
5081 User's Manual

Doc. #03280 Rev 0194

OCTAGON SYSTEMS CORPORATION®

6510 W. 91st Ave. Westminster, CO 80030

Tech. Support: 303-426-4521

COPYRIGHT

Copyright 1992–94—Octagon Systems Corporation. All rights reserved. However, any part of this document may be reproduced, provided that Octagon Systems Corporation is cited as the source. The contents of this manual and the specifications herein may change without notice.

TRADEMARKS

Micro PC™, PC SmartLink™, Octagon Systems Corporation®, the Octagon logo and the Micro PC logo are trademarks of Octagon Systems Corporation. SmartSocket™ and SmartWatch™ are trademarks of Dallas Semiconductor.

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	1
Conventions used in this manual	1
Symbols and terminology	2
Technical support	4
CHAPTER 1: OVERVIEW	5
Description	5
Major Features	6
CHAPTER 2: SETUP & OPERATION	9
Equipment	9
Installing the 5081 Card	10
Uploading & Downloading Programs	13
Uploading Your Program	13
Downloading Your Program	13
Where To Go From Here	14
Troubleshooting	15
CHAPTER 3: ADDING MEMORY	17
Introduction	17
Types of Memory	17
Adding Memory	17
Installing RAM Chips	17
Installing a Battery Backup Module	17
Installing EPROM and EEPROM Chips	18
CHAPTER 4: STORING IN RAM	21
Programming Examples	22
Saving and Retrieving a Variable	22
Saving and Retrieving a String	22
Saving and Retrieving an Array	23
Commands	23

CHAPTER 5: STORING IN EEPROM..... 25

Introduction	25
Saving a Program in EEPROM	25
Loading Programs To RAM	27
Deleting a Program From EEPROM	27
Automatic Program Execution On Power-Up	28
Autorunning Object Code Programs	29
Preventing Autorun	29
Linking Multiple Programs	30
Commands	30

CHAPTER 6: SERIAL PORTS..... 31

Introduction	31
COM1 (J2)	31
COM2 (J1)	32
Multidrop Communications Network	32
Accessing COM1 And COM2 Buffers	34
Serial Port File Numbers	35
COM1 And COM2 Port Status	36
Changing The Baud Rate	36
Commands	37

CHAPTER 7: DIGITAL I/O LINES 39

Introduction	39
Organization of Connectors and Ports	39
Input and Output Ports	40
Example: J3 Digital I/O	41
Outo-Module Rack Interface	41
Interfacing To Switches And Other Devices	42
Commands	43

CHAPTER 8: ANALOG INPUT 45

Introduction	45
AIN Function	45
Mode Options	45
Single-ended Mode	46
Differential Mode	46
Pseudo-differential Mode	46

CHAPTER 9: DISPLAY PORT	47
Introduction	47
Connecting A Display To J6	47
Connecting A Display TO 5600 Digital I/O CARD	47
Programming Example	48
Program Explanation	48
Commands	49
CHAPTER 10: KEYPAD PORT	51
Introduction	51
Connecting The Keypad To J5	52
Connecting The Keypad TO J3 or a 5600 I/O Expansion Card	52
Programming Example	53
Program Explanation	53
Commands	54
CHAPTER 11: HIGH CURRENT PORTS	55
Introduction	55
Considerations For High Current Outputs	56
CHAPTER 12: WATCHDOG TIMER	57
CHAPTER 13: CALENDAR/CLOCK	59
DS-1216EM SmartWatch	59
Commands	60
CHAPTER 14: INTERRUPT HANDLING	61
Description	61
Interrupt Generation	61
Programming Example	62
Program Explanation	62
Commands	62

APPENDIX A: COMPONENTS 63

Custom Communication Cable 63

Power Module 63

Using Other Serial Communications Software 64

APPENDIX B: TECHNICAL INFORMATION..... 67

Specifications 67

Connector Pinouts 70

WARRANTY

PREFACE

This manual provides all the information required to install, configure and operate the 5081 Microcontroller.

By using this manual, you will be able to:

- Interface the 5081 to your IBM-compatible PC and the Micro PC Expansion Cards.
- Set up communications between the 5081 and a terminal or PC.
- Gain an understanding of the operation of 5081 hardware using CAMBASIC IV programming software.

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 5081 uses a 64180 processor. Additional information on this component can be obtained from Hitachi (hardware manual #U77, software manual #U92, phone: 408-435-8300) or Zilog (technical manual #03-8276-01, phone: 408-370-8016).

