BY JAMES J. BARBARELLO

ATARI JOYSTICK INTERFACE

If you've ever played video games in an arcade or on another microcomputer and then tried them on the CoCo, chances are you were frustrated with Radio Shack's joysticks. They just aren't sensitive enough for some of the fast, precise action necessary in some games. And they often don't hold up well in the intense heat of play.

After the fifth or sixth joystick repair job, I knew there had to be a better way. The easiest alternative would be to buy one of the many available Atari-to-CoCo joystick interfaces, but these cost about $20 per joystick, and you still have to buy the joysticks.

Since Atari joysticks are available from a number of sources for well under $10 a pair, why not build a simple interface myself and pocket the difference?

That thought resulted in the CoCo Joycard—a simple construction project that interfaces an Atari joystick to the CoCo. In addition, the Joycard provides a jumper-selectable option for either single-shot or rapid firing. Best of all, the total cost for one joystick and interface should be under $10.

Tired of joysticks that don't stand up to your style of play? Adapt the inexpensive Atari sticks.

Number Please

Take a look at the Radio Shack joystick. Figure 1 shows one of the two joystick potentiometers (pots). Note that one end is attached to 5 volts and the other to ground (0 volts). As you move the pot's wiper, the output voltage varies between the two voltage extremes. When the wiper is in the middle of the pot (at rest) the output voltage is one-half the maximum (2.5 volts).

Figure 2 shows a circuit that forms the basis of the interface. First, when the joystick is at rest, the output must be 2.5 volts. This occurs with switches S1 and S2 open. Closing S1 raises the output to 5 volts (indicating left or up, for instance). Reopen S1 and close S2 to drop the output to 0 volts (indicating right or down).

Figure 3 shows a representation of an Atari joystick. As you move the stick to a vertical or horizontal position, one switch closes (moving in a diagonal position closes two adjacent switches). But notice that all switches (including the fire switch) are connected together on one side.) Therefore, you can't use the joystick directly in our circuit of Fig. 2, since the common points for the up/down, left/right, and fire switches must be isolated from each other.

Enter the circuit in Fig. 4. This complete schematic of the CoCo Joycard and joystick contains the 4016 CMOS Quad Analog switch, an integrated circuit that has four separate,

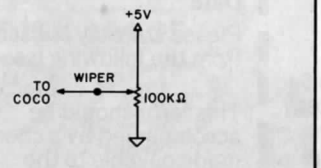

Figure 1.

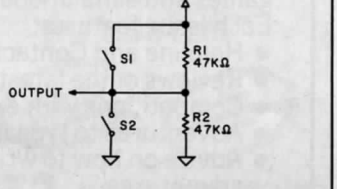

Figure 2.

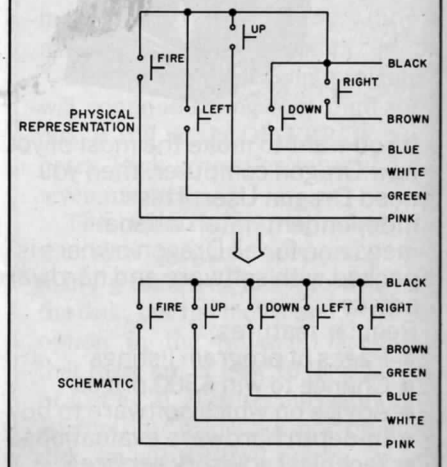

Figure 3.

electrically controlled transmission gates (switches). Each switch has an input (pin 1 for the first switch), an output (pin 2), and a control (pin 13).

When no voltage reaches the control pin, the resistance between input and output becomes so large that you can consider it an open circuit. When it receives 5 volts, the resistance drops to below 200 ohms (low enough to consider it a short circuit). You can consider each gate a single-pole, single-throw switch that is activated by a control signal.

Figure 4 contains two circuits that are similar to Fig. 2. In Fig. 4, however, transmission gates replace the two switches. R1, R3, R5, and R7 provide the 0-volt reference when the joystick switches are open.

