

CODEN: LUTFD2/(TFRT-5449)/1-24/(1992)

Utveckling av system för registrering av andningsaktivitet

Krister Tham

Institutionen för Reglerteknik
Lund Tekniska Högskola
Januari 1992

**TILLHÖR REFERENSBIBLIOTEKET
UTLÅNAS EJ**

Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden		<i>Document name</i> MASTER THESIS	
		<i>Date of issue</i> January 1992	
		<i>Document Number</i> CODEN: LUTFD2/(TFRT-5449)/1-24/(1992)	
<i>Author(s)</i> Kristers Tham		<i>Supervisor</i> Lars Gösta Hellström (KI), Dag Brück	
		<i>Sponsoring organisation</i>	
<i>Title and subtitle</i> Utveckling av system för registrering av andningsaktivitet (Developing a system for measuring respiratory drive)			
<i>Abstract</i> <p>The thesis describes the development of software for a computer controlled measuring system, which calculates the respiratory drive and some additional breath parameters. The respiratory drive, i.e., a signal from breath center to the lung, is estimated by calculating the airway pressure drop. The software is based on an interrupt driven structure designed for IBM PC and compatibles.</p>			
<i>Key words</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i>			<i>ISBN</i>
<i>Language</i> Swedish	<i>Number of pages</i> 24	<i>Recipient's notes</i>	
<i>Security classification</i>			

Innehåll

1 Inledning	2
2 Medicinsk bakgrund	3
3 Syfte med examensarbetet	4
4 Beskrivning av befintlig utrustning	5
5 Definition och beräkning av parametrar	6
5.1 Definition och beräkning av "P _{0.1} "	6
5.2 Beräkning av inspirationsvolym	7
5.3 Definition av koldioxidkoncentration	8
6 Programutveckling	9
6.1 Foreground/Background - scheduler	9
6.2 Abstrakt datatyp - SAMPLPAC	12
6.3 Abstrakt datatyp - COMPAC	15
6.4 Huvudprogram - RESPUNIT	16
7 Utprovning	18
7.1 Validering av andningsparametrar	18
7.2 Beräkning av maximal samplingshastighet	19
7.3 Test av utrustning vid återändring	20
8 Utvecklingsmöjligheter	21
8.1 Förbättring av kalibreringsrutin för flödesgivare	21
8.2 Beräkning av processorkapacitet	22
8.3 Maximering av samplingsfrekvens	23
8.4 Datorstyrd öppning av ocklusionsventil	23
8.5 Statistiska analyser av inhämtad andningsdata	24
9 Referenser	24

1. Inledning

På institutionen för medicinsk teknik på Karolinska institutet har ett mätsystem för mätning av andningsfunktion utvecklats. Systemet ska användas för forskningsändamål och även kliniskt ute i sjukvården. Mätutrustningen består av ett specialutvecklat munstycke och ett antal analoga tryckgivare som via ett AD-kort är anslutna till en dator.

Syftet med examensarbetet är att utveckla programvara för mätdatainsamling och styrning av mätutrustningen. För att kunna mäta de relativt högfrekventa förlopp som uppkommer vid andningsregistrering krävs en hög samplingsfrekvens. För att åstadkomma detta har programvaruutvecklingen baserats på en enkel realtidskärna, "Foreground/background scheduler".

Examensarbetet har resulterat i en programvara där de analoga signalerna läses in, behandlas och presenteras i realtid. Inläsningsfrekvensen av signalerna är 500 - 2000 Hz, beroende av aktuell processorkapacitet. En relativt stor del av arbetet har lagts ner på att uppnå ett enkelt och lättbegripligt användargränssnitt. Programvaran har provkörts och utvärderats tillsammans med den analoga mätutrustningen.

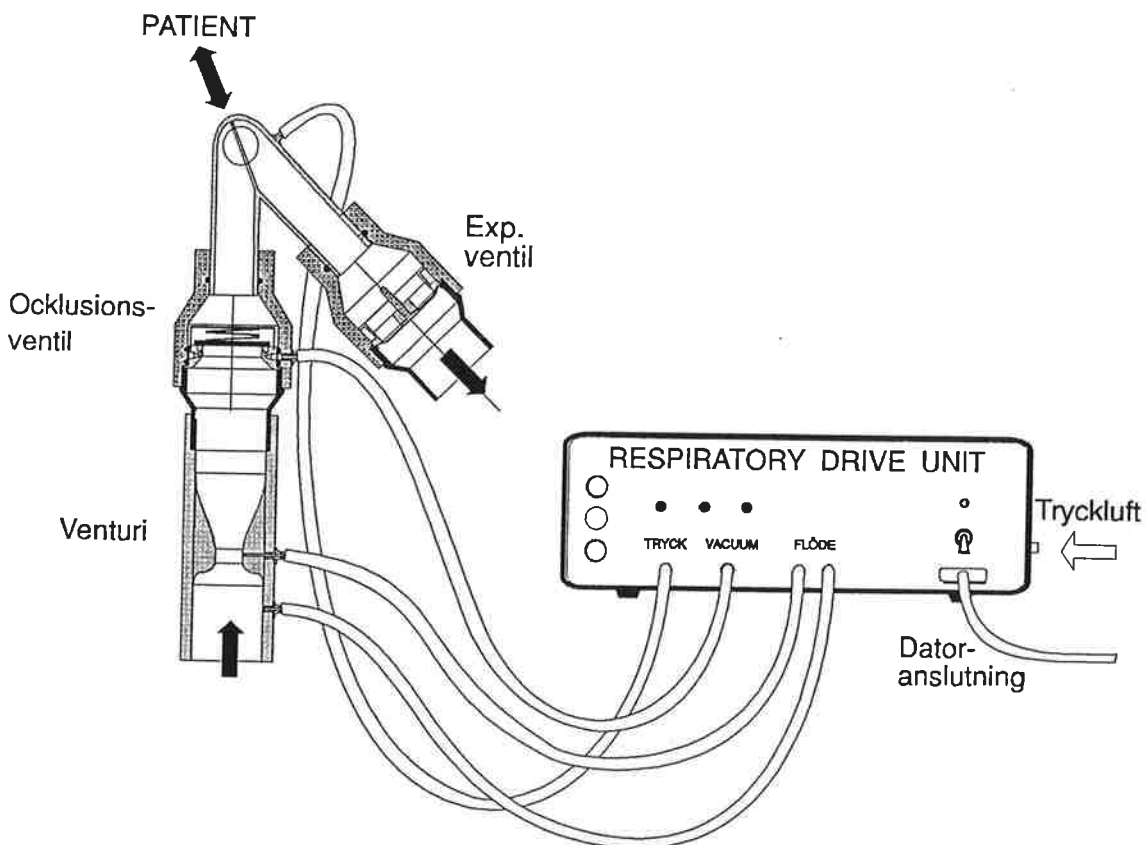


Bild 1.1 System för registrering av andningsfunktion

I kapitel 2 ges en översiktlig medicinsk bakgrund till projektet. Därefter definieras i kapitel 3 syfte och målsättning för examensarbetet. Den analoga mätutrustningen beskrivs i kapitel 4. I kapitel 5 ges en utförlig beskrivning av de parametrar som ska beräknas. Här ges också en kort beskrivning av hur beräkningsalgoritmerna är implementerade i programvaran. Kapitel 6 syftar till att ge en översiktlig bild av programmets uppläggning och struktur. I detta kapitel ges även en kortare beskrivning av funktion och gjorda justeringar av "Foreground/background scheduler". Resultat från utprovning och tester av mätutrustningen beskrivs i kapitel 7. Därefter ges en del förslag på förbättringar och utvecklingsmöjligheter av programvara och mätutrustning i kapitel 8.

2. Medicinsk bakgrund

Andningsregleringen är en viktig del av den basala hjärnfunktionen. Vid såväl medicinsk behandling som vid forskning finns ofta ett behov av att objektivt kunna bedöma andningscentrums funktion. Exempel på sådana områden är forskning kring plötslig spädbarnsdöd, sömnapné syndrom (andningsuppehåll under sömn) och effekter av morfinadministrering.

