

# **PC 41**

## **8 Channel Tracking LVDT to Digital Convertor Card**

### **User's Manual**

**For the IBM PC, PC/XT, PC/AT, PS/2  
386, 486, ISA and EISA computers**

All rights reserved. No part of this Publication may be copied, stored in a retrieval system, or transmitted, in any form by any means, electronic, mechanical, by photocopying, recording, or otherwise, in whole or in part, without prior written permission from the Publishers.

First Edition

May 1995

May 1995 Printing

Information furnished in this manual is believed to be accurate and reliable; however no responsibility is assumed for its use, nor for any infringements of patents or other rights of third parties which may result from its use.

IBM, IBM PC/XT/AT and IBM PS/2 are trademarks of International Business Machine Corporation. MicroSoft and MSDOS are trademarks of MicroSoft Corporation.

© Copyright: Boston Technology; Eagle Technology

## TABLE OF CONTENTS:

PREFACE.

FEATURES.

1.0 INTRODUCTION.	1
1.1 Software.	1
1.2 Applications.	1
1.3 Ordering information.	2
2.0 HARDWARE.	3
2.1 Architecture.	3
2.2 Addressing Requirements.	5
2.3 Interconnections.	6
2.3.1 Output Connector.	6
2.3.2 Pin Definitions.	7
2.4 Timing Requirements.	9
3.0 SOFTWARE.	10
3.1 Programming the PC41	10
3.1.1 Address Allocation.	10
3.1.2 Register Functions.	10
3.1.3 Data Format.	12
3.2 Software Drivers and Demonstration Software.	12
3.2.1 Operation.	12
3.2.2 Software Drivers.	15
3.2.3 Sample Programs.	19
4.0 SINGLE AND DUAL SPEED SYSTEM SETUP.	21
BIBLIOGRAPHY	22

APPENDICES

APPENDIX A      SPECIFICATIONS

**PREFACE:**

This manual is written for users of the PC41 series I/O cards. It provides all the necessary information required to successfully operate and program the PC41 series.

This manual assumes:

- a) That you have a detailed knowledge of synchro and resolver operation.
- b) That you are familiar with the PC environment.
- c) That you are capable of writing your own programs or modifying the demonstration software for your own use.

**FEATURES:**

- LOW COST
- ACCURACY OF 0.1% OF FULL-SCALE
- OVER-RANGE BIT PROVIDED
- 1 TO 8 CHANNELS
- 11 BIT POSITION OUTPUT
- HIGH TRACKING RATE TO 100 RPS
- EXCITATION FREQUENCIES FROM 400Hz TO 10kHz
- SEPARATE REFERENCE INPUTS FOR EACH CHANNEL
- INSENSITIVE TO AMPLITUDE CHANGES
- HIGH REPEATABILITY
- DRIFT FREE
- VELOCITY OUTPUTS PROVIDED

## **1.0 INTRODUCTION.**

The PC41 is a versatile, full size IBM PC card designed for 1 to 8 channels of tracking LVDT to Digital conversion.

Input signal ranges are 2.5V rms. Options of frequency range vary from 400Hz to 10kHz.

A high quality velocity output is provided for each channel which has a full scale voltage of  $\pm 10V$ . An over-range bit is also provided for each channel, which is readable by software.

The card is double buffered and the 11 bit angle information is provided in two 8-bit bytes to the computer. The I/O addressing space is switch selectable with a DIP switch.

### **1.1 Software.**

Complete Pascal and C software drivers and demonstration programs are provided.

### **1.2 Applications.**

The multi-channel PC41 is designed for use in high performance control and simulation systems. With its large number of channels, the PC41 is an excellent choice for applications including avionics control, machine tooling, and process control.

### 1.3 Ordering Information.

Numerous models are provided:

Two models are provided:

