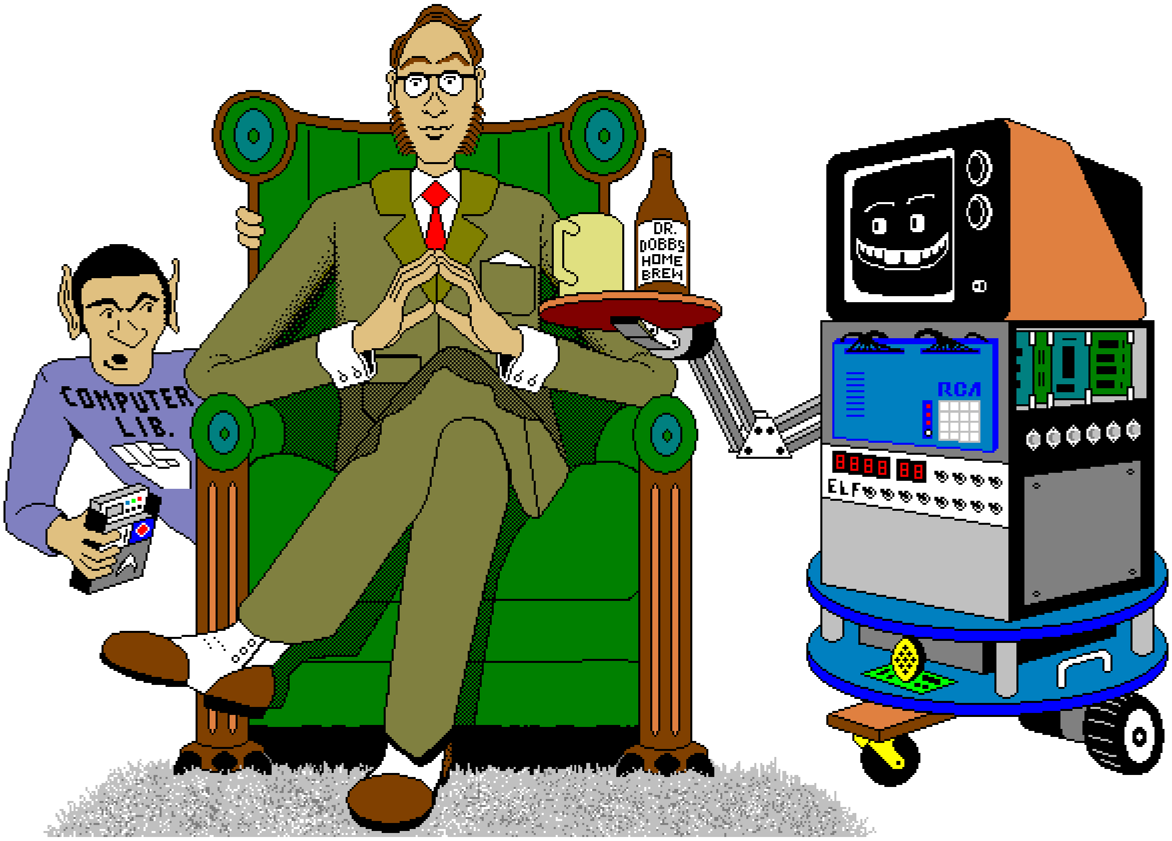
The 1802 Membership Card



For me and you and the 1802

TMSI c/o Lee Hart
814 8th Ave N
Sartell MN 56377 (USA)
leeahart@earthlink.net
http://www.sunrise-ev.com/membershipcard.htm

Rev. H -- last revised: 4/7/2015

What the heck is this? It's an adventure, by cracky! The "Membership Card" is your ticket to the weird and wonderful world of microcomputing. Our guide will be the COSMAC 1802, perhaps the oddest and most entertaining microprocessor yet invented. I hope you'll find this manual to be equally odd and entertaining.

The COSMAC 1802 was created in the 1970's at the dawn of the microcomputer revolution, by Joseph Weisbecker of RCA Corporation. It used their new CMOS fabrication process, which had very low power consumption, high noise immunity, and was very simple to use. It was intended for military and aerospace; applications too tough for other microcomputers to survive.

But Joe was a hacker at heart. He wrote a series of articles starting in the August 1976 issue of Popular Electronics magazine called "Build the COSMAC ELF". It described a simple low-cost computer, using the 1802 microprocessor. At the time, microcomputer systems cost hundreds to thousands of dollars. (Hmm... they still do today!) But Weisbecker's ELF cost about $80! Yet, it was an honest-to-goodness real live computer, able to do anything its much bigger cousins could do -- albeit a bit slower and cruder.

It was the ideal computer trainer. Hobbyists built thousands of ELFs, learning about computer design, construction, and programming in the process. A dozen companies began producing versions of the ELF, also selling for low prices. It was the "Legos" of computers; a simple building-block computer that could be assembled many ways to become almost anything, limited only by your imagination.

I learned about computing on my ELF. It set me on a career in engineering, as it did for thousands of others. 1802's were designed into all sorts of amazing things; video games, music synthesizers, Chrysler engine computers, military weapon systems, and even spacecraft such as NASA's Galileo. Eat stardust, x86 PCs!

So return with me now to those thrilling days of yesteryear, when the heroic pioneers of the microcomputer revolution built their own computers from scratch, and programmed them to do incredible things, all for a tiny amount of money!

