Handleiding voor het werken met DT-2821 data-acquisitiekaarten voor de IBM AT Personal Computer en compatibles

Handleiding voor het werken met DT-2821 data-acquisitiekaarten voor de IBM AT Personal Computer en compatibles

Theo Heeren

WFW rapport nr. 87068

September 1987

Inhoudsopgave :

Pagina Hoofdstuk

Appendices :

Literatuur :

[1] : The IBM Personal Computer AT Technical Reference

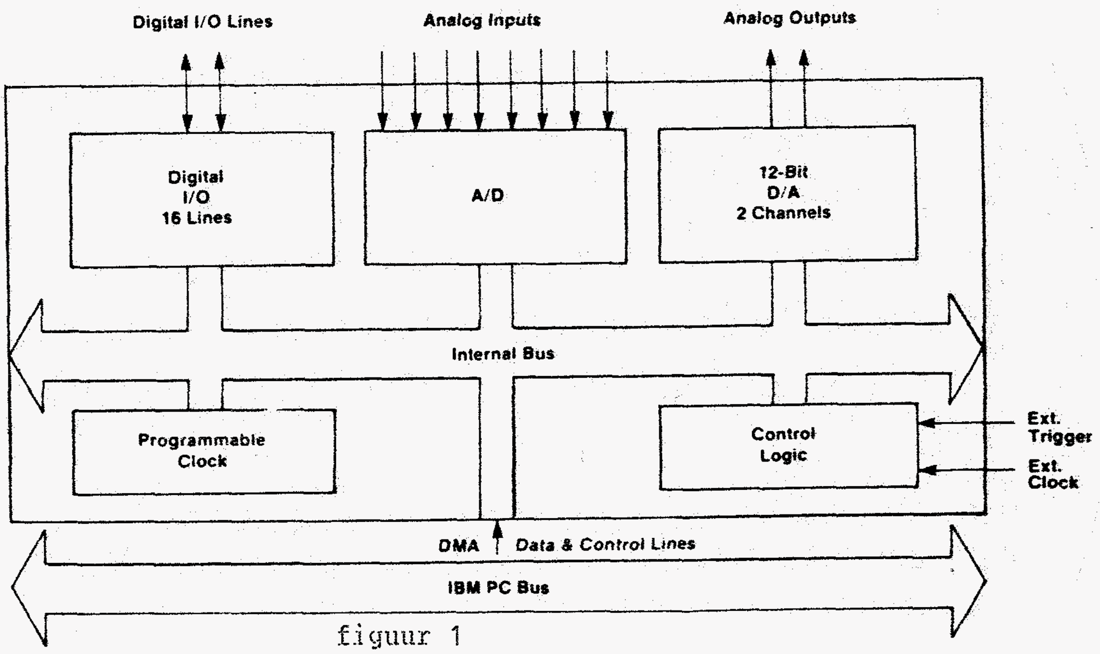
[2] : The DT2821 series user manual (Data Translation)

Hoofstuk 1 : Inleiding

In deze handleiding wordt beschreven welke mogelijkheden de data-acquisitiekaarten uit de DT2821 serie van Data Translation hebben en hoe de daarbij behorende software gebruikt moet worden. Data-acquisitiekaarten dienen hoofdzakelijk voor het naar binnen halen en naar buiten sturen van analoge data (in de vorm van analoge elektrische spanningen) door een PC. Daarnaast kunnen ze ook digitale in- en output voor hun rekening nemen. De struktuur van de kaart zal globaal beschreven worden. De PC moet een IBM-AT of daarmee compatibel zijn. De kaart moet worden aangebracht in een van de lege expansion slots van de PC.
Voor een goed begrip van de inhoud van de handleiding is enige voorkennis gewenst. Men moet bekend zijn met het werken met PC's (MSDOS) en met een of meerdere hogere programmeertalen die op PC's gebruikt worden, bv. Turbo Pascal, Fortran, Basic. Verder moet enige elementaire kennis over de struktuur van een PC aanwezig zijn : een centrale processor chip (CPU), met daaromheen een hoeveelheid RAM-geheugen, die via I/O-poorten communiceert met zijn omgeving. Voor het snel uitwisselen van data tussen randapparatuur en geheugen wordt de funktie van de CPU overgenomen door een speciaal daarvoor bedoelde chip, een zogenaamde DMA-controller.
De software bestaat uit een aantal in assembly geschreven routines die bedoeld zijn om vanuit een hogere programmeertaal aangeroepen te worden.

Hoofdstuk 2 : Beschrijving van struktuur en werking van de kaart

De DT2821 serie bestaat uit een aantal data-acquisitiekaarten voor de IBM PC AT of compatibles, die anno 1987 tot de technisch meest geavanceerde gerekend moeten worden. Dit zijn de DT2821 die ca. 50.000 A/D- of D/A-conversies per sekonde kan doen, de DT2821-F (ca. 160.000 conversies per sekonde), de DT2821-G (ca. 260.000 conversies per sekonde), de DT2825, de DT2823 en de DT2828, die vier kanalen tegelijkertijd kan bemonsteren. De struktuur van de kaarten is afgebeeld in figuur 1 :



figuur 1

De kaart-typen verschillen in conversiesnelheid, resolutie , spanningsbereiken van de A/D- en D/A-converters en de programmeerbare versterkingsfaktoren waarmee analoge signalen versterkt kunnen worden alvorens een conversie plaatsvindt. De prijzen variëren van 4000 tot 9000 gulden (1987). Voor gedetailleerde informatie hierover zij verwezen naar appendix C. We zullen ons bij de verdere beschrijving beperken tot de typen DT2821 en DT2821-F. In figuur 1 kunnen vijf gedeelten onderscheiden worden :

-een A/D sektie, die de analoge ingangsspanningen converteert tot digitale signalen met een resolutie van 12 bits

-een D/A sektie, die digitale signalen converteert naar analoge spanningen, eveneens met een resolutie van 12 bits

-een digitale I/O sektie, die 16 I/O lijnen bevat die per groep van 8 lijnen als invoer- of uitvoerlijnen kunnen fungeren

-een programmeerbare klok. Deze klok genereert pulsen met regelmatige tussenpozen. De pulsen kunnen dienen als startsignaal voor A/D- of D/A-conversies

-een stuk controle logica dat de voorafgaande gedeelten bestuurt en de communicatie met de PC verzorgt

We zullen ieder van de gedeelten nu nader omschrijven.

De A/D sectie bevat 16 analoge ingangen. Van ieder van de ingangen kan op een bepaald tijdstip een sample genomen worden. Vóórdat de conversie begint wordt het analoge ingangssignaal versterkt met een programmeerbare faktor: 1, 2, 4 of 8. Daarna vindt de conversie plaats met een resolutie van 12 bits (voor de DT2823 is de A/D- en D/A-resolutie 16 bits). De volgorde waarin de ingangen gesampled worden, de faktor waarmee iedere ingang versterkt moet worden en het aantal ingangen dat gesampled moet worden, worden vastgelegd in de zogenaamde Channel Gain List, een stukje geheugen op de data-acquisitiekaart dat door de PC van data voorzien wordt. Vóór het starten van een serie conversies wordt de Channel Gain List geladen met de gewenste kanaalvolgorde en de gewenste versterkingsfaktoren. De Channel Gain List biedt geheugenplaatsen voor maximaal 16 conversies (16 kanaalnummers, 16 versterkingsfaktoren). Indien er meer conversies gedaan moeten worden dan het aantal conversies opgeslagen in de Channel Gain List wordt de kanaalvolgorde in de Channel Gain List steeds gerepeteerd. Het is ook mogelijk om hetzelfde kanaal meerdere malen in de Channel Gain List op te nemen, indien gewenst met verschillende versterkingsfaktoren. Het tijdstip waarop een sample van een ingang genomen wordt, wordt bepaald door de pulsen van een klok, die op zijn beurt weer door een trigger-signaal gestart wordt. Op de mogelijkheden van de klok en de triggersignalen zal bij de klok-sektie nader ingegaan worden. Bij het verrichten van een serie conversies kent het

A/D-subsysteem twee verschillende modes, namelijk A/D clocked DMA mode en A/D triggered scan DMA mode. Bij A/D clocked DMA wordt na het geven van een triggersignaal bij iedere klokpuls een conversie uitgevoerd. Steeds wanneer het einde van de Channel Gain List bereikt wordt, gaan de conversies weer verder vanaf het begin. Bij A/D triggered scan DMA worden na het geven van een triggersignaal de conversies vanaf het begin tot het einde van de Channel Gain List uitgevoerd (bij iedere klokpuls èèn conversie). Daarna wacht het A/D-subsysteem op het volgende triggersignaal om de volgende "scan" van de Channel Gain List te starten. In beide modes gaat de conversie-serie zolang door totdat een aantal van te voren gespecificeerde geheugenbuffers volgeschreven is met conversieresultaten. Het maximum aantal A/D conversies per sekonde bedraagt bij de DT2821 ca. 50.000 en bij de DT2821-F ca. 160.000. Er zijn twee ranges mogelijk waarbinnen de analoge ingangsspanningen moeten liggen : 0 tot +10 Volt (unipolair) of -10 tot +10 Volt (bipolair). De range wordt met schakelaars op de kaart zelf ingesteld en kan dus niet softwarematig veranderd worden.

De D/A sektie bevat 2 analoge uitgangen, die ieder hun eigen 12 bits D/A converter hebben (voor de DT2823 is de resolutie 16 bits). De converters kunnen ieder apart of simultaan werken. Evenals bij de A/D conversies worden de D/A conversies gestart door de pulsen van een klok. Bij het uitvoeren van een serie conversies opereert de D/A sektie in de zogenaamde D/A clocked DMA mode waarbij de digitale data voor èèn of beide converters gelezen wordt uit een aantal van te voren gespecificeerde geheugenbuffers. De conversies gaan zolang door totdat alle buffers afgewerkt zijn. Het maximum aantal D/A conversies per sekonde en per converter is voor de DT2821 ca. 50.000 en voor de DT2821-F ca. 130.000. Er zijn vijf ranges mogelijk waarbinnen de uitgangsspanningen kunnen liggen : 0 tot 5 Volt, 0 tot 10 Volt, +/- 2.5 Volt, +/- 5 Volt en +/- 10 Volt. De range wordt met schakelaars op de kaart ingesteld en is dus niet programmeerbaar.

De digitale I/O sektie bestaat uit 16 I/O lijnen die verdeeld zijn in twee groepen van 8 lijnen. Iedere groep kan afzonderlijk als groep van inputlijnen of als groep van outputlijnen geprogrammeerd worden. Van een groep inputlijnen kan op ieder gewenst moment gelezen worden en naar een groep outputlijnen kan op ieder gewenst moment geschreven worden. De logische waarde 1 komt overeen met ca. 5 Volt en de logische waarde 0 met ca. 0 Volt.

De programmeerbare klok genereert pulsen die kunnen worden gebruikt voor het starten van A/D- en/of D/A-conversies. De tijd die verstrijkt tussen twee pulsen is vrij te kiezen en kan variëren van 250 nanosekonden tot 2.097 sekonden. Het kleinst mogelijke inkrement bedraagt 250 nanosekonden. Het is dus mogelijk om een het aantal klokpulsen per sekonde groter te kiezen dan het maximum aantal A/D of D/A conversies per sekonde. In dat geval zal de kaart niet kapot gaan, maar onbetrouwbaar werken. Voor het geval de range of de nauwkeurigheid van de klok niet groot genoeg is bestaat ook de mogelijkheid om de funktie van de klok op de kaart over te laten nemen door een externe klok, die door de gebruiker moet worden aangeleverd en op een aparte klok-ingang van de kaart aangesloten moet worden. Het is mogelijk om softwarematig te kiezen tussen de klok op de kaart of de externe klok. De klok op de kaart kan worden gestart met een trigger. Dit trigger-signaal kan zowel vanuit de PC naar de kaart gestuurd worden als op een aparte externe trigger-ingang op de kaart aangeboden worden. Een conversie of serie van conversies wordt altijd gestart door een trigger. De trigger stelt de klok in staat om A/D of D/A conversies te initialiseren. De eerste conversie van een serie conversies wordt dus geinitialiseerd door de eerste klok-puls na het geven van het trigger-signaal.

Voor het geval we maar één conversie willen hebben is het mogelijk om de A/D- en D/A-sektie afzonderlijk los te koppelen van de klok en alleen een trigger-signaal te geven. Dit loskoppelen is ook programmeerbaar.

De controle logica zorgt voor de correcte aansturing van alle onderdelen van de kaart en neemt de communicatie tussen de PC en de kaart voor zijn rekening. De communicatie met de kaart geschiedt door data te schrijven naar en te lezen van een achttal 16 bits I/O registers, die nader omschreven worden in appendix C en in het DT2821 series user manual [2].

De analoge en digitale in- en uitgangen komen de PC binnen door middel van een connector aan de achterkant van de data-acquisitiekaart. Voor de funktie

van iedere connectoraansluiting wordt verwezen naar het DT2821 series user manual [2]. Langs deze weg kan de gebruiker de te verwerken signalen aanbieden aan de PC. Het is mogelijk om op de connector een DT707 screw terminal panel aan te sluiten. Dit is een paneel dat voorziet in een robuuste schroefaansluiting voor iedere analoge en digitale in- en uitgang, alsmede voor de externe klok en de externe trigger. Voor meer informatie over de DT707 en andere typen screw terminal panels wordt verwezen naar appendix C en het DT2821 series user manual [2].
De kaart is redelijk beveiligd tegen het aanbieden van te grote spanningen en het veroorzaken van kortsluiting aan de connectoraansluitingen.

