VIPER

VOLUME 1 JANUARY ISSUE 6

GAMES!

MASTERMIND
MUSIC
BASEBALL
RUSSIAN ROULETTE
COIN FLIP
CRAPS
NIM
SUB CHASE
BREAKOUT

plus

Motor Control
Studio II

$2.00

AN ARESCO PUBLICATION

EDITORIAL

Happy new year! Our only New Year's resolution was to try to get the VIPER out on time. So far, so good. I hope you enjoy this fun-and-games issue. I realized that the first five issues of the VIPER were full of pretty heavy stuff, and thought we might all relax for a month.

Our most technical article this month shows how to control small DC motors with a VIP. The author used it with a model railroad engine. Let us know what you use it for! And who will "close the loop" and use the input port to feed back positional information?

We always need good articles, but here's a special request: I have a display module with an eight-digit, latched BCD LED display. It's yours if you're willing to write a monitor which allows use of the display module in place of the TV display on the VIP - and write an article telling all of us how to use it. Anyone interested? Drop me a line for the spec sheets.

Starting on page 5 is the article I promised last issue re: Studio II. I haven't heard from very many of you regarding whether you'd like to have a special section in the VIPER on Studio II, but the few who have written are enthusiastic. Come on, folks - let us know what you think about it!

SUBSCRIPTION RATES, ADVERTISING RATES AND OTHER ESSENTIAL INFORMATION

The VIPER is published ten times per year and mailed to subscribers on the 15th day of each month except June and December. Single copy price is $2.00 per issue, subscription price is $15.00 per year (all ten issues of one volume). Dealer prices upon request. Outside of Continental U.S. and Canada, add $10.00 per subscription for postage ($1.00 for single copy).

Advertising rates are as follows:

1/4 page – 25.	3/4 page – 65.
1/2 page – 45.	full page – 85.

Less than 30% of the VIPER will be available for advertising. Please send camera ready copy in the exact page size of your ad on 8-1/2 x 11 white stock by the 1st day of the month in which you'd like the ad to appear. Photos should be glossy black & white in the exact size to be printed. Payment required with copy.

The VIPER is an Aresco Publication, edited by Rick Simpson. For information contact Editor, VIPER, P.O. Box 43, Audubon, PA 19407.

Six VIP Games
by
Carmelo Cortez

Dear Sirs:

I have six CHIP-8 games I would like to share with you. These are not the greatest of programs but I have just started using the VIP.

The first game is called Russian Roulette. Press any key to Spin and pull the Trigger. A "Click" or "Bang" will show, get ten "Clicks" in a row and you win.

The next game is a Coin Flipping program. Flip run up and the computer starts to flip a coin, and at the same time showing heads and tails on the screen, stopping at the value set in VC at 0204 (32=50). 0232 to 023A is a Delay routine by resetting the value at 0233 you can speed-up or slow-down the Coin Flipping.

To use the Craps program, press any key to roll dice, 7 or 11 wins, 12, 2 or 3 loses on first roll. The second roll must match the first to win, but if you roll a seven you lose. This program could be expanded to include on-the-screen scoring of bets.

The Nim Game is a little less graphic than most VIP games. The player may go first by pressing "F" key, any other let the VIP go first. You subtract 1, 2 or 3 from the score. The one who ends up with the last number loses!

The Sub Game is my favorite. Press "5" key to fire depth charges at the subs below. You score 15 points for a small sub and 5 points for the larger. You get 25 depth charges to start.

The last game, Breakout, is a variation of the Wipe-Off game. You have six walls and 20 balls to start. To win you must get through all walls to the top of the screen. At the end of the game the program will show the number of times you hit the walls and will show "FREE!" if you get through.

##

NEW APPLICATION NOTE FROM RCA

"Optimizing Hardware/Software Trade-Offs in RCA CDP 1802 Microprocessor Application," Microprocessor Application Note 1CAN-6704, is now available on request from RCA, Publication Services, Box 3200, Somerville, NJ 08876. The 12-page note shows how the interfacing of a keyboard and display can be accomplished in a variety of ways given differing mixtures of hardware and software. The brochure provides valuable information for anyone working on 1800-series system design.

BREAKOUT

```
0200 A2 CC 6A 06 61 03 6B 08
0208 60 00 D0 11 70 08 7B FF
0210 3B 00 12 0A 71 02 7A FF
0218 3A 00 12 06 66 00 67 14
0220 A2 CD 60 20 61 1E D0 11
0228 63 1D 62 3F 82 02 77 FF
0230 47 00 12 AA FF 0A A2 CB
0238 D2 31 65 FF C4 01 34 01
0240 64 FF A2 CD 6C 00 6E 04
0248 EE A1 6C FE 6E 06 EE A1
0250 6C 02 D0 11 80 C4 D0 11
0258 4F 01 12 98 42 00 64 01
0260 42 3F 64 FF 43 00 12 CE
0268 43 1F 12 A4 A2 CB D2 31
0270 82 44 83 54 D2 31 3F 01
0278 12 42 43 1E 12 98 6A 02
0280 FA 18 76 01 A2 CA 12 88
0288 D2 31 C4 01 34 01 64 FF
0290 C5 01 35 01 65 01 12 42
0298 6A 03 FA 18 A2 CB D2 31
02A0 73 FF 12 36 A2 CB D2 31
02A8 12 28 A2 CD D0 11 A2 F0
02B0 F6 33 F2 65 63 18 64 1B
02B8 F0 29 D3 45 73 05 F1 29
02C0 D3 45 73 05 F2 29 D3 45
02C8 12 C8 F0 80 FF FF A2 DE
02D0 63 15 62 10 D3 25 A2 E3
02D8 63 1D D3 25 12 AA EE 8A
02E0 CE 8C 8A EE 88 CC 88 EE
```

COIN FLIPPING

```
0200 6E 00 6D 00 6C 32 A2 62
0208 66 05 67 00 D6 75 A2 67
0210 66 36 D6 75 A2 F0 FE 33
0218 63 00 22 42 A2 F0 FD 33
0220 63 32 22 42 CB 01 4B 00
0228 7E 01 4B 01 7D 01 4C 00
0230 12 30 6A 10 FA 15 FA 07
0238 3A 00 12 36 7C FF 00 E0
0240 12 06 65 09 F2 65 F0 29
0248 D3 55 73 05 F1 29 D3 55
0250 73 05 F2 29 D3 55 4B 00
0258 68 01 4B 01 68 02 F8 18
0260 00 EE 88 88 F8 88 88 F8
0268 20 20 20 20
```

