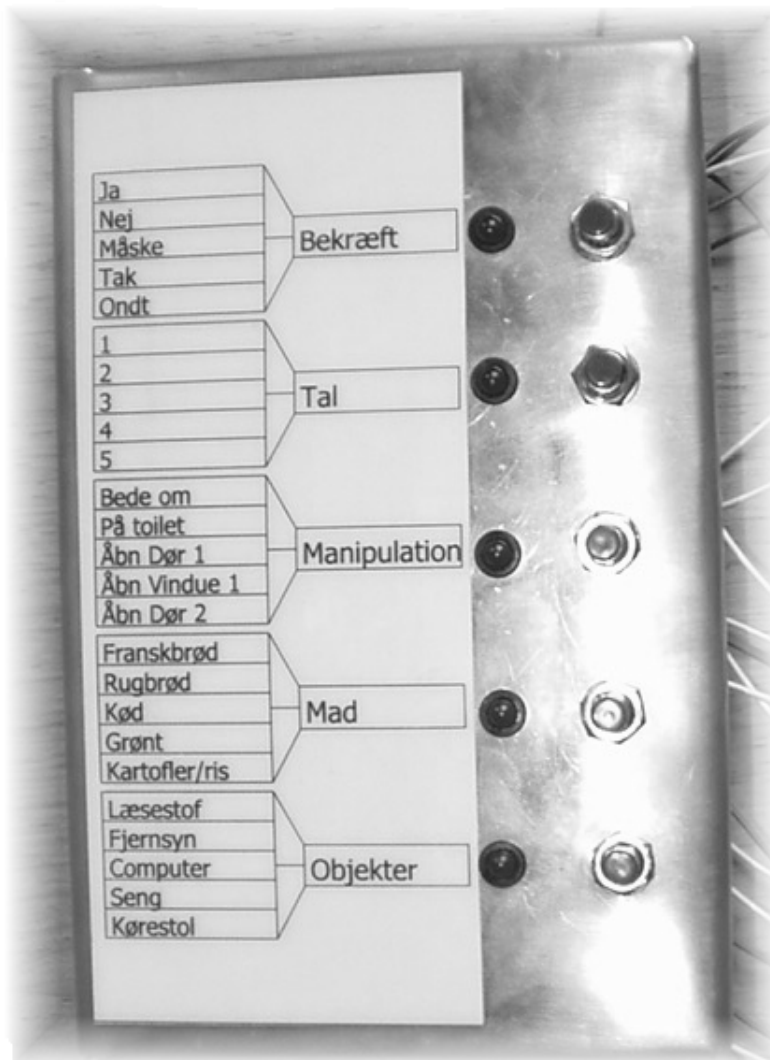


## Talkboard



Af Michael Hansen  
Jes Toft Kristensen

Teknik A  
HTX, CEU Herning  
6. semester  
27. april 2003

## 0.1 Forord

Denne rapport er et afsluttende projekt på htx uddannelsen, CEU Herning. Den er opbygget med projektbeskrivelsen først, så en beskrivelse af den egentlige fremstilling af produktet. Refleksioner mv. er til sidst i hver del, idet vi mener dette vil være mest overskueligt. Personen der har stået for hvert enkelt kapitel er nævnt med initialer i indexet. Disse initialer er MH - Michael Hansen og JTK - Jes Toft Kristensen. Hvor der ikke er noget anført står vi begge inde for emnet.

Vores vejledere til projektet er Hjalmar K. Andersen (HKA) og Henrik Larsen (HEL).

## **Resumé**

Dette projekt omhandler fremstillingen af et talkboard. Dette er et hjælpemiddel til handicappede, helt eksakt til personer der har problemer med at tale. Talkboardet er i sig selv en letanvendelig konsol koblet til en PC, hvor konsollen udgør interfacet mellem menneske og system, imens PCen sørger for at afspille lyde.

Michael Hansen  
Jes Toft Kristensen  
CEU Herning  
27. april 2003

# Indhold

0.1	Forord . . . . .	i
<b>I</b>	<b>Projektbeskrivelse</b>	<b>1</b>
<b>1</b>	<b>Projektbeskrivelsen</b>	<b>2</b>
1.1	Gruppen . . . . .	2
1.2	Projektanalyse . . . . .	2
1.2.1	Handicaphjælpemidler . . . . .	2
1.2.2	Det moderne kontor . . . . .	2
1.2.3	Overvågning og alarm . . . . .	3
1.3	Valg af tema . . . . .	3
1.4	Ide-fase . . . . .	3
1.5	Interview . . . . .	4
1.5.1	Målgruppen . . . . .	4
1.5.2	Funktioner . . . . .	5
1.5.3	Formen . . . . .	5
1.5.4	Konklusion på interviewet . . . . .	5
1.6	Problemformulering . . . . .	5
1.7	Afgrænsning . . . . .	6
1.8	Tidsplan . . . . .	7
<b>II</b>	<b>Elektronikdelen</b>	<b>9</b>
<b>2</b>	<b>Introduktion</b>	<b>10</b>
2.1	Formålet med elektronikken . . . . .	10
2.2	Den overordnede bløde gennemgang . . . . .	10
<b>3</b>	<b>Knapper (JTK)</b>	<b>12</b>
3.1	Opbygning/beregning . . . . .	12
3.2	Print til knap-delen . . . . .	15
<b>4</b>	<b>LED styring (MH)</b>	<b>16</b>
4.1	Opbygning . . . . .	16
4.2	Fremstillingen . . . . .	17
<b>5</b>	<b>Global out (JTK)</b>	<b>19</b>
5.1	Opbygning . . . . .	19

<b>6 Latche (JTK)</b>	<b>21</b>
6.1 Funktion . . . . .	21
6.2 Om latchene . . . . .	21
6.3 Opbygning . . . . .	22
<b>7 Delay-delen (MH)</b>	<b>25</b>
7.1 Ide/teori . . . . .	25
7.2 Udførelse . . . . .	28
<b>8 EPROM'en (JTK)</b>	<b>30</b>
8.1 Inputsekvens/programmering . . . . .	30
8.2 Fremstilling . . . . .	31
<b>9 Parallelporten (MH)</b>	<b>34</b>
<b>10 Produktudvikling</b>	<b>37</b>
10.1 Udgangspunktet . . . . .	37
10.2 Print . . . . .	37
<b>III Mekanikdelen</b>	<b>40</b>
<b>11 Chassis (JTK)</b>	<b>41</b>
<b>12 Døråbneren (MH)</b>	<b>43</b>
12.1 Udarbejdelse . . . . .	43
<b>IV Software programmet</b>	<b>47</b>
<b>13 Software programmet (MH)</b>	<b>48</b>
13.1 Programmeringen . . . . .	49
<b>14 Lyde</b>	<b>52</b>
<b>V Opsamling</b>	<b>54</b>
<b>15 Opsamling</b>	<b>55</b>
15.1 Projektet rent tidsmæssigt . . . . .	55
15.2 Projektet økonomisk . . . . .	55
15.3 Ændringer/fejl . . . . .	56
<b>16 Vurdering</b>	<b>58</b>
<b>17 Konklusion</b>	<b>59</b>
<b>VI Bilag</b>	<b>I</b>
<b>18 Elektronik bilag</b>	<b>II</b>
18.1 Folde ud ark . . . . .	II

18.2 Programscriptet . . . . .	III
18.3 Dagbog . . . . .	XV

# Figurer

1.1	Ide til talkboardets udformning . . . . .	3
1.2	Vores nuværende ide som blokdiagram . . . . .	6
3.1	Diagram over knapperne . . . . .	13
3.2	Oscilloskopsimulering af RC-ledet 1 . . . . .	14
3.3	Oscilloskopsimulering af RC-ledet 2 . . . . .	14
3.4	Print og komponentplacering til knapperne . . . . .	15
4.1	LED styringen i Multisim opstilling . . . . .	17
4.2	Printudlægget til LED-styring . . . . .	18
4.3	Printudlægget til LED-styring med komponenter . . . . .	18
5.1	Diagram over global out kredsen . . . . .	19
5.2	Printlayout og komponentplacering til global out kredsen . . . . .	20
6.1	Hændelsesdiagram for latche . . . . .	22
6.2	Diagram over Latch blokken . . . . .	23
6.3	Print til Latch blokkene . . . . .	23
6.4	Komponentplacering til Latch blokkene . . . . .	24
7.1	Diagrammet for delay-kredsen . . . . .	26
7.2	Oscilloskopstilling . . . . .	27
7.3	Printlayoutet for delay-kredsen, med og uden Copper-area . . . . .	28
7.4	Komponentplacering til delaydelen . . . . .	29
8.1	Print til EPROM . . . . .	32
8.2	Komponentplacering til EPROM . . . . .	33
10.1	Vores første udgangspunkt . . . . .	38
11.1	Arbejdstegninger til chassiet . . . . .	41
11.2	Mærkat til chassiet . . . . .	42
12.1	Princippet illustreret . . . . .	44
12.2	Arbejdstegninger for døråbneren . . . . .	44
12.3	Illustrering af de angivne mål . . . . .	45
13.1	Flowchart for programmets funktion . . . . .	48
18.1	Bilag, Blokdiagram over elektronik delen . . . . .	II

# Tabeller

1.1	Første tidsplan . . . . .	8
6.1	Benforbindelser til Latchblokkene . . . . .	24
8.1	EPROM programmeringstabel . . . . .	31
9.1	Oversigt over LPT portens pins . . . . .	35
9.2	Oversigt over forbindelser mellem EPROM og LPT . . . . .	35
9.3	Oversigt over funktioner i nibble mode . . . . .	36
14.1	Tabel over lyde . . . . .	53
15.1	Oversigt for projektperioden . . . . .	55
15.2	Det totale budget . . . . .	56
15.3	Materialer til produkt . . . . .	57



# Del I

# Projektbeskrivelse

# Kapitel 1

## Projektbeskrivelsen

### 1.1 Gruppen

I projektet må der arbejdes individuelt eller i grupper af enten to eller tre personer. Vi har valgt at danne en gruppe af to, bestående af:

- Jes Toft Kristensen (JTK)
- Michael Hansen (MH)

Bogstaverne i parentes angiver medlemmets initialer, og vil i rapporten blive brugt til at illustrere hvem der har været hovedansvarlig for de forskellige afsnit.

### 1.2 Projektanalyse

Projektoplægget beskriver tre forskellige temaer, indenfor hvilke der er opstillet en problemstilling, som skal løses i projektforsløbet. Oplægget dækker over temaerne: "Handicaphjælpemidler", "Det moderne kontor" og "Overvågning og alarm". Gældende for alle oplæg er, at produktet skal indeholde på elektroniske og mekaniske elementer. Desuden skal der udarbejdes en dokumenterende rapport.

#### 1.2.1 Handicaphjælpemidler

Der er brug for hjælpemidler til handicappede, for at de er i stand til at klare hverdagen. Dette kan være alt ligefra små tilpasninger, til større hjælpekonstruktioner. Produktet kan laves i skalamodel i formindsket størrelse.

#### 1.2.2 Det moderne kontor

Computeren har haft en uovertruffen indflydelse på det moderne kontor, men har samtidig betydet, at der er blevet meget stillesiddende arbejde. Derfor er der kommet fokus på indretning på det moderne kontor, idet der er brug for en række forbedringer, for at højne arbejdsmiljøet.

### 1.2.3 Overvågning og alarm

Brug af overvågning er blevet mere almindeligt, hvadenten det er et fysisk område, eller en proces der overvåges. Ved overvågelse af et område, er det hensigten at forhindre tyveri, mens en procesovervågning er med henblik på at styre, eller kontrollere den givne proces.

## 1.3 Valg af tema

På baggrund af de fremsatte temaer, har vi valgt at arbejde med hjælpemidler til handicappede. Vi har i den forbindelse tænkt på et kommunikationssystem, der gør det muligt for svært handicappede personer, at komme i kontakt med plejepersonale. Vi vil i den forbindelse forsøge at koble forskellige funktioner til dette kommunikationssystem, dvs. en elektronisk og mekanisk udførelse af en hverdagsopgave, såsom at åbne en dør, slukke lyset eller lignende. Det bliver altså på den måde en "Multi-controller" til den handicappede, så han/hun i et bredere omfang kan klare hverdagsopgaven selv.

## 1.4 Ide-fase

Som beskrevet i det ovenstående, har vi som udgangspunkt tænkt på et kommunikationssystem, hvor der skal tilkobles en række dagligdagsfunktioner. Ideen har vi valgt at kalde et "Talkboard", og vil i det efterfølgende blive omtalt således. Uden at have opstillet et blokdiagram for selve funktionen, har vi fremsat illustrationen vist på figur 1.1 af vores ide.



Figur 1.1: Ide til talkboardets udformning

Ideen er at lade den handicappede vælge et ord eller en sætning fra talkboardets menu, der så er tilsluttet en computer, som kan afspille sætningen. Samtidig har vi tænkt på at lade f.eks. åbningen og lukningen af døre, være styret fra talkboardet, så den handicappede har mulighed for at varetage enkelte basale opgaver selv. Åbning og lukning af døre behøver ikke at foregå igennem computeren, men kunne derimod ske igennem en radiosender.

I hele ide-fasen har vi bygget vores løsningsforslag på egne forestillinger om handicappedes behov. Det betyder at vi ikke umiddelbart er klar over hvorvidt der er en målgruppe for produktet, og dermed overhovedet behov for det. Derfor valgte vi at opsøge en person der har haft kontakt med svært handicappede, og derfor har indsigt i de forskellige former for handicap, samt hvilke behov de har. Vi har derfor i den indledende fase i vores projekt, foretaget et interview med Helle Gøtzsche, som har arbejdet med svært handicappede mennesker, som et led i sin uddannelse til sygeplejerske. Dette er beskrevet i det efterfølgende afsnit.

## 1.5 Interview

Formålet med interviewet var til dels at få afdækket om der var en målgruppe for vores ide, og helt præcist hvilke problemer disse har, samt hvordan vi kan afhjælpe problemerne.

Denne sektion er en struktureret gennemgang, af det vi kom frem til i interviewet.

### 1.5.1 Målgruppen

Vi kom frem til at den målgruppe der kunne få meget ud af vores ide, var ældre personer med Apoplexia (blodprop i hjernen), som har medført afasi (besvær med at udtrykke sig). Afasi'en er muligvis en følge af et lammet talecenter, hvilket ofte optræder når en person får en blodprop i en af hjernehalvdelene. I en sådan situation vil den handicappede også ofte være ramt af lammelser, og typisk er det hele den ene side af kroppen der er lammet.

Denne gruppe ville have stort udbytte af et "talkboard", idet dette vil give dem udvidede muligheder for at klare basale ting selv, eller i det mindste formulere de udtryk de vil. På denne måde kan de øge deres selvstændighed, og dermed deres selvværd, hvilket kan være med til at undgå en del depressioner. Problemet er nemlig ofte når folk der hele livet har været aktive, pludselig bliver ramt af svære lammelser. At skulle omstille sig fra at kunne klare alting selv, til pludselig at skulle have hjælp til selv de mest basale ting, er nemlig et meget stort psykisk pres. Derfor er det af stor betydning, at den handicappede får hjælpemidler til at løse de mange basale opgaver selv.

Der rammes årligt 10'000<sup>1</sup> mennesker af Apoplexia. En stor del heraf vil måske kunne få brug for et talkboard, idet følgerne af blodproppen som sagt ofte er i form af lammelser i kroppen, og i talecenteret.

<sup>1</sup>Kilde: Basisbog i Medicin og Kireugi

### 1.5.2 Funktioner

De funktioner vi snakkede om, var primært at kunne udtrykke ord. Men funktioner som dør- og vinduesåbning er også en ting der ville afhjælpe mange handicappede. Derudover ville en radiosender med ”mindre” alarmer (jeg er tørstig eller, jeg vil gerne have en bog...) også være meget handy, idet man så kunne tilkalde personale der ikke er til stede i rummet.

### 1.5.3 Formen

Den form vi præsenterede var figur 1.1. Denne kunne monteres på elektriske kørestole eller lignende, alt efter behov. Det primære med formen er, at det ikke skal være for stort, men samtidig skal knapperne også være til at bruge, hvis man er motorisk besværet.

### 1.5.4 Konklusion på interviewet

Interviewet bekræftede os i, at der var brug for vores produkt. Der ”massefremstilles” ikke noget lignende system, men der findes systemer der tilpasses den individuelle. Fordelen ved vores produkt ville derfor være at det vil være billigt, og nemt at bruge. Idet mange hvert år rammes af blodpropper i hjernen, er der også et potentielt marked for produktet. Især fordi produktets enkelthed gør det nemmere at anvende for f.eks. ældre der kan have besvær med at sætte sig ind i komplicerede hjælpe-systemer.

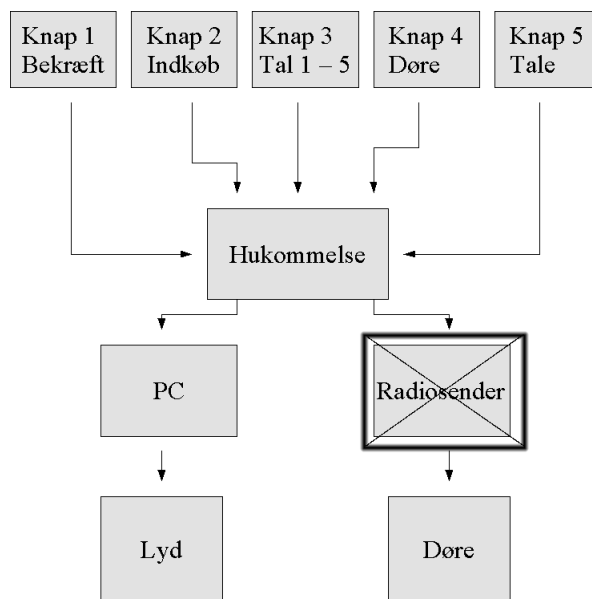
## 1.6 Problemformulering

Hvert år rammes 10'000 personer mellem 65-74 år, af enten en blodprop eller en hjerneblodning, der medfølger svære lammelser. Disse lammelser medfører ofte talebesvær, og i det hele taget besvær med at varetage daglige opgaver. For personer der hele livet har været aktive, er det et stort psykisk pres, lige pludselig ikke at være i stand til at ordne selv de mindste ting i hverdagen. Specielt en manglende evne til at tale med andre mennesker, er et problem for sådanne handicappede.

