

M68SPGMR11UM/D

MAY 1995

**M68SPGMR11**  
**SERIAL PROGRAMMER**  
**USER'S MANUAL**

© MOTOROLA Inc., 1994, 1995; All Rights Reserved

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function, or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

Motorola and the Motorola logo are registered trademarks of Motorola Inc.

Motorola Inc. is an Equal Opportunity/Affirmative Action Employer.

PROG11 software is © P & E Microcomputer Systems, Inc.\*, 1990, 1993; All Rights Reserved. Portions of the software are © Borland International, 1987. Portions of the software are © TurboPower Software, 1988.

\* P & E Microcomputer Systems, Inc.,  
PO Box 2044  
Woburn, MA 01888-2044  
(617)-353-9206

---

---

## CONTENTS

### CHAPTER 1 INTRODUCTION

|     |                                   |     |
|-----|-----------------------------------|-----|
| 1.1 | FEATURES AND SPECIFICATIONS ..... | 1-3 |
| 1.2 | USER REQUIREMENTS.....            | 1-4 |
| 1.3 | OVERVIEW.....                     | 1-4 |

### CHAPTER 2 HARDWARE PREPARATION

|     |                                      |     |
|-----|--------------------------------------|-----|
| 2.1 | MAKING STANDALONE CONNECTIONS.....   | 2-1 |
| 2.2 | MAKING MMDS11 CONNECTIONS.....       | 2-3 |
| 2.3 | INSTALLING PROGRAMMING ADAPTERS..... | 2-4 |

### CHAPTER 3 PROGRAMMING

|     |                                      |      |
|-----|--------------------------------------|------|
| 3.1 | PREPARING TO PROGRAM .....           | 3-1  |
| 3.2 | STARTING PROGRAMMING.....            | 3-2  |
| 3.3 | PROG11 SCREEN.....                   | 3-4  |
| 3.4 | PROGRAMMING COMMANDS.....            | 3-6  |
| 3.5 | TYPICAL PROGRAMMING SEQUENCE.....    | 3-13 |
| 3.6 | PROGRAMMING THE CONFIG REGISTER..... | 3-14 |

### CHAPTER 4 SUPPORT INFORMATION

|     |                            |     |
|-----|----------------------------|-----|
| 4.1 | CABLE CONNECTORS.....      | 4-1 |
| 4.2 | TROUBLESHOOTING GUIDE..... | 4-2 |
| 4.3 | ZIF SOCKET .....           | 4-3 |

## INDEX

## FIGURES

|      |   |     |
|------|---|-----|
| 1-1. | SPGMR11 Programmer .....                          | 1-1 |
| 2-1. | Standalone Connection (Top View).....             | 2-1 |
| 2-2. | MMDS11 Connection (Top View) .....                | 2-3 |
| 2-3. | SPGMR11 Programming Adapter .....                 | 2-4 |
| 2-4. | Programming Adapter Installation (Side View)..... | 2-5 |
| 3-1. | SPGMR11 Programming Screen .....                  | 3-5 |

## TABLES

|      |  |     |
|------|--|-----|
| 1-1. | SPGMR11 Programming Adapters.....                  | 1-2 |
| 1-2. | SPGMR11 Specifications.....                        | 1-3 |
| 4-1. | HOST, EMULATOR Connector Signal Descriptions ..... | 4-1 |
| 4-2. | SPGMR11 Troubleshooting.....                       | 4-2 |

## CHAPTER 1

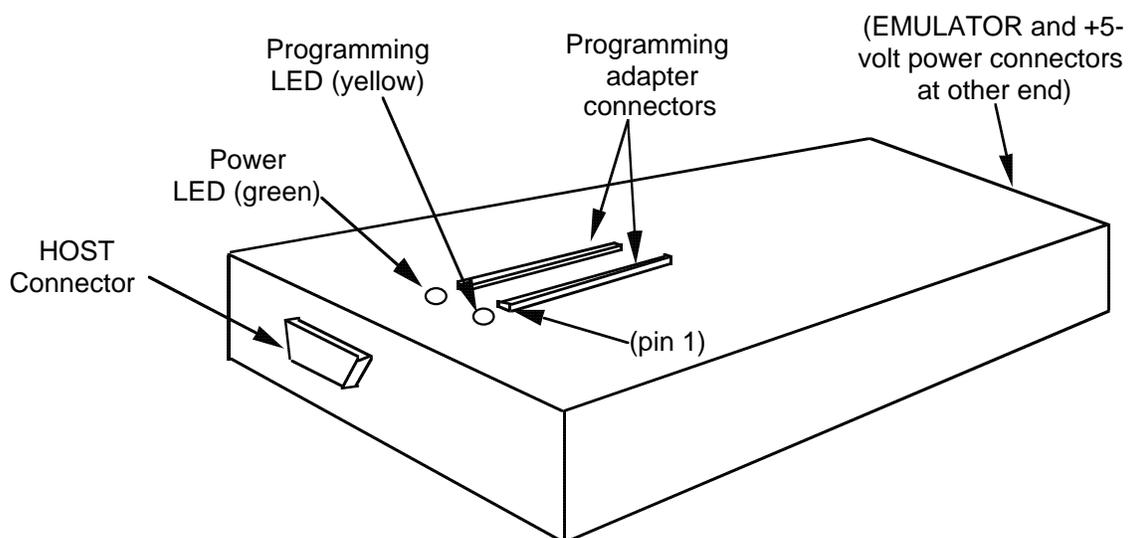
### INTRODUCTION

The M68SPGMR11 Serial Programmer (SPGMR11) lets you program your code into OTPROM, EPROM, and EEPROM microcontroller (MCU) devices of the MC68HC11 family. The SPGMR11 also can generate an S-record file from data programmed into such an MCU, and can program the CONFIG register of EEPROM-based MCUs.

The SPGMR11 is an optional peripheral for your personal computer or for an HC11 Motorola Modular Development System (MMDS11). To use the SPGMR11 as a standalone programmer, you connect it to an IBM-compatible computer and to a 5-volt power supply. To use the SPGMR11 as an MMDS11 peripheral, you connect it between the MMDS11 and the host computer.

To use the SPGMR11, you need a separately-purchased programming adapter. A programming adapter has the programming socket for one or more types of HC11 MCU; the programming adapter plugs into the top of the SPGMR11. The variety of programming adapters available means that you can use the SPGMR11 to program virtually any HC11 MCU.