When a joystick switch is closed, 5 volts from the common line pass through it to its companion transmission gate. That gate shorts out its corresponding resistor, changing the output voltage provided to the CoCo.

When the stick is at rest, all joystick switches are open, and all transmission gates act like open switches. Thus, no resistors are shorted, and the CoCo receives a midposition indication.

The remaining portion of the circuit is for the fire switch. The CoCo expects to see an open circuit or 5 volts when the fire switch is open. When you press the fire switch, the CoCo expects 0 volts.

IC2 is half of a 4011 CMOS Quad two-input NAND gate. As wired, the two NAND gates form a "gated" oscillator operating at about 10 Hz (10 cycles per second). As long as pin 1 is referenced to ground (as when the joystick fire button is open), the oscillator does not operate and the output from pin 3 is at 5 volts.

When you press the fire button, 5 volts pass to pin 1 and the oscillator begins running. The output then varies rapidly between 0 and 5 volts, simulating a player with extremely fast responses! If you remove jumper J (single-shot mode) and press the fire button, the output from pin 3 goes high and stays there until you release the button.

Construction

This construction approach presumes that you will only use the Atari joysticks on the CoCo. First, construct the printed circuit board (PCB) shown in Fig. 5. When you've finished it, begin installing resistors R1–R10 and C1 as shown in Fig. 6. Finally, install CMOS integrated circuits IC1

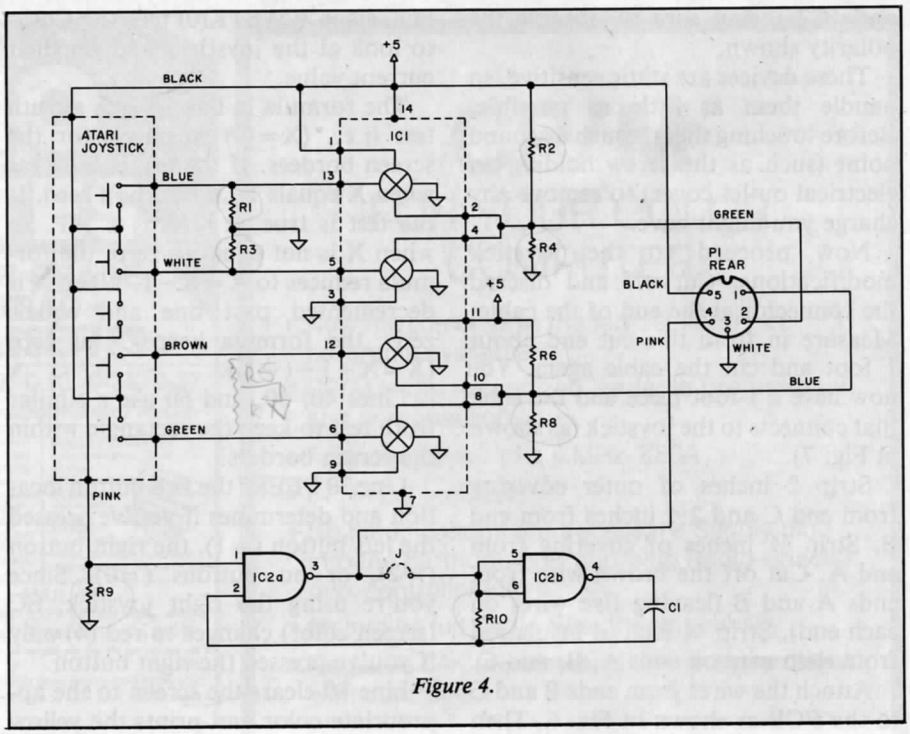

Figure 4.

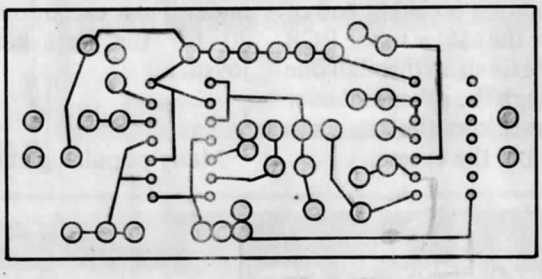

Figure 5.

