megatel

THE MEGATEL QUARK®/100 SINGLE BOARD COMPUTER

ENTIRE CONTENTS COPYRIGHT
MEGATEL COMPUTER TECHNOLOGIES
150 TURBINE DRIVE
WESTON, ONTARIO, CANADA M9L 2S2

MA-223010-R0

TABLE OF CONTENTS

SOFTWARE — **S/W PAGE NUMBER**

APPENDIX — **APPENDIX PAGE NUMBER**

Limited Warranty

All products sold (including software) are under LIMITED WARRANTY on a return to factory basis against defects in workmanship and material for a period of 90 days from the date of shipment.

WARRANTY is contingent upon proper use of the Product. WARRANTY will not apply if any repairs are necessary due to accident, unusual physical, electrical or electro-magnetic stress, neglect, misuse, or causes other than ordinary use. WARRANTY will also not apply if the product has been modified by BUYER, or if the product has been disassembled by BUYER. Disassembly includes the removal of the serial number label on the Product without prior written approval from MEGATEL.

Customer attempted repairs will void the WARRANTY. Any tips suggested in the manual which involve physical changes to the board or a reconfiguration of the software, if attempted, will void the WARRANTY.

Copyright Notice and Trademarks

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Megatel Computer Technologies.

Entire contents copyright Megatel Computer Technologies, Toronto, Canada 1985. Printed in Canada.

"Megatel Quark" is a registered Canadian trademark of Megatel Computer Technologies.

"CP/M" is a registered trademark of Digital Research, Inc.
"Digital Research", "CP/M Plus", "DDT", "LIB-80", "LINK-80", "ASM", "MAC", "MP/M II", "RMAC", and "SID" are registered trademarks of Digital Research, Inc.
"Microsoft" is a registered trademark of Microsoft Corporation.
"Z80" is a registered trademark of Zilog Inc.
"PAL" is a registered trademark of Monolithic Memories Inc.
"Apple" and "Apple III" are registered trademarks of Apple Computer Inc.
"Corvus OMNINET" is a trademark of Corvus Systems, Inc.
"IBM" is a tradename of International Business Machines, Inc.
"Centronics" is a registered trademark of Centronics Data Computer Corporation.

Disclaimer

Megatel makes no representations nor warranties with respect to any circuitry not embodied within a Megatel product. Megatel specifically disclaims any implied warranties of merchantability or fitness for any particular purpose of any Megatel product. Megatel assumes no responsibility for errors which may appear in this document, and reserves the right to revise this document without notice.

Service Information

Should a QUARK board require service please contact the Megatel Engineering office at 416-745-7214 for instructions.

Introduction to the QUARK family of single board microcomputers

The QUARK family of microcomputer products comprises a number of high-performance single board computers and support software, all designed and manufactured by Megatel. Quarks are designed to provide all the working functions of a basic computer on a single, compact board for the end user, computer OEM and controller markets. The QUARK hardware and software systems are created as flexible and complete components which allow their users to skip most of the expensive and time-consuming steps in their systems design, programming and testing.

QUARK hardware is designed to minimize space requirements and parts count, while simultaneously maximizing processing power, speed, and flexibility. The architecture of the QUARK/100 serves as the "template" for the entire family. All of the QUARKs share its wide range of CPU, memory, video, disk and I/O features. The QUARK/100 is available in 3 models, 64K, 128K or 256K of RAM. The QUARK/150 adds RGB color capability to the QUARK/100's list of features. The QUARK/200 includes an intelligent local area network interface as well as every feature of the QUARK/100. The QUARK/300 extends the QUARK/100's floppy disk capability by providing direct control of ST-506-type Winchester hard disk drives. The Quark/400 combines all of the features of the forementioned boards and adds real a time clock with battery backup, EPROM and EEPROM support, yet you need choose only the features that you require. Each of the models is available with various memory and I/O options. Hardware accessories available from Megatel include connectors, and transition boards for quick connections to drives, monitors and other I/O devices.

QUARK software consists of configurable operating systems, complete device drivers, utility programs, an installation procedure, and source code on disk. QUARK software is distributed on floppy disk and in a form which allows it to boot up immediately assuming the "lowest common denominator" in user hardware.

An SBC package purchased from the factory should include the following items:

-Quark 64K SBC single board computer (or which ever model was ordered)
-QTB-3 Quark Transition Board
-Megatel's hardware and software manual which includes a QTB-3 manual
-Software on either 5.25" or 8" diskettes

Recommended drives for the Megatel QUARK

The QUARK is designed to interface easily with most 8-, 5.25-, and 3.5-inch floppy disk drives. You should be aware, however, that disk drive specifications vary widely from manufacturer to manufacturer.

Megatel has tried and tested a variety of floppy disk drives and the following is a list of drives which we recommend as being particularly well-suited for use with the QUARK.

Manufacturer	Model#	Size
Shugart	SA-300	3.5"
Shugart	SA-350	3.5"
Sony	OA-D30V	3.5"
Shugart	SA-400	5.25"
Shugart	SA-455	5.25"
Shugart	SA-465	5.25"

Manufacturer	Model#	Size
Tandon	TM100-1A	5.25"
Tandon	TM100-2	5.25"
Tandon	TM100-3	5.25"
Tandon	TM100-4	5.25"
Teac	FD-55F	5.25"
Shugart	SA850/851	8"
Tandon	TM848-2	8"

On the QTB-3 there is a 34-pin header denoted by J3. A standard 5.25 inch floppy disk cable will plug into this board in this area and connect directly to the edge connectors of your floppy disk drives. This same 34-pin header is used for connecting 8 inch drives. This can be done by taking a regular 50 pin ribbon cable and replacing the 50-pin female header with a 34-pin female header. Strip away lines 48 through 50 and 1 through 13 to give you the required 34 position cable. Either set of cable can be purchased from Megatel.

To allow a mixed 5.25 and 8 inch system Megatel can supply a cable which provides both a 34-pin and a 50-pin connector. Assembling your own mixed drive cable, although possible, is difficult due to the difference between 8 and 5.25 inch drive interfaces.

The next step, for a mixed drive system, is to wire wrap from the floppy disk data transfer rate (pin B-17 on the 96-pin DIN connector) to the drive select line (see appendix for pinout) you have chosen for your 8 inch drive. This allows the data transfer rate to switch automatically with drive selection.

Notes regarding Floppy Disk Drives:

Megatel Quarks do not provide a MOTOR ON signal. Most 5.25 inch drives will provide a jumper option on the drive motor control board to have the motor come on with the selection of the drive. Please consult your drive manual to insure this is done before you try to boot up the system.

Some 5.25 inch drives have a READY signal. If you are using such a drive, you must isolate the READY line in a mixed drive installation. If you are using an 8 and 5.25 inch drive system the READY line should be taken directly from the QTB-3 to the 8 inch drive, while bypassing the 5.25 inch drive.

Megatel Quarks do not provide a HEAD LOAD signal for 8 inch drives. Most 8 inch drives will have a HEAD LOAD with drive select jumper option on the drive motor control board. Please consult your drive manual to ensure this is done.

Recommended Monitors for the Megatel QUARK

Many direct-drive and composite monitors can be used successfully with the QUARK. The user should beware, however, that some of the cheaper composite monitors on the market could operate poorly. The most common difficulty with such monitors is their inability to display a full 80 characters. This is because they do not meet the tight timing requirements of the Quark composite signal's horizontal retrace time. These monitors can still be used if the number of characters per row is reduced through software modifications.(only on SBC's with graphics options)

Before purchasing any monitor check with the vendor to see if you will be allowed an evaluation period. This is the best way to avoid possible disappointment.

The following is a list of recommended monitors that display a full 80 characters per row:

Manufacturer	Model#	Input Type	Manufacturer	Model#	Input Type
Zenith*	ZVM-121	Composite	Electrohome	EVM 1719	Composite
Electrohome	EVM 1220	Composite	Electrohome	EVM 2319	Composite
Electrohome	EVM 1519	Composite			

*Note - The Zenith ZVM-121 is a widely excepted monitor which works well with the Megatel QUARK. The Zenith ZVM-121 is available at most retail chains and sells for approximately $100.00.

Using the Megatel QUARK with a television

Using a commercially available RF converter the Megatel QUARK can use a black and white televsion as a monitor.

As with some of the cheaper composite monitors, this method does not allow the display of a full 80 characters.

Recommended keyboards for the QUARK

Any ASCII-encoded 7- or 8-bit parallel-output keyboard with active-high DATA outputs and an active-low STROBE output will work with the QUARK.

Note that a computer terminal (or its equivalent) with a full-duplex RS-232C asynchronous serial interface capable of operating at **1200** baud can be used instead of a monitor and keyboard.

If using a, terminal simply connect it to the 25-pin D-shell connector at location J5 on the QTB-3 making sure it is set at 1200 baud. Adjacent to the 25-pin D-shell is a dual row 26-pin header at location J4. This too is the full duplex RS-232C port.

You will notice that two pairs of headers have been wire wrapped together. If you are using a terminal they may be cut only if your terminal supports the protocol involved (CTS to RTS, DTR to DSR).

Required power supply for the QUARK

The Megatel QUARK requires a regulated +5 and +12 volt supply with a common ground. These are the same voltages required by 5.25-inch floppy disk drives. The current drawn by a QUARK 128K SBC at +5 volts is 2A. The current required at +12 volts is about 100mA. If this supply must also supply power to operate the floppy disk drives, a monitor, and a keyboard, then the power requirements of these devices must be allowed for.

The color scheme for the DC power cable enclosed with the package is:

RED.....+5
BLACK.....GRD
ORANGE.....+12

The -12V required by the RS-232C drivers on the QUARK is developed by an on-card charge pump, so no negative power supply is necessary.

If a QUARK-based system fails to operate properly, the power supply connections and voltage levels in the system should be one of the first things to be checked. Be aware that reversed or incorrect polarities may cause damage to the QUARK which is not covered by Megatel's limited warranty.

The power supply chosen for a system must be able to handle normal current surges. Some QUARK users have experienced video display or CPU problems because their power supply was not able to maintain +12 and +5 volts during periods of disk drive activity. Since the QUARK is largely TTL logic, the +5V supply must never fall below +4.75V.

Megatel QUARK CPU Overview

The Megatel QUARK uses the Z80B microprocessor, manufactured by Zilog Inc. The Z80B clock frequency is 5.97MHz (6.2MHz on 50 Hz models), leading to an execution time of 667ns (645ns on 50 Hz models) for a typical 4-cycle instruction, such as a register-to-register ADD.

The main memory of the QUARK is either 64/128/256kbytes in size. A memory-management scheme employing the Z80B's I-register is used to provide simple yet flexible bank-switching to allow the full use of the extended physical memory. The Synchronous Address Multiplexer (SAM) also participates in the memory-mapping process, and permits the use of some special mapping modes.

To accomodate the Video Display Interface, the main memory of the QUARK operates as a dual-ported RAM. One of the ports is a bidirectional input/output port to the CPU, while the other port is essentially an unidirectional output-only port for the Video Display Interface. Through the programmable registers of the SAM, the size and location of the area of main memory used as the Video Display Memory can be set under software control.

As a consequence of the operation of this dual-ported memory, all Z80B memory cycles have wait states added to extend them to a multiple of 4 T-states. Thus an instruction which would execute in, say, 7 T-states if no wait states were added will have one wait state added, to extend the instruction to 8 cycles. The insertion of these wait states is fundamental to the synchronization of memory accesses by the Video Display Interface and by the CPU.

Input/output instructions are also extended to modulo-4 cycles, but then have four additional wait states added. For instance, an 11-state IN A,r type of instruction will have a total of five wait states added, extending it to 16 T-states. The addition of the extra four cycles relaxes the speed requirements for the peripheral controller devices, both those on the board as well as user-added external controllers.

CPU Memory Management

The Z80B processor used on the QUARK has a sixteen-bit address bus, and is therefore able to directly address 65536 (64k) locations in memory. However, the QUARK has 128k of memory, and therefore a process is required whereby the 64k possible addresses generated by the CPU are "mapped" into the 128k of main memory provided. In this discussion, the addresses generated by the CPU on its Address Bus will be called "logical addresses", whereas the actual locations in memory where data is stored will be called "physical addresses". Logical addresses are therefore the hexidecimal values normally used in programming.

The Z80's I-register and the Synchronous Address Multiplexer (SAM) are used in the mapping process between the logical and physical address spaces. The I-register is used to control bank-switching between bank A and bank B of the main memory, and to determine in which bank the Video Display Memory is located. The SAM determines the location of the Video Display Memory within a 64k address space, and controls the mapping of the upper and lower halves of the logical address space.

The most-significant three bits (bits 5, 6, and 7) of the I-register are used to define an 8k address boundary within the CPU's 64k logical address space. Any logical address appearing on the Address Bus is compared to the address boundary as determined by these bits. The result of this comparison determines in which memory bank the physical address is to be found.

If the value of the logical address is greater than or equal to the specified address boundary, the logical address will be mapped onto a physical address (that is, a memory location) in memory bank A. Conversely, if the value of the logical address is less than this boundary, then the logical address will be mapped onto a physical address in memory bank B. For example, if bits 5, 6, and 7 of the I-register were all zero, then the address boundary

specified would be 0000. Since all logical addresses are greater than or equal to 0000, then the CPU's entire logical address space would be mapped into the physical address space of bank A. If bits 6 and 7 were set and bit 5 were clear (I-register = C0), then logical addresses between 0000 and BFFF would be mapped onto physical addresses in bank B, while logical addresses equal to or above C000 would be mapped onto physical addresses in bank A.

To read or write the contents of the I-register the Z80's LD A,I and LD I,A instructions are used. After changing the value in the I-register, the new boundary will take effect after the next opcode fetch. Consider the following code sequence:

```
LD   A,BDYVAL       ;LOAD A WITH NEW BOUNDARY VALUE
LD   I,A            ;LOAD I WITH VALUE
RET                 ;RETURN
```

The return address used for the RET instruction will be popped off the stack in the memory bank determined by the new boundary value in the I-register, whereas the opcode for the RET instruction will be fetched from the memory bank in effect prior to loading the new value into the I-register.

Note that the use of the Z80 I-register for selecting bank-switching boundaries precludes its usual use in the Z80 Mode 2 (vectored) interrupt.

It might appear that a difficulty with this method of bank-switching is that there is no logical address which maps onto any physical address above E000 in bank B, that is, that the top 8k of bank B cannot be directly accessed by the CPU. This would seem to be the case because the highest 8k address boundary which can be specified by bits 5, 6, and 7 of the I-register is E000, so that logical addresses above E000 will always be mapped onto physical addresses in bank A, and never in bank B. This problem can be overcome through the use of the MPA TYPE and PAGE MODE bits of the SAM, as explained below.

The MAP TYPE bit in the SAM (designated "TY") allows addresses in either the upper of the lower halves of the 64k logical address space to be effectively translated into the other half of the logical address space. In the "normal" mode of operation (the mode which is initialized at power-up) the TY bit is set. In this mode, each logical address is mapped onto a unique physical address in bank A or bank B by mapping process described previously. If, however, the TY bit is cleared, then the upper and lower halves of the 64k logical address space map onto a physical address space whose size is between 32k and 64k. The actual size and location of the physical address space within the two Main Memory Banks is determined by both the bank-switching boundary set up in the I-register and by the PAGE MODE bit in the SAM (designated "P1"), as described below.

With both the MAP TYPE and the PAGE MODE bits cleared, the position of the bank-switch boundary will determine both the bank into which the logical addresses are mapped as well as whether the mapping is one-to-one or many-to-one. Consider first the simplest case, where the bank-switch boundary is 8000hex. In this special case, logical addresses less than the bank-switch boundary will be mapped into the physical address space of bank B, just as they would be if the MAP TYPE bit were set ("Normal" mode). However, logical addresses equal to or greater than the bank-switch boundary are mapped onto physical addresses in the "bottom" half of bank A. This part of the physical address space of bank A is the space into which logical address from 0 to 7FFFhex are mapped when the bank-switch boundary is set to 0000 and the MAP TYPE bit is set (normal mode). Thus logical address from 8000 to FFFFhex are essentially translated to 0000 to 7FFFhex. Note, however that this is still a one-to-one mapping, since each half of the logical address space is mapped into the bottom half of a distinct Main Memory Bank.

Continuing with the case where the bank-switch boundary is at 8000hex and the MAP TYPE bit cleared, the effect of the PAGE MODE bit is to determine whether it is the upper or lower half of the logical address space which is mapped into a "foreign" part of the physical address space. As explained above, if the PAGE MODE bit is cleared, the upper half of the logical address space is mapped into the bottom half of bank A, while the lower half is mapped into the lower half of bank B. If, instead, the PAGE MODE bit is set, then the upper half of the logical space will be mapped into the upper half of bank A. This is the physical space into which the upper half of the logical space would be mapped if the MAP TYPE bit were set. The

lower half of the logical address space is now mapped into the upper half of bank B.

With the MAP TYPE bit cleared and the PAGE MODE bit set, it becomes possible to access the otherwise-hidden upper 8k of bank B. Setting the bank-switch boundary to 8000H, as in the description above, forces logical addresses from 0000 to 7FFFhex to be mapped onto physical address in bank B from 8000hex to FFFFhex, a range which includes the hidden 8k. Thus the logical address range 6000hex to 7FFFhex will be mapped onto the upper 8k of bank B.

By appropriately setting the F-bits in the SAM control register and bit 0 of the I-register, it is possible to position the Video Display Memory so that this hidden 8k region is used for all or part of the Video Display Memory. Table V in the Appendix gives suggested values for the F-bits and the resultant Video Memory address boundaries.

QUARK Boot Mode Memory Mapping

The QUARK has a special memory-mapping mode which is used after the system has been reset by a low-level signal at the RESET input, or when the internal BOOT MODE line has been set. This memory mapping mode is called the Boot Mode. This mode is used when the system is loaded from the floppy disk drive (in a so-called "bootstrap" manner), and whenever the SAM control registers are to be written.

In this mode, any memory read operation by the CPU (including instruction fetching) will read from the contents of the 512-byte Bootstrap PROM, rather than from Main Memory. Memory write operations will write to Main Memory, using the memory mapping process described. I/O read or write operations will reference I/O ports in the usual way, except that the Character Generator (I/O addresses 00-3F) cannot be written when in the Boot Mode.

When the QUARK is reset, the registers of the Z80B, including the Program Counter, are cleared. Also, various I/O lines on the VIA and PIA are cleared and set to act as inputs. In particular, the BOOT MODE control line, which is the CB2 control line on the PIA, will be reset to act as an input, allowing a pullup resistor on this line to assert a "BOOT MODE" condition.

The Z80B will now fetch the instruction at location 0000. Since the BOOT MODE control line is high, this will cause the instruction at location 0000 of the Bootstrap PROM to be fetched. Successive instructions will be fetched from the PROM until the BOOT MODE control line is cleared to a low-level output condition. This puts the QUARK into the Normal Mode of operation, where all memory read or write operations will reference the Main Memory. In the Normal Mode, the PROM cannot be accessed.

The logical address space occupied by the Bootstrap PROM consists of the first 64 addresses (i.e. 00-3Fhex) of each 256-byte page in memory. Thus the entire PROM can be accessed in eight 64-byte "chunks" in the bottom 2k of memory. This logical address space is aliased between 0000 and 073F(hex) are used in accessing the Bootstrap PROM.

Addresses between the 64-byte chunks cause I/O read or write operations. It is not recomended that these addresses be used. I/O operations may be performed in the usual manner using I/O instructions while in the Boot Mode.

It should be pointed out that it is unlikely that the user will ever need to directly use the Boot Mode of operation, nor is it ever necessary to directly access any part of the Bootstrap PROM. After the operating system has been booted from the floppy disk, the only operation requiring a return to the Boot Mode is when the SAM control registers are to be re-written. In this case, a special routine called the SAM Loader Access Routine, resident on the bottom page of Main Memory, will automatically handle the entry into and return from the Boot Mode.

The size of the Video Display Memory and its location within the 128k Main Memory are under the control of registers in the SAM and by bit Ø of the I-register.

Bit Ø of the I-register determines in which memory bank the Video Display Memory is located. When this bit is clear, the Video Display Memory is located entirely within bank A, and when it is set, the Video Display Memory is located entirely within bank B. The bank switching apparatus for the Main Memory using bits 5-7 of the I-register does not apply to addresses generated by the Video Display Controller, but only to addresses generated by the CPU, regardless of whether these CPU addresses fall within the Video Display Memory or not.

In either bank A or bank B, the physical address boundaries of the Video Display memory within the selected bank are determined by the settings of bits FØ, F1, F2, F3, F5, and F6 in the SAM control Register, and by the Video Display Mode (ALPHA or GRAPHICS). The starting address (or the lower bound) of the Video Display Memory is the binary address

(F6)(F5)(F3)(F2) (F1)(FØ)ØØ ØØØØ ØØØØ,

where (Fn) represents the contents on the Fn bit in the SAM Control Register. The final address (or the upper bound) of the Video Display Memory depends on the Video Display Mode. In Alphanumeric Mode, the final address is the first 16k address boundary followimg the starting address, whereas in Graphics Mode, the final address is the second 16k address boundary following the starting address.

From the above, it can be seen that the size of the Video Display Memory may be set anywhere from zero to 32k in 1024-byte increments. However, only a certain set of sizes are likely to be of use in most applications. First, when operating in Graphics mode, the size of the Video Display memory must be an integer multiple of 3k (3072) bytes in order for the horizontal sync signal to be generated correctly. Second, if it is desired to have the Vertical Sync frequency match the frequency of the local AC power system (to avoid moving "hum-bars" on the CRT and related phenomena), the size of the Video Display Memory must be adjusted so that its entire contents will be read and displayed once during one cycle of the AC power line. In practice, this means that the most useful sizes are likely to be:

LINE FREQ.	MODE	SIZE
50Hz	ALPHA	5k
50Hz	GRAPHIC	30k
60Hz	ALPHA	4k
60Hz	GRAPHIC	24k

In order to achieve a Vertical Sync frequency of exactly 50 Hz or 60Hz, it is necessary that the appropriate frequency crystal be used in the Master Clock generator. Thus it is not possible to generate a 50Hz vertical sync frequency on a board equipped with a crystal intended to permit operation at 60Hz, and vice versa. However, a "60Hz" QUARK can be programmed to operate with a Vertical Sync frequency of 48.1Hz, and a "50Hz" QUARK can be operated at 62.5Hz. See Table VI in the Appendix for suggested settings and the resultant address ranges.

Note that the F4 bit in the SAM in not used in determining the starting address of the Video Display Memory, and must always be set. Bits 1 through 4 of the I-register are "don't care" bits, and have no effect on the operation of the QUARK.

Programming the SAM Registers

The Synchronous Address Multiplexer (SAM) is an essential part of the CPU, and the Video Display Interface.

The Control Registers of the SAM are initialized by the Bootstrap PROM after a system reset. It may be necessary at some time to change some of the values in the Registers of the SAM. To allow the user to do this, a special routine is automatically loaded into Page Zero of of the Main Memory. This routine is called the SAM Loader Access Routine, since it provides access to the SAM Register Loader routine in the Bootstrap PROM.