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 - 29422
```

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. Names of other sections or manuals are also italicized.

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 5081.
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	Transferring a program or data from a PC to the RAM on the 5081; 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	Any personal computer with terminal emulation software, such as an IBM PC with PC SmartLINK.
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, Refers to all versions of PC SmartLINK.
Reset	Resetting the 5081 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	Transferring a program or data from the 5081 to a PC.

TECHNICAL SUPPORT

If you have a question about the 5081 card and can't find the answer in this manual, call Technical Support. They will be ready to give you the support you need.

When you call please have the following at hand:

- Your *5081 Microcontroller User's Manual*.
- A description of your problem

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

DESCRIPTION

The Micro PC 5081 Microcontroller is a 4.5 in. x 4.9 in. computer board 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 and your IBM-compatible PC.

The 5081 uses one slot of the Micro PC card cage. All communication between the 5081 and your PC is done through the serial ports. Figure 1-1 shows the basic system configuration. With this configuration, you can:

- Write the program using the PC as a “Smart” terminal;
- Download the program to the 5081;
- Edit and alter the program on the 5081;
- 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 5081 card. You do not need any other software to run programs on the 5081. You will need communications software, such as PC SmartLINK, to link your PC to the 5081 card.

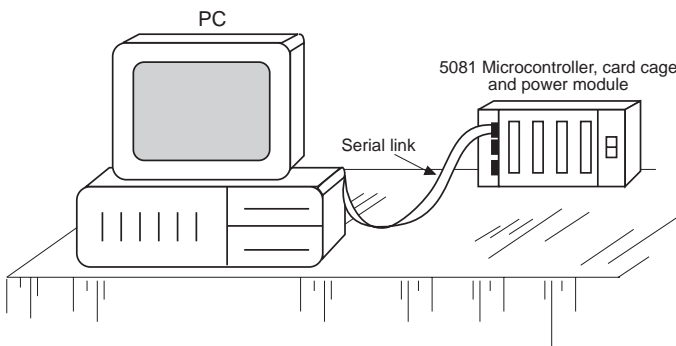


Figure 1-1—Basic System Configuration

MAJOR FEATURES

Resident CAMBASIC IV Software

The 5081 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.

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 5081 will match baud rates of 300, 1200, 2400, 4800 and 9600.

Serial Ports

Two programmable RS-232 serial ports are standard on the 5081. The baud rate, parity, length, and number of stop bits are software programmable for both ports.

Keypad Port

4 x 4 keypads can be used with the 5081 card. The keypad is automatically scanned and is read by using the KEYPAD command. Pressing a key generates a software interrupt. This port can also be used as a general purpose digital I/O port if a keypad is not connected.

High Current Output Port

The 5081 has eight high current sinking outputs for driving relays, lamps, small solenoids, etc. This port can also be used as a general purpose digital I/O port if the output driver chip is replaced with a

DIP shunt. The BIT and OUT statements are used to switch the outputs off and on.

Display Port

The display port can be used to connect liquid crystal displays (LCD) from two lines by 20 characters to four lines by 20 characters. This port can also be used as a general purpose digital I/O port. You send data to this port with the DISPLAY command.

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.

Opto Rack Interface

The 5081 has 24 digital I/O lines which are logic level compatible and can also be used to drive Opto 22 style opto modules. The BIT command is used to control opto modules.

Watchdog Timer

The watchdog timer is a fail-safe against program crashes or processor lockups. It times out every 1.2 sec. The timer is enabled and disabled by software commands.

Analog Inputs

There are four analog input channels. Inputs can be either single-ended or differential. The card has provisions to increase the 0-5V input range on two channels to a maximum of 32V.

This page intentionally left blank.

The 5081 is designed to be ready to program from the time you plug it in. Installation is simple and straightforward. The 5081 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 5081 card and before inserting or removing cables.
2. Disconnect power before removing or inserting the 5081 card.
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 5081.

5081 Microcontroller
Micro PC cardcage
5100 power module, or equivalent
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 5081 CARD

Before installing the 5081 card, refer to Figure 2-1 for the location of various connectors and jumpers.

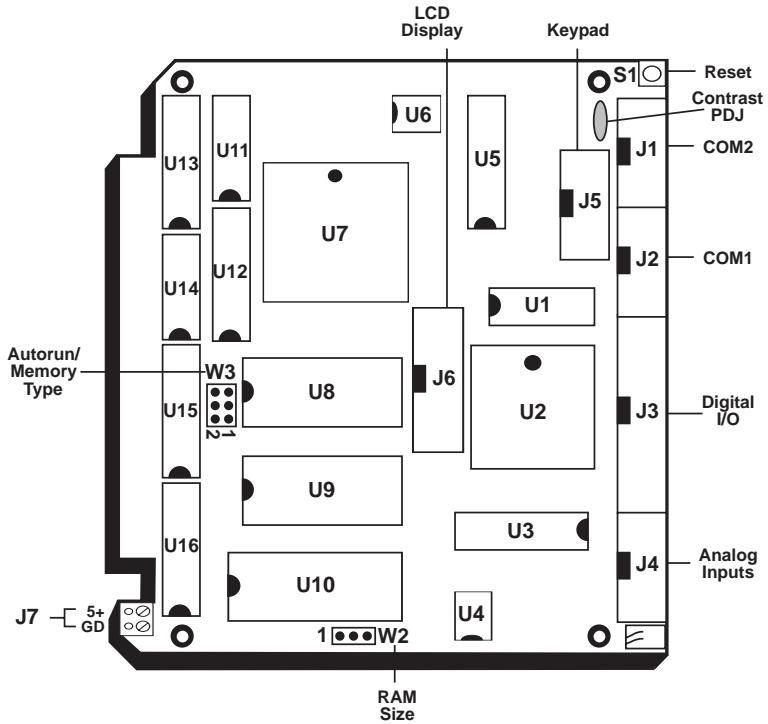


Figure 2-1—5081 Component Diagram

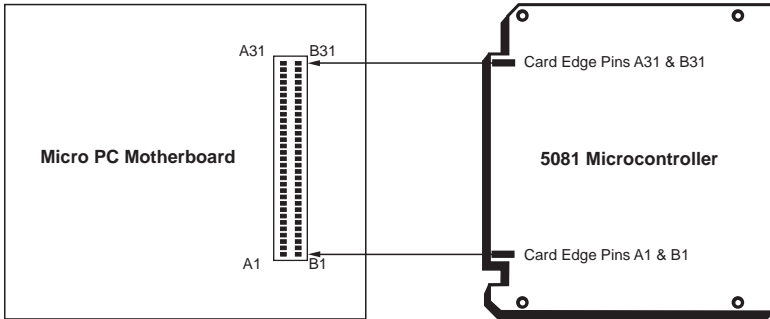


Figure 2-2—Edge Connector Orientation

To install the 5081 in the card cage:

1. Turn card cage power off.
2. Slide the 5081 into the card cage. It will fit only one way—components face left.
3. Seat the 5081 firmly into the backplane socket.
4. The serial cable (VTC-9F) for the 5081 has a 10-pin IDC connector on the 5081 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 (J2) on the 5081. (Please refer to Creating a Custom Communications Cable in Appendix A if you are making your own serial cable.)
5. Plug the other end of the serial cable into COM1 of your PC using a gender changer. 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.)

-
6. Copy PC SmartLINK onto your hard disk and type

```
A:SL <RETURN>
```

7. 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.

8. Press the <ESC> key.
9. 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  
V4.xx - all rights reserved  
Bytes free - 29422
```

10. If you don't get the proper logon message, check the serial parameters of your terminal to make sure they are set correctly. Parameters should be 9600 baud, 8 data bits, no parity, and 1 stop bit; you can also use baud rates of 300, 1200, 2400 or 4800. If the system still does not respond, refer to the Troubleshooting section at the end of this chapter.
11. 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

Once you have completed your application program on your PC, you can download the files to the 5081. You can also upload files from the 5081 to your PC for further editing and debugging. Please refer to Appendix A if you are using communications software other than PC SmartLINK.

Uploading Your Program

In the previous section, you created a test program on the 5081. To upload that program and save it to the disk on your PC:

1. Start PC SmartLINK on your PC. You are now ready to establish a communications link between the 5081 and your PC.
2. Refer to your PC SmartLINK manual for specific steps on uploading files from RAM on the 5081 to your PC using the XMODEM protocol.
3. To save the program to disk:
Enter the *filename* of your program.