MODEL	REF VOLTAGE	L-L VOLTAGE	FREQUENCY
A	2.5 V	2.5 V	400 Hz
B	2.5 V	2.5 V	1kHz-10kHz

SLEW RATE	
400 Hz	200 LSB/ms
1kHz-10kHz	400 LSB/ms

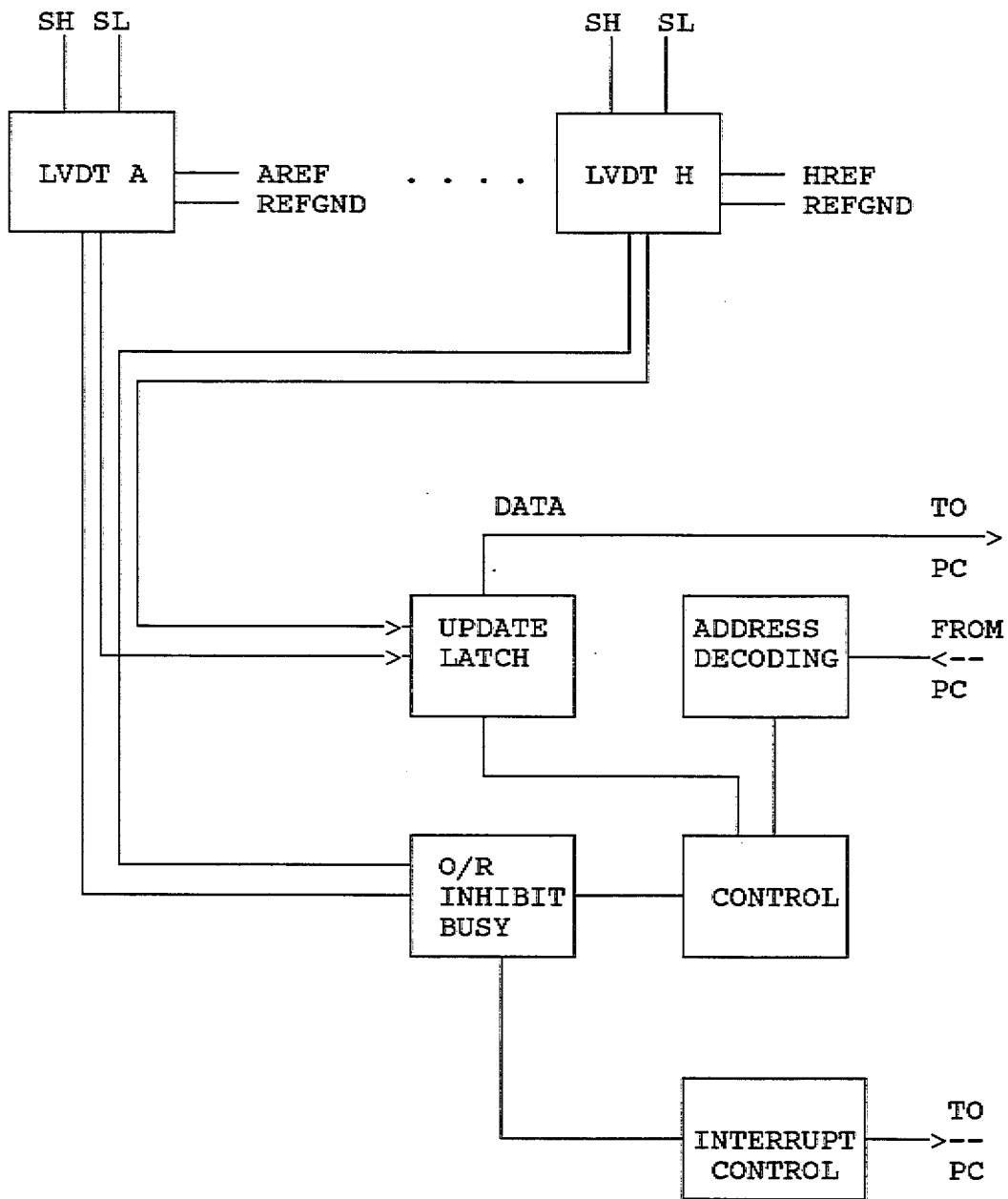
SETTLING TIME	
400 Hz	50 ms
1kHz-10kHz	25 ms

## 2.0 HARDWARE.

### 2.1 Architecture:

The PC41 is a full size IBM PC card which can accept up to eight channels LVDT to Digital conversion channels. The maximum resolution is 11 bits.

The LVDT signals are converted to digital position information, which is updated at each equivalent least significant bit change of the piston. The data is transferred via latches to the PC at a read instruction from the PC. It is arranged in two bytes per channel, therefore two reads from the PC will be required. An over-range bit is provided. An inhibit latch is also provided to prevent a channel latch from being updated while it is being read. A BUSY status-latch can be polled for busy status. The BUSY status is set when a channel has had a least significant bit change. The BUSY latch can be reset. An interrupt mode can also be set up to update the channels transparently.