Figure 1-1 shows the SPGMR11 without any programming adapter. Table 1-1 lists the programming adapters available, or scheduled to be available soon, as this manual went to print.



**Figure 1-1. SPGMR11 Programmer**

**Table 1-1. SPGMR11 Programming Adapters**

| <b>Adapter</b>  | <b>Package</b>     | <b>Programs MCUs</b>   |
|---|--------------------|--|
| M68PA11A8P48  | 48-pin DIP         | MC68HC11A0 <sup>(1)</sup> , -A1, -A8, -811E2                                       |
| M68PA11D3FB44   | 44-pin QFP         | MC68HC11D3 <sup>(2)</sup> , -711D3   |
| M68PA11D3FN44   | 44-pin PLCC, CLCC  | MC68HC11D3 <sup>(2)</sup> , -711D3   |
| M68PA11D3P40  | 40-pin DIP, CerDIP | MC68HC11D3 <sup>(2)</sup> , -711D3   |
| M68PA11E20FN52  | 52-pin PLCC, CLCC  | MC68HC11A0 <sup>(1)</sup> , -A1, -A8, -E0, -E1, -E9, -E20, -711E9, -711E20, -811E2 |
| M68PA11E20FU64  | 64-pin QFP         | MC68HC11A0 <sup>(1)</sup> , -A1, -A8, -E0, -E1, -E9, -E20, -711E9, -711E20         |
| M68PA11E20PB52  | 52-pin TQFP        | MC68HC11E1, -E9, -E20  |
| M68PA11F1FN68   | 68-pin PLCC        | MC68HC11F1   |
| M68PA11F1PU80   | 80-pin QFP         | MC68HC11F1   |
| M68PA11G5FN84   | 84-pin PLCC, CLCC  | MC68HC11G5 <sup>(2)</sup> , -G7 <sup>(2)</sup> , -711G5                            |
| M68PA11K4FN84   | 84-pin PLCC, CLCC  | MC68HC11K4, -P2, -711K4, -711P2  |
| M68PA11K4FU80   | 80-pin QFP         | MC68HC11K4, -M2 <sup>(2)</sup> , -711K4, -711M2                                    |
| M68PA11KA4FN68  | 68-pin PLCC, CLCC  | MC68HC11KA4, -711KA4   |
| M68PA11L6FN68   | 68-pin PLCC, CLCC  | MC68HC11L6, -711L6   |
| M68PA11L6FU64   | 64-pin QFP         | MC68HC11L6, -711L6   |
| M68PA11PA8FU64  | 64-pin QFP         | MC68HC11PA8, -711PA8   |
| M68PA11PH8FN84  | 84-pin PLCC, CLCC  | MC68HC11PH8, -711PH8   |
| M68PA11PH8PV112   | 112-pin QFP        | MC68HC11PH8, -711PH8   |
| (1) Programs the CONFIG register.   |                    |  |
| (2) This MCU does not have programmable ranges; the SPGMR11 gives read access to the CONFIG register and ROM. |                    |  |

PROG11 programming software, on a 3.5-inch diskette, comes with the SPGMR11. The SPGMR11 includes a programming voltage circuit, eliminating the need for an external 12-volt supply.

The SPGMR11 comes with a 2-lead cable for 5-volt SPGMR11 operating power. (An MMDS11 or external 5-volt power supply is the source for this power.) The SPGMR11 also comes with a 9- to 25-pin communications adapter, which lets you connect the separately purchased 9-lead cable to a 25-pin computer serial communication port.

## 1.1 FEATURES AND SPECIFICATIONS

SPGMR11 features include:

- An economical means of programming OTPROM, EPROM, and EEPROM MC68HC11 MCUs.
- Standalone hookup to your computer or hookup between a development system and your computer.
- An on-board programming voltage circuit, eliminating the need for an external programming voltage supply.
- 19,200 baud communication with the computer, for faster programming time.
- Compatibility with a variety of separately-purchased programming adapters. (Each programming adapter has a programming socket for an MCU type or package. To expand your programming capability, you need only purchase another programming adapter.)

Table 1-2 gives SPGMR11 specifications.

**Table 1-2. SPGMR11 Specifications**

| <b>Characteristics</b>                    | <b>Specifications</b>                    |
|---|--|
| Internal clock                            | 4.9152 MHz oscillator                    |
| Communications ports                      | RS-232 compatible, 19200 baud            |
| Power requirements                        | 5 volts (4.9 to 5.5 Vdc), @ 200 mA (max) |
| Temperature<br>Operating<br>Storage       | 0° to 50° C<br>-40° to +85° C            |
| Relative humidity                         | 0 to 90% (non-condensing)                |
| SPGMR11 dimensions                        | 6.6 x 3.5 x 1.0 in. (168 x 89 x 25 mm)   |
| SPGMR11 weight                            | 7.4 oz. (210 g)                          |
| Adapter board dimensions<br>(approximate) | 2.0 x 2.0 x 1.0 in. (51 x 51 x 25 mm)    |
| Adapter board weight<br>(approximate)     | 2.0 oz. (57 g)                           |

---

---

## 1.2 USER REQUIREMENTS

To use the SPGMR11, you must provide:

- An IBM PC or compatible computer, with serial communications port and 3.5-inch diskette drive. The computer must be at least 286-based, and must have at least 640 Kbytes of memory.
- Programming adapters, separately purchased: one adapter for each MCU type and package to be programmed.
- An external 5-volt power supply that produces 4.9 to 5.5 Vdc, at a maximum of 200 mA. (The external supply is not required if you will use the SPGMR11 with an MMDS11; the MMDS11 provides 5-volt power for the SPGMR11.)
- A separately purchased 9-lead cable, for connection to the serial communication port of your computer.

### NOTE

You must use 9-lead communication cables between the SPGMR11 and the computer, and (optionally) between the SPGMR and an MMDS11. Three-lead connections (RXD, TXD, and GROUND) will not work correctly.

## 1.3 OVERVIEW

The SPGMR11 lets you program OTPROM, EPROM, or EEPROM MCUs of many different types. You connect the SPGMR11 to your host computer or between your host computer and your development system. Programming software, which resides in the computer, is inactive except during MCU programming.

You must provide +5-volt power to the SPGMR11. If you connect the SPGMR11 to an MMDS11 development system, use MMDS11 connector P6 as the source of 5-volt power. Otherwise, use any regulated +5-volt external power source.

