esetén az azonnali beavatkozás alkalmazása.

2. SPECIFIKÁCIÓ

Egy pulzoximeter fő feladata a vér-oxigén szint, valamint a pulzusszám mérése és kijelzése, lehetőleg minél kompaktabb, hordozható formában. Azonban ez folyamatos megfigyeléshez már nem elegendő.

Az általam elképzelt műszer viszonylag kis méretű, akkumulátorról hosszú ideig működőképes és zavartalanul viselhető, ezzel folyamatos megfigyelési lehetőséget biztosítva.

A pulzus és oxigénszint adatok begyűjtéséért a pulzoximéter szenzor szükséges, mely optikai módszerrel, infravörös és piros fény segítségével megvilágítja a bőrfelületet, majd az arról visszaverődő fényt elemzi. A modul a kapott adatokat a mikrokontroller felé küldi a kiértékelésre.

Az adatok feldolgozására és kiértékelésére egy olyan mikrokontroller szükséges, amely képes sok adat feldolgozására, valamint Bluetooth kapcsolatra. A kontroller egy kijelző segítségével megjeleníti az adatokat a felhasználó számára. A rendszer vezérlésére a gombok segítségével van lehetőség.

A hosszú üzemidőre való tekintettel a rendszer tartalmaz alacsony fogyasztású módot, mely lehetővé teheti az akár a több napos használati időt.

A különböző riasztásokat hangjelzéssel, valamint Bluetooth kapcsolat segítségével mobilalkalmazáson keresztül jelezheti a felhasználónak.

A rendszerben helyet foglal egy nagypontosságú mozgásszenzor modul is, mellyel érzékelhető, ha a felhasználó tétlen, vagy esést szenved.

A kész eszköz egy 3D nyomtatott ház segítségével lenne rögzíthető az adott felhasználó csuklóján.

3. LOGIKAI RENDSZERTERV

A felügyelő rendszert elsősorban olyan felhasználók számára tenném használhatóvá, akik kórházban tartózkodnak és azonnali segítségre szorulhatnak, valamint későbbiekben olyan időseknek, akik nem rendelkeznek folyamatos felügyelettel és biztonságban szeretnék magukat tudni.

A rendszer a „Gondos óra” idős felügyelő rendszerhez hasonlóan lenne használható, mely egy olyan eszköz, ami képes jelzést leadni, valamint hívást indítani egy diszpécser központ felé, ha a felhasználónak problémája akad, vagy esést szenved el. Az eszköz mobilhálózat kapcsolattal rendelkezik, tartalmaz a híváshoz használható mikrofont és hangszórót, rezgőmotort, gyorsulás érzékelőt, valamint egy gombot a vészhívás indításához. Az üzemideje akár elérheti a 4 napot is töltés nélkül. [1]

Ez a rendszer azonban csak a felhasználó jelzésére vagy elesésére riaszt. Az én rendszerem azonban esetleges közelgő problémákat is jelezhet, révén a pulzus és oxigénszint mérési képessége miatt.

3.1 Mechanika

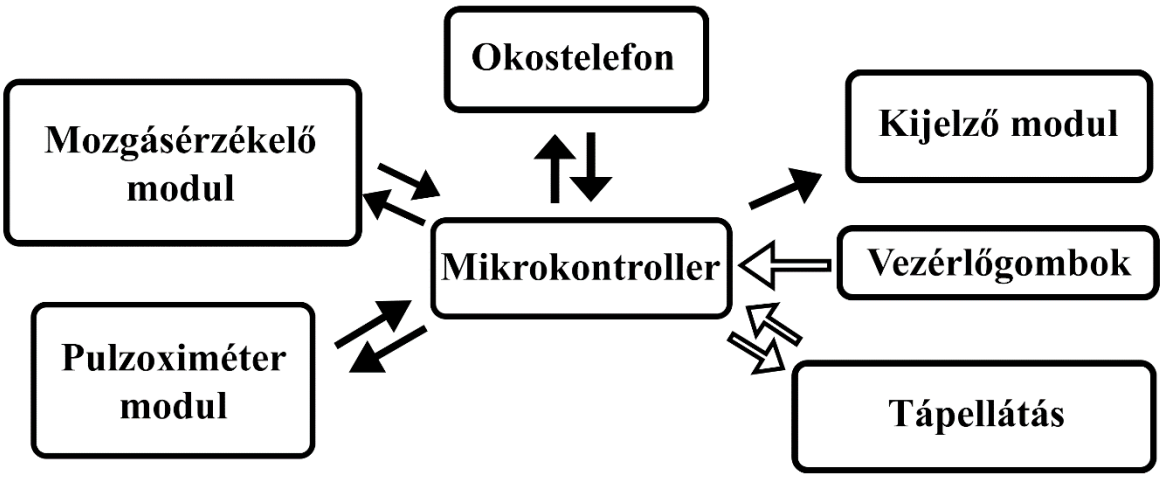
A pácienshez való rögzítés többféleképpen is kivitelezhető. A pulzoximéterek többsége egy ujjvégre rögzíthető csiptetős szerkezet, melyet könnyen lehet felvenni és levenni. A problémát itt az okozza, hogy a csiptető szerkezet viszonylag nagy, és a kézmozgásban gátló, valamint hosszútávon zavaró tényező. Ezért az ihletet az okosórák által használt rendszerből merítettem, melyben az óra hátlapján helyezik el a szenzorokat. Az én megoldásom annyiban különbözne ettől, hogy a szenzor a pánt másik oldalán helyezkedne el, nem pedig magában a felső részben, méretcsökkentés gyanánt.

A rögzítő szerkezet három részből áll:

- Felső rész: Ebben a részben fog helyet kapni a mikrokontroller, az akkumulátor, a kijelző és a csipogó.
- Alsó rész: Ebben a részben kapnak helyet a szenzorok, felső részében a pulzoximéter egy kivágáson keresztül érintkezik a bőrfelülettel, az alsó részben pedig a mozgásérzékelő kap helyet.
- Összeköttetés: A két részt egy gumipánt fogja össze, mely a biztos rögzítést is elvégzi.

3.2 Elektronika

Az elektronikai modulok felépítését a következő ábrán láthatóan végezném el:



1. ábra: Az elektronikai modulok felépítése

Az eszköz tartalmaz egy mikrokontrollert, mely feldolgozza és továbbítja a mért adatokat, és melyhez különböző modulok csatlakoznak. Ilyen modulok a kijelző modul, mely kijelzi az aktuális adatokat és státuszt. A pulzoximéter modul, mely a pulzus és az oxigénszint adatokat méri, valamint a mozgásérzékelő modul, mely az elesés érzékelésére szolgálna. A mikrokontroller vezérlésére gombokkal van lehetőség. Az ábrán a tápellátásért felelős rész is helyett kapott, valamint az okostelefonnal való kommunikáció is megjelenik.

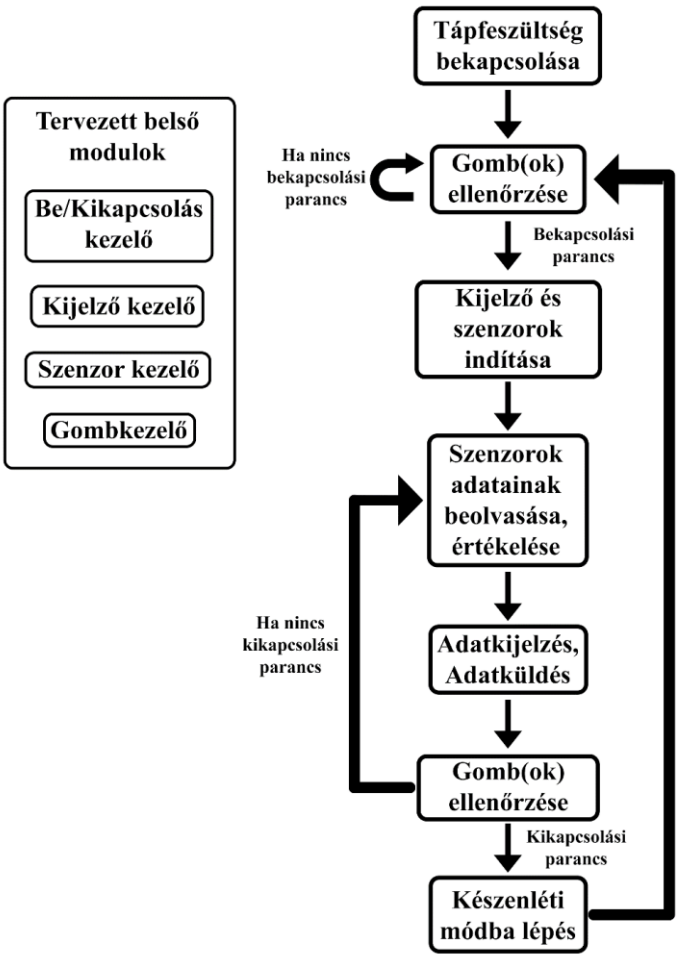
A továbbiakban az eszközt felépítő hardvereket a „MODULOK LEÍRÁSA” részben mutatom be részletesen.

3.3 Szoftver

3.3.1 A mikrokontroller szoftverének elképzelt működése

A mikrokontroller szoftverének képesnek kell lennie a szenzorok adatainak folyamatos mérésére és számolására, a kijelző folyamatos frissítésére, a vezérlőgombok figyelésére, az okostelefonnal való kommunikációra, riasztások jelzésére. A szoftvernek tartalmaznia kell egy menürendszert, melyben az eszköz testre szabható, valamint amelyben egyéb információk tekinthetők meg. A különböző funkciókat modulos rendszerben szeretném a programban létrehozni, mert ez megkönnyíti a teljes program létrehozását és tesztelését.

A szoftver elképzelt működésének folyamata, valamint moduljainak ábrája:



2. ábra: Az ESP32 szoftverének elképzelt folyamat ábrája

3.3.2 A kijelző kezelő modul működése

A kijelző modul felelős a kijelzőn megjelenő karakterek kiírásáért, frissítéséért és törléséért.

A modul indítása a kijelző jelenlétének ellenőrzésével majd felkonfigurálásával indul, ezután a fontos értékek kiíratása történik meg. A kijelző frissítése megadott időközönként történik, mely folyamat során először elvégzi kijelző törlését, majd ezután az aktuális karakterek megjelenítését. A megjelenítendő adatokat vagy parancsokat függvények segítségével lehet a modulnak megadni, ami ezután a következő frissítés alkalmával végrehajtja azokat.

3.3.3 A szenzor kezelő modul működése

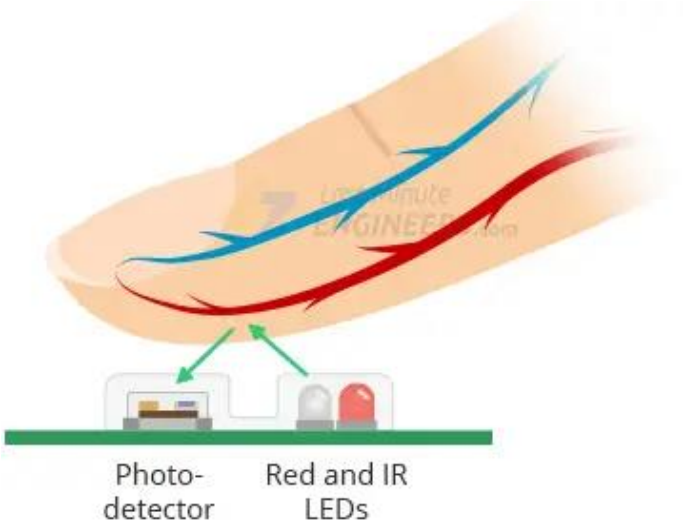
A szenzorokat kezelő modul feladata a pulzoximéter, valamint az orientációs modul felkonfigurálása, valamint adatainak lekérése, számolása.

A kezelő modulok működésének menete hasonló. A modul a szenzor jelenlétének ellenőrzésével majd annak felkonfigurálásával indul. A szenzor adatainak lekérését megadott időközönként végi el, melyeket függvények segítségével lehetséges kikérni.

3.3.4 A pulzus és oxigénszint mérésének menete, algoritmusa

A pulzoximéter szenzor működése optikai alapon történik. A szenzor tartalmaz egy piros és egy infravörös LED-et, valamint egy fotoszenzort, mely ezen két LED visszavert fényét elemzi.

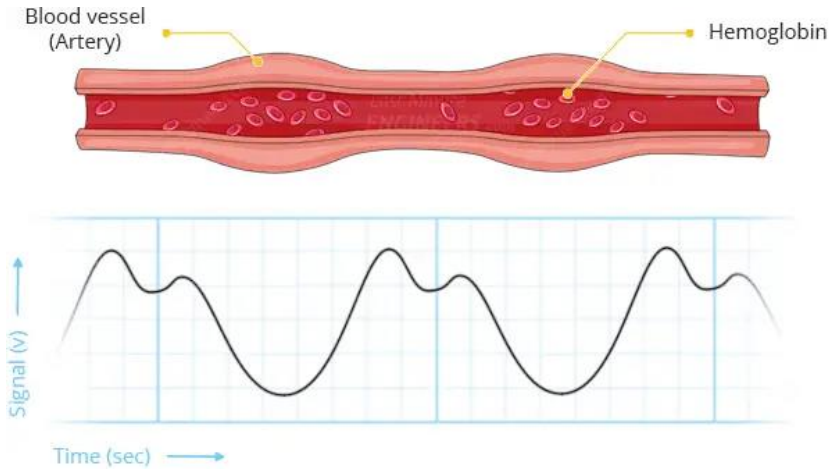
A LED-ek fénye megvilágítja a bőrfelületet, és az abban lévő vér pulzálásának hatására a fény visszaverődésének mértéke periodikusan változik. Ezt a visszaverődő, pulzáló fényt elemzi a szenzor, majd a mért adatokat elküldi a mikrokontroller felé a további értékelésre. Ezt a fajta pulzus és oxigénszint mérést „Photoplethysmogram”-nak nevezik.