Vi vil derfor i vores projekt, udarbejde et produkt der kan foretage en kommunikation mellem den handicappede, og andre personer. Selve kommunikationen skal styres via en letbetjent kontrolpanel. Kontrolpanelet skal være udformet så det er muligt også at styre andre funktioner. Med disse funktioner har vi især tænkt på styring af døre, med hensyn til åbning og lukning, samt styring af lyset.

Vi har hertil overvejet at lade de ”ekstra” funktioner som døråbningen og lyset, være styret via en radiosender. Selve kommunikationsenheden, dvs. den del af produktet der skal afspille den ønskede lyd, vil formodentlig tage form af en computer med lydkort. For at det sidst nævnte kan lade sig gøre, må vi tilpasse produktet så det er i stand til at kommunikere med en computer, evt. igennem en parallel port.

For at kunne få vores produkt til at kommunikere med computeren, er vi desuden nødt til at skrive et program til computeren, som kan få computeren til at udfører de ønskede kommandoer, udfra signalerne fra vort produkt.



Figur 1.2: Vores nuværende ide som blokdiagram

Den mekaniske del vil hovedsaglig bestå af døråbneren, og chassiset til Talkboardet. Hvordan vi helt nøjagtigt skal have udformet konstruktionen af døråbneren, har vi på nuværende tidspunkt ikke fastlagt, men vi vil om muligt lade den være drevet af en elmotor, så vi nemt kan styre den via elektronikken.

Opgaverne må derfor kort beskrevet være at:

- Præcist definere de funktioner som vores produkt skal have.
- Tilpasse vores kommunikationsenhed så den kommunikerer med computeren, og styre de ønskede funktioner.
- Have udarbejdet et program til computeren, så den kan samarbejde med vores produkt.
- Udarbejde en konstruktion til den mekaniske døråbner, så den kan styres via vores produkt.

Vi har illustreret vores ide i det overordnede blokdiagram på figur 1.2

## 1.7 Afgrænsning

Til projektet hører der sig nogle begrænsninger som vi er nødt til at tage højde for i projektet. Der er for det første de økonomiske begrænsninger, dvs. i vores tilfælde et budget på 1000 kr, som skal overholdes. Denne begrænsning stiller krav til vores valg af komponenter, hvor vi er nødt til at medtage et økonomisk aspekt med hensyn til valget. Dog er det ikke nødvendigt at foretage nogen umiddelbart afgrænsning, idet komponenterne til vores nuværende ide, ikke vil overskride budgettet.

Desuden er der visse lovkrav der gør sig gældende i forbindelse med radiosenderen, hvilket besværliggør fremstillingen af denne. Samtidig er der nogle tidsmæssige begrænsninger vi også skal tage højde for, hvilket har været årsag til, at vi har valgt ikke at fremstille radiosenderen og modtageren i praksis. Dette fravalg er også gjort på baggrund af, at det resterende produkt fint kan vise den ønskede funktion, uden denne blok er fremstillet.

Som udgangspunkt har vi defineret, at en computer (PC) skal stå for at styre lyden. Derfor vil vi gerne have en PC stillet til rådighed. Denne skal have en fungerende parallel port, et lydkort og et *ikke-NT* styresystem (windows 95 eller 98 er fint)<sup>2</sup>.

Vil vil på nuværende tidspunkt ikke fastlægge flere afgrænsninger, idet vi med skolens ressourcer i form af materialer, og værksteder, burde være udemærket i stand til at realisere vores nuværende ide.

## 1.8 Tidsplan

Vores tidsplan for projektet kan ses på tabel 1.1 side 8.

---

<sup>2</sup>Windows 2000 og NT blokerer for direkte adgang til parallel porten

Uge	Dag	Dato	Lekt.	Aktivitet	Bemærkning
9	Mandag	24-02	8	Projektbeskrivelse færdig	Ingen
9	Fredag	28-02	4	EPROM-kredsen, teoretisk.	Afl. forløbig projektbeskrivelse
10	Mandag	03-03	8	Rette projektbeskrivelse. ”Markedsundersøgelse”	Ingen
10	Fredag	07-03	4	EPROM + Pc.	Ingen
11	Mandag	10-03	8	EPROM + Pc.	Ingen
11	Fredag	14-03	4	EPROM + Pc. Finpudse projektbeskrivelse	Afl. endelig projektbeskrivelse
12	Mandag	17-03	8	EPROM + Pc.	Ingen
12	Fredag	21-03	4	EPROM + Pc kreds færdig	EPROM + Pc kreds færdig
13	Mandag	24-03	8	LED-styring - teori	Ingen
13	Fredag	28-03	4	LED-styring - teori	Ingen
14	Mandag	31-03	8	LED-styring - udarbejdning	LED-styring færdig
14	Fredag	04-04	4	Styring til døråbner	Ingen
15	Mandag	07-04	8	Styring til døråbner	Ingen
15	Fredag	11-04	4	Styring til døråbner	Styring færdig
16	Alle	-	-	Påskeferie. Arbejde på døråbningen	Døråbning færdig
17	Tirsdag	22-04	8	Chassis + rapportskrivning	Ønske om værksted
17	Onsdag	23-04	8	Chassis + rapportskrivning	Ønske om værksted
17	Torsdag	24-04	8	”Buffer” (sikkerhed) Rapportskrivning	Chassis færdig Ingen
17	Fredag	25-04	8	”Buffer” (sikkerhed) Rapportskrivning	Produkt færdig
18	Mandag	28-04	4	”Buffer”(sikkerhed)	

Tabel 1.1: Første tidsplan



# Del II

# Elektronikdelen

## Kapitel 2

# Introduktion

### 2.1 Formålet med elektronikken

Formålet med elektronikdelen er at reagere på brugerens input, for så at sørge for, at computeren får besked på at afspille en lyd. Idet inputtet sker af to omgange, får vi en seriel kommunikation, som vi skal have omsat til parallel information, for at kunne anvende computerens parallel port. Dette sker ved at skrive informationerne ind i flere latche, som vi så kan aflæse, som om de var parallelle. En EPROM sørger for at "oversætte" dataene fra latchene til data på parallelporten.

### 2.2 Den overordnede bløde gennemgang

Hvis vi betragter blokdiagrammet (figur 18.1) på side II, ser vi at det hele starter med nogen knapper. Fra disse knapper går outputtene 3 veje.

Den første vej går til latchene, som kommer til at fungere som en slags lager. Disse lagre er seriekoblede på en sådan måde, at første bølge af information skrives i Latch 1. Når den næste bølge kommer, læses informationen fra Latch 1 over i Latch 2, imens fyldes den første latch med nye data. Disse to sæt data kan vi så koble ind på en EPROM, som så omsætter dem til det parallel porten vil have.

Den anden vej ud af knap-blokken går først til en "global out" blok. Ideen er at denne datavej skal bruges som en form for impulsgenerator, som vi kan trigge latchene med. Derfor skal der ske en trigning, ligemeget hvilken knap der trykkes på. Global out blokken sender altså et signal videre, så snart der er trykket på en vilkårlig knap. Dette signal går videre til delay-delene, der skal sørge for at vores data bliver opfanget på den rigtige måde. Helt eksakt skal de time det, så Latch 1 læser senere end Latch 2, således at Latch 2 opfanger de korrekte data.

Den sidste vej ud af knap-blokken er til nogen indikator LEDer, der fortæller brugeren hvilken knap denne har trykket på tidligere, altså en ren service funktion.

## Benævnelser mm.

Fortløbende har komponenter i diagrammer fået et unikt navn. Dette fik komponenterne i print-tegningerne ikke, da vi fremstillede printene. De har derimod fået mere meningsfulde navne. Vi har valgt at gengive printene som de er. Dog vil navnene i komponentplaceringen svare til de navne der er brugt i diagrammerne.

Idet vi er blevet nødsaget til at behandle printene som billeder er de blevet skaleret, og er derfor ikke til deres originale størrelse. De rigtige print-tegninger er vedlagt på diskette, enten i AutoCAD R14 .dwg formatet, eller i Ultiboards format.

Forsyningen til al elektronikken er på 5 volt.

## Kapitel 3

# Knapper

Knappernes funktion er at sende brugerens valg videre, til de andre blokke i systemet. Det primære i forbindelse med knapperne er at tage hensyn til prel og, idet vi ville bruge dem til at teste resten af systemet i projektet, at der er pull-down modstande på. Vi skal bruge i alt fem knapper.

### 3.1 Opbygning/beregning

Knapperne er ikke af den bedste kvalitet<sup>1</sup>, hvorfor vi testede dem for prel, men det var svært at se noget på oscilloskopet (det går jo meget hurtigt). Derfor indsatte vi et RC-led, for at modvirke den prel der måske var der. Opladetiden for dette skal ikke være særlig stor, men da der samtidig skulle pull-down modstande på, ville det blive lidt et tilpasningsarbejde. Vi endte med et diagram, som på figur 3.1.

Her vil opladetiden tau ( $\tau$ ) jo opfylde  $RC = \tau$ , men samtidig bliver der en spændingsdeler over modstandene  $R_{K1}$  og  $R_{K2}$ . Vi besluttede at et rimeligt niveau for spændingsdeleren (med udgangspunkt på 5 volt), ville være at den opnåede minimum 4,7 volt, således at der faldt 0,3 volt over  $R_{k1}$ . Altså kan vi beregne størrelsesforholdet mellem modstandene.

$$\frac{R_{k1}}{R_{k2}} = \frac{U_{k1}}{U_{k2}} = \frac{0,3V}{4,7V} = 0,064 \quad (3.1)$$

$$\begin{aligned} & \Downarrow \\ \frac{R_{k1}}{0,064} &= R_{k2} \end{aligned} \quad (3.2)$$

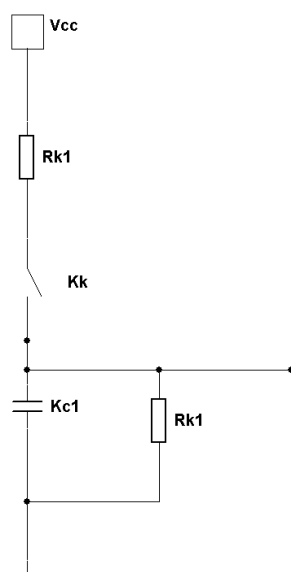
Vi har valgt en tidskonstant på 2ms, og kan således bestemme vores modstandsværdi, når vi sætter kondensatorens værdi til 680nF (det var hvad vi lige havde), og vælger fra E12 rækken.

$$\tau = R_{k1} \cdot C_k \quad (3.3)$$

$$\frac{\tau}{C_k} = R_{k1} \quad (3.4)$$

$$\frac{2ms}{680nF} \approx 2K7\Omega \quad (3.5)$$

<sup>1</sup>Mere om valget af knapper på side 56



Figur 3.1: Diagram over knapperne

Herefter bestemmes  $R_{k2}$  ud fra sætning (3.2), hvor vi så vælger fra E12 rækken.

$$R_{k2} = \frac{2K7\Omega}{0,064} \approx 47K\Omega \quad (3.6)$$

Altså har vi fundet følgende data, hvilket vi kan regne videre på

$$R_{k1} = 2K7\Omega \quad (3.7)$$

$$R_{k2} = 47K\Omega \quad (3.8)$$

$$C_k = 680nF \quad (3.9)$$

$$\tau = R_k \cdot C_k = 1,8ms \quad (3.10)$$

$$U_{Rk1} = 0,271V \quad (3.11)$$

$$U_{Rk2} = 4,72V \quad (3.12)$$

På figur 3.2 ses den første  $\tau$  simuleret i multisim (her måles spændingen over kondensatoren vha. et oscilloskob).

Vi kan nu regne på, hvor lang tid der vil gå, før vores udgange på knap-printet opnår 4,5 volt, som er der hvor gatene videre i systemet med sikkerhed går høje. Vi bruger forskriften for et RC-leds opladning, hvori vi kan isolerer t.  $U_C$  er det totalt opnåelige potentialetab over  $R_{k2}$  (og dermed kondensatoren), imens U er spændingen på 4,5V. Modstanden R, er så  $R_{k1}$ , imens C er kondensatorens værdi på 680nF.

$$U_C = U \cdot \left(1 - \frac{1}{e^{\frac{t}{RC}}}\right) \quad (3.13)$$

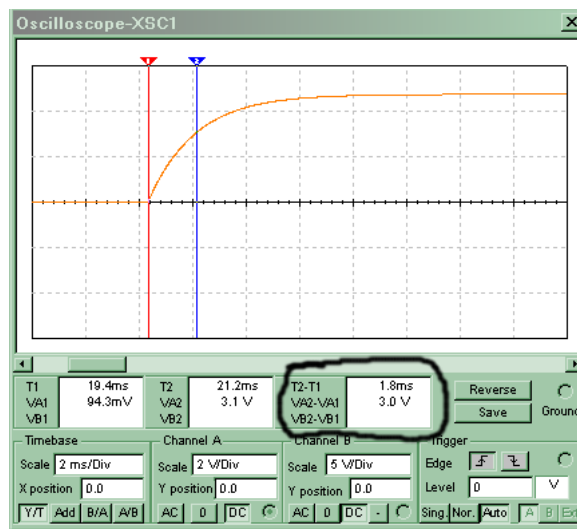
$$\Downarrow \quad (3.14)$$

$$t = -RC \cdot \ln\left(1 - \frac{U_C}{U}\right) \quad (3.15)$$

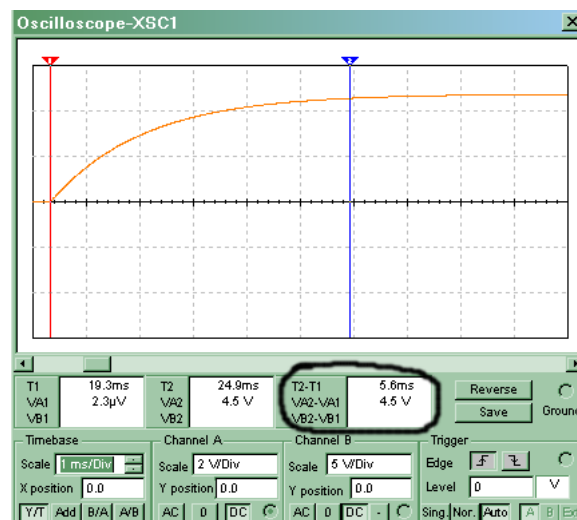
$$t = -2K7\Omega \cdot 680nF \cdot \ln\left(1 - \frac{4,5V}{4,72V}\right) \quad (3.16)$$

$$t = 5,6ms \quad (3.17)$$

Dette har vi også simuleret i multisim (vha. et oscilloskop der måler spændingen over  $C_k$ ), dette ses på figur 3.3, hvor det ser ud til at passe ret godt.



Figur 3.2: Oscilloskopsimulering af RC-ledet 1

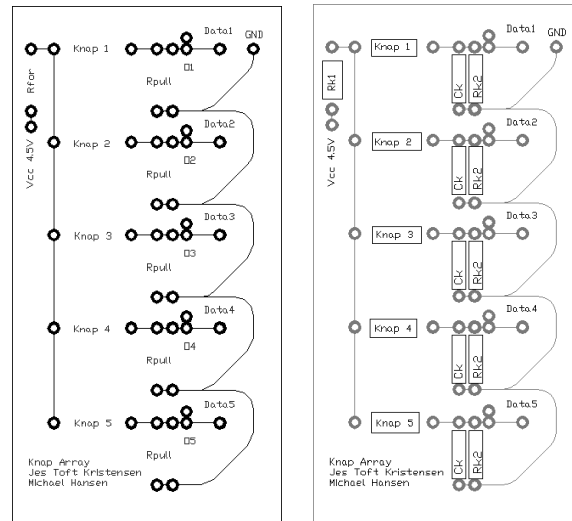


Figur 3.3: Oscilloskopsimulering af RC-ledet 2

## 3.2 Print til knap-delen

Med disse informationer var det bare at udtænke en smart arrangering, og tegne et print med et passende antal udgang. Desuden kan vi nøjes med at have en formodstand, så længe vi kun trykker en knap ned ad gangen. Dette er på figur 3.4.

Komponentplaceringen ses til højre på figur 3.4, set fra printsiden.



Figur 3.4: Print og komponentplacering til knapperne, begge fra printsiden

# Kapitel 4

## LED styring

Led styringen har til formål at indikere hvilken knap man har trykket på, ganske enkelt for at lette overblikket over kombinationsmulighederne. Styringen bygger på en række RS flip flops, der settes via knapperne på Talkboardet, og resettes direkte fra LPT porten.

### 4.1 Opbygning

Der er flere måder hvorpå man kunne opbygge denne LED styring. RS flip flops kan på flere måder konstrueres af diverse gates der kobles sammen. Vi så dog ingen grund til at bruge unyttig tid på at konstruere så mange flip flops ved hjælp af gates, idet det er både kræver en del plads, samt en del besvær når printet skal udarbejdes fordi der skal bruges en stor del gates i forskellige pakninger.

Vi valgte derfor den nemme løsning, og brugte en RS latch, der blot er en samling RS flip flops i samme pakning. Typen vi valgte at anvende var en *CD4043BC*, som jo ganske tydeligt er fra 4000 serien. Normalt bør man undlade at anvende både 4000 serien og 74hc i samme kredse, idet de i enkelte tilfælde kan vise sig ikke at virke sammen. I dette tilfælde spiller det dog ingen rolle, idet denne del af kredsen er helt isoleret fra alt det andet logik, og der er derfor ingen steder hvor 4000 serien og 74hc serien er direkte forbundet til hinanden, udover reset, hvilket er uden betydning.

