PROGRAMMING AIDS T2
062-5972-01

PROGRAMMING AIDS T2 is a tape collection of 15 programs PLUS the Programming Tips handbook to aid you in creating or dissecting a 4050 BASIC program. Employ these routines to follow a program's structure, track variables or change them, enhance graphs. Take a look at the abstract describing the novel program which converts bases (Hexadecimal Operations). The Programming Tips handbook has been a best seller and is sure to expand and streamline your 4050 operations. The individual abstracts describe each program.

Title/
Previous Abstract #

Flowchart Program for 4051 Basic Programs
51/00-8005/1

Automatic Tape Directory
51/00-8022/0

Tape File Header Expander
51/00-8039/0

List Program's Statement Types
51/00-8001/0

Sort & List Program Variables
51/00-8003/0

Change and List Program Variables
51/00-8028/0

Variable Name Changer
51/07-8034/0

Program Module Cross Reference
51/00-8027/0

Program Module Map
51/00-8027/0

Dash, Dot, Dash-Dot Routine
51/00-9501/0

Neat Tics and Axis Labeling
51/00-9502/0

Linear Axis Labeling Routine
51/00-9503/0

Hexadecimal Program
51/00-5503/0

Disk Directory
51/07-8049/0

File Identifier
51/07-8031/0

Programming Tips Handbook, Vol. 1
51/00-7004/0

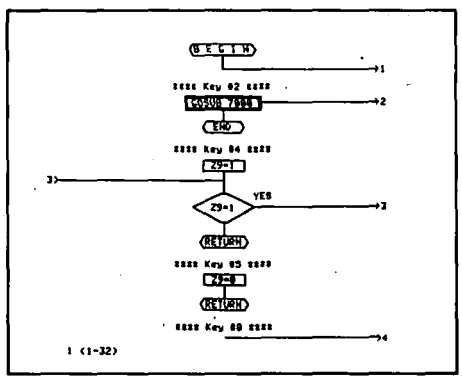
Program 1

Title: **Flowchart Program for 4051 Basic Programs**
Author: Han Klinkspoor
Tektronix, Inc.
Amstelveen, The Netherlands
Revised by Leland C. Sheppard
Sunnyvale, CA
Memory Requirement: 9K for the program; will run on 16K machine and chart programs with up to 170 branches. On 24K or 32K machines, it will chart programs with 700 or more branches.
Statements: 405
Files: 1 ASCII Program

This program will flowchart any 4050 BASIC program from a tape file. It does the job in the following way:

In the first pass, a map of the branches is made to enable "look ahead" in the second pass. In the second pass, program lines are processed one at a time. The line number is stripped off and the branch table is examined to draw incoming or outgoing branches, if any. As each entry in the branch table is processed, the page number on which that reference occurred is plugged back into the branch table for subsequent printing. As the program is charted the current page number and the starting and ending statement numbers shown on that page are printed on the bottom of the page.

Restrictions: Limit of 4 character statement numbers to allow the program to run on a 16K machine. This may be modified to 5. Maximum of 20 FOR/NEXT loops unless modified to increase the limit. Page limit is 99 but may be modified.



Program 2

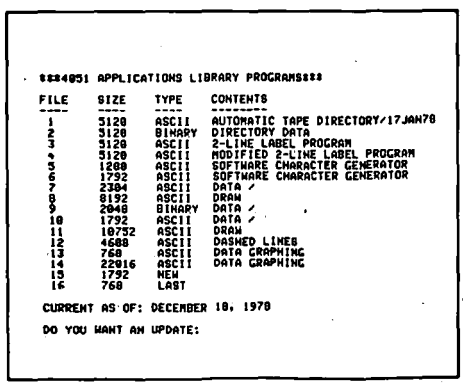
Title: **Automatic Tape Directory**
Author: E.A. Bleiweiss
University of New Mexico
Civil Engineering Research
Albuquerque, NM
Memory Requirement: 24K
Statements: 150
Files: 1 ASCII Program
Requires 1 data file

The program will list in order the contents of a magnetic tape cartridge by file number, size, type (ASCII or BINARY) and contents (title of Program). It will then load and run a selected program if requested.

The program requires the first 2 files on the tape. File 1 is for the program; file 2 for the data (Table of Contents).

When used for the first time the program will scan each file starting with file 3, then read and list the first line of each file. The first line of each program must be a REM statement and contain the title in braces [ ]. Up to 48 characters may be used in the title.

