



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Automatizálási és Alkalmazott Informatikai Tanszék

Okosóra készítése

Projektfeladat mechatronikusoknak

Készítette: Nemes Balázs Boldizsár

Konzulens:

Dr. Iváncsy Szabolcs
címzetes egyetemi docens

Budapest, 2023

Tartalomjegyzék

1. Feladat leírása	3
2. Használt szoftverek, programnyelvek	4
3. Hardver kiválasztása	4
3.1. Mikrovezérlő	4
3.2. Perifériák	5
4. Blokkdiagram	7
5. A szoftver fejlesztése	8
5.1. Gyártói LCD kezelő szoftver	8
5.2. A feladathoz szükséges struktúrák	8
5.3. Fő programot egyszerűsítő könyvtárak	9
5.4. Fő program	9
5.4.1. A program állapotábrája	9
5.5. Gombok elhelyezkedése, funkciója	10
5.6. Időbeli pontosság	10
5.7. Üzem módok	11
5.8. Az aktív üzemmód menü szerkezete	12
5.8.1. Time display menu	12
5.8.2. Set alarm menü	12
6. Továbbfejlesztési ötletek, lehetőségek	14
7. Jelenlegi állapot	15

1. Feladat leírása

A projektfeladat célja egy viselhető, okos funkciókkal ellátott óra készítése volt.

A megvalósítandó feladatok:

- Dátum és idő pontos kijelzése
- Beállítható ébresztések (dátum és időpont)
- Időzítő funkció
- Vezeték nélküli kommunikáció
- Alacsony energiafogyasztás

2. Használt szoftverek, programnyelvek

Az arduino programozása előre definiált könyvtárakkal kiegészített C++ nyelven történik.

Az Arduino felprogramozását az Arduino IDE szoftver^[1], és az Arduino IoT rendszer^[2] segítségével végeztem el.

3. Hardver kiválasztása

3.1. Mikrovezérlő

A feladat megvalósításához egy Arduino Nano 33 IoT mikrokontrollert választottam.

A mikrovezérlő előnyei:

- Magas órajel (fontos az SPI kommunikációt használó LCD kijelző megfelelő frissítéséhez)
- Integrált WiFi és Bluetooth modul (vezeték nélküli kapcsolathoz)
- RTC modul (pontos idő nyomonkövetéséhez)
- Fejlesztői környezet (Arduino IDE szoftver)
- Előre készített könyvtárak könnyebbé teszik a perifériák kezelését
- Kiterjedt support (interneten könnyen lehet megoldást találni a felmerülő problémákra)

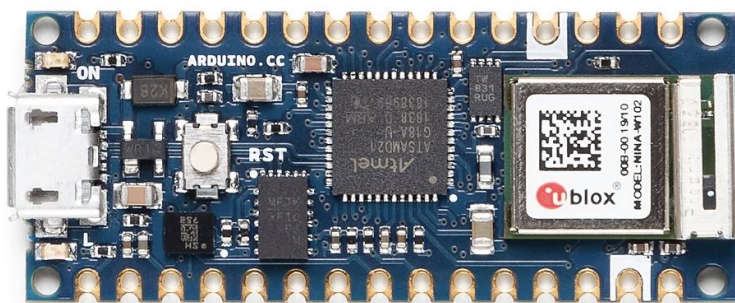
Hátrányai:

- A feladat esetpéén felmerülő hely-korlátozás miatt relatív nagy méret
- Kis mértékben személyre szabható (pl.: pin-ek tűskesorban -> nehéz NYÁK-ra integrálni)
- Ár (a feladat megvalósításához feleslegesen nagy tudás; jóval drágább, mint egy kisebb tudású mikrovezérlő)

Műszaki paraméterek:

USB csatlakozó	Micro USB
WiFi és Bluetooth	Nina W102 uBlox modul
Üzemi feszültség	3.3 V
Processzor	ATSAMD21G18A
Órajel	48 MHz
Memória	256 KB SRAM, 1 MB flash

3.1. táblázat – Arduino Nano 33 IoT műszaki paraméterei^[4]



3.1. ábra – Arduino Nano 33 IoT

3.2. Perifériák

A feladat megoldásához választott perifériák:

- Waveshare 1.28 inch kerek LCD kijelző^[5]
 - Felbontás: 240x240 pixel
 - 65K RGB szín kijelzés
- Nyomógombok^[6]
- Rezgő motor^[7]
- Elektromágneses hangjelző^[8]