Udgangene på latchene skal som navnet indikere, drive en række LED'er, og der er hertil brug for en række formodstande. En standard LED kan drives ved ca. 1,5V og 20mA. Idet latchen har 5V på udgangen når den er høj, vil formodstanden som følge deraf have en størrelse af:

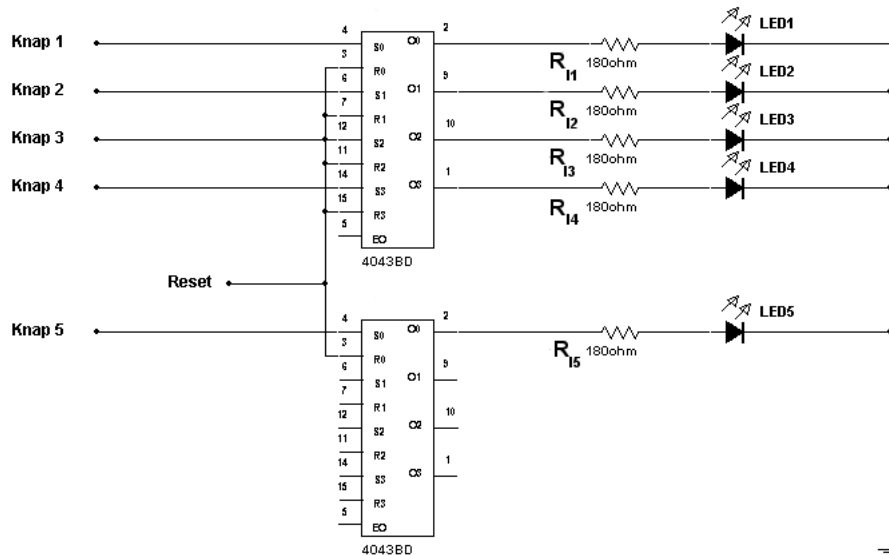
$$U_R = U_{latch} - U_{LED} = 5V - 1,5V = 3,5V \quad (4.1)$$

$$\Downarrow$$

$$R_{l1-l5} = \frac{U_R}{I_{LED}} = 175\Omega \quad (4.2)$$

Omregnet til E-12 rækken bliver det 180  $\Omega$ . Kredsen kunne nu konstrueres, og opbygningen er vist på figur 4.1.





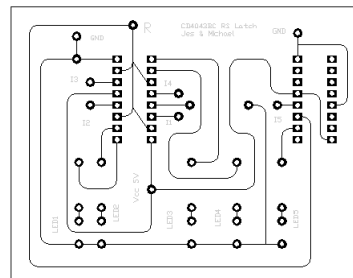
Figur 4.1: LED styringen i Multisim opstilling

Figur 4.1 viser hvordan de 5 knapper er koblet til 5 set-indgange på de to latche, og hvordan samtlige anvendte reset-indgange er koblet sammen, og føres til LPT porten (Pin 2). Ved denne opstilling vil der efter første tryk, være en lysdiode der indikere hvilken knap der er blevet trykket ned. Ved andet tryk vil en anden lysdiode indikere hvilken knap der nu er trykket på (medmindre det selvfølgelig er den samme). Begge LED'er vil nu holdes tændte indtil softwaren i computeren har opfattet inputtene, og efterfølgende resetter Talkboardet, og dermed også latchene, således at LED'erne slukkes.

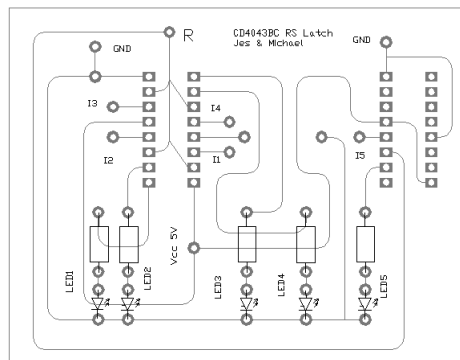
## 4.2 Fremstillingen

Som opstillingen i Multisim indikerer, er der tale om quad pakninger, dvs. fire latche per IC. Vi skulle kun bruge 5, hvorfor vi fik 3 i overskud, der derfor ikke er tilsluttet noget på printet. Printudlægget blev fremstillet i AutoCad, og kan ses på figur 4.2

Komponentplaceringen kan ses på figur 4.3.



Figur 4.2: Printudlægget til LED-styring



Figur 4.3: Printudlægget til LED-styring med komponenter

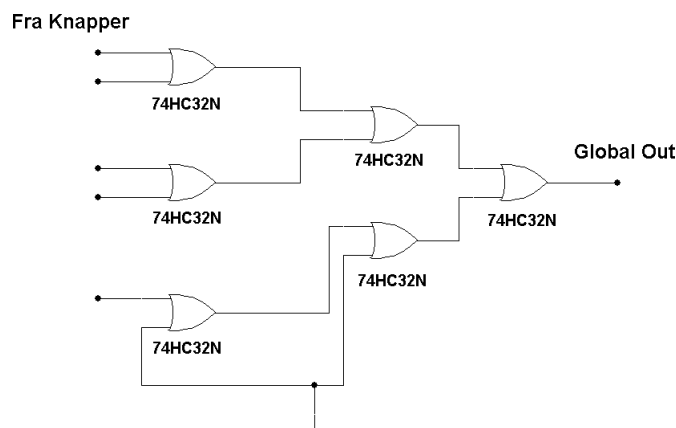
# Kapitel 5

## Global out

Idet vi skal bruge nogen trigger-impulser senere i kredsen, skal vi reagerer på knapperne, uanset hvilken der trykkes på. Til dette bruger vi en opkobling af OR-gates. Denne blok får sine inputs direkte fra knapperne. Denne bloks indgange skal kobles direkte på en af "data n" udgangene, på højresiden af figur 3.4.

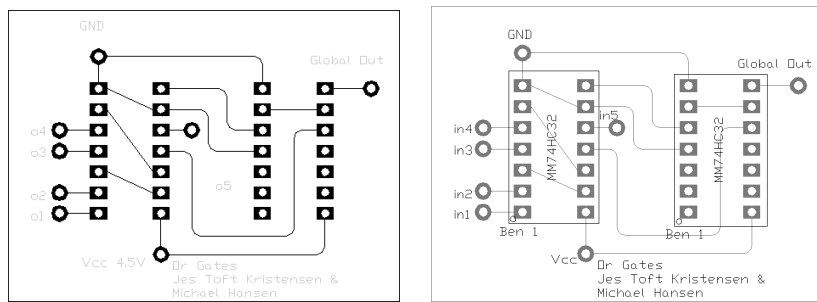
### 5.1 Opbygning

Selve konstruktionen lavede vi ikke noget karnaugh-kort for, idet vi mere bruger en logisk egenskab og opbygning, end et egentligt "sandhedstabelsproblem". Diagrammet for global out kredsen ses på figur 5.1. Grunden til at en ekstra gate er trukket til jord er, at der helst skal være lige stor forsinkelse igennem delen, ellers ville det måske forskyde nogen knapper i forhold til hinanden, hvilket vi gerne vil undgå.



Figur 5.1: Diagram over global out kredsen

Selve printet er til venstre på figur 5.2, imens komponentplaceringen er til højre på samme figur.



Figur 5.2: Printlayout og komponentplacering til global out kredsen, begge fra printsiden

## Kapitel 6

# Latche

Latchene (2 stk.) i denne kreds indgår som sagt i en lagerfunktion, samt i at tilføre et stabilt input til EPROMmen. Latchenes triggerimpuls får de fra delay- og global out delene. Latchenes clear styres via parallel porten.

### 6.1 Funktion

For at gøre det mere overskueligt har vi lavet et ”hændelsesdiagram”. På figur 6.1 ses de fire trin, som processen sker i. Her ses de fem databanke/Flip-flops vi bruger i hver latch. Disse er koblet op, så latch 1 læser på inputtet fra knapperne (kaldet data A og B, se eventuelt diagrammet på figur 6.2) når den får en clock impuls fra delay delen. Latch 2 læser på latch 1, når denne får en clock impuls fra global out-delen. Altså triggles latch 2 før latch 1.

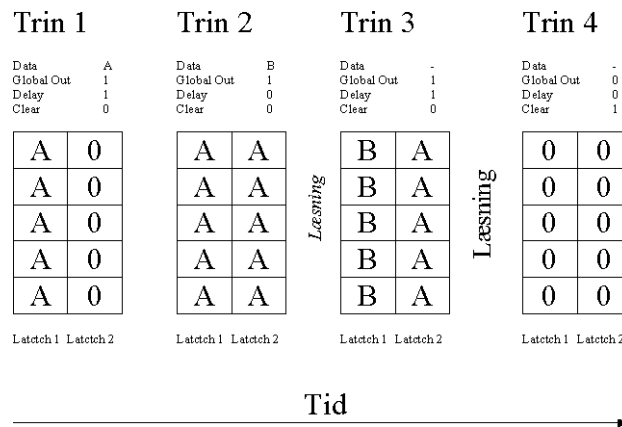
I trin 1 har latch 1 læst dataene ind, idet den får en clock impuls fra delay delen. Før trin 1 har latch 2 også læst på latch 1, men her står der jo 0 over hele linien, derfor er dette trin udeladt. Altså har latch 1 sine databanke fyldt, imens latch 2 er helt tom.

Når så der trykkes anden gang kommer vi til trin 2. Idet der trykkes på en knap går udgangen fra global out høj. Dette får latch 2 til at læse, hvorfor den læser datasæt A ind i sine databanke. Efter et stykke tid går delay-delens udgang også høj, og latch 1 læser datasæt B ind (datasæt B stammer fra det andet tryk). Efter trin 3 er det så meningen at EPROMmen skal læse værdierne ud af latchene, hvilket den så gør, og sender et clear signal tilbage til latchene. Dette bringer os til trin 4, hvor cyklussen kan begynde forfra.

En ting vi har opdaget, idet vi har stillet tingene op som på figur 6.1, er at der vil ske en udlæsning efter trin 2, hvilket er en fejl. Dette medfører at EPROMmen læser AA ind, i stedet for AB (som vi havde tiltænkt den). Dette vil vi dog diskutere på side 57.

### 6.2 Om latchene

Den type latch vi bruger er en MM74HC273 med fælles clear og clock (kanttrigget). En ting at bemærke er at clear skal køres lav, for at cleare latchen. Dvs. at den som standard skal være trukket høj, for at inputs skal have nogen effekt.



Figur 6.1: Hændelsesdiagram for latche

Idet clear styres fra parallelporten, som sættes høj når der skal cleares, skal vi også bruge en inverter.

En anden ting man skal beregne med er at dataene skal være på latchen et stykke tid før den kan læses ind. Dette tidsrum er i datasheetene anført til typisk at være på  $t_s = 20ns$  (Data to clock)<sup>1</sup>. For at vi kan være sikre på at vores latche fungerer optimalt skal deres clock-impuls altså forsinkes.

Idet vi i starten af projektet havde en anden forestilling om hvordan kredsen skulle laves, end vi har nu, har vi brugt 4 NOT-gates til at skabe denne forsinkelse. I NOT-gatenes datasheets står der at deres Maximum propagation delay typisk er på  $t_{PHL/PLH} = 11ns$ , det er den tid det tager for data "at fise igennem gaten"<sup>2</sup>. Altså får vi med 4 NOT-gates en forsinkelse på 44ns, hvilket bringer os helt på den sikre side. Det skal bemærkes at dette delay kun ville være nødvendigt på den første latch, idet dataene står på latch 2 i det tidsrum der er mellem første og andet tryk, hvilket vi ikke i vores vildeste fantasi kunne forestille os faldt under de 20ns ( $t_s$ ).

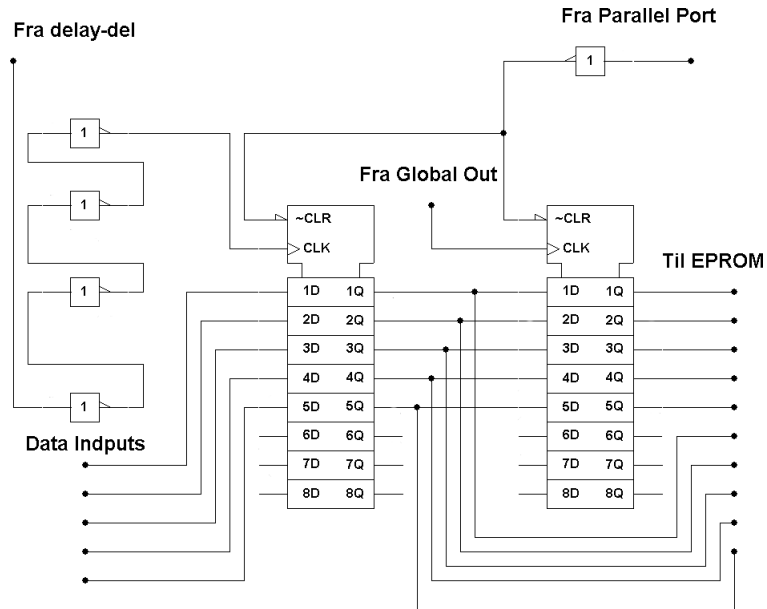
### 6.3 Opbygning

Denne blok består af to print. Det første print har en latch og en inverter, imens det andet print kun har en latch. Grunden til denne opdeling er, at vi ikke var opmærksomme på at vi skulle bruge 2 latche, for at få vores ønskede effekt. I starten har vi simpelthen kun regnet med at bruge 1 latch (vi troede at den havde en serie databanke i sig selv), derfor måtte vi lave et ekstra print, til den ekstra latch. I alt ser det færdig diagram ud som på figur 6.2.

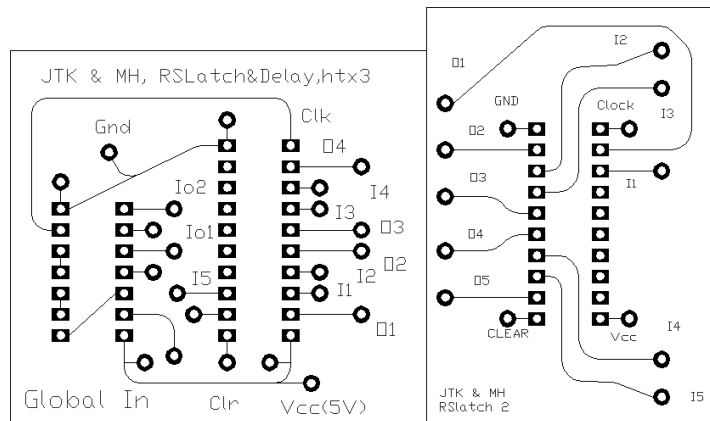
Printet ses på figur 6.3, hvor komponentplaceringen er på figur 6.4, set fra printsiden. Det første print vi lavede er det til venstre (kaldet RSLatch & Delay). På printet er der en del forskellige ud- og indgange, hvis forbindelser er anført i tabel 6.1. Her er dog kun anført de forbindelser man selv skal lave, i en eventuel produktionssituation.

<sup>1</sup>Se bilagsmappen, datasheet for MM74HC273 side 4, Minimum Setup Time

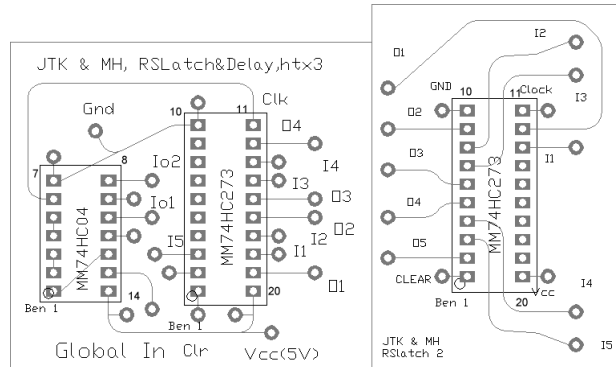
<sup>2</sup>Se bilagsmappen, datasheet for MM74HC04, side 3, Maximum Propagation Delay



Figur 6.2: Diagram over Latch blokken



Figur 6.3: Print til Latch blokkene



Figur 6.4: Komponentplacering til Latch blokkene, begge fra printsiden

IC/Ben	Print/forbindelse	IC/Ben	Print/forbindelse
74HC04	Latch 1 (inverter)	74HC273	Latch 1
14	Støjkondensator	1 (Clear)	fra 74HC04 (inverter) ben 10
7	Støjkondensator	2	Output 5, til EPROM A09 og latch 2 ben 3 (I5)
8 (Io2)	Til Delay-delen	3 (I5)	Input 5, fra Data 5 (knapper)
9	Fra global out	12 (O4)	Output 4, til EPROM A08 og latch 2 ben 4 (I4)
10	Til clear på de to MM74HC273	13 (I4)	Input 4, fra Data 4 (knapper)
11 (Io2)	Fra Parallelport (pin 2) (Clear styring)	14 (I3)	Input 3, fra Data 3 (knapper)
		15 (O3)	Output 3, til EPROM A07 og latch 2 ben 7 (I3)
		16 (O2)	Output 2, til EPROM A06 og latch 2 ben 8 (I2)
		17 (I2)	Input 2, fra Data 2 (knapper)
		18 (I1)	Input 1, fra Data 1 (knapper)
		19 (O1)	Output 1, til EPROM A05 og latch 2 ben 13 (I1)
		74HC273	Latch 2
		1 (Clear)	fra 74HC04 (inverter) ben 10
		2 (O5)	Til EPROM A04
		3 (I5)	Input 5, fra latch 1 ben 2 (O5)
		4 (I4)	Input 4, fra latch 1 ben 12 (O4)
		5 (O4)	Output 4, til EPROM A03
		6 (O3)	Output 3, til EPROM A02
		7 (I3)	Input 3, fra latch 1 ben 15 (O3)
		8 (I2)	Input 2, fra latch 1 ben 16 (O2)
		9 (O2)	Output 2, til EPROM A01
		11 (Clock)	Fra global out
		12 (O1)	Output 1, til EPROM A00
		13 (I1)	Input 1, fra latch 1 ben 19 (O1)

Tabel 6.1: Benforbindelser til Latchblokkene



## Kapitel 7

# Delay-delen

Idet adressen til EPROM'en sammensættes af to omgange (første og andet tryk), er det nødvendigt med et delay, der forsinker signalet, sådan at signalet kommer rigtigt igennem kredsen. Til dette anvender vi en opkobling af to timerkredse af typen lm555.

### 7.1 Ide/teori

Delaykredsens funktion er at forsinke trigger signalet til latch 1, således at der ikke står noget på udgangen af denne når latch 2 læser ind. Næste gang der så bliver trykket, vil latch 2 igen læse ind først, hvor signalet fra sidste indtryk nu står på indgangen. Forsinkelsen skal altså være den tid det tager for latch 2 at læse det der står på indgangen, idet den skal være færdig med dette, inden udgangen på latch 1 ændres.

Latchen vi anvender (MM74HC273<sup>1</sup>) skal "kant-trigges" dvs. at den trigges ved ændringen fra lav til høj (transition from low to high level). Derfor skal forsinkelsen i teorien ikke være meget længere end den tid dataerne skal være på indgangen inden latchen bliver trigget. I tilfældet med MM74HC273 er denne tid (Minimum Setup Time, Clock to Data) garanteret til at være 20 ns under normale forhold (25° og ved 4,5V).

En sådan forsinkelse kunne i teorien opnåes ved at sammenkoble en række invertere, og bruge deres reaktionsforsinkelse, til at frembringe det ønskede delay.