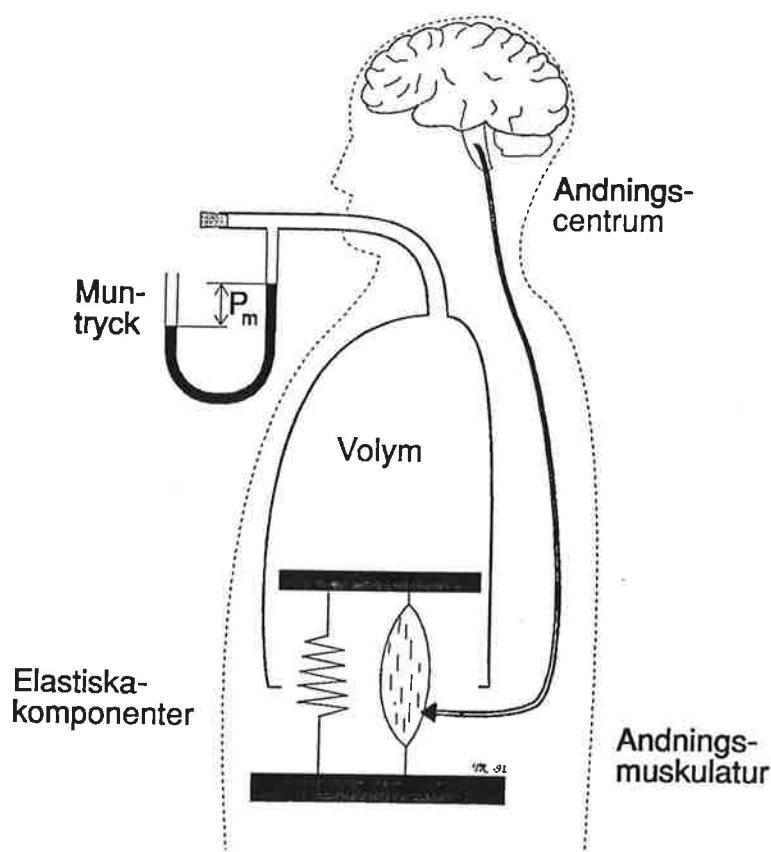


Bild 2.1 Modell för mätning av "P_{0.1}"

En teknik för att avgöra andningscentrums funktion är s.k. ocklusionstryckmätning. Metoden går ut på att studera tryckfallet i andningsvägarna under en inandning. Tryckfallets hastighet kan sedan indirekt ge en uppskattning av den centrala inspiratoriska aktiviteten. Tryckfallshastigheten benämns i dessa sammanhang "P_{0.1}".

För att ett tryckfall skall uppstå måste andningsvägarna under ett kort ögonblick vara stängda - ockluderade. Ocklusionstiden måste dock vara så liten att den inte påverkar andningsrytmen. I vissa situationer kan en ocklusionstid vara så kort som fem millisekunder. För att under detta tidsintervall kunna inhämta tillräcklig information krävs därför en snabb samplingsfrekvens.

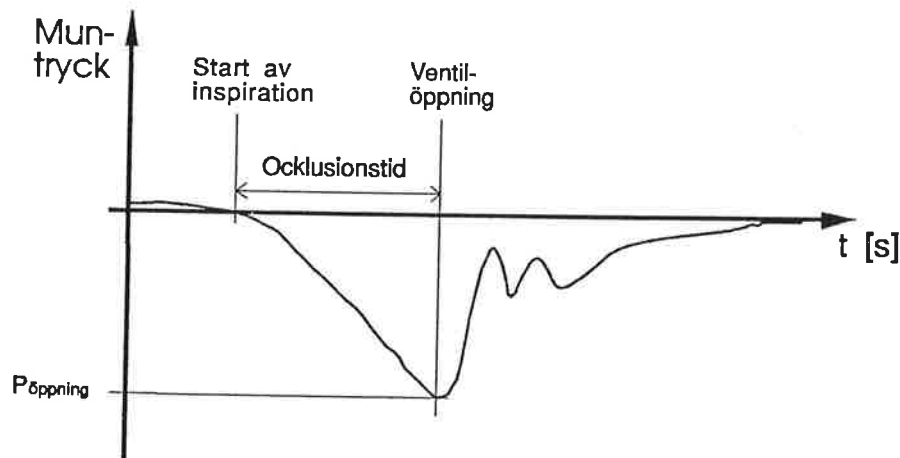


Bild 2.2 Tryckkurva med ocklusion.

En kliniskt användbar metod för mätning av andningsregleringen förutsätter ett integrerat system för kontinuerlig övervakning av olika parametrar såsom P_{0.1}, inandnings- och utandningsvolym, inandnings- och utandningstid, koldioxid koncentration, syrgasmättnad m.m .

3. Syfte med examensarbetet

Syftet med examensarbetet är att utveckla en lämplig programvara för insamling och presentation av data från mätutrustningen. En målsättning är att utrustningen ska kunna användas för forskningsändamål och kliniskt ute i sjukvården.

På den färdiga programvaran ställs följande krav:

- * Ett antal analoga signaler skall samplas med minst 1000 Hz.
- * Insamlade data skall kontinuerligt behandlas och presenteras för användaren.
- * Beräknade parametrar skall kunna lagras på en fil. På den aktuella filen skall också patientdata såsom namn, personnummer, datum mm kunna lagras.
- * Kalibrering av mätutrustning och AD-kort skall kunna göras.
- * Datainsamling skall baseras på en avbrottsstyrd programvara.

- * Programvaran skall skrivas i C.
- * Programvaran skall vara kompatibel med olika PC-datorer med olika processorkapacitet.
- * Eftersom målsättningen är att utrustningen ska kunna användas av sjukvårdspersonal skall utformningen av användargränssnittet vara enkel och lättbegriplig.

I examensarbetet ingår också utprovning och utvärdering av programvara och analog mätutrustning.

4. Beskrivning av befintlig apparatur

Systemet är uppbyggt kring ett speciellt munstycke. I detta munstycke sitter en s k ocklusionsventil (bild 4.1). Ocklusionsventilen består av ett plastmembran som sugs fast av ett utifrån förinställt vakuum.

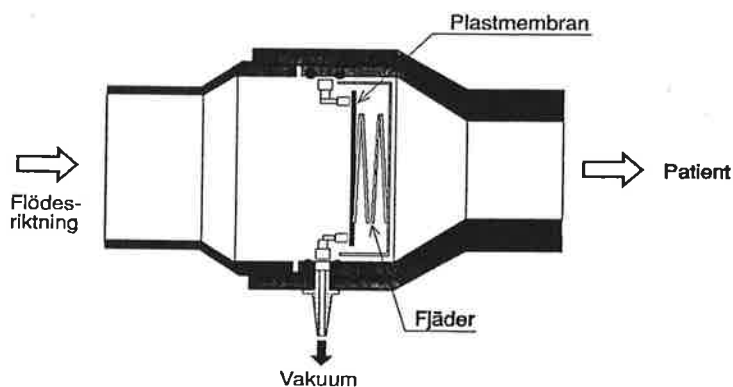


Bild 4.1 Ocklusionsventil

Ventilen är normalt stängd, men vid ett visst undertryck i patientens luftvägar (vid början av en inandning) öppnas ventilen och luft kan strömma igenom. Undertrycket vid öppningsögonblicket kallas för öppningstryck eller "ocklusionstryck" (se bild 2.2).

Mätutrustningen består av tre differentialtryckgivare (bild 4.2). Den första givaren mäter det faktiska trycket i patientens luftvägar. Det är med denna givare som tryckfallet ($P_{0.1}$) mäts. Den andra mäter differentialtrycket från en flödesmätare (venturirör). Utifrån denna flödesmätning kan inandningsvolymen beräknas. Den tredje tryckgivaren mäter det vakuumpycket som används för att hålla ocklusionsventilen stängd. När ocklusionsventilen öppnas försvinner vakuumpycket. Med hjälp av tryckgivaren kan man indikera när ocklusionsventilen öppnas.

I utrustningen ingår också en tryckventil som påverkar vakuumpycket. Denna ventil styrs av en elektrisk signal. Ventilen har endast två lägen, på eller av. Med denna kan man utifrån stänga av vakuumpycket så att ocklusionsventilen hålls konstant öppen. Ocklusionsventilen kan alltså öppnas dels av ett undertryck i patientens luftvägar och dels av en från datorn styrd elektrisk signal.

Vid ocklusionsmätning finns också ett intresse av att kunna variera ocklusionsventilens öppningstryck. För detta ändamål finns en manuell "ratt" på mätutrustningen. Med denna justeras steglöst storleken av vacuumtrycket. Vakuumpycket påverkar i sin tur öppningstrycket för ocklusionsventilen.

I utrustningen finns också en extern gasanalysator. Denna har en egen panel där bl.a. utandningsluftens koldioxidkoncentration visas. Från gasanalysatorn finns analoga signaler som indikerar utandningsluftens gassammansättning.

De tre tryckgivarna och den elektriskt styrda tryckventilen samt koldioxidsignalen är anslutna till ett AD-kort och en dator. Under utprovningen har AD-kortet installerats dels i en PC-XT (8086) och dels i en LT3400 LapTop (-386 SX).

För programutvecklingen finns dels en Microsoft QuickC - kompilator och dels en Microsoft Optimizing C (ver 6.0) - kompilator.

5. Definition och beräkning av parametrar

5.1 Definition och beräkning av $P_{0.1}$

Tryckfallshastigheten $P_{0.1}$ är ursprungligen definierad som "tryckfallet under 0.1 sekunder", därav beteckningen "0.1". På institutionen för medicinsk teknik används en något modifierad algoritm för att på ett mer flexibelt sätt kunna beräkna $P_{0.1}$. Parametern skattas istället genom att beräkna tryckets regressionslinje i området 30 - 90% av öppningstrycket (se bild 5.1). $P_{0.1}$ fås som en tiondel av linjens riktningskoefficient. Trycket anges i dessa sammanhang i enheten cmH_2O .