To set or clear any of the SAM registers, the address corresponding to the register is loaded into the HL register, and then the SAM Register Loader Access routine at location 000Bhex is called. For example, the following sequence will clear register F1:

```
LXI     H,0FF88H        ;LOAD ADDRESS FOR F1-CLEAR
CALL    000BH           ;CALL SAM LOADER ACCESS ROUTINE
```

Table V in the Appendix lists the addresses to be used for clearing or setting each of the registers in the SAM, as well as indicating the functions of some of the register for the QUARK. Note that only some of the registers are ever likely to be changed. Even though there are no actual restrictions on the settings for any of the registers, only a particular set of combinations is useful on the QUARK.

Table VI indicates the useful set of values for the F-registers. Registers F0, F1, F2, F3, F5, and F6 are used in determining the starting address of the Video Display Memory. The location of the Video Display Memory which results from each of the combinations is also shown in Table V.

If it is necessary to use the area of Main Memory occupied by the SAM Loader Access Routine, then the routine can be copied to the corresponding location in any 2k block of memory. Only the section of the Access Routine between locations 000BH and 0017H inclusive need be copied.

For instance, if the area from 000BH to 0017H is copied into the area from F00BH to F017H, then the following example will have the same effect as the previous example:

```
LXI     H,0FF88H        ;LOAD ADDRESS FOR F1-CLEAR
CALL    F00BH           ;CALL SAM LOADER ACCESS ROUT
```

QUARK Interrupt system

On the QUARK, both of the two Z80B interrupt inputs $\overline{\text{NMI}}$ and $\overline{\text{INT}}$ are used. The maskable interrupt input $\overline{\text{INT}}$ is wired-ORed to the interrupt request outputs from the VIA, the PIA, and the ACIA. The non-maskable interrupt input $\overline{\text{NMI}}$ is connected to the 1793 Floppy-disk Controller's Data Request output (DRQ) through an inverter.

Interrupt requests from each of the VIA, PIA, and ACIA can be individually enabled or disabled by writing to the appropriate control registers in these devices.

The Vertical Sync line is connected to the CA1 input of the PIA. Since the frequency on this line is normally 60Hz, the CA1 input can be used to generate a real-time clock interrupt to the CPU. Note that if the size of the Video Display Memory is changed from the "standard" sizes for Alphanumeric and Graphics Modes, the Vertical Sync frequency will change proportionately. Thus, if the Vertical Sync Frequency is changed, any software which relies on this interrupt as a time base should take this into account.

The Interrupt Request line INTRQ from the 1793 Floppy-disk Controller is connected to the CB1 input on the VIA. This interrupt output is set at the completion of any command to the

1793. In order for the 1793 to interrupt the CPU, the CB1 interrupt on the PIA must be enabled for low-to-high transitions on the CB1 pin.

The Floppy-disk Controller's Data Request output is set each time a byte is ready for the CPU or each time the controller is ready to receive a byte from the CPU. The DRQ output is inverted to drive the $\overline{\text{NMI}}$ input. When transferring data, the DRQ signal will go active at a rate proportional to the data transfer rate for the floppy disk drives in use. The QUARK is sufficiently fast so that even with the worst-case data transfer rate (500kbits/second for 8-inch double-density disks), only 70% of the CPU's time is used in moving data between the controller and main memory. This leaves 30% of the processor's capacity available to perform other operations. The left-over capacity is essential in certain applications, such as when an interrupt-driven communications routine must continue despite data transfers to or from the disk. Lesser transfer rates will of course leave correspondingly more of the CPU's time available.

Version 2.2 operating systems place a jump instruction at location 38hex in both bank A and bank B. In bank A, the jump address points to the interrupt handler for the INT interrupt in the BIOS. In bank B, the jump address points to the entry point of a special routine in the BIOS. This routine saves the current value in the I-register (which determines the bank-switching boundary when the interrupt occured), sets the bank-switching boundary to location 0000 (so that all CPU memory is in bank A), and then pushes a special return address onto the stack. The routine then jumps to location 38hex in bank A. This causes the interrupt to proceed as it would have if the bank-switching boundary had been at location 0000 when the interrupt occured.

When the routine handling the interrupt returns, it will pop the return address off the stack. If this return address is the special address pushed earlier by the bank B interrupt handler, then control will be transferred back to this routine. It will then restore the former bank-switching boundary, and return from the original interrupt.

The version 2.23 and 2.24 operating systems distributed with the QUARK enables only two interrupt sources, the real-time clock interrupt from the CA1 input on the PIA, and the $\overline{\text{NMI}}$ interrupt, which is connected to the 1793 floppy-disk controller's DRQ output. When an $\overline{\text{INT}}$ interrupt occurs (which vectors to location 38hex), the interrupt handler in the distributed version of the BIOS does not verify that the source of the interrupt is the real-time clock. Instead, it immediately updates the displayed clock on the screen, if the system was installed with the clock option.

In order to use other interrupt sources on the QUARK, the user may either patch the source file for the BIOS himself, or include in the application program sufficient code to handle the additional interrupts.

For the latter approach, the general idea would be for the application program to save the jump address at locations 39hex and 3Ahex, and replace them with a pointer to the interrupt entry point in the program. When an interrupt occurs, the application program would check for the source of the interrupt, and take the appropriate action if the interrupt was intended for use by the application. If the interrupt was not for the application program, then the program should jump to the regular interrupt entry point in the BIOS, the address of which was saved by the application program initially. Thus the BIOS routine will have its opportunity to respond to the interrupt.

Notice that the mechanism handling interrupts which occur while operating in bank B will mesh perfectly with the above approach. Of course, the user is free to handle bank B interrupts in his own way.

The $\overline{\text{NMI}}$ interrupt is largely a system rather than a user feature on the QUARK. The operating system always handles the $\overline{\text{NMI}}$ interrupt from the floppy disk controller in an appropriate fashion. No mechanism is provided by the operating system for trapping bank B NMI interrupts, as none are expected.

The on-card Video Display Interface is an integral part of the Megatel QUARK. It is capable of operating in either Alphanumeric or Graphics modes. TTL-driven video, horizontal sync, and vertical sync signals are provided for connection to direct-drive CRT data displays. Additionally, a composite video output at approximately $1V_{p-p}$ is provided for use with monochrome displays with composite video inputs.

The data displayed on the CRT are stored in a segment of the Main Memory of the QUARK. The meory bank in which the Video Display Memory is located is determined by bit 0 of the CPU's I-register.

In Alphanumeric mode, a total of 32 character rows are scanned for each video frame on the 60Hz version (assuming a 60Hz frame rate) and 40 rows on the 50Hz version (assuming a 50Hz frame rate). However, not all of the scanned rows may be displayed. The first row, which represents data within the first half-page of the Video Display Memory, is displayed during the Vertical Retrace Period. There being no hardware mechanism to blank the video output during the Vertical Retrace period, this part of the Video Display Memory must be loaded with data that will generate a null video output. The simplest manner in which to do this is to load 00 bytes into that part of the Video Display Memory which is scanned during the retrace period. The Bootstrap PROM routine loads a "blank" character into the Character Generator for the 00 character code, so that when a row of 00 bytes is displayed, no video output will result.

The second line of the display can be used for display purposes if the CRT monitor employed terminates the Vertical Retrace Interval sufficiently quickly. On many monitors, however, characters displayed in this row will appear slanted because of the monitor's inability to recover from the Vertical Retrace Interval in time to properly display the first several scan lines. Thus the video driver routines included with standard QUARK operating systems do not use the second displayable row on the video display.

The QUARK is equipped with a Programmable Character Generator. This allows the eight-bit by eight-bit patterns for the characters displayed on the CRT to be loaded or altered under software control. Custom character sets may be designed, saved on floppy disk, and loaded when needed. This simplifies the task of accomodating application programs requiring languages other than English, or running programs using special symbols, or of using programming languages which employ special character sets (e.g. the APL programming language). While normally an entire character set (256 characters) would be loaded as one step (as is done by the utility program CHRLD.COM) character patterns can be loaded or modified on a byte-by-byte basis, so it is not necessary to load an entire set of characters. This may permit special video effects in some applications.

The Programmable Character Generator used on the QUARK is a 2048-byte static memory which is independent of the Main Memory. A utility routine to load the Programmable Character Generator with a standard character set as the operating system is booted is included with the QUARK operating system software. Also included is a character set editor utility, which can be used to customize the standard character set or to create new character sets. These user-defined character sets can be saved on a floppy diskette.

The Bit-mapped Graphics Mode of the Monochrome Video Display Interface allows graphic data to be displayed on the CRT. In this mode, individual bits in the Video Display Memory are mapped onto single dots (pixels) on the CRT. For 60Hz models, 24k of Main Memory is used to display (typically) 143,360 pixels, organized as 640 horizontally by 224 vertically. On 50Hz models, 30k is used to display (typically) 179,200 pixels, organized as 640 by 280 pixels.

Video Display Memory

The size and location of the Video Display Memory within the Main Memory of the QUARK is under software control through the values stored in the control register of the SAM, as well as by the value in the Z-80B's I register. The size of the Video Display memory may be set to values between 1k and 32k, although only certain memory sizes are appropriate, as will be explained below. The starting address for the Video Display memory is determined by the values loaded into the F6, F5, and F3 to F0 bits of the SAM Control Register. (Bit F4 is not used in the determination of the starting address.) The top address of the Video Display Memory is determined by the mode (Alphanumeric or Graphics) and by the values of the SAM Video Display Counter bits corresponding to F3 and F4.

The Video Display Interface reads 96 consecutive bytes from the Video Display Memory for each raster scan line displayed on the CRT. In Alphanumeric mode, these 96 bytes are the last 96 of each half-page (128 bytes) within the Video Display Memory. The first 32 bytes of each half-page are not read by the Video Display Interface in Alphanumeric mode.

In Graphics mode, the 96 bytes scanned for each line displayed are contiguous within the Video Display Memory. Thus there are no unscanned memory areas within the Video Display Memory when operating in Graphics mode.

In both display modes, the second through sixteenth of the 96 bytes, plus one more from the displayed 80 bytes, are read during the horizontal retrace interval between succesive scan lines on the CRT monitor. The Video output is blanked during this interval, so the contents of these bytes will not directly cause any visible output on the CRT. The last eighty bytes contain either the ASCII codes to be translated into character data on the CRT, or the graphic information to be displayed as pixels.

In the Alphanumeric mode, the same 96-byte block is read eight times for each character row displayed. It is necessary to do this because each character row is built up from eight raster scan lines, each line adding one horizontal "slice" of the character patterns. The ASCII code for each of the eighty characters in the row must be read eight times while the Scan Line Counter counts from 0 to 7.

The Graphics mode operates in a similar fashion to the Alphanumeric mode, except that each of the sets of 96 bytes is read only once for each raster scan line displayed, rather than eight times as in the Alphanumeric Mode. In the Monochrome Video Display Interface, the data in the last 80 of the 96 bytes is sent directly to the Video Shift Register, bypassing the Character Generator. Thus the pattern of bits in each of the 80 bytes determines the pattern of pixels displayed on each scan line of the CRT. Since 80 bytes of eight bits each are read for each line, a total of 640 pixels can be displayed horizontally. The most-significant bit of each byte (bit 7) is the first bit shifted out by the Video Shift Register, and thus appears on the CRT as the left-most pixel of each group of eight.

The time required to read the 96 bytes from the Video Display Memory determines the Horizontal Sync pulse period and frequency. Four Z-80B T-states are required for each byte read. Thus the Horizontal Sync frequency is 15.540kHz (64.368 us), and on the 50Hz version it is 16.1145kHz (61.939 us).

In the Graphics mode, the top address of the Video Display Memory is equal to the output from the Video Counter at the time when the Video Counter bits corresponding to F3 and F4 reach 0 and 1, respectively. In Alphanumeric mode, the top address is such that the Video Counter bit corresponding to F4 reaches zero. Thus in Alphanumeric mode the top address of the Video Display memory is the address of the first 16k boundary following the starting address minus one. In Graphics mode the top address is the address of the second 16k boundary following the starting address minus one.

The Vertical Sync frequency is determined by the display mode, the size of the Video Display Memory area, and the master clock frequency. The exact Vertical Sync period is given by

$$t_V = (96(n+2) + 88) * t_E,$$

where n is the number of scan lines being displayed, and t_E is the period of the E-clock. On the QUARK, this is 670ns. In Alphanumerics mode, the number of displayed lines is the Video Display Memory size (in bytes) divided by 16, while in Graphics mode, it is the Video Display Memory size divided by 96.

For a 60Hz Vertical Sync frequency, the Video RAM area should be 4k in Alphanumeric Mode, and 24k in Graphics Mode (assuming that the master clock frequency is 23.86176Mhz, as is the case for the "60Hz" versions of the QUARK). For a 50Hz Vertical Sync frequency, the memory sizes should be 5k and 30k, respectively (assuming the 24.8MHz master clock frequency used on "50Hz" versions of the QUARK).

If it is not necessary for the Vertical Sync frequency to be exactly 60Hz then one is free to pick the starting address of one's choice. The starting address of the Video Display Memory may be placed on any 1k address boundary, although in Graphics mode only 3k boundaries will result in a prpoer Horizontal Sync signal. For example, a 60Hz board could be operated in the Graphics mode with a Video Display Memory size of 27k instead of 24k. This larger size would result in a vertical sync frequency of 53.4Hz, with a total of 288 lines per frame, instead of 256. The horizontal sync frequency would remain unchanged at 15540Hz.

See Table VI in the Appendix for SAM Control Register values needed to select various Video Display Memory sizes and locations.

Alphanumeric Mode

In Alphanumeric Mode, data stored in the Video Display Memory are interpreted as eight-bit character codes. These codes are fetched from the 96 scanned locations in each half-page of the Video Display Memory and presented to the input of the Character Generator. The Character Generator contains the patterns which represent the characters to be displayed on the CRT.

The standard character set for the QUARK uses characters formed from a 5-by-7 matrix of dots. This matrix is imbedded in a larger 8-by-8 background field. The background field is effectively part of the character; in the standard QUARK character set the top row of dots and the first two and the last columns of dots for each character are blank to provide the necessary space between adjacent characters. The standard character set includes 128 "normal video" characters and 128 "reverse video" characters, and is designed to use seven-bit ASCII codes. The reverse video characters are essentially a second set of 128 characters where the dots forming both the character and the background are inverted from the corresponding normal video characters. In the standard character set, a reverse video character is displayed whenever bit 7 (the most significant bit) of any 8-bit value stored in the Video Display Memory is set. The remaining seven bits form the ASCII code for the character that is to be displayed.

There are no extra dot columns or dot rows between the 8-by-8 background fields of each character, that is, that the 8-by-8 fields completely fill the displayable area of the screen. Thus it is also possible to create graphic characters (as distinguished from Bit-mapped Graphics, below) which allow continuous lines, bars, and other figures to be displayed on the screen in Alphanumeric Mode.

In the Bit-mapped Graphics Mode, the Monochrome Video Display Interface fetches data from the Video Display Memory in exactly the same manner as is used in the Alphanumeric mode. However, instead of this data going to the character generator (together with the three Scan Line Counter bits), the data goes directly to the Video Shift Register. Thus the patterns of "ones" and "zeroes" stored in each byte of the displayed portion of the Video Display Memory will directly determine which dots, or pixels, are illuminated on the CRT.

Because the Character Generator is not used to map eight-bit bytes onto eight-by-eight dot patterns in the Graphics Mode, eight times as much memory must be allocated for the Video Display Memory when operating in Graphics Mode as when in Alphanumeric mode. Note, however that in Graphics mode the 96-byte blocks scanned for each raster line displayed are contiguous in the QUARK's Main Memory, whereas in Alphanumeric Mode, these blocks have 32-byte gaps between each block. Thus the actual memory area used in Graphics mode need be only six times that used for Alphanumeric mode. The size of the Video Display Memory is increased by moving its starting address downward within the Main Memory.

To switch between the Bit-mapped Graphics Mode and Alphanumeric Mode, the Graphics Mode Bit of the QUARK and the V2 Mode bit of the SAM must be changed. The Graphics Mode Bit is bit 6 of the PIA Port B output (I/O address 76hex). The Graphics Mode bit is at a logic low level for Graphics Mode and a logic high level for Alphanumeric Mode. This bit is cleared by writing to Port B with accumulator bit 6 set to zero, and is set by writing to the same port with the same accumulator bit set to one. Note that bit 6 of the PIA Data Direction Register (I/O address 74hex) must be high in order that PB6 be enabled as an output line. The V2 Mode bit of the SAM must be set to 1 for the Graphics Mode, and cleared for the Alphanumeric Mode. The V0 and V1 mode bits are left cleared in both display modes.

The Graphics Mode bit and the V2 mode bit must be changed in synchrony with the Vertical Sync (VS) signal. A suggested approach using the Real-time Clock interrupt (which is generated by the positive-going edge of the VS signal) to initiate the sequence of mode bit changes is described below.

When the Real-time Clock interrupt occurs, create a delay of at least five microseconds. This delay ensures that the seven pulses (at the E-clock frequency) immediately following the first rising edge of the VS signal will be bypassed. At the end of this delay, load the new starting address of the Video Display Memory (defined by bits F0-F3, F5, and F6) into the SAM. Then change the polarity of the active transition on the CA1 input of the PIA from positive-transition active to negative-transition active. (This requires that bit 1 of the PIA Control Register A be set low.) Now wait for the negative transition of the VS signal by polling the PIA IRQA1 flag (bit 7 of Control Register A, I/O address 75hex). Immediately upon detecting this negative transition change the Graphics mode bit and the V2 bit of the SAM to the values required for the mode to be selected. After changing these bits restore the original value of bit 1 of the PIA Control Register A by loading a one into this bit. This completes the sequence of operations required to change from Alphanumeric mode to Graphics mode.

The following routine can be called to enter into the Graphic Mode. It is assumed here that the Real-time Clock interrupt is disabled before entering the routine.

```
;Graphics-entry example routine
;revised sept 15, 1983 to restore piacra and eliminate unnecessary
;sync loops
;
SAMV2S  EQU     0FF85H  ;ADDRESS TO SET V2 BIT IN SAM
SAMF0C  EQU     0FF86H  ;ADDRESS TO CLEAR F0 BIT IN SAM
SAMF1C  EQU     0FF88H  ;ADDRESS TO CLEAR F1 BIT IN SAM
SAMF2C  EQU     0FF8AH  ;ADDRESS TO CLEAR F2 BIT IN SAM
SAMF3S  EQU     0FF8DH  ;ADDRESS TO SET F3 BIT IN SAM
SAMF5C  EQU     0FF90H  ;ADDRESS TO CLEAR F5 BIT IN SAM
SAMF6C  EQU     0FF92H  ;ADDRESS TO CLEAR F6 BIT IN SAM
```

```
SAMSET  EQU     0000BH  ;ADDRESS OF ROUTINE TO SET/CLEAR SAM BITS
PIAPA   EQU     074H
PIACRA  EQU     075H
PIAPB   EQU     076H
        ;
GRAPHICS:
        DI
        IN      PIAPB   ;READ THE CURRENT STATE OF PIA PORT B
        ANI     0BFH    ;CLEAR BIT 6 (GRAHICS/ALPHA BIT)
        MOV     B,A     ;SAVE THIS IN REGISTER B
        CALL    SYNC    ;CALL ROUTINE TO SYNCHRONIZE WITH
                        ; VERTICAL SYNC LINE
        MOV     A,B     ;RESTORE VALUE TO SEND TO PIA PORT B
        OUT     PIAPB   ;RESTORE PIA PORT B WITH GRAPHICS BIT CLEARED
        LXI     H,SAMV2S        ;GET SAM V2-SET ADDRESS
        CALL    SAMSET  ;SET V2 FOR GRAPHICS MODE
        LDA     PIASAVE ;GET FORMER VALUE FOR PIA CRA
        OUT     PIACRA  ;RESTORE PIA CRA
        RET             ;RETURN FROM GRAPHIC-ENTRY ROUTINE
        ;
SYNC:
        IN      PIACRA  ;GET CURRENT PIA CRA
        STA     PIASAVE ;SAVE IT
        MVI     A,0C4H  ;VALUE FOR PIA CONTROL REGISTER A
        OUT     PIACRA  ;DISABLE INTERRUPTS FROM CA2, SET IRQA2 ON
                        ; HIGH-TO-LOW TRANSITION OF CA1 (VERT. SYNC)
        IN      PIAPA   ;READ PIA PORT A TO CLEAR IRQA-1 AND -2 FLAGS
LOOP1:
        IN      PIACRA  ;READ PIA CRA
        ANI     080H    ;EXAMINE BIT 7 (=IRQA1 FLAG)
        JZ      LOOP1   ;LOOP UNTIL IRQA1 IS SET BY HIGH-TO-LOW
                        ; TRANSITION OF VERTICAL SYNC LINE
;
;THE FOLLOWING WILL CONFIGURE THE VIDEO DISPLAY MEMORY BETWEEN
;ADDRESSES 2000H AND 7FFFH
;SEE TABLE VI IN THE APPENDIX FOR OTHER LOCATIONS FOR THE VIDEO
;DISPLAY MEMORY
;
        LXI     H,SAMF0C  ;CLEAR F0 BIT
        CALL    SAMSET
        LXI     H,SAMF1C  ;CLEAR F1 BIT
        CALL    SAMSET
        LXI     H,SAMF2C  ;CLEAR F2 BIT
        CALL    SAMSET
        LXI     H,SAMF3S  ;SET F3 BIT
        CALL    SAMSET
        LXI     H,SAMF5C  ;CLEAR F5 BIT
        CALL    SAMSET
        LXI     H,SAMF6C  ;CLEAR F6 BIT
        CALL    SAMSET
        ;
        IN      PIAPA    ;READ PIA PORT A TO CLEAR IRQA-1 AND -2 FLAGS
LOOP3:
        IN      PIACRA   ;READ PIA CRA
        ANI     080H     ;EXAMINE BIT 7 (=IRQA1 FLAG)
        JZ      LOOP3    ;LOOP UNTIL IRQA11 IS SET BY HIGH-TO-LOW
                         ; TRANSITION OF VERTICAL SYNC LINE
        RET
PIASAVE:
        DS      1        ;ONE BYTE FOR SAVING PIA CRA
```

The example and description above assumed that the interrupts were disabled throughout the procedure. It would also be possible to re-enable interrupts after loading the Video Memory starting address and changing the polarity of the CA1 active transition. The next negative transition of the VS signal would generate an interrupt, immediately following which the Graphics mode bit and the V2 bit would be changed, just as in the non-interrupt procedure above.