SUBMARINE

```
0200 A2 CD 69 38 6A 1E D9 A2
0208 A2 D0 6B 00 6C 1A DB C2
0210 A2 D4 64 3C 66 06 D4 63
0218 67 00 68 19 22 A2 22 AC
0220 48 00 12 D8 65 09 A2 D7
0228 63 00 6D 05 ED A1 63 01
0230 8E 40 ED A1 DE 51 12 3C
0238 ED A1 22 D8 A2 D4 D4 63
0240 12 42 74 FF D4 63 A2 D0
0248 DB C2 CD 04 8B D4 DB C2
0250 3F 00 12 92 A2 CD D9 A2
0258 CD 07 4D 00 79 03 79 FD
0260 D9 A2 3F 00 12 8C 43 00
0268 12 2A A2 D7 DE 51 45 1F
0270 12 86 75 02 F3 18 DE 51
0278 3F 01 12 3C 6D 1F 8D 52
0280 4D 1F 12 8C 12 92 22 AC
0288 78 FF 12 1E 22 A2 77 05
0290 12 96 22 A2 77 0A 22 A2
0298 6D 03 FD 18 A2 D7 DE 51
02A0 12 86 A2 F8 F7 33 63 00
02A8 22 B6 00 EE A2 F8 F8 33
02B0 63 32 22 B6 00 EE 6D 00
02B8 F2 65 F0 29 D3 D5 73 05
02C0 F1 29 D3 D5 73 05 F2 29
02C8 D3 D5 00 EE 01 08 7F 7C
02D0 08 3E 60 08 18 3C FF 08
02D8 A3 00 63 11 6D 0B D3 D5
02E0 A3 05 63 19 D3 D5 A3 0A
02E8 63 23 D3 D5 A3 0F 63 2B
02F0 D3 D5 63 00 12 F4 6D 0B
02F8 00 01 04 00 00 EE 00 01
0300 EE 8A 8A AA EE EF A5 A5
0308 A5 EF 7A 2A 3B 29 79 BA
0310 A2 B2 20 3A 34 3A 3C D6
0318 54 1C 0C 40 9E 25 68 0C
```

RUSSIAN ROULETTE

```
0200 65 0A 22 88 F2 0A C3 05
0208 43 04 12 32 A2 5A 61 10
0210 D1 35 A2 5F 61 18 D1 35
0218 A2 64 61 20 D1 35 64 30
0220 F4 15 F4 07 34 00 12 22
0228 00 E0 75 FF 45 00 12 46
0230 12 02 A2 69 61 10 D1 35
0238 A2 6E 61 18 D1 35 A2 73
0240 61 20 D1 35 12 44 A2 78
0248 61 10 D1 35 A2 7D 61 18
0250 D1 35 A2 82 61 20 D1 35
0258 12 58 E8 88 88 88 EE EE
0260 48 48 48 EE AA CA CA C0
0268 AA F7 55 77 55 F5 65 65
0270 55 4D 4D D5 15 15 40 D5
0278 8B 89 89 A9 DB B2 32 2A
0280 26 A2 AA AA AA 00 AA 20
0288 A2 92 66 18 67 10 D6 77
0290 00 EE 08 2A 7F 63 6B 63
0298 7F 7F 63 7F
```

NIM

```
0200 6E 23 6D 02 22 62 FC 0A
0208 4C 0F 12 18 6A 01 8E A5
0210 22 62 8D A5 4E 00 12 82
0218 68 01 E8 A1 12 26 78 01
0220 38 04 12 1A 12 18 85 E0
0228 85 85 3F 01 12 18 8E 85
0230 22 62 4E 00 12 92 69 70
0238 F9 15 F9 07 39 00 12 3A
0240 98 D0 12 50 87 80 86 D0
0248 87 65 4F 00 12 50 7D 04
0250 8D 85 3D 00 12 5A 6D 04
0258 12 0C 8E D5 22 62 6D 04
0260 12 18 FA 18 00 E0 A3 50
0268 FE 33 63 10 6B 00 F2 65
0270 F0 29 D3 B5 73 05 F1 29
0278 D3 B5 73 05 F2 29 D3 B5
0280 00 EE A2 A2 60 00 61 15
0288 D0 15 A2 A7 60 08 D0 15
0290 12 90 A2 AC 60 00 61 10
0298 D0 15 A2 B1 60 08 D0 15
02A0 12 A0 8B 89 89 A9 DB B2
02A8 32 2A 26 A6 8E 8A 8A 8A
02B0 EE EE 88 EC 28 EE
```

CRAPS

```
0200 61 08 22 58 61 12 22 58
0208 FE 0A 61 09 22 60 87 00
0210 61 13 22 60 88 00 89 80
0218 89 74 49 02 12 9C 49 03
0220 12 9C 49 0C 12 9C 49 07
0228 12 8C 49 0B 12 8C 8A 90
0230 60 40 F0 15 F0 07 30 00
0238 12 34 22 7C FE 0A 61 09
0240 22 60 87 00 61 13 22 60
0248 88 00 89 80 89 74 99 A0
0250 12 8C 49 07 12 9C 12 30
0258 A2 F0 62 08 D1 27 00 EE
0260 66 01 62 09 60 01 F0 29
0268 D1 25 F6 18 C3 07 43 00
0270 00 EE 70 01 D1 25 30 07
0278 12 66 12 64 62 09 61 09
0280 F7 29 D1 25 61 13 F8 29
0288 D1 25 00 EE A2 AC 60 00
0290 61 15 D0 15 A2 B1 60 08
0298 D0 15 12 9A A2 B6 60 00
02A0 61 10 D0 15 A2 BB 60 08
02A8 D0 15 12 AA 8B 89 89 A9
02B0 DB B2 32 2A 26 A6 8E 8A
02B8 8A 8A EE EE 88 EC 28 EE
```

