

Radioamatør computer.

Af OZ1BOV Karsten Frahm, Vestertorp 61, 7400 Herning

En computer for de, som vil eksperimentere med en lille styringscomputer.

Historien bag projektet.

Mange af de ideer, jeg går og får om små styringsopgaver, ender som regel ud i at være så store og komplekse, at de næsten ikke lader sig udføre med traditionelle komponenter. De kræver ligefrem at blive løst med en mikroprocessor. Efter at have udtænkt løsningen af opgaven med en mikroprocessor, kommer det altid tilbagevendende problem: printudlægget. En mikroprocessor og tilhørende komponenter kan ofte ikke udlægges på enkelt-sidede print, og da jeg heller ikke kan lave gennempletter et dobbeltsidet print hjemme, skal jeg ud i byen for at få printet lavet, og så bliver det dyrt. Derfor ideen til at lave dette dobbeltsidet gennempletteret print med mikroprocessor og tilhørende komponenter på en ring for alle. Det er så ideen, at interfacekredsløb med optokobler, relæer, display, tastatur, strømforsyning og meget andet laves på enkeltsideprint, eller på et stykke vero-board.

Introduktion.

Lad os lige slå en ting fast med det samme: det er ikke en PC der er tale om, men et lille print med en styringscomputer på. Det er lige som om at digital / computer feberen ikke rigtig er kommet over radioamatørerne endnu. Det tror jeg nu ikke rigtig på, der

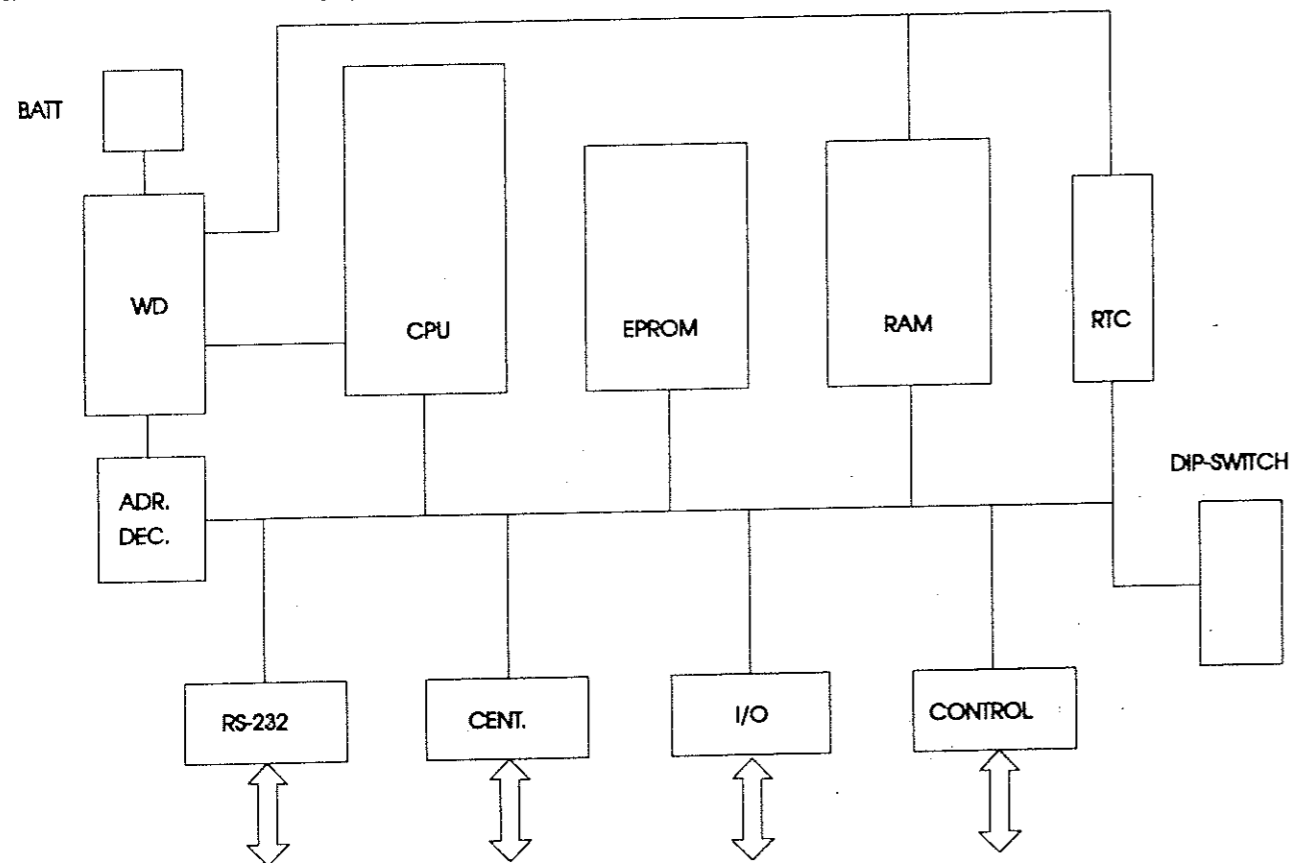
er vel mange der eksperimenterer med C64'er, PC, packet radio og lignende, der bliver bare ikke rigtig noget stof til OZ om dette. Det er også svært, for det man sidder hjemme med, er der vel ingen andre der har interesse for. Det er der nu nok, så derfor tager jeg springet med følgende beskrivelse af hvad jeg kalder et lille mikrocomputer print til amatør brug.

Krav til computeren.

Hvilke krav vil man så stille til sådan en lille computer? Den skal selvfølgelig have så mange input og output som det er muligt, det er jo styringsopgaver, den skal bruges til. En RS-232 kanal, så vi kan kommunikere med andre computere, og en centronic parallel printer port. Batteri backup på memory, så de data vi har i ram'en ikke bliver slettet, når der slukkes for computeren. Real time clock - så man altid ved, hvad klokken er. Watchdog til at overvåge og resette computeren, hvis den skulle blive forstyrret.

Hvad kan den så bruges til?

Jeg vil ikke komme med et færdigt projekt, men overlade det helt til læseren, hvad denne computer kunne bruges til. Jeg har flere gange anvendt den ved afprøvning af små ideer, bla. læsning af stregkoder, test af små software rutiner o. lig. Hvis man vil lære noget om mikrocomputere, er det måske nemmere og sjovere efter princippet: Learning by doing !



Hardware.

Hvad blev det så til ! Det blev en lille computer på 100 · 160 mm² europakort størrelse. Opbygget omkring Intel's 80C31, med plads til 32 eller 64 Kbyte epröm og 32 eller 64 K byte ram, Cmos Real Time Clock, RAM og RTC med batteri back-up, RS-232 seriel port med 300 til 9600 Baud, Centronic parallel printer port, 12 input og 12 output linier, 16 CPU I/O linier og en dip-switch.

Hvorfor blev det så lige en 80C31 fra Intel? Fordi den anvendes af mange professionelle, d.v.s. der er mange, man kan spørge til råds. Ikke mindre vigtig, så er der et firma/klub der har udviklet en pascal compiler til denne CPU. Det betyder, at vi kan komme i gang uden store investeringer i assembler/compiler, ikke helt uvæsentligt når det er hobby.