BLOCK DIAGRAM OF PC41

PC41                      HARDWARE REQUIREMENTS		
COMPUTER HARWARE REQUIREMENTS: IBM PC/XT/AT or Compatible with 512k RAM		
PARAMETER	UNITS	VALUE
POWER SUPPLY: current consump. (5V)	A	3.0 max. with 8 chan.
TEMPERATURE RANGE operating temp. storage	°C °C	0 to 70 -55 to 125
PHYSICAL ATTRIBUTES full size IBM PC card	in. cm	4.3 x 13.1 x 0.59 11.0 x 33.4 x 1.50

## 2.2 Addressing Requirements.

The PC41 card uses 20 I/O addresses and the base address is factory preset to be located at \$720. This location may be adjusted by setting a DIP switch on the PC41 card. The address locations may be varied from \$0000 to \$0FE0 in steps of 32.

The address in decimal is set by the following formula:

$$A = 16 * ( SW1 * 128 + SW2 * 64 + SW3 * 32 + SW4 * 16 + \\ + SW5 * 8 + SW6 * 4 + SW7 * 2 ).$$

( ON = 1; OFF = 0 )

A = \$720

SW1	SW2	SW3	SW4	SW5	SW6	SW7
OFF	ON	ON	ON	OFF	OFF	ON

eg. A = \$700

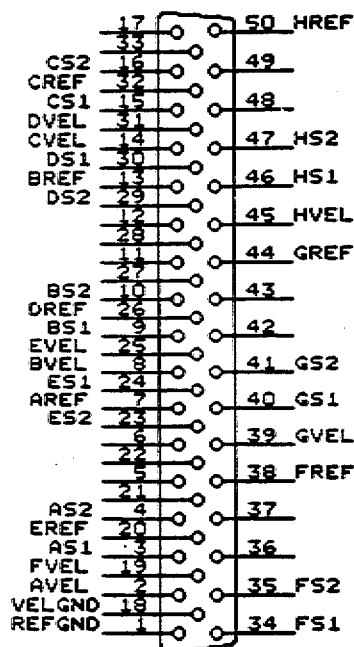
SW1	SW2	SW3	SW4	SW5	SW6	SW7
OFF	ON	ON	ON	OFF	OFF	OFF

## 2.3 Interconnections.

### 2.3.1 Output Connector.

The output connector on the PC41 board is a **male** DB50 connector. The connections are shown below:

1	REFGND	18	VELGND	34	FS1
2	AVEL	19	FVEL	35	FS2
3	AS1	20	EREF	36	
4	AS2	21		37	
5		22		38	FREF
6		23	ES2	39	GVEL
7	AREF	24	ES1	40	GS1
8	BVEL	25	EVEL	41	GS2
9	BS1	26	DREF	42	
10	BS2	27		43	
11		28		44	GREF
12		29	DS2	45	HVEL
13	BREF	30	DS1	46	HS1
14	CVEL	31	DVEL	47	HS2
15	CS1	32	CREF	48	
16	CS2	33		49	
17				50	HREF



Male DB50 connector as seen from rear of PC41.

### 2.3.2 Pin Definitions.

Note that prefix A, B, C, D, E, F, G or H refers to the channel numbers 1, 2, 3, 4, 5, 6, 7 or 8 respectively.

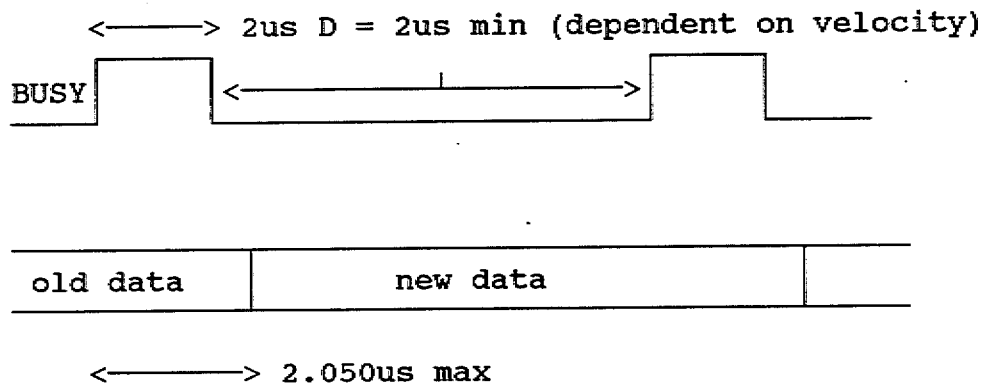
- a) **REFGND and xREF:** REFGND is the common reference point to which all REFLO points from each channel are connected. The xREF points are REFHI points and can be of different frequencies, depending on the modules. Therefore the reference sources must be connected to REFGND and AREF, BREF, etc. The reference voltage and frequency must be accurate to within  $\pm 20\%$  of voltage to maintain the quoted accuracies of the converters. A solid state reference oscillator can be used. The ordering code is PC42.