Hoofdstuk 3 : Datatransfer tussen PC-geheugen en data-acquisitiekaart

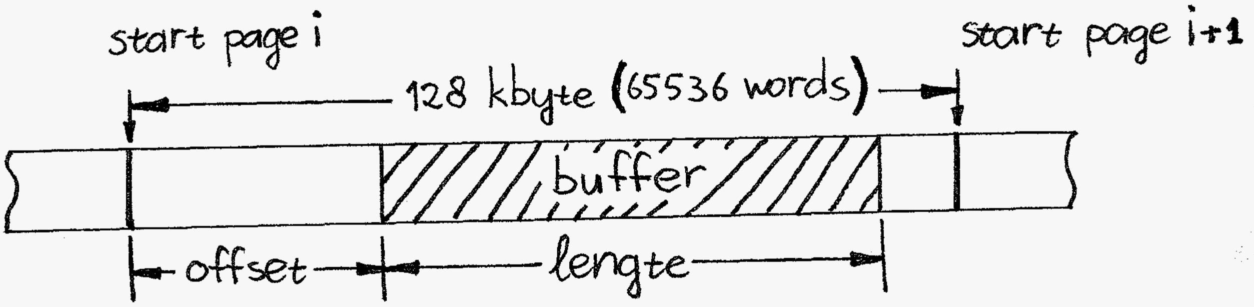
De data transfer tussen de kaart en het PC-geheugen, nodig voor het opbergen van A/D conversie-resultaten, het aanbieden van data aan de D/A converters, voor de digitale I/O en voor het besturen van de kaart, kan geschieden op drie manieren :

- Programmed I/O. Door een kommando naar een van de I/O poorten te schrijven kan een A/D of D/A conversie worden gestart (een trigger-kommando). Door de data van een andere I/O poort te lezen kan statusinformatie verkregen worden waaruit kan worden opgemaakt of een conversie voltooid is. Indien dat het geval is kan het resultaat van een A/D konversie gelezen worden van weer een andere I/O poort. Deze werkwijze heeft het grote nadeel dat de PC voortdurend moet checken of een conversie voltooid is, waardoor veel tijd van de PC in beslag genomen wordt tijdens de data-acquisitie. De digitale I/O en de besturing van de kaart kunnen uitsluitend via programmed I/O geschieden.
- Interrupt driven I/O. De kaart kan ook zodanig geprogrammeerd worden dat na de voltooiing van een conversie een interrupt sigaal door de kaart gegenereerd wordt. De PC kan daar dan op reageren door het programma waar hij op dat moment mee bezig is te onderbreken en het resultaat van een A/D conversie van een I/O register te lezen, op te slaan in het geheugen en daarna weer verder gaan met datgene waar hij mee bezig was. Deze werkwijze heeft het voordeel dat de PC niet de hele tijd de statusinformatie moet checken en dus tijd over heeft voor andere zaken. Het uitvoeren van A/D of D/A conversies en het draaien van programma's op de PC kan nu simultaan gebeuren. Met name wanneer er een hele serie conversies achter elkaar gedaan moet worden en de tijdstippen waarop ze moeten worden uitgevoerd ver uit elkaar liggen is deze werkwijze erg aantrekkelijk.
- Direct Memory Acces (DMA). Indien er een grote serie conversies gedaan moet worden en de conversies snel na elkaar moeten gebeuren (bv.10.000

conversies per sekonde of meer), dan is de PC ondanks het feit dat gebruik gemaakt wordt van Interrupt Driven I/O toch nog een groot gedeelte van de beschikbare tijd bezig met data transfers van of naar de kaart. Indien er meer dan ca. 100.000 conversies per sekonde gedaan moeten worden, is de data tranfersnelheid zelfs zo groot dat de Centrale Processor Unit van de PC deze niet meer aankan. In die gevallen moet gebruikt gemaakt worden van een speciaal voor snelle datatransfer ontworpen chip, een zogenaamde DMA-controller. Hiervan zijn er twee aanwezig in iedere IBM PC AT compatibele PC. Alleen de tweede kan gebruikt worden door de data-acquisitiekaart. De kaart kan zodanig geprogrammeerd worden dat na voltooiing van een conversie een signaal aan de DMA-controller gegeven wordt, die daarop de programmaverwerking door de CPU gedurende korte tijd (minder dan 1 mikrosekonde) onderbreekt om de datatransfer tussen de kaart en het PC geheugen uit te voeren. Zodoende neemt zelfs bij 200.000 conversies per sekonde de datatransfertijd minder dan 20% van de beschikbare tijd in beslag en kan er dus meer dan 80% van de tijd besteed worden aan programmaverwerking door de CPU. Voor series van A/D of D/A conversies is deze werkwijze zonder meer te verkiezen boven alle andere en daarom zal voor conversie-series uitsluitend van DMA data transfer gebruik gemaakt worden.

Hoofdstuk 4 : DMA-buffers

Indien de kaart gebruik maakt van DMA data transfer moet er data uitgewisseld worden tussen de data-acquisitiekaart en een zogenaamde DMA-buffer. Een DMA-buffer is een aaneengesloten deel van het geheugen van de PC dat door de DMA-controller geadresseerd kan worden. Omdat de gebruiker zelf kan uitmaken waar in de PC-geheugenruimte DMA-buffers mogen liggen, is het van belang om wat meer te weten over de geheugen-indeling en de adressering daarvan door de DMA-controller. De kleinste geheugen-eenheid, die door de CPU geadresseerd kan worden heet een byte. Een byte is groep van 8 bits (afkorting voor binary digits), die ieder de logische waarden 0 of 1 kunnen aannemen. De totale geheugenruimte van de PC omvat 16 Megabyte (1 Megabyte = 1024 * 1024 = 1048576 byte). De Intel 80286 CPU van de PC heeft 24 adreslijnen waarmee 2 tot de macht 24 byte (= 16 Megabyte) ieder afzonderlijk geadresseerd kunnen worden. De tweede DMA-controller van de PC, die alle transfers tussen de kaart en een DMA-buffer voor zijn rekening neemt, adresseert dezelfde 16 Megabyte op zijn eigen manier. De geheugenruimte wordt onderverdeeld in 128 pages ("bladzijden") van 128 kilobyte (1 kilobyte = 1024 byte) aaneengesloten geheugen. Het nummer van een page, die door de DMA-controller geadresseerd wordt, staat in een page register, behorend bij één van de drie voor randapparatuur beschikbare DMA-kanalen van de DMA controller. Omdat het nummer van een page tijdens de DMA-operatie niet kan veranderen, moet een DMA-buffer in ieder geval binnen een page liggen, met andere woorden : DMA buffers mogen page-grenzen niet overschrijden. Een DMA-buffer wordt naast het page-nummer gekenmerkt door twee andere parameters : een offset vanaf de onderste page-grens tot aan het begin van de buffer en de lengte van de buffer. Een en ander wordt verduidelijkt in figuur 2 :



figuur 2

De DMA-controller kan het geheugen alleen in words adresseren. Een word is een eenheid van twee naburige bytes, waarvan de eerste op een even geheugenadres moet liggen (de minst significante bit van het 24 bits adres is bij een DMA-transfer altijd 0). Daarom wordt de offset en de lengte van een buffer ook gemeten in words. Het zal duidelijk zijn dat de som van offset en lengte van een buffer nooit groter mag zijn dan 65536 (= 64 * 1024) words, omdat anders een page-grens overschreden zou worden.

Bij het kiezen van de bufferlocaties en de bufferlengten moeten we ervoor oppassen dat er geen geheugen overschreven wordt waar waardevolle data instaat, bv. het geheugen dat in beslag genomen wordt door BIOS, MSDOS of een programma dat aan het draaien is. Raadpleeg hiervoor de memory map in het IBM AT Technical Reference Manual [1]. Verder moeten buffers altijd gelegd worden in die delen van de geheugenruimte, waar RAM geheugen aanwezig is. Bij de PC AT is het grootste gedeelte van de geheugenruimte leeg. Schrijven naar lege geheugenplaatsen betekent verlies van data, lezen van lege plaatsen binnenhalen van random data. Het is aan te bevelen om de PC te voorzien van een 512 kilobyte geheugenuitbreiding, die loopt van de hexadecimale adressen 100000h tot en met 17FFFFh. Hierin zouden bijvoorbeeld vier buffers ter lengte van 65536 words gelegd kunnen worden. Om de keuze van de bufferligging iets te vergemakkelijken is in de software een routine opgenomen, die een in een hogere programmeertaal gedeclareerd programma-array tot buffer kan verklaren, waardoor de A/D- of D/A-data direct vanuit het programma toegankelijk is en we zeker weten dat het gedeelte van het RAM-geheugen dat door het array in beslag genomen wordt, nergens anders voor bestemd is.

De data-acquisitiekaart kan voor een bepaalde DMA-operatie meerdere buffers nodig hebben, bv. wanneer er veel data overgedragen moet worden of wanneer de buffers klein zijn. De software is zodanig opgezet dat de gebruiker in de initialisatiefase van een DMA-operatie een lijst met daarvoor beschikbare buffers moet opstellen, een zogenaamde Buffer Transfer List (BTL). Dit kan in termen van een hogere programmeertaal een array van korte integers zijn, waarin voor elke buffer de drie parameters achter elkaar staan : pagenummer, offset en lengte. De volgorde waarin de buffers gebruikt worden is dezelfde als de volgorde waarin de bufferparameters in de BTL staan. Steeds wanneer een buffer gebruikt is (volgeschreven met resultaten van A/D conversies of leeggelezen met waarden voor D/A conversies) schakelt de data-acquisitiekaart automatisch over naar de volgende DMA-buffer zonder daarbij samples te verliezen. Dit wordt gerealiseerd door twee kanalen van de DMA-controller te gebruiken. Wanneer het ene kanaal aktief is, kan het andere kanaal door de PC geherprogrammeerd worden met de parameters van de volgende buffer. De kaart geeft zelf aan de PC te kennen wanneer een DMA-buffer volgeschreven is door het genereren van een interrupt signaal naar de processor. De software handelt deze interrupt af en het programma van de gebruiker kan tijdens de data-acquisitie gewoon door draaien. Het wordt alleen af en toe zeer kort onderbroken door DMA data transfers en zo nu en dan een interrupt-afhandeling zonder dat het programma van de gebruiker hiervan grote hinder ondervindt. Om een indruk te geven : bij 130.000 A/D-conversies per sekonde wordt de PC ca. 10% langzamer dan normaal.

Hoofdstuk 5 : De software

De software voor het aansturen van de kaart bestaat uit een bibliotheek van enkele tientallen routines, die bedoeld zijn om vanuit een hogere programmeertaal (Fortan, Basic, Turbo-Pascal, etc.) aangeroepen te worden. Ze zijn geschreven in 80286 Assembly (een op de 80286 CPU toegespitste programmeertaal), waardoor ze zeer weinig processortijd verbruiken. Bij de opzet van de software is ervan uitgegaan dat er een of meerdere data-acquisitie kaarten uit de DT2821 familie in de PC voorkomen, die niet allemaal van hetzelfde type hoeven te zijn. Een tweede uitgangspunt is dat alle kaarten uitsluitend gebruik maken van de DMA kanalen 5 en 6 en niet van kanaal 7. Met behulp van jumper-schakelaars op de kaart kunnen de door de kaart te gebruiken DMA kanalen ingesteld worden. De software is zodanig geschreven dat het niet kan voorkomen dat twee kaarten of twee verschillende gedeelten van eenzelfde kaart (A/D en D/A sektie) tegelijkertijd van de DMA-controller kanalen 5 en 6 gebruik kunnen maken. De goede werking hiervan kan echter alleen dan worden gegarandeerd indien de gebruiker voor het communiceren met de kaart uitsluitend de routines uit de software-bibliotheek gebruikt en niet zelf rechtstreeks data gaat sturen naar de I/O registers van een data-acquisitiekaart.
De routines uit de software-bibliotheek zijn onderverdeeld in 7 categorieën:

- Initialisatie routine voor het opstarten van een kaart
- Digitale I/O routines
- Klok besturingsroutines
- A/D conversie routines
- D/A conversie routines
- DMA operatie besturingsroutines
- DMA-buffer management routines

We zullen ieder van de routines per categorie de naam en een korte funktie-omschrijving geven. Een uitgebreide funktie-omschrijving, manier van aanroepen en verduidelijking van de formele parameters voor alle routines wordt gegeven in appendix A, waarin de routines op alfabetische volgorde gerangschikt zijn.

Initialisatie routine :

BRD_INI : INItialize board. Moet altijd worden aangeroepen voordat een data-acquisitiekaart gebruikt gaat worden

Digitale I/O routines :

DIO_ERB : Enable for input and Read low or high Byte
DIO_ERW : Enable for input and Read Word
DIO_EWB : Enable for output and Write low or high Byte
DIO_EWW : Enable for output and Write Word
DIO_RDB : ReaD low or high Byte
DIO_RDW : ReaD Word
DIO_WRB : WRite low or high Byte
DIO_WRW : WRite Word
DIO_INQ : INQuire I/O directions of DIO ports
DIO_SDB : Set I/O Direction of DIO port low or high Byte
DIO_SDW : Set I/O Direction of DIO port Word

Clock besturingsroutines :

CLK_SET : SET pacer clock register
CLK_PER : set PERiod of pacer clock
CLK_INQ : INQuire about period of pacer clock
CLK_DEC : Disable External Clock, enable pacer clock
CLK_EEC : Enable External Clock, disable pacer clock
CLK_DSC : Disable PC AT System Clock
CLK_ESC : Enable PC AT System Clock

A/D conversie routines :

ADC_STC : Software Triggered single Conversion
ADC_ETC : Externally Triggered single Conversion
ADC_STS : Software Triggered Scan of channel gain list
ADC_ETS : Externally Triggered Scan of channel gain list
ADC_LCG : Load Channel Gain list
ADC_SAD : Series of A/D conversions with DMA data transfer

D/A conversie routines :

DAC_SSC : Software triggered Single Conversion
DAC_SDC : Software triggered Dual Conversions
DAC_ESC : Externally triggered Single Conversion
DAC_EDC : Externally triggered Dual Conversions
DAC_SDD : Series of D/A conversions with DMA data transfer

DMA operatie besturingsroutines :

DMA_ADT : A/D Trigger to start series of A/D conversions
DMA_DAT : D/A Trigger to start series of D/A conversions
DMA_HLT : HaLT any DMA operation in progress
DMA_WFC : Wait For Completion of DMA operation in progress
DMA_INQ : INQuire about status of DMA operation in progress

DMA buffer management routines :

MEM_DAB : Declare user program Array as Buffer
MEM_MBD : Move a Block of Data between buffer and program array

In appendix B worden enkele voorbeelden gegeven van Turbo-Pascal programma's die met de data-acquisitiekaart werken. Verder wordt daar uitgelegd hoe men de software-bibliotheek "aan de praat" kan krijgen voor het geval het geval er gewerkt wordt met Turbo-Pascal of met een programmeertaal-compiler, die

object (.OBJ) modules produceert, die vervolgens gelinkt moeten worden tot een executable (.EXE) file.
De software-bibliotheek neemt ongeveer 4 kilobyte RAM-geheugen in beslag.

Appendix A : Description of the DT2821 software library

Functional description and formal parameter specification of the procedures

Information about implementation and examples can be found in appendix B

ADC_ETC (Externally Triggered Conversion)

Function : perform an Externally Triggered single Conversion :
load the first channel-gain list entry of the DT2821 board with the specified channel & gain pair and perform an A/D conversion on this channel at the first clock pulse after the occurrence of the external trigger.

Warnings : -this procedure changes the contents of the channel-gain list.
-the execution of this routine will be skipped without further notice if it is called while a DMA operation is in progress on the selected board.

Call from Turbo Pascal : adc_etc(result,channel,gain,base);

Inputs :-integer channel contains the channel number (ranging from 0 to 15) that is to be sampled
-integer gain contains the gain that is applied to the conversion result. This is a number ranging from 0 to 3 with the following meaning : 0 : gain = 1
1 : gain = 2 (DT2825 : gain = 10)
2 : gain = 4 (DT2825 : gain = 100)
3 : gain = 8 (DT2825 : gain = 500)
-integer base contains the base address of the I/O buffer of a DT2821 board.

