Rapport - Digitala projekt 5p

Department of Information Technology
LTH

Munin

Grupp 4

Igor Tasevski
Henrik Palmér

Handledare: Bertil Lindvall

Abstract

In this project we have designed a construction and we have named it Munin. The construction can be used by anybody and would certainly help a lot of people. Munin stores all forms of passwords that the user wants to save. These passwords can, for an example be, to your bank account, cell phone or maybe to your e-mail. Munin uses an AVR AT Mega 16 processor, a keypad and a display. The hardware was first put together and thereafter the software was added. During the process of making this construction,some problems appeared, but in the end we managed to complete Munin.

Innehållsförteckning

1 Inledning

Målsättningen med vårt projekt var att bygga en konstruktion som lagrar användarnamn och lösenord. Konstruktionen, som döptes till Munin, fungerar som en komihåg dosa. Uppgiften bestod i att först planera arbetet, bygga Munin och därefter skapa mjukvaran. Till en början valdes processor, samt knappsats och display. Därefter planerades hur komponenterna ska samverka och hur de ska kopplas samman.

Då komponenterna monterats och kommunikationen mellan dem fungerade återstod programmeringen. I denna rapport sammanfattas tillvägagångssättet för skapandet av Munin. Det kommer även lite mer ingående att presenteras hur mjukvaran är utformad följt utav en diskussion samt resultat.

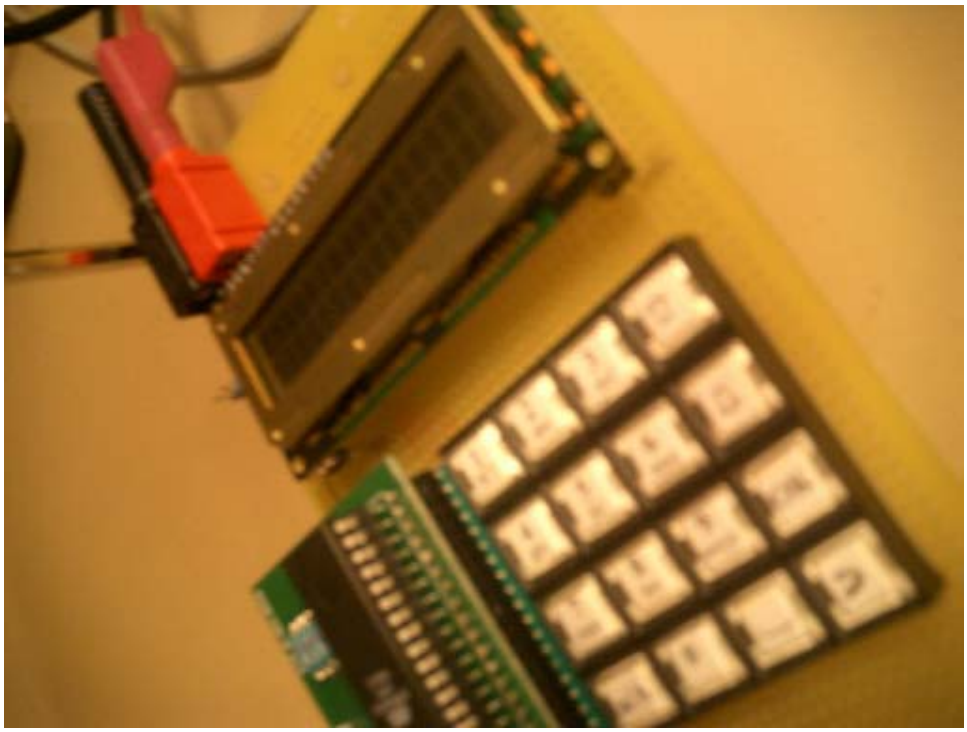

Bild 1: Munin

2 Metod

Det första som gjordes i projektet var att bestämma vad för applikation som ska konstrueras och vilka funktioner denna ska innehålla. Detta gjordes relativt snabbt och därefter påbörjades arbetet lite mer ingående. Vi inledde med att bestämma oss för vilka komponenter vi ska använda och kom relativt snabbt fram till att det vi behövde i form av hårdvara var följande:

- Processor - AVR AT Mega 16 [1]
- Knappsats - 16 stycken knappar
- Alfanumrisk teckendisplay - SHARP Dot-Matrix LCD Units [2]
- Jord till displayen

Eftersom vår applikation inte består av många olika komponenter gick även denna del relativt snabbt. Därefter fortsatte arbetet med att studera databladen för respektive komponent och göra upp en plan för hur dessa ska kopplas samman. Ett kopplingsschema ritades upp och efter ytterligare diskussioner var det dags till att börja bygga.

Komponenterna monterades på en kopplingsplatta och sammankopplades med hjälp utav virning och lödning. Vi stötte på en del problem, men efter konsultation med vår handledare kom vi snabbt fram till en färdig konstruktion av Munin.

2.1 Huvudmenyn

Huvudmenyn navigeras som andra menyer med den skillnaden att Munin stängs av när användaren går ur den. Alternativen är ”View Passwords”, ”Edit Passwords” och “Munin”. I menyn ”View Passwords” visas en lista med de sparade lösenordens identifierare. När användaren väljer ett menyalternativ skrivs dettas identifierare ut följt av lösenordet på nästa rad. Användaren kommer tillbaka genom att trycka på ”Clear”.