- b) **xS1, xS2** are the inputs to the converters. These pins are connected to the LVDT. S1, and S2 of the LVDT go to SL, and SH of the converter. The LVDT reference connections are RL to xREF (REFHI) and RH to REFGND (REFLO).
- c) **xVEL** are the velocity outputs from the channels. The converter generates an analogue voltage proportional to the angular velocity of the shaft. The velocity output has a full scale voltage of  $\pm 10V$ . An external device can be connected to xVEL and VELGND.

## 2.4 Timing Requirements.

Up to a maximum of 2.050 us after the converter busy line has gone high, the data is available.

The converter ignores an inhibit signal if it is applied during an increment command.



The PC41 uses high-speed components, thereby allowing it to be addressed with zero wait states.

### 3.0 SOFTWARE.

#### 3.1 Programming the PC41.

##### 3.1.1 Address Allocation:

The PC41 uses 20 consecutive address locations in the I/O space:

ADDRESS ALLOCATION TABLE	
ADDRESS OFFSET FROM BASE	FUNCTION
OFFSET = 0	CHANNEL 1 LOW BYTE
OFFSET = 1	CHANNEL 1 HIGH BYTE
OFFSET = 2	CHANNEL 2 LOW BYTE
OFFSET = 3	CHANNEL 2 HIGH BYTE
OFFSET = 4	CHANNEL 3 LOW BYTE
OFFSET = 5	CHANNEL 3 HIGH BYTE
OFFSET = 6	CHANNEL 4 LOW BYTE
OFFSET = 7	CHANNEL 4 HIGH BYTE
OFFSET = 8	CHANNEL 5 LOW BYTE
OFFSET = 9	CHANNEL 5 HIGH BYTE
OFFSET = 10	CHANNEL 6 LOW BYTE
OFFSET = 11	CHANNEL 6 HIGH BYTE
OFFSET = 12	CHANNEL 7 LOW BYTE
OFFSET = 13	CHANNEL 7 HIGH BYTE
OFFSET = 14	CHANNEL 8 LOW BYTE
OFFSET = 15	CHANNEL 8 HIGH BYTE
OFFSET = 16	CHANNEL INHIBIT
OFFSET = 17	OVER-RANGE STATUS
OFFSET = 18	BUSY STATUS
OFFSET = 19	CLEAR BUSY STATUS

##### 3.1.2 Register Functions.

OFFSET 0 TO 15 - POSITION DATA - READ ONLY
--

Note that the data is right justified and hence the lower 8 bits are read from the LOW BYTE LATCH, whilst the upper 4 bits are read from the HIGH BYTE LATCH. See bit weights in the latter bit weight table.

OFFSET 16 - CHANNEL INHIBIT REGISTER - WRITE ONLY								
BIT:	D7	D6	D5	D4	D3	D2	D1	D0
CHAN. NUMBER:	8	7	6	5	4	3	2	1
BIT SET = INHIBIT BIT CLEAR = ENABLE								

OFFSET 17 - OVER-RANGE STATUS REGISTER - READ ONLY								
BIT:	D7	D6	D5	D4	D3	D2	D1	D0
CHAN. NUMBER:	8	7	6	5	4	3	2	1
BIT SET = OVER-RANGE BIT CLEAR = OK								

OFFSET 18 - CHANNEL BUSY STATUS - READ ONLY								
BIT:	D7	D6	D5	D4	D3	D2	D1	D0
CHAN. NUMBER:	8	7	6	5	4	3	2	1
BIT SET = BUSY								

OFFSET 19 - CLEAR CHANNEL BUSY STATUS - WRITE ONLY								
WRITE A '0' TO THE RELEVANT LOCATION								

### 3.1.3 Data Format.

BIT WEIGHTS			
BIT		SHAFT POSITION (*)	
MSB	D11	0.500000000	] high byte
	D10	0.250000000	
	D9	0.125000000	
	D8	0.062500000	
	D7	0.031250000	] low byte
	D6	0.015625000	
	D5	0.007812500	
D4	0.003906250		
D3	0.001953130		
D2	0.000976563		
LSB	D1	0.000488281	

\* in fractions of a pitch.

eg. An angle of 224 degrees would be 101000000000 binary  
or 2800 Hex.