To return to Alphnumeric mode, it is necessary only to set the Graphics/Alphanumeric mode bit, and then clear the V2 bit in the SAM, and set up the F-bits for the desired Video Display Memory location. The routine below will set up the Alphanumeric mode with the Video Display Memory located between F000 and FFFF, as is set up by the Bootstrap PROM.

```
;alpha-entry example routine
SAMV2C  EQU     0FF84H  ;ADDRESS TO CLEAR V2 BIT IN SAM
SAMF0C  EQU     0FF86H  ;ADDRESS TO CLEAR F0 BIT IN SAM
SAMF1C  EQU     0FF88H  ;ADDRESS TO CLEAR F1 BIT IN SAM
SAMF2S  EQU     0FF8BH  ;ADDRESS TO SET F2 BIT IN SAM
SAMF3S  EQU     0FF8DH  ;ADDRESS TO SET F3 BIT IN SAM
SAMF5S  EQU     0FF91H  ;ADDRESS TO SET F5 BIT IN SAM
SAMF6S  EQU     0FF93H  ;ADDRESS TO SET F6 BIT IN SAM
PIAPA   EQU     074H
PIACRA  EQU     075H
PIAPB   EQU     076H
SAMSET  EQU     0000BH
        ;
        ;
ALPHA:
        ;TO AVOID GLITCHING SCREEN, THIS POINT SHOULD SYNCHRONIZED WITH
        ;VERTICAL SYNC SIGNAL (SEE GRAPHICS-ENTRY EXAMPLE)
        DI              ;DISABLE INTERRUPTS
        IN      PIAPB   ;READ CONTENTS OF PIA PORT B OUTPUT
        ORI     40H     ;SET BIT 6 FOR ALPHANUMERIC MODE
        OUT     PIAPB
        LXI     H,SAMV2C
        CALL    SAMSET
        LXI     H,SAMF0C
        CALL    SAMSET
        LXI     H,SAMF1C
        CALL    SAMSET
        LXI     H,SAMF2S
        CALL    SAMSET
        LXI     H,SAMF3S
        CALL    SAMSET
        LXI     H,SAMF5S
        CALL    SAMSET
        LXI     H,SAMF6S
        CALL    SAMSET
        EI
        RET
```

This completes the routine to switch from graphics mode to alphanumeric mode.

Loading the Programmable Character Generator

On the QUARK the patterns for the characters displayed in the Alphanumeric Mode are loaded into the Character Generator under software control. Once loaded, the contents of the Character Generator remain until another character set is loaded or until power to the board is removed. The standard set of utility routines included with the Distribution Software include programs for loading the Programmable Character Generator and for designing character patterns to be loaded. It is therefore not necessary to understand how the Programmable Character Generator operates in order to be able to make use of this feature.

Peripheral Interfaces

The Megatel QUARK provides a number of parallel and serial I/O lines. While some of these lines are intended for use with specific peripherals, such as parallel- or serial-interface printers, many of these I/O lines may be used for more general purposes if the intended devices are not being used in a particular application. A discussion of these I/O ports follows. In addition, specialized peripheral subsystems, such as the Floppy-disk Controller provide a high level of I/O support.

Parallel Printer Interface

The QUARK includes a port intended for the connection of an eight-bit parallel-interface printer. This port consists of an eight-bit latch for the output data, a Data Strobe output line, and an Acknowledge input line. All of the input and output lines for this port are TTL-compatible.

Eight-bit parallel data is written to the port by an output instruction to I/O address 5Fhex. The data appearing on the output pins of the QUARK (see Table I for the pinout of the connector) represent the true state of the data written to the port.

The Data Strobe line for the port is controlled by the CA2 control line of the VIA. The CA2 control line must be configured as an output in the Peripheral Control Register (PCR) of the VIA, by setting bits 2 and 3 of the PCR. The CA2 output drives a TTL inverter, the output of which is connected to the Data Strobe output pin (pin C-13). Thus the logic state of the actual Data Strobe output is the inverse of the state of the CA2 line, as determined by bit 1 of the PCR. This TTL inverter is capable of sinking 24mA.

The Acknowledge input for the Parallel Printer Port is directly connected to the CA1 input line of the VIA. The active transition of the CA1 input of this line will set a flag in the Interrupt Flag Register (IFR) of the VIA. The setting of this flag may also generate an interrupt if the appropriate bit in the Interrupt Enable Register (IER) of the VIA is set. This interrupt can be used to interrupt the CPU when the printer is ready to accept another character, depending on the operation of the printer.

Handshaking using the Data Strobe and Acknowledge lines is not automatic, that is that the routine handling the Parallel Printer Interface must write the output latch, toggle the Data Strobe line in the manner required for the interface of the printer, and then act accordingly for the printer's response on the Acknowledge input.

If it is not desired to use this port with a parallel printer, then the eight Data lines, the Acknowledge Input line, and the inverted output line from CA2 may be used for other purposes such as might be required for a particular application of the QUARK.

Parallel Keyboard Interface

The QUARK provides an interface for an ASCII-encoded parallel-output keyboard. This interface uses the Port A I/O lines and the CA2 control line of the PIA. As initialized by standard CP/M operating systems provided on the Distribution Diskettes, active-high ASCII data present on the PA0-7 lines of the PIA will be read after a negative-going stobe pulse on CA2. Note that the data present on the input lines is not actually latched into the Input Data register when the strobe occurs, so the keyboard data must remain valid between the strobe and the read. (This is not usually a problem.)

If it is not intended to use an encoded keyboard for a particular application of the QUARK, then these eight I/O lines and the CA2 control line may be used for any other I/O functions which might be required. It is possible, for instance, to connect an un-encoded

keyboard to the QUARK using the I/O lines from Port A of the PIA and Ports A and B of the VIA. For such a keyboard the user would include his own keyboard-scanning routine as part of the CP/M BIOS. (This software is not provided by Megatel.)

The I/O address for reading or writing either Port A or Data Direction Register A of the PIA is 74hex. The addresses of the PIA Control Register A is 75hex. Table VIId in the Appendix gives the pin connections required for connecting a keyboard to this interface.

Full-Duplex Serial Interface

The QUARK provides a full-duplex asynchronous serial data port. This port uses an Asynchronous Communications Interface Adapter (part number 68A50) and includes line drivers and receivers for compatibilty with RS-232C signal levels. As implied by the term full-duplex, this port can perform bidirectional simultaneous communication.

This port allows the QUARK to be connected to standard terminals and communications peripherals, such as telephone-line modems. The port also allows a terminal to be used as the console I/O device.

Parallel data written to the Transmit Data Register of the ACIA will be transmitted serially on the TxDATA pin of the QUARK connector (pin A-3). Serial data received on the RxDATA pin (pin C-2) is read from the Receive Data Register. The I/O address of both the Transmit and the Receive Registers is 79hex.

A total of four protocol lines are provided for the Port. The Clear-to-Send input line (pin C-4) provides direct control of the transmitter of the ACIA. The Data Set Ready input (pin B-4) drives the ACIA's Data Carrier Detect input, which provides direct control of the receiver of the ACIA. The Request-to-Send output (pin B-3) is controlled by the RTS bit in the Control register of the ACIA. The Data Terminal Ready output (pin C-3) is controlled by the PB7 I/O line of the PIA. The PB7 line should be configured as an output by setting bit 7 of Data Direction Register B in the PIA (I/O address 76hex). All of the four protocol lines are RS-232C compatible.

The serial communications speed (or baud rate) for the serial input and output data is controlled by one or both of the programmable timers in the VIA. On the QUARK the PB7 line from the VIA is connected directly to the Transmit Clock input of the ACIA. When Timer 1 is operated in the free-run mode a square wave is generated on the PB7 output line. This square wave provides the basic Transmit Clock frequency, which is then divided by 1, 16, or 64, according to the settings of bit 0 and bit 1 of the ACIA Control Register.

The Receive Clock input of the ACIA can be connected to the Transmit Clock by jumper J3. Unless ordered otherwise, this jumper is installed at the factory. With J3 installed, the Transmit and Receive baud rates will be identical, both being generated by Timer 1 of the VIA.

Timer 1 uses a 16-bit counter which, in the free-run mode, is automatically re-loaded from the 16-bit Timer 1 latch each time the counter reaches zero. The value in the latch determines the period of the square wave appearing on PB7. The period of this square wave is given by

$$(2N+3.5) * t_E,$$

where t_E is the period of the system E-clock and N is the value in the Timer 1 latch. The count-down clock for Timer 1 is the system E-clock. The period of the E-clock is 670.5 ns. Table II in the Appendix gives the values for the Timer 1 latch required to generate commonly-used baud rates on this serial port. Because Timer 1 is a true 16-bit timer, it is able to produce the widest range of baud rates.

If "split" baud rates on the Full-duplex Serial Port - where the Transmit and Receive baud rates are independently generated - are required, then J3 should be removed and J4 installed. The installation of J4 connects the CB1 line from the VIA to the Receiver Clock input on the ACIA. Pulses may be generated on the CB1 line when the Shift Register (SR) of the VIA is operated in the "Shift out free-running at Timer 2 rate" mode. Values for the Timer 2 low-

order latch (I/O address 68hex) to produce commonly-used baud rates can be found in Table III(a) and III(b) in the Appendix. Jumpers J5 and J6 should not both be installed when attempting to use split baud rates, or the Shift Register output on CB2 will short out the T2-generated clock output on CB1.

To enable this Shift Register Mode, the Auxiliary Control Register (ACR, I/O address 6Bhex) of the VIA must be written with bits 2 and 3 low and bit 4 high. According to the manufacturer's data sheet for the VIA, it is also necessary to perform an I/O read or write operation to the SR after setting up the Shift Mode to start the clocking of the SR (and the clock pulse output on CB1). Note that using Timer 2 and the Shift Register for this purpose precludes the use of the Simplex Serial Port, which uses the VIA's Shift Register, as a serial data channel.

Although Timer 2 uses a 16-bit counter, only the least-significant eight bits are automatically re-loaded in its free-run mode. Thus, the range of baud rates directly available from this timer is more limited than that of Timer 1. However, this range can be extended by using the divide-by-16 and divide-by-64 modes of the ACIA. Note that the divide ratio selected applies to both the Transmit and Receive clocks. Table IV gives the values for the Timer 2 latch required for common baud rates. The period of the basic clock signal generated by Timer 2 is given by

$$(2N+4) * t_E,$$

where t_E is the period of the system E-clock (given previously) and N is the value in the Timer 2 latch.

Note that on the QUARK the negative supply voltage used for the RS-232C drivers is developed on-card by a charge-pump circuit. As a result, the voltage swing on the RS-232C outputs is usually in the range of +11V to -8V.

Simplex Serial Interface

In addition to the Full-Duplex Serial Port, the QUARK provides a simplex (unidirectional) asynchronous serial data port. This port uses the Shift Register of the VIA (part number 6522A) and includes line drivers and receivers for compatibilty with RS-232C signal levels.

The Simplex Port allows the QUARK to be connected to receive-only or transmit-only peripherals with serial interfaces, such as printers or serial-output encoded keyboards. The port can also be used for other purposes, such as generating tones. By means of jumpers J5-8, the port can be configured for serial output or input (but not both). One protocol line is also provided. Depending on which jumpers are installed, this protocol line may be used as an input or an output.

For serial output, the Shift Register (SR), the CB2 control line, and the PB6 I/O line of the VIA are used. In this mode, data in the shift register is shifted out on the CB2 pin of the VIA. With J6 installed (done at the factory unless ordered otherwise), the serial data is shifted to RS232C output voltage levels and is available on pin A-4 of the QUARK's connector. The protocol line, an RS-232C compatible input for this mode, is pin B-2. Jumper J8 (also installed at the factory) allows the state of the protocol line to be read on the PB6 I/O line of the VIA. Bit 6 in Data Direction Register B of the VIA must be zero to allow the use of PB6 as an input.

In order for the Shift Register to be used in the output mode, bit 4 in the Auxilliary Control Register (ACR) of the VIA must be set. With this bit set, bits 2 and 3 of the ACR will determine the rate at which the SR is shifted, as well as its operational mode.

Summary of Shift Register output modes (ACR-4 = 1)

ACR-3	ACR-2	Remarks
0	0	Continuous shifting at T2 rate. Useful for waveform-generation applications.
0	1	8 bits only shifted at T2 rate after each SR load. SR Interrupt Flag set after 8 bits shifted.
1	0	8 bits only shifted at E-clock rate after each SR load. SR Interrupt Flag set after 8 bits shifted.
1	1	8 bits or more shifted at CB1 input rate. SR Interrupt Flag set after 8 bits shifted. Install J3 & J4 for Timer 1 clock to CB1 input.

With ACR bit 2 and 3 cleared, Timer 2 determines the rate at which data is shifted out of the SR. In this mode, shifting is continuous, and does not stop automatically after eight bits have been shifted. This mode can be used for generating retangular waveforms (repeating patterns of eight bits) on pin A-4. This might be useful in some applications for generating audio signals.

In the next mode (ACR-2=1, ACR-3=0), shifting stops automatically after eight bits are shifted, and the SR Interrupt Flag in the Interrupt Flag Register (IFR) of the VIA is set. If the SR Interrupt Enable bit in the Interrupt Enable Register (bit 2 of the IER) is set, the VIA will assert its Interrupt Request output, which will cause a Z80 interrupt if the Z80's interrupt system is enabled. The shifting rate is determined by Timer 2, as in the previous mode. This is the mode used by the Serial Printer Handler included in the Operating System.

The next mode (ACR-2=0, ACR-3=1) operates in the same manner as the previous mode, except that the System E-clock is used as the shift clock, rather than Timer 2. The frequency of the E-clock is 1.49MHz for 60Hz models, and 1.55MHz for 50Hz models.

In the final mode (ACR-2=1, ACR-3=1), the shift rate is controlled by pulses applied to the CB1 input on the VIA. If jumpers J3 and J4 are installed, then the PB7 I/O line will be connected to the CB1 line (as well as to the Transmit and Receive clocks on the ACIA - see Sec 3.3). This allows Timer 1, normally used to generate the baud rates for the ACIA, to also generate the shift clock for the Shift Register. Note that J5 must not be installed in this mode, or the CB1 clock input will be shorted to the CB2 SR output.

For serial input, the Shift Register, and the CB1 and CB2 control lines are used. In this mode, data is shifted into the SR on the CB2 line. Pin B-2 on the QUARK's connector provides an RS-232C compatible input line for the Simplex Port input mode. Jumper J7 should be installed to allow the signal from pin B-2 to reach the CB2 input on the VIA. (Note that J8, installed at the factory unless ordered otherwise, will connect CB2 and PB6 together when J7 is installed. If both J7 and J8 are installed, PB6 must be configured as an input or it will contend with the signal at CB2.)

In order for the Shift Register to be used in the input mode, bit 4 of the ACR must be cleared. As in the output mode, bits 2 and 3 of the ACR will determine the rate at which the SR is shifted, as well as its operational mode.

Summary of Shift Register input modes (ACR-4 = 0)

ACR-3	ACR-2	Remarks
0	0	Shift register disabled.
0	1	8 bits only shifted in at T2 rate. SR Interrupt Flag set after 8 bits shifted. Shift pulses generated on CB1 during shifting.
1	0	8 bits only shifted in at E-clock rate. SR Interrupt Flag set after 8 bits shifted. Shift pulses generated on CB1 during shifting.
1	1	8 bits or more shifted at CB1 input rate. SR Interrupt Flag set each time 8 bits shifted in. Install J3 & J4 for Timer 1 clock to CB1 input.

In the first mode (ACR-2=0, ACR-3=0) the SR is disabled. The SR can be read or written, but no shifting occurs, and the CB1 and CB2 lines are under the control of the appropriate bits in the Peripheral Control Register.

In the next mode (ACR-2=1, ACR-3=0), data is shifted into the SR at a rate controlled by Timer 2. Shift pulses are generated on the CB1 line. If J5 is installed, then these pulses will appear (at RS-232C signal levels) on pin A-4 of the connector. (Note that if J4 is installed, these pulses, at TTL signal levels, will also appear on the Receive Clock input of the ACIA. This may interfere with the operation of the ACIA.)

The third mode is similar to the previous, except that the E-clock is used to control the shift rate.

In the final mode, pulses on the CB1 line control the shift rate. The only way to provide a signal input to the CB1 line while CB2 is being used for the SR input is to install J3 and J4. This connects the PB7 line to the Transmit and Receive clocks of the ACIA and to the CB1 input. This allows Timer 1 to generate a clock signal for the ACIA and for the Shift Register.

Values for Timer 1 to generate commonly-used baud rates for the Simplex Port are given in Table IV.

By various combinations of straps, the lines and RS-232C drivers and receivers associated with this port may be used for a variety of purposes.

Parallel I/O lines

The QUARK provides fourteen general-purpose I/O lines (not including the nine lines on the PIA used for the parallel keyboard interface). These lines are connected to Ports A and B of the VIA. Each of these lines may be programmed to act as an input or as an output by setting or clearing the corresponding bit in the Data Direction Register.

Parallel Port 2 of the QUARK provides eight I/O lines. These are connected to the PA0 to PA7 lines on the VIA. These lines are not dedicated to any particular purpose in the system software for the QUARK.

Parallel Port 3 of the QUARK provides six I/O lines, which are connected to the PB0 to PB5 lines. Under the standard distributed operating system, the PB0 line is configured as the "bell" output from the QUARK. When an ASCII control-G character (code 07) is encountered by the terminal driver, a square wave will be produced on this output. The PB0 output line may be capable of driving some piezo-electric acoustic transducers directly, or an external buffer amplifier using a transistor or gate can be used to drive a small speaker.

The two remaining I/O lines of Port B on the VIA are intended for some specific uses. PB7 is normally used as the output line for Timer 1-generated baud rates for the Full-Duplex Serial

Port (ACIA) or for the Simplex Serial Port. PB6 is used in the output mode of the Simplex Serial Port as a protocol input line. Neither PB6 nor PB7 are available directly on the QUARK's connector.

Note that when the VIA's Port B lines are configured as outputs, the value read in Input Register B (I/O address 60hex) is the value programmed for the corresponding bit in Output Register B, not the logic level actually present on the output pin of the VIA.

Serial Keyboard Interface

The QUARK can be used with serial-output keyboards. Many applications may require the use of a "detachable", or stand-alone keyboard. These keyboards generally use asynchronous serial communication over a single pair of conductors to reduce the size of the cable connecting the keyboard to the main housing of the system.

The Simplex Serial Interface can be used for this application. When strapped in the input mode (explained below), the CB2 control line of the VIA as well as that device's Shift Register can be used as a receiver for the asynchronous keyboard output. Timer 2 can be used to generate the clock for the Shift Register.

The basic method of operation is to enable the CB2 control line to act as an edge-triggered independent-interrupt input. This mode is enabled by setting bit 5 and clearing bit 7 of the VIA's Peripheral Control Register (PCR, I/O address 6Chex). Also, the state of bit 6 in the PCR determines the active edge on CB2 which will set the CB2 Interrupt flag. The active transition will be positive-to-negative when bit 6 is cleared, and negative-to-positive when bit 6 is set. Bit 6 should be configured so that the transmission of the start bit from the keyboard will be recognized as the active transition. Note that the RS-232C line receiver driving the CB2 control line will invert the signal from the keyboard.

Jumper J7 must be installed to allow the output of the RS-232C receiver whose input is from pin B-2 to be connected to the CB2 line on the VIA. Jumper J8, which connects the output of the receiver to the PB6 I/O should either be removed, or PB6 must be configured as an input. If this is not done, then the output on PB6 will short out the input on CB2. Although pin B-2 would ordinarily be driven by an RS-232C driver, most TTL serial keyboard outputs should be able to drive this input properly.

When the keyboard transmits the start bit, the active transition of the CB2 line will set the CB2 Interrupt flag in the Interrupt Flag Register. If the CB2 Interrupt Enable bit in the Interrupt Enable Register is set, the VIA will generate an interrupt to the Z80B.

Once in the interrupt service routine for the serial keyboard, the VIA shift register must be enabled in one of its input modes. The most useful mode is likely to be the "Shift in Under Control of T2", since in this mode Timer 2 is used to generate the Shift Register clock. If the Shift Register Flag in the IFR is not set, then shifting will begin as soon as this mode is enabled. At the completion of eight shifts, the SR flag in the IFR will be set, and, if the SR Interrupt in the IER is enabled, the VIA will generate an interrupt to the Z80B. The contents of the SR can then be read, and the SR disabled. This allows the CB2 input to return to the edge-triggered independent-interrupt mode, so that the next start bit from the serial keyboard will be recognized.

Baud rates for the Shift Register can be found in Table III. It should be noted that depending on the interrupt service routine for the serial keyboard as well as the timing of the serial data stream it is possible that the first bit (or the start bit) will not be recognized as a data bit. As of the date of issue of this document, Megatel does not offer software for a serial keyboard interface using the simplex port.

The Full-duplex port may also be used for a serial keyboard input. This port would be particularly simple to implement if the keyboard's output is standard 7- or 8-bit RS-232C protocol. If this is the case, then it is necessary only to enable the ACIA properly for the receiving mode used, and then poll the Received Data Register Full lag for an incoming

character. The CP/M operating systems distributed with the QUARK generally allow the "TTY:" or "AUX:" input (the full-duplex port) to be installed as the console input device. See the Installation maunual for more information.

Special I/O functions

Some of the control lines on the PIA are connected internally on the QUARK for special functions. Care should be taken when programming the PIA that these functions are not disturbed by inadvertently modifying the Control Registers.

The CA1 control line on the PIA is connected to the Vertical Sync output for the Video Display. The CA1 line can be programmed to generate an interrupt to the Z80B on either the rising or falling edge of the Vertical Sync line. This allows the use of a real-time clock function, a feature which is incorporated into some of the QUARK operating systems.

The CB1 control line on the PIA is connected to the Interrupt Request (INTRQ) output from the 1793 Floppy Disk Controller. The 1793 produces an interrupt request signal at the completion of any command. The CB1 input is initialized as a positive-transition active input. When a Floppy Disk Controller interrupt occurs, the IRQB flag in the PIA will be set. This configuration of the CB1 line must not be altered if the Floppy Disk Interface is to operate properly.

The IRQ bit can be programmmed to generate an interrupt to the Z80B when this occurs, although this interrupt is not used under the standard operating system for the QUARK.

The CB2 control line is the internal Boot Mode control line. This line will be set to act as an input after a system reset has occured. When it is configured as an input, or as an active-high output, a pull-up resistor pulls the Boot Mode line high, putting the QUARK into Boot Mode operation. At the end of the Bootstrap routine, this line will be cleared and configured to act as an output, returning the system to Normal mode operation. There is no need for the user to change the state or configuration of this line during normal operation of the QUARK.

All of the PIA Port B I/O lines are used for specific output functions. PB0-3 are the Floppy Disk SEL0 to SEL3 outputs. PB4 is the Floppy Disk SIDE select output. If floppy disk drives requiring a "Low Write-current" input are used, QUARK operating systems can be installed to allow any of the SELect lines or the SIDE line to be used for this purpose. (See the Installation manual.)

PB5 is the Floppy Disk Single/Double-density control line. Single-density (FM) operation is selected when this line is at a high level. PB6 is the Alphanumeric/Graphics Mode line. A high level on this output selects Alphanumeric Mode. PB7 is the Full-duplex Serial Interface DTR output. A high level on PB7 causes a negative-voltage output on pin C-3.

Floppy Disk Interface

The on-card Floppy Disk Interface is capable of controlling up to four double-sided floppy disk drives. Both single-density (FM) and double-density (MFM) recording formats are supported, and either 8-inch or 5.25-inch drives may be used. The selection of single- or double-density operation is under software control, through bit 5 of PIA port B (PIA PB5). This output is set (logical 1) for single-density operation, and is cleared (logical 0) for double-density. Jumper 2 (designated J2), located adjacent to the crystal, is open (not installed) when 5.25 inch drives are to be used, and is closed (installed) when 8-inch drives are to be used. The QUARK board is shipped with J2 installed if the system was ordered with the distribution software on an 8-inch diskette, while J2 is not installed on boards ordered with 5.25-inch diskettes.

