



SUPER TLEGS

onboard relocater

16K LEVEL II

BY
ALLEN GELDER
© 1979

Super TLEGS

copyright 1978 Allen Gelder

Super TLEGS is a machine language program designed to be used in conjunction with T-BUGtm, the TRS-80tm monitor. Super TLEGS relocates T-BUG to the user's choice of RAM, in addition, Super TLEGS relocates itself along with T-BUG. This means that the compact unit of T-BUG and Super TLEGS can move again from the new location. In practice this means escape from those revolting wipeouts that occur when your monitor coincides with interesting program material. Now just relocate T-BUG to an unoccupied corner of RAM and you are again free to examine and modify, # P punch backup copies, etc. Locate two distinct copies of T-BUG in different areas of RAM; if one gets contaminated, generate a clean copy from the survivor and save reloading. Or experiment with T-BUG itself, exchanging control between unmodified and your own kustom versions.

INSTRUCTIONS FOR USE:

~~T-BUG~~
*? ~~T-BUG~~ ENTER Load your copy of T-BUG using SYSTEM.
*? ~~T-BUG~~ ENTER Load Super TLEGS. File name is TLEGS.
~~TLEGS~~ Volume 5½.
*? / ENTER Enter T-BUG.

Control is now with T-BUG. Relocating involves passing two parameters to Super TLEGS via CPU registers DE and HL. Recall that Level II T-BUG occupies 4380-4980 in RAM. Super TLEGS occupies 49A0-4AA0, nearly contiguous. The first parameter is the new starting address of T-BUG, placed by the user in DE. The second parameter is the present starting address of T-BUG, initially 4380. This goes in register HL.

NOTE constraints: The following constraints are on DE.
1) E must be 80H.
2) D may be any hex byte such that DE has at least 720H bytes vertical clearance.

It helps to displace T-BUG in round thousands because then three digits of the familiar RAM register storage locations are retained. Registers AF'-PC are stored by Level II T-BUG at 4825-483C. Displacing by round thousands to, say, 7380-7980, means that your CPU registers will be stored at 7825-783C. While not an absolute

necessity, since TLEGS relocates to the nearest hundred, this practice does make remembering things a little easier.

EXAMPLE: Relocating T-BUG to 7380-7980 from 4380-4980.

# M 4831 FF 80	Load DE with new starting address.
4832 FF 73	
4833 FF 80	Load HL with present starting address.
4834 FF 43	
4835 FF	<u>X</u> Exit the # M mode.
# R	Check DE and HL in the #R register mode.
# J 49A0	Jump to Super TLEGS.

Control has now been assumed by the high RAM copy of T-BUG.

AF'-PC are now at 7825-783C, Super TLEGS is at 79A0-7AA0. The T-BUG register configuration is meaningful as follows.

IX holds your present starting address. Here 7380.

IY holds your previous starting address. Here 4380.

If you wish to distinguish between the proliferating T-BUGS, the prompt character byte is found at 43DE or relocated equivalent.

EXAMPLE: Changing the prompt character on 7380-7980 T-BUG.

# M 73DE 23 5C	Change the prompt character byte.
73DF CD	<u>X</u> Exit the # M mode.
↓	New prompt character

EXAMPLE: Relocating T-BUG to 5080-5680 from 7380-7980.

↓ M 7831 FF 80	Load DE with new starting address.
7832 FF 50	
7833 FF 80	Load HL with present starting address.
7834 FF 73	
7835 FF	<u>X</u>
↓ R	Just checking.
↓ J 79A0	Go.

Control is now with 5080-5680 T-BUG and AF'-PC are at 5525-553C.
Super TLEGS entry point is 56A0. Note IX and IY for orientation.

EXAMPLE: # Punch a copy of T-BUG and Super TLEGS at a
desired out of the way location. Load this copy
for a relocatable T-BUG.

↓ P 7380 7AA0 73A0 TBIG

Why Super TLEGS? Sometimes interesting program material coincides with the means by which it is to be examined. Then we see an extreme case of the observer affecting what is to be observed, for in your RAM is a world of finite, perfect resolution. No programs sharing locations can co-reside in such a world, and so in particular T-BUG cannot examine, modify or # P punch backup copies of anything that overlaps 4380-4980. This is clearly undesirable, and various solutions are available to ensure that a monitor will always be present and applicable to any RAM material.

One way to be sure the monitor will always be there is to put it in ROM, frozen forever. This solution does not apply to the TRS-80 as sold. Other methods keep your RAM based monitor out of the way, for instance loading the program into different levels of RAM, e.g., top of 16K, top of 32K, etc. There are monitors using this scheme available now, and it certainly beats locating in the densely populated lower 4K, as does your T-BUG. Since we wish to preserve T-BUG, it must be moved and all self referential aspects must reflect the change. The overhead for this task comprises Super TLEGS, including routines for the computation of displacement and transfer, and an uncondensable table that refers to T-BUG in a semantic way. Super TLEGS itself is place independant and moves with the formal T-BUG material. Super TLEGS is therefore an onboard relocater.

Finally, it should be noted that Super TLEGS confers two distinct new abilities on T-BUG. The first is the ability to move. This lets the user examine, alter, # P punch backup copies and in general brings the powers of the monitor to bear on subject program material. The second new ability is that of self replication. Simultaneous copies of T-BUG can coexist in your RAM. This not only enhances survivability, in that runaway program flow often will inflict only local damage that leaves a distinct monitor untouched,

but opens the way to mutation, in the form of kustom monitors with properties developed by the user. Generating parallel T-BUGs allows such experimentation, putting the user in the position of an evolutionary force! Anyway, it's hoped you will get a kick out of running with TLEGS.

ALLEN GELDER

Box 11721 Main Post Office

San Francisco, CA. 94101