COORDINATE TRANSFORMATION
A POINT STORAGE CONVERSION
for the HP-41CV/CX SURVEYING PAC

5. Keystroke

[ ] [RCL] [.] [0] [8] [8]

to go to step 88.

Delete steps 88 (DSE 13) through 77 (1.1), and then type in new steps as listed to the right.

(5) delete these steps

insert these

```
74 GTO 02
75♦LBL 10
76 SF 10
77 1.1
78 STO 13
79 XROM "NE"
80 STO 01
81 RDN
82 STO 00
83 ISG 13
84 XROM "NE"
85 STO 05
86 RDN
87 STO 04
88 DSE 13
89 RTN
90♦LBL 00
91 RCL 05
```

```
ADV
"BEG. PT. NO.?"
PROMPT
XEQ "POUT"
STO 01
RDN
STO 00
CF 10
R↑
XROM "NE"
AVIEW
"END PT. NO.?"
PROMPT
STO 16
XEQ "POUT"
STO 05
RDN
STO 04
"SAVE AS?"
PROMPT
STO 15
1.1
STO 13
SF 10
```

```
71 RCL 03
72 X<> 12
73 STO 03
74 GTO 02
75♦LBL 10
```

XEQ 01

(6) insert

6. [ ] [RCL] [.] [0] [7] [3]

to go to step 73 (STO 03) and insert XEQ 01.

[ ] [RCL] [.] [0] [0] [1] and single step through the program to check it. A complete listing of the program steps is shown on the next page.

Software by D'Zign
P. O. BOX 1370 • PACIFICA, CA 94044

COORDINATE TRANSFORMATION
A POINT STORAGE CONVERSION
for the HP-41CV/CX SURVEYING PAC

published by D'Zign **Land Survey & Development**
Pacifica, California 1987

ISBN 0-9616846-7-4

Introduction

A number of surveyors have requested programs which let them store coordinates by point number, without having to buy a module. Since most surveyors already have the HP-41 Surveying Pac, they are reluctant to spent three or four hundred dollars for a second module which, essentially, does the same work as the HP Surveying Pac.

This series of booklets was produced as a low-cost way of getting more out of the module which you already have.

Each booklet contains complete instructions for **editing** a routine in your Surveying Pac to use the point number storage. The point storage program is the same for all of the conversions and, by itself, provides a complete system for radial inversing between coordinates for 'spraying' in points in the field.

This booklet converts the Coordinate Transformation routine in the Surveying Pac to work with pre-stored point numbers. It may be used independently, or in combination with the converted Traverse, Inverse & Sideshots routine (D'Zign Booklet #521). If the latter converted program is already in memory, there is no need to re-enter the utility program "**XX**" with this program.

The conversion in this booklet is compatible with the converted Intersections Solutions (Booklet #523) and the converted Predetermined Areas program (Booklet #531). If you have a CX, the intersection and transformation programs may be stored into extended memory until needed, used, and then deleted again from main program memory.

As an added bonus, this conversion does away with the need to inverse between the rotated coordinates with the Traverse routine after transformation. It contains an AUTO-INVERSE routine that lets you output the new bearings and distances between the transformed coordinates just by setting a flag.

This booklet has been published in two versions, one with and one without pre-programmed magnetic cards. If your version contains the cards, you will still find the editing lists helpful, and can further modify the programs to suit your particular needs.

Utility Programs

There is a short set of utility programs that acts as the 'driver' for the point storage system. The main routines are "**PIN**" (short for "**P**oint **IN**"), "**POUT**" (**P**oint **OUT**), and one called "**XX**" (no particular reason). After these have been entered into program memory we will assign them to the keyboard for convenient use.

Start by executing a **GTO··**, and enter program mode, by keystroking [ ] RCL · · PRGM. The display should show **00 REG**, and a number, **NNN**, which is the number of available registers in memory. You will need at least 21 registers in order to type in these UTILITY programs.

