



R6500 Microcomputer System APPLICATION NOTE

Preparing an AIM 65 BASIC Program for PROM/ROM Operation

PURPOSE

This Application Note describes a method of preparing a user-written BASIC program for operation in a PROM or ROM in the AIM 65. Specifically illustrated is a BASIC program which will execute at addresses D000-DFFF, and can be started by typing the N key. The program variables are assigned to RAM, in addresses 0200-0FFF. Two short support programs, a BASIC Relocator and a BASIC Driver, are listed with key variables described. The procedure can easily be adapted to meet other memory and relocation requirements.

The procedure describes how to do the following:

1. Load and run a BASIC program. The program is relocated in RAM from its normal starting address, however, to allow the BASIC Driver to be added.
2. Load the BASIC Driver and run the BASIC program again to verify proper driver coding.
3. Change the BASIC program statement linkage addresses to the new addresses, using the BASIC Relocator.
4. Change the starting addresses of the BASIC program and BASIC variables in the BASIC Driver to the new addresses.
5. Dump the BASIC Driver and BASIC program to audio cassette for programming into PROM/ROM.
6. Store the BASIC Driver and BASIC Program into PROM/ROM.
7. Install the PROM/ROM in AIM 65 and operate the BASIC program.

DESCRIPTION

1. Load and Run BASIC Program

The first step is to verify that the desired BASIC program will execute properly in the new address space.

- a. Enter BASIC by typing 5. Enter the desired memory size and terminal width values.
- b. When the " ^ AIM 65 BASIC V1.1" message is displayed, press ESC to return to the Monitor.

- c. Change the initial starting and ending addresses of the BASIC program as follows, to allow the BASIC Driver to be added later:

Address	Parameter	Initial Value	New Value
073	BASIC Program Starting Address Low + 1	12	81
074	BASIC Program Starting Address High	02	02
075	BASIC-Variables Starting Address Low*	14	83
076	BASIC Variables Starting Address High*	02	02

*Normally, the initial program start address + 3.

(M) 0073 12 02 14 02
(/) 0073 81 83

- d. Change the contents of the first three addresses of the BASIC program to S00.

Program Address	New Value
0280	S00
0281	S00
0282	S00

(M) 0280 AA AA AA AA
(/) 0280 00 00 00

- e. Type 6 to re-enter BASIC. Type or load the BASIC program. Run the BASIC program to verify proper operation.

Example:

```
100PRINT"TYPE A STRI
NG"
110INPUTAS
120FORN=1TOLEN(AS)
130PRINTLEFTS(AS,N)
140NEXT
```

If the program will not enter or run properly, either the addresses or the S00 values are not entered properly.

- f. Save the BASIC program on audio cassette for future reference.

Preparing an AIM 65 BASIC Program for PROM/ROM Operation

- g. Return control the AIM 65 Monitor by pressing the RESET button (or the ESC key if in the BASIC command mode).
- h. Determine the address of the second BASIC statement by examining memory starting at the BASIC program starting address with the Monitor M command (location given for previous example):

(M) 0280 00 96 02 64

2. Load and Verify the BASIC Driver

The BASIC Driver tells the BASIC Interpreter the new locations of the BASIC program and variables. The following steps must be taken to load and verify the Driver:

- a. Load the BASIC Driver object code. The object code may be prepared using the AIM 65 Assembler (refer to the assembly listing in Figure 1).
- b. Enter the required values of the BASIC program starting address, BASIC variables starting address, and BASIC variables ending address, if not loaded with the BASIC Driver object code. (Note example is for 1K of RAM only.)

Driver Address	Parameter	Value
0248	BASIC Program Starting Address Low + 1	81
024A	BASIC Program Starting Address High	02
0250	BASIC Variables Starting Address Low	00
0252	BASIC Variables Starting Address High	03
0258	BASIC Variables Ending Address Low + 1	00
025A	BASIC Variables Ending Address High	04

(M) 0248 XX A9 XX 86
 (/) 0248 81 02
 (M) 0250 XX A9 XX 86
 (/) 0250 00 03
 (M) 0258 XX A9 XX 86
 (/) 0258 00 04

- c. Run the BASIC program by using the Monitor * and G commands to jump to the BASIC Driver:

* = 0200
 G/.

If the BASIC program does not run properly, either the BASIC Driver instructions or the entered addresses are incorrect.

- d. Return control to the Monitor by pressing RESET or typing ESC (if in the BASIC command mode).

The BASIC Driver is started at 0200 to allow simple relocation to D000 and to minimize use of PROM/ROM memory. However, some BASIC functions use addresses 0200-0210, so check these addresses after running the BASIC program to verify they have not been altered.

3. Change the BASIC Program Statement Addresses

The BASIC Relocator changes all the addresses in the BASIC program to the addresses required at the new location; D000-DFFF in this case.