## 3.2 Software Drivers and Demonstration Software.

### 3.2.1 Operation.

There are two modes of operation:

- a) **Synchronous mode** where the data is read randomly.
- b) **Asynchronous mode** where the data is read after each  
LSB shaft angle change.

- a) For **synchronous mode**, the following algorithm is required:

INITIALIZE: SET INHIBIT = 0.

STEP 1: SET INHIBIT = 1.

STEP 2: WAIT MINIMUM OF 2us.

STEP 3: READ HIGH BYTE OF WHATEVER CHANNEL REQUIRED.

STEP 4: READ LOW BYTE OF WHATEVER CHANNEL REQUIRED.

STEP 5: SET INHIBIT = 0.

- b) For **synchronous mode**, the following algorithm is required:

INITIALIZE: SET INHIBIT = 0.

STEP 1: WAIT FOR INTERRUPT OR POLL BUSY STATUS REGISTER FOR CONVERTER BUSY.

STEP 2: CHECK BUSY STATUS REGISTER TO DETERMINE WHICH CHANNEL IS READY.

STEP 3: SET INHIBIT = 1.

STEP 4: READ HIGH BYTE OF WHATEVER CHANNEL REQUIRED.

STEP 5: READ LOW BYTE OF WHATEVER CHANNEL REQUIRED.

STEP 6: SET INHIBIT = 0.

NOTE: THE ORDER OF BYTE RETRIEVAL IS IRRELEVANT IE. HIGH BYTE / LOW BYTE RETRIEVAL IS UNIMPORTANT.

If all 8 channels are used, the above routines can be modified by merely looping the algorithms and reading the relevant channel on each iteration of the loop.

If fewer channels are used, the INHIBIT register can be set up to disable the unused channels.

If interrupt mode is used, an interrupt jumper must be inserted in one of four positions to set up INT2, INT3, INT4 or INT5.

Note that INT2 should only be used on PC XT systems.

The normal interrupt channel allocation is:

JUMPER	PC XT		PC AT
W1	INT2	unused	int. controller 2
W2	INT3	unused	serial port 2
W3	INT4	serial port 1	serial port 1
W4	INT5	unused	parallel port 2

The following formula can be used to convert the two bytes of each channel to a position:

$$\text{position} = \frac{\text{HIGH BYTE} * 256 + \text{LOW BYTE}}{2048} * \text{pitch}$$

[ shaft position in units of pitch ]

Full software drivers are provided. The drivers are written in Borland C++ and Turbo Pascal. Fortran and Basic drivers will be available shortly. Also included, is full demonstration software, which can be used in real applications.

### 3.2.2 Software Drivers.

A number of driver routines are provided in Turbo Pascal, Borland C++, and Microsoft QuickC on the accompanying disc. These can be converted to other versions of C and Pascal by modifying the interrupt functions. The PC41 driver files are as follows:

PC41.c : the c drivers

PC41.pas : the general Pascal drivers  
and the Pascal interrupt drivers

#### The functions are:

set_base :	sets card base address.
set_num_chan :	sets number of channels available on the card.
clr_chan_array :	clears the channel array for initialization.
status :	returns the status of a single channel. Returns True if the channel latch has been updated.
over-range :	returns the True if an over-range condition occurs.
read_position :	returns the position of a single channel in mm.
process_CB :	returns the status of the busy register. Returns True if the channel latch has been updated. Also automatically updates the channel array each time it is called.

**I\_read\_angle :** returns the position of a single channel in integer format.

**I\_to\_R :** converts integer values to floating point values.

**init\_int :** initialises the interrupt routine. The interrupt routine automatically updates the channel array.

**restate\_old\_int :** reinstates the old interrupt handler.

**Channel array** is a global array and the data from the `process_CB` function and interrupt handler are stored in this array. Channels 0 to 7 are stored in sequence. The variable `num_chan` is also global and holds the number of channels to be processed.

