5082 User's Manual

Doc. #03281 Rev. 0296

OCTAGON SYSTEMS CORPORATION®

6510 W. 91st Ave. Westminster, CO 80030
Tech. Support: 303–426–4521

COPYRIGHT

TRADEMARKS

Micro PC™, PC SmartLink™, Octagon Systems Corporation®, the Octagon logo and the Micro PC logo are trademarks of Octagon Systems Corporation. QuickBASIC® is a registered trademark of Microsoft Corporation. ROM-DOS™ is a trademark of Datalight. SmartSocket™ and SmartWatch™ are trademarks of Dallas Semiconductor. IBM PC® is a register trademark of IBM Corporation.

NOTICE TO USER

The information contained in this manual is believed to be correct. However, Octagon assumes no responsibility for any of the circuits described herein, conveys no license under any patent or other right, and makes no representations that the circuits are free from patent infringement. Octagon makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification.

Octagon Systems Corporation general policy does not recommend the use of its products in life support applications where the failure or malfunction of a component may directly threaten life or injury. It is a Condition of Sale that the user of Octagon products in life support applications assumes all the risk of such use and indemnifies Octagon against all damage.

IMPORTANT!

Please read before installing your product.

Octagon's products are designed to be high in performance while consuming very little power. In order to maintain this advantage, CMOS circuitry is used.

CMOS chips have specific needs and some special requirements that the user must be aware of. Read the following to help avoid damage to your card from the use of CMOS chips.

Using CMOS Circuitry in Industrial Control

Industrial computers originally used LSTTL circuits. Because many PC components are used in laptop computers, IC manufacturers are exclusively using CMOS technology. Both TTL and CMOS have failure mechanisms, but they are different. This section describes some of the common failures which are common to all manufacturers of CMOS equipment. However, much of the information has been put in the context of the Micro PC.

Octagon has developed a reliable database of customer-induced, field failures. The average MTBF of Micro PC cards exceeds 11 years, yet there are failures. Most failures have been identified as customer-induced, but there is a small percentage that cannot be identified. As expected, virtually all the failures occur when bringing up the first system. On subsequent systems, the failure rate drops dramatically.

- Approximately 20% of the returned cards are problem-free. These cards, typically, have the wrong jumper settings or the customer has problems with the software. This causes frustration for the customer and incurs a testing charge from Octagon.

- Of the remaining 80% of the cards, 90% of these cards fail due to customer misuse and accident. Customers often cannot pinpoint the cause of the misuse.

- Therefore, 72% of the returned cards are damaged through some type of misuse. Of the remaining 8%, Octagon is unable to determine the cause of the failure and repairs these cards at no charge if they are under warranty.

The most common failures on CPU cards are over voltage of the power supply, static discharge, and damage to the serial and parallel ports. On expansion cards, the most common failures are static discharge, over voltage of inputs, over current of outputs, and misuse of the CMOS circuitry with regards to power supply sequencing. In the case of the video cards, the most common failure is to miswire the card to the flat panel display. Miswiring can damage both the card and an expensive display.

- **Multiple component failures** - The chance of a random component failure is very rare since the average MTBF of an Octagon card is greater than 11 years. In a 7 year study,

Octagon has never found a single case where multiple IC failures were not caused by misuse or accident. It is very probable that multiple component failures indicate that they were user-induced.

- **Testing "dead" cards** - For a card that is "completely nonfunctional", there is a simple test to determine accidental over voltage, reverse voltage or other "forced" current situations. Unplug the card from the bus and remove all cables. Using an ordinary digital ohmmeter on the 2,000 ohm scale, measure the resistance between power and ground. Record this number. Reverse the ohmmeter leads and measure the resistance again. If the ratio of the resistances is 2:1 or greater, fault conditions most likely have occurred. A common cause is miswiring the power supply.

- **Improper power causes catastrophic failure** - If a card has had reverse polarity or high voltage applied, replacing a failed component is not an adequate fix. Other components probably have been partially damaged or a failure mechanism has been induced. Therefore, a failure will probably occur in the future. For such cards, Octagon highly recommends that these cards be replaced.

- **Other over-voltage symptoms** - In over-voltage situations, the programmable logic devices, EPROMs and CPU chips, usually fail in this order. The failed device may be hot to the touch. It is usually the case that only one IC will be overheated at a time.

- **Power sequencing** - The major failure of I/O chips is caused by the external application of input voltage while the Micro PC power is off. If you apply 5V to the input of a TTL chip with the power off, nothing will happen. Applying a 5V input to a CMOS card will cause the current to flow through the input and out the 5V power pin. This current attempts to power up the card. Most inputs are rated at 25 mA maximum. When this is exceeded, the chip may be damaged.

- **Failure on power-up** - Even when there is not enough current to destroy an input described above, the chip may be destroyed when the power to the card is applied. This is due to the fact that the input current biases the IC so that it acts as a forward biased diode on power-up. This type of failure is typical on serial interface chips.

- **Serial and parallel** - Customers sometimes connect the serial and printer devices to the Micro PC while the power is off. This can cause the failure mentioned in the above section, *Failure upon power-up*. Even if they are connected with the Micro PC on, there can be another failure mechanism. Some serial and printer devices do not share the same power (AC) grounding. The leakage can cause the serial or parallel signals to be 20-40V above the Micro PC ground, thus, damaging the ports as they are plugged in. This would not be a problem if the ground pin is connected first, but there is no guarantee of this. Damage to the printer port chip will cause the serial ports to fail as they share the same chip.

- **Hot insertion** - Plugging cards into the card cage with the power on will usually not cause a problem. (**Octagon urges that you do not do this!**) However, the card may be damaged if the right sequence of pins contacts as the card is pushed into the socket. This usually damages bus driver chips and they may become hot when the power is applied. This is one of the most common failures of expansion cards.

- **Using desktop PC power supplies** - Occasionally, a customer will use a regular desktop PC power supply when bringing up a system. Most of these are rated at 5V at 20A or more. Switching supplies usually require a 20% load to operate properly. This means 4A or more. Since a typical Micro PC system takes less than 2A, the supply does not regulate properly. Customers have reported that the output can drift up to 7V and/or with 7-8V voltage spikes. Unless a scope is connected, you may not see these transients.

- **Terminated backplanes** - Some customers try to use Micro PC cards in backplanes that have resistor/capacitor termination networks. CMOS cards cannot be used with termination networks. Generally, the cards will function erratically or the bus drivers may fail due to excessive output currents.

- **Excessive signal lead lengths** - Another source of failure that was identified years ago at Octagon was excessive lead lengths on digital inputs. Long leads act as an antenna to pick up noise. They can also act as unterminated transmission lines. When 5V is switch onto a line, it creates a transient waveform. Octagon has seen submicrosecond pulses of 8V or more. The solution is to place a capacitor, for example 0.1 µF, across the switch contact. This will also eliminate radio frequency and other high frequency pickup.

TABLE OF CONTENTS

PREFACE

This manual provides all the information required to install, configure and operate the 5082 Microcontroller.

By using this manual, you will be able to:

- Interface the 5082 to your IBM–compatible PC and the Micro PC Expansion Cards.
- Set up communications between the 5082 and a terminal or PC.
- Gain an understanding of the operation of 5082 hardware using CAMBASIC IV programming language.

This manual assumes that you are familiar with some type of BASIC programming software. If you are not experienced with BASIC–type software, you may want to refer to some of the excellent books on BASIC which are available through your local bookstore. Refer also to the *CAMBASIC IV Programming Guide* for information and examples of all commands.

NOTE: The 5082 uses a Z80181 processor. Additional information on this component can be obtained from Zilog (technical manual #DC–2519–02, phone: 408–370–8000).

CONVENTIONS USED IN THIS MANUAL

1. Information which appears on your screen (output from your system or commands or data that you key in) is shown in a different type face.

   Example 1:

   ```
   CAMBASIC IV (tm) (c) 1985-92
   Octagon Systems Corporation
   V xx.x - all rights reserved
   Bytes free — 33518
   ```

   Example 2:

   Press the <ESC> key.

2. *Italicized* refers to information that is specific to your particular system or program, for example,

   Enter *filename*

   means enter the name of your file.

3. Warnings always appear in this format:

   WARNING: The warning message appears here.

4. Paired angle brackets are used to indicate a specific key on your keyboard, for example, <ESC> means the escape key; <CTRL> means the control key; <F1> means the F1 function key.

5. All addresses given in hexadecimal.

SYMBOLS AND TERMINOLOGY

Throughout this manual, the following symbols and terminology are used:

W[–]	Denotes a jumper block and the pins to connect.
NOTE	Information under this heading presents helpful tips for using the 5082.
WARNING:	Information under this heading warns you of situations which might cause catastrophic or irreversible damage.
Autorun	Automatic execution of a program on power–up or reset.
Download	To transfer a program or data from a PC to the RAM on the 5082; also refers to loading the program stored in the EEPROM to the RAM.

Free Memory	The amount of memory available for program and data storage.
Immediate Mode	The system is on and ready for you to use CAMBASIC IV.
Industrial Command Extensions	Specialized CAMBASIC IV commands designed for industrial programming applications (i.e., AIN, AOT, BIT).
LCD	Liquid Crystal Display.
Multidrop Network	A method of multiprocessor communication using RS–485.
PC SmartLINK	A serial communications software package designed by Octagon. It provides communications between a PC and other equipment and may be used with any PC software package, including CAMBASIC IV.
Reset	Resetting the 5082 hardware and software by pushing the reset switch. Has the same results as disconnecting power to the system without the potential side effects of a cold reset.
System RAM	Memory used by CAMBASIC IV for program, data and variable storage.
TTL Compatible	0–5V logic levels.
Upload	To transfer a program or data from the 5082 to a PC.
XON/XOFF	A sender/receiver protocol in which data transmission is suspended until the equipment receiving the data is ready to accept the incoming information.

TECHNICAL SUPPORT

If you have a question about the 5082 Microcontroller and can't find the answer in this manual, call our Technical Support. They will be ready to give you the support you need.

When you call please have the following at hand:

- Your *5082 Microcontroller User's Manual.*
- A description of your problem.

The direct line to Technical Support is (303)–426–4521.

Chapter 1 OVERVIEW

INTRODUCTION

The 5082 Microcontroller is a 4.5 in. x 4.9 in. computer that contains all the hardware and software necessary to create a control system. It can be used alone or in conjunction with Octagon Micro PC expansion cards.

The 5082 uses one slot of the Micro PC card cage. All communication between the 5082 and your PC is done through the serial ports. Figure 1–1 shows the basic system configuration. With this configuration, you can:

- Use your PC's editor or word processor to write your program;
- Download the program to the 5082;
- Edit and alter the program on the 5082;
- Save the program back to disk or to on–board EEPROM.

CAMBASIC IV, an operating system and compiler designed specifically for control and data acquisition applications, is included as part of your 5082 Microcontroller. You do not need any other software to run programs on the 5082. You will need communications software, such as PC SmartLINK, to link your PC to the 5082 card.