What can you do with an 1802?

The Membership Card is much like the Arduino and Parallax BASIC Stamp. You can program it to do simple tasks, and be the "brains" of your projects. It can blink lights, drive relays or small motors, run displays, make annoying sounds with a speaker, read switches or sensors to measure things, perform calculations, make decisions, and more. Unlike the Arduino and BASIC Stamp, the 1802 is far simpler and uses much less power. Its machine language is very simple, and free assemblers are available. There are also open source BASIC interpreters and C compilers for the 1802 to write and run really ambitious programs.

Once a program is loaded, it can be retained without power for hours by the onboard supercapacitor, or indefinitely by maintaining power. Remember, this is a low power computer; it will run for a year on three AA cells! Or, you can put your program in an EPROM so it won't be lost even without power.

The Front Panel can be unplugged, and the Membership Card used by itself. The Front Panel isn't needed until you want to change the program, or debug or observe operation. For stand-alone use, connect power, ground, and your desired inputs and outputs to J1 (the 30-pin header). Jumper J1 pins 13-14 (RUN to VDD) to turn it on. Jumper pins 10-11 (/WE to /MWR) to enable writing to memory.

--

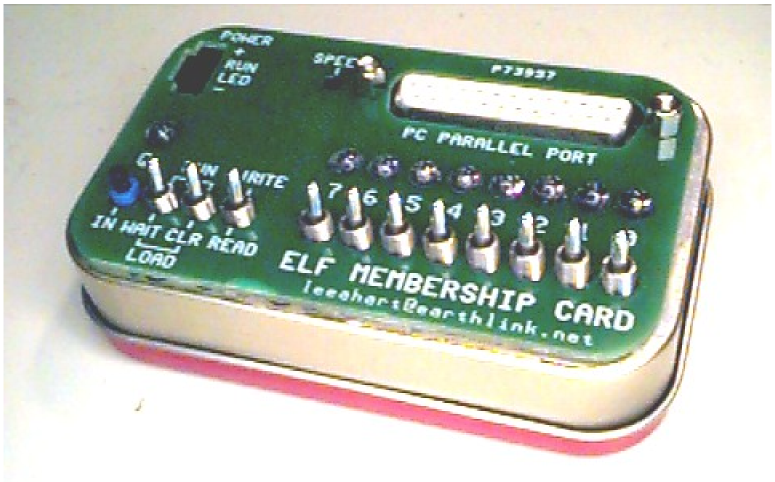
Inspiration for this manual came from Jeff Duntemann's "Captain Cosmo's Whizbang" (c) 1980. The cover was inspired by a cartoon by Chris Cloutier, and the back cover cartoon by one from Bill Higgins.

Specifications: What have we got here?

The Membership Card is a miniature copy of the original Popular Electronics ELF, built using modern parts and repackaged to fit in an Altoids tin. It's got the basics of every computer; a CPU, memory, and I/O.

CPU:	RCA CDP1802ACE microprocessor (the brains of this outfit)
Clock:	1.8 MHz ceramic resonator, plus RC oscillator adjustable from 20-2000 KHz (no, that's not MHz or GHz)
Memory:	2k to 64k bytes, RAM and/or EPROM (that's kilobytes, not megabytes) U2 holds 2k-32k of RAM or EPROM, plus optional U8 for an additional 32k of RAM supercapacitor holds data and programs in RAM without power
I/O:	one 8-bit output port, with LEDs one 8-bit input port, with switches one 1-bit output, with LED four 1-bit flag inputs, one with a pushbutton switch, one with LED one interrupt input
Connectors:	4-pin power connector (+V, RUN, LED, and -V) 25-pin DB25 connector with all the I/O and power on it DB25 can plug into a PC parallel port to operate the Membership Card with the PC DB25 also has RS-232 and TTL serial I/O (using Q and EF3)
Size:	3-1/2" x 2-1/8" x 3/4" (89 x 54 x 19 mm)
Power:	voltage: 3v to 5v DC current: 0.1 to 2ma (depending on clock speed and supply voltage) plus 1-3ma for each LED that is on
Aroma:	a hint of curiously strong peppermint

Unlike modern microcomputers that require an expensive PC and huge software programs to do anything, the Membership Card is totally self-sufficient. No PC is needed, and no special software is required. You can power it from a small solar panel, and program it with nothing but the front panel switches and lights. If you ever get stranded on a desert island and need to compute, this is the computer to have in your pocket.



The complete package: The Membership Card, Front Panel, and Cover Card all fit in an Altoids tin.

Assembly: Getting it all together

This ain't no Heathkit, but I'm working to make it as easy to build as possible. I want it to be something you can give to your kids, and have them experience the thrill of saying, "I did it! It's alive! Bwoo ha ha hah..."