Tekniske data.

Strømforsyning:	5 Vdc - 30 mA. 5 Vdc - 10 mA i idle mode, med monitor programmet, ellers alt efter hvor stor aktivitet der skal udføres.
Forbrug fra batteri:	Ca. 6 µA, når der er slukket for computeren.
Batteri back-up:	3 Vdc - 175 mAh, lithium.
X-tal:	11,0592 MHz standard, ellers valgfrit.
Print gennempletteret, med komponenttryk.	100 · 160 mm ² dobbeltsidet, gennempletteret, med komponenttryk.
Tilslutning:	Med 26 polet fladkabel eller 25 polet Sub-D stik.

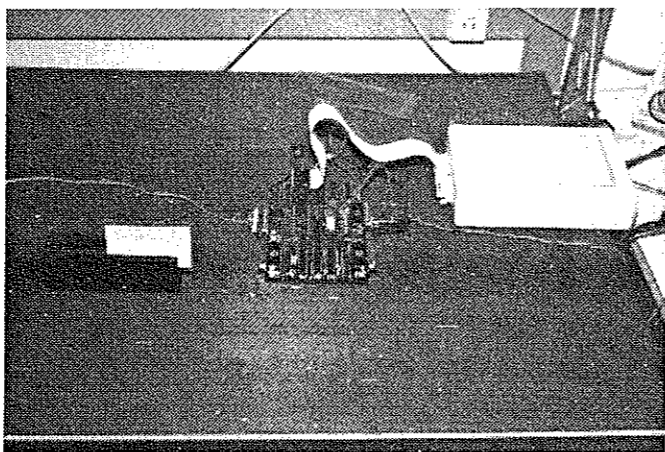
Udstyr krav.

Hvad kræves der så af udstyr for at komme i gang? For at kunne skrive programmer til computeren skal man råde over en PC og et tekstbehandlingsprogram til at skrive selve programmet i. En oversætter af det program man har skrevet, assembler eller compiler, alt efter hvilket sprog, man programmerer i. Mulighed for at programmere EPROM'er med det program, man har lavet. Sidst, men ikke mindst, man skal også kunne slette de EPROM'er man får brændt med et program, der ikke vil virke!

Monitor program.

For at man kan få computeren afprøvet så hurtig og nemt som mulig, har jeg skrevet et lille monitor program. Med det kan man afprøve og se at computeren virker, uden at man skal først til at skrive program. Det er ikke altid let at stå over for nyt hardware og software på en gang. Det monitor program, jeg her har skrevet til computeren, indeholder mulighed for test af alle de ting, der er indbygget.

Tilslutning: Computeren tilsluttes til en terminal eller PC med kommunikations program, 9600 baud,



8 data bit, 1 stop bit, ingen paritet. 1 sek efter computeren tændes, kommer der en hjælpeoversigt frem på skærmen, som vist nedenfor.

Beskrivelse af monitor, oversigt over kommandoer:

- h: **Hjælp**, giver en oversigt over alle kommandoer.
- m: **Udlæser data fra RAM**, 128 byte ad gangen, kaldes med adresse [m0100].
- e: **Udlæser data fra EPROM**, 128 byte ad gangen, kaldes med adresse [e0000].
- c: **Udlæser klokken og dato**, forsætter indtil der sendes en karakter.
- s: **Indstiller klokken**, kaldes med data [sTTMMSS-DDMMÅÅ].
- f: **Fylder ram med data**, kaldes med data [f5A].
- t: **Tester ram** og udskriver højeste ram adresse.
- p: **Sætter i printer mode**, d.v.s. data modtaget på den serielle linie sendes ud på centronic output. ESC returnerer til normal mode.
- u: **Upload programfil fra PC til ram**, Intel hex format.
- b: **Upload programfil fra PC til ram**, binær format.
- w: **Udlæser Dipswitch** stilling.
- o: Sætter i **Power down** mode.
- i: Sætter i **Idle mode**.
- g: Afprøver **Watchdog**.
- r: **Skriv Data til RAM**, kaldes med adresse og data [r0100.3D].

Anvendelse.

Hvad kan man så bruge sådan en lille computer til? Ja, det er faktisk, som altid, kun fantasien der sætter begrænsningerne. Man kunne f.eks. bruge den til at sende morse-koderne til rævesenderen, tiden skal også registreres hver gang en rævejæger finder en ræv, udprintning af aftenens løb ! Hvad med for resten at rævejægeren ikke bruger et DIN-stik som nogle, men bruger et magnetkort, f.eks. dankort? Frekvensudlæsning til synteseradioen. Hvad med et tænd/sluk ur med digital udlæsning? Antennerotor med hukommelse. Repeater styring. Morsetræner, intelligent accu keyer. Data konverter seriel/parallel, en lille printer buffer, buffer for WEFAX signaler.

Ordforklaring.

Watchdog:	Kredsløb der overvåger CPU'en. Watchdog skal have en impuls for hver maximum 100 mS, ellers resettes CPU'en.
RTC:	Real Time Clock - sand tids ur, klokke og dato, der fortsætter med at gå, når der ikke er strøm på printet, batteri back-up skal da være monteret.
Batteri backup:	Lithium batteri der via U5 forsyner RTC og RAM når der ikke er strøm på printet.
Idle mode:	Hvile tilstand, CPU'en stopper al aktivitet, dog forsætter X-tal oscillator og interrupt overvågningen, sparer på strømforbruget. Skal genstartes med et interrupt, og skal sættes i idle mode igen hvis man fortsat ønsker at spare på strømmen. Data i intern ram bevares.
Power down:	CPU'en stopper og slukker for alt aktivitet, X-tal oscillator stoppes. Kan kun genstartes med reset. Data i intern ram bevares.
Compiler:	Oversætter program skrevet i et højniveau sprog, pascal eller C, til maskinkode.
Assembler:	Oversætter program skrevet i mnemonic, en art læsbar maskinkode, til maskinkode.

Her følger så en gennemgang af hardwaren i radioamatør computeren. Hardwaren er ikke beskrevet ned i yderste detalje, men med beskrivelsen prøver jeg at begrunde de tanker og valg, jeg har haft under udviklingen af computeren. I følgende tekst betyder f.eks. ICS - not CS - ikke CS - at signalet er aktiv lav.

CPU, U1.

