DOWNLOAD PDF TRS-80 DATA FILE PROGRAMMING

Chapter 1 : can not find the "program data" folder. - Microsoft Community

Tandy/Radio Shack Book: TRS Data File Programming ()(John Wiley and Sons) Skip to main content Search the history of over billion web pages on the Internet.

The screen is reflective, not backlit. These last 12 are tactile "button"-style keys. Almost all keys other than the 12 function keys are capable of rollover without phantom keys appearing depressed , so multi-key combinations can be used. The basic unit includes: System bus interface DIP socket under a cover on the bottom of the machine. It could not run from the rechargeable nickel-cadmium batteries available at the time, but a hardware modification was available that made this possible. Many times several attempts to read a tape were required, along with much adjustment of the volume setting. This allowed the Model to display 40 or 80 column video on an external television set or video monitor. One empty drive bay permitted the installation of a second disk drive which proved handy for backing up disks. A bar code reader wand was also offered. ROM firmware[edit] When first switched on, the Model displays a menu of applications and files and the date and time. Cursor keys are used to navigate the menu and select one of the internal or added application programs, or any data file to be worked upon. This is similar to other Microsoft BASICs of the time and includes good support for the hardware features of the machine: Like previous Microsoft BASIC interpreters, variable names were restricted to two characters and all program lines and subroutines were numbered and not named. However, the default for floating point numbers is double-precision. Often after speed-typing a sentence or two, the user would have to wait several seconds for the computer to "catch up". As a small compensation, many users found it somewhat amusing watching the text appearing on the display, particularly text inserted into the body of a paragraph. A perhaps not well-known but documented feature of TEXT was that it partially supported the WordStar command interface. Another invisible deleted file named "RickY" refers to Rick Yamashita. According to Gates, "part of my nostalgia about this machine is this was the last machine where I wrote a very high percentage of the code in the product". Optional ROMs can be installed in the Model , providing a range of customized application software. Some commercial software applications for the Model were also distributed on cassette. Workarounds exist for this problem. Since the century of the date is not important for any of the software functions, and the real-time clock hardware in the Model does not have a calendar and requires the day of the week to be set independently of the date, the flaw does not at all impair the usability of the computer; it is cosmetic. Applications[edit] When introduced, the portability and simplicity of the Model made it attractive to journalists, [14] [15] [16] who could type about 11 pages of text and then transmit it for electronic editing and production using the built-in modem and TELCOM program. Data is protected by a built-in rechargeable Ni-Cd battery when the AA batteries discharge or are removed for replacement. There are several simple programs available on the Internet for transferring files between a Model and a modern personal computer or a vintage one. The Model was also used for industrial applications and in science laboratories as a programming terminal for configuration of control systems and instruments. Its compactness ease of handling and small space requirements , low maintenance needs, lack of air vents a plus for dusty or dirty environments , full complement of ports, and easy portability made it very well suited for these applications. Third-party peripherals for the Model extended its battery life and file storage capacity. Software was designed, and is still available, to extend the display capabilities to 60 columns and 10 rows of text using smaller characters and to provide more advanced word-processing or calculation software than the supplied programs. To this day, hobbyists continue to design games, applications, and hardware for this device. Simple drawing programs and games using the pixel-addressable display were favorites among users. As with virtually all other contemporary home computers, users are able to create their own applications using the included BASIC programming language. There are no built-in facilities for assembler programming, but the thoroughly-documented BASIC interpreter by Microsoft offered the clever coder tricks for accessing machine code subroutines. These tricks usually involved packing the raw object code into strings or integer arrays, and would be familiar to veteran programmers for the older TRS Models I and III. Peers and successors[edit] The Tandy [17] was introduced in as a more capable sister product of the Model Rather than

the "button" style keys of the Model , its four arrow keys are a cluster of keys of the same size and shape as those comprising the keyboard, though the function and command "keys" are still of the button type. The Tandy includes Multiplan , a spreadsheet application. It also added DTMF tone dialing for the internal modem, whereas the Model only supports pulse dialing. The last new model that could be considered part of this line was the Tandy , introduced in October Similar to the Tandy , it features a flip-up screen, but with 80 columns rather than Built-in features include a 3. This reduction in size and weight was made possible by the substitution of surface-mount chip packaging. The earlier and smaller Epson HX of used a much smaller LCD display, four lines of twenty characters, and had an internal cassette tape drive for program and file storage. GRiD was later acquired by Tandy. It was released in The Zenith ZP , also of , was introduced prior to the Tandy The two computers were notably similar, although the ZP did include BASIC and could be configured with more memory, but did not have a built-in diskette drive. Computers from two other British companies were similar in form and functionality to the Model The Cambridge Z88 of , developed by British inventor Sir Clive Sinclair , had greater expansion capacity due to its built-in cartridge slots. It had a far more sophisticated operating system called OZ that could run multiple applications in a task-switched environment. The firmware contained a powerful application called Pipedream that was a spreadsheet that could also serve capably as a word processor and database. Many modern laptop computers are larger and heavier than the Model However, practically none of them run 20 hours on batteries without a battery change. The Laser 50 educationa l computer is in the same size and form factor as the Model , and was sold by Video Technology. It concluded that the computer "is an ingenious, capable device Radio Shack Catalogs dot com. Retrieved April 10, Radio Shack Catalog. Retrieved January 8,

Chapter 2 : TRS - Wikipedia

Open Library is an initiative of the Internet Archive, a (c)(3) non-profit, building a digital library of Internet sites and other cultural artifacts in digital form.