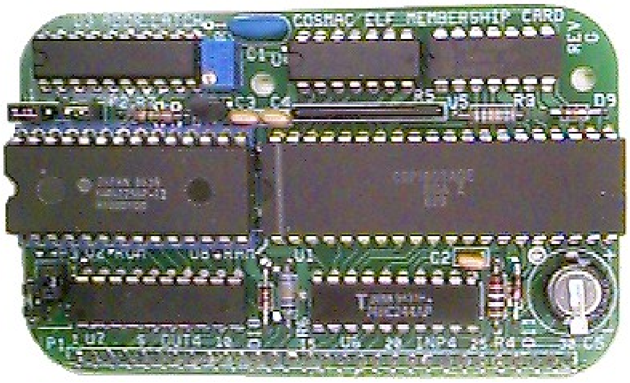
To assemble it, you'll need the following tools:

- A clean, well-lit, place to work. Preferably one without cats or small children. Or if the kids are old enough, let them help. (Cats are never old enough to help.)
- Soldering iron with a small tip. Don't use a soldering gun unless you're desperate. Soldering these tiny parts and pads with a big fat tip will be a real challenge.
- Solder. 63/37 tin/lead is best, but 60/40 is also good. Lead-free electronics solder is also OK, though it doesn't solder as well. It **<u>must</u>** be rosin core electronics solder; NOT acid core plumbing or sheet metal solder!
- Wire cutters. The smaller the better. Nothing is big here.
- Needle-nosed pliers. For bending or straightening lead wires, holding nuts, etc.
- Screwdriver with a 1/8" wide or smaller blade. For tightening mounting screws, and adjusting the clock frequency pot.
- A magnifying glass. My old eyeballs aren't good enough to read the markings on some parts, or see whether a solder joint is done right or is shorted to the pad next to it. Your eyes may not be that good, either.

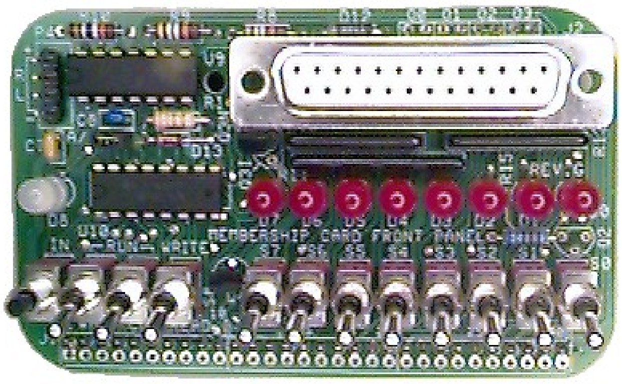
You'll need to know how to solder. This isn't the right kit to learn how to solder. The pads are pretty small, and if you make a mistake, it can be a real fight to get the part off and put back on the right way.

First, make sure you have all the parts. I supplied everything that goes on the printed circuit boards, but you may need to gather a few other things yourself. I'll supply Altoids tins as long as my supply holds out, but you will have to drill or cut the holes in it for the switches, LEDs and connectors.

The Membership Card is your entry into the **COSMAC College of Computer Knowledge**. We'll start with an aptitude test. See if you can find and identify the common electronic components listed on the next page. Place a check mark in the box (X) as you find them. If any are missing, let me know so I can send it out before you get bored and go back to watching TV. Ready? Turn the page to begin an adventure that could last a lifetime...