If your version of this book includes the program cards, see Appendix A for input instructions. If not, type in the program steps listed below:

```
01♦LBL "XX"
02 XROM "*IN"
03 XEQ "C*"
04 CF 10
05 XEQ "POUT"
06 XROM "NE"
07 AVIEW
08 ADV
09 STO 08
10 RDN
11 STO 07
12♦LBL 03
13 CF 01
14 "TO?"
15 PROMPT
16 XEQ "POUT"
17 DSE 13
18 STO X
19 XROM "INVERSE"
20 GTO 03
21 RTN
22♦LBL "POUT"
23 STO 13
24♦LBL "OUT"
25 RCL 13
26 2
27 *
28 19
29 +
30 ENTER↑
31 ENTER↑
32 1
33 -
34 RCL IND X
35 RCL IND Z
36 RTN
37♦LBL "PIN"
38 SF 10
39 STO 13
40 XROM "NE"
41♦LBL "IN"
42 RCL 13
43 2
44 *
45 19
46 +
47 X<>Y
48 STO IND Y
49 RDN
50 1
51 -
52 X<>Y
53 STO IND Y
54 RTN
*55♦LBL "C*"
56 .019
57 0
58♦LBL 11
59 STO IND Y
60 ISG Y
61 GTO 11
62 RDN
63 RDN
64 RTN
65 END
```

* If you are using a 41CX, you can save a few program steps (and registers) by substituting the steps shown to the right for the program steps from 55 through 65.

```
55♦LBL "C*"
56 .019
57 CLRGX
58 RDN
59 RTN
```

If you are not fully familiar with programming on the HP41 calculators, don't panic; all of the program steps are normal ones, and you shouldn't have any trouble. To put in an XROM, such as are shown at steps 02, 06, 19 and 40, you input the step as "XEQ".

The calculator will change the command to "XROM" for you, as long as the Surveying Pac is in the calculator. For instance, the keystrokes for input of step 06 are [XEQ] [ALPHA] [N] [E] [ALPHA].

Any portion of a program step which is listed in quotation marks (" ") indicates that it is **alpha** input, and must be input with the calculator in **alpha mode**.

Program steps which are **indirect** store or recall instructions are input by stroking the shift key after the STO or RCL key. For example, program step 34 would be input by keystroking [RCL] [ ] [·] [6].

Step 49, RDN is "roll down", sometimes mistaken for "round" which is printed as "RND". The ♦ symbol at each label is inserted by the printer when it lists the programs ... you don't need to input it. Finish typing in the program.

Once that chore is completed, we will assign the program "**PIN**" to the shifted STO key, and try storing a coordinate pair from the keyboard. Size your calculator to **040** by stroking [XEQ] [ALPHA] [S] [I] [Z] [E] [ALPHA], and, when the display shows the prompt **SIZE___**, stroke [0] [4] [0].

Keystroke [ ] [XEQ] [ALPHA] [P] [I] [N] [ALPHA] [ ] [STO], to assign "**PIN**". The keystroke instructions are simple: Input the point number, stroke [ ] [STO], and the display will prompt for the North coordinate. Input the coordinate and stroke [R/S], and the display will prompt for the East coordinate. Input the coordinate, stroke [R/S]. Now the coordinates are stored.

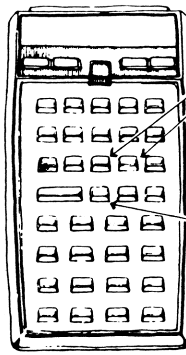
Practice this by inputting the coordinates of the traverse on the next page. A step by step example is shown, so just follow the procedures.

User instructions

First, let's assign the "**XX**" program to the shifted CHS (-42) key, by stroking

[ ] XEQ ALPHA X X ALPHA [ ] CHS

We can also assign the program "**POUT**" to the shifted RCL button at this time, and our assigned keys will be as shown to the right.



When **FLAG 00** is clear, output will be **azimuth**. When set, the output will be shown as a **bearing**.

The keystroke instructions for **RADIAL INVERSING** are as follows:

1. Put the calculator into USER mode.

2. Input the beginning (setup) point number, and stroke [ ] CHS

output: **N**x=**XXXXX.xxxx**

3. If a printer is attached, the output is automatic. If no printer, stroke R/S

output: Ex=**XXXXX.xxxx**

R/S if no printer

prompt: **TO?**

4. Input the point number of the point you are inversing to, and stroke R/S

output: **AZ=DDD.mmss** (bearing if flag 00 is set)