Downloading Your Program

1. Start PC SmartLINK on your PC.
2. Type
NEW <RETURN>

-
3. SmartLINK has a buffer which is used to temporarily store the program. Download the program from the PC disk to the buffer. Refer to the PC SmartLINK manual for specific information on downloading files using the XMODEM protocol.
 4. Enter your *filename*.
 5. Download the file to the 5081
 6. List the program by typing

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. Check the power module; make sure the power cord is properly connected between the power module and the power source.
2. Check the serial parameters on your PC. The default setting should be:

9600 baud, no parity, 8 data bits, 1 stop bit
3. Make sure the serial cable is properly connected between the 5081 and your PC or terminal.
4. Make sure the serial cable is working properly by performing a point-to-point check on the connectors; and make sure you are using a gender changer. Refer to Appendix B for corresponding connector, signal, and pin numbers.
5. Make sure the 5081 is receiving power. The power module voltage should be in the $5 \pm 0.25V$ range when measured at the connector pin, B3. The power module ripple should be less than 50 mV.
6. 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 5081. To verify that the 5081 is communicating with your PC:
 - a. Connect an oscilloscope to J2, pin 3. Press the reset switch on the 5081. A burst of signal activity that switches at least between +3V and -3V verifies that the 5081 is sending a signal.
 - b. If you see the logon message but nothing else happens, connect the oscilloscope to U5, pin 12. 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 J2, pin 5 on the 5081 for a signal change of at least $\pm 3V$.

If you cannot get a signal at J3, pin 5, check your computer and make sure it is transmitting.

For communications packages other than PC SmartLINK, please note:

- The 5081 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 5081 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.

INTRODUCTION

The 5081 Microcontroller is shipped with 32K RAM and 32K EEPROM in sockets U10 and U8, respectively. This chapter describes these memory chips and how to add memory to the 5081 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 and 32K EPROM or 32K EEPROM in your 5081 card. The RAM chip you select is installed in socket U10. The EPROM or EEPROM is installed in the user socket, U8.

RAM is used to store data, variables, recipes, and formulas. It provides volatile memory storage unless you are using a battery backup (the DS-1213D 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 EEPROM is used to store programs. EEPROMs are nonvolatile but have a limited number of write cycles (about 10,000). 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 U10, and run from system RAM.

ADDING MEMORY

Installing RAM Chips

The 5081 is shipped with 32K RAM installed in socket U10. The card is designed to accept 128K and 512K RAM chips. To add a battery backup module or to change the memory chip, you need to remove the original chip, insert the optional battery backup module and memory chip.

To install a RAM chip:

1. Turn off power to the 5081.
2. Remove any memory chip from socket U10.
3. Install the RAM in socket U10.
4. For 32K and 128K, jumper W2 [2–3]. For 512K, jumper W2 [1–2].

Installing a Battery Backup Module

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

The DS–1213 is a 32–pin DIP sockets with built–in power fail circuitry and a dual battery system. The DS–1213D provides battery backup for a low power 32K and 128K CMOS RAMs. The DS–1213D is installed in socket U10; the RAM is inserted into the top of the SmartSocket. A DS–1213DM is required for the 512K RAM.

To install the SmartSocket:

1. Turn off power to the 5081.
2. Remove any memory chip from socket U10.
3. Install the DS–1213D or DS–1213DM into socket U10 with the index mark pointing towards the 5081 gold contact fingers.
4. Install the low power CMOS RAM into the top of the module.

Installing EPROM and EEPROM Chips

The 5081 accepts a 32K EEPROM and 32K EPROM. It is shipped with a 32K EEPROM.

To change the memory chip, you need to remove the original chip, install the new memory, and change the jumper settings in jumper block W2.

WARNING:

Improper setting of W2 or W3 jumper may result in temporary inoperation.

To install an EPROM or EEPROM chip:

1. Turn off power to the 5081.
2. Remove any memory chip from socket U8.
3. Install the new EEPROM or EPROM in socket U8.
4. Make the appropriate jumper connections in jumper block W3:

5081 Memory Select: W3	
Pins Jumpered	Memory Size
[3-4][5-6]	32K 29C256 EEPROM
[3-4][5-6]	32K 27C256 EPROM

5. If you are installing an EEPROM for the first time, you must format it first using the CONFIG SSD statement. Make sure jumper [1-2] is in place on W3.

CONFIG SSD *type*

5081 Program Size	
Program Size	Type
32K (1 file)	0
32K (4 files)	1

This page intentionally left blank.

This chapter describes how to save information to the RAM installed in socket U10.

In general, data (including formulas, recipes, and variables) are saved to and retrieved from RAM. The data can be arrays, PEEK and POKE numbers, or strings (using PEEK\$ and POKES). This memory is volatile; it does not retain the data when power is turned off unless used with a battery backup.

Programs are stored in the EPROM and EEPROM in socket U8, not in RAM. You can also transfer a program from your PC to RAM (downloading) or from RAM to your PC (uploading). Refer to Chapter 2 for information on transferring a program between your PC and the 5081. Refer to Chapter 5 if you want to download a program from or save a program to the EEPROM. Maximum program size is 32K.

Programs and CAMBASIC IV variables reside in segment 0; all user data is saved to segment 1 (see Memory Map in Figure 4-1). On power-up, the program in segment 8 (the EEPROM) is loaded into segment 0.

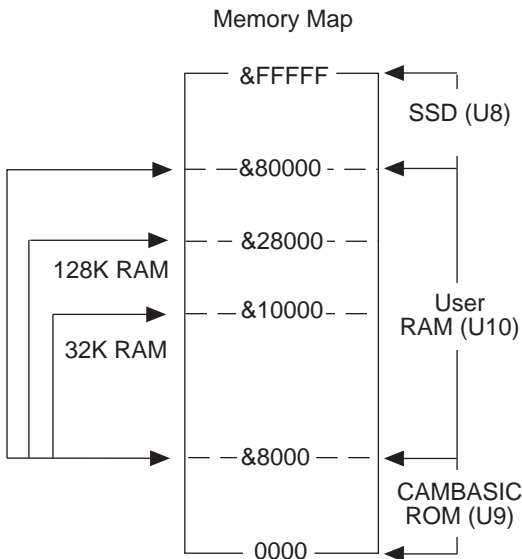


Figure 4-1—Memory Map Diagram

You can use the PEEK and POKE commands to store and retrieve the values. For example,

```
20 POKE 0,A,1
```

pokes the value of A into segment 1, address zero.

Later in your program, use the PEEK statement to retrieve the variable:

```
50 A=PEEK (0,1)
```

You can store and retrieve arrays, strings, and PEEK and POKE bytes this way.

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. A byte is any number from 0 to 255. Words and floating point numbers can also be saved using DPOKE (2 bytes) and POKE! (4 bytes).

```
50 A=PEEK(200,1)
```

 retrieves a byte value. Words and floating point numbers can also be retrieved by using DPEEK and PEEK!

Saving and Retrieving a String

100 POKE \$300,A\$,1	pokes string variable AS 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.

5081 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

This page intentionally left blank.