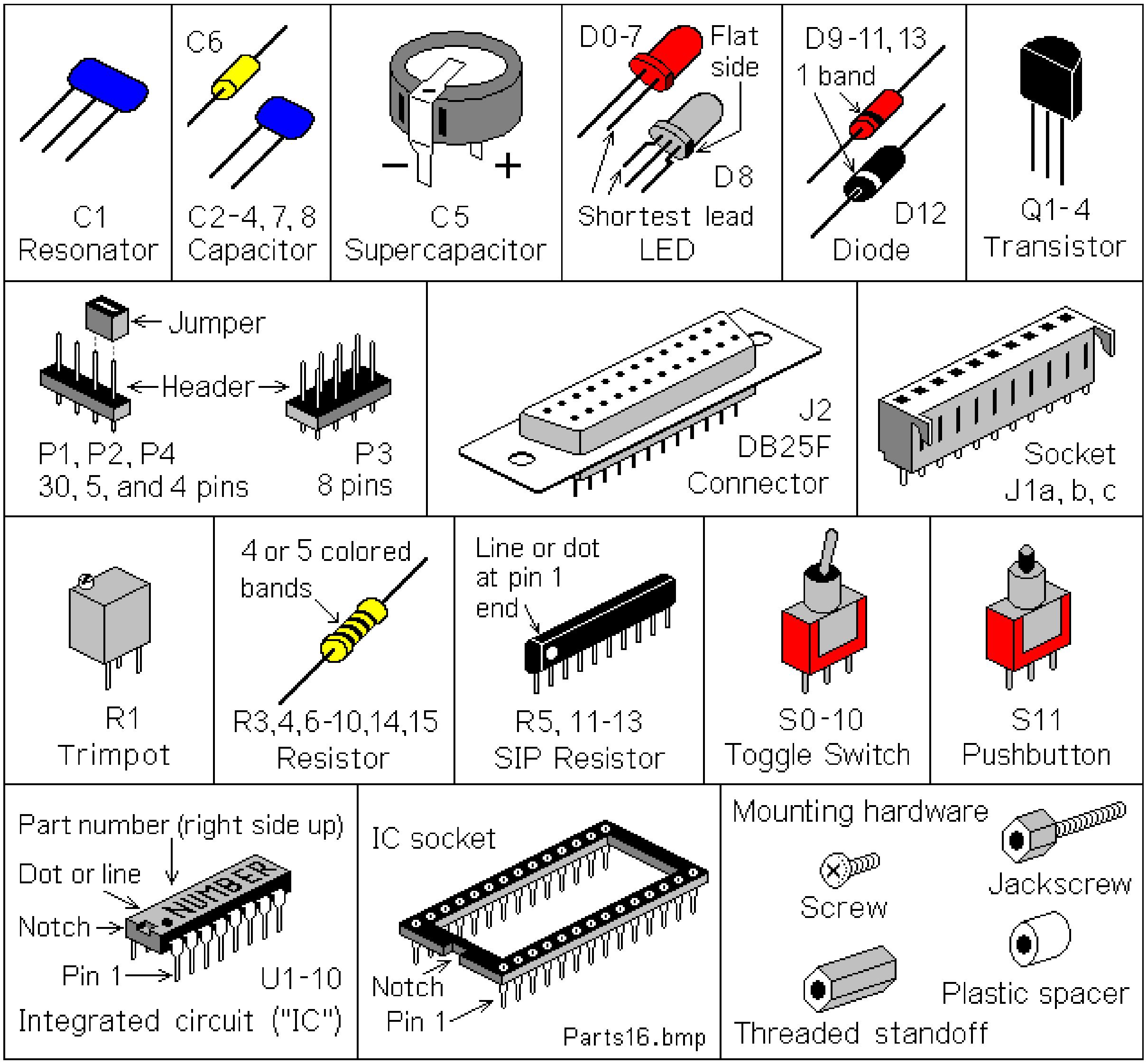
Membership Card



Front Panel Card

Quantity	Identifier	**PARTS LIST**	Source
( ) 1	C1	ceramic resonator, 1.8 MHz (blue, marked 1.80Y)	Mouser 520-ZTT180MG
( ) 4	C2-4,7	capacitor, 0.1uF X7R ceramic (blue, marked 104M)	Jameco 544868
( ) 1	C5	supercapacitor, 0.10F, 5.5vdc (black disk, 0.4" dia.)	Digikey 283-2818-ND
( ) 1	C8	capacitor, 4.7uF ceramic (blue, marked 475)	Mouser 810-FK14X7R1E475K
( ) 8	D0-7	LED, T1-3/4, red	Jameco 253753
( ) 1	D8	LED, T1-3/4, red/green, common cathode (clear with 3 leads)	Digikey 754-1232-ND
( ) 4	D9,10,13,14	diode, 1N4148 (clear red case)	Jameco 36038
( ) 1	D11	diode, 1N5231B 5.1v zener (clear red case)	Jameco 179047
( ) 1	D12	diode, Schottky 1N5818 (black case)	Jameco 177957
( ) 3	J1a,b,c	socket, 10-pin top entry, Molex 22-18-2101	Digikey WM3241-ND
( ) 1	J2	connector, 25-pin, DB25 female, vertical PC mount	Jameco 15165
( ) 1	P1	header, 30-pin, with 0.025" sq.pins on 0.1" centers	Jameco 103342
( ) 1	P2	header, 5-pin, with 0.025" sq.pins on 0.1" centers	Jameco 2076789
( ) 1	P3	header, 4x2 pin, with 0.025" sq.pins on 0.1" centers	Jameco 109517
( ) 1	P4	header, 4-pin, with 0.025" sq.pins on 0.1" centers	Jameco 117560
( ) 1	Q1	transistor, NPN w.base resistors, FJN3301 (3 leads, black)	Digikey FJN3301RTACT-ND
( ) 1	Q2	transistor, NPN w.base resistors, FJN3307 (3 leads, black)	Digikey FJN3303RTACT-ND
( ) 1	Q3	transistor, PNP w.base resistors, FJN4303 (3 leads, black)	Digikey FJN4305RTACT-ND
( ) 1	Q4	transistor, NPN, 2N3904 (3 leads, black)	Jameco 38359
( ) 1	R1	trimpot, 1meg, BI Tech 64WR1MEG (3 leads, gray cube)	Jameco 2194902
( ) 2	R3,6	resistor, 100k 1/4w (brown-black-black-orange-violet)	Jameco 691340
( ) 4	R4,8-10	resistor, 499k 1/4w (yellow-white-white-orange-brown)	Jameco 691500 (470k)
( ) 1	R5	SIP resistor, 9-pin, 8 x 10k (black, marked L91S103)	Mouser 652-4609X-1LF-10K
( ) 1	R7	resistor, 1meg 1/4w (brown-black-green-gold)	Jameco 691585
( ) 1	R11	SIP resistor, 10-pin, 9 x 1k (black, marked L101S102)	Jameco 97877
( ) 2	R12,13	SIP resistor, 8-pin, 7 x 100k (black, marked L81S104)	Jameco 276358
( ) 1	R14	resistor, 3.3k 1/4w (orange-orange-red-gold)	Jameco 690988
( ) 1	R15	resistor, 6.8k 1/4w (blue-gray-red-gold)	Jameco 691067
( ) 11	S0-10	toggle switch, subminiature SPDT (plus 3 nuts)	Digikey CKN1091-ND
( ) 1	S11	pushbutton switch, subminiature SPDT	Digikey CKN1740-ND
( ) 1	U1	1802 microprocessor	eBay, or TMSI (that's me :-)
( ) 1	U1 socket	40-pin, ultra-low-height, Mill-Max 115-43-640-41-003000	Digikey ED90220-ND
( ) 1	U2	32k RAM supplied. Use 2k-32k RAM (6116, 6264, 62256 etc)	Jameco 42850 (32k RAM)
		or EPROM (2716-27256 or 26C16-27C256 etc.)	Jameco 39731 (32k EPROM)
( ) 1	U2 socket	28-pin, very-low-height, Mill-Max 115-43-628-41-001000	Digikey ED90205-ND
( ) 1	U3	74HC373 or 74HCT373 octal latch	Jameco 45831
( ) 1	U4	4093 quad 2-input NAND gate, Schmitt trigger	Jameco 13400
( ) 1	U5	4013 dual D flip-flop	Jameco 893443
( ) 1	U6	74HC244 or 74HCT244 octal buffer	Jameco 45655
( ) 1	U7	74HC374 or 74HCT374 octal D flip-flop	Jameco 45858
( ) 1	U9	4071 quad 2-input OR	Jameco 13274
( ) 1	U10	74HC157 or 74HCT157 quad data selector	Jameco 272170
( ) 1	PCB	Membership Card (rev.H) and Front Panel Card (rev.G)	TMSI (me again)
( ) 2	screw	#4-40 x 3/16" round head machine screw	Fastenal 1128626
( ) 2	spacer	#4 x 3/16" dia, 1/8" long plastic spacer	Mouser 749-9908-125
( ) 2	standoff	#4-40 x 3/16" dia, 5/16" long, hex female threaded standoff	Mouser 728-FC2054-440-A
( ) 2	jackscrew	#4-40 x 3/16" dia, .187" long female, 0.515" long male	Jameco 108987
( ) 4	jumper	jumpers for P2 and P3 headers	Mouser 737-MSB-G
( ) 1	case	"Altoids" (or equivalent) tin candy box	grocery or candy store