A CHEAP GRAPHICS COMPUTER - CONVERT THE RCA STUDIO II

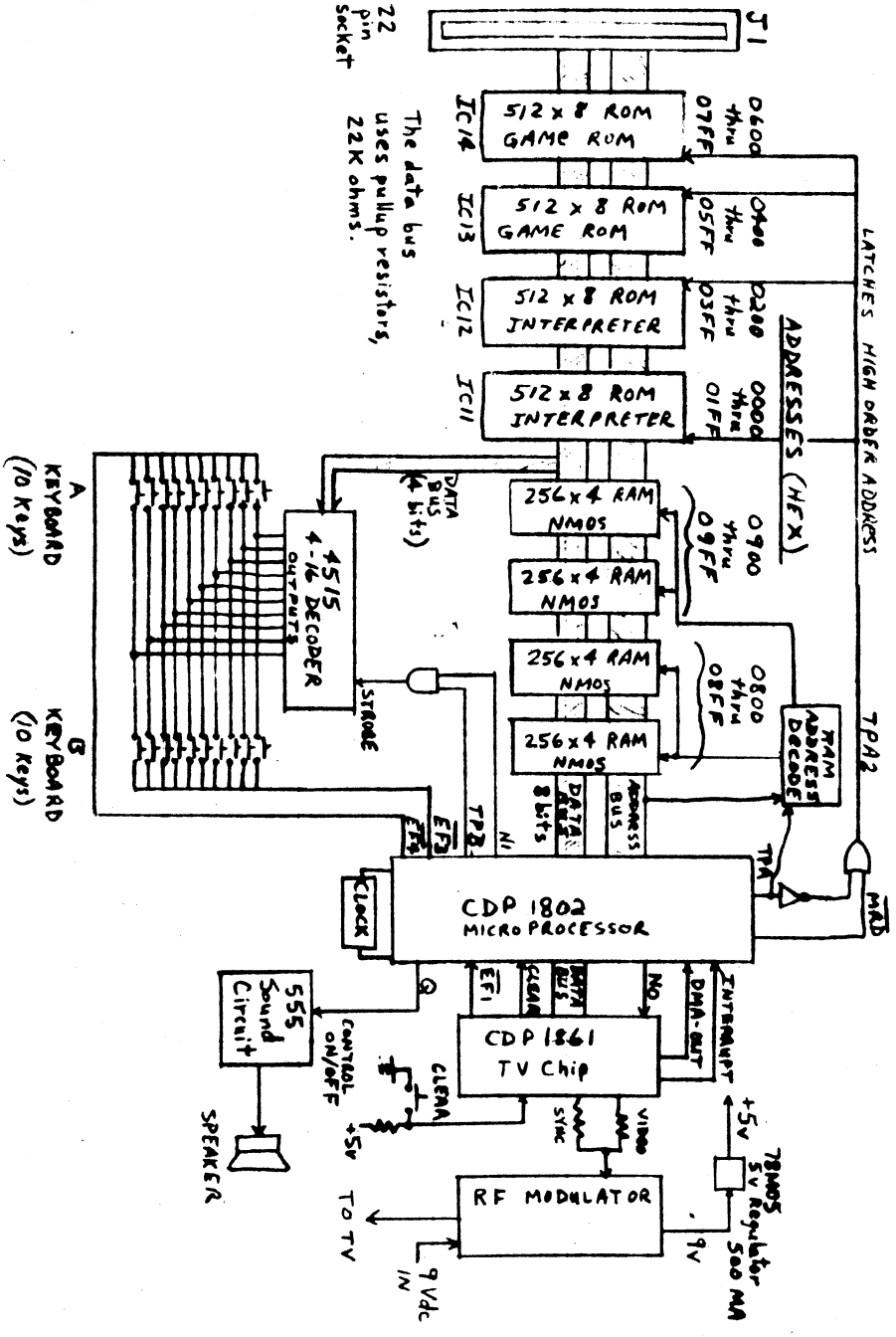
The RCA Studio II is a programmable TV game which contains a video-oriented microcomputer based on the RCA 1802 microprocessor. RCA made over 150,000 Studio IIs and sold their remaining inventory to Radio Shack. For about $59.95, you could buy one of these sophisticated TV games at Radio Shack, and for an additional $15 - $20 or so, can now build a plug-in card (similar to the extra plug-in game cards sold with the Studio II) which will allow you to enter your own machine language programs into the Studio II RAM, edit them, and run them. So for less than $80, you will have a small CMOS microcomputer which has sound output, video output with its own RF modulator, and two keyboards for input. This can be done without any internal modifications to the Studio II; all you need is the plug-in card to be described in this series, which contains a PROM and a few support ICs. If you are involved in teaching microcomputer fundamentals to a group of students, this is a cheap way to give everybody their own computer. With a copy of the 1802 User's Manual ("User Manual for the CDP1802 COSMAC Microprocessor", MPM-201A, RCA, $5.00) in front of you, it is an excellent and fun way to learn about microprocessor systems and machine language.

The Studio II home video game unit was designed to be programmable for an almost unlimited variety of games via the insertion of new ROM cartridges. The basic circuitry is shown in Figure 1. The entire circuit operates from a 5 Vdc supply at a processor clock frequency of 1.76 MHZ. Pushing "CLEAR" resets the 1802 and starts execution of the stored program in the ROM, IC11. This ROM and IC12 contain an interpreter which fetches instructions from the other 2 ROMs, IC13 and IC14. When a card is plugged into the 22 pin socket, J1, IC13 and 14 are disabled, thus causing the interpreter to now fetch instructions from the ROMs on the plug-in card. The PROM card you will build works the same way, except that its instructions tell the Studio II to accept and run your program, rather than telling it to play a specific game. The heart of the Studio II TV display system is made up of the 1802 microprocessor, the 1861 TV chip, and the 1822 RAMs. This combination of ICs bit-maps half of the RAM (hexadecimal addresses 0900-09FF) onto the TV screen, as shown in Figure 2. Thus, the game program puts objects in the display and moves them around by writing 1s to this RAM (1s = white, 0s = black). This is done very easily by a write-to-memory machine language instruction (see the 1802 User's Manual). The 1802 shifts this RAM data out to the TV display 60 times per second by using its DMA-OUT and INTERRUPT capabilities, and special features of the 1861 TV Chip.

The keyboard is scanned by latching four-bit numbers into the CD4515 (4 to 16 decoder) for the keys we want to check. As each number is latched in, this causes that particular decoder output to go low and either EF3 (A Keyboard) or EF4 (B Keyboard) is checked by the 1802 at the same time to see if it is low (If EF3 or EF4 is low, it means that key has been pressed).

To summarize, the technique for writing your own game or graphics

program is to have your program first write 1s to the appropriate addresses in the display RAM to form your display. Then make it modify the display based either on the keyboard inputs or on your own internal algorithm. The interrupt routine and the hardware will take care of the display refresh. The plug-in card to be described next month will allow you to store your game or graphics programs in the Studio II scratchpad RAM (0800-08FF) and then execute them.



SOFTWARE LIBRARY

Our second addition to the Software Library is an excellent game of BASEBALL, written by Clarke Hottel. The game fills the entire 2K of VIP memory and is very well done. A tape of the program is available for $10 from the VIPER. Here is Clarke's description of the game:

1. Set RUN/RESET to RUN.

2. "BASEBALL" appears on the screen.

3. Press A for an automatic game or press B for manual game.

4. Automatic Game: The ball is pitched automatically unless Key 5 is held depressed.

5. Manual Game (Best for two players): The ball is pitched when key 5 is depressed.

6. The bat is moved to the left with key 4.

7. The bat is moved to the right with key 6.

8. If key 4 or 6 is depressed when the ball reaches the area of the bat, a STRIKE or FOUL BALL will result.

9. If key 4 or 6 is not depressed when the ball reaches the area of the bat, a BALL or STRIKE will result.

10. If the ball hits the bat, the ball will fly at one of four speeds in one of five directions (unfortunately, a bunt occasionally looks like a line drive, but I only had a 1/4 page of space left).

11. A hit results in a BUNT, POPUP, GROUND BALL, FLY BALL, or LINE DRIVE.

12. This in turn results in a SINGLE, DOUBLE, TRIPLE, HOME RUN, or OUT.

13. Singles load bases, while doubles or triples clean the bases of all but one runner. Walks also load bases with runners, and runners can be forced home.