R/S if no printer

HD=XXX.xxx

R/S if no printer

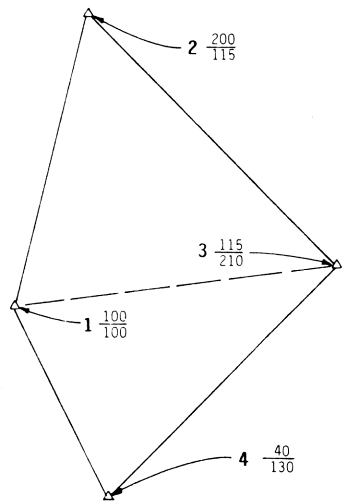
Ny=**XXXXX.xxxx**

R/S if no printer

Ey=**XXXXX.xxxx**

R/S if no printer

To illustrate the keystrokes for storing coordinate pairs, we'll use the little traverse shown below, and input the coordinates of the angle points. The keystrokes shown assume that "**PIN**" is assigned to the shifted STO key, as shown on the previous page.



keystroke:

[1] [ ] [STO]

prompt: **N1=?**

keystrokes:

[1] [0] [0] [R/S]

prompt: **E1=?**

keystrokes:

[1] [0] [0] [R/S]

we've stored point #1

keystrokes:

[2] [ ] [STO]

prompt: **N2=?**

keystrokes:

[2] [0] [0] [R/S]

prompt: **E2=?**

keystrokes:

[1] [1] [5] [R/S]

we've stored point #2

keystrokes:

[3] [ ] [STO]

prompt: **N3=?**

keystrokes:

[1] [1] [5] [R/S]

prompt: **E3=?**

keystrokes:

[2] [1] [0] [R/S]

that stored point #3

keystrokes:

[4] [ ] [STO]

prompt: **N4=?**

keystrokes:

[4] [0] [R/S]

prompt: **E4=?**

keystrokes:

[1] [3] [0] [R/S]

We now have all of the points in storage, and we can use these same points as examples of how to inverse with "**XX**".

keystroke examples

To give you an example of the keystrokes, let's assume that point #1 is the setup point, and do radial inverses to the other points.

These outputs are all "north" azimuths, so if you set the azimuth of the backsight in the instrument before backsighting, 0°00'00" would have you looking "north".

This works whether the coordinate grid is set up on 'true' north, 'project' north, or 'assumed' north. When you set the azimuth to the next point, it is the same as turning an 'angle right'.

This means that, with pre-stored job coordinates, you have a really fast way to calculate "spray" ties in the field. And you can pick the setup and backsight points when you get to the job, no matter what the job conditions are.

Clear flag 00 before you start, and then follow these procedures:

keystrokes:

1 [ ] CHS

output:

* N1=100.0000
* E1=100.0000

prompt: **TO?**

keystrokes:

2 R/S

output:

* AZ=8.3151
* HD=101.1187

* N2=200.0000
* E2=115.0000

prompt: **TO?**

keystrokes:

3 R/S

output:

* AZ=82.1405
* HD=111.0180

* N3=115.0000
* E3=210.0000

prompt: **TO?**

keystrokes:

4 R/S

output:

* AZ=153.2606
* HD=67.0820

* N4=40.0000
* E4=130.0000

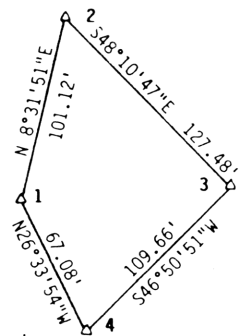
prompt: **TO?**

* R/S if no printer

If you are using the program with a printer attached, the output is automatic. If no printer is used, continue stroking R/S and writing down the answers until the prompt **TO?** appears. Then input the next point number.

We'll use the same stored coordinates to do a keystroke example of an **INVERSE TRAVERSE**, using "**XX**". The only difference between this example and the last is that we set FLAG 01 each time, prior to entering the point number.

We will also set FLAG 00 before we start, to have **bearing** output instead of azimuth.



keystrokes:

1 CHS

output:

N1=100.0000
E1=100.0000

* prompt: **TO?**

keystrokes:

7 0 0 0

7 0 1 2 R/S

output:

N 8.3151 E
HD=101.1187

N2=200.0000
E2=115.0000

* prompt: **TO?**

keystrokes:

7 0 1 3 R/S

output:

S 48.1047 E
HD=127.4755

N3=115.0000
E3=210.0000

* prompt: **TO?**

*An additional R/S will be needed if used without printer

keystrokes:

7 0 1 4 R/S

output:

S 46.5051 W
HD=109.6586

N4=40.0000
E4=130.0000

* prompt: **TO?**

keystrokes:

7 0 1 1 R/S

output:

N 26.3354 W
HD=67.0820

N1=100.0000
E1=100.0000

* prompt: **TO?**

To use "**POUT**", simply input the point number, and stroke RCL. This recalls the point's coordinates to the x and y registers. The Easting will be displayed, and you may verify that the northing is in the y-register by stroking either X⇄Y or R↓.

