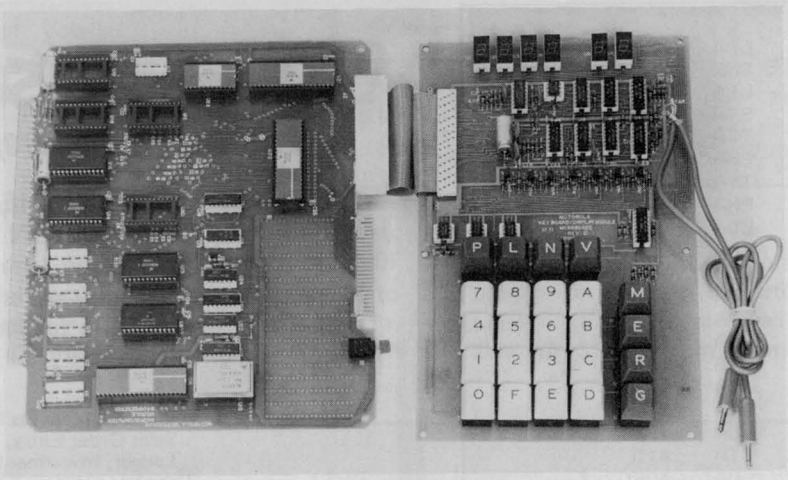
Terry Perdue
1470 Wilson Rd.
St. Joseph MI 49085



The Motorola MEK6800D2 Evaluation Kit.

Micro Maestro

a musical review of Motorola's MEK6800D2

For the last several years I have been noting with interest the ever-widening variety of applications for microprocessors. However, only recently did I decide that, as a design engineer, I had better stop procrastinating and start learning about this mysterious device, lest I be replaced by one.

I enrolled in a night class that began with number theory, proceeded through the workings of a hypothetical MPU, and finally covered the Motorola 6800 family. After learning the instruction set and addressing modes associated with the 6800, we began to write simple programs. As the programs became more complex, I began to wish that I had some means of running them to see if they would do what I intended. My employer agreed to purchase a trainer; the choice as to which one was left entirely up to me.

There are many trainer/evaluation systems available, and their features vary enough to make selection somewhat difficult. Having been introduced to the 6800 instructions, it seemed sensible to find a trainer using this chip. After a brief investigation of what was available, I came across a data sheet on the recently introduced Motorola MEK6800D2 Evaluation Kit. It seemed to have everything I needed to become more proficient in the use of the 6800 in particular, and programming in general, and the price of $235 was somewhat less than for others I had seen, so I ordered it.

Assembly

Upon opening the box, I found only a large three-ring binder full of documentation. Inside the binder, however, two of the "pages" contained circuit boards, with the associated components in plastic bubbles over each board. The assembly took about five hours and was aided by a manual containing pictures, parts placement drawings, and a page of construction hints.

The two boards, the Keyboard/Display Module and the Microcomputer Module, are joined by a 50-conductor ribbon cable. An edge connector on one end plugs onto the microcomputer circuit board, and the other end becomes permanently attached to the keyboard/display circuit board by means of another connector. Although the manual describes the installation of this connector, which is rather involved and requires a bench vise, the cable furnished had this connector already installed.

Sockets are provided for all of the 6800 family devices. These include the MC6800 MPU, an MCM6830 ROM containing the JBUG monitor, three MCM6810 RAMs (128 x 8), one of which is used as a scratchpad by JBUG, leaving 256 bytes for program storage, two MC6820 PIAs (Peripheral Interface Adapter), an MC6850 ACIA (Asynchronous Communications Interface Adapter), and the MC6871B crystal-controlled clock generator, operating at 614.4 kHz. The board has provisions for two additional RAMs, and two MCM68708 EPROMs (sockets provided), as well as for the buffers required to make the kit compatible with Motorola's EXORcisor. You may provide your own sockets for the remainder of the ICs, as I chose to do, but quality, low-profile types should be used.