14. Except for a HOME RUN, as each runner crosses home plate the word RUN appears.

15. Base loading is kept track of internally, but the number of runs for BUMS or DOGS is continually updated.

16. The number of balls, strikes, and outs are continually updated and printed under BSO.

17. The team at bat is printed along with the inning. Example: D7, DOGS up, seventh inning.

18. At the end of the ninth inning (B9), the final scores are held in view and "BASEBALL" is printed on the screen as in Step 2. A new game, automatic or manual, as in Step 3 may be started.

19. Pressing 0 during a pitch cancels the game (goes to B9) and a new game may be started. (I don't like to use the run/reset switch during games).

20. Tie games end up a tie. No 10th inning (because of rain?).

21. A full ninth inning is played regardless of score.

****

More Help for VIPers

RCA has completed printing on the new VIP Users Guide, RCA Publication VIP-320. The User Guide is intended for novice VIP programmers and greatly expands on the CHIP-8 programming information contained in Chapter III of the VIP Instruction Manual. The User Guide will be packed with all new VIPs and is available to previous VIP purchasers for $5. The User Guide is available directly from the VIPER for $5 plus $1 postage.

The 48-page Guide covers initial setup of the VIP, background information on hexidecimal arithmatic and programming in general, and detailed information on the CHIP-8 instruction set.

The Guide is intended to take an intelligent but unknowledgeable new VIP owner and give him all the information he needs to become a competent CHIP-8 programmer. VIP owners who learned "the hard way" (reading the old Instruction Manual and experimenting) may still want to purchase a copy of the Guide for use by others in their family or acquaintances. The VIP is so portable that it can be lent out over a weekend along with a copy of the Guide, to give a friend a taste of what microcomputing is all about.

The Guide was written by VIPER "staff member" Terry Laudereau. (She will be happy to autograph any copies ordered from the VIPER!) It focuses exclusively on programming and contains no information on hardware or machine language programming. For this you'll still have to rely on the VIPER.

A MOTOR CONTROLLER FOR THE VIP

by
Steven Medwin

INTRODUCTION

One of the first projects I wanted to do with my VIP was develop a motor controller that would allow use of the video display along with all CHIP-8 instructions. More specifically I wanted to build a pulse width modulated controller for small D.C. motors. This interface would allow software control of speed and direction, would handle up to four motors independently, and would be suitable for model railroads, robots or any other control application. The ability to control the interface with CHIP-8 and use the display simultaneously would make it flexible and easy to use.

BACKGROUND

The simplest and cheapest way to control a motor is to output pulses with a constant frequency, but variable width. The pulse width, or time the pulse is high, determines the average output voltage. For instance, if "FF" was sent to VIP's output port 60 times a second, for 1/60 of a second each time, the average output voltage would be 5 volts. However if each pulse was "FF" for half the period (1/120 of a second) and "00" for the other half, then the average output voltage would be 2.5 volts. By varying the time the pulse is high relative to the period of the whole pulse, the average output voltage can be precisely set.

To drive a D.C. motor this way, an amplifier is the only additional hardware needed. A motor should never be driven directly from VIP's output port. In order to control both speed and direction, two bits of information are needed. The circuit reproduced below will amplify these two bits to drive up to a 15 volt D.C. motor drawing at most 1 amp. All that is needed now is a way to generate the pulses.

HOW IT WORKS

Basically the display interrupt routine is used as a constant frequency source to output certain bytes. These bytes in turn generate the desired pulses.

The display routine interrupts the VIP 60 times a second, for about 1/120 of a second each time. Within this routine (for the one page display) there is a loop that is executed 32 times on each interrupt. If an output instruction is inserted in this loop, then 32 different bytes can be outputted in 1/120 of a second. By carefully setting these bytes, very fine control of the pulse width, and output voltage, is possible.

Since each byte is latched on output, the last byte sent determines the status until the next interrupt (a time span of 1/120 of a second). For short pulse widths ($<$1/120 sec.) this last byte, or the "flag" byte,

is set to "00". For longer pulse widths, the flag byte is set so the appropriate bits are high between interrupts. For any specific speed within each range, the rest of the output bytes are set for the exact pulse width needed.

These output bytes are stored in a table in memory. They can be modified at any time by a machine language subroutine ("update") called from a CHIP-8 program. Since the interrupt routine is always outputting bytes (and controlling the display), the timing to update the table is not critical. In addition all CHIP-8 instructions, including those controlling the display are still implemented.

Because there are eight bits per byte, eight different variable width pulses can be generated, all synchronized at 60 cycles per second. This means that four bi-directional motor drives (using two bits each) can be controlled by a CHIP-8 program or three drives and two relays (using one bit each) and so on. (Note: This technique shouldn't be used if only relays are being controlled, since there are simpler ways.)

HARDWARE

The only interface needed is a suitable amplifier. A circuit designed to drive small D.C. motors first appeared in Byte Magazine (July 1978, P. 72). I modified it to provide up to one amp at 15 volts since that is what my model railroad needs. This motor driver needs two signals; one turns the amplifier on while the other determines the direction. These signals come directly from the VIP output port. The Byte article contains a complete description of this circuit and how it works. If you build this circuit, make sure you heat sink all power transistors. Also the integrated circuit (7426) can't handle more than 15 volts: Use a heat sinked voltage regulator (like Radio Shack 7815 15 volt regulator IC) if your power supply provides more than this.

Below is a table of the logic needed to drive this amplifier:

Function	Bit 0 (VIP PIN M)	Bit 1 (VIP PIN N)
OFF	0	0
Forward	1	0
Reverse	1	1

SOFTWARE

The necessary software consists of the following four parts:

- CHIP-8 Modification
- New Display Interrupt Routine
- "Update" Machine Language Subroutine
- CHIP-8 Program

The only modification to the CHIP-8 interpreter consists of changing the address of the interrupt routine from 8146 to 0205.

The interrupt routine is basically the same as the one in the ROM operating system except for the following changes. In the first part of the routine, a data pointer (R7) is initialized to the top of the output table. Then in the timing loop that follows, two instructions are changed so a byte is outputted, and the pointer incremented, once each loop. Since the loop is executed 32 times, 32 bytes are outputted during each interrupt.

It is the job of the update subroutine to set and change those output bytes. The update routine modified the output table depending on four CHIP-8 variables (V0 to V3). These variables determine which bits will be on and for how long. Below is a table describing the function of each:

V0=Byte Set:	Determines which bits will be set high.
V1=Byte Mask:	All other bits are masked.
V2=Flag:	Value depends on speed range - =0 if low; =byte set if high range.
V3=Speed:	Any hex digit.