It used a Zilog Z80 processor clocked at 1. Performing a read from the keyboard area of the memory would return the state of a particular set of keys. Many users complained about the TRS Model I keyboards, which used mechanical switches and suffered from " keyboard bounce ", resulting in multiple letters being typed per keystroke. The change was subsequently added to a firmware ROM revision. The keyboard hardware was also modified to minimize bounce. The root cause of the key bounce was found to be dust, dirt, etc. This problem was most common among heavy smokers smoke and ash particles. The key switches could be cleaned, rectifying the problem, but the bounce would be back again when the keyboard was re-exposed to the contaminating environment. Later production runs of the Model I computer replaced the nameplate with a numeric keypad. Older versions could be upgraded with a numeric keypad add-on by keyboard replacement, which also remedied the key bounce problem because other parts of the system would be updated by the service technician during installation. The color of the screen text is faintly blue the standard P4 phosphor used in black-and white televisions. Green and amber filters, or replacement tubes to reduce eye fatigue were popular aftermarket items. Later models came with a green-on-black display. Complaints about the video display quality were common. As Green wrote, "hells bells, [the monitor] is a cheap black and white television set with a bit of conversion for computer use". This had little effect on normal BASIC programs, but fast programs made with assembly language could be affected. Software authors worked to minimize the effect. Notwithstanding the primitive display hardware, many arcade-style games were available for the Tandy TRS A simple half-hour hardware fix could correct that. By installing a seventh done by piggybacking it onto chip 6 , it was possible to display all characters which includes lowercase letters. The manual for the popular word processor Electric Pencil came with instructions for modifying the computer. Although the modification needed to be disabled for Level II BASIC, its design became the industry standard and was widely sold in kit form, [37] along with an eighth chip with descenders for the lowercase letters. Later models came with the hardware for the lowercase character set to be displayed with descenders. Both 64x16 and 32x16 video modes are supported, which is in contrast to 40x25 used by Apple and 80x24 used by some dumb terminals. The choice of 64x16 was part in economics, part in physics and part in usability. To support 80x24, twice as much static ram would be required 7 additional RAM chips, 9 chips total with the extra address decoding needed , plus there was no room on the board to put any extra chips. The Model I was originally intended to be used with a user supplied black and white television, what could physically be displayed on a s black-and-white TV. With 1K of RAM address space, a maximum of 16 lines are possible. With higher density RAM chips and dedicated purpose build monitors, higher resolution crisp displays are obtainableâ€”80x24 character displays were available in the Model II, Model 4, and later systems. The Model I has no built-in speaker. Square wave tones can be produced by outputting data to the cassette port and plugging an amplifier into the cassette "Mic" line. Most games used this ability for sound effects. An adapter was available to use Atari joysticks. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. July Learn how and when to remove this template message Cassette tape drive[edit] User data was originally stored on cassette tape. To find the correct volume at first use, the load was started and the volume was adjusted until the TRS picked up the data. Then it was halted to rewind the tape and restart the load. Users were instructed to save multiple copies of a software program file, especially if audio tape cassettes instead of certified data tape was used. A special program was loaded using the conventional tape interface. Then the radio broadcast was connected to the cassette tape interface. This helped the situation, but tape operation was still unreliable. Some programmers wrote machine-language programs that would increase the speed to up to 2, bits per second without a loss of reliability on their tape recorders. With the Model III and improved electronics in the cassette interface, the standard speed was increased to 1, baud that worked quite reliably on most tape recorders. For loading and storing data from tape, the CPU created the sound by switching the