Output :-integer result contains the result of the A/D conversion in a coding selected by the jumper switches on the DT2821 board

ADC_ETS (Externally Triggered Scan)

Function : perform an Externally Triggered Scan. This is a series of A/D conversions beginning from the first entry in the channel-gain list to the final entry. Each conversion is started at the occurrence of a clock pulse. The first conversion starts at the occurrence of the first clock pulse (either from the on-board pacer clock or an external clock, see routines CLK_EEC and CLK_DEC) after the external trigger.
It is assumed that the Channel Gain List has previously been loaded by a call to routine ADC_LCG

Warning : The execution of this routine will be skipped without further notice if it is called while a DMA operation is in progress on any DT2821 board.

Call from Turbo Pascal : adc_ets(resultlist,error,base);

Input :-integer base contains the base address of the I/O buffer of a DT2821 board

Outputs :-integer array resultlist contains the results of the A/D conversions in the order as specified in the channel-gain list in a coding selected by the jumper switches on the DT2821 board
-integer error is equal to zero if no error has occurred and equal to 1 if an A/D error has occurred.

ADC_LCG (Load Channel-Gain list)

Function : Load the channel-gain list of the DT2821 board with the specified channel and gain numbers (input).

Warning : The execution of this routine will be skipped without further notice if the selected board is operating in A/D clocked or triggered scan DMA mode

Call from Turbo Pascal : adc_lcg(chanlist,gainlist,n,base);

Inputs :-integer array chanlist contains the channel numbers (max. 16 numbers ranging from 0 to 15) in the order in which the channels have to be sampled.

-integer array gainlist contains the gain numbers (max. 16 numbers ranging from 0 to 3) in the order in which the channels have to be sampled (as specified in chanlist). The gain numbers have the following meaning : 0 : gain = 1

1 : gain = 2 (DT2825 : gain = 10)

2 : gain = 4 (DT2825 : gain = 100)

3 : gain = 8 (DT2821 : gain = 500)

-integer n contains the number of channel-gain list entries ranging from 1 to 16.

-integer base contains the base address of the I/O buffer of a DT2821 board.

ADC_SAD (Series of A/D conv. with DMA transfers)

Function : perform a series of A/D conversions using DMA data tranfer from the selected DT2821 board to the RAM-buffers defined in a Buffer Transfer List (BTL). The transfer requires one or two DMA channels to do the transfer, depending on the number of A/D conversions to be done and on the magnitude of the buffers. If transfer requires two DMA channels an interrupt is issued by the selected DT2821 board each time a buffer has been filled and then the board immediately switches to the other DMA channel in order to fill the next DMA buffer.
The interrupt causes the execution of an interrupt service routine embedded within this routine. If necessary it sets up the parameters for the DMA channel that is currently not in use so that the board can switch to it after the DMA operation on the channel that is currently in use has terminated. The board(s) are jumper selected to use DMA channels 5 and 6 and interrupt levels 10 or 15.
The series of conversions starts at the first clock pulse after an external trigger or a software trigger. The first conversion corresponds with the first entry in the channel gain list. When the final entry in the channel gain list is reached selection loops around to the first entry again.
This routine only initializes the transfer. So upon exit the transfer is not completed, but is taking place throughout the execution of the rest of the user program. Use the routines DMA_INQ (INQuire), DMA_HLT (HaLT) or DMA_WFC (Wait For Completion) in order to monitor or influence the DMA operation in progress.

Warning : the execution of this routine will be skipped without further notice if it is called while a DMA operation is in progress on any DT2821 board.

Call from Turbo Pascal : adc_sad(nbuf,btlist,mode,trigger,base);

Inputs :-integer nbuf contains the number of buffers (specified in the transfer list) to be used. The first nbuf buffers are used. nbuf must not be greater than the total number of buffers in the list. nbuf must be greater than zero.

-integer array btlist contains the parameters of each transfer buffer. For buffer i these parameters are :

btlist [3*i-2] : page number of buffer i
btlist [3*i-1] : offset address from the page boundary in words
btlist [3*i]   : length of the buffer in words (1 word = 2 byte)

The 16 Mbyte memory space is divided in pages (units of 128 Kbyte) starting at addresses that are a multiple of 2**17. If the PC has 1 Mbyte of RAM located at the bottom of the address space the page number can range from 0 to 7. The page number determines bits 23 to 17 of the buffer starting address. The offset address determines bits 16 to 1 of the buffer starting address. Because buffers may not cross page boundaries the length of a buffer in words cannot be larger than 65536 - offset address.

The BTL must point to possibly large areas (buffers) of spare RAM Upon completion of the DMA operation the buffers themselves contain the A/D conversion results. Each value is an integer (2 byte) coded in a way that is selected by jumper switches on the DT2821 board.

The total number of A/D conversions carried out is equal to the sum of the lengths of the first nbuf buffers.

-integer mode selects the A/D mode to be used :

mode = 1 : A/D clocked DMA. The first A/D conversion is initiated by the first clock pulse after an external or software trigger. Then at each clock pulse another conversion is done.

The conversions and transfers transfers wil continue until the first nbuf buffers in the BTL are filled

mode = 2 : A/D triggered scan DMA. The first and all other conversions corresponding with the first entry in the channel gain list are initiated by the first clock pulse after an external or software trigger. All conversions corresponding with the second to the final channel gain list entries are initiated by subsequent clock pulses. So at each trigger the DT2821 board scans through the channel gain list once and then waits for the next trigger. This continues until the the first nbuf buffers in the BTL are filled. If the software trigger is selected the user program can issue this trigger by calling DMA_ADT (A/D Trigger).

mode = 3 : Continuous A/D clocked DMA. This is the same as mode = 1, except that when the first nbuf buffers are filled the transfer will be repeated over and over. This proces can only be stopped by calling DMA_HLT or BRD_INI.

mode = 4 : Continuous A/D triggered scan DMA. This is the same as mode = 2, except that the transfer will be repeated over and over. This process can only be stopped by calling DMA_HLT or BRD_INI.

-integer trigger selects wether the external or the software trigger is to be used to initiate the series of conversions :

trigger = 0 : software trigger. No trigger is generated within this routine. Use DMA_ADT to issue a software trigger

trigger <> 0 : external trigger

-integer base contains the base address of the I/O buffer of a DT2821 board

ADC_STC (Software Triggered Conversion)

Function : perform a Software Triggered single Conversion :
load the first channel-gain list entry of the DT2821 board with the specified channel & gain pair and perform an A/D conversion on this channel at the occurrence of the first clock pulse after the software trigger generated within this routine.

Warnings : -this procedure changes the contents of the channel-gain list.
-the execution of this routine will be skipped without further notice if it is called while a DMA operation is in progress on the selected board.

Call from Turbo Pascal : adc_stc(result,channel,gain,base);

Inputs :-integer channel contains the channel number (ranging from 0 to 15) that is to be sampled
-integer gain contains the gain that is applied to the conversion result. This is a number ranging from 0 to 3 with the following meaning : 0 : gain = 1
1 : gain = 2 (DT2825 : gain = 10)
2 : gain = 4 (DT2825 : gain = 100)
3 : gain = 8 (DT2825 : gain = 500)
-integer base contains the base address of the I/O buffer of a DT2821 board.

Output :-integer result contains the result of the A/D conversion in a coding selected by the jumper switches on the DT2821 board.

ADC_STS (Software Triggered Scan)

Function : perform a Software Triggered Scan. This is a series of A/D conversions beginning from the first entry in the channel-gain list to the final entry. Each conversion is started at the occurrence of a clock pulse. The first conversion starts at the occurrence of the first clock pulse (either from the on-board pacer clock or an external clock, see routines CLK_EEC and CLK_DEC) after the software trigger generated within this routine. It is assumed that the Channel Gain List has previously been loaded by a call to routine ADC_LCG

Warning : the execution of this routine will be skipped without further notice if it is called while a DMA operation is in progress on any DT2821 board.

Call from Turbo Pascal : adc_sts(resultlist,error,base);

Input :-integer base contains the base address of the I/O buffer of a DT2821 board

Outputs :-integer array resultlist contains the results of the A/D conversions in the order as specified in the channel-gain list and in a coding selected by the jumper switches on the DT2821 board.

-integer error is equal to zero if no error has occurred and equal to 1 if an A/D error has occurred.

BRD_INI (INItialize board)

Function : (re)initialize the DT2821 board :

- set the initialization bits
- clear all error bits and data buffers
- disable the external clock and external trigger
- initialize both DAC's with the specified values
- disable clock-initiated A/D or D/A conversions
- clear all interrupt bits, disabling board interrupts

This routine should always be called before calling any other routine involving the selected DT2821 board.

Call from Turbo Pascal : brd_ini(dac0,dac1,base);

Inputs :-integers dac0 and dac1 contain the initial values for DAC0 and DAC1 respectively. The values are expected in offset binary and will be send to the DAC's without any modification

-integer base contains the base address of the I/O buffer of a DT2821 board

CLK_DEC (Disable External Clock)

Function : use the on-board pacer clock to initiate A/D and/or D/A conversions instead of the external clock

Call from Turbo Pascal : clk_dec(base);

Input :-integer base contains the base address of the I/O buffer of a DT2821 board

CLK_DSC (Disable System Clock)

Function : disable the PC AT system clock by ignoring IRQ0 (Timer interrupt), IRQ8 (Realtime Clock Interrupt) and IRQ3 to IRQ7 (Serial & Parallel Ports) except IRQ6 (Diskette Controller)

Call from Turbo Pascal : clk_dsc;

CLK_EEC (Enable External Clock)

Function : use the external clock to initiate A/D and/or D/A conversions
instead of the on-board pacer clock

Call from Turbo Pascal : clk_eec(base);

Input :-integer base contains the base address of the I/O buffer of a
DT2821 board

CLK_ESC (Enable System Clock)

Function : enable the PC AT system clock by acknowledging IRQ0 and all other interrupts (IRQ1 to IRQ15).

Call from Turbo Pascal : clk_esc;

CLK_INQ (INQuire clock period)

Function : output a 4 byte integer (called p from now on) that contains the period of time between two pulses of the current DT2821 board pacer clock in units of 250 ns.

Call from Turbo Pascal : clk_inq(phigh,plow,base);

Input :-integer base contains the address of the I/O buffer of a DT2821 board

Outputs :-integer phigh contains most significant word of p
-integer plow contains least significant word of p

CLK_PER (set clock PERiod)

Function : find the prescaler and counter value of the pacer clock register so that the period of time between two clock pulses matches the desired period (input) as closely as possible and then send these values to the pacer clock register.

Call from Turbo Pascal : clk_per(phigh,plow,base);

Inputs :-integers phigh and plow (2 byte each) specifying the desired period p (a 4 byte integer) in units of 250 ns.
phigh contains most significant word of p
plow contains least significant word of p
-integer base contains the base address of the I/O buffer of a DT2821 board

CLK_SET (SET clock register)

Function : Set the prescaler and counter bits of the Pacer Clock register of the current DT2821 board to the specified values (input)

Call from Turbo Pascal : clk_set(prescaler,counter,base);

Inputs :-integer prescaler contains the prescaler value, ranging from 0 to 15.

-integer counter contains the counter value ranging from 0 to 255. The pacer clock period can be calculated as follows :

clock period = 250 * (2 ** prescaler) * (256 - counter) nanosec.

if prescaler = 1 then substitute 0 for prescaler in this formula in order to correct the clock discontinuity for prescaler = 1

-integer base contains the base address of the I/O buffer of a DT2821 board

DAC_EDC (Externally triggered Dual Conversion)

Function : perform an Externally Triggered dual D/A Conversion on the two DAC channels. Because there are two conversions to be done the D/A subsystem is put in dual-channel mode. The actual D/A conversions take place at the first clock pulse after the occurrence of an external trigger.

Warning : the execution of this routine will be skipped without further notice if it is called while a DMA operation is in progress on the selected board

Call from Turbo Pascal : dac_edc(valuex,valuey,base);

Inputs :-integer valuex contains the value to be converted to analog by DAC0. The value sent to the DAC (expected in offset binary) is valuex + 2048, thus converting from 2'complement to 12 bit offset binary for bipolar DAC operation. The most significant 4 bits of valuex are ignored in case the DT2821 board has 12 bit DAC's. For 16 bit DAC's an extra offset has to be added by the user on top of the 2048 offset added by this routine

-integer valuey contains the value to be converted to analog by DAC1 The same comment as for valuex is applicable here

-integer base contains the base address of the I/O buffer of a DT2821 board

DAC_ESC (Externally triggered Single Conversion)

Function : perform an Externally Triggered single D/A Conversion on one of the two DAC channels. Because there is only one conversion to be done the D/A subsystem is put in single-channel mode. The actual D/A conversion takes place at the first clock pulse after the occurrence of an external trigger

Warning : the execution of this routine will be skipped without further notice if it is called while a DMA operation is in progress on the selected board.

Call from Turbo Pascal : dac_esc(value,xory,base);

Inputs :-integer value contains the value to be converted to analog by the DAC. The value sent to the DAC (expected in offset binary) is value + 2048, thus converting from 2'complement to 12 bit offset binary for bipolar DAC operation. The most significant 4 bits of value are ignored in case the DT2821 board has 12 bit DAC's. For 16 bit DAC's an extra offset has to be added by the user on top of the 2048 offset added by this routine

-integer xory selects the x- or y-channel :
xory = 0 : x-channel ; xory <> 0 : y-channel

-integer base contains the base address of the I/O buffer of a DT2821 board

DAC_SDC (Software triggered Dual Conversion)

Function : perform a software triggered dual D/A conversion on the two DAC channels. Because there are two conversions to be done the D/A subsystem is put in dual-channel mode. The software trigger that initiates the D/A conversion is generated within this routine

Warning : the execution of this routine will be skipped without further notice if it is called while the selected board is operating in D/A clocked DMA mode.

Call from Turbo Pascal : dac_sdc(valuex,valuey,base);

Inputs :-integer valuex contains the value to be converted to analog by DAC0. The value sent to the DAC (expected in offset binary) is valuex + 2048, thus converting from 2'complement to 12 bit offset binary for bipolar DAC operation. The most significant 4 bits of valuex are ignored in case the DT2821 board has 12 bit DAC's. For 16 bit DAC's an extra offset has to be added by the user on top of the 2048 offset added by this routine

-integer valuey contains the value to be converted to analog by DAC1 The same comment as for valuex is applicable here

-integer base contains the base address of the I/O buffer of a DT2821 board

DAC_SDD (Series of D/A conv. with DMA data transfer)

Function : perform a series of D/A conversions using DMA data tranfer from the RAM-buffers defined in a Buffer Transfer List (BTL) to the selected DT2821 board. The transfer requires one or two DMA channels to do the transfer, depending on the number of D/A conversions to be done and on the magnitude of the buffers. If transfer requires two DMA channels an interrupt is issued by the selected DT2821 board each time a buffer has been filled and then the board immediately switches to the other DMA channel in order to fill the next DMA buffer.