In order to control the interface described above, connected at bits 0 and 1, the following values are used by the update routine:

Function	V0	V1	V2	V3
Off (initialize)	00	00	00	00
Foward, Low	01	FC	00	00-0F
Forward, High	01	FC	01	00-0F
Reverse, Low	03	FC	00	00-0F
Reverse, High	03	FC	03	00-0F

To control a relay, driven by an amplifier (see Aug. '78 VIPER) at bit 3, use the following variable values:

Function	V0	V1	V2	V3
Relay Off	04	FB	00	00
Relay On	04	FB	04	0F

This control is not included in the CHIP-8 program described below, but could be easily added.

The CHIP-8 Program is a demonstration of one way to generate the values needed by the update routine. The program accepts two hex key inputs: The first sets the direction and speed range and the second sets the exact speed. The first digit is used as a pointer to a table of "update values." Then V0, V1 and V2 are set to the appropriate values. The second digit (speed) is doubled and stored in V3. This is done because there are twice as many possible speeds (32) as hex keys. Since V0 through V3 are now set (the timing wasn't critical) the update subroutine is executed. The video display is then cleared and the two key inputs displayed. Then the program waits for two more inputs.

NEXT STEP

The CHIP-8 program described above could really be a subroutine called from a much larger program. The speed could be determined by a few inputs or conditions. Also the display could be used to show the motor in its environment: The model railroad layout or the room a robot is exploring. With a little bit of work, this idea could even be used with a two-page display.

MEMORY USAGE:

```
0200-022D   Display Interrupt Routine
0230-0257   Update Subroutine
0260-029B   CHIP-8 Program
0400-041F   Output Table (in 2K system, 04=06;
              in 4K system, 04=0E)
```

REGISTER ASSIGNMENTS:

```
R(7)  = Output Table Pointer (interrupt routine)
R(C)  = CHIP-8 Variable Pointer
R(D)  = Output Table Pointer (update routine)
R(E).0= Byte Set
R(E).1= Byte Mask

V0    =Byte Set
V1    =Byte Mask
V2    =Flag
V3    =Speed
V8,V9 =x,y Coordinates for Display
VA,VB =Input Keys
```

++

Here are some corrections to programs published in the VIPER.

1. "EDITOR" by Sam Hersh (Oct.) - To eliminate display jitter:

```
Change:  0293 from C4 to ED
  "      032D  "   C4 to EC
```

2. "VIP ARITHMETIC" (Aug. '78, P. 3) - Should be:

```
01F5 06 F6 F6 F6 F6 30 2F
```

Steve Medwin

CHIP 8 - Program Sheet

Page 1

Program Name: VIP MOTOR CONTROLLER

Date 12/78

Address	Code	Comments
0		
2		
4		
6		
8		CHIP-8 MODIFICATION
000A	02 B1	} CHANGE DISPLAY INTERRUPT POINTER
C	F8 05	}
E		
0200	1260	GO TO CHIP-8 INSTRUCTIONS
2	7A42	DISPLAY INTERRUPT ROUTINE
4	70 22	
6	78 22	
8	52 C4	
A	F8 00	}
C	A0 A7	} 0YOO → R(7)
E	9B B0	}
0210	92 B7	}
2	80 E7	7 → X
4	63 20	M(R(X)) → BUS , R(X)+1
6	A0 E2	
8	20 A0	
A	E2 20	
C	A0 3C	
E	12 19	
0220	98 32	
2	27 AB	
4	2B 8B	
6	B8 88	
8	32 02	
A	7B 28	
C	30 03	
E	00 00	

CHIP 8 - Program Sheet

Page 2
Date 12/78

Program Name: VIP MOTOR CONTROLLER

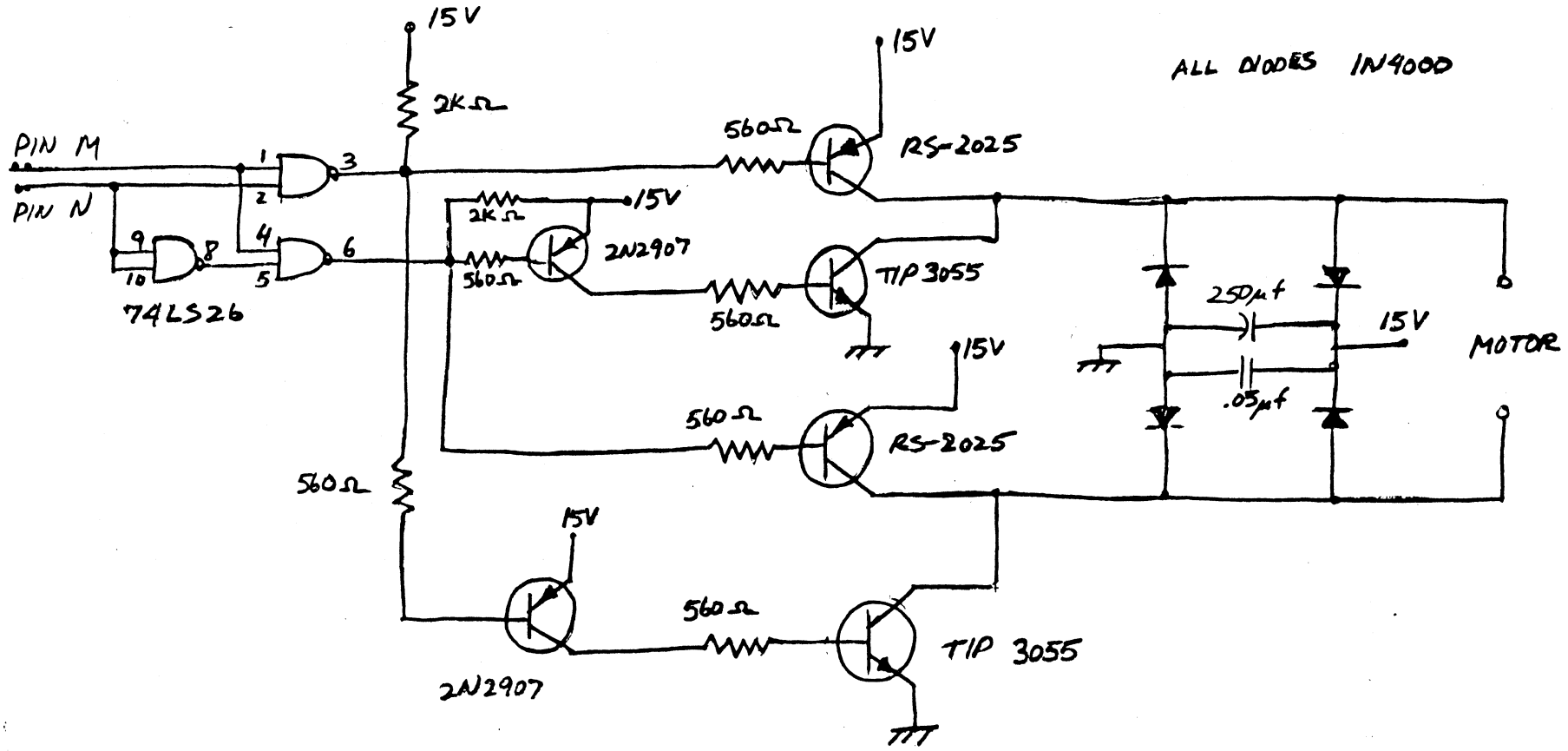
Address	Code	Comments
0230	97 BC	UPDATE SUBROUTINE
2	BD F8	0YF0 → R(C)
4	F0 AC	04 → R(D).1
6	4C AE	V0 → R(E).0
8	4C BE	V1 → R(E).1
A	F8 1F	0Y1F → R(D)
C	AD ED	INITIALIZE
E	9E F2	
0240	5D 4C	SET FLAG BYTE = V2
2	F1 5D	
4	2D 9E	
6	F2 5D	CLEAR TABLE
8	8D 3A	
A	44 4C	BRANCH TO RETURN
C	AD 32	~~CHECK~~ IF SPEED = 0
E	57 1D	
0250	2D 8E	
2	F1 5D	SET TABLE
4	8D 3A	
6	50 D4	RETURN
8		
A		
C		
E		CHIP-8 PROGRAM
0260	AF00	USE A700 FOR 2K SYSTEM — INITIALIZE ALL VARIABLES TO ZERO
2	FF65	
4	0230	DO UPDATE SUBROUTINE
6	00E0	CLEAR SCREEN
8	6800	
A	FA29	
C	D895	DISPLAY INPUT KEYS
E	7806	

