DALLAS TI HOME COMPUTER GROUP
JANUARY, 1985

Meetings: 7:00pm, 3rd Friday each month
At Northlake College, Room B-206, Irving, Texas

>>> NEXT MEETING: Friday, 18 January, 1985 <<<

President: Robert Bayne
Vice-President: Keith Althar (STARTEXT: 50528)
Secretary: Richard Roberts (SOURCE: TI3352; STARTEXT: 8762)
Treasurer: Earl Bullock
Editor: Robert Lee Hoffpauer (SOURCE: TI3700; STARTEXT: 42437)

This newsletter is the official publication of the DALLAS TI HOME COMPUTER GROUP, a non-profit organization serving member/users of the Texas Instruments 99/4A HOME COMPUTER. For more information you are invited to attend our next meeting or send a SASE to: DALLAS TI HOME COMPUTER GROUP, [redacted] Irving, Texas 75061.

SECRETARY'S MINUTES: Richard Roberts

Last month's meeting of the Dallas TI Home Computer Group was on December 14, 1984. The meeting was called to order by club vice-president Keith Althar.The minutes from the November meeting were read by Richard Roberts.

A request from software library chairman Glen Ashe to buy 50 disks for the use of the library was put to a vote, and passed. Another list was made of members who need a copy of the FORTH documentation, and an attempt will be made to secure these copies soon.

For the evening's program, Robert Hoffpauer (newsletter editor) presented a FORTH program which was based on a BASIC program in a recent edition of MICROpendium. The program counted the number of ways to make change for $1.00, and takes just under 30 minutes to complete. Robert presented two FORTH versions, one that mimics the BASIC program, and one that is written especially for speed. He then showed how the two programs differ, but accomplish the same task. Of course, the first FORTH version ran about 30 times faster than the BASIC version, but the second FORTH version beat that by a mile, running in only 18 seconds.

Resorting to plain ole programmer's ingenuity, Dan Johnson proceeded to show how the BASIC version could be maximized for speed, and managed to cut the time down to under a minute, by eliminating many of the redundant loops, and PRINTs to the screen. It really is amazing, the things that can be done in computer programming, if one would just try to look logically at a problem, rather than resorting to time honored, and well worn methods, that sometimes are just simply inefficient.

EDITOR/ASSEMBLER: Robert Lee Hoffpauer

The nominating committee that Kieth appointed at the November meeting met this past Sunday to finalize their recommendations. It is the job of this committee, under our by-laws, to assure that there will be at least one candidate for each of our club's four elected positions -- president, vice-president, secretary, and treasurer -- by persuading four people to serve.

It is not the situation that the jobs are so very demanding that no one will consider serving, simply that most of us do not seek to serve. It is this reticence that hinders the growth of our club. We have a variety of small tasks that need to be done, done regularly, and done well. For instance, somebody needs to meet people at the door and welcome new members; somebody should make the commitment to lead a special interest group for newer, and less experienced members. There are more tasks to be done that will ever be done, but still, we should be doing more as a group than we are; and the sad part is that we could be.

If you cannot decipher the listings, analyze them. The way to analyze a FORTH program is to sit down with a pencil, a pad of paper, and your reference manual, and diagram what each word does. The key to diagramming the word is to build a picture of the stack. Remember that in FORTH, numbers in a program are PUSHed onto the stack, words POP their arguments from the stack, and words PUSH their results back onto the stack. So, when doing a diagram, the convention is to just show what values are on the stack before the word is invoked. Using one sheet of lined paper for each word defined, write the words (and numbers) contained in the definition down the right side of the paper, in the order that the definition is written in the FORTH program. Now, build the picture of the stack from the left side of the paper toward the words and numbers on the right side. Show what the stack looks like before the word or number on the right side is reached. Of course, the stack for one word is the result of the previous word above it. FORTH is not easier than BASIC -- it is more powerful than BASIC, and easier than assembly language.

SCR #32

```
( CHANGEMAKER VER 2.0 22 NOV 84 RLH ) BASE->R DECIMAL
: CLROUT2 ;    0 VARIABLE CNTR
: TST1 DUP 100 = ;   : TST2 DUP 100 > ;
: CHNG DROP CNTR DUP 1 SWAP +! @ 5 .R ;
: PNYS TST1 IF CHNG ELSE                   5 + MYSELF ENDIF ;
: NKLS TST1 IF CHNG ELSE  DUP PNYS  5 + MYSELF ENDIF ;
: DIMS TST1 IF CHNG ELSE  TST2 IF DROP
                    ELSE  DUP NKLS 10 + MYSELF ENDIF ENDIF ;
: QTRS TST1 IF CHNG ELSE  DUP DIMS 25 + MYSELF ENDIF ;
: HLVS TST1 IF CHNG ELSE  DUP QTRS 50 + MYSELF ENDIF ;
: FIN CR CR ." WAYS TO MAKE CHANGE FOR $1:" CNTR @ 5 .R ;

: CHANGEMAKER 0 HLVS FIN ;

R->BASE  CHANGEMAKER FORGET CLROUT2

```

DALLAS TI HOME COMPUTER GROUP
[redacted] Place
Irving, Texas 75061

TRAVEL A NEW

FIRST CLASS MAIL TO:

MEMBER 116 EXP: 5/685
RICHARD ROBERTS
[redacted]
VING, TEX 75041