The interrupt causes the execution of an interrupt service routine embedded within this routine. If necessary it sets up the parameters for the DMA channel that is currently not in use so that the board can switch to it after the DMA operation on the channel that is currently in use has terminated. The board(s) are jumper selected to use DMA channels 5 and 6 and interrupt levels 10 or 15 .

The series of conversions starts at the occurrence of the first clock pulse after the detection of an external trigger or a software trigger.

This routine only initializes the transfer. So upon exit the transfer is not completed, but is taking place throughout the execution of the rest of the user program. Use the routines DMA_INQ (INQuire), DMA_HLT (HaLT) or DMA_WFC (Wait For Completion) in order to monitor or influence the DMA operation in progress.

Warning : the execution of this routine will be skipped without further notice if it is called while a DMA operation is in progress on any DT2821 board.

Call from Turbo Pascal : dac_sdd(nbuf,btlist,mode,trigger,base);

Inputs :-integer nbuf contains the number of buffers (specified in the transfer list) to be used. The first nbuf buffers are used. nbuf must not be greater than the total number of buffers in the list. nbuf must be greater than zero.

-integer array btlist contains the parameters of each transfer buffer. For buffer i these parameters are :

btlist [3*i-2] : page number of buffer i
btlist [3*i-1] : offset address from the page boundary in words
btlist [3*i]   : length of the buffer in words (1 word = 2 byte)

The 16 Mbyte memory space is divided in pages (units of 128 Kbyte) starting at addresses that are a multiple of 2**17. If the PC has 1 Mbyte of RAM located at the bottom of the address space the page number can range from 0 to 7. The page number determines bits 23 to 17 of the buffer starting address. The offset address determines bits 16 to 1 of the buffer starting address. Because buffers may not cross page boundaries the length of a buffer in words cannot be larger than 65536 - offset address.

The BTL must point to possibly large areas (buffers) of spare RAM The buffers themselves must contain the values to be converted to analog. Each value is an integer (2 byte) ranging from 0 to 4096 coded in offset binary (= 2's complement + c , where c = 2048 for bipolar 12 bit DAC's, 0 for unipolar 12 or 16 bit DAC's and 32768 for 16 bit bipolar DAC's). The conversion from 2's complement to offset binary must be done by the user.

If the board is operating in dual channel mode the first value of each pair of values is sent to channel X (DAC0), the second value to channel Y (DAC1). The total number of D/A conversions carried out is equal to the sum of the lengths of the first nbuf buffers.

-integer mode selects the D/A mode to be used :

mode = 1 : dual D/A mode, simultaneous D/A conversions on both channels X and Y. DMA data transfer stops when the first nbuf buffers in the BTL are used

mode = 2 : single D/A mode using channel X. DMA data transfer stops when the first nbuf buffers in the BTL are used

mode = 3 : single D/A mode using channel Y. DMA data transfer stops when the first nbuf buffers in the BTL are used

mode = 4 : dual D/A mode, simultaneous conversions on both channels X and Y. The first nbuf buffers in the BTL are continuously scanned until DMA_HLT is called.

mode = 5 : single D/A mode using channel X. The first nbuf buffers in the BTL are continuously scanned until DMA_HLT is called.

mode = 6 : single D/A mode using channel Y. The first nbuf buffers in the BTL are continuously scanned until DMA_HLT is called.

-integer trigger selects wether the external or the software trigger is to be used to initiate the series of conversions :

trigger = 0 : software trigger

trigger <> 0 : external trigger

No trigger is generated within this routine. If the software trigger is selected, the user can issue this trigger by calling routine DMA_DAT (D/A Trigger).

-integer base contains the base address of the I/O buffer of a DT2821 board

DAC_SSC (Software triggered Single Conversion)

Function : perform a software triggered single D/A conversion on one of the two DAC channels. Because there is only one conversion to be done the D/A subsystem is put in single-channel mode. The software trigger that initiates the D/A conversion is generated within this routine

Warning : the execution of this routine will be skipped without further notice if it is called while the selected board is operating in D/A clocked DMA mode.

Call from Turbo Pascal : dac_ssc(value,xory,base);

Inputs :-integer value contains the value to be converted to analog by the DAC. The value sent to the DAC (expected in offset binary) is value + 2048, thus converting from 2'complement to 12 bit offset binary for bipolar DAC operation. The most significant 4 bits of value are ignored in case the DT2821 board has 12 bit DAC's. For 16 bit DAC's an extra offset has to be added by the user on top of the 2048 offset added by this routine

-integer xory selects the x- or y-channel :
xory = 0 : x-channel ; xory <> 0 : y-channel

-integer base contains the base address of the I/O buffer of a DT2821 board

DIO_ERB (Enable & Read Byte)

Function : enable the high byte or the low byte of the DIO register for input and read a byte from the enabled byte of the DIO register.

Call from Turbo Pascal : dio_erb(inbyte,i,base);

Output : integer inbyte contains the byte read from the DIO register
Inputs : integer i selects the high or low byte of DIO register :
i = 0 : put the contents of the DIO low byte in inbyte
i <> 0 : put the contents of the DIO high byte in inbyte
integer base containing the base address of the I/O buffer of a DT2821 board

DIO_ERW (Enable & Read Word)

Function : enable the high and low byte of the DIO register for input and read a word from the DIO register

Call from Turbo Pascal : dio_erw(inword,base);

Output : integer inword contains the word read from the DIO register
Input : integer base contains the base address of the I/O buffer of a DT2821 board

DIO_EWB (Enable & Write Byte)

Function : enable the high byte or the low byte of the DIO register for output and write a byte to the enabled byte of the DIO register

Call from Turbo Pascal : dio_ewb(outbyte,i,base);

Inputs : byte outbyte contains the byte to be sent to the DIO register ranging from 0 to 255.
integer i selects the high or the low byte of the DIO register :
i = 0 : send the contents of outbyte to the DIO low byte
i <> 0 : send the contents of outbyte to the DIO high byte
integer base containing the base address of the I/O buffer of a DT2821 board

DIO_EWW (Enable & Write Word)

Function : enable the high and low byte of the DIO register for output and write a word to the DIO register

Call from Turbo Pascal : dio_eww(outword,base);

Inputs : integer outword contains the word to be written to the DIO register
integer base contains the base address of the I/O buffer of a DT2821 board

DIO_INQ (INQuire)

Function : read the current direction of the data flow on the DIO register high and low bytes

Call from Turbo Pascal : dio_inq(diodir,base)

Input : integer base contains the base address of the I/O buffer of a DT2821 board

Output : integer diodir contains a number ranging from 0 to 3 :

diodir = 0 : high byte input , low byte input
diodir = 1 : high byte input , low byte output
diodir = 2 : high byte output, low byte input
diodir = 3 : high byte output, low byte output

DIO_RDB (ReaD Byte)

Function : read a byte from the high or low byte of the DIO register.
It is assumed that the specified byte (high or low) is enabled for input

Call from Turbo Pascal : dio_rdb(inbyte,i,base);

Output : integer inbyte contains the byte read from the DIO register

Inputs : integer i selects the high or low byte of the DIO register
i = 0 : put the contents of the DIO low byte in inbyte
i <> 0 : put the contents of the DIO high byte in inbyte
integer base containing the base address of the I/O buffer of a DT2821 board

DIO_RDW (ReaD Word)

Function : read a word from the DIO register. It is assumed that both the high and the low byte of the DIO register are enabled for input

Call from Turbo Pascal : dio_rdw(inword,base);

Output : integer inword contains the word read from the DIO register

Input : integer base contains the base address of the I/O buffer of a DT2821 board

DIO_SDB (Set Direction Byte)

Function : set the direction of the data flow on the DIO register high byte or the low byte (input or output)

Call from Turbo Pascal : dio_sdb(i,base);

Inputs : integer i containing a value ranging from 0 to 3 :

i = 0 : make DIO low byte an input port
i = 1 : make DIO low byte an output port
i = 2 : make DIO high byte an input port
i = 3 : make DIO high byte an output port

integer base pointing to the base address of the I/O buffer of a DT2821 board

DIO_SDW (Set Direction Word)

Function : set the data flow direction of the DIO register high and low byte (input or output)

Call from Turbo Pascal : dio_sdw(i,base);

Inputs : integer i ranging from 0 to 3 :

i = 0 : high byte input, low byte input

i = 1 : high byte input, low byte output

i = 2 : high byte output, low byte input

i = 3 : high byte output, low byte output

integer base contains the base address of the I/O buffer of a DT2821 board

DIO_WRB (WRite Byte)

Function : write a byte to the specified part (high or low byte) of the
DIO register.
It is assumed that the specified byte is enabled for output.

Call from Turbo Pascal : dio_wrb(outbyte,i,base);

Inputs : integer outbyte contains the byte to be sent to the DIO register
ranging from 0 to 255
integer i selects the high or low byte of the DIO register :
i = 0 : send the contents of outbyte to the DIO low byte
i <> 0 : send the contents of outbyte to the DIO high byte
integer base containing the base address of the I/O buffer of a
DT2821 board

DIO_WRW (WRite Word)

Function : write a word to the DIO register. It is assumed that both the high and the low byte of the DIO register are enabled for output

Call from Turbo Pascal : dio_wrw(outword,base);

Inputs : integer outword contains the word to be written to the DIO register

integer base contains the base address of the I/O buffer of a DT2821 board

DMA_ADT (A/D Trigger)

Function : If a board is operating in A/D triggered scan DMA mode or in A/D clocked DMA mode, this routine issues a soft start trigger that initiates a series of A/D conversions.

Warning : if this routine is called in A/D triggered scan DMA mode while a scan of the channel gain list is not yet completed, no trigger will be given in order to prevent an A/D trigger error. If this routine is called in A/D clocked DMA mode only the first trigger call will lead to a trigger.

Call from Turbo Pascal : dma_adt(base);

Output : integer base contains the base address of the I/O buffer of the DT2821 board that is operating in A/D DMA mode. If no DMA operation is in progress on any board, base equals 0.

DMA_DAT (D/A Trigger)

Function : If the board is operating in D/A clocked DMA mode, this routine issues a soft start trigger that initiates a series of D/A conversions. The trigger will only be given at the first call to DMA_DAT after the initialization of the DMA operation by DAC_SDD. Subsequent calls to DMA_DAT during the DMA operation will be ignored.

Call from Turbo Pascal : dma_dat(base);

Output : integer base contains the base address of the I/O buffer of the DT2821 board that is operating in A/D DMA mode. If no DMA operation is in progress on any board, base equals 0.

DMA_INQ (INQuire)

Function : inquire about the DMA operation that is currently in progress. This routine acquires information about which DMA mode is selected, wether of not the selected board is operating in dual DMA mode, which DMA channel is currently in use, which buffer is currently in use and what is the DMA controller's current word count and the base address of the DT2821 board.

```
Call from Turbo Pascal : dma_inq(error,dualdma,dmamode,bufferb,nbuf,bfcount,
                                 curwcount,base);
```

Outputs: -integer error reports the status of the A/D and D/A error bits :

error = 0 : no A/D or D/A error
error = 1 : A/D error has occurred
error = 2 : D/A error has occurred
error = 3 : A/D and D/A errors have occurred

-integer dualdma (bit 12 of SUPCSR register) indicates wether the board is operating in dual channel mode (dualdma = 1) or not (dualdma = 0)

-integer dmamode (bits 11 and 10 of SUPCSR register) indicates the current DMA mode of the board :

dmamode = 0 : No DMA
dmamode = 1 : A/D clocked DMA
dmamode = 2 : D/A clocked DMA
dmamode = 3 : A/D triggered scan DMA

-integer bufferb (bit 9 of SUPCSR register) indicates which of the two DMA channels is currently in use (only valid if dmamode > 0):

bufferb = 0 : DMA channel 5
bufferb = 1 : DMA channel 6

-integer nbuf equals 0 if non-continuous DMA was selected at the initialization done by ADC_SAD or DAC_SDD. If continuous DMA was selected nbuf equals the total number of buffers used for this DMA operation

-integer bfcount gives the number of buffers in the buffer transfer list that still have to be used. If nbuf (as specified in the call to "dac_sdd" or to "adc_sad") is greater than 1 bufid ranges from nbuf-1 to 0. If nbuf is equal to 1 bufid is zero.

-integer curwcount gives the current word count of the DMA channel that is currently in use. This is the number of data words that still have to be transferred before operation on this channel is completed

-integer base contains the base address of the I/O buffer of the DT2821 board that is operating in A/D DMA mode. If no DMA operation is in progress on any board, base equals 0.

DMA_HLT (HaLT)

Function : halt any DMA operation that is currently in progress on channels 5 and 6. If the A/D system is or has been using the DMA mode, the A/D subsystem is reset. If the D/A subsystem is or has been using the DMA mode, the D/A subsystem is reset

Call from Turbo Pascal : dma_hlt(base);

Output : integer base contains the base address of the I/O buffer of the DT2821 board on which a DMA operation was in progress. If there was no DMA operation in progress on any board, base equals 0.

DMA_WFC (Wait For Completion)

Function : wait for the completion of the DMA operation that is currently in progress on the selected board if the board is operating in either non-continuous A/D or non-continuous D/A clocked DMA mode. If it is not , calling this routine has no effect. A DMA operation is completed if all the buffers in the BTL are used or if an error bit is set.

Warning: If the software trigger is selected in the call to ADC_SAD or to DAC_SDD, calling this routine before the trigger is given by calling DMA_ADT or DMA_DAT will have no effect. This prevents the program from getting into an endless loop.

Call from Turbo Pascal : dma_wfc(error,base);

Outputs :-integer error contains information about the A/D or D/A error bits at the termination of the DMA operation :

error = 0 : no error
error = 1 : A/D error bit was set
error = 2 : D/A error bit was set

-integer base contains the base address of the I/O buffer of the DT2821 board that is operating in A/D DMA mode. If no DMA operation is in progress on any board, base equals 0.

MEM_DAB (Declare Array as Buffer)

Function : Declare a user program Array as Buffer. This routine makes an entry in a Buffer Transfer List (BTL) point to a user program array. This has the advantage that the data in the buffer is directly accessible to the user without having to make a call to routine MEM_MBD that moves blocks of data between a buffer and an array. A disadvantage could be that the buffers cannot be larger than the maximum allowable array size and that they cannot be located in extended memory.

Warning : This routine can modify a BTL while a DT2821 board is performing a DMA operation using the same BTL. Be sure to know what you are doing if you call this routine while a DMA operation is in progress.

