BSc Önálló laboratórium téma

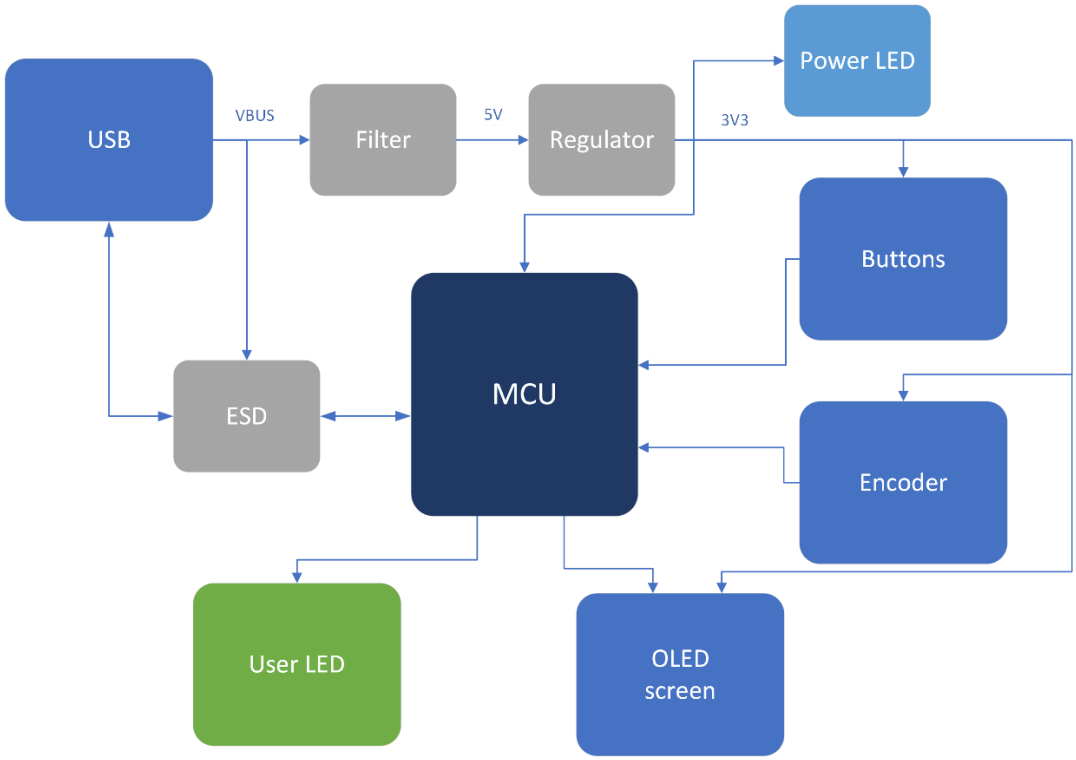
Jelszókezelő hardver fejlesztése

SÁGHY ATTILA
FZFZXF

Téma leírása:

Cél STM32 alapon kifejleszteni egy beágyazott szoftvert, ami USB-n keresztül HID eszköznek látszik, egy gombnyomásra képes jelszót billentyűleütésekként beírni. Egy kiegészítő szoftver küldi el a kívánt oldal nevét, amihez a jelszó tartozik, illetve az eszközzel együtt autentikálja a felhasználót (challenge-response / ECC). Az eszköz titkosítva (chacha20) tárolja a jelszavakat a flash memóriában, lehetőség van újat hozzáadni és korábbi módosítani.

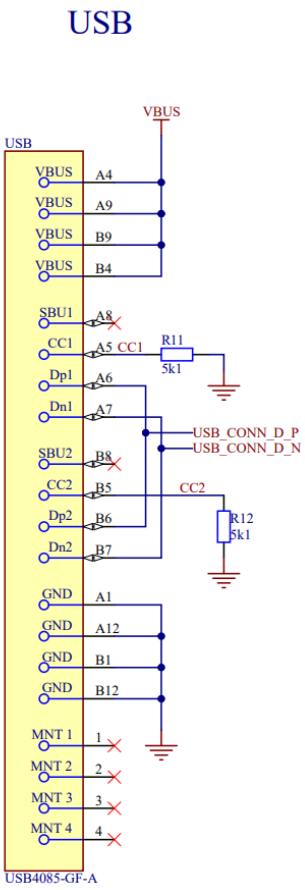
Kapcsolási rajz:



Kapcsolási rajz blokkdiagrammja

Röviden a hardverről:

A megvalósított jelszókezelő hardver egy STM32L422 mikrokontrollert használ, ami USB-n keresztül kap tápellátást. Mivel a mikrokontroller 3.3V-on működik és az USB-ből 5V jön, ezért azt elő kell állítani, amit egy LDO segítségével oldottam meg. Mielőtt a csatlakozóból érkező feszültséget rákötnénk a szabályozóra, előtte azt megszűrjük a zajoktól egy Pi szűrő segítségével. Az LDO kimenetéről kap minden a PCB-n található eszköz tápellátást, aminek a meglétét egy kék LED jelzi. Az USB adat ki- és bemeneteit sem közvetlenül kötjük a kontrollerre, hanem egy elektorsztatikus kisülés ellen védő IC-n keresztül. Ezeken túl a nyákon megtalálható még két nyomógomb, egy enkóder, még egy LED és egy OLED kijelző, amiken keresztül vezérelhetjük az eszközt.