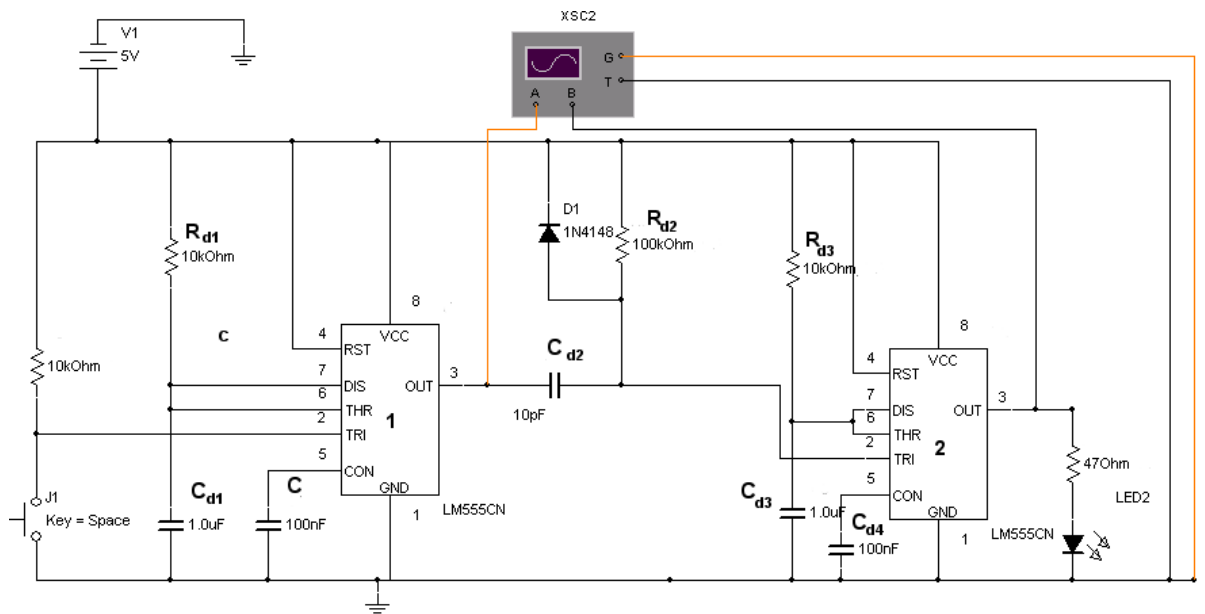
Men teori og praksis er to vidt forskellige ting, og i så små tidsstørrelser er det umuligt at foretage nogen praktiske målinger, i en eventuel fejlfindingssituation. Da vi samtidig ikke mente at en yderlig forsinkelse ville have nogen indflydelse på produktets funktion, valgte vi at bruge en opkobling med to lm555 der er beregnet til noget større tidsforsinkelser (fra  $\mu s$  og opefter).

Konstruktionen består som sagt af to lm555<sup>2</sup> der er koblet sammen, således at forsinkelsen og længden af det forsinkede signal, kan moduleres. Diagrammet fra multisim er vist på figur 7.1.

Det skal hertil nævnes at kredsen ikke blev trigget ved en kontakt som vist, men derimod fra vores "global out" der blev kørt igennem en inverter, idet lm555 trigges ved at spændningen på ben 2 bliver mindre end 1/3 af forsyning,

<sup>1</sup>se evt. datablad for MM74HC273 i bilagsmappen

<sup>2</sup>Se evt. datablad for LM 555 i bilagsmappen



Figur 7.1: Diagrammet for delay-kredsen

dvs. ca. 1,67V. Derudover var der ikke koblet en LED op som vist. Denne var blot med det formål at lette simuleringen.

Kredsen minder på mange måder om to monostabile multivibratorer der er koblet sammen, hvilket det til dels også er. Den første lm555 er koblet op nøjagtigt som man normalt ville opkoble en monostabil multivibrator. Udgangen er så koblet til indgangen på den anden lm555 via en kondensator. Det betyder, at når den første lm555 bliver trigget, vil udgangen gå høj som havde man brugt den som monostabil multivibrator.

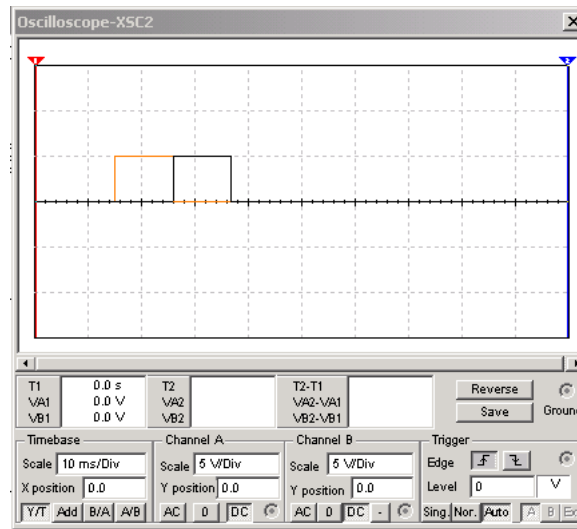
Så snart udgangen på den første lm555'er går lav igen, vil den trigge den anden lm555'er, der så går høj i den forudindstillede tidsperiode.

Herved kan vi justere signalets forsinkelse, samt længde, ved at ændre på de modstande og kondensatorer som styrer timingen af de to lm555. Dvs. at for at ændre forsinkelsen, skal størrelsen af modstanden ( $R_{d1}$ ) og kondensatoren ( $C_1$ ) der sidder i forbindelse med den første lm555 (mellem forsyning og stel, med forbindelse til pin 7 og 6). Tiden er bestemt ud fra formlen:

$$t = 1,1 \cdot R \cdot C \quad (7.1)$$

På samme måde med længden af signalet. Her er det blot modstanden ( $R_{d3}$ ) og kondensatoren ( $C_3$ ) i forbindelse med den anden lm555 der skal tilpasses.

Vi testede den opstillede kreds i Multisim, med et oscilloskop indsat vist på figur 7.1), således at den måler på udgangen af både den første og anden timer. Dermed illustreres funktionen tydeligt, nemlig at når den første timer har kørt det indstillede tidsinterval, aktiveres den næste. Simuleringen er grafisk illustreret ved hjælp af et oscilloskop, vist på figur 7.2. Den orange linie viser altså forsinkelsen, mens den sorte viser det der egentlig står på kredsens udgang.



Figur 7.2: Oscilloskopopstilling

For at tilpasse delayet skulle  $R_{d1}$  og  $C_{d1}$  og altså skaleres. Den nemmeste måde at gøre dette på er at finde en tilpas kondensator først, og derefter indpasse modstanden herefter. Vi valgte derfor at gøre brug af  $2\ 1\mu F$ , for henholdsvis  $C_{d1}$  og  $C_{d3}$ .

Problemet var, at vi ikke var afklaret med nøjagtigt hvor lang en forsinkelse vi havde brug for. Vi skulle være sikre på at latch 2 havde læst igennem inden latch 1 ændrer på udgangen. Forsinkelsen burde ikke være længere end 100 ns, men for at være sikker, valgte vi at indsætte et potentiometer på  $47k\Omega$ , så vi kunne indstille den fra timerkredsens mindste forsinkelse og indtil en forsinkelse givet ved:

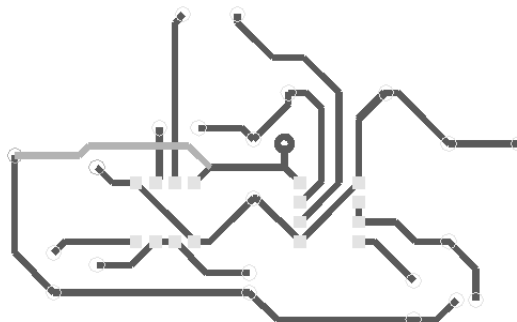
$$t = 1,1 \cdot R_{d1} \cdot C_{d1} \quad (7.2)$$

$$t = 1,1 \cdot 47 \cdot 10^3 \Omega \cdot 1 \cdot 10^{-6} F = 0,0517s \approx 52ms \quad (7.3)$$

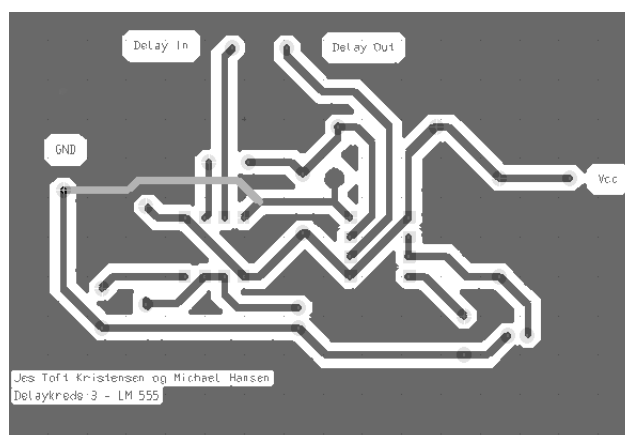
Længden af impulsen kunne som sagt også moduleres ved at ændre på  $R_{d3}$  og  $C_3$ , hvilket vi dog ikke havde behov for. Latchene var jo kant-triggeret, og længden af impulsen skulle derfor, ifølge databladet, have en længde af blot 16 ns under normale forhold (stadig  $25^\circ$ ). For god ordens skyld indsatte vi en  $10k\Omega$  modstand, hvorved trigger impulsen fik en størrelse af:

$$t = 1,1 \cdot R_{d1} \cdot C_{d1} \quad (7.4)$$

$$t = 1,1 \cdot 10 \cdot 10^3 \Omega \cdot 1 \cdot 10^{-6} F = 0,011s \quad (7.5)$$



Jes Toft Kristensen og Michael Hansen  
Delaykreds - LM 555



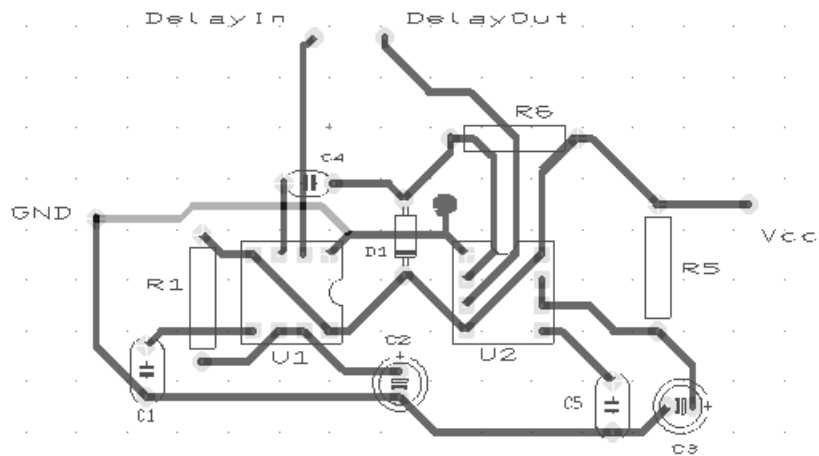
Figur 7.3: Printlayoutet for delay-kredsen, med og uden Copper-area

## 7.2 Udførelse

Når vi nu havde lavet diagrammet, og en succesfuld simulering i Multisim, kunne vi ligeså godt anvende Ultiboard til udarbejdelse af printet. Dette blev det eneste print vi producerede i Ultiboard, men nok også det mest velegnede dertil. Det endelige printlayout kan ses på figurene 7.3. Den lysere streg angiver placeringen af en lus.

Vi valgte at udarbejde vores dette print med såkaldte copperareas, dvs. områder hvor der normalt ikke ville være noget, og hvor man så vælger at lade kobberet blive der, fremfor at ætse det væk. Det har som sådan ingen betydning for kredsen, selvom man i teorien måske kunne sætte stel til disse områder, og anvende dem som en slags isolator for støj og lignende. Det har vi dog ikke gjort, ganske enkelt fordi denne kreds ikke er specielt følsom overfor støj. Eneste begrundelse for at vi har lavet printet med copperareas, er æstetiske årsager (det ser godt ud...!).

Komponentplaceringen ses på figur 7.4.



Jes Toft Kristensen og Michael Hansen  
 Delaykreds 3 - LM 555

Figur 7.4: Komponentplacering til delaydelen

# Kapitel 8

## EPROM'en

Vores EPROM fungerer som en oversætter og et interface til parallelporten. En oversætter idet den tager de 10 input-bits fra latchene, og omsætter dem til 5 bit, som der er plads til på parallelporten. Samtidig passer niveauerne mellem eprom'en og parallelporten sammen, så vi ikke skal tænke over dette. (begge passer på TTL niveauer<sup>1</sup>).

Af krav til eprom'en er der følgende:

- De 10 input-bits skal omsættes til 5 bits
- Første gang der trykkes ned skal parallelporten modtage decimalværdien 88, som er programmets startsignal(se programmeringsafsnittet side 49).
- Skemaet over lyde skal passe sammen med de outputs der gives til porten.

### 8.1 Inputsekvens/programmering

Følgende forklaring tager udgangspunkt i tabel 8.1 på side 31. Følg eksemplet nederst i tabellen.

Her får vi allerførst bitsekvensen "aaaaa" ind, hvor der så sendes et startsignal. Efter dette gives der en bitsekvens "bbbbb". Disse "sættes sammen" idet vores latche giver eprom'en disse data på samme tid. På grund af opkoblingen vil bbbbb gå til de 5 første mest betydningsfulde bits (A09 - A05), imens aaaaa går til adresse A05 - A00. Dette omsætter vi til en hex-adresse, som er det register der vil blive aktiveret af vores input.

Til dette register skal vi så bestemme en outputværdi. Idet vi har 5 bits ind på parallelporten får vi 32 muligheder. Altså har vi decimalværdien 0 - 32 i muligheder. Hver lyd skal selvfølgelig have sin egen decimalværdi ud af eprom'en. Vi tager altså "mulighed" nr. 27 (simpelthen tallet 27) og konverterer denne til en binær værdi. Parallelportens bits starter efter de første 3 cifre, så vi lægger 3 nuller efter. Dette konverterer vi til en hex-værdi, som eprom'en skal sende ud.

Vi mangler så at bestemme hvilke decimalværdier computeren kan læse, når vi sender vores eksempel-værdi til parallelporten. Imidlertid er det vigtigste bit (bit nr. 7, startende fra 0) på parallelporten inverteret. Altså skal vi ændre denne

<sup>1</sup>Se bilagsmappe, datasheet for AM27C256, side 1, general description

værdi, og derefter konvertere den til en decimalværdi, som vores computer kan associere med den korrekte lyd.

Lyd	Grp.	U.Grp.	Adr.	Adr.	Mul	Mul	Prog	Port	Port
Format	Bin	Bin	Bin	Hex	Dec	Bin	Hex	Bin	Dec
Ja	10000	10000	1000010000	0210	1	00001000	0008	10001000	136
Nej	10000	01000	0100010000	0110	2	00010000	0010	10010000	144
Måske	10000	00100	0010010000	0090	3	00011000	0018	10011000	152
Tak	10000	00010	0001010000	0050	4	00100000	0020	10100000	160
Ondt	10000	00001	0000110000	0030	5	00101000	0028	10101000	168
1	01000	10000	1000001000	0208	6	00110000	0030	10110000	176
2	01000	01000	0100001000	0108	7	00111000	0038	10111000	184
3	01000	00100	0010001000	0088	8	01000000	0040	11000000	192
4	01000	00010	0001001000	0048	9	01001000	0048	11001000	200
5	01000	00001	0000101000	0028	10	01010000	0050	11010000	208
Bede om	00100	10000	1000000100	0204	11	01011000	0058	11011000	216
På toilet	00100	01000	0100000100	0104	12	01100000	0060	11100000	224
Åbn dør 1	00100	00100	0010000100	0084	13	01101000	0068	11101000	232
Åbn vind 1	00100	00010	0001000100	0044	14	01110000	0070	11110000	240
Åbn dør 2	00100	00001	0000100100	0024	15	01111000	0078	11111000	248
Franskbrød	00010	10000	1000000010	0202	16	10000000	0080	00000000	0
Rugbrød	00010	01000	0100000010	0102	17	10001000	0088	00001000	8
Kød	00010	00100	0010000010	0082	18	10010000	0090	00010000	16
Grønt	00010	00010	0001000010	0042	19	10011000	0098	00011000	24
Kartofler	00010	00001	0000100010	0022	20	10100000	00A0	00100000	32
Læsestof	00001	10000	1000000001	0201	21	10101000	00A8	00101000	40
Fjernsyn	00001	01000	0100000001	0101	22	10110000	00B0	00110000	48
Computer	00001	00100	0010000001	0081	23	10111000	00B8	00111000	56
Seng	00001	00010	0001000001	0041	24	11000000	00C0	01000000	64
Kørestol	00001	00001	0000100001	0021	25	11001000	00C8	01001000	72
Startsignal	10000	-	1000000000	0200	26	11011000	00D8	01011000	88
Startsignal	01000	-	0100000000	0100	26	11011000	00D8	01011000	88
Startsignal	00100	-	0010000000	0080	26	11011000	00D8	01011000	88
Startsignal	00010	-	0001000000	0040	26	11011000	00D8	01011000	88
Startsignal	00001	-	0000100000	0020	26	11011000	00D8	01011000	88
Eksempel	aaaaa	bbbb	bbbbbaaaa	ADR.	27	11110xxx	F0	01110xxx	112

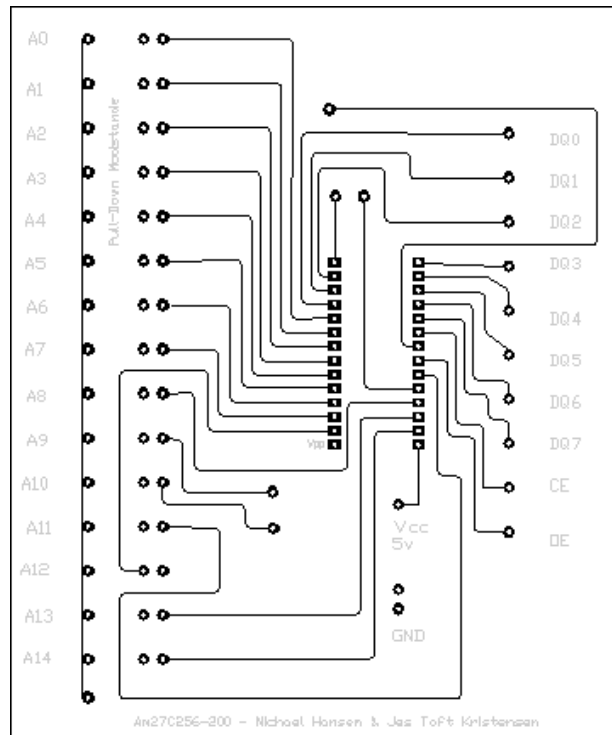
Tabel 8.1: EPROM programmeringstabel

Det vi rent faktisk skal programmere eprom'en efter, er dataene i "Adr" og "Prog" kolonnerne. "Adr" er den adresse der skal ændres, imens "Prog" er den værdi som adressen/registret skal ændres til. Disse to kolonner er derfor også indrammet med en dobbelt streg til hver side.