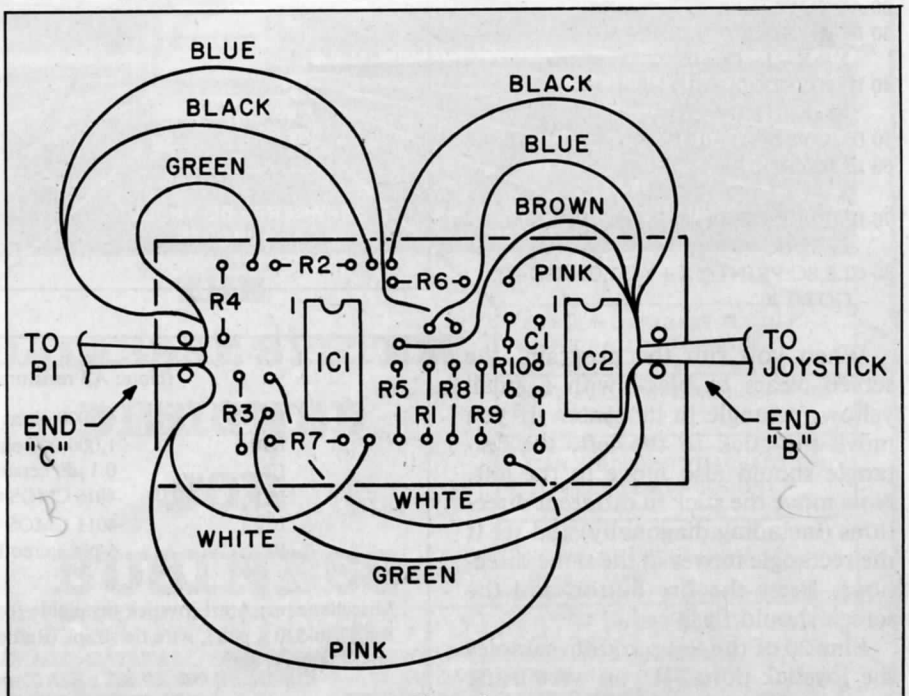

Figure 6.

and IC2, being sure to observe the polarity shown.

These devices are static sensitive, so handle them as little as possible. Before touching them, touch a ground point (such as the screw holding an electrical outlet cover) to remove any charge you might have.

Now proceed to the joystick modifications. Cut off and discard the connector at the end of the cable. Measure in from this cut end about 1 foot and cut the cable again. You now have a 1-foot piece and the cable that connects to the joystick (as shown in Fig. 7).

Strip 2 inches of outer covering from end C and 2½ inches from end B. Strip ¾ inches of covering from end A. Cut off the brown wire from ends A and B (leaving five wires on each end). Strip ¼ inch of insulation from each wire on ends A, B, and C.

Attach the wires from ends B and C to the PCB as shown in Fig. 6. Then attach the wires from end A to the DIN connector. Place ends B and C over the double holes on either end of the PCB. Secure the cables to the PCB by passing a wire tie strap through one hole, back through the other and over the cable. Then close the tie strap tightly and clip off the excess.

Testing

Turn on your CoCo and plug the joystick into the right joystick port. Enter this short test program:

```
10 CLS:X=16:Y=8
20 A=JOYSTK(0)
30 IF JOYSTK(0)=0 THEN
    X=X-1-(X=0)
40 IF JOYSTK(0)=63 THEN
    X=X+1+(X=31)
50 IF JOYSTK(1)=0 THEN Y=Y-1-(Y=0)
60 IF JOYSTK(1)=63 THEN
    Y=Y+1+(Y=14)
70 IF (PEEK(65280) AND 3)=2 THEN SC=4
    ELSE SC=0
80 CLS SC:PRINT@X+Y*32,CHR$(148);:
    GOTO 20
```

When you run the program, the screen clears to black with a small yellow rectangle in its center. If you move the stick to the left, the rectangle should also move to the left. Now move the stick in different directions (including diagonally) and see if the rectangle moves in the same directions. Press the fire button and the screen should flash red.

