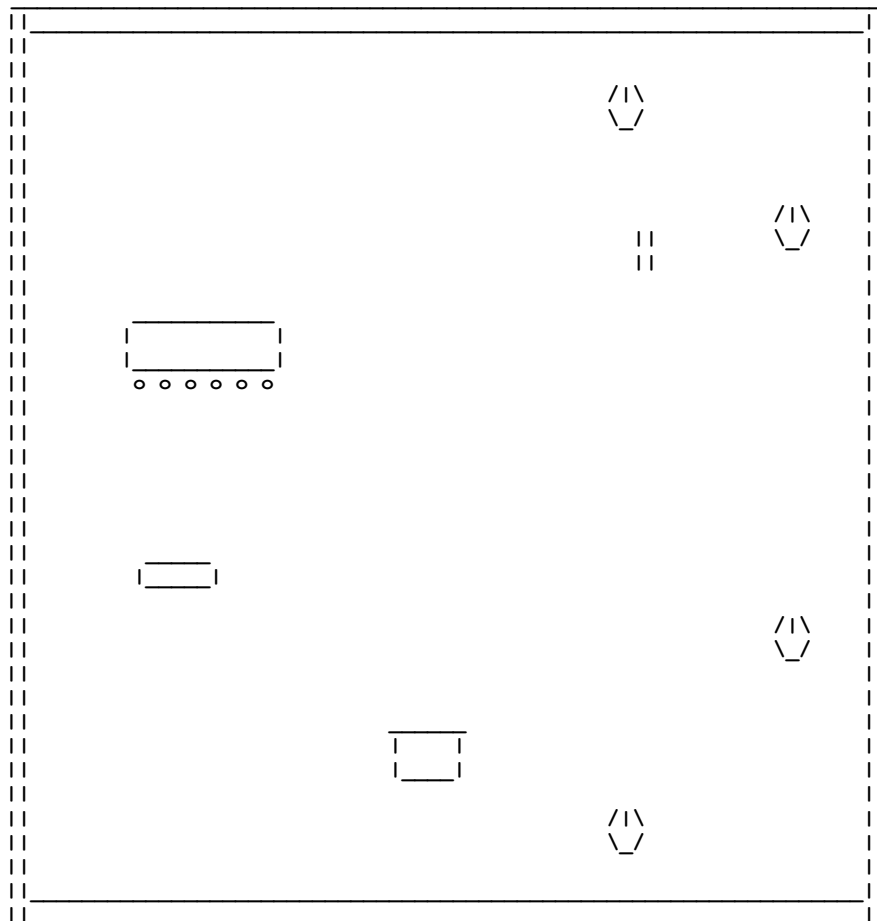


Digitala projekt 8p

Växelmojten

||
M A N T V / T E K
| | \ / | | | |



Abstract

This report describes a project in the course Digital projects EDI022. The project described is a remaking of an existing exchange machine called Våxelmojten. The remaking consists of a total rebuild, although this report only describes the user interface and the software. The hardware used in the course and described in this report consists of two Atmel AVR mega16 processors communicating using the built in I2C or TWI interface. The program was written in C and AVR studio 4 was used for the development. The reports describes the process of the project, the outcome and the problems encountered.

Innehållsförteckning

1 Inledning.....	4
2 En Växelmojt.....	5
2.1 En Växelmojts verksamhetsområde.....	5
3 Hårdvara och funktion.....	6
3.1 Processorerna.....	6
3.2 Display, knappar och acceptorer.....	6
3.3 Motorer och sensorer.....	6
3.4 PC.....	7
4 Mjukvara.....	8
4.1 Mainloopar.....	8
4.2 Kommunikation.....	8
5 Användargränssnittet.....	9
5.1 Knappar och display	9
6 Mål och resultat.....	10
7 Problem.....	12
8 Kommentarer.....	13
9 Sammanfattning.....	14

1 Inledning

Den här konstruktionsuppgiften är gjord som ett projekt i kursen Digitala projekt, större kurs. Kursen går ut på att självständigt konstruera och programmera en fungerande elektronisk prototyp baserad på en microprocessor. Projektet som beskrivs här är ett försök att skapa styrning och användargränssnitt till en växelautomat. Själva maskinvaran har på det här stadiet simulerats. Konstruktionen baseras på två microprocessorer av typen Atmel AVR Mega16. Koden är skriven i C. Prototypen är avsedd att resultera i en färdig applikation för användning i en befintlig växelautomat, kallad Växelmojten, som finns i drift i E-sektionens lokaler på LTH i Lund.