CPU'en 80C31 kører med multiplexet adresse/data bus, hvilket betyder, at den på signalbenene AD0 - AD7 først sender low byte af adressen ud, og dernæst kommer data. Det kræver, at low byte af adressen gemmes i en 8-bit latch U2; dette styres af ALE signalet. Ud af U2 har vi altså low byte adresse og på CPU benene A8 - A15 har vi high byte adresse. X-tal til CPU'en er på 11,0592 MHz, da dette giver mulighed for at køre med de gængse baud hastigheder på den serielle linie, RS-232. Hvis man skal lave noget, hvor timingen spiller en stor rolle, kan man f.eks. anvende 12.000 MHz krystal, da det er lidt nemme at regne med 1 μ S instruktionstider. Med J17 er det muligt at vælge, om CPU'en skal anvende 8 Kbyte intern program memory, som f.eks. 80C51 eller

8052AH med intern BASIC eller, om den kun skal køre med ekstern EPROM program memory. Default kører CPU'en med ekstern program lager, d.v.s. 80C31 med programmet placeret i EPROM. R1 og R4 er begge pull-up modstande for RESET input på CPU'en og kun en af disse modstande skal monteres. Hvis modstanden R4 monteres kan man forsyne den interne CPU ram med strøm fra batteriet, når der er slukket for printet, d.v.s. gemme interne data i CPU'ens ram, og derved forsætte med de samme data efter en ny power on. Normalt monteres R1. R1 eller R4 monteres, fordi reset kredsløbet ikke alene kan levere strøm nok til reset indgangen, og derved sikrer en korrekt opstart af CPU'en.

EPROM, U3.

Det er muligt at vælge 32 eller 64 Kbyte EPROM. Dette gøres med J1. Er der forbindelse mellem ben 1 og 2, default, skal der anvendes 32 Kbyte EPROM, 27C256-150, er forbindelsen mellem ben 2 og 3, skal der anvendes 64 Kbyte EPROM, 27C512-150. Med J2 er det mulig at bestemme, om CPU'en skal hente programmet fra EPROM'en. Er der forbindelse mellem ben 1 og 2, default, hentes programmet fra EPROM'en. Er forbindelsen mellem ben 2 og 3, hentes programmet ikke fra EPROM'en men fra RAM'en, se endvidere under RAM.

RAM, U4

Her er valgt at bruge en 32 Kbyte RAM som standard. U4 forsynes fra Vbb, så data lagt i ram'en huskes. Hvis det ikke er nok med 32 Kbyte ram, kan man lodde en ekstra 32 Kbyte RAM ovenpå U4. Ud for U4 ben 20 er der en loddeø kaldet J16. Her er ICSHR signalet til den ekstra RAM, man loddede ovenpå U4, hvis man ønsker 64 Kbyte RAM. Ben 20 bøjes lidt ud og ved hjælp af lidt tråd forbindes ben 20 til J16. Denne lidt specielle monteringsform er anvendt på grund af pladsmangel. Med J3 er det mulig at vælge RAM'en ind som program memory. Er der forbindelse mellem ben 1 og 2, default, vil RAM'en virke som data memory. Er der forbindelse mellem ben 2 og 3 vil RAM'en virke både som program memory og som data memory. Det skal lige nævnes, at det er vigtigt, at J2 og J3 står til samme side, og at skiftet sker uden strøm på printkortet. Vil man anvende RAM'en som program memory under program udvikling kan man montere en 2 pol omskifter i stedet for jumper J2 og J3.

RTC, U7

Som real time clock blev valgt Seiko's SMC 5242C. Den forsynes fra Vbb, hvorved den skulle fortsætte selv om der bliver slukket for printet. Med J4 er det mulig at sende et interrupt signal til CPU'en. SMC 5242C kan programmeres til at afgive interrupt signalet med 1/64 sek, 1 sek, 1 minut eller 1 times

interval. Det kan bruges til at lade CPU'en gå i stand-by, Idle mode, for at spare på strømmen. Så lader man RTC'en genstarte CPU'en en gang imellem, for lige at se om noget har ændret sig, og hvis så noget har ændret sig, så kan CPU'en svare med et eller andet. RTC'en har en 4 bit adresse og data bus. Det betyder at man skal hente 4-bit data adgangen. Skal man f.eks hente minutter, henter man en'er minutter i register 2 og ti'er minutter i register 3, begge som BCD tal.

Watchdog, U5.

Her blev valgt en MAX 691, der indeholder supply monitor, watchdog, batteri backup switch, gate for CS signaler til ram. Reset signalet til CPU'en bliver først frigivet, d.v.s. går lav og CPU'en starter, når Vcc kommer op over 4,65 Vdc. Herefter har man 1,6 sekund inden watchdog'en skal resettes igen, og derefter skal watchdog'en resettes indenfor maksimum 100 mS, ellers vil den resette CPU'en. Watchdog'en resettes ved at skrive til adresse FFFCH; hvilke data, der skrives med, betyder ingenting, det er skrive adresseringen, der resetter watchdog'en. Med jumper J18 kan man afbryde reset signalet til watchdog input, indgangen kommer derved til at svæve, watchdog'en frakobles og programmet behøver ikke at skulle skrive til adresse FFFCH. Nyttig ved afprøvning af nyt program. Kommer forsyningsspændingen under 4,65 Vdc, resettes CPU'en, og det forhindres, at der kan skrives til RAM via de interne gates i MAX691 og til RTC via BATT ON output fra MAX691, samt gaten U12B. RAM'en og RTC'en forsynes fra Vcc, indtil denne kommer under Vbb, derefter forsynes fra batteriet.