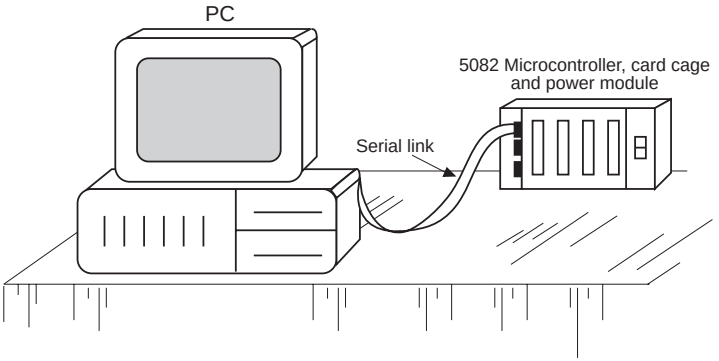


Figure 1–1 — Basic System Configuration

MAJOR FEATURES

Resident CAMBASIC IV Software

The 5082 includes CAMBASIC IV software for program development. This software is designed for developing control and data acquisition programs. Its syntax is very similar to Microsoft BASIC. Industrial command extensions have been added to help you interface with both built–in and external real time hardware.

Digital I/O Port and Opto Rack Interface

The 5082 has 64 I/O lines to interface with logic devices, switch inputs, LEDs, and drive an opto mounting rack. The input and output levels are 0–5V logic compatible. Eight of the digital I/O lines are also supported as an LCD display port, while another eight lines are also supported as a keypad port. The remaining 48 I/O lines can be programmed as inputs or outputs in groups of four and eight.

Matrix Keypad Port

This port directly supports all Octagon 16–position keypads. The KEYPAD$ command in CAMBASIC IV automatically scans and debounces the keypad and interrupts the program when a key is pressed.

Display Port

The display port interfaces to the LCD series displays. The DISPLAY command in CAMBASIC IV supports LCDs up to 80 characters at this port. Eight of the 64 I/O lines are used for this port.

High Current Drivers

The 5082 has 16 high current outputs for driving LEDs, lamps, relays, small motors, etc. Each line can switch DC loads up to 100 mA @ 50V. These lines can also be configured as standard logic levels and can be programmed as inputs or outputs in groups of eight.

COM1, COM2 and COM3 Serial Ports

Three programmable RS-232 serial ports are standard on the 5082. The baud rate, parity, length, and number of stop bits are software programmable for all ports.

Watchdog Timer

The watchdog timer is a fail–safe against program crashes or processor lockups. It times out every 1.2 seconds.

Reset Indicator

W3 is a three pin connector that allows you to connect any kind of indicator (light, LED, buzzer, etc.) that is triggered when the system resets. Pins 1 and 3 are ground. Pin 2 goes low on reset. The indicator must be disconnected to reset the SCR at Q1.

Autorun On Power–up

Autorun refers to the automatic execution of a program on power–up or reset. When the autorun is on, the program stored in the EEPROM (in socket U6) will automatically execute on power–up or reset.

AutoBaud

The AutoBaud feature automatically determines and operates at the baud rate of your PC. The 5082 will match baud rates of 300, 1200, 2400, 4800 and 9600.

On–Card Programmer

Once your program has been debugged, you can store it in the EEPROM by using the on–card programmer and the SAVE command.

Battery–backed RAM

The static RAM on the 5082 may be battery–backed for saving process data during power down. 32K and 128K may be batter–backed with the DS–1213D SmartSocket. The 512K RAMs may be battery–backed with the DS–1213DM. Battery life is typically 10 years at room temperature.

Calendar/clock

A battery–backed calendar may be added to the system by inserting the DS–1216EM SmartWatch into the EEPROM socket and plugging the EEPROM into the DS–1216EM. The dual battery system in this module has a life of about 10 years at room temperature.

Chapter 2 INSTALLATION

The 5082 is designed to be ready to program from the time you plug it in. Installation is simple and straightforward. The 5082 requires one slot in the Micro PC card cage and plugs directly into the backplane (motherboard).

WARNING: Always use antistatic procedures when handling any electronic components. The greatest dangers occur when the card is plugged into a card cage and when cables are plugged into the card. To avoid damaging your card and its components:

1. Ground yourself before handling the 5082 Microcontroller and before inserting or removing cables.
2. Disconnect power before removing or inserting the 5082.
3. Do not insert or remove chips from the board while power is applied.

EQUIPMENT

You will need the following equipment (or equivalent) to use your 5082.

5082 Microcontroller
Micro PC Card Cage
Power Module
VTC–9F Cable
PC SmartLINK

(Please refer to Appendix A if you are making your own serial cable, using a communications program other than PC SmartLINK, or using other non–Octagon components.)

INSTALLING THE 5082 MICROCONTROLLER

Before installing the 5082, refer to Figure 2–1 for the location of various connectors and jumpers.

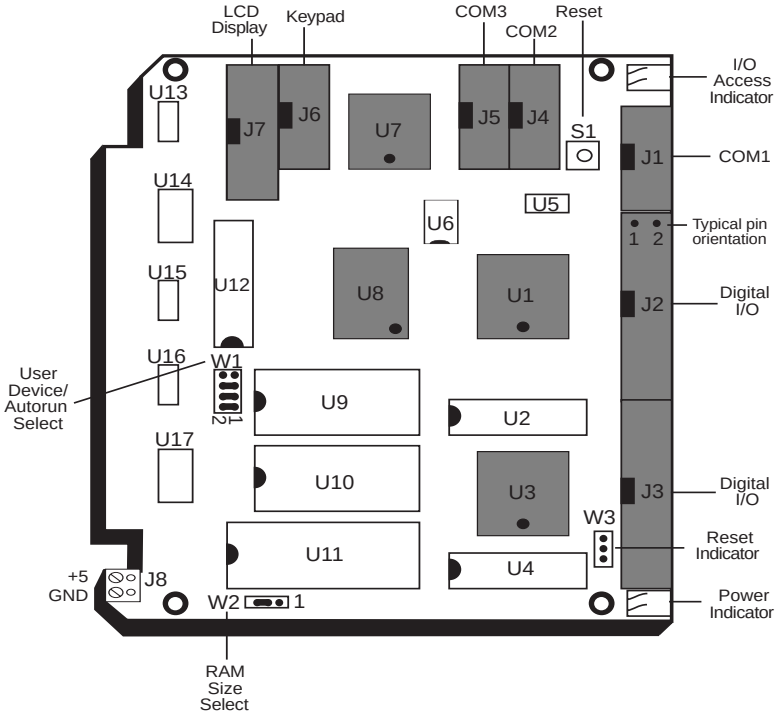


Figure 2–1—5082 Component Diagram

WARNING: Take care to correctly position the 5082 in the card cage. The VCC and ground signals must match those on the backplane. Figure 2–2 shows the relative positions of the 5082 edge connector and the motherboard connector.

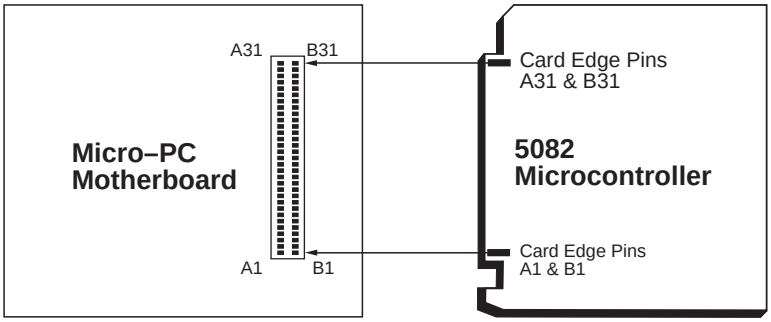


Figure 2–2—Edge Connector Orientation

To Install The 5082 In The Card Cage

1. Turn power module or supply off.

2. Slide the 5082 into the card cage. The components on the card should face to the left or up depending on the type of card cage.

   WARNING: Plugging the card in incorrectly will destroy the card!

3. The serial cable (VTC–9F) for the 5082 has a 10–pin IDC connector on the 5082 end and a connector that is compatible with the PC you are using on the other end. Plug the 10–pin connector on the cable into COM1 (J3) on the 5082. (Please refer to Creating a Custom Communications Cable in Appendix A if you are making your own serial cable.)

4. Plug the other end of the serial cable into COM1 of your PC. If your PC has only one serial port, it is probably COM1. If you have two serial ports, check your computer manual to see which is COM1. (If you are already using COM1, you can use COM2. See the PC SmartLINK Manual for information on using COM2 instead of COM1.)

5. Copy PC SmartLINK onto your hard drive and type

```
SL <RETURN>
```

6. Turn on your power module. If the baud rate of your PC is set to 9600, the following message appears on your screen:

```
Press <ESC>
```

If the baud rate of your PC is not set to 9600, a nonsense message appears instead. Ignore the nonsense message and continue with the next step.

7. Press the <ESC> key.

8. The system will automatically adapt to baud rates of 300, 1200, 2400, 4800, or 9600 and display a logon message showing the current version of CAMBASIC IV and the amount of free memory available:

```
CAMBASIC IV (tm) (c) 1985-92
Octagon Systems Corporation
Vxx.x – All rights reserved
Bytes free – 33518
```

9. If you don't get the proper logon message, refer to the Troubleshooting section at the end of this chapter.

10. The system is now in "immediate mode" and is ready for you to start programming with CAMBASIC IV.

Type the following test program:

```
10 FOR X=0 TO 2
20 PRINT "Hello ";
30 NEXT
40 PRINT
```

Now, type

```
RUN
```

The system will display:

```
Hello Hello Hello
```

UPLOADING & DOWNLOADING PROGRAMS

Downloading programs means transferring them from your PC to RAM on your 5082. Uploading means transferring them from RAM on the 5082 back to the PC. This section explains how to do both of these procedures if you are using PC SmartLINK. Please refer to Appendix A if you are using other communications software.

Uploading Your Program

In the previous section, you created a test program on the 5082. To upload that program and save it to the disk on your PC:

1. Start PC SmartLINK on your PC.

2. Hold down the <ALT> key and type U (for upload).

3. Type in the filename of the program to be uploaded.

Downloading Your Program

To transfer a program from the hard drive on your PC back to the 5082:

1. Start PC SmartLINK on your PC.

2. Hold down the <ALT> key and type D (for download).

3. Type in the filename of the program to be downloaded.

4. To view the program after it downloads, type:

```
LIST<RETURN>
```

WHERE TO GO FROM HERE

If you want to do this:	Turn to:
Add memory	Chapter 3
Battery back memory	Chapter 3
Save data to RAM	Chapter 4
Save programs to EEPROM	Chapter 5
Use autorun feature	Chapter 5
Use serial ports	Chapter 6
Use on–card digital I/O lines	Chapter 7
Connect a display	Chapter 8
Connect a keypad	Chapter 9
Connect high current devices	Chapter 10
Use the watchdog timer	Chapter 11
Use a calendar/clock	Chapter 12
Connect interrupt devices	Chapter 13
Use non–standard communications software or equipment	Appendix A
Use a Micro PC expansion card	The User Manual for that card

TROUBLESHOOTING

No Logon Message

1. Press <ESC> again.

2. Check the power module; make sure the power cord is properly connected between the power module and the power source.

3. Check the serial parameters on your PC. The default setting should be:

   9600 baud, no parity, 8 data bits, 1 stop bit

4. Make sure the serial cable is properly connected between the 5082 and your PC or terminal.

5. Make sure the serial cable is working properly by performing a point-to-point check on the connectors. Refer to Appendix B for corresponding connector, signal, and pin numbers.

6. Make sure the 5082 is receiving power. The power module voltage should be in the 5 +/–0.25V range when measured at the connector pin, B3. The power module ripple should be less than 50 mV.

