```
{
*************************************************************
            F A G P R Ø V E N  V Å R E N  1 9 9 9
                  Universitetet i Bergen
            Institutt for den Faste Jords Fysikk

                A D - K O R T  P R O G R A M
                 Skrevet av Anders ...


Beskrivelse :   Programmet skal kjøres på et Flashlite kort
                med V25+ prosessor. produsert av JKMicro.
                Det skal kommunisere med en RS-232 linje
                og et AD-kort. Når <SPACE> mottas serielt
                skal AD-kortet sample en gang i sekundet
                og sende dataene fortløpende ut serielt. Denne
                rutinen avsluttes når vilkårlig karakter mottas.
                Når 'B' eller 'b' mottas skal AD-kortet sample
                2048 ganger med en samplingsrate på 500Hz. Så
                skal alle dataene sendes ut serielt i en blokk.
                I tillegg avsluttes programmet når 'Q' eller 'q'
                mottas serielt.
CopyRight   :   Universitetet i Bergen,
                Institutt for den Faste Jords Fysikk (1999)
Språk       :   Turbo Pascal 7.0

*************************************************************



*************************************************************
*                  T I M E R  C O N S T A N T S             *
*************************************************************
*           10MHz Klokke: 12.8us timer resolution.          *
*           0.1ms = 7.8125 tics ->     8 tics =   0.1024ms  *
*           1.0ms = 78.125 tics ->    78 tics =   0.9984ms  *
*          10.0ms = 781.25 tics ->   781 tics =   9.9968ms  *
*         100.0ms = 7812.5 tics ->  7813 tics = 100.0064ms  *
*************************************************************



*************************************************************
*          C O N T R O L  B Y T E  -  A D  K O R T          *
*************************************************************
*   D7      D6      D5      D4      D3      D2      D1      D0  *
*-----------------------------------------------------------*
*   0       1       0       RNG     BIP     A2      A1      A0  *
*************************************************************



*************************************************************
*          R A N G E  T A B L E  -  A D  K O R T            *
*************************************************************
*   RNG     BIP         INPUT RANGE (V)                     *
*-----------------------------------------------------------*
*   0       0               0 - 5                           *
*   1       0               0 - 10                          *
*   0       1                 ñ5                            *
*   1       1                 ñ10                           *
*************************************************************



*************************************************************
*       C H A N N E L  T A B L E  -  A D  K O R T           *
*************************************************************
*   A2  A1  A0      CH0 CH1 CH2 CH3 CH4 CH5 CH6 CH7         *
*-----------------------------------------------------------*
*   0   0   0        X                                      *
*   0   0   1            X                                  *
```

```
*   0   1   0               X                                   *
*   0   1   1                   X                               *
*   1   0   0                       X                           *
*   1   0   1                           X                       *
*   1   1   0                               X                   *
*   1   1   1                                   X               *
*************************************************************


}

{$N-} {Ingen numerisk co-prosessor}

Program AD_kort;


Uses
    Dos; {Unit som inneholder de mest brukte rutinene}


Const
    constTimerInterval      = 1;    {Timer interval i ms}

    constNumbOfSamples      = 2048; {Antall samplinger ved rate 500HZ}
    constSampleInterval     = 2;    {Samplingsinterval i ms ved rate 500Hz}
    constContSampleInterval = 1000; {Samplingsinterval i ms ved rate 1Hz}

    constADCh0              = 0;    {Konstant for å velge AD kanal 0}

    constAD_Uni5Volts       = 0;    {Konstant for å velge AD input range 0-5V}
    constAD_Uni10Volts      = 2;    {Konstant for å velge AD input range 0-10V}
    constAD_Bip5Volts       = 1;    {Konstant for å velge AD input range ñ5V}
    constAD_Bip10Volts      = 3;    {Konstant for å velge AD input range ñ10V}

    Seg     =   $F000;  {Segment adresse for CPU special registers}

    SRIC0   =   $FF6D;  {Receive completion interrupt request register, port 0}
    RxB0    =   $FF60;  {Receive buffer}

    MD1     =   $FF8A;  {Modulo/Timer register for timer 1}
    TMC1    =   $FF91;  {Timer control register for timer 1}
    TMIC2   =   $FF9E;  {Timer unit interrupt controll register, timer 1}