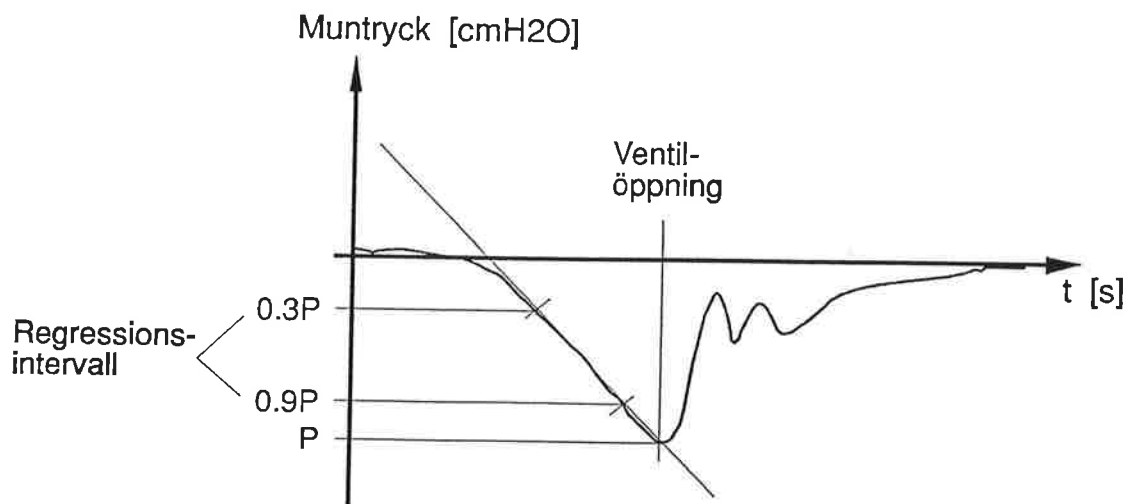


Bild 5.1 Definition av regressionsintervall

Vid normal andning har $P_{0.1}$ ett värde i storleksordningen 0.5 till 3.0 cmH₂O/0.1s. Vid extrem andning kan en person ha $P_{0.1}$ -värden på upp till 25.0 cmH₂O/0.1s. Eftersom ocklusionstiden (se bild 2.2) i dessa fall kan vara så kort som 5 ms krävs en samplingshastighet på minst 1000 Hz för att erhålla tillräckligt många punkter för beräkning av $P_{0.1}$ genom regression.

Vid mätning av $P_{0.1}$ är luftvägstryckgivarens signal av intresse endast under ocklusionstiden. I programmet sparas kontinuerligt de 300 sista sampelvärdena (vid samplingsfrekvensen 1000 Hz). När ett nytt andetag har påbörjats indikeras detta av en "triggpuls". Utifrån de sparade sampelvärdena bestäms först det minimala trycket (öppningstrycket). Därefter bestäms 90%- och 30%:s nivåerna (se bild 5.1). Därefter beräknas kurvans riktningskoefficient genom regression inom det aktuella intervallet. Dessutom beräknas regressionens korrelationskoefficient som kan användas för att göra en kvalitetsbedömning av $P_{0.1}$.

5.2 Beräkning av inspirationsvolym

För att beräkna inspirationsvolymen används en flödesmätare av venturityp. Volymen fås genom att integrera flödet med avseende på tiden. Eftersom flödeskurvan är samplad får integralen approximeras (se bild 5.2). I programmet beräknas integralen genom att under ett andetag kontinuerligt summera sampelvärdena för flödet och vid slutet av andetaget multiplicera den totala summan med sampelintervallet.

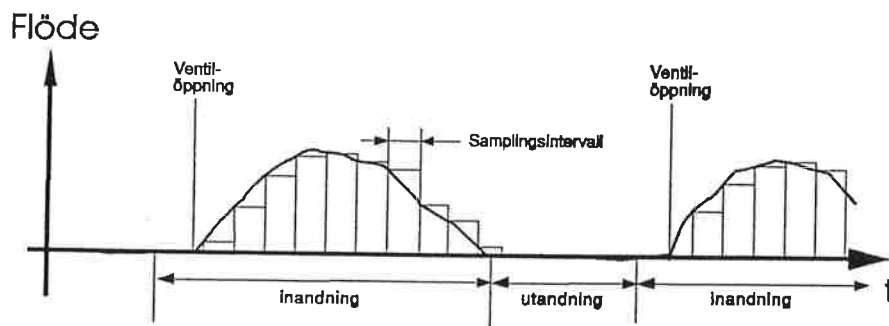
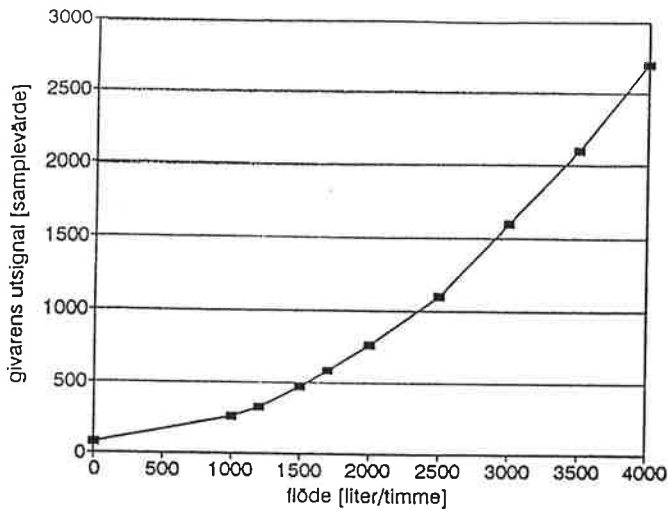
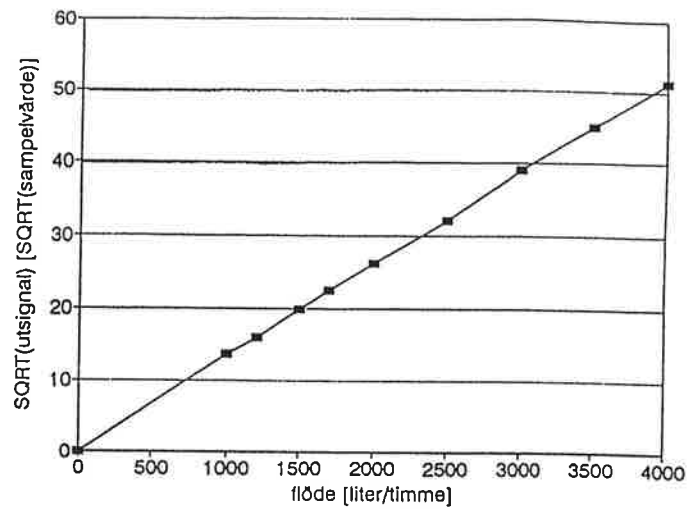


Bild 5.2 Illustration av den samplade integralen

Ett problem med flödesgivaren är att den inte är linjär (se bild 5.3a). Men eftersom förhållandet har ett kvadratisk beroende fås en betydligt mer hanterbar kurva om kvadratroten ur sampelvärdet avsätts på y-axeln (se bild 5.3b).



(a)



(b)

Bild 5.3(a) Flödesgivarens utsignal som funktion av flödet

(b) Kvadratroten av flödesgivarens utsignal som funktion an flödet

5.3 Definition av koldioxidkoncentration

Vid studier av andningsfunktion har ofta koldioxidkoncentrationen i utandningsluften stor betydelse. Den medicinska benämningen för detta är "endtidal koldioxidkoncentration" som betecknar koldioxidhalten av utandningsluften vid slutet av ett andetag. Det är denna utandningsluft som under längst tid utsätts för gasutbyte i lungalveolerna. Genom att mäta den endtidala koldioxidkoncentrationen kan man få en uppfattning av hur väl gasutbytet i lungalveolerna fungerar.

Den koldioxidgivare som används i mätutrustningen har en viss tröghet. Detta innebär att det uppmätta värdet inte motsvarar det sanna värdet framför allt vid högfrekventa förlopp. Dessutom har givarsignalen ca två sekunders fördröjning. För att förenkla beräkningar definieras den endtidala koldioxidkoncentrationen som "det maximala värdet från koldioxidgivaren under ett helt andetag" (se bild 5.4).