7. If you are using serial communications software other than PC SmartLINK and your test program does not work, it is probably due to communication problems between the PC and the 5082. To verify that the 5082 is communicating with your PC:

   a. Connect an oscilloscope to J1, pin 3. Press the reset switch on the 5082. A burst of signal activity that switches at least between +3V and –3V verifies that the 5082 is sending a signal.

   b. If you see the logon message but nothing else happens, connect the oscilloscope to U7, pin 14. Press any key on the PC keyboard. The signal should switch between +5V and ground.

      If the signal does not switch between +5V and ground, check J1, pin 5 on the 5082 for a signal change of at least +/–3V.

      If you cannot get a signal at J1, pin 5, check your computer and make sure it is transmitting.

For communications packages other than PC SmartLINK, please note:

- The 5082 does not send a CTS signal to the host. This line is tied high. If your terminal or communications software requires other signals (DCD, DSR), you may have to tie these signals to the appropriate levels. You may be able to ignore these lines in software.

- The 5082 does not recognize the RTS line from the host to pace communications. This is important if your terminal's communications become garbled or otherwise unreadable. Most often this happens because the 5082 is continually sending out data and the receiving unit cannot handle all of it.

Please refer to *Using Other Serial Communications Software* in Appendix A for additional information.

Chapter 3 ADDING MEMORY

INTRODUCTION

The 5082 Microcontroller is shipped with a 32K static RAM and a 32K EEPROM in sockets U11 and U9, respectively. This chapter describes these memory chips and how to add memory to the 5082 card.

To store information in RAM, please see Chapter 4.

To store programs in the EEPROM or to autorun programs, see Chapter 5.

TYPES OF MEMORY

You can use 32K, 128K or 512K RAM in your 5082 Microcontroller. The RAM chip you select is installed in socket U11. RAM is used to temporarily store data, variables, and user programs. It provides volatile memory storage unless you are using a battery backup (the DS–1213D/DM SmartSocket, for example). You can also download programs from your PC to RAM and load information or programs from RAM back to your PC disk.

The 5082 supports a 32K EEPROM. The EEPROM is used to store programs. EEPROMs are nonvolatile and may be written to more than 10,000 times. The number of read cycles is unlimited. Programs are written to the EEPROM by using the on–card programmer. Programs are read from the EEPROM, downloaded (written to) the system RAM in U11, and run from system RAM.

You can also use a 32K EPROM in the 5082. It also provides nonvolatile memory and can be used to store programs. However, to store programs to the EPROM, you must use an off–card programmer.

ADDING MEMORY

Installing RAM Chips

The 5082 is shipped with a 32K static RAM installed in socket U11. The card is designed to accept 32K, 128K or 512K RAM chips.

To install a RAM chip:

1. Turn off power to the 5082.

2. Remove the memory chip from socket U11.

3. Install the new memory chip in socket U11.

4. Make the appropriate jumper connections in jumper W2:

5082 RAM Select: W2	
Pins Jumpered	**RAM Size**
[2-3]	32K, 128K*
[1-2]	512K

* = default

NOTE: Improper setting of W2 jumper may result in faulty operation.

Installing a Battery Backup Module

You can use a battery backup module, such as the Dallas DS–1213D/DM SmartSocket, for nonvolatile storage of system variables and data stored in your system RAM.

The DS–1213D/DM is a 32–pin module with built–in power fail circuitry and a dual battery system. The DS–1213D provides battery backup for a low power 32K or 128K static RAMs. The DS–1213DM battery backs 128K or 512K RAMs.

To install the SmartSocket:

1. Turn off power to the 5082.

2. Remove the memory chip from socket U11.

3. Install the DS–1213D or DS–1213DM into socket U11 with the index mark pointing towards the 5082 gold contact fingers.

4. Install the memory chip into the top of the module.

5. Make the appropriate jumper connections in jumper W2:

5082 RAM Select: W2	
Pins Jumpered	**RAM Size**
[2-3]	32K, 128K*
[1-2]	512K

* = default

NOTE: Special low power memory chips must be used or battery life will be drastically reduced.

Installing EEPROM Chips

The 5082 is shipped with a 32K EEPROM installed in socket U9 (the "user socket"). To change the device, you need to remove the original chip, install the new device, and change the jumper settings in jumper block W1.

NOTE: Improper setting of W1 may result in lack of operation.

To install an EEPROM chip:

1. Turn off power to the 5082.

2. Remove any memory chip from socket U9.

3. Install the new EEPROM in socket U9.

4. Make the appropriate jumper connections in jumper block W1:

5082 User Device Select: W1		
Device Size	**Device Type**	**Pins Jumpered**
32K	29C256 (EEPROM)*	[3-4][5-6]
32K	27C256 (EPROM)**	[3-4][5-6]

* = default

**NOTE: The on-card EEPROM programmer cannot be used with a 27C256. The EPROM must be programmed off card and then installed in U9 on the 5082.

This page intentionally left blank.

INTRODUCTION

Although RAM is volatile, it is usually the most convenient means of temporarily storing data, program variables, etc. When permanent storage is required, a battery–backup module can be added. This chapter describes the memory usage of CAMBASIC IV, and how to save and retrieve information from the static RAM.

User programs and CAMBASIC IV variables reside in segment 0 of RAM. Except for program and variable space, the remainder of RAM space is available for user data. When using a 128K or 512K RAM, it is recommended that user data be stored in segment 1 and higher so that there will be no conflict with CAMBASIC operations. With a 32K RAM only segment 0 is available, so care must be taken to properly reserve RAM space for user data.

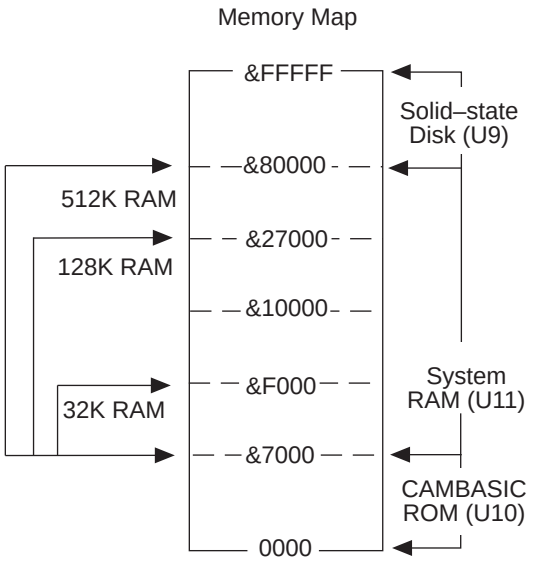
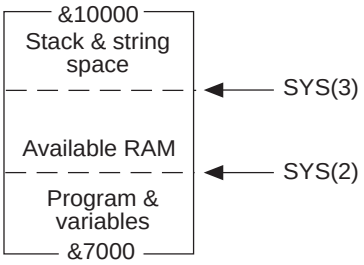
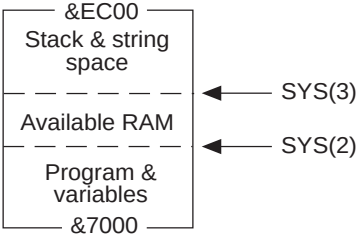


Figure 4–1—Memory Map



128K & 512K RAM System



32K RAM System

Figure 4–2—32K, 128K and 512K RAM

The following program demonstrates how to reserve 100 bytes of RAM space:

```
10 'Allocate 100 bytes for user data,
20 'e.g., assembly program space, in segment 0
30 'assuming 32K RAM installed.
40 ..
50 ..
60  SIZE=SYS(3)-SYS(2)       'Size=Total avail RAM
70 IF SIZE<100 THEN 200      'If no space, then error
80  ADR=SYS(3)-100           'Alloc 100 bytes from top
90 ..
100 'Do your stuff with ADR here.
110 END

200 PRINT "Not enough space for allocation!"
210 END
```

PROGRAMMING EXAMPLES

This section includes three examples: saving and retrieving a variable, a string, and an array. For more information, please refer to the *CAMBASIC IV Programming Guide*.

Saving and Retrieving a Variable

```
10 POKE 200,A,1     'Saves byte value into address 200 of
                     segment 1. A byte is any number
                     from 0 to 255. Words and floating
                     point numbers also saved by using
                     DPOKE (2 bytes) and POKE! (4
                     bytes).

50 A=PEEK(200,1)    'Retrieves a byte value. Words and
                     floating point numbers also retrieved
                     by using DPEEK and PEEK!
```

Saving and Retrieving a String

```
100 POKE 300,A$,1       'Pokes string variable A$ to
                         memory address 300 in segment
                         1

120 B$=PEEK$(300,1)     'Retrieves a string at address
                         300, segment 1
```

Saving and Retrieving an Array

A single dimensioned array may be stored and retrieved from RAM. Refer to the ARRAY statement and function in the *CAMBASIC IV Programming Guide* for additional information.

COMMANDS

The following is a list of the CAMBASIC IV commands used with the RAM:

5082 Memory Commands	
Command	**Function**
ARRAY	Save or return a value from extended memory
PEEK	The PEEK and POKE commands save and retrieve byte values
DPEEK	Returns 16-bit value from memory
DPOKE	Writes 16-bit value to memory address
PEEK$	Returns string from memory
PEEK!	Returns floating point number from memory
POKE	Writes byte into memory location
POKE$	Sends string to memory address
POKE!	Stores data in memory

Chapter 5 STORING IN EEPROM

INTRODUCTION

Programs are stored in the EEPROM, which is installed in socket U9. You can store one program with a maximum size of 32K if you are using a 32K EEPROM or you can store up to four programs, each with a maximum size of 8K. Refer to the CONFIG SSD command in the *CAMBASIC IV Programming Guide* for storage options.

The 5082 Microcontroller is shipped with a 32K EEPROM in socket U9. The EEPROM is nonvolatile (retains memory even when power is disconnected), has an unlimited number of read cycles, and each memory location can be written to more than 10,000 times. Programs are not run from EEPROM. Instead, the program is downloaded to RAM in socket U11. Programs in RAM can be run or modified, then uploaded to either your PC or the EEPROM. This chapter includes directions for loading and storing your program in the EEPROM. Please refer to Chapter 2 for information on transferring programs between RAM and your PC and to Chapter 4 for more on storing information in RAM.

You can set up your system to "autorun" your program (or series of programs). When autorun is on, the program in the EEPROM is loaded into RAM and begins to execute immediately on power–up. You can also link multiple programs, so that the first program to run calls the second, which calls the third, which calls the fourth.

The EEPROM is write–protected with a software lock so accidental writes on power–on or –off are almost impossible. You cannot disable or overwrite the lock except when executing the SAVE command. This means you cannot "POKE" data into the EEPROM.

CONFIGURING THE EEPROM

Before the EEPROM can be used to save programs, jumper block W1 must be configured for the "autorun enable" option. This is the factory default setting:

5082 Autorun Select: W1	
Pins Jumpered	**Function**
[1-2]*	Autorun enabled
Not jumpered	Autorun disabled

* = default

Next, the EEPROM must be configured using the CONFIG SSD command. This command configures the EEPROM for either one 32K program (type=0) or four 8K programs (type=1). It is executed only when you install a new EEPROM or want to change the configuration of an existing EEPROM. To configure the EEPROM, enter:

```
CONFIG SSD type
```

Type refers to the size of EEPROM and the number and size of files stored there. Refer to the *CAMBASIC IV Programming Guide* for additional information.