The maximum title storage capability in file 2 is 43 files. If more are required, file 2 may be marked larger and the appropriate lines of code changed in the program.



```
****4051 APPLICATIONS LIBRARY PROGRAMS****

FILE    SIZE    TYPE     CONTENTS
----    ----    ----     --------
  1      5120   ASCII    AUTOMATIC TAPE DIRECTORY/17JAN78
  2      5120   BINARY   DIRECTORY DATA
  3      5120   ASCII    2-LINE LABEL PROGRAM
  4      5120   ASCII    MODIFIED 2-LINE LABEL PROGRAM
  5      1280   ASCII    SOFTWARE CHARACTER GENERATOR
  6      1792   ASCII    SOFTWARE CHARACTER GENERATOR
  7      2304   ASCII    DATA /
  8      8192   ASCII    DRAW
  9      2048   BINARY   DATA /
 10      1792   ASCII    DATA /
 11     10752   ASCII    DRAW
 12      4608   ASCII    DASHED LINES
 13       768   ASCII    DATA GRAPHING
 14     22016   ASCII    DATA GRAPHING
 15      1792   NEW
 16       768   LAST

CURRENT AS OF: DECEMBER 18, 1978

DO YOU WANT AN UPDATE:
```

Program 3

Title: **Tape File Header Expander**
Author: Randy Bowling
Tennessee Valley Authority
Chattanooga, TN
Memory Requirement: 8K

Statements: 197
Files: 1 ASCII Program

This program permits you to annotate the standard file header as well as add information in the remaining 256-byte header block not occupied by the standard header.

A descriptively annotated file allows quick identification during the normal TLIS. The expanded header would only be displayed using the expanded TLIS portion of this program.

The annotations of the standard header or its expansion does not affect the data contained in any tape file. The file may contain a program or data in ASCII or binary. Programs may be secret or open.

```
 1   ASCII   PROG    2304 MANNING'S EQUATION DEPTH-FLOW
 2   ASCII   PROG   13056 BABY ANNOUNCEMENT CARD
 3   ASCII   PROG    5120 MC6800 DISASSEMBLER PROGRAM
 4   ASCII   DATA     768 MC6800 DISASSEMBLER DATA
 5   ASCII   PROG    6400 QUANTEX DS-12 INTERFACE
 6   BINARY  PROG   28224 MASS STORAGE MANAGEMENT SYSTEM
 7   ASCII   PROG    6144 4924 MASS TAPE DUPLICATION
 8   ASCII   PROG    2560 PROGRAM MODULE CROSS REF & MAP
 9   ASCII   PROG    3840 PROGRAM MODULE CROSS REF & MAP
10   ASCII   PROG    1536 CP1B GET COMMAND TRIGGER HP3438A
11   ASCII   PROG    4096 CP1B GET COMMAND TRIGGER IC54000
12   ASCII   PROG    2816 FILE IDENTIFIER
13   ASCII   PROG   19968 SLIDEMAKER II (REVISION) PROGRAM
14   ASCII   DATA    5632 SLIDEMAKER II (REVISION) DATA
15   ASCII   PROG    3072 SLIDEMAKER II (REVISION) CONVERT
16   ASCII   PROG   21248 Q-PLOT
17   ASCII   PROG    7424 SYMBOLGEN
18   LAST             768

Please press the RETURN key when you are ready to continue.
```

Program 4

Title: **List Program's Statement Types**
Author: Brian Diehm
Tektronix, Inc.
Wilsonville, OR
Memory Requirement: 8K
Statements: 113
Files: 1 ASCII Program

This program reads a tape file containing a BASIC ASCII program and prints an alphabetized list of all the statement types used in that program, together with the number of occurrences of each statement type. Then the list is sorted in decreasing order by number of occurrences and reprinted. The program first asks the user which tape file contains the program to be analyzed. Then after reading the file, the alphabetized list of statement types is printed, with the count of the occurrences of each type. This is followed by a total of the number of statements in the analyzed program. Provision is made to allow processing of several files, combining the results into one list.

```
STATEMENT TYPES USED (USE ORDER):

LET(IMPLIED)    41
IF              19
PRINT           15
REM             10
FOR              5
NEXT             5
GO TO            4
INPUT            4
PAGE             3
FIND             2
DIM              1
END              1
GOSUB            1
ON               1
RETURN           1

TOTAL NUMBER OF STATEMENTS: 113
```

```
STATEMENT TYPES USED (ALPHABETICALLY):

DIM             1
END             1
FIND            2
FOR             5
GO TO           4
GOSUB           1
IF              19
INPUT           4
LET(IMPLIED)    41
NEXT            5
ON              1
PAGE            3
PRINT           15
REM             10
RETURN          1

TOTAL NUMBER OF STATEMENTS: 113
Press RETURN to continue.
```