Four Drive Select lines (SEL 0 to SEL 3) are provided to select one of four floppy disk

drives. Additionally, a Side Select output (SIDE) is provided for use with double-sided drives. The Select and SIDE outputs are controlled by PB0, 1, 2, 3, and 4 of Port B of the PIA. These lines are initialized to act as outputs in the Bootstrap PROM routine.

The Side Select line or one of the Drive Select lines may be used as a Low Write-current line. Some floppy disk drives require an external control line to reduce the write current to the recording head when writing the inner tracks on the diskette. If the drives to be connected to the QUARK require such a control line, then the system installation program allows any of the Drive Select lines or the Side Select line to be used for this purpose. Note that using a Drive Select line for this purpose would reduce the maximum number of drives that could be connected to the QUARK to three. If only single-sided drives are used, then the Side Select line may be used as the Low Write-current line with no effect on the maximum number of drives permitted. #B17 #C18

Note that normally all of the drives connected to the QUARK must be of the same size, that is, that a combination of 8-inch and 5.25-inch drives cannot be directly supported without a modification to the board. However, since the installation of J2 merely ties the internal Disk Size Select line low (permitting 8-inch operation), a special control line connected to the Disk Size Select line can be used to change the selected Disk Size under software control. If a particular application of the QUARK requires between one and three 5.25-inch floppy disk drives and only one 8-inch drive, then the simplest solution is to install a strap from the Disk Select line (SEL0, 1, 2, or 3) which is to be used to select the 8-inch drive to the Disk Size Select line. Since the Drive Select lines are active low, then whenever the 8-inch drive is selected, the Disk Size Select line will be pulled low, enabling 8-inch drive operation. When any of the 5-inch drives are selected, the Drive Select line for the 8-inch drive, and thus the Disk Size Select line, will be at a logical high level, putting the Floppy Disk Interface into 5.25-inch operation, just as is required.

If more then one 8-inch drive is to be used with some number of 5.25-inch drives, then the Disk Size Select line must either be connected to an unused parallel output line from the PIA or VIA on the QUARK (which can be then controlled in a software patch to the BIOS), or the Disk Select lines used to select the 8-inch drives must be AND-ed together and the result of this combination used to control the Disk Size Select line. The latter approach will, of course, require an external AND gate as well as the strap connecting the gate's output to the Disk Size Select line.

The outputs from the QUARK to the floppy disk drives are driven by medium-current low-power Schottky TTL drivers. Inputs from the floppy disk drives are terminated by 150ohm pullup resistors to the +5V supply on the QUARK. The floppy disk drive most distant from the computer should have 150ohm terminating pullups on the output lines for proper transmission line characteristics. None of the intermediate drives should have passive loads on either input or output lines.

Most of the remaining hardware for the Floppy Disk Interface is incorporated within the Western Digital 1793-02 Floppy Disk Controller (or equivalent).

Expansion of the Megatel QUARK

Although the various members of the Megatel QUARK family of single-board computers provide all of the functions necessary to integrate standard peripheral devices into a complete, stand-alone computer system, there may be special or additional I/O functions needed in some applications. The QUARK can accomodate these special functions through its Peripheral Expansion Bus. This bus provides external access to the eight data lines, the six least-significant address lines, and to appropriate timing and select lines.

The Read ($\overline{RD}$) and Write ($\overline{WR}$) lines generated by the Z80B are brought out on the Peripheral Expansion Bus. These two lines would be used with 8080-compatible peripheral devices, such as the 8251 UART, or the 8255 PPI. To connect 6500- and 6800-compatible devices the E-clock output and the $\overline{WR}$ line are used. For these devices, the $\overline{WR}$ line functions as the 6500/6800 RD/$\overline{WR}$ line.

The Z80B's Interrupt input line (INT) is available on the Peripheral Expansion Bus to allow external devices to generate interrupts to the Z80B. A power-on reset output (active low) is also provided to reset external peripheral devices. Finally, a decoded active-low chip-select line responding to I/O addresses between C0 and FF(hex) is created on-board and can be used to select a single external peripheral device, or to qualify the decoding of some of the address lines for several external chip-select lines.

The Power-on-reset (POR) output on the Peripheral Expansion Bus is an active-low buffered reset line which should be used in resetting external peripheral devices. Note that 8080-type devices require an active-high reset signal, so the POR line would have to be inverted to service these devices.

To ease timing requirements for interfacing external peripheral devices to the six megahertz CPU, four wait states (T-states) are added to all Z80 I/O machine cycles. This is over and above the standard extension to mod-4 cycles for any memory cycle. Thus an I/O instruction which might require, for example, 10 cycles would be first extended to 12 cycles, and then further extend to 16 cycles. This allows "A"-version (1.5MHz) peripheral devices to be used. Because of the exact rule used for wait-state insertion, Table XII should be consulted for precise instruction timing information.

If several external peripheral devices are to be connected to the QUARK's Peripheral Expansion Bus, or if the Bus is to be extended any significant distance, it is recommended that the address, data, and control lines being used be buffered by TTL drivers. Two TTL packages are sufficient for this purpose if only 8080- or only 6500/6800-type peripheral chips are used; if both types are employed then an additional buffer is necessary, unless five or fewer address lines are required. An application note on use of this bus is available.

PortB PIO = FDC

1. The QUARK component computers use some high-speed bipolar integrated circuits in order to achieve their high-performance specifications. Some of these parts become very warm in the course of operation. This is perfectly normal, and is no cause for alarm. However, adequate ventilation for the board should be provided, as it should for any piece of electronic equipment. The QUARK boards are tested at the factory for proper operation at an ambient temperature of over 50 degrees Celsius, a temperature that exceeds the specifications of standard floppy-disk drives. The total power dissipation of any of the QUARK boards is in the range of 10 to 14 watts.
2. Further technical information regarding the expansion of the QUARK hardware and utilization of the Peripheral Expansion Bus is available in the form of "Quark Application Notes - Q-Tips".

3. The red light-emitting diode (LED) on any of the QUARKs is connected to the Z80B's $\overline{\text{HLT}}$ output. This LED will be on when the Z80 is running. Executing a $\overline{\text{HLT}}$ instruction, for instance, will extinguish the LED.

4. If a QUARK computer is unsucessful in locating track 0, sector 1 of the system disk drive when cold-booting, it will, after 10 tries, display a diagonal "staircase" pattern on the Video Display interface, and then halt (red LED off). No other indication of the error will be given.

5. The Parallel Printer STROBE output (from the CA2 line on the 6522A to pin C-13 on the connnector) is buffered by a medium-current TTL driver. On all QUARK/100s bearing revision 05R00 or greater, this driver is a non-inverting buffer, one of the eight such buffers in the 74LS241. On all earlier QUARK/100s, this driver is an inverting buffer, part of the 74LS240.

As a consequence of this change, the sense of the Parallel Printer STROBE output will be inverted with respect to its former sense. This means, among other things, that a software driver intended for use with the inverter-driven STROBE line will not, strictly speaking, be compatible with the non-inverting buffer. However, this may not be as much of a problen as it seems.

Most printers latch the input data on the rising edge of the STROBE line. As long as the data is valid on its data inputs for at least the minimum set-up period before the edge, the data will be accepted by the printer. Since the parallel printer driver in the BIOS of the distributed CP/M 2.23 or 2.24 systems generates a short low-to-high-to-low pulse (as seen on the CA2 pin) after it sends the parallel data to the output port, the printer will see either a high-to-low-to-high pulse (if an inverter drives the STROBE line) or a low-to-high-to-low pulse (if a non-inverting buffer drives the STROBE line). If the inverter is driving the line, then the printer will latch the data on the **second** transition of the pulse. If the non-inverting buffer drives the line, it will latch the data on the **first** transition. Since the parallel data is valid on the output port well before either transition occurs, and because the both transitions of the pulse occur while the data is still valid, the printer will have no difficulty acquiring the data, regardless of the polarity of the signal.

Software for the QUARK

The Megatel QUARK is available with an extensive operating system and utility software package. This package, may be ordered on one of the following sets of disk media:

- 5.25 inch, double-density encoding (MFM), Megatel format
  48 tracks per inch, 35 tracks per side, single-sided
- 5.25 inch, double-density encoding (MFM), Megatel format,
  96 tracks per inch, 80 tracks per side, single-sided
- 8 inch, double-density encoding (MFM), Megatel format
  48 tracks per inch, 77 tracks per side, single-sided

When shipped from the factory, all of the floppy diskettes are write-protected to guard against accidental erasure in an improperly-set-up system. It will not be possible for the QUARK to write on a write-protected diskette if the disk drive incorporates the appropriate write-protection mechanism.

As shipped, the distributed operating system is configured with the following logical-to-physical device assignments:

LOGICAL DEVICE	PHYSICAL DEVICE
CON:	UC1:
LST:	LPT:
RDR:	TTY:
PUN:	TTY:

The peripheral interfaces on the QUARK computer and their identities under the CP/M BIOS are given in the table below:

PHYSICAL DEVICE ID	PHYSICAL DEVICE ASSIGNED
TTY:	Full-duplex serial interface
CRT:	Memory-mapped Video Display Interface
BAT:	Console Input from TTY: Console Output to CRT:
UC1:	Simultaneous Console I/O to and from both TTY: and CRT:
LPT:	Parallel printer interface
UL1:	Simplex serial interface (output)
PTP:	Simplex serial interface (output)
PTR:	Not implemented
UR1:	" "
UR2:	" "
UP1:	" "
UP2:	" "

An initial operating system is ready to be loaded from the Distribution Diskettes. This operating system is configured by Megatel to operate under a "worst-case" system configuration- that is, a hardware configuration consisting of only one single-sided low-performance floppy disk drive, and either a keyboard/CRT console device or a RS-232C terminal.

After the initial operating system has been successfully loaded from one of the Distribution Diskettes, the actual parameters of the "target" system may be entered using the submit procedure QINSTALL. The QINSTALL submit file contains menu-driven CP/M customization programs which have been written by Megatel specifically for use with the QUARK. QINSTALL allows the QUARK user to change the basic I/O configuration of the CP/M operating system without requiring a thorough understanding of the operation of the BIOS or of the programming of the Z80B.

With QINSTALL the user may modify the parameters in the operating system for optimum performance with his final system hardware. As well, various I/O ports not needed for functions assumed by the Distribution operating system may be freed for other uses. Finally, other system parameters, such as the baud rates to be initialized when the operating system is loaded, can be specified through QINSTALL.

The CP/M operating system supplied on the Distribution Diskettes is configured to support one physical drive (i.e. a single floppy disk drive unit) and four logical drives, named A:, B:, C: and D:. Before installation of the system, the disk formats of all logical drives A:, B:, C:, and D: will be compatible with the "Megatel format" used on the Distribution Diskettes.

By running the installation procedure QINSTALL, the user can customize up to four physical drives and four logical drives.

Because different logical drives can access the same physical drive, the user must ensure that the correct diskette is inserted into the correct disk drive.

The Megatel QUARK Software Package

Three classes of utility routines and other files are included in the software package for the QUARK. In the following sections those commands and routines which are part of the CP/M operating system, utilities written by Megatel specifically for the QUARK, and CP/M Users' Group files are described.

Note that a copy of the customer's serialized CP/M system is included on each of the diskettes shipped with the software package, so it is possible to "boot up" from any of these diskettes.

CP/M Commands and Utilities

The following utilities are provided by Digital Research for CP/M 2.2 and are included on the QUARK Distribution Diskettes. A full discussion of the use of these utilities can be found in the CP/M 2.2 Manual. Utilities identified as "Built-in" are part of the CP/M operating system, whereas other utilities are ".COM" files on the diskette.

DIR (Built-in)

This command is used to display the directory of a diskette.

ERA (Built-in)

This command is used to erase files from a diskette. For the QUARK CP/M implementation, this command has been modified to display the names of those files which are being erased.

REN (Built-in)

This command is used to rename a file on a diskette.

TYPE (Built-in)

This command displays the contents the named ASCII file on the CONSOLE device. For the QUARK CP/M implementation, this command has been modified to format the output into pages. When the display is paused between pages, typing any key will cause the next page to be displayed. Typing Control-C will abort the TYPE command.

SAVE (Built-in)

The SAVE command allows the user to save the contents of a part of the Main Memory on a diskette. The SAVE command creates a file on the diskette, and writes into the file the specified number of pages, starting from location 0100hex.

USER (Built-in)

The USER command allows access to different user areas. User 0 is automatically logged on start up.

PIP.COM

The Peripheral Interchange Program is used to move files from one logical device to another. It can be used, for instance, to copy files from one diskette to another, or for printing a file.

DDT4.COM

The Dynamic Debugging Tool can be used to trace the execution of programs, to write, modify,

load, or save files, and to examine and modify CPU registers or areas of memory.

ED.COM

ED is the CP/M line editor. It can be used in the creation of source files to be assembled.

ASM.COM

ASM is an 8080-code assembler. Only 8080 mnemonics are recognized.

LOAD.COM

LOAD is used to load ".HEX" files (such as those produced by the ASM assembler into memory) and produces an executable file with the extension ".COM". A restriction on the .HEX file is that the file must have an ascending sequence of addresses. LOAD displays the size of the file it loads as a convenience for use with a subsequent SAVE command.

SUBMIT.COM

SUBMIT allows CP/M commands to be batched for automatic processing.

SUB.COM

SUB allows CP/M commands to be batched for automatic processing. Console input is accepted in SUB.

XSUB.COM

XSUB allows CP/M commands to be batched for automatic processing. XSUB will intercept CP/M BDOS call for Console Buffer Input, thereby allowing the console responses to be batched as well.

XDUMPD.COM

This utility displays the contents of the named file in hexidecimal and ASCII form.

MOVCPM.COM

This utility allows the user to reconfigure the CP/M system for any particular memory size. Please note that this version does not contain the QUARK BIOS and therefore will not operate in the auto-execute mode.

ERAQ.COM

Queries the user before erasing single or groups of disk files.

STAT.COM

Provides general statistical information about file storage. STAT can also be used to examine and alter device assignments.

Megatel Utilities and Files

The following utilities are written by Megatel and are included on the QUARK Distribution Diskettes.

QINSTALL.COM, QINSTALL.MSG

CP/M customization/installation utility, and message file.

QINSTALL.SUB

CP/M submit file containing the commands and files to be executed during the operating system customization and installation procedure.

QASETUP.SUB

Submit file containing the commands and files to be executed when an ALPHANUMERIC terminal driver is being installed.

QGSETUP.SUB

Submit file containing the commands and files to be executed when a GRAPHICS terminal driver is being installed. On boards with only 64K of RAM this file is not used because the graphics terminal emulation is not available.

QCERT.COM

Diskette formatting utility.

QCOPY.COM

Diskette-to-diskette copy utility. Multiple track buffering permits rapid copying of entire diskettes with same storage capacity and format.

QSYSGEN.COM

System generation program. Used to copy a CP/M system onto a formatted diskette.

QDSKTWO.COM

Program to patch the CP/M system in memory to enable the use of a second physical drive in the operating system before installation. The CP/M system as distributed allows for a one-physical-drive set up.

QSYS.DAT

File containing default parameters for standard distribution diskette, which is updated with user-supplied parameters after running the QINSTALL procedure.

QBOOT.ASM

The QUARK terminal driver bootstrap assembler source listing. The function of the boostrap is to load and initialize the operating system.

QBIOS.ASM

Basic I/O routines in 8080 assembler mnemonics and Z80-compatible machine-language instructions. This assembly-language file is modified by QINSTALL.

MEMNAGE.ASM

Memory management module for the BIOS.

ALPHTERM.ASM

Alphanumeric terminal emulator.

GRPHTERM.ASM

Graphics terminal emulator.

DFCU.COM, DFCU.MSG

Disk format configuration utility and message file which allows the user to reconfigure disk drive formats at run time. On boards with only 64K of RAM this file is not used because the graphics terminal emulation is not available.

QCPM.SYS

User's CP/M System File, needed by the QINSTALL procedure.

CHRLD.COM

Character set loader. Can be used to load standard or custom character sets.

CHR.DAT

Standard character set supplied on the Distribution Diskettes.

CHRED.COM

Character set editor. Can be used to modify existing character sets or create new ones.

QBAUD.COM

The Full-duplex and Simplex serial interface configuration utility. Allows baud rates and operating modes for these serial interfaces to be configured after the system is loaded.

CCP.DOC

Text file containing description of features of the Console Command Processor.

QTCONFIG.COM

Utility for configuring the terminal control codes used by the QUARK operating system's video driver. The video driver may be configured to emulate the control codes used on the Televideo 920 terminal, the standard Megatel control codes, or the user's own codes.

CP/M Users' Group Utilities

The following utilities are CPMUG (CP/M Users' Group) utilities, and have been modified for operation on the QUARK.

DU.COM

DU is a disk dump utility.

SWEEP.COM

SWEEP is a file maintenance utility.

COMPARE.COM

Utility used to compare two files, byte by byte.

QMOD790.COM

QMOD790 is a version of MODEM7 configured for use with the QUARK.

Installing the Customized CP/M System

To generate a customized CP/M system for the target hardware configuration the following steps should be performed:

1. Make two sets of back-up copies of the Distribution Diskettes. One of these sets of copies will be used in the course of the Installation. The other is a back-up copy for security purposes.

2. Start the first step of the automatic installation procedure, QINSTALL. This procedure is in the form of a CP/M SUBMIT file. The procedure will query the user regarding the parameters of the target system, and use the responses to the queries to create modified versions of QSYS.DAT and QBIOS.ASM.

3. Start the second installation procedure. QASETUP.SUB is used when an alphanumeric terminal driver is being installed, while QGSETUP.SUB is used for the graphics terminal driver. These procedures will assemble and link all files required to create the target CP/M operating system.

4. Using the DFCU (Disk Format Configuration) utility, temporarily assign the "target" system drive A format to be accessed by drive B.

5. Format a diskette using QCERT. Write the new CP/M system onto the diskette in drive B, then transfer all files from the Distribution Disk over.

6. Finally, re-boot the system with the diskette created in drive B. The CP/M system loaded into memory will be the new CP/M system, customized for the target system configuration.

This completes the list of steps needed to install a CP/M system for the target QUARK system. Each of the steps is described in greater detail in the next several sections.

Throughout the software examples in this manual, all **boldface** text indicates a user response to a prompt or query from the computer. All prompts or communication which will be displayed on the screen will be indented. This convention is also followed in the CP/M Operating System Manual.

Operation of a one-drive system

This section is for users intending to set up a system with only one floppy disk drive. If more than one drive is to be used, this section may be ignored.

Although it is possible to use the QUARK with only one floppy disk drive, most users will find this configuration inconvenient because of the extra effort involved when making copies of files or entire diskettes. On such a one-drive system, a substantial amount of time will be spent removing and inserting diskettes in the single drive. However, if only one drive is available, it is nonetheless possible to bring up a prototype system and operate the QUARK.

With a one-drive system, all of the logical drives access the same physical drive unit. With drive A: in operation (which is always the case when the CP/M system is first booted up), to access drive B: requires the removal of the diskette in the drive and the insertion of the diskette which is to be accessed on logical drive B:. Whenever logical drive B: is to be accessed after having previously accesed drive A:, the following prompt will be displayed on the screen:

Please mount disk B in drive 0, press <RET> when ready.

This prompt indicates that the diskette then in the drive should be removed, and that the "B:" diskette should be inserted. (<RET> indicates that the carriage return key should be typed.)

When logical drive A: is again accessed, the following prompt will be displayed:

Please mount disk A in drive 0, press <RET> when ready.

At this point, the "B:" diskette should be removed from the drive, and the "A:" diskette (the original diskette) be re-inserted in the drive. This disk-swapping procedure will take place any time a logical drive other than the presently logged-in drive is accessed. The same holds true for drives C: and D:. **It is essential that when diskettes are swapped that the "A:", "B:", "C:" and "D:" diskettes do not become confused.**

Temporary enabling of the second drive on a two-drive system

As stated previously, the CP/M operating system on the Distribution Diskettes is configured to support only one physical disk drive. Such a system can be tedious to use because of the necessity of removing and inserting diskettes into the single drive when copies of diskettes are being made. However, the CP/M system on the Distribution Diskettes can be patched directly to allow the use of a second physical drive when accessing B:, C:, or D:.

To patch the system for two physical disk drives, the following program should be run:

-QDSKTWO.COM which patches the system in memory, so that the operating system and the utilities will recognize a second drive.

Note that the patches performed by QDSKTWO are temporary.

Formatting diskettes to make copies of the Distribution Diskettes

---

Before the QINSTALL program is run, a secure backup of the QUARK Distribution Diskettes should be made in the same format as is used on the Distribution Diskette. It is advised that two copies of each diskette received be made.

To make the copies, it is necessary first to format the appropriate number of diskettes on which the copies will be made. The utility QCERT.COM is used for this purpose.

When using a one drive system please ensure that the correct diskette is inserted in the drive

To run QCERT, the example below should be followed:

```
A0>QCERT

MEGATEL DISC FORMATTER ROUTINE

Which drive, (A - B, then RETURN)? B

Formatting disk B with ..............

Put disk in B: and press return to continue  <ret>

        FORMATTING BEGUN ON DRIVE B
```

QCERT will indicate the number of tracks and sectors used in the format in the space indicated by "..............".

When QCERT is complete, the following prompt will appear, to which the user should reply by typing the return key to re-boot the operating system, or by typing any other key to format another diskette.

****** HIT RETURN to Reboot, any other key to restart

The user should format extra diskettes using QCERT. These diskettes will be used to make backup copies of the Distribution Diskette. It is essential that the diskettes be inserted in the "B" drive and not in the "A" drive. To ensure that the Distribution Diskette will not be accidentally erased, the write-protection feature on the diskette should be enabled if the floppy disk drives in use recognizes a write-protected diskette. (The user is reminded that covering the notch will write-protect 5.25-inch diskettes, while exposing the notch will write-protect 8-inch diskettes.)

Making backup copies of the Distribution Diskette

The utility routine QCOPY.COM is included on the distribution diskettes. This utility is used primarily to make backup copies of diskettes. Two sets of copies of the Distribution Diskettes should be made on diskettes formatted using the QCERT.COM utility. To make the copies of the Distribution Diskettes, insert each diskette in drive A, and follow the example shown below.

When using a one drive system please ensure that the correct diskette is inserted in the drive

```
A0>QCOPY

MEGATEL DISK COPY UTILITY

Please enter the source disk to be copied  A

Please enter the destination disk to be copied to  B

Enter  (Y/N)? Y

Copy done

A0>
```

This procedure must be performed once for each copy of the Distribution Diskettes. When completed, two sets of copies of the Distribution Diskettes should have been made, complete with the CP/M operating system. The Master Distribution Diskettes should now be stored in a safe place. The remaining two sets of diskettes should be labeled the same as the distribution copies, with one set labeled "Work Copy" and the other "Back Up Copy".