(a) Waveshare 1.28 inch LCD



(b) Nyomógombok



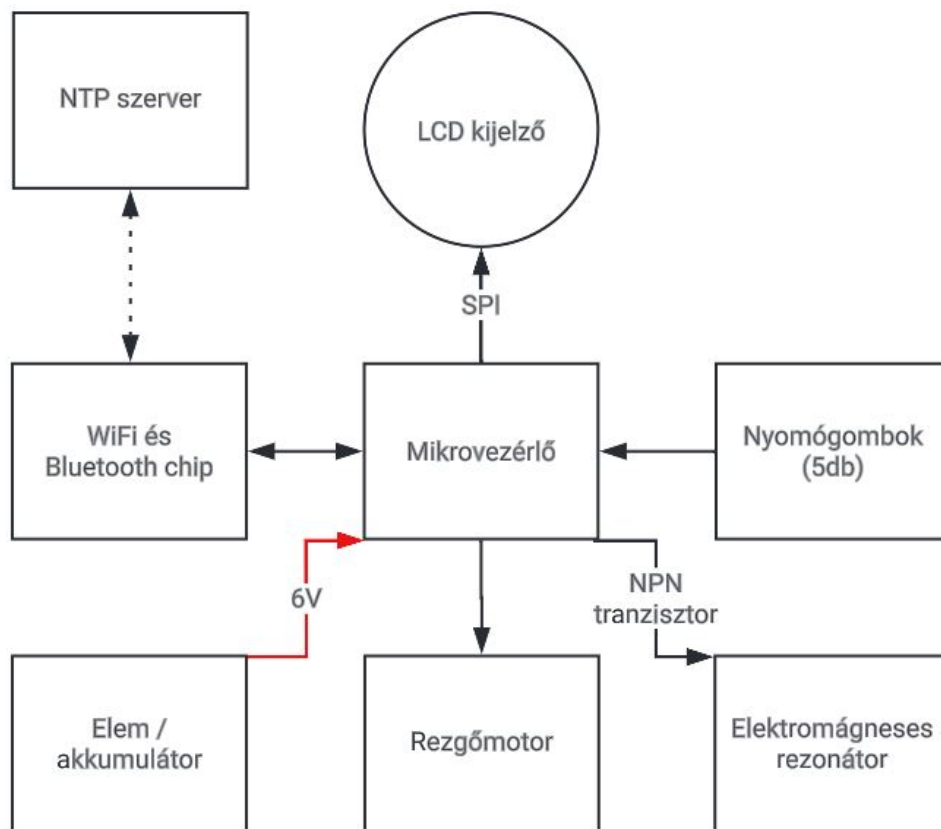
(c) Rezgő motor



(d) Elektromágneses hangjelző

3.2. ábra – A felhasznált perifériák

4. Blokkdiagram



4.1. ábra – A rendszer blokkdiagramja

5. A szoftver fejlesztése

5.1. Gyártói LCD kezelő szoftver

A gyártó által kiadott szoftver az LCD felkonfigurálását végzi el. A szoftver öt különböző méretű betűtípust tartalmaz, emellett alapvető "rajzoló" funkciókat tartalmaz.

A szoftver függvényei:

- A képernyő elforgatása, tükrözése (inicializáláskor)
- A kijelző "törlése" (egy szín beállítása)
- String-ek, számok kiírása adott helyre
- Pixelek rajzolása
- Szakaszok, vastagabb vonalak, téglalapok és körök rajzolása

5.2. A feladathoz szükséges struktúrák

A feladat megoldásához létrehoztam két struktúrát. Ezek a *date* és *alarm* nevet kapták.

```
1 struct date {
2     int year;
3     int month;
4     int day;
5     int hour;
6     int minute;
7     int second;
8     int weekday;
9     unsigned long TIME;
10    unsigned long DATE;
11 };
```

5.1. kód – "date" struktúra

```
1 struct alarm {
2     String name;    // slot name
3     date time;      // time of alarm
4     int style;       // 0-OFF 1-vibrate
5                     // 2-vibrate and sound
6     int repeat;     // 0-once 1-weekly
7                     // 2-monthly
8 };;
```