Call from Turbo Pascal : mem_mbd(btlist,bufid,progarray,nwords,offset);

Output :-integer array btlist contains the parameters of each transfer buffer. For buffer bufid these parameters are :

btlist [3*bufid-2] : page number of buffer
btlist [3*bufid-1] : offset address from the page boundary in words
btlist [3*bufid] : length of the buffer in words

The 16 Mbyte memory space is divided in pages (units of 128 Kbyte) starting at addresses that are a multiple of 2**17. If the PC has 0.5 Mbyte of RAM located at the bottom of the address space the page number can range from 0 to 4. The page number determines bits 23 to 17 of the buffer starting address. The offset address determines bits 16 to 1 of the buffer starting address. Because buffers may not cross page boundaries the length of a buffer in words cannot be larger than 65536 - offset address.

The BTL must point to possibly large areas (buffers) of spare RAM. Only one set of buffer parameters is modifies by this routine.

Inputs :-integer bufid contains the number of the buffer whose parameters in the BTL are set so that the buffer and the user program array occupy the same memory area.

-integer array progarray is the array that has to be declared as a buffer. It does not have to contain any valid data.

-integer nwords specifies the number of words within progarray that the buffer must occupy. nwords may not be larger than the total number of words in progarray

-integer offset specifies the offset in words from the beginning of progarray to the start of the buffer. So, if progarray is a one dimensional array declared as progarray[6..1000] the buffer occupies progarray[6+offset] to progarray[6+offset+nwords]. The sum of offset and nwords may not be larger than the total array length.

Warning : buffers may not cross page boundaries, but the selected part of progarray may well do this. In this case the routine will truncate the buffer length to the end of the current page. Check btlist[3*bufid] (the real buffer length) to see if it is equal to nwords (the desired buffer length) upon exit of the routine

MEM_MBD (Move Block of Data)

Function : move a block of data between a data buffer (possibly in extended memory) and a user accessible program array

Warning : This routine can access a buffer while the DT2821 board is performing a DMA operation on the same buffer. Be sure to know what you are doing if you call this routine while a DMA operation is in progress.

Call from Turbo Pascal : mem_mbd(btlist,bufid,offset,nwords,progarray,
direction);

Inputs :-integer array btlist contains the parameters of each transfer buffer. For buffer bufid these parameters are :

btlist [3*bufid-2] : page number of buffer
btlist [3*bufid-1] : offset address from the page boundary in words
btlist [3*bufid] : length of the buffer in words

The 16 Mbyte memory space is divided in pages (units of 128 Kbyte) starting at addresses that are a multiple of 2**17. If the PC has 0.5 Mbyte of RAM located at the bottom of the address space the page number can range from 0 to 4. The page number determines bits 23 to 17 of the buffer starting address. The offset address determines bits 16 to 1 of the buffer starting address. Because buffers may not cross page boundaries the length of a buffer in words cannot be larger than 65536 - offset address.

The BTL must point to possibly large areas (buffers) of spare RAM

-integer bufid contains the number of the buffer to be used for the data transfer.

-integer offset contains the offset address in words from the bottom of the buffer that points to the bottom of the block of data to be moved.

-integer nwords contains the size in words of the block of data to be moved. This size cannot be larger than the buffer length (specified in btlist[3*bufid]) minus the offset from the bottom of the buffer.

-integer direction selects the data transfer direction :

direction = 0 : from buffer to user program array

direction <> 0 : form user program array to buffer.

Input/Output (depending on the value of direction) :

-integer array progarray is used for the data transfer. This array is user accessible in a Turbo Pascal program

Appendix B : Information about implementation and examples

In order to use the dt2821 library routines in a Turbo Pascal program the files "dt2821.inc" and "dt2821.com" are required. Include the file "dt2821.inc" in the program (so the first statement in the program should be "{$I dt2821.inc}"). "dt2821.inc" declares all the routines in the dt2821 library and loads "dt2821.com" as an external file, that contains the code of all dt2821-routines. It is assumed that these files are on the same directory as the program source file.
In case the dt2821 library is to be used in a Fortran program or with any other language using a compiler that produces object (.OBJ) modules the modules can be linked with the library "dt2821.lib" into an executable (.EXE) file using the program LINK in Microsoft's Macro Assembler package.
There are two versions of the dt2821 library. The Turbo Pascal version (dt2821.com) consists of near procedures and uses value parameters. So the value of all input parameters is passed on the stack instead of a pointer to the parameter. The Fortran version (dt2821.lib) consists of far procedures and does not use value parameters. All parameters are passed on the stack by means of pointers.

Example demo ad.pas : perform a series of A/D conversions

```
{
 This program gives an example of the usage of A/D Clocked or triggered Scan
 DMA mode in order to perform a series of A/D conversions.

 The first thing to do is include the file "dt2821.inc" that declares all the
 procedures in the DT2821 library and loads the external file "dt2821.com",
 that contains the code for all routines
}
{$I dt2821.inc}
type list   = array [1..24]   of integer;
     buffer = array [1..1000] of integer;
var base,i,j,k,l,m,n,error,clkhigh,clklow,
    dualdma,dmamode,bufferb,nbuf,bfcount,curwcount : integer;
    choice : string[1];
    chanlist,gainlist,btlist : list;
    buf1,buf2,buf3 : buffer;
begin
{
 variable "base" must contain a valid base address of the I/O buffer of a
 DT2821 board
}
base := $0240;      { after a $ follows a hexadecimally interpreted number}
{
 Initialize the board. Set the initial output value of both DAC's to 0 Volt,
 assuming that they are jumper configured for bipolar output and that we are
 dealing with a board containing 12 bit DAC's. These initial values must be
 given in offset binary coding (= 2's complement + $0800 for 12 bit DAC's),
 not in 2's complement
}
brd_ini ($0800,$0800,base);
{
 Read the desired clock period and set the clock period of the pacer clock on
 the board to a period as closely as possible matching the desired period.
 Then inquire about the exact clock period (not necessarily equal to the
 desired period)
 The period is specified by two short integers in the following way :

    clock period = [ clkhigh * 65536 + clklow ] * 250 nanoseconds

 in this formula 2's complement conventions are ignored : the short integers
 can range from 0 to 65535 (always positive)
}
write('Clockperiod high and low word : ');readln(clkhigh,clklow);
clk_per (clkhigh,clklow,base);
clk_inq (clkhigh,clklow,base);
writeln ('Actual clock period : high word ',clkhigh,'  low word ',clklow);
{
 Set up the contents of the Channel Gain List. Here we choose to work with 12
 entries (maximum is 16). For each entry the channel number can range from
 0 to 15 and the gain number from 0 to 3. Here we select the channels 0 to 11
 in downward order. The associated gains are all the same : gain number 0
 (so gain factor = 1)
 Call the routine that loads the Channel Gain List
}
for i := 1 to 12 do
  begin
     chanlist[i] := 12 - i;
     gainlist[i] := 0;
  end;
adc_lcg (chanlist,gainlist,12,base);
{
 Set up a Buffer Transfer List with parameters of 3 DMA buffers :
 declare 3 arrays as buffers and put each buffer's parameters (page number,
 offset from page boundary and buffer length in words) in an array "btlist."
 that will function as the Buffer Transfer List (BTL).
 Clean the buffers
}
mem_dab (btlist,1,buf1,1000,0);
mem_dab (btlist,2,buf2,1000,0);
mem_dab (btlist,3,buf3,1000,0);
for i := 1 to 1000 do
  begin
      buf1[i] := 0; buf2[i] := 0; buf3[i] := 0
  end;
{
 Report the real buffer lengths (not necessarily equal to the desired buffer
 lengths in case any of the three arrays crosses a page boundary)
}
writeln ('Real buffer lengths : ',btlist[3],'  ',btlist[6],'  ',btlist[9]);
{
 Read the number of buffers and the DMA mode to be used in routine adc_sad
 (initialization of the DMA operation) and then call adc_sad. Read the
 description of adc_sad before choosing parameters. Here we have selected
 the software trigger to initiate the series of conversions. In case
 non-continous DMA is selected the DMA operation continues until the selected
 number of buffers are filled with A/D-conversion results
}
write ('Number of buffers to be used (max. 3) : '); readln (n);
if n > 3 then n := 3; if n < 1 then n := 1;
write ('DMA mode to be used (1 to 4) : '); readln (m);
if m > 4 then m := 4; if m < 1 then m := 1;
adc_sad (n,btlist,m,0,base);
{
 The DMA operation is now initialized and will start at the first trigger
 to be given by a call to routine dma_adt or by repetitive calls to dma_adt
 in case triggered scan DMA is selected. The program will keep running while
 A/D conversions are taking place. The following menu-oriented loop allows
 you to give triggers, inquire about the DMA operation, halt it, wait for its
 completion or to watch parts of the buffers filling up with A/D conversion
 results. It is also possible to carry out D/A conversions while the A/D DMA
 operation is going on. Connect the A/D input channels 0 to 11 with tunable
 voltage sources in order to check the board's operation
}
choice :=  ' ';
while choice <> 'q' do
  begin
    writeln ('t = trigger ; q = quit; d = display buffer ; i = inquire DMA');
    writeln ('h = halt DMA; w = wait for DMA completion  ; c = D/A conv.');
    write ('Choice : '); readln (choice);
    if choice = 't' then
      begin
        dma_adt (k); writeln ('DMA_ADT base : ',k)
      end;
    if choice = 'i' then
      begin
        dma_inq (error,dualdma,dmamode,bufferb,nbuf,bfcount,curwcount,k);
        writeln ('error        : ',error);
        writeln ('dualdma      : ',dualdma);
        writeln ('DMA mode     : ',dmamode);
        writeln ('Buffer B     : ',bufferb);
        writeln ('nbuf         : ',nbuf);
        writeln ('buffer count : ',bfcount);
        writeln ('word count   : ',curwcount);
        writeln ('base address : ',k);
      end;
    if choice = 'd' then
      begin
        write ('Give first array-index : '); readln (i);
        if 1 > 980 then i := 980; if i < 1 then i := 1;
        for k := i to i + 20 do writeln (buf1[k],'         ',buf2[k],
                                         '         ',buf3[k])
      end;
    if choice = 'c' then
      begin
        write ('Give D/A values : '); readln (k,l);
        dac_sdc (k,l,base)
      end;
    if choice = 'h' then
      begin
        dma_hlt (error,k); writeln ('DMA_HLT error : ',error,'  base : ',k)
      end;
    if choice = 'w' then
      begin
        dma_wfc (error,k); writeln ('DMA_WFC error : ',error,'  base : ',k)
      end;
  end;
{
 Halt any DMA operation in progress and reset the board before leaving the
 program
}
dma_hlt (error,k);
brd_ini ($0800,$0800,base);
end.
```

Example demo da.pas : perform a series of D/A conversions

```
{
 This program gives an example of the usage of D/A Clocked DMA mode in order
to
 perform a series of D/A conversions.

 The first thing to do is include the file "dt2821.inc" that declares all the
 procedures in the DT2821 library and loads the external file "dt2821.com",
 that contains the code for all routines
}
{$I dt2821.inc}
type list   = array [1..24]   of integer;
     buffer = array [1..1000] of integer;
var base,i,j,k,l,m,n,error,clkhigh,clklow,
    dualdma,dmamode,bufferb,nbuf,bfcount,curwcount : integer;
    choice : string[1];
    btlist : list;
    buf1,buf2,buf3 : buffer;
begin
{
 variable "base" must contain a valid base address of the I/O buffer of a
 DT2821 board
}
base := $0240;      { after a $ follows a hexadecimally interpreted number}
{
 Initialize the board. Set the initial output value of both DAC's to 0 Volt,
 assuming that they are jumper configured for bipolar output and that we are
 dealing with a board containing 12 bit DAC's. These initial values must be
 given in offset binary coding (= 2's complement + $0800 for 12 bit DAC's),
 not in 2's complement
}
brd_ini ($0800,$0800,base);
```

```
{
 Read the desired clock period and set the clock period of the pacer clock on
 the board to a period as closely as possible matching the desired period.
 Then inquire about the exact clock period (not necessarily equal to the
 desired period)
 The period is specified by two short integers in the following way :

    clock period = [ clkhigh * 65536 + clklow ] * 250 nanoseconds

 in this formula 2's complement conventions are ignored : the short integers
 can range from 0 to 65535 (always positive)
}
write('Clockperiod high and low word : ');readln(clkhigh,clklow);
clk_per (clkhigh,clklow,base);
clk_inq (clkhigh,clklow,base);
writeln ('Actual clock period : high word ',clkhigh,'  low word ',clklow);
{
 Set up a Buffer Transfer List with parameters of 3 DMA buffers :
 declare 3 arrays as buffers and put each buffer's parameters (page number,
 offset from page boundary and buffer length in words) in an array "btlist."
 that will function as the Buffer Transfer List (BTL).
}
mem_dab (btlist,1,buf1,1000,0);
mem_dab (btlist,2,buf2,1000,0);
mem_dab (btlist,3,buf3,1000,0);
{
 Report the real buffer lengths (not necessarily equal to the desired buffer
 lengths in case any of the three arrays crosses a page boundary)
}
writeln ('Real buffer lengths : ',btlist[3],'  ',btlist[6],'  ',btlist[9]);
{
 Fill the buffers with some data to be output by the D/A converters (it is
 assumed that they are jumper configured for bipolar output with a range
 from -10 to +10 Volts) :

 buf1 : a square wave from 0 Volt to 5 Volt
 buf2 : a ramp from -10 Volt to +10 Volt
 buf3 : a sine with amplitude 10 Volt

 All D/A data has to be supplied in offset binary, not in 2's complement
 (offset binary = 2's complement + 2048 for 12 bit bipolar DAC's)
}
for i := 1 to 1000 do
  begin
     if i < 500 then buf1[i] := 2048 else buf1[i] := 3072;
     buf2[i] := round ( 4.096 * (i-1) );
     buf3[i] := 2048 + round ( 2047 * sin ( 0.006283 * (i-1) ) )
  end;
{
 Read the number of buffers and the DMA mode to be used in routine dac_sdd
 (initialization of the DMA operation) and then call dac_sdd. Read the
 description of dac_sdd before choosing the parameters. Here we have
 selected the software trigger to initiate the series of conversions. In case
 non-continous DMA is selected the DMA operation continues until the selected
 number of buffers are read. In case Dual DMA is selected the first data word
 of each pair of words is sent to DAC0 and the second to DAC1
}
write ('Number of buffers to be used (max. 3) : '); readln (n);
if n > 3 then n := 3; if n < 1 then n := 1;
write ('DMA mode to be used (1 to 6) : '); readln (m);
if m > 6 then m := 6; if m < 1 then m := 1;
dac_sdd (n,btlist,m,0,base);
{
 The DMA operation is now initialized and will start at the first trigger
 to be given by a call to routine dma_dat. The program will keep running while
 D/A conversions are taking place. The following menu-oriented loop allows
 you to give a trigger, inquire about the DMA operation, halt it or wait for
 its completion. Monitor the D/A output channels with an oscilloscope in order
 to check the board's operation
}
choice :=  ' ';
while choice <> 'q' do
  begin
    writeln ('t = trigger ; q = quit; d = display buffer ; i = inquire DMA');
    writeln ('h = halt DMA; w = wait for DMA completion');
    write ('Choice : '); readln (choice);
    if choice = 't' then
      begin
        dma_dat (k); writeln ('DMA_DAT base : ',k)
      end;
    if choice = 'i' then
      begin
        dma_inq (error,dualdma,dmamode,bufferb,nbuf,bfcount,curwcount,k);
        writeln ('error        : ',error);
        writeln ('dualdma      : ',dualdma);
        writeln ('DMA mode     : ',dmamode);
        writeln ('Buffer B     : ',bufferb);
        writeln ('nbuf         : ',nbuf);
        writeln ('buffer count : ',bfcount);
        writeln ('word count   : ',curwcount);
        writeln ('base address : ',k);
      end;
    if choice = 'd' then
      begin
        write ('Give first array-index : '); readln (i);
        if 1 > 980 then i := 980; if i < 1 then i := 1;
        for k := i to i + 20 do writeln (buf1[k],'         ',buf2[k],
                                        '         ',buf3[k])

      end;
    if choice = 'h' then
      begin
        dma_hlt (error,k); writeln ('DMA_HLT error : ',error,'  base : ',k)
      end;
    if choice = 'w' then
      begin
        dma_wfc (error,k); writeln ('DMA_WFC error : ',error,'  base : ',k)
      end;
  end;
{
 Halt any DMA operation in progress and reset the board before leaving the
 program
}
dma_hlt (error,k);
brd_ini ($0800,$0800,base);
end.
```

