

Institutionen för Informationsteknologi LTH
Projektarbete

Nummerfjolla

Henrik Svensson
Johan Strand
1999-12-14



Abstract

The purpose of this project was to develop and build a Caller ID Box with an Industrial approach. The features contains presentation of: number of new calls, date and time for incoming calls and a opportunity to automatically dial an incoming telephone number by a press of a button. It was important to allow further development of the product and that the project was well documented. A number of additional requirements were specified in the project requirement specification to be implemented in amount of time. The additional features included a Telephone book with an opportunity to store thirty names and telephone numbers and presentation of the name of the dialer's name instead of the dialer's telephone number. Even a feature to blacklist any number to prevent the dial signal from getting through was included in the requirement specification. The final goal was to fulfill as many of those requirements as possible. The project turned out to be a success and all requirements were fulfilled even an opportunity to dial a number stored in telephone book were included.

Innehåll:

1.	Inledning	4
2.	Målsättning	4
3.	Förutsättningar	4
3.1.	Nummerpresentatörs tjänsten	4
4.	Grundläggande strategi	5
4.1.	Kravspecifikation.....	5
4.2.	Högnivå design	5
4.3.	Lågnivå design.....	6
5.	Grunduppkoppling med M68HC11	6
6.	Kringutrustning.....	6
6.1.	DTMF-transceiver	6
6.1.1.	Kringkomponenter	7
6.2.	LCD-skärm	7
6.3.	Övriga komponenter	7
7.	Programvara.....	8
7.1.	Strategi.....	8
7.2.	Styr signaler till hårdvaran.....	8
7.2.1.	Styr signaler till DTMF	8
7.2.2.	Styr signaler till Display	8
7.2.3.	Övriga styr signaler.....	9
7.3.	Viktiga globala variabler	9
7.4.	Lågnivå specifikation.....	10
7.4.1.	DTMF-transceiver	10
7.4.2.	LCD-skärm	11
7.4.3.	68HC11	11
7.4.4.	Knappar	12
7.5.	Högnivå specifikation.....	12
7.5.1.	DTMF-transceiver	12
7.5.2.	LCD-skärm	12
7.5.3.	68HC11	13
8.	Utbyggnadsmöjligheter.....	13
9.	Brister	14
10.	Referenser	14
11.	Appendix	14

1. Inledning

Detta dokument beskriver projektet Nummerfjolla (hårdvara och programvara) som har utvecklats i kursen Digitala Projekt under höstterminen 1999. Kursen ges av Institutionen för Informationsteknologi vid Lunds Tekniska Högskola. Dokumentet beskriver målsättningar (kapitel 3), förutsättningar (kapitel 4), en kort genomgång av nummerpresentatörstjänsten samt grundläggande strategier vid högnivå respektive lågnivå design (kapitel 5). Vidare beskrivs den grundkoppling (kapitel 6) och kringutrustning (kapitel 7) som användes. Slutligen specificeras programvaran som utvecklats i samband med projektet (kapitel 8), utbyggnadsmöjligheter (kapitel 9) och brister (kapitel 10).

2. Målsättning

Målsättningen för projektet var först och främst att implementera en nummerpresentatör för visning av nya inkommande samtal med nummer, datum och tid, samt uppringningsmöjlighet av visat nummer. Vidare skulle vår lösning ge möjligheter till utbyggnad för mer avancerade funktioner såsom: Telefonbok med uppringningsfunktion, möjlighet att lägga till nytt namn och nummer i telefonboken, möjlighet att ställa in tid och datum samt att kunna svartlista vissa telefonnummer.

Målsättningen för projektet sattes redan första veckan i kravspecifikationen (se Appendix) och projektets uppgift blev sedan att uppfylla så många som möjligt av de ställda kraven.

Två av kraven som specificerades i kravspecifikationen [Krav 9, Krav 10] fick lägre prioritet än övriga krav och skulle därmed implementeras först sedan övriga krav var uppfyllda och tidsutrymme fanns.

3. Förutsättningar

Förutsättningarna för projektets arbete var att bygga nummerpresentatören antingen kring M68HC11 eller M68000 och redan första veckan bestämdes att M68HC11 skulle användas. Anledningarna till detta var dels den förenklade hårdvarubilden. Denna skulle ge projektet mer tid och möjlighet att uppfylla kraven som till stor del låg på mjukvaran. Dels ansågs M68HC11 ge tillräckliga resurser för att kunna uppfylla målbilden.