output voltage between three states, creating crude sine wave audio. The first version of the Model I also had a hardware problem that complicated loading programs from cassette recorders. Tandy offered a small board which was installed at a service center to correct the issue. The ROMs in later models were modified to correct this. Model I Expansion interface[edit] Only the Model I used an Expansion interface; all later models had everything integrated in the same housing. It went through several revisions. This demanded periodic cleaning with a pencil eraser in order to avoid spontaneous reboots, which contributed to its "Trash" sobriquet. Aftermarket connectors plated with gold solved this problem permanently. Software developers also responded by devising a recovery method which became a standard feature of many commercial programs. They accepted an "asterisk parameter", an asterisk star character typed following the program name when the program is run from the TRSDOS Ready prompt. When used following a spontaneous reboot or an accidental reset or program crash , the program loaded without initializing its data area s , preserving any program data present from the pre-reboot session. The power button on the EI was also somewhat difficult to operate as it was recessed so as to guard against the user accidentally hitting it and turning it off while in use. A pencil eraser or similar object would be used to depress the power button and the EI had no power LED, making it difficult to determine if it was running or not. The expansion unit required a second power supply, identical to the base unit power supply. An interior recess held both supplies. The user was instructed to power on and power off all peripherals in proper order to avoid corrupting data or potentially damaging hardware components. The manuals for the TRS stipulated turning on the monitor first, then any peripherals attached to the EI if multiple disk drives were attached, the last drive on the chain was to be powered on first and work down from there , the EI, and the computer last. When powering down, the computer was to be turned off first, followed by the monitor, EI, and peripherals. Due to the above-mentioned problems with potentially corrupting disks, it was recommended to power up to the garbage screen with the disk drives empty, insert a system disk, and then hit Reset. It could accommodate the complete computer system plus up to four floppy drives and the Quick Printer. Since the cable connecting the expansion interface carried the system bus, it was kept short about 6 inches. The user had no choice but to place it directly behind the computer with the monitor on top of it. This caused problems for a non-Tandy monitor whose case did not fit the mounting holes. Also, the friction fit of the edge connector on the already short interconnect cable made it possible to disconnect the system bus from the CPU if either unit was inadvertently moved during operation. Floppy disk drives[edit] Radio Shack introduced floppy drives in July , about six months after the Model I went on sale. The first version released to the public was a buggy v2. The industry standard Shugart Associates SA minifloppy disk drive was used. Four floppy drives could be daisy-chained to the Model I. The last drive in the chain was supposed to have a termination resistor installed but often it was not needed as it was integrated into later cables. A common method of handling the delay was to issue a command to the , perform several "NOP" instructions, then query the for the result. Early TRS-DOS neglected the required yet undocumented wait period, and thus false status was often returned to the OS, generating random errors and crashes. Once the delay was implemented, it was fairly reliable. The disk drives offered on the Model I were Shugart SAs which supported 35 tracks and was the sole 5. By , other manufacturers began offering drives. The combination of 40 tracks and double-density gave a capacity of kilobytes per single-sided floppy disk. The use of index-sync meant that a " flippy disk " required a second index hole and write-enable notch. One could purchase factory-made "flippies". Some software publishers formatted one side for Apple systems and the other for the TRS The usual method of connecting floppy drives involved setting the drive letter via jumper blocks on the drive controller board, however Tandy opted for a slightly more user-friendly technique where all four select pins on the drives were jumpered and the ribbon cable was missing the Drive Select line, thus the user did not need to worry about moving jumpers around depending on which position on the chain a drive was in. A standard flat floppy ribbon cable was perfectly usable on the Model I, in which case the drives would have to be jumpered to their number on the chain, or even an IBM PC "twist" cable, which required setting each drive number to 1, but would only permit two drives on the chain. Although third party DOSes allowed the user to define virtually any floppy format he wanted, the "lowest common denominator" format for TRSs was the baseline single density, single sided, track format of the Model I. These new drives were all half-height and

therefore required different or modified drive housings. Exatron Stringy Floppy[edit] An alternative to cassette tape and floppy disk storage was provided by a company called Exatron. The device was a continuous loop tape drive, dubbed the "stringy floppy" or ESF. Exatron tape cartridges stored over 64 kB of data. It was popular with TRS enthusiasts, selling over units by Up to four hard disks could be daisychained for 20 MB of storage. The LDOS operating system by Logical Systems was bundled, which provided utilities for managing the storage space and flexible backup. Later, a 15MB hard disk was offered in a white case, which could be daisychained for up to 60 MB. Like most hard disks used on 8-bit machines, there was no provision for subdirectories, but the DiskDISK utility was a useful alternative that could create virtual hard disk ". DSK" files that could be mounted as another disk drive, and used like a subdirectory would. Printers[edit] The "Quick Printer", [49] was an unusual electrostatic rotary printer that scanned the video memory through the bus connector, and printed an image of the screen onto aluminum-coated paper in about one second. Unfortunately, it was incompatible with both the final, buffered version of the expansion interface, and with the "heartbeat" interrupt used for the real-time clock under Disk BASIC. This could be overcome by using special cabling, and by doing a "dummy" write to the cassette port while triggering the printer. It had only 7 pins, so letters with descenders such as lowercase "g" did not reach under the baseline, but were elevated within the normal line.

Chapter 3 : TRS â€“ Books | Ira Goldklang's TRS Revived Site

Note: Citations are based on reference standards. However, formatting rules can vary widely between applications and fields of interest or study. The specific requirements or preferences of your reviewing publisher, classroom teacher, institution or organization should be applied.

The SAM is a multifunction device that performs the following functions: Its main purpose is to control the DRAM but, as outlined above, it integrates several other functions as well. It is connected to a crystal at 4 times the television colorburst frequency This is divided by 4 internally and is fed to the VDG for its own internal timing 3. Switching the SAM into 1. As such, the display shows garbage; this mode was seldom used. Writing even bytes sets that register bit to 0, Writing to odd bytes sets it to 1. However, this is not required and results in the creation of some new display modes not possible when the VDG is used in a system alone. There are two versions of the SAM. There are some minor timing differences, but the major difference is the support of an 8-bit refresh counter in the version. Some third party bank-switching memory upgrades that used K DRAMs needed this 8-bit refresh counter to work. VDG[edit] Power-on screen of a CoCo 2 The MC is display generator capable of displaying text and graphics contained within a roughly square display matrix pixels wide by lines high. It can display 9 colors: In alphanumeric mode, each character is a 5 dot wide by 7 dot high character in a box 8 dots wide and 12 lines high. This display mode consumes bytes of memory and is a 32 character wide screen with 16 lines. The internal ROM character generator only holds 64 characters, so no lower case characters are provided. Lower case characters were rendered as upper case characters with inverted color. Although simulated screen shots would show this as green on black, on most CoCo generations it was actually green on very dark green. Semigraphics is a hybrid display mode where alphanumerics and chunky block graphics can be mixed together on the same screen. If the 8th bit of the character is set, it is a semigraphics character. If cleared, it is an alphanumeric. When the 8th bit is set, the next three bits determine the color and last 4 bits determine which "quadrant" of the character box is either the selected color or black. This is the only mode where it is possible without sneaky tricks to display all 9 colors on the screen simultaneously. The alphanumeric display has two colorsets. The one used by default on the CoCo has black characters on a green background. The alternate has black characters on an orange background. The colorset selection does not affect semigraphics characters. The border in this mode is always black. In this mode only 4 colors are possible but the Colorset bit of the VDG can select two different groups of the 4 colors. Due to a peculiarity of its hardware, only two colors are available in graphics blocks when using Semigraphics 6 on the CoCo. This creates the "virtual" modes Semigraphics 8, 12, and These modes were an interesting curiosity but not widely used, as the Semigraphics screen consumed bytes of memory. These modes were not implemented on the CoCo 3. CoCo enthusiasts created experimental programs to try to reverse engineer the modes, and were able to reconstruct the missing documentation. In resolution modes, each pixel is addressable as either on or off. There are two colorsets available, the first was black dots on a green background and green border, the second, more commonly used one has white dots on a black background with a white border. In color modes, each pixel was two bits, selecting one of four colors. Again the colorset input to the VDG determined which colors were used. The first colorset has a green border, and the colors green, yellow, red, and blue were available. The second colorset has a white border and the colors white, cyan, magenta and orange were available. The maximum size of a graphics screen is bytes. It is not possible to reliably display dots across the screen due to the limitations of the NTSC signal and the phase relationship between the VDG clock and colorburst frequency. In the first colorset, where green and black dots are available, alternating columns of green and black are not distinct and appear as a muddy green color. However, when one switches to the white and black colorset, instead of a muddy gray as expected, the result is either orange or blue. Reversing the order of the alternating dots will give the opposite color. Most CoCo games used this mode as the colors available are more useful than the ones provided in the hardware 4 color modes. Unfortunately the VDG internally can power up on either the rising or falling edge of the clock, so the bit patterns that represent orange and blue are not predictable. Most CoCo games would start up with a title screen and invited the user