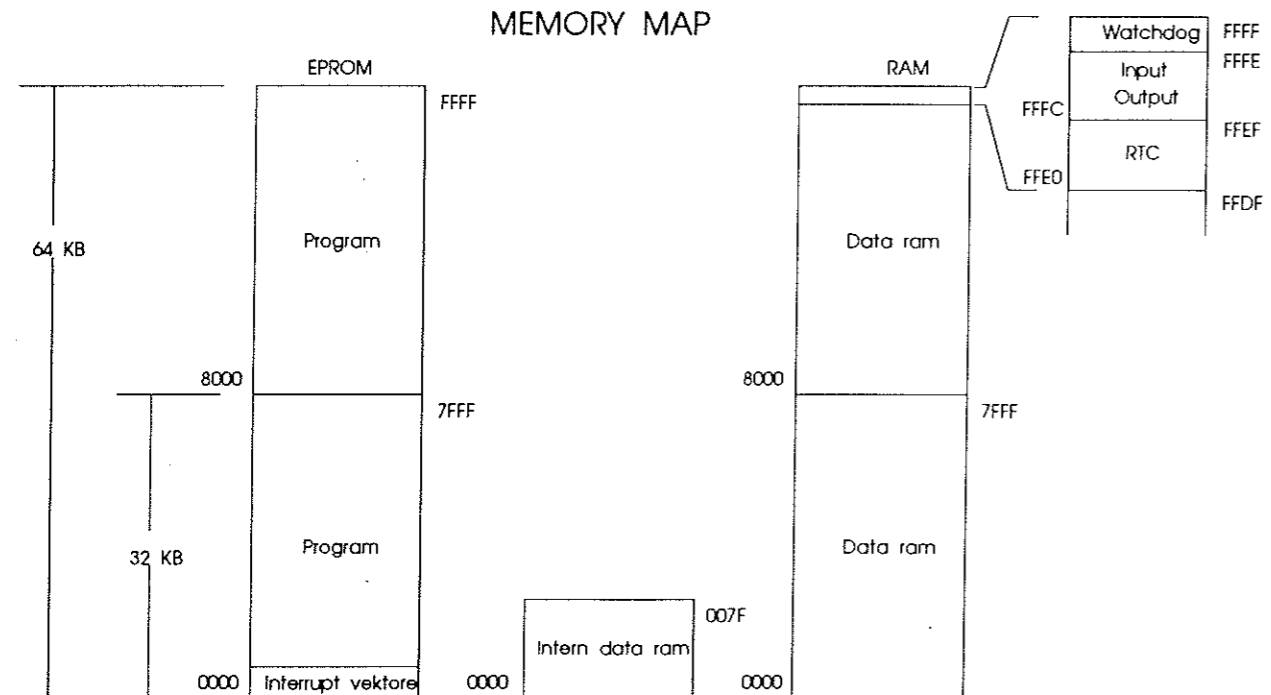
Adressering

Adresse dekodingen er lidt af et kapitel for sig. 80C31 anvender 2 eksterne parallel memory banke, en til program memory og en til ekstern data memory. Herved kan vi have 64 Kbyte program memory og 64 Kbyte data memory, med en 16 bit adresse bus. Program memory skal altid starte fra adresse 0000H, da de forskellige interrupt vektorer starter fra adresse 0000H til ca 0025H alt efter CPU type. Program memory starter altså i adresse 0000H og går til adresse 7FFFH hvis der er monteret 32 Kbyte EPROM, eller til FFFFH hvis der er monteret 64 Kbyte EPROM. RAM'en er adresseret således, at første RAM ligger i området fra 0000H til 7FFFH. Hvis der loddes en RAM ovenpå den første, kommer den til at ligge i adresseområdet fra 8000H til FDFH. Vi kommer altså til at mangle ram adresserne fra FFE0H til FFFFH i den øverste RAM. Adresseområdet fra FFE0H til FFEFH bruges til at selektere RTC'en, så vi kan komme i kontakt med de interne registre i RTC'en. Register 0 får altså adresse FFE0H og register 15 får adresse FFEFH. På U10 ben 8, har vi signalet der fortæller at vi er i adresse området fra FFFCH til FFFFH. Dette signal sammen med A0, A1 og IWD føres til U 14, der adresserer vores input / output samt watchdog.

Adresse. Read/Write.
 FFE0H - FFEFH Læs / skriv til registre i ur.

Adresse.	Write.
FFFCH	Reset watch dog.
FFFDH	U20, Output O10 - O17.
FFFEH	U18, Output data til Centronic, O30 - O37.
FFFFH	U19, Output O20 - O25, Centronic control output O26 - O27.

MEMORY MAP



Adresse.	Read.
FFFCH	Bruges ikke.
FFFDH	U16, Input I10 - I17.
FFFEH	U17, Læs dip switch på print.
FFFFH	U15, Input I20 - I25, Centronic control input I26 - I27.

Adresse dekodingen kunne være udført med PAL'er, men jeg og mange andre råder ikke over en PAL brænder, så derfor blev det med helt almindelige kredse. U12 forsynes fra Vbb. Det er nødvendig, for at holde ICS til RAM og RTC høj, når der ikke er strøm på printet. Hvis ICS til RAM og RTC går lav, selekterer man disse kredse, og hvis adresse bus'en er lav, det er den når der ingen strøm er på printet, vil man ødelægge de data der er i adresse 0000H i RAM'en, byten overskrives med 00H.

Digital input.

Datainput læses via 74HCT373, og indgangskredsløbene på kredsen bliver trukket lav med 100 kohm, for at være sikker på at ingen efterlader dem svævende. Når man læser fra en af indgangs adresserne, læser man 8 bit ind adgangen. Indgangs strømmen i en 74HCT373 er max $\pm 1 \mu A$, $V_{in} = 0$ til V_{cc} .

Digital output.

Til output er der anvendt 74HCT373. Her skriver 8 bit data udgangen til den aktuelle adresse, og data bliver sendt ud på udgangene med det samme. Der er ingen mulighed for at læse, hvad der rent faktisk står på udgangene, så hvis man i programmet senere skal bruge den aktuelle udgangsstilling, er det nødvendigt at man selv gemmer stillingen i ram. Selv om input og output ligger på samme adresse, har de ingen forbindelse, d.v.s. udgangene kan være lav og indgangene kan være høj. Udgangene kan maksimum belastes med ± 35 mA.

RS-232, U6.

Her bruger vi en MAX-232, der kun skal forsynes med 4 kondensatorer og + 5 Vdc, så laver den selv ± 9 Vdc til RS-232. Tx output fra CPU'en føres til U6 og ud af U6 kan vi så hente TXD, det RS-232 kompatible signal. Modtaget RS-232 signal føres til U6 og ud af U6 kan vi så hente Rx signalet til CPU'en. CPU port 3.4 føres via jumper J5 til U6, og ud fra U6 fås output kontrol signalet RTS. Forbindes J5 ben 2-3 fås en ekstra seriel printer udgang, til brug ved basic computeren. Input kontrol signalet CTS føres til U6 og forbindes via J6 til CPU'en port 3.3. Alle signaler inverteres i MAX232. Med X-tal oscillator på 11.0592 MHz kan man vælge mellem følgende standard hastigheder: 300, 600, 1200, 2400, 4800, 9600 baud.

Rs-232 tilslutning:

Computer	Pc - 9 pol	Pc - 25 pol
2 - TxD	2 - RxD	3 - RxD
3 - RxD	3 - TxD	2 - TxD
7 - GND	7 - GND	7 - GND

Batteri backup

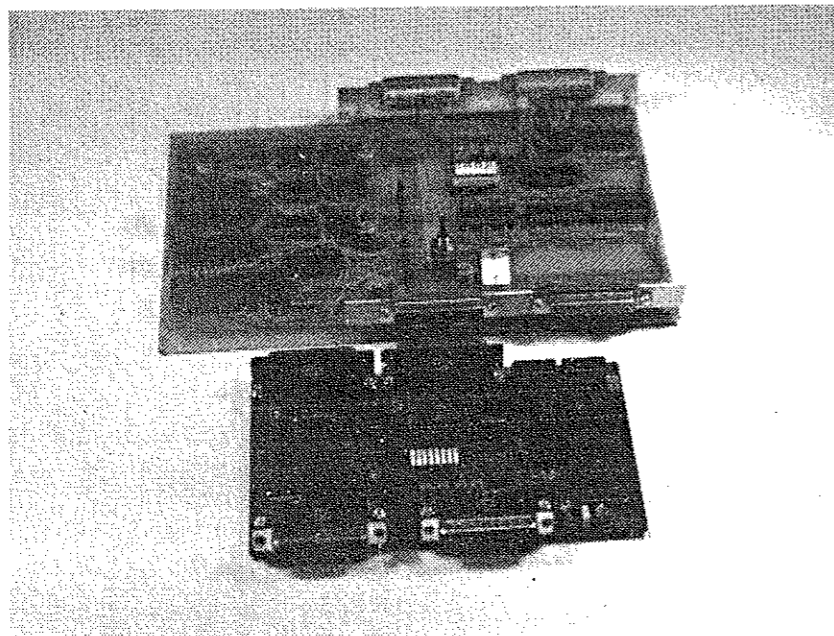
Lithium celle på 3 Vdc - 175 mAh. Med et målt forbrug på $3 \mu A$, giver det $175 \text{ mAh} : 3 \mu A = 58.333$ timer eller 6 1/2 år, hvor der kan være slukket for computeren, uden at data eller klokken går tabt.