5.2. kód – "alarm" struktúra

5.3. Fő programot egyszerűsítő könyvtárak

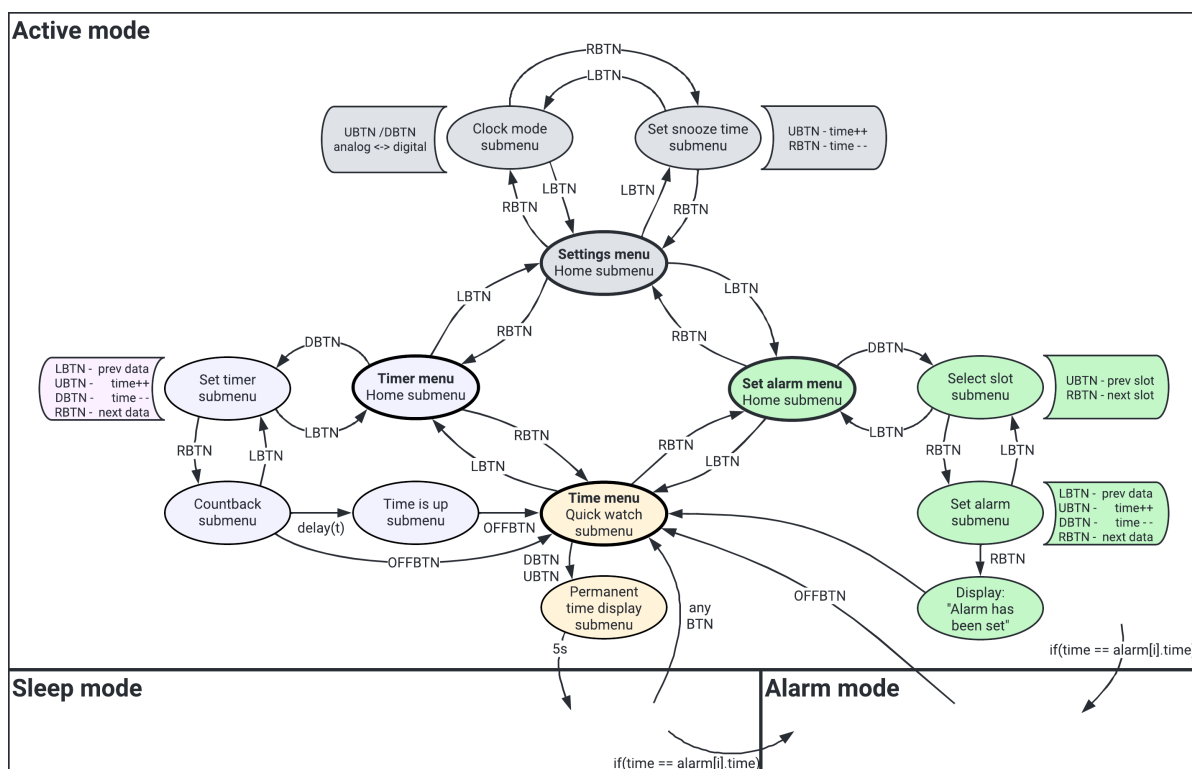
A program függvényeit kiszerveztem header fájl formájában.

A header fájlok:

- LCD setup - inicializálás
- LCD GUI - rajzoló, kiírató függvények
- Idő kijelző függvények
- WiFi init, kezelés
- RTC modul kezelés

5.4. Fő program

5.4.1. A program állapotábrája



5.1. ábra – A program állapotábrája

5.5. Gombok elhelyezkedése, funkciója

Az óra számlapja alá be van építve 4 gomb (12, 3, 6 és 9 óránál). A számlapot négy csavar tartja, melyek elmozognak a NYÁK furataiban, így a számlap minimálisan mozgatható. A gombok beépített rugója kellően erős hozzá, hogy megfelelő ellenerőt biztosítson a számlap súlyának (így nem nyomódnak meg "maguktól"). Emellett egy ötödik gomb is beépítésre került a számlaptól jobbra, annak forgatási tengelyére merőleges irányban (az óra oldalán).