2 En Växelmöjt

Växelmöjten, som automaten i fråga alltså kallas för i teknologmun, är från utsidan en enkel apparat i vilken 20-kronorssedlar kan växlas till mynt och mynt växlas till andra mynt. På senare tid har Växelmöjten varit något ostadig och inte uppnått önskvärd pålitlighet. Med anledning av detta ska hela innanmätet byggas om. Det här projektet syftar endast till att konstruera styrning och mjukvara. Funktionerna ska vara de samma som tidigare och dessutom ska Växelmöjten förberedas för laddning av E-sektionen egna cashkort, ett system som byggs i ett annat projekt i kursen.

2.1 En Växelmöjts verksamhetsområde

Kort och gott kan Växelmöjten följande:

- Ta emot mynt eller 20-kronorssedlar för växling
- Mata ut mynt: 1x5 kr
 5x1 kr
 10x1 kr
- Mata ut resternade saldo i mynt
- Då summan inmatade kontanter uppgår till exakt 20 kr mata ut en summa slumpad mellan 16 och 24 kr. Funktionen kallas Jackpott.

3 Hårdvara och funktion

Byggstenarna i den fysiska konstruktionen är en display, en sedelacceptor, en myntacceptor, två processorkort med Atmel AVR Mega16 processorer, knappar för styrning av användargränssnittet, sensorer och motorer. I figur 3.1 finns en översiktsbild över ingående hårdvara. Förutom ovan nämnda komponenter ska den ena processorn också kommunicera med en PC för att kunna lagra statistik och felkoder i en databas.

Figur 3.1 Översiktsbild över hårdvaran

3.1 Processorerna

Valet att använda två processorer gjordes med hänsyn till att delar av konstruktionen är tänkt att sitta i en dörr och andra delar inne i ett skåp. Kablar riskerar då att skadas av dörrens gångjärn och av ständigt vikande. Kommunikationen mellan processorerna tar betydligt färre trådar i anspråk än vad styrningen av projektets ingående komponenter gör. Scheman för processorkorten finns i bilaga 1.

3.2 Display, knappar och acceptorer

Display och knappar är till för att hantera kommunikationen mellan människa och maskin. En av knapparna är tänkt att sitta på insidan av dörren för användning endast av administratören. Kretsar för att simulera acceptorerna har byggts efter mätningar av signalerna från de befintliga acceptorerna. Både mynt och sedelacceptor sänder ut en puls på en pinne specifik för mynt- eller sedelslaget. I prototypen simuleras mynt och sedlar av knapptryckningar som sänder motsvarande pulser. Knappar, display och acceptorer hanteras av processorn som sitter i dörren.

3.3 Motorer och sensorer

Inne i automaten finns 4 motorer som hanterar själva myntutmatningen och sorteringen

av inmatade mynt. Motorerna är kopplade till enkla drivkretsar. Dessa motorer måste styras utifrån användarens handlingar, såsom knapptryckningar och myntinmatningar. I den här delen av automaten finns även 12 stycken sensorer som även de används för själva mynthanteringen. Sensorer såväl som motorer styrs av processorn inne i skåpet.