Dip switch.

Laver man en styringscomputer uden tastatur og display, kan man lave programmet således, at man v.h.a. dipswitchen kan vælge forskellige parametre, f.eks baud hastighed, en ud af flere funktioner og lignende.

Centronic port

Den parallelle Centronic port er opbygget med 74HCT373 kredse, og er afprøvet med en STAR LC24-10 printer. Der er serie modstande i alle output



ledninger, for at begrænse strømmen i output kredse, hvis printeren er tilsluttet og slukket.

Tilslutning

Man kan vælge mellem to typer stik, 26 polet fladkabel eller 25 polet D-sub stik. Begge typer stik kan dog monteres samtidig. Det 26 poledede fladkabel er praktisk, hvis computerprintet skal tilsluttes et andet printkort. Det 25 poledede D-sub stik er godt hvis man skal eksperimentere med forskellige ting, det er nemt at tilslutte og bytte om på forbindelserne. RS-232 forbindelsen er dog kun lavet med D-sub stik. Vil man have monteret D-sub stik i chassis eller forplade for ekstern tilslutning, er det nemmest at have et 26 polet fladkabel stik på printet og så forlænge med fladkabel til et D-sub stik med fladkabel tilslutning. Bennumrene på fladkabel stikket er lagt ud således, at der ikke ændres på stik nummerer på D-sub stikket ved denne forlængelse.

Jumper.

Oversigt over jumper på printet. Default forbindelsen er på printet, så ønsker man ikke en anden, behøves der ikke at blive monteret stiftlist og jumper. Default forbindelsen er på loddensiden af printet, så det skulle være mulig at afbryde forbindelsen, selv med alle komponenter monteret på printet.

J 1:	1 - 2	Default	Enable 32 KB eprom. 27C256.
	2 - 3		Enable 64 KB eprom, 27C512.
J 2:	1 - 2	Default	Enable eprom som program memory.
	2 - 3		Disable eprom som program memory.
J 3:	1 - 2	Default	Ram memory som data memory.
	2 - 3		Ram memory som data og program memory.
J 4:	1 - 2		Enable interrupt fra RTC til IINT1.
J 5:	1 - 2		Enable T0 output til RS-232 output !RTS.
	2 - 3		Enable P1.7 til RS-232 output !RTS seriel printer output ved basic computer.
J 6:	1 - 2		Enable IINT0 interrupt til RS-232 input
J 7:	1 - 2		Centronic !ACK til IINT1 input.
J12:	1 - 2	Default	Batteri tilslutning.
J16:	1 - 2		Chip select til high 32 KB ram.
J17:	1 - 2	Default	!EA forbindes til GND, CPU'en køre med ekstern program lager.
	2 - 3		!EA forbindes til + 5 Vdc. De første 8 KB program udføres fra den intern eprom. CPU P8051AH-BASIC.

J18: 1 - 2 Default Watchdog input til MAX691 fra U14, hvis ingen forbindelse er watchdog frakoblet.

Ordforklaring

Supply monitor:	Overvågning af forsyningsspænding. Resetter cpu'en hvis spændingen går under 4,65 Vdc.
Chip select:	Signal til udvælgelse af et bestemt kredsløb. f.eks ram, eprom, rtc og ligene.
Interrupt vektor:	Et interrupt er en udefra kommende hændelse, der kræver at igangværende program afbrydes, og at der afvikles et program som tager højde for hændelsen. For at CPU'en ved, hvor programmet den skal udføre er, hopper den til en bestemt adresse, kaldet interrupt vektor. Når interrupt programmet er udført, returneres til det før igangværende program.

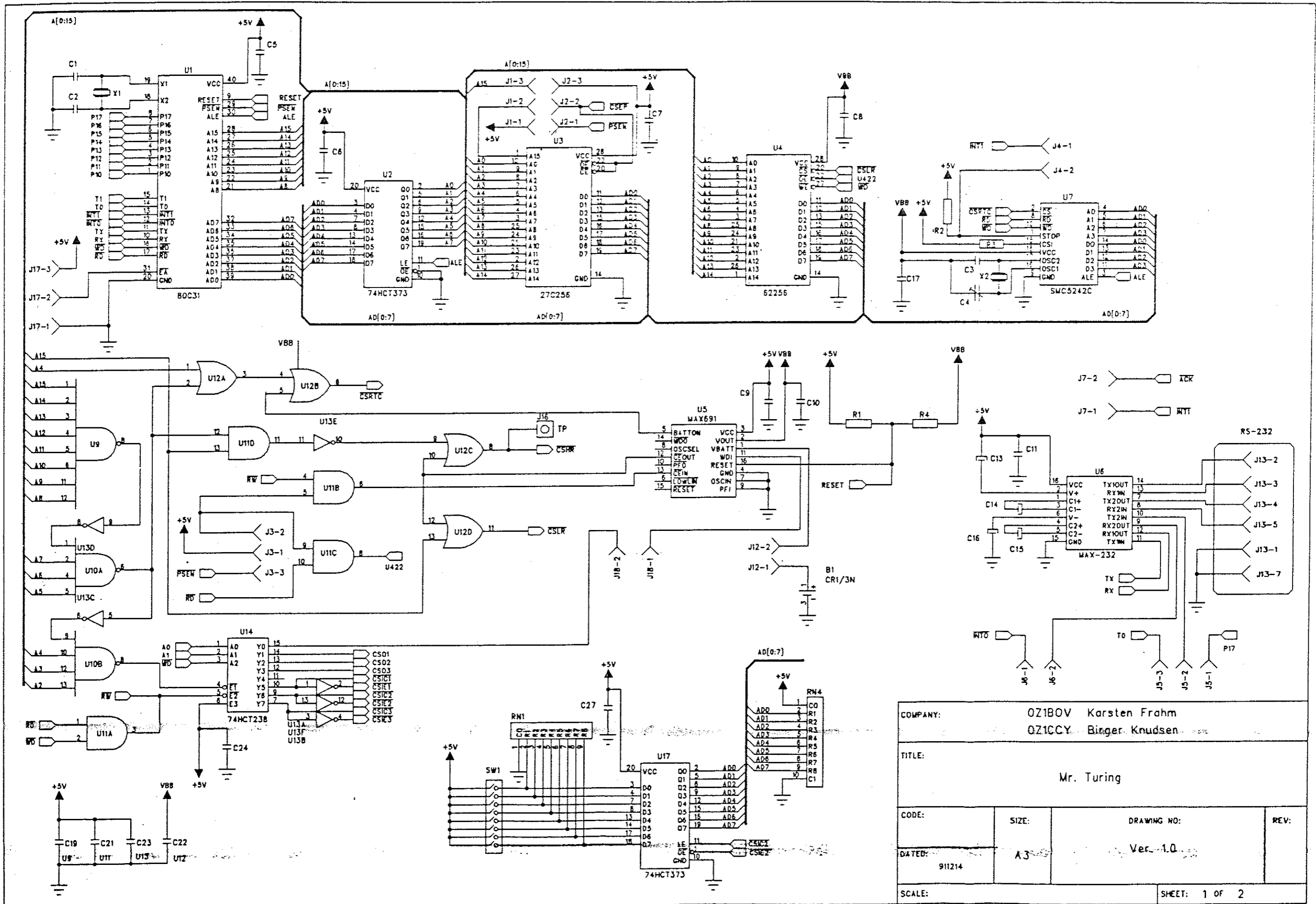
Interesseret ? så har jeg ekstra print og komponenter liggende.