Program 5

Title: **Sort & List Program Variables**
Author: Dan Taylor
Tektronix, Inc.
Wilsonville, OR
Memory Requirement: 8K
Statements: 118
Files: 1 ASCII Program

This program reads a BASIC ASCII program from tape and produces an alphabetized table of the variables used in that program. The BASIC program may be stored on multiple tape files as long as the files are sequential on tape. No operator intervention is required between first and last files.

```
VARIABLES:

A    A$
     B$
     C$

F    F$   F0   F1   F2   F3   F4   F5   F6   F7
     G$
     H$
J    J$   J0   J1   J2   J3   J4   J5   J6   J7   J8   J9
     K$
     L$

N    N$


     R$
     S$
     T$

     V$
     W$
```

Program 6

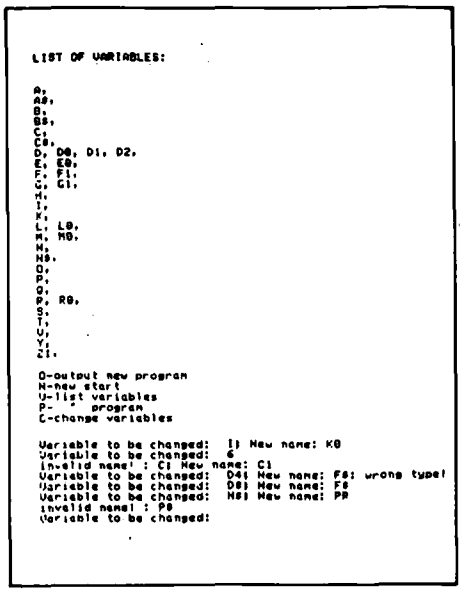
Title: **Change and List Program Variables**
Author: S. Schicktanz
Technical University
Munich, Germany
Memory Requirement: 8K
Statements: 143
Files: 1 ASCII Program

The program allows listing or changing the names of the variables of an ASCII program from tape. Listing the program is also available. When changing variable names, input is tested for validity and correct type; errors are indicated by an appropriate message. The changed program can then be output to the original or another tape file.

The program can be used either with the menu or the User-Definable Keys. The user is prompted for necessary input information by use of the POInter-statement. It is not necessary to terminate input with the RETURN key.

User-Definable Keys provide:

Menu
Program input
Variable list
Variable change
Program list
Program output

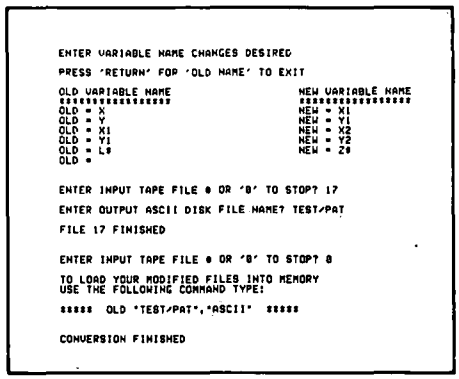


Program 7

Title: **Variable Name Changer**
Author: Mallory M. Green
U.S. Dept. of HUD
Washington, D.C.
Memory Requirement: 8K
Peripherals: 4907 File Manager
Statements: 167
Files: 1 ASCII Program

Variable Name Changer allows a programmer to change any number of variable names in his program with ease. The program to be changed must be an ASCII file on tape in the 4050 internal tape drive. **Variable names in REMark statements are not changed.** The revised program is written as an ASCII file on the 4907 disk, leaving the original program intact on tape.

Variable Name Changer prompts the user for a list of current variables to be changed and for their new names; it then prompts for the file number of the program to be revised. The file is read a statement at a time. If it is not a REMark or IMAge statement, it is scanned character by character for variables. When a variable to be changed is discovered, it is replaced with the new variable name. The process continues until the whole program has been read from tape and written to disk