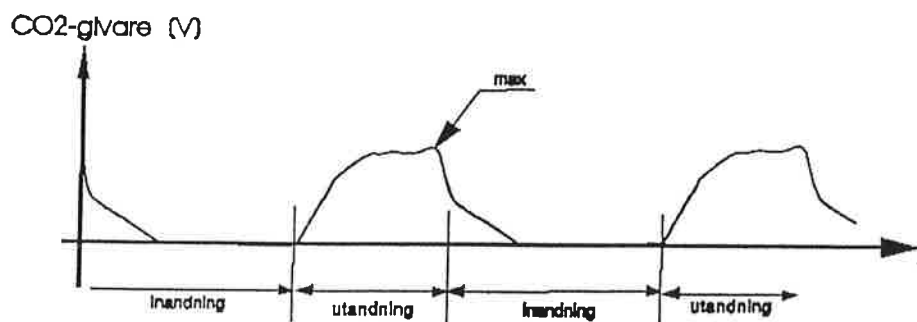


Bild 5.4 Koldioxidgivarens utsignal som funktion av tiden (utan fördröjning)

6 Programutveckling

Detta kapitel är en beskrivning av examensarbetets huvuduppgift - att utveckla programvara till mätsystemet. Framställningen är huvudsakligen uppbyggd i den kronologiska ordning utvecklingsarbetet har genomlöpt.

Bild 6.1 illustrerar den processgraf som ligger till grund för programutvecklingen. Framställningen i detta kapitel syftar till att ge en översiktlig förklaring av denna. För en mera detaljerad beskrivning av de funktioner som exporteras av de abstrakta datatyperna hänvisas till motsvarande "headerfiler". För beskrivning av övriga funktioner som används i programmet hänvisas till källfilerna (se appendix).

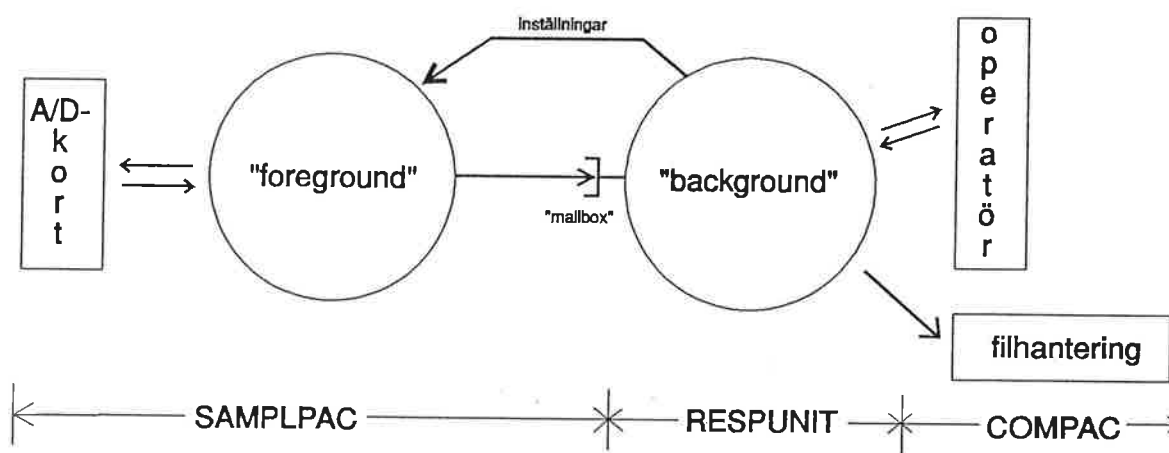


Bild 6.1 Processgraf

I kap 6.1 ges en beskrivning av avbrottshandlingens uppbyggnad och funktion. Denna struktur ligger sedan som grund för utformningen av den abstrakta datatypen "SAMPLPAC", som beskrivs i kap 6.2. I "SAMPLPAC" finns de funktioner som kontinuerligt läser in data från AD-kortet och lagrar data temporärt i speciella ringbuffertar. I kap 6.3 beskrivs den abstrakta datatypen "COMPAC". Från "COMPAC" exporteras funktioner som handhar skärmhantering och filhantering. I kap 6.4 beskrivs implementeringen av huvudprogrammet "-RESPUNIT". I "RESPUNIT" finns bl.a. huvudfunktionen "main()" som använder sig av funktioner från de ovan nämnda abstrakta datatyperna. I "RESPUNIT" finns dessutom samtliga funktioner som utför beräkningar på insamlad data.

6.1 "Foreground/Background-scheduler" - för avbrottshandling

En av målsättningarna för programvaruutvecklingen är att kunna avläsa de analoga mätsignalerna med samplingsfrekvensen 1000 Hz. För att åstadkomma en så snabb samplingsfrekvens kan man inte använda sig av den systemklocka som normalt används som tidsbas. Dels har systemklockan inte större

upplösning än 10 millisekunder, dels uppstår svårigheten att sampla med konstanta tidsintervall, samtidigt som beräkningar och kommunikation med användaren skall utföras.

Istället krävs en sk avbrottsstyrd programvara. I denna använder man sig av en programmerbar hårdvaruklocka med en maximal upplösning på 0.84 mikrosekunder. Med hjälp av denna kan man generera avbrottssignaler med konstanta tidsintervall (t.ex. 1 ms). Vid en avbrottssignal laddas programräknaren med adressen till en "avbrottsrutin". När avbrottsrutinen är genomgången laddas programräknaren återigen med den adress vid vilken programräknaren befann sig före avbrottet.

På institutionen för reglerteknik vid Lunds Tekniska Högskola har ett avbrotts hanteringssystem utvecklats - "Foreground/Background-scheduler" (Brück, 1988). Detta är ursprungligen framtaget för att enkelt kunna implementera olika regleralgoritmer i dator och sedan med hög samplingsfrekvens kunna styra en mekanisk process. "Foreground" syftar på avbrottsrutinen, där regleralgoritmen implementeras. "Background" syftar på den del av programmet som handhar kommunikation med operatören.

"Foreground/Background-scheduler" är skriven i assembler och är konstruerad för att användas i högnivåspråket C. De två huvudfunktioner som anropas i C-programmet är "schedule()" och "reset()". Vid anrop av funktionen "schedule()" initieras avbrotts hanteringssystemet. Avbrottsrutinens adress och önskad samplingsfrekvens registreras. Gamla avbrottsvektorer sparas och nya vektorer skapas. Den programmerbara hårdvaruklockan ställs om. Vid anrop av funktionen "reset()" återställs avbrottsvektorerna och hårdvaruklockan.

Den stora fördelen med "Foreground/Background-scheduler" är dess snabbhet. Genom dess enkla struktur kan man på en relativt långsam processor implementera programvara med mycket hög samplingsfrekvens. Samtidigt undviker man den komplexitet som vanlig realtidsprogrammering medför (prioritetsnivåer, semaforer, events, time-outs osv). "Foreground/Background-scheduler" har dock vissa begränsningar:

- * I avbrottsrutinen ("foreground") kan inte flyttalsoperationer utföras. Detta beror på att flyttalsregisterna inte sparas vid processbyte.
- * DOS-kommandon som är beroende av BIOS kan ej anropas från avbrottsrutinen. Detta beror på att "Foreground/Background-scheduler" manipulerar den avbrottsvektor som normalt genererar avbrott till BIOS.
- * "Foreground/Background-scheduler" är i nuvarande utförande begränsat till "small memory model". Detta innebär att den exekverbara programkoden inte kan överstiga 64kB.

Jämfört med en vanligt realtidskärna har "Foreground/Background-scheduler" också helt naturligt vissa strukturella begränsningar. "Foreground/Background-scheduler" består i princip endast av två processer. Men eftersom den ena processen ("foreground") saknar möjligheten att utföra avancerade beräk-

ningar, måste detta utföras i den andra processen ("background"). För att samtidigt kommunicera med operatören (tangentbordet) skulle det behövas ytterligare en tredje process. Problemet kan dock lösas genom pollning. "Background"-processen kontrollerar kontinuerligt dels om det finns nya data att behandla från "foreground"-processen och dels om någon tangentnedtryckning har gjorts.

Då man i princip kan se "Foreground/Background-scheduler" som två olika processer måste man ta hänsyn till ömsesidig uteslutning för gemensamma resurser. Någon form av synkronisering av processerna måste också implementeras. Detta kan göras i form av en "mail box" (se kap 6.2).

"Foreground/Background-scheduler" är utvecklad med avseende på MS-DOS 3.0 och är avsedd att användas till ett specifikt AD-kort. Andningsövervakningssystemet är däremot avsett att användas på MS-DOS 4.0 samt på ett annat AD-kort. Detta medför att vissa ändringar och tillägg var nödvändiga för att "Foreground/Background-scheduler" skall kunna användas i programutvecklingen. Den återstående delen av kap 6.1 ägnas åt att beskriva dessa förändringar.

I "Foreground/Background-scheduler" finns en "Ctrl-Break"-funktion implementerad. Denna används för att göra avbrottsanrop direkt från tangentbordet och därmed avsluta exekvering av ett program. Funktionen fungerar bra på institutionens XT-maskin (80186) med DOS 3.0. På institutionens LapTop (80386) med DOS 4.0 återställs däremot inte tangentbordets avbrottsvektor korrekt vid "Ctrl-Break". Exekveringen avbryts visserligen men därefter svarar inte datorn på tangentnedtryckningar. Felet beror antagligen på skillnader i operativsystemen. Felet avhjälpas genom att korrigera ett funktionsanrop (interrupt 21H) i assemblerkoden.

