

THE TARGET

MAY/JUNE 1980

-- an AIM65 newsletter

Many readers have expressed an interest in back issues. The March/April and January/February 1980 issues are now available for \$1.00 a piece. The 1979 issues are no longer available individually. These are being consolidated into a Best of 1979 volume. Other than the information contained in those issues all corrections will be made plus a new index to provide quicker reference to material. Readers who already have those issues will gain only the quick reference mentioned above.

The price for Canadian and US subscribers is increasing to \$6.00. The Postal Service will be increasing the rates later on in the year so this is one reason for the increase. Another reason is to give a little overhead to allow some slight format changes in future issues.

George Sellers passes along the following comments on Basic. Have you ever inadvertently pressed the "5" when you are in the monitor mode and have a Basic program loaded in memory? Addresses \$0-2 should have 4C ~~00~~ B9 and \$212-213 should be the address of the next (2nd) Basic program statement. If these are fixed pressing "6" will allow one to reenter Basic.

Ron Riley offers the following. Litronix is offering a kit of parts to make an intelligent display similar to Aim 65 for \$135. It comes with 16 digits and is expandable to a 40 digit display by plugging in more DL-1416 readouts. They also offer three application notes of interest, one is "The Intelligent Display" and the other two are Application Notes 9A and 13. Write Litronix 19000 Homestead Rd Vallco Park, Cupertino, CA 95014.

The price for The Best of 1979 will be \$6.00 and will be available by the end of June. Please don't write for it before hand unless you are willing to wait! The rate for foreign subscribers is \$12.00.

Once again I need feedback from you the reader. Some questions appear below. All you have to do is drop me a postcard with all the questions answered. Easy enough?

Do you have an Aim?
How much memory?
Do you have the Basic Rom?
Do you have the Assembler Rom?
Do you have the PL-65 Rom?
Which language do you use the most?
Why did you purchase the Aim?
What peripherals do you have?
What changes would you like to see?
What magazines do you read?
Are you a novice, intermediate, good or excellent programmer?
Comments.
Comments.
Comments.

SOFTWARE	RENUMBER	2
SOFTWARE	SUBROUTINES	4
SOFTWARE	LUNAR LANDER	5
HARDWARE	TAPE RECORDERS	5
HARDWARE	RS-232	6
HARDWARE	PRINTER	7
PRODUCTS	PL-65	5
PRODUCTS	BETA 32K RAM	3
PRODUCTS	ABC FLOPPY	8
PRODUCTS	CSB1	8

Copyright © 1980 Donald Clem Jr

Renumber

Steve Bresson
1302 Strawberry Lane
Hanover, MD 21076

RENUMber is a Basic program to renumber Basic programs. It is based on a Pet program in Personal Computing, March 1979, by Mark Zimmermann.

Renumbering is a built in command in many of the larger Basics. It is useful when a program has been modified extensively, when you wish to insert lines within the program, or in cleaning up a listing so that it looks nice and is easy to follow.

The program is started with RUN 63000. It asks for the START and END line numbers of the block to be renumbered. Then you supply the new line number to start numbering that block, and the increment. The original program renumbered the whole program with a start of ten and an increment of ten. I felt that the ability to renumber any block would be a nice addition. It enables you to offset subroutines on 100 or 1000 boundaries, which makes it easy to remember what subroutine is where, and to easily distinguish between unrelated portions of the program.

As written, storage for 100 line numbers is provided. Those of you with more memory may wish to set MX to a larger value to accommodate larger programs.

Since you probably won't want to save RENUM at the end of every program you use it on, you have three alternatives:

- 1) Delete it by hand each time.
- 2) Save a text file on tape. On each line duplicate one line number from RENUM. End it with CTRL Z. when you load this file, it will overwrite RENUM.
- 3) Go in and change the Basic pointers so it does not know RENUM is part of the program.