- a. Load the BASIC Relocator object code. The object code may be prepared by using the AIM 65 Assembler (refer to the assembly listing in Figure 2). Note that the assembly object code cannot be directed to memory during assembly since it is located in page zero; instead, direct the object code to audio cassette.
- b. Enter the required values of the old and new BASIC program starting addresses, if not loaded with the BASIC Relocator object code.

Relocator Address	Parameter	Value
0009	New BASIC Program Start Address Low + 1	81
000B	New BASIC Program Start Address High	D0
000F	Old BASIC Program Start Address Low + 1	81
0014	Old BASIC Program Start Address High	02
001C	Old BASIC Program Start Address Low + 1	81
0020	Old BASIC Program Start Address High	02

- c. Run the BASIC Relocator by using the Monitor * and G commands:

* = 0008
 G/.

- d. The Monitor prompt will be displayed upon completion.
- e. Verify the relocation process by comparing the new address of the second BASIC statement with the old address as recorded in Step 1-h.

(M) 0280 00 96 D0 64

NOTE: The BASIC program cannot be executed after the statement addresses have been changed until the BASIC program is installed at the new addresses, e.g., D000-DFFF.

4. Change the BASIC Driver Addresses

Change the starting addresses of the BASIC program and variables in the BASIC Driver to these new values:

Driver Address	Parameter	Value
0248	BASIC Program Starting Address Low + 1	81
024A	BASIC Program Starting Address High	D0
0250	BASIC Variables Starting Address Low	12
0252	BASIC Variables Starting Address High	02

(M) 0248 81 A9 02 86
 (/) 0248 D0
 (M) 0250 00 A9 03 86
 (/) 0250 12 02

5. Dump the BASIC Driver and BASIC Program to Audio Cassette

Dump the BASIC Driver and BASIC Program to audio cassette using the Monitor D command:

```
(D) OUT=T F=DATA
    FROM=0200      (Start of BASIC Driver)
    TO=0FFF        (End of BASIC Program)
```

6. Incorporate the BASIC Driver and BASIC Program into PROM/ROM

- a. Identify the PROM or ROM that is to contain the BASIC Driver and BASIC Program. Some typical PROMs/ROMs and their capacities are:

2708 PROM	=	400 (hex)	=	1024 (dec)	bytes
2716 PROM	=	800	=	2048	bytes
TMS 2532 PROM	=	1000	=	4096	bytes
R2316 ROM	=	800	=	2048	bytes
R2332 ROM	=	1000	=	4096	bytes

- b. The listed PROMs can be programmed using the Rockwell SYSTEM 65 Microcomputer Development System with PROM Programmer Option. The object code may be transmitted from AIM 65 to SYSTEM 65 by using the interface described in Application Note R6500 N04.

- c. The BASIC Driver and BASIC Program can also be masked into ROM by submitting the object code on audio cassette to Rockwell. Refer to the data sheets for the R2316 or R2332.

7. Install the PROM/ROM in AIM 65 and Operate the BASIC Program

- a. Install the programmed PROM/ROM in AIM 65 socket Z24 (D000-DFFF).
- b. Start the BASIC Program by typing N.
- c. Return to the Monitor by pressing RESET or typing ESC (if in the BASIC command mode).

```

2000          PGMST  =#$2B1          ;BASIC PROGRAM START ADDRESS
2000          RAMBOT =#$300          ;BASIC VARIABLE START ADDR
2000          RAMTOP =#$400          ;VARIABLE END ADDRESS
2000          *=$200
0200 A9 E1      START LDA  ##E1      ;INITIALIZE BASIC POINTERS
0202 B5 B2      STA  #B2
0204 A2 FE      LDX  ##FE
0206 9A         TXS
0207 D8         CLD
0208 A9 4C      LDA  ##4C
020A B5 00      STA  #00
020C B5 03      STA  #03
020E B5 9C      STA  #9C
0210 B5 BB      STA  #BB
0212 A2 B7      LDX  ##B7
0214 A9 BF      LDA  ##BF
0216 B6 BC      STX  #BC
0218 B5 BD      STA  #BD
021A B6 04      STX  #04
021C B5 05      STA  #05
021E A9 14      LDA  ##14
0220 B5 12      STA  #12
0222 A9 0A      LDA  ##0A
0224 B5 13      STA  #13
0226 A2 1C      LDX  ##1C
0228 BD B5 CE   L1    LDA  #CEB5>X
022A 95 BE      STA  #BE>X
022C CA         DEX
022E D0 FB      BNE  L1
0230 A9 03      LDA  #03
0232 B5 9B      STA  #9B
0234 A9 00      LDA  ##00
0236 B5 01      STA  #01
0238 B5 B0      STA  #B0
023A 4B         PHA
023C B5 60      STA  #60
023E B5 10      STA  #10
0240 A9 61      LDA  ##61
0242 B5 5E      STA  #5E
0244 A9 B9      LDA  ##B9
0246 B5 02      STA  #02
0248 A2 B1      LDX  <<PGMST          ;BASIC PROG START ADR
024A A9 02      LDA  >>PGMST          ;CHG WHEN RELOCATED
024C B6 73      STX  #73
024E B5 74      STA  #74
0250 A2 00      LDX  <<RAMBOT          ;BASIC VARIABLE START ADDR
0252 A9 03      LDA  >>RAMBOT
0254 B6 75      STX  #75
0256 B5 76      STA  #76
0258 A2 00      LDX  <<RAMTOP          ;BASIC VARIABLE END ADDR
025A A9 04      LDA  >>RAMTOP
025C B6 7F      STX  #7F
025E B5 80      STA  #80
0260 20 7C B4   JSR  #B47C          ;CALL RUN
0262 4C B8 B5   JMP  #B5CB          ;JUMP TO FIRST STATEMENT
0264          ;
0265          ;
0266          .END