to press the reset button until the colors were correct. The CoCo 3 fixed the clock-edge problem so it was always the same; a user would hold the F1 key during reset to choose the other color set. Users in PAL countries saw green and purple stripes instead of solid red and blue colors. Thus, the entire set of Extended Color Basic graphics commands could be used with the artifact colors. Some users went on to develop a set of 16 artifact colors[how? Use of POKE commands also made these colors available to the graphics commands, although the colors had to be drawn one horizontal line at a time. Some interesting artworks were produced from these effects, especially since the CoCo Max art package provided them in its palette of colors. Lower case and the T1[edit] Sample character set display of T1 VDG in true lowercase mode The is capable of using an external character generator. Several third party add-on adapter boards would allow the CoCo to display real lowercase characters. Its other changes were mainly to reduce parts count by incorporating an internal data latch. The lower case capability of this VDG is not enabled by default on this system and is not even mentioned in the manual. Only through some tinkering and research was this feature discovered by intrepid CoCo users. The T1 may also carry the part number XCP; these may have been pre-release parts. The earliest CoCo models had two standard chips. Functionally the and are identical and one can put a in place of the if that part is bad. Some external pull-up resistors may be needed to use a to replace a in a CoCo for normal keyboard operation. This means that precise software controlled timing loops are easily implemented. This reduces hardware cost, but reduces system performance as the MPU is unavailable during these operations. They must wait until an error occurs or all the data needed is read. CoCo 3 hardware changes[edit] The hardware in the CoCo 1 and CoCo 2 models was functionally the same; the only differences were in packaging and integration of some functions into small ASICs. The CoCo 3 radically changed this. Aside from the graphics enhancements outlined above, the CoCo 3 offered true lower case, 40 and 80 column text display capability, and the ability to run at 1. While the CoCo sported perhaps the most advanced 8-bit processor ever made, that processing power came at a significant price premium. In order to sell the CoCo at a competitive price, its relatively expensive processor was not tied to any specialized video or sound hardware. In comparison, the derived processor in the Commodore, Apple and Atari systems was much less expensive. Both Commodore and Atari had invested in advanced graphics and sound chip designed for arcade games and home game consoles. By tying these specialized circuits with an inexpensive processor, Atari and Commodore systems were able to play sophisticated games with high quality graphics and sound. The trade-off is between a system with an expensive CPU that does a lot of work, or an inexpensive CPU that controls the registers of its sound and video hardware. The CoCo video hardware was derived from a chip designed as display for a character based terminal, and is a completely "dumb" device. Similarly, the sound hardware is little more than a 6-bit DAC under software control. All graphics and sound require direct CPU intervention, and while this allows for great flexibility, its performance is much lower than dedicated hardware. Games drove system sales then as they do now, and with its poor gameplay performance, the CoCo attracted little interest in officially licensed ports of popular games. The CoCo 3 did improve graphics capability and doubled CPU performance, but still contained no hardware graphics or sound acceleration. Drawing was performed by the CPU, and most of the new graphics modes required at least twice as much processor time due to increased display resolution and color depth. The sound hardware was not changed at all. Every computer platform is a compromise, and despite the significant graphics and sound handicap the CoCo may have had, it still had a sophisticated CPU under its hood with extremely high performance. There were many independent clones of popular games available, but far more important was the availability of " killer apps " for the CoCo. Even the BASIC interpreter was one of the most powerful available, and provided the user with a rich set of easy-to-use commands for manipulating on-screen graphics and playing sounds. Some of the hardware limitations were overcome with external add-ons, particularly expansion cartridges. Some were made by Tandy, some by other manufacturers. OS-9 applications communicate with OS-9 and its drivers. This allows for a degree of hardware independence. In order to support such programs or at least, those that bypassed BASIC and addressed hardware directly , any future CoCo version would have to be hardware-compatible with the CoCo, or perfectly emulate every aspect of the CoCo. In contrast, OS-9 programs relied only on OS-9 functions, and its drivers could be rewritten to work with different hardware. OS-9 was an additional product that had to be