INTRODUCTION

Programs are stored in the EEPROM, which is installed in socket U8. You can store one program with a maximum size of 32K, or four 8K programs, if you are using a 32K EEPROM. If you are using an 8K EEPROM, you can store one 8K program. Refer to the CONFIG SSD command in the *CAMBASIC IV Programming Guide* for storage options.

The EEPROM is nonvolatile (retains memory even when power is disconnected), has an unlimited number of read cycles and a limited number (about 10,000) of write cycles. Programs are not run from EEPROM. Instead, the program is downloaded to RAM in socket U10. 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.

SAVING A PROGRAM IN EEPROM

Before a program can be saved in EEPROM, it must first be configured using the CONFIG SSD command. This command is only executed 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 and programs cannot be recovered

To save a program to EEPROM:

1. You need to specify a file name, and
2. Jumper block W3 must be configured so that autorun is ON (it is shipped in this configuration).

If you plan to autorun your program later, the first four letters of the first file name on the SSD must be "BOOT". The rest of the file name can be anything you want it to be, up to twelve characters total. If the first four letters of the first file name on the SSD are not "BOOT", the system will not execute your program on power up.

To store your program, enter

```
SAVE "filename"
```

The file name can have up to 12 characters. Your program is in system RAM until you type SAVE.

Programming progress is displayed on the screen:

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

```
>__
```

Though compiling typically takes about 5 seconds, it can take as long as 15 seconds with a very long or complex program. The prompt (>) appears when the program has been successfully saved to the EEPROM. If the program does not write to EEPROM, an error message appears:

```
Failed @ XXXX>
```

```
>__
```

XXXX is a 4 byte address.

If you are using a 32K EEPROM, you can store up to four programs of 8K each, or one program up to 32K in length. To store subsequent programs, use the SAVE command again. Each program will be stored to the next block in the order that you save them. For example,

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

saves "BOOT" in the first block; secondprogram in the second block; and thirdprogram in the third block. If the autorun jumper is in place, the program "BOOT" will autoexecute on power up.

LOADING PROGRAMS TO RAM

Enter

```
FILES
```

to see the file names of the programs stored in your EEPROM. Information stored in RAM does not have file names.

To load a program from EEPROM into RAM, enter

```
LOAD "filename"
```

DELETING A PROGRAM FROM EEPROM

To delete a program from the EEPROM, enter

```
DEL "filename"
```

To see the file names of remaining programs enter

```
FILES
```

AUTOMATIC PROGRAM EXECUTION ON POWER-UP

Your 5081 Control Card 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 autoexec file (the first file on the disk) any arbitrary name, then `RENAME` it to "BOOT" when it comes to run time.

To autorun a program:

1. Configure jumper block W3 [1-2] for autorun ON (this is its default).
2. Enter

```
CONFIG SSD type
```

NOTE: This command is only executed once when first configuring the EEPROM.

3. `SAVE` your program to the EEPROM. The first four letters of the filename must be `BOOT` (see previous section).
4. On power-up or reset, the program "BOOT....." will automatically download to RAM and execute.

Autorunning Object Code Programs

CAMBASIC IV checks only the first file for an object code program. If that fails, it then checks the beginning of the SSD for an autoexecute object code file.

Your object code file must have a four byte header that starts at an offset of 40H from the beginning of the EEPROM. The first two bytes must be zero. The next two bytes are the length bytes with the LSB first. The length tells the system the number of bytes to move. Thus, this number must be equal to or greater than the actual code length. Your program starts at an offset of 44H into the EEPROM.

If the system finds an object code file, it will load the file into RAM and execute it. The system turns off all maskable interrupts before moving the file. This gives you a chance to change the interrupt vectors before reenabling the interrupts.

Preventing Autorun

When troubleshooting a program, it may not be convenient for an autoexecute file to run. This is especially true for those that have been configured to ignore the break <ESC> key.

To prevent autorun:

1. Remove the jumper at W3 [1-2].
2. Power up the system.
3. Replace the jumper at W3 [1-2].
4. Use the RENAME command to rename "BOOT....." to any name not beginning with BOOT.

The configuration of jumper block W3 controls whether the EEPROM is enabled (autorun is ON) or disabled (autorun is OFF):

ENABLED is jumpered (W3 [1-2]).
DISABLED is not jumpered.

LINKING MULTIPLE PROGRAMS

You can store up to four 8K programs in an EEPROM. Use the RUN command to link programs. For example, you can use the RUN command in your "BOOT" program to call any of the other three programs. You can also use the RUN command in each program to call the next program.

As each subsequent program is downloaded to RAM, it completely replaces the previous program. All scalars, arrays and strings are erased, except for protected variables A%–Z%.

COMMANDS

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

5081 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

INTRODUCTION

The 5081 has two serial ports that can be used for interfacing to a printer, terminal or other serial device. COM1 is used for program development. During run time, it can be used in the same manner as COM2. It can be used with printers, displays, or other devices that do not require handshaking during run time. COM2 can be used to interface the 5081 to a multidrop network or to equipment that requires hardware handshaking. Both ports can receive the XON/XOFF characters, so that a remote device can tell the 5081 to start and stop data transmission.

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 using a gender changer.

Default parameters for both ports are 9600 baud, 8 data bits, no parity, and 1 stop bit. When you press the <ESC> on power-up, the 5081 will automatically adjust to baud rates of 300, 1200, 2400, 4800 and 9600. Any of these parameters can be changed with the CONFIG BAUD command. Refer to the *CAMBASIC IV Programming Guide* for further information.

COM1 (J2)

COM1 is located at J2. Use a VTC-9F cable to connect external serial devices to the port. This port does not use hardware handshake lines but does support the receive XON/XOFF protocol. Under this protocol, the external device controls the flow of characters from the 5081 to the device.

CAMBASIC IV has a 256 character output buffer that is interrupt driven and sends characters out the serial port without slowing down program execution. For example, when you execute any of the PRINT statements, characters are sent to a buffer and transmitted (printed) in the background while CAMBASIC continues program execution. If the buffer fills, program execution stops until there is room in the buffer.

COM1 is normally used in programming the 5081, but during run time it may be used as a general purpose serial port. When used

for programming or with the INPUT statement, it 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. Its buffer size is 256 characters. If the buffer becomes full, all subsequent characters are discarded.

COM2 (J1)

COM2 is located at J1. Use a VTC-9F cable to connect this port for general purpose serial data transfer or when the 5081 is part of a multidrop communications network (see next section). You cannot download or program through this port.

COM2 is identical to COM1 except that COM2 also has two hardware handshaking lines, CTS and RTS. In the example below, line 100 returns the status of the CTS line:

```
100 B=BIT(194,5)
```

Here, line 400 sets the RTS line low; line 500 sets the RTS line high.

```
400 BIT 192, 4, 1
500 BIT 192, 4, 0
```

MULTIDROP COMMUNICATIONS NETWORK

You can use your 5081 card with a multidrop communication network by connecting COM2 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 5081 cards) 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 Converter Product Sheet*.