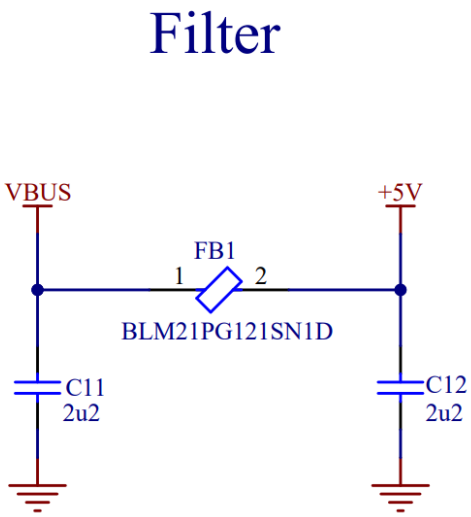


USB:

A választott csatlakozó egy USB-C 2.0. A szabvány szerint a CC lábakat 5.1kΩ-os ellenállásokon keresztül földre kell húzni, hogy tudassuk a PC-vel, hogy ő a host és ez egy device. Így a PC fogja szolgáltatni az áramot (max 1.5A). A Dp és Dn lábakat összekötjük, így biztosítható, hogy mindkét orientációban legyen adatátvitel. Az SBU pineket nem használjuk.

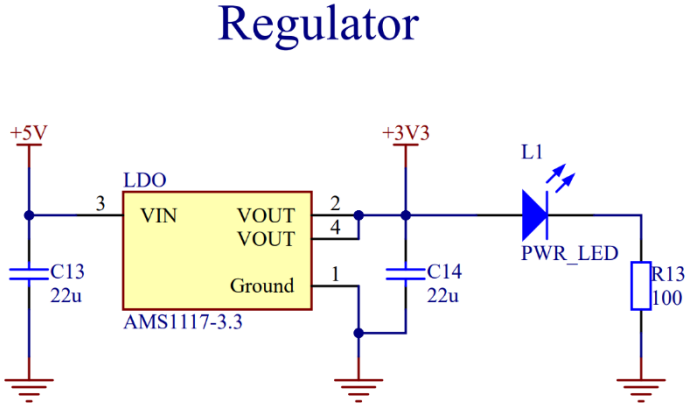
Szűrő:

A PC-ből érkező zajos feszültséget egy egyszerű aluláteresztő ferrit gyöngyös pi szűrővel szűrjük meg.



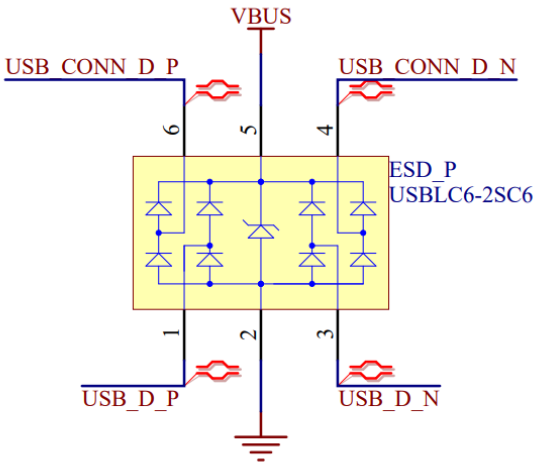
LDO:

Szintillesztést valósítja meg. 5V-ból 3.3V-ot állít elő, az adatlapnak megfelelő kondenzátorokat alkalmazva. A kimenetén megtalálható egy LED, ami a 3.3V meglétét jelzi. Minden alkatrész innen kap tápellátást.