loaded manually each time the computer was started. Writing an OS-9 program meant appealing to a smaller subset of the CoCo community; this discouraged development of OS-9 products. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. December Learn how and when to remove this template message On October 26, , Ed Juge of Tandy announced that the CoCo 3 would be dropped from its computer line. With no apparent successor mentioned, the announcement was disheartening to many[who? To this day, current and former CoCo owners agree [21] that Tandy did not take the CoCo very seriously, despite it having been their best-selling computer for several years. Radio Shack fliers and stores typically depicted the CoCo 3 running CoCo 2 games, and offered a very limited selection of CoCo 3 specific software.

Chapter 4 : TRS Model - Wikipedia

Tandy/Radio Shack Book: TRS Data File Programming In BASIC ()(Leroy Finkel) Skip to main content Search the history of over billion web pages on the Internet.

The following are a few exceptions. KEY where these keys are disabled. The following are some of the major differences. The following conditions apply to both keyboard modes. NumLock has no effect in the emulator, the number pad always sends numbers. The cap-lock light does not reflect the actual cap-lock state within the emulator. Operation - The emulator display screen: The emulator operates in xx16 VGA mode. At the bottom of the display screen is a status line. The four boxes on the left represent the 4 floppy drive select lights. Between drive lights 1 and 2 is the track number of the currently selected drive. Following the drives lights is where is cassette counter is displayed. The available speeds are 1, 2 or 1. The turbo speed is represented by a question mark. If the MHZ messages is in reverse-video the the speed is locked. The four boxes on the left represent the 4 hard drive access lights. Between drive lights 1 and 2 is the cylinder number the hard drive is currently accessing. When the CPU is in single step mode the section of the status line following the speed is the Z80 register display. When in model 4 mode the status line is within the TRS display area. This status line is in background to prevent it interfering with data on the TRS display. If the Status line does interfere however it can be turned off in the configuration screen. Operation - The virtual floppy disk selection screens: As documented in a preceding chapter virtual floppies may be entered on the command line at startup. Since the user can only enter the 4 virtual floppies to be assigned to the 4 emulated drives a way to remove, insert and create new virtual floppies from within the emulator was needed. The virtual disk selection screens are for that purpose. In fact these screens eliminate the need for the command line entry method entirely. To bring up the first of the selection screens push the F9 key. The emulator display screen will be replaced by the virtual drive status screen. Here is a table of the functions available once the drive status screen is displayed: F10 Re-boot and return to the emulator. This function is most often used when the disk in drive 0 is changed in order to boot the new system disk upon return to the emulator the user can also use F9 to return to the emulator and then push F10 to re-boot. Enter Insert or change the virtual floppy inserted in the currently selected virtual drive. This key will bring up the virtual disk selection screen. Insert Insert or change the virtual floppy inserted in the currently selected virtual drive. This will bring up a box where the user can enter the virtual disk file to be mounted. By default the file will be loaded from the current default drive path shown when selecting virtual disks , with a default extension of. Delete Remove the virtual floppy from the currently selected virtual drive this function does not delete the virtual floppy file. Home Toggle the write protect status of the virtual floppy in the currently selected virtual drive. End Toggle the writing of Model 1 compatible address marks when writing in single density. Here is a table of the functions available once the floppy disk selection screen is displayed: Arrow keys Change which filename is highlighted. Enter, JV[1] or JV[3] If a virtual floppy filename is highlighted then that disk will be mounted and the drive status screen will return showing the new floppy mounted or an error message. If [1] is pushed the emulator will assume the virtual floppy is in JV1 format pushing [1] is the only way to open a JV1 type virtual disk If [3] is pushed the emulator will assume the virtual floppy is in JV3 format. After pushing [C] the user will be prompted for a filename. The file will be created in the currently displayed directory. The default file extension is. DSK but the user can override this. After entering a valid filename the user will be prompted for 2 more parameters. The default is double. If single is selected the floppy can only be formatted single density. This option is to save space only. If single is selected the floppy can only be formatted single sided. Pushing [Enter] for the above prompts will accept the default. Pushing [Escape] at any of the above prompts will abort the creation process. Then that disk will be mounted and the drive status screen will return showing the new floppy mounted or an error message if the file was unable to be created. If the file does exist then the user will be prompted whether the user wishes to overwrite it. If the user answers yes the file will be created as above, all data currently on the virtual floppy will be lost. After pushing [D] the users will be prompted to enter a new drive letter. Enter a letter A-Z or nothing to leave the drive unchanged. After pushing [S] the user will be