Efter hardware beskrivelsen kommer her en kort gennemgang af forskellige programmeringsværktøjer til amatørcomputeren. Derefter en kort gennemgang af monitor programmet som jeg har skrevet til computeren.

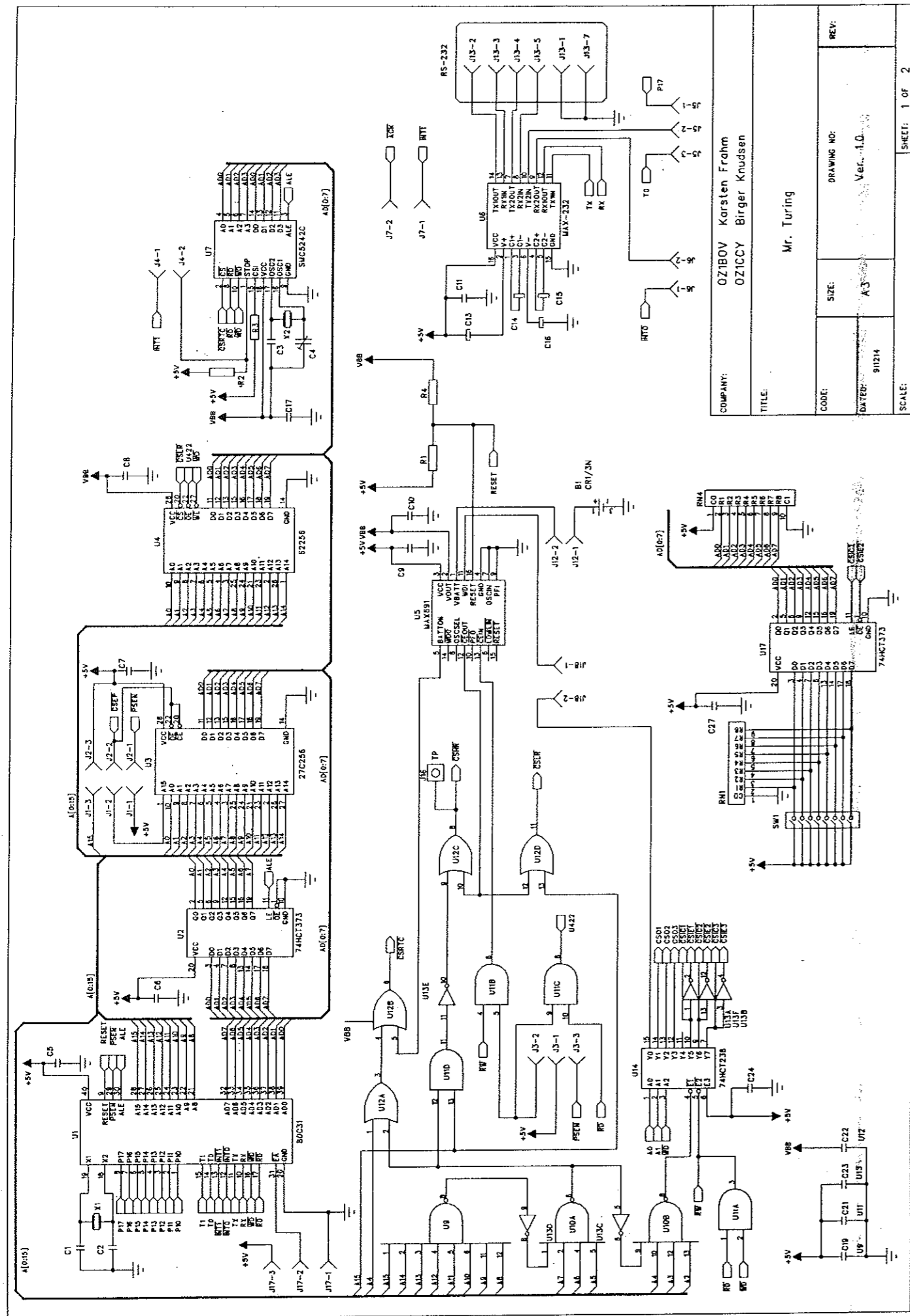
EPROM programmer og sletter.

Når man nu laver programmer til computeren, og disse virker efter hensigten, skal man have mulighed for at brænde en EPROM med det færdige program. Upload funktionen er tænkt som en hjælp i starten, programmerne kan udvikles og af testes uden EPROM programmer, og når man så har et program der virker, kender man nok en eller anden der kan hjælpe med programmeringen af EPROM'en. Det fås flere forskellige typer EPROM-brændere, nogle med egen strømforsyning, hvor man via RS-232 overflytter data til denne og så foretager brændingen af EPROM'en. Anden type består af et kort til at sætte i sin PC og en ekstern boks med EPROM sokkel. V.h.a et medfølgende program kan man så direkte brænde en EPROM med det program man har liggende på harddisken. Sidste omtalte type ligger i et prisleje der bedst passer til hobby brug. Man kunne jo selvfølgelig også selv lave en, hvor computeren her sørger for at styre programmeringen af EPROM'en.

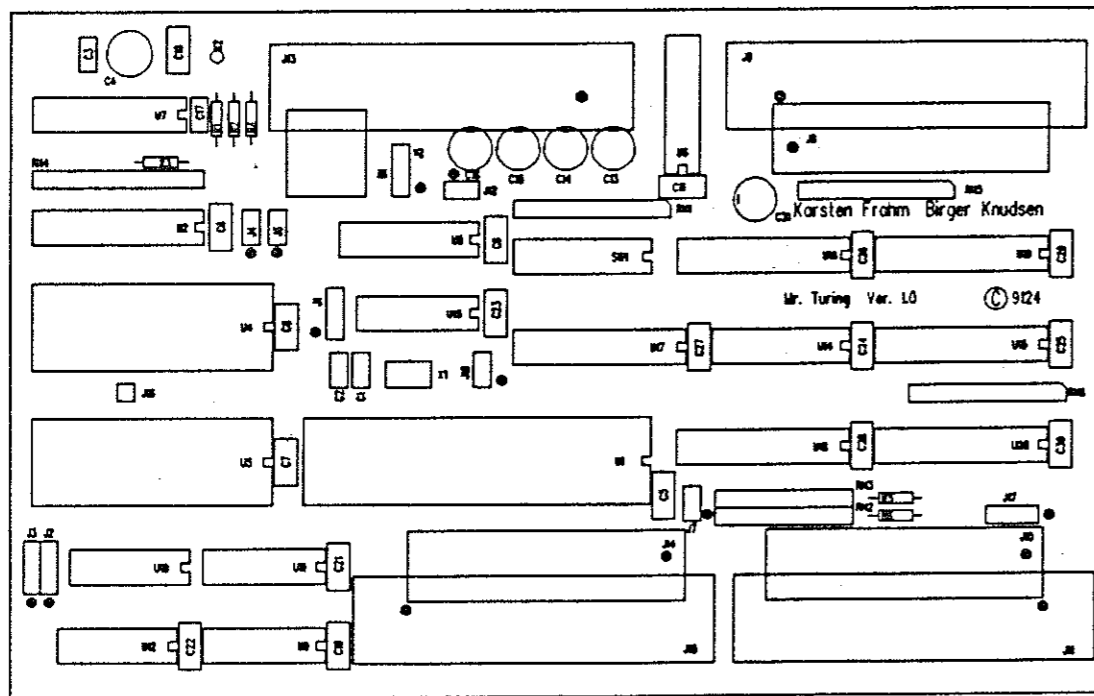
Sidst men ikke mindst, man skal kunne slette de EPROM'er man får brændt med et program der ikke vil virke. Det gøres med en epromsletter, der indeholder en UV-lampe til sletning af EPROM'en. Her kan evt. anvendes en højffeldssol; EPROM'en skal i så