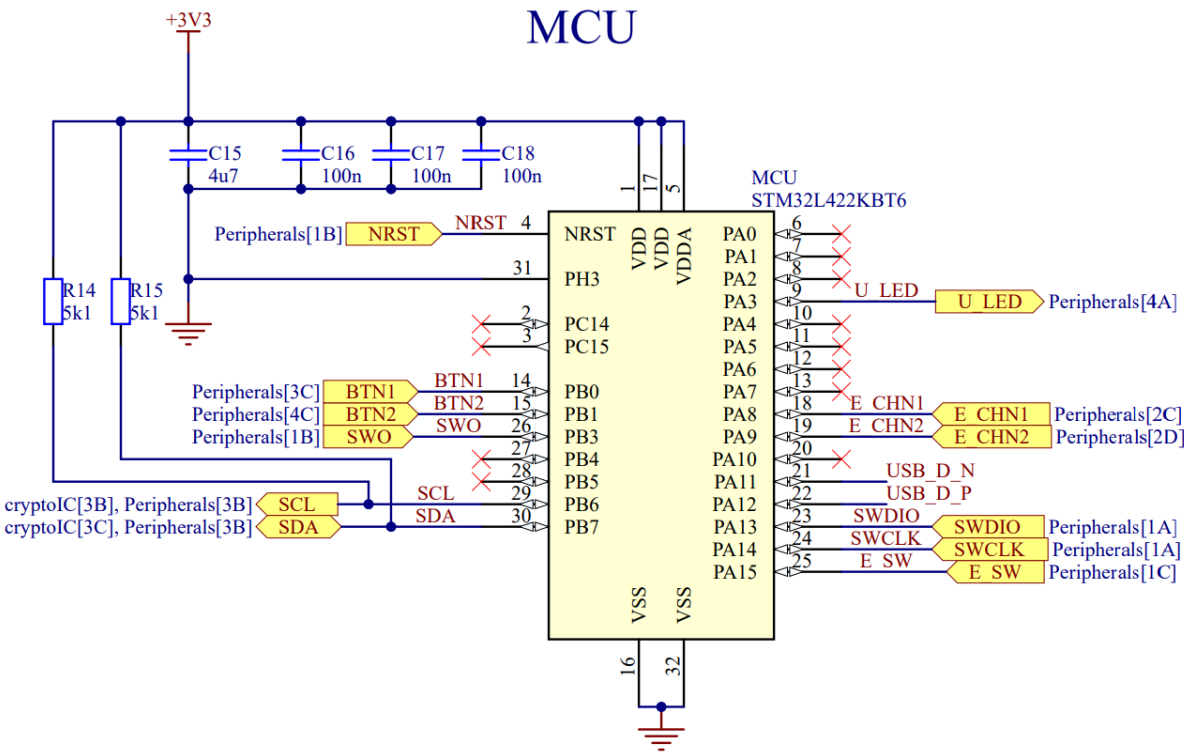


ESD protection:

Elektrosztatikus kisülés elleni védelmet valósítja meg, melyek kárt tehetnek a mikrokontrollerben. Az IC-ben TVS diódák találhatók, amelyek elfolytják a nagy feszültség tüskéket.



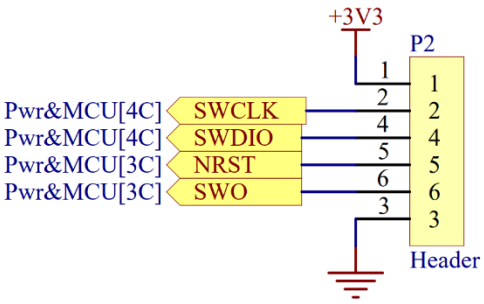
Mikrokontroller:

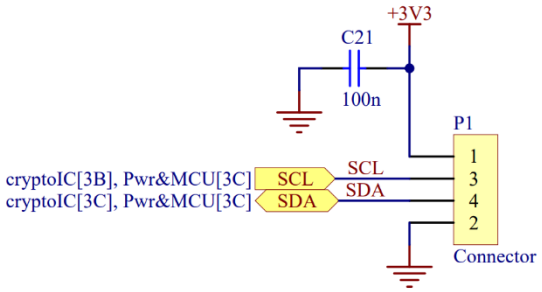


A mikrokontroller a hardver középpontja, minden periféria rá csatlakozik. Rajta futnak az algoritmusok és a jelszavakat is itt tároljuk. Minden tápfeszültség lábhoz tartozik egy 100nF-os hidegítő kondenzátor. Az esetleges feszültség esés ellen megtalálható egy 4.7µF-os buffer kondenzátor. Az I2C buszon 5.1kΩ-os felhúzó ellenállások vannak. Látható még hogy a kontroller melyik pinjéhez melyik periféria tartozik.

SWD:

Ezen keresztül történik a mikrokontroller programozása és debuggolása. A nyákon egy tüskesor formályában jelenik meg.



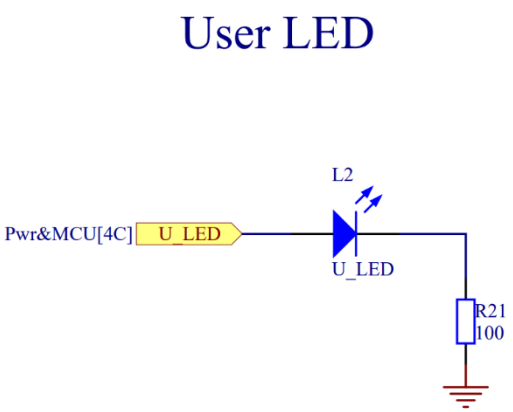


OLED kijelző:

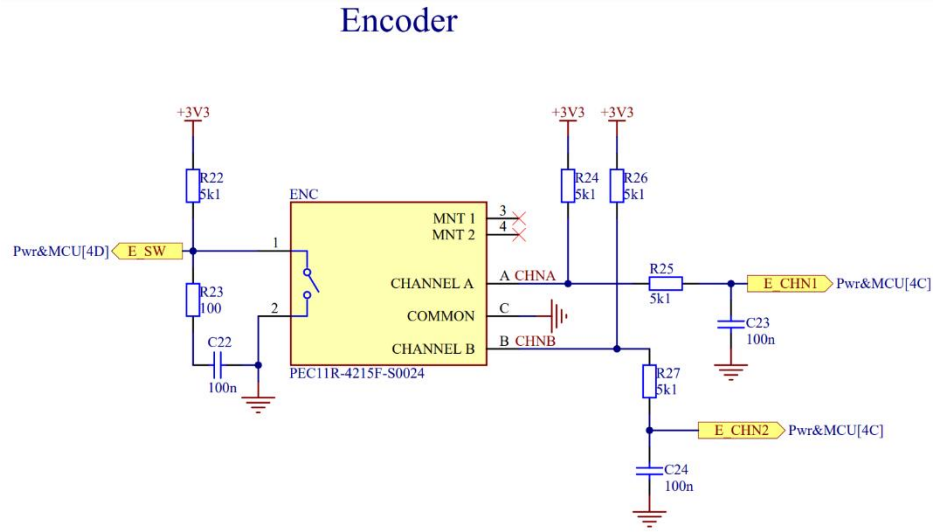
128x64 pixel felbontású 0.96in méretű I2C kijelző. SSD1306 drivert használ. A táp bemeneten megtalálható egy 100nF-os hidegítő kondenzátor.