NOTE: The Octagon BBS (303-427-5368) has additional information on multidrop.

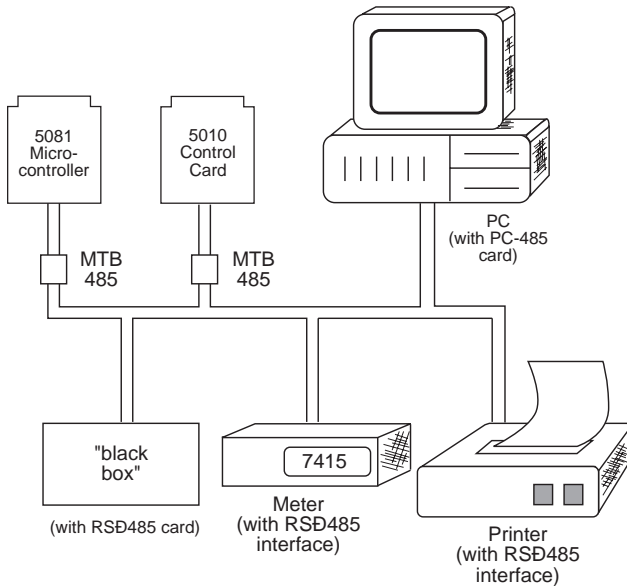


Figure 6-1—Multidrop Communications Network

ACCESSING COM1 AND COM2 BUFFERS

You can access the COM1 and COM2 buffers in three ways. You may need to consult the *CAMBASIC IV Programming Guide* for more information.

1. **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.

2. **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.

3. **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.

SERIAL PORT FILE NUMBERS

CAMBASIC IV references the serial I/O ports by file numbers:

5080 Serial Port File Numbers			
Serial Port	File #	Example	
COM1 (J3)	1	PRINT "HELLO" INPUT A\$	Prints "hello" to COM1 Returns input from COM1
		or Prints#1, "hello" INPUT#1,A\$	Prints "hello" to COM1 Returns input from COM1
COM2 (J6)	2	PRINT#2,"hello" INPUT#2,A\$	Prints "hello" to COM2 Returns input from COM2

COM1 AND COM2 PORT STATUS

Use the `SYS(10)` and `SYS(11)` commands to check the status of the buffers. 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.

CHANGING THE BAUD RATE

Use `CONFIG BAUD` to change the baud rate of COM1 and COM2 serial ports. The autobaud feature sets both serial ports to the same baud rate on power-up. You may need to use a different baud rate for one of the ports. For example, if you connect a printer which uses a baud rate of 1200 to COM2, and your system is set to 9600, you must change the baud rate of COM2 before you can use your printer. Refer to the *CAMBASIC IV Programming Guide* for further information.

COMMANDS

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

5081 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\$	Jumps to subroutine on serial string
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.

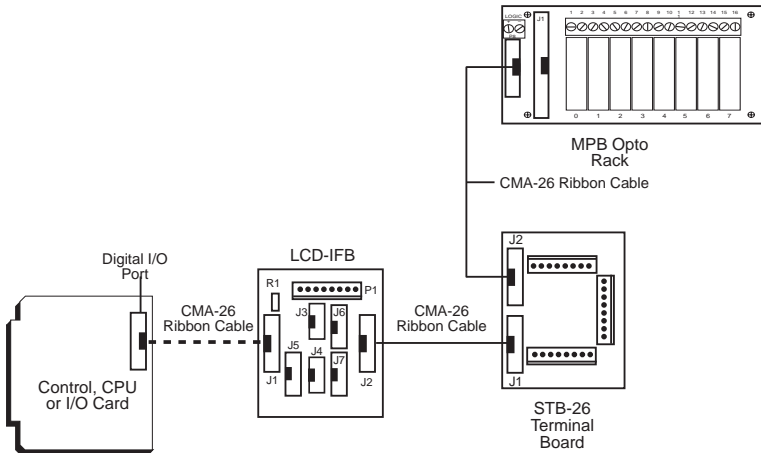


Figure 7-1—Typical Opto Rack Configuration

WARNING:

Apply power to the 5081 card before applying an input voltage to the digital I/O lines. This prevents excessive currents from flowing and damaging input devices. If you cannot apply power to the 5081 first, contact the Technical Support Department for suggestions appropriate to your application.

ORGANIZATION OF CONNECTORS AND PORTS

The digital I/O lines on the 5081 are supplied by a 82C55 chip (high current version). The 82C55 is located in socket U2, which is connected to J3.

J6 and J5 have been designed for use as display and keypad ports, respectively. These specific uses are described in Chapters 9, 10,

and 11. The keypad and display ports have lines in common with the digital I/O.

Input and Output Ports

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

The 82C55 has three ports with eight parallel I/O lines (bits) per port. Each port has a unique I/O address. Port A and Port B can be programmed as all inputs or all outputs. Port B is also a high current port. Refer to Chapter 11, *High Current Port*, for more information. 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 12 mA at 1.0V.

The keypad and display ports share lines with the digital I/O port.

5081 Digital I/O Port Configuration			
Connector	Port	I/O Address	Description
J3	A	00	8 lines which can be programmed as all inputs or all outputs
J3	B	01	8 lines which can be programmed as all inputs or all outputs or 8 high current outputs.
J3	C	02	8 lines that can be programmed as one group of 8 lines or two groups of 4 lines as all inputs or all outputs.
J5	Keypad	02	Used for keypad. Connected to Port C above.
J5	LCD	00	Used for display. Connected to Port A above.

Example: J3 Digital I/O

```
10 'Connect an 8-, 16-, or 24-position opto
    rack to J3
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
```

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

OPTO-MODULE RACK INTERFACE

You can interface digital I/O lines from J3 to an 8-, 16-, or 24-position opto-module rack. One end of the CMA-26 cable plugs into J1 and the other plugs into an MPB-8S, MPB-16S, or MPB-24S opto module rack.

You can also use a STB-26 terminal cable to connect J3 on the 5081 to a terminal board and then to the opto module rack. The STB-26 has two 26-pin connectors, one of which plugs into J3; the other plugs into the opto module 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:

5081 Opto Rack Interface		
Opto Channels	82C55 Port	I/O Address
0-3	Lower C	2
4-7	Upper C	2
8-15	A	1
16-23	B	0

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 5081 Control Card. All 82C55 lines have 10K pull-up resistors. I/O lines at connector 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 the input status as follows:

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

State of the input is returned in the variable "D" as a zero (connected to ground) or a one (open).

See connector pinouts in Appendix B for port and pin number information.