I attempted the last method, from Basic, but had problems when I tried to change the End of Basic pointer at \$75/76. However, if all you want to do is SAVE the program, it is enough to change the line pointer for line 63000 to \$0000. This may cause problems if you try to RUN or change the program, but it will not hurt the LIST or SAVE. It will list up to that point and quit. After the SAVE you can POKE the values back in and have use of RENUM again. As shown, the change would be made by a RUN 63900. (See Listing)

```
LIST 1-10000
10 GOTO 110
20 GOTO 95
50 IF A=0 GOTO 65
65 GOSUB 125
80 END
95 GOTO 50
110 GOTO 20
125 PRINT!"SUB 1":R
ETURN
RUN
SUB 1
RUN 63000
RENUM START,END=? 1,
1000
NEW #, INCR=? 1, 2
8 LINE NUMBERS CORR
ECTED
FINISH
RUN
SUB 1
LIST 1-100
1 GOTO 13
3 GOTO 11
5 IF A=0 GOTO 7
7 GOSUB 15
9 END
11 GOTO 5
13 GOTO 3
15 PRINT!"SUB 1":RE
TURN
```

```
RUN 63000
RENUM START,END=? 13
,50
NEW #, INCR=? 950, 100
2 LINE NUMBERS CORR
ECTED
LINE 7 NO ROOM TO RE
PLACE 15 WITH 1050
FINISH
```

```
LIST 1-10000
1 GOTO 950
3 GOTO 11
5 IF A=0 GOTO 7
7 GOSUB 15
9 END
11 GOTO 5
950 GOTO 3
1050 PRINT!"SUB 1":
RETURN
```

```
LIST 1-60000
950 GOTO 3
1050 PRINT!"SUB 1":
RETURN
FRE= 832
```

```
RUN 63000
RENUM START,END=? 1,
50000
NEW #, INCR=? 10, 10
47 LINE NUMBERS COR
RECTED
LINE 100 NO ROOM TO
REPLACE 90 WITH 130
FINISH
LIST 100
100 FOR I=1 TO 5: X=SI
N(I, 1): NEXT: GOTO 80
100 FOR I=1 TO 5: X=SI
N(I, 2) NEXT: GOTO 130
```

```
LIST 1-50
10 DY=40
20 PRINT!"--TWO DIA
MONDS--"
30 REM KB, APR 79, P.
115
40 REM V1A, 8, 15, 79.
5LB
50 INPUT"INSTRUCTIO
NS": I#
```

```
RUN 63000
RENUM START,END=? 40
,100
NEW #, INCR=? 2, 2
7 LINE NUMBERS CORR
ECTED
FINISH
LIST 1-110
10 DY=40
20 PRINT!"--TWO DIA
MONDS--"
30 REM KB, APR 79, P.
115
40 REM V1A, 8, 15, 79. 5
LB
50 INPUT"INSTRUCTION
5": I#
6 IF "Y"=LEFT$(I#, 1
) THEN GOSUB 500
8 X1=3: Y1=5: M=1
10 DIM B%(5, 9), C%(3
)
12 C%(0)=" ": C%(1)=
"*": C%(2)=" ": C%(3)=
"X"
14 FOR I=1 TO 5: B%(
I, 0)=I: NEXT I
110 FOR I=1 TO 5: FOR J=
1 TO 9: READ B%(I, J): NEX
T J, I
```

```
CRAZY HUH?!!
4/14/80 5LB
```


LIST

```

10 REM TEST OF DELETE
20 REM KILLS LAST PART OF A PROGRAM
30 REM V1A.4.15.80.SLB
63900 L=530
63905 LN=256*PEEK(L+3)+PEEK(L+2)
63910 IF LN>=63000 GOTO 63930
63920 L=PEEK(L)+PEEK(L+1)*256:GOTO 63905
63930 PRINT"OLD ";L;"=",PEEK(L),PEEK(L+1)
63940 POKE L,0:POKE L+1,0
63950 END

```

RUN

```

OLD 608 =
106      2

```

SAVE

```
OUT=P
```

```

10 REM TEST OF DELETE
20 REM KILLS LAST PART OF A PROGRAM
30 REM V1A.4.15.80.SLB

```

```
Z
```

```

POKE 608,106
POKE 609,2

```

LIST

```

10 REM TEST OF DELETE
20 REM KILLS LAST PART OF A PROGRAM
30 REM V1A.4.15.80.SLB
63900 L=530
63905 LN=256*PEEK(L+3)+PEEK(L+2)
63910 IF LN>=63000 GOTO 63930
63920 L=PEEK(L)+PEEK(L+1)*256:GOTO 63905
63930 PRINT"OLD ";L;"=",PEEK(L),PEEK(L+1)
63940 POKE L,0:POKE L+1,0
63950 END

```

LIST 63000-

```

63000 REM V1F.4.14.
80.SLB RENUMBER AIM
65
63002 REM AIM65 VER
SION BY STEVE BRESSO
N
63004 REM 1302 STRA
WBERRY LN. HANOVER, M
D 21076
63006 REM BASED ON
PGM IN P.C. MAR 79, B
Y MARK ZIMMERMANN
63008 MX=100
63010 DIM OLX(MX):D
M=0:AL=18:AH=2
63012 INPUT"RENUM
START,END=";SR,ER
63014 INPUT"NEW #,
INCR=";LX,IX

```

```

63016 NL=LX:IF ER>6
2999THEN ER=62999
63018 NH=0:IFNL>255
THENNH=INT(NL/256):N
L=NL-NH*256
63020 AD=256*AH+AL:
LL=PEEK(AD+2):LH=PEE
K(AD+3):OL=256*LH+LL
63030 IF OL>ER GOTO
63500
63035 IF OL<SR GOTO
63070
63040 OLX(OM)=OL:OM
=OM+1
63050 POKE AD+2,NL:
POKE AD+3,NH:NH=NL+I
X

```

Mel Evans

1027 Redeemer

Ann Arbor, MI 48103

Here is a report on the new Beta 32K Ram. It's great! (Model 6502DM 32K dynamic Ram, Beta Computer Devices, PO Box 3465, Orange, CA 92665, \$419 + \$3 UPS).

It's on a little 4-1/2 by 6 inch board that draws less than 1 amp of 5V only (the other voltages are generated on-board). It comes with an ingenious right-angle connector that allows it to fit either behind or below the Aim board, and either way, the expansion edge connector is still available for other things. With a little trimming of the Enclosure Plus connector hole, it fits in the enclosure under the Aim with room left over for another board.