Editing the Surveying Pac

We are going to "edit" the program "**COORD**" which is in the HP Surveying Pac so that it will work with our stored coordinates, instead of requiring that the coordinates be input. Because the programs in the module cannot be altered, we do the next best thing by having a modified version of the same program in our program memory.

First, make sure that you are NOT in **user mode**, because two of the keys which we will be using are assigned keys. Next do a **GTO··** and shift into **program mode** by stroking the PRGM key. You will need at least 55 available registers, so verify that you have enough registers to do the editing, then stroke PRGM again.

Stroke XEQ ALPHA C O P Y ALPHA , and at the prompt COPY_, stroke

ALPHA C O O R D ALPHA

Stroke PRGM , and you should see **01 LBL⊤COORD** in the display. We will make our first change to the program by typing in the following:

STO ALPHA C T ALPHA SST ← SST

This has given the program a new name, and deleted the old one. From now on, when we want to call up this program we will call it "**CT**".

Now, follow these procedures in the order shown:

1. Stroke RCL · 1 2 8 , to go to program step 128, and type in these additional steps:

```
129 RTN
130♦LBL 07
131 FIX 0
132 ARCL 13
133 "├ "
134 RTN
135♦LBL 09
136 XEQ "OUT"
137 RCL 08
138 -
139 X<>Y
140 RCL 07
141 -
142 R-P
143 "HD="
144 ARCL X
145 AVIEW
146 X<>Y
147 X<0?
148 XEQ 05
149 HMS
150 XROM "BRG"
151 AVIEW
152 ADV
153 RTN
154♦LBL 11
155 R↑
156 CF 10
157 XROM "NE"
158 AVIEW
159 ADV
160 RTN
```

2. Keystroke

[ ] [RCL] [·] [1] [2] [2]

to go to step 122, and type in the steps shown to the right. Stroke [ ] [SST] 4 times to backstep to program step 120 (STO 03), and type in STO 07.

② insert

```
123 STO 08
124 XEQ 11
```

here

```
120 STO 03
121 RDN
122 STO 02
123 RTN
124♦LBL 05
```

and insert 121 STO 07

```
120 STO 03
121 RDN
122 STO 02
```

here

③ delete these steps

```
117 "1 NEW"
118 ASTO 13
```

replace them with these

```
117 "NEW COORDS"
118 XROM "*YH"
119 CLX
120 FS? 10
121 XEQ 08
122 RTN
123♦LBL 08
124 "NEW"
125 AVIEW
126 RCL 13
127 SF 10
```

3. Backstep three times to step 118 (ASTO 13) and delete steps 118 & 117. The display should now read **116 LBL 04.**

Type in the new steps shown to the left.

As a check, stroking [SST] should give a display of **128 XROM "NE".**

4. Stroke

[ ] [RCL] [·] [1] [1] [4]

to go to step 114 (STO 00) and add the steps shown (right).

Backstep to step 112 (STO 01), and insert **STO 07, STO 03.**

④ insert

```
115 STO 08
116 STO 02
117 XEQ 11
```

```
112 STO 01
113 RDN
114 STO 00
115 RTN
116♦LBL 04
```

backstep and insert

```
113 STO 07
114 STO 03
```

```
111
112 STO 01
113 STO
```

Editing the Surveying Pac

5. Backstep to step 111 (XROM "NE"), and delete thru 109 ("1 OLD"), replacing them with the steps shown to the right.

(5) delete these

```
109 "1 OLD"
110 ASTO 13
111 XROM "NE"
```

add these

```
109 "ROT. PT.?"
110 PROMPT
111 XEQ "POUT"
```

6. Go to program step 106 (ADV), and delete all of the steps thru 99 (CLA).

Type in the new program steps shown to the right.

If you stroke SST after typing in the new steps, you should see **111 STOP** in the display.