To program an MCU, you must install the appropriate programming adapter on the SPGMR11. The only differences between programming adapters are the size and pin arrangement of the programming sockets. Before changing programming adapters, make sure that the yellow programming LED is not lit.

The remaining chapters of this manual explain the general steps for programming your MCU:

- Chapter 2 explains how to connect the SPGMR11 and how to install the correct programming adapter.
- Chapter 3 explains how to load SPGMR11 software into your host computer, how to install the MCU to be programmed into the programming socket, and how to use the programming software commands.
- Chapter 4 explains connector pin assignments, signal descriptions, and other support information.



## CHAPTER 2

### HARDWARE PREPARATION

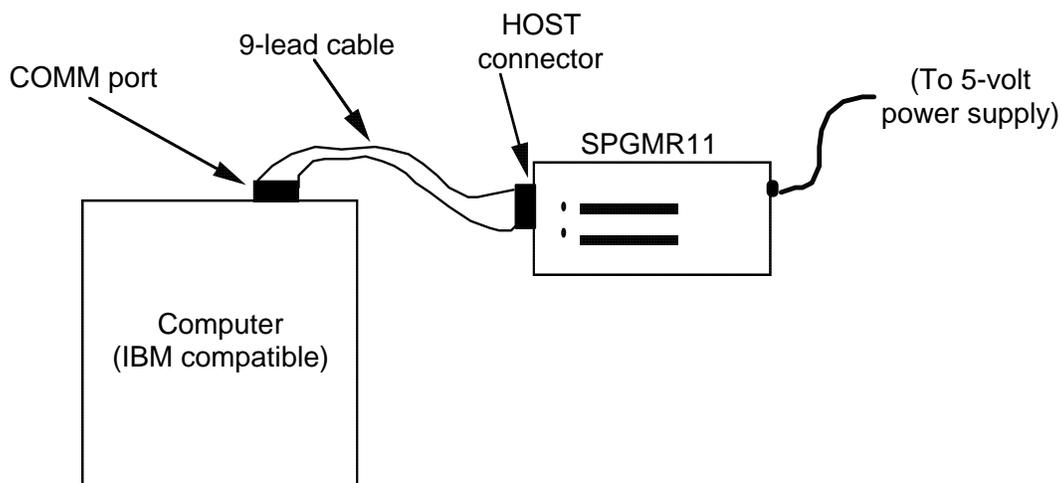
This chapter explains hardware preparation for your SPGMR11 programmer. Paragraph 2.1 explains cable connections for standalone use. Paragraph 2.2 explains cable connections for use with an MMDS11. Paragraph 2.3 explains how to install a programming adapter.

#### ELECTROSTATIC DISCHARGE PRECAUTION

Ordinary amounts of static electricity from your clothing or work environment can damage or degrade electronic devices and equipment. For example, the electronic components installed on your printed circuit board are extremely sensitive to electrostatic discharge. You should wear a grounding wrist strap whenever you handle any printed circuit board. This strap provides a conductive path for safely discharging static electricity to ground.

### 2.1 MAKING STANDALONE CONNECTIONS

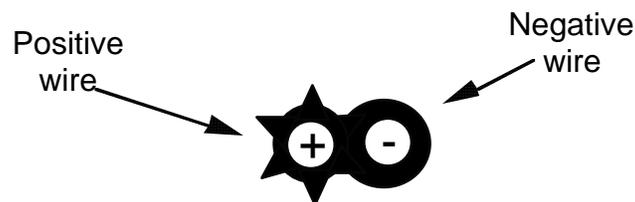
Figure 2-1 represents SPGMR11 cable connection for standalone use. (The figure shows the SPGMR11 without a programming adapter; you may connect these cables whether a programming adapter is installed or not.)



**Figure 2-1. Standalone Connection (Top View)**

Follow steps 1 through 6 to make standalone connections:

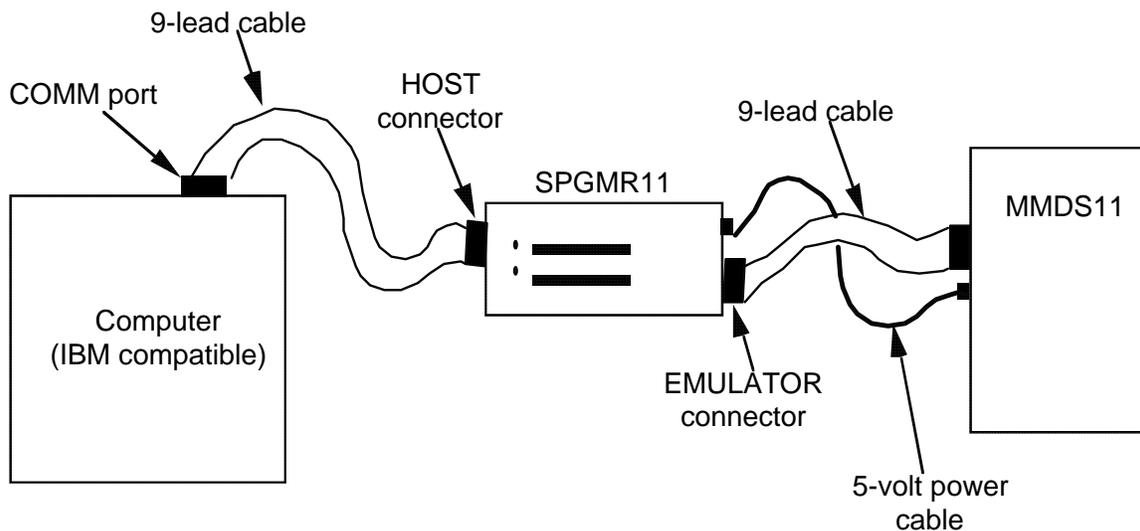
1. Turn OFF the computer power switch. Turn OFF your 5-volt power supply.
2. Connect a 9-lead serial cable (separately purchased) between the SPGMR11 HOST connector and the COM1 or COM2 serial port of your computer.
  - a. If necessary, use the 25-to-9-pin adapter at the COM1 or COM2 port end of the cable.
  - b. The COM1 port is the default port for SPGMR11 software. You can specify either COM port when you run the software.
3. Compare the output connector of your 5-volt power supply with the 5V connector of the SPGMR11. Note that the center pin of the 5V connector is the + contact; the ring of the 5V connector is the – contact. If the two connectors match, connect the 5-volt power cable between them. **Do not turn on 5-volt power until you are ready to program.** Advance to Step 5.
4. If the output connector of your power supply does *not* match the 5V connector of the SPGMR11, you may modify the 5-volt power cable:
  - a. Cut off *one* cable connector, then adapt this end of the cable for your power supply connector. Note that the positive wire of the cable has ribs, but the negative wire is smooth, per this cross-section diagram:



- b. Connect the modified end of the 5-volt power cable to the power supply. Verify that the center pin of the 5V connector is the + contact and the ring of the 5V connector is the – contact. Connect the unmodified end of the cable to the SPGMR11 5V connector. **Do not turn on 5-volt power until you are ready to program.**
5. Install a programming adapter on the SPGMR11, per paragraph 2.3.
6. System connections are complete; you may turn on power for the computer and the SPGMR11. For programming instructions, turn to Chapter 3.

## 2.2 MAKING MMDS11 CONNECTIONS

Figure 2-2 represents SPGMR11 cable connections for use with an MMDS11 system, as well as a computer. (The figure shows the SPGMR11 without a programming adapter; you may connect these cables whether a programming adapter is installed or not.)



**Figure 2-2. MMDS11 Connection (Top View)**

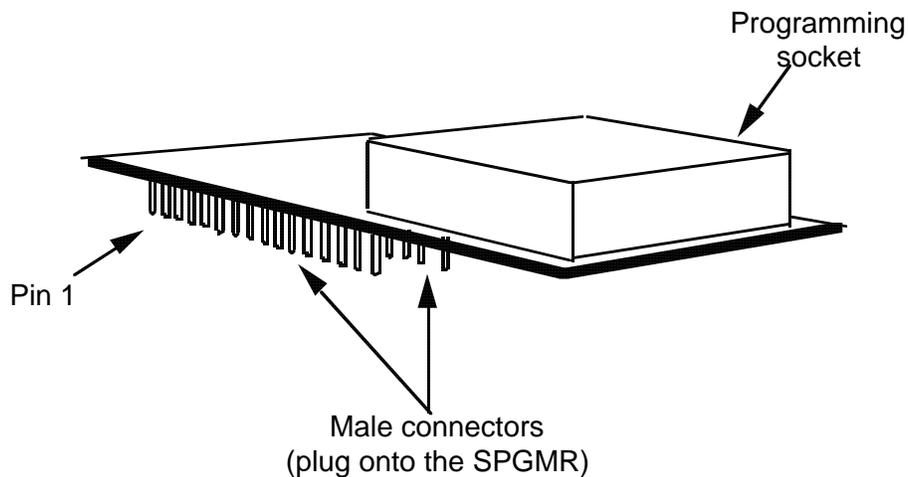
Follow steps 1 through 6 to make MMDS11 connections:

1. Turn OFF the computer power switch. Turn OFF the MMDS11 power switch.
2. Connect a 9-lead serial cable (separately purchased) between the SPGMR11 HOST connector and the COM1 or COM2 serial port of your computer.
  - a. If necessary, use the 25-to-9-pin adapter at the COM1 or COM2 port end of the cable.
  - b. The COM1 port is the default port for SPGMR11 software. You can specify either COM port when you run the software.
3. Another 9-lead cable comes with the MMDS11. Connect this cable between the SPGMR11 EMULATOR connector and the 9-pin serial port of the MMDS11 station module.

4. Connect the 5-volt power cable between the SPGMR11 power connector and the 5-volt-out connector (P6) of the MMDS11 station module. **Do not turn on power until you are ready to program.**
5. Install a programming adapter on the SPGMR11, per paragraph 2.3.
6. System connections are complete; you may turn on power for the computer, the SPGMR11, and the MMDS11. For programming instructions, turn to Chapter 3.

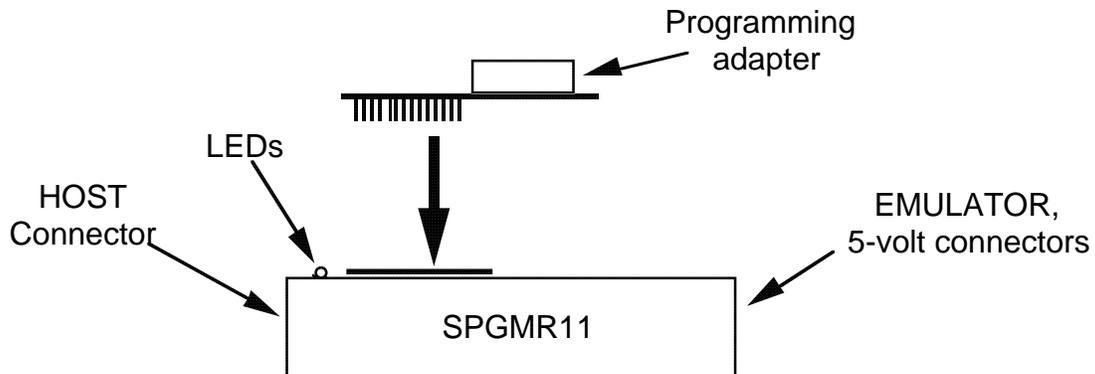
## 2.3 INSTALLING PROGRAMMING ADAPTERS

Figure 2-3 represents an SPGMR11 programming adapter. Adapters are identical functionally, but accommodate different microcontroller units (MCUs) or different MCU package configurations. Table 1-1 lists programming adapters; your Motorola representative can explain which is correct for your programming application.



**Figure 2-3. SPGMR11 Programming Adapter**

To install a programming adapter, plug it onto the top of the SPGMR11: the pins on the bottom of the programming adapter fit into the connectors on the top of the SPGMR11. Be sure to match the pin 1 connection: orient the programming adapter so that the programming socket is as far as possible from the SPGMR11 LEDs. Figure 2-4 is a side view of programming adapter installation.



**Figure 2-4. Programming Adapter Installation (Side View)**

#### **NOTE**

The connectors on the top of the SPGMR11 are rated for at least 50 insertion cycles. If you anticipate changing programming adapters often, use a zero insertion force (ZIF) socket. Chapter 4 gives more information about a ZIF socket.



## CHAPTER 3

### PROGRAMMING

This chapter explains how to load programming software and how to program MCUs.