3. ábra: A pulzoximéter fényének útja

A pulzus mérésének elve:

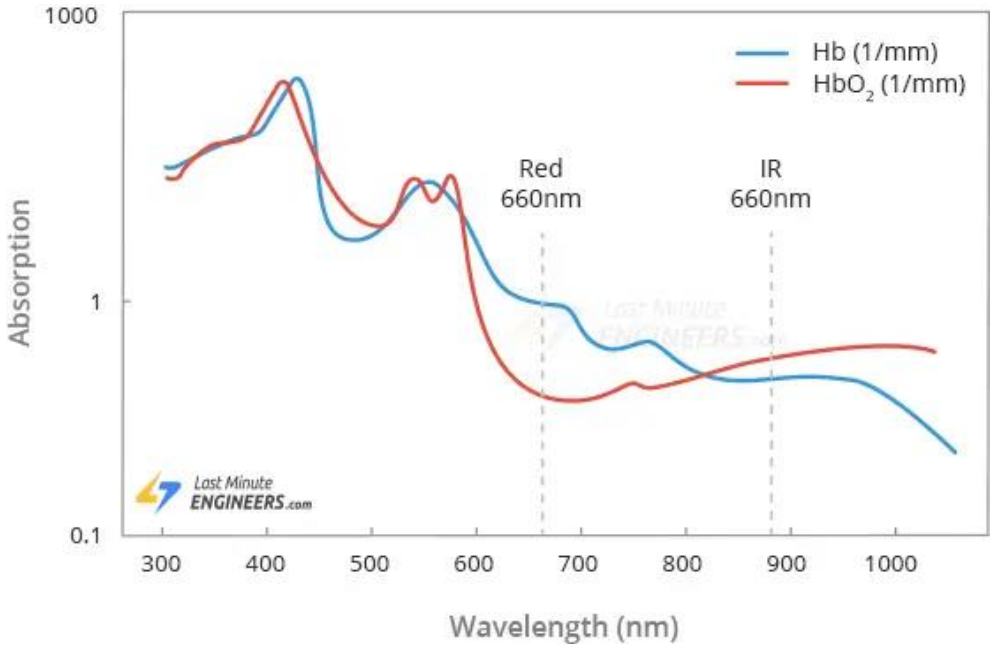
A vérben lévő hemoglobin karakterisztikus infravörös fény elnyelési képességgel rendelkezik. A szívverés pumpálásának hatására periodikusan változik a visszavert fényük mértéke az erekben, és ebből kikövetkeztethető a szívverés mértéke.



4. ábra: A pulzus hatására visszavert fény grafikonja

A véroxigénszint mérésének elve:

A mérése a vér infravörös és piros fény elnyelési képességén alapul, mely a vér oxigéntelítettségétől függően változik. A következő grafikon mutatja a hemoglobin fényelnyelő képességét, melyben a „HbO_2” az oxigénnel telített, és a „Hb” az oxigént nem tartalmazó hemoglobin fényelnyelési spektruma.



5. ábra: A hemoglobin fényelnyelési képességének spektruma

Az oxigént nem tartalmazó vér több piros fényt, míg az oxigénnel telített vér több infravörös fényt nyel el. Az oxigénszaturáció mértéke a két érték közötti viszonyszámból kapható meg.

Ezen viszonyszám kiszámolását a mikrokontroller végzi a modul által rögzített adatokból. [2]

3.3.5 Az esésérzékelés menete, algoritmusa

Az esés érzékelése viszonylag egyszerű módszerrel lehetséges.

A mozgásszenzor rögzíti mind a három mozgástengely (X, Y, Z) gyorsulását. Ezeket az értékeket négyzetesen összeadva az eredő gyorsulás értékét kapjuk.

A gyorsulások összeadásának matematikai képlete a következő:

$$accAmp = \sqrt{accX^2 + accY^2 + accZ^2}$$

Az eredő gyorsulás értéke nyugalmi állapotban közel egyenlő a gravitációs gyorsulással, aminek értéke ~9,8m/s^2.

Ez az érték szabadesés közben megközelíti, vagy eléri a nullát, mellyel egy pont adható meg az algoritmusnak, mely jelzi, hogy szabadesés van folyamatban.

Az esés végén azonban egy nagyobb gyorsulás érzékelése történik meg, a hirtelen megállás és/vagy a felületről való visszaverődés hatására. Ezen pont szintén érzékelhető, mert az érték viszonylag hirtelen, pozitív ugrással reagál.

Az algoritmus ezenfelül kiegészíthető mozgásérzékeléssel, mely megakadályozza a hamis riasztásokat. Ha az alsó és a felső érték feltételei teljesülnek, akkor az algoritmus következőnek azt vizsgálja meg, hogy éppen történik-e mozgás vagy sem. Ha nem történik mozgás egy ideig vár, majd ismételten megvizsgálja, hogy történik-e mozgás. Ha ekkor sem történik, akkor jelzés kerül küldésre.

3.3.6 A tétlenség érzékelésének menete, algoritmusa

A tétlenség érzékelése mozgáson alapul. Ha a mozgás megszűnik, egy időzítő indul, mely, ha elér egy megadott időt, jelzést ad. Ha időközben mozgás történik, akkor ez az időzítő törlődik, és újrakezdi a számlálást.

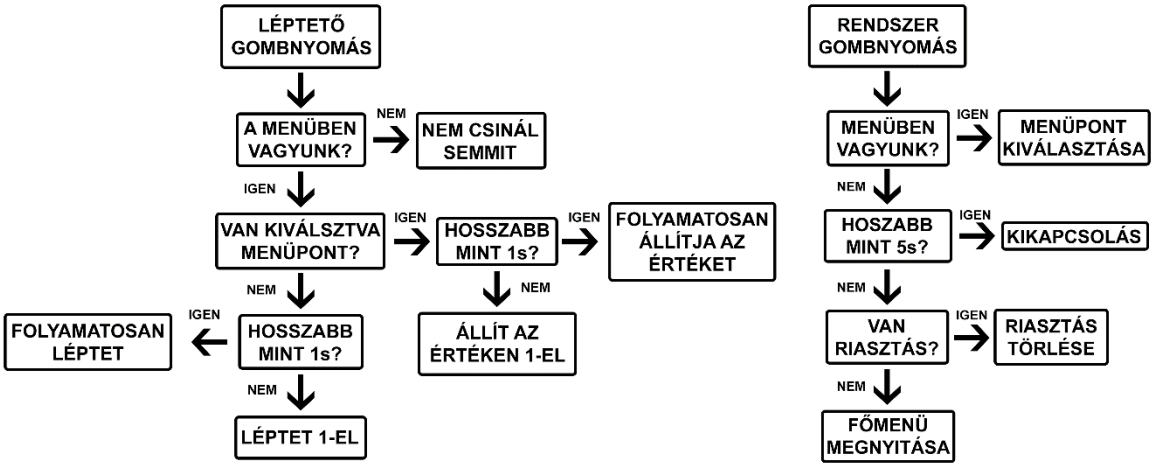
3.3.7 Gombkezelő modul működésének megtervezése

A gombkezelő modul feladata a vezérlőgombok funkcióinak megadása, végrehajtása.

A vezérlésre három gomb segítségével lenne lehetőség, melyben a gombok a következő funkciókat töltenék be:

- A menübe való belépés, a kijelölt elem kiválasztása, és hosszas megnyomása esetén az eszköz kikapcsolása. Kikapcsolt állapotból való ébredés indítása.
- A menüben való fel/le lépegetésre, valamint változtatható opciók esetén azok értékének növelésére és csökkentésére. A fel és le funkciók saját gombokkal rendelkeznének. Hosszas megnyomásuk esetén az érték folyamatosan változna.

A gombkezelő működését a következő ábra szemlélteti:



6. ábra: A gombkezelő működésének ábrái

3.3.8 Az Android szoftver elképzelt működése

Az Android alkalmazás segítségével lehetséges az eszközhöz való csatlakozás, annak státuszának folyamatos figyelése.

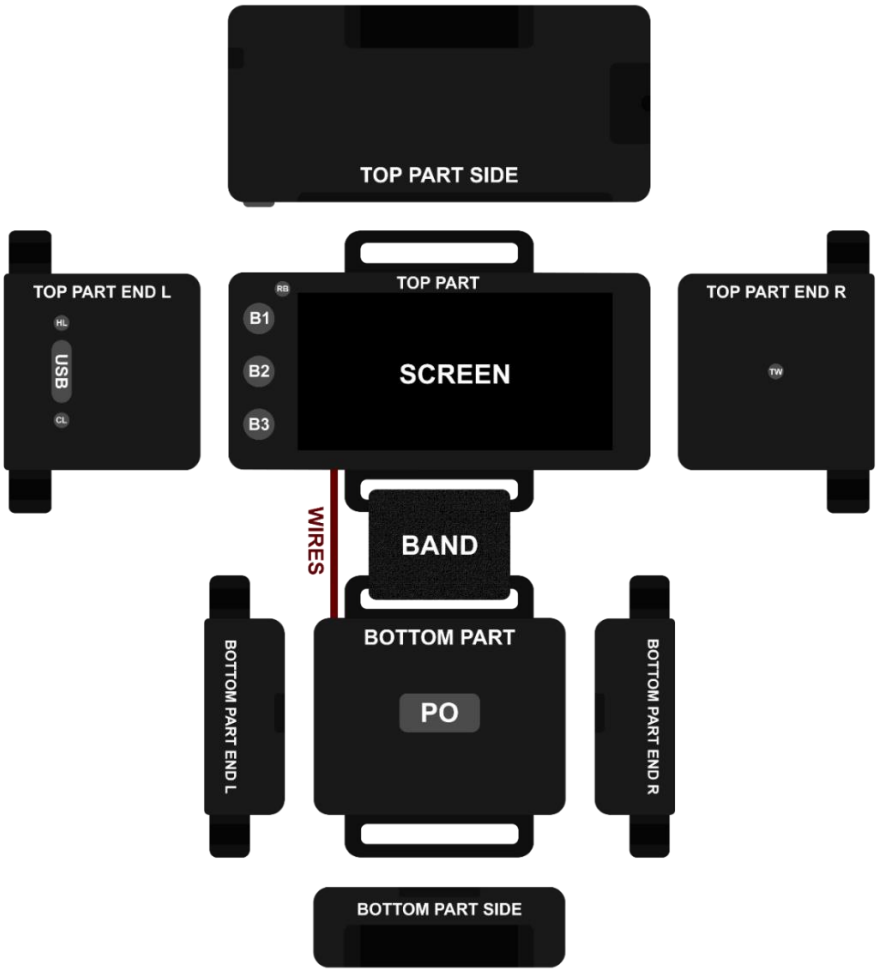
Az alkalmazásnak képesnek kell lennie a pulzusszám, oxigén-szaturáció, az akkumulátor állapotának, valamint a jelenlegi riasztás státuszának lekérésére, a jelenlegi idő továbbítására az eszköz felé.

Az eszköz riasztása esetén az alkalmazásnak is riasztást kell küldenie, valamint a hely jelzésére is képesnek kell lennie, ahol ez bekövetkezett. Az eseménynaplóba ezen események idejét és helyét rögzítenie kell. Az eszköz állapotát 24 órára visszamenőlegesen egy grafikonon ábrázolni tudja, mely az egyszerű visszatekintést és elemzést garantálja.

4. FIZIKAI RENDSZERTERV

4.1 Mechanika

A felépítés egy egyszerű csuklópántos kivitel, melynek különböző részeit háromdimenziós program segítségével tervezném meg, majd ezeket a részeket háromdimenziós nyomtató segítségével, adott műanyagból kinyomtatnám és összeszerelném. Az elképzelt felépítésről a következő kihajtott háromdimenziós ábrát készítettem a Adobe Photoshop program segítségével:



7. ábra: A rögzítőmechanizmus kihajtott háromdimenziós ábrája

A pánt egy felső és egy alsó részből áll, melyet egy gumipánt tart össze és rögzít a karon.

A felső rész tartalmazza a kijelzőt a gombokkal, a vezérlésért felelős mikrokontrollert, valamint az akkumulátort és a csipogót. Az alsó rész tartalmazza a mozgásszenzort, valamint a pulzoximéter szenzort. A két rész közötti összeköttetést vezetékek biztosítják, amelyek a pánton futnak.

4.2 Elektronika

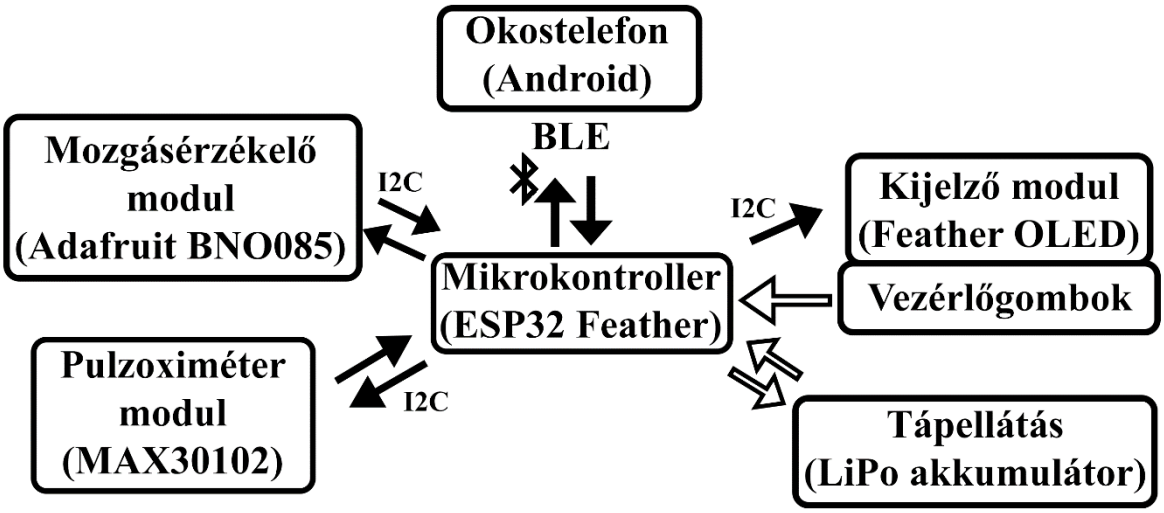
A vezérlésre egy ESP32 mikrokontrollert választottam, ez kezeli a kijelzőt, valamint dolgozza fel a szenzorok által rögzített adatokat, fogadja a gombok lenyomását, jelzéseket küld a külvilág és a mobilkészülék felé.