User LED:

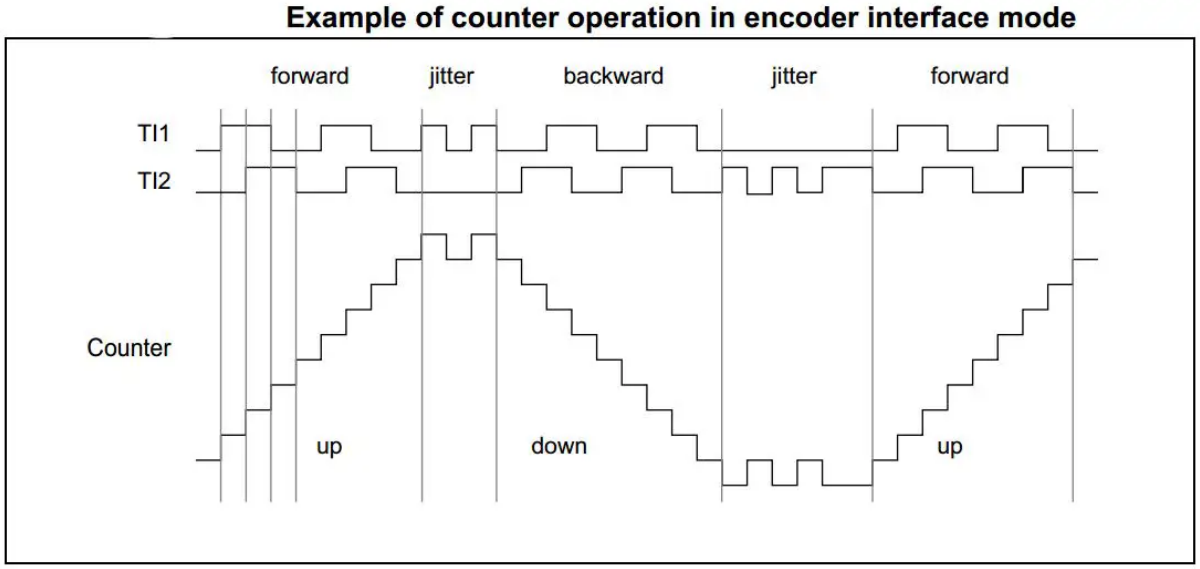
Egyszerű 0805-ös zöld LED, 100Ω-os ellenállással sorba kötve. Közvetlenül a mikrokontroller vezérli. Státusz visszajelzésre használt. Utólag nagyon fényesnek bizonyult, ezért PWM-el vezérelt.



Enkóder:



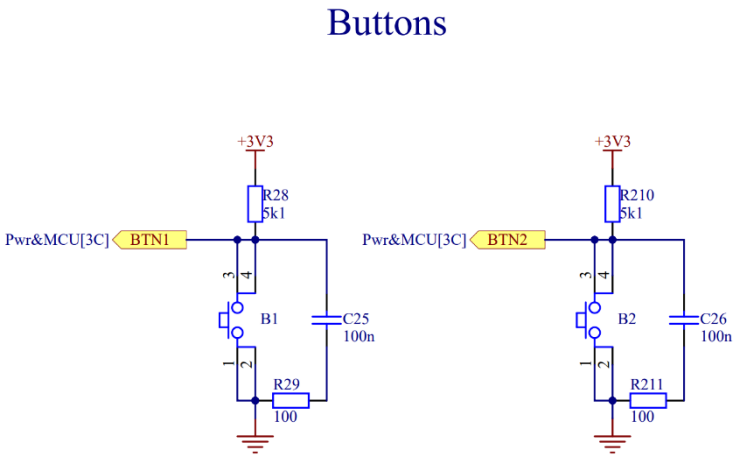
Menürendszerben való navigálásra és a PIN kód bevitelére használt beviteli eszköz. Áll egy nyomógombból és két csatornából. A két csatorna a mikrokotrolleren egy timer-re van kötve, ami enkóder üzemmódban van konfigurálva. Ez az jelenti, hogy ha az enkódert pozitív irányba forgatjuk akkor kattanásonként a timer számlálója 4-gyel nő, ha negatívba, akkor pedig 4-gyel csökken. Az enkóder tökéletlenségeiből fakadóan kattanásonként nem mindig 4-gyel osztható számot kapunk (jitter), ezért nem azt vizsgáljuk, hanem a két kattanás közti differenciát. A csatornákon és a gomboknál megtalálható felhúzó ellenállások, valamint pergés mentesítés céljából RC aluláteresztő szűrők.



Gombok:

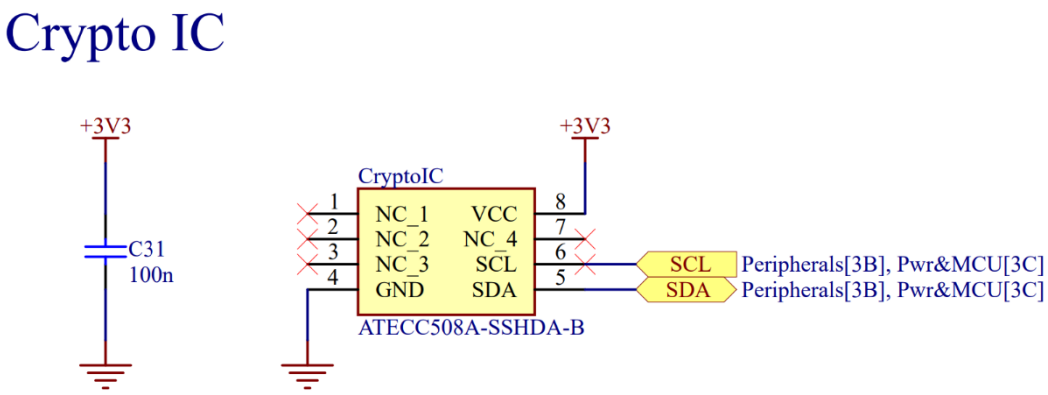
Egyszerű nyomógombok felhúzó ellenállásokkal és RC aluláteresztő szűrőkkel.