The boards have mounting holes that allow spacers to be attached as feet, or the boards may be mounted into a chassis or enclosure. I found a large plastic box with carrying handle and integral hinge, and mounted the Microcomputer Module to the inside of the lid and the Keyboard/Display Module on long spacers to the bottom of the case. The long spacers provided room for a power supply beneath the board, making the unit self-contained and easily portable. The trainer requires a single 5 volt supply, and draws less than an Amp. I used a 6.3 volt filament transformer and bridge circuit, a 6000 uF electrolytic, and a 7805 regulator with heat sink. This should handle the two additional ROMs if added later, although if the use of the EPROMs is contemplated, plus and minus 12 volt supplies will also be required.

Features

The trainer provides a hexadecimal keyboard for program entry, and a six-digit

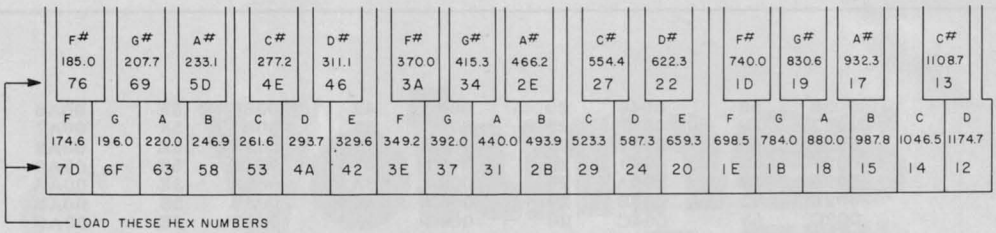


Fig. 1. Musical note-code chart.

hexadecimal LED display. Eight function keys are also provided. They are labeled M, E, R, G, V, N, L and P. The M key allows examination and modification of any memory location. E stands for Escape, and allows an exit from a particular operation. R provides a look at the MPU's internal registers. G is used to go to any step in the user's program and begin execution, and is also used to single-step through memory when loading or reviewing a program. The V key is used to set breakpoints (up to five) for checking and debugging, while the N key traces through a program by executing one instruction at a time.

The L and P keys are used with one of the really nice features of the unit. The MEK6800D2 has built-in facilities for storing programs on audio cassettes. The trainer is simply attached to the microphone and external speaker jacks of the recorder. The first and last addresses of the program to be stored are entered, and the P (punch) key is pressed with the recorder in record mode.

To load a program from a cassette, the L button is pressed with the recorder in play mode. The user is notified of completion of either process by means of the prompt character (a dash in the left LED which means ready). After reloading a long program by hand, this feature is really appreciated.

Another thoughtful feature is a grid of plated holes on standard .1" x .3" centers which is provided on the microcomputer board to accommodate a large number of wire-wrap sockets for breadboarding additional circuitry.

Using the Trainer

While programming proficiency may take a long time, mastery of the trainer comes quickly. The manual that accompanies the kit explains its operation in detail, and leads the user through a sample program which illustrates the use of the various functions. In addition to the functions mentioned earlier, the JBUG monitor provides a useful routine for calculating the offset required for a branch instruction without manually counting addresses. This eliminates one of the most common programming errors.

While operation is quite straightforward, there are two things which I consider unhandy. When reviewing the MPU's registers using the R key, the value displayed as the location of the stack pointer is always seven less than the true value. When entering instructions or data, the program counter must be manually advanced with the G key. This adds many unnecessary steps to the loading of a program. These are only minor inconveniences, however.

The documentation is excellent. The assembly/operation manual contains a listing of the JBUG monitor with flowcharts for each of its functions, and schematic diagrams for each board. Also included are the System Design Manual, containing detailed data on the 6800 series devices, and the Programming Reference Manual, which fully describes the instruction set and addressing modes.