**For C the function prototypes are:**

```
void set_base(unsigned int base_add);
    input parameters: base_add

void set_num_chan(int num);
    input parameters: num [1 to 8]

int over-range(int chan_num);
    input parameters: chan_num [0 to 7]

void clr_chan_array();
    input parameters: none

int status(int chan_num);
    input parameters: chan_num [0 to 7]
```

```
float read_position(int chan_num, float pitch);
```

```
    input parameters:  chan_num [0 to 7]
                       pitch
```

```
int process_CB();
```

```
    input parameters:  none
```

```
int I_read_angle(int chan_num);
```

```
    input parameters:  chan_num [0 to 7]
```

```
float I_to_R(int angle_in);
```

```
    input parameters:  angle_in
```

```
void init_int(int int_num);
```

```
    input parameters:  int_num [2, 3, 4, or 5]
```

```
void restate_old_int();
```

```
    input parameters:  none
```

**Global variables:**

```
int chan_array[8]
```

```
int num_chan
```

When using the drivers, the include file **PC41.h** must be incorporated in the main source file. Please do not forget to create project files.

The correct function prototypes and calls for Microsoft QuickC and Borland C++ must be set up. This is done by setting the constant:

```
#define lang QC for Microsoft QuickC
```

```
or #define lang BC for Borland C++.
```

The project is then compiled. Most versions of Microsoft C and Borland C will also work with these settings.

For Pascal the function prototypes are:

```
procedure set_base(base_add : word);
```

```
  input parameters:  base_add
```

```
procedure set_num_chan(num : integer);
```

```
  input parameters:  num [1 to 8]
```

```
function over-range(chan_num : integer) : boolean;
```

```
  input parameters:  chan_num [0 to 7]
```

```
procedure clr_chan_array;
```

```
  input parameters:  none
```

```
function status(chan_num : integer) : boolean;
```

```
  input parameters:  chan_num [0 to 7]
```

```
function read_position(chan_num : integer, pitch : real)  
                    : real;
```

```
  input parameters:  chan_num [0 to 7]  
                    pitch
```

```
function process_CB : boolean;
```

```
  input parameters:  none
```

```
function I_read_angle(chan_num : integer) : integer;
```

```
  input parameters:  chan_num [0 to 7]
```

```
float I_to_R(angle_in : integer);
```

```
  input parameters:  angle_in
```

```
procedure init_int(int_num : integer);
```

```
  input parameters:  int_num [2, 3, 4, or 5]
```

```
procedure restate_old_int;
```

```
  input parameters:  none
```

**Global variables:**

```
chan_array[0..7] of integer;  
num_chan : integer;
```

When using the drivers, the unit **PC41g** must be incorporated in the main source file. If the interrupt drivers are used, then the include file **PC41i** must also be incorporated in the main source file.

**Note I:** The **PC41i.pas** unit is written in Turbo Pascal and will have to be modified and recompiled for Microsoft Pascal.

**Note II:** If an invalid channel number is read with **read\_position**, -1 is returned.

**3.2.3 Sample Programs.**

Six sample programs are provided in Turbo Pascal and Borland C++. These can be converted to other versions of C and Pascal by modifying the **clrscr** functions, **gotoxy** functions, and **delay** functions to equivalents. The 'exe' files are also provided.

Programs for single speed systems:

```
PC41eg1.c and PC41eg1.pas : synchronous system  
PC41eg2.c and PC41eg2.pas : asynchronous system  
PC41eg3.c and PC41g3.pas  : interrupt driven
```

The 'exe' files for **PC41eg1** to **PC41eg3** are also provided.

**Program Description:**

**PC41eg1.X** reads all the channels synchronously and prints the positions on the screen. The **read\_position** function is used to read in the data.

**PC41eg2.X** reads all the channels asynchronously and prints the positions on the screen. The **process\_CB** function is used to poll the busy status register for a latch update and automatically update the relevant locations in the channel array. The **I\_to\_R** function is used in the main program to convert the integer data from the channel array to floating point position values.

**PC41eg3.X** reads all the channels asynchronously using interrupts and prints the positions on the screen. The interrupt handler is used to update the channel array transparent to the main program. The **I\_to\_R** function is used in the main program to convert the integer data from the channel array to floating point position values.