A kommunikáció a mikrokontroller és a perifériák között I2C kapcsolat segítségével, a mobilkészülék felé pedig Bluetooth kapcsolat segítségével valósul meg.

A perifériák tápellátását a mikrokontroller végzi el, melyet egy akkumulátor táplál, amit a mikrokontroller nyákján lévő vezérlő segítségével lehetséges tölteni is.

A mikrokontroller vezérléséért felelős gombok a kijelző paneljén találhatóak meg.

A modulok felépítése a következő ábrán látható:



8. ábra: Hardvermodulok felépítése

A modulok specifikációit a „MODULOK LEÍRÁSA” részben részletesen tárgyalom.

4.3 Szoftver

A be/ki kapcsoló gomb megnyomásával indítható a készülék, ezzel léphetünk a menübe, nyugtázhatjuk a riasztásokat, illetve hosszas megnyomása esetén kapcsolhatjuk ki a készüléket. A további gombok a menüben való léptetésre, értékek megváltoztatására, újraindításra adnak lehetőséget.

Bekapcsoláskor a mikrokontroller ellenőrzi a hardverek megfelelő működését, a kijelző és a szenzorok bekapcsolnak, majd megkezdi a szenzorok által rögzített adatok feldolgozását és kijelzését. A perifériák frissítése megadott időközönként történik, a modulok adatainak kikérése és vezérlése függvényekkel lehetséges.

A menüben lehetőség nyílik különböző dolgok konfigurálására, például a kijelző fényerejének beállítására, kikapcsolási időzítőjének módosítására.

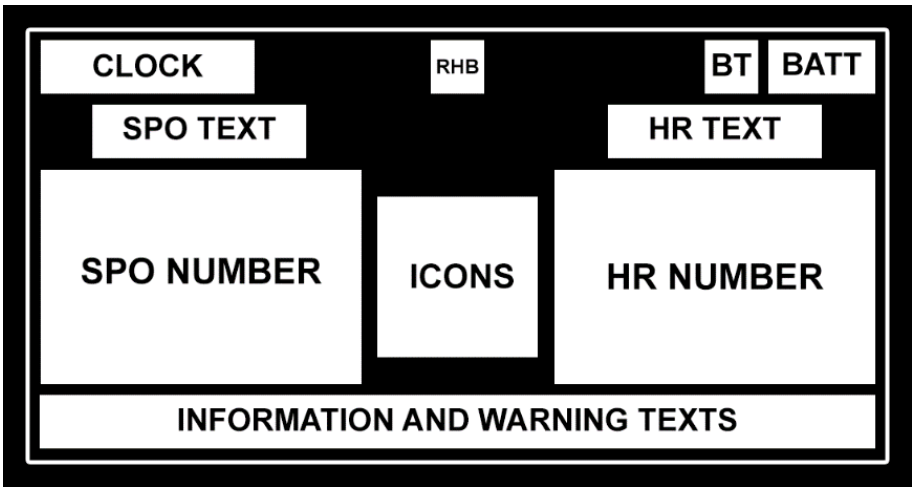
Hiba vagy riasztás esetén az eszköz az adott riasztás szövegét kiírja a kijelzőre, valamint hangjelzést ad. Ha van mobileszköz csatlakoztatva az szintén jelzést ad le.

4.3.1 A kijelzés és kinézet megtervezése

A kijelző kinézetének vázlatát az Adobe Photoshop, illetve a GIMP szerkesztő programok segítségével készítettem el, mely egyszerűnek és könnyűnek bizonyult, hogy azt a későbbiekben csak megvalósítani kelljen. A rajzot a kijelző méretarányával azonosan készítettem el.

A kijelzésnek minden fontos adatot tartalmaznia kell, melyeknek jól láthatónak kell lenniük.

Alapesetben kijelzésre kerül a mért pulzusszám és oxigénszint, az akkumulátor jelenlegi állapota, a vezeték nélküli kapcsolat állapota, valamint a jelenlegi riasztás szövege. A kijelző közepén helyet kapott a kijelző frissülését jelző animáció, mely a kijelzés befagyása esetén könnyen észrevehető jelzést ad a felhasználónak.



9. ábra: Az ESP32 szoftverének grafikai terve

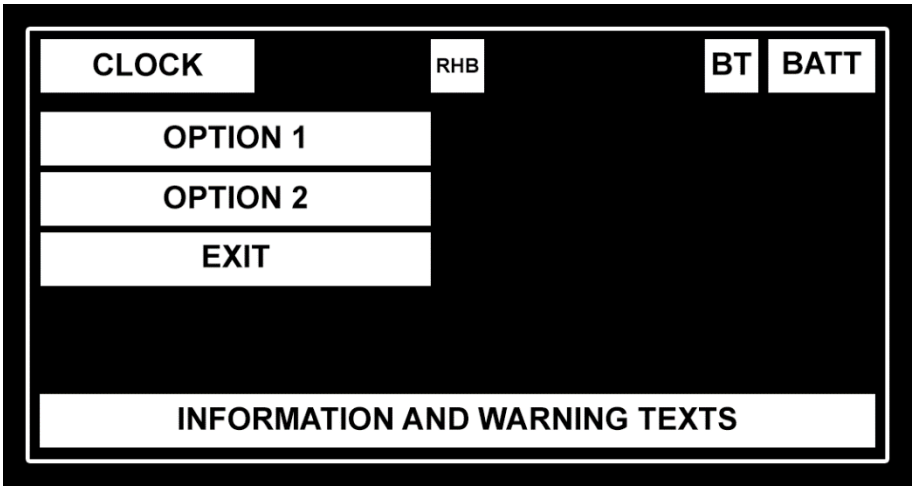
A grafikán megjelenő rövidítések leírása:

- CLOCK: Bluetooth kapcsolat esetén a jelenlegi időt mutatja
- RHB: „Refresh Heart Beat” mutató, animációt játszik a kijelző frissülésekor
- BT: Bluetooth jel, a Bluetooth aktuális státuszát mutatja
- BATT: Az akkumulátor jelenlegi töltöttségi szintjét mutatja
- SPO TEXT: Az oxigén szaturáció, felirata („SPO2 %”)
- HR TEXT: A pulzusszám felirata („HR Bpm”)
- SPO NUMBER: Az aktuális oxigénszint százalékban
- HR NUMBER: Az aktuális pulzusszám ütés/percben
- ICONS: Pulzus(szív) ikon vagy figyelmeztető(felkiáltó jel) ikon animációja
- INFORMATION …: A riasztások megjelenítésének helye

4.3.2 Menürendszer kezelő kinézetének tervezése

A programban több menü és funkció is helyet kapna, és ezeket a menürendszer modul segítségével készíteném el, melyben lehetőség nyílik egyszerűen hozzáadni új menüpontokat és kijelezni kívánt oldalakat.

A menüpontok egymás alatt, és két oszlop szélességben jelennének meg. Az opciók között a fel és le gombokkal lehet navigálni vagy értéküket megváltoztatni. A kijelzésen továbbra is szerepel az óra, a kapcsolat állapotának jelzése, az akkumulátor státuszának jelzése, valamint a jelenlegi riasztás állapotának kijelzése, a kijelző frissülését jelző animáció.



10. ábra: Az ESP32 menürendszerének grafikai terve

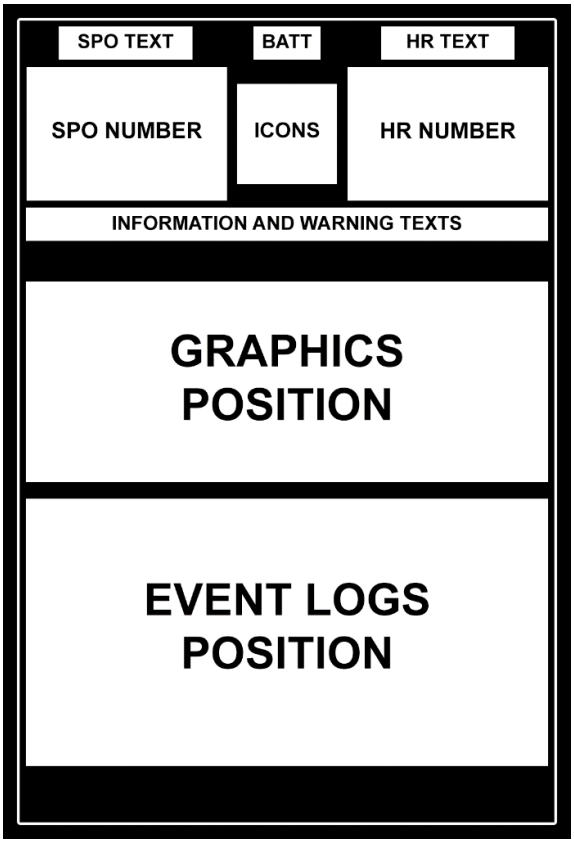
A grafikán megjelenő rövidítések leírása:

- CLOCK: Bluetooth kapcsolat esetén a jelenlegi időt mutatja
- RHB: „Refresh Heart Beat” mutató, animációt játszik a kijelző frissülésekor
- BT: Bluetooth jel, a Bluetooth aktuális státuszát mutatja
- BATT: Az akkumulátor jelenlegi töltöttségi szintjét mutatja
- OPTION n: Menüpontok egyedi elnevezéssel és funkciókkal
- EXIT / BACK: A menüben való visszalépés gombja
- INFORMATION …: A riasztások megjelenítésének helye

4.3.3 Az Android alkalmazás kinézetének megtervezése

Az alkalmazás kinézetének vázlatát, valamint ikonjainak, képeinek elkészítését szintén a már korábban említett Photoshop, illetve GIMP programok használatával végeztem el. Az elkészítéskor törekedtem a lehető legegyszerűbb kinézetre, a fontos adatok kiemelésére.

A kijelzésben szerepelnek ugyanazon értékek, melyek az eszközön is megjelennek, valamint ezek alatt a kiegészítő részek, a grafikon, a térkép és az eseménylista.



11. ábra: Az Android alkalmazás kinézetének grafikai terve

A grafikán megjelenő rövidítések leírása:

- BATT: Az akkumulátor jelenlegi töltöttségi szintjét mutatja
- SPO TEXT: Az oxigén-szaturáció, felirata („SPO2 [%]”)
- HR TEXT: A pulzusszám felirata („HR [Bpm]”)
- SPO NUMBER: Az aktuális oxigénszint százalékban
- HR NUMBER: Az aktuális pulzusszám ütés/percben
- ICONS: Pulzus(szív) ikon vagy figyelmeztető(felkiáltó jel) ikon animációja
- INFORMATION: A riasztások megjelenítésének helye
- GRAPHICS: Grafikon és térkép megjelenítésének helye
- EVENT LOGS: Eseménynapló megjelenítésének helye

5. MODULOK LEÍRÁSA

5.1 Pulzoximéter modul

A pulzus és oxigénszint adatok gyűjtéséért egy MAX30102 pulzoximéter szenzorral ellátott modul felel. Lehetőség van a logikai szint feszültség kiválasztására is, ez ESP32 esetén 3.3V-ot jelent, de beállítható 1.8V-ra is, így bármely kontrollerhez univerzálisan illeszthető.

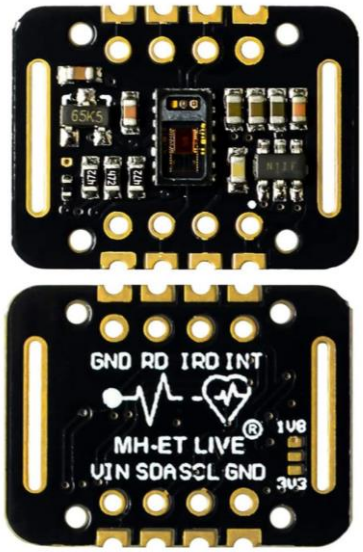
Elődjéhez képest (MAX30100) kevesebb fogyasztással rendelkezik, illetve képes az alvó üzemmódra való váltásra is, ezzel lehetőség nyílik arra, hogy hordozható rendszerekben is használható legyen. Rendelkezik beépített hőmérő szenzorral is, így lehetőség van kompenzálni a hőmérsékletből adódó eltéréseket. A modullal való kommunikáció I2C protokoll segítségével történik. Az adatok ideiglenes tárolása is lehetséges a modulon, amely egy 32 helyes FIFO tárban valósul meg.