All refresh is automatic on-board, so it behaves like static ram. And each 4K block is individually addressable to any 4K boundary, so you can make it contiguous with Aim ram by putting it at \$1000-8FFF. Aim 65 BASIC now responds with 36334 Bytes Free. Doesn't that "Bytes Free" give you a feeling of unlimited power?

I got mine four days after a phone order, wired the connector for "under", wired the header for \$1000-8FFF, plugged it in, and it's still working six months later.

```

63055 IF NL>255 THE
N NL=NL-256:NH=NH+1
63060 IF OM>MX THEN
PRINT"SYM TBL FULL"
GOTO 63500
63070 AL=PEEK(AD):A
H=PEEK(AD+1):GOTO 63
020
63500 REM LINE NUMB
ERS FIXED- NOW CORRE
CT GOTO TYPE STMTS.
63510 PRINT!OM;"LIN
E NUMBERS CORRECTED"
:OM=OM-1:L=529
63515 SR=OLX(0):ER=
OLX(OM)
63520 L=L+4:LN=256*
PEEK(L)+PEEK(L-1)
63525 IF LN=63000 T
HEN PRINT"FINISH":E
ND
63530 L=L+1:CH=PEEK
(L):IF CH=0 GOTO 635
20
63540 IF(CH<136)AN
D(CH<140)AND(CH<16
1)GOTO 63530
63550 L0=L:N=0
63560 L=L+1:CH=PEEK
(L):IF CH=32 GOTO 63
560
63570 CH=CH-48:IF (
0<=CH)AND(CH<10)THEN
N=N*10+CH:GOTO63560
63580 IF N=0 GOTO 6
3530

```

```

63590 IF ND=SR AND
NK=ER THEN GOSUB 638
00
63595 IF CH=0 GOT
O 63550
63600 IF CH+48=0 GO
TO 63520
63610 GOTO 63530
63800 J=-1:FORI=0TO
OM:IF OLX(I)=NTHENJ=
I:I=OM
63810 NEXTI:IF J<=
-1GOTO63820
63815 PRINT"LINE":
LN:" #":N:"NOT FOUN
D":GOTO 63890
63820 NL=J*IX+LX:NL
#=STR$(NL):NL#=RIGHT
$(NL#,LEN(NL#)-1)
63830 IF LEN(NL#)<=
(L-L0-1)GOTO 63840
63835 PRINT"LINE":
LN:"NO ROOM TO REPLA
CE":N:"WITH":NL:GOTO
63890
63840 IFLEN(NL#)<L-
L0-1THEN NL#=" "+NL#
:GOTO 63840
63850 FOR I=L0+1 TO
L-1:N#=MID$(NL#,I-L
0,1):N=VAL(N#)+48:PO
KE I,N
63860 IF N#=" " THE
N POKE I,32
63870 NEXT I
63890 N=0:N#="":RET
URN

```


Subroutines

Dale Lowery
431 Redwood Dr.
Pasadena, CA 91105

Here are some assembly routines used to get to individual or specific characters in the display. HXASC is NUMA, part of the monitor, with output to a buffer rather than to the AOD. PNWHL creates the pinwheel effect in character 20. You should call a 1/2 second delay between PNWHL calls or it will look like a star. PBUF contains the ASCII characters 1,/,,-,\.

I always JSR RCHEK in assembly code especially in any sequence calls for keyboard input. This way you can get out of a program without doing a reset.

```

RA=$AC00           ;HXASC
RB=$AC02           ;HXASC CONVERTS HEX
;DIXCH             ;NUMBERS IN A TO
;SUBROUTINE OUTPUTS ;ASCII AND SAVES IN
;DATA TO EXPLICIT  ;INDEXED BUFFER,
;DISPLAY DIGIT     ;MSD FIRST
DIXCH PHA;SAVE DATA  HXASCII CLC
TYA ;GET DIGIT WORD  LDY #2 ;INITIALIZE
ORA #80             ;FOR TWO CHARS
STA RA             PHA ;SAVE DATA
TAY ;SAVE A IN Y    LSR A
PLA ;PULL DATA     LSR A
EOR #80 ;MASK OUT CURSOR LSR A
PHA ;SAVE DATA     LSR A
STA RB ;DISPLAY IT  JMP NOUT ;GOCONVERT
TYA ;GET DIGIT ADDR AGAIN NUM2 PLA ;RESTORE
EOR #80 ;SET W=0    ; DATA
STA RA             AND #F ;DUMP MS
ORA #7C ;SET CE=1  ; NIBBLE
STA RA             NOUT CLC
LDA #FF ;SET W=1   ADC #30 ;ADDTOASCII
STA RA             CMP #3A ;BIGGERTHAN
PLA ;GET DATA AND.. ; NINE?
RTS ;RETURN        BCC STON ;NO KEEP IT
;                 ADC #06 ;YES-MAKE
;                 ; IT"A"
;                 ; THRU
;                 ; "F"
;TO USE DIXCH AS  ;
;INDEPENDENT     ;
;SUBROUTINE      ;
;CALL=LDA ASCII DATA STON STA BUFF,X
; LDY DIGIT ADDR WORD ;SAVE IN BUFFER
; JSR DIXCH       INX ;UP INDEC
;                 DEY ;DOWN COUNTER
;                 BNE NUM2 ;NOT DONE--
;                 ; GET MS NIBBLE
RTS ;FINISHED