(6) delete these

```
97 RCL 01
98 +
99 CLA
100 ARCL 11
101♦LBL 06
102 ASTO 13
103 CF 10
104 XROM "NE"
105 AVIEW
106 ADV
107 STOP
```

replace with these

```
99 CF 10
100 XROM "NE"
101 AVIEW
102 ADV
103 RTN
104♦LBL 06
105 RCL 16
106 STO 13
107 RDN
108 XEQ "IN"
109 "STORED"
110 AVIEW
```

(7) insert

```
81 "OLD"
82 AVIEW
83 STO 16
84 XEQ "POUT"
```

```
79 GTO 06
80♦LBL B
81 RCL 03
```

delete

```
77 CLA
78 ARCL 12
79 GTO 06
80♦LBL B
```

add these

```
78 XEQ "IN"
79 FS? 09
80 XEQ 09
81 XEQ "OUT"
82 STO 08
83 X<>Y
84 STO 07
85 X<>Y
86 CF 10
87 "NEW"
88 AVIEW
89 XROM "NE"
90 AVIEW
91 ADV
92 STOP
```

7. Keystroke [shift] RCL · 0 8 0 to go to step 080 (LBL B). Type in **STO 16, XEQ "POUT"**.

Backstep to 79, and delete steps 79 (GTO 06) and 78 (ARCL 12). Type in the new steps shown to the left.

8. Keystroke [ ] [RCL] [·] [0] [5] [8] to go to step 058, and type in **STO 16, XEQ "POUT".**

Backstep to 056 and delete program steps 56 (ASTO 12) thru 52 (LBL 01).

```
57 STOP
58♦LBL A
59 RCL 01
60 -
```

⑧ **insert**

```
59 STO 16
60 XEQ "POUT"
```

backstep, then delete

```
52♦LBL 01
53 " OLD"
54 ASTO 11
55 " NEW"
56 ASTO 12
```

⑨ **delete**

```
36 RCL 03
37 STO 04
38 XEQ 04
39 "2 NEW"
40 ASTO 13
41 XROM "NE"
42 RCL 03
43 -
```

type in

```
38 "NEW "
39 XEQ 07
40 "├N↑E"
41 FIX 4
42 PROMPT
```

9. [ ] [RCL] [·] [0] [4] [1] and delete steps 41 (XROM "NE") thru 38 (XEQ 04), replacing them with the steps shown to the left.

10. Keystroke [ ] [RCL] [·] [0] [2] [6] Delete steps 26 (XROM "NE") thru 24 ("2 OLD"), and type in the steps shown to the right.

Backstep to program step 021 (GTO 01), delete it and replace it with **STOP.**

```
22 ...
23 XEQ 03
24 "2 OLD"
25 ASTO 13
26 XROM "NE"
27 RCL 01
```