Det aktuella programavsnittet har följande ursprungliga utseende:

```
; _schedule - register FG procedure and period
...

; Catch CTRL-BREAK to make sure we reset the interrupt

MOV  AX,CS
MOV  DS,AX
MOV  DX,OFFSET BREAK_int
MOV  AH,25H
MOV  AL,1BH
INT  21H
...
```

När den näst sista raden ändras till:

```
MOV  AL,23H
```

fungerar "Ctrl_break" felfritt i både DOS 3.0 och DOS 4.0.

De funktioner som i "Foreground/Background-scheduler" används för hantering av AD-kortet måste anpassas till institutionens speciella AD-kort. Dels finns de två funktionerna "ADin()" och "DAout()"

som är avsedda att användas i C-programmet och dels finns funktionen "DAOUT" som endast används i assemblerkoden. Eftersom C-biblioteket erbjuder relativt enkla och snabba funktioner för porthantering, hanteras inläsning och utskrifter på AD-kortet enklast direkt från C-programmet. De två förstnämnda funktionerna behöver alltså inte ersättas i assemblerkod. Den sistnämnda funktionen (DAOUT) används i assemblerprogrammet för att man på ett oscilloskop ska kunna studera de olika faserna i avbrotts hanteringen. Denna funktion är nödvändig och en motsvarande funktion som är anpassad till institutionens AD-kort måste implementeras i assemblerkoden. Den modifierade funktionen kallas "DAOUTA" (D/A-omvandlare A) och har följande utseende:

```

; DAOUTA - output integer voltage on 8-bits D/A-converter (A)
;
; Note:      AX is destroyed.
;           VOLTAGE must be a constant expression.

DAOUTA    MACRO    VOLTAGE
    IF TSIGNAL
        PUSH DX
        CLI
        MOV DX,DA8A           ;DA8A    EQU    0714H
        MOV AX,-VOLTAGE*12
        OUT DX,AX
        STI
        POP DX
    ENDIF
    ENDM

```

AD-kortet har två 8 bitars D/A-omvandlare (A och B). Funktionen skickar ut signaler på A.

Utprovning och testresultat av "Foreground/Background-scheduler" finns beskrivet i kap. 7.

6.2 Abstrakt datatyp - SAMPLPAC

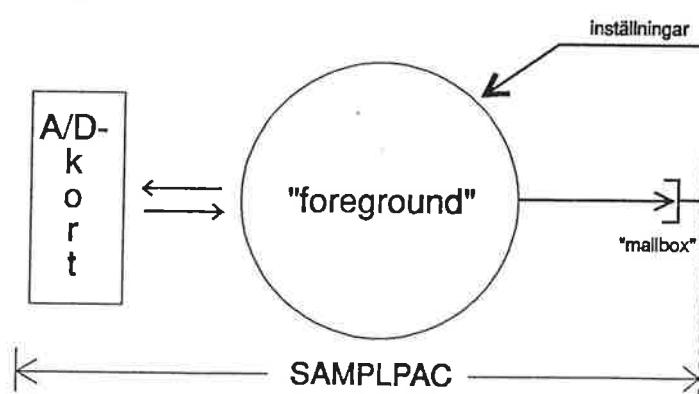


Bild 6.2 Illustration av SAMPLPAC (se även bild 6.1)

I den abstrakta datatypen "SAMPLPAC" implementeras avbrottsrutinen (el. processen) "foreground". Huvudsyftet med "SAMPLPAC" är att åstadkomma en programstruktur som möjliggör att insamling av data blir oberoende av vad som sker i bakgrundsprocessen "background". Detta löses genom att efter-

likna vad som i realtidssammanhang kallas "mailbox". Insamlad data placeras i brevlådan allt eftersom ny data produceras. Data kan sedan hämtas av en annan process när denna har tid och möjlighet. Om brevlådan är tom kan den aktuella processen (här "background") antingen vänta tills ny data har inkommit eller under tiden utföra andra operationer t.ex. kommunicera med användaren. Brevlådans fysiska storlek är dock begränsad (pga begränsat minnesutrymme). Om brevlådan skulle vara full måste "foreground" tyvärr ignorera nyinkommen data.

Ovanstående resonemang implementeras i programmet med en länkad ringbuffert, bestående av ett antal poster. När ett nytt andetag indikeras samlas all data från föregående andetag ihop och infogas som ett dataelement i en av ringbuffertens poster (se bild 6.3). I posten sätts dessutom den logiska variabeln "DataAvailable", som anger att posten innehåller ny data. "Foreground" kan sedan obehindrat fortsätta med insamling av ny mätdata från påföljande andetag. Bakgrundsprocessen kan i mån av tid hämta lagrad data. När data har behandlats markeras detta genom att återställa den logiska variabeln "DataAvailable".

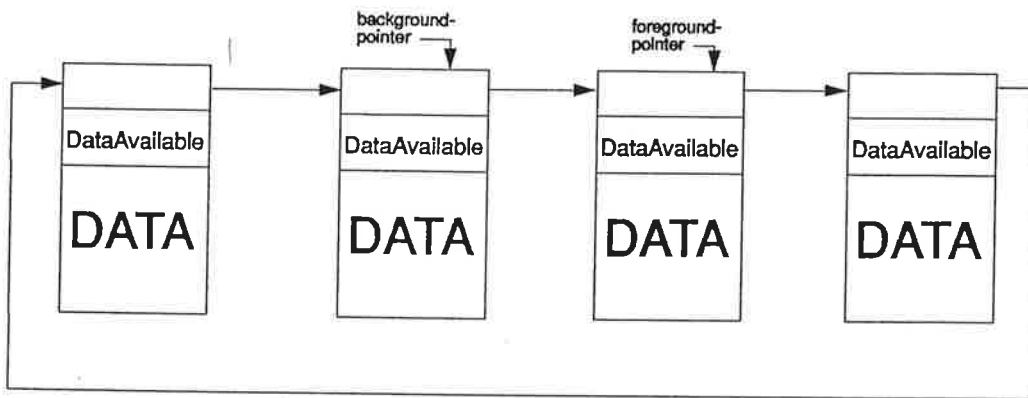


Bild 6.3 Länkad "data-ringbuffert"

Posterna i den länkade ringbufferten har följande utseende ("struct" motsvaras av "record" i Pascal):

```
struct databuffstruct {
    int DataAvailable;
    struct datastruct Data;
    struct databuffstruct *NextElement;
};
```

Dataelementet definieras av följande "structure":

```
struct datastruct {
    unsigned int BreathNo;
    unsigned long TotNumOfSamples;
    unsigned long NumOfInspSamples;
    unsigned long InspFlowSum;
    unsigned int Event;
    struct dlinkbuffstruct *FirstDLinkBuffPtr;
};
```

Det sista elementet "FirstDLinkBuffPtr" är en pekare till en dubbellänkad lista. Denna avses innehålla de senaste sampelvärdena av luftvägstrycket. Utifrån dessa sampelvärden beräknas parametern "Po.1" (se kap 5.1).

I den återstående delen av kapitel 6.2 beskrivs de viktigaste funktionerna i den abstrakta datatypen SAMPLPAC. För en mer detaljerad beskrivning hänvisas till motsvarande headerfil och källfil (se appendix A.2).

De två viktigaste funktionerna som exporteras från SAMPLPAC är "InitKernel()" och "GetNewData()". I "InitKernel" skapas de ovan nämnda ringbuffertarna, AD-kortet initieras och "Foreground/Background-scheduler" startas. Den andra funktionen "GetNewData()" motsvarar den "mailbox" som har beskrivits tidigare i detta kapitel. Om det finns ny data att inhämta, returnerar funktionen vid anrop en pekare till en post av typen "datastruct" (se ovan). Saknas däremot nyinkommen data returneras en pekare med värdet "NULL".

Från SAMPLPAC exporteras också ett antal funktioner som anropas för att ändra vissa variabler i avbrottsrutinen "foreground". En sådan funktion är t.ex. "ChangePvacuTrigLevel()" som används för att ändra triggnivån för vakumtrycket. Dessa funktioner symboliseras av pilen "inställningar" i process-grafen (bild 6.1).