```

```

;PNWHL
;PBUF=4 BYTES
;XSAV = 1 BYTE
;CONTENTS OF PBUF=
;$31,$2F,$2D,$5C
PNWHL JSR PHXY ;SAVE
;INDEXES ($EB9E)
LDX XSAV ;GETCHARINDEX
LDA PBUF,X ;GET
; CHAR
LDY #3C ;SET DIGIT
; 20 INDEX
JSR DIXCH ;DISPLAYIT
INX ;UP INDEX
CPX #4 ;DONE 4?
BNE RUN ;NO-GOFORIT
LDX #0 ;YES-RESET
RUN STX XSAV ;SAVE
; CHAR INDEX
JSR PLXY ;GET PGM
; INDEXES
;PLXY=$EBAC
RTS ;RESUME
;
;DISPLAY ADDRESS
; DIGIT WORD
; 01 7B
; 02 7A
; 03 79
; 04 78
; 05 77
; 06 76
; 07 75
; 08 74
; 09 6F
; 10 6E
; 11 6D
; 12 6C
; 13 5F
; 14 5E
; 15 5D
; 16 5C
; 17 3F
; 18 3E
; 19 3D
; 20 3C
;TO USE HXASC.LAS
;INDEPENDENT
;SUBROUTINE
;CALL=LDA DATA(HEX)
; LDX BUFF INDEX
; JSR HXASC

```

6502 Assembly Language Programming by
Lance A. Leventhal from Osborne/McGraw-
Hill. \$12.50 plus \$1.00 handling (non-
US or Can-\$4.00) C/O Donald Clem

Lunar Lander

Steve Bresson
1302 Strawberry Ln
Hanover, MD 21076

This version of Lunar Lander is suited to the Aim 65 because of the way it uses the display. The only things shown during a game are; the altitude (miles), the speed(MPH), and the fuel(pounds). The program expects that PB7 and PB6 of the user VIA are tied together--It uses these to implement the real time clock.

The keyboard is scanned using the GET function. (Note: A special poke to 42111 is done so that the program does not hangup on the GET function).

The 1 thru 0 keys are held down to give a thrust of 20 to 200 LBS/SEC. If nothing is pressed, then no thrust is applied. The 'C' in the first column indicates the coarse fuel scale is used. By holding the F(fine), V(very fine), or S(super fine) keys down, the thrust produced by a given key can be changed. (Hit 'C' to go back to the coarse setting).

Your rating will determine the quality of fuel used--A higher rating gives less thrust per pound of fuel. If the game goes too slow for you, hit 'U' to speed things up, and 'D' will slow things down.

Another feature of this version is that it doesn't use the approximation method used by the others I've seen. Instead, it uses the exact value obtained by integration. This is more exact, but requires that you have the LOG function available. (A series approximation to the LOG is easily done, however).

With each flash of the display, the time since the last update has been found, the new speed calculated, and the keyboard scanned.

Product

PL/65 for the Aim 65.

PL/65 is designed to improve the productivity of the programmer and to increase program reliability. Control statements such as conditional execution (IF-THEN-ELSE), conditional looping (FOR-TO-BY), coupled with a simplified block capability, support structured program design techniques.

Tape Recorders

Ron Riley
PO Box 4310
Flint, MI 48504

Here are a few comments on tapes and tape recorders. They should help to eliminate some of your mass storage problems.

Use short cassettes because long tapes are too thin and also cause tape speed variations. Don't use the 3 for a dollar bargain tapes. I have had good luck with tapes from Microsette Co. 777 Palomar Ave Sunnyvale, CA 94086. Sample tape is \$1.00. Ten tapes for \$7.50, and 50 for \$32.50. These are C-10 tapes(5 minutes per side).

Several tape recorders work well with the Aim. Some are;

GE 5145 with	counter	aprox.	\$50
GE 5121 with	counter	aprox.	\$47
GE 5003	cheap	aprox.	\$18 to 27
Sears 934			depends where purchased

I strongly recommend these especially the ones with a tape counter.

From personal experience I can say that Lloyds recorders don't work very well with the Aim. Note that the price of the recorder isn't a good indication of its usability with an Aim. The Lloyds unit that I was trying to use was about \$45 and therefore about the same price as the GE unit with similiar features yet the GE unit works and the Lloyds doesn't.

Announcement

The PL/65 compiler generates assembly language code. In addition, PL/65 allows assembly language instructions to be incorporated in-line in portions of programs where timing or code optimization requirements are critical. The result is a system implementation language which has the power and flexibility of assembly language and structuring potential of a high level language.

The PL/65 is contained in two 4K byte ROMs which plug directly into the Aim 65. The price including user's manual is \$125. The manual is available separately for \$10.

