Hardware Password Manager

Autor

Victor Barbu

Introducere

Exista numeroase solutii de password manager, dar majoritatea implica stocarea parolelor undeva online, intr-un cloud. Parolele rar sunt “cu adevarat” criptate deci, in cazul unui leak de date, ele pot fi expuse publicului. Proiectul are ca scop realizarea unui Hardware Password Manager care sa ofere o interfata sigura si simpla. El se va conecta la calculator si, de acolo, utilizatorul va interactiona doar cu programul care stie sa comunice cu acest device.

Descriere generală

Hardware Design

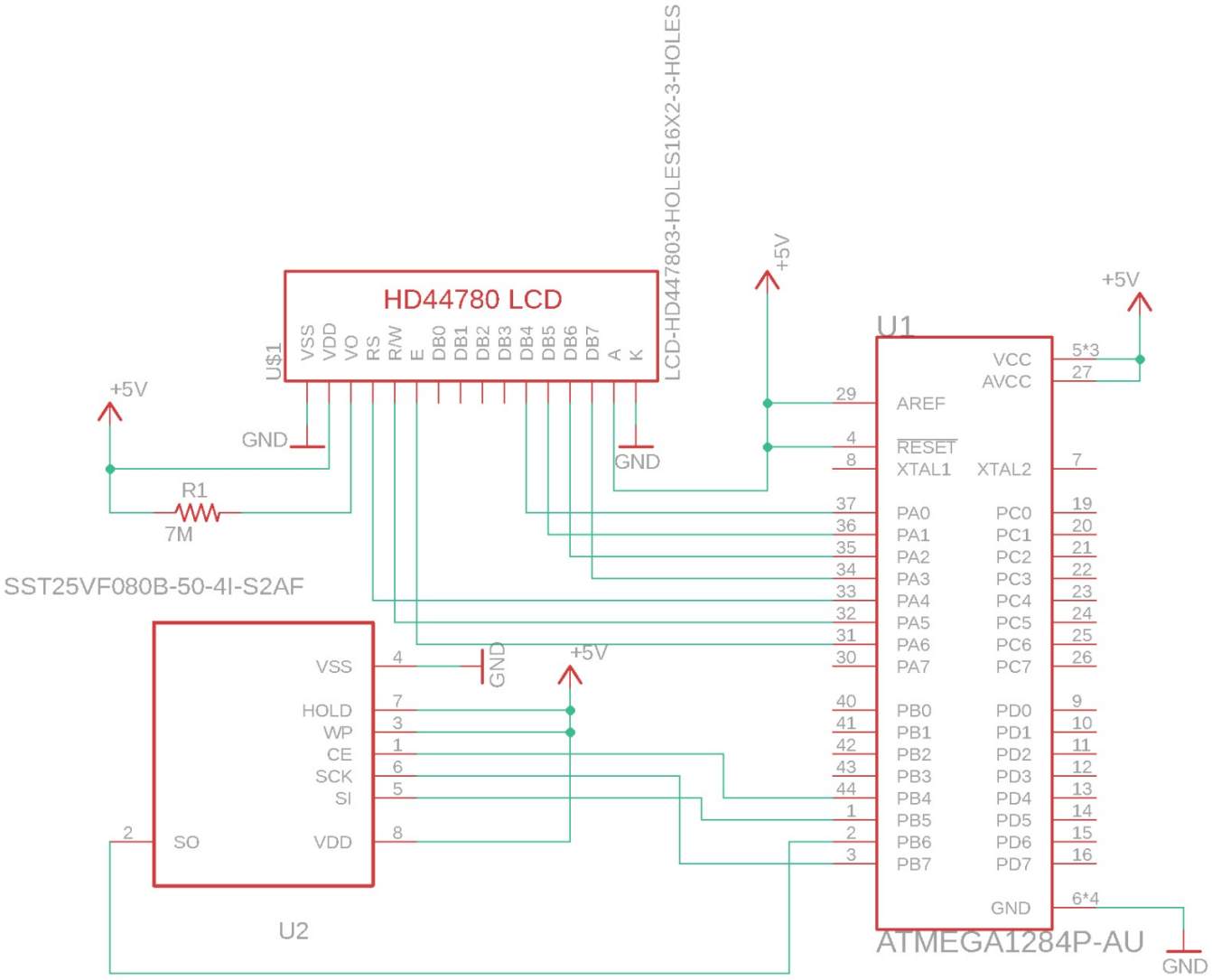
Urmatoarele componente vor fi necesare:

- ATMega1284P
- ATECC508A-SSHCZ-B (criptoprocesor)
- SST25VF080B-50-4C-PAE (memorie flash 8M)
- interfata USB-serial
- LEDuri + rezistoare
- condensator de ordinul uF pentru stabilizarea tensiunii de la sursa
- cristal quartz 20MHz + 2 condensatori de 22pF

NOTA Lista este deocamdata orientativa.