COMMANDS

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

5081 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

There are four analog input channels with an input range of 0–5V. The resolution is 8 bits (about 0.4% of full scale). The input range can be extended to 32V with the addition of resistors. The maximum input on the 5V range is 16V. While this level will not damage the 5081, the inputs may give erroneous readings above 5V.

NOTE: For the beta units, please contact the factory, if input voltages greater than 5V are to be used.

AIN FUNCTION

This function is used in CAMBASIC to read the analog input. You do not use CONFIG AIN for the 5081. The syntax is

```
10 B=AIN! (channel)
    or
10 B=AIN! (channel, mode, scaler)
```

Notice that the AIN is followed by an "!". The first syntax will return the value at the specified channel. The channel number ranges from 0–3. The result will range from 0–255.

The second syntax is used when differential mode or a scaling factor is required. The mode determines whether the input is single-ended, pseudo-differential or differential. The scaler is the number that you want to multiply the result by. For example, if you have a pressure gauge with a 0–50 PSI rating, the scaler would be:

$$\text{Scaler} = 50/255 = 0.196$$

Mode Options

There are three mode options: single-ended, differential and pseudo-differential.

In 8-bit systems differential and pseudo-differential are rarely needed. Indeed, these modes can cause problems if used improperly. The “mode” parameter is 0 for single-ended, 1 for differential, and 2 for pseudo-differential.

Single-ended Mode

In the syntax case:

```
10 D=AIN! (channel)
```

single-ended mode is assumed. The signal is applied to the appropriate channel, and the signal ground is applied to the common ground (even numbered pins in J4).

Differential Mode

In the differential mode two adjacent channels have differential inputs. There are two combinations, channels 0 and 1, and channels 2 and 3. In this case neither side of the signal is connected to the common ground.

Pseudo-differential Mode

This mode reduces the total number of channels to three. The fourth channel becomes the differential signal ground for channels 0-2. Thus, none of the signal inputs are connected to the common ground.

INTRODUCTION

You can use an LCD series display with your 5081 Microcontroller. Display sizes range from two lines by 20 characters to four lines by 20 characters. Only LCD displays are supported with the DISPLAY command.

Setting up a display with your Micro PC system can give the operator feedback on operation status. When used in conjunction with a keypad, it can provide input feedback.

CONNECTING A DISPLAY TO J6

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

To connect your display to J6: Connect the cable included with your display to J6.

Execute the CONFIG PIO and CONFIG DISPLAY commands to set up the port address and display type.

CONNECTING A DISPLAY TO 5600 DIGITAL I/O CARD

You can also connect your display to the optional 5600-48 Digital I/O Card. This is especially useful if you are using a 17-key or larger keypad, as well as a display. The display and the keypad both connect to the IFB, which connects to the 5600. For more information, please refer to the *5600 I/O Expansion Card User's Manual*.

1. Connect the cable included with your display to the display and the IFB.
2. If you are using a keypad, connect the cable included with your keypad to the keypad and the IFB.
3. Connect a CMA-26 cable from the IFB to the port on the 5600 that you want to use.

-
4. Line 10 of the programming example (at the end of the chapter) determines which port is connected to your display. When you use a port on the 5600, the command should be:

```
CONFIG DISPLAY X,4,0
```

The 5600 has four 26-pin connectors. All four are identical, and all have three ports: A, B, and C. X is the address of port A of the connector you're using on the 5600. (Refer to the *5600 I/O Expansion Card User's Manual* for information on calculating addresses.)

5. Add the following line to the program example to configure the PIO:

```
7 CONFIG PIO x,0,0,y,y,y      x is the ad-
                               dress of port
                               A; y initial-
                               izes values
                               for port B and
                               C
```

PROGRAMMING EXAMPLE

The example below is for a 2-line by 40-character LCD display. Notice that all DISPLAY statements end with a semicolon so that a carriage return/line feed will not be sent.

```
10 CONFIG DISPLAY 0,3,0
20 A$="PRESENTING THE NEW OCTAGON"
30 C$="80 Character LCD display"
40 DISPLAY (0,5) A$;
50 DISPLAY (1,6) C$;
60 DELAY 2
70 DISPLAY (0,0) CHR$(32,80);
80 DELAY .5
90 GOTO 40
```

Program Explanation

Line 10	specifies which multi-line display you want to use with the display statement
Lines 20, 30	define the two strings to be printed
Lines 40	displays A\$ at row 0, column 5

Line 50 displays C\$ at row 1, column 6
Line 60 has a 2 second delay
Line 70 erase both lines by writing a string of 80
 spaces
Line 80 is a 0.5 second delay and the program
 repeats

COMMANDS

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

5081 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.

This page intentionally left blank.

INTRODUCTION

You can use 4 x 4 (16 key) and larger keypads with the 5081 card. A 4 x 4 keypad connects directly to J5 on the 5081 via the included cable. Larger keypads must be connected to an interface board (IFB); the IFB is then connected to either J3 on the 5081 or to a port on the 5600 I/O Expansion Card. If you want to use your 4 x 4 keypad with J3 or the 5600 expansion card, you must also use an IFB (see Figure 9-1).

There are two types of IFBs: LCD and DP (DP is used with vacuum fluorescent displays). If you are using a display, select the IFB appropriate for your display. If you are not using a display, you can use either IFB.

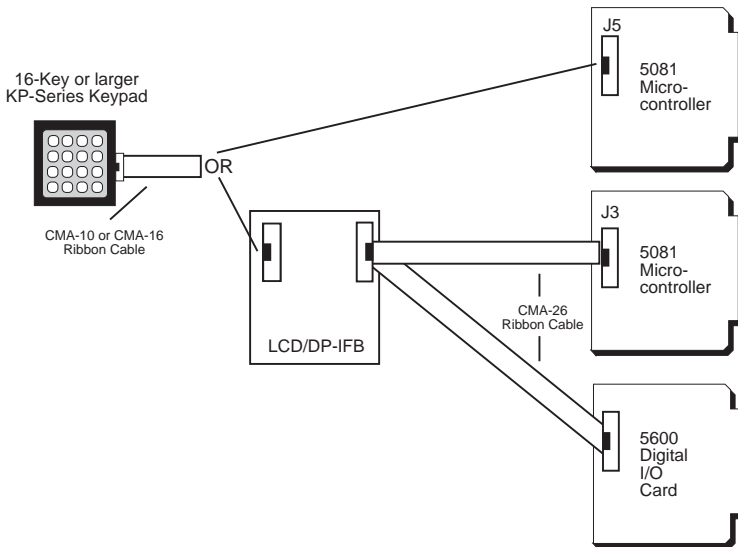


Figure 9-1—Keypad Configurations

CONNECTING THE KEYPAD TO J5

The most common configuration uses J5, 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 J5 on the 5081 Control Card.
3. You are now ready to initialize the keypad (see example at the end of this chapter).