RS-232

Steve Roberts
7100 Mimosa Dr
Carlsbad, CA 92008

My approach to interfacing RS-232 terminals is quite simple. It uses only the TTL voltage levels and so is not a proper RS-232 implementation. It does work fine with my Aim running a DECwriter at 300 baud and also running into a serial port on a H/P 9845 desktop computer at 2400 baud. No handshaking is used.

Sending data from a RS-232 to Aim-65

Data is input from the terminal to the Aim through J1-Y. Input voltages are clamped between 0 and +5 volts by diodes CR8 and CR9. You may need to replace R24 with a 1.6K or larger resistor if the voltage level on Z5 pin 5 won't swing low enough.

Sending data from Aim-65 to RS-232

Data is sent to the terminal from the Aim through J1-U. Z5 pin 3 needs to be pulled up to +5 volts. This may be accomplished by connecting Z5 pin 3 to +5 volts through a 3.3K resistor, which should still allow the TTY loop to work.

Aim Application Connector J1	RS-232 Connector * DB-25S
Pin Y	Pin 2
Pin U	Pin 3
Pin 1	Pin 7

Depending on the terminal, you may have to also connect pins 5, 6, and 20 on the DB-25 connector to +5 volts.

* These connections are for a terminal. For a modem make up an adapter cable which reverses the connections to pins 2 and 3:

Y	2	2
U	3	3
1	7	7

AIM Term. Modem
J1 cable adapter
cable

Directing output to the serial port, but retaining input on the Aim keyboard. Designed to be called from Basic. Check baud rate in A417-A418.

```
*=0D00
LDA A9 ;Check for zero
BEQ AIM
LDA #00 ;Assumes output
STA A406 ;routine starts
LDA #0F ;in OF00
STA A407 ;Change if desired
RTS
```

```
AIM LDA #05 ;Use Aim
STA A406 ;Display
LDA #EF ;and Printer
STA A407
RTS
```

```
*=0F00 ;Serial port output
CMP #0D ;Check for CR
BNE CR8D
JMP LF
CR8D CMP #8D ;Check for CR
BNE OUT ;not a CR
LF JSR EEA8 ;send the CR
LDA #0A ;get a LF
OUT JMP EEA8 ;send character
;and return
;to basic
```

Call from Basic

```
N=0:GOSUB 9200:REM OUTPUT GOES TO AIM
N=1:GOSUB 9200:REM OUTPUT GOES TO PORT
9200 POKE04,00:POKE05,13:XX=USR(N):RETURN
```

The patches may reside anywhere, but the POKEd address in line 9200 will have to be changed accordingly. Also if the patch at OF00 is moved be sure to change the listing at OD00 to point at the new address. If desired line 9200 could also POKE the baud rate into A417 and A418.

Note also the KB-TTY switch stays in the KB position.

TARGET Back issues- The two 1980 back issues are available for \$1.00 a piece in the US and Canada, \$2.00 elsewhere.

The 1979 issues are being consolidated into a best of edition and will be available shortly for \$6.00, \$12 outside US and Canada.

Printer

Reprinted by permission of Olivetti Corporation from Document No. 75041. Olivetti Corporation of America, 2800 Valley Road, Harrisburg, PA 17112.

Model PU1800 Printer

During a print cycle, the motor turns clockwise as viewed to the right. Its pinion (not visible here) engages the large spur on the transmission gear, turning it counterclockwise. The helical portion of the transmission gear meshes with the helical portion of the main cam and strobe disc assembly, turning it clockwise in this view.