Vidare var förutsättningarna att den samlade arbetsinsatsen för projektet skulle var cirka 240 timmar, fördelade på en sju veckors period.

Till projektets förfogande fanns en informationsbank på nätet över tillgängliga kretsar, kringkomponenter som fanns på institutionen för Informationsteknologi och tillgång till kurslaboratoriet dygnet runt.

3.1. Nummerpresentatörstjänsten

En självklar förutsättning för att bygga en nummerpresentatör är att det finns en tjänst som stöder detta d.v.s. numret måste på något sätt sändas från växel till abonnent. Detta görs genom DTMF signalering (Dual Tone Multi Frequency). I korthet går denna signalering ut på att ett tonpar som skickas representerar ett tecken. Ett tecken sänds ut under 70 ms och följs av en 70 ms lång paus. Protokoll för sändning av nummer från växel till abonnent ser ut som följer:

([A S1 S2 S3 . . . Sn] v [D S1 S2 S3 . . . Sn] v [D S1 S2 S3 . . . Sn] v [B S1 S2]) & C

A	Starttecken för anropandes nummer.
$S_1 \dots S_n$	Siffror, 0 till 9, n max 15.
D	Starttecken för vidarekopplat nummer.
C	Stopptecken.
B	Starttecken för informationskod. Skickas vid utlandssamtal, hemligt nummer eller då en växel ej stödjer nummerpresentation.

4. Grundläggande strategi

4.1. Kravspecifikation

Kravspecifikationen skrevs under den första arbetsveckan och inledde arbetet med projektet. Kravspecifikationen finns i Appendix 1.

4.2. Högnivå design

Under andra veckan gick projektet in på högnivå designen. Det fastslogs då vilka kretsar som skulle ingå för att kunna uppfylla de krav som ställdes i kravspecifikationen samt hur dessa skulle kommunicera med M68HC11.

Följande kretsar/komponenter togs upp i högnivå designen:

- DTMF transceiver MT8880C (för mottagning och sändning av DTMF signaler)
- LCD skärm VK 2x16 (display för presentation)
- Knappsats (tre stycken: Pil upp, pil ner och ring upp)

Den fjärde knappen (Mode) lades till i ett senare skede då det upptäcktes att den specificerade funktionaliteten inte enkelt kunde implementeras med tre knappar.

Port C på M68HC11 konstruerad för dubbelriktat dataflöde [2] vilket gör den mycket lämplig att använda till parallell datakommunikation mellan de olika kretsarna och M68HC11. Detta gjorde att det var naturligt att ansluta alla enheters data in och utgångar till denna port. Förutsättningen för en sådan lösning är att det finns en styrsignal på varje krets som gör att in/utgångar på kretsen blir högimpediva, för att inte kortsluta kretsen.

Till styrsignalerna, som bestämmer vilken av de ingående kretsarna som skall vara aktiv (sända eller ta emot data) och vilka som skall var inaktiva (högimpediva ingångar), är M68HC11:ans port B lämplig eftersom denna bara innehåller utgångar [2]. Genom att använda port B enbart till styrsignaler kan man kommunicera med maximalt fyra om man begränsar sig till att varje enhet har maximalt fyra styrsignaler förutom chipset select. Med en yttre avkodare kan ytterligare enheter anslutas utan större förändringar i programvaran. Detta gynnar en framtida utbyggnad av nummerpresentatören med kommunikation mellan fler enheter.

RAM minnet i M68HC11:an är 512 byte [1]. Detta ansågs fullt tillräckligt då telefonnummer tillsammans med datum och tid skulle uppta 12 byte (8 byte för 16 siffror, 0 till 9, plus 4 byte för datum och tid). D.v.s. det minnesutrymme som totalt skulle krävas för lagring av 30 inkommande samtal skulle vara 360 byte.

Projektet hade i denna fas av arbetet en farhåga att minnesutrymmet i M68HC11:an inte skulle räcka till för att kunna implementera en telefonbok med 30 namn och nummer, eftersom storleken på M68HC11:ans interna EEPROM berodde på version och typ av krets. Därför togs även ett seriellt EEPROM med i högnivådesignen. Senare visade det sig att den typ av M68HC11 som skulle användas hade 512 byte [1] internt EEPROM vilket är lagom för lagring av 30 namn och

telefonnummer. Något extra seriellt EEPROM behövdes då inte användas i konstruktionen. Högnivådesignen finns i Appendix 2.