prompted to enter a new search file spec. Enter nothing to leave the search file spec unchanged. Be careful not to select non-virtual floppy files to mount, doing so could cause the file selected to be corrupted. Type JV1 virtual disk files must be mounted by pushing [1] when selecting. This type cannot be determined automatically. The current path and file spec being displayed is shown at the top of the screen. Virtual floppies cannot be deleted from within the emulator. Operation - The virtual hard disk selection screens: The emulators support hard drives in two ways. The second method is by emulating the actual hardware of a WD hard disk controller. The second method is new for v5. Method one will be discussed in a later chapter. Emulating the actual hardware allows the emulator to use the same hard disk drivers that are used on an actual TRS, no special drivers need to be written. Virtual disks selected on these screens are accessed with standard TRS hard disk drivers. The emulators can support up to 4 hard drives. Each drive can be up to cylinders and 8 heads. This allows support of up to a 67meg hard drive. Most DOSs cannot support a single drive this large but support partitioning of a single hard drive into smaller logical drives. The emulator however can change hard drives on the fly just like floppy disks. This allows creating many virtual hard drives each with its own DOS. Further, since LS-DOS can only store files on a hard drive, and it is doubtful that files would require 15megs of space, smaller hard drives would probably be more desirable. Since the user can only have 4 virtual hard disks mounted at a time 2 with some drivers a way to remove, insert and create new virtual hard disks from within the emulator was needed. The virtual hard disk selection screens are for that purpose. To bring up the first of the selection screens push the Shift-F9 key. Enter Insert or change the virtual hard disk inserted in the currently selected virtual drive. Insert Insert or change the virtual hard disk inserted in the currently selected virtual drive. Delete Remove the virtual disk from the currently selected virtual drive this function does not delete the virtual hard disk file. Home Toggle the write protect status of the virtual hard disk in the currently selected virtual drive. If the user pushes [Enter] to insert a virtual floppy in a drive the drive status screen will be replaced by the floppy disk selection screen. Enter If a virtual hard disk filename is highlighted then that disk will be mounted and the drive status screen will return showing the new floppy mounted or an error message. A virtual hard drive can have 1 to 8 heads. This parameter and that of the driver not matching will result in wasted space or an unreadable hard disk. A virtual hard drive can have 10 to cylinders. This value limits the size of the virtual hard drive and prevents the file from expanding beyond the desired size. Pushing [Enter] for the above prompts without entering a value will abort the creation process. Then that disk will be mounted and the drive status screen will return showing the new disk mounted or an error message if the file was unable to be created. If the user answers yes the file will be created as above, all data currently on the virtual disk will be lost. Virtual hard disks cannot be deleted from within the emulator. Operation - The virtual cassette control screen: Support is at the hardware level so any program that reads or writes to cassette should be supported.

Chapter 5 : TRS Color Computer - Wikipedia

Programming Enter your mobile number or email address below and we'll send you a link to download the free Kindle App. Then you can start reading Kindle books on your smartphone, tablet, or computer - no Kindle device required.

It will also come up automatically when a breakpoint is hit. They also can be set on the command line by the -b option. The debugger will also activate if the Emulator Extensions are used and the Z tries to access a protected area of memory. All values are displayed in hexadecimal. There is also a view of the top of the stack and a T-state counter which can be changed as desired to measure intervals interactively. The Step button move execution forward a single instruction. Step Over sets a breakpoint after the current instruction and resumes execution. This is useful for CALLs to run quickly though a subroutine. The "Bus Fault Enabled" checkbox may be turned off to disable protection checks. When single stepping the display will turn gray to give an indication where the CRT beam is at that moment in execution. The debugger is still operational when the TRS is running. Since the screen shows the contents of the previous frame and the drawing of the current frame you will not usually see an immediate change when writing to screen memory. It only shows up when the CRT beam reads and draws it. The debugger memory view gives you the ability to see immediate changes to the various different RAM systems. You can also select just the RAM to focus on the 48K or memory. But keep in mind these other view uses their own addressing. In a clunky way RAM can be changed. The easiest approach to to select a memory byte and write a new hexadecimal value for it. The emulator simply reads back the memory dump so you can also delete a line and enter any address followed by a colon and a series of space-separated hexadecimal bytes to change memory locations without having to look at them. A few pseudo-memory regions are viewable but not changeable. They are intended to give a partial view of the TRS hardware state. Z Port Writes The last value written by the Z to a port. At the bottom of the window are line of check boxes and drop-downs to control bus tracing which is dicussed later. Source Level Debugging My zmac cross assembler will output machine language programs in. It is a text format so by looking at it and the zmac source code you can probably figure out how to generate it yourself. But the important part here is that loading. It will look a bit like an assembler listing file. The current program location will be highlighted and follow the execution of the Z The format also defines symbolic labels so you can type these labels in to the breakpoint or register windows instead of having to look up the hexadecimal values yourself. You can also use labels for the -b command line option to set breakpoints. Advanced Recording Sometimes examining memory in the debugger is too cumbersome. The normal recording options can assist debugging. It may be helpful to step through a video a frame at a time to see some graphical glitch in detail. The "MHz Audio" option takes this to the extreme by recording audio output a sample rate equal to the speed of the Z In effect this lets you see exactly when the audio changes. The other options attempt to self-document in their output. When they do their final output the use whatever value is in RAM at the time for the disassembly. If the program changes you may seen confusing output. This gets even worse if the memory mapping changes. All these recording options can be activated and stopped at any time. It is useful and often desirable to start them when the program is stopped in the debugger and then stop them at the next breakpoint after an interesting subroutine or full step of a game simulation has run. It is intended to help measure where your program spends its time to be used as a guide for optimization. It can also be used to simply track what a program as done during an interval. However, "Bus Use" is better for that task and Trace will show every instruction in order. In theory you can use this to respond to a crash. The output is much like a disassembler but with markup indicating how memory was accessed: The intent here is to distinguish the loaded program from the ROM or operating system routines it uses. If the program is sufficiently put through its paces the result should be a good disassembly that can be assembled to produce the original code. However, this is a problem if the program relocates itself. At least "Bus Use" will help understand the relocator code. Emulator Extensions These are enabled by the -ee command line option. Here is a brief overview: Function is typically used to end a test in batch mode. The bus permissions are very helpful in tracking down nasty bugs. For example, you can set your code section to execute-only. The emulator will trap into the debugger the instant something tries to overwrite over your code.