Avbrottsrutinen "foreground()" kan i stora drag beskrivas med följande "pseudokod":

```
void foreground()
{
    inkrementera "antalet sampel i detta andetag"-räknaren;

    avläs luftvägstryckets sampelvärde;
    infoga sampelvärdet i den dubbellänkade listan;

    avläs flödets sampelvärde;
    beräkna kvadratroten på sampelvärdet och addera detta till "flödessumman"; (se kap 5.2)

    avläs vakuumtryckets sampelvärde;
    OM (vakuumtrycket indikerar att ett nytt andetag har påbörjats)
    {
        placera all data från föregående andetag i "DataBuff"-listan ("mailbox");
        gör nollställningar och rotera ringbuffertar;
    }
};
```

Eftersom inga flyttalsoperationer kan utföras i "foreground" kan inte C-bibliotekets vanliga "sqrt()" -funktion anropas för att beräkna roten av flödessignalens sampelvärde. Dessutom bör samtliga operationer i "foreground" vara så tidssnåla som möjligt. Därför används en förenklad algoritm för beräkning av roten (Newton-Raphsons metod). Metoden bygger på ett iterativt förfarande.


```

unsigned int SQRT(unsigned int sample)
/* Returns the square root of "sample". */
{
    unsigned int i, sqrt_sample;

    sqrt_sample = (unsigned int) 4 + 6*sample;
    for (i=1;i<=10;i++)
    {
        sqrt_sample = (unsigned int) (sqrt_sample + sample/sqrt_sample) / 2;
    }
    return (sqrt_sample);
}

```

Denna förenklade algoritm konvergerar endast för heltal i området 0 - 5000. Funktionen är dock fullt tillräcklig eftersom ett sampelvärde maximalt antar värdet 4096.

6.3 Abstrakt datatyp - COMPAC

Den abstrakta datatypen COMPAC erbjuder funktioner för kommunikation med användaren samt funktioner för filhantering. Datatypen är uppdelad i fyra delar; hantering av "rullgardinsmenyer", fönsterhantering, allmän skärmhantering och filhantering. Dessa delar är inbördes oberoende och skulle i princip var och en kunna utgöra en mindre abstrakt datatyp. Av praktiska skäl är det dock enklare att samla dessa i en och samma fil.

"Rullgardinsmenyn" är uppbyggd genom att ett antal huvudmenyer visas på skärmen. Genom att använda piltangenterna kan man "stega sig runt" bland menyalternativen och med "return"-tangenten aktivera önskad funktion. Det aktuella menyvalet markeras med avvikande färg som har valts för att fungera även på svart/vit- och plasmaskärm. Fördelen med menyhanteringssystemet är att det kan användas i en programstruktur med "pollning". Användarens tangentnedtryckning kan (med funktionen "kbhit()") pollas parallellt med att andra funktioner pollas. En utförlig beskrivning av hur menyhanteringssystemet ska implementeras finns i headerfilen till COMPAC (appendix A.2).

Fönsterhanteringsfunktionerna baseras på programbiblioteket "CodeBase 4" (Sequiter software inc., 1988). COMPAC erbjuder ett antal funktioner som vid anrop aktiverar ett "pop up"-fönster. Fönstret kan t.ex. användas för inläsning av flyttal eller textsträngar i kombination med lämplig ledtext. Fönstret kan också användas för att enbart visa en informationstext. Efter att användaren har tryckt på "escape"- eller "return"-tangenten försvinner fönstret och tidigare skärminformation visades på nytt.

Den allmänna skärmhanteringsdelen avser funktioner som skriver ut ledtexter och värden av olika parametrar. Funktionerna baseras på det grafiska bibliotek "graphics.lib" som Microsoft C-kompilatorerna erbjuder.

Filhanteringsdelen exporterar funktioner för inläsning och utskrifter på filer (se bild 6.4). För filhanteringen används "top-level functions". Detta innebär att en filbuffert fylls på med data och skriver ut på aktuell fil först när bufferten är fylld.

```

"File name      :kriste.002
"Identity number :650603-1175
"Name          :Krister Tham
"Date          :Wed Jan 08 13:47:13 1992

"Abbreviations:
"BrNo          = breath number
"Time          = time scale [min] (e.g time when data is saved on file)
"TT           = total time of the breath [sec]
"TI           = total time of the inspiration [sec]
"TO           = total time of the occlusion [sec]
"P0.1         = inspiratory drive [ cmH2O/0.1sec ]
"r            = correlation coefficient of drive regression [-1 to 1]
"n            = number of samples which drive regression is based on
"Pmin         = minimal occlusion pressure [cmH2O]
"VT           = inspired tidal volume [liter]
"CO2          = end tidal CO2 airway pressure [%]
"RF           = 30percent-level of drive regression found [ 0(=false) or 1(=true) ]
"PS           = Pressure sensor Amplitude OK (sensor not saturated) [ 0(=false) or 1(=true)]
"FS           = Flow sensor Amplitude OK (sensor not saturated) [ 0(=false) or 1(=true) ]
"Ev           = event number
"EventMess    = message inserted when the event counter is incremented

"BrNo Time TT TI TO P0.1 r n Pmin VT CO2 RF PS FS Ev EventMess
1, 0.50, 12.96, 1.15, 0.069, -1.68, -0.961, 46, -1.16, 0.91, 4.6, 1, 1, 1, 0,
2, 0.54, 2.80, 1.07, 0.037, -3.41, -0.979, 21, -1.25, 0.76, 4.6, 1, 1, 1, 0,
3, 0.59, 2.71, 1.05, 0.066, -1.75, -0.962, 44, -1.16, 0.73, 4.7, 1, 1, 1, 0,
4, 0.64, 2.72, 1.10, 0.069, -2.06, -0.988, 38, -1.42, 0.72, 4.6, 1, 1, 1, 0,
5, 0.68, 2.69, 1.18, 0.081, -1.65, -0.982, 46, -1.33, 0.84, 4.8, 1, 1, 1, 0,
6, 0.73, 2.71, 1.25, 0.075, -1.48, -0.971, 39, -1.11, 0.75, 4.6, 1, 1, 1, 0,
7, 0.77, 3.02, 1.13, 0.061, -1.83, -0.974, 37, -1.12, 0.56, 4.5, 1, 1, 1, 1, kommentar
8, 0.81, 2.53, 1.21, 0.119, -0.81, -0.961, 60, -0.97, 0.57, 4.6, 1, 1, 1, 1,
9, 0.86, 2.84, 1.73, 0.099, -1.12, -0.971, 57, -1.11, 0.57, 4.6, 1, 1, 1, 1,

```

Bild 6.4 Utskrift av fil för insamlad andningsdata

6.4 Huvudprogram - RESPUNIT

I RESPUNIT finns huvudfunktionen "main()". Dessutom finns alla funktioner för beräkningar av andningsparametrar, trigg nivåer, nollnivåer, kalibreringsparametrar m.m.

I huvudfunktionen "main()" anropas två funktioner; "run_statistics()" och "run_measure()". Tanken är att mätdatainsamlingsprogrammet i en senare version dels skall kunna anropa "run_measure()" för att

samla in andningsdata och göra kontinuerliga beräkningar och dels skall kunna anropa "run_statistics()" efter en mätning för att göra vissa enklare statistiska beräkningar och analyser på insamlad data. Funktionen "run_statistics()" är ännu inte implementerad utan avses att implementeras vid framtida utveckling av programmet.

Huvudprogrammets viktigaste funktion är alltså "run_measure()". Denna funktion använder sig dels av importerade funktioner från SAMPLPAC och COMPAC och dels av funktioner som är implementerade i RESPUNIT (t.ex. beräkning av P0.1).

Vid anrop av "run_measure()" görs först ett antal initieringar, bl.a. läses ett antal startparametrar från den senaste exekveringen av programmet in från en "setupfil". Därefter anropas den loop som kan sägas utgöra programmets bakgrundsprocess - "background". I loopen pollas dels tangentbordsnedtryckningar och dels "GetNewData()" som är den "mailbox"-funktion som tidigare har beskrivits (kap 6.2). Loopens huvudstruktur kan illustreras med följande "pseudokod":

```
LOOP
{
    OM (tangentnedtryckning)
        {
            läs in tangentnummer;
            anropa lämpliga funktioner;
        }
    OM ( GetNewData() != NULL )
        {
            utför beräkningar på ny data;
            lagra uträkningarna på fil;
            uppdatera aktuella parametrar på skärmen;
        }
}
```

När användaren anser sig vara färdig med insamling av data avslutas loopen. Ett antal återställningsfunktioner anropas och "run_measure()" avslutas.

7 Utprovning

7.1 Validering av andningsparametrar.

För att kunna bedöma det av programmet uträknade värdet av P0.1 kopplas den analoga signalen från luftvägstryckgivaren till en plotter. Därigenom kan en grafisk beräkning av P0.1 göras. Den grafiska beräkningen kan anses ha en noggrannhet som är bättre än ± 0.5 cmH₂O/0.1s. Öppningstrycket är 1 cmH₂O. Resultatet beskrivs i tabell 7.1. Av tabellens värden kan man dra slutsatsen att de av datorn beräknade värdena av P0.1 har god noggrannhet.