A modullal különböző megszakítások kezdeményezésére is van lehetőség, mellyel a használata egyszerűbben történhet. Öt különbözőféle megszakítási esemény beállítása lehetséges:

- **Power Ready**: tápfeszültség bekapcsoláskor fut le
- **New Data Ready**: egy mérési adat elkészülése esetén hívódik meg
- **Ambient Light Cancellation:** akkor jelez, ha a szenzor túllépte a maximális fény-zaj elnyomási képességét
- **Temperature Ready**: akkor küld megszakítást, ha a belső hőmérséklet kiértékelése befejeződött
- **FIFO Almost Full**: megszakítást küld, ha a FIFO tár megtelt vagy meg fog telni

Egyéb működési paraméterek:

- Feszültség: 3.3-tól 5.5V-ig
- Áramfelvétel:
  - Mérés közben: ~600µA
  - Alvó állapotban: ~0.7µA
- A LEDek paraméterei:
  - Piros hullámhossz: 660nm
  - Infravörös hullámhossz: 880nm
- Hőmérséklet tartomány: -40°C-től +85°C-ig
- Hőmérési pontosság: ± 1˚C



12. ábra: MAX30102 pulzoximéter modul

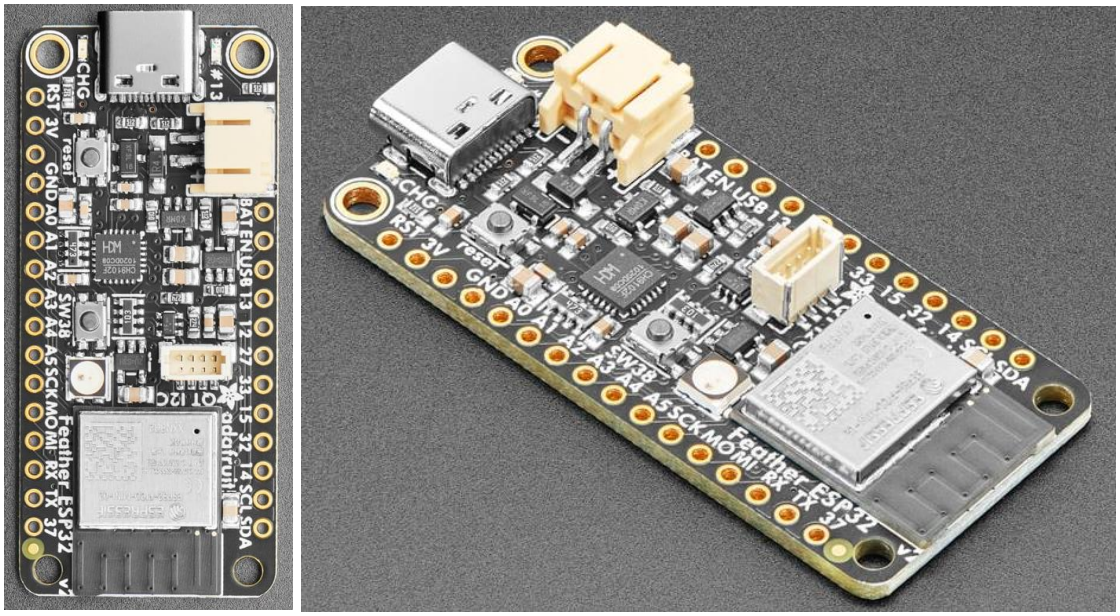
5.2 Mikrokontroller modul

Az adatok feldolgozásáért, a perifériák vezérléséért, valamint az okostelefonnal való kommunikációért egy Adafruit ESP32 Feather V2 fejlesztői nyák felel.

A nyákon egy ESP32 található, mely rendelkezik WiFi-vel és Bluetooth-al. A nyákra integráltak LiPo akkumulátort illesztő áramkört, egy újraindító és egy szabadon felhasználható gombot, valamint egy programozható RGB LED-et, viszonylag sok memóriát, dedikált I2C csatlakozót, melyet a projektemben fel is használok a könnyebb összeépítés érdekében.

Egyéb adatok a mikrokontrollerről:

- ESP32 Dual core Xtensa mikrovezérlő
  - Órajel: 240MHz
  - Flash: 8MB
  - PSRAM: 2MB
  - SRAM: 520KB
  - 3db UART, 3db SPI, 2db I2C, 12db ADC, 2db DAC
- Beépített WiFi / Bluetooth, integrált antennával
- USB-C csatlakozás programozáshoz és adattovábbításhoz
- Li-Po akkumulátor kezelés (töltés, akkumulátoros üzem, feszültség mérés)
- Egy beépített programozható és újraindító gomb
- Programozható NeoPixel RGB LED, és egy piros LED
- STEMMA QT gyorscsatlakozó I2C eszközök számára
- Működési feszültség 5V USB-ről, 3.7V LiPo akkumulátorról. vagy 3V külső tápról
- Maximális áramfelvétel ~500mA [3]



13. ábra: Adafruit ESP32 Feather V2 board

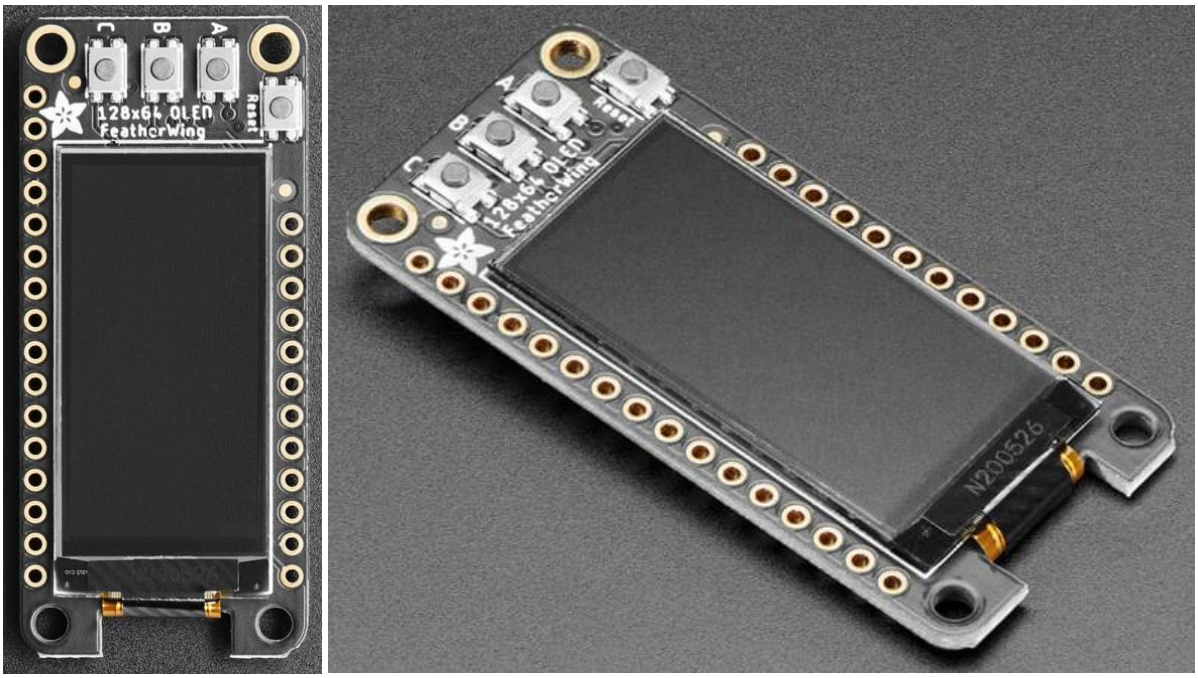
5.3 Kijelző modul

Adafruit FeatherWing OLED kijelző panel szolgál az adatok kijelzésére, illetve a gombokkal való vezérlésre.

A kijelzés színe fehér, 128x64 pixel maximális felbontással rendelkezik, valamint OLED megoldáson alapul. A nyákra építettek be egy a mikrokontroller újraindítását lehetővé tévő, és három szabadon programozható gombot. A nyák túloldalára került egy STEMMA QT I2C gyorscsatlakozó, amivel lehetőség nyílik a kijelző egyszerű csatlakoztatására vagy más eszköz csatlakoztatására a kontrollerhez.

A kijelző főbb paraméterei:

- A kijelző OLED technológiájú
- Képátlója 1.3”
- Kijelzés felbontása 128*64 pixel
- A kommunikáció I2C interfészen zajlik
- A kijelző 3.3V-ról működik
- Egyszínű, fehér képpontokat képes megjeleníteni
- A panel tartalmaz 3 konfigurálható gombot és egy, a mikrokontrollert újraindító gombot
- STEMMA QT gyorscsatlakozó I2C eszközökhöz a hátoldalon [4]



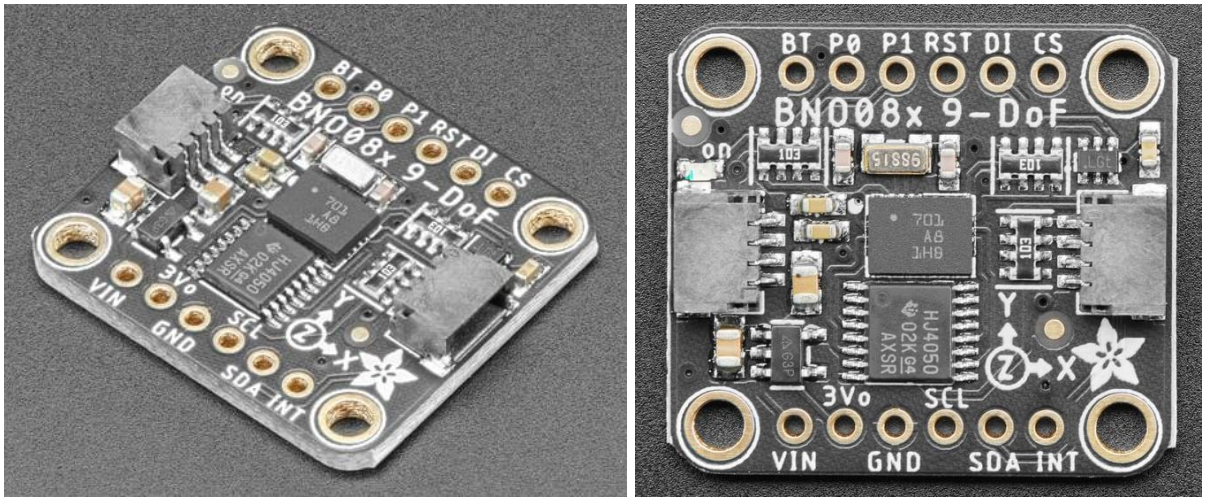
14. ábra: Adafruit FeatherWing OLED kijelző

5.4 Mozgásérzékelő modul

Adafruit 9-DOF Orientation IMU BNO085 mozgásérzékelő és orientációs modul.

A modul képes a gyorsulás nagy pontosságú követésére, valamint a mágneses orientáció és gravitáció mérésére alacsony válaszidő mellett. A kommunikációra több interfészen is lehetőség van, amelyen keresztül a modulon lévő mikrokontroller üzenetek formájában válaszol. A nyákra építettek két I2C gyorscsatlakozót, így lehetővé téve a modul és egyéb más eszköz gyorscsatlakoztatását az I2C buszhoz.

- Beépített Cortex M0 mikroprocesszor az adatok feldolgozására és átkonvertálására
- Kommunikáció lehetséges I2C/UART/SPI interfészen keresztül
- Nagy pontosságú gyorsulásmérő (egyenes irányú és forgás irányú)
- Magnetométer (orientáció szenzor)
- Adatkimenet akár m/s^2-ben
- Gravitáció szenzor
- Alacsony válaszidő
- Alacsony fogyasztás
- Működési feszültség 3-tól 5V-ig
- 2 darab STEMMA QT I2C gyorscsatlakozó
- Egyéb beépített funkciók, mint a lépésszámmérés, közlekedési eszköz típusának figyelése, VR eszközként való funkcionalitás [5]



15. ábra: Adafruit 9-DOF Orientation IMU BNO085 mozgásérzékelő és orientációs modul

5.5 Tápellátás és hangjelzés

5.5.1 Tápellátás akkumulátorról

A szünetmentes tápellátáshoz egy LiPo akkumulátor beépítését tervezem, mellyel az eszköz akár több napig is üzemképes lehet. Az akkumulátor rendelkezik beépített védő áramkörrel is.

Az akkumulátor specifikációi:

- Feszültség: 3.7V
- Kapacitás: 500mAh
- Méretkód: 602040 (6mm vastag, 20mm széles, 40mm hosszú)
- Beépített túltöltés, túlmerítés és zárlatvédelem
- JST csatlakozóval van szerelve a Feather boardhoz való könnyű csatlakoztatásért

16. ábra: LiPo akkumulátor illusztrációja

5.5.2 Hangjelzés csipogó segítségével

A riasztások jelzésére egy csipogó beépítését is tervezem, mellyel hangjelzés adható. A csipogó egy egyszerű, saját hanggenerátorral ellátott piezo hangszóró, mely feszültség hatására folyamatos sípolásra képes.

A csipogó specifikációi:

- Működési feszültség: 3-tól 5V-ig
- Hang frekvencia: 2500Hz
- Hangerő: ~85dB
- Működési elv: Piezo
- Méretek: 12x96mm, 2,5mm

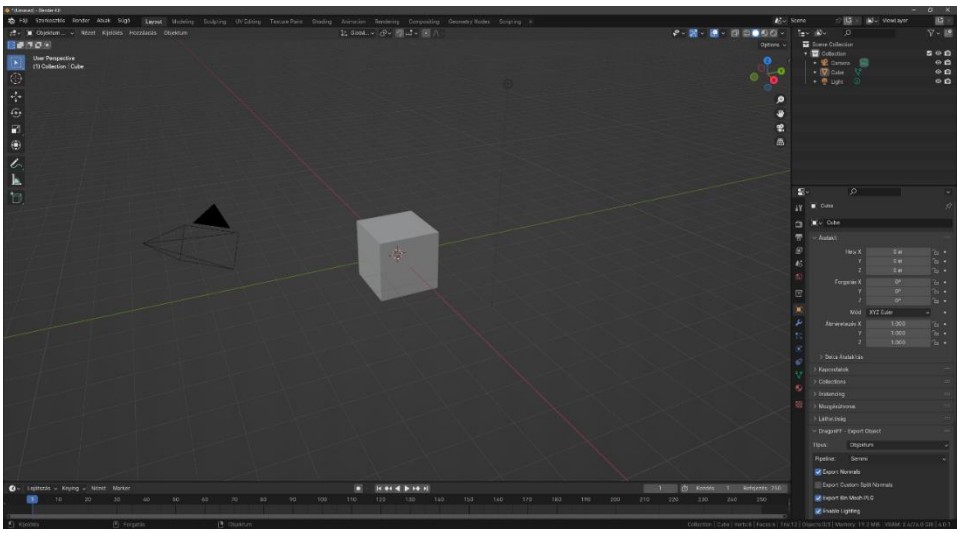
17. ábra: A csipogó illusztrációja

6. INTEGRÁCIÓ

6.1 Mechanika

6.1.1 A rögzítő mechanizmus tervezése

A csuklópánt műanyag részeit a Blender nevű ingyenes program segítségével tervezem meg, majd ezeket a modelleket háromdimenziós nyomtató segítségével legyártanám.