3.4 PC

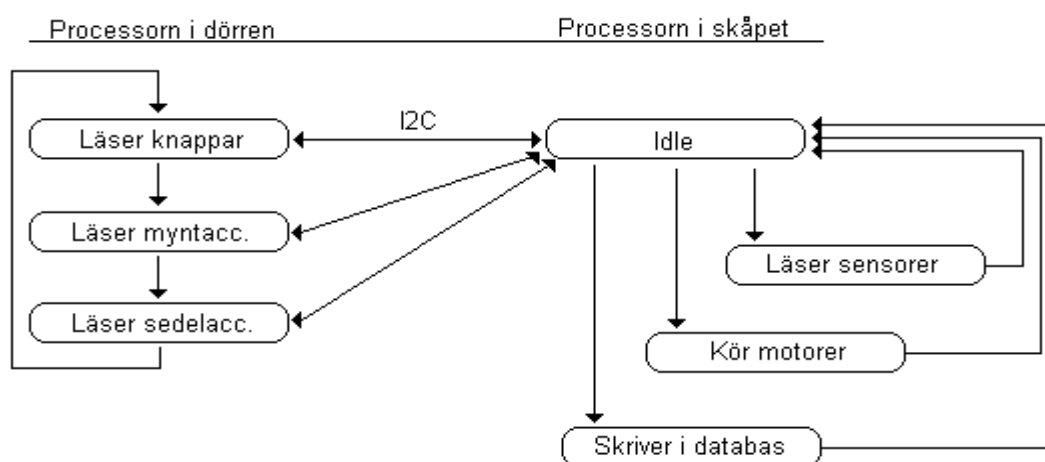
Anledningen till att koppla ihop processorerna med en PC är att kunna hantera ett felkodssystem med hjälp av databaser samt att i databaser lagra statistik över användning och funktion. Felkodsystemet är tänkt att vara ett skydd för användarens pengar, om växel inte kan matas ut ska en felkod lämnas. Denna felkod kan sedan lämnas till Växelmojtens administratörer för utjämning av penningbalans. Felkoderna ska kunna studeras och raderas med hjälp av knappsatsen, den extra knappen inne i automaten och displayen.

4 Mjukvara

All kod är skriven i C och som utvecklingsmiljö har AVRstudio 4 använts. På varje processor finns en mainloop som styr vad som händer i automaten med hjälp av en mängd olika rutiner. En översiktslista över rutinerna finns i bilaga 2. Processorerna kommunicerar med varandra via I2C-protokollet. Kommunikationen med datorn kommer att ske via UART och SPI men på prototypstadiet har den utelämnats.

4.1 Mainloopar

Den viktigaste delen av mjukvaran är de två mainloopar som styr allt som händer i automaten. En schematisk bild av looparnas funktion finns i figur 4.1. Processorerna är omväxlande master och slave beroende på vilken som har något att säga.



Figur 4.1 Mainloopar och internkommunikation

Dörrloopen kör hela tiden och väntar på interaktion från användaren. På signal från knappar eller acceptorer kommunicerar den med skåploopen som hela tiden väntar på instruktioner. Generellt sett kan det sägas att dörrloopen interagerar med användaren och skåploopen sköter maskinvaran.

4.2 Kommunikation

Kommunikationen med PC:n är till för att läsa och skriva i SQL-databaser men är ännu inte implementerad. Kommunikationen kommer att skötas via UART och SPI med hjälp av ett externt kommunikationskort. Då kortet ännu inte finns färdigt har databaser simulerats med hjälp av en vektor. Statistikdatabasen har än så länge utelämnats helt. Som redan nämnts sker kommunikationen mellan processorerna via I2C.

4.3 I2C-protokollet

I2c-protokollet är ett kommunikationsprotokoll som har uppfunnits och patenterats av Philips. På Atmels AVR-processorer heter den därför TWI, eller Two Wire Interface, för att undvika licensavgifter. I2C använder, som man kan gissa av namnet på Atmels version, två trådar för kommunikation (tre om man räknar med att det behövs en gemensam

jordpunkt också). De två trådarna är SDA och SCL som är seriell data respektive klocksignal. Dessa dras höga av pull-up-motstånd när ingen kommunikation sker, och dras låga av kretsarna under kommunikationsfasen. Kretsarna på en I2C-buss har en egen adress som används av den sändande enheten (Master) för att adressera datapaketet till mottagaren (Slave). Adressen består av 7 bitar i det första paketet i det enklaste standardutförandet, vilket betyder att det går att koppla in 127 enheter på bussen. Det går även att använda större adresser om antalet enheter kräver det. De följande paketet består av data, 8 bitar i taget tills utbufferten på mastern är tom.