CONNECTING THE KEYPAD TO J3 OR A 5600 I/O EXPANSION CARD

If you have a 17-key or larger keypad, or if you want to reserve connector J5 for another use, you can connect your keypad to J3 on the 5081 card or to an optional 5600 I/O Expansion Card. Either configuration requires an IFB.

1. Connect your keypad cable (included with your keypad) from the keypad to the IFB. If you're using a 4 x 4 keypad, use the 10-pin connection; for larger keypads, use the 16-pin connection.
2. Connect the IFB to J3 on the 5081 card or to the selected port on the 5600 I/O Expansion Card.
3. You are now ready to initialize the keypad. Use the CONFIG PIO command:

```
CONFIG PIO X,1,Y,0,1,1
```

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).

4. In the programming example at the end of this chapter, use the following in line 10:

```
CONFIG KEYPAD$ 8,64,Z
```

The value 64 indicates that you are using a 17–key or larger keypad. Z is the address of the port and the connector you're using. For J3, Z is zero. If you are using the 5600 I/O Expansion Card, the value for Z depends upon the address of the port and connector you're using. Please refer to the *5600 I/O Expansion Card User's Manual* for information on determining your 5600 port address.

PROGRAMMING EXAMPLE

The following example shows how to use each of the functions and commands described above. Your application may not need all of the commands. When the program is run, each character is printed to the screen as it is entered from the keypad.

```
5  CONFIG PIO &0,1,0,0,1,0
10 CONFIG KEYPAD$ 8,16,2
20 'optionally change a keypad value.
30 'change the "B" key to the letter "M"
   (ASCII 77)
40 POKE SYS(12) + 7,77
80 ON KEYPAD$ GOSUB 500
90 'Dummy loop for this example
100 GOTO 100
500 A$=KEYPAD$(0)
510 IF A$="C" THEN B$=" "
520 IF A$="#" THEN ..enter
530 PRINT A$;
540 B$=B$+A$
550 RETURN
560 ..enter
570 FL=1
600 RETURN
```

Program Explanation

10	sets debounce time 80 ms; keypad address = 0; keypad size is 16
20–30	comment lines
40	get address, add offset
80	set up interrupt
100	dummy loop for this example
500	get keypad value

-
- 510 clear buffer if "C" hit
 - 540 build string
 - 570 the variable FL is a flag read by the main program to determine if a keypad string has been entered. The flag should be cleared for subsequent tests. BS must be cleared before subsequent characters are entered.

COMMANDS

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

5081 Keypad Commands	
Command	Function
CONFIG KEYPADS	Sets keypad parameters
KEYPADS(0)	Returns the last key from the keypad port
ON KEYPADS	Causes a program branch when a key is pressed on keypad
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 eight I/O lines at J3 on the 5081 card. 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, U1, with a DIP shunt jumper and use the lines as general purpose digital I/O lines. The lines are TTL compatible with the DIP shunt in place. The address of the high current port is 01. The following table lists the connector pinouts for the high current port:

5081 Digital I/O: J3			
I/O Line	Port A	Port B (high current)	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			

On power-up, all high current driver inputs are pulled LOW. This forces all high current outputs OFF. A "1" or "ON" written to the high current port causes the output to switch HIGH 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. Refer to Figure 10-1.

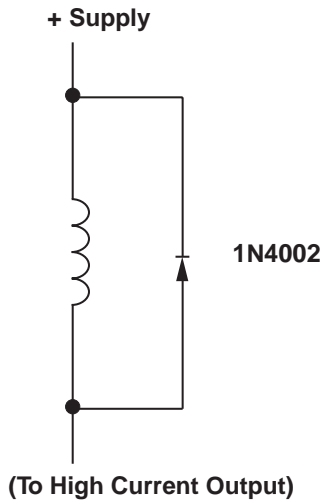


Figure 10-1—Inductive Load Protection Circuitry

- 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 5081 and can cause erratic operation.

- Lines at J3 may be turned ON and OFF using the BIT statement.

DESCRIPTION

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. The timeout is 1.2 sec.

5081 Watchdog Timer	
Task	Command
To enable	OUT &60,0
To disable	OUT &60,1
To reset	A=BIT(&60,0)

WATCHDOG LATCH

The watchdog latch allows you to add a buzzer or light in series with a button which can be reset should the watchdog timeout. The latch is available through jumper block W4, pins 1, 2 and 3:

5081 Watchdog Latch: W4	
Pin #	Description
1	Gnd
2	Output
3	Gnd

The following diagram shows a typical circuit for the watchdog latch. the LED could be replaced by a relay for driving devices

that require high 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.

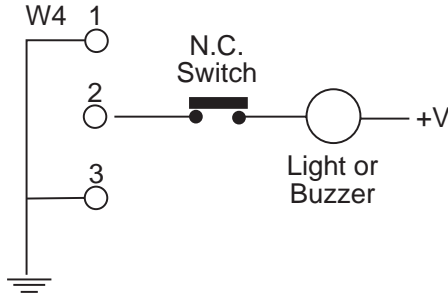


Figure 12-1—Typical Watchdog Latch Circuit

DS-1216EM SMARTWATCH

The Dallas DS-1216EM is a 28-pin DIP socket which contains a calendar/clock and a dual battery system with a life of 5 -10 years. Accuracy of the clock is about 1 minute/month at 25° C. Install the SmartWatch in user socket U8, then install your EEPROM in the SmartWatch.

To install the DS-1216EM SmartWatch

1. Remove any memory chip from socket U8.
2. Install the DS-1216EM SmartWatch into socket U8 with the index mark pointing towards the 5081 gold contact fingers.
3. Install the memory chip into the top of the SmartWatch.
4. The clock 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. 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.

5081 Calendar/Clock Commands	
Command	Function
DATES	Reads/sets date from calendar/clock
TIMES	Reads/sets timer for calendar/clock

DESCRIPTION

The 5081 can be mounted in a passive backplane for use with external interrupt devices. You can use a switch closure, magnetic pick up, or other devices capable of switching between ground and +5V for external interrupt generation.

5081 Interrupt Input Sources				
64180 Designation	CAMBASIC Interrupt #	Activ Level	Bus Name	Edge Conn Pins
INT0	2	High	IRQ4	B24
INT1	3	High	IRQ3	B25
INT2	4	High	Comes from A/D converter	

INTERRUPT GENERATION

An interrupt is generated when a logic signal level on an interrupt input goes high. If you include an ON ITR *n* command in your program, it will branch to a subroutine when the interrupt is generated. The average latency time in CAMBASIC IV is less than 0.5 mS. The current statement must finish executing before the interrupt subroutine executes. However, the interrupt itself need only be high for 10 μ S.