Micro-maestro

Now that I've told you all about the trainer, let me give you an example of what fun it can be, aside from being instructive. After a few days of writing simple programs to gain confidence, I began to tire of watching the LED display for results, and decided to interface with the PIA. After learning how to address this versatile device, which can provide inputs and outputs in any desired software-controlled combination up to sixteen total, I wrote the following program. You can be sure that it didn't work the first time, or the second. But it does now, and as one of my first attempts at programming, I'm rather pleased with the result. It turns the trainer into a musical instrument which will play any tune you wish to load, using the note-codes given in the chart of Fig. 1 corresponding to the desired notes. The "instrument" is permanently tuned, due to

```
00001                                  NAM     MUSIC
00002                                  OPT     NOP,S
00003  0000                            ORG     0
00004  0000  7F  8005  REPLAY          CLR     $8005       CLEAR CRA-2 BIT
00005  0003  7C  8004                  INC     $8004       INCR DATA DIRECTION REG
00006  0006  73  8005                  COM     $8005       SET CR A-2 BIT
00007  0009  8E  0025                  LDS     *TEMP       POINT TO FIRST NOTE-1
00008  000C  CE  08FF  RUNOTE          LDX     *$08FF      TIME PER NOTE
00009  000F  33                        PUL     B           PULL NEXT NOTE FROM STACK
00010  0010  5D                        TST     B           HAS LAST NOTE BEEN PLAYED?
00011  0011  27  ED                    BEQ     REPLAY      IF NOT, CONTINUE
00012  0013  F7  0025                  STA     B  TEMP     STORE NOTE-CODE
00013  0016  4C                TONLUP  INC     A           TOGGLES OUTPUT WHEN STORED
00014  0017  F6  0025                  LDA     B  TEMP     GET NOTE-CODE
00015  001A  09                COUNT   DEX                 HOLD TONE FOR AWHILE
00016  001B  27  EF                    BEQ     NUNOTE      LONG ENOUGH YET?
00017  001D  5A                        DEC     B
00018  001E  26  FA                    BNE     COUNT
00019  0020  B7  8004                  STA     A  $8004    TOGGLES OUTPUT
00020  0023  20  F1                    BRA     TONLUP
00021  0025  0001              TEMP    RMB     1
```

Fig. 2. Music program listing.

0026	53	0046	20	0066	42	0086	3E	00A6	2B	00C6	3E
0027	42	0047	29	0067	37	0087	58	00A7	37	00C7	4A
0028	53	0048	20	0068	42	0088	3E	00A8	2B	00C8	3E
0029	42	0049	29	0069	37	0089	58	00A9	37	00C9	4A
002A	53	004A	20	006A	3A	008A	3E	00AA	2B	00CA	24
002B	42	004B	29	006B	46	008B	58	00AB	37	00CB	3E
002C	53	004C	20	006C	3A	008C	3E	00AC	2B	00CC	24
002D	42	004D	29	006D	46	008D	58	00AD	37	00CD	3E
002E	53	004E	20	006E	3E	008E	3E	00AE	2B	00CE	29
002F	42	004F	29	006F	4A	008F	58	00AF	37	00CF	42
0030	53	0050	20	0070	3E	0090	3E	00B0	2B	00D0	29
0031	42	0051	29	0071	4A	0091	58	00B1	37	00D1	42
0032	37	0052	20	0072	3E	0092	37	00B2	2B	00D2	29
0033	42	0053	29	0073	4A	0093	4A	00B3	37	00D3	42
0034	37	0054	20	0074	3E	0094	37	00B4	2B	00D4	29
0035	42	0055	29	0075	4A	0095	4A	00B5	37	00D5	42
0036	37	0056	2B	0076	3E	0096	37	00B6	31	00D6	29
0037	42	0057	37	0077	4A	0097	4A	00B7	3E	00D7	42
0038	37	0058	2B	0078	3E	0098	37	00B8	31	00D8	29
0039	42	0059	37	0079	4A	0099	4A	00B9	3E	00D9	42
003A	22	005A	24	007A	3E	009A	31	00BA	37	00DA	29
003B	2B	005B	2B	007B	4A	009B	3E	00BB	42	00DB	42
003C	22	005C	24	007C	3E	009C	31	00BC	37	00DC	29
003D	2B	005D	2B	007D	4A	009D	3E	00BD	42	00DD	42
003E	20	005E	29	007E	3E	009E	2B	00BE	3A	00DE	2B
003F	29	005F	37	007F	4A	009F	37	00BF	46	00DF	31
0040	20	0060	29	0080	3E	00A0	2B	00C0	3A	00E0	37
0041	29	0061	37	0081	4A	00A1	37	00C1	46	00E1	3E
0042	20	0062	42	0082	3E	00A2	2B	00C2	3E	00E2	42
0043	29	0063	37	0083	4A	00A3	37	00C3	4A	00E3	4A
0044	20	0064	42	0084	3E	00A4	2B	00C4	3E	00E4	00
0045	29	0065	37	0085	4A	00A5	37	00C5	4A		