#### 3.1 PREPARING TO PROGRAM

Before you begin programming, you must load the PROG11 programming software into your host computer. To do so:

1. Insert the software diskette into the 3.5-inch drive of your computer.
2. Type the drive letter and a colon, followed by the word `install`. For example, if your 3.5-inch drive is drive B, type: `b:install`
3. The install program automatically loads the software. Follow the instructions that appear on the screen. This completes software loading.

As the SPGMR11 downloads S-records to the MCU device, the code you program into the MCU must be in Motorola S-record format. To create S-records, use a 6811 assembler that outputs S-records. You may store S-records anywhere on your computer hard drive.

As long as you do not turn on power to the SPGMR11 or run the PROG11 software, your computer functions as if the SPGMR11 were not present. (Similarly, the SPGMR11 has no effect on an MMDS11 until you begin programming; MMDS11 activity stops during SPGMR11 programming.)

#### NOTE

To select commands or other choices, click the mouse button once, type the command, or use the arrow keys. This highlights your selection. To activate your selection, click the mouse button a second time or press RETURN.

## 3.2 STARTING PROGRAMMING

When you are ready to program MCUs, follow steps 1 through 11:

1. Make sure that 5-volt power to the SPGMR11 is OFF.
2. Double check that the correct programming adapter is installed on the SPGMR11.
3. Insert the MCU to be programmed into the programming socket, according to the MCU package, matching MCU pin 1 with pin 1 of the socket. (The silk-screen marking of the programming adapter identifies pin 1 of the socket.)
  - Insert a PLCC- or CLCC-package MCU with the pins *up*. MCU pin 1 is centered above the part number or centered on the edge to the right of the chamfered corner. A dimple may identify pin 1.
  - Insert a QFP-package MCU with the pins *down*. The largest corner dimple identifies pin 1.
  - Insert a DIP- or CerDIP-package MCU with the pins *down*. MCU pin 1 is at the left corner of the U-cut edge. (That is, if the MCU's U-cut edge points north, pin 1 is at the northwest corner.)
4. Turn ON 5-volt power to the SPGMR11. The green power LED lights.
5. Bring up the DOS prompt on the computer. To start the PROG11 software, enter the command

```
PROG11 [2] [/filename.11P]
```

The optional parameter **2** specifies the COM2 serial port of the computer; without this parameter value, the system uses the default COM1 port.

An optional **filename.11P** parameter identifies the MCU type to be programmed. If you enter this parameter value (this filename must be in the same directory as the PROG11.EXE file), these messages appear: `Resetting the HC11 program...` and `Loading boot code to HC11...`. The yellow programming LED lights. The spinning symbol on the computer screen indicates that downloading is in progress. Advance to Step 7.

6. If you did not enter the **filename.11P** parameter in Step 5, the MCU select window appears. Select the MCU type that matches the MCU device in the programming socket. The messages, `Resetting the HC11 program...` and `Loading boot code to HC11...` appear, and the yellow programming LED lights. The spinning symbol on the computer screen indicates that downloading is in progress.

7. The main PROG11 screen appears. (Paragraph 3.3 explains this screen.)
8. To program your MCU, select appropriate commands from the main screen. (Paragraph 3.4 explains each command.)
9. To program another MCU of the same type, select the X (reset MCU) command. The yellow LED goes out, and the prompt . . . change MCU if necessary . . . appears. Remove the current MCU from the programming socket, insert the new MCU into the socket, press RETURN, and return to Step 8.
10. To program an MCU of a different type, select the Q (quit) command to stop PROG11 software. The yellow LED goes out, and the prompt . . . Remove the MCU now . . . appears. Remove the MCU and press RETURN; the DOS prompt replaces the PROG11 screen. Return to Step 2.
11. To quit programming, select the Q (quit) command to stop PROG11 software. The yellow LED goes out, and the prompt . . . Remove the MCU now . . . appears. Remove the MCU and press RETURN; the DOS prompt replaces the PROG11 screen. Turn OFF 5-volt power to the SPGMR11. The green power LED goes out, and the computer returns to normal operation (as does an MMDS11).

#### NOTE

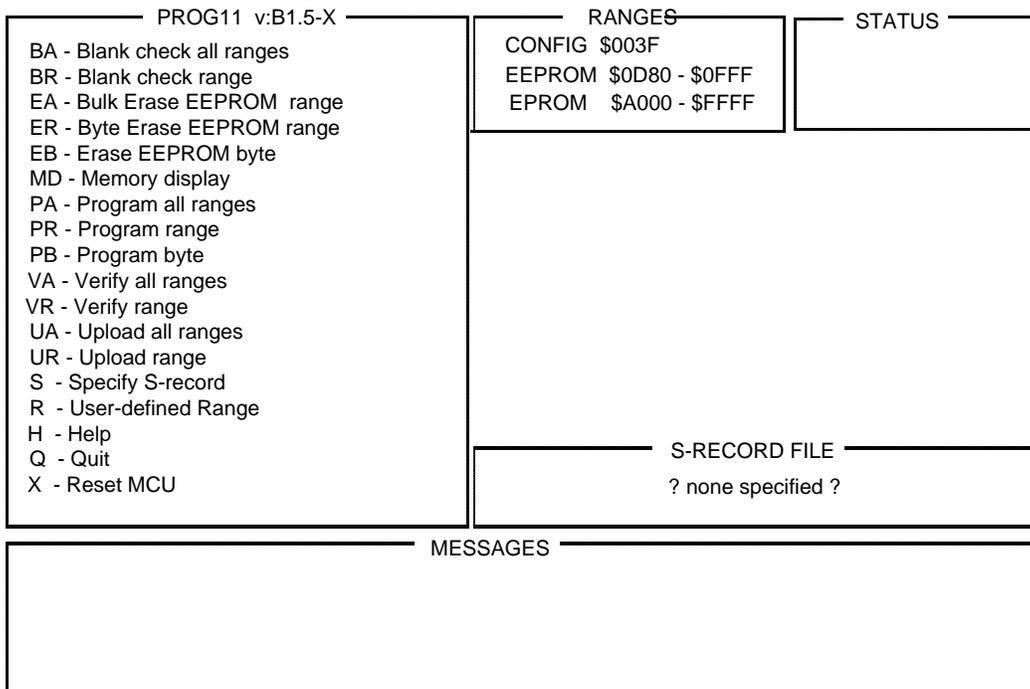
An alternative to starting PROG11 from DOS is to start it as a programmer option from within the RAPID integrated development environment. RAPID comes with MMDS11 software. (The RAPID user's manual gives instructions.)