Programs 8-9

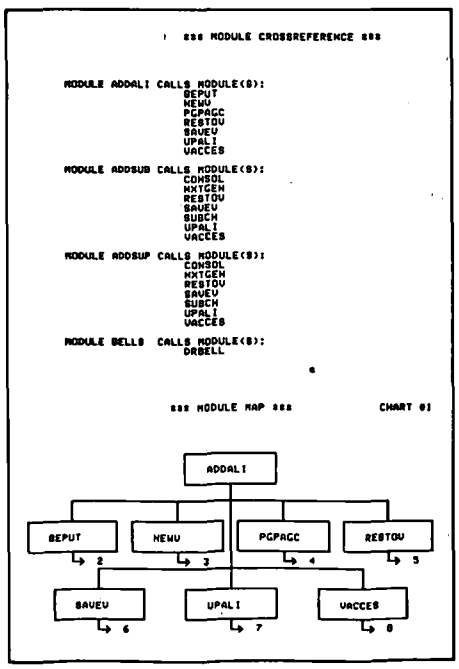
Title: **Program Module Cross Reference and Map**
Author: Captain S.K. Sanford
Aberdeen Proving Ground, MD
Memory Requirement: 16K
Peripherals: Optional-4051R06 Editor ROM Pack
4924 Tape Drive
Statements: 276
Files: 2 ASCII Program
Requires user-created data files

The cross reference program requires that the user create two files of calling and called subprogram names using the 4051R06 Editor ROM or a simple BASIC program. The first file must be sorted in calling program sequence (alphabetically), while the second, which is identical, must be sorted in called program sequence.

The program reads the created data files from the 4050 or a 4924 one at a time and produces a listing of the calling programs with their called programs, then a listing of the called programs with their calling programs. The pages of output are numbered alphabetically from "a" to "zz", and may be automatically copied by the 4631.

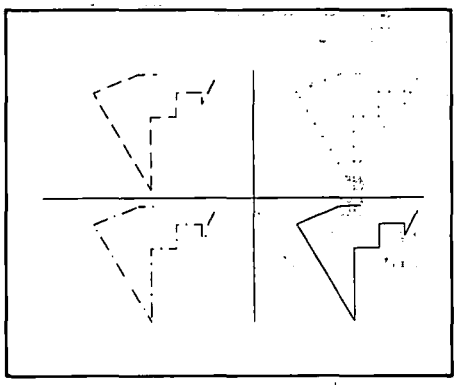
The module map program requires a file on tape showing the calling and called module names, and their interrelationships. The program searches the tape for all occurrences of a calling program and records the called program modules. In order for a program to appear on the module map, it must be called by another program, with the exception of the "MAIN" program and an optional "BLOCK DATA" program for use with FORTRAN program systems.

The map appears as multiple pages on the 4050 and may be automatically copied. Each page is headed by one program. The programs called by this first program are displayed in blocks linked by arrows to the first block. From each called program block is an arrowhead and the number of the page on which that program appears.



Program 10

Title: **Dash, Dot, Dash-Dot Routine**
Author: Nick Fkiaras
Tektronix, Inc.
Wilsonville, OR
Memory Requirement: 8K
Peripherals: Optional-4662 Plotter
Statements: 151
Files: 1 ASCII Program



The program draws a solid, dotted, dashed, or dot-dashed line between any two points X1,Y1, and X2,Y2 regardless of the window and viewport used. User inputs X and Y coordinate points, viewport, window, line type and output device. User-Definable Keys enable the user to:

Enter data and draw
Redraw with a different line type

Program 11

Title: **Neat Tics and Axis Labeling**
Author: Dan Taylor, Kathy Thurman
Tektronix, Inc.
Wilsonville, OR
Memory Requirement: 8K
Statements: 78
Files: 1 ASCII Program

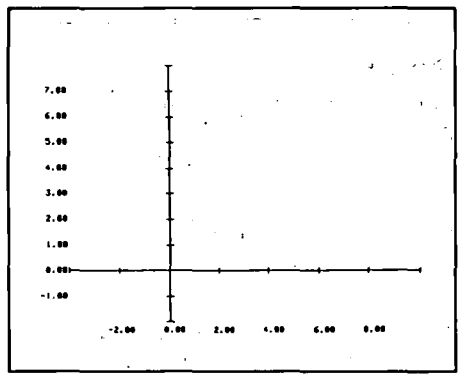
This program is a subroutine designed to be used with a user program. The subroutine prepares the screen for a user's graph by:

1. Calculating "neat" tic lengths
2. Setting the WINDOW
3. Setting the VIEWPORT
4. Drawing an axis and labeling the tic marks. The axis is drawn through user data value 0 (or data min if 0 is not in the WINDOW). Tic labels always appear to the left and bottom of the screen.

Tic marks on the axes are presumed to be evenly spaced (not logarithmic).