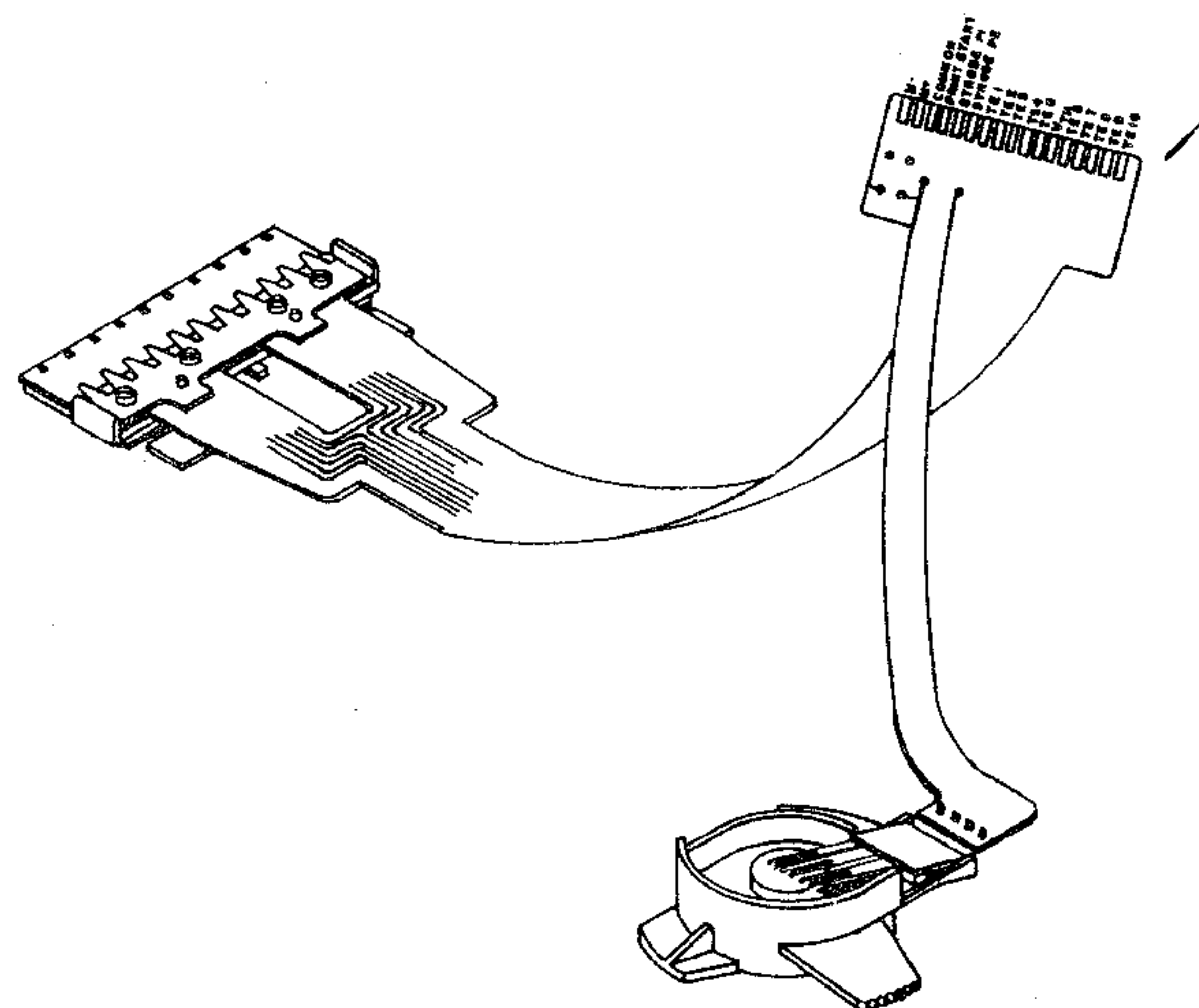
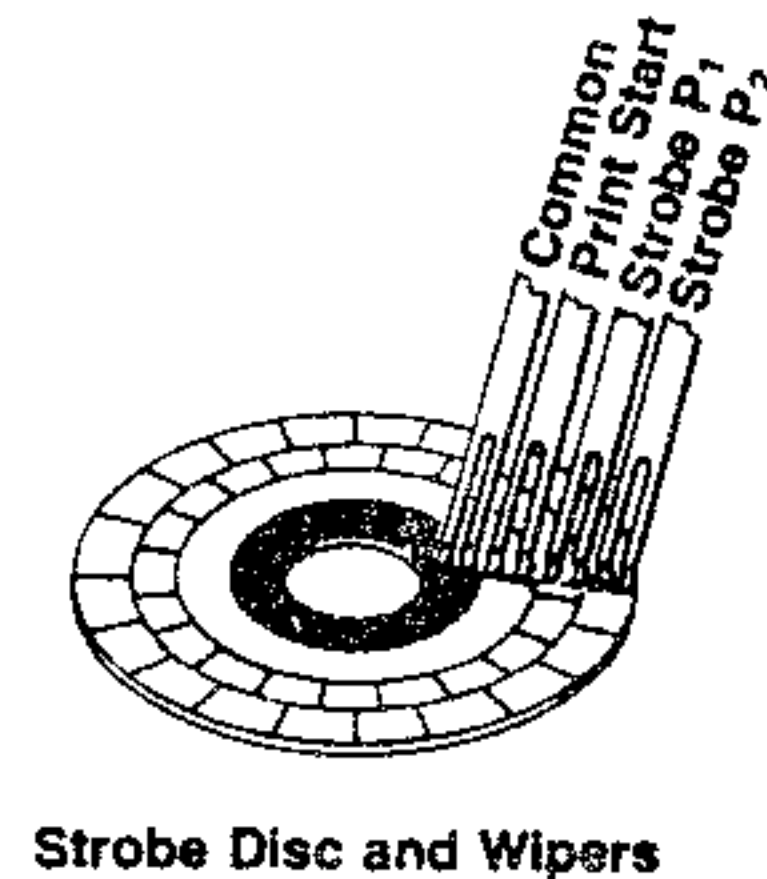
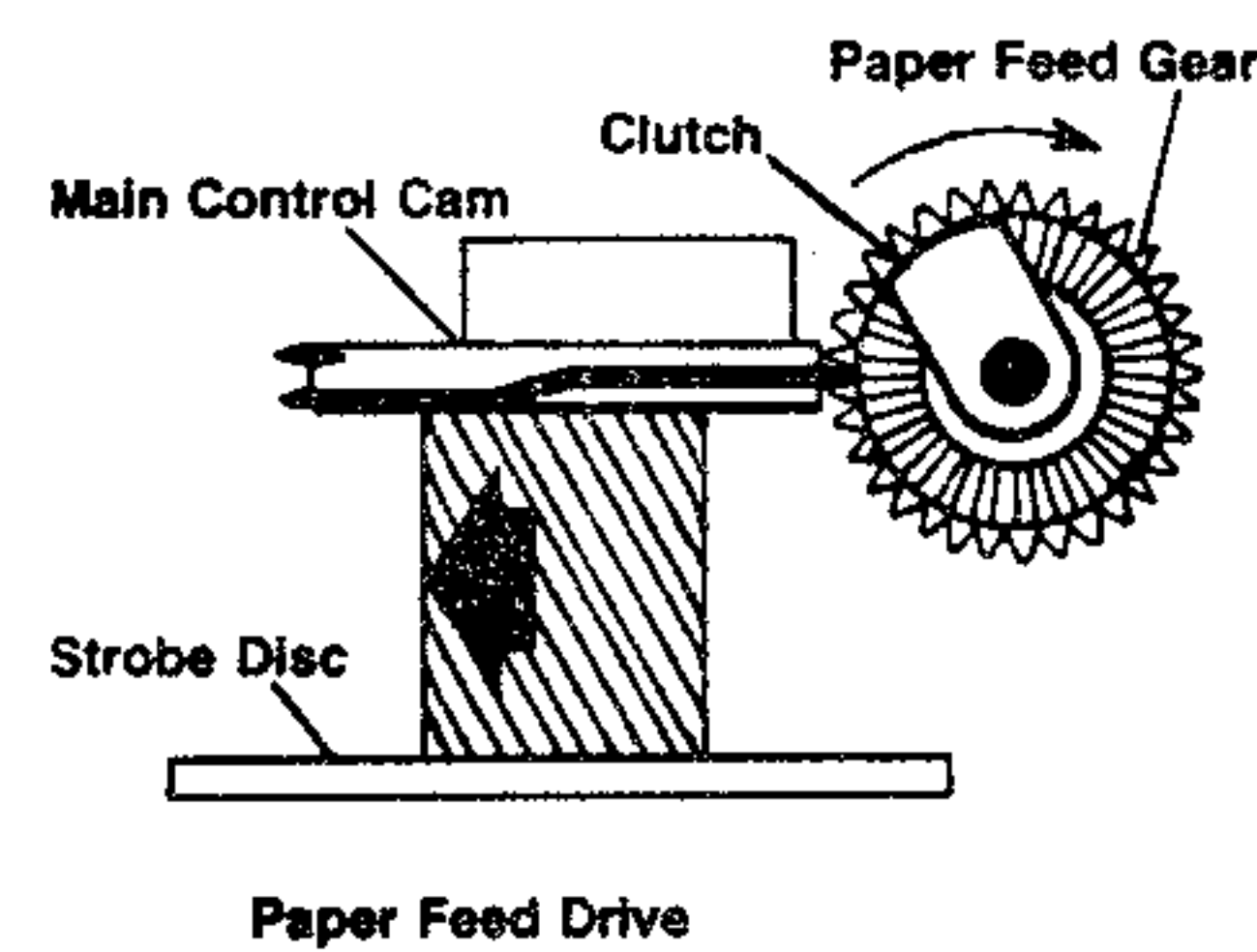
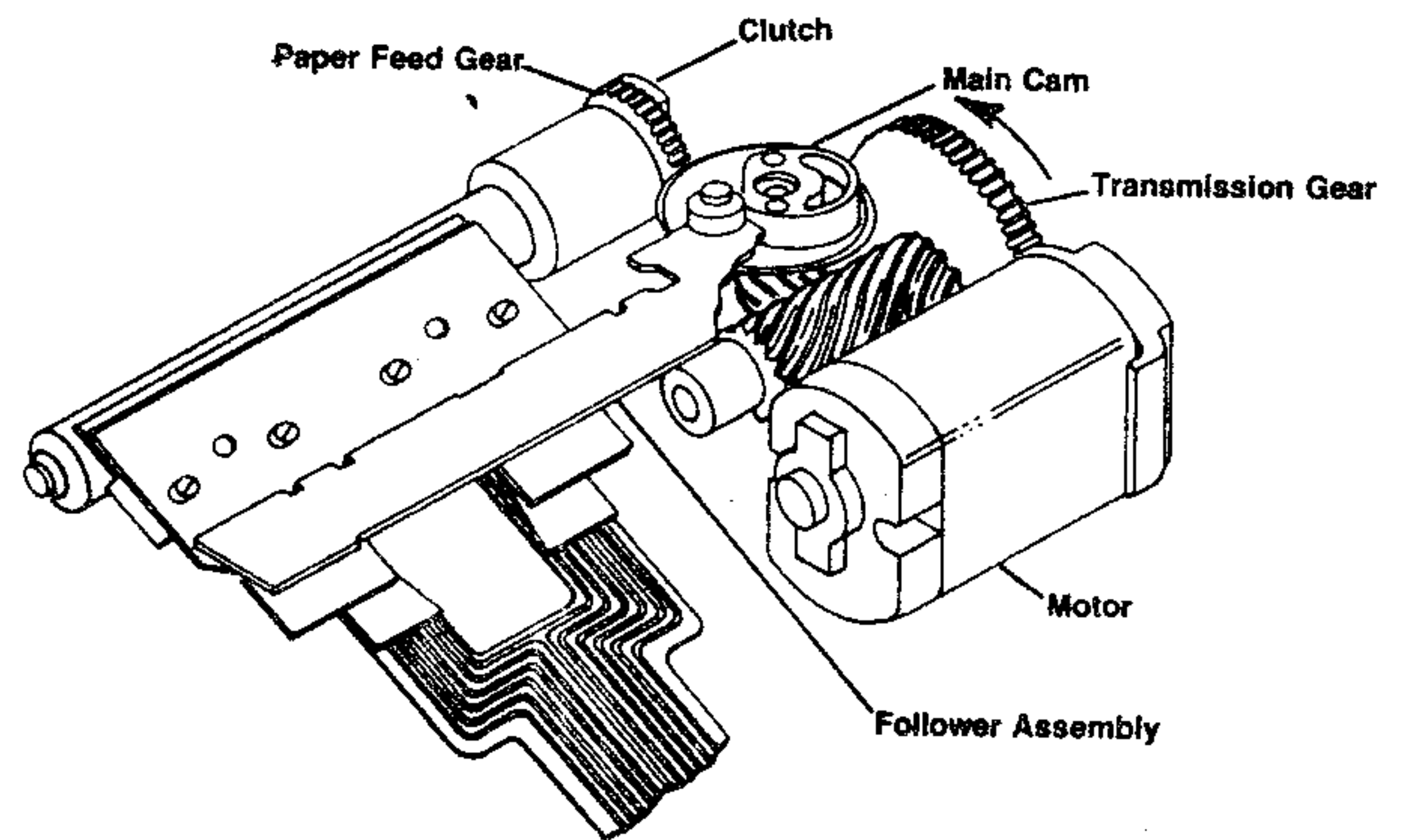
The eccentric top of the main cam moves the follower assembly from side-to-side over a distance that corresponds to the width of two characters. The roller is kept in contact with the cam by a spring and return follower (omitted here for clarity). The thermal head is clipped into the follower assembly, and slides on the paper.