    AD_Base =   $2B0;   {Base adresse til AD-konverter kort}


Var
    TimerLimit          : Word;     {Verdi timertelleren ikke må overstige}

    TimerCounter        : Word;     {Variabel timeren øker}

    NumberOfSamples     : Word;     {Antall samplinger}

    {Her lagres 2048 samplinger når samplingsraten er 500Hz}
    AD_values           : Array [1..constNumbOfSamples] of Integer;


    Int13Vector         : Pointer;  {Peker til gammel serieport 0
                                     receive interrupt rutine}
    Int30Vector         : Pointer;  {Peker til gammel timer 1
                                     interrupt rutine}

    EndProgram          : Boolean;  {Flagg som indikerer om
                                     programmet skal avsluttes}
    Serial0Interrupt    : Boolean;  {Flagg som indikerer at
                                     serielle data er mottatt}

    ContinuousSampling  : Boolean;  {Flagg som indikerer 1Hz
                                     kontinuerlig sampling}
```

```
    Sampling            : Boolean;  {Flagg som indikerer at
                                     sampling skal utføres}



(*************************************************
            I N I T I A L I S E R I N G
*************************************************)

Procedure VarInitialize;

Begin
    {Beregner øvre grense for timer teller}
    TimerLimit          := (65536 - constTimerInterval);

    {initialiserer noen variabler}
    TimerCounter        := 0;

    EndProgram          := False;
    Serial0Interrupt    := False;

    ContinuousSampling  := False;
    Sampling            := False;

    NumberOfSamples     := 1;
End;


Procedure InitTimers;
Var
    MD1_Number: Word;

Begin
    {Initialiserer Timer 1 til 1ms interval}
    MD1_Number:= Round(constTimerInterval * 78.125);
    memw[Seg:MD1]:= MD1_Number;

    {Starter timer}
    mem[Seg:TMC1]:= $C0;
End;


(*************************************************
                    U T I L S
*************************************************)

Procedure VeiledningsText; {Instruksjonstekst som sendes ut serielt}

Begin
    Writeln;
    Writeln;
    Writeln('Trykk "SPACE" for kontinuerlig sampling, 1Hz samplingfrekvens.');
    Writeln('Trykk "B" eller "b" for 2048 samplinger, 500Hz samplingfrekvens.');
    Writeln;
    Writeln('Trykk paa en vilkaarlig bokstav for aa stoppe kontinuerlig sampling');
    Writeln;
    Writeln('Trykk "Q" eller "q" for aa avslutte');
End;


Procedure SampleADCard(Chan: Byte; Gain: Byte;
                       SampleInterval,NumbOfSamples: Word);

Var
    ADControlValue  : Byte;         {Inneholder kontrolverdien til AD kort}
    I,CounterOffset : Word;
    ADDataValue     : Integer;      {Inneholder dataverdier fra AD kort}

Begin
```

```
            Sampling        := False;

            TimerCounter    := 0;

            {Initialize AD-card control value}
            ADControlValue  := ((Gain Shl 3) And $18);
            ADControlValue  := (ADControlValue Or (Chan And $07) Or $40);

            {Utføres ved 500Hz samplingsrate (ikke kontinuerlig sampling)}
            If Not(ContinuousSampling) Then Begin

                {Synkroniserer}
                CounterOffset   := TimerCounter;

                While (TimerCounter < (CounterOffset + 1)) Do;
            End;


            {Start sampling}
            For I:= 1 To NumbOfSamples Do Begin

                CounterOffset   := TimerCounter;

                {Sampler}
                Port[AD_Base]   := ADControlValue;

                {Vent 1ms}
                While (TimerCounter < (CounterOffset + 1)) Do;

                {Leser data}
                ADDataValue     := PortW[AD_Base];

                {Skriver data til serieport 0 ved kontinuerlig sampling, hvis ikke
                lagres data i et array som senere skrives til serieport 0}
                If ContinuousSampling Then Writeln(ADDataValue:5)
                Else Begin
                    {Lagrer data i et array}
                    AD_values[I]    := ADDataValue;

                    {Vent}
                    While (TimerCounter < (CounterOffset + SampleInterval)) Do;
                End;

            End;

            {Skriver ut data til serieport 0 ved 500Hz samplingsrate
            (ikke kontinuerlig sampling}
            If Not(ContinuousSampling) Then Begin
                {Sender data til serieport}
                For I:= 1 To NumbOfSamples Do Writeln(AD_Values[I]:5);

                {Sender veiledningstekst til serieport}
                VeiledningsText;
            End;

        End;


        Procedure CheckChar; {Tester hvilken bokstav som mottas serielt}

        Var
            RawData : Byte;
            TestChar: Char;

        Begin
            Serial0Interrupt:= False;

            {Leser seriell data fra receive buffer}
            RawData:= mem[Seg:RxB0];
```

```
        {Konverterer data til ASCII tekst, store bokstaver}
        TestChar:= UpCase(Chr(RawData));