## 8.2 Fremstilling

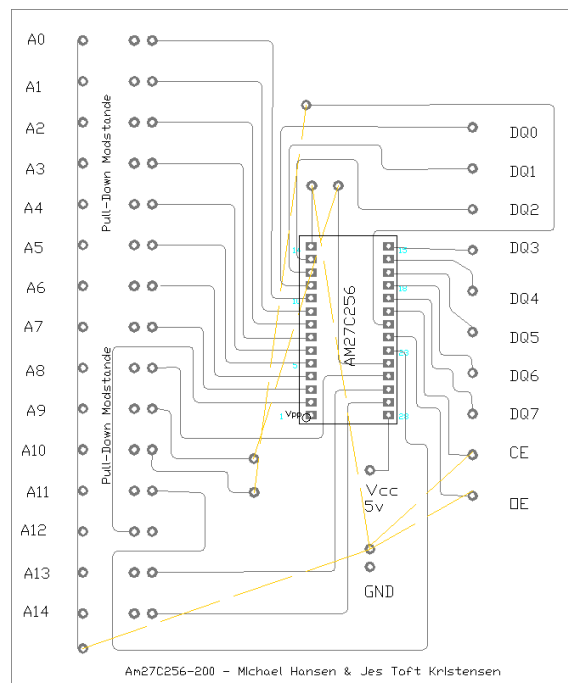
Det print vi fremstillede til eprom'en var originalt tiltænkt en eksperimenterende rolle. Altså regnede vi med at skulle teste os til en del viden om eprom'en. Derfor er printet meget stort, idet alle ud- og indgange er trukket ud i periferien, så de er til at håndtere. Printet blev tegnet i AutoCAD, hvilket har givet en mindre fejl, idet nogen loddeøer blev skaleret ned, så de ikke længere overlappede hinanden.

Vi måtte altså bore ekstra huller til at sætte pull-down modstandene på. Printet er på figur 8.1. Komponentplaceringen er vist på figur 8.2. De stiplede linier er lus der skal sættes ind.



Figur 8.1: Print til EPROM





Figur 8.2: Komponentplacering til EPROM, fra printsiden

## Kapitel 9

# Parallelporten

Kommunikationen mellem EPROM'en og parallelporten var en af de mest grundlæggende problemstillinger i projektet, og afgørende for om produktet i det hele taget ville virke. EPROM'en vi valgte at gøre brug af, var af typen AM27C256 og gør brug af CMOS (Complementary Metal Oxide Semiconductor) teknologi. Parallelporten derimod gør brug af det ældre TTL (transistor-transistor logic) system, og vores første bekymring var om de ville fungere sammen, idet niveauerne for høj og lav, ikke er det samme for de typer.

Ved nærmere undersøgelse af datablade for EPROM'en viste det sig imidlertid at denne var fuld kompatibel med TTL, idet alle anvendte niveauer lagde inden for TTL grænserne. Forskellen på TTL og CMOS, udover de forskellige niveauer, er flere. TTL'en er som antydnet den ældste teknologi, og har sammenlignet med CMOS'en et højt strømforbrug. Desuden fylder den del mere. CMOS'en har en høj indgangsimpedans, fylder mindre og som sagt også et lavt strømforbrug. Derudover er den mere modstandsdygtig overfor støj. Til gengæld er den også langsommere en TTL'en.

Idag findes det flere måde hvorpå en parallel port kan bruges til kommunikation. Disse går under betegnelserne SPP, EPP og ECP, der står for henholdsvis Standard Parallel Port, Enhanced Parallel Port og Extended Capabilities Port. Som antydnet er standarden idag SPP, men de fleste styresystemer kan køre dem alle, og konfigurere selv efter behov. Vi vil ikke yderligere beskrive disse kommunikationsmuligheder, idet de ikke har nogen betydning for vores projekt. Vi valgte nemlig fra starten af at anvende den oprindelige konfiguration (ingen af de ovenstående) for printerporten, hvilken betyder at dataerne kun kan gå i en retning for hvert bit. Derfor er der en række bit der anvendes til kommunikation ind i porten og en række bit til kommunikation ud af porten.

Oprindeligt var porten jo designet til printere, og kommunikationen ville derfor hovedsaglig forgå fra computeren og ind i printeren. Hertil anvendes de 8 adressebit der gør brug af pin 2-9. For at printeren har mulighed for at give computeren besked om fejl, papirmangel eller andet, har man reserveret 5 bit til diverse statussignaler. Det er hovedsaglig disse bit vi gør brug af i vores projekt. Derudover har man også nogle bit, der bliver brugt til kontrol, men disse er mindre interessante i denne sammenhæng. For at overskueliggøre situationen har vi fremstillet tabel 9.1

Som det første skal det bemærkes at 'Busy' er inverteret inde i computeren, dvs. når printeren sender høj ud, vil computeren læse det som en lav, og om-

Pin	Bit	Navn	Typisk adresse (Decimal)
1	0	Strobe	890
2	0	Data 0	888
3	1	Data 1	888
4	2	Data 3	888
5	3	Data 4	888
6	4	Data 5	888
7	5	Data 6	888
8	6	Data 7	888
9	7	Data 8	888
10	6	Acknowledge	889
11	7	Busy	889
12	5	Paper Empty	889
13	4	Select In	889
14	1	Auto Linefeed	890
15	3	Error	889
16	2	Initialise	890
17	3	Select out	890
18-25	-	Ground	-

Tabel 9.1: Oversigt over LPT portens pins

vendt. Dette havde stor betydning da vi opstillede de værdier som programmet skal sammenligne inputtene med. Det man sender fra EPROM'en er jo ikke det samme som det man læser på porten, hvilket der skulle tages højde for i programmeringen af softwaren til computeren.

I kolonnen med 'Bit' er der angivet hvilken værdi det pågældende bit har på den angivne adresse. Bit 0 er det mindst betydningsfulde, mens bit 7 er det mest betydningsfulde (most and least significant). Dette er især vigtigt at holde for øje når man skal konvertere det binære input på porten, til et decimal tal i programmet.

Navnene stammer fra da man udviklede porten, hvor det var hensigten at denne udelukkende skulle bruges til printere. De har dog ingen umiddelbart betydning for de enkelte bits funktion, men antyder blot hvilken anvendelse de i sin tid var tiltænkt.

Talkboardets tilslutning til parallelporten forgår direkte fra EPROM'ens udgange. Vi har opstillet tabel 9.2 for forbindelserne mellem EPROM og LPT port.

EPROM	LPT pin (Navn)
DQ7	→ 11 ( <i>Busy</i> )
DQ6	→ 10 ( <i>Acknowledge</i> )
DQ5	→ 12 ( <i>PaperEmpty</i> )
DQ4	→ 13 ( <i>SelectIn</i> )
DQ3	→ 15 ( <i>Error</i> )
GND	→ 18 ( <i>Ground</i> )

Tabel 9.2: Oversigt over forbindelser mellem EPROM og LPT

Sammenligner man de to tabeller (9.1 og 9.2), kan det ses at det mest be-

tydningsfulde bit på EPROM'ens udgang er koblet til det mest betydningsfulde bit på indgangen af adresse 889. Dette lattede programmeringen en del, idet vi så ikke skulle til at tage en masse forbehold i programmeringen af EPROM'en. Dette er yderligere beskrevet under afsnittet omhandlende EPROM'en, på side 30.

Idet vi har gjort brug af den gamle metode til kommunikation, er vores muligheder begrænset til det antal kombination man kan frembringe af 5 bit. Antal kombinationer er givet ved:

$$2^5 = 32 \text{ muligheder} \quad (9.1)$$

I vores projekt er dette rigeligt, men skulle man udbygge produktet til at have flere muligheder, ville man være tvunget til at finde en anden kommunikationsform med LPT porten. En af de muligheder man her kunne gøre brug af, er den såkaldte nibble mode, der gør det muligt at sende 8 bits data, med kun 5 bit. Teorien er simpel, idet man blot lader de 8 bit overføre af to omgange, dvs.  $2 \times 4\text{bit}$ . De databærende bit får altså forskellig betydning, alt efter om det er første eller anden del af signalet der sendes. Dette er overskueliggjort i tabel 9.3.

LPT Pin (Navn)	1. nibble	2. nibble
10 ( <i>Acknowledge</i> )	Sættes lav for gyldig nibble	Sættes lav for gyldig nibble
11 ( <i>Busy</i> )	Databit 3	Databit 7
12 ( <i>PaperEmpty</i> )	Databit 2	Databit 6
13 ( <i>SelectIn</i> )	Databit 1	Databit 5
15 ( <i>Error</i> )	Databit 0	Databit 4

Tabel 9.3: Oversigt over funktioner i nibble mode

Skulle produktet udvides med flere muligheder, ville ovenstående metode være en måde at gøre det på. Dog ville det kræve en omfattende ombygning af Talkboardet, men det ville helt klart være en mulighed.

# Kapitel 10

## Produktudvikling

### 10.1 Udgangspunktet

Da vi påbegyndte projektet havde vi en ide om hvordan Talkboardet skulle udformes. Denne ide indebar bl.a. en avanceret timing af de forskellige signaler, som senere hen blev skåret væk, ganske enkelt fordi produktet ville kunne fungere bedre uden disse.

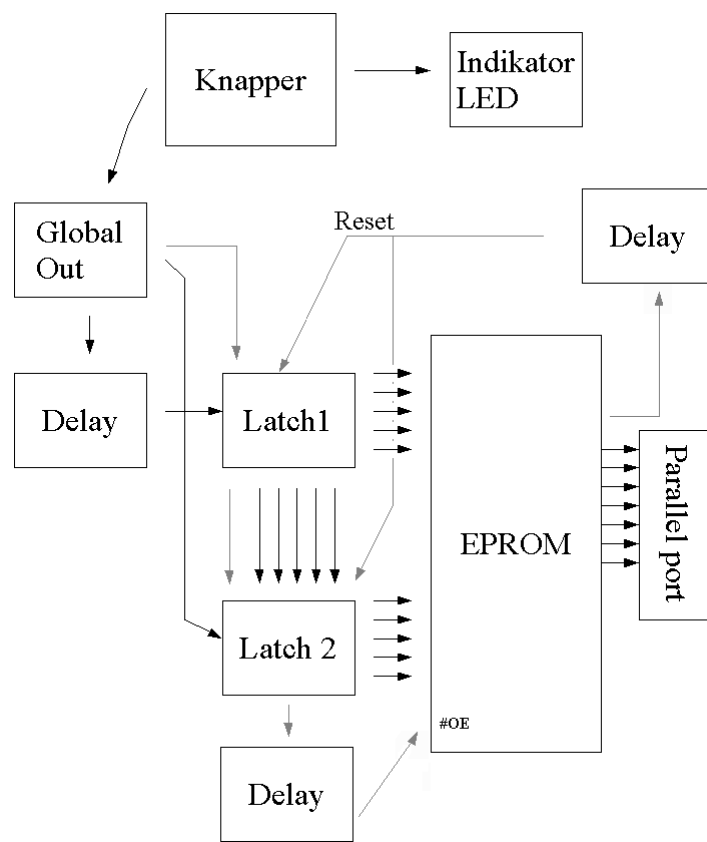
Det var bl.a. som udgangspunkt antaget at kredsen skulle resettes af EPROM, ved at tilbagekoble en af dennes udgange, og så forsinke dette signal, så LPT porten havde mulighed for at læse på indgangen, inden kredsen blev reset. På denne måde ville vi dog ikke være sikre på at LPT porten havde læst signalet ordentligt, og vi valgte derfor at koble en af LPT portens udgange tilbage som reset, og simpelthen lade programmet resette Talkboardet. På den måde ville vi være sikre på at signalet fra Talkboardet var blevet opfattet af computeren, og vi ville samtidig spare en delaykreds.

Derudover havde vi planlagt at anvende EPROM'ens  $\#OE$  (Output Enabled), for at sikre os at porten ikke ville få noget signal, inden alle ti adressebit var klar på EPROM'ens indgang. Dette kunne gøres ved at sende Global Out igennem de to latche, og via en inverter ind i EPROM'ens  $\#OE$ , efter en kort forsinkelse. På den måde ville der først læses noget ud fra EPROM'en når hele adressen var sammensat og klar på EPROM'ens indgang. Det viste sig imidlertid i forbindelsen med programmeringen af softwaren til computeren, at det ville være praktisk med en sådan mellemværdi, idet programmet så ville være klar over at der nu kom en værdi den skulle sammenligne med. Derfor undlod vi dette, hvorved vi igen sparede en delaykreds, så vi var nede på en enkelt, imellem de to latche.

Vores første ide er skitseret på figur 10.1, og til sammenligning er den endelige løsning illustreret på figur 18.1. Det der er tegnet med gråt, er de ændringer der er foretaget fra udgangspunktet til den endelige løsning.

### 10.2 Print

Næsten alt printet er tegnet i AutoCAD, da dette var hurtigst. På printbillederne i rapporten er banerne tynde, men dette er fordi stregtykkelsen styres under udprintning. Denne er sat til en tykkelse på 0,7 mm til 0,8 mm.



Figur 10.1: Vores første udgangspunkt

Printfremstillingen blev for det meste gjort i hånden. Dette skyldes at det i vores tilfælde var nemmere at overskue, fremfor at sætte det op i Multisim, og videre til Ultiboard. Når man opsætter et diagram i Multisim, hvor der indgår IC med flere af samme type kreds (eks. hex inverter), er man hele tiden nødt til at holde et overblik over hvilken section der er placeret hvor i IC-en, for ikke at være tvunget til at lave alt for mange lus. Nå man tegner forbindelserne i hånden, kan man bedre overskue at bruge de sectioner af IC-en der ligger nærmest for, og dermed skaber færrest lus.

Vi har i bilagsmappen vedlagt en brugsvejledning til produktet, der beskriver anvendelsen samt de forskellige funktioner.

**Del III**

**Mekanikdelen**

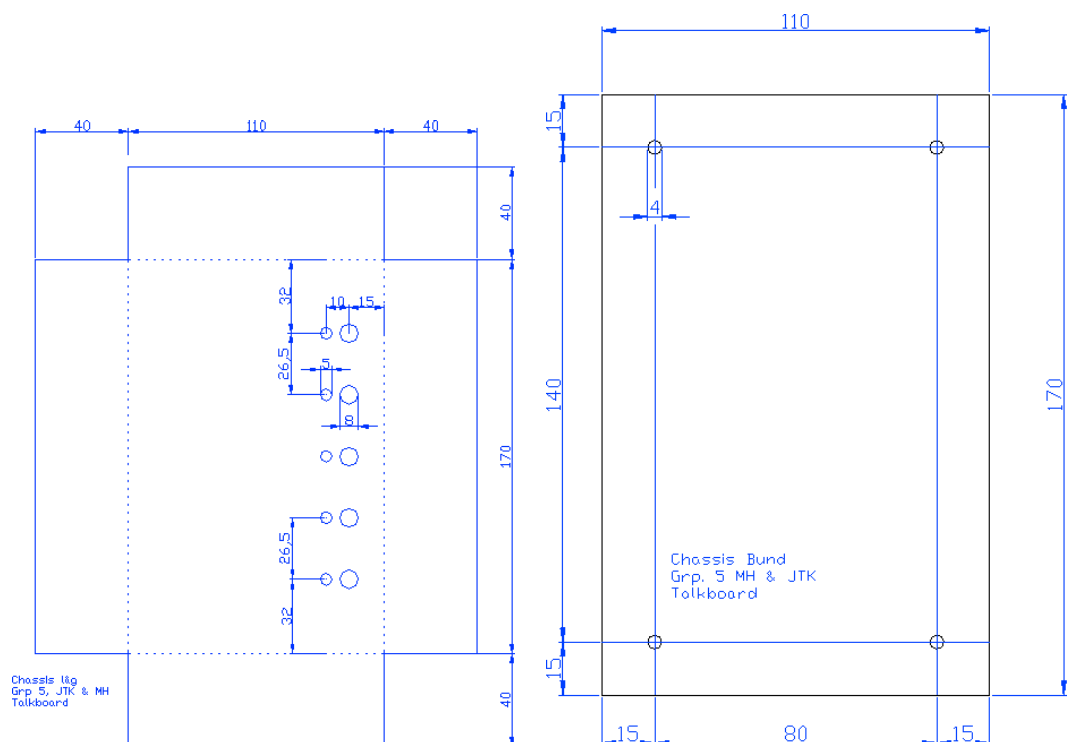


# Kapitel 11

## Chassis

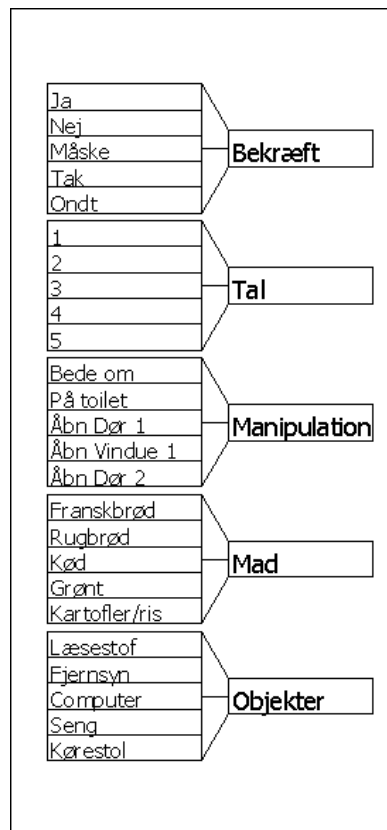
Chassiets funktion er at indeholde elektronikken, og samtidig gøre det nemt at betjene talkboardet. Da vi skulle bestemme dimensionerne på chassiet foldede vi en model af papir, og lavede mål efter dette. Et billede af chassiet er på forsiden.

