

NATIONAL RADIO ASTRONOMY OBSERVATORY  
GREEN BANK, WEST VIRGINIA

ELECTRONICS DIVISION INTERNAL REPORT No. 191

AN EVALUATION KIT TO  
ROM PROGRAMMER INTERFACE

STEPHEN MACMINN\*

AUGUST 1978

NUMBER OF COPIES: 150

---

\* SUMMER STUDENT.

## AN EVALUATION KIT TO ROM PROGRAMMER INTERFACE

Stephen MacMinn

The Read Only Memory programmer/verifier currently in use at NRAO (see Electronics Division Internal Report No. 132) is cumbersome to use, often requiring more than an hour of the programmer's time. To facilitate use of this device and software development in general, a hardware/software interface has been devised to connect the PROM programmer to the MEK6800D2 evaluation kit. This allows software to be developed, edited, debugged, etc., on the evaluation kit, stored on cassette tapes thru the kit's cassette interface and, when ready, to be automatically burned into a 1702 EPROM using the ROM programmer and this interface.

The hardware portion of this interface is simply a ribbon cable (see Appendix A) connecting the ROM to verify socket on the ROM programmer and the application connector (J1) on the evaluation kit. Through this cable, the processor reads the current address being programmed, and sends out the data to be put in that address to the ROM programmer. Please note that no handshaking takes place in this system! Functioning of this interface depends entirely on the slow speed of the ROM programmer to allow the microprocessor time to read the address and supply the data before the programmer attempts to use that data. Since the ROM programmer allows 490 ms between supplying the address and reading the data and it takes the processor less than 100  $\mu$ s to supply the data after reading the address, a very wide margin for error is allowed. If attempts are made to speed up the ROM programmer, however, this timing margin should be considered or the ROM programmer may attempt to use the data before it becomes valid!

The software portion of this interface is shown in Appendix B. This program is very short and it may be easily modified to suit the individual user's purpose. This program may be found in the digital cassette library, and is easily loaded into the evaluation kit's memory by following the procedure described in the kit's instruction manual. A short description of the software follows:

The first 16 bytes of this program initialize the PIA ports, port A to read the current address from the PROM programmer, and port B to write the corresponding data out to the PROM programmer. The address of the first RAM location to be transferred is loaded into the index register and the data at that address is transferred to the ROM programmer. The program then waits for the address input to change and compares the new value to the last ROM address to be programmed. If they match, the program outputs 00 hex to the ROM programmer and returns to the system monitor. If they don't match, the program increments the index register and sends the new data.

Note that the first RAM address to be sent and the last ROM location to be programmed must be loaded into the program in the locations marked by R's in the program listing. The RAM address follows the LDX instruction and the last ROM address to be programmed follows the LDAB instruction (see program listing, Appendix B).

Example: Suppose we want to program locations 10 thru D0 of the evaluation kit's RAM into locations 00 thru C0 of a 1702 EPROM:

1. With the ROM programmer power off, plug the ROM you wish to program into the ROM to verify socket. Apply power and verify that it is erased. (Sample a few random locations; they should contain zeroes.)

2. Turn the programmer power off. Remove the ROM and put it in the ROM to program socket. Plug the 24 pin side of the interface cable into the ROM to verify socket, taking care to align pin 1 properly. Never plug the interface cable into the ROM to program socket!!!!
3. Plug the other end of the interface cable into applications connector J1 on the evaluation kit taking care to align pin A properly.
4. Load the ROM programmer interface software and the software to be programmed into the evaluation kit.  
Note: The tape version of the interface software loads at location 0120H.
5. Type 0137 M  
Type 00 G  
Type 10 G  
This sets the first RAM address to be programmed as 0010H.
6. Type E  
Type 013C M  
Type C0  
This sets the last ROM address to be programmed.
7. Turn on the programmer. (Make sure the  $V_{DD}$  light comes on. If not, toggle the power switch until it does.)
8. Enter 000 on the thumbwheel switches and push the ENTER ADDR button. Verify that 000 appears on the 7 segment display. Put the mode switch on Copy Program.