CHIP 8 - Program Sheet

Page 3
Date 12/78

Program Name: VIP MOTOR CONTROLLER

Address	Code	Comments
0270	FB29	(DISPLAY CON'T)
2	D895	
4	FA0A	ENTER VA
6	6F03	BRANCH TO INITIALIZE IF VA > 03
8	8FA5	
A	3F01	
C	1260	
E	FB0A	ENTER VB
0280	A290	SET I
2	FA1E	
4	FA1E	
6	FA1E	
8	F265	SET V0, V1, V2, V3
A	83BE	V3=VBx2
C	1264	BRANCH TO UPDATE
E		
0290	01FC	UPDATE ROUTINE DATA
2	0001	
4	FC01	
6	03FC	
8	0003	
A	FC03	
C		
E		
0		HOW TO RUN CONTROLLER PROGRAM:
2		1) WHEN START PROGRAM, SHOULD SEE 2 ZEROS ON SCREEN
4		2) ENTER 2 DIGITS: FIRST ONE SETS RANGE AS FOLLOWS -
6		0 FOR LOW FORWARD, 1 FOR HIGH FORWARD,
8		2 FOR LOW REVERSE, 3 FOR HIGH REVERSE. (ANY OTHER KEY RESETS OUTPUT TO 0
A		SECOND KEY SETS EXACT SPEED FROM 0 → F.
C		3) 2 DIGITS DISPLAYED AND PULSES OUTPUTTED
E		4) TO CHANGE SPEED, JUST ENTER 2 MORE DIGITS.



VIP MOTOR CONTROLLER SCHEMATIC

S. MEDWIN 12/78

MASTERMIND
by
Robert Lindley

I have programmed two versions of the game Mastermind. This game is distributed by Invicta Plastics, Suite 940, 200 - 5th Ave., New York, NY 10010, and is available most places where toys and games are sold. For complete details of the game, please refer to their instructions. The information given here refers to this particular VIP implementation.

The two versions are essentially the same except that the first has a four number code and the second has a five number hidden code selected at random. In the four number version, the digits one through six are used and in the five number version the digits zero through seven are used. While the game is running, the other hex keys, except key F, have no effect. Key F is used when you change your mind and want to change your input. This key erases the current partial entry.

The game starts by displaying a series of dashes arranged in rows. As the game progresses, the player attempts to deduce the hidden code by replacing the dashes with digits entered via the hex keyboard. Each time a hex key is used, the selected digit replaces a dash in one vertical column. This vertical column is one guess of the ten allowed to deduce the hidden number. When the bottom dash in any column is replaced by a digit, that try is immediately scored. This score appears below the current column. If any digit in the column exactly matches the hidden number digit in the same row, a broken bar will appear. When four or five (one for each row in the game) appear, the hidden number has been deduced and it will be revealed at the right end of the screen. If any digit in the column matches a hidden digit, but in an incorrect row, a white bar will appear. Note that the scoring is across all rows. For example, if there are two fives in a column and one of them is in the correct row and there is only one five in the hidden numbers, one broken bar will appear. When all allowed ten tries have been used, the hidden number will be revealed.

The four row version uses standard CHIP-8 and the five row version uses two page CHIP-8 described in issue three of the VIPER.

==========

----------CORRECTION----------

Errors seem to creep into even the shortest of programs. In the Joystick program on P. 14 of Issue #5, the "S" in locations 16 and 1C should be a "5".

LABEL	ADDR.	INSTR.	COMMENTS
	2ØØ	A2FC	Point to dash
	2	6DØØ	Display X to upper left
	4	6EØØ	Display Y to upper left
DASH LOOP	6	DDE3	Show dash
	8	7EØ6	Move Y down
	A	3E18	Stop at 4th dash
	C	12Ø6	Continue down
	E	7DØ6	Move X across
	21Ø	3D3C	Stop at 10th dash
	2	12Ø4	Continue across
	4	A3F4	ANS (answer) location
	6	23ØØ	Get 1st random #
	8	23ØØ	" 2nd " "
	A	23ØØ	" 3rd " "
	C	23ØØ	" 4th " "
	E	6DØØ	Display X to upper left
	22Ø	6EØØ	" Y " " "
	2	6CØØ	1Ø try loop counter
TRY LOOP	4	A3F4	ANS location
	6	F365	get ANS
	8	F355	Put in working COPY of ANS
	A	6BØØ	Input key counter
INPUT KEY	C	FØØA	Key → VØ
	E	3ØØF	Test for cancel signal - Key F
	23Ø	124A	Bypass input cancel
CANCEL LOOP	2	4BØØ	Cancel loop counter test
	4	122C	Exit - all done
	6	7EFA	Back up display Y
	8	A3FC	IN (input) save location
	A	7BFF	Back up input loop count
	C	FB1E	Loop count is array offset
	E	FØ65	Recover previous input