Knapperne er placeret til højre for LED-erne (disse indikerer hvilken knap man har trykket på), idet højrehåndede så nemmere kan se. Problemet er at deres hånd ellers ville "skygge" for LEDerne og knapperne. Udover dette er der placeret et mærket med et index over lydene. Arbejdstegninger til fremstillingen er på figur 11.1



Figur 11.1: Arbejdstegninger til chassiet

Mærkatet er fremstillet til at være læsbart fra stor afstand, og er derfor ikke så farverigt. Mærkatet er sat fast med dobbeltklæbende tape. Tegningen til mærkatet er på figur 11.2. Selve chassiset er fremstillet af aluminium.



Figur 11.2: Mærkat til chassiset

## Kapitel 12

# Døråbneren

I løbet af projektperioden, valgte vi at begrænse døråbneren med hensyn til den praktiske fremstilling. Vi ville i stedet nøjes med udarbejdelse af arbejds-tegninger, og beregninger, og så lade chassiset være det eneste der praktisk blev fremstillet. Dette afsnit vil derfor omhandle teorien bag døråbneren, samt tegninger og beregninger for denne.

### 12.1 Udarbejdelse

Vi havde flere løsningsforslag oppe at vende da vi arbejdede på døråbneren. Forskellen var hovedsaglig hvad der skulle drive døråbneren, dvs. hvorvidt det skulle være en momentstærk elmotor, pneumatisk stempel, eller noget helt tredje. Vi valgte dog hurtigt at tage udgangspunkt i den momentstærke motor, idet dette var noget vi havde et reelt kendskab til, og samtidig ville være det nemmeste at styre.

Selve den mekaniske del, dvs. den del der skulle kunne åbne døren ved hjælp at motoren, minder i høj grad om den mekanisme man finder på dørpumper, dvs. anordninger der påmonteres døre for automatisk lukning. Denne mekanisme består af en leddelt arm der 'skubber' på døren når den påmonterede motor foretager en vandring. Princippet er vist på figur 12.1.

Denne metode er den der umiddelbart fylder mindst, og dermed er til mindst gene. Samtidig er den simpel, hvilket gør produktionen nemmere og billigere.

Arbejdstegningen for den mekaniske arm er vist på figur 12.2.

I forbindelse med dimensioneringen af motoren kan det være praktisk at kende noget til den dør den skal påmonteres. Vi har derfor lavet nogle beregninger for en indvendig fyrre-dør med følgende data:

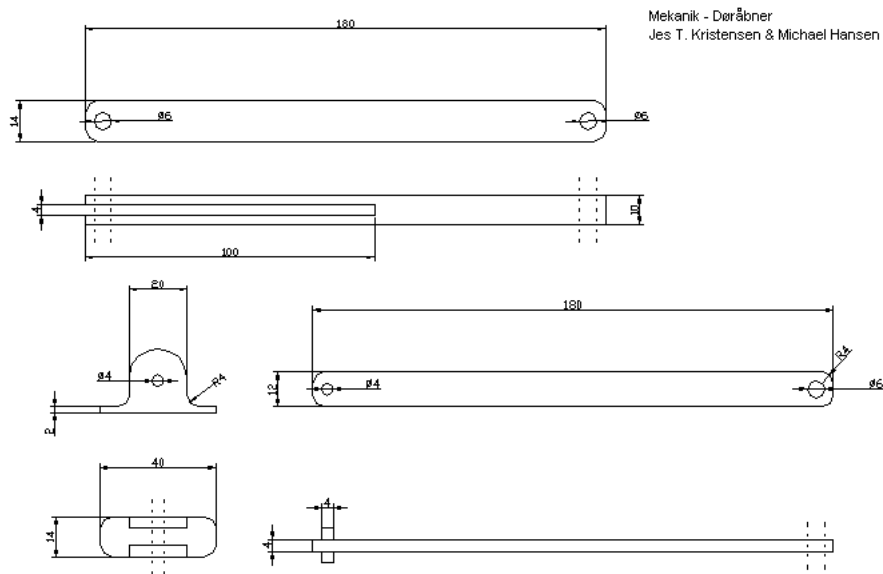
$$m = 25kg \quad h = 190cm \quad l = 80cm \quad t = 4cm$$

Figur 12.3 illustrere de enkelte mål.

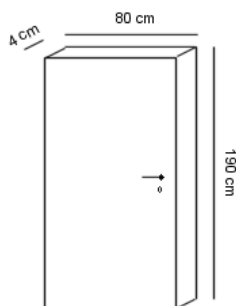
Det første vi vil beregne er dørens inertimoment, idet denne har betydning for dørens impulsmoment, samt den energi der skal tilføres for at døren opnår en given hastighed. Inertimomentet kan findes ved hjælp af Steiners sætning:



Figur 12.1: Princippet illustreret



Figur 12.2: Arbejdstegninger for døråbneren



Figur 12.3: Illustrering af de angivene mål

$$I = m \cdot d^2 + I_0 \quad (12.1)$$

$$(12.2)$$

$d$  angiver den afstand som omdrejningspunktet er forskudt fra massemidt-punktet.  $I_0$  er inertimomentet omkring en akse gennem massemidt-punktet. Døren kan, set ovenfra, betragtes som en stang, hvorfor  $I_0$  er givet ved:

$$I_0 = \frac{1}{12} \cdot m \cdot l^2 \quad (12.3)$$

Inertimomentet for vores dør er derfor givet ved:

$$I = m \cdot d^2 + \frac{1}{12} \cdot m \cdot l^2 \quad (12.4)$$

$$\Downarrow$$

$$I = m \cdot \left(\frac{l}{2}\right)^2 + \frac{1}{12} \cdot m \cdot l^2 \quad (12.5)$$

$$\Downarrow$$

$$I = \frac{1}{3} \cdot m \cdot l^2 = \frac{1}{3} \cdot 25\text{kg} \cdot (0,8\text{m})^2 = 5,33\text{kg} \cdot \text{m}^2 \quad (12.6)$$

At beregne impulsmomentet er derimod lidt mere besværligt, idet denne beregning bygger på at legemet bevæger sig med en konstant vinkelhastighed  $\omega$ . Det vil jo ikke være tilfældet, for selvom vores motor kører med konstant hastighed, vil cirkelbevægelsen i armen gøre at døren ikke gør det. Vi vil i det efterfølgende derfor blot antage at døren kommer til at bevæge sig med en konstant hastighed, som vi selv fastsætter. Antager vi at døren bruger 3 sekunder på at åbne  $90^\circ$ , vil vinkelhastigheden være givet ved:

$$\omega = \frac{d\varphi}{dt} = \frac{0,5\pi}{3\text{s}} = 0,524\text{s}^{-1} \quad (12.7)$$

Dermed vi impulsmomentet være givet ved:

$$L = I \cdot \omega = 5,33\text{kg} \cdot \text{m}^2 \cdot 0,524\text{s}^{-1} = 2,79\text{kg} \cdot \text{m}^2 \cdot \text{s}^{-1} \quad (12.8)$$

Den energi som motoren skal bruge til at bringe døren op i denne givne hastighed, må være givet ved den rotationenergi som døren har, hvis man ser bort fra gnidning i hængsler og luftmodstand. Denne energi er givet ved:

$$E_{rot} = \frac{1}{2} \cdot I \cdot \omega^2 = \frac{1}{2} \cdot 5,33\text{kg} \cdot \text{m}^2 \cdot (0,524\text{s}^{-1})^2 = 0,73\text{J} \quad (12.9)$$

Dette holder selvfølgelig ikke i praksis, idet der vil være en betydelig gnidning i hængsler, samt tab når kræfterne skal overføres fra motoren til døren igennem armen.

**Del IV**

**Software programmet**

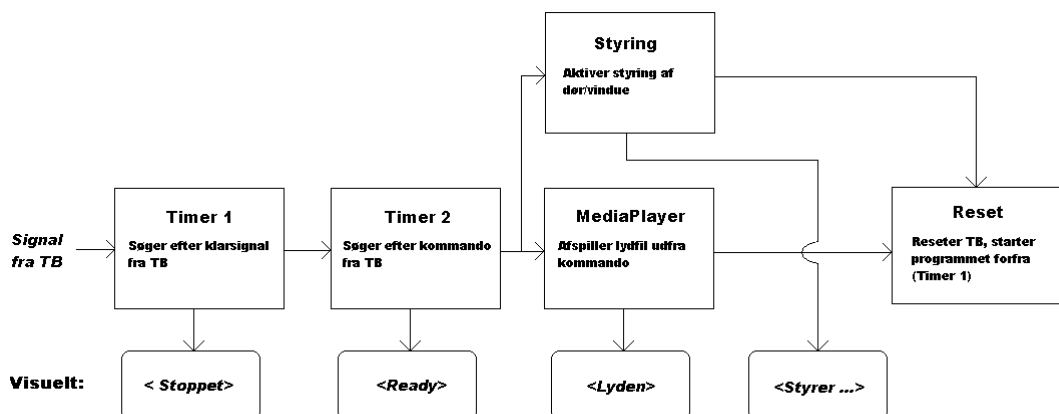
## Kapitel 13

# Software programmet

Software til computeren blev skrevet ved hjælp af programmet Delphi 6, der på mange måder minder om Visual Basic, men blot anvender Pascal som programmeringssprog, fremfor Basic. Altså er det et program der i høj grad hjælper med den grafiske side af programmet, samtidig med at det indeholder en række forudprogrammerede elementer, som man kan gøre brug af i sine programmer. Det betyder at man i vid udstrækning ikke behøver at skrive anden kode, end den der skal til for at aktivere og styre de forskellige elementer som programmet skal indeholde.

Vores kendskab til programmering i Delphi, og dermed programmeringssproget Pascal, var meget begrænset. Det betød at vi fra start af havde besluttet at holde det så simpelt som muligt, og ikke ville spekulere videre over muligheder for opdatering af programmet eller lignende. Programmet skulle blot kunne udføre den funktion som vi ønskede i forbindelse med dette specifikke projekt.

For at overskueliggøre programmets funktion, har vi visualiseret det på figur 13.1, hvor vi har opdelt de enkelte funktioner i forskellige blokke.



Figur 13.1: Flowchart for programmets funktion

I vores tilfælde hvor vi kører Windows 98 som styresystem, kan kommu-



nikationen med parallelporten klares ved hjælp af en enkelt dll fil. Havde vi anvendt Windows 2k eller NT, ville dette ikke umiddelbart have været muligt, uden at skulle foretage nogle omfattende driver-opdateringer. Filen vi gør brug af, er betegnet IO.dll, og er hentet fra nettet<sup>1</sup>. Ved at referere til denne fil fra programmet, er det muligt at henholdsvis skrive til, og læse fra LPT porten, ved hjælp af to forskellige kommandoer, PortIn og PortOut. Det eneste der er påkrævet er, at dll'en ligger i samme mappe som programmet, og at der refereres til korrekt til filen fra programmet.

Parallelporten har flere forskellige adresser, afhængig af hvilke bit man er interesseret i at anvende. Skriver man f.eks. til computeren, foregår det på en standard parallelport på adressen 889 (angivet i decimal, ofte bruger man hex, hvorved adressen kommer til at hedde 379h). Denne adresse har 5 bit at gøre godt med, og er normalt tiltænkt den kommunikation som printerne giver computeren ved fejl, papirmangel eller lignende. Kommunikationen ud af computeren foregår på adresse 888 (378h), og der er her 8 bit til rådighed.

Tager vi udgangspunkt i den første blok i flowchartet, Timer 1, så har denne til opgave at læse på portadressen, og aktivere Timer 2, hvis det rigtige startsignal (decimalværdien 88 ) kommer fra Talkboardet. Grunden til at vi kalder den en timer, er fordi det er et forudprogrammeret timerelement fra delphi vi har brugt, til at gennemføre en læsning på porten med et fastsat tidsinterval. Hvis værdien på porten ændrer sig til en fastsat værdi, i dette tilfælde 88 (decimal), vil blokken udføre de instruktioner vi har givet den.

Dette er bl.a. at aktivere den næste timer, nemlig Timer 2, der med et andet tidsinterval læser på porten. Når denne timer genkender en værdi på portadressen, aktiverer den en handling svarende til det vi har defineret for den pågældende værdi.

Ofte vil det være at aktivere MediaPlayeren som så afspiller den lyd-fil der er defineret for værdien. Det kunne dog også være et signal til styring af døre/vinduer, hvorved den så blot vil sende et signal ud af porten som skulle aktivere en evt. døråbning. Med hensyn til MediaPlayeren så er den et af de mange forudprogrammerede elementer man kan gøre brug af i sin kodning. Det eneste der skal gøres, er blot at definere de indstillinger som afspilleren skal have i det pågældende program.

Når den definerede handling er udført, sendes der et signal ud af porten, på adresse 888, der reset'er Talkboardet så den er klar til brug igen. Herefter startes Timer 1 igen, og programmet køres igennem på ny.

## 13.1 Programmeringen

Vi vil ikke minutiøst gennemgå hele koden til programmet, idet programmeringsdelen jo som sådan kun er en lille del af projektet, og det element der har mindst med Design og Produktion-faget at gøre. Vi har i stedet valgt at gennemgå en lille del af koden, nemlig den del som vi i det ovenstående har beskrevet som Timer 2, og delvist også MediaPlayeren.

<sup>1</sup><http://www.geekhideout.com/iodll.shtml>

Vi antager nu at programmet har modtaget klarsignalet fra Talkboardet, og dermed er fortsat videre til Timer 2. Denne timer sammenligner som sagt værdien fra porten, med værdierne opstillet i programmet. Der er mange måder hvorpå man kunne havde gjort dette, og den mest åbenlyse ville være ved brug af en database der indeholdt disse værdier. På den måde ville det have været nemt at opdatere, og formodentlig også nemmere at overskue koden. Men som tidligere nævnt havde ingen af os speciel stor erfaring med Pascal programmering, og kendte derfor intet til brug af databaser i programmeringen. Fremfor at bruge en masse tid på at sætte os ind i det, valgte vi i stedet at bruge ”copy-paste metoden”. Vi har simpelthen opstillet en lang række if-sætninger, der bruges til sammenligning med værdierne på porten, hvilket betyder at der er ligeså mange if-sætninger som der er kombinationsmuligheder på Talkboardet, dvs. 25. Et lille kodeeksempel vil se sådan ud:

```
procedure TTalkboard.RenewClick(Sender: TObject);

var

InValue : Byte;
Sound: string;

begin

Invalue:=Portin(889);
Sound:=inttostr(Invalue);

if

Sound=inttostr(136) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Bekræftelser\ja.wav';
    Open;
    wait:=true;
    play;

end;

Talkboard.Label1.caption:='Ja';
Portout(888,1);
exit;

end
```

I Delphi skal alle variabler defineres inden brug. Derfor skrives der i starten af proceduren: *var*, og efterfølgende de variabler man har tænkt sig at bruge.

En sådan variabel kaldes *lokal*, idet den kun kan anvendes i den pågældende procedure. Vi har her defineret *InValue* og *Sound*, som henholdsvis "Byte" og "string". Det viste sig dog senere at denne konvertering faktisk ikke var nødvendigt, idet man kunne hente værdien direkte ud som decimaltal.

Først sættes variabelen *InValue* til portens værdi. Herefter konverteres den til en streng, der så sættes lig med variabelen "Sound".

If-sætningen fungerer ligesom i alle andre programmeringssprog, altså hvis udtrykket er opfyldt, udføres den efterfølgende kode, ellers fortsættes der længere ned i scriptet. Koden *inttostr(136)* konvertere tallet (integer) til en streng, hvilket er nødvendigt idet man ikke kan sammenligne en streng med et tal. Antager vi at der nu står 136 (decimal) på portadresse 889 (stadig decimal), vil den viste kode altså blive udført, idet udtrykket er opfyldt (Variabelen *Sound* er netop lig med 136).

I udtrykket omkring MediaPlayerens handlinger, er syntaksen skrevet en smule anderledes end man normalt ville have gjort det. Funktionen er den samme, idet ovenstående metode blot er lidt mere elegant måde at skrive tingene på, når man har mange handlinger for det samme objekt. Den første linie i denne kode beskriver bare at der nu påbegyndes en række handlinger med objektet "MediaPlayer1"

Der angives nu hvilken fil man ønsker at åbne, hvilket gøres med kommandoen *Filename*. For at man ikke skal til at ændre i programmet hver gang man flytter rundt på det, har vi angivet en relativ sti for filnavnet, således at mappen med lyde blot behøver at ligge i samme mappe som programmet gør, og man dermed isolerer programmet fra den resterende biblioteksstruktur.

Herefter åbnes filen med kommandoen *open*;. Den efterfølgende kode bruges til at sikre, at hele lydfile bliver afspillet. Den forhindrer nemlig andre procedure i at begynde, før MediaPlayeren har fuldført sin afspilning. Til sidst bedes MediaPlayeren om at afspille lyden med kommandoen *Play*;

Når MediaPlayeren har afspillet lyden, skal den påbegyndte procedure afsluttes, hvilket gøres med *end*;. For at gøre brug af nogle af Delphi's muligheder, har vi valgt at lade et mindre visuelt element indgå, nemlig en label der beskriver hvilken handlingen programmet er i gang med at udføre. Idet vi nu er i gang med at afspille lydfile "Ja.wav", ændrer vi i det der står i label1, til teksten "Ja". Nu har vi udført de ønskede funktioner, og skal nu have reset'et Talkboardet. Dette gøres ved at skrive værdien '1' til adressen 888, hvorved bit 0 på de 8 databit ud af porten, går høj (5V). Denne er tilbagekoblet til Talkboardet, der herved reset'es.

Herefter skal proceduren afsluttes, dvs. at programmet skal ud af if-sætningen og tilbage for at sammenligne med andre værdier, hvilket gøres ved kommandoen: *exit*;. Når Talkboardet er reset, vil der altid stå en fast værdi på porten, som vi så senere kan bruge til at initialisere programmet forfra.

Som sagt har vi kun valgt at beskrive et stykke af koden her i rapporten, men vi har vedlagt scriptet til programmet hvor der er indsat kommentarer til de enkelte linier i koden. Dette script kan findes i bilagsmappen.

## Kapitel 14

# Lyde

De .wav filer (lyde) vi skulle bruge til Talkboardet skulle jo passe til dennes funktion, og vi var derfor selv nødt til at indtale dem. Dette gjordes ved hjælp af programmet SoundForge, og en mikrofon fra et headset. Sidst nævnte er skyld i en lettere forringet lyd kvalitet, der dog ikke forvolder de store problemer med hensyn til forståelsen.

Med hensyn til de indtalte ord, har vi forsøgt at udvælge den sammensætning vi umiddelbart fandt mest grundlæggende, men vi har dog ikke gennemgået noget videre informationssøgning i den forbindelse. Det må være op til eksperterne i en eventuel salgssituation. Lydene kan ses i tabel 14.1.