BTN1 a menüben való visszalépéshez használt, míg BTN2 vel menüt lehet váltani, valamint jelszót kiküldeni.

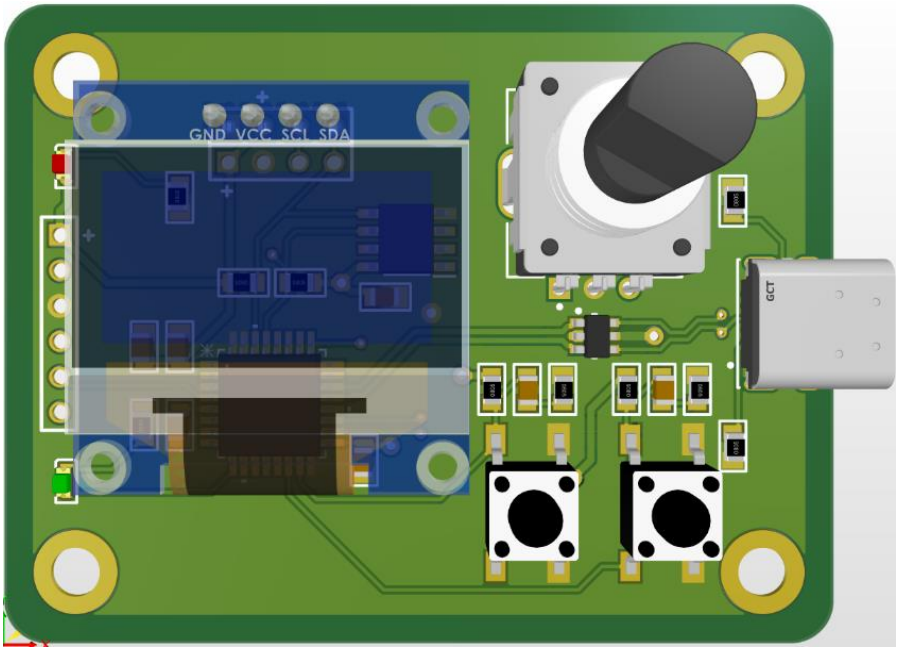


Crypto Authentication Device:

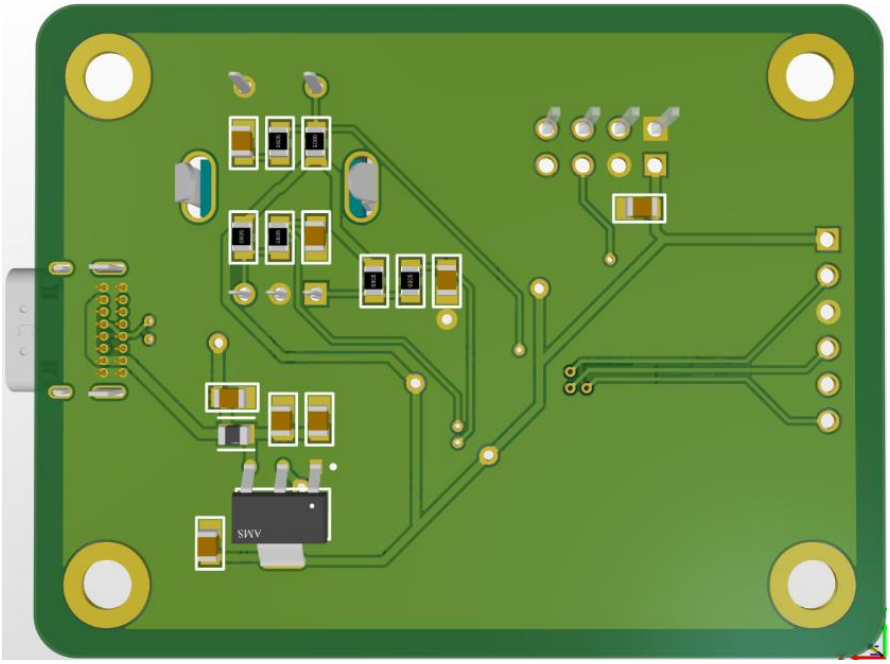
I2C-s kriptográfia eszköz. Valódi random szám generálására és különböző titkosító algoritmusokra képes. Végül nem került bele a végleges hardverbe.