LABEL	ADDR.	INSTR.	COMMENTS
	24Φ	FΦ29	
	2	DDE5	Erase previous input
	4	A2FC	dash location
	6	DDE3	Return dash to display
	8	1232	Continue cancel loop
SAVE INPUT	A	6AF9	-7
	C	8AΦ4	Add to key value
	E	3FΦΦ	Test key for <7
	25Φ	122C	key>6 → try again
	2	4ΦΦΦ	Test key=Φ
	4	122C	key=Φ → try again
	6	A2FC	dash location
	8	DDE3	Erase dash
	A	A3FC	IN location
	C	FB1E	Loop count is IN array offset
	E	FΦ55	Input key → IN
	26Φ	FΦ29	
	2	DDE5	Show input value
	4	7EΦ6	Move down
	6	7BΦ1	Incr. input loop count
	8	3BΦ4	Test for 4 inputs
	A	122C	Get another input
	C	64ΦΦ	V4 is signal for dotted bar
	E	68ΦΦ	V8 correct count for this try
	27Φ	6BΦΦ	"Test for correct" loop counter
DOTTED LOOP	2	A3FC	IN location
	4	FB1E	Loop count is array offset
	6	FΦ65	Get next IN #
	8	82ΦΦ	Move to V2
	A	A3F8	COPY location
	C	FB1E	array offset
	E	FΦ65	Get COPY #

LABEL	ADDR.	INSTR.	COMMENTS
	28Φ	83ΦΦ	COPY # → V3
	2	23ΦE	Call BAR subroutine
	4	A3FC	IN location
	6	FB1E	array offset
	8	8Φ2Φ	
	A	FΦ55	Save (possibly modified) IN #
	C	A3F8	COPY location
	E	FB1E	array offset
	29Φ	8Φ3Φ	
	2	FΦ55	Save (possibly modified) COPY #
	4	7BΦ1	Incr. loop count
	6	3BΦ4	Are all 4 rows tested?
	8	1272	continue testing
	A	38Φ4	If all correct, end of game
	C	12C2	Branch to white bar test
SHOW ANSWER	E	6EΦΦ	Display to top - Show answer
	2AΦ	6D3C	Display to right end
	2	69ΦΦ	Answer loop count
ANS LOOP	4	A3F4	ANS location
	6	F91E	array offset
	8	FΦ65	Get next ANS #
	A	FΦ29	
	C	DDE5	Show ANS #
	E	7EΦ6	Move down
	2BΦ	79Φ1	Incr. loop count
	2	39Φ4	Stop at 4 #
	4	12A4	Get more
	6	6Φ2Φ	Blink time
	8	FΦ15	Set timer
	A	FΦΦ7	Test timer
	C	3ΦΦΦ	" "
	E	12BA	" "

LABEL	ADDR.	INSTR.	COMMENTS
	2CØ	129E	Blink answer - end of loop
WHITE BAR TEST	2	64Ø1	V4 has white bar signal
	4	6AØØ	IN loop index - outer loop
WHITE BAR LOOP	6	6BØØ	COPY loop index - inner loop
	8	A3FC	IN location
	A	FA1E	array offset
	C	FØ65	Get IN #
	E	82ØØ	
INNER LOOP	2DØ	A3F8	COPY location
	2	FB1E	array offset
	4	FØ65	Get COPY #
	6	83ØØ	
	8	23ØE	Call BAR subroutine
	A	A3F8	COPY location
	C	FB1E	array offset
	E	8Ø3Ø	
	2EØ	FØ55	Save (possibly modified) COPY #
	2	7BØ1	Incr. COPY loop counter
	4	3BØ4	Test for end of loop
	6	12DØ	continue loop
	8	7AØ1	Incr. IN loop counter
	A	3AØ4	Test for end of loop
	C	12C6	continue loop
END OF TRY	E	7DØ6	Move to next column
	2FØ	6EØØ	Move to top row
	2	7CØ1	Incr. Try loop count
	4	3CØA	Is this the 10th try?
	6	1224	Process next try
	8	129E	Done - Go to show answer
	A	9ØFØ	Dotted bar / White bar
	C	ØØØØ	Dash
	E	6ØØØ	"

LABEL	ADDR.	INSTR.	COMMENTS
RANDOM # SR	300	C007	Get random # 0 → 7
	2	4000	Test for 0
	4	1300	Do not use 0
	6	4007	Test for 7
	8	1300	Do not use 7
	A	F055	Put random # in arroy
	C	00EE	Return
BAR SR	E	5230	V2=V3 ?
	310	00EE	NO - Just return
	2	A2FA	Dotted bar location
	4	3400	0 means use dotted bar
	6	A2FB	White bar location
	8	DDE1	Show bar
	A	7E02	Move down
	C	7801	Incr. correct count
	E	620E	Cross out IN #
	320	630F	Cross out COPY #
		00EE	Return
Storage Arrays			
	3F4	FOUR LOCATIONS	ANS - answer
	3F8	FOUR LOCATIONS	COPY - working copy of answer
	3FC	FOUR LOCATIONS	IN - column input

VIP Music
by
Carmelo Cortez

I have a music program I would like to share with all your readers. The program was taken from an article in Popular Electronics.* I modified it for the VIP and added a few more notes, plus a few bits of hardware.

The program starts at Loc. 0000, and takes less than 1 page of memory. The song notes start at Loc. 0046 and can continue to extra pages of memory.

The music program requires two bytes for each note. The first byte is the duration, the second byte is the pitch.+ You also can change the tempo by changing the byte at Loc. 0011, and the rest time between notes by changing the byte at Loc. 003E. (Eg; 01 or 02 will give a smoother sounding song, but you will have to decrease the tempo to 19 or 1B to compensate for the increased speed.)

I have also included a circuit that uses a transistor, a speaker, and three resistors. (Simple huh?)

And lastly I have included a simple circuit to display the note being played. It uses LEDs and 8 resistors. This circuit can be left out if desired. The readout is binary and can be read in Hex, (Eg; 1001 0101=95). The ones being light and zeros dark (of course).

This and the speaker circuit can be built on a Radio Shack plug in P.C. board (C.T. No. 276-154). This board will plug into the VIP I/O port.

Oh yes, if you end a song with an 00 byte, it will start over.

*"How to Upgrade a Basic ELF Microcomputer" by Edward M. McCormick, Feb. 1978
+Fig. 3 is a Table of Notes with Durations + Pitch

Table 1

```
0000 E5 F8 00 B5 F8 46 A5 F0
0008 32 01 A8 15 63 25 F0 A7
0010 F8 10 A9 87 FC B4 33 20
0018 31 1D 7B 30 20 7A 30 20
0020 87 FF 01 3A 21 89 FF 01
0028 A9 3A 33 88 FF 01 A8 3A
0030 10 30 3B C4 C4 30 37 30
0038 39 30 13 7A 15 F8 0E B3
0040 23 93 3A 40 30 07
0046 First Note of Song
```