Line 20 of the test program samples the joystick ports. If you were using the left joysticks (JOYSTK(2) and JOYSTK(3)) you would still need this line, since JOYSTK(0) tells the CoCo to look at the joysticks and get their current value.

The formula in line 30 uses a truth test (i.e., (X=0)) to check for the screen borders. If the test is false (as when X equals 1), it returns a zero. If the test is true, it returns a −1. So when X is not equal to zero, the formula reduces to X=X−1. When X is decremented past one and equals zero, the formula keeps X at zero (X=X−1−(−1)).

Lines 40, 50, and 60 use a similar truth test to keep the rectangle within the screen borders.

Line 70 PEEKs the fire-button location and determines if you've pressed the left button (=1), the right button (=2), or no buttons (=0). Since you're using the right joystick, SC (screen color) changes to red (4) only if you've pressed the right button.

Line 80 clears the screen to the appropriate color and prints the yellow rectangle (CHR$(148)) at the updated screen position that X and Y determine. Then execution returns to line 20 for the next sampling of the joystick.

Use

Many popular games (like Donkey Kong and Pacdroids) can use the joystick without any modification, since these games look to see whether the joystick is at one extreme (0) or the other (63). You can adapt Basic games that rely on a specific number from the joystick but are actually just determining direction of movement by using the logic of the test program.

You should realize that the number produced when the joystick is at rest cannot be exactly 32. From Fig. 2, note that when both S1 and S2 are open (stick at rest), the output is the ratio R1/(R1+R2). If R1 equals R2, then the output ratio is exactly 1:2 which produces the number 32.

Since the resistors' values can vary by as much as 5 percent, the ratio can be somewhat less or greater than 1:2. However, with standard 5-percent resistors, the number generated should not be less than 30 or more than 34. If necessary, you can select resistor values very close to each other so your joystick produces a 32 at rest. In most instances, however, this will not be necessary. ■

Address correspondence to James J. Barbarello, R.D. #1, Box 241H, Tennent Road, Englishtown, NJ 07726.

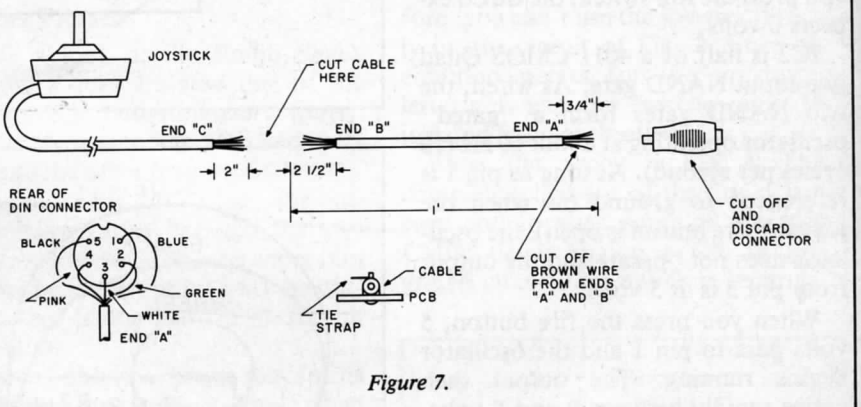

Figure 7.

(Note: All resistors are ¼ watt, 5 percent)

R1–R9	47,000 ohms
R10	1,000,000 ohms
C1	0.1 μF, ceramic disk capacitor (10 volts or greater)
IC1	4016 CMOS Quad Analog switch
IC2	4011 CMOS Quad 2-Input NAND gate
P1	6-pin stereo DIN plug (Philmore EA-9 or equal)

Miscellaneous: Atari joystick (available from Digikey Corp. or other mail-order suppliers for $7 to $10 a pair), wire tie straps (Radio Shack part number 278–1632), printed circuit board, solder, etc.

Table 1. List of Materials