Or even read it. Another useful technique is turning off stack permissions at the bottom and top of your stack to detect stack overflow or underflow. For function 0 the lower 6 bits in B are set to indicate what Z operation is allowed on that memory location. It also places markers in the output file to indicate when a frame has ended and when one second of execution has finished. It can be activated a program start with the -trace option. The full log is recorded in the output file. The output can be voluminous. The "Tracing" checkpoint in the Z Debugger is a convenient shortcut. This is handled by the device drop-down. The interface is awkward. As you select each device in the drop-down the checkmark to the right changes to indicate if that device is being logged. Obviously these controls should be in some other window but the debugger happened to be handy at the time. The actual logging looks something like this: The second is the PC of the Z when the operation occurred. The type of access follows. Most are "ex" for instruction execution with a disassembly of the instruction following. Other possible operations are: Some devices are very simple in that any byte read or written can only have one meaning. But for the CRTC a write to a register depends on which register was previously selected. Without the annotation you would have to search backwards for the last register selection. Not all devices provide annotations. If they do then you can bet they were giving us trouble in developing the emulator. Most of the Model 2 devices have annotations. By the way, the underscore and signs in front of addresses are intentional and useful. But searching the word ffb2 will find instructions that reference the address. Or you can search for ffb2 explicitly to restrict your search to only instructions executed at that address. George Phillips, August 27,

Chapter 6 : Back to the future: the TRS Model | Ars Technica

TRS Revived Site is an archive of everything related to the Tandy Radio Shack TRS microcomputer lines. Site contains emulators, programs, manuals, books, patches, games, hints, discussions, questions, forums, and tons more.

It was written by Gordon Letwin and was among the first four products sold by Microsoft Consumer Products, a new division created to sell software to consumers. It had unusually high system requirements for a game at the time; both the Model I and Apple II versions required 32K and a floppy drive. Adventure was the first computer adventure game and became very popular on mainframe installations in and Only Microsoft has it all. Adventure fills an entire disk with everything you need for your exploration. But in an odd omission, the manual never mentions Will Crowther and Don Woods, who were the authors of the original Adventure. Crowther was an experienced cave explorer also known as caving or spelunking. He set Adventure in with some creative license the Colossal Cave that is part of the Mammoth Cave system in Kentucky 1. The goal is to use simple text commands to explore the cave, solve puzzles, and collect treasure. It began with this introduction: Joseph Weizenbaum with Eliza in Crowther stopped developing Adventure sometime in In , Don Woods, having played a copy of Adventure at Stanford University, wanted to further develop the game. Woods doubled the size of the Adventure data file with longer descriptions and more rooms. He added more treasures, improved scorekeeping, and created the concept of a player inventory. He also put in this message: It was inevitable that programmers would write Adventure adaptations for these new computers. Microsoft Adventure, written by Gordon Letwin through his company Softwin Associates 3 , was the first commercial adaptation to implement the complete Adventure 4. He reimplemented Adventure in assembly language and managed to fit it in 32K, which was quite an accomplishment. Unfortunately, these two sessions had to be saved to the original program disk. If after repeated efforts, you still cannot solve the problem, you can send for one of four pamphlets available from Softwin Associates, implementers of the game, that will help you through specific parts of the cave. The game setting is a vast network of caves beneath the earth and the land outside. The fantasy world contains rooms or nodes, 15 treasures, 40 useful objects and 12 problems to solve. The program allows players to store the status of two games on a diskette. Microsoft ended official support for Microsoft Adventure in Software Toolworks sold a modified version of this conversion in as The Original Adventure. This was the only version of Adventure that paid a royalty to Crowther and Woods. Many programmers at the time worked through their own companies, even when also working elsewhere.

Chapter 7 : Copying Files To and From TRS Model II TRSDOS â€“ PSKI Software Development

The TRS Micro Computer System (TRS, later renamed the Model I to distinguish it from successors) is a desktop microcomputer launched in and sold by Tandy Corporation through their Radio Shack stores.