PCB design:



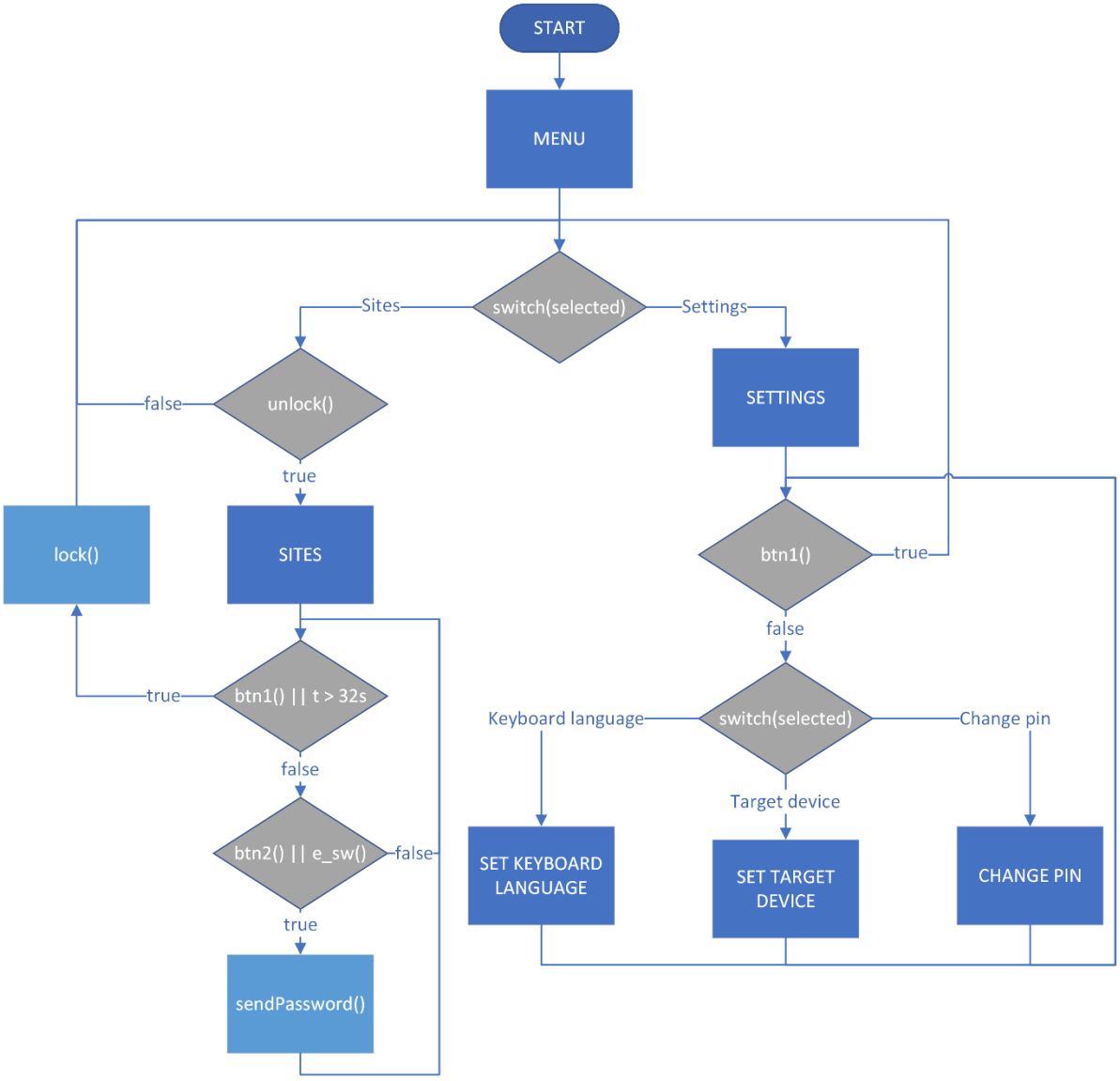
PCB design front



1. ábraPCB design back

A megvalósított áramkörön két hüvelysor is helyett kapott az OLED kijelzőnek, mert a VCC és GND lábak sorrendje gyártónként eltér. Egy + szimbólum jelzi a VCC pint.

Firmware:



Firmware folyamatábrája

Főmenü

A program indulásakor rögtön a főmenüben találjuk magunkat. Itt kiválaszthatjuk, hogy a weboldalak listáját szeretnénk-e látni, vagy a beállításokat. Ha az előbbit választjuk, akkor fel kell oldanunk az eszközt. Ezt egy PIN kód segítségével tehetjük meg, amit az enkóderrel tudunk bevinni. Ha az enkódert az óramutató járásával megegyező irányba tekerjük, akkor egyre nagyobb számokat kapunk, ha pedig az óramutató járásával ellentétes irányba, akkor pedig csökkenő számokat. Ha az adott helyiértéken megfelelő számot állítottunk be, akkor az enkóder lenyomásával válthatunk a következőre. Ha az utolsó helyiértéken váltunk, akkor a kiválasztott digit újra az első lesz így, ha esetleg elrontottuk valamelyiket, akkor ki tudjuk azt javítani. Magák a bevitt számok is le vannak kezelve túl- illetve alulcsordulás ellen hisz, ha 9-nél nagyobb számot szeretnénk bevinni, akkor 0 jelenik meg és ha 0-nál kisebbet, akkor pedig

9-es. Csak az épp kiválasztott digit értékét látjuk, a többi helyiértéken csillag van, hogy egy másik személy ne tudja lelesni azt. Ha bevittük a PIN kódunkat, akkor a 2-es gomb megnyomásával ellenőrizhetjük, hogy érvényes-e. Ha nem, akkor törlődik a bevitt PIN kód és a kiválasztott digit újra a legelső lesz.