P0.1 [cmH ₂ O/0.1s] (grafisk beräkning)	P0.1 [cmH ₂ O/0.1s] (datorns värde)
1.23	1.25
2.38	2.35
7.20	7.20
11.58	11.50
20.70	21.03

Tabell 7.1

För bedömning av den beräknade inandningsvolymen används en "volymkalibreringsspruta". Sprutan har en noggrannhet på 0.01 liter. I tabell 7.2 finns de uppmätta och beräknade värdena. Vid en direkt beräkning av flödeskurvens integral blev volymvärdet något för litet. Detta kan bero på att den rotameter som har använts för att kalibrera flödesgivaren inte är korrekt. Avvikelsen visar sig dock vara linjär i förhållande till volymens storlek. Om den beräknade integralen korrigeras med en faktor 1.15 erhålls en inandningsvolym med godtagbar noggrannhet.

Volym [liter] (kalibreringsspruta)	Volym [liter] (datorns värde)	Volym [liter] (korrigerat värde)
0.10	0.09	0.10
0.20	0.18	0.21
0.50	0.44	0.51
1.00	0.86	0.99
3.00	2.65	3.07

Tabell 7.2

Den tredje parametern som beräknas i programmet är den "endtidala koldioxidkoncentrationen" (kap 5.3). Programmets beräknade värde jämförs med den externa koldioxidmätarens utslag enligt tabell 7.3.

CO ₂ -konc. [%] (externt värde)	CO ₂ -konc. [%] (datorns värde)
2.3	2.3
3.5	3.5
4.7	4.7
5.5	5.6
6.9	6.9

Tabell 7.3

Vad gäller programmets beräkningar av andetagets in- och utandningstid stämmer dessa väl överens med manuellt uppmätta tider.

Slutsatsen är att datorns beräknade andningsparametrar har god noggannhet.

7.2 Beräkning av maximal samplingshastighet

Ett av de krav som ställdes på den färdiga utrustningen var att programvaran skulle kunna användas på PC-datorer med olika processorkapacitet. Detta har lösts genom att använda en avbrottstyrd programstruktur. Vad som dock begränsar programmets användningsområden är dess maximala samplingsfrekvens. Den maximala samplingsfrekvensen beror på hur snabbt avbrottsrutinen kan genomlöpas dvs exekveringstiden för avbrottsrutinens samtliga operationer och funktionsanrop.

För att kunna studera detta har i "Foreground/Background-scheduler" implementerats en funktion som på en D/A-omvandlare varierar spänningen beroende på var i programmet exekvering sker (se kap 6.1). När avbrottsrutinen anropas sätts D/A-omvandlaren till 10 volt och när avbrottsrutinen är genomlöst återställs D/A-omvandlaren till 1 volt (se bild 7.1).

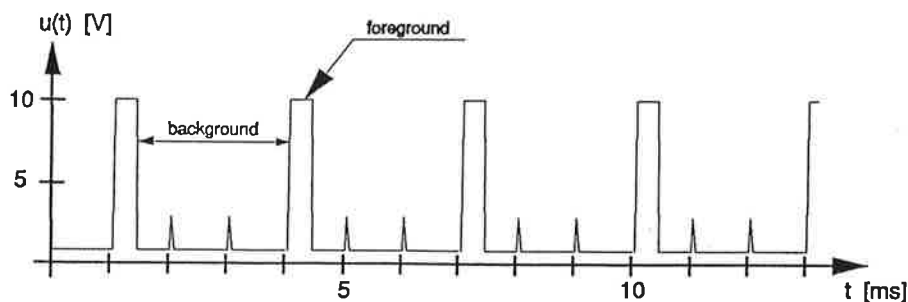


Bild 7.1 D/A-omvandlarens utsignal vid exekvering på en 386-processor
(samplingsintervall = 3 ms).

I bild 7.1 illustreras också "spänningsspikar" med amplituden tre volt. En sådan "spik" markerar att det har genererats en avbrottsignal från hårdvaruklockan. I "Foreground/Background-scheduler" genereras en avbrottssignal varje millisekund oavsett inställt samplingsintervall. Vid varje avbrott inkrementeras en räknare som håller reda på vid vilka avbrotts signaler "foreground" skall anropas. I bild 7.1 anropas "foreground" vid var tredje avbrottssignal.

Exekveringstiden för avbrottsrutinen varierar enligt tabell 7.4 .

Processorbeteckning med klockfrekvens	Exekveringstid för "foreground" [ms]	Maximal samplings- frekvens [Hz]
8086 (8 MHz)	1.40	500
80286 (16 MHz)	0.70	1000
80386 SX (16 MHz)	0.50	1000
80386 (25 MHz)	0.30	2000

Tabell 7.4

"Foreground/Background-scheduler" är implementerad så att samplingsintervallet måste väljas som ett heltal i millisekunder. Slutsatsen är att kravet på samplingsfrekvensen 1000 Hz går att uppfylla för 286- och 386-processorn. XT-maskinen med 8086-processorn klarar dock endast att sampla med 500 Hz.

Avbrottsrutinens mest tidskrävande operationer är dels "sample and hold"-tiden och dels beräkningen av roten av flödessamplet. Genom att modifiera eller utesluta den sistnämnda operationen skulle en snabbare exekvering av avbrottsrutinen kunna uppnås. Därigenom skulle även 8086-processorn kunna användas för sampling med 1000 Hz (se även kap 8).

7.3 Test av utrustningen vid "återandning"

Vid institutionen för medicinsk teknik bedrivs flera olika studier av andningsfunktion där analys av P_{0.1} ingår. En av metoderna går ut på att studera hur P_{0.1} påverkas av inandningsluftens koldioxidkoncentration. Koldioxidkoncentrationen kan varieras i form av pulser, steg eller ramper.

Ett enkelt sätt att åstadkomma en rampliknande ökning av inandningsluftens koldioxidkoncentration är genom s.k. återandning. Vid återandning låter man försökspersonen andas in och ut i en sluten behållare. Allteftersom patienten andas, konsumeras syre och produceras koldioxid i behållaren. Koldioxidhalten ökar successivt i form av en ramp. I bild 7.2 illustreras resultatet av en sådan test där mätutrustning har använts tillsammans med den nyutvecklade programvaran.

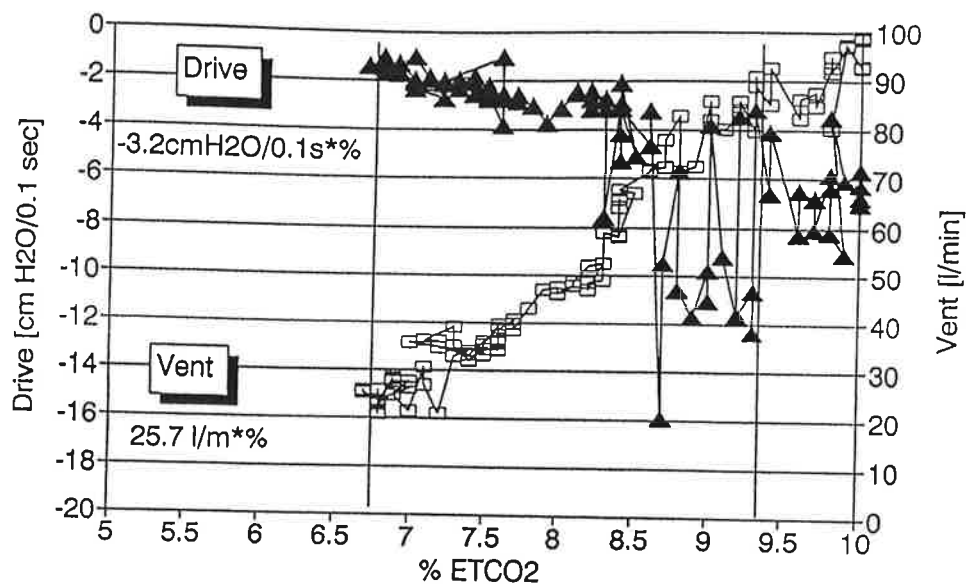


Bild 7.2 Resultat av återandningstest där $P_{O.1}$ och inandningsvolym per minut avsätts som funktion av koldioxidkoncentrationen i utandningsluften.

I grafen kan man notera att en ökad koldioxidhalt i inandningsluften resulterar i en ökad "drive" vilket kan sägas motsvara en ökad vilja att andas in luft. Denna "ökade vilja" resulterar i en ökad ventilation vilket tyder på att försökspersonen har en fungerande andningsreglering. Om "drive"-värdena däremot inte hade ökat vid en ökad koldioxidkoncentration skulle detta tytt på en defekt andningsreglering. På samma sätt hade en konstant ventilation trots ökade "drive"-värden tytt på en defekt i lungorna.

Resultatet av återandningstestet stämmer väl överens med tidigare utförda tester på normala personer vid institutionen och i litteraturen. Det visar vidare att mätutrustningen har kapacitet att användas vid relativt höga ventilationer med minutvolymen på upp till ca 100 l/min.