WARNING: CONFIG SSD erases the EEPROM. Any programs stored in the EEPROM cannot be recovered.

SAVING A PROGRAM IN EEPROM

To store a 32K program or the first of four 8K programs to the EEPROM, enter:

```
SAVE filename
```

The filename can have up to 12 characters.

Though compiling typically takes about 5 seconds, it can take as long as 15 seconds with very long or complex programs. The programming status displays on the screen:

```
compile. .program. .verify
```

The prompt (>) appears when the program has been successfully saved to the EEPROM. If the program does not properly write to the EEPROM, an error message appears:

```
Failed @ XXXX>
>__
```

XXXX is the hexadecimal address where the program failed.

If you have configured the EEPROM to store four 8K programs, subsequent programs can be stored using the SAVE command:

```
SAVE "BOOT"
SAVE "secondprog"
SAVE "thirdprogram"
```

If you plan to autorun your program on power–up, the first four letters of the first filename on the EEPROM must be “BOOT”. The filename can be anything you choose, up to twelve characters total.

LOADING PROGRAMS TO RAM

To examine the sizes and filenames of the files saved in the EEPROM, enter:

```
FILES
```

To load a program from EEPROM into RAM, enter:

```
LOAD "filename"
```

RENAMING A PROGRAM IN THE EEPROM

To rename a program that is stored in the EEPROM, enter:

```
RENAME "oldname","newname"
```

DELETING A PROGRAM FROM EEPROM

To delete a program from the EEPROM, enter:

```
DEL "filename"
```

RUNNING A PROGRAM FROM EEPROM

To run a program from the EEPROM, enter:

```
RUN "filename"
```

The file will load from the EEPROM to RAM and execute immediately. It has the same effect as entering LOAD “filename” and then entering RUN.

AUTOMATIC PROGRAM EXECUTION ON POWER–UP

Your 5082 Microcontroller is shipped with autorun configured as the default setting. On power–up CAMBASIC IV checks the contents of the files in the solid–state disk (SSD). If the first four letters of the first file are “BOOT”, it will load the file into RAM and run the program. It ignores any other part of the filename. For example, any one of the following will autoexecute, if it is the first program stored on the SSD:

BOOT
BOOT–PUMP2
BOOT12–1–90

The following files will not autoexecute:

BOOM
ABOOT
12BOOT

During program development, give your autoexecute file any name, then RENAME it to “BOOT” when it comes to run time.

To autorun a program:

1. Configure jumper block W1 [1–2] for autorun ON (this is its default).

2. If the EEPROM will store one 32K file, enter:

   CONFIG SSD 0

   If the EEPROM will store four 8K files, enter:

   ```
   CONFIG SSD 1
   ```

   NOTE: This command is only executed once when first configuring the EEPROM. This command will erase any existing data in the EEPROM.

3. SAVE your program from RAM to the EEPROM. The first four letters of the filename must be BOOT.

4. On power–up or reset, the program "BOOT....." will automatically download to RAM and execute.

Preventing Autorun

When troubleshooting a program, it may be necessary to disable the autorun option. This is especially true for those programs that have been configured to ignore the break <ESC> key.

To prevent autorun:

1. Power–off the 5082.

2. Remove the jumper at W1 [1–2].

3. Power–on the system.

4. Replace the jumper at W1 [1–2].

5. Use the RENAME command to rename "BOOT....." to any name not beginning with BOOT.

COMMANDS

The following is a list of CAMBASIC IV commands used in storing and running programs saved in the EEPROM.

5082 Commands to Store and Run Programs	
Command	**Function**
CONFIG SSD	Sets up EEPROM as a disk
DEL	Deletes program from EEPROM
FILES	Displays SSD files
LOAD	Downloads program from EEPROM to RAM
RENAME	Renames SSD files
RUN	Downloads and runs the specified program
SAVE	Saves program to EEPROM

Chapter 6 SERIAL PORTS

INTRODUCTION

The 5082 has three serial ports, COM1, COM2 and COM3, which can be used for interfacing to printers, terminals or other serial devices. Each port can be programmed independently with built–in CAMBASIC IV commands to operate on an interrupt–driven basis.

Each serial port has 256–character input and output buffers. Characters are stored in the input buffer and removed from the output buffer on an interrupt driven basis. Information is handled through the serial port without slowing down program execution. For example, when executing a PRINT statement, the processor sends the characters to the output buffer which prints in the background while the program continues to run. If the output buffer fills, program execution stops until there is room in the buffer. If the input buffer fills, all subsequent characters are discarded.

All three serial ports support XON/XOFF software protocol when transmitting so that a remote device can tell the 5082 to start and stop data transmission. XON and XOFF are special characters chosen to establish a flow control protocol. When XON is received, transmission is temporarily suspended until XOFF is received.

Use a VTC–9F cable to connect the ports to the external serial equipment. The pinout of the connector allows you to plug the cable directly into a 9–pin PC serial connector.

COM1 (J1)

During program development, COM1 is used to establish the communications link between your PC and the 5082. When used for programming or with the INPUT statement, COM1 will only accept ASCII characters with values from 0 to 127. When used with the INKEY$ and COM$ functions, it will also return ASCII data from 0 to 255.

During run time, COM1 can be used as a general purpose serial port. However, it does not support hardware handshaking.

COM2 (J4)

COM2 can be used as a general purpose serial port. It cannot be used to download or edit your programs. It supports two hardware handshake lines, CTS and RTS. The 5082 serial ports, COM2 and COM3, are configured as DCE (Data Communication Equipment), therefore RTS is an input and CTS is an output. On power–up, the 5082 does not enable the RTS or CTS control signals. It is up to the programmer to control these signals with CAMBASIC commands. The BIT command can be used to monitor the logic levels of the RTS line and control the CTS line. The following example demonstrates how to monitor and control the handshake lines:

```
70 ..
80 'The following program refers to COM2 only.
90 ..
100 'Set COM2 CTS low
110 'CTS is an output from the 5082
120 ..
130 BIT &80,4,OFF :'Set bit 4 of port addr &80 to low

200 'Loop to monitor COM2 RTS
210 'RTS is an input to the 5082
220 ..
230 ..rts_loop
240 RTS=BIT(&82,5)   :'Monitor bit 5 of port addr &82
250 IF RTS <> 1 THEN GOTO ..rts_loop
270 'Program continues when receiver has detected RTS

300 'Set COM2 CTS high
310 'CTS is an output from the 508X
320 ..
330 BIT &80,4,ON  :'Set bit 4 of port address &80

400 'RTS high detected, CTS set high, the program is
ready to receive characters.
410 ..
420 GOSUB 1000
430 ..

500 'After characters are received, CTS must be set low
510 'Set COM2 CTS low
520 'CTS is an output from the 508X
530 ..
540 BIT &80,4,OFF    :'Reset bit 4 of port addr &80
550 GOTO ..rts_loop
```

```
1000 'Receive character subroutine
1010 ..

1500 RETURN
```

COM3 (J5)

In addition to RTS and CTS, COM3 also supports DTR and DCD. The 5082 serial ports, COM2 and COM3, are configured as DCE (Data Communication Equipment), therefore RTS and DTR are inputs, CTS and DCD are outputs. On power–up, the 5082 does not enable the RTS, CTS, DCD or DTR control signals. It is up to the programmer to control these signals with CAMBASIC commands. COM3 can also be configured for a baud rate of 19200. These special features make this serial port ideal for modem operations. The following program is an example of how COM3 might be used in a typical application:

```
70 ..
80 'The following program refers to COM3 only.
90 ..
100 'Set COM3 CTS low
110 'CTS is an output from the 5082
120 ..
130 OUT &E8,5        :'Indicates access to Z80181 reg 5
140 OUT &E8,&68      :'Set bit 2, of Z80181 reg 5, low
                     (*see &68 below)
200 'Loop to monitor COM3 RTS
210 'RTS is an input to the 5082
220 ..
230 OUT &E8,15       :'Indicates access to Z80181 reg 15
240 OUT &E8,&D0      :'Enable bits 3 and 5
250 ..rts_loop
260  RTS=BIT(&E8,5) :'Monitor bit 5 of port address &E8
270 IF RTS <> 1 THEN GOTO ..rts_loop

270 'Program continues when receiver has detected RTS

300 'Set COM3 CTS high
310 'CTS is an output from the 5082
320 ..
330 OUT &E8,5        :'Indicates access to Z80181 reg 5
340 OUT &E8,&6A      :'Set bit 2, of Z80181 reg 5, high
                     (*see &6A below)
```

```
400 'RTS detected, CTS set high, the program is ready to
receive characters.
410 ..
420 GOSUB 1000
430 ..

500 'After characters are received, CTS must be set low
510 'Set COM3 CTS low
520 'CTS is an output from the 5082
530 ..
540 OUT &E8,5       :'Indicates access to Z80181 reg 5
550 OUT &E8,&68     :'Set bit 2, of Z80181 reg 5, low
                     (*see &68 below)
560 GOTO ..rts_loop

1000 'Receive character subroutine
1010 ..

1500 RETURN

The following code is for additional control
of COM3 control lines.

2000 'Set COM3 DCD low
2010 'DCD is an output from the 5082
2020 ..
2030 OUT &E8,5      :'Indicates access to Z80181 reg 5
2040 OUT &E8,&68    :'Set bit 3, of Z80181 reg 5, low
                     (*see &68 below)
2100 'Set COM3 DCD high
2110 'DCD is an output from the 5082
2120 ..
2130 OUT &E8,5      :'Indicates access to Z80181 reg 5
2140 OUT &E8,&E8    :'Set bit 7, of Z80181 reg 5, high
                     (*see &E8 below)
2200 'Loop to monitor COM3 DTR
2210 'DTR is an input to the 5082
2220 ..
2230 OUT &E8,15     :'Indicates access to Z80181 reg 15
2240 OUT &E8,&D0    :'Enable bits 3 and 5, if this is done
earlier in the program it does not have to be repeated
2260 ..dtr_loop
2270 DTR=BIT(&E8,3):'Monitor bit 3 of port address &E8
2280 IF DTR <> 1 THEN GOTO ..dtr_loop
```

* CTS and DCD are both affected by writing to register 5. Both can be controlled at the same time.

```
&68 = CTS and DCD low
&E8 = CTS low and DCD high
&6A = CTS high and DCD low
&EA = CTS and DCD high
```

NOTE: For further information on the Zilog Z80181 processor, please contact Zilog for the technical manual #DC-2519-02.

CHANGING THE SERIAL PARAMETERS

AutoBaud on COM1

During program development, when you power–on the 5082, the system displays a nonsense message if the baud rate of your PC is not set to 9600. If you press the <ESC> key, the autobaud feature automatically adjusts the baud rate to 300, 1200, 2400, 4800 or 9600 and a communications link is established with your PC. The autobaud feature also configures COM2 to match the parameters of COM1. The serial parameters of COM3 are not affected by the autobaud feature.