4.3. Lågnivå design

Den tredje projektveckan påbörjades lågnivådesignen som gick ut på att läsa datablad och rita kretsschema. I stort sätt blev detta arbete en förlängning av det arbete som gjordes i samband med högnivådesignen.

Samtliga ingående kretsar hade färdiga standarduppkopplingar i databladerna [1][3] och dessa användes rakt av med undantag av M68HC11:ans resetkrets (se kapitel 6).

Lågnivådesignen finns i Appendix 3.

5. Grunduppkoppling med M68HC11

M68HC11 körs som enchipps dator i det så kallade single chip mode. Kopplingen är enligt [1] Basic Single-Chip-Mode med undantag av reset kretsen som utelämnades på grund av att denna prototyp alltid kommer att användas med en stabil matningsspänning. Den slutgiltiga produkten borde innehålla resetkrets för att gardera sig mot oväntade spänningsfall.

6. Kringutrustning

6.1. DTMF-transceiver

För att både kunna ta emot och skicka information valdes kretsen MT8880C från Mitel. Detta är en krets som kan användas i många sammanhang då man vill generera DTMF (Dual Tone Multi Frequency) toner. Dessa toner är de man hör då man slår ett nummer på en normal tonvals telefon. DTMF tonerna är en blandning av två toner som normalt genereras från ett telefon tangentbord med en frekvens för varje rad och en för varje kolumn enligt tabell 1. Dessa symboler kodas i transceivern enligt tabell 2.

	1209Hz	1336Hz	1477Hz	1633Hz
697Hz	1	2	3	A
770Hz	4	5	6	B
852Hz	7	8	9	C
941Hz	*	0	#	D

Tabell 1.

Siffr	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
0	1	0	1	0
*	1	0	1	1
#	1	1	0	0
A	1	1	0	1
B	1	1	1	0
C	1	1	1	1
D	0	0	0	0

Tabell 2.

Kretsen var mycket enkel att använda eftersom den sköter om allt som har med sändning och mottagning av DTMF toner att göra, med få yttre komponenter. Det finns ett antal olika modes som kretsen kan arbeta i. I Burst mode genereras tonstötter med 51ms längd/mellanrum som är standard för automatiska uppringare mm.

Kretsen är väl anpassad för att arbeta ihop med en microprocessor som M68HC11 med avbrottsignal, dubbelriktade fyra bitars data in/utgångar av tri-state typ (D₃-D₀), registervalsgång, läs/skriv och Chipset select signaler. MT8880C har fem interna register, två kontrollregister där de olika arbetsätten bestäms, ett status register, mottagarregister och sändarregister.

6.1.1. Kringkomponenter

Kringkomponenter föreslagna som standard lösning i databladet [3] användes: kristall, avkopplingskondensatorer och motstånd. Styr kretsen bestämmer hur lång tid en viss DTMF ton skall finnas på ingången för att denna skall detekteras. Tiden sattes till 23ms som föreslagits i databladet [3]. Om man skall använda mottagaren på teleledningar med mer störningar kan denna krets behövas modifieras så att man ökar detekteringstiden.

6.2. LCD-skärm

En standard tvåradig LCD display med plats för 32 tecken används för att presentera användargränssnittet. Displayen använder en instruktionsuppsättning med 11 instruktioner [4] för skrivning, läsning av data samt inställningar av olika visnings och inmatningslägen. Data består av ASCII koder för olika tecken vid skrivning till eller läsning från skärmen.

6.3. Övriga komponenter

De fyra knapparna implementerades som aktivt låga med pull-up motstånd. Vidare användes avkopplingskondensatorer på 0,1µF till samtliga kretsar. Avkopplingskondensatorerna placerades nära varje krets så att goda avstörnings egenskaper erhöles.

Till M68HC11 användes pull-up motstånd på alla ingångar [1] eller portar som kan konfigureras som ingångar för att se till att inga ingångar blev oanslutna och därmed få ett odefinierat värde.

7. Programvara

All programvara skrevs i C på Windows NT arbetsstationer med hjälp av IAR's Embedded Workbench och debuggern CSPY.