COMPANY:				OZ1BOV Karsten Frahm			
				OZ1CCY Binger Knudsen			
TITLE:				Mr. Turing			
CODE:	SIZE:	DRAWING NO:		REV:			
DATED:	911214	A3	Ver. 1.0				
SCALE:				SHEET: 1 OF 2			



COMPANY: OZIBOV Korsten Frahm
 OZICCY Birger Knudsen
 TITLE: Mr. Turing
 CODE: DRAWING NO: Ver. 1.0
 DATE: 9/12/14
 SCALE: 1 OF 2



SILKSCREEN TOP SIDE BK PCB DESIGN 74 45 45 23 Mr. Turing Ver. 1.0 9124

tilfælde placeres tæt, 5 cm, fra lampen og belyses i ca. 15 minutter. Pas på øjnene med UV-stråler.

Programmerings software

En computer kan jo ingen ting uden et program. For at kunne skrive et program kræver det, at vi har valgt, hvilket programmerings sprog vi vil anvende. Jeg har i det følgende prøvet at beskrive fordele og ulemper ved nogle af de forskellige programmeringssprog som kan anvendes til denne computer.

Assembler.

Jeg har en public domain assembler til 8051. Jeg har ingen erfaring med den, men den skulle virke efter de almene principper for assemblere, d.v.s. Intel syntaks. Programmeringen kræver en god teknisk viden af programmøren omkring netop den CPU der anvendes. Det er ikke noget jeg vil anbefale til nybegynder, men det bør prøves - det er meget interessant. Jeg programmer meget i assembler, da jeg synes at der ligger store udfordringer i at skrive programmer så hurtige og små som mulige. Skal man skrive et program til f.eks en stregkortlæser, kommer man nok ikke uden om assemblerprogrammering, da læsningen at stregkort foregår i rimeligt tempo.