I växelmojten finns det, som tidigare nämnts, två processorer, och följaktligen endast två enheter på I2C-bussen. Dessa växlar mellan att vara master och slave eftersom implementationen av I2C-protokollet är baserat på interrupt istället för polling. Detta möjliggör strukturen i programlooparna där det alltid är en av processorerna som väntar på att den andra skall bli klar med den pågående processen innan de båda fortsätter exekveringen.

5 Användargränssnittet

Idag har Växelmojten ett tämligen undermåligt användargränssnitt bestående av tre stycken sju-segments LED-displayer (egentligen fyra displayer, men en är trasig) och några knappar. Då något går fel visas en, även för den insatte, i de närmaste obegriplig felkod. Den stackars användaren har inte mycket annat att göra än att hämta någon med nyckel för att få hjälp. Då Växelmojten har den dåliga vanan att titt som tätt lägga beslag på användarens pengar blir det även en fråga om att bli trodd för att få tillbaka pengarna osv. Med felkodssystemet som nämnts tidigare är detta problem tänkt att förenklas.



Figur 5.1 Användargränssnitt

5.1 Knappar och display

Användaren kommunicerar med Växelmojten via ett gränssnitt baserat på knappar med variabla funktioner och en display, placerat enligt principen i figur 5.1. Knapparnas funktion klargörs för användaren genom text på displayen och kan därmed bytas ut för olika steg i processen. Till exempel då ett myntslag är slut visas inte detta som utmatningsmöjlighet. Displayen används för att instruera och informera användaren om vad som händer och vilka möjligheter och val som finns för tillfället och naturligtvis det aktuella saldot.

6 Mål och resultat

Nedan finns kravspecifikationen i sin helhet med tillhörande kommentarer gällande resultatet.

Mojten ska:

- kunna känna igen mynt som stoppas in
Myntacceptorn känner igen myntslagen och ger pulser på olika pinnar beroende på myntslag. Detta utnyttjas i koden.
- kunna känna igen sedlar som stoppas in (endast 20 kr vid växling)
Precis som myntacceptorn berättar acceptorn vilken typ av sedel det rör sig om. Även här används detta i koden.
- visa inmatat saldo på display
Det finns en variabel för att hålla reda på saldot.
- mata ut inmatat belopp i valfria myntvalörer i mån av tillgång
Det finns funktioner både för att exekvera och för att välja myntslag för utmatning.
- mata ut resterande belopp
Det finns en funktion för att mata ut resterade belopp.
- visa vilka myntvalörer som är slut på display
Detta är löst så att alternativet att mata ut en myntvalör inte finns då mynten av denna typ är slut.
- stänga av växlingsfunktionen då kort är isatt samt börja ta emot 100-lappar
Detta kommer att skötas av kortläsarprogrammet då det tagit över verksamheten.
- visa, i procent, hur mycket mynt av olika valörer som finns kvar, vid knapptryck
Funktionen finns förberedd men inte implementerad. Implementering kräver inkoppling av de sensorer som känner av hur många mynt som finns i varje rör.
- om något går fel någonstans ska det gå att få reda på via displayen var i processen
Ännu ej implementerat.
- återcirkulera instoppade mynt
Programvara och styrning är förberedd för detta.
- generera felkoder då inmatat belopp av någon anledning inte kan matas ut. Dessa ska visas på display.
Funktionen implementerad.

- lagra felkoder i en databas ur vilken en administratör kan radera dem
Implementerat temporärt med vektorer då den slutliga databasen ej är färdigställd.
- lagra statistik över vad som händer i databas
Uppskjutet i väntan på databaser, då funktionerna inte skiljer sig nämnvärt från felkodshanteringen men är betydligt rörigare att hantera med vektorer.