A gombok funkciói:

- Négy számlapi gomb (UBTN, RBTN, DBTN, LBTN)
 - Sleep üzemmódból való
 - Alarm mód esetén szundi aktiválása
 - Jobb- és baloldali gomb
 - * Navigálás a főmenük között
 - * Navigálás a beállítható adatok között (pl.: óra - perc - másodperc)
 - * Mentés, elvetés
 - Alsó és felső gomb
 - * Főmenük és almenüjeik közötti átjárás
 - * Értékek beállítása (fel -> +1, le -> -1)
- Oldalsó gomb (OFFBTN)
 - Ébresztés és időzítő kikapcsolása

5.6. Időbeli pontosság

A megfelelő időbeli pontosság eléréséhez az alábbi megoldásokat találtam:

- Minél kevesebb számítás elvégzése egy-egy loop alatt
- Ehhez dátum struktúrában TIME és DATE tagok - így elég futásonként ezt a két tagot összehasonlítani a pillanatnyi idővel -> harmadannyi for ciklus (ébresztések számának növelésével így hatványozott gyorsítás)

- RTC modul alkalmazásával pontos idő és egyszerű time interrupt-ok (pl.: másodpercenként frissíteni a képernyőt)
- Óra bekapcsolásakor pontos idő lekérése NTP szerverről
- Nyomógombok interrupt-két való használata (nem a loop egy adott pillanatában nézi, hogy meg van-e nyomva a gomb, hanem a fő programot megszakítva ”lekönyveli” /*interrupt függvény*/, hogy megnyomtuk, majd a loop egy adott részében megnézi, hogy az előző lefutás óta meg lett-e nyomva valamelyik gomb)

5.7. Üzem módok

Az okosórának három üzem módja van:

- Sleep mode
 - LCD kikapcsolva - energiafogyasztás csökkentése
 - Beállított ébresztések figyelése, összevetése a jelenlegi idővel (egyezés esetén *Alarm mode*)
- Active mode
 - LCD bekapcsolva
 - Interakció a programmal
 - Beállítások, vezérlés
- Alarm mode
 - LCD bekapcsolva (piros háttér)
 - Az éppen aktuális ébresztés nevének kiírása
 - Az ébresztés beállításától függően *csak villogás/rezgés/rezgés és hangjelzés*
 - Számlapi gombok - szundi
 - Oldalsó gomb - ébresztés kikapcsolása

5.8. Az aktív üzemmód menüszerkezete

5.8.1. Time display menu

- Quick watch almenü
 - *Sleep* módból bármelyik "számlapi gomb" megnyomásával ide kerülünk
 - 5 másodpercre kijelzi az időt
 - Utána visszatérés *sleep* módba
- Permanent time display almenü
 - A pontos időt mutatja, kikapcsolásig
 - A *quick watch* almenüből ide jutunk az alsó vagy a felső gomb megnyomásával
 - A felső vagy alsó gomb újbóli megnyomásával visszatérés *sleep* módba

5.8.2. Set alarm menü

- Home almenü
 - Kijelzi, hogy a *Set alarm* menüben vagyunk
 - Jobbra-balra tudunk a többi menü között váltani
- Select slot almenü
 - Itt lehet kiválasztani, hogy melyik helyre akarunk ébresztést beállítani
 - Jobb gomb - OK
 - Bal gomb - Cancel
- Set alarm almenü
 - Ébresztés idejének megadása
 - Ébresztési stílus megadása (néma, rezgő, hangos)
 - Ismétlődés megadása (naponta, hetente, havonta)

- Jobb-bal: váltás a fentiek között
- Fel-le: érték növelése/csökkentése
- Settings menü
 - Home almenü
 - Time format almenü
 - * Óra formátum kiválasztása
 - * Analóg/digitális
 - Set snooze time almenü
 - * Szundi idejének megadása
- Timer menü
 - Home almenü
 - Set timer almenü
 - * Időzítő megadása
 - Countback almenü
 - * Hátralévő idő kiírása
 - Time is up almenü
 - * Hangjelzés és rezgés, ha lejárt a beállított idő