I menyn ”Edit Passwords” finns tre stycken undermenyer: ”Add Password”, ”Delete Password” och ”Edit Password”. De olika undermenyerna låter användaren utföra de operationer som namnen antyder.

Den sista menyn ”Munin” innehåller tre stycken undermenyer: ”Change Password”, ”Reset” och ”About”. Om användaren väljer ”Reset” skrivs Munins minne över med nollor och den stängs av.

2.2 Navigering

Menyer i Munin navigeras med hjälp av ”↑”- och ”↓”-knapparna på knappsatsen. För att välja det markerade alternativet används ”OK”-knappen. Om användaren trycker på ”Clear” avslutas nuvarande meny och Munin går en nivå upp i menyhierarkin.

3 Mjukvara

3.1 Programkoden

Styrsystemet till Munin är skrivet i ANSI-C med hjälp av AVR Studio, där koden är uppdelad i olika moduler. Underst ligger moduler för grundläggande kontroll av maskinvaran: mkeypad för att läsa knapptryckningar, mdisplay för att göra utskrifter på den alfanumeriska displayen och mmem för att göra skrivningar till EEPROM.

Denna används i sin tur av en modul för grundläggande användargränssnitts operationer: mui för att visa meddelanden, läsa in textidentifierare och lösenord samt navigera i menyer. Dessa menyer ligger i separata filer.

Det finns även en krypteringsmodul. Denna består av des, som sköter DES-kryptering, lfsr som implementerar ett linjärt skiftat återkopplingsregister, samt mcrypto som kombinerar dessa och implementerar funktioner för att läsa och skriva till EEPROMoc arbetsminne samt kontrollera att det som dekrypteras verkligen är korrekt.

Utöver dessa moduler finns mcommon som består av generella funktioner och makron, som fördröjningar och bitläsning.

3.2 Huvudprogrammet

Huvudprogrammet ligger i Munin.c, som bara direkt använder användargränssnittsmodulen och krypteringsmodulen för att vid uppstart av Munin läsa in ett lösenord från användaren och kontrollera detta genom att försöka dekryptera data lagrad på EEPROM.

Om lösenordet är korrekt kommer exekveringen att fortsätta med att huvudmenyn öppnas, annars visas ett felmeddelande och lösenordet efterfrågas igen.

Om EEPROM är oinitierat, kommer programmet istället att fråga efter ett nytt lösenord och sedan gå vidare som om ett korrekt lösenord matats in.

3.3 Maskinvaruhantering

Denna modul är tillsammans med mcommon de enda som är plattformsberoende. Mkeypad skriver till och läser från AVR:ens port B för att kommunicera med knappsatsen, och mdisplay använder port A och D för att skicka kommandon och data till displayen.

Utvecklandet av mjukvara för att styra displayen är det som tagit avgjort mest tid. När vi ersatt vår första AVR med en som fungerar tog det dock inte lång tid att få den att göra som den ska.

3.4 Användargränssnitt

Då Munins konstruktion är så enkel med bara en två-raders alfanumerisk display och en knappsats med endast 16 knappar är koden för användargränssnittet relativt enkel. Den funktion som är mest komplicerad är inläsning av text från användaren, vilket vi löst genom att emulera textinmatning på en telefon.

3.5 Kryptering

Då Munin är tänkt att lagra många lösenord är det mycket viktigt att informationen lagras säkert. På denna punkt har vi tyvärr tvingats göra en kompromiss på grund av tidsbrist.

Vi använder DES för kryptering, vilket inte kan anses vara säkert nog nuförtiden eftersom nyckelstorleken endast är 56 bitar. Vi finner dock kompromissen acceptabel, eftersom vi ändå genererar nyckeln av ett lösenord som inte behöver vara längre än 8 tecken, eller 56 bitar.

Innan data krypteras och lagras på EEPROM, XOR-maskas den med utdatan från ett linjärt skiftat återkopplingsregister för att undvika de långa sekvenser med nollor som uppkommer när identifierare och lösenord är kortare än 16 tecken. Återkopplingsregistrets startläge härleds ur lösenordet.

4 Resultat och diskussion

Sammanfattningsvis kan det sägas att konstruktionen blev sådan som vi förväntat och hoppats på att den skulle bli. Efter att vi hade bestämt oss får vilka komponenter som ska användas och hur vi ska koppla samman dem gick det, som tidigare nämnts, snabbt att bygga ihop den. Trots att vi från början kopplade till pinnar som upptogs av vår JTAG och att vi sedan fick koppla om allting gick det hyfsat smärtfritt till. De stora och tidskrävande problemen kom då vi skulle programmera mjukvara som skulle sköta kommunikation mellan processor och display.

Efter många timmar av felsökande kom vi till sist fram till att vi var tvungna att byta processor. Detta gjordes och därefter fungerade kommunikation mellan våra komponenter utmärkt.

Resultatet blev en färdig konstruktion som uppfyllde de krav vi ställt i början av projektet. Vi vidhåller att krypteringen vi använder oss av kunde gjorts bättre, men med tanke på de veckor vi hade på oss, så har vi gjort Munin tillräckligt säker.

Projektet har varit mycket lärorikt och inspirerande. Det har ökat vår förmåga att angripa nya problem samt ökat vår kunskap. Det speciella med projektet är att man skapat en konstruktion från grunden och att det blev så som man tänkt sig.

5 Referenser

[1] http://www.it.lth.se/datablad/Processors/ATmega16.pdf

[2] http://www.it.lth.se/datablad/display/LCD.pdf