18. ábra: A Blender háromdimenziós szerkesztőprogram felülete

6.1.2 A rögzítő mechanizmus legyártása

Az elkészített háromdimenziós modellt Creality Ender 3 típusú háromdimenziós nyomtatóval gyártom le, fekete, PLA típusú műanyag felhasználásával. [6]

19. ábra: Creality Ender3 háromdimenziós nyomtató

6.1.3 A mechanizmus összeszerelése

Sajnos idő hiányában a csuklópánt tervezése, valamint elkészítése nem valósulhatott meg.

6.2 Elektronika

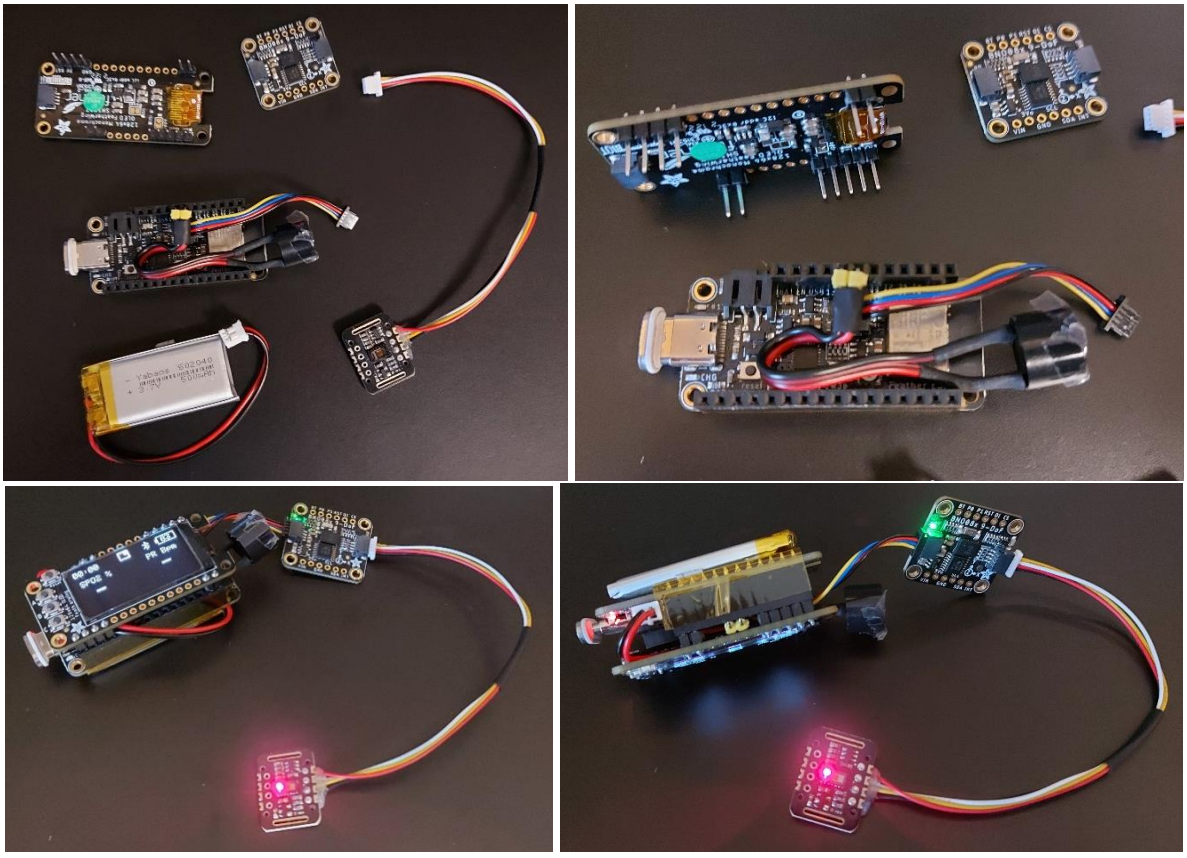
6.2.1 A különböző elektronikai modulok összeépítése

A kijelző modult és az ESP32 nyákját sorcsatlakozókkal egészítettem ki, mely szükség esetén könnyen bontható, valamint stabilan tartja össze a modulokat.

A csipogót szintén ennek a sorkapocsnak a segítségével csatlakoztattam, mely egy vezetékkel lett meghosszabbítva a megfelelő pozíció eléréséhez.

Az orientációs modult a Feather nyákon található STEMMA QT I2C gyorscsatlakozóval csatlakoztattam a mikrokontrollerhez. Az orientációs modul gyártója külön gondolt a szenzor soros felfűzésére, így a nyák kettő darab I2C gyorscsatlakozót tartalmaz, melyre így a továbbiakban a pulzoximéter modulját is rácsatlakoztathattam.

Az akkumulátor rögzítéséről két oldalú ragasztó gondoskodik, mellyel a Feather nyákjára rögzítettem.

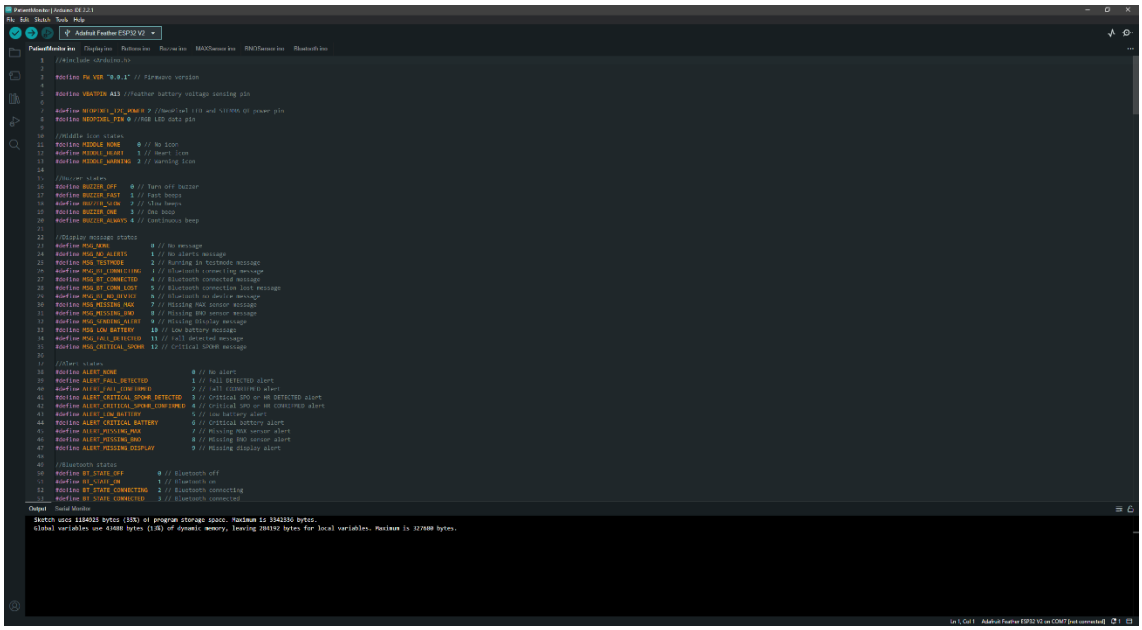


20. ábra: Az eszköz felépítésének képei

6.3 Szoftver

6.3.1 Az ESP32 szoftverének elkészítése

Az ESP32 szoftverét Arduino IDE programkörnyezetben készítettem el. A program kezelése egyszerű, valamint lehetőséget ad különböző kiegészítő könyvtárak és fejlesztői nyákok telepítésére is. Tartalmaz konzolt a soros kommunikációhoz, valamint egy grafikon felületet az ott érkezett adatok grafikai megjelenítéséhez.



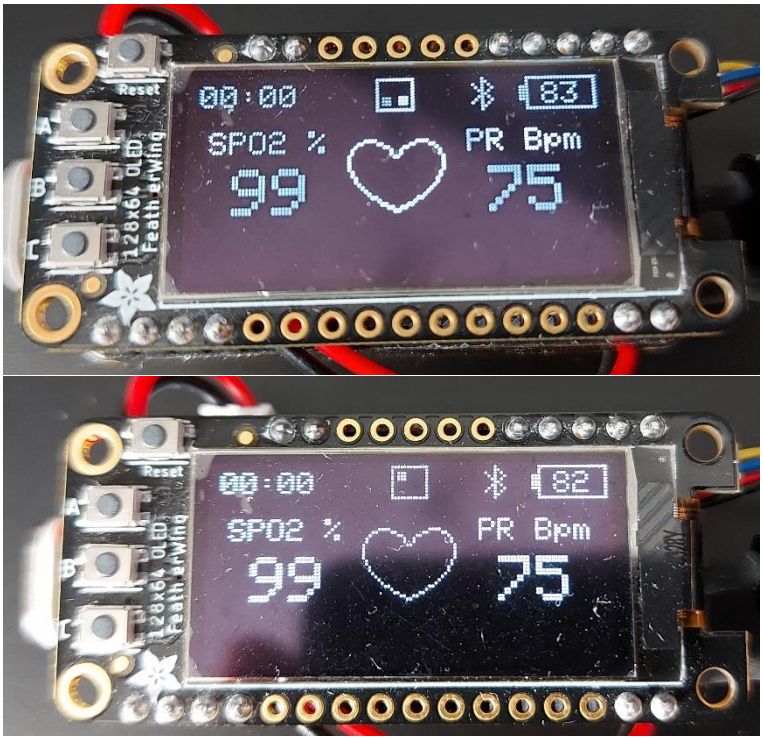
21. ábra: Az Arduino IDE programfelülete

Jelenleg a szoftver képes a szenzorból kinyerni az adatokat, azokat feldolgozni, majd kiírni az eredményt a kijelzőre. Képes soros kommunikációval való adattovábbításra is, így akár kijelző nélkül is lehetséges az adatok kinyerése. Elesés esetén a rendszer jelzést képes leadni egy csipogó segítségével, illetve a szenzorok működését is ellenőrzi indításkor, és hibájuk esetén szintén jelezni tud. Bluetooth kapcsolattal mobilapplikáción nyomon követhető az eszköz állapota, valamint ez az adatokat szintén képes kijelezni, riasztást küldeni.

A tétlenség érzékelésére szolgáló algoritmus nem került megírásra idő hiányában.

6.3.2 Kijelző modul, és kinézetének elkészítése

A kijelzést szinte az elképzelt rendszerrel teljesen megegyezőre sikerült megvalósítanom. A kijelzésen szerepel minden fontos információ melyet az eszköz mér. A kijelzőn a frissítés animáció folyamatosan változik, valamint a középső ikonok is. A riasztásmentes mérés közben a szív ikon animációja a pulzusszámmal megegyező mértékben változik, riasztás esetén pedig egy felkiáltó jel animációja jelenik meg.



22. ábra: Az ESP32 elkészült kijelzésének képei

6.3.3 Gombkezelő modul elkészítése

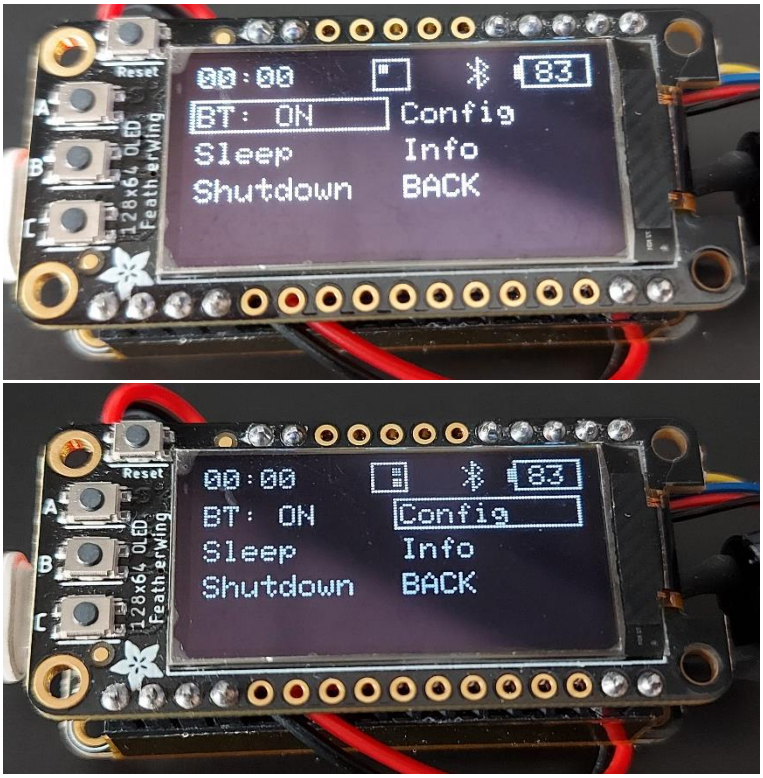
A gombkezelő modul működése az elképzelthez hasonló. A középső gomb segítségével lehetséges a menübe való belépés, a menüpont kiválasztása, hosszas megnyomása esetén az eszköz kikapcsolása.

A felső és alsó gombokkal lehetséges a léptetés vagy a kiválasztott menüpont értékének változtatása. Hosszas megnyomásuk esetén az érték folyamatosan változtatásra kerül.

A gyors reagálást szem előtt tartva, a gombok frissítési rátáját 100ms-onként terveztem meg, ezért a folyamatos lépegetés és a gombok nyomásának ellenőrzése ebben az ütemben valósul meg.

6.3.4 Menürendszer kezelő elkészítése

A menürendszert sikerült az elképzelttel megegyezőre elkészíteni, az opciók egymás alatt, két oszlopban helyezkednek el. A közöttük való léptetés a felső és alsó gombok segítségével lehetséges, a kiválasztásuk pedig a középső gombbal. A kijelölést a kijelölt opció körüli keret mutatja.