Around the periphery of the main cam is a worm that engages the paper feed gear, and drives it upward by one-dot height increments. Teeth on the right face of the gear drive the clutch, which is connected to the platen. The clutch protects the worm from damage caused by pulling on the paper.

The strobe disc on the bottom of the main control cam has four circular tracks. Four brush contacts (wipers) are held against these circular tracks. During a print cycle, the strobe disc rotates with the main control cam, and these wipers contact the tracks and send the strobe signals to the control electronics for selection. During a print cycle all wipers are in continuous contact with the circular track.

The next illustration shows the flexible flat cable, which is the connection point to the control electronics. The motor leads and ground wire are omitted here. There are seventeen positions; the center-to-center spacing is 2.54 mm (0.1 inch).

Also shown are the thermal head and the strobe cap. The thermal head, flat cable, and strobe cap and wipers are one assembly, which is easily removed from the printer if necessary.



Product Announcements

Applied Business Computer Co. introduces an advanced Floppy Disk System for the Aim-65 Microcomputer. The FP-950 Controller Module controls up to 4 two-sided Floppy Drives, eight heads.

It uses State-of-the-Art technology to handle either Single or Double Density formats. It can control 35,40 or 77 tracks/side, as well as overlapped seek capability and independent motor control.

The new Advance Operating System (ADOS) uses a new concept on the file structure, allocating files dynamically which results in quick deletion and compression of files.

ADOS allows manipulation of input/output files in Buffered Sector or Block Transfer Modes up to 63KBytes. Files can be read/write from Editor, Monitor, Assembler or Basic, besides the normal commands for manipulation of files. Module FP-950 \$475, Single sided driver with power supply and case \$375, and Double sided driver with power supply and case \$460

Applied Business Computer Co. 707 G So. State College Blvd. Fullerton, CA 92631.

Compas Microsystems 224 S.E. 16th Street
Ames, Iowa 50010

Compas Microsystems announces the CSB1 and CSB1D. The CSB1 adheres to the Rockwell System 65 bus and provides 2K static RAM, sockets for up to 32K EPROM/ROM, 50 input/output lines and 10 buffered output lines.

CSB1D is a development oriented model of the CSB1. The difference between the two boards is that the CSB1D has been modified to disable the on-board 6502 processor. This allows the CSB1D to be plugged into the Rockwell expansion motherboard (A65-009) for the Aim 65. The CSB1D appears like an expansion board to the Aim 65.

Once the project software (and hardware) is developed, the software may be programmed into EPROM/ROM and installed in a standard CSB1. If desired, the CSB1D may be easily modified back to the standard CSB1 configuration. Contact Mike Corder at 515-232-8187.

8 TARGET May/June 1980

THE TARGET
c/o DONALD CLEM
R.R. #2, CONANT RD.
SPENCERVILLE, OHIO 45887