CONFIG BAUD Command

The default parameters for all ports are 9600 baud, 8 data bits, no parity and 1 stop bit. The CONFIG BAUD command allows you to change these parameters to fit your application needs. The following example shows a typical configuration:

```
1 'CONFIG BAUD a,b,c,d          a=COM port, b=baud rate, c=# of data & stop bits, d=even/odd parity
10 'Configure COM1 for 9600 baud, 8 data bits, no parity, 2 stop bits
20 ..
30 CONFIG BAUD 1,6,5,0
40  ..
50 'Configure COM2 for 2400 baud, 7 data bits, even parity, 2 stop bits
60 ..
70 CONFIG BAUD 2,4,3,0
80 ..
100 'Configure COM3 for 300 baud, 7 data bits, odd parity, 1 stop bit
```

```
120 ..
130 CONFIG BAUD 3,1,2,1
```

For further information, please refer to the *CAMBASIC IV Programming Guide.*

MULTIDROP COMMUNICATIONS NETWORK

You can use your 5082 Microcontroller with a multidrop communication network by connecting COM2 or COM3 to the network with an MTB–485 converter. The MTB–485 converts the serial port to RS–422 or RS–485. With the MTB–485, you can connect up to 32 units (including one or more 5082 Microcontrollers, PCs and other devices) over a 4,000 foot range.
Figure 6–1 shows an example of a multidrop communications network. The network includes a host and one or more devices. The host transmits signals to all the devices in the network. The signal includes an identifier as well as commands or data for one or more devices. The signal is received by all devices, and ignored by all devices except the one identified at the beginning of the signal. The identified device will transmit any required response back to the host.

There are several different communication protocols which establish how to encode the device identity in the signal. For example, you might set up your protocol so that the host sends a message in this format:

```
:03C4A52237
```

The colon (:) indicates that this is a message for everyone in the network; the 03 specifies which device actually needs to hear the message; and the rest of the information is the actual data or instructions. You can also have an identifier, such as 00, that identifies the message for all devices in the network. You can create your own protocol, but it must be consistent throughout the network. For example, you could use the symbol > instead of the colon.

To receive a string, use the CAMBASIC IV commands, CONFIG COM$ and ON COM$. The device will generate an ON COM$ interrupt when it receives a string from the host. You may want to

use the LEFT$ function to capture the first three characters and to determine if those characters of the string match the particular device's address (essentially parsing out the command). If the first three characters match the device, then the software parses the next character, determines what it means, moves to the next character, and so on. The MTB–485 automatically turns the transmitter on and off. For more information on the MTB–485, please refer to the *MTB–485 Product Sheet.*

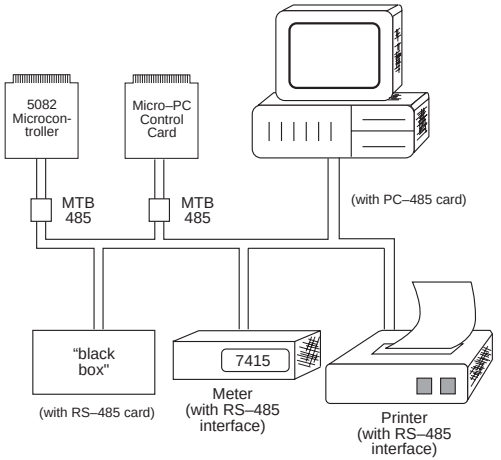


Figure 6–1—Multidrop Communications Network

ACCESSING COM1, COM2 AND COM3 INPUT BUFFERS

CAMBASIC IV supports three different methods of reading the input buffers. The following are examples of each method. You may need to consult the *CAMBASIC IV Programming Guide* for more information.

INPUT Statement

This removes all characters in the buffer up to the terminator character and puts them into a CAMBASIC IV variable. When you use the INPUT statement, CAMBASIC suspends operation until you enter the terminator character (carriage return). If the buffer

fills (256 bytes) and you have not entered a carriage return, all subsequent characters will be discarded. Whether this will be a problem depends on your particular application.

INKEY$ Function

Characters may be removed one at a time with this function. A null string is returned when the buffer is empty. In this mode, you have access to the full 256 bytes. If you don't read the buffer and the buffer fills, all subsequent characters will be discarded. The INKEY$ function may be used anywhere in the program.

ON COM$ Multitasking Statement

Characters are automatically buffered until a termination condition (which you specify) is reached. The program will then jump to a subroutine that removes the entire string from the buffer. In this mode you have access to only 128 of the 256 bytes at a time. If the number of characters in the buffer reaches 128 before meeting your termination conditions, the program will still jump to the subroutine. If you don't read the buffer and the buffer fills to 256 characters, all subsequent characters will be discarded. The advantage of this method is that the whole string is captured without halting program execution. Refer to the Multitasking Chapter, in the *CAMBASIC IV Programming Guide* for more information.

For information on using the commands, INPUT, INKEY$ and ON COM$ together, download the application note #AN–0059 from the Octagon Bulletin Board (303) 427–5368. (The settings are 2400, 8, N, 1.)

COM1 AND COM2 PORT STATUS

Use the SYS command to check the status of the COM port. Refer to the *CAMBASIC IV Programming Guide* for more specific information. Knowing the status helps to determine if there are any characters in the buffers or if a communication error has occurred. It is important to check for communication errors, especially if you are using the ON COM$ tasking statement. This ensures the integrity of the string that generated the interrupt,

that is, whether the interrupt was generated because the string was transmitted properly or because there was a communications error.

COMMANDS

The following is a list of the CAMBASIC IV commands used for serial I/O functions.

5082 Serial I/O Commands	
Command	**Function**
CLEAR COM$	Resets serial input buffer
CLS	Clears screen
COM$	Returns string from autobuffer
CONFIG BAUD	Sets serial port parameters (baud rate, start and stop bits, parity, and data length)
CONFIG COM$	Sets serial input buffer
INKEY$	Returns character from serial buffer
INPUT	Receives data from serial port
LIST	Outputs program listing
ON COM$	Enables serial tasking
PRINT	Outputs data through serial ports
PRINT!	Prints formatted strings or number
PRINT$	Prints list of numbers as characters
TAB	Tabs to specified position

This page intentionally left blank.

INTRODUCTION

Digital I/O lines can be used to interface with opto–module racks, operate switches, turn on low–current LEDs, and interface with other devices that have TTL input or output (for example, printers and scales). Figure 7–1 shows a typical opto–rack configuration:

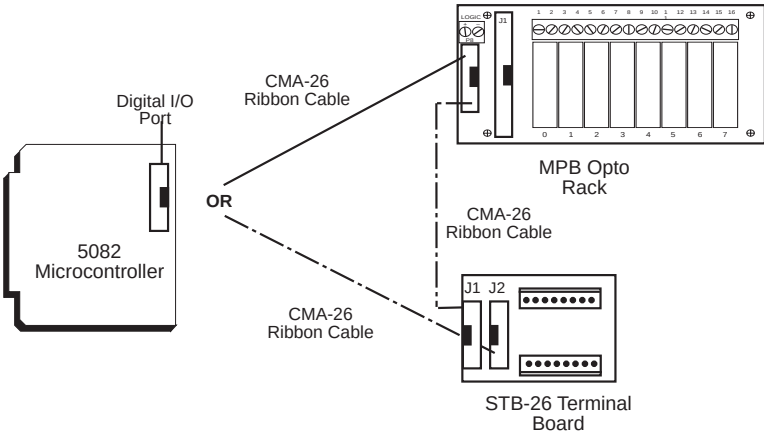


Figure 7-1—Typical Opto Rack Configuration

WARNING: Apply power to the 5082 Microcontroller before applying an input voltage to the digital I/O lines. This prevents excessive currents from flowing and damaging input devices.

ORGANIZATION OF CONNECTORS AND PORTS

The digital I/O lines on the 5082 are supplied by two 82C55 chips. One 82C55 is located in socket U1 which is connected to J2. The other is in socket U3 which is connected to J3. J6 and J7 are the keypad and display ports. However, they may be used as general digital I/O ports.

Input and Output Ports

On power–up and software or hardware reset, all the 82C55 digital I/O lines in J2 and J3 are configured as inputs. All lines are TTL logic level compatible (0–5V) and have 10K pull–up resistors to the 5V supply.

5082 Digital I/O Port: J2 and J3			
Port	I/O Address		Description
	J2	J3	
A	00	H40	8 lines which can be programmed as all inputs or all outputs. 8 lines interface to a high current driver.
B	01	H41	8 lines which can be programmed as all inputs or all outputs. 8 lines interface to a high current driver.
C	02	H42	8 lines which can be programmed as one group of 8 lines or two groups of 4 lines as all inputs or all outputs.

The 82C55 has three ports with eight parallel I/O lines (bits) per port. Each port has a unique I/O address. Ports A and B can be programmed as all inputs or all outputs. Port C can be programmed in one group of eight lines (all inputs or all outputs) or as two groups of four lines (upper and lower C). The four lines in upper or lower C can each be programmed as all inputs or all outputs.

When a line is configured as an output, it can sink a maximum of 2.5 mA at 0.4V and can source over 2.5 mA at 2.4V. When driving opto–modules, the output can sink 15 mA at 1.0V.

The keypad and display ports can also be used as general purpose digital I/O lines.

CONFIGURING THE 82C55 DIGITAL I/O LINES

On power–up or reset, all ports are inputs. Use the CONFIG PIO command to reconfigure I/O lines. The following programs demonstrate typical configurations:

Example 1: J2 Digital I/O

```
10 'Connect an 8–, 16–, or 24–position opto rack to J2
30 CONFIG PIO 0,1,1,1,0,1    'Configures lower port C
                              for outputs; all others for
                              inputs.

100 OUT 2,3                   'Outputs a "3" to lower
                              port C
110 B=INP(0)                  'Gets status from port A
```

Example 2: J3 Digital I/O

```
10 'Connect a STB-26 to J3
30 CONFIG PIO &40,0,0,0,1,1   'Configures 82C55
                               ports A & B as low
                               outputs, C as all inputs
40 'Perform other initialization routines
100 OUT &40,1                 'Sets port A, bit 0 to a 1
110 D=INP(&42)                'Reads all lines on port C
120 B=BIT(&42,1)              'Reads port C, bit 1
130 BIT &41,3,1               'Turns on bit 3 of high
                              current port B
```

NOTE: For Ports A and B, a "0" turns on a high current line; a "1" turns it off.

For further information, please refer to the *CAMBASIC IV Programming Guide.*

OPTO–MODULE RACK INTERFACE

You can interface digital I/O lines from J2 and/or J3 to an 8–, 16–, or 24–position opto–module rack. One end of the CMA–26 cable plugs into J2/J3 and the other plugs into an MPB–8, MPB–16, or MPB–24 mounting rack.

You can also use a CMA–26 cable to connect J2 and/or J3 on the 5082 to a STB–26 terminal board and then to the opto rack. The STB–26 has two 26–pin connectors, one of which plugs into J2/J3; the other plugs into the opto rack.

For either configuration, run a separate line to 5V and ground on the opto–module rack. Use the following table to determine the corresponding opto channel for a particular port.

5082 Opto Rack Interface		
Opto Channels	**82C55 Port**	**I/O Address**
0-3	Lower C	2
4-7	Upper C	2
8-15	A	0
16-23	B	1

INTERFACING TO SWITCHES AND OTHER DEVICES

The STB–26 terminal board provides a convenient way of interfacing switches or other digital I/O devices to the 82C55 digital port on the 5082 Microcontroller. All 82C55 lines have 10K pull–up resistors. I/O lines at connector J2 and/or J3 can be connected to the STB–26 with a CMA–26 cable. Parallel I/O devices are then connected to the screw terminals on the STB–26.