## 9. Type E

Type 0120G

The evaluation kit display should go dark and the first byte of data should appear on the programmer's 8 Data LED's.

10. Push the program sequence button on the programmer. The lights above this button should flash and the programmer should begin stepping through addresses. Verify that the proper data appears for the first couple locations (about 10 sec/location).
11. Go have a cup of coffee.

Tips on Use:

- 1) Use caution when handling 1702's; they're static sensitive.
- 2) Never plug or unplug ROM's or the interface cable with the programmer power on.
- 3) The 7 segment display on the ROM programmer is in decimal, not hex. If you want to know the current hex address, look at the 8 address LED's just above the 7 segment display.
- 4) To re-run the program reset the ROM programmer to 000; type E; type 0120G, and push the sequence button.
- 5) Program the final copy of your ROM twice to insure that it gets the data, and verify it!
- 6) Make sure a ROM is thoroughly erased before you try to program it.

## APPENDIX A

## ROM Programmer Interface Cable - Connection List

Definition	MEK6800D2 Applications Connector (J1)	ROM Programmer ROM to Verify Socket
Address $\emptyset$	H	3
1	J	2
2	K	1
3	L	21
4	M	2 $\emptyset$
5	N	19
6	P	18
7	R	17
Ground	X	14
Data $\emptyset$	7	4
1	8	5
2	9	6
3	10	7
4	11	8
5	12	9
6	13	1 $\emptyset$
7	14	11

## APPENDIX B

\*\*\* \$PROMINT - PROM Programmer Interface, V2.0

\*

\* ENTRY - Start, Start + 1 = 1st RAM location to be sent

\* LASTADDR = last ROM location to be programmed

\* EXIT - Exits to monitor after programming LASTADDR;

\* leaves rest of ROM in unprogrammed state.

0000	CE 80 04	PROMINT	LDX	PIADDR	INITIALIZE THE PIA PORTS.
0003	6F 01		CLR	CONREGA	
0005	6F 03		CLR	CONREGB	
0007	86 FF		LOAD	FFH	
0009	A7 02		STAA	DATDIRB	B TO WRITE.
000B	6F 00		CLR	DATDIRA	A TO READ.
000D	86 04		LDAA	04H	
000F	A7 01		STAA	CONREGA	
0011	A7 03		STAA	CONREGB	
0013	8E 01 7F		LDS	STACKTOP	
0016	CE RR RR		LDX	START	GET FIRST ADDRESS.
0019	8D 12 X		BSR	SENDIMM	SEND THE DATA.
001B	C6 RR		LDAB	LASTADDR	
001D	08	.AGAIN	INX		

## Appendix B (continued):

001E 8D 0B	X	BSR	SEND	
0020 F1 80 04		CMPB	PIAIN	FINISHED YET?
0023 26 F8	X	BNE	.AGAIN	
0025 8D 0C	X	BSR	WAIT	YES,
0027 7F 80 06		CLR	PIAOUT	FILL WITH ZEROES,
002A 3F		SWI		AND EXIT.

\*\*\* \$SEND - Send a byte to ROM programmer when ready

\*

\* ENTRY - X points to data

\* EXIT - None

\* USES - A

\* CALLS - Wait

002B 8D 06	X	SEND	BSR	WAIT	WAIT UNTIL IT IS READY,
002D A6 00		SENDIMM	LDAA	(X + 0)	
002F B7 80 06			STAA	PIAOUT	THEN SENT IT.
0032 39			RTS		



## Appendix B (continued):

\*\*\* \$WAIT - Wait for ADDR change on input

\*

\* ENTRY - None

\* EXIT - Ready to send

\* USES - A

\* CALLS - None

\*

0033 B6 80 04	WAIT	LDAA	PIAIN	
0036 B1 80 04	.BACK	CMPA	PIAIN	READY YET?
0039 27 FB		BEQ	.BACK	NO, TRY AGAIN.
003B 39		RTS		YES, EXIT.