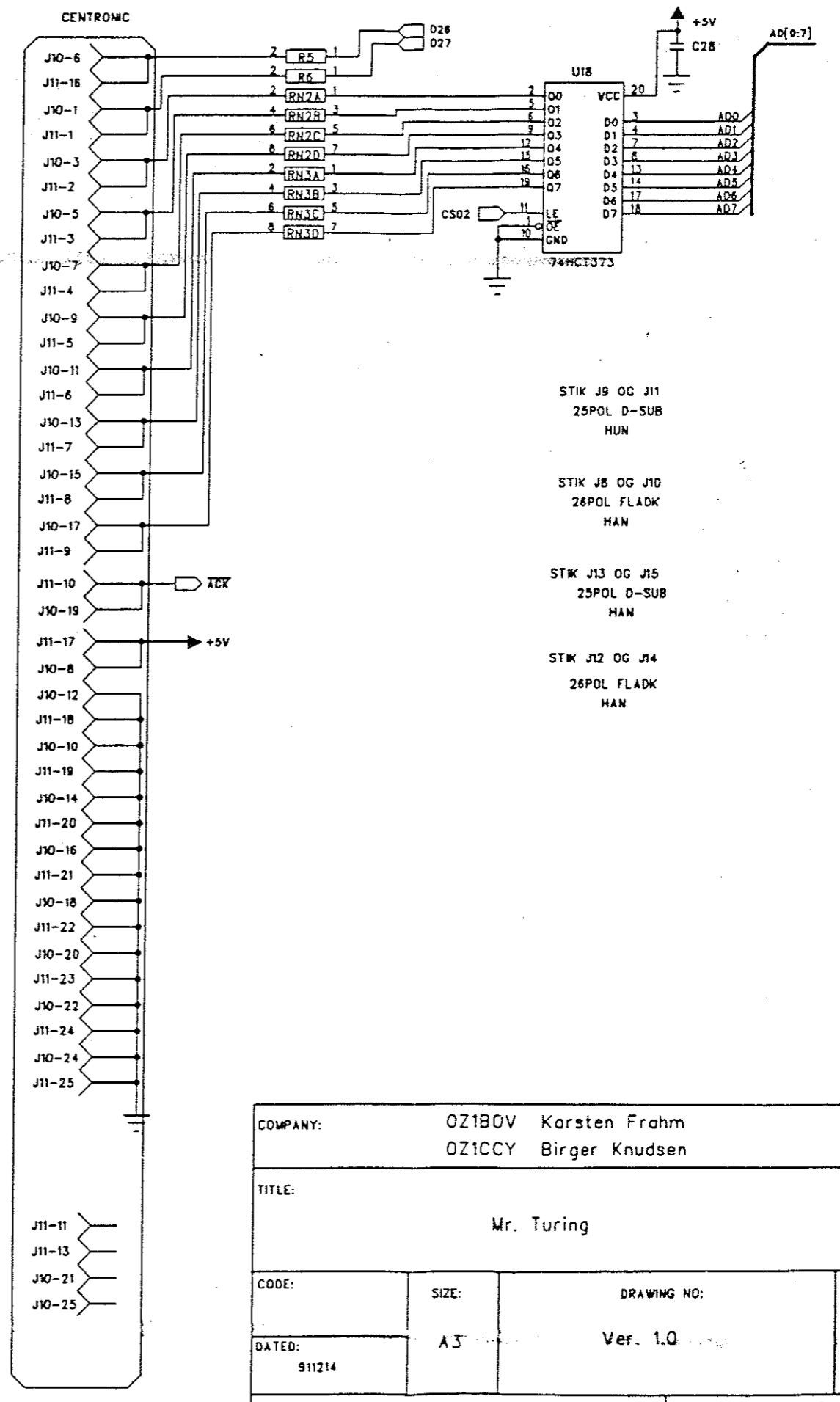
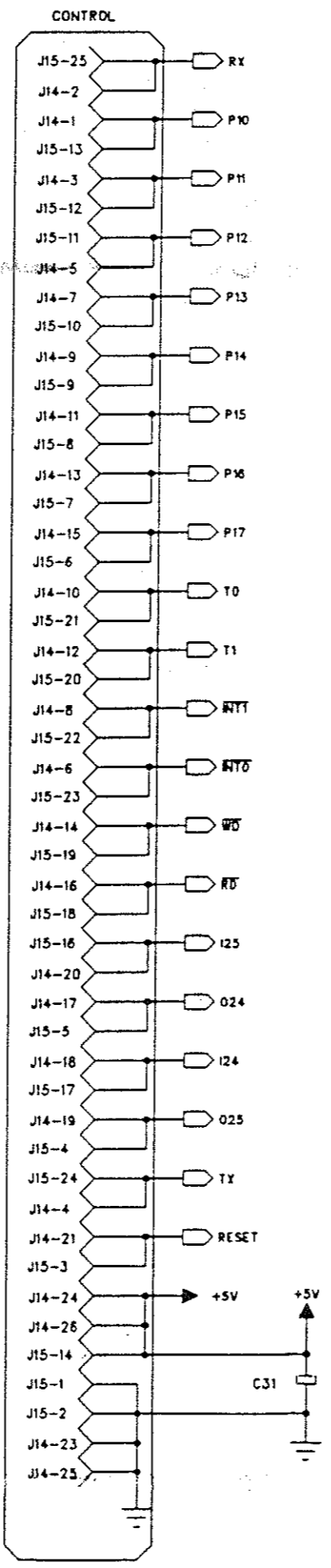
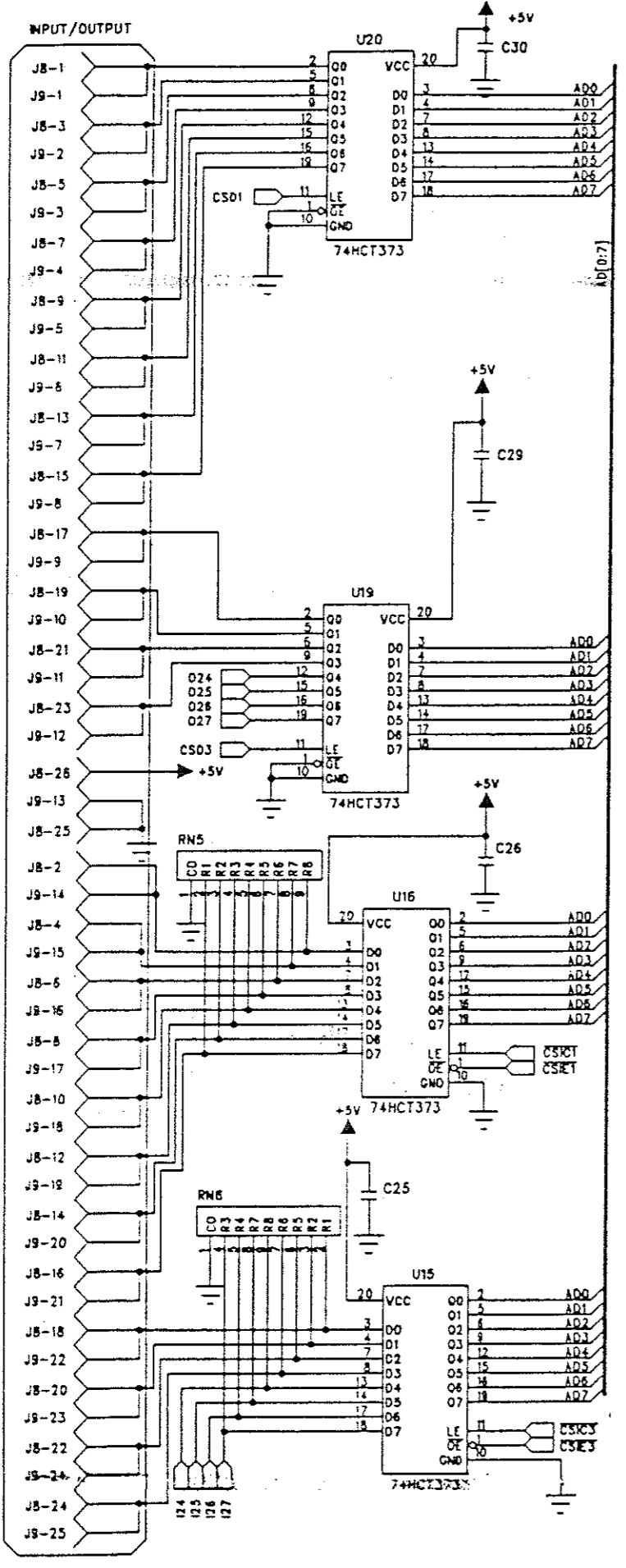
Pascal

Man kan hos Circuit Design Aps købe en diskette med en pascal compiler, der er udviklet af Circuit Design. Disketten hedder PCPAS51. Et fint og stort stykke arbejde er der lagt i denne pascal compiler, og prisen er kun 298,- kr. i modsætning til en professionel pascal compiler, der nemt koster 7.000,- kr. Hvis

man vælger at anvende Circuit Design's compiler vil jeg henstille til alle, at man køber sit eget program, ikke noget med at bruge en sikkerhedskopi, ellers ødelægger vi muligheden for at i fremtiden at kunne købe billige og gode programmer hos CD. Har man prøvet at skrive programmer i pascal til sin PC, er det ikke noget problem at skrive et styringsprogram til denne computer. Forskellen mellem pascal til en PC og Circuit's pascal til denne lille computer er, at Circuit's pascal ikke indeholder reals, strings, sets og pointers, samt alle specielle funktioner og procedurer så som skærmstyring, filhåndtering og lignende, der følger med en stor pascal. Ingen problem, man må så bare selv skrive de manglende funktioner og procedurer.

C.

Monitor programmet er skrevet i C, da det er det højniveau-sprog jeg anvender til daglig. Her kender jeg desværre ikke nogen public domain compiler, er der nogen der gør, hører jeg det meget gerne. Kan man programmere i pascal kender man også sproget C. C har ikke flere end 26 funktioner, så C er nemt at lære. Forskellen mellem C og pascal er, at med pascal går man med livrem og seler, mens at man med C selv må holde bukserne oppe med den ene hånd. Det betyder, at pascal kontrollerer programmet for noget som ikke kan lade sig gøre, mens C oversætter programmet, og så må man selv om, hvad man har skrevet! Fordelen ved C fremfor pascal er, at man i C kan opbygge et bibliotek af assembler-rutiner, der kan kaldes direkte fra C. Disse assembler-rutiner skrives i en editor og assembleres og derefter



STIK J9 OG J11
25POL D-SUB
HUN

STIK J8 OG J10
26POL FLADK
HAN

STIK J13 OG J15
25POL D-SUB
HAN

STIK J12 OG J14
26POL FLADK
HAN

COMPANY:		OZ180V Karsten Frahm OZ1CCY Birger Knudsen	
TITLE:			
Mr. Turing			
CODE:	SIZE:	DRAWING NO:	REV:
DATED:	A3	Ver. 1.0	
911214			
SCALE:	SHEET: 2 OF 2		