You can determine switch status as follows:

```
300  D=BIT(1,2)
```

Status of the switch is returned in the variable “D” as a zero (closed) or a one (open).

COMMANDS

The following is a list of the CAMBASIC IV commands used for digital I/O:

5082 Digital I/O Commands	
Command	**Function**
BIT	Function returns status of bit at I/O address
BIT	Statement sets a bit to 0 or 1 at an I/O address
CONFIG PIO	Initializes 82C55 drivers
CONFIG COUNT	Configures an I/O line for counter input
CONFIG FREQ	Sets the I/O address of the frequency inputs
CONFIG TIMER	Configures an I/O line for a timed output
DINP	Returns 16-bit value from I/O
DOUT	Writes 16-bit value to I/O
INP	Returns a byte from I/O
ON BIT	Declares I/O line to monitor logic level
ON INP	Causes interrupt when preset input pattern is detected
OUT	Writes a byte to I/O

This page intentionally left blank.

INTRODUCTION

The 5082 has a dedicated LCD display port (J7) with built–in contrast control. Supported displays range in size from two lines by 20 characters to four lines by 20 characters.

WARNING: The display port lines are unbuffered and connect directly to the microcontroller chip (Z80181). Care should be taken to avoid static discharge as this could damage the processor.

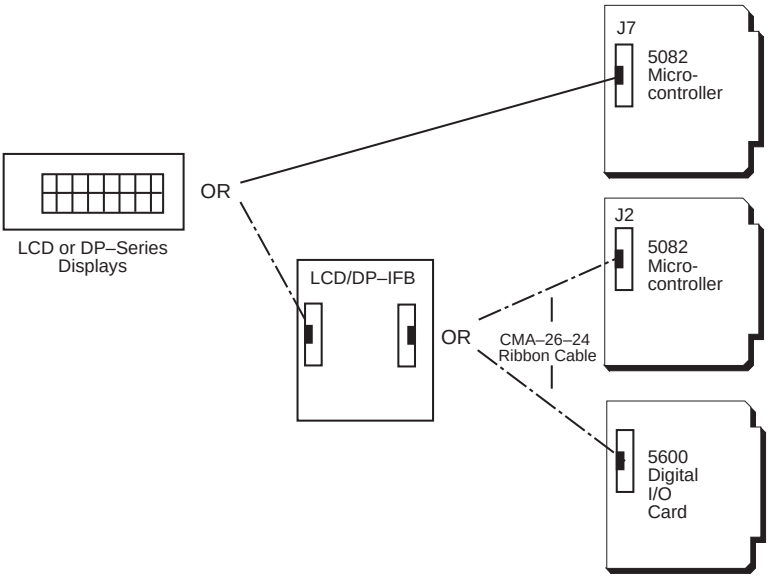


Figure 8–1—Display Configurations

CONNECTING A DISPLAY

Using J7

The most common configuration uses connector J7, which is configured as an output and directly supports an LCD display. If you don't use these lines to connect your display, you can use them as general purpose digital I/O lines.

1. Connect one end of the display cable (included with your display) to the display.

2. Connect the other end of the cable to J7 on the 5082.

3. You are now ready to initialize the display.

   The example below is for a 4x20 LCD display. Notice that all DISPLAY statements end with a semicolon so that a carriage return/line feed is not sent.

```
10 CONFIG DISPLAY &E2,6,0    'Config display port
20 A$="OCTAGON'S BUILT-IN"   'Build display strings
30 B$="DISPLAY PORT IS"
40 C$="EASY TO USE"
50 DISPLAY (0,1) A$;         'Display strings
60 DISPLAY (1,1) B$;
70 DISPLAY (2,1) C$;
80 END
```

NOTE: If your LCD does not display a good contrast between the characters and the background, adjust the LCD contrast pot.

Using J2

If you are using an LCD display larger than 4x20 or are using a DP–series (vacuum fluorescent) display with the 5082, you must connect the display to the digital I/O port (J2) using an LCD– or DP–IFB board.

1. Connect the cable included with your display to the display and the IFB.

2. Connect a CMA–26 cable from the IFB to J2 on the 5082.

3. You are now ready to initialize the display.

   The following program demonstrates how to drive a 4x20 LCD via the J2 digital port:

```
10   CONFIG PIO 0,0,0,0,0,0    'Config dig I/O port
20   CONFIG DISPLAY 0,6,0      'Config display type
30   A$="OCTAGON'S BUILT-IN"   'Build display strings
30   B$="DISPLAY COMMAND"
50   C$="IS EASY TO USE"
60   DISPLAY (0,1) A$;         'Display strings
70   DISPLAY (1,1) B$;
80   DISPLAY (2,1) C$;
90   END
```

CONNECTING A DISPLAY TO A 5600 DIGITAL I/O CARD

You can also connect your display to the optional 5600 or 5600–48 Digital I/O Expansion Cards with a LCD– or DP–IFB. This is especially useful if you are using a keypad larger than 4x4 or want to use J2 for something other than a display.

1. Connect the cable included with your display to the display and the IFB.

2. Connect the cable included with your keypad (if you are using one) to the keypad and the IFB.

3. Connect a CMA–26 cable from the IFB to the 5600 port you want to use. The 5600 has four 26–pin connectors. All four are identical and all have three ports: A, B and C.

4. You are now ready to initialize the display:

```
10 CONFIG PIO x,0,0,0,0,0      'Config dig I/O port;
                               x=addr of port A
20 CONFIG DISPLAY x,6,y        'Config display type
30 A$="OCTAGON'S BUILT-IN"     'Build display strings
40 B$="DISPLAY COMMAND"
50 C$="IS EASY TO USE"
60 DISPLAY (0,1) A$;           'Display strings
70 DISPLAY (1,1) B$;
80 DISPLAY (2,1) C$;
90 END
```

Line 10: x specifies the address of the 5600 Card, Port A.
Example: `CONFIG PIO &100,0,0,0,1,1`

Line 20: x specifies the address of the 5600 Card, Port A.
y specifies the type of cursor.
Example: `CONFIG DISPLAY &100, 6 0`

COMMANDS

The following is a list of the CAMBASIC IV commands for the display:

5082 Display Commands	
Command	**Function**
CONFIG DISPLAY	Specifies which multi-line display you want to use with the display statement. For more information on specific displays, refer to the CAMBASIC IV Programming Guide.
DISPLAY	Displays the strings at the row and column specified.

INTRODUCTION

You can use 4x4 (16–key) and larger keypads with the 5082 Microcontroller. You can connect a 4x4 keypad directly to J6 on the 5082 with the cable included with your keypad. Larger keypads must be connected to an interface board (IFB); the IFB is then connected to either J2 on the 5082 or to a port on the 5600 Digital I/O Expansion Card.

WARNING: The keypad port lines are unbuffered and connect directly to the microcontroller chip (Z80181). Care should be taken to avoid static discharge as this could damage the processor.

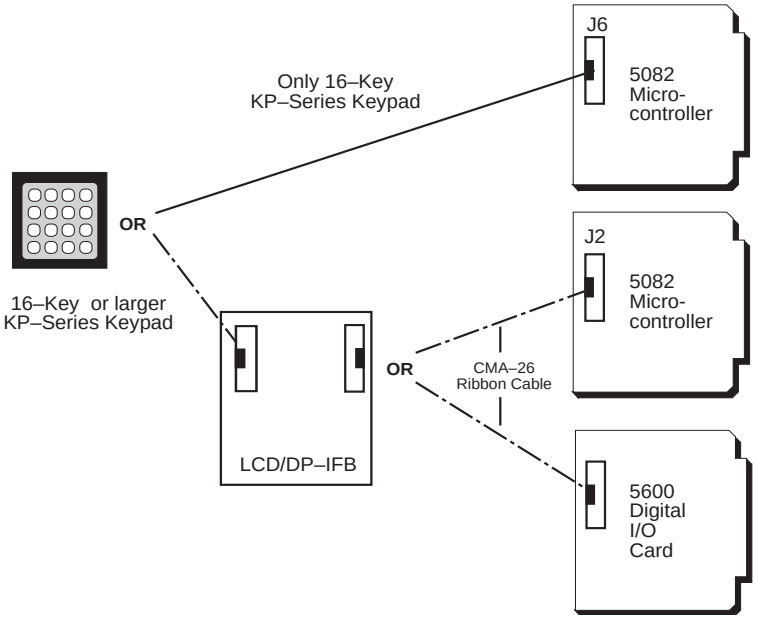


Figure 9–1—Keypad Configurations

CONNECTING THE KEYPAD

Using J6

The most common configuration uses J6, which is already configured. If you don't connect your keypad to these lines, you can use them as general purpose digital I/O lines.

1. Connect one end of the keypad cable (included with your keypad) to the keypad.

2. Connect the other end of the cable to J6 on the 5082 card.

3. You are now ready to initialize the keypad. The example below is for a 16–key keypad:

```
10 CONFIG KEYPAD$ 8              'Use def 16–kp type
20 ON KEYPAD$ GOSUB ..GETKEY     'Enable kp task

100 ..MAIN                       'Idle loop
110 GOTO .. MAIN

200 ..GETKEY                     'Kp service routine
210 A$=KEYPAD$(0)                'Get key
220 PRINT A$                     'Echo key pressed
230 RETURN
```

Using J2

If you are using a keypad with more than 16 keys, you must it to the digital I/O port (J2) using an LCD– or DP–IFB board. The DP–IFB is used with vacuum fluorescent displays and the LCD–IFB for liquid crystal displays.

1. Connect the cable included with your keypad to the LCD–IFB or DP–IFB.

2. Connect a CMA–26 cable from the IFB to J2 on the 5082.

3. You are now ready to initialize the keypad. The following example is for a 64–key keypad:

```
10 CONFIG PIO &0,0,0,0,1,1    'Set up PIO port
20 CONFIG KEYPAD$ 8,64,&1     'Use 64–keypad type
30 ON KEYPAD$ GOSUB ..GETKEY 'Enable keypad task

100 ..MAIN                    'Idle loop
110 GOTO ..MAIN

200 ..GETKEY                  'Kypd service routine
210 A$=KEYPAD$(0)             'Get key
220 PRINT A$                  'Echo key pressed
230 RETURN
```

NOTE: The CONFIG PIO and CONFIG KEYPAD$ statements are required for a 64–key keypad. Other size keypads may require modification to the statement. For example, the following is required for a 16–key keypad connected to J2:

```
CONFIG PIO &0,0,0,0,1,0
CONFIG KEYPAD$ 8,16,&2
```

CONNECTING THE KEYPAD TO A 5600 DIGITAL I/O CARD

You can also connect your keypad to the optional 5600 or 5600–48 Digital I/O Expansion Cards with a LCD– or DP–IFB. This is especially useful if you are using a keypad larger than 4x4 or want to use J2 for something other than a keypad.

1. Connect your keypad cable (included with your keypad) from the keypad to the IFB. If you're using a 4x4 keypad, use the 10–pin connection; for larger keypads, use the 16–pin connection.