Did you find them all? Here are some hints:



Comments on components

The resistors use colored rings to identify their resistance in Ohms. The other parts have numbers, but you may need a magnifying glass to read them! Capacitors have their value in Farads (usually picoFarads). For example, "104" means 10 with four zeroes after it; that's 100,000pF (picoFarad), which is also 0.1uF (microFarad).

ICs have room for a part number, but it's hidden between extra letters (i.e. the 1802 is actually marked CDP1802ACE). Leave the ICs in their packaging until you're ready to install them. They are easily damaged by static electricity. You know that tiny little spark you get if you touch something metal after petting the cat or walking across a carpet? That's static electricity. In the microscopic world inside an IC, it hits like a lightning strike. **KABOOM**! Your IC is dead.

Note: U8 and C6 are <u>optional</u> parts to expand memory to 64k. They are <u>not</u> supplied with the standard kit.

Assembly

Assembly is a work in process. I'll just describe how I built mine. Please let me know if you find better ways to do it! Check off the steps as you go (X), in case you get interrupted and have to come back to it later.

A couple things to keep in mind: To fit both boards in an Altoids tin, they must be built with the parts as low as possible. Standard IC sockets are too high, so I've supplied special low-profile sockets. Likewise, the plastic bodies of the square pin headers (P1 to P3) make them too tall, so the plastic bodies have to be removed after soldering. You'll also need to trim the leads on the back of the boards very short as you go.

Ready? Let's get started!

Membership Card assembly

All parts go on the side of the board with the white silkscreened lettering (the top or "component side"). All soldering is done on the other side (the back or "solder" side). The only exceptions are for connectors (and I'll remind you when we get to them).

First, we'll install the lowest parts (the resistors):

(   ) R3: 100k ohm resistor (gray body, with brown-black-black-orange-violet color bands). Bend the wire leads, and place it on the board at the location marked "R3". Bend the leads outward slightly to hold it in place. Turn the board over, solder each lead, and cut off the excess as short as possible.

Install the rest of the parts the same way.

(   ) R4: 499k (brown body, with yellow-white-white-orange-brown bands).
(   ) R6: 100k (gray body, with brown-black-black-orange-violet bands).
(   ) R7: 1 meg (tan body, with brown-black-green-gold bands).
(   ) R2 is easy to install -- there isn't one!

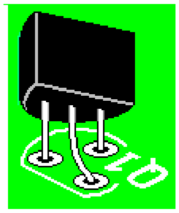
Capacitors have two leads, but are installed pretty much the same. If the leads are bent, straighten them with your needle-nosed pliers to fit the board. Then repeat the above steps to install each one.

(   ) C2: 0.1uF (blue, marked "104M").
(   ) C3: 0.1uF (blue, marked "104M").
(   ) C4: 0.1uF (blue, marked "104M").
(   ) C5: 0.10F (a black disk, marked "0.10F"). It has two flat leads, one with a "-" sign next to it. This lead goes in the hole near the "-" and "C5" markings, closest to the corner of the board.

These next parts are a little bigger, with more leads:

(   ) R5: 9-pin SIP (Single Inline Package) with eight 10k resistors in it (black, marked "L91S103"). Put it at "R5" with pin 1 (the end with the white line) next to C4.
(   ) C1: 1.8 MHz ceramic resonator (blue body with 3 pins, marked "1.80Y").

( ) Q1: NPN transistor FJN3301 (black with 3 leads, marked "R3301"). Install it as shown at right, (not as shown by the silkscreen on the board). Position the board with the "Q1" lettering upside down. Hold the transistor with its flat side toward you, and the leads pointing down. Put the left lead in the left hole. Put the center lead in the LOWER RIGHT hole. Put the right lead in the UPPER CENTER hole. Push it down close to the board. Solder each lead, and cut off the excess.