Gruppe	Indhold
Bekræft	Ja Nej Måske Tak Ondt
Tal	1 2 3 4 5
Manipulation	Bede om På toilet Åbn dør 1 Åbn vindue 1 Åbn dør 2
Mad	Franskbrød Rugbrød Kød Grønt Kartofler/ris
Ojekter	Læsestof Fjernsyn Computer Seng Kørestol

Tabel 14.1: Tabel over lyde

Del V

**Opsamling**

# Kapitel 15

## Opsamling

I dette kapitel er opsamlingen og refleksionen på projektet.

### 15.1 Projektet rent tidsmæssigt

Vi må nok erkende at tidsplanen ikke helt blev holdt. Selve forløbet blev fundamentalt anderledes, idet vi havde gang i flere blokke på en gang, hvilket vi ikke havde regnet med i tidsplanen. Desuden blev styringen til døråbningen helt udfaset, hvilket vi blev nødt til pga. tidspres.

For at lette overblikket over projektforløbet har vi konstrueret tabel 15.1 der viser hvornår vi har haft de forskellige faser færdigt i forløbet.

Element	Udtænkt	Fumlet	Printlayout	Ætset	Færdigt
Knaparray	17/3	21/3	17/3	24/3	31/3
LED-styring	17/3	21/3	17/3	24/3	31/3
Global Out	16/3	21/3	16/3	24/3	31/3
Delay kreds	28/3	28/3	31/3	31/3	7/4
PC program	11/4	-	-	-	23/4
Latche	7/4	11/4	7/4	11/4	11/4
EPROM'en	17/3	-	17/3	24/3	31/3
Døråbning (tegninger)	11/4	-	-	-	22/4
Forsyning	23/4	23/4	23/4	-	23/4
Montering	-	-	-	-	23/4

Tabel 15.1: Oversigt for projektperioden

Se eventuelt dagbogen i bilagene (side XV) for mere information.

### 15.2 Projektet økonomisk

Budgettet for projektet var som beskrevet i projektbeskrivelsen, på 1200 kr. for en tomandsgruppe. For at lette overblikket udgifterne for projektet, har vi fremstillet tabellerne 15.2 og 15.3. Disse illustrere henholdsvis de samlede udgifter

samt de udgifter der er brugt på komponenter til det færdige produkt, dvs. kun prisen på de anvendte materialer til produktet, uden ødelagte komponenter.

Type	Navn	Best. nr.	Antal	Pris	Ialt
EPROM	AM27C256-200	203-415	1 stk	66,16 kr	66,16 kr
Latch	MM74HC273	379-505	2 stk	4,37 kr	8,74 kr
Inverter	MM74HC04N	378-252	2 stk	2,60 kr	5,20 kr
Knapper	Sluttekont.	-	8 stk	5,00 kr	40 kr
RS-latch	CD4043BC	385-141	2 stk	5,20 kr	10,4 kr
OR-gates	MM74HC32	370-453	3 stk	3,77 kr	11,31 kr
Spænd. reg.	LM7805CT	412-776	1 stk	4,93 kr	4,93 kr
LED std.	Rød LED	-	10 stk	1,79 kr	17,9 kr
Timer	LM555	409-327	4 stk	3,38 kr	10,14 kr
Diode	1N4148	368-106	1 stk	0,14 kr	0,14 kr
Modstande	Div. E12	-	30 stk	0,10 kr	3,0 kr
Kond.	Div. E12	-	20 stk	1,00 kr	20,0 kr
IC sokkel	Div.	-	10 stk	5,00 kr	50 kr
Print	Fotoprint	-	600 $cm^2$	0,25 kr/ $cm^2$	150 kr
Hulprint	Hulprint	-	25 $cm^2$	0,50 kr/ $cm^2$	12,5 kr
Alupld.	Alu. plade	-	0,75 kg	20 kr/kg	15 kr
Montering	Skruer m/sok	-	20 stk	3 kr	60 kr
Ialt					485,42 kr

Tabel 15.2: Det totale budget

Mange af priserne er anslåede idet komponenterne (især de passive) er købt af skolen, og der derfor ikke er fastsat nogen egentlig pris. Vi har bare gættet os frem efter bedste evne. Prisen for aluminium er korrekt, men vægten er anslået, ligesom det samlede printareal også er anslået.

### 15.3 Ændringer/fejl

Hvis produktet skulle markedsføres, ville der være en del ændringer der skulle foretages. Som det første ville det være praktisk at få tingene ind i chassiset, hvilket jo ikke kan lade sig gøre med den nuværende opstilling (vi lavede den meget markante blokopdeling for nemmere at kunne finde fejl). Ved en større produktion ville man gøre brug af SMD komponenter, der er betydelig mindre og billigere i en omfangsrig produktion. Derved ville vores størrelsesproblem være løst. Desuden ville man højst sandsynligt ekstrudere et chassi i plast i stedet, det er både hurtigere og sandsynligvis også billigere.

Derudover er der også et problem med knapperne. De anvendte knapper er udelukkende brugt, idet større og mere brugervenlige knapper, ville belaste vores budget markant, hvilket der ikke var nogen grund til. Dog ville det være noget af det mest grundlæggende at ændre, idet de anvendte knapper er for små, og skrøbelige til praktisk brug. Desuden er man aldrig helt sikker på hvornår man har trykket dem nok ned, der er ikke nogen mærkbar ”grænse”, hvilket ville blive et problem.

Rent elektronisk er der en design fejl i latchene, som burde rettes. Dette ses tydeligt på figur 6.1 side 22 hvor den med kursiv markerede *læsning* rent faktisk



Type	Navn	Best. nr.	Antal	Pris	Ialt
EPROM	AM27C256-200	203-415	1 stk	66,16 kr	66,16 kr
Latch	MM74HC273	379-505	2 stk	4,37 kr	8,74 kr
Inverter	MM74HC04N	378-252	2 stk	2,60 kr	5,20 kr
Knapper	Sluttekont.	-	5 stk	5,00 kr	25 kr
RS-latch	CD4043BC	385-141	2 stk	5,20 kr	10,4 kr
OR-gates	MM74HC32	370-453	2 stk	3,77 kr	7,54 kr
Spænd. reg.	LM7805CT	412-776	1 stk	4,93 kr	4,93 kr
LED std.	Rød LED	-	5 stk	1,79 kr	8,95 kr
Timer	LM555	409-327	2 stk	3,38 kr	6,76 kr
Diode	1N4148	368-106	1 stk	0,14 kr	0,14 kr
Modstande	Div. E12	-	28 stk	0,10 kr	2,8 kr
Kond.	Div. E12	-	19 stk	1,00 kr	19,0 kr
IC sokkel	Div.	-	10 stk	5,00 kr	50 kr
Print	Fotoprint	-	600 $cm^2$	0,25 kr/ $cm^2$	150 kr
Hulprint	Hulprint	-	25 $cm^2$	0,50 kr/ $cm^2$	12,5 kr
Alupld.	Alu. plade	-	0,75 kg	20 kr/kg	15 kr
Montering	Skruer m/sok	-	20 stk	3 kr	60 kr
Ialt					453,12 kr

Tabel 15.3: Materialer til produkt

er en fejl. Dette kunne man programmere sig ud af (tjekke for at værdien skulle være der et stykke tid), eller sætte endnu et led latches på, som først aflæste efter det andet tryk. Denne fejl resulterer i at man engang imellem får læst forkert ud.

Rent softwaremæssigt burde man skrive programmet om, så det brugte en database til at hente sine filer fra (sammenlignede decimalværdiene med en reference i en database). Dette ville være meget mere overskueligt, og samtidig nemmere at rette i.

## Kapitel 16

# Vurdering

Vi kan ud fra projektføreløbet og resultatet heraf i form af produktet og denne rapport, vurdere at projektet er forløbet succesfuldt, Vi har i projektføreløbet lavet en del ændringer fra vores udgangspunkt, og til det endelige produkt. Samtidig har vi været nødt til at lave nogle prioriteter indenfor det praktiske arbejde, hvilket gik ud over den praktiske fremstilling af døråbneren. Vi fik dog udarbejdet arbejdstegningerne til døråbneren, og fik da også fremstillet noget praktisk i mekanik, nemlig chassiset.

Med hensyn til elektronikdelen, lykkedes det os at få fremstillet et funktionelt produkt, som lever op til vores forventninger. Dog ser det endelige produkt noget anderledes ud rent elektronisk, end vi havde regnet med fra starten. Specielt med hensyn til timing af signalet har vi lavet en del modifikationer, ganske enkelt for at lette programmeringsarbejdet såvel som for at spare tid på udarbejdelsen af ekstra delaykredse.

Med hensyn til tidsplanen, har det nok været det element der har afviget mest fra hvad vi havde forventet. Fremfor at arbejde på hver del for sig, havde vi stort set hele tiden gang i flere dele på samme tid. Dette skyldes at der er visse faser i en produktudvikling der går igen for de enkelte kredse, eks. simulering, dimensionering og fremstilling af print. Det har ikke til alle tider i projektet været muligt at f.eks. at fremstille printet, hvorfor det simpelthen har været en nødvendighed at have flere opgaver i gang på samme tid, hvis ikke man skulle komme bagefter tidsplanen.

Derudover er samarbejdet forløbet rigtigt godt. Opdelingen af mekanik og elektronik har været ligelig, og vi har været i stand til at supplere hinanden på en fornuftig måde.

## Kapitel 17

# Konklusion

Vi kan udfra det netop veloverståede projektforsløb, konkludere at vi har løst den problemstilling vi fremsatte i projektbeskrivelsen. Vi fik udarbejdet et produkt, der kan afhjælpe talebesvædere i deres hverdag. Samtidigt fik vi klargjort produktet til at der kunne kobles en automatisk døråbner på, hvilken vi dog valgte ikke at fremstille.

Problemstillingen med kommunikationen med parallelporten fik vi løst, og på en sådan måde, at Talkboardet burde være kompatibelt med langt størstedelen af alle computere, såfremt disse ikke anvender Windows 2000, NT eller XP. Dette skyldes at vi har gjort brug af den originale kommunikationsmetode, som alle printerporte burde være i stand til at anvende.

Programmet til computeren fik vi ligeledes udarbejdet, og fik i den forbindelse opfyldt alle de krav vi havde til dets funktionalitet. Programmet er i stand til at læse inputtene fra Talkboardet, og sammenligne det med indlagte værdier, hvorudfra den udfører en fastsat handling. Denne handling er enten i form af en afspilning af en lydfil, eller et signal videre til en evt. dørstyring.

Som bekendt valgte vi ikke at udarbejde døråbningsmekanismen, men har i stedet fremstillet arbejdstegninger samt beregninger til en sådan. Det praktiske arbejde indenfor mekanikområdet, blev derfor reduceret til udarbejdelsen af chassiset.

Alt i alt kan vi konkludere at vi har fået udarbejdet et funktionsdygtigt produkt, samt en dertilhørende dokumentation, i form af denne rapport. Altså et vellykket projekt!

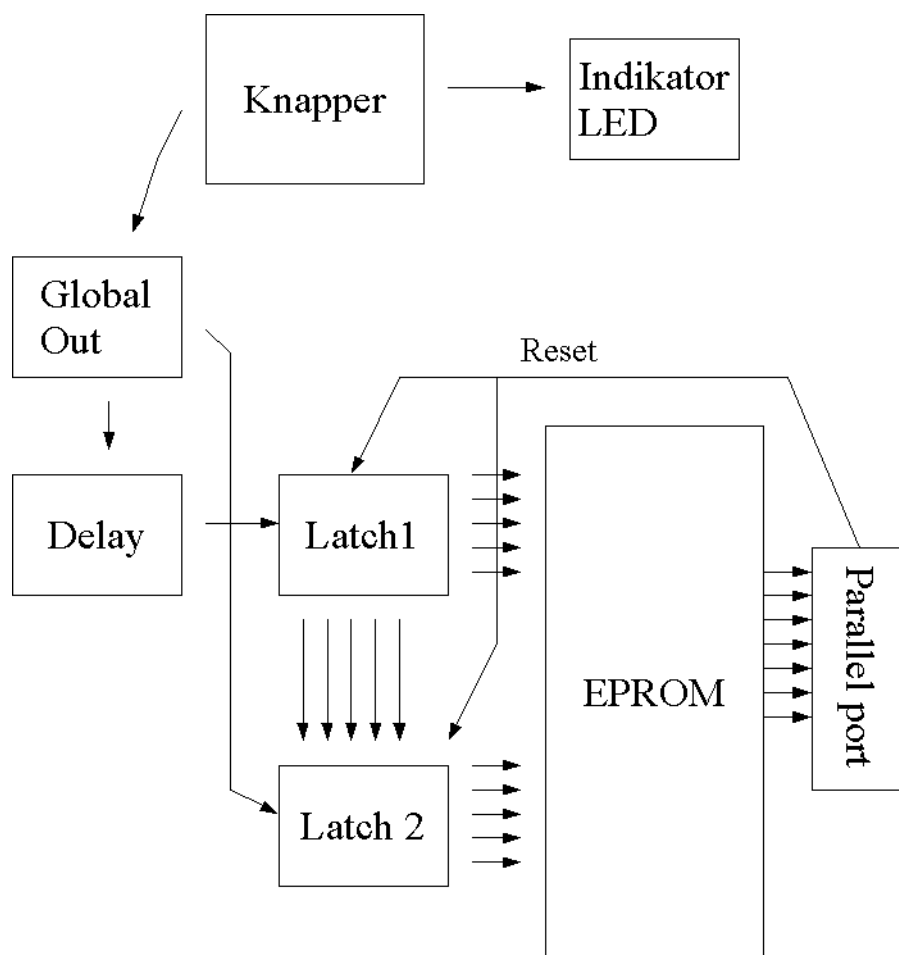
# Del VI

# Bilag

# Kapitel 18

## Elektronik bilag

### 18.1 Folde ud ark



Figur 18.1: Blokdiagram over elektronik delen

## 18.2 Programscriptet

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, MPlayer, Menus;

{Ovenstående beskriver de elementer der indgår i programmet.}

type
  TTalkboard = class(TForm)                                {'type' beskriver de objekter samt}
    MainMenu1: TMainMenu;                                {procedure der er i programmet}
    Filer1: TMenuItem;
    Afslut1: TMenuItem;
    MediaPlayer1: TMediaPlayer;
    Timer1: TTimer;
    Renew: TButton;
    Timer2: TTimer;
    About1: TMenuItem;
    alkboard1: TMenuItem;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Reset: TButton;
    procedure RenewClick(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure alkboard1Click(Sender: TObject);
    procedure Afslut1Click(Sender: TObject);
    procedure ResetClick(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Talkboard: TTalkboard;                                {Variablerne der her kan angives er globale}
                                                    {dvs. de kan kaldes i alle procedures.}

implementation

{implementation er de andre 'units' (dette er 'Unit 1') der bruges.
Unit2 er en aboutbox}

uses Unit2;
```

```
{ $R *.dfm }

procedure PortOut(Port : Word; Data : Byte); stdcall; external 'io.dll';
function PortIn(Port:Word):Byte; stdcall; external 'io.dll';

{Ovenstående kalder io.dll, som programmet anvender til at kommunikere
med LPT porten. Samtidig beskrives de anvendte funktioner (PortIn, PortOut)}

procedure TTalkboard.Timer2Timer(Sender: TObject);

{Fortæller at der startes en en procedure, og med hvilket element}

begin

if

{Traditionel if sætning}

portin(889)=128

then begin
Talkboard.Timer1.Enabled:=false;
Talkboard.Label1.caption:='Stoppet';
portout(888,0);
Talkboard.Timer2.Enabled:=true;

{Hvis if-sætningen opfyldes, eksekveres den efterfølgende kode. Først beskrives
objektet, efterfuldt af handlingen den ønskede handling.
Alle linier afsluttes med ;}

end

else if

{Hvis ikke den første if-sætning er opfyldt, forsættes der videre til den næste,
osv.}

portin(889)=88

then begin

Talkboard.Timer2.Enabled:=false;
Talkboard.Label1.caption:='Ready';
Talkboard.Timer1.enabled:=true;

end;
end;

{Proceduren afsluttes med End;}
```

```
procedure TTalkboard.RenewClick(Sender: TObject);

var
  InValue : Byte;
  Sound: string;

{Her er der først startet en ny procedure, hvorefter de lokale variabler er
defineret. Det efterfølgende indeholder en lang række if-sætninger som denne
procedure sammenligner med}

begin
  Invalue:=Portin(889);
  Sound:=inttostr(Invalue);

  if
    Sound=inttostr(136) then

    begin

      {Er if-sætningen opfyldt, eksekveres det efterfølgende som sagt. I dette
      tilfælde er det MediaPlayer1 der skal udføre en række handlinger. Syntaksen er
      her en smule anderledes, men meningen den samme. Blot en pænere måde at skrive
      det på.}

      With MediaPlayer1 do begin
        Filename := 'sounds\Bekræftelser\ja.wav';
        Open;
        wait:=true;
        play;

      end;
      Talkboard.Label1.caption:='Ja';
      Portout(888,1);
      exit;

      {Når MediaPlayeren afspiller den ønskede lyd, skal displayet ændre indhold
      svarende til den lydfil der afspilles. Herefter skal Talkboardet resettes,
      hvilket gøres ved at sende 1 til adresse 888. Herefter skal programmet hoppe
      ud af denne procedure, hvilket gøres med exit;}

    end

  else

  if
    Sound=inttostr(144) then

    begin
```



```
With MediaPlayer1 do begin
    Filename := 'sounds\Bekræftelser\nej.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Nej';
Portout(888,1);
exit;
end

else

if
Sound=Inttostr(152) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Bekræftelser\måske.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Måske';
Portout(888,1);
exit;
end

else

if
Sound=Inttostr(160) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Bekræftelser\tak.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Tak';
Portout(888,1);
exit;
end
```

```
else

if
Sound=inttostr(168) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Bekræftelser\ondt.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Ondt';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(176) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\tal\1.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='1';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(184) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\tal\2.wav';
    Open;
    wait:=true;
    play;
```

```
end;
Talkboard.Label1.caption:='2';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(192) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\tal\3.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='3';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(200) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\tal\4.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='4';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(208) then
```

```
begin

With MediaPlayer1 do begin
    Filename := 'sounds\tal\5.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='5';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(216) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Manipulation\jegvilgernebedeom.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Bede om';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(224) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Manipulation\jegvilgernepatoilet.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='På toilet';
Portout(888,1);
exit;
```