Starting PROG11 from within RAPID by-passes Steps 5 and 6, above, automatically specifying the S-record file that corresponds to the source file being edited. This advances you to Step 8, ready to enter a program command.

### 3.3 PROG11 SCREEN

Figure 3-1 shows the PROG11 screen, which consists of five windows:

- **Commands** — This window, which takes up most of the left half of the screen, lists and gives you access to the programming commands. (Paragraph 3.4 explains each command.) The PROG11 title includes the software version number.
- **Ranges** — This window, at the top center of the screen, shows the types and addresses of MCU memory ranges. This window expands to show additional sub-ranges, which you may define via the R (user-defined range) command. (Figure 3-2 shows ranges for a KA4 MCU; these values will be different for other MCUs.)
- **Status** — This window, at the upper right of the screen, shows the status of the memory ranges listed in the ranges window. When the PROG11 screen first appears, this window is empty. Status indicators, such as BLANK, NOT BLANK, or VERIFIED, appear in the status window as you execute PROG11 commands.
- **S-Record File** — This window, near the lower right corner of the screen, shows the path and filename of the specified S-record. If you have not yet specified an S-record, the words `none specified` appear in this window. (Starting PROG11 from within RAPID includes automatic specification of an S-record.)
- **Messages** — This window extends across the bottom of the screen. Prompts, amplifications of status indications, and error messages appear in this window.



**Figure 3-1. SPGMR11 Programming Screen**

### 3.4 PROGRAMMING COMMANDS

Table 3-1 lists the programming commands. Individual explanations of each command follow the table. Paragraph 3.5 gives programming examples.

**Table 3-1. Programming Commands**

| Command | Function                | Command | Function           |
|---------|-------------------------|---------|--------------------|
| BA      | Blank check all ranges  | VA      | Verify all ranges  |
| BR      | Blank check range       | VR      | Verify range       |
| EA      | Bulk erase EEPROM range | UA      | Upload all ranges  |
| ER      | Byte erase EEPROM range | UR      | Upload range       |
| EB      | Erase EEPROM byte       | S       | Specify S-record   |
| MD      | Memory display          | R       | User-defined range |
| PA      | Program all ranges      | H       | Help               |
| PR      | Program range           | Q       | Quit               |
| PB      | Program byte            | X       | Reset MCU          |

#### BA

#### Blank Check All Ranges

This command blank checks all ROM, EPROM, or EEPROM ranges of the MCU in the programming socket. (The ranges window lists these ranges.) That is, the system verifies that each range byte contains the hexadecimal value \$FF. The system verifies that the CONFIG register is in the erased state defined in the MCU personality file.

Blank checking verifies successful erasure of memory ranges. This is important, as EPROM or EEPROM bytes must be blank before you can program them.

**BR****Blank Check Range**

This command blank checks one ROM, EPROM, or EEPROM range of the MCU in the programming socket. When you activate this command, a prompt asks you to select one of the ranges in the ranges window. Then the system verifies that each byte of the specified range contains the hexadecimal value \$FF. The system verifies that the CONFIG register is in the erased state defined in the MCU personality file.

---

**EA****Bulk Erase EEPROM Range**

This command erases all EEPROM ranges of the MCU in the programming socket, possibly including the CONFIG register. When you activate this command, a prompt asks you to select one of the EEPROM ranges listed in the ranges window (the CONFIG register is one of these EEPROM ranges). If you select the CONFIG register range, the system erases all EEPROM ranges, including the CONFIG register. (New CONFIG-register settings take effect when you reset the MCU.) If you select any other EEPROM range, the system erases all EEPROM ranges except the CONFIG register.

---

**ER****Byte Erase EEPROM Range**

This command erases an EEPROM range of the MCU in the programming socket. When you activate this command, a prompt asks you to select one of the EEPROM ranges in the ranges window. When you select the range, the system erases each byte of the range.

---

**EB****Erase EEPROM Byte**

This command erases one byte of EEPROM of the MCU in the programming socket. When you activate this command, the system prompts for the address of the byte. When you enter the address and press RETURN, the system erases the byte.

**MD****Memory Display**

This command displays the contents of 48 (\$30) bytes of memory. When you activate this command, the system prompts for a starting address. Enter the hexadecimal byte address; in the messages window, the system displays the contents of the 48 bytes starting with that address.

---

**PA****Program All Ranges**

This command programs all EPROM or EEPROM ranges of the MCU in the programming socket. When you activate this command, the system prompts for the S-record file that contains the new code. Select the S-record; the system programs the S-record contents into all ranges that have corresponding addresses. (If you have not yet specified an S-record, the system prompts for one.) (The S command information explains how to customize the system prompt for S-records.) (Starting PROG11 from within RAPID includes automatic specification of an S-record.)

**NOTE**

An S-record includes address range definitions. The system programs (that is, changes the contents of) EPROM or EEPROM bytes included in the S-record address definitions.

**PR****Program Range**

This command programs an EPROM or EEPROM range of the MCU in the programming socket. When you activate this command, the system prompts for the EPROM or EEPROM range. Select this range, and the system programs the contents of the specified S-record into the range. (If you have not yet specified an S-record, the system prompts for one.) (The S command information explains how to customize the system prompt for S-records.) (Starting PROG11 from within RAPID includes automatic specification of an S-record.)

**NOTE**

An S-record includes address range definitions. The system programs (that is, changes the contents of) EPROM or EEPROM bytes included in the S-record address definitions.

---

**PB****Program Byte**

This command programs one byte of EPROM or EEPROM of the MCU in the programming socket. When you activate this command, the system prompts for the address of the byte. Enter the address and press RETURN; the system prompts for the data. Enter the data and press RETURN; the system programs the byte.

---

**VA****Verify All Ranges**

This command verifies all ROM, EPROM, or EEPROM ranges of the MCU in the programming socket. That is, the system makes sure that contents of all ranges match those of a specified S-record. When you activate this command, the system begins verifying. (If you have not yet specified an S-record, the system prompts for one.) (The S command information explains how to customize the system prompt for S-records.) (Starting PROG11 from within RAPID includes automatic specification of an S-record.) If the system finds a discrepancy, verification activity ends and a failure message appears. If contents match, a success message appears.