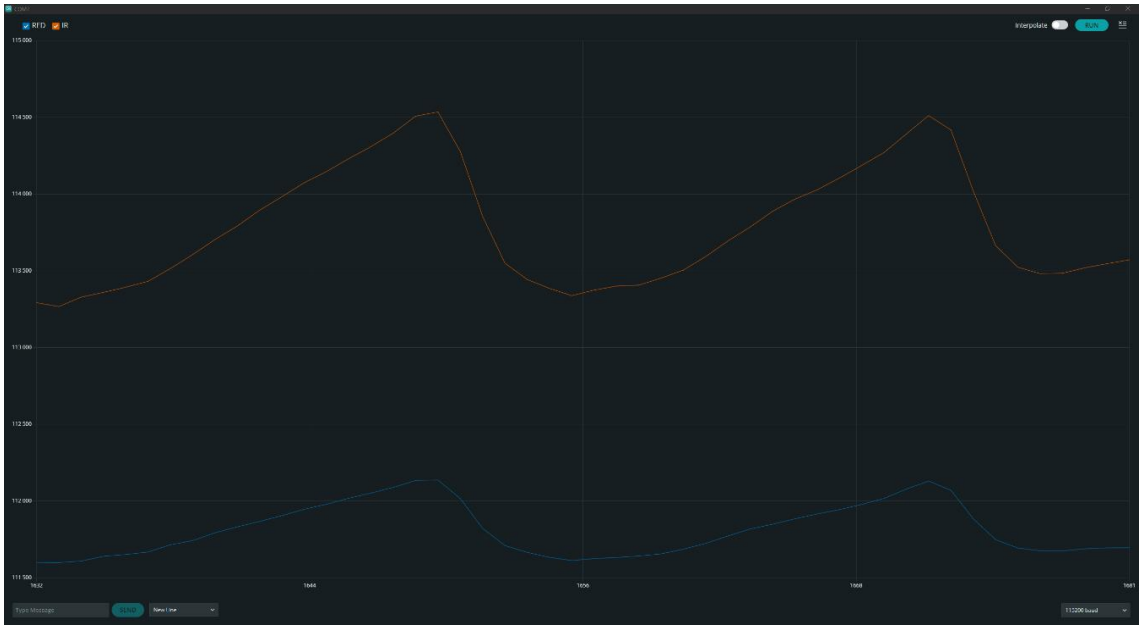
23. ábra: Az elkészült menürendszer kinézetének képei

6.3.5 A pulzoximéter szenzor kezelő elkészítése

A MAX30102 szenzor használatához lehetőségem nyílt már elkészített könyvtárak alkalmazására, melyek tartalmazzák a szenzor által gyűjtött adatokhoz való feldolgozó algoritmust.

A szenzor használatbavételekor számos problémába ütköztem, mert a szenzor sehogy sem akart jó értékeket mérni. Hosszas kutatás után egy másik algoritmus kipróbálását próbáltam meg, ám az sem jutott eredményre. Szerencsére ezen algoritmus készítőjével lehetséges a kommunikáció, valamint már más is szembesült hasonló problémával. Mint kiderült, a hiba a szenzorban van, ugyanis a nem eredeti gyártású szenzorokon a piros, illetve az infravörös LED felcserélve lett beültetve, ezért a két jel fordítva van rögzítve. Szerencsére a hiba javítása egyszerű, a programkódban megoldható, a két rögzített jel tömbjét csak megcserélni szükséges. A hiba orvoslása után a mérés már jó és pontos értékeket mutatott.

A szenzor által rögzített adatok kijelzésére a fejlesztőkörnyezet egy grafikai megoldást is kínál, mellyel jól szemléltethetőek az algoritmus által felhasznált jelek hullámzásai:



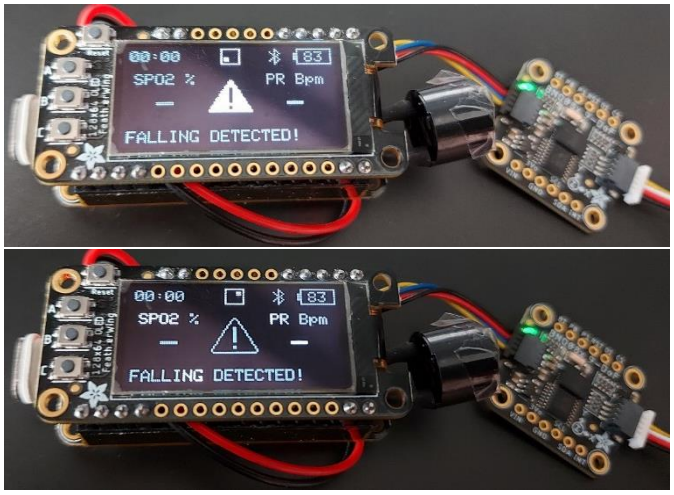
24. ábra: A pulzoximéter által rögzített nyers értékek grafikonja

Az ábrán található jelölések magyarázata:

- RED: (kék vonal) a piros LED hatására történő lüktetés értékei
- IR: (piros vonal) az infravörös LED hatására történő lüktetés értékei

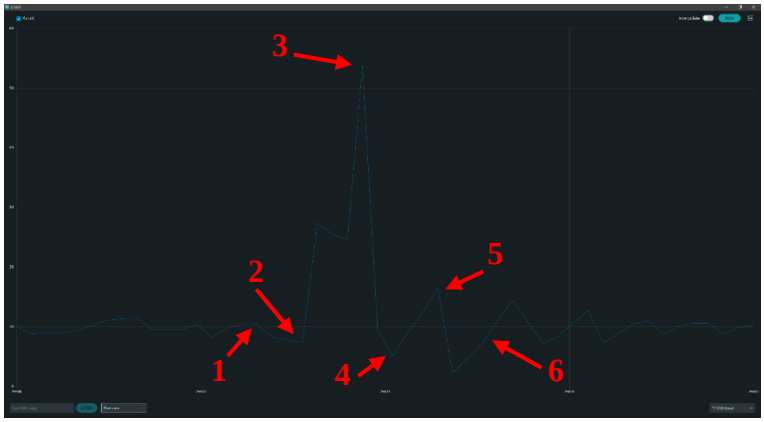
6.3.6 Orientáció és mozgásszenzor kezelő elkészítése

Az mozgásszenzor kezelésére szintén létezik elkészített könyvtár, mellyel a használata egyszerűen megoldható volt. A kapott adatokat saját függvényekkel dolgoztam fel, mely az elképzelt mód alapján működik és lefutása esetén jelzést küld.



25. ábra: Az elesés riasztásának képe

A korábban bemutatott grafikai megjelenítővel szemléltethető a mozgásszenzor által mért gyorsulás is. A következő ábrán jól kivehető az esés kezdete, annak vége, majd a felületről való visszapattanás pillanatai.



26. ábra: Az orientációs modul által rögzített gyorsulás grafikonja

Az ábrán található jelölések magyarázata:

1. Nyugalmi helyzet
2. Az esés kezdete, gyorsulás
3. Gyorsulás vége, maximális esési sebesség elérése
4. Az esés vége, becsapódás pillanata
5. A felületről való visszapattanás pillanata
6. Az visszapattanás lecsengése, nyugalmi helyzetbe kerülés

6.3.7 Bluetooth kapcsolat kezelő elkészítése

A Bluetooth kapcsolathoz az úgynevezett BLE(Bluetooth Low Energy) protokollt választottam, mivel ez nem igényel folyamatos kapcsolattartást, így energiát spórolva, és az akkumulátoros üzemidőt növelve.

A BLE protokoll nem folyamatos, hanem egy csomagorientált kapcsolat, ezért tud hatékonyan, kevés energiafelhasználással is „aktív” maradni.

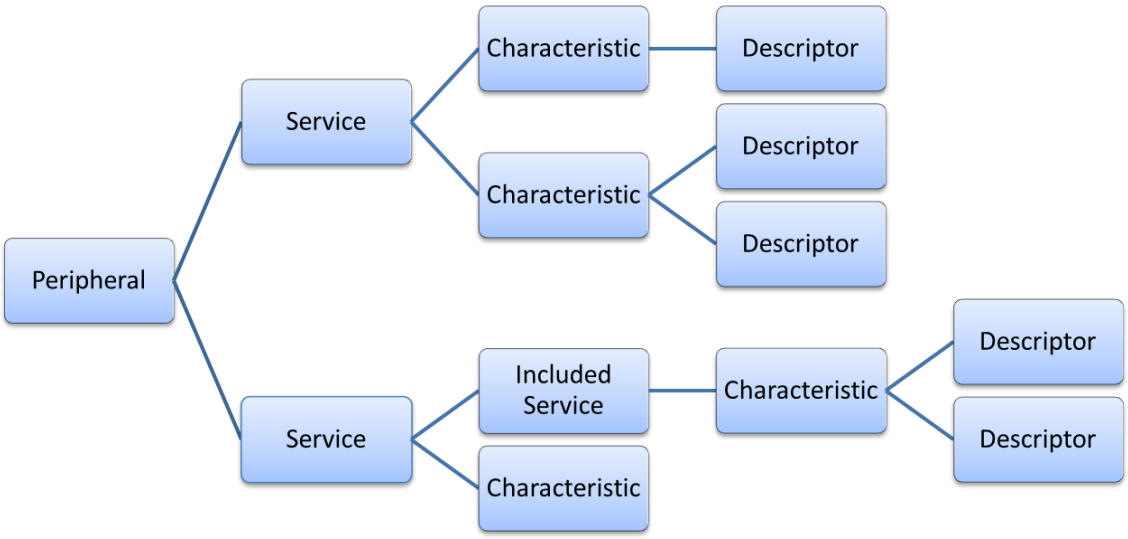
A kapcsolat „szerver-kliens”(service-client) felállásban valósul meg. A szerver esetünkben az ESP32, mely megadott értékeket, például a pulzust, oxigénszintet meghatározott időpontokban „hirdeti”.

Ezeket a hirdetett „szolgáltatásokat”(service) a kliens, esetünkben a mobiltelefon fogadja, jelzi ki, és visszajelzést küldhet rá.

A szolgáltatások egy vagy több „karakterisztikát”(characteristic) tartalmazhatnak, melyek az adatot tartalmazzák. Az karakterisztikák tartalma egyénileg testre szabható, több részre bontható.

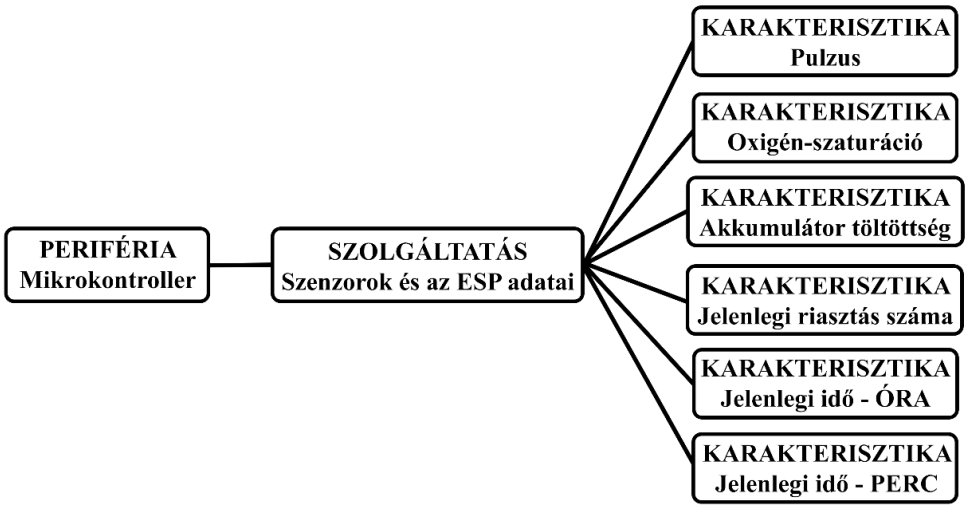
A „leírók”(descriptor) segítségével további információk adhatóak meg, hogy az adott karakterisztika a szolgáltatáson belül milyen feladatot lát el, mit tartalmaz.

A szolgáltatásoknak és karakterisztikáknak egyedi azonosító számuk un. „UUID” számuk van, mely egy számokból és betűkből álló egyedileg generált sorozat. [7]



27. ábra: A BLE szerver által hirdetett szolgáltatások bemutatóvázlata

Az általam elkészített modul BLE hirdetési ábrája a következő:



28. ábra: Az elkészített BLE modul hirdetési ábrája

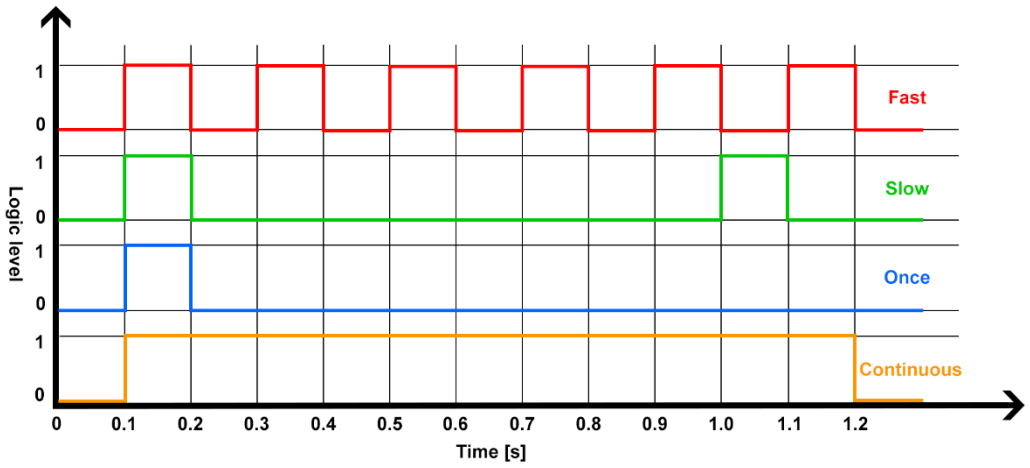
6.3.8 A csipogó kezelőmoduljának elkészítése

A csipogó modulja lehetővé teszi különböző hangriasztások lejátszását.