Appendix C: Data-acquisitie

DATA TRANSLATION® IBM PERSONAL COMPUTER AT™ SERIES
HIGH SPEED SINGLE BOARD ANALOG AND DIGITAL I/O SYSTEMS

DT2821 SERIES

FEATURES

- PC/AT-compatible boards featuring up to 130kHz A/D, 130kHz D/A, 16 digital I/O lines, and on-board pacer clock
- A/D features:
  - □ Throughput to 130kHz
  - □ RAM channel-gain list for random channel sequencing
  - □ 12-bit resolution
  - □ Optional 16-bit resolution (DT2827)
  - □ 100kHz simultaneous sampling (DT2828)
  - □ Programmable gains of 1, 2, 4, 8
  - □ Up to 16SE/8DI channels
- D/A features:
  - □ 130kHz throughput per DAC (260kHz aggregate)
  - □ 2 independent DACs
  - □ Deglitching circuitry reduces noise
  - □ 12-bit resolution
  - □ Outputs are short circuit protected
- Support for five-level bus interrupt
- Programmable Pacer Clock initiates A/D or D/A conversions
- Dual DMA channel architecture supports Continuous Performance (no gap) data collection
- Optional subroutine library (ATLAB) and application software packages
- Shipped with comprehensive user manual which includes programming instructions

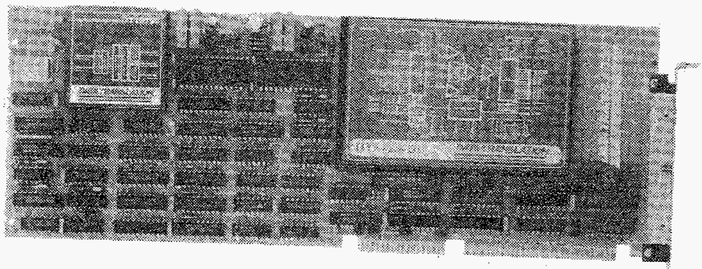


Figure 1. The DT2821 is a complete high performance analog and digital I/O interface for the IBM Personal Computer AT.

MODELS AVAILABLE

DT2821-F-16SE: 130kHz A/D; 16SE inputs.
DT2821-F-8DI: 130kHz A/D; 8DI inputs.
DT2821: 50kHz A/D; 16SE/8DI inputs.
DT2827: 100kHz 16-bit A/D; 4DI inputs.
DT2828: 100kHz simultaneous sampling A/D; 4SE inputs.

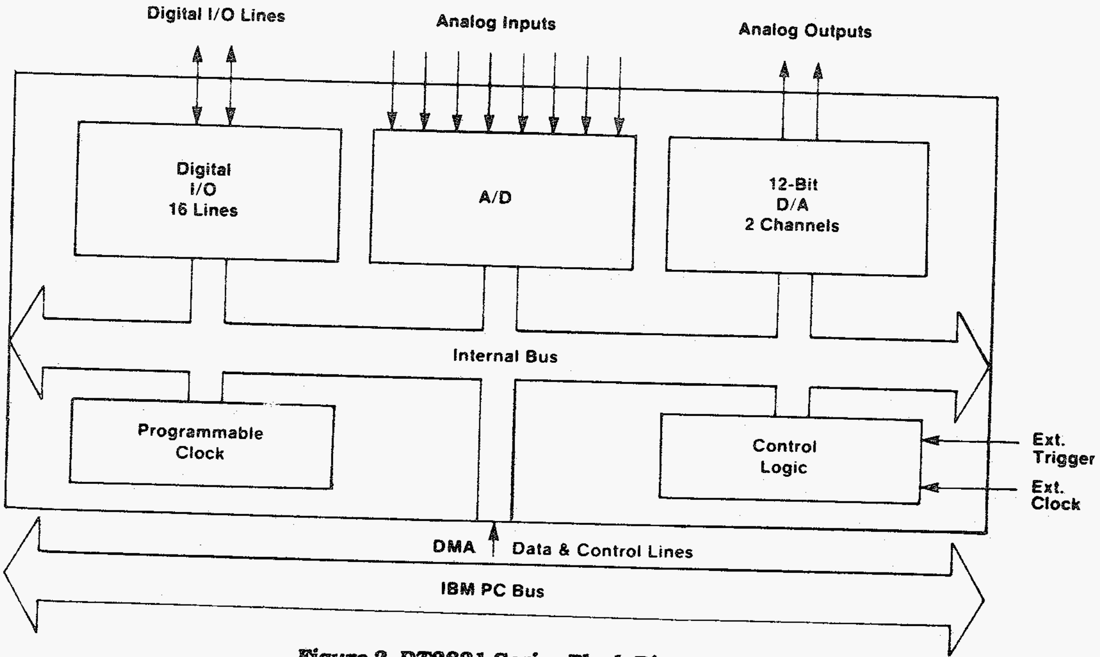


Figure 2. DT2821 Series Block Diagram

Data Translation, Inc. 100 Locke Drive, Marlboro, MA U.S.A. 01752/(617) 481-3700/TLX 951646
Data Translation Ltd., The Business Centre, Wokingham, Berks RG11 2QZ, U.K., Tel. Reading (0734) 793838

DESCRIPTION

The DT2821 Series are high speed analog and digital I/O boards for the IBM Personal Computer AT featuring up to 130kHz throughput and 12-bit or 16-bit resolution to fit any laboratory, industrial, and control applications.

The series consists of four models which differ only in their A/D converters. The DT2821-F has a throughput of 130,000 samples per second, and is factory configured for 16 single-ended or 8 differential inputs. The DT2821 has a throughput of 50,000 samples per second, and is jumper selectable for single-ended or differential inputs. The DT2827 has a throughput of 100,000 samples per second at 16-bit resolution with 4 differential input channels. The DT2828 has a throughput of 100,000 samples per second and simultaneously samples 4 singled-ended channels at the same time. All models except the DT2827 have 12 bits of A/D resolution; the DT2821 models have a programmable gain amplifier for software-selectable gains of 1, 2, 4, and 8; and selectable input ranges of 0 to +10V and ±10V full scale.

A/D channel selection on the DT2821 Series is accomplished using a unique RAM channel-gain list. The RAM channel-gain list is a 16-location memory which allows any of the channels at any gain to be sampled in any sequence at the full throughput rate. It also permits the same channel to be sampled at different gains.

The DT2821 Series' digital to analog (D/A) subsystem consists of two 12-bit deglitched D/A converters. These can provide either single outputs or dual simultaneous outputs. When used in DMA transfer mode, analog data can be output at up to 130kHz per channel from system memory.

In addition, the DT2821 Series boards contain two 8-line digital I/O ports which can be programmed for either input or output transfers.

The DT2821 Series boards also contain a programmable pacer clock which is used to initiate A/D and D/A conversions. The pacer clock operates under program control, and provides a usable range of 7.75μs (129kHz) to 2 seconds. Alternatively an external clock can be selected to start conversions, and an external trigger may be selected to gate A/D and D/A conversion events. The A/D and D/A converters can be operated simultaneously at the same clock speed; alternatively the D/A subsystem can be operated in single conversion mode at any desired rate while the A/D subsystem operates from the pacer or external clock.

An on-board dc-to-dc power converter generates all required analog supply voltages from the +5 volts provided on the IBM PC/AT backplane and provides high noise isolation from the computer system's power supplies.

The DMA interface is compatible with 16-bit data transfers and can be jumper-selected to use DMA channel 5, 6, or 7. DMA buffers may be located anywhere in the 16Mbyte memory space of the PC/AT and may be up to 65,536 words long.

A unique configuration utilizing two DMA channels is also included to support Continuous Performance DMA. Continuous Performance is a data sampling method which provides gap-free transfers of large volumes of data from memory or disk (D/A conversion) or to memory or disk (A/D conversions) without any loss of samples.

The DT2821 Series boards also support interrupts. The interrupt level is jumper-selectable to level 3, 5, 7, 10, or 15.

ARCHITECTURAL FEATURES

High Speed Design

The DT2821 Series boards were designed for high speed applications using the IBM Personal Computer AT. The analog input section uses a pipelined architecture which overlaps the multiplexer's and instrumentation amplifier's settling time on the next sample with the A/D converter's conversion period on the present sample. This pipelined architecture permits the A/D section of the DT2821 Series to achieve throughput rates in excess of 100kHz.

Continuous Performance DMA Operation

The DT2821 Series boards support Continuous Performance DMA operation, a mode which allows the board to perform gap-free DMA transfers of large amounts of data. In Continuous Performance operation, data transfers occur by chaining through a pair of buffers, each associated with a separate DMA channel. Data transfers require both buffers: when the end of the first buffer is reached, the DT2821 Series board chains automatically to the second buffer without pausing or missing any samples. During this chaining, the host CPU is interrupted to enable the controlling program to set up the next DMA channel parameters.

Subsequent transfers may occur to different buffer pairs up to the limit of available system memory. Alternatively, the same two buffers may be used again, provided there is a completion routine which writes data from each buffer to disk (on A/D transfers), or reads data into each buffer from disk (on D/A transfers), while the other buffer is performing data transfers.

Deglitched DAC Operation

The DT2821 Series's D/A circuits contain a proprietary deglitching circuit for reducing noise on DAC outputs.

Conventional D/A converters contain glitches: pulses of short duration (a few microseconds) but high energy (10 to 100 LSBs) caused by charge imbalances in the converter's data switches. Glitches can significantly reduce the effective accuracy of the converter.

The DT2821 Series' deglitching circuit consists of a sample and hold connected to the output of each of the board's D/A converter chips. Since glitches occur when new data is written to the D/A converter, the DT2821 Series board's D/A converter output is disconnected whenever a new input value is written. After the glitch has passed, the D/A converter's output is reconnected to the DT2821 Series board's output connections.

nterrupts.
vel 3, 5, 7,

for high
Computer
l architec-
istrumen-
mple with
e present
the A/D
roughput

us Perfor-
the board
nounts of
on, data
f buffers,
nel. Data
id of the
d chains
using or
the host
program

it buffer
memory.
ed again,
h writes
sfers), or
on D/A
ing data

propri-
on DAC

: pulses
n energy
s in the
ificantly

sts of a
h of the
ır when
DT2821
nnected
e glitch
nnected
ns.

RAM Channel-Gain List

The DT2821 Series boards feature a RAM channel-gain list. This sampling mechanism is a 16-location RAM memory which is used to specify the sequence of channels and the gain at which each channel will be sampled. Successive locations in RAM memory are loaded with channel-gain pairs. When an A/D scan begins, the DT2821 Series board sequences through the RAM, sampling each channel entry at the specified gain (there is no programmable gain option on the DT2827 or the DT2828, gain=1). This sampling mechanism is very flexible and allows the user to predefine virtually any sampling sequence desired: sequential sampling (0 to n, or n to m channels, for example); random channel sampling (channels 0, 3, 2, 5, for example); or successive samples on a single channel at the same or different gain. The user can also specify that not all elements in the channel-gain list be used, thus accommodating sequences of fewer than 16 samples.

Interrupt Support

The DT2821 Series boards support interrupts to the PC/AT's processor upon the completion of significant events. The board's interrupt source is selected under program control, and includes the following: A/D Done (programmed I/O transfers); A/D DMA Done (DMA transfers); A/D Scan Done; A/D Error; D/A Ready (programmed I/O transfers); D/A DMA Done (DMA); or D/A Error.

The DT2821 Series boards can interrupt the processor on any of five levels: 10 (highest), 15, 3, 5, or 7 (lowest). The interrupt level is jumper-selectable by the user.

Programmable Pacer Clock

The on-board pacer clock may be used to initiate A/D, D/A, or simultaneous A/D and D/A conversions. The pacer clock is made up of three elements: a 4MHz on-board pulse generator; a clock prescaler, which serves as a programmable divider, and which can be set for powers of 2 from 0 to 15; and a counter divisor. The counter divisor is an 8-bit programmable counter that divides the prescaler output by any number from 1 to 256.

PRINCIPLES OF OPERATION

DIO Section

The digital I/O section of the DT2821 Series is controlled by programmed I/O (PIO) reads and writes. Data transfers can occur at any time asynchronously, ie. you can output to or input from the DIO section at any time without having to check any status bits.

D/A Section

The D/A section can be controlled by programmed I/O or DMA transfers. In the PIO mode, data values are output to the DACs either singly or alternating in dual channel. The actual D/A conversion does not take place until a DAC single conversion command is given or the clock (internal or external) has been enabled and started with a software start command. The next value can be written to the DAC's when the program detects D/A Ready or can use a system interrupt if Interrupt on D/A Ready is enabled. In the DMA mode, the user programs the DMA controller, enables the clock (internal or external), and initiates the operation with either a software start or external trigger. The operation terminates when the DMA controller finishes. If dual DMA is selected, when the first DMA controller is finished, the DT2821 Series board automatically switches to the second DMA channel, and when finished switches back to the first. This continues until the program disables the dual DMA mode.