**VR****Verify Range**

This command verifies a ROM, EPROM, or EEPROM range of the MCU in the programming socket. That is, the system makes sure that contents of the ROM range match those of a specified S-record. When you activate this command, the system begins verifying. (If you have not yet specified an S-record, the system prompts for one.) (The S command information explains how to customize the system prompt for S-records.) (Starting PROG11 from within RAPID includes automatic specification of an S-record.) If the system finds a discrepancy, verification activity ends and a failure message appears. If contents match, a success message appears.

---

**UA****Upload All Ranges**

This command uploads all ROM, EPROM, or EEPROM contents from the MCU in the programming socket to a specified file, in S-record format. When you activate this command, the system prompts for a filename. Enter the filename and press RETURN. The system begins uploading, showing contents in the messages window. When uploading is done, the system restores the PROG11 screen.

---

**UR****Upload Range**

This command uploads contents of one ROM, EPROM, or EEPROM range of the MCU in the programming socket to a specified file, in S-record format. When you activate this command, the system prompts for a filename. Enter the filename and press RETURN. The system prompts for a range. Select the range and press RETURN; the system begins uploading. When uploading is done, the system restores the PROG11 screen.

**S****Specify S-Record**

This command specifies an S-record file. When you activate this command, the search pattern window appears. This window contains the default search string **\*.S19**. To accept the default string, press RETURN. Otherwise, enter a new search string in DOS format, then press RETURN. Another temporary window appears, showing all files that fit the search pattern, as well as the names of other directories. Select a file. This restores the main screen, which shows the pathname of the newly specified file in the S-record file window. (Starting PROG11 from within RAPID includes automatic specification of an S-record.)

**R****User-Defined Range**

This command lets you define a custom range in the memory map of the MCU in the programming socket. When you activate this command, prompts ask for the hexadecimal starting and ending addresses of the range. When you enter these addresses, the system defines the range; this new range appears in the ranges window.

**H****Help**

This command calls up help information. When you activate this command, help topics appear in a temporary window. When you select a topic, the help information replaces the topic list. Clicking on the diamond symbol at the upper left corner of a help window returns you to a previous window (depending on your location in the help-window hierarchy). When you are done with help information, press the ESCAPE key two or three times to return to the main screen.

**Q****Quit**

This command quits the PROG11 software, returning you to the DOS prompt. When you activate this command, the yellow LED goes out, and the prompt . . . remove the MCU . . . appears. Remove the MCU and press RETURN; the DOS prompt replaces the PROG11 screen.

---

**X****Reset MCU**

This command powers down the HC11 MCU in the programming socket, enabling you to program another MCU of the same type without quitting and restarting the PROG11 software. When you activate this command, the yellow LED goes out, and the prompt . . . change MCU if necessary . . . appears. Remove the current MCU from the programming socket, insert the new MCU into the socket, then press RETURN. The messages, *Resetting the HC11 program . . .* and *Loading boot code to HC11 . . .* appear in the messages window. The spinning symbol indicates that downloading is in progress. When the MCU is ready to be programmed, the system returns cursor control to the commands window.

Another use for this command is to reset the MCU currently in the programming socket. This action may be appropriate as part of programming the CONFIG register (as paragraph 3.6 explains) or as part of troubleshooting.

### 3.5 TYPICAL PROGRAMMING SEQUENCE

Although your specific situation determines the most appropriate programming actions, steps 1 through 9 will be typical for most users:

1. Use the **BA** command to blank check all ranges. Memory ranges must be blank before you can program them.
2. If EEPROM ranges are not blank, use the **EA** or **ER** command to erase them.
3. If EPROM ranges are not blank, use ultraviolet light to erase them.
4. Use the **S** command to specify an S-record. (Starting PROG11 from within RAPID includes automatic specification of an S-record.)
5. Use the **PA** command to program all ranges.
6. Optionally, use the **MD** command to see memory contents.
7. To program another MCU of the same type, use the **X** command. At the prompt, remove the current MCU and insert the new one. Return to Step 1, above.
8. When you are done programming MCUs of the current type, use the **Q** command to quit the PROG11 programming software.
9. To program an MCU of a different type, restart the PROG11 software and return to Step 1, above.

### 3.6 PROGRAMMING THE CONFIG REGISTER

Most M68HC11 MCUs have a nonvolatile configuration (CONFIG) register. This special control register enables or disables ROM or EPROM, EEPROM, the computer operating properly (COP) watchdog system, and optional EEPROM security. The SPGMR11 can program the CONFIG register, as this register is a byte of EEPROM (separate from the EEPROM range of your MCU).

#### NOTES

Before programming the CONFIG register, be sure that it is nonvolatile in your MCU. (The technical data book for the MCU explains whether the CONFIG register is volatile or nonvolatile.) Programming a volatile CONFIG register is futile, as the programmed value disappears when you remove power from the MCU.

A new CONFIG-register value does not take effect until an MCU reset occurs. Thus, PROG11 software can display a not verified, not blank, or error message that is not relevant to the CONFIG register.

Follow steps 1 through 10 to program the CONFIG register:

1. Select the **EB** (erase byte) command. When the system prompts for the byte address, enter the address of the CONFIG register (\$103F or \$003F, according to the MCU.) Press RETURN and the system erases the byte.  
**EXCEPTION:** For A-family MCUs, select the **EA** (bulk erase EEPROM range) instead of the **EB** command.
2. Depending on the specific MCU, the message `Erase not verified...` may appear. If so, disregard this message.
3. Select the **X** command. The yellow LED goes out, and the prompt `...` `change MCU if necessary...` appears. Do not remove the MCU from the programming socket. Press RETURN. The messages, `Resetting the HC11 program...` and `Loading boot code to HC11...` appear in the messages window. The spinning symbol indicates that downloading is in progress.
4. Select the **BR** (blank check range) command. When the system prompts for the range, select the first range of the ranges window and press RETURN. (The first range of the ranges window is the CONFIG register.)