Funktionen enligt Appenix 1 krävde ett antal olika arbetssätt för nummerpresentatören. Därför introducerades fyra olika modes: Normal Mode, Set Time Mode, Telephone book Mode och Edit Telephone book Mode. Nummerpresentatören skulle enligt Appendix 1 arbeta i Normal Mode då inga specialfunktioner användes varför Normal Mode implementerades i huvudprogrammet. De övriga arbetssätten implementerades sedan i olika procedurer som ändrar funktionen på knappar och skriver ut för arbetssättet väsentliga saker på skärmen se Appendix 4.

7.1. Strategi

Strategin när det gäller mjukvaran har varit att först bygga upp en funktionsuppsättning som ligger närmast hårdvaran. Denna funktionsuppsättning byggdes ut allt eftersom kommunikation med enheter testades. Den egentliga programmeringen påbörjades först då en gedigen instruktions uppsättning till alla hårdvarudelar var klar och testad.

7.2. Styr signaler till hårdvaran

7.2.1. Styr signaler till DTMF

const char PB_WRITE_DTMF=0x00;

Läggs ut på port B då skrivning till sändarregistret skall göras.

const char PB_READ_DTMF=0x02;

Läggs ut på port B då läsning av mottagarregistret skall göras.

const char PB_WRITE_DTMF_REG=0x04;

Läggs ut på port B då skrivning till controlregister A skall göras.

const char PB_READ_DTMF_REG=0x06;

Läggs ut på port B då läsning av statusregistret skall göras.

7.2.2. Styr signaler till Display

const char PB_WRITE_DISP_INSTR1=0x01;

Läggs ut på port B, sätter RS Och R/W' för skrivning av instruktion.

const char PB_WRITE_DISP_INSTR2=0x11;

Läggs ut på port B, instruktion till display skall finnas på port C.

const char PB_READ_DISP_BUSY1=0x21;

Läggs ut på port B, sätter RS och R/W' för läsning av busyflaggan.

const char PB_READ_DISP_BUSY2=0x31;

Läggs ut på port B, läser busy flaggan (bit 7 på Port C) +adress.

const char PB_WRITE_DISP1=0x09;

Läggs ut på port B, sätter RS och R/W' för skrivning av data.

const char PB_WRITE_DISP2=0x19;

Läggs ut på port B, för skrivning av data.

const char PB_WRITE_DISP_ADDRESS1=0x01;

Läggs ut på port B, sätter RS och R/W' för skrivning av adress.


```
const char PB_WRITE_DISP_ADDRESS2=0x11;
    Läggs ut på port B, för skrivning av adress.
```

7.2.3. Övriga styrsignaler

```
const char PB_ALL_DISABLE=0x01;
    Läggs ut på port B, sätter Display och DTMF disable.
const char DDRC_INPUT=0x00;
    Skrivs till DDRC, sätter Port C till ingång.
const char DDRC_OUTPUT=0xff;
    Skrivs till DDRC, sätter Port C är utgång.
```

7.3. Viktiga globala variabler

```
char StackSize
    Antal element i NbrStack (se nedan).
```

```
char NbrStack[StackSize][12]
    Matris där inkommande nummer samt tid lagras på formen:
```

Ex. Inkommande samtal från 55590000 den 24/12 kl 23:59

0	1	2	3	4	5	6	7	8	9	10	11								
5	5	5	9	0	0	0	0	A	-	-	-	-	-	-	b	23	11	23	59

↖ Slutsymbol

↖ Datum och månad lagras som aktuellt datum minus ett.
↖ Indikerar ej besökt nummer. Blir a efter nummret har bläddrats fram

Kommentar: Det lagras alltså två siffror i telefonnumret på en byte. De fyra mest signifikanta bitarna representerar den första siffran i en byte och de fyra minst signifikanta bitarna representerar den andra siffran.

```
char TBSize
    Antalet element i TeleBook.
```

```
no_init char TeleBook[TBSize][16]
    Matris där namn och telefonnummer lagras. Nummret lagras på samma sätt som i NbrStack medan namnet lagras på position 8-15.
```

```
char Timer
    Räknas ner varje sekund tills den blir noll.
```

```
char RecMode
    En etta anger att DTMF:en håller på att ta emot ett nummer. Sätts till noll då slutsymbol (C) kommer.
```

char TBStackMode

En etta anger att nummerpresentatören är i TBStackDispMode.

char NewCalls

Räknas upp varje gång ett nytt samtal inkommer och räknas ned varje gång ett nytt samtal bläddras fram. Blir dock aldrig större än StackSize.

char StackDispNbr

Anger vilken av elementen på stacken som displayen visar.

char Time[4]

Vektor som innehåller aktuellt datum och tid.

char StackNbr

Nästa inkommande nummer lagras på platsen NbrStack[StackNbr].