Fig. 3. The "Mystery Song."

the crystal-controlled clock, so you can play accompaniment on another instrument if you like. In addition, each time the song plays, the program turns the trainer into a different instrument, giving the song a new character each time it repeats. This is done by selecting different harmonic combinations from the PIA. The selected harmonics appear on edge connector J1, pins H, J and K (the three lowest order bits of the A port of the PIA). They are mixed through 10k resistors and coupled to an audio amplifier. If a mating edge connector is not available, note that there are plated-through holes into which the resistors may be soldered to avoid defacing the board connector. I found that the B plus bus gives less clock noise than the ground bus when used as a ground return for the amplifier.

The program listing is given in Fig. 2. Load the program through address 0025. Begin loading your chosen song at address 0026 and follow the last note-code with a 00 to identify the end of the song. Now press E, 0000, G, and out comes music! To stop the song, use the master reset on the Microcomputer Module; sometimes using the E key to stop it will alter a few addresses in this program. I don't know why. To alter the speed at which the song is played, change the 08 at address 000D to something else.

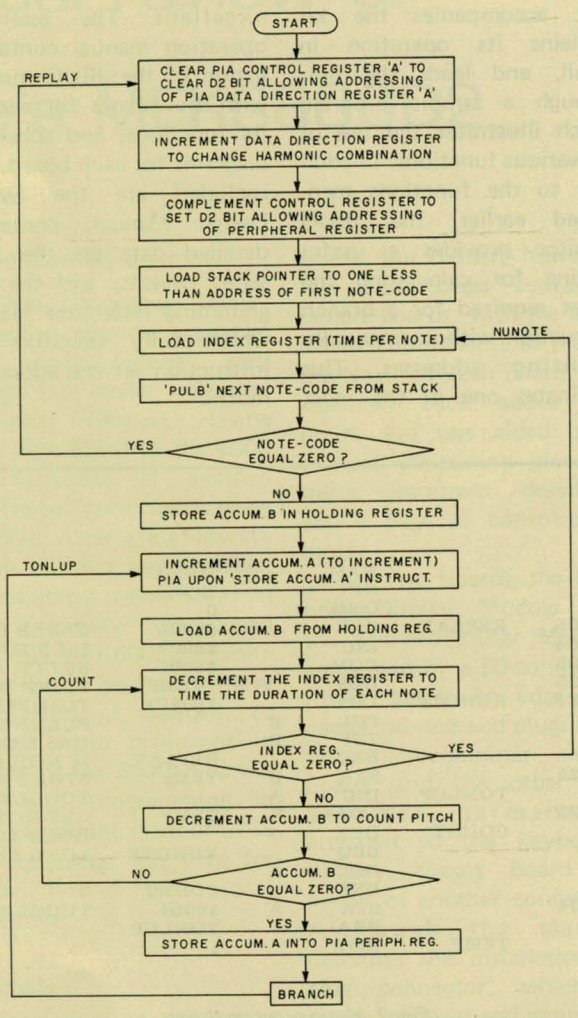
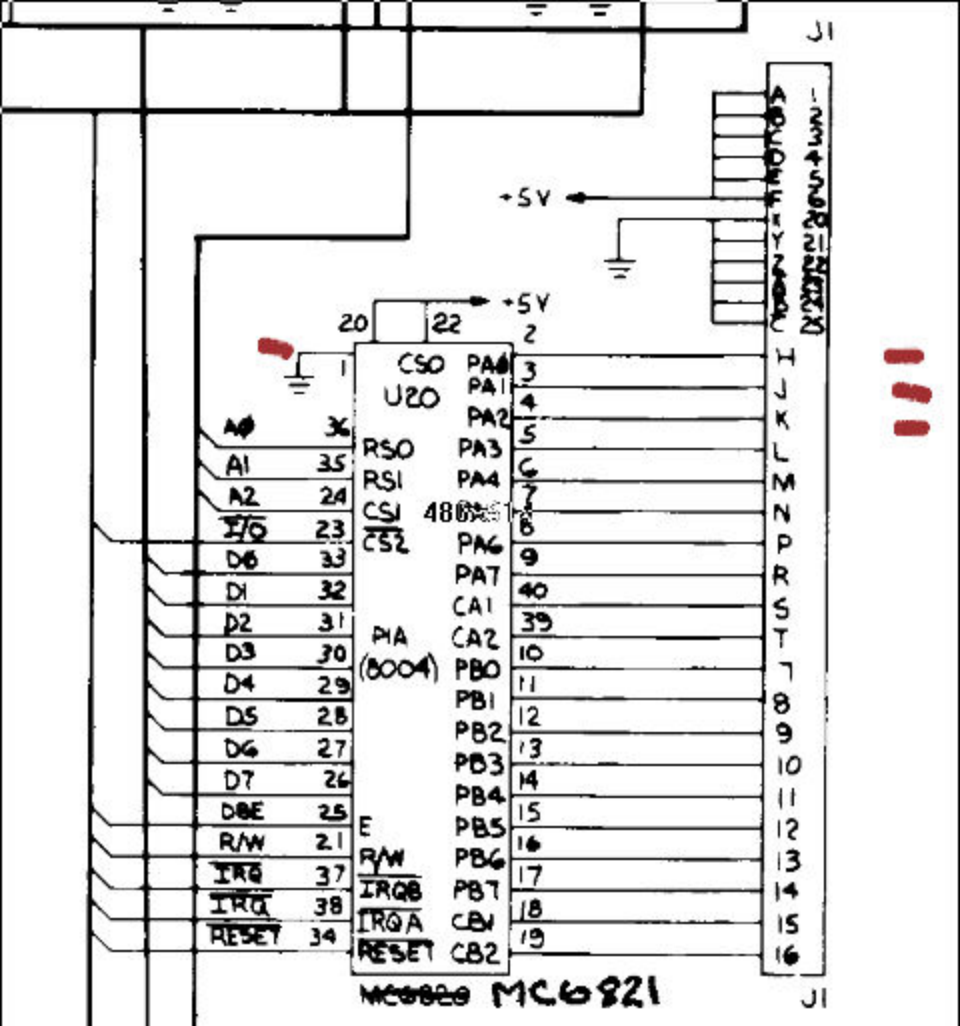


Fig. 4. Music program flowchart.

Fig. 3 provides a song sequence for you that gives the impression of two notes played together by rapidly alternating. I won't tell you the name of the song, but I'm sure you'll enjoy it. This program should be easily adaptable to other 6800 systems by following the flowchart in Fig. 4.

I realize that this may be an elementary approach to computer music for many readers, but one has to start somewhere. Playing with this trainer has sparked my interest to the point that I have my own 6800 system on order, and look forward to finding other uses for it.

My thanks to Doug Bonham and Bob Furtaw for bringing me this far. ■


J1
+5V
20
22
2
1
U20
CS0
CS1
CS2
RS0
RS1
PIA
(8004)
E
R/W
IRQB
IRQA
RESET
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
CA1
CA2
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
CB1
CB2
A0 36
A1 35
A2 24
I/O 23
D6 33
D1 32
D2 31
D3 30
D4 29
D5 28
D6 27
D7 26
DBE 25
R/W 21
IRQ 37
IRQ 38
RESET 34
3
4
5
6
7
8
9
40
39
10
11
12
13
14
15
16
18
19
H
J
K
L
M
N
P
R
S
T
7
8
9
10
11
12
13
14
15
16
MC6820 MC6821
J1