⑩ **delete**

type in

```
24 XEQ 04
25 "2ND PT?"
26 PROMPT
27 XEQ "POUT"
```

replace

```
19 RCL 03
20 XEQ 04
21 STOP ── ~~21 GTO 01~~
22♦LBL 02
```

11. Stroke [ ] [RCL] [·] [0] [0] [3] to go to program step 003, and insert **XEQ "C*".**

⑪ **insert**

```
02♦LBL E
03 XROM "*IN"
04 XEQ "C*"
04 SF 10
05 1
```

12. Stroke [ ] [RCL] [·] [0] [0] [1] to go to the beginning of the program, and use [SST] to check your program against the listing on page 13.

```
01♦LBL "CT"
02♦LBL E
03 XROM "*IN"
04 XEQ "C*"
05 SF 10
06 1
07 STO 05
08 CF 22
09 "ROT. ∡=?"
10 PROMPT
11 FC? 22
12 GTO 02
13 HR
14 STO 04
15 CF 22
16 "SCALE FACT.=?"
17 PROMPT
18 FS? 22
19 STO 05
20 XEQ 03
21 XEQ 04
22 STOP
23♦LBL 02
24 XEQ 03
25 XEQ 04
26 "2ND PT?"
27 PROMPT
28 XEQ "POUT"
29 RCL 01
30 -
31 X<>Y
32 RCL 00
33 -
34 R-P
35 1/X
36 STO 05
37 X<>Y
38 XEQ 05
39 STO 04
40 "NEW "
41 XEQ 07
42 "├N↑E"
43 FIX 4
44 PROMPT
45 RCL 03
46 -
47 X<>Y
48 RCL 02
49 -
50 R-P
51 ST* 05
52 X<>Y
53 XEQ 05
54 ST- 04
55 STOP
56♦LBL A
57 STO 16
58 XEQ "POUT"
59 RCL 01
60 -
61 X<>Y
62 RCL 00
63 -
64 R-P
65 RCL 05
66 *
67 X<>Y
68 RCL 04
69 -
70 X<>Y
71 P-R
72 RCL 02
73 +
74 X<>Y
75 RCL 03
76 +
77 CLA
78 XEQ "IN"
79 FS? 09
80 XEQ 09
81 XEQ "OUT"
82 STO 08
83 X<>Y
84 STO 07
85 X<>Y
86 CF 10
87 "NEW"
88 AVIEW
89 XROM "NE"
90 AVIEW
91 ADV
92 STOP
93♦LBL B
94 "OLD"
95 AVIEW
96 STO 16
97 XEQ "POUT"
98 RCL 03
99 -
100 X<>Y
101 RCL 02
102 -
103 R-P
104 RCL 05
105 /
106 X<>Y
107 RCL 04
108 +
109 X<>Y
110 P-R
111 RCL 00
112 +
113 X<>Y
114 RCL 01
115 +
116 CF 10
117 XROM "NE"
118 AVIEW
119 ADV
120 RTN
121♦LBL 06
122 RCL 16
123 STO 13
124 RDN
125 XEQ "IN"
126 "STORED"
127 AVIEW
128 STOP
129♦LBL 03
130 "ROT. PT.?"
131 PROMPT
132 XEQ "POUT"
133 STO 01
134 STO 07
135 STO 03
136 RDN
137 STO 00
138 STO 08
139 STO 02
140 XEQ 11
141 RTN
142♦LBL 04
143 "NEW COORDS"
144 XROM "*YN"
145 CLX
146 FS? 10
147 XEQ 08
148 RTN
149♦LBL 08
150 "NEW"
151 AVIEW
152 RCL 13
153 SF 10
154 XROM "NE"
155 STO 03
156 STO 07
157 RDN
158 STO 02
159 STO 08
160 XEQ 11
161 RTN
162♦LBL 05
163 X>0?
164 RTN
165 360
166 +
167 RTN
168♦LBL 07
169 FIX 0
170 ARCL 13
171 "├ "
172 RTN
173♦LBL 09
174 XEQ "OUT"
175 RCL 08
176 -
177 X<>Y
178 RCL 07
179 -
180 R-P
181 "HD="
182 ARCL X
183 AVIEW
184 X<>Y
185 X<0?
186 XEQ 05
187 HMS
188 XROM "BRG"
189 AVIEW
190 ADV
191 RTN
192♦LBL 11
193 R↑
194 CF 10
195 XROM "NE"
196 AVIEW
197 ADV
198 RTN
199 END
```

With your program in the calculator, and proof-reading completed, you are probably anxious to try it out. The program functions the same as the HP Surveying Pac version, but with different prompts. Of course, the main differences are that this version **stores** the coordinates, and can also inverse between them when you want it to.

Let's cover that aspect first. If you want to inverse as the new coordinates are calculated, set flag 09. If you do not need the new distances and bearings output, clear flag 09. The inverses, when output, are from the last point calculated to the new point.

The two types of setup input are the same as in the Surveying Pac; **if rotation angle is known**, and **if two points in each system are known.** If the rotation angle is entered, the option for changing the scale factor prompts for it.

As in the Hewlett Packard version, the rotation angle is entered as POSITIVE FOR COUNTERCLOCKWISE and negative for clockwise. That's backwards in surveying, left is negative, but we haven't changed it in this version because we want the program to work as closely as possible to the original, to avoid confusion.

The keystroke procedures for both types are shown below. Begin by calling up the program with XEQ ALPHA C T ALPHA. The first prompt, **ROT.∠=?** is displayed.

IF ROTATION ANGLE IS KNOWN:

1. Input the rotation angle, in Degrees, Minutes and Seconds (if CLOCKWISE, CHS) R/S

prompt: **SCALE FACT.**

2. If scale factor is not 1:1, input the new scale factor. If the factor is 1, it is not necessary to input anything R/S

prompt: **ROT. PT.?**

3. Input the point number of the pivot point R/S

output: Nx=XXXX.xxxx R/S if no printer

Ex=XXXX.xxxx R/S if no printer

prompt: **NEW COORDS?**

4\. Stroke N R/S if the coordinates of the pivot point are the same in the new system, **OR** stroke Y R/S if the coordinates of the pivot point are different in the new system. If the answer is YES, the additional prompts (marked**) will appear.