The "Work" copies of the diskettes will be used during the installation of the operating system. Some of the files on these disks will be modified by the Installation procedure. The "Back-up" copies are intended as additional security against accidental loss of data.

This section should not be attempted without having made additional copies of the Distribution Disks.

To run the QINSTALL procedure, specifications for the floppy disk drives to be used should be at hand. The QINSTALL program will require some information regarding the characteristics of the disk drives in order to provide a nearly-optimal implementation of CP/M.

Because of the numerous system configurations possible with the QUARK, only a limited set of combinations can be created using QINSTALL. Most users should find the configurations provided by QINSTALL adequate. Advanced users may wish to modify the BIOS source themselves to tailor their operating system more closely to their needs.

To run the installation procedure, insert the following:

```
8 inch      - "Work" System disk in drive A
5.25" 96tpi- "Work" System disk in drive A
             "Work" Installation Source disk in drive B
5.25" 48tpi- "Work" System disk in drive A
             "Work" Installation Source 1 in drive B
             "Work" Installation Source 2 in drive C
```

When using a one drive system with 5.25 inch drives, the submit file QINSTALL will prompt the user to insert either disk A or B. Disk A is the "System" and B is the "Installation Source". Please ensure you read the prompt that is displayed on the screen and insert the correct diskette in the drive.

Type the following:

```
A0>SUB QINSTALL
```

This "SUBMIT" command causes a number of CP/M commands and programs to be executed in a sequential fashion. The file QINSTALL.SUB contains the list of commands to be automatically executed. The program QINSTALL.COM will be the first to execute in the procedure list. QINSTALL.COM will clear the screen and display its version number, and then produce the queries described below.

The default values for each query will appear at ENTER. Initially, the actual value used as the default value depends on the disk size distributed. Entering a "carriage return" as the response to a query will cause the displayed default value to be entered.

To install the operating system properly, **one must select and answer all queries** from options A, B, and C. Each query requires a response, the default values are included in the manual after each question. Option D must then be selected to update to continue with the alteration and assembling of the source files.

The first procedure invoked by QINSTALL.SUB will display the following message:

```
SYSTEM INSTALLATION PROGRAM Version 2.23

Please select option A, B, or C to enter the CP/M
system's configuration for your MEGATEL QUARK.
When you have completed the configurations, select
option D to continue with the installation
procedure or select option X to abort.  If option
D is selected, all user input will be saved in
QSYS.DAT, and will be used as defaults for the
next installation run.

A - DISK DRIVE HARDWARE SPECIFICATIONS
B - DISKETTE FORMAT SPECIFICATIONS
C - OTHER PERIPHERALS CONFIGURATION
D - CONTINUE THE INSTALLATION PROCEDURE
X - ABORT PROCEDURE

ENTER -
```

Each of the above menu selections is described in the sections which follow.

Selection "A" - Disk drive hardware specifications

When **A** is chosen, the following queries will appear on the screen and must be answered:

```
SELECT DRIVE UNIT TO BE DEFINED
 1.  Drive 0 on SEL 0
 2.  Drive 1 on SEL 1
 3.  Drive 2 on SEL 2
 4.  Drive 3 on SEL 3
 5.  Return to main menu
Enter -


DRIVE UNIT 0 SPECIFICATIONS
 Define drive to be the same as
 1.  Drive 0
 2.  Drive 1
 3.  Drive 2
 4.  Drive 3
 5.  To be defined
Enter -


Type of drive
 1.  5 1-4 inch floppy disk
 2.  8 inch floppy disk
 ENTER -


Number of Tracks Per Inch
 1.  96 TPI
 2.  48 TPI
Enter -
```

```
Stepping rate mode
 1.  Use hardware stepping rate
 2.  Use software stepping rate
Enter -


If hardware stepping rate
    5.25 inch            8 inch
 1.  6  ms               3  ms
 2.  10 ms               6  ms
 3.  15 ms               10 ms
 4.  30 ms               15 ms
 ENTER -


If software stepping rate
    5.25 inch     8 inch
 1.  2  ms        1  ms
 2.  4  ms        2  ms
 3.  6  ms        3  ms
 4.  8  ms        4  ms
ENTER -


Motor start time in milliseconds
 Minimum of 0.1  msec
 Maximum of 1000 msec
Enter 'X' if the above is not required
ENTER -
```

Enter the time required for the disk drive to come up to full speed.

```
Wait time for the head load operation
 Minimum of 0.1  msec
 Maximum of 1000 msec
 ENTER -
```

This is the delay inserted between the selecting of a drive and the beginning of disk operations.

```
Head settling time after positioning
 Minimum of 0.1  msec
 Maximum of 200  msec
 ENTER -
```

This is the delay inserted after any disk operation in which the heads are moved.

```
Delay between drive selects in msec.
 Minimum of 0.1  msec
 Maximum of 1000 msec.
 Enter 'X' if the above is not required
ENTER -


Tunnel erase delay in milliseconds
 Minimum of 0.1  msec
 Maximum of 10   msec
 ENTER -
```

The write head must not be allowed to move from track to track unless the tunnel erase is turned off following a disk write operation. The "Tunnel-erase delay" is the time required for the drive to turn off the tunnel erase after a write operation.

This completes the queries for drive "0". QINSTALL will then continue querying the user about the second, or "1", physical drive. The first question of the QINSTALL utility will appear on the screen:

```
SELECT DRIVE UNIT TO BE DEFINED
 1.  Drive 0 on SEL 0
 2.  Drive 1 on SEL 1
 3.  Drive 2 on SEL 2
 4.  Drive 3 on SEL 3
 5.  Return to main menu
Enter -
```

The user can continue to define drives by choosing number 2 Drive 1 on SEL 1 or choose number 5 to return to the main menu. If number 2 is chosen the second question will appear on the screen:

```
DRIVE UNIT 1 SPECIFICATIONS
 Define drive to be the same as
 1.  Drive 0
 2.  Drive 1
 3.  Drive 2
 4.  Drive 3
 5.  To be defined
Enter -
```

The user can choose 1 which will take the parameters entered for Drive 0 or if 2 is entered (Drive 1) all the questions asked previously will be asked again.

Selection "B" - Diskette formatting specifications

When B is chosen from the Main Menu (DISKETTE FORMAT SPECIFICATIONS), the following queries will appear:

```
SELECT LOGICAL DRIVE TO BE DEFINED
 1.  Drive A
 2.  Drive B
 3.  Drive C
 4.  Drive D
 5.  Return to main menu
ENTER -
```

```
LOGICAL DRIVE A FORMAT
 Define drive to be the same as
 1.  Drive A
 2.  Drive B
 3.  Drive C
 4.  Drive D
 5.  IBM 3740 format
 6.  KAYPRO II
 7.  MEGATEL 5.25" 48 TPI (CP/M 2.23)
 8.  MEGATEL 5.25" 96 TPI (CP/M 2.23)
 9.  MEGATEL 8"           (CP/M 2.23)
10.  MEGATEL 5.25" 48 TPI (CP/M 2.22)
11.  MEGATEL 5.25" 96 TPI (CP/M 2.22)
12.  MEGATEL 8"           (CP/M 2.22)
13.  To be defined
ENTER -
```

You can make logical drive A through D to be same as any of the above formats, or you can define your own format. However, we recommend that you maintain at least one drive (say drive C) to have the same format as your distribution disk, or data transfer from the distribution disk to your own format after the installation will be ~~impossible.~~ possible only with the use of DFCU utility.

```
Single or double density
 1.  Single density
 2.  Double density
 ENTER -
```

```
Single or double sided
 1.  Double sided
 2.  Single sided
 ENTER -
```

```
Double tracking
 1.  Use double tracking
 2.  Do not use double tracking
 ENTER -
```

If double tracking is invoked, two step pulses will be issued each time the heads move in or out one track, so that alternate physical tracks are skipped. This feature allows 48 TPI diskettes to be read on a 96 TPI drive.

```
Physical sector size
 1.  128  bytes
 2.  256  bytes
 3.  512  bytes
 4.  1024 bytes
ENTER -
```

```
Number of physical sectors per track
 Minimum of 1
 Maximum of 48
ENTER -
```

The physical sector is the unit of data read or written by the floppy disc controller. The following table shows the estimated byted per track available:

Recording method	8 inch	5.25 inch
Single Density (FM)	5208	3125
Double Density (MFM)	10416	6250

```
Skew factor
 Minimum of 1
ENTER -
```

```
Gap III Size
 Minimum of 4   BYTES
 Maximum of 100 BYTES
ENTER -
```

The gap size is the number of bytes between the end of the data field and the ID mark. Its purpose is to act as a "safe zone" to allow for uncertainty in the position of the end of the data field.

```
Block size
 1.  1024  bytes
 2.  2048  bytes
 3.  4096  bytes
 4.  8192  bytes
 5.  16384 bytes
ENTER -
```

```
Number of directory entries
 Minimum of 64
 Maximum of 4096
ENTER -
```

```
Total number of tracks
 ENTER -
```

This is the total number of logical tracks. If a double sided 8 inch format is used and there are 77 tracks per side then the response to this query would be 154.

```
Number of reserved tracks
 ENTER -
```

The reserved tracks are those that are set aside for the operating system. The Megatel distribution diskettes use (#) reserved tracks for 8 inch and 4 reserved tracks for 5.25 inch. The Megatel operating system requires about 150 sectors at 128 bytes per sector.

```
This logical drive is selected by
 1.  SEL 0
 2.  SEL 1
 3.  SEL 2
 4.  SEL 3
ENTER -


Do you wish to allocate additional
 BIOS memory for this drive
 1.  Yes
 2.  No
ENTER -
```

The special utility DFCU allows the user to reconfigure a logical drive format at run time. If through the QINSTALL the user has defined a drive with smaller capacity and then intends to use the DFCU utility to read/write a disk format of greater capacity, then additional memory will be required. To ensure the success of the DFCU utility the user should answer Yes to the above questions which will automatically allocate 128 bytes for the translate table, check sum and allocation vectors of the drive being defined.

This completes the queries for "A" drive. QINSTALL will then continue to prompt the user to define another logical drive. The first question of the QINSTALL utility will appear on the screen:

```
SELECT LOGICAL DRIVE TO BE DEFINED
 1.  Drive A
 2.  Drive B
 3.  Drive C
 4.  Drive D
 5.  Return to main menu
ENTER -
```

The user can choose to define more drives or return to main menu.

Selection C - Other peripherals configuration

When C is chosen from Main Menu, the following queries will appear.

```
Hardware board revision number
 1.  05R00 (Revision 5)
 2.  REV 1.0, REV 2.0, REV 3.0, REV 4.0
ENTER -
```

The hardware revision number can be found on the solder side of the pc board beneath the Z80 chip.

```
CON: Console input and output
 1.  TTY:  Input from and output to the
           full-duplex serial port
 2.  CRT:  Input from the parallel keyboard port
           and output to memory mapped crt
 3.  BAT:  Input from full-duplex serial port
           and output to memory mapped crt
 4.  UC1:  Combined functions of 1 and 3
ENTER -
```

```
LST: List device output to
 1.  TTY: The full-duplex serial port
 2.  CRT: Memory mapped crt
 3.  LPT: The Centronics parallel printer port
 4.  UL1: The simplex serial port
ENTER -


List device end-of-line to next-line delay
 Maximum of 255  NULLS
 Enter 'X' if the above is not required
ENTER -
```

Some printers require that there be a delay between sending a carriage return character to the printer and the transmission of the next printable character. The number of "null" characters (=00 hex) sent after a carriage return can be specified here.

```
Simplex Serial Port baud rate
 1.  9600 baud
 2.  7200 baud
 3.  4800 baud
 4.  3600 baud
 5.  2400 baud
 6.  1800 baud
 7.  1200 baud
 ENTER -
```

The response to this query determines the baud rate initialized for the Simplex Serial Port when the system is booted.

```
Full duplex serial port baud rate
 1.  9600   baud
 2.  7200   baud
 3.  4800   baud
 4.  3600   baud
 5.  2400   baud
 6.  1800   baud
 7.  1200   baud
 8.   600   baud
 9.   300   baud
10.   150   baud
11.   134.5 baud
12.   110   baud
13.    75   baud
14.    50   baud
 ENTER -
```

The response to this query determines the baud rate initialized for the Full-duplex Serial Port when the system is booted.

```
Video display terminal emulation mode.
 1.  Alphanumeric mode
 2.  Graphic mode
ENTER -
```

The response to this query will determine whether the Alphanumeric Mode or the Graphics Mode of the Video Display Interface is used to display the CRT alphanumeric data. Note that the

graphics emulation mode is not implemented if you have a 64k board.

```
Number of rows for the DISPLAY screen
 Enter 'X' if the above is not required
ENTER -


Number of rows for the MAIN screen
 Enter 'X' if the above is not required
ENTER -


Number of rows for the STATUS screen
 Enter 'X' if the above is not required
ENTER -


If the alphanumeric mode is chosen
 1.  Enable the clock display
 2.  Disable the clock display
ENTER -


If the clock displayed is enabled
 1.  Display at top left corner
 2.  Display at bottom right corner
ENTER -


 If the Graphics display mode is chosen, select
  column offset from the left edge of the screen
  Minimum of 1    colms
  Maximum of 7    colms
  Enter 'X' if the above is not applicable
 Enter -
```

The number of (7-dot-wide) columns specified determines the number of blank or background characters that will appear between the left-most edge of the video display and the left-most character of the "working" area of the CRT. This can be used to create a border around the working area, or to extend the horizontal blanking period.

When menu options **A**, **B**, and **C** have been complete to the user's satisfaction, the **D** option should be chosen.

Function D will continue with the QINSTALL submit procedure. The first file to be altered is the system parameter file QSYS.DAT. The default values obtained from the copy of QSYS.DAT on the Distribution diskette will be replaced by the parameters entered during QINSTALL which were, until this point, stored in memory.

QINSTALL will then alter the files QBIOS.ASM (the source file for the QUARK CP/M Basic I/O System) and QBOOT.ASM (the source file for the operating system boot loader). Then ALPHTERM.ASM

When a one drive 5.25 inch system is being used, QINSTALL will prompt the user to insert either disk A or B at specific times during the alteration of the source files. It is very important to follow the prompts carefully to ensure a working system, remembering that A is the "System" disk and B is the "Installation Source" disk.

In the case of 5.25" 48 TPI software, the "Installation Source 2" disk will be required during the installation procedure. This disk will be mounted as disk C.

Second Installation procedure

The second part of the Installation procedure requires another submit to be executed. The submit to be executed is dependent on the mode chosen for the video output. If the alphanumeric mode is to be used then QASETUP.SUB is used. If the graphics mode is to be used QGSETUP.SUB is executed. To continue with the procedure enter one of the following:

```
A>SUB QASETUP
```

if the alphanumeric mode terminal driver is being installed,

-or-

```
A>SUB QGSETUP
```

if the graphics mode terminal driver is being installed.

The second part of the installation procedure will assemble and link all pertinent files. At the completion of this section the new system will be written to a file called QCPM.SYS.

Final steps in the Installation procedure

At this point a temporary CP/M system will be required to transfer the system and data from the distribution disk to the target system drive. The following system is required:

-Drive 'A' format should be the same as the Distribution disk format

-Drive 'B' format should be the same as the user's target system drive 'A'

This way the user can access his own format through drive B while operating under this temporary system. Additionally, he can read and write to the Distribution Diskettes through drive A.

To create the temporary system as described above the DFCU utility is used. Follow the example below:

```
A>DFCU QSYS.DAT

DISK FORMAT CONFIGURATION UTILITY VERSION 2.23

DFCU prompts user for the specifications of
disk drives and disk formats, and lets the user
configure the disk drive parameter tables
in memory directly, all without the need to
reassemble or regenerate the O/S. However
the configuration is only temporary until
the next system reset.

          MAIN MENU SELECTION

A. DISK DRIVE SPECIFICATIONS AND OPERATIONS
B. LOGICAL DISK FORMATS AND OPERATIONS
X. RETURN TO O/S

ENTER -B
```

```
SELECT LOGICAL DRIVE TO BE DEFINED
 1.  DRIVE A
 2.  DRIVE B
 3.  DRIVE C
 4.  DRIVE D
 5.  Return to main menu
ENTER -2


LOGICAL DRIVE B FORMAT
 Define drive to be the same as
 1.  DRIVE A
 2.  DRIVE B
 3.  DRIVE C
 4.  DRIVE D
 5.  IBM 3740 format
 6.  KAYPRO II
 7.  MEGATEL 5.25" 48 TPI (CP/M 2.23)
 8.  MEGATEL 5.25" 96 TPI (CP/M 2.23)
 9.  MEGATEL 8"           (CP/M 2.23)
10.  MEGATEL 5.25" 48 TPI (CP/M 2.22)
11.  MEGATEL 5.25" 96 TPI (CP/M 2.22)
12.  MEGATEL 8"           (CP/M 2.22)
13.  To be defined
ENTER -1


This logical drive is selected by
 1.  SEL 0
 2.  SEL 1
 3.  SEL 2
 4.  SEL 3
ENTER -2
```

Or another SEL line to which a physical drive is wired to read and write the target system's drive 'A' format.

```
CONFIGURE LOGICAL DRIVE B PARAMETER TABLES IN
MEMORY ACCORDING TO SPECIFICATIONS (Y/N)
ENTER -Y
```

At this point the DFCU menu will be displayed. The user should return to the main menu and exit by enter **X**. Do not save the data file as limited disk space is provided on the distribution disks. The temporary system will now be in memory and the following operations can be performed.

New disk format and new CP/M system

After the Installation procedure has been completed, the temporary system in memory will be used to format several diskettes in the user's customized format, and to transfer the new system from the "Work" diskettes, which are in the Megatel format, to the user's formatted diskettes.

Note that it is important not to reset the QUARK while the temporary system is in memory, as a reset would cause a copy of the Distribution Diskette's operating system to be loaded and the temporary system to be lost.

It is also important to ensure that while the temporary system is resident and before the user's customized operating system has been written to the "B" diskette that **only** the "A" drive be accessed. If the user attempts to read from any other logical drives, and a properly-formatted diskette is not inserted, an error would result and the QUARK system would have to be reset, resulting in the loss of the temporary system.

If the temporary system is lost, through either using the reset or by accessing the wrong logical drive, the following procedure will recover the temporary system:

-reset (load a copy of the distribution system)

-run **DFCU QSYS.DAT** and answer all questions as before

Formatting a diskette under the new format.

To format a new system diskette with the user's customized format, under the temporary system, simply run QCERT.

When using a one drive system, the system will prompt the user to insert either disk A or B. Disk A is the "System" and B is the "target drive A". Please ensure you read the prompt that displays on the screen correctly.

```
A0>QCERT

MEGATEL DISC FORMATTER ROUTINE

Which drive, (A - B, then RETURN)? B

Formatting disk B with ..............

Put disk in B: and press return to continue  <ret>

        FORMATTING BEGUN ON DRIVE B
```

If any other drive were inadvertently accessed at this point, the temporary system would have to be recovered using the procedure described in above.

Writing the new system on a diskette.

After formatting one or several diskettes with the user's customized format, the customized CP/M operating system may be generated and written onto the system tracks of one of these diskettes. To do this, the program QSYSGEN.COM is used. This file is included on the Distribution Diskette, and hence should also appear on the "Work" disk copy of the Distribution Diskette.

To use the QSYSGEN utility at this point, follow the example presented below.

```
A0>QSYSGEN QCPM.SYS

Megatel QUARK SYSTEM GENERATION UTILITY
DESTINATION DRIVE? B
DESTINATION ON B, TYPE  RET  <ret>
Function complete.
A0>
```

By specifying **QCPM.SYS**, QSYSGEN will read the file QCPM.SYS from drive A: and write the file onto the system tracks of the diskette in drive C:. The file QCPM.SYS was created in the Installation procedure. QSYSGEN waits for the proper disk to be inserted, after which the user types a carriage return to start the system generation process.

After generating and writing the customized CP/M system, this diskette can now be used as the new "system diskette", containing the CP/M operating system together with the bootstrap loader. What remains to be done is transferring of the CP/M and Megatel utility files from the Distribution Diskette (or the "Work" copy) to the user's system diskette and testing of the customized system.

Transferring files from the Work Diskette

The various files may be transferred from one of the copies of the Distribution Diskette to the new System Diskette. This will be done using the CP/M utility PIP (for Peripheral Interchange Program). The file PIP.COM should be on the "Work" diskette. To copy all of the files on the "Work" diskette to the new System Diskette, follow the example below.

```
A0>PIP B:=A:*.*[VOR]
.............
.............
.............
A0>
```

As PIP copies each file from drive A to drive B, the names of the files will be displayed. When PIP is finished, the A0> prompt will be displayed. Having completed the transferring of all of the files, additional copies of the new system diskette may be made on other diskettes formatted using the (customized) QCERT utility.

At this point, the following diskettes should be on hand:

1. The Distribution Diskettes received with the QUARK. These diskettes should be stored in a safe place.

2. An exact set of backup copies of the Distribution Diskettes, labelled "Back-up".

3. A diskette with the CP/M system used on the Distribution Diskette and new utilities, labelled "Work". This diskette, used in running the QINSTALL procedure, can be re-formatted.

4. The new master Target System Diskette together with all backup copies made.

Booting the new system

To verify that the new CP/M system can be properly loaded, insert the new system diskette into the "A:" drive, and reset the QUARK system. If the system operates properly, the new system will be loaded and the "A0>" prompt will be displayed on the screen. At this point, utilities and other files on one of the copies of the Distribution diskette may be copied onto the new system diskette. For details on drive assignments of the new system, please refer to the beginning of the installation procedure.

If the boot fails, improper data may have been entered in the QINSTALL procedure. It is recommended that the user restore all the original files from the Distribution Diskettes to their "Work" disks, and then return to the beginning of the installation section.

QCERT.COM - The diskette formatting utility

QCERT is a diskette formatting program supplied on one of the QUARK distribution diskettes. QCERT allows new or existing diskettes to be formatted. After formatting a diskette, QCERT also checks the format that has just been written by reading back every sector.

The version of QCERT that has been supplied is capable of formatting a diskette in a number of different formats. QCERT obtains a list of parameters concerning the physical drive characteristics and logical disk formats from various BIOS tables in memory.

Operation of QCERT

When QCERT is started (by typing **QCERT** in response to a CP/M prompt), it queries the user for the identity of the logical drive which is to be formatted, as shown below.

```
Which Drive (A - P), then RETURN ?

Formatting drive x with nnn tracks mmm sectors

FORMATTING BEGUN ON DRIVE

    TRACK  nnn
```

If the diskette is formatted successfully, the following message will appear when finished.

```
Hit RETURN to Reboot, any other key to re-start.
```

To abort QCERT at any time, type the **ESC**ape key. If typing this key does not seem to have any immediate effect, type the key repeatedly until QCERT halts and displays an appropriate message.

The following list describes all the error messages that may appear during the operation of QCERT.COM.

Speed Error - x MS. per revolution

This error indicates that QCERT has detected that the rotational speed of the diskette is not within the required range. This is probably a hardware fault in the disk drive and should be corrected.

ERROR - Cannot write physical track n

This indicates that QCERT was not successful in writing track number "n". The most common reason for this fault is because the diskette is write-protected. It might also indicate that there is a fault in the floppy-disk interface hardware, in the cable connecting the QUARK board to the floppy disk drive, or in the diskette itself, and this is stopping the program from writing the data for formatting purposes. This error will only occur in the formatting phase.