Q1revH.bmp

( ) R1: 1 megohm trimpot (a gray cube with 3 leads, marked "WR1MEG").

Diodes are polarity sensitive, and have a colored a band around one end. Be **sure** the banded end matches the band shown on the board.

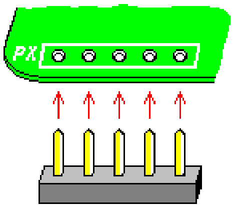
( ) D9: 1N4148 (a reddish glass tube, with a wire at each end, and "1N4148" printed on it in tiny letters). Bend the leads, and place it on the board at "D9" with the banded end as shown on the board. Solder each lead, and cut off the excess.
( ) D10: 1N4148.
( ) D14: 1N4148.
( ) D11: 1N5231B (reddish glass, marked "1N5231B"). It has tape on its leads, marked "D11".

Now for the pin headers: They get soldered on top, which is easier without the ICs in the way. Hint: Cover the nearby IC holes with a piece of masking tape so you don't accidentally plug them with solder.

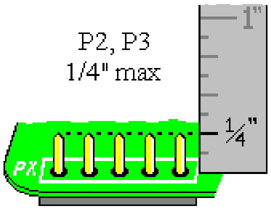
( ) P1: 30-pin header. Insert the pins so the plastic body is against the **bottom** of the board (see drawing below). Solder the pins on the **top** side. Don't use too much solder. Pry off the plastic body, and cut the pins on the bottom as short as possible. Hint: To remove excess solder from the top, temporarily push the plastic onto the top of the pins to hold them in place, and re-solder the pins from the bottom.

( ) P2: 5-pin header. Install P2 the same as P1. Measure the height of the pins on top. If more than 1/4" (6.3mm) high, tap them down into the plastic body. They **must not** be higher, or they will short to the Front Panel. Solder on **top**, and cut the plastic body and pins on the **bottom** as short as possible.

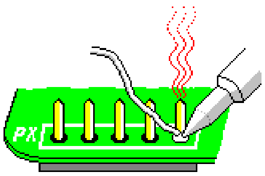
( ) P3: 6-pin header. Install it the same as P1 and P2. I supplied an 8-pin part. Remove 2 pins so it fits on the board. Be sure the pins are no more than 1/4" (6.3mm) above the board.



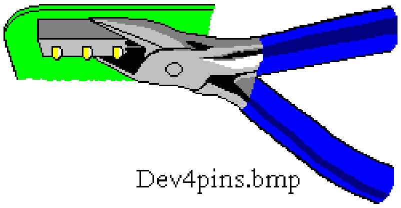
Insert from bottom



check height



solder from top



cut off pin and body under board

ICs and static electricity: (Old pros and young fools can skip this paragraph.) ICs are easy to damage with static electricity! Keep them in their protective packaging until needed. When you remove an IC, keep it in your hand until it is on the board. Pick up the board or tool with your other hand. Do not have the IC be the first thing to touch the tool or board. That way, any static electricity discharges into you, and not the IC.

ICs are polarity sensitive; they **must** be installed with the pin 1 end matching the marking on the board. The pin 1 end may be marked in a number of ways; with a dot, notch, or line, etc. When the printing on the IC is right side up and facing you, pin 1 is in the lower left corner. See the illustration on page 6 to find pin 1.

Lay each IC on its side on the table. Bend the pins slightly inward so they are straight and parallel to each other. If it still does not fit into the holes on the board, use your needle-nosed pliers to straighten the leads.

IC sockets are only supplied for U1 and U2 (the 1802 and memory chip). Sockets add cost, and make the board taller so it won't fit in the Altoids tin. Sockets are also the least reliable part of the whole computer, especially if you use el-cheapo sockets! But they make troubleshooting and chip replacement easier, so you can add them if desired.

( ) U1a: 40-pin IC socket. Install it on the board at location "U1". The notched end must match the notch shown on the board. Solder each pin.

( ) U1: 1802 (marked CDP1802ACE). Plug it into the socket, so the notched end matches the board.

( ) U2a: 28-pin IC socket. Install it as described for U1a.

New! Improved! (yada yada...) Optional Memory Expansion RAM

The rev.G Membership Card has room for a second memory chip, U8. This is an **option**, not included with the standard kit. It adds 32k bytes of RAM, addressed from 32k to 64k. To add it, you need two extra parts:

1. U8: 32k RAM in 0.3" wide DIP package (Cypress CY7C199; Jameco 242376).
2. C6: 0.1uF X7R axial lead ceramic capacitor (Mallory P20R104K5; Jameco 536542).

If you are **not** installing U8, skip the steps in the following box. To **install** U8, do steps A, B, and C:

```
 ------------------------------------------------------------------------------------------------------
| A. ( ) U2: If U2 is installed, unplug it and set it aside. On the socket for U2, remove the 3 plastic bars |
|          that connect its two rows of pins. The easiest way is to melt them in two with your soldering     |
|          iron, then cut off the excess with your diagonal cutters. This makes room for U8 to fit under U2.|
|                                                                                                            |
| B. ( ) U8: 32k 0.3" wide 28-pin RAM (marked CY7C199 or equivalent). Install it at location U8 on           |
|          the board. Be sure the pin 1 end matches the board. Solder U8 directly to the board, without a    |
|          socket; a socket makes it too tall to fit under U2.                                               |
|                                                                                                            |
| C. ( ) C6: 0.1uF (yellow, marked "104"). Install it like you did the resistors. Yes, it is a tight fit!    |
|          If necessary, carefully remove a little plastic from socket U2a to make room for capacitor C6.   |
|          The easiest way to do this is to melt it away with your soldering iron.                           |
 ------------------------------------------------------------------------------------------------------
```

( ) U2: 32k 0.6" wide 28-pin memory IC (marked "CXK58256P-10L" or equivalent). Plug it into the socket so the notched end matches the board.