7.4. Lågnivå specifikation

7.4.1. DTMF-transceiver

void Fi2(void)

Ger en puls på insignalen Fi2 på DTMF transceivern. Användes vid skrivning till MT8880C efter att data finns på PortC.

void FiHi(void)

Sätter insignalen Fi2 på DTMF transceivern till hög.

void FiLow(void)

Sätter insignalen Fi2 på DTMF transceivern till låg.

Funktionerna FiHi och FiLow användes vid läsning av datautgångarna från DTMF transceivern. Efter FiHi finns data på utgångarna varefter FiLow används för att återställa vilonivån på signalen.

void DTMFSetRecMode(void)

Sätter MT8880C i mottagningsläge och konfigurerar PortC som ingång.

void DTMFSetSendMode(void)

Sätter MT8880C i sändningsläge och konfigurerar PortC som utgång.

char DTMFReadSR(void)

Läser MT8880C statusregister. Användes för att kvittera avbrott.

void InitDTMF(void)

Initerar DTMF transceivern och sätter denna i mottagningsläge.

char DTMFGetDigit(void)
Läser en mottagen symbol från DTMF transeivern. Användes med fördel vid avbrott.

void DTMFSendDigit(char Digit)
DTMF:en sänder symbolen Digit.

7.4.2. LCD-skärm

void ClearDisplay(void)
Ger tom display och sätter markören längst upp till vänster.

void DispReturnHome(void)
Sätter markören längst upp till vänster.

void DispOnCursorOff(void)
Sätter på displayen med ingen markör synlig.

void DispOnCursorOn(void)
Sätter på displayen med markör synlig.

char DispBusy(void)
Returnerar ett om displayen är upptagen med läsning eller skrivning annars noll.

void DispWrite(char ch)
Skriver ett tecken (ch) på displayen och skiftar markören ett steg åt höger.

void SetDispPos(char address)
Sätter markörens position till address.

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Rad 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Rad 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

Kursivt = ram adresser

void InitDisp(void)
Initierar displayen till tvåradig, sätter högerskift av markören efter skrivning. Skall köras innan användning av displayen.

7.4.3. 68HC11

void EEMemErase(char ItemNbr,char Pos)
Raderar en byte på EEPROM. Data i TeleBook[ItemNbr][Pos] raderas.

void TBWrite(char ItemNbr,char Pos,char Data)
Skriver en byte data till Telefonboken. Data skrivs till TeleBook[ItemNbr][Pos].

void init(void)
Kör alla initieringsrutiner och konfigurerar portar och avbrott. Körs en gång vid uppstart.

7.4.4. Knappar

char IsKeyUpDown(void)

Returnerar 1 om ”knapp upp” är nedtryckt, annars 0.

char IsKeyDownDown(void)

Returnerar 1 om ”knapp ned” är nedtryckt, annars 0.

char IsKeyRDown(void)

Returnerar 1 om ”ring upp” är nedtryckt, annars 0.

char IsKeyModeDown(void)

Returnerar 1 om ”mode” är nedtryckt, annars 0.

7.5. Högnivå specifikation

7.5.1. DTMF-transceiver

void DTMFSendStackNbr(char StackItem[8])

DTMF:en sänder numret som ligger i vektorn StackItem. Numret i StackItem skall vara lagrat på samma sätt som i NbrStack.

void RecNbrIntoNbrStack(char RecNbrAndTime[20])

Tar vektorn RecNbrAndTime packar om den till rätt format och lägger in den i NbrStack. Position 0 till 15 ska innehålla telefonnumret och position 16 till 19 innehåller datum och tid.