I mån av tid ska den också kunna:

- vid inmatning av 20 kr kunna leverera slumpad summa mellan 16 och 24 kr vid tryck på jackpott.
Funktionen implementerad.
- visa statistik på display
Uppskjutet i väntan på databas.

7 Problem

Problem som stötts på under projektets gång:

- Det första lite större problemet som stöttes på var att funktionen för att skriva till displayen inte fungerade riktigt som den borde. Detta berodde på en bortglömd pekare. När detta hade rättats till fungerade funktionen som den skulle.
- Ett annat problem som uppstått under projektets gång var att få en bra jackpot-funktion. De matematiska funktionerna för att räkna ut en bra normalfördelning accepterades inte av kompilatorn. Detta visade sig vara ett länkningsproblem till följd av att kompilatorn inte inkluderade mathbiblioteket. Detta löstes med en parameter `-lm` till kompilatorn.
- Ett annat problem vid beräkning av ett normalfördelat jackpot-resultat berodde på processorns något knackiga flyttalshantering samt det faktum att division mellan två stycken 32-bitars flyttal inte är en enkel operation på en 8-bitars processor. Detta löstes genom att skapa en vektor med önskad fördelning och slumpa vilket element som skulle användas med hjälp av en rektangelfördelat slumpvalsgenerator istället.
- När kommunikationen mellan de två processorerna skulle fås igång uppstod problem med att interuptet vid inkommande packet inte fungerade. Även detta var ett kompilatorproblem. Den "optimerade" bort en del av funktionen som skulle hantera detta interuptet. Detta löstes med att deklarerera variabeln som påverkades av denna något oönskade optimering som volatile.
- Vid ett tillfälle bråkade även hårdvaran. Det resulterade i två trasiga processorer. Detta problem uppstod till följd av att det under uppstart fanns flera komponenter som samtidigt försökte driva samma buss. Detta löstes med hjälp av några pullup-motstånd.
- Bortsett från ovanstående har det uppstått diverse problem med c-syntaxen och en del enklare problem med hårdvaran, oftast har den mänskliga faktorn här spelat en avgörande roll.