Now we'll do the rest of the ICs. Install each one so the notched end matches the board. Bend a couple pins if necessary to hold it in place. Then solder all the pins, and cut off the excess. (Note: These ICs don't have sockets. If you add sockets, remember they must be very low profile to fit in the Altoids case.)

( ) U3: 74HC373 or 74HCT373 20-pin IC.
( ) U4: 4093 14-pin IC.
( ) U5: 4013 14-pin IC.
( ) U6: 74HC244 or 74HCT244 20-pin IC.
( ) U7: 74HC374 or 74HCT374 20-pin IC.

Front Panel assembly

Install these parts on the Front Panel PC board the same way you did on the Membership Card:

( ) C7: 0.1uF (blue, marked "104M").
( ) D12: 1N5818 (black body). Be sure the end with the white band matches the board.
( ) D13: 1N4148 (clear red body). Be sure the band matches the board.
( ) R8: 499k (brown body, with yellow-white-white-orange-brown bands).
( ) R9: 499k (yellow-white-white-orange-brown).
( ) R10: 499k (yellow-white-white-orange-brown.
( ) R14: 3.3k (tan body, with orange-orange-red-gold bands).
( ) Q4: 2N3904 NPN transistor (black with 3 leads, marked 2N3904). Position it as shown on the board.
( ) P4: 4-pin header. This time, you can keep the black plastic body on it. Or if you prefer, leave P4 off, and solder 4 small wires to the board instead, with any connector you like on the other end.

Next, install the SIP resistor networks and ICs. The end with the dot or line is "pin 1". It goes on the top in the illustration below, to match the silkscreening on the board. Solder each lead, and cut off the excess.

( ) R11: 10-pin SIP resistor network with nine 1K resistors (black, marked "L101S102").
( ) R12: 8-pin SIP with seven 100K resistors (black, marked "L81S104").
( ) R13: 8-pin SIP with seven 100K resistors (black, marked "L81S104").
( ) U9: 4071 14-pin IC. Be sure pin 1 is at the correct end!
( ) U10: 74HC157 16-pin IC. (That goes for U10 as well.)
( ) J1a, J1b, J1c: Three 10-pin parts are supplied. Mount them on the BOTTOM of the board. The pins go in the row of holes toward the CENTER of the board, with the connector's "cat ears" also toward the center. Do not solder the empty holes close to the edge. Hint: Put a piece of masking tape over these holes so you don't solder them closed. Solder the connector's pins, and cut off the "ears".

New! Serial Input and Output Options

Serial I/O uses the 1802's Q output and /EF3 input pins (note that /EF3 is inverted; Q is not). A program is needed to "bit bang" the data in and out. For example, RCA's UT4 monitor, TMSI's IDIOT monitor or Tiny BASIC, or Spare Time Gizmo's Elf2K EPROM. You can toggle the program in with the Front Panel switches (tedious). Or download it with the parallel port (requires a PC with a parallel port). Or, install an EPROM at U2 with the program in it (you'll need the EPROM, and the optional RAM at U8 -- see page 9).

Hardware-wise, serial I/O is on J2, the 25-pin D-connector on the Front Panel. Output TXD is on pin 15, and input RXD is on pin 20. You can assemble the Membership Card for TTL (0-5v) or RS-232 (+/-12v) levels, and either normal or inverted data. But you'll have to look up, or experiment to figure out what method the thing you want to connect is using. See http://www.retrotechnology.com/memship/mem_rom_serial.html.

Do **ONE** of the following sets of steps (A or B). If you're not going to use serial I/O, or don't know what serial format you'll need, build it for NORMAL data (A), with TTL output levels (i.e. C8 shorted).

( ) A. For NORMAL non-inverted data (used for RS-232, where the idle state is a LOW voltage):

( ) R15: 6.8k resistor (tan body, with blue-gray-red-gold bands).
( ) Q2: Do not install Q2.
( ) Q3: Install PNP transistor FJN4303 (marked "R4303") in the Q3N position.

( ) B. For INVERTED data (used by TTL serial-to-USB adapters, where idle is a HIGH voltage):

( ) Q2: NPN transistor FJN3307 (black with 3 leads, marked "R3307").
( ) R15: Do not install R15.
( ) Q3: Install PNP transistor FJN4303 (marked "R4303") in the Q3I position.

( ) C8: 4.7uF (blue, marked "475"). Yes it really is 47 times more capacitance than C7! Amazing, isn't it?

( ) For RS-232 serial output levels (+v/-v), install C8.
( ) For TTL serial output levels (0v-5v), install a piece of wire in place of C8.

D8 is a 2-color LED. It is red when the 1802's Q output is high, and green when the 1802's EF3 input pin is high. Since Q and EF3 are used for serial I/O, this LED is off when idle, and blinks red or green when data is sent or received. If nothing is connected to the serial input, the LED is green if configured for NORMAL serial input (think of it as a "power on" indicator). If configured for INVERTED serial input, D8 will be off.