8 Utvecklingsmöjligheter

8.1 Förbättring av kalibreringsrutin för flödesgivare

Vid utprovningen i kapitel 7.1 visar det sig att den beräknade volymen avviker från den sanna volymen. Avvikelsen är dock linjär och kan lätt avhjälpas genom att korrigera värdet med en empiriskt framtagen korrektionsfaktor. Eftersom avvikelsen är linjär kan man dra slutsatsen att felet beror på dålig noggrannhet vid kalibreringen av flödesgivaren.

För volymberäkningar vore en mer rättfram metod att kalibrera flödesgivaren med avseende på volymen, dvs med hjälp av en volymkalibreringspruta. Vid kalibreringen skattas alltså riktnings-

koefficienten för grafen i bild 8.1. Detta kan lätt implementeras med befintliga kalibreringsfunktioner. En nackdel är dock att man utifrån denna kalibrering inte kan beräkna det momentana flödet.

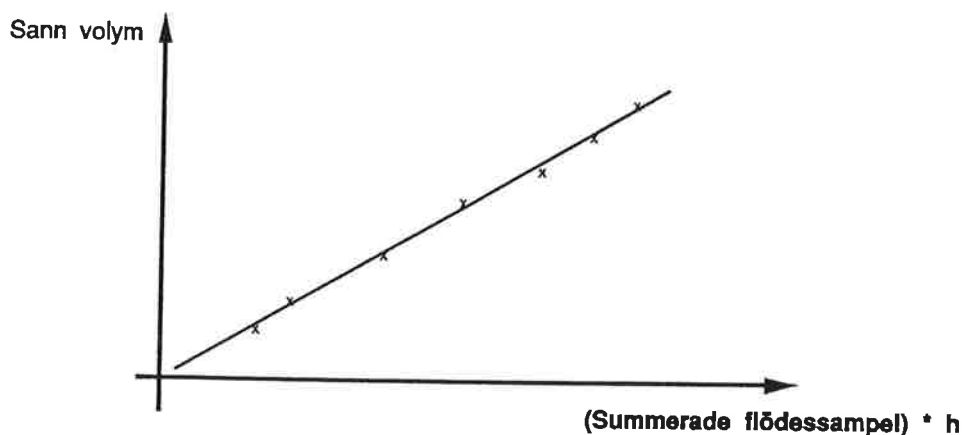


Bild 8.1 Regressionskurva för kalibrering av volym (h =sampelintervall)

8.2 Beräkning av processorkapacitet

Vad gäller programvarans kompatibilitet på datorer med olika processorkapacitet har det nuvarande programmet två brister:

Det första problemet gäller implementering av AD-omvandling. Eftersom AD-kortet saknar funktion som markerar när omvandling är klar ("busy-flagga") måste omvandlingstiden implementeras med en vänteloop (40 mikrosek). När processorhastigheten ändras uppstår problemet att hålla omvandlingstiden konstant. Antalet loopar i vänteloopen måste anpassas till den nya processorhastigheten.

Det andra problemet vid varierande processorkapacitet uppstår när "Foreground/Background-scheduler" anropas med ett önskat samplingsintervall som är mindre än exekveringstiden för avbrottsrutinen. Resultatet blir att avbrottsrutinen ständigt anropas så fort tidigare avbrottsanrop avslutats och "background" får därmed aldrig tillfälle att exekvera. Programmet "låser sig" och antingen "ctrl-break" eller "ombootning" blir nödvändigt för att avbryta programmet.

Ovanstående problem skulle kunna lösas om man på något sätt kunde skatta processorkapaciteten. Ett sätt vore att med hjälp av systemklockan beräkna exekveringstiden för ett stort antal loopar. Utifrån detta värde skulle man kunna beräkna lämpligt antal loopar för omvandlingstiden och även maximal samplingshastighet. Denna procedur skulle utföras en gång vid varje ny start av programmet.

För att få ett mer exakt värde av avbrottsrutinens exekveringstid skulle funktionen "foreground()" kunna anropas med ett "vanligt" funktionsanrop. Genom att göra ett större antal sådana anrop skulle systemklockan med sin relativt dåliga upplösning kunna användas för att beräkna exekveringstiden.

Att använda "Foreground/Background-scheduler" för att beräkna exekveringstiden är däremot inte lämpligt eftersom avbrottsshantering skulle påverka exekveringstiden i bakgrundsprocessen. Därmed skulle ett systematiskt fel uppstå.

8.3 Maximering av samplingsfrekvens

För att kunna uppnå samplingsfrekvensen 1000 Hz även vid exekvering på en 8086-processor (se kap 7.2) kan vissa modifieringar av avbrottsrutinen göras.

Ett sätt är att utnyttja omvandlingstiden vid sampling på ett effektivare sätt. I stället för att genomgå en vänteloop kan t.ex. rotutdragning av flödessamplet utföras under denna tid. På detta sätt kan man "tjäna" 120 mikrosekunder (tre vänteloopar á 40 mikrosekunder). Detta är dock inte tillräckligt för att uppnå frekvensen 1000 Hz.

Den största tidsvinsten kan göras vid den tidskrävande rotutdragningsalgoritmen där det framför allt är de tio divisionerna som tar tid. Ett enkelt sätt att undvika operationen är att använda en tabell. Tabellen skulle kunna beräknas eller läsas in från fil vid varje ny start av programmet. Därigenom skulle man även kunna ta hänsyn till flödesgivarens eventuella olinjäriteter. Dessutom skulle man kunna uppnå en större noggrannhet än de 64 olika kvantiseringsnivåer som den nuvarande algoritmen medger. Nackdelen är att en tabell upptar en del minnesutrymme, framför allt i "datasegment-arean". Detta borde dock inte innebära några problem eftersom det (för närvarande) endast är kritiskt vad gäller "kodsegment-arean".

8.4 Datorstyrd öppning av ocklusionsventil

En nackdel med den nuvarande mätutrustningen är att öppningstrycket för ocklusionsventilen ställs in med en manuell "ratt". Efter en inställning måste först en inandning genomföras, innan det nya öppningstrycket kan beräknas. Eventuellt måste även en omkalibrering göras för att anpassa triggnivån till det nya vakuumentrycket. En elegantare lösning vore att kunna ställa in önskat öppningstryck från datorn. Detta kan lösas på olika sätt.

Ett sätt är att använda sig av en elektriskt reglerbar vakuumpump. Vakuumentrycket och därmed öppningstrycket skulle då kunna regleras från datorn. Vid en ändring av önskat öppningstryck skulle en grovinställning av lämplig styrspänning till pumpen kunna beräknas utifrån en tabell eller en kalibreringskurva. Därefter skulle styrspänningen kontinuerligt kunna finjusteras beroende av uppmätta öppningstryck.

Ett annat sätt vore att använda sig av en elektriskt styrd "switch"-ventil för vakuumtrycket. Vakuumtrycket skulle då läggas på en konstant hög nivå. När luftvägstrycket vid en inandning understiger det önskade öppningstrycket, kopplar datorn bort vakuumtrycket med hjälp av ventilen och ocklusionsventilen öppnas. En fördel jämfört med den första lösningen är att speciella kalibreringsfunktioner och eventuella regleralgoritmer inte behövs för att beräkna lämplig styrsignal till vakuumpumpen. Metoden kan dessutom lätt implementeras med befintlig programvara. Nackdelen är dock att lösningen förutsätter en snabb bortkoppling av vakuumtrycket. För att uppnå en noggrannhet på ca 0.3 cmH₂O av öppningstrycket krävs en omslagstid på ca 3 ms. Den nuvarande elektriska ventilen har tillsammans med slangar en omslagstid på ca 12 ms. Största delen av denna omslagstid beror på att ventilen även i öppet tillstånd har ett mycket begränsat flöde. Detta medför att det tar relativt lång tid för vakuumtrycket i slangarn att utjämnas när ventilen öppnas. En ventil med mindre flödesmotstånd dvs större diameter skulle antagligen minska omslagstiden betydligt.

8.5 Statistiska analyser av inhämtad andningsdata

Eftersom nuvarande insamlad andningsdata är relativt svåröverskådlig och måste behandlas i separata statistiska program för att slutsatser ska kunna dras, skulle det vara önskvärt om enklare statistiska beräkningar kunde utföras i direkt anslutning till mätdatainsamlingen. Behandlingen av data skulle kunna innefatta enklare statistiska beräkningar såsom trender och regressionsanalyser. För detta ändamål har det reserverats plats i "rullgardinsmenyn" - Modes/Statistics.

9 Referenser

- Shoemaker, Richard L. (1986): "The IBM PC from the Inside Out", Addison-Wesley Publishing, USA
- Brück, Dag (1988): "A foreground/background real-time scheduler for the IBM AT",
Inst. f. Reglerteknik, LTH, TFRT-7393
- Waite, Mitchell (1990): "Microsoft Quick C Programming", Penguin Books Ltd, USA
- Sequiter software inc. (1988): "Code Base4 - the C library for database and screenmanagement",
Canada
- Whitelaw, William A (1975) "Occlusion pressure as a measure of respiratory center", Dep. of
physiology, McGill University, Canada