PROGRAMMING EXAMPLE

The following programming example shows how to set up the interrupt and service routine. When the IRQ4 line goes high, the interrupt service routine is executed and the message "Door Open" is printed.

```
20 ON ITR 2 GOSUB 500
.
.           your program continues
.
500 'interrupt routine here
510 PRINT "Door Open"
520 DR = 1
530 RETURN ITR 2
```

Program Explanation

```
20   Sets up interrupt for line 500
520   Sets flag to signal door is open
520   Returns from interrupt routine to normal program
      flow
```

COMMANDS

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

5081 Interrupt Commands	
Command	Function
ON ITR	Enables a program branch on an interrupt
RETURN ITR	Re-enables an interrupt and returns program control

This appendix contains information which may be helpful to you if you are using non-Octagon components with your 5081 Control Card. For additional information, please refer to the user manuals supplied with your equipment.

CUSTOM COMMUNICATION CABLE

The 5081 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 5081.

5081 Custom Communication Cable				
J3	Signal	Function	DB-25	DB-9
9	Gnd	Common	7	5
3	TxD	Transmitted data	3	3
5	RxD	Received data	2	2
10	+5	Power		

POWER MODULE

The 5081 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 $5 \pm 0.25V$ 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 $5V \pm 0.25V$ range.

To use your power module with the 5081 Control Card:

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 more than 18 inches long and be 18 gauge or larger.

Please note that with some expansion cards (for example, the 5700 12-bit Analog Input Card), you will need a power module that also supplies ± 12 volts. 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 packaged 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 successful, they cannot be 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. If your communications program does not look for a pacing prompt and you do not use an adequate delay, you will begin to lose characters when the program gets too big to compile quickly.

- To upload a program (transfer it from the 5081 RAM to your PC), your PC needs to receive a file. Refer to your communications software documentation for details.
- The 5081 Control Card 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 enable your communications program to receive or transmit a file. To save a program, type “LIST” before receiving a file. After a <RETURN> is sent, the listing will follow.

-
- COM1 does not use a CTS line to prevent the host from sending data. You may need to tie this line to RTS or a plus voltage to enable CTS on you system.
 - The serial port COM1 (J2) does NOT use handshaking or control lines. The 5081 uses only three lines (TxD, RxD, and ground) to send and receive data between it and the host. These are the only signals available.
 - Put your serial communications software in “transmit mode” to download a program to the 5081.

SPECIFICATIONS

- CPU
- 64180 or Z80180 (Z80 code compatible), 9.216 MHz

Memory

- 32K, CAMBASIC IV ROM
- 32K, 128K or 512K Static RAM
- 32K nonvolatile memory (EEPROM or EPROM)

Digital I/O

The 5081 has 24 digital I/O lines using a 82C55 chip (high current version). The I/O lines are terminated in three connectors for a variety of uses. All lines have 10K pull-up resistors. You can use an MPB-8, -16, or -24 opto-module rack by connecting the rack to J2 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.

Analog Inputs (J4)

There are four analog input channels. Each has a 0-5 input range with a resolution of 8 bits. The inputs are protected against damage up to 16 VDC. Through a software command, the inputs can also be differential. The precision reference does not require calibration.

High Current Outputs (J3)

Eight of the digital I/O lines are capable of driving relays and other loads rated to 50V and 100 mA. A dip shunt jumper is provided to convert high current outputs to TTL logic levels. I/O address is 01.

Keypad Input (J5)

Port C, connector J5 may be used with a 4 x 4 keypad. It will also accept 4 switch contacts. Debouncing and scanning are done with software commands. Uses a 10-pin IDC connector. Use Octagon

KP-1, KP-2-16, or KP-3 keypad and cable. I/O address is 02. The keypad port uses eight of the lines that terminate at J2.

LCD Display Port (J6)

This port accepts LCD displays only. Port A of 8255 I/O address is 0.

Serial Ports (J1 and J2)

Two RS-232C serial ports. COM1 (J2) supports the RxD and TxD signals. COM2 (J4) supports the RxD and TxD signals as well as the RTS and CTS handshake lines. Refer to the *CAMBASIC IV Programming Guide* and the CONFIG BAUD command for more information.

EEPROM Programmer

Programs 8K or 32K EEPROMs. Speed for all EEPROMs is 200 nS or faster.

5081 EEPROMs		
Vendor	Size	Order #
Octagon 29C256	32K	2527
Atmel	32K	29C256-20PC

Pushbutton Reset

The pushbutton reset switch will reboot the system.

Power Requirements

5V \pm 5% @120 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.

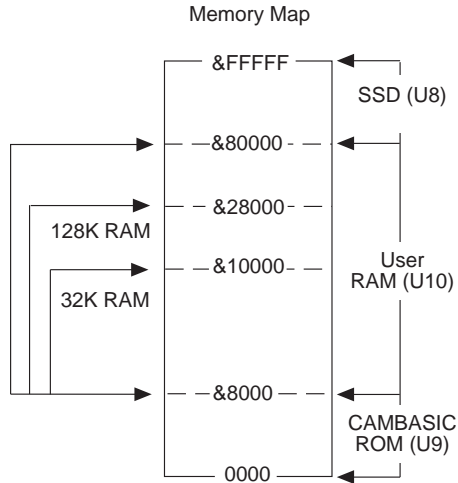
Temperature Range

Operating: -25° to 70° C.
Storage: -50° to 90° C
Operating humidity: 5% to 95% RH noncondensing.

Size

4.5 in. X 4.9 in.
(11.4 X 12.4 cm)

Memory Map



I/O Map

5081 I/O Map	
Description	Address
J2 Digital I/O	00-03
J6 Display	0
J5 Keypad	2
J4 Analog	40-5F
Watchdog Timer	60-7F
Not used	80-BF
64180 Internal	C0-FF
Expansion	100-3FF

CONNECTOR PINOUTS

5081 COM2: J1	
Function	Pin #
+5V	10
Gnd	9
TxD (Out)	3
RxD (In)	5
RTS (In)	6
CTS (Out)	4

5081 COM1: J2	
Function	Pin #
+5V	10
Gnd	9
TxD (Out)	3
RxD (In)	5

5081 Digital I/O: J3			
I/O Line	Port A	Port B (high current)	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			

5081 Analog Port: J4	
Function	Pin #
IN 0	1
Gnd	2
IN 1	3
Gnd	4
IN 2	5
Gnd	6
IN 3	7
Gnd	8
NC	9
Gnd	10

5081 Keypad Port: J5	
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

5081 Display Port: J6	
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

EDGE CONNECTOR 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	DACK1*	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.