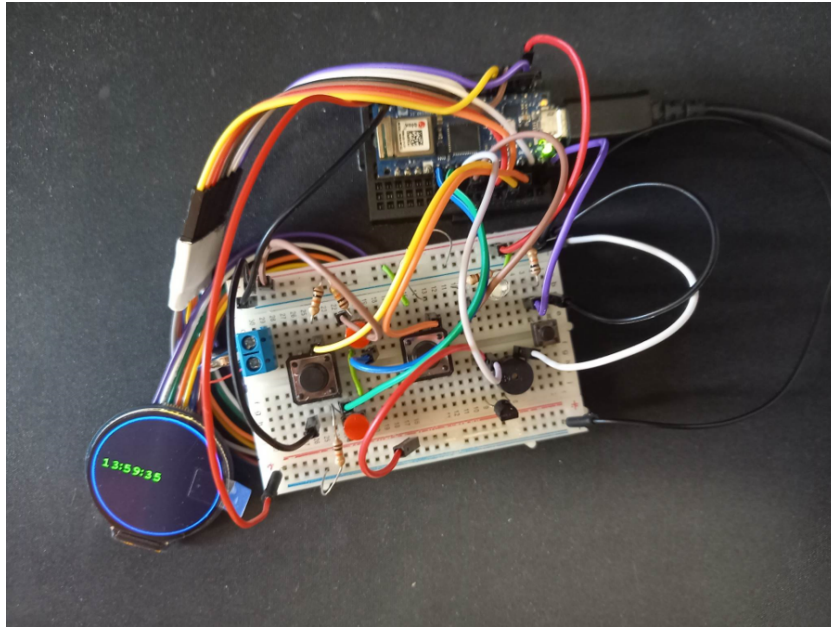
6. Továbbfejlesztési ötletek, lehetőségek

A projekt során megvalósított eszköz rengeteg módon továbbfejleszthető, kibővíthető. Ezen funkciók közül szeretnék kiemelni néhányat.

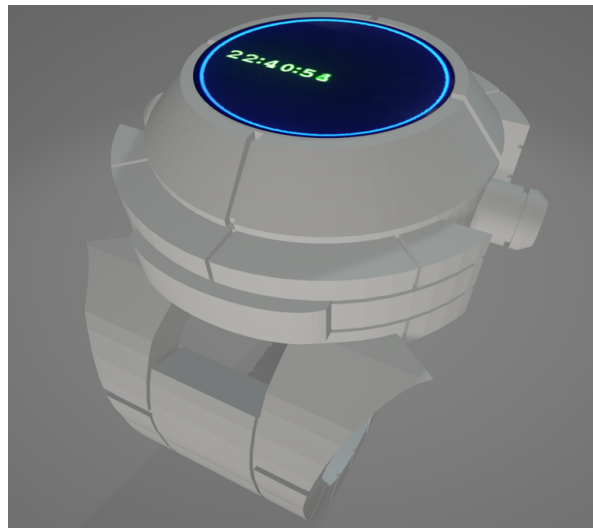
- Ébresztések kezelése menü létrehozása
 - Ébresztések ki- és bekapcsolása
 - Stílus, ismétlődés átállítása
- Esztétikai fejlesztések
 - Nagyobb betűkészletek létrehozása
 - Kijelzett String-eket, számokat automatikusan középre rendező függvény
 - Nem aktuális objektumok automatikus eltakarása, maszkolása
- Bluetooth-kommunikáció okostelefonnal
 - Mobil applikáció létrehozása
 - Ébresztések küldése, szerkesztése
 - Naptár szinkronizálása
 - Háttér/számlap beállítása
- Mikrofon és hangszóró beépítése
 - Kihangosító multimédia eszközként való használat
- Giroszkóp beépítése
 - Lépésszámlálás
 - Képernyő bekapcsolása, ha hirtelen vízszintes helyzetbe fordítjuk (megnézzük az időt gesztus)

7. Jelenlegi állapot

Az okosóra jelenleg nincs összeszerelve, csak bróbbapanelen összehuzalozva. A funkciók működését így is tudtam tesztelni, viszont sajnos nem maradt időm a tok és a szükséges NYÁKok megtervezésére.



7.1. ábra – Az okosóra jelenlegi állapota



7.2. ábra – A kész okosóra látványterve

Hivatkozások

- [1] <https://www.arduino.cc/en/software> (2023.06.09)
- [2] <https://docs.arduino.cc/arduino-cloud/> (2023.06.09)
- [3] <https://fritzing.org/> (2023.06.09)
- [4] <https://docs.arduino.cc/hardware/nano-33-iot> (2023.06.09)
- [5] <https://www.waveshare.com/1.28inch-lcd-module.htm> (2023.06.09)
- [6] https://www.hestore.hu/prod_10040903.html (2023.06.09)
- [7] https://www.hestore.hu/prod_10042054.html (2023.06.09)
- [8] https://www.hestore.hu/prod_10026555.html (2023.06.09)