        {Sjekker bokstav}
        Case TestChar of

            ' ':    Begin
                        {Starter 1Hz sampling}
                        ContinuousSampling  := True;
                        NumberOfSamples     := 1;
                    End;


            'B':    Begin
                        {Starter 500Hz sampling, stopper 1Hz sampling}
                        Sampling            := True;
                        ContinuousSampling  := False;
                        NumberOfSamples     := constNumbOfSamples;
                    End;

                    {Avslutter program}
            'Q':    EndProgram:= True;

        Else Begin
            {Stopper 1Hz sampling}
            ContinuousSampling  := False;
            VeiledningsText;
        End;

        End;
    End;


    (*******************************************
                I N T E R R U P T S
    *******************************************)

    Procedure HandleSerial0; Interrupt;

    Begin
        {Resetter interuppt flagg og disabler interrupt}
        mem[Seg:SRIC0]:= $47;

        {Klarsignal for å lese data}
        Serial0Interrupt:= True;

        {Enabler interrupt}
        mem[Seg:SRIC0]:= $7;

        {Slutt på interrupt}
        Inline($0F/$92);
    End;


    Procedure HandleTimer1; Interrupt;

    Begin
        {Resetter interupt flagg og disabler interrupt}
        mem[Seg:TMIC2]:= $47;

        {•ker teller}
        If TimerCounter>= TimerLimit Then TimerCounter:= 0;
        Inc(TimerCounter, constTimerInterval);

        {Enabler interrupt}
        mem[Seg:TMIC2]:= $7;

        {Slutt på interrupt}
```

```
    Inline($0F/$92);
End;


Procedure EnableSerialsInterrupt;

Begin
    {Lagrer DOS interrupt vector}
    GetIntVec(13, Int13Vector);

    {Setter interrupt vector til 'HandleSerial0'}
    SetIntVec(13, @HandleSerial0);

    {Enabler seriell 0 receive interrupt}
    mem[Seg:SRIC0]:= $7;
End;


Procedure EnableTimersInterrupt;

Begin
    {Lagrer DOS interrupt vector}
    GetIntVec(30, Int30Vector);

    {Setter interrupt vector til 'HandleTimer0'}
    SetIntVec(30, @HandleTimer1);

    {Enabler timer 1 interrupt}
    mem[Seg:TMIC2]:= $7;
End;


(*******************************************************
        A V S L U T N I N G S P R O S E D Y R E R
*******************************************************)

Procedure DisableSerialsInterrupt;

Begin
    {Disabler seriell 0 receive interrupt}
    mem[Seg:SRIC0]:= $47;

    {Setter interruptvektor tilbake til DOS rutine}
    SetIntVec(13, Int13Vector);
End;


Procedure DisableTimersInterrupt;

Begin
    {Disabler timer 1 interrupt}
    mem[Seg:TMIC2]:= $47;

    {Setter interruptvektor tilbake til DOS rutine}
    SetIntVec(30, Int30Vector);
End;


Procedure StopTimers;

Begin
    {Stopper timer 1}
    mem[Seg:TMC1]:= $0;
End;


(*********************************************
        M A I N  P R O C E D U R E
*********************************************)
```

```
Begin
    VarInitialize;              {Initaliserer variabler}

    EnableSerialsInterrupt;     {Oppretter handle for interrupt
                                 fra seriell 0 receiver}

    EnableTimersInterrupt;      {Oppretter handle for interrupt
                                 fra timer 1}

    InitTimers;                 {Setter opp timer interval}

    Writeln('************************************************');
    Writeln('*     F A G P R 0 V E N  V A A R E N  1 9 9 9     *');
    Writeln('*                                                *');
    Writeln('*              Universitetet i Bergen            *');
    Writeln('*       Institutt for den Faste Jords Fysikk     *');
    Writeln('*                                                *');
    Writeln('*           A D - K O R T  P R O G R A M         *');
    Writeln('*            Skrevet av Anders .........          *');
    Writeln('************************************************');

    VeiledningsText;            {Skriver veiledningstekst}


    {Program loop, går helt til programmet avsluttes}
    Repeat
        {Sjekker mottatt bokstav ved seriell 0 receive interrupt}
        If Serial0Interrupt Then CheckChar;

        {Starter 1Hz sampling hvis flagg er satt}
        If ContinuousSampling Then
            If (TimerCounter >= 1000) Then
            SampleADCard(constADCh0, constAD_Bip5Volts,
                         constSampleInterval, NumberOfSamples);

        {Starter 500Hz sampling hvis flagg er satt}
        If Sampling Then
        SampleADCard(constADCh0, constAD_Bip5Volts,
                       constSampleInterval, NumberOfSamples);

    Until EndProgram;

    DisableSerialsInterrupt;    {Setter handle for seriell 0 receive
                                 interrupt tilbake til DOS}

    DisableTimersInterrupt;     {Setter handle for timer 1 interrupt
                                 tilbake til DOS}

    StopTimers;                 {Stopper timer 1}
End.
```