Prompt: **Nx=?

Input the pivot point north-coordinate in the NEW system R/S

prompt: **Ex=?

Input the east-coordinate of the pivot point in the NEW system R/S

All of the parameters for rotating or transforming the other coordinates are now stored, and the program will halt to await input of the first point number (see **SOLUTIONS**). At this point, if you want to inverse between the new points, set FLAG 09 before continuing.

IF TWO POINTS IN EACH SYSTEM ARE KNOWN:

1\. When the prompt, **ROT.∡=?** is displayed, stroke R/S

prompt: **ROT.PT.?**

2\. Input the point number of the pivot point R/S

output: Nx=XXXX.xxxx R/S if no printer

Ex=XXXX.xxxx R/S if no printer

prompt: **NEW COORDS?**

3. Stroke **N** **R/S** if the coordinates of the pivot point are the same in the new system, **OR** stroke **Y** **R/S** if the coordinates of the pivot point are different in the new system. If the answer is YES, the additional prompts and output (marked**) will appear.

prompt: **Nx=?

Input the pivot point north-coordinate in the NEW system **R/S**

prompt: **Ex=?

Input the east-coordinate of the pivot point in the NEW system **R/S**

****output: Nx=XXXX.xxxx** **R/S** if no printer

Ex=XXXX.xxxx **R/S** if no printer

prompt: **2ND PT?**

4. Input the point number of the second point **R/S**

prompt: **NEW X. N↑E**

5. Input the north-coordinate of the second point, in the NEW system **ENTER↑**

Input the east-coordinate of the second point, in the NEW system

All of the parameters for rotating or transforming the other coordinates are now stored, and the program will halt to await input of the first point number (see **SOLUTIONS**, below). At this point, if you want to inverse between the new points, set FLAG 09 before continuing.

SOLUTIONS:

1. Input the point number of the first point which you wish to transform (NOTE: the point numbers input for setup **have not yet** been transformed)

2a. To transform coordinates **from old system to new system** **A**

***output: HD=XXX.xxxx** from last point to this point

R/S if no printer

***output:** BEARING from last point to this point

R/S if no printer

output: NEW

R/S if no printer

Nx=XXXX.xxxx

R/S if no printer

Ex=XXXX.xxxx

2b. To transform coordinates **from new system to old system** **B**

output: OLD

R/S if no printer

Nx=XXXX.xxxx

R/S if no printer

Ex=XXXX.xxxx

Note: These coordinates in the old system are calculated, but NOT stored. If you wish to store them as OLD system, stroke R/S. The calculator will output "**STORED**" to indicate that you have stored the coordinates under that point number in the old system.

3. Continue repeating steps 1 and 2 as desired.

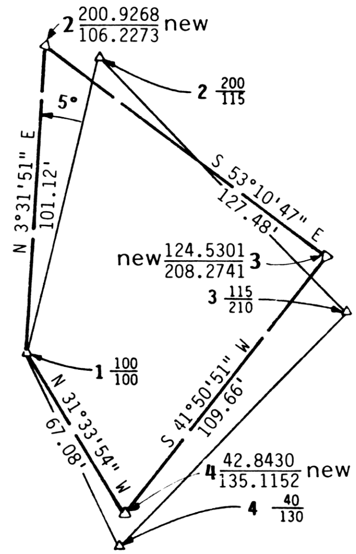
4. For a new case, stroke **E**

We can use the points that we have already stored (page 5) for the keystroke examples. If you have already cleared them, it only takes a minute to put them back in.

Before beginning, the calculator should be sized at least to 040, and the utility programs must be in program memory.

For the first example, we will assume that we want to rotate the traverse 5° to the left, using point number 1 as the pivot, and with the coordinates of the pivot point the same in both systems.

*Output when FLAG 09 is set.