Since most of the files handled by Z80Asm will actually reside on a Windows PC, this instruction manual uses the Windows style of naming except in a few cases where the files reside on a TRS virtual disk. Installation The Z80Asm assembler is a single executable file that runs from the command line. It does not have or need an installation program; just copy it into the directory where you want to use it, or copy it into a binaries directory and use a PATH statement to make it visible to the system. The command line looks like this: LST, which is a listing of the assembled program. Please note that because different people use different extensions for their assembly language source files ASM, SRC, etc. If you want to assemble FILE. This default behavior will probably be enough in most circumstances, but Z80Asm features the following list of options to serve your needs. Output an Intel-style HEX file. This setting is only available in the registered assembler. Output a core image file. Disable the display of false conditionals. You can keep your program listings shorter and neater by only including true conditionals. Disable the generation of a CMD header record. Disable display of macro expansions. Suppress display of local labels. If you use the dollar sign to start all your local labels and then specify the -sl option when you assemble your source code, local labels will not be included in the symbol table listing. This will make that listing much less cluttered and much easier to follow and understand. Disable generation of an output file. You might just want to check your source file for errors, or generate a program listing but no program file. Disable generation of a listing file. Z80Asm creates a program listing by default, but you may not want or need one. Enable support of Hitachi HD instructions. One of the more popular TRS add-ons in its later years was the XLR8er board, which provided additional memory and a faster and more capable Hitachi HD microprocessor. Enable generation of an equate listing. An equate listing is a list of all the equate values that were used in a source file; if you have ever looked at The Source, the published program listings for the TRSDOS 6 operating system, you know what one looks like. Set the include directory. By default Z80Asm looks for include files in the current directory. You may want it to look somewhere else to keep your subdirectory structure neater. Set a virtual disk as input. Please note that the virtual disk cannot currently be in use by another program â€” for example, mounted in a drive on a currently-running TRS emulator. Set a virtual disk as output. Again please note that the virtual disk cannot currently be in use by another program. Rename the output file. Use this option to change the name of the output files. Here are some examples of possible Z80Asm command lines. ASM source file, assemble it with support for the Hitachi HD instructions, generate an equate listing of non-alphanumeric equates, and write out the XLR8. Each line of the program looks something like this, but keep in mind that each of the four fields is optional: You can make each field line up in columns or not; your programming style is completely up to you. Z80Asm treats upper and lower case letters as identical except inside character strings , so you can decide whether to program in upper or lower case. The label is a symbolic name for a memory address. The label can be from two to fifteen characters long. Labels may be terminated by a colon, a space, a tab, or any combination of the three. The first character of a label must be a letter, an underline, a dollar sign, or an at sign. The subsequent characters of a label may be any of these characters as well as a digit 0 through 9 or a question mark. Fields 2 and 3: The comment always begins with a semi-colon and ends with the end of the line. Expressions As can most other assemblers, Z80Asm can evaluate expressions as part of the source code. The expressions can include simple arithmetic: Logical bitwise AND the word. Z80Asm handles all of the familiar pseudo-ops, including but not limited to: ORG specifies the location in memory where the program will start assembling. END marks the end of a program file and optionally specifies the location of the start of the program. EQU sets a label to a specified value. DB inserts a single byte or a string of bytes or characters into the program. DATE inserts the system date into your program as an eight-character string. TIME inserts the system time into your program as an eight-character string. DC inserts the specified number of byte constants into the program. Assembler Directives Assembler directives are commands embedded in the program that tell the assembler to do

something different. Z80Asm supports several directives: Include files are very useful for loading macro definitions or for breaking a larger program up into multiple files. Z80Asm normally uses a radix of ten, which means that if your source file includes the instruction LD A,10, the decimal value ten will be loaded into the A register. You can set the radix anywhere from two to sixteen. If the radix were set to sixteen, the instruction LD A,10 would load the hexadecimal base sixteen value of 10H â€" 16 in decimal â€" into the A register. Just as a suggestion, it is a good idea to leave the radix at ten and explicitly specify numbers in other bases. Parameters are not required but there may be as many parameters as will fit on a line. Each parameter should be preceeded by a number sign or. Conditional statements can be used inside macros to make them a little more flexible. Z80Asm also supports the three standard built-in macros. The first is REPT number, which will repeat the macro definition a specified number of times.

Chapter 8 : Ira Goldklang's TRS Revived Site: Documentation - Books

The TRS Model is a portable computer introduced in It is one of the first notebook-style computers, featuring a keyboard and liquid crystal display, in a battery-powered package roughly the size and shape of a notepad or large book.

The TRS Model , the "original laptop. The 8-line, 40 column reflective LCD display no backlight is great for outdoor use and is extremely low-power. Files you create get a. DO extension and get stored in RAM for you to work with from the main menu screen. A little light HTML editing. The bottom of the Model provides access to the ROM sockets and battery compartment. This ROM chip holds the operating system and basic applications. This ROM has been rocking since A second ROM socket is available for add-on applications. All this computing power is driven by 4 AA batteries. All the peripheral ports anyone could ever want: This smaller connector is for a barcode reading wand. A while ago, as we at Ars were discussing our first mobile computing experiences, I recalled the first "laptop" computer I was ever issued to use as a journalist: Back on my first field assignment for Government Computer News, I was given the only computing device in the company inventory that could be used to file a story from the fieldâ€”via acoustic couplers and an MCI Mail account. Overcome with nostalgia, I put a bid in on a Model on an eBay auction shortly afterwardâ€”and then completely forgot about it. Or I forgot about it until about a week and a half ago, when I saw that I had the winning bid. The Model was the future of mobile computing when it arrived on the market in Bill Gates told the National Museum of American History in an interview that the Model was "in a sense my favorite machine. Powered by the 2. The core software for the Model is stored on a ROM chip. It includes a BASIC interpreter, a text editor, a "Telecom" program for remote connections, and an address book application that stores phone numbers for dialing up connections via the built-in baud modem. If you needed longterm storage or to free up some of the RAM , files could be written to a cassette tape drive or transferred to another computerâ€”either by modem or serial connection. Radio Shack offered a portable 3. So to get the Model online again may end up pushing the final price of my purchase a bit closer to its price tag. Your suggestions on what to attempt with this particular piece of our computing heritage are welcome, as always. Listing image by Sean Gallagher Promoted Comments.

Chapter 9 : TRS data file programming | Open Library

If a file doesn't seem to do anything when loaded, or won't load, it's probably an auxiliary program which is normally called by a larger one. The NewDos/80 directory can be seen by typing DIR 0.