7.5.2. LCD-skärm

char DigitToASCII(char Digit)

Konverterar ett ensiffrigt tal till motsvarande ASCII tecken.

void DispWriteStr(char Str[])

Skriver ut en Ansi C sträng. Inparametern Str[] skall vara en vektor med tecken som avslutas med NULL ('\0')

void DispWriteTime(char TimeV[4])

Skriver ut innehållet i TimeV på formen TimeV[0]/TimeV[1] TimeV[2]:TimeV[3] med början på adress 45. Funktionen konverterar siffrorna i TimeV till motsvarande ASCII tecken innan utskrift.

void DispWriteStackItem(char ItemNbr)

Skriver ut ett element i NbrStack på formen:

Telefonnummer	
Nya samtal	Datum

Om ItemNbr är lika med StackSize så skrivs ”Nr. Fjolla vX.X” ut samt aktuell tid, datum och antal nya samtal. Om platsen ItemNbr i NbrStack inte innehåller något nummer skrivs ”---Empty---” ut samt aktuell tid, datum och nya samtal.

void DispWriteTB(char ItemNbr)

Skriver ut elementet ItemNbr från telefonboken. TeleBook[ItemNbr] skrivs ut på skärmen på följande form:

Telefonnummer (max 16 tecken)	
Namn (max 8 tecken)	ItemNbr

Om ItemNbr är lika med TBSize så skrivs "Telephonebook". Om platsen ItemNbr i TeleBook inte innehåller något nummer skrivs "---Empty---".

7.5.3. 68HC11

void AddTime(char Pos)

Lägger till ett datum, en månad, timma eller minut till aktuell tidpunkt. Time[Pos] ökas med ett.

void SubTime(char Pos)

Drar ifrån ett datum, en månad, timma eller minut från aktuell tidpunkt. Time[Pos] minskas med ett.

void SetTime(void)

Ett av fyra modes. Stänger av alla avbrott och möjliggör inställning av tid med hjälp av knappar. Avbrott sätts på igen efter avslutad SetTime

void TBWriteItem(char TBVector[16],char ItemNbr)

Skriver TBVector till TeleBook[ItemNbr]. TBVector skall ha samma format som ett element i TeleBook.

void TBEditMode(char ItemNbr)

Ett av fyra modes. Stänger av avbrott och möjliggör redigering av elementet TeleBook[ItemNbr] med hjälp av knapparna. Avbrotten sätts på efter avslutad TBEditMode.

char ItemNbrInTB(char StackItemNbr)

Returnerar elementnummer i Telefonboken om nummret i NbrStack[StackItemNbr] finns i Telefonboken. Annars returneras TBSize.

void TBStackDispMode(void)

Ett av fyra modes. Möjliggör bläddring i Telefonboken samt uppringning av nummer lagrade i Telefonboken med hjälp av knapparna.

8. Utbyggnadsmöjligheter

Genom att låta all datatrafik gå mellan port C och övriga enheter kan man på ett enkelt sätt bygga ut kopplingen med fler enheter. Dessa förändringar kan implementeras utan större förändringar i programvaran.

Eftersom programvaran är byggd med generella funktioner för olika hårdvaruoperationer kan den lätt byggas på och nya arbetssätt introduceras. M68HC11:an har inbyggd hårdvara för seriell kommunikation till externa enheter som datorer. Detta kan man utnyttja för att t.ex. presentera

nummer över dator nätverk. Man kan även utnyttja MT8880C för andra ändamål än nummerpresentation så som i Home Automation tillämpningar.

9. Brister

Den absolut största bristen för denna prototyp är minnes utrymmet i RAM som kraftigt begränsar utbyggnadsmöjligheterna i programvaran. Vidare så uppkommer lågfrekvent brus då man kopplar in utgången Tout från MT8880C på telefon nätet. Detta kan åtgärdas genom att låta ett relä styra inkoppling av Tout då denna behövs eller genom att konstruera ett filter.

10. Referenser

- [1] Motorola M68HC11 Preference Manual.
- [2] Motorola Semiconductor Technical Data MC68HC11E9/D Advance Information.
- [3] Mitel MT8880C Integrated DTMF Transceiver Datasheet.
- [4] Institutionen för Informationsteknologi LTH Digitala projekt Datablad HC11

11. Appendix

1. Kravspecifikation.
2. Högnivådesign.
3. Lågnivådesign.
4. Användarmanual för Nummerfjolla.
5. Programutskrift.