PIN kód bevitel

Weboldalak listája és zöld LED

Ha érvényes volt a bevitt PIN kódunk, akkor sikeresen feloldottuk az eszközt, felvillan a zöld LED, és megjelenik előttünk a weboldalak listája, melyekhez tartoznak az elmentett jelszavak. Az enkóder segítségével kiválaszthatjuk, hogy melyik weboldalra szeretnénk belépni, majd annak, vagy a kettes gombnak lenyomása után az eszköz, mint egy billentyűzet, begépeli az adott jelszavunkat. Minderre van 32 másodpercünk, mert ez után a jelszókezelő biztonsági okokból automatikusan zárol és visszalép a főmenübe. Természetesen előbb is lehet zárolni, valamint visszalépni az egyes gomb segítségével.

A beállítások menüben három lehetőség közül választhatunk. Be lehet állítani a billentyűzet nyelvét, a cél eszközt, valamint PIN kódot lehet változtatni. Mivel az eszköz konkrétan úgy működik, mint egy billentyűzet fontos, hogy ugyan az a nyelv legyen rajta beállítva, mint a céleszközön, mert speciális karakterek esetén biztosan mást gépel be. Ezen túl fontos, hogy maga a céleszköz is jól legyen beállítva, mert tesztelések alapján pl. Android-on magyar billentyűzetet beállítva a speciális karaktereknek nem ugyan az a billentyű kombinációja, mint Windows-on. PIN kódot változtatni feloldás után lehet, 2x beírva az újat. A különböző almenükből itt is az 1-es gombbal lehet visszalépni.

Beállítások

A programon végigmenve 2 féle megjenítéssel találkozhatunk. Egy adott menü, valamint a PIN kód bevitelével. A menükhöz tartozót egy page struktúra kezeli.

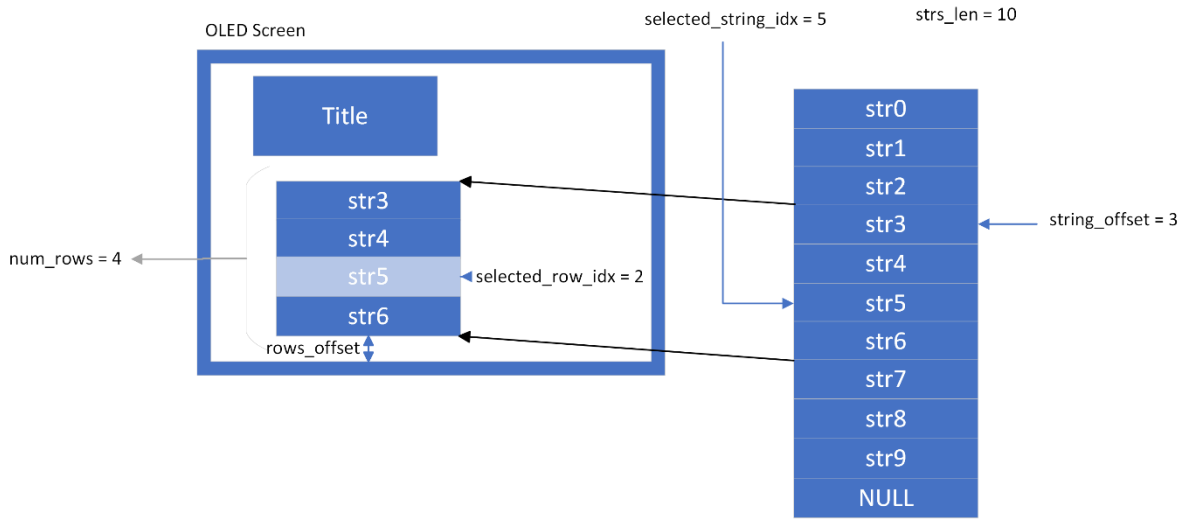
Page:

Egy page egy címből és egy NULL terminált string tömbből áll.

```
// set by page functions
typedef struct {
    uint8_t string_offset;       // index of the first string that is being drawn
    int16_t selected_row_idx;    // highlighted row
    uint8_t selected_string_idx; // selected_row + string_offset
    uint8_t num_rows;            // number of strings that fits the screen
    uint8_t strs_len;            // number of strings in the string array
    uint8_t rows_offset;         // adjusts the spacing between the title and rows
    FontDef title_font;
    FontDef rows_font;
    char *title;
    char **rows;
} Page;

Page initPage(FontDef *title_font, char *title_str, FontDef *rows_font, char *rows_str[]);
void drawPage(Page *p);
```

Használata a kövezkező: Létre kell hozni egy Page struktúra példányt az initPage() segítségével. Ennek a függvénynek 4 paramétere van. A címhez tartozó betűtípus és string, valamint a sorokhoz tartozó betűtípus és string tömb. Az initPage() kiszámolja a megjelenítéshez szükséges adatokat, többek közt hogy hány sor fér el és a sorok között mekkora távolságok legyenek. Ha nem szeretnénk címet megadni, akkor NULL pointert kell helyette átadni, és következésképpen az initPage() több helyet fog adni a soroknak. Ezek után a drawPage() függvény segítségével a Page struktúrát átadva kirajzolható az adott oldal.