A/D Section

The A/D section is also controlled by either programmed I/O or DMA transfers. In PIO mode, if the pacer clock is disabled, each time the program issues a software start the DT2821 Series board does a conversion on the channel/gain of the next channel-gain list RAM entry; when the final address in the the channel-gain list is reached, selection loops around to the first value. If the pacer clock is enabled (either internal or external), when the program issues a software start (or an external trigger if enabled), the DT2821 Series board starts at channel-gain list entry 0 and, at each pacer clock tick, cycles through to the RAM channel-gain entry designated as the final address and then loops back to location 0. The program can detect when to read the converted data by checking the status of A/D Done or by a system interrupt if enabled. In the first DMA mode, the user programs the DMA controller, enables the clock, and issues a software start (or issues an external trigger). The conversions and transfers will continue until the DMA controller is finished. If dual DMA mode is selected, when the first DMA controller is finished, the DT2821 Series board switches to the second DMA channel; it switches back to the first when the second is complete. In the second DMA mode (triggered scan), the user starts as he would in the first DMA mode. The board then scans through all the channel-gain entries to the final address each time it receives a software start or an external trigger. The DMA operation continues until the DMA controller is finished.

REGISTER DESCRIPTION

All functions on the DT2821 Series boards are controlled and monitored by writing commands, command parameters, and data to, or by reading the board status and data from, registers on the board. The board contains eight I/O mapped 16-bit registers, and occupies eight contiguous word locations. The starting location of the following registers can be anywhere between 200 (hex) and 3E0 (hex) in increments of 20 (hex) in the host CPU's I/O address space.

Register Name	Address	Access
A/D Control/Status (ADCSR)	Base	Read/Write
Channel List CSR (CHANCSR)	Base + 2	Read/Write
A/D Data (ADDAT)	Base + 4	Read only
D/A, DIO Control/Status (DA/DIOCSR)	Base + 6	Read/Write
D/A Data (DADAT)	Base + 8	Write only
DIO Data (DIODAT)	Base + A	Read/Write
Supervisory Control/Status (SUPCSR)	Base + C	Read/Write
Pacer Clock (TMRCTR)	Base + E	Read/Write

A/D Control/Status Register

The ADCSR is a read/write register which controls and monitors all activity associated with the A/D section of the DT2821 Series board and is located at the base address. A write operation accesses the control functions and a read accesses the status information of the A/D section.

BIT DEFINITIONS

A/D Error	Reserved					A/D Clock Enable	Multiplexer Busy
15	14	13	12	11	10	9	8

A/D Done	Inter. on A/D Done	Gain Select		A/D Mux Channel Select			
7	6	5	4	3	2	1	0

A/D Error
This bit indicates that one of the following A/D errors has occurred: attempting an A/D conversion while the module is busy (trigger error); attempting an A/D conversion while the DMA pipeline is full (data overrun error).

Enable A/D Clock
Enables the output of the pacer clock or external clock to initiate A/D conversions.

Multiplexer Busy
Indicates when the input mux has settled and the input A/D sample is stable.

A/D Done
This bit indicates that the A/D has completed a conversion. In PIO operations, this bit is polled to control a software read operation.

Interrupt on A/D Done
This bit enables or disables interrupts from the A/D section on the completion of data conversions.

Gain Select
These bits specify the gain which will be associated in the RAM channel-gain list with the channel specified in bits 8 to 11. 7 bits 0 to 3

A/D Mux Channel Select
These bits specify the next channel number which will be loaded into the RAM channel-gain list.

CHANNEL LIST CONTROL/STATUS REGISTER
The CHANCSR is a read/write register that controls and monitors most of the activity associated with the Channel-gain list RAM section of the DT2821 Series board and is located at the base address + 2.

BIT DEFINITIONS

Load List Enable	Reserved			Present List Address			
15	14	13	12	11	10	9	8

Reserved				Final List Address			
7	6	5	4	3	2	1	0

Load List Enable
When set enables the channel-gain information from the ADCSR register to be loaded in successive locations of the channel-gain list RAM. When reset, inhibits any channel-gain list changes.

Present List Address
Read only input that indicates the present address of the channel-gain list.

Final List Address
Used for setting or getting the final address to be used for the channel-gain list sequence.

A/D DATA REGISTER
This register contains the digital data resulting from an A/D conversion. Data obtained using unipolar input ranges is coded in binary; bipolar input ranges may be coded in offset binary, two's complement binary, or two's complement binary with sign extension.

D/A, DIO CONTROL/STATUS REGISTER
The DADIOCSR is a register which controls and monitors the D/A and Digital I/O (DIO) activity of the DT2821 Series board.

BIT DEFINITIONS

D/A Error	Reserved					Y Channel Select	Single Channel
15	14	13	12	11	10	9	8

D/A Ready	Intr. on D/A Ready	Enable D/A Clock	Reserved			DIO High Byte Output	DIO Low Byte Output
7	6	5	4	3	2	1	0

D/A Error
This indicates that one of the following errors has occurred: attempting any conversion while the D/A ready bit is set (trigger error); attempting any conversion while the DMA pipeline is empty (data late error).

Y Channel Select
When the DT2821 Series board is in single-channel D/A mode, this bit selects between the X D/A channel (DAC 0) and the Y D/A channel (DAC 1).

Single Channel
This bit selects whether the DT2821 Series board is operating in single-channel or alternating dual-channel mode.

D/A Ready
This bit indicates that the D/A has completed a conversion and a new conversion can begin.

Interrupt on D/A Ready
This bit enables or disables interrupts from the D/A section upon D/A Ready or DMA Done.

Enable D/A Clock
This bit enables the output of the pacer clock to initiate the D/A conversions.

DIO High B
This bit de
DIO high b

DIO Low B
This bit de
DIO low byt

D/A DATA
This regist
analog by o

DIO DATA
This regist
digital I/O

SUPERVIS
The SUPC
including t
sion mode
and clock
board.

DMA Done	Inter E						
15	14						

Single D/A Conve	Ini						
7	6						

DMA Done
This bit in
has compl
operation.

Interrupt
This bit e
A/D error

Clear DM
This bit cl

Dual DMA
This bit e
DMA oper
used to tra
When the
the secon
reached o
Series boa
This conti

DMA Sele
These bits
clocked D
DMA, and

Buffer B
In dual-D
Performan
being use

DIO High Byte Output
This bit determines the direction of data flow on the DIO high byte.

DIO Low Byte Output
This bit determines the direction of data flow on the DIO low byte.

D/A DATA REGISTER
This register receives a digital value to be converted to analog by one of the D/A converters.

DIO DATA REGISTER
This register is used for data transfer to and from the digital I/O lines.

SUPERVISORY CONTROL/STATUS REGISTER
The SUPCSR controls a variety of supervisory tasks, including the DMA section, error clearing, DAC conversion mode, board initialization, multiplexer preload, and clock source selection for the DT2821 Series board.

BIT DEFINITIONS

DMA Done	Interrupt on Error	Clear DMA Done	Dual DMA	DMA Select		Buffer B	Scan Done
15	14	13	12	11	10	9	8

Single D/A Convert	A/D Initialize	DAC Initialize	Preload A/D Mux	Soft Start	Enable Ext Trigger	Enable Ext Clock	Init
7	6	5	4	3	2	1	0

DMA Done
This bit indicates when the requested DMA operation has completed and the board is ready for a new DMA operation.

Interrupt on Error
This bit enables an interrrupt to the system when an A/D error or when a D/A error occurs.

Clear DMA Done
This bit clears the DMA Done and the DMA Error bits.

Dual DMA
This bit enables or disables Continuous Performance DMA operation. In this mode, two DMA channels are used to transfer data continuously to or from memory. When the terminal count is reached, control is given to the second DMA channel. When the terminal count is reached on the second DMA channel, the DT2821 Series board switches back to the first DMA channel. This continues until this bit is cleared.

DMA Select
These bits select among the following DMA modes: A/D clocked DMA, D/A clocked DMA, A/D triggered scan DMA, and no DMA.

Buffer B
In dual-DMA mode (which is used for Continuous Performance), this bit indicates which DMA buffer is being used.

Scan Done
This bit indicates that a scan of A/D conversions has completed. That is, that the A/D section has sequenced through the RAM channel-gain list until the scan address equals the final address in the list.

Single D/A Convert
This bit causes a D/A conversion. This bit can be used to initiate D/A conversions simultaneous with A/D conversions, but occurring at a different rate.

A/D Initialize
This bit is used to initialize the A/D subsystem.

DAC Initialize
This bit is used to intialize the D/A subsystem.

Preload A/D Mux
This acts as an initialization of the pipeline that overlaps the A/D conversion and the next channel selection and sample.

Soft Start
This bit issues a software trigger to initiate A/D, D/A, or simultaneous A/D and D/A conversions.

Enable External Trigger
This bit allows the external trigger signal on DT2821 Series board's J1 connector to initiate A/D, D/A, or simultaneous A/D and D/A conversions

Enable External Clock
This bit enables the external clock input on the DT2821 Series board's connector J1 to be used in place of the on-board pacer clock to initiate the second and subsequent conversions in multiple A/D, multiple D/A, or multiple A/D and D/A modes

Board Initialization
This bit initializes the board. This clears all the error bits and data buffers.

PACER CLOCK REGISTER
The Pacer Clock Register determines the frequency of the on-board pacer clock. The pacer clock consists of a 4MHz oscillator which can be divided down by a prescaler value and by a divisor. The prescaler can assume the value of any power of two from 0 to 15 (that is, from 1 to 32,768). The divisor can assume any integer value from 1 to 256. The usable range of the pacer clock is 7.75μs (129kHz) to 2 seconds in 250ns increments.

2^{15}
2^{6}

BIT DEFINITIONS

Reserved				Prescaler Select Bits			
15	14	13	12	11	10	9	8

Counter Bits							
7	6	5	4	3	2	1	0

Prescaler Select Bits
These bits specify the prescaler value. The prescaler value equals two raised to the power specified in these bits.

Counter Bits
These bits specify the value by which the prescaled clock frequency is divided before being output as the pacer clock signal.

SPECIFICATIONS

(Typical at +25°C and rated power, unless otherwise specified)

A/D SUBSYSTEM	DT2821	DT2821-F	DT2827	DT2828
Number of Inputs	16SE/8DI (jumper)	16SE or 8DI (factory)	4DI	4SE
Resolution	12 bits	12 bits	16 bits	12 bits
Programmable Gain Range	1,2,4,8	1,2,4,8	None	None
A/D Throughput	50kHz	130kHz	100kHz	100kHz
A/D Conversion Time	10μs	4μs	6μs	4μs
Channel Acquisition Time ±.5 LSB	15μs	4μs	6μs	6μs
Sample and Hold Aperture Uncertainty	10ns	5ns	5ns	5ns
Sample and Hold Aperture Delay	50ns	100ns	100ns	100ns
System Accuracy percent of FSR	±.03%	±.03% (G=1) ±.04% (G=2) ±.05% (G=4) ±.07% (G=8)	±.03%	±.03%
Input Ranges (jumper)				
Unipolar	0 to +10V	0 to +10V	None	0 to +10V
Bipolar	±10V	±10V	±10V	±10V
Output Coding (Jumper)	Binary Offset Bin. Two's Comp.	Binary Offset Bin. Two's Comp.	Binary Offset Bin. Two's Comp.	Binary Offset Bin. Two's Comp.
Common Mode Input Voltage, Maximum	±11V	±10.5V	±10.5V	±10.5V
Common Mode Rejection Ratio, Gain=1,@60Hz	80dB, 1kΩ	80dB, 1kΩ	80dB, 1kΩ	80dB, 1kΩ
Maximum Input Voltage Without Damage				
Power on	±35V	±27V	±27V	±31V
Power off	±20V	±12V	±12V	±20V
Input Impedance				
Off Channel	100MΩ,10pF	100MΩ,10pF	100MΩ,20pF	10MΩ,20pF
On Channel	100MΩ,100pF	100MΩ,50pF	100MΩ,100pF	100MΩ,100pF
Bias Current	±20nA	±10nA	±50nA	±200nA
Nonlinearity	±0.5 LSB	±0.5 LSB	±0.5 LSB	±0.5 LSB
Differential Nonlinearity	±0.5 LSB	±0.5 LSB	±0.5 LSB	±0.5 LSB
Inherent Quantizing Error	±1/2 LSB	±1/2 LSB	±1/2 LSB	±1/2 LSB
A/D Zero Drift	±10ppm/°C	±50μV/°C	±50μV/°C	±20ppm/°C
Gain Drift (of FSR/°C)	±30ppm	±30ppm	±30ppm	±30ppm
Differential Linearity Drift (of FSR/°C)	±3ppm	±3ppm	±3ppm	±3ppm
Monotonicity	0 to +70°C	0 to +70°C	0 to 70°C	0 to 70°C

D/A SUBSYSTEM (ALL MODELS)

Number of DACs	2
Resolution	12 bits
Settling Time to 0.01% of FSR	
20V step	5μs
100mV step	1μs
Throughput	130kHz max. (single channel) 260kHz max. (aggregate)
Slew Rate	10V/μs
Glitch Energy	15mV/μs
Output Ranges (jumper)	0 to +5V, 0 to +10V ±2.5V, ±5V, ±10V
Input Data Coding (jumper)	
Unipolar	Straight binary
Bipolar	Offset binary
Output Current	±5mA
Output Impedance	0.1Ω
Capacitive Drive Capability	0.004μF
Protection Against	Short circuit to analog common
Nonlinearity	1/2 LSB
Differential Nonlinearity	1/2 LSB
Inherent Quantizing Error	1/2 LSB
Gain Error	Adjustable to zero
Zero Error	Adjustable to zero
Gain Drift	±30ppm of FSR/°C
Zero Drift (Bipolar)	±10ppm of FSR/°C
Monotonicity	0 to +70°C

DIGITAL
Number c
Number c
Logic Fan
Logic Ser
Input Typ
Input Ter
Input Log
Logic Hig
Logic Low
Logic Hig
Logic Low
Fanout
Logic Hig
Logic Low
Logic Hig
Logic Low

EXTERN
Input Typ
Logic Fan
Logic Loa
Input Ter
Logic Hig
Logic Low
Logic Hig
Logic Low
Minimum
Clock H
Clock L

ON-BOAI
Base Freq
Prescaler

Divisor R
Usable Ra

EXTERN
Input Typ
Logic Fan
Logic Loa
Input Ter
Logic Hig
Logic Low
Logic Hig
Logic Low
Minimum
Clock H
Clock L

DIGITAL I/O SUBSYSTEM (ALL MODELS)

Number of DIO lines	16
Number of Ports	Two 8-bit ports
Logic Family	LSTTL
Logic Sense	Positive true
Input Type	Level sensitive
Input Termination	None; unused inputs float
Input Logic Load	1 LSTTL load
Logic High Input Voltage	2.0V minimum
Logic Low Input Voltage	0.8V maximum
Logic High Input Current	20μA maximum
Logic Low Input Current	-0.4mA maximum
Fanout	8 LSTTL loads
Logic High Output Voltage	2.4V minimum
Logic Low Output Voltage	0.4V maximum
Logic High Output Current	-15mA maximum
Logic Low Output Current	24mA maximum