Mystery Song - Table 2

```
0046 181F 0A27 0A27 181F 241F 0C1F 0D1B 0F17
0056 2115 4115 0A27 0A24 181F 181F 181F 0D1B
0066 0C1F 1524 2B24 1427 181F 1033 1427 122D
0076 2B24 122D 4133 164C 0B4C 054C 0C1F 0A27
0086 0A24 181F 241F 0C1F 0D1F 0F17 2115 4115
0096 0A27 0A24 181F 181F 181F 0D1B 0C1F 1524
00A6 2B24 1427 181F 1033 1427 122D 2B24 122D
00B6 6133 0B46 2115 2412 2415 1E17 2115 1B1B
00C6 371B 2115 2412 2115 1E17 2115 521B 2412
00D6 2910 2412 2214 2412 1E17 1E17 1E17 0F17
00E6 1015 2412 2214 1E17 1B1B 181F 0B4C 241F
00F6 0C1F 0A27 0A24 181F 241F 0C1F 0D1B 0F17
0106 2115 4115 0A27 0A24 181F 181F 181F 0D1B
0116 0C1F 1524 2B24 1427 181F 1033 1427 122D
0126 2B24 2412 6215 0B4C 00
```

Table 3

.Durations and Notes

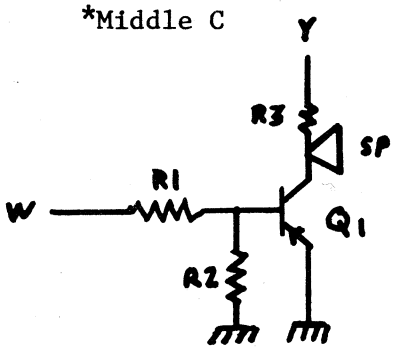
Notes	1	1/2	1/4	1/8	1/16	Pitch
E	A4	52	29	14	0A	10
D#	9C	4E	27	13	09	11
D	93	49	24	12	09	12
C#	8B	45	22	11	08	14
C	83	41	21	10	08	15
B	7B	3D	1E	0F	07	17
A#	75	3A	1D	0E	07	19
A	6E	37	1B	0D	06	1B
G#	68	34	1A	0D	06	1D
G	62	31	18	0C	06	1F
F#	5D	2E	17	0B	05	22
F	57	2B	15	0A	05	24
E	52	29	14	0A	05	27
D#	4E	27	13	09	04	2A
D	49	24	12	09	04	2D
C#	45	22	11	08	04	30
*C	41	20	10	08	04	33
B	3E	1F	0F	07	03	37
A#	3A	1D	0E	07	03	3B
A	37	1B	0D	06	03	3F
G#	34	1A	0A	06	03	43
G	31	18	0C	06	03	47
Rest	2D	16	0B	05	02	4C

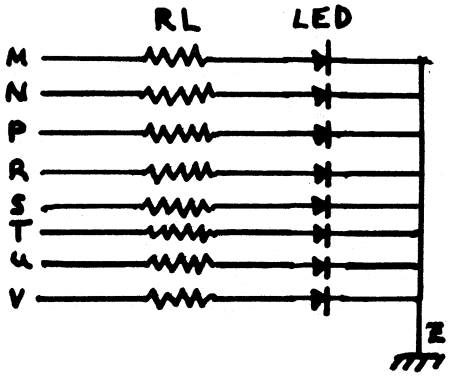
1/4 middle C note

would be - 1033

Duration Pitch

*Middle C





R1=10K R2=47K R3=47 ohm SP= 8 ohm speaker RL=470
LED = subminiature LEDs Q1= 2N2222

An Advertisement

ARESCO is now an authorized dealer for VIP products. This means you can order VIP peripherals using your VISA or Master Charge card. Furthermore, if you live in the States, we can accept C.O.D. orders if you're willing to pay the shipping costs. We won't charge your credit card account - or cash your check - until the day we're ready to ship your order. But, like everyone else, we have to wait our turn for delivery from RCA - so we can't promise any better a delivery schedule than RCA promises - and our schedule will be met in the same timely fashion as RCA's.

If you've been holding off on your order, waiting for off-the-shelf delivery, you'll wait quite a while, since orders are pouring in from all over the country. If you've been holding off because you don't want your money tied up, you can now order through ARESCO and not have your money tied up - since we won't charge your account until the day we ship. Sound fair?

ARESCO doesn't warranty RCA products, however, since RCA is well-known for the service it provides for all its products, their guarantee is worth twice what anyone else's is. If your VIP peripheral fails to work as you expect, you just return it to RCA - the return address is provided in the package with the product.

You can get your name on the waiting list without putting money "up front" by ordering through ARESCO. We need the following information - and then all you do is wait for RCA to deliver to us:

- Your name
- Your street address (we can't ship UPS to a post office box)
- Your city, state, and zip code
- Your name and number (see pages 7, 8, & 9 of Issue #3, VIPER) of the items you want to order
- Your credit card number, expiration date, and, if it's Master Charge card, the interbank number OR a statement that you want C.O.D. delivery and are willing to pay shipping costs
- A statement saying that you are aware that delivery may take more than 30 days (we need this to keep out of trouble for mail fraud, since the law says 30 days is maximum and we can't promise delivery in less than 30 days if RCA can't).

STUDIO II OWNERS!

How fast would you run to purchase a powerful microcomputer for only $50.00 - with all these features!

* Dual Keyboard
* 512 Bytes of RAM
* Interval Timer
* TV display controller
* 256 Bytes of ROM
* Audio "Beeper"

* ROM Monitor - to allow you to

- Examine Memory
- Start Program Execution
- Access TV display and Keyboard reading sub-routines for use in your own 1802 machine language programs
- Load Memory
- Interrupt program Execution
- Write your own Studio II games

No modification to your Studio II is required - the ARESCO Studio II Conversion Kits contain everything you need to build your own ROM cartridge. The Kits are available in a variety of forms - to suit your budget and your specific needs:

KIT A: Schematics, ROM program listing, operating instructions and sample programs - $5.00. **(to be published in the VIPER)**

KIT B: All the items in KIT A plus the printed circuit board for $15.00. Or get the PC board by itself for $10.00.

KIT C: All the items in KITS A & B plus the pre-programmed ROM containing the monitor program for $30.00. Or just the ROM by itself for $15.00.

KIT D: All items in KITS A, B, & C fully assembled and tested and guaranteed for 90 days - for $50.00.

Offer good while supply lasts - contact ARESCO, P.O. Box 43, Audubon, PA 19407. Sorry, no C.O.D. or billing available. MC/VISA/BAC okay. Specify Kit letter. Please give street address, not P.O. Box, for UPS shipping within four weeks of receipt of your order.

ARESCO
P.O. Box 43
Audubon, PA 19407

YES! I'd like to subscribe to the VIPER and receive all ten issues of this year's volume! I enclose $15.00 in full payment.

Name _____
(Please print or type)

Address _____

City _____ State _____ Zip _____

Cash, Check, Money Order Enclosed _____

MC/VISA/BAC Number _____ Exp. Date _____

MC Interbank No. _____

Required Credit Card Signature _____

☐ You may let other VIP owners in my area know I have a VIP, so they can contact me.

☐ I am interested in forming/joining (circle one) a VIP Users group

☐ I'd like to see articles in the VIPER about:

_____

_____

_____

_____

MAIL TO: VIPER; P.O. Box 43; Audubon, PA; 19407