Physical track x does not validate

This error occurs during the verification phase when the program finds that it cannot read back the data that it has just written during the formatting phase.

Unable to home drive - ABORTING
Unable to restore drive

These two errors are likely to occur at the same time and indicate a hardware fault. The program has been unable to reset the drive to its normal rest position on track 00. This error will occur before any formatting has taken place.

ABORTING FORMATTER - DISK STATUS = n

This message will always appear after every error, to indicate the status of the disk at the time the error occurred. The value of "n" is the value in the Status Register of the Floppy Disk Controller (part number 1793-02) when the error was detected.

****ERROR HAS OCCURRED****
Please press the down arrow (Control J) to acknowledge

This error message will appear after every error to ensure that the display has been read and acknowledged. QCERT will pause until a line feed key (Control-J) is pressed.

QSYSGEN.COM - The System Generation Utility

QSYSGEN is a utility routine used to put a system image on the system tracks of a given logical drive. QSYSGEN does not destroy or alter any files on the destination diskette, as only the system tracks of the destination diskette are written. It is necessary to "Sysgen" only those diskettes which are to be used as "system" diskettes (i.e., those diskettes from which the QUARK is to booted).

A special version of QSYSGEN has been developed by Megatel for all Megatel floppy disk systems running under CP/M 2.2. This new version offers two advanced features:

QSYSGEN will copy from and write to any diskette with standard or non-standard disk formats;

QSYSGEN will also read and write standard or non-standard configurations of the CP/M modules on the system tracks.

SYSTEM IMAGE CONFIGURATION

When the CP/M 2.23 system is read in by the QSYSGEN either from a file containing the system created by the installation procedures, or from the system tracks of a disk, the system image in memory will look like this :

Memory Address	System Modules
900h	bootstrap - CP/M standard address
980h	ccp, bdos - CP/M standard address
1f80h	bios - CP/M standard address
	- the remaining are stand along or banked bios modules of the O/S
3b80h	system initialization
3f00h	terminal emulator
4a00h	teminal emulator variables and stack
4b00h	memory management module
4e00h	character font data set

TRACK IMAGE CONFIGURATION

We choose to write the system image to the system tracks in the following sequence and size(in 128 byte sectors) :

1) bootstrap - 1 sector
2) system initialization, terminal emulator, emulator variables and stack, and the memory management moduel - 37 sectors
3) ccp, bdos, and bios - 100 sectors
4) character font data set - 16 sectors

A table in QSYSGEN can be patched to resequence the system image when writting to the system tracks, or resequence the track image when reading to memory.

Presently, the tables is filled out as follows :

Tabel Patch Address	Word	Word	
274h	0900h	0001h	
278h	3b80h	0025h	
27ch	0980h	0064h	
280h	4e00h	0010h	
284h	0000h	0000h	- terminate table by 4 bytes of zeros

In case of reading from the system tracks, the table dictates to the program to read the first sector to memory address 900h, the next 37 sectors to address 3b80h and on, the next 100 sectors to address 980h and on, and the next 16 sectors to 4e00h.

In case of writting to the system tracks, the program will write the first sector from memory address 900h, the next 37 sectors from address 3b80h, and so on.

SEQUENCIAL/STANDARD SYSTEM GENERATION

In case the user wish to patch the QSYSGEN program to read/write in sequencial order without any resequencing, fill the table as follows :

Tabel Patch Address	Word	Word
274h	0900h	00a0h
278h	0000h	0000h

DFCU.COM - The Disk Format Configuration Utility

The DFCU program is very similar in function as the QINSTALL program with the following similarities and differences.

SIMILARITIES

1. Both use the same input data structure and ask the same questions for the physical and logical drive definitions.

2. Both serves to change system confiqurations.

DIFFERENT

1. QINSTALL patches source files, DFCU patches memory to configure a drive. The DFCU configuration is temporary.

2. DFCU does not save data file unless requested by user through selction 'X' of the main menu.

3. DFCU uses DFCU.MSG and DFCU.DAT files by default. A data file name other than the default, entered at the command line is accepted, but the data file will always be saved as DFCU.DAT.

4. DFCU does not configure peripherals other than disks.

5. DFCU can use the QSYS.DAT file generated by QINSTALL, but DFCU.DAT is not sufficient to run QINSTALL.

CREATING A DFCU.DAT FILE

Run **DFCU QSYS.DAT,** and select option 'X' to exit, and answer 'Yes' to the save data file prompt.

PLEASE CONSULT INSTALLATION DOCUMENTATION REGARDING THE QUESTIONS.

Software for the Video Display Interface

The utility routine CHRLD.COM, supplied on the distribution diskettes, is used to load a character set for either the alphanumeric or graphic display drivers which are part of the QUARK operating system.

In case of alphanumeric terminal emulation, CHRLD.COM is used to load a character set into the Programmable Character Generator. In case of graphics terminal emulation, the program is used to load a character set into the RAM. This utility can be used to load either the default character set imbedded within the CHRLD.COM file, or any other character set saved in a file on a diskette.

The character set used for alphanumeric display is a full 256-character set, including both normal- and reverse-video characters for the QUARK. The character set is intended to be used with ASCII codes. Each character is stored as eight contiguous bytes within the data file.

For further information on the operation of the Programmable Character Generator or the Video Display Interface, refer to the Hardware section of this manual.

CHRLD.COM - The Character Generator Loader for CP/M Plus

This file will load the default character set, the data for which is contained within CHRLD.COM itself. If the user wishes to load the default character set after the system is up, it is necessary only to enter the following command:

```
A>CHRLD
A>
```

To load another character set into either the Programmable Character Generator or RAM without altering the CHRLD.COM file, specify the character set file name on the command line immediately following the command CHRLD. CHRLD will find the character set file and load the specified character set file. The following is an example of this use of CHRLD.COM:

```
A>CHRLD CHR.DAT
A>
```

The character set "CHR.DAT" set will remain until another character set is loaded, or until power is removed. Character set files can be created or modified using the Character Set Editor CHRED.COM.

To change the default character set within CHRLD.COM a desired character set file must be patched into CHRLD.COM. To perform this patch, load CHRLD.COM at address 0100H, and load the file containing the new character set to address 103h. Make sure the new data set is not greater than 8K. An example is presented below.

```
A>DDT4 CHRLD.COM

#ICHR.DAT
#R103
#GO

A>SAVE 13 TCHRLD.COM
```

Changing Character Set Loaded at Cold-boot

The following procedure shows how to change the character font loaded at cold-boot time.

```
A>DDT4 QCPM.SYS
#ICHR.DAT
#R4D00
#GO

A>SAVE 85 TQCPM.SYS
A>QSYSGEN TQCPM.SYS
DESTINATION DISK ON DRIVE A

A>
```

CHRED.COM - The Character Set Editor

The Character Set Editor utility CHRED.COM is included on the Distribution Diskettes. It can be used to modify existing character sets, or to create new ones. The Editor displays the eight-by-eight matrix of dots forming the character pattern in an eight-by-eight edit frame on the CRT. With a given set of control keys, the user may move a "dot cursor" around within the grid, stopping on any dot and turning the dot on or off. The 8-bit ASCII code which is to represent the character may be entered, and the set of characters may be scanned in ascending or descending order of codes.

The calling syntax of the Character Set Editor is as follows:

```
A>CHRED dr:filename.ext
```

where "dr:" is the optional logical drive on which the character set file can be found, and "filename.ext" is the name of the character set file to be edited.

After CHRLD has been loaded, it will prompt the user for the hexidecimal ASCII code for the character to be edited. When a code is entered, the pattern for that character in the character set file will be displayed in a the large 8-by-8 edit frame, as well as in a small single-character cell below the grid. Also displayed is a "painting cursor". Control keys ^E, ^X, ^S, and ^D will move the cursor up, down, left, or right within the edit frame. Positioning the cursor on any cell of the frame and typing a "space" will cause that cell to be inverted, ie., the dot will be switched on or off. Also, the corresponding pixel in the single-character cell below the edit frame will change to show the actual appearance of the character.

Typing control-C or control-R will move the edit frame onto the next or previous ASCII character. Typing a carriage return allows a new code for the character to be edited to be entered. If "00" is entered as the new code, CHRLD will ask if the edited character set is to be saved. If the response to this prompt is "Y", then it will save the edited file under the filename given when CHRED was called.

A file CHR.DAT is included on the Distribution Diskettes. This file contains a character set identical to the set imbedded within the distributed copy of CHRLD.COM, and can be used as a starting point when creating new character sets.

QTCONFIG.COM - Terminal Code configuration utility

The purpose of this utility is to allow the user to change the terminal control codes used by the terminal driver. QTCONFIG allows the standard Megatel control codes, the user's own set of codes, or the set of control codes used on a Televideo 920 terminal. To run this utility the user should enter:

A>QTCONFIG

The screen will now display:

```
Terminal Emulation Utility Vers. 3.01

    1.  Televideo 920
    2.  QUARK
    3.  User Defined

    ENTER -
```

Option 1 will configure the QUARK to emulate a Televideo 920. Option 2 will configure the QUARK with the Megatel control codes used on the Distribution Diskettes. These control codes are given in Table XI of the Appendix.

Option 3 allows the user to configure his own terminal codes or to load these codes from another file. This option will prompt the user with queries and then save the responses in a user-specified file.

When setting up the user-specified control codes, QTCONFIG will display a description of the terminal function and then allow the user to enter a two-byte sequence representing the control code to be used for that terminal function. If it is desired that a particular function have only a one-byte control code, the desired hexidecimal value for the code should be entered as the first value, and FFhex be entered as the second. Only the first byte entered will then be the control code; the value FF will not become part of the code. If a particular terminal function is not to be implemented, then the value FFhex should be entered for both parts of the code.

Note that the number of character rows displayed by the video driver will not be changed when QTCONFIG is run. Since the standard number of rows displayed on a Televideo 920 terminal is 24, application programs designed to run on a Televideo 920 should be patched to allow for number of rows installed for the "Normal" screen area of the QUARK.

With the QUARK terminal driver the control codes for high intensity will cause the characters to be displayed in reverse video.

ELECTRICAL SPECIFICATIONS

Parameter	60Hz models	50Hz models	units
Master clock frequency	23.86176	24.80000	MHz
Z80B clock frequency	5.96544	6.20000	MHz
Z80B T-state period	167.63223	161.29032	ns
E-clock frequency	1.49136	1.55000	MHz
Write-precompensation:			
8-inch drives	125	125	ns
5.25 inch drives	250	250	ns
Horizontal sync frequency	15.540	16.150	kHz
Horizontal sync period	64.35	61.92	us
H-sync pulse-width	21.45	20.64	us
Horizontal sync polarity	positive	positive	
V-sync pulse-width	187.7	180.6	us
Vertical sync polarity	negative	negative	
Length of Video data	53.6	51.6	us
Percentage line utilization	83.3	83.3	%
Video output amplitude	4.0	4.0	Vp-p±3dB
H-sync output amplitude	4.0	4.0	Vp-p±3dB
V-sync output amplitude	4.0	4.0	Vp-p±3dB
H-sync pulse width	8.0	8.0	us
Front porch (data to start H-sync)	1.8	1.8	us
Back porch (end H-sync to data)	0.7	0.7	us
Composite video amplitude	1.0	1.0	Vp-p±3dB
Composite sync level relative to black level	-0.5	-0.5	V

TABLE I I/O ADDRESSES AND FUNCTIONS FOR THE QUARK

I/O ADDRESS	DEVICE & REGISTER	FUNCTION (READ / WRITE)
00-3F	Character generator	Special procedure must be invoked to write
40-5E	I/O alias	Not recommended for use
5F	Parallel printer output port	-VIA CA1: parallel printer Acknowledge input -VIA CA2: parallel printer Data Strobe output
60	VIA ORB/IRB	Output register B / Input Register B -PB0-5: General-purpose I/O lines PB0 configured for bell output -PB6: Simplex port protocol/data input -PB7: ACIA transmitter/transmitter & receiver clock output
61	" ORA/IRA	Output register A / Input Register A -PA0-7: General-purpose I/O lines
62	" DDRB	Data direction register B
63	" DDRA	Data direction register A
64	" T1C-L	T1 low-order counter / T1 low-order latch
65	" T1C-H	T1 high-order cntr / T1 high-order latch/cntr
66	" T1L-L	T1 low-order latch
67	" T1L-H	T1 high-order latch -T1 counter used as ACIA baud rate generator
68	" T2C-L	T2 low-order counter / T2 low-order latch
69	" T2C-H	T2 high-order counter -T2 counter used as baud rate generator for VIA Shift Register (simplex serial port), or for full-duplex serial port transmitter with split baud rates, or as a timer
6A	" SR	Shift register for simplex serial port -CB1: SR clock from PB7 if J3 & J4 installed -CB2: Tx DATA output for simplex serial port
6B	" ACR	Auxiliary control register
6C	" PCR	Peripheral control register -CA1: Parallel printer acknowledge input -CA2: Parallel printer data strobe output
6D	" IFR	Interrupt flag register
6E	" IER	Interrupt enable register
6F	" ORA/IRA	Same as address 61 except no "handshake"
70-73	I/O alias	Not recommended for use

TABLE I I/O ADDRESSES AND FUNCTIONS FOR THE QUARK (CONTINUED)

I/O ADDRESS	DEVICE & REGISTER	FUNCTION (READ / WRITE)
74	PIA PA or DDRA	Peripheral reg A or data direction reg A -PA0-7: 8-bit encoded keyboard input or general-purpose I/O lines
75	" CRA	Control register A -CA1: Vertical Sync interrupt input -CA2: External interrupt input, or keyboard strobe, or bell output
76	" PB or DDRB	Peripheral reg B or data direction reg B -PB0-2: Floppy Disk SEL 0-2 outputs(act. high) -PB3: Floppy Disk SEL 3/LOW CURRENT(act. high) -PB4: Floppy Disk SIDE select output -PB5: Floppy Disk SNGL/DBLE select (high=SNGL) -PB6: GRAPHICS/ALPHA mode bit (high=ALPHA) -PB7: Full-duplex port DTR output (active low)
77	" CRB	Control register B -CB1: Floppy Disk Controller INTERRUPT REQUEST -CB2: Boot Mode: low = normal operation high= PROM selected, RAM deselected
78	ACIA SR/CR	Status register / Control register
79	" RDR/TDR	Receive data register / Transmit data register -VIA Timer 1 sets baud rate on transmitter and receiver under non-split baud rates, receiver-only under split baud rates -VIA Timer 2 sets transmitter baud rate under split baud rates
7A-7F	I/O alias	Not recommended for use
80	FDC STR/CR	Status register / Control register -FDC interrupts are sent to PIA CB1 input
81	" TR	Track register
82	" SR	Sector register
83	" DR	Data register
84-8F	I/O alias	Not recommended for use
90	LAN STS/CMD (Q/200 only)	Local-area network status/command register -stat reg bit 7: LAN READY flag (active high) -control block address for Omninet interface
91-97	I/O alias	Not recommended for use.

HI = side 0
LO = side 1

at ~~DRQ~~ IRQ ?

CB7 is actually IRQ

same as PIA ; but select port is 76h vice 84h

TABLE I I/O ADDRESSES AND FUNCTIONS FOR THE QUARK (CONTINUED)

I/O ADDRESS	DEVICE & REGISTER	FUNCTION (READ / WRITE)
98	LAN INTERRUPT (Q/200 only)	LAN interrupt status/control register -status reg bit 7=1: LAN interrupt pending -control reg bit 7=1: LAN interrupts enabled -control reg bit 7=0: LAN interrupts disabled -writing to this port clears LAN interrupt
99-BF	I/O alias	Not recommended for use.
C0-FF	User /CS output	Active low output on pin A-14 of the ESIC connector. Use to select a single external peripheral device, or to qualify decoding of address lines for multiple external peripheral devices.

Notes:
1. For software compatibility between current and future QUARK single-board computers, it is recommended that I/O addresses indicated as "Reserved" or "I/O alias" not be used in programming the QUARK.
2. The Parallel Printer Data Strobe output is driven an inverting TTL buffer on version 04R01 and earlier QUARK/100s, on revision 05R00 and later QUARK/100s, this output is driven by a non-inverting buffer.

TABLE IIa TIMER-1 BAUD RATES FOR FULL-DUPLEX INTERFACE - 50HZ VERSION

BAUD RATE	DIVIDE BY 1 VALUE	(HEX)	ERROR	DIVIDE BY 16 VALUE	(HEX)	ERROR	DIVIDE BY 64 VALUE	(HEX)	ERROR
19200	39	0027	-.95	1	0001	-8.26	-1	000-1	-15.91
9600	79	004F	-.03	3	0003	6.22	0	0000	-27.92
7200	106	006A	-.10	5	0005	-.33	0	0000	-3.89
4800	160	00A0	-.18	8	0008	3.50	1	0001	-8.26
3600	214	00D6	-.22	C	000C	-2.15	2	0002	-10.30
2400	321	0141	.05	18	0012	2.19	3	0003	6.22
1800	429	01AD	-.05	25	0019	.60	5	0005	-.33
1200	644	0284	.01	39	0027	-.95	8	0008	3.50
600	1290	050A	-.01	79	004F	-.03	18	0012	1.55
300	2582	0A16	-.02	160	00A0	-.18	39	0027	-.95
150	5165	142D	.00	321	0141	.05	79	004F	-.03
134.5	5760	1680	.01	358	0166	.11	88	0058	.31
110	7044	1B84	.00	439	01B7	-.09	108	006C	.31
75	10332	285C	.00	644	0284	.01	160	00A0	-.18
50	15498	3C8A	.00	967	03C7	.00	240	00F0	.18

* Not recommended. Use the Divide-by-16 mode.

TABLE IIb TIMER-1 BAUD RATES FOR FULL-DUPLEX INTERFACE - 60HZ VERSION

BAUD RATE	DIVIDE BY 1 VALUE	(HEX)	ERROR	DIVIDE BY 16 VALUE	(HEX)	ERROR	DIVIDE BY 64 VALUE	(HEX)	ERROR
19200	37	0025	.23%	1	0001	-11.73%	—	—	—
9600	76	004C	-.10	3	0003	2.20	—	—	—
7200	102	0066	-.18	5	0005	-4.10	—	—	—
4800	154	009A	-.26	8	0008	-.42	1	0001	-11.73
3600	205	00CD	.19	B	000B	1.54	2	0002	-13.69
2400	309	0135	-.02	18	0012	-1.68	3	0003	2.20
1800	413	019D	-.12	24	0018	.55	5	0005	-4.10
1200	620	026C	-.06	37	0025	.23	8	0008	-.42
600	1241	04D9	.03	76	004C	-.10	17	0011	.23
300	2484	09B4	-.01	154	009A	-.26	37	0025	.23
150	4969	1369	.01	309	0135	-.02	76	004C	-.10
134.5	5542	15A6	.01	345	0159	-.07	85	0055	-.14
110	6777	1A79	.00	422	01A6	-.02	104	0068	.16
75	9941	26D5	.00	620	026C	-.06	154	009A	-.26
50	14912	3A40	.00	930	03A2	.04	231	00E7	.12

* Not recommended. Use the Divide-by-16 mode.

Notes for Table II(a) and II(b)

1. The frequency of the Timer 1-generated clock output on PB7 of the VIA is given by

$$f_E/(2N+3.5),$$

where f_E is the frequency of the E-clock and N is the value in the Timer 1 latch. The baud rate is this frequency divided by 1, 16, or 64, according to the divide ratio bits in the ACIA.

2. Baud rates other than those shown above are possible by loading the Timer 1 latches whith the value determined by the equation above. Consult the data sheets for the VIA (6522A) and the ACIA (68A50).
3. Baud rate errors exceeding out 5% may be unacceptable in some applications. If this is the case, Timer-2 may be used to generate the Transmit and Receive clocks for the Full-duplex port. To do this, install jumpers J3 and J4, set PB7 of the VIA to act as an input, and use the Timer-2 generated baud rates (Table IIIa) for the ACIA Transmit and Receive clocks.
4. In the divide-by-one mode, the ACIA receiver clock should be synchronized with the incoming data. The ACIA transmitter will operate normally in this mode.

TABLE IIIa TIMER-2 BAUD RATES FOR FULL-DUPLEX RECEIVER IN SPLIT BAUD MODE - 50HZ VERSION

BAUD RATE	DIVIDE BY 1 VALUE	(HEX)	ERROR	DIVIDE BY 16 VALUE	(HEX)	ERROR	DIVIDE BY 64 VALUE	(HEX)	ERROR
19200	38	26	.91	1	01	-15.61	0*	00	-68.47
9600	79	4F	-.33	3	03	.91	0*	00	-36.93
7200	106	6A	-.33	5	05	-3.89	0*	00	-15.91
4800	159	9F	.28	8	08	.91	1*	01	-15.91
3600	213	D5	.12	11	0B	3.89	1	01	12.31
2400	—	—	—	18	12	.91	3	03	.91
1800	—	—	—	25	19	-.33	5	05	-3.89
1200	—	—	—	38	26	.91	8	08	.91
600	—	—	—	79	4F	-.33	18	12	.91
300	—	—	—	159	9F	.28	38	26	.91
150	—	—	—	321	—	—	79	4F	-.33
134.5	—	—	—	358	—	—	88	58	.04
110	—	—	—	438	—	—	108	6C	.08
75	—	—	—	644	—	—	159	9F	.28
50	—	—	—	967	—	—	240	F0	.08

* Not recommended. Use the Divide-by-16 mode.

TABLE IIIb TIMER-2 BAUD RATES FOR FULL-DUPLEX RECEIVER IN SPLIT BAUD MODE - 60HZ VERSION

BAUD RATE	DIVIDE BY 1 VALUE	(HEX)	ERROR	DIVIDE BY 16 VALUE	(HEX)	ERROR	DIVIDE BY 64 VALUE	(HEX)	ERROR
19200	37	25	-.42	1	01	-19.09	0*	00	-69.66
9600	76	4C	-.42	3	03	-2.91	0*	00	-39.32
7200	102	66	-.42	5	05	-7.53	0*	00	-19.02
4800	153	99	-.23	8	08	-2.91	0*	00	21.37
3600	205	CD	-.06	11	0B	-.42	1	01	7.88
2400	—	—	—	17	11	2.20	3	03	-2.91
1800	—	—	—	24	18	-.42	5	05	-7.53
1200	—	—	—	37	25	-.42	8	08	-2.91
600	—	—	—	76	4C	-.42	17	11	2.20
300	—	—	—	153	99	.23	37	25	-.42
150	—	—	—	309	—	—	76	4C	-.42
134.5	—	—	—	345	—	—	85	55	-.43
110	—	—	—	422	—	—	104	68	-.08
75	—	—	—	619	—	—	153	99	.23
50	—	—	—	930	—	—	231	E7	.01

* Not recommended. Use the Divide-by-16 mode.

Notes for Table III(a) and III(b)

1. The frequency of the Timer-2 clock output on CB1 of the VIA is given by

$$f_E/(2N+4),$$

where f_E is the frequency of the E-clock and N is the value in the Timer-2 latch. The baud rate is this frequency divided by 1, 16, or 64, according to the divide ratio bits (bits 0 and 1) in the ACIA control register.