EXTERNAL TRIGGER (ALL MODELS)

Input Type	Schmitt trigger, edge sensitive, clocks on falling edge
Logic Family	LSTTL
Logic Load	1 LSTTL load
Input Termination	22kΩ pullup to +5V
Logic High Input Voltage	2.0V minimum
Logic Low Input Voltage	0.8V maximum
Logic High Input Current	-0.25mA maximum
Logic Low Input Current	25μA maximum
Minimum Pulse Width	
Clock High	200ns
Clock Low	50ns

ON-BOARD CLOCK (ALL MODELS)

Base Frequency	4.00MHz ±0.01%
Prescaler Range	Powers of two from 0 to 15 (values from 1 to 32,768)
Divisor Range	Integer values from 1 to 256
Usable Range	7.75μs (129kHz) to 2 seconds

EXTERNAL CLOCK (ALL MODELS)

Input Type	Schmitt trigger, edge sensitive, clocks on falling edge
Logic Family	LSTTL
Logic Load	1 LSTTL load
Input Termination	22kΩ pullup to +5V
Logic High Input Voltage	2.0V minimum
Logic Low Input Voltage	0.8V maximum
Logic High Input Current	-0.25mA maximum
Logic Low Input Current	25μA maximum
Minimum Pulse Width	
Clock High	200ns
Clock Low	110ns

INTERFACE CHARACTERISTICS (ALL MODELS)

Compatible Bus	IBM PC/AT
Interface Type	I/O mapped with 10-bit addressing
Number of Locations Occupied	8 words
Data Path	16 bits
Base Address Range (jumper)	200 (hex) to 3E0 (hex) in increments of 20 (hex)
Factory-Assigned Base Address	240 (hex)
Interrupts	1 interrupt, jumper-configurable to any of five interrupt lines
Interrupt Levels	3, 5, 7, 10, and 15
Interrupt Sources	A/D, D/A Error; A/D Done; D/A Ready; A/D Scan Done; DMA Done
Mating Connector	50-pin 3M type, user-supplied Ansley 609-5030 or equivalent Ansley 609-5031 or equivalent strain relief recommended
Power Requirements +5V	
DT2821	±5%, @ 1.8A maximum (1.5A typical)
DT2821-F	±5%, @ 2.2A maximum (1.8A typical)
DT2827	±5%, @ 2.5A maximum (2.0A typical)
DT2828	±5%, @ 2.0A maximum (1.7A typical)
Physical/Environmental Dimensions	4.5"H x 13.25"W x .75"D (11.4 X 33.7 X 1.9 cm)
Weight	20 ounces (567g)
Operating Temperature Range	0 to +70°C (32 to 158°F)
Storage Temperature Range	-25 to +85°C (-13 to +185°F)
Relative Humidity	To 95% non-condensing

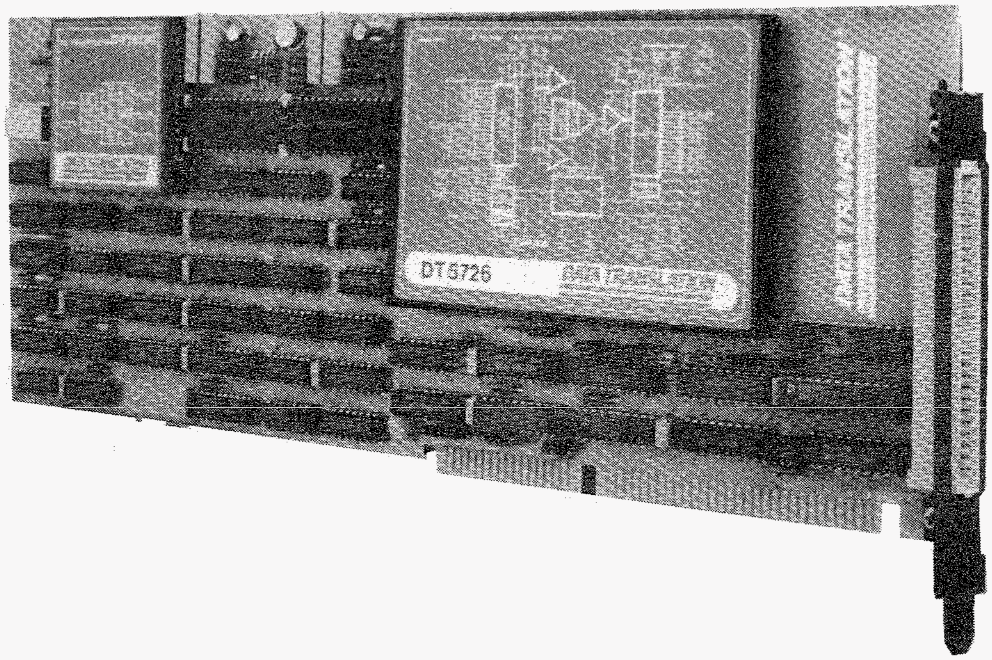


Figure 3. The DT2827 single board high performance analog and digital I/O system supports an A/D resolution of 16-bits at 100kHz throughput.

Screw Ter

Two screw
with the I
DT752.

The DT7C
user conn
convenien
tions are l
tions, ana
clock inpu
blank cir
current s
inputs (c
differentia
board). Th
(3.3-foot) f
DT2821 S
designed f
with rubb

The DT75
input lines
This conn
used with
signal co
isolated D
DT6700 s
used with
board pro
also provic
signals, ar
thermocou
lated sign
features,
junction c
completion

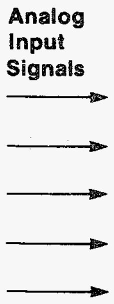


***Figure4.
includes
analog in***

Screw Terminal Panel Accessories

Two screw terminal panels are recommended for use with the DT2821 Series boards: the DT707 and the DT752.

The DT707 screw terminal panel accommodates all user connections to the DT2821 Series board on convenient screw terminal connections. Screw connections are labeled in silkscreen for analog input connections, analog output connections, DIO lines, external clock input, and external trigger input. In addition, blank circuit pads are provided for user-supplied current shunts to accommodate 0 to 20mA current inputs (current inputs should be used only with differential input versions of the DT2821 Series board). The DT707 contains an integral one-meter (3.3-foot) flat ribbon cable which plugs directly into the DT2821 Series board's J1 connector. The DT707 is designed for table-top mounting, and comes equipped with rubber feet.

The DT752 is similar to the DT707, except analog input lines are directed to a separate 20-pin connector. This connector permits the DT2821 Series board to be used with any of a wide range of Data Translation signal conditioning accessories, including the non-isolated DT709-Y or DT756-Y and the isolated DT750/DT6700 series of signal conditioning modules. When used with the DT709-Y or DT756-Y, a DT2821 Series board provides 16 differential inputs. The DT756-Y also provides selectable gain to accommodate low level signals, and offset and cold-junction compensation for thermocouple inputs. The DT750/DT6700 series isolated signal conditioning products offer a variety of features, including differential inputs, gain, cold-junction compensation, RTD linearization, or bridge-completion circuitry.

ATLAB - SUBROUTINE LIBRARY

Features:

- Easy to use, real-time software package to support the DT2821 series boards
- CALLable from FORTRAN, C, and Pascal
- Library routines for control of all on-board analog and digital I/O functions
- Supports Continuous Performance to memory or disk
- Shipped complete on a single diskette together with a comprehensive user manual

ATLAB is a real-time software package for Data Translation's IBM PC/AT compatible DT2821 series of analog and digital I/O system interfaces. The package consists of libraries of routines designed to be CALLed from Microsoft's FORTRAN (rev 3.3 or higher), C (rev 3.0 or higher) and Pascal (rev 3.3 or higher) and operated under PC DOS (rev 3.0 or higher). The ATLAB package allows the user to control all the analog and digital I/O capabilities of the interface boards through operation-specific routines which greatly simplify the programming of the hardware interfaces. With ATLAB, direct access to the control registers is not required. An error processing system checks for argument errors and will generate an error report if any are detected. Attempts to operate the interfaces in illegal modes of operation are reported and not executed. Routines are included for analog input, analog output, digital input, digital output, continuous sampling (memory and disk), pacer clock control, and configuration setup.

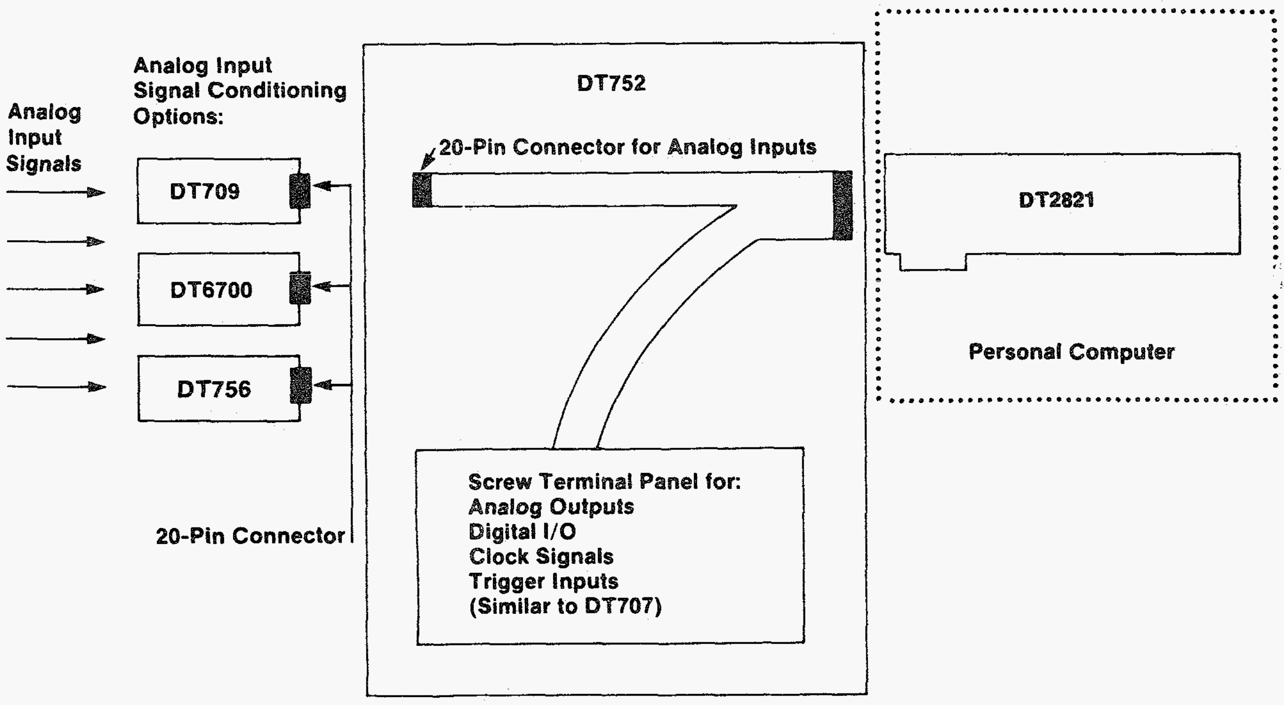


Figure 4. The DT752 is a screw terminal panel for the analog outputs, digital I/O, and clock signals of the DT2821 and includes a 20-pin connection for the analog input signals. The DT752 is compatible with isolated and non-isolated analog input signal conditioning options.

DT2821 SERIES SUMMARY

Model	A/D Channels	A/D Throughput	Resolution	D/A (12-bit)	DIO
DT2821-F	16SE or 8DI	130kHz	12-bit	2 channels	16 line
DT2821	16SE/8DI	50kHz	12-bit	2 channels	16 line
DT2827	4DI	100kHz	16-bit	2 channels	16 line
DT2828	4SE*	100kHz	12-bit	2 channels	16 line

*Simultaneous Sampling

ORDERING GUIDE

All DT2821 Series boards are shipped with a comprehensive user manual that includes programming instructions.

DT2821-F-16SE

IBM PC/AT-compatible analog and digital I/O board with A/D, D/A, digital I/O, and on-board pacer clock. The A/D subsystem features 16SE analog inputs, 12-bit resolution, programmable gain (gains of 1, 2, 4, or 8), and 130kHz throughput. The D/A subsystem features two 12-bit deglitched D/A converters with 130kHz throughput. The digital I/O subsystem features 16 lines configurable for input or output in two 8-bit ports.

DT2821-F-8DI

As the DT2821-F-16SE, but 8DI input channels only, and 130kHz A/D throughput.

DT2821

As the DT2821-F, but 16SE/8DI input channels (user selectable), and 50kHz A/D throughput.

DT2827

As the DT2821-F, but 4DI channels of 16-bit A/D resolution input and throughput rate of 100kHz.

DT2828

As the DT2821-F, but 4SE channels of A/D inut with a throughput rate of 100kHz where the 4 input channels are simultaneously sampled within a ±5ns aperture.

ATLAB

SP0143

Software subroutine package that provides the user high level language calls to subroutines that support all of the analog I/O, digital I/O, and clock functions of the DT2821 Series of boards. Languages supported are Microsoft C, Pascal, and FORTRAN.

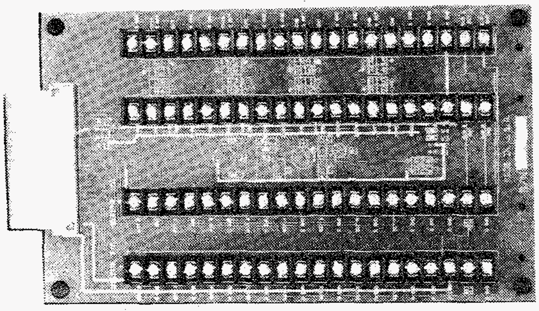


Figure 5. The DT707 supports all the analog I/O, digital I/O, and clock functions of the DT2821 and provides for convenient connection of user input signals.

ACCESSORIES

DT707 Screw Terminal Panel

The DT707 is a screw terminal panel for all A/D, D/A, DIO, external trigger, and external clock connections. It includes a 1 meter (3.3-foot) flat ribbon cable.

DT752 Signal Conditioning/Screw Terminal Panel

Complete with cable and screw terminal panel for D/A, digital I/O, and trigger functions, the DT752 allows several signal conditioning products such as the DT709-Y, DT756-Y, and DT6700 to be used with the A/D sections of the DT2821 Series boards. See individual signal conditioning product data sheets for complete details.

DATA TRANSLATION®

10-05607-1.

Data Translation, Inc. 100 Locke Drive, Marlboro, MA U.S.A. 01752/(617) 481-3700/TLX 951646
Data Translation Ltd., The Business Centre, Wokingham, Berks RG11 2QZ, U.K., Tel. Reading (0734) 793838