**Note I:** Do not forget to set up the project files in C.

### 4.0 Two Wire System Setup.

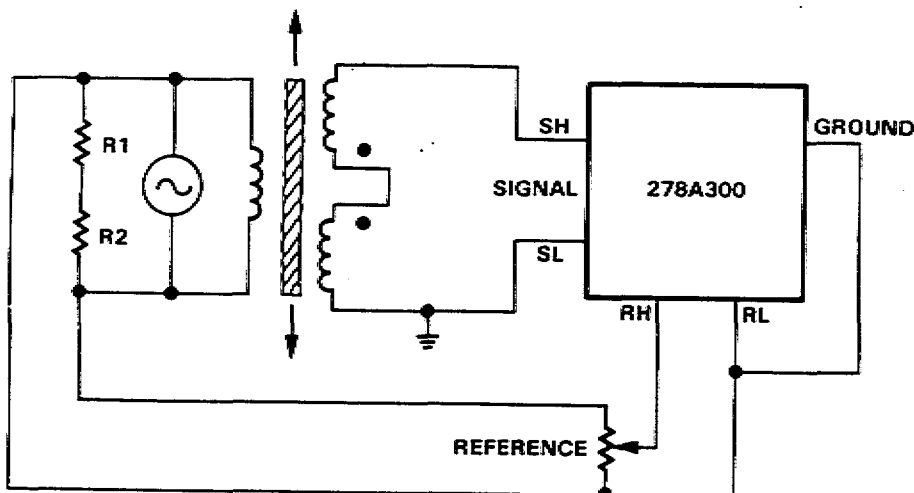
#### TWO-WIRE SYSTEM

The two-wire system of connection converts the widest range of LVDT sensors. It should be used in cases where the sum of the output voltages (SH + SL) is not constant with piston displacement. Where the A input is in phase with the RH input the digital output increases towards positive full scale.

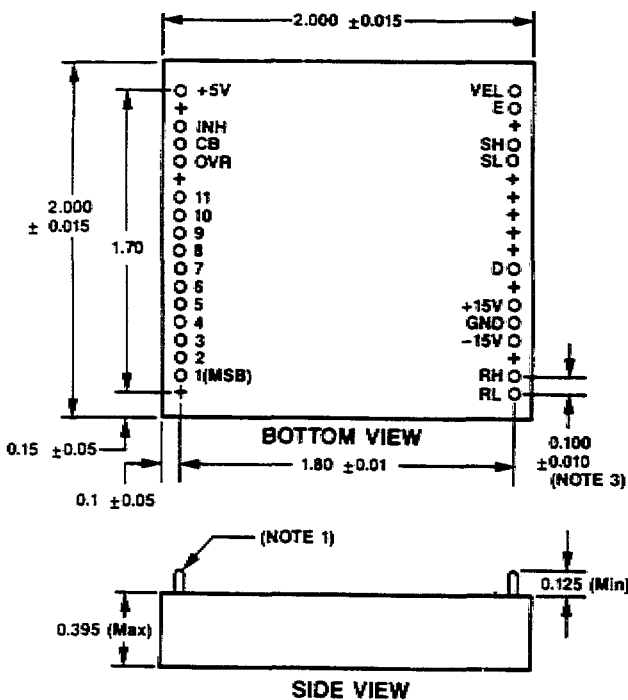
#### SCALE ADJUSTMENT

Scale adjustment can be accomplished by inserting a potentiometer in the reference circuit as shown. This can be used instead of the fine mechanical trimming of piston position in some applications. Typical pot value is 10 Kohms.

#### TWO-WIRE SYSTEM BLOCK DIAGRAM



#### MECHANICAL OUTLINE



#### ORDERING INFORMATION

278A Suffix	Operating Frequency	Logic Type
X00	400 Hz	LSTTL
X01	1 kHz-10 kHz	LSTTL
X02	400 Hz	HCMOS
X03	1 kHz-10 kHz	HCMOS

Standard temperature range (0° to 70°C).

#### NOTES

1. Rigid .025 diameter pins suitable for solder-in or plug-in applications.
2. Non-cumulative.
3. Dimensions are in inches.