2. Connect the IFB to the selected port on the 5600 Digital I/O Expansion Card.

3. You are now ready to initialize the keypad:

```
10 CONFIG PIO x,0,y,0,1,1     'Set up PIO port
20 CONFIG KEYPAD$ 8,64,x      'Use 64–keypd type
30 ON KEYPAD$ GOSUB ..GETKEY 'Enable keypd task

100 ..MAIN                    'Idle loop
110 GOTO ..MAIN
```

```
200  ..GETKEY                    'Kypd service routine
210  A$=KEYPAD$(0)               'Get key
220 PRINT A$                     'Echo key pressed
230 RETURN
```

Line 10: X specifies the address of port A; Y specifies the status of Port A as an input or output. If you are using a display, Y should be zero. If you are not using a display, Y can be 1 or 0 (input or output). Example: `CONFIG PIO &100,0,0,0,1,1`

Line 20: The value 64 indicates that you are using a keypad larger than 4x4. (This example is for a 64–key keypad.) X is the address of the Port B. When using the 5600 Digital I/O Expansion Card, the value for X depends upon the address of the port and connector you are using. Please refer to the *5600 I/O Expansion Card User's Manual* for information on determining your 5600 port address. Example: `CONFIG KEYPDA$ 8,64,&101`

COMMANDS

The following is a list of the CAMBASIC IV commands for the keypad:

5082 Keypad Commands	
Command	**Function**
CONFIG KEYPAD$	Sets keypad parameters
KEYPAD$(0)	Returns the last key from the keypad port
ON KEYPAD$	Enables keypad tasking
SYS(12)	Returns keypad string table address

INTRODUCTION

The high current port can be used to drive relays, LEDs, solenoids, and similar devices. The port includes 16 I/O lines at J3 on the 5082. These outputs switch load to ground. If you do not use this port to drive high current devices, you can replace the output driver chip, U2 and U4, with DIP shunt jumpers and use the lines as general purpose digital I/O lines. The lines are TTL compatible with the DIP shunt in place. The addresses of the high current ports are 40H for Port A at J3 and 41H for Port B at J3.

On power–up, all high current outputs are OFF. A “1” or “ON” written to the high current port causes the output to switch low or ON.

NOTE: When ON, the saturation voltages are incompatible with TTL logic levels and should not be used to drive other logic devices.

Considerations For High Current Outputs

- Each of the high current outputs can sink 100 mA at 50V.
- Incandescent lamps have a “cold” current of 11 times that of their “hot” current. It is recommended that lamps requiring more than 50 mA not be used.
- When inductive loads are used, protection diodes or other schemes must be used.

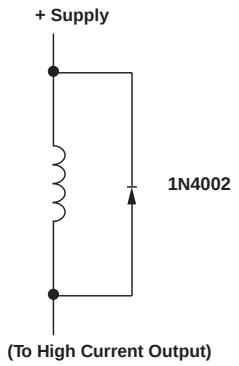


Figure 10–1 — Inductive Load Protection Circuitry

- Configuring outputs in parallel for higher drive is NOT recommended and could result in damage since the outputs will not share current equally.

WARNING: If external devices, such as 24 VDC relays, are driven, the ground of the external 24V supply must be connected to J3, pin 26 and NOT the power ground. Failure to do so will produce a ground loop within the 5082 and can cause erratic operation.

The watchdog timer is a fail-safe against program crashes or processor lockups. The following table shows how to enable, disable, and reset the timer.

5082 Watchdog Timer	
Task	**Command**
To enable	OUT &60,0
To disable	variable=INP(&60)
To reset	OUT &60,1

Once the watchdog time is enabled, it must be reset at a maximum of 1.2 second intervals, otherwise the system will reboot.

WATCHDOG LATCH

A watchdog latch allows the addition of a buzzer or light in series with a button which can be reset should the watchdog timeout. The latch is accessed through W3.

5082 Watchdog Latch	
Pin #	**Signal**
1	Gnd
2	Output
3	Gnd

The following is an example circuit for the watchdog latch. The LED could be replaced by a relay for driving devices that require higher voltage or current. The SCR output used is a Trecor EB103B or a Motorola MCR100–4. Refer to the appropriate data sheets for maximum ratings.

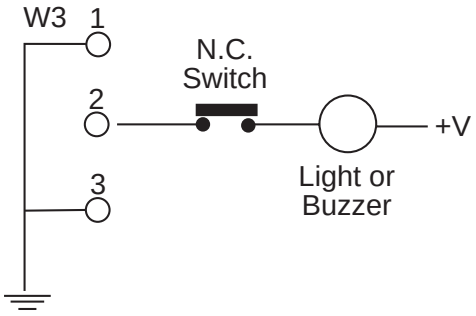


Figure 11–1—Example Watchdog Latch Circuit

DS–1216EM SMARTWATCH

The Dallas DS–1216EM is a 28–pin DIP socket which contains a calender/clock and a dual battery system with a minimum life of 5 years. Accuracy of the clock is about 1 minute/month at 25°C.

To install the DS–1216EM SmartWatch at U9:

1. Remove the memory chip from socket U9.

2. Install the DS–1216EM SmartWatch into socket U9 with the index mark pointing towards the 5082 gold contact fingers.

3. Install the EEPROM into the top of the SmartWatch.

4. The SmartWatch is shipped with its internal oscillator turned off to maximize battery life. To turn it on, type:

   ```
   CONFIG CLOCK ON
   ```

   You need to do this only once.

5. To test the clock, type:

   ```
   DATE$="01-15-91"
   PRINT DATE$(0)
   ```

   If the date prints out as shown below, the clock is working properly.

   ```
   01-15-91
   ```

6. You are now ready to set the time. Use the TIME$ statement as follows:

```
TIME$="11:23:45"
```

To read the time, type:

```
PRINT  TIME$(0)
```

NOTE: Writing to the clock will not affect any information stored in the EEPROM.

COMMANDS

The following is a list of the CAMBASIC IV commands for the calendar/clock:

5082 Calendar/Clock Commands	
Command	**Function**
DATE$	Reads/sets date from calendar/clock
TIME$	Reads/sets timer for calendar/clock

Chapter 13 INTERRUPT HANDLING

INTRODUCTION

The 5082 can be mounted in a passive backplane for use with external interrupt devices. You can use a switch closure, magnetic pick up, or other device capable of switching between +5V and ground for external interrupt generation. Refer to the following table for 5082 interrupt connections:

5082 Interrupt Input Sources				
Z80181 Designation	**CAMBASIC Interrupt #**	**Active Level**	**Bus Name**	**Edge Conn Pins**
INT1	3	High	IRQ3	B25
INT2	4	High	IRQ4	B24

INTERRUPT GENERATION

An interrupt is generated when a logic signal level on an interrupt input switch goes high for 10 uS. If you include an ON ITR *n* GOSUB label command in your program, it will branch to the subroutine label when the interrupt is generated. The subroutine designated by this command is executed after the current CAMBASIC IV statement completes execution. The average latency is about 0.5 mS.

PROGRAMMING EXAMPLE

The following programming example shows how to set up the interrupt and service routine. When the IRQ3 line goes high, the interrupt service routine is executed and the message “Door Open” is printed.

```
20 ON ITR 3 GOSUB 500   'Sets interrupt for line 500
 .
 . your program continues
 .
500 'interrupt routine here
510 PRINT "Door Open"
520 DR = 1               'Sets flag to signal door open
530 RETURN ITR 3         'Ret from interrupt routine to
                          normal program flow
```

COMMANDS

The following is a list of the CAMBASIC IV commands used for interrupt functions:

5082 Interrupt Commands	
Command	**Function**
ON ITR	Enables a program branch on an interrupt
RETURN ITR	Re-enables an interrupt and returns program control

Appendix A COMPONENTS

This appendix contains information which may be helpful to you if you are using non–Octagon components with your 5082 Microcontroller. For additional information, please refer to the user manuals supplied with your equipment.

CUSTOM COMMUNICATION CABLE

The 5082 requires an RS–232 serial communications cable to interface to the PC. If you are not using a VTC series cable, you can make your own communications cable.

1. Determine if your PC requires a male or female connector.

2. Refer to the following table for cable connections needed with the 5082:

5082 Custom Communication Cable				
J1	**Signal**	**Function**	**DB-25**	**DB-9**
9	Gnd	Common	7	5
3	TxD	Transmitted data	3	2
5	RxD	Received data	2	3

POWER MODULE

The 5082 needs a 5V power module, which is generally purchased and installed with your card cage. If you are not using the Octagon power module, you can use any well–regulated power module that can furnish at least 200 mA.

Power module voltage should be in the 5 +/–0.25V range when measured at the connector pins. Ripple should be less than 50 mV. If you are using a “switching” power module, excessive high frequency noise may cause erratic operation. The noise plus the power module voltage must not vary outside the 5V +/–0.25V range.

To use your power module with the 5082 Microcontroller:

- Connect the negative lead from your power module to the terminal marked DG on the backplane. Connect the +5 lead to the terminal marked +5. The leads should be no longer than 18 inches and must be 18 gauge or larger.

Please note that with some expansion cards (for example, the 5700 12–bit Analog Input Card), you will need +/–12 volts power module. Refer to the expansion card's user's manual for more technical specifications.

USING OTHER SERIAL COMMUNICATIONS SOFTWARE

PC SmartLINK is a serial communications software package designed by Octagon. It provides communications between a PC and other equipment. If you are using a software package other than PC SmartLINK, you may need to make some modifications before your serial communications will work. (Please note that although other packages may be used successfully, they are not supported by Octagon.)

Some issues and considerations:

- CAMBASIC IV is an incremental line compiler. As you type in a line of your program, for example,

  ```
  1000 IF A=3 then print "A=3"
  ```

  CAMBASIC compiles that line into an intermediate code. The time needed to compile the line depends on the complexity of the statement and how many statements have already been compiled. The longer and/or more complex the program, the longer it takes to compile. It also takes longer to download a program that is replacing an existing program.

  CAMBASIC must finish compiling one line before beginning the next line. When the line has been compiled, CAMBASIC sends a pacing prompt (the “>” character), which is recognized by PC SmartLINK. When PC SmartLINK detects the pacing prompt, it sends the next line of code to CAMBASIC.

CAMBASIC uses the carriage return line feed sequence followed by the ">" as the pacing prompt.

Your communications program should wait for a > character at the beginning of each line, otherwise you will begin to lose characters when the program gets too big.

- To upload a program (transfer it from the 5082 RAM to your PC), set up your PC to receive a file. Refer to your communications software documentation for details.

- The 5082 Microcontroller sends out a "clear screen" command on power–up or reset. This command is essentially an escape sequence that clears the screen. You may see the symbol <—; on your screen in response to the command. Ignore the symbol. It will not affect any other aspect of your system.

- If your communications setup runs at a baud rate other than 9600, you will get a nonsense message when you power-up. Ignore the nonsense message (and the symbol <—; if it appears) and press the <ESC> key; the logon message should appear. See Chapter 2 for more information.

- CAMBASIC IV sends out escape sequences to control functions on your PC. If you are not using PC SmartLINK, you may get unpredictable results on your software package or terminal.

  If you are having problems, turn off the escape sequences while in CAMBASIC IV by typing:

  ```
  CONFIG TERM 1
  ```

  The CLS command sends out an <ESC>;, which may appear as <—; on your screen.

- Saving and loading CAMBASIC IV programs using another communications program depends upon its command set. Usually, you must set up your communications program to receive or transmit a file. To save a program, type "LIST", set up your communications program to receive a file, then press <RETURN>.

- The serial port COM1 (J1) does NOT use handshaking or control lines.

- Set up your serial communications software to transmit a file when downloading a program to the 5082.