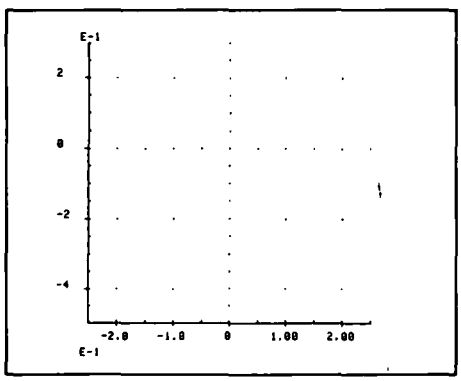
Requires minimum and maximum data values and number of tic intervals desired on each axis. Labels are printed in scientific notation for either axis if any label on that axis = 10.

The viewport allows room on the screen for a title to be printed above the graph.



Program 12

Title: **Linear Axis Labeling Routine**
Author: Steven Den Beste
Tektronix, Inc.
Wilsonville, OR
Memory Requirement: 8K
Statements: 196
Files: 1 ASCII Program



This program is a subroutine designed to be used with a user program. The subroutine generates an L-shaped axis, with tics and labels, covering any plot range, and places it anywhere on the screen. It requires 10 input variables and passes back 8 of them to describe the plot exactly.

All labels are 4 characters, including a decimal point and a sign (if negative).

For orientation, a grid of points is generated within the plottable areas. A point is placed at the intersection of any two intersections and at the intersection of any tic and zero, if zero is within range. (This is optional.)

An example program is included.

Program 13

Title: **Hexadecimal Program**
Author: Marv Abe
Tektronix, Inc.
Wilsonville, OR
Memory Requirement: 16K
Statements: 360
Files: 1 ASCII Program

This routine allows the user to perform some miscellaneous hexadecimal functions using the 4050 System. Each one of the routines provided are called by the User-Definable Keys.

Conversion routines provided are:

1. Decimal Values—Hex Representation
2. Hexadecimal Values—Decimal Representation
3. RAD40—ASCII*
4. ASCII—RAD40 Representation*
5. Hexadecimal #—Bit Pattern
*NOTE: For TEKTRONIX 4081

Arithmetic functions provided are:

1. Hexadecimal Subtraction
2. Hexadecimal Addition

Both functions are provided in cumulative form and add and subtract from some constant value. The different routines prompt the user for the required input and most always terminate on a carriage return.

Program 14

Title: **Disk Directory**
Author: Nick Ogbourne
Comalco Aluminium (Bell Bay) Ltd.
George Town, Tasmania
Memory Requirement: 8K
Peripherals: 4907 File Manager
Statements: 193
Files: 1 ASCII Program

Disk Directory maintains a directory of up to 50 disk programs and controls access to and execution of those programs.

Disk Directory creates and maintains an index file. This index file includes the file identifier, program # (sequenced in the order entered), and user-input information (up to 44 characters) about the file. Programs may be added or deleted through the User-Definable Keys.

Disk Directory reads the index file, prints a directory of the files (multipage if necessary) and prompts for the program of your choice. It will warn you if a selected file is a binary data file. Any other type of file it will attempt to load.

Program 15

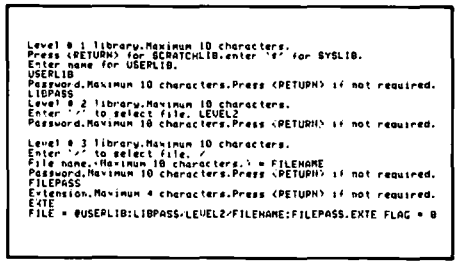
Title: **File Identifier**
Author: Nick Ogbourne
Comalco Aluminum Ltd.
George Town, Tasmania,
Australia
Memory Requirement: 8K
Peripherals: 4907 File Manager
Statements: 111
Files: 1 ASCII Program

The program is a subroutine to compile a file identifier that will comply with the 4907 File Manager rules.

The program prompts the user to select libraries to the selected level, up to level 4, including SYSLIB or SCRATCHLIB. Passwords for any or all libraries may be added.

Following library selection, file selection on the same basis occurs, plus the selection of a file extension.

The valid file name is then returned in E$ and a flag, E0, assumes a value of 0 if the file does not exist and 1 if it does exist on the currently mounted disk



Program 16

Title: **Programming Tips Handbook, Vol. 1**
Author: Applications Library Members
Documentation Only

The PROGRAMMING Tip Handbook is a collection of all of the Programming Tips and BASIC Bits from the first three volumes of TEKniques (1977—1979), published as one volume. Corrections from issues subsequent to the tip are incorporated into the handbook in the text of the tip. In this way, all of the Programming Tips and BASIC Bits from these issues can be found in one place. And the Programming Tips Handbook is well indexed to find the desired tip easily. For instance, information can be found by programming area, rather than by title or TEKniques issue alone.