```

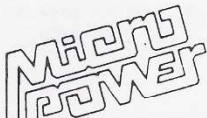
Figure 1. BASIC Driver

```

2000          PGMST  =#D081          #NEW PROGRAM START ADDRESS
2000          *=$04
0004          NEXT  **=*+2
0006          OFF   **=*+2
0008
0008 A9 81     START  LDA #<PGMST     #NEW PROGRAM START ADDR
000A A2 D0     LDX #>PGMST
000C D8        CLD
000D 38        SEC
000E E9 81     SBC #81
0010 85 06     STA OFF              #OLD START ADDR LOW
0012 8A        TXA
0013 E9 02     SBC #2
0015 85 07     STA OFF+1            #OLD START ADDR HIGH
0017 A2 00     LDX #0
0019 A0 01     LDY #1
001B A9 81     LDA #81              #OLD START ADDR LOW
001D 85 04     STA NEXT
001F A9 02     LDA #2              #OLD START ADDR HIGH
0021 85 05     STA NEXT+1
0023 A1 04     J2      LDA (NEXT),X
0025 11 04     ORA (NEXT),Y
0027 D0 03     BNE J1
0029 4C A1 E1  J1      JMP $E1A1     #RETURN TO AIM MONITOR
002C 18        CLC
002D A1 04     LDA (NEXT),X
002F 48        PHA
0030 65 06     ADC OFF
0032 81 04     STA (NEXT),X
0034 B1 04     LDA (NEXT),Y
0036 48        PHA
0037 65 07     ADC OFF+1
0039 91 04     STA (NEXT),Y
003B 68        PLA
003C 85 05     STA NEXT+1
003E 68        PLA
003F 85 04     STA NEXT
0041 4C 23 00  JMP J2
0044          .END

```

Figure 2. BASIC Relocator



ELECTRONIC DEVICES DIVISION REGIONAL ROCKWELL SALES OFFICES

HOME OFFICE

Electronic Devices Division
Rockwell International
3310 Miraloma Avenue
P.O. Box 3669
Anaheim, California 92803
(714) 632-3729
TWX: 910 591-1698

UNITED STATES

Electronic Devices Division
Rockwell International
1842 Reynolds
Irvine, California 92626
(714) 632-3710
DDD (714) 545-6227

Electronic Devices Division
Rockwell International
921 Bowser Road
Richardson, Texas 75080
(214) 996-6500
Telex 73-307

Electronic Devices Division
Rockwell International
10700 West Higgins Rd., Suite 102
Rosemont, Illinois 60018
(312) 297-8862
TWX: 910 233-0179 (RI MED ROSM)

Electronic Devices Division
Rockwell International
5001B Greentree
Executive Campus, Rt. 73
Marlton, New Jersey 08053
(609) 596-0090
TWX: 710 940-1377

EUROPE

Electronic Devices Division
Rockwell International GmbH
Fraunhoferstrasse 11
D-8033 Munchen-Martinsried
Germany
(089) 859-9575
Telex 0521/2650

Electronic Devices Division
Rockwell International
Heathrow House, Bath Rd.
Cranford, Hounslow,
Middlesex, England
(01) 759-9911
Telex: 851-25463

FAR EAST

Electronic Devices Division
Rockwell International Overseas Corp
Itohpia Hirakawa-cho Bldg.
7-6, 2-chome, Hirakawa-cho
Chiyoda-ku, Tokyo 102, Japan
(03) 265-8806
Telex J22198

YOUR LOCAL REPRESENTATIVE

3-81



Rockwell International

...where science gets down to business