```
end

else

if
Sound=inttostr(232) then

begin
Talkboard.Label1.caption:='Styrer dør';
Portout(888,33);
exit;
end

else

if
Sound=inttostr(240) then

begin
Talkboard.Label1.caption:='Styrer vindue';
Portout(888,65);
exit;
end

else

if
Sound=inttostr(248) then

begin
Talkboard.Label1.caption:='Styrer dør';
Portout(888,129);

exit;
end

else

if
Sound=inttostr(0) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Mad\Franskbrød.wav';
    Open;
    wait:=true;
    play;

end;
```

```
Talkboard.Label1.caption:='Franskbrød';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(8) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Mad\rugbrød.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Rugbrød';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(16) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Mad\Kød.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Kød';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(26) then

begin
```

```
With MediaPlayer1 do begin
    Filename := 'sounds\Mad\Grønt.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Grønt';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(32) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Mad\Kartofler.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Kartofler';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(40) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Objekter\Læsestof.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Læsestof';
Portout(888,1);
exit;
end
```

```
else

if
Sound=inttostr(48) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Objekter\Fjernsyn.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Fjernsyn';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(56) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Objekter\Computer.wav';
    Open;
    wait:=true;
    play;

end;
Talkboard.Label1.caption:='Computer';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(64) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Objekter\Seng.wav';
    Open;
    wait:=true;
    play;
```



```
end;
Talkboard.Label1.caption:='Seng';
Portout(888,1);
exit;
end

else

if
Sound=inttostr(72) then

begin

With MediaPlayer1 do begin
    Filename := 'sounds\Objekter\Kørestol.wav';
    Open;
    wait:=true;
    play;

end;
    Talkboard.Label1.caption:='Kørestol';
    Portout(888,1);
    exit;
end

else if

Portin(889)=128

then

Talkboard.Timer2.Enabled:=true;
exit;

{Når programmet har reset Talkboardet, vil værdien på porten ændre sig til
128. Dette udnytter vi til at starte den første timer (Timer2...logisk ikke?)
med igen, hvorefter programmet hopper helt ud af denne procedure med exit;}

end;

procedure TTalkboard.alkboard1Click(Sender: TObject);

begin
{Denne procedure kalder aboutboxen frem.}

AboutBox.ShowModal;
end;

procedure TTalkboard.Afslut1Click(Sender: TObject);
```

```
begin
{Denne procedure afslutter programmet}

close;
end;

procedure TTalkboard.ResetClick(Sender: TObject);
begin
{Denne procedure afslutter programmet}
PortOut(888,1)
end;

end.
```

### 18.3 Dagbog

Her er dagbogen for projektforløbet.

#### Mandag d. 17. Feb 2003

Fik foretaget en brainstorm over vores muligheder. De to muligheder vi tror mest på, er en overvågningsmekanisme til proces, eller et "talebræt" (EPROM med undermenuer =, PC-lydkort) til handicappede.

HKA anbefalede at opsøge et plejehjem o.lin. for at få præciseret behovet. Så vi er sikre på at vi får noget der kan bruges.

Har fået nogen forslag af Gürli, omkring hvad der kunne overvåges mht. proces-faget. Dette vil vi også overveje.

Ellers gik vi på nettet, og fandt en del info om parallel-porte, og deres brug.  
11:20 Vi pakker sammen.

#### Fredag d. 21. Feb 2003

Pga. sygdom var Jes ikke i til stede, hvilke gav visse komplikationer med hensyn til udarbejdelse af projektbeskrivelse. Via telefon aftalte vi derfor at Michael skulle foretage informationssøgning ud fra de ideer vi frembragte Mandag d. 17. Feb 2003.

Under informationssøgningen fik vi fat i materiale omkring digital teknik, som kunne være anvendelig til vores ideer.

Derudover fik vi hver for sig arbejdet videre på den tidligere fremsatte ide, omkring hjælp til handicappede, og har på baggrund heraf, udarbejdet et overordnet blokdiagram der forklarer ideen.

#### Mandag d. 24 Feb 2003

Idag skal vi have defineret vores projekt yderligere. Vi har valgt gruppen med Ebbe, Kevin og Steffen som sparringspartner. Målet for de første 4 timer må

være at gennemtænke projektet dybere. De sidste 4 timer kan vi så bruge på projektbeskrivelsen.

Gennemtænkningen skal hovedsageligt finde ud af, om vores projekt kan gennemføres. Rent teknisk.

10:00 Vi har snakket med HKA omkring EPROMen, og om hvordan den fungerer. Han henviste os til en afdeling i vores fundne bog om skiftehukommelse eller latch-systemer. Dette skulle ses som alternativ til vores hidtidige plan, med at bruge sekventiel-teknik i vores EPROM. Men et er sikkert, det er gangbart.

At bygge et radiosystem var dog ikke gangbart. Problemet i dette var, at man lavede parasit-komponenter, hvilket blev svært at undgå, idet frekvenserne blev så høje, og komponenterne dermed meget små. HKA anbefalede derfor at se på/tage en trådløs ringeklokke, eller styringen fra en radiostyret bil.

12:07 Efter at have set og regnet lidt på priser, er vi kommet frem til at vi skal bruge en latch-IC. Denne kommer til at fungere i en form for seriel til parallel konverter.

### **Fredag d. 28 Feb 2003**

Overskriften for idag er EPROM, som vi så vil kigge nærmere på. Desuden skal vi have skrevet vores projektbeskrivelse færdig, så vi kan få den ud af verden.

Når vi ser tilbage på dagen, har vi brugt det mest af den på at lave projektbeskrivelsen færdig. Denne blev dog også færdig, så HKA har en version med, som han kigger på til mandag.

### **Mandag d. 3 Mar 2003**

Planen for idag er, at få planlagt og udført et interview med Helle Gøtzsche, som har arbejdet med handicappede. Derudover vil vi skrive vores projektbeskrivelse færdig, specielt med henblik på den nærmere koordinering af tiden (vi har fået den rettede projektbeskrivelse tilbage).

### **Fredag d. 7 Mar 2003**

Gennem ugen har vi gået og overvejet vores tidsplan, og besluttet at skære lidt ned på den. Det vi gør er, at fjerne analysen og monteringen af radiosystemet, da dette er endnu et helt nyt emne, som alligevel er rent eksempelstadie. Derfor vil vi hellere bruge tiden på EPROM, døråbning og LED-styring.

Programmet for idag er, at få EPROMens grundlæggende opgave på plads, og så få fundet ud af hvordan vi kan opbygge vores system. Desuden har vi lavet en "notatpapirsskabelon". Så vi er tvunget til at skrive navn, dato og emne på alt det vi skribler ned gennem projektet. Desuden har vi vedtaget, at papirer i mappen, skal ligge efter emne, og så derefter nyeste dato.

Projektbeskrivelsen blev en version nyere, og vi fik et overblik over EPROMmen, og hvordan den kan kobles op. Desuden kom HKA med den glædelige nyhed, at vi i næste uge kan få en computer.

### **Mandag d. 10 Mar 2003**

Med projektbeskrivelsen ude af verden, vil vi nu til at gå igang med det dybere projekt. Temaet for idag er EPROM, som vi vil dybere ind i. Mht. bestilling

af komponenter skal vi enten give HKA en seddel inden 12:00, eller fange ham onsdag før middag (der har han teknologi med 2. års).

08:40 Den helt præcise målsætning for idag, er at få udarbejdet et flowdiagram for EPROMen. Først når dette er gjort, kan vi konkret sige hvad vi vil nå idag, så vi vender tilbage.

09:15 Med flowdiagrammet i hånden kan vi se, at vi gerne vil nå: (EPROM) tidslinje, dimensionering og at lege med programmeringen. Til dette skal vi også nå at se på parallel porten, finde et dataformat til denne, som vi så kan tage højde for i tidslinjen og printudlægget.

14:52 Idag kom vi ind i tidslinjen. Det vi mangler at få bestemt/lavet, er delay<sub>2</sub>, som skal være på 175ns til 200ns, hvilket er lidt svært at lave (billigt), og så skal vi finde delay<sub>3</sub> værdien, som afhænger af parallel porten (vi skulle gerne have computeren på fredag, så vi kan teste det). Udover dette fandt vi kodeeksempler på port->computer interfacet. En modulopbygning af EPROM-printet fik vi også lavet.

Omkring dimensioneringen fandt vi ud af, at der ikke er så meget, alt afhængigt af hvad delay<sub>2</sub> delen kommer til at bestå af. Vi er kommet frem til at vi skal bruge inverter gates, til delay. Til sidst vil vi færdiggøre bestillingssedlen, så vi efterhånden kan få nogen komponenter hjem og arbejde med.

Det vi ikke nåede idag var programmering, printudlæg samt simulering. Simuleringen bliver måske en anelse problematisk, da vi ikke lige ved hvordan Multisim opfører sig overfor EPROM (hvis den overhovedet har et reaktionsmønster?).

### **Fredag d. 14 Mar 2003**

Vi har arbejdet på et printlayout til EPROMen, denne skal lige pudses af. De store hængepartier er det 200ns delay og specifikationerne på parallel porten. Når vi har løst disse problemer er der kun fodarbejde tilbage mht. EPROM delen. Planen for idag er altså:

At få tegnet printet færdigt, at få set på parallel porten, og når HKA kommer, vil vi snakke med ham om det delay og om den PC der burde være klar nu.

10:30 Tegningen er færdig og vi går videre med informationssøgning om parallel portens virkemåde.

11:15 Vi har fundet en del info om parallel porten og vi har besluttet os for en måde at benytte den på. Derudover har vi bestilt en håndbog hjem, måske lidt sent, min problemerne var uforudset store i forhold til hvad vi forventede.

### **Mandag d. 17 Mar 2003**

I weekenden fik vi sat hovederne sammen, og fik lavet print til knapperne og samlingen af den globale udgang. Hængepartiet er således stadig parallel portens opkobling og tider. Vi regner på det bestemtteste med at HKA dukker op kl. 10, hvor vi så kan få vores computer og komme igang med tests osv. Indtil da vil vi arbejde videre på printene, så de er helt klar til fremstilling.

12:30 Nu har vi fået vores computer, og er blevet installeret i lokalet 2.228. Dette er herligt, idet vi nu kan komme rigtigt igang med tingene. Udover dette er vi blevet færdige med de 4 første printplader (EPROM montering, global out, knapper og LED-styring), hvilke vi forventer at gå igang med at lave på fredag. Resten af dagen vil vi arbejde på at få parallel porten til at reagere.

15:07 Nu kører systemerne, og det med porten ser meget lovende ud! Vi vil nu lave de sidste bestillinger og gå hjem. Vi vil skaffe parallelkabel og huse til næste gang.

### **Onsdag d. 19 Mar 2003**

Idag har besluttet at tage noget ekstra tid. Det eneste vi har fokuseret på idag er at lege med parallel porten, hvilket er lykket til fulde. Desuden har vi meldt os til konkurrencen unge forskere, vi har jo en chance for at vinde noget. Desuden har vi nu viden nok omkring timing osv. til at gå videre med de andre print, idet parallel porten ikke har nogen specielle timingskrav/strobesignaler.

### **Fredag d. 21 Mar 2003**

Målet for idag må delvist være at opbygge et overblik over hvor vi står. Desuden har vi lige fået komponenter, så vi kan efterhånden begynde at fumle lidt. Michael vil derfor fokusere på parallel porten, imens Jes vil koncentrere sig om at afprøve de forskellige kredse. Desuden er der lige røget en kontakt, idet der blev loddet for kraftigt på benet, hvorved den smeltede.

Når HKA engang kommer, skal vi også lige have hørt ham om hvordan vi nemmest kan undgå kontaktprel, som godt kunne betyde noget i vores kontaktkreds.

Idag fik vi fumlet global out og LED-styringskredsen. Desuden fik vi set på parallel porten, så vi næsten er klar til at påbegynde det endelige program. Nu er vi som sådan også klar til at begynde på ætsning af vores print.

### **Mandag d. 24 Mar 2003**

08:00 Vi er nu cirka halvvejes i projektet, og vi føler at det kører udemærket. Vi har efterhånden de enkelte dele på plads, og er således nået til fremstillingen. Denne regner vi med at gå dybere ind i idag. Det optimale ville være at kunne få ætset alle vores print-dele, og få loddet lidt på dem.

Således vil vi bruge de to første timer på at få gjort printene helt klar, og så må vi se hvad vi har af muligheder. Hvis ikke det er muligt at få ætset efter de 2 første timer, vil vi gå videre med at kigge på delay delene.

10:45 Vi er nu klar til at ætse. Vi blev ikke færdige efter 2 timer, idet vi ikke kunne ætse, så vi brugte i stedet tiden på at kigge på noget kontaktprel, som vi så har fået fjernet i vores nye print.

11:30 Vi har dimensioneret lidt anti-prel, og fået snakket om mekanikdelen. Nu er der mad.

Efter middagen vil vi gå videre med at få lavet print. Ellers vil vi have parallelportenes nummerering helt på plads, og så rettet vores tidslinie til, så vi efterhånden kan få udstykket delay-delene.

15:10 Dagens udbytte er 4 stk print af en udmærket kvalitet, samt en større viden om parallelportens opbygning. Næste gang bør vi således have alle muligheder for at teste samspillet mellem vores verden og parallel porten. Timingen fik vi ikke set nærmere på idag, desværre.

### **Fredag d. 28 Marts 2003**

Nu er der kommet ekstra RAM i maskinen, og vi skal igang med at samle tingene. Det regner vi med at tiden går med.

Vi har kigget på delay-delene, og fundet ud af at vores delay2 (mellem latch og EPROMmens Output Enable) egentligt kan gøres overflødig, ved at programmere EPROMen rigtigt. Dette vil vi så gøre, idet et delay på 200ns er svært at opnå. Desuden er der jo ingen grund til at gå over vejen efter FlashCam-engangskamerare.

Men idag har vi ikke helt nået så meget som vi gerne ville. Vi er kommet lidt videre med delay-delene, og ellers har vi fået loddet lidt. Knap-printet er således næsten færdigt, men vi mangler dog stadig at sætte en enkelt knap på.

### **Mandag d. 31 Marts 2003**

Idag er det vist store loddedag, samtidig med at delay-delene skal mere på plads. Så det er vist planen for idag.

08:55 Så virker knap-delen endelig. Videre til de andre (testet med måleopstilling med LED + formodstande).

10:20 Nu virker LED-displayet. Testet i forlængelse af knapperne, med +5v over det hele. Totalt elite der!

11:00 Global out kredsen virker nu. Den snoede røde ledning er global out. Den er testet sammen med knap-kredsen.

I dag fik vi endda ætset kredsen til RS-latchen og delay-delen. Michael laver delay delen (den anden) færdig, så den er klar til at ætse på fredag. Ellers må vi sige at det er en OK tilfredsstillende dag, idet vi er kommet et stort stykke videre, samtidig med at tingene virker.

### **Fredag d. 4 April 2003**

Idag vil vi komme hurtigt igang, og få lavet de sidste print færdigt. I weekenden vil vi få arbejdet på PC-programmet, så vi på mandag kan begynde at programmere EPROMmen. Men nu er det middag, og det skal man jo holde.

Efter middage blev vi bedt om at mødes i klasselokalet, hvor vi fik at vide, at vi skal aflevere en status på mandag. Delay delen blev udformet idag, men den nåede ikke at blive ætset. Latchen drillede lidt, så den må vi arbejde videre på næste gang.

### **Mandag d. 7 April 2003**

Weekendens party affødte ikke noget delphi-program, men sådan er det jo. Udover dette har vi tjekket op på unge-forskere, hvor vi fandt ud af, at det kun er finaledeltagerne der får direkte besked. Alle andre må vente til d. 23 April, med at vide noget som helst.

Status for idag er jo først og fremmest status, og så noget print. Michael arbejder videre med printet fra sidste gang, og bør være færdig efter ca. 1 time. Imens vil Jes gå igang med at få lavet den status, så vi kan komme videre med andre ting.

10:05 Så er vores status endelig færdig, om Michael er ved at ætse printet. Vi vil derfor sætte os ned i fysik, for at komme videre med det praktiske arbejde. Vi

forventer derfor at få lavet latch/delay-delen færdig, samt at få den sidste delay del klar. Således at vi kun mangler programmering af EPROMmen og selve PC programmet.

12:05 Nu virker latch og delay delen. Fejlen fra før var, at chippen var vendt forkert, og at CLEAR skulle trækkes høj, for at den ikke clearer det hele. Ved nærmere eftertestning, ses det at en enkelt flip-flop i vores latch er defækt. Vi skal derfor have en ny. Når den sidste delay del er færdig (snart), skal vi vist lige have en dybere snak om, hvad vi videre gør. Denne snak kunne godt ende med at tage resten af dagen. Eller også kunne vi koble det hele sammen, bare for at se om det virker. Men dette skal vi lige have snakket dybere sammen om.

13:15 Vi har opdaget en fejl. Idet Latch delen læser ind på hver clock, vil vi ikke få gemt den tidligere data. Vi vil derfor hurtigst muligt gå videre og få lavet endnu et print, der kan løse vores problem. Dette består egentilgt bare i en "nøgen" RS-latch, af samme type, som sidder lige efter. Dette burde vi have forudset, men det er altså smuttet lige her. Vi ætser printet idag, så vi er klar til næste gang.

## Påskeferien

Vi satte os sammen i påskeferien, lige for at have det hele klart til den koncentrede uge. Dette fik vi også, sammen med en slagplan for tirsdag. Der skal vi: Brænde EPROM (programmet er skrevet), skifte LM555 ud i delay-kredsen (de opførte sig underligt), teste vores program til PC (lydene er oven i købet klare), folde et chassis og sætte al elektronikken sammen, og putte det på et bræt. Husk også at se efter sådan nogen "huse" til vores LEDer, det ser lidt mere prof ud.

## Tirsdag d. 22 April

Idag nåede vi at få koblet tingene helt sammen, fik lavet starten på chassiet og fik koblet computeren til. Så mangler vi bare lige at bestemme den helt eksakte forbindelse til parallel porten, og så burde det rent faktisk virke... Så burde vi have masser af tid til at gøre det hele smukt.

## Onsdag d. 23 April

Arbejdede i den helt store stil med at få det til at virke idag. Det ser lyst ud.

## Torsdag d. 24 April

Skrev rapport

## Fredag d. 25 April

Fik rettet det sidste til idag. Det hele virker sørme. Dog er der nogen mellemværdier vi godt kunne undvære.