SPECIFICATIONS

- CPU
- Z80181 (Z80 code compatible), 9.216 MHz

Memory

- 29K, CAMBASIC IV ROM
- up to 512K static RAM
- 32K nonvolatile memory (EEPROM)

5082 Digital I/O

The 5082 has 64 digital I/O lines using two 82C55 chips. Forty–eight of the I/O lines are terminated with 26–pin IDC connectors. The remaining 16 I/O lines are terminated with 10K pull–up resistors on IDC connectors for a keypad and display.

You can use an MPB–8, –16, or –24 opto–module rack by connecting the rack to J2 or J3 with a CMA–26 cable.

Drive Current:	2.5 mA max. per line, sink or source. (TTL compatible)
Output Low Voltage:	0.45V max. at 2.5 mA, 1V max. at 12 mA. (opto rack compatible)
Output High Voltage:	2.4V min.

Keypad Input–J6

Connector J6 may be used with a 4x4 keypad. It will also accept four switch contacts. Debouncing and scanning are done with software commands. Use Octagon KP–1, KP–2–16, or KP–3 keypad and cable. I/O address is E1H.

Display Port – J7

Can be used with any LCD display with up to 80 characters. I/O address is E2H.

Serial Ports – J1, J4, J5

Three RS–232C serial ports. COM1 (J1) supports the RxD and TxD signals. COM2 (J4) supports the RxD and TxD signals as well as the RTS and CTS handshake lines. COM3 (J5) supports the RxD

and TxD signals, as well as the RTS, CTS, DCD and DTR handshake signals.

EEPROM Programmer

Programs 32K, including:

5082 EEPROMs		
Vendor	**Size**	**Order #**
Octagon 29C256	32K	2527
Atmel	32K	29C256-20PC

Speed for all EEPROMs must be 200 nS or faster.

Pushbutton Reset

The pushbutton reset switch will reboot the system.

Power Requirements

5V +/–5% @170 mA typical

The RS–232C supply voltages are generated on–card.

When using opto mounting racks with the system, the 5V supply requirements increase by 12 mA per opto module.

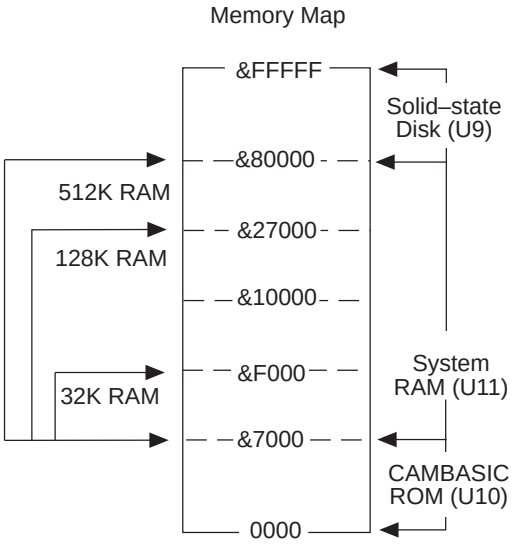
Environmental

–40° to 70° C operating
–40° to 85° C nonoperating
RH 5% to 95% noncondensing

Size

4.5 in. X 4.9 in.
(12.45 X 11.43 cm)

Memory Map


Memory Map
&FFFFF
Solid–state
Disk (U9)
&80000
512K RAM
&27000
128K RAM
&10000
&F000
System
RAM (U11)
32K RAM
&7000
CAMBASIC
ROM (U10)
0000

I/O Map

5082 I/O Map	
Description	**Address**
J2 Digital I/O	00-03
J3 Digital I/O	40H-43H
Watchdog enable	60H
Z80181 Internal Register	80H-BFH
J6 Keypad	E1H
J7 Display	E2H
Expansion	100H-3FFH
	1E0H-1EFH Reserved
	2E0H-2EFH Reserved
	3E0H-3EFH Reserved

JUMPER CONFIGURATIONS

5082 User Device Select: W1		
Device Size	**Device Type**	**Pins Jumpered**
32K	29C256 (EEPROM)*	[3-4][5-6]
32K	27C256 (EPROM)**	[3-4][5-6]

* = default
**NOTE: The on-card EEPROM programmer cannot be used with a 27C256. The EPROM must be programmed off card and then installed in U9 on the 5082.

5082 Autorun Select: W1	
Pins Jumpered	**Function**
[1-2]*	Autorun enabled
Not jumpered	Autorun disabled

* = default

5082 RAM Select: W2	
Pins Jumpered	**RAM Size**
[2-3]	32K, 128K*
[1-2]	512K

* = default

5082 Reset Indicator: W3	
Pin #	**Function**
1 and 3	Gnd
2	Reset output

CONNECTOR PINOUTS

5082 COM1, 2, 3 serial ports: J1, J4, J5					
Function	**Pin #**	**Dir**	**1**	**2**	**3**
DCD	1	Out			●
TxD	3	Out	●	●	●
RTS	4	In		●	●
RxD	5	In	●	●	●
CTS	6	Out	●	●	●
DTR	7	In			●
Gnd	9	Out	●	●	●
+5V	10	Out	●	●	●

5082 Digital I/O: J2			
I/O Line	**Port A**	**Port B**	**Port C**
Line 0	19	10	13
Line 1	21	8	16
Line 2	23	4	15
Line 3	25	6	17
Line 4	24	1	14
Line 5	22	3	11
Line 6	20	5	12
Line 7	18	7	9
+5V - Pin 2 Gnd - Pin 26			

5082 Digital I/O: J3			
I/O Line	**Port A**	**Port B**	**Port C**
Line 0	19*	10*	13
Line 1	21*	8*	16
Line 2	23*	4*	15
Line 3	25*	6*	17
Line 4	24*	1*	14
Line 5	22*	3*	11
Line 6	20*	5*	12
Line 7	18*	7*	9
+5V - Pin 2			
Gnd - Pin 26			

* = These lines are also high current.

5082 Keypad Port: J6	
Function	**Pin #**
Row 1	1
Row 2	4
Row 3	5
Row 4	8
Gnd	10
Column 1	6
Column 2	3
Column 3	2
Column 4	7

5082 Display Port: J7	
Function	**Pin #**
+5V	1
Gnd	2
Data 4	3
Contrast	4
Data 6	5
Data 5	6
NC	7
NC	8
NC	9
NC	10
Data 1	11
Data 0	12
Data 3	13
Data 2	14

PC BUS PINOUTS

Microcontroller Side "A"					
Pin #	**Description**	**Signal**	**Pin #**	**Description**	**Signal**
A1	I/O CH CK*	NMI	A17	A14	Not used
A2	D7	I/O	A18	A13	Not used
A3	D6	I/O	A19	A12	Not used
A4	D5	I/O	A20	A11	Not used
A5	D4	I/O	A21	A10	Not used
A6	D3	I/O	A22	A9	O
A7	D2	I/O	A23	A8	O
A8	D1	I/O	A24	A7	O
A9	D0	I/O	A25	A6	O
A10	I/O CH RDY	I	A26	A5	O
A11	AEN	Gnd	A27	A4	O
A12	A19	Not used	A28	A3	O
A13	A18	Not used	A29	A2	O
A14	A17	Not used	A30	A1	O
A15	A16	Not used	A31	A0	O
A16	A15	Not used			

* = active low

Microcontroller Side "B"					
Pin #	**Description**	**Signal**	**Pin #**	**Description**	**Signal**
B1	GND	O	B17	DACKI*	Not used
B2	RESET	O	B18	DRQ1	Not used
B3	+5V	I	B19	DACK0*	Not used
B4	IRQ2	I	B20	CLOCK	O
B5	-5V	Not used	B21	IRQ7	Not used
B6	DRQ2	Not used	B22	IRQ6	Not used
B7	-12V	Not used	B23	IRQ5	Not used
B8	Reserved	Not used	B24	IRQ4	I
B9	+12V	I	B25	IRQ3	I
B10	Analog Gnd	Not used	B26	DACK2*	Not used
B11	MEMW*	Not used	B27	T/C	Not used
B12	MEMR*	Not used	B28	ALE	Not used
B13	IOW*	O	B29	Aux +5V	Not used
B14	IOR*	O	B30	OSC	Not used
B15	DACK3*	Not used	B31	Aux Gnd	I
B16	DRQ3	Not used			

* = active low

WARRANTY

Octagon Systems Corporation (Octagon), warrants that its standard hardware products will be free from defects in materials and workmanship under normal use and service for the current established warranty period. Octagon's obligation under this warranty shall not arise until Buyer returns the defective product, freight prepaid to Octagon's facility or another specified location. Octagon's only responsibility under this warranty is, at its option, to replace or repair, free of charge, any defective component part of such products.

LIMITATIONS ON WARRANTY

The warranty set forth above does not extend to and shall not apply to:

1. Products, including software, which have been repaired or altered by other than Octagon personnel, unless Buyer has properly altered or repaired the products in accordance with procedures previously approved in writing by Octagon.
2. Products which have been subject to power supply reversal, misuse, neglect, accident, or improper installation.
3. The design, capability, capacity, or suitability for use of the Software. Software is licensed on an "AS IS" basis without warranty.

The warranty and remedies set forth above are in lieu of all other warranties expressed or implied, oral or written, either in fact or by operation of law, statutory or otherwise, including warranties of merchantability and fitness for a particular purpose, which Octagon specifically disclaims. Octagon neither assumes nor authorizes any other liability in connection with the sale, installation or use of its products. Octagon shall have no liability for incidental or consequential damages of any kind arising out of the sale, delay in delivery, installation, or use of its products.

SERVICE POLICY

1. Octagon's goal is to ship your product within 10 working days of receipt.
2. If a product should fail during the warranty period, it will be repaired free of charge. For out of warranty repairs, the customer will be invoiced for repair charges at current standard labor and materials rates.
3. Customers that return products for repairs, within the warranty period, and the product is found to be free of defect, may be liable for the minimum current repair charge.

RETURNING A PRODUCT FOR REPAIR

Upon determining that repair services are required, the customer must:

1. Obtain an RMA (Return Material Authorization) number from the Customer Service Department, 303-430–1500.
2. If the request is for an out of warranty repair, a purchase order number or other acceptable information must be supplied by the customer.
3. Include a list of problems encountered along with your name, address, telephone, and RMA number.
4. Carefully package the product in an antistatic bag. (Failure to package in antistatic material will VOID all warranties.) Then package in a safe container for shipping.
5. Write RMA number on the outside of the box.
6. For products under warranty, the customer pays for shipping to Octagon. Octagon pays for shipping back to customer.
7. Other conditions and limitations may apply to international shipments.

NOTE: PRODUCTS RETURNED TO OCTAGON FREIGHT COLLECT OR WITHOUT AN RMA NUMBER CANNOT BE ACCEPTED AND WILL BE RETURNED FREIGHT COLLECT.

RETURNS

There will be a 15% restocking charge on returned product that is unopened and unused, if Octagon accepts such a return. Returns will not be accepted 30 days after purchase. Opened and/or used products, non-standard products, software and printed materials are not returnable without prior written agreement.

GOVERNING LAW

This agreement is made in, governed by and shall be construed in accordance with the laws of the State of Colorado.

The information in this manual is provided for reference only. Octagon does not assume any liability arising out of the application or use of the information or products described in this manual. This manual may contain or reference information and products protected by copyrights or patents. No license is conveyed under the rights of Octagon or others.