A modullal lehetőség van folyamatos sípszó vagy riasztás esetén gyors, illetve lassú ütemű sípszó küldésére, valamint egyszeri sípszó lejátszására.

A modul indításkor felkonfigurálja a kimeneteket melyen a csipogó található és megadott időközönként frissíti ezen kimenetek állapotát. A csipogó vezérlésére függvények segítségével van lehetőség, mely lefutása után a csipogó a következő frissítés alkalmával reagál.

A csipogó jeldiagramja:



29. ábra: A csipogó vezérlésének logikai diagramja

6.3.9 Az akkumulátor töltöttségének mérése

Az akkumulátor töltöttségének mérésére egy dedikált bemenet szolgál a Feather tesztnyákon, mely analóg feszültség értékének lekérésével megkapható a jelenlegi töltöttség szintje.

A feszültség mérése:

$$vbatt = analogReadMilliVolts(VBATPIN) * 0.002;$$

- **vbatt:** az akkumulátor jelenleg mért feszültsége, voltban
- **analogReadMilliVolts:** Az adott bemeneten lévő feszültség értéke millivoltban
- **VBATPIN:** Az adott kimenet száma/neve
- **0,002:** Az adott bemeneten egy feszültség osztó van, ezért a mért feszültséget szükséges 2-vel felszorozni, majd az értéket 1000-rel elosztani a voltba való átszámításhoz

A mért feszültségből százalékká való átalakítás a következő képlet szerint végezhető el:

$$state\_battery = round(vbatt - 3{,}2) * 100;$$

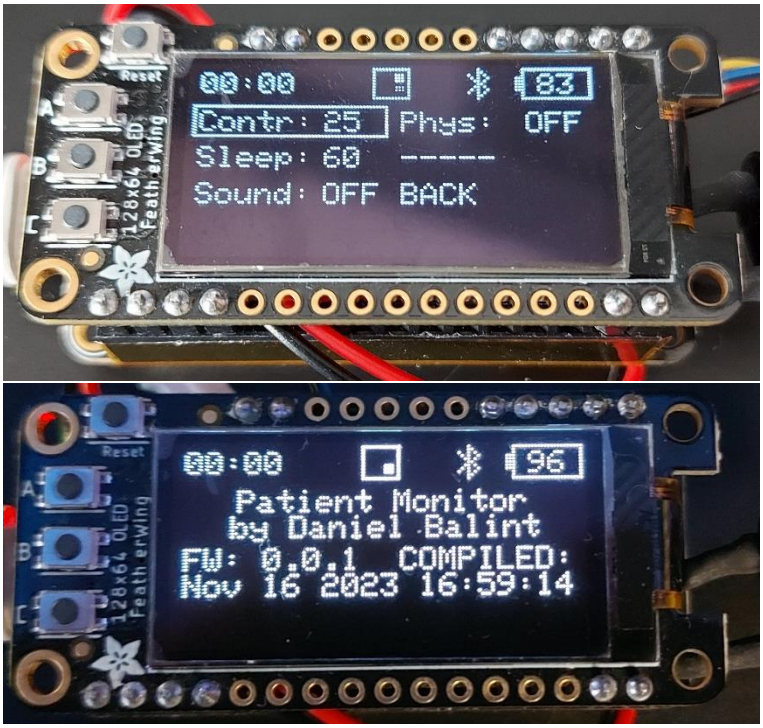
- **state_battery:** a töltöttségi érték százalékban
- **round:** szükséges az érték egész számmá kerekítése
- **vbatt:** az akkumulátor jelenleg mért feszültsége, voltban
- **3,2:** az akkumulátor feszültsége, teljesen lemerült állapotban (0%)

6.3.10 Kényelmi és egyéb funkciók hozzáadása

A menürendszerben hozzáadásra kerültek különböző kényelmi, illetve beállítási lehetőségek.

Ezek között szerepel a program létrehozásának dátuma, a kijelző paramétereinek megváltoztatása (fényerő, kikapcsolási időzítő), a kikapcsolási, valamint alvó módba lépési lehetőség, a Bluetooth kapcsolat engedélyezésének vagy tiltásának lehetősége, a hangjelzések be és kikapcsolásának lehetősége.

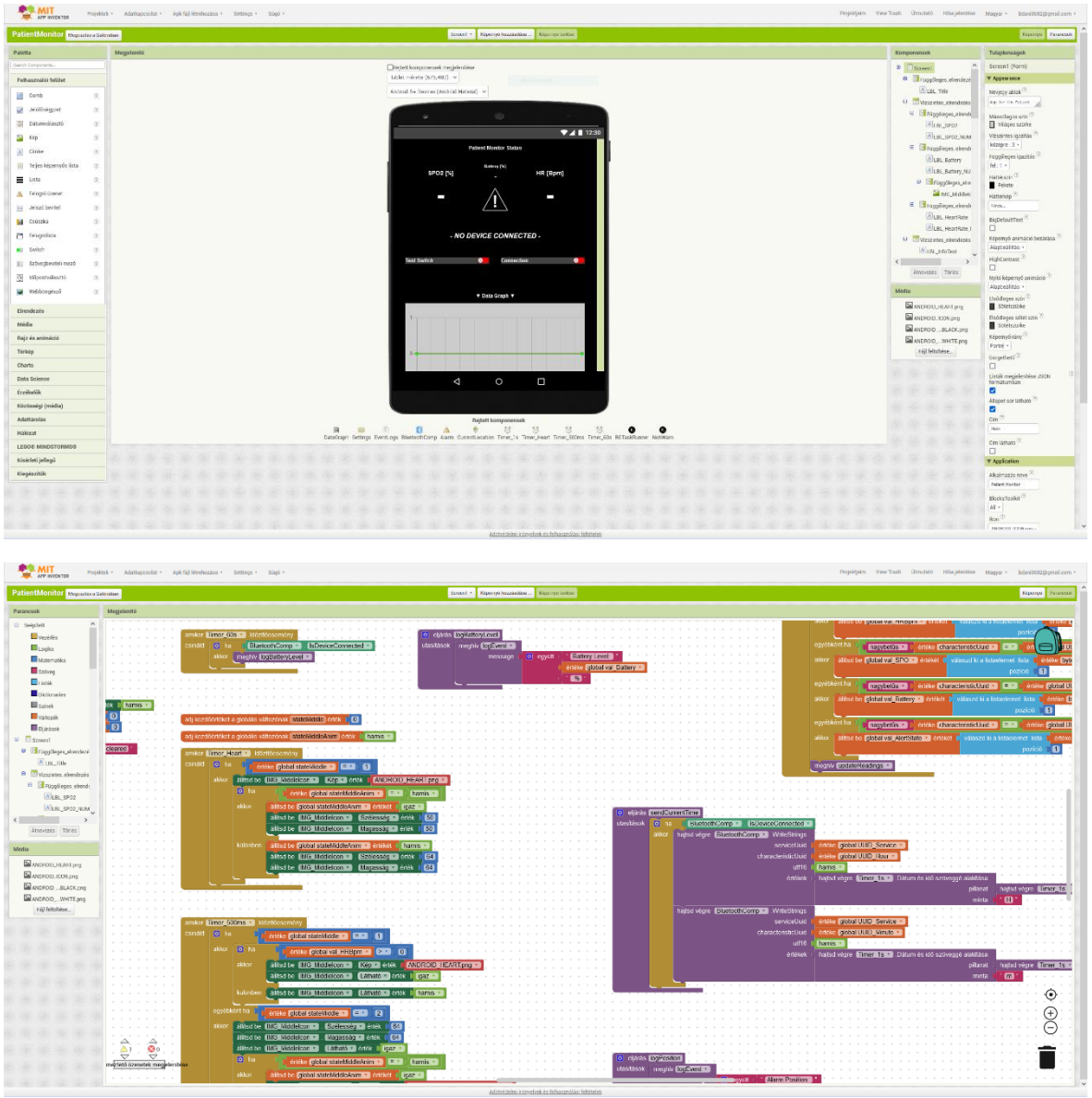
Jelenlegi állapotában a szoftver még nem képes tárolni ezeket a beállításokat, ezért újraindítás után a beállítások elvégzése ismételten szükségessé válik.



30. ábra: A beállítás és információs menü képei

6.3.11 Az Android mobilapplikáció elkészítése

Az Android applikációt az MIT App Inventor webes tervező program segítségével készítettem el. A program segítségével lehetőség nyílik egyszerűbb alkalmazások létrehozására, valamint azok azonnali tesztelésére is a tervezés közben. Az elkészítéshez szükséges volt különböző külső bővítmények importálására is, ilyenek voltak például a BLE kezelő beépülő modul és a figyelmeztető üzeneteket küldő kiegészítő modul. Az alkalmazás két részre bontható, egy grafikai tervező felületre, valamint a programozás felületére, ahol a bizonyos objektumok funkciója szabható meg. A programozás blokk rendszerű, aminek elsajátítása és kezelése egyszerű volt.

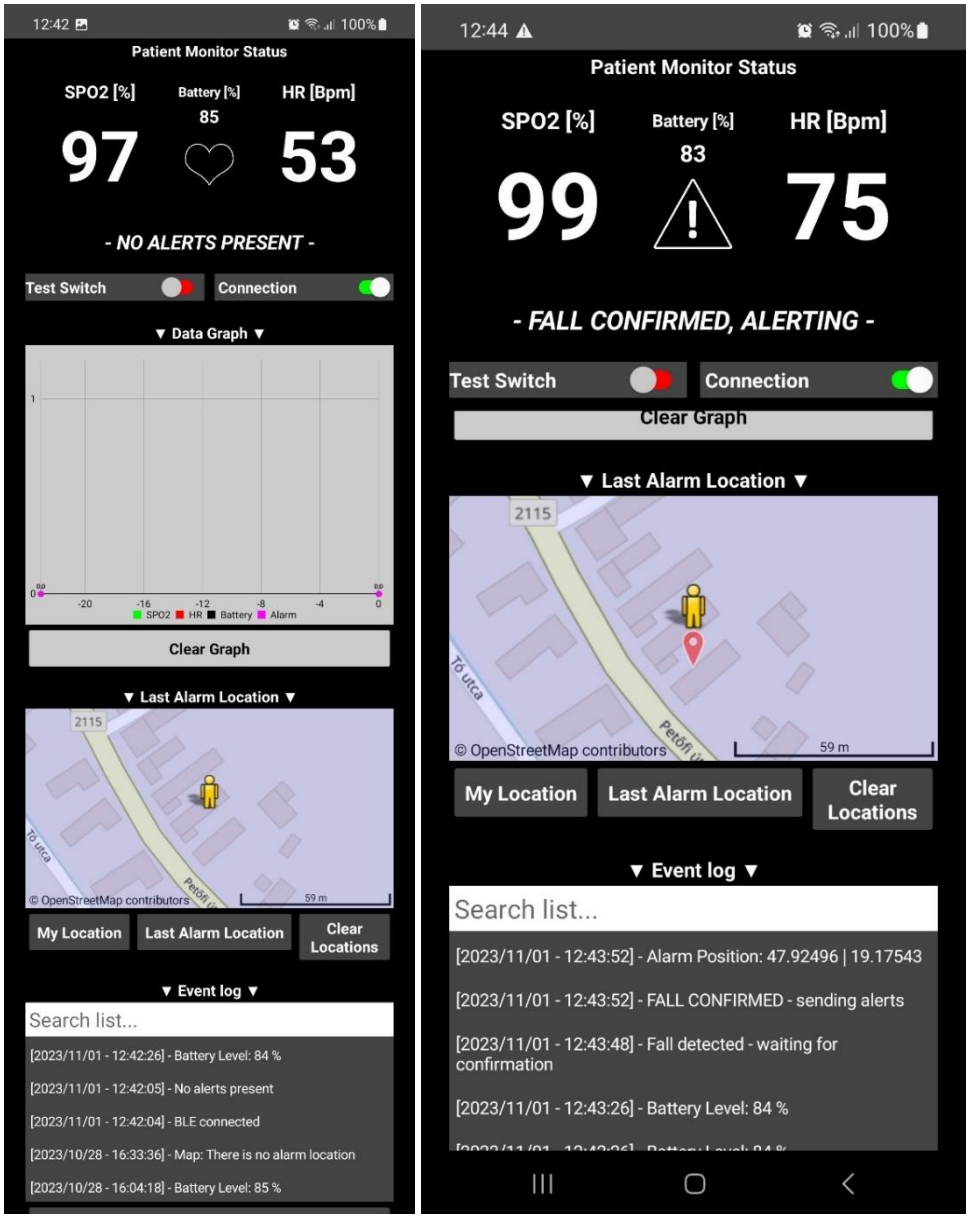


31. ábra: Az MIT App Inventor felületének képei

6.3.12 Az Android alkalmazás funkcióinak elkészítése és leprogramozása

A program a tervezetthez képest kissé eltérőre sikeredett, de minden elképzelt funkcióját sikerült elkészítenem.

A program képes a mikrokontroller által rögzített adatok fogadására és azok kijelzésére. A program ezenkívül képes riasztás esetén a hely pontos rögzítésére, valamint riasztás leadására. Az eseményeket egy listában rögzíti, melyt egy fájlba ment el. Az elmúlt 24 órában rögzített adatokat képes egy grafikonon ábrázolni.