2. For the split baud rate mode, J3 must be opened and J4 closed. Timer 1 is used to generate a square wave on PB7, the frequency of which determines the Full-duplex port transmitter baud rate. (Use the values given in Table II for these transmitter baud rates). With the Shift Register in the free-running output mode, the receiver clock frequency is determined by Timer 2.

TABLE IVa SIMPLEX SERIAL PORT BAUD RATE SELECTION - 50HZ

BAUD RATE	VALUE FOR VIA TIMER 2 LATCH VALUE	(HEX)	MODE	ERROR-%
9600	79	4F	1 bit/bit	-.33
4800	159	8F	1 bit/bit	.28
9600	38	26	2 bits/bit	.91
4800	79	4E	2 bits/bit	-.33
2400	159	8F	2 bits/bit	.28
19200	8	8	4 bits/bit	.91
9600	18	12	4 bits/bit	.91
7200	25	19	4 bits/bit	-.33
4800	38	26	4 bits/bit	.91
3600	52	34	4 bits/bit	-.33
2400	79	4E	4 bits/bit	-.33
1800	106	6A	4 bits/bit	-.33
1200	159	8F	4 bits/bit	.28

TABLE IVb SIMPLEX SERIAL PORT BAUD RATE SELECTION - 60HZ

BAUD RATE	VALUE FOR VIA TIMER 2 LATCH VALUE	(HEX)	MODE	ERROR-%
9600	76	4C	1 bit/bit	-.42
4800	153	99	1 bit/bit	.23
9600	37	25	2 bits/bit	-.42
4800	76	4C	2 bits/bit	-.42
2400	153	99	2 bits/bit	.23
19200	8	09	4 bits/bit	-2.9
9600	17	11	4 bits/bit	2.2
7200	24	18	4 bits/bit	-.42
4800	37	25	4 bits/bit	-.42
3600	50	32	4 bits/bit	-.42
2400	76	4C	4 bits/bit	-.42
1800	102	66	4 bits/bit	-.42
1200	153	99	4 bits/bit	.23

Notes for Table IV

1. "Mode" indicates the number of bits in the VIA Shift Register which are used to generate one "bit" of output. The expansion of bits in this manner must be handled in software. The Simplex Serial Port drivers in the QUARK operating systems use the "4 bits/bit" mode.
2. The actual shift frequency is given by

$$f_E/(2N+4),$$

where f_E is the frequency of the E-clock and N is the value in the Timer 2 latch.

3. Connecting jumpers J3 and J4 will connect the PB7 I/O line from the VIA to the CB1 control line of the VIA. This allows Timer 1, normally used to generate the transmit and receive clocks for the full-duplex serial port, to generate the simplex serial port clock as well. However, split baud rates on the full-duplex channel are not possible when Timer 1 is used in this way.

TABLE V SYNCHRONOUS ADDRESS MULTIPLEXER ADDRESS ASSIGNMENTS

ADDRESS (HEX)	CONTROL REGISTER BIT	SET/CLEAR	REMARKS
FF80	V0	CLR	Normally cleared
FF81	V0	SET	
FF82	V1	CLR	Normally cleared
FF83	V1	SET	
FF84	V2	CLR	Cleared for Alphanumeric Mode
FF85	V2	SET	Set for Graphics Mode
FF86	F0	CLR	Start address bit 10
FF87	F0	SET	
FF88	F1	CLR	Start address bit 11
FF89	F1	SET	
FF8A	F2	CLR	Start address bit 12
FF8B	F2	SET	
FF8C	F3	CLR	Start address bit 13
FF8D	F3	SET	
FF8E	F4	CLR	Normally set
FF8F	F4	SET	
FF90	F5	CLR	Start address bit 14
FF91	F5	SET	
FF92	F6	CLR	Start address bit 15
FF93	F6	SET	
FF94	P1	CLR	Page bit
FF95	P1	SET	
FF96	R0	CLR	Normally cleared
FF97	R0	SET	
FF98	R1	CLR	Normally cleared
FF99	R1	SET	
FF9A	M0	CLR	Normally cleared
FF9B	M0	SET	
FF9C	M1	CLR	Normally set
FF9D	M1	SET	
FF9E	TY	CLR	Map type-see sec T/1.1
FF9F	TY	SET	(Normally set)

Notes:

1. To set or clear any of the bits in SAM registers, load the address in the above table corresponding to the bit to be set or cleared into the HL register, and then execute a CALL to the subroutine at location 0Bhex. (Note that this routine alters the contents of the C register.)

TABLE VI SUGGESTED VALUES FOR THE SAM CONTROL REGISTER

f_{VERT}	MODE	F6	F5	F4	F3	F2	F1	F0	VIDEO MEMORY ADDRESS RANGE
60Hz	ALPHA	0	0	1	1	1	0	0	3000-3FFF
60Hz	ALPHA	0	1	1	1	1	0	0	7000-7FFF
60Hz	ALPHA	1	0	1	1	1	0	0	B000-BFFF
60Hz	ALPHA	1	1	1	1	1	0	0	F000-FFFF
60Hz	GRAPHICS	0	0	1	1	0	0	0	2000-7FFF
60Hz	GRAPHICS	0	1	1	1	0	0	0	6000-BFFF
60Hz	GRAPHICS	1	0	1	1	0	0	0	A000-FFFF
50Hz	ALPHA	0	0	1	1	0	1	1	2C00-3FFF
50Hz	ALPHA	0	1	1	1	0	1	1	6C00-7FFF
50Hz	ALPHA	1	0	1	1	0	1	1	AC00-BFFF
50Hz	ALPHA	1	1	1	1	0	1	1	EC00-FFFF
50Hz	GRAPHICS	0	0	1	0	0	1	0	0800-7FFF
50Hz	GRAPHICS	0	1	1	0	0	1	0	4800-BFFF
50Hz	GRAPHICS	1	0	1	0	0	1	0	8800-FFFF

Notes:

1. The starting address is calculated from the polynomial

   $SA=(F6)*2^{15}+(F5)*2^{14}+(F3)*2^{13}+(F2)*2^{12}+(F1)*2^{11}+(F0)*2^{10}$.

   Thus the starting address is the binary number

   (F6)(F5)(F3)(F2) (F1)(F0)00 0000 0000.
2. The final address is the first 16k boundary following the starting address in Alphanumeric mode, or the second 16k boundary following the starting address in Graphics mode.
3. Bit F4 in the SAM Control Register must always be one.
4. On Quarks with 128k memory, the Video Memory is located in the memory bank determined by bit 0 of the I register.
5. In Graphics mode, the starting address must be on a 3k boundary.

TABLE VII QUARK PIN CONNECTIONS AND FUNCTIONS

PIN	GROUP	DESCRIPTION
A-1	CRT	Ground
A-2	CRT	Vertical sync output (RED output on QUARK/150)
A-3	FULL-DUP	RS-232C transmit data from ACIA (full-duplex port)
A-4	SIMPLEX	RS-232C transmit data from VIA
A-5	FULL-DUP	Ground
A-6	P3	VIA PB1 parallel I/O line
A-7	P3	VIA PB3 parallel I/O line
A-8	P3	VIA PB5 parallel I/O line
A-9	PAR PTR	Parallel printer output bit 7
A-10	PAR PTR	Parallel printer output bit 5
A-11	PAR PTR	Parallel printer output bit 2
A-12	PAR PTR	Parallel printer output bit 0
A-13	P/S PTR	Ground for parallel/serial printer
A-14	EXP BUS	User chip select output (active low)
A-15	EXP BUS	E-CLK output
A-16	DISK	Ground

TABLE VII QUARK PIN CONNECTIONS AND FUNCTIONS (CONTINUED)

PIN	GROUP	DESCRIPTION
A-17	DISK	Step output to floppy disk drive (active low)
A-18	DISK	Write gate output to floppy disk drive (active low)
A-19	DISK	Floppy disk drive side select (active low)
A-20	DISK	Ground
A-21	P1	PIA PA2 parallel I/O line (KBD2)
A-22	P1	PIA PA4 parallel I/O line (KBD4)
A-23	P1	PIA PA6 parallel I/O line (KBD6)
A-24	EXP BUS	Ground
A-25	EXP BUS	Z80B data bus D1
A-26	EXP BUS	Z80B data bus D7
A-27	EXP BUS	Z80B data bus D6
A-28	EXP BUS	Z80B data bus D3
A-29	FLOPPY	Floppy disk index signal input (active low)
A-30	RESERVED	Reserved for future hard disk version
A-31	POWER	Ground return line
A-32	POWER	Ground return line
B-1	CRT	TTL video signal out (GREEN output on QUARK/150)
B-2	SIMPLEX	Printer busy input
B-3	FULL-DUP	RS-232C RTS output from ACIA
B-4	FULL-DUP	RS-232C DSR input to ACIA
B-5	P2	VIA PA0 parallel I/O line
B-6	P2	VIA PA1 parallel I/O line
B-7	P2	VIA PA2 parallel I/O line
B-8	P2	VIA PA3 parallel I/O line
B-9	P2	VIA PA4 parallel I/O line
B-10	P2	VIA PA5 parallel I/O line
B-11	P2	VIA PA6 parallel I/O line
B-12	P2	VIA PA7 parallel I/O line
B-13	PAR PTR	Parallel printer acknowledge input to VIA
B-14	EXP BUS	Z80B address bus A2
B-15	EXP BUS	Z80B address bus A4
B-16	DISK	Ground
B-17	DISK	Use for Disk Size Mode (user modification)
B-18	DISK	Floppy disk drive select #3 output (active low)
B-19	DISK	Floppy disk drive select #0 output (active low)
B-20	DISK	Ground
B-21	P1	PIA PA0 parallel I/O line (KBD0)
B-22	P1	PIA PA7 parallel I/O line (KBD7)
B-23	P1	PIA CA2 control line (KB STROBE)
B-24	EXP BUS	Z80B address bus A1
B-25	EXP BUS	Power-on-reset output (active low)
B-26	EXP BUS	Z80B WR control line (active low)
B-27	EXP BUS	Z80B RD control line (active low)
B-28	EXP BUS	Z80B INT input line (active low)
B-29	FLOPPY	Track 00 sense input (active low)
B-30	RESERVED	Reserved for future hard disk version
B-31	POWER	+5V regulated input
B-32	POWER	+5V regulated input

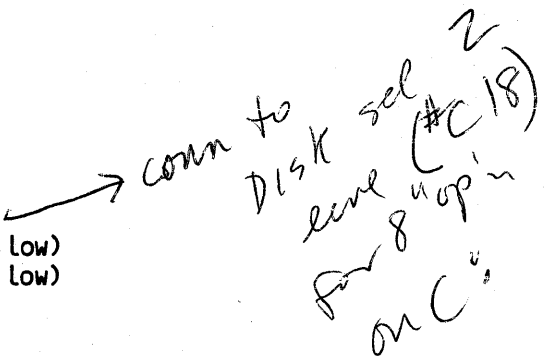
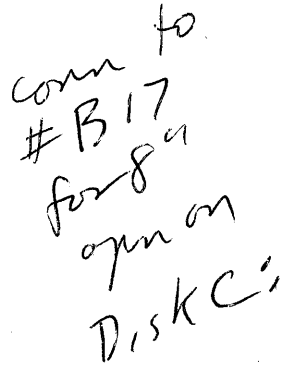


TABLE VII QUARK PIN CONNECTIONS AND FUNCTIONS (CONTINUED)

PIN	GROUP	DESCRIPTION
C-1	CRT	Horizontal sync output (BLUE output on QUARK/150)
C-2	FULL-DUP	RS-232C RxD serial data input to ACIA
C-3	FULL-DUP	RS-232C DTR output (driven by PIA PB7)
C-4	FULL-DUP	RS-232C CTS input to ACIA
C-5	CRT	Composite video output (COMPOSITE SYNC on QUARK/150)
C-6	P3	VIA PB2 parallel I/O line
C-7	P3	VIA PB0 parallel I/O line
C-8	P3	VIA PB4 parallel I/O line
C-9	PAR PTR	Parallel printer interface output bit 4
C-10	PAR PTR	Parallel printer interface output bit 3
C-11	PAR PTR	Parallel printer interface output bit 6
C-12	PAR PTR	Parallel printer interface outputbit 1
C-13	PAR PTR	Parallel printer interface data strobe output
C-14	EXP BUS	Z80B address bus A3
C-15	EXP BUS	Z80B address bus A5
C-16	DISK	Read data input from floppy disk (active low)
C-17	DISK	Direction control output to floppy (active low)
C-18	DISK	SEL 2 floppy disk drive select (active low)
C-19	DISK	SEL 1 floppy disk drive select (active low)
C-20	DISK	Write data output to floppy disk drive (active low)
C-21	P1	PIA PA1 parallel I/O line (KBD1)
C-22	P1	PIA PA3 parallel I/O line (KBD3)
C-23	P1	PIA PA5 parallel I/O line (KBD5)
C-24	EXP BUS	Z80B address bus A0
C-25	EXP BUS	Z80B data bus D0
C-26	EXP BUS	Z80B data bus D2
C-27	EXP BUS	Z80B data bus D4
C-28	EXP BUS	Z80B data bus D5
C-29	DISK	Floppy disk write protect sense input (active low)
C-30	RESERVED	Reserved for future hard disk version
C-31	RESET	Reset input (active low)
C-32	POWER	+12 V regulated input



1. P1, P2, P3 refer to general-purpose parallel ports 1, 2, and 3. In standard Quark operating systems port P1 is configured as a standard encoded-keyboard data input, with CA2 of the PIA as the keyboard data strobe input.
2. EXP BUS refers to the set of connections to the Z80 address, data, and control lines, and to the E-clock and external chip-select lines. These lines form the Peripheral Expansion Bus.
3. DISK refers to connections for the floppy disk drives.
4. All inputs and outputs for the Parallel Printer (PAR PTR) interface are TTL-compatible.
5. The Full-duplex Serial Port (FULL-DUP) is implemented using the ACIA and PB7 of the PIA.
6. The Simplex Serial Port (SIMPLEX) is implemented using the VIA.
7. For proper operation, the composite video output should be terminated by a bridging 75 ohm load at the monitor.
8. Bit 0 is the least-significant, bit 7 the most-significant.

TABLE VIII QUARK PERIPHERAL CONNECTIONS

TABLE VIIIa FULL-DUPLEX PORT CONNECTIONS

DB-25S PIN (MODEM)	DB-25S PIN (TERMINAL)	FUNCTION	QUARK PIN	REMARKS
1	1	PROT GND	——	(Opt) Chassis ground
2	3	Tx DATA	A-3	Twist with ground wire
3	2	Rx DATA	C-2	Twist with ground wire
4	5	RTS	B-3	
5	4	CTS	C-4	
6	20	DSR	B-4	
7	7	GROUND	A-5	Ground
20	6	DTR	C-3	

Notes:
1. The pin numbers in the column labelled "MODEM" represent the pinouts for connecting to industry-standard modems using a DB-25S connector.
2. The pin numbers in the column labelled "TERMINAL" represent the pinouts for connecting to a standard computer terminal with an RS-232C serial interface using a DB-25S connector. Use these pinouts for connecting an external terminal to be used as the Console device.
3. If a serial printer is to be connected on the Full-Duplex Serial Port, the DB-25S pinout given in the column labelled "TERMINAL" should be used.

TABLE VIIIb PARALLEL PRINTER INTERFACE CONNECTIONS

PRINTER PIN	FUNCTION	QUARK PIN	REMARKS
1	DATA STROBE	C-13	Data strobe, Acknowledge,
2	BIT 0 (LSB)	A-12	and Bits 0-7 may each be
3	BIT 1	C-12	twisted with ground wires
4	BIT 2	A-11	
5	BIT 3	C-10	
6	BIT 4	C-9	
7	BIT 5	A-10	
8	BIT 6	C-11	
9	BIT 7 (MSB)	A-9	
10	ACKNOWLEDGE	B-13	
19-29	GROUND	A-13	

Notes:
1. Printer pin numbers shown for the Parallel Printer connections are the pin numbers for a Centronics 739 printer connector.

TABLE VIIIc 5-1/4-INCH FLOPPY DISK DRIVE CONNECTIONS

EDGE CONNECTOR	FUNCTION	QUARK PIN	REMARKS
1	GROUND	——	
2	OPT	——	
3	GROUND	——	
4	BUSY	——	
5	GROUND	——	
6	SEL 3	B-18	
7	GROUND	A-20	Read Data, Write Data, Index,
8	INDEX	A-29	and Step may each be twisted
9	GROUND	——	with ground wires.

10	SEL 0	B-19	All disk drive lines are
11	GROUND	----	active low.
12	SEL 1	C-19	
13	GROUND	----	
14	SEL 2	C-18	
15	GROUND	----	
16	MOTOR ON	----	
17	GROUND	----	
18	DIRECTION	C-17	
19	GROUND	A-16	
20	STEP	A-17	
21	GROUND	B-20	
22	WRITE DATA	C-20	
23	GROUND	----	
24	WRITE GATE	A-18	
25	GROUND	----	
26	TRACK 00	B-29	
27	GROUND	----	
28	WRITE PROTECT	C-29	
29	GROUND	B-16	
30	READ DATA	C-16	
31	GROUND	----	
32	SIDE	A-19	
33	GROUND	----	
34	SPARE	----	
	RESERVED	B-17	Use for Disk Size Mode
	RESERVED	A-30	Reserved for hard disk version
	RESERVED	B-30	" " " " "
	RESERVED	C-30	" " " " "

Notes:
1. Pin numbers in the above table refer to standard edge connector pin numbers for a 5.25 inch floppy disk drive.
2. If the Motor control line must be used, it could be controlled by on of the QUARK parallel I/O lines (if suitably buffered), or by one of the Drive Select lines. Either of these configurations would require a patch to the BIOS.

TABLE VIIId 8-INCH FLOPPY DISK DRIVE INTERFACE

EDGE CONNECTOR	FUNCTION	QUARK PIN	REMARKS
1	GROUND	----	
2	LOW CURRENT	B-18	
3	GROUND	----	
4,6,8,10	N/C	----	
5,7,9,11	GROUND	----	
12	DISK CHANGE	----	
13	GROUND	----	
14	SIDE	A-19	
15	GROUND	----	
16	IN USE	----	
17	GROUND	----	
18	HEAD LOAD	----	
19	GROUND	----	
20	INDEX	A-29	Read Data, Write Data, Index,
21	GROUND	A-20	and Step may each be twisted
22	READY	----	with ground wires.
23	GROUND	----	All disk drive lines are
24	SECTOR	----	active low.
25	GROUND	----	
26	SEL 0	B-19	

27	GROUND	----
28	SEL 1	C-19
29	GROUND	----
30	SEL 2	C-18
31	GROUND	----
32	SEL 3	B-18
33	GROUND	----
34	DIRECTION	C-17
35	GROUND	----
36	STEP	A-17
37	GROUND	A-16
38	WRITE DATA	C-20
39	GROUND	B-20
40	WRITE GATE	A-18
41	GROUND	----
42	TRACK 00	B-29
43	GROUND	----
44	WRITE PROTECT	C-29
45	GROUND	----
46	READ DATA	C-16
47	GROUND	B-16
48	SEP DATA	----
49	GROUND	----
50	SEP CLK	----

Notes:
1. Pin numbers in the above table refer to standard edge connector pin numbers for an 8-inch floppy disk drive.

TABLE VIIIe ASCII-ENCODED PARALLEL-OUTPUT KEYBOARD CONNECTIONS

FUNCTION	QUARK PIN	REMARKS
KEYBOARD D0	B-21	D0 is the least significant bit
KEYBOARD D1	C-21	of the ASCII code
KEYBOARD D2	A-21	
KEYBOARD D3	C-22	
KEYBOARD D4	A-22	
KEYBOARD D5	C-23	
KEYBOARD D6	A-23	
KEYBOARD D7	B-22	D7 is ignored by the BIOS
KEYBOARD STROBE	B-23	STROBE is initialized active low

Notes:
1. The keyboard input uses Port A of the PIA and CA2.
2. All of the keyboard input lines are TTL-compatible.
3. If a keyboard with an active high STROBE output is to be connected to the QUARK, the STROBE input line may have to be inverted until the console input routine in the BIOS is patched to recognize an active high STROBE.

TABLE VIIIf DIRECT-DRIVE DATA DISPLAY MONITOR CONNECTIONS

FUNCTION	QUARK PIN	REMARKS
Video input on monitor	B-1	The video output from the QUARK
Horizontal sync input	C-1	can be twisted with a ground wire.
Vertical sync input	A-2	
Signal ground on monitor	A-1	

TABLE VIIIg COMPOSITE VIDEO DATA DISPLAY MONITOR CONNECTIONS

FUNCTION	QUARK PIN	REMARKS
Video input on monitor	C-5	The video output from the QUARK
Signal ground on monitor	A-1	can be twisted with a ground wire.

TABLE VIIIh ANALOG RGB COLOUR DISPLAY MONITOR CONNECTIONS

FUNCTION	QUARK PIN	REMARKS
GREEN video output	B-1	Each video output from the
BLUE video output	C-1	QUARK/150 can be twisted with
RED video output	A-2	a ground wire.
COMPOSITE SYNC output	C-5	
Signal ground on monitor	A-1	

TABLE IX JUMPER OPTIONS

JUMPER ID	FUNCTION	USE
J1	MM PAL RAS bypass	Connect for 64k-only
J2	8" / 5.25" floppy	Connect for 8" floppy
J3	Non-split baud rate	Connect for non-split
J4	Split baud rate	Connect for split
J5	A-4 GP serial out	Connect for VIA CB1 to A-4
J6	A-4 GP serial out	Connect for VIA CB2 to A-4
J7	B-2 GP serial in	Connect for B-2 to VIA CB2
J8	B-2 GP serial in	Connect for B-2 to VIA PB6
J9	68A50 CS2	Connect for 68A50
J10	68A52 Reset	Connect for 68A52
J11	68A50 CS1	Connect for 68A50
J12	68A52 CS	Connect for 68A52
J13	68A50 CS0	Connect for 68A50
J14	POSITIVE SYNC (Q/150)	Connect for positive sync
J15	NEGATIVE SYNC (Q/150)	Connect for negative sync

Notes:
1. The standard installed jumper configuration is:
   J2(if 8-inch diskettes are ordered), J3, J6, J8, J9, J11, J13.
2. Connect both J3 and J4 for Timer 1-generated baud rates on the simplex serial port as well as on transmitter and receiver of full-duplex serial port.
3. Pin A-4 is driven by an RS-232C driver output. Pin B-2 drives an RS-232C receiver input.
4. The 68A52 SSDA is available by special order only. Contact the factory for further information.
5. J1 is installed at the factory on 64k memory versions.
6. J14 and J15 apply on to the QUARK/150, where they determine the polarity of the composite sync output to the RGB monitor. Only one of J14 and J15 should be connected at any time.

TABLE X TERMINAL CONTROL CODES

This table gives the hexidecimal values to be sent to the Quark's Terminal Emulator (or video driver) to perform the various functions it supports. The "Megatel" codes are the codes recognized by the terminal emulator used with any of the distributed operating systems. These codes will be the control codes used in all installed operating systems when the are first booted. QTCONFIG (see below) can be used to change the control codes used.