**BIBLIOGRAPHY:**

- 1) **SYNCHRO CONVERSION : ILC DATA DEVICE CORPORATION,  
1990.**
- 2) **SYNCHRO AND RESOLVER CONVERSION : ANALOG DEVICES,  
1980.**

## APPENDIX A

## ELECTRICAL SPECIFICATIONS

Parameter	Value
Resolution	11 bits
Accuracy <sup>(1)</sup>	
0°C to 70°C	0.1% (full scale)
Linearity	±1/2 LSB
Reference Frequency	400 Hz - 10 kHz
Signal Inputs <sup>(2)</sup>	2.5V rms (RH-RL) (SH-SL)
Input Impedance	200 Kohms differential
Slew Rate	
400 Hz	200 LSB/ms
1 kHz - 10 kHz	400 LSB/ms
Settling Time (99% FS Step)	
400 Hz	50 ms
1 kHz - 10 kHz	25 ms
Acceleration Constant (K <sub>a</sub> )	
400 Hz	70,000 sec <sup>-2</sup>
1 kHz - 10 kHz	650,000 sec <sup>-2</sup>
Converter Busy(CB)	1.0 usec positive pulse
Over-Range (OVR)	Logic '1' = Over-range
Inhibit Input	Logic '0' inhibits 0.5 LSTTL load
Power Supplies <sup>(3)</sup>	
+15V @	25 mA max. (18 mA typ.)
-15V @	25 mA max. (18 mA typ.)
+ 5V @	100 mA max. (80 mA typ.) LSTTL 15 mA max. ( 5 mA typ.)-HCMOS
Temperature Ranges	
Operating	0°C to 70°C
Storage	-55° to +125°C
Dimensions	2.00" X 2.00" X 0.395"
Weight	1.25 oz

## NOTES:

- Accuracy applies for:
  - ±20% signal voltage variation
  - over operating temperature range
  - over operating frequency range
  - not greater than 3° phase error between reference and signal inputs
- This is a nominal value.
- All units can operate on voltages between ±12V to ±17V. The tolerance on the 5 Vdc supply is +4.75 to +5.25V.

## DIGITAL OUTPUT CODES

	MSB	LSB	PHASE
+FULL SCALE	1	1	IN
ZERO	1	0	0
-FULL SCALE	0	0	OUT

Irrespective of the method of input connection the output codes will be offset binary.

## LOGIC INPUTS/OUTPUTS

This converter series is available with HCMOS or LSTTL logic with load and drive capabilities as specified under Electrical Specifications. The Converter Busy (CB) output is a positive 1.0 microsecond positive pulse which brackets the output code change to indicate output update. The inhibit (INH) input locks the internal up-down counter, thus preventing the converter from tracking. Logic "0" or ground inhibits, logic "1" or open allows tracking. Application of extremely long inhibit times can cause erratic operation.

## TIMING

Whenever an input angle change occurs, the converter changes the digital angle in steps of 1 LSB and generates a Converter Busy pulse. During the Converter Busy time the output data is changing and should not be transferred. The converter will ignore an inhibit command applied during the Converter Busy interval. There are two methods of interfacing with a computer: (1) synchronous and (2) asynchronous. A simple method of synchronous loading is to: (a) apply the inhibit, (b) wait 2 microseconds, (c) transfer the data, and (d) release the inhibit. Asynchronous loading is accomplished by transferring data on the trailing edge of the CB pulse.

## ACCELERATION ERROR

A tracking converter like a 278A300 employing a "Type II" servo loop does not suffer any velocity lag, however, there is an additional error due to acceleration. This additional error can be defined using the acceleration constant K<sub>a</sub> of the converter.

$$K_a = \frac{\text{Input acceleration}}{\text{Error in output angle}}$$

The numerator and denominator have the same units. K<sub>a</sub> does not define maximum acceleration — only the error due to acceleration. Maximum acceleration is in the region of 5 times the K<sub>a</sub> figure (deg/sec<sup>2</sup>).

## VELOCITY VOLTAGE

The Velocity output (VEL) is a dc voltage proportional to the angular velocity of the synchro or resolver shaft. Voltage polarity is positive for an increasing angle. This output can be used in many applications to provide loop stabilization and velocity feedback data. Scaling is ±10 Vdc for maximum velocity.

## INTER LSB VOLTAGE

The Inter LSB (E) voltage is a dc analog voltage representing the synchro or resolver shaft position within the least significant bit of the digital angle output. Polarity is negative for increasing angle. Scaling is ±4.5 V/LSB.

## DC ERROR VOLTAGE

The DC Error Voltage (D) is the signal at the output of the phase sensitive demodulator and is proportional to the error between the analog input angle and digital output angle. This is an unfiltered output and will increase if the output angle fails to track the input for any reason. Polarity is negative for increasing angle. Scaling is 15 mv/LSB.