5. Select the **PB** (program byte) command. When the system prompts for the byte address, enter the address of the CONFIG register. Press RETURN, and the system prompts for the data. Enter the new value for the CONFIG register, then press RETURN.
6. Depending on the specific MCU, a `...not verified...error...` message may appear. Disregard such a message, which is not relevant to the CONFIG register.
7. Select the **X** command. The yellow LED goes out, and the prompt `... change MCU if necessary...` appears. Do not remove the MCU from the programming socket. Press RETURN. The messages, `Resetting the HC11 program...` and `Loading boot code to HC11...` appear in the messages window. The spinning symbol indicates that downloading is in progress.
8. Wait for control and the cursor to return to the commands window. Then select the **MD** command. When the system prompts for the starting address, enter the address of the CONFIG register and press RETURN.
9. The system displays the contents of 48 (\$30) bytes of memory, starting with the CONFIG register. This lets you verify that the new value is programmed into the CONFIG register.
10. This completes CONFIG register programming.

#### NOTE

The SPGMR11 automatically puts the MCU into special bootstrap mode. For some types of MCU, reset in special bootstrap mode forces a specific value into the CONFIG register static latches. (If so, the corresponding technical data book gives the value.) The above instructions do program the CONFIG register EEPROM byte of any MCU type. But the Step 8 **MD** command shows (in Step 9) any value forced by the reset instead of the programmed value. This prevents visual verification of programming, per Step 9.



## CHAPTER 4

### SUPPORT INFORMATION

This chapter contains connector pin assignments and signal descriptions, troubleshooting information, and logic diagrams for the SPGMR11.

#### 4.1 CABLE CONNECTORS

The SPGMR11 HOST and EMULATOR connectors are 9-pin D-shell connectors that have identical pin assignments and signal descriptions. Table 4-1 describes these signals.

**Table 4-1. HOST, EMULATOR Connector Signal Descriptions**

| Pin | Mnemonic | Signal  |
|-----|----------|---|
| 1   | DCD      | DATA CARRIER DETECT - Output signal (held high) that indicates detection of an acceptable carrier signal. |
| 2   | RX       | RECEIVE DATA - Serial data output line.   |
| 3   | TX       | TRANSMIT DATA - Serial data input line  |
| 4   | DTR      | DATA TERMINAL READY - Input signal that indicates on-line/in-service/active status.                       |
| 5   | GND      | GROUND - Signal ground or common return connection between the SPGMR11 and the computer.                  |
| 6   | DSR      | DATA SET READY - Output signal (held high) that indicates on-line/in-service/active status.               |
| 7   | RTS      | REQUEST TO SEND - Input signal that requests permission to transfer data.                                 |
| 8   | CTS      | CLEAR TO SEND - Output signal that indicates a ready-to-transfer data status.                             |
| 9   | RI       | RING INTERRUPT - Not used for the SPGMR11.  |

## 4.2 TROUBLESHOOTING GUIDE

Table 4-2 lists possible problems and corrective actions for your SPGMR11.

**Table 4-2. SPGMR11 Troubleshooting**

| <b>Problem</b>   | <b>Probable Cause</b>   | <b>Corrective Action</b>  |
|--|---|---|
| PROG11 software does not start, message <b>Error downloading boot code</b> appears | 1. COM port mismatch  | Make sure that the communications cable runs to the COM port specified in the PROG11 command. (COM1 is the default.)          |
|  | 2. Inappropriate cable  | Be sure to use a <i>straight-through</i> cable between the SPGMR11 and the computer.  |
|  | 3. Mismatch between the actual and selected MCU type                  | Select the MCU file that matches the MCU in the programming socket.   |
|  | 4. Programming adapter not properly inserted                          | Remove and re-insert the adapter, making sure that all pins are plugged into the SPGMR11 connector.                           |
| Repeated PUT error and GET error messages: loss of communication with the MCU      | 1. Mismatch between the actual and selected MCU type                  | Select the MCU file that matches the MCU in the programming socket.   |
|  | 2. Improper use of PB command to program RAM or certain SCI registers | Select the X command to reset the MCU, then reprogram correctly. (Do not use PROG11 commands to modify RAM or SCI registers.) |
| Repeated resetting and loading messages  | Mismatch between the actual and selected MCU type                     | Select the MCU file that matches the MCU in the programming socket.   |

### 4.3 ZIF SOCKET

The connectors on the top of the SPGMR11 are rated for at least 50 insertion cycles. If you anticipate changing programming adapters often, you should use a zero insertion force (ZIF) socket:

1. Assemble the ZIF socket and its mating receptacle. (Aries part numbers are 32-6554-16 and 32-6556-11. A comparable socket and mating receptacle from another manufacturer also may be satisfactory.)
2. Plug the completed ZIF socket assembly onto the top of the SPGMR, so that the socket lever is close to the SPGMR11 LEDs.
3. To change programming adapters:
  - a. Move the lever to open the socket contacts,
  - b. Remove the current adapter,
  - c. Insert the new adapter, and
  - d. Move the lever to close the socket contacts.



## INDEX

cables: 1-2, 1-4, 2-1 — 2-3

commands, programming: 3-6 — 3-12

CONFIG register, programming: 3-13, 3-14

connections, system: 2-1 — 2-4

connectors:

- EMULATOR: 1-1, 2-3, 4-1
- HOST: 1-1, 2-1 — 2-4, 4-1
- power: 1-1, 2-1 — 2-4

EMULATOR connector: 1-1, 2-3, 4-1

features: 1-3

hardware preparation: 2-1 — 2-5

HOST connector: 1-1, 2-1 — 2-4, 4-1

installing programming adapters: 2-4, 2-5

introduction: 1-1 — 1-5

loading software: 3-1

making system connections: 2-1 — 2-4

overview: 1-4

pin assignments, connector: 4-1

power connector, 5-volt: 1-1, 2-1 — 2-4

preparing to program: 3-1

programming: 3-1 — 3-15:  
    commands: 3-6 — 3-12  
    CONFIG register: 3-14, 3-15  
    preparing to program: 3-1  
    PROG11 screen: 3-5  
    starting: 3-2, 3-3  
    typical sequence: 3-13

programming adapters:  
    available: 1-2  
    installing: 2-4, 2-5

RAPID (integrated development environment): 3-4

screen, PROG11 programming: 3-5

sequence, typical programming: 3-13

signal descriptions, connector: 4-1

software, loading: 3-1

specifications: 1-3

starting programming: 3-2, 3-3

support information: 4-1 — 4-3

system connections: 2-1 — 2-4

troubleshooting: 4-2

typical programming sequence: 3-13  
    user requirements: 1-4  
    ZIF socket: 2-5, 4-3