The "Televideo* 920" codes are an alternative set of control codes which can be used instead of the standard Megatel codes. These codes are a subset of the terminal control codes used by a Televideo 920 terminal. Only some of the functions performed by that terminal are supported by the Quark operating system, so codes other than those given below will be ignored. When running programs which have been designed to run on this terminal, it may be more convenient to load the Televideo 920 code set rather than to modify the program to use the Megatel codes.

The utility program QTCONFIG.COM can be used to load either sets of codes after the system is booted. This utility also allows a set of user-defined codes to be created. This set can be saved as a disk file which can be subsequently loaded through QTCONFIG.COM.

* "Televideo" is a registered trademark of Televideo Systems, Inc.

TABLE X TERMINAL CONTROL CODES

	"MEGATEL" CODE		"TELEVIDEO 920" CODE	
FUNCTION	LEAD-IN (HEX)	CONTROL (HEX)	LEAD-IN (HEX)	CONTROL (HEX)
Bell	none	07	none	07
Cursor Down (line feed)	none	0A	none	0A
Cursor Up (vertical tab)	none	0B	none	0B
Cursor Left (backspace)	none	08	none	08
Cursor Right	none	0C	none	0C
Carriage Return	none	0D	none	0D
Clear Screen (same as Clear Screen & Home)	n/a	n/a	1B	2B
Clear Screen and Home	none	1A	n/a	n/a
Cursor Home	none	1E	none	1E
Reverse Video Off	1B	19	1B	6B
Reverse Video On	1B	1F	1B	6A
Half-intensity Off (same as Reverse Off)	n/a	n/a	1B	28
Half-intensity On (same as Reverse On)	n/a	n/a	1B	29

Clear to End of Line (includes character at cursor position)	1B	15	1B	54
Clear to End of Page (includes characters at or right of cursor position)	1B	16	1B	59
Cursor Addressing (see note 1 for order of co-ordinates)	1B	3D	1B	3D
Cursor On	1B	11	1B	11
Cursor Off	1B	12	1B	12
Insert Line	1B	13	1B	45
Delete Line	1B	14	1B	52
Use STATUS screen	1B	17	1B	17
Use MAIN screen	1B	18	1B	18
Use DISPLAY screen	1B	1D	1B	1D
No-Check Mode Off (interprets control codes)	1B	09	1B	09
No-Check Mode On (displays control codes)	1B	10	1B	10
Load Character Pattern (for alpha mode, and for graphics mode,) followed by: -ASCII character code (0-FF) -eight pattern bytes First pattern byte is top line of character pattern.	1B	01	1B	01

Print Screen - 1C hex - This is not a terminal driver function but a special trap in the console input routine.

Notes:

1. If the Megatel control codes are being used, the cursor addressing character sequence is

   1B 3D xx yy

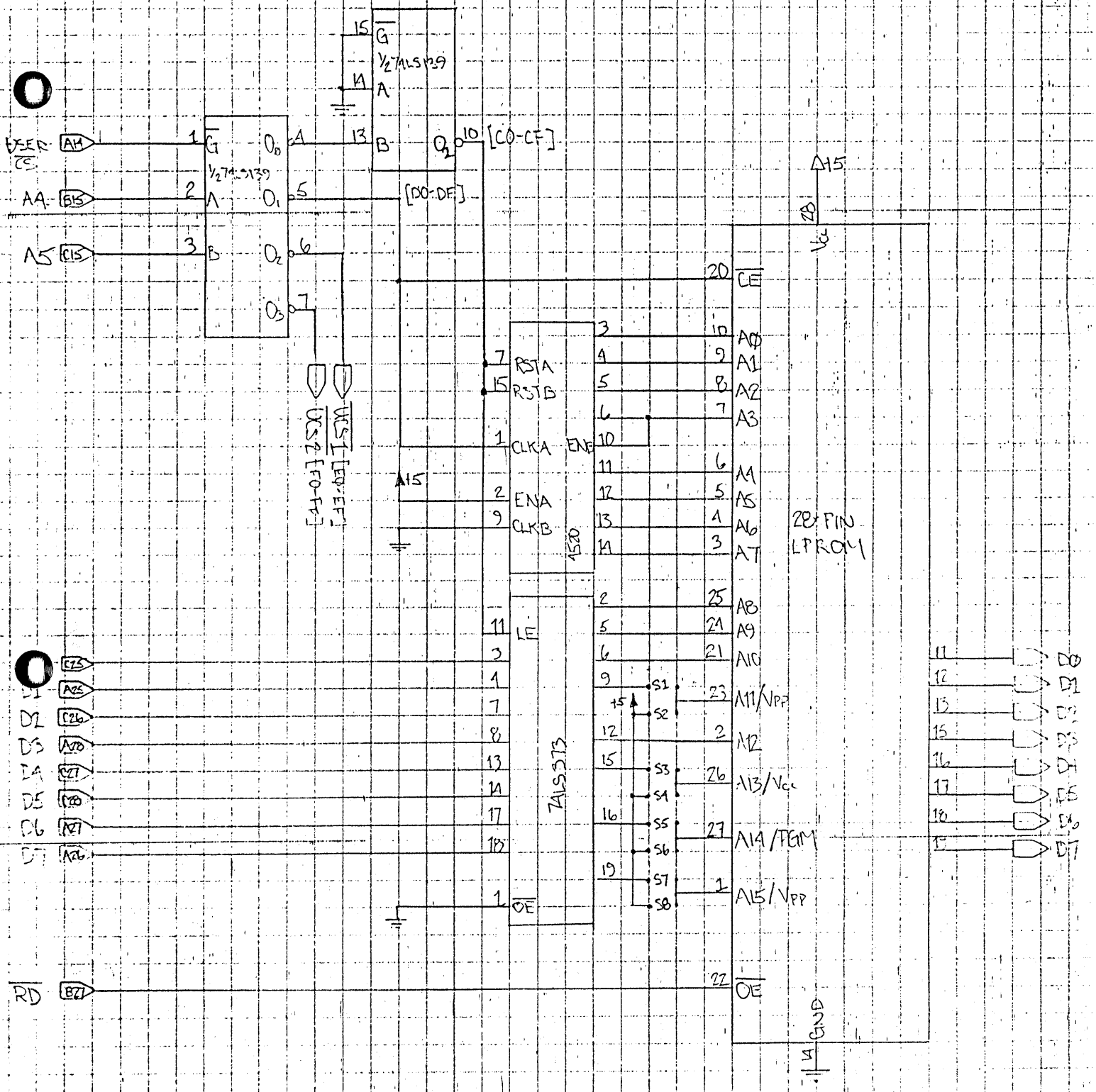
   where xx and yy are the one-byte values for the x-(horizontal) and y-(vertical) co-ordinates. If the Televideo 920 control codes are in use, the the cursor addressing sequence is

   1B 3D yy xx

   In the CP/M 2.2 terminal drivers, there are 27 lines (0-26) and 80 columns (0-79) on the 60Hz video driver, and 35 (0-34) lines and 80 columns in the 50Hz video driver. In CP/M 3.0 terminal drivers, the vertical size of the screens are defined during system installation. The top left-hand corner of the in-use screen area (DISPLAY, MAIN, or STATUS) is address 0,0 (ie, there is no offset in the addresses). Cursor addresses outside the area of the screen will leave the cursor at the boundary of the screen.

For instance, if the x-co-ordinate is outside the range 0 to 80, the cursor will be left at the left edge of the screen at the line specified by the y-co-ordinate. Bit 7 of the address bytes is ignored, offsets of 80(HEX) do not change the address value.

2. Some language processors will not properly address column 9 or line 9 when using the Cursor Address facility, because they automatically expand a "09" code into a TAB, and insert several spaces. The suggested solution is to add an offset of 80(HEX) to both the x- and y-co-ordinates, since the terminal emulator ignores the top bit of the address.
3. Cursor Up will have no effect if the cursor is positioned on line 0.
4. Cursor Down will perform a scroll (ie. entire screen up one line) if the cursor is positioned on the bottom line of the in-use screen.
5. Cursor Right has no effect if the cursor is positioned in column 80.
6. Cursor Left will have no effect if the cursor is positioned in column 0 of any line.
7. In the CP/M 2.2 terminal drivers, there are 27 lines (0-26) and 80 columns (0-79) on the 60Hz video driver, and 35 (0-34) lines and 80 columns in the 50Hz video driver.
8. The Televideo 920 "Half-intensity" control codes perform the same effects as the "Reverse Video" control codes. The Televideo 920 "Clear Screen" code performs a "Clear Screen and Home".
9. Programs using these control codes should be made flexible as additional control codes may be implemented in the future.



ADDRESS	FUNCTION
C0-CF	WRITE HIGH ORDER EPROM ADDRESS LATCH, CLEAR LOW ORDER COUNTER
D0-DF	READ EPROM, THEN INCREMENT LOW ORDER ADDRESS COUNTER
E0-EF	USER CHIP SELECT 1 - USED FOR COMPANION SERIAL CARD
F0-FF	USER CHIP SELECT 2 - USE FOR OTHER EXTERNAL I/O

EPROM EXPANSION CIRCUIT

DBS 07 26 G.SKELLS

```
        TITLE   'QUARK BOOTSTRAP PROM FOR EPROM BOOT..VER 1.0  APRIL 3/85'
;
;QUARK BOOTSTRAP PROM FOR EPROM BOOT                                    1.
;
;       -COPYRIGHT (C) 1985
;       MEGATEL COMPUTER TECHNOLOGIES
;       A DIVISION OF F. & K. MANUFACTURING CO. LIMITED
;       150 TURBINE DRIVE
;       WESTON, ONTARIO
;       M9L 2S2
;
;               VERSION 1.0
;
;-APRIL 3/85    -TAKEN FROM QUARK BOOTSTRAP PROM VER 1.5
;               -WILL READ THE FIRST 256 BYTES OF EPROM TO LOCATION
;                80 HEX AND THEN PASS CONTROL TO CODE AT 80 HEX. B.MISKETIS
;
;
FALSE           EQU     0
TRUE            EQU     NOT FALSE
;
F50HZ           EQU     true    ;PROM SETUP FOR 50 HERTZ OPERATION
;
;
TIMEOUT         EQU     25000   ;250MS TIMEOUT ON ALL OTHER DISK OPERATIONS
;                               ; THIS VALUE MUST BE LESS THAN 8000H
        IF      F50HZ
SAMBITS         EQU     1BE2H
VDMBASE         EQU     6C00H           ;START OF SCREEN MEMORY. TY=0,P1=1 MAP
NUMLINES        EQU     40              ;NUMBER OF LINES TO CLEAR
BOTSCREEN       EQU     0EC00H          ;START OF SCREEN MEMORY
VERSION         EQU     01550H          ;VERSION 1.5, 50HZ
        ELSE
SAMBITS         EQU     07E2H
VDMBASE         EQU     7000H           ;START OF SCREEN MEMORY. MAPPED
NUMLINES        EQU     32              ;NUMBER OF LINES TO CLEAR
BOTSCREEN       EQU     0F000H
VERSION         EQU     01560H          ;VERSION 1.5, 60HZ
        ENDIF
HIDCODE         EQU     0F000H          ;PUT CODE IN VIDEO HOLES
;
LOADADD         EQU     80H             ;EXECUTION ADDRESS OF BOOTSTRAP SECTOR
LATCHADD        EQU     0C0H            ;I/O ADDRESS OF LATCH USED FOR
                                        ;READING EPROM
EPROMADD        EQU     0D0H            ;I/O ADDRESS OF FIRST EPROM
CODECNT         EQU     255             ;NUMBER OF BYTES TO READ FROM EPROM
                                        ; MAXIMUM 255
PIAPORT         EQU     74H
DSKPORT         EQU     80H
PIACRAPORT      EQU     75H             ;PERIPHERAL INTERFACE ADAPTER CONTROL REGIS
PIADRAPORT      EQU     74H             ;PIA DATA REGISTER A, OR PIA DATA DIRECTION
SELVAL          EQU     7FH             ; PIAPB -NO DRIVE,SIDE 0,SNGL DENS,ALFA
;-----------------------------------------------------------------------
;
        MACLIB  Z80
;
SPACE MACRO ?XA,?XB
        LOCAL   ?XC,?XD
        ?XC     SET     LOW(?XA)
        IF NUL ?XB
                ?XD     SET     40H
        ELSE
                ?XD     SET     ?XB
        ENDIF
        IF ?XC LT ?XD
```

```
                REPT ?XD-?XC
                        DB      0
                ENDM
        ENDIF
        ENDM
;
        ORG     0
;
        DI                      ;INIT INTERRUPTS
        IM1
        JMP     START
        DW      VERSION
;
        ENDSPACE $,0BH
;
        ORG     0BH             ;NOTE: THIS CODE IS OVERLAID IN MEMORY/ROM
;
        MOV     C,M             ;SECTION FOR ROUTINE TO TALK TO SAM REGISTERS
        IN      PIAPORT+3       ;GO TO BOOT MODE TO GET TO SAM REGISTERS
        JR      SKIPV           ;SHOULD COME OUT TO JR 13H
        DB      0,0,0           ;NOTE: ROOM FOR RST VECTOR
SKIPV:
        ORI     8
RAMLOOP:
        OUT     PIAPORT+3
RAMRET:
        MOV     M,C             ;OVERLAID IN RAM AS A RETURN
UNBOOT:
        IN      PIAPORT+3       ;UNBOOT
        ANI     0F7H
        JR      RAMLOOP         ;NOW OUTPUT-SHOULD BE 15H
;
        ENDSPACE $,34H
;
        ORG     34H
;
INTERFIX:                       ;THIS ROUTINE IS TO CATCH INTERRUPTS
        ANI     0F7H            ; WHEN REPROGRAMING THE SAM
        OUT     PIAPORT+3
        INX     SP              ;THIS DOES NOT GET EXECUTED TWICE
        INX     SP
        MOV     M,C
        IN      PIAPORT+3
        JR      INTERFIX
;
CEND0   EQU     $
        ENDSPACE CEND0
;
;I/O INITIALIZATION
;
        ORG     100H
START:
        XRA     A               ;SAM INIT---CF LEO
        STA     0FF98H          ;R1 CLEAR
        STA     0FF96H          ;R0 CLEAR

        OUT     99H             ;DISABLE OMNINET INTERRUPT-ALIASES TO TRACK

        MVI     A,0FFH          ;SET DATA DIRECTION OF PIA B TO OUTPUT
        OUT     PIAPORT+2
        MVI     A,0FEH          ;CRB
        OUT     PIAPORT+3
        MVI     A,0C6H          ;CRA
        OUT     PIAPORT+1
        MVI     A,selval        ;piaB DATA - density, side, sel, alpha
```

```
                JMP     200H
;
                JMP     200H
RESET3:
                DCR     C
                JNZ     RESET4
;
                LXI     H,SAMBITS       ;GET SAM INITIALIZATION
                LXI     D,0FF00H
                LXI     B,1040H
;
SAMINIT:
                MOV     A,C
                DAD     H
                RAL
                MOV     E,A
                STAX    D
                INR     C
                DCR     B
                JNZ     SAMINIT
;
                MVI     A,0
WAIT:
                DCR     A
                JNZ     WAIT
                JMP     SCREEN
;
CEND1   EQU     $
                ENDSPACE CEND1
;
                ORG     200H
                ;THIS SECTION DOES A PROPER HARDWARE RESET TO THE SAM PART
RESET:                  ; IF YOU WANT TO KNOW DETAILS, SEE LEO
                LXI     B,1004H
RESET4:
                LXI     D,1
                LXI     H,0F000H
                IN      PIAPORT
RESET1:
                IN      PIAPORT+1
                ANA     A
                JM      RESET3
                DAD     D
                JNC     RESET1
                LXI     H,0FF80H
RESET2:
                MOV     M,A
                INR     L
                INR     L
                DCR     B
                JNZ     RESET2
                STA     0FF9DH          ;M1 SET
                STA     0FF9BH          ;M0 SET
                LXI     H,0FF8DH        ;F3 SET
                MOV     M,A
                DCR     L
                MOV     M,A
                INR     L
                MOV     M,A
                DCR     L
                MOV     M,A
                INR     L
                MOV     M,A
                DCR     L
```

```
        INR     L
        MOV     M,A
        DCR     L
        MOV     M,A
        STA     0FF9AH
        JMP     RESET

CEND2   EQU     $
        ENDSPACE CEND2
;
        ORG     300H
;
SCREEN:
        MVI     A,00$00000$1B   ;256K SYSTEM SET USER BANK TO 0
        STAI
        XRA     A               ;256K SYSTEM SET COMMON/VIDEO BANK, OR 128K
        STAI                    ; SYSTEM SET COMMON/VIDEO BANK AND USER BANK
;
        LXI     H,VDMBASE       ;CLEAR VDM
CLEAR:
        MVI     M,0
        INX     H
        MOV     A,H
        RLC
        JRNC    CLEAR

        STA     0FF9FH          ;SET TYPE BIT - UNMAP MEMORY CONFIG

        MVI     L,LOW(MOVETABLE)        ;COPY CODE IN PROM TO RAM
MOVENEXT:
        MVI     H,HIGH(MOVETABLE)       ;TABLE CAN NOT CROSS PAGE BOUNDARY
        MOV     C,M
        MOV     A,C
        INR     A
        JRZ     SCREEN1
        INR     L
        MOV     E,M
        INR     L
        MOV     D,M
        INR     L
        MOV     B,M
        INR     L
        MOV     H,M
        MOV     A,L
        MOV     L,B
        MVI     B,0
        LDIR                    ;MEMORY WRITE
        MOV     L,A
        INR     L
        JR      MOVENEXT
SCREEN1:
        MVI     A,0C9H          ;SETUP RETURN FOR CALL 0BH FROM UNBOOT MODE
        STA     RAMRET
;                               ;DO NOT USE STACK BEFORE UNBOOT-CAN NOT POP
        LXI     SP,STACK        ;INIT STACK - MINIMUM OF THREE LEVEL
                                ;RETURN TO CHARLD IS ALREADY ON STACK
        JMP     UNBOOT

CEND3   EQU     $
        ENDSPACE CEND3
;
        ORG     400H

;REG C - BYTE COUNT
;REG DE - DESTINATION
```

```
MOVETABLE:
        DB      0CH                             ;SAM INTERFACE
        DW      0BH,0BH
        DB      (EBOOTEND-EBOOT) AND 0FFH       ;READ IN EPROM
        DW      BOOT,EBOOT
        DB      30H                             ;CHARACTER LOADER, MAXIMUM 48 BYTES
        DW      CHARLD,XCHARLD                  ;DESTINATION IS SECOND HOLE IN VDM
        DB      (XENDFL-EBOOTFL) AND 0FFH       ;BOOT FAIL
;       DW      BOOTFL,EBOOTFL
        DW      0FF00H,EBOOTFL
;       dw      038h,ebootfl
        DB      0FFH                            ;END OF TABLE
;
        ENDSPACE $,24H
;
CEND4   EQU     $
        ENDSPACE CEND4
;
        ORG     500H
                        ;THIS SECTION EXECUTES IN THE VIDEO MEMORY TO LOAD
;                       ; THE 0 AND 80H CHARACTER POSITIONS
;                       ; CODE RUNS AT BOTSCREEN+80H TO BOTSCREEN+80H+48
XCHARLD:
CHARLD          EQU     BOTSCREEN+80H
                                ;RETURN TO BOOTSTRT IS ALREADY ON THE STACK
;
        LXI     B,BOTSCREEN     ;THIS ADDRESS MUST CONTAIN 00H TO START
        CALL    VSLOOP
;
        LXI     H,BOTSCREEN+20H
        MVI     A,16
XCONTMOV:
CONTMOV         EQU     $-XCHARLD+CHARLD
        MVI     M,080H
        INX     H
        DCR     A
        JRNZ    XCONTMOV
        CMA
        STAX    B
;
XVSLOOP:
VSLOOP          EQU     $-XCHARLD+CHARLD
;
;************************************************
; CODE STARTING FROM HERE MUST BE PRECISELY TIMED
;
;CLEAR ANY PENDING IRQA INTERRUPTS
;
        IN      PIAPORT
;
;WAIT FOR VERTICAL SYNC INTERRUPT CYCLE
;
XWAITINTERRUPT:
WAITINTERRUPT   EQU     $-XCHARLD+CHARLD
        IN      PIAPORT+1
        ANA     A                       ;LOOK FOR IRAQA INTERRUPT
        JP      WAITINTERRUPT           ;12 T STATES
;
        MVI     E,8
;WE SHOULD BE AT THE HORIZONTAL RETRACE AT THE START OF THE
;FIRST LINE OF VDM.(0F020H)
;
XHSLOOP:
HSLOOP          EQU     $-XCHARLD+CHARLD
```

```
        INR     A
        DCR     E
        RZ
;
;FINISH COUNT DOWN TO COMPLETE 96 CHAR COUNT
;
                MVI     A,21
CNTDWN:
CNTDWN          EQU     $-XCHARLD+CHARLD
                DCR     A
                JNZ     CNTDWN
                JR      XHSLOOP
;
;MAKE SURE THE SIZE OF THE CHAR LOADER IS NOT MORE THAN 30H
;
CEND5   EQU     $
        ENDSPACE CEND5
;
        ORG     600H            ;THIS SECTION WILL EXECUTE FROM 40H
;
EBOOT:
BOOT            EQU     40H
;
        MVI     A,0
        OUT     LATCHADD        ;INITIALIZE TO ACCESS FIRST PAGE OF
                                ;THE FIRST EPROM
        LXI     H,LOADADD       ;PUT EXECUTION ADDRESS OF BOOTSTRAP CODE IN HL
        MVI     C,EPROMADD      ;PUT I/O ADDRESS OF EPROM INTO REG C
        MVI     B,CODECNT       ;PUT NUMBER OF BYTES TO READ FROM EPROM MAX-256
        INIR
        JMP     LOADADD         ;JUMP TO BOOTSTRAP CODE
;
 BOOTEND        EQU     $
,
CEND6   EQU     $
        ENDSPACE $
;
        ORG     700H
;
EBOOTFL:
BOOTFL          EQU     HIDCODE+0F00H
        MVI     B,24            ;TWENTY FOUR LINES
        LXI     D,130           ;LINE LENGTH-SORT OF
        LXI     H,BOTSCREEN+230H
BOOTLOOP:
        MVI     M,80H           ;REVERSE VIDEO
        DAD     D               ;NEXT LINE
        DJNZ    BOOTLOOP
        HLT
                DW      0               ;MINIMUM OF THREE LEVEL STACK
STACK           EQU     $-EBOOTFL+BOOTFL
                DW      CHARLD,BOOT     ;PREPARE STACK FOR A COUPLE OF RET'S
;
XENDFL          EQU     $
;
;
;
CEND7   EQU     $
        ENDSPACE CEND7

;CHECK TO SEE CODE IS EXTENDED ABOVE 64 BYTE OF EACH PAGE

        IRPC    ?X,01234567
?XX     SET     LOW(CEND&?X)
        IF ?XX GT 40H
                DS 'PAGE &?X IS EXTENDED ABOVE 3FH'.
```