We will begin with the keystrokes

XEQ **ALPHA** **C** **T** **ALPHA**

prompt: **ROT. ∡=?**

keystrokes:

5 **R/S**

prompt: **SCALE FACT.=?**

keystrokes:

R/S

prompt: **ROT PT?**

keystrokes:

1 **R/S**

output:

```
 N1=100.0000
*E1=100.0000
*
```

prompt: **NEW COORDS?**

keystrokes:

N **R/S**

7 **0** **9** **R/S**

display: **0.0000**

keystrokes:

2 **A**

output:

```
*HD=101.1187
*N 3.3151 E

*NEW
*N2=200.9268
 E2=106.2273
```

keystrokes:

3 **A**

output:

```
 HD=127.4755
*S 53.1047 E
*
 NEW
*N3=124.5301
*E3=208.2741
```

keystrokes:

4 **A**

output:

```
*HD=109.6586
*S 41.5051 W

*NEW
*N4=42.8430
 E4=135.1152
```

keystrokes:

1 **A**

output:

```
 HD=67.0820
*N 31.3354 W
*
 NEW
*N1=100.0000
*E1=100.0000
```

If, for some reason, we wanted to see what point number 3 coordinates were in the old system, we could do it with the keystrokes:

3 **B**

output:

```
 OLD
*N3=115.0000
*E3=210.0000
```

* **R/S** if no printer

Store the original coordinates again, for another example. Let's assume that we want to **move** the traverse 100' north, and 100' east. This can be done by using a 0° rotation angle and "NEW" coordinates of 200/200 for point number 1, with the same keystrokes we just practiced.

Another way to do it would be to use 200/200 for NEW #1, and 300/215 for NEW #2, then working the problem as **two points in each system known**. Clear FLAG 09.

keystrokes: XEQ ALPHA C T ALPHA

prompt: **ROT. ∡=?**

keystrokes: R/S

prompt: **ROT. PT.?**

keystrokes: 1 R/S

output:
* N1=100.0000
* E1=100.0000

prompt: **NEW COORDS?**

keystrokes: Y R/S

output:
* NEW

prompt: **N1=?**

keystrokes: 2 0 0 R/S

prompt: **E1=?**

keystrokes: 2 0 0 R/S

output:
* N1=200.0000
* E1=200.0000

prompt: **2ND PT?**

keystrokes: 2 R/S

prompt: **NEW 2. N↑E**

keystrokes: 3 0 0 ENTER↑ 2 1 5 R/S

The program will stop and wait for input of the first point number.

keystrokes: 2 A

output:
* NEW
* N2=300.0000
E2=215.0000

keystrokes: 3 A

output:
* NEW
* N3=215.0000
E3=310.0000

keystrokes: 4 A

output:
* NEW
* N4=140.0000
E4=230.0000

keystrokes: 1 A

output:
* NEW
* N1=200.0000
E1=200.0000

* R/S if no printer

Appendix A

If your version of this booklet includes the programmed magnetic cards, cards A and B contain the "**CT**", and card C contains the UTILITY program, "**XX**".

"**CT**" contains 375 bytes, and will occupy 54 registers. If the converted TRAVERSE, INVERSE & SIDESHOT (booklet #521) is in program memory, it is not necessary to enter the UTILITY programs.

	R19=	0.0000
N1	R20=	100.0000
E1	R21=	100.0000
N2	R22=	215.8520
E2	R23=	155.2698
N3	R24=	64.8508
E3	R25=	445.1249
N4	R26=	204.0489
E4	R27=	287.8866
N5	R28=	240.3085
E5	R29=	386.4769
	R30=	0.0000
	R31=	0.0000

The storage register location of any point may be found as follows:

north coordinate

18 + 2(PT#)

east coordinate

19 + 2(PT#)

To store data onto magnetic cards, input **20.eee** (where eee is the three-digit number of the highest register used) and execute WDTAX.

The card reader function RDTAX may be used to input the coordinates in the same way. place **20.eee** in the x-register before executing the function. The coordinates may also be input and output from extended memory or external memory in a similar manner.

A number of the Surveying Pac programs, as well as user-written programs, contain the command "**CLRG**" as a part of their 'housekeeping' routine. This command will clear ALL of the registers, including the ones which are holding your stored coordinates.

The routine, "**C***" will clear only those registers directed by the number in the x-register, and will work in any HP-41. If you are using a CX, there is a function, "**CLRGX**" which will do this, and you should (where possible) substitute it for "**CLRG**" (see page 3) in your programs.

$6.95

41 886

ISBN 0-9616846-7-4