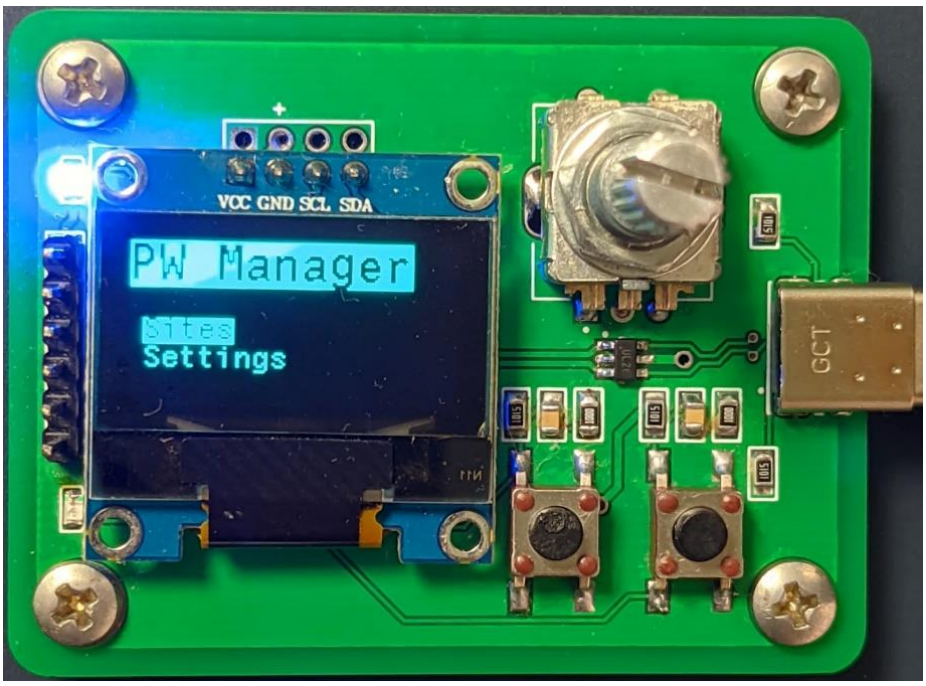


Az ábrán látható egy példa paraméterekkel együtt. A drawPage() a string tömbből egyszerre annyi sort jelenít meg amennyit az initPage()-ben kiszámoltunk, jelenleg ez 4. A kiválasztott sort az enkóderrel módosíthatjuk. Ha az óramutató járásával megegyező irányba tekerjük, akkor felfelé lépünk, ha ellentétes irányba, akkor pedig lefelé. Ha tekerés során a következő sor kilóg a képernyőről, akkor a 4 stringet tartalmazó ablak a megfelelő irányba tolódik. Például, ha str6 után még lefelé megyünk, akkor a megjelenített stringek str4-str7 lesznek. Ha a string tömb végére értünk, akkor újabb lefelé lépésnél az ablak a string tömb elejére ugrik és a kiválasztott sor a legelső lesz. Értelemszerűen, ha felfelé megyünk, akkor is érvényesek az ablak eltolások a megfelelő irányba. A kiválasztott string indexe a selected_string_idx-ben van tárolva, amit pl. a következő menü kiválasztásánál egy switch()-ben használhatunk.

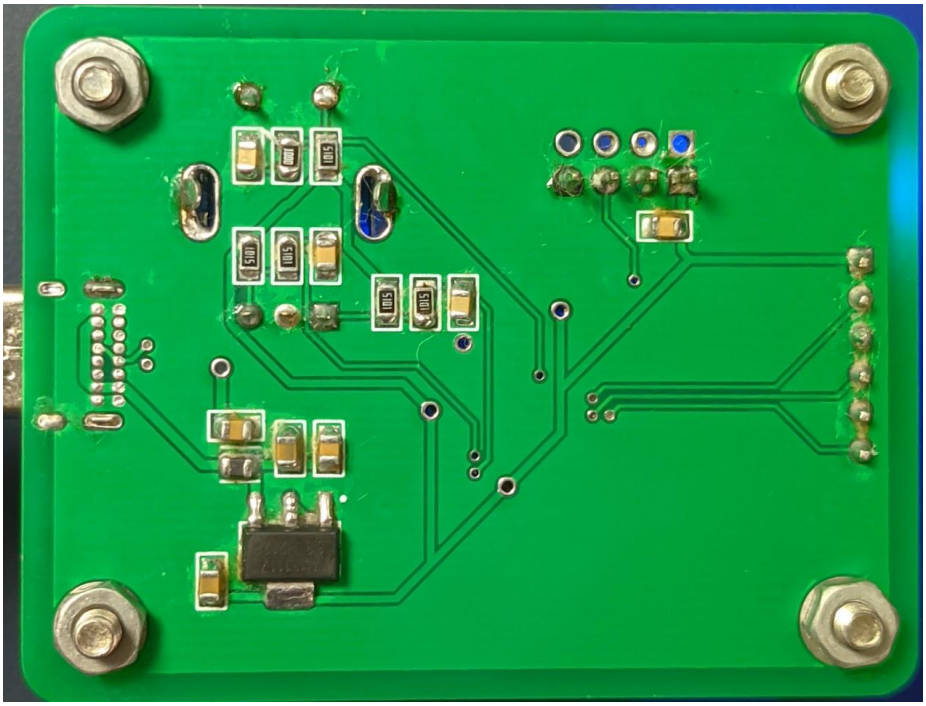
PIN

A PIN kód bevitele és megjelenítése a setDigits(), illetve a drawDigits() fügvényekkel történik. A PIN mérete egy define segítségével állítható. A drawDigits() a megadott méretnek megfelelően egyenlő távolságokra középen rajzolja ki a számokat.

Elkészült hardware:



Hardware front



Hardware back

Projekt github linkje:

https://github.com/saghya/passwordManager