8 Kommentarer

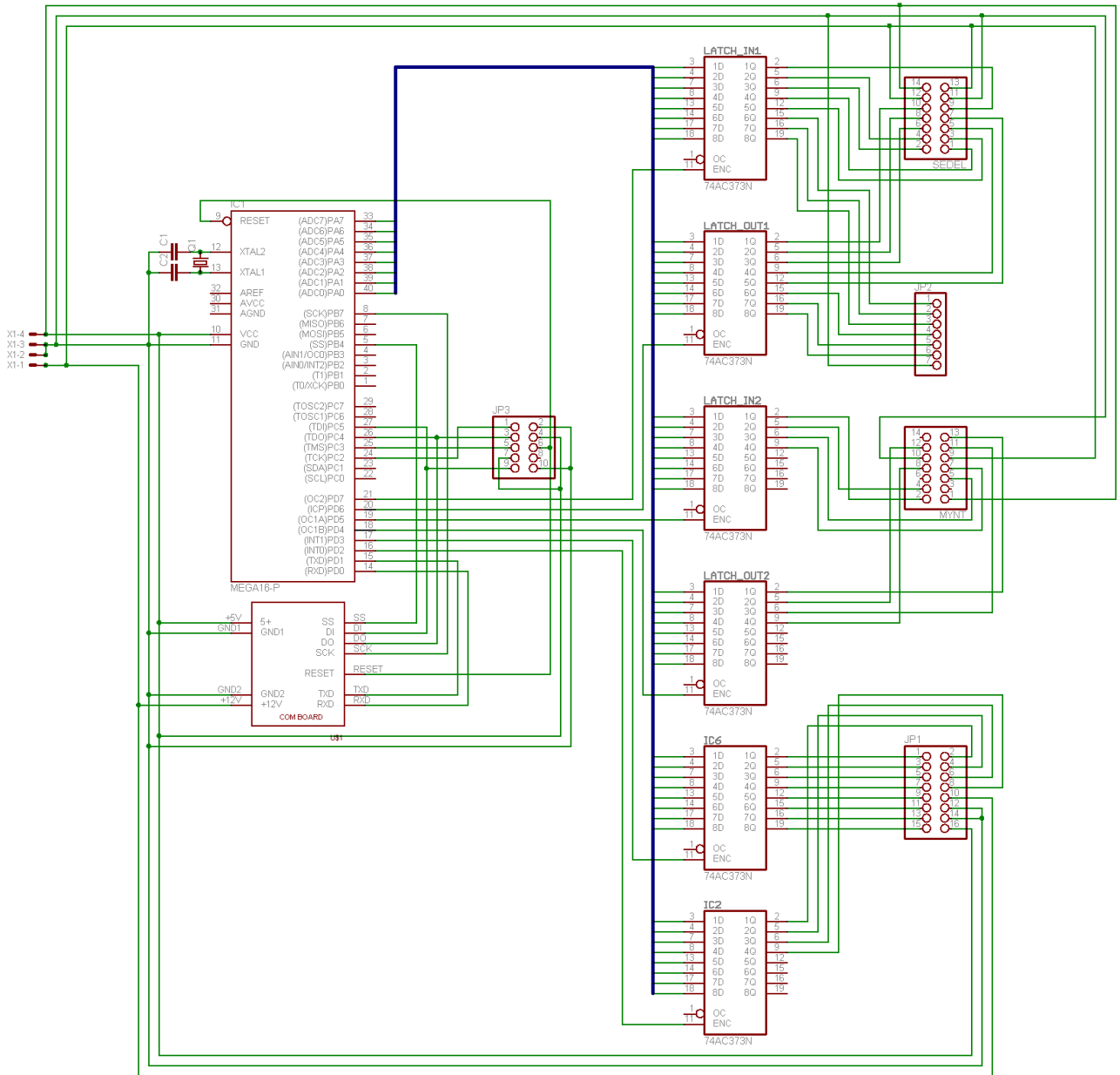
Projektet har i stort sett flutit på som planerat efter en något seg start, mycket beroende på problem med att reda ut acceptoreernas pinnkonfiguration. Förutom de problem som tagits upp i tidigare avsnitt har arbetet gott framåt utan direkta avbrott. De funktioner som hittills implementerats känns tillfredställande. Än finns en del arbete kvar att göra innan prototypen kan realiseras i sin verkliga tillämpning, inte minns ombyggnation av själva maskinvaran. Det råder dock tillförsikt om att projektet i sin helhet, alltså även det som faller utanför ramen för denna kurs, kommer att slutföras. Förhoppningsvis kommer det leda till att E-sektionen kan njuta av en bättre fungerande Växelmojt än idag.

9 Sammanfattning

Rapporten behandlar konstruktionen av en elektronisk prototyp, baserad på en microprocessor, som byggts i kursen Digitala projekt, större kurs. Prototypen i fråga är ett försök att skapa styrning och användargränssnitt till en växelautomat, kallad Växelmojten. Prototypen är uppbyggd av två Atmel AVR Mega16 processorer som tillsammans styr maskinvara, användargränssnitt och övriga operationer i Växelmojten. Ingående komponenter är bl a display, knappar, motorer och sensorer. Processorerna kommunicerar med varandra över I2C-protokollet och med en dator via UART och SPI. Kopplingen till datorn är huvudsakligen till för att lagra statistik och felkoder i SQL-databaser.

Koden är skriven i programmeringsspråket C och utvecklingsmiljön som använts är AVR studio 4. Meningen är att prototypen ska komma till användning i verkligheten. Projektet har flutit på utan större missöden.

Bilaga 1:1 Processorkortet i dörren



Bilaga 1:2 Processorkortet inuti skåpet