Pentru debugging pe partea de hardware, voi folosi un logic analyzer.

Urmatoarea este schema electrica:



Software Design

Codul pentru microcontroller este scris in C, compilat cu `avr-gcc` si scris cu `avrdude` si programatorul ATMEL-ICE.

Codul pentru aplicatia de linux va fi scris intr-un limbaj high-level, de genul Python, pentru ca operatia lui nu este dificila (implica doar calluri de `read` si `write` pe serial device).

Partea software contine urmatoarele module:

1. SPI (`spi.h`) faciliteaza transferul de date prin SPI catre memoria flash.
2. Flash (`flash.h`) abstractizeaza accesul la memoria flash folosindu-se de modulul SPI si de instructiunile suportate de memorie.
3. MemHdr (`mem_hdr.h`) abstractizeaza structura de date prin care se stocheaza indexul de parole in memoria flash intr-o maniera care salveaza spatiu de stocare.
4. IoUtil (`io_util.h`) si LCD (`lcd.h`) ajuta la folosirea LCD-ului HD44780 intr-o maniera transparenta, folosind `printf`.
5. DB (`db.h`) este inca un layer de abstractizare peste SPI, Flash si MemHdr si expune functionalitati high-level de initializare, adaugare, stergere si citire din indexul de parole.

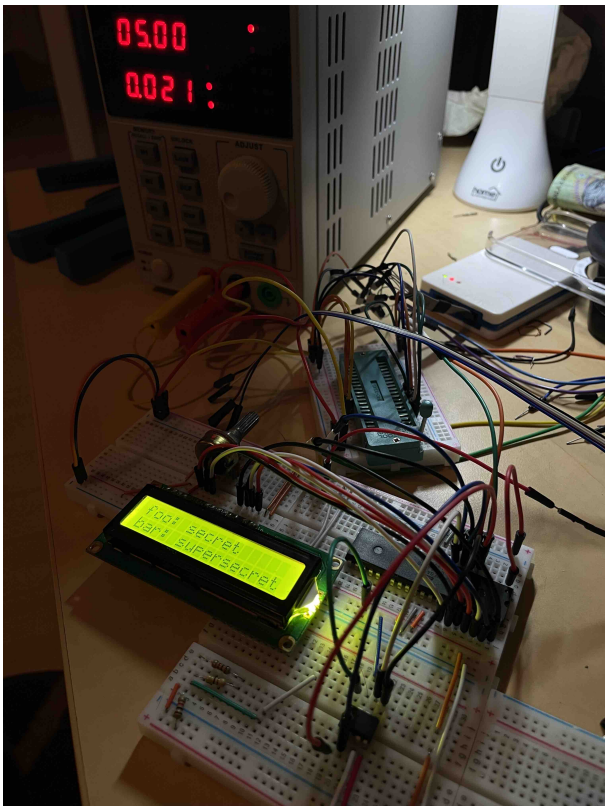
Rezultate Obţinute

Functionalitatea de comunicare prin USB cu un computer si cea de criptare a datelor vor fi adaugate intr-o versiune ulterioara. In acest moment, partea hardware este pregatita pentru a scrie, citi, sterge si lista parole din memoria flash.

Device-ul suporta chiar si hot-swap al memoriei flash. Cand aceasta se schimba, ea va fi initializata fara ca macar sa fie nevoie de restartarea microcontrollerului.

Pentru a prezenta un demo al produsului, microcontrollerul a fost programat sa scrie doua parole, iar, in alta iteratie, sa le citeasca din memorie si sa le afiseze pe display.

Mai jos se poate vedea ruland iteratia in care parolele sunt citite cu succes din memorie. Este au labelurile “foo” respectiv “bar”, iar parola insasi apare dupa ”:”.



Concluzii

Device-ul isi poate atinge toate functionalitatile prezentate in introducerea acestui document. Implementarea a durat mai mult decat era anticipat, deci criptarea si comunicarea prin USB cu USART vor veni intr-o versiune ulterioara a proiectului.

Ca directii viitoare, imi doresc sa comand componente SOIC si sa pun dispozitivul pe un PCB si intr-o carcasa astfel incat el sa poata fi folosit cu adevarat. Cu mici imbunatatiri generale atat la nivel hardware cat si software, acest lucru este posibil.

Consider ca proiectul a fost util pentru ca a dat start a ceva ce chiar poate deveni un device util.

Download

- Cod sursa: hw_pass_mgr.zip
- PDF: https://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/hw_pass_mgr?do=export_pdf

Jurnal

- 20 aprilie - am ales tema proiectului
- 26 aprilie - creat pagina de wiki
- 10 mai - 1 iunie - lucru la montaj si software
- 2 iunie - finalizarea acestui wiki

Bibliografie/Resurse

- Datasheet ATmega1284P
- Datasheet SST25VF080B
- Datasheet HD44780