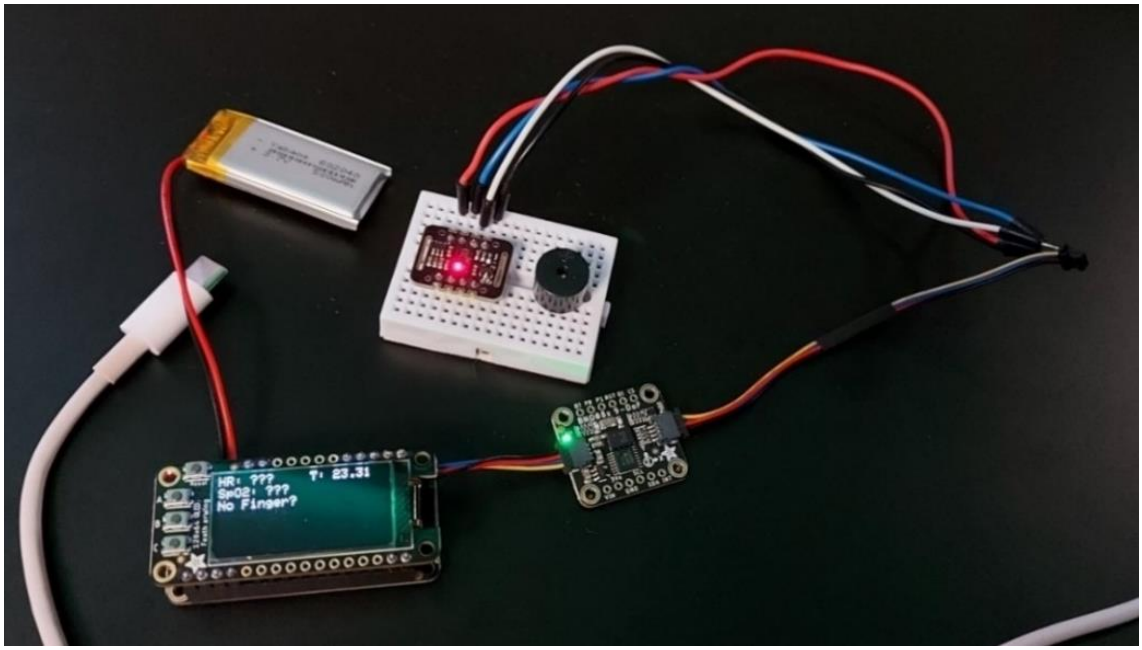
32. ábra: Az elkészült Android alkalmazás képernyőképe

7. TESZTELÉS

7.1 Az ESP32 szoftverének tesztelése

7.1.1 A modulok működésének ellenőrzése

A modulok működését az első összeállítás után egy egyszerűbb tesztkódokkal végeztem, hogy megbizonyosodjak azok hibátlan működéséről. Ezen teszten sajnos a pulzoximéter modulja megbukott, melyt egy hosszas kutatás követett, ami kiderítette, hogy a szenzor gyártási hibával rendelkezik, valamint, hogy ennek a hibának a javítása nagyon egyszerű, a programkódban javítható.



33. ábra: A pulzoximéter tesztelése breadboardos felépítésben

7.1.2 A kijelzés és menü tesztelése

A kijelzést a programrész megírása után egy 0-tól 200-ig felfelé számoló segítségével ellenőriztem, hogy megbizonyosodjak arról, hogy a kijelzett értékek teljes egészében kiférnek a kijelzőre.

A kijelzett értékek minden alkalommal kifértek a kijelzőre, így további finomhangolás nem volt szükséges.

7.1.3 A pulzoximéter modul tesztelése és mérésének hitelesítése

A pulzoximéter modul által mért adatokat egy készen vásárolt pulzoximéter segítségével hitelesítettem, hogy megbizonyosodjak a mérés pontosságáról.

Az algoritmus és a szenzor beállításainak finomhangolása után a két eszköz szinte megegyező adatokat mért.

34. ábra: A hitelesítéshez felhasznált pulzoximéter

7.1.4 Az orientációs modul és esésérzékelés tesztelése

Az orientációs modul működését a mellékelt könyvtárak segítségével ellenőriztem le, mellyel a szenzor működőképesnek bizonyult.

Az esésérzékelő algoritmus megírása után az esés érzékelésének tesztelését az eszköz hirtelen elmozdításával, megütésével szimuláltam, melyen az algoritmus több próbálkozás és finomhangolás után riasztást jelzett.

7.1.5 A csipogó tesztelése

A csipogó tesztelését a már kész programmal végeztem el, mellyel egy esést szimulálva riasztás történt és a csipogó ennek hatására jelezni kezdett.

7.1.6 A riasztások tesztelése

A különböző riasztásokat az őket kezdeményező módszer szimulálásával végeztem el.

- A hibás vagy hiányzó szenzorok riasztásait az adott szenzor rendszerből való eltávolításával végeztem, melyen az adott riasztások meg is szólaltak.
- Esés riasztása esetén a már korábban említett ütéses tesztelési módszert alkalmaztam, melyen a riasztás szintén megszólalt.
- Alacsony akkutöltöttség esetén megszólaló jelzést az akkumulátor teljes egészében való lemerítésével teszteltem le, melyen a riasztások szintén megszólaltak.

7.1.7 Az akkumulátoros üzemidő tesztelése

Az akkumulátoros üzemidő tesztelését az akkumulátor teljes feltöltésével kezdtem, majd a töltés megszüntetése után stopperórával elkezdtem mérni az eltelt időt.

A tesztelés során az akkutöltöttség mérőt figyelve ellenőriztem a kijelzés pontosságát, valamint a megadott töltöttségi szint esetén jelentkező riasztásokat is. Riasztás történik 25%-on, mely a felhasználó tudtára hozza, hogy hamarosan töltés válik szükségessé, ez a riasztás 5 óra 20 perc után szólalt meg, valamint 15%-on, ahol egy sokkal intenzívebb jelzés történik, hogy a töltöttség kritikusan alacsony, ez a jelzés 5 óra és 30 perc után történt meg.

A további használat során kiderült, hogy az eszköz ez után még képes működni amíg az akkumulátor védő áramköre le nem kapcsol, és így az üzemideje maximálisan 5 óra és 40 perc.

Az üzemidő kissé kevésnek tűnik, mivel így több töltés is szükséges a nap folyamán, és ez zavaró. Az üzemidőt esetlegesen szoftveres javítással lehetne növelni, például a pulzoximéter szenzor mérési idejének korlátozásával, hogy ne folyamatosan mérjen, csak bizonyos időnként. Lehetséges lenne az üzemidőt növelni egy nagyobb kapacitású akkumulátorra való váltással is.

7.2 Az Android alkalmazás tesztelése

7.2.1 A kijelzés tesztelése

A kijelzést a korábbi tesztmódszerhez hasonlóan, egy 0-tól 200-ig számlálóval ellenőriztem, hogy megbizonyosodjak a kijelzés hibátlanságáról.

A teszt közben a kijelzés nem mindig volt megfelelő, ezért finomhangolásra volt szükség. A kijelzés ezután már megfelelően működött.

7.2.2 A BLE csatlakozás és adattovábbítás, riasztások tesztelése

A vezeték nélküli kapcsolatot a már kész eszközön lévő programmal teszteltem, mely ekkor már a valós, mért adatokat küldte, és ezt az Android alkalmazás pontosan ki is jelezte.

A különböző riasztások megszólalása esetén az alkalmazás is megfelelően reagált, azonos jelzést adott le a telefonon, az eszköz riasztásától számítva kisebb reakcióidővel eltérően.

Az eszközön lévő óra is megfelelően működik amíg a kapcsolat aktív. A kapcsolat megszűnésével az óra az utolsó időpontot mutatja amikor a kapcsolat még működött.

8. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Az eszköz továbbfejlesztésre látok pár lehetőséget, mivel a jelenlegi rendszer eléggé körülményesnek tekinthető.

Lehetőséget látok a rendszerbe mobil internet kapcsolat tartására való modem beépítésére, mivel így nem szükséges a telefonnal való kapcsolattartás az adatok nyomon követéséhez. Ez azt a hátrányt is megoldja, mely akkor jelentkezik, ha a felhasználó a mobiltól túlságosan távol kerül, mivel ekkor a Bluetooth kapcsolat megszűnik. Ebben az esteben szükséges GPS vevő beépítése is, hogy a hely is pontosan lekérdezhető legyen. Egy központi gyűjtőszerver felhasználása is szükségessé válik az adatgyűjtésre és tárolásra.

Hardveres továbbfejlesztésben esetlegesen a fizikai visszajelzés is hozzáadható lenne (riasztás esetén rezgő visszajelzés), vagy beszélgetéses kommunikációhoz mikrofon és hangszóró. Az akkumulátor üzem idejének növelése is egy cél lehet, valamint az egész műszer méretének további csökkentése.

Szoftveres oldalon szintén sok lehetőség nyílna még a továbbfejlesztésre, például egy jobb mobilalkalmazásra, valamint az eszköz esetén könnyebben kezelhető és testre szabhatóbb felületre.

A műszer házának vízállóvá tételével szintén továbbfejleszthető lenne a projekt, mivel jelenlegi esetben az eszköz nem vízálló, és így például fürdés közben nem viselhető, mely egy nagy veszélyforrás.

9. ÖSSZEFOGLALÁS ÉS TAPASZTALATOK

A szakdolgozatom során az elképzelt terveim nagy részének megvalósítása sikerült, a kisebb elakadások nem befolyásolták számottevően az eredményt. Az egyik legnagyobb probléma a pulzoximéter moduljával akadt, ezen hibának megoldásával sok időt töltöttem el. A program megírása szintén sok időt emésztett fel, ezért már nem jutott idő pár dolog kivitelezésére, például a tétlenség érzékelésének megírására, valamint a rögzítőszerkezet elkészítésére.

A rendszer működési alapjainak kidolgozását a projekt 1. tárgy keretében kezdtem meg, és itt még egy egyszerűbb rendszert alkalmaztam. Ez működőképesnek bizonyult, viszont sok hibával rendelkezett, köztük a pulzoximéter mérési hibájával. Ennek a rendszernek az elkészítése viszonylag sok időt emésztett fel, mert a szenzor csak külföldről volt megrendelhető, és a kiszállítás sok időt vett el.

A rendszert a projekt 2. tárgy keretében tovább javítottam, itt került a rendszer nagy átalakításra, melyben a korábbi mikrokontroller és kijelző le lett cserélve egy erősebbre, valamint itt került hozzáadásra a mozgásérzékelő modul is. A pulzoximéter mérési hibájának okát itt kezdtem el komolyabban vizsgálni. A mozgásérzékelő modul tesztelésekor egy új problémával is szembesültem, ugyanis a modul nem volt hajlandó kommunikálni a mikrokontrollerrel. Ezen hibát viszonylag gyorsan orvosolni tudtam a kezelőkönyvtár frissítésével.

A szakdolgozat tárgy keretében a rendszer teljes vezérlő programjának megírása, valamint Android eszköz támogatásának megírása valósult meg. A rögzítő szerkezet megvalósítását is ezen tárgy keretében szerettem volna kivitelezni, ám ez idő hiányában nem valósult meg. A vezérlőprogram az általam elképzelt funkciók nagy részével rendelkezik, feladatát ellátja. Az Android program szintén rendelkezik minden általam elképzelt funkcióval, melyek működése is megfelel az elvártaknak.

Összességében a projektem és szakdolgozatom eredményével elégedett vagyok, kitűzött céljaim nagy részének elérését sikeresnek tartom. A jövőben az időbeosztáson javítanék, mert a jelenlegi projekt kivitelezésének egy része ezért nem valósulhatott meg.

10. SUMMARY AND EXPERIENCES

During my thesis, most of my plans are implemented, there was small obstacles, but they did not modified the result that much. One of the biggest problem was with the pulseoximeter module, i have spent a lot of time investigating what caused it. Writing the firmware also took a lot of time, and by this reason i had no time for a few things to finish, for example the inactivity detection, or the holder structure.

I have started working on the basic principle during the subject of project 1. Here i have used a much simpler system than the finished one. It worked, but it did had a lot of problems, like the pulseoximeter sensor's measuring problem. Making this system took a bit too long, because i was only able to order the sensor from abroad, and the shipping took time.

On the subject of project 2. i further corrected the system, here i did big modifications, where i changed the microcontroller to a much stronger one, and the screen to a better one, and i also added the motion sensor into the system. I also started to investigate the problem with the pulseoximeter sensor here. While setting up the motion sensor, i ran into a new problem, because it was not able to communicate with the microcontroller. Fixing this problem was fortunately not hard, only its library needed an update.

During my subject of thesis, i have implemented both the system's complete firmware, and the support for Android devices. During the subject, i would have liked it to make the holding structure, but i was not able due to running out of time. The firmware has almost every function i have planned for it, and it do function well. The Android application also have every planned function, and it also do function as well.

Overall, i'm satisfied with the result of my project and thesis, i did reached most of my set goals, and i think most of it was successful. In the future, i would adjust on my schedule in time, because some part of this project wasn't finished due to this reason.

11. IRODALOMJEGYZÉK

[1] „GondosÓra program,” [Online]. Available: https://gondosora.hu/. [Hozzáférés dátuma: 19. 11. 2023.].

[2] „Pulzoximéter szenzor összeállítása, működése,” [Online]. Available: https://lastminuteengineers.com/max30102-pulse-oximeter-heart-rate-sensor-arduino-tutorial/. [Hozzáférés dátuma: 25. 10. 2022.].

[3] „Adafruit ESP32 Feather V2 fejlesztői nyák leírása,” [Online]. Available: https://www.adafruit.com/product/5400. [Hozzáférés dátuma: 10. 04. 2023.].

[4] „Adafruit FeatherWing OLED modul leírásai,” [Online]. Available: https://www.adafruit.com/product/4650. [Hozzáférés dátuma: 10. 04. 2023.].

[5] „Adafruit 9-DOF Orientációs modul leírása,” [Online]. Available: https://www.adafruit.com/product/4754. [Hozzáférés dátuma: 10. 04. 2023.].

[6] „Creality Ender 3 3D nyomtató,” [Online]. Available: https://www.creality.com/products/ender-3-3d-printer. [Hozzáférés dátuma: 19. 11. 2023.].

[7] „Bluetooth Low Energy működése,” [Online]. Available: https://doc.qt.io/qt-6/qtbluetooth-le-overview.html. [Hozzáférés dátuma: 19. 11. 2023.].

12. ÁRBAJEGYZÉK