( ) A. If you don't have a **Cover Card**, use the bare Front Panel board as a template to drill the Altoids tin (or your own plastic or metal panel). I use the **bottom** of the Altoids tin as my "front panel". Drill small 1/32" holes for the switches, LEDs, D-connector J2, and power connector P4. Then remove the PC board, and enlarge the holes to the final size. Use a "step" drill bit, which won't tear the thin metal. This is a long drill bit with a single cutting edge and steps in it to drill a dozen or so sizes from 1/8" to 1/2". Cut the D-connector opening with a nibbling tool or a lot of hand filing.

B. The optional **Cover Card** is available at http://www.sunrise-ev.com/membershipcard.htm. It has all the labels and holes made for you (see photo on page 3). Just cut a big hole in the Altoids tin, and use the card to cover the hole. Solder, screw, or glue it in place when done assembling the boards.

( ) S11: Pushbutton switch. Put it on the board with the printed side up in this view. Bend one of the pins slightly; just enough so it won't fall off the board.

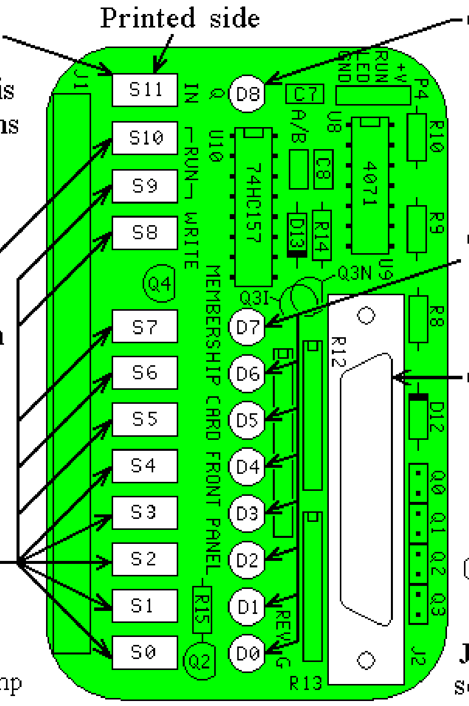
Do not solder.

( ) S10: Toggle switch. Put it on the same as S11, with the "120v" side up. Bend a pin so it won't fall off.

Do not solder.

( ) S9-S0: Place them on the board the same as S10.

Do not solder.



Dev4FP-A.bmp

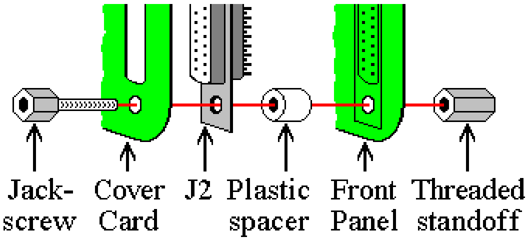
( ) D8: White 3-lead LED. Put the shortest lead in the left hole, and the flat side of the case on the right. Bend a lead so it won't fall off.

Do not solder.

( ) D0-7: Red LEDs. Short wire in right hole. Ignore the flat side.

Do not solder.

( ) J2: D-connector. Loosely mount J2 to the Front Panel Card and Cover Card with #4 hardware as shown.



( ) Wiggle the parts and boards around so all the LEDs and switches fit neatly in the holes in your Cover Card or front panel. Work the board as close as you can. Be careful! The switches can break if the holes in your front panel don't match well enough. Hint: It looks nicer if the LEDs stick out the front a little bit. **Now** solder all the parts to the board, with the Cover Card holding everything in position.

( ) Remove the mounting hardware from D-connector J2. Make sure you can easily remove, and re-install the Cover Card. You may have to re-heat or reposition some parts, or enlarge some holes in your front panel to make it fit easily. This may seem tedious, but it makes **sure** that all the parts are soldered in the right places so they won't get forced or broken! :-)

Final Assembly -- Jumpers and Options

Next, install jumpers on the Membership Card to configure it for the type of memory chips at U2 and U8.

( ) P2 and P3: Look at the type of memory chip installed at U2. Ignore the letters at the beginning and end of the part number, and focus on the 3-digit number in the center. For example, if it is marked "CXK58256P-10L" -- then it's a generic 256k bit (32k byte) RAM -- the standard chip I supply.

( ) Find **TABLE 1** in the bottom left corner of the schematic on page 20. Look up your chip number to see what jumpers to install. For a 32k byte RAM (62256 etc.), install jumpers at P2 between pins 1-2 and 4-5, and at P3 between pins 1-3 and 2-4.

( ) If U8 is installed, look on the bottom of the board, under R7. There are 4 pads, marked U2, U8, HI, and LO. Jumpers at these pads control the addressing of U2 and U8. There are two options:

1. U2 to LO (0-32k) U8 to HI (32-64k) Standard; jumpers are etched onto the board for this.

   a. RAM at U2 (the classic all-RAM "ELF" configuration): Load programs with the Front Panel.
   b. Or, install a 2k to 32k EPROM at U2, and RAM at U8: Automatically runs programs in the EPROM. For stand-alone controllers, since no Front Panel or manual program loading is needed.
   c. Or, install RAMs at both U2 and U8. Now you have a 64k all-RAM system.

2. U2 to HI (32-64k) U8 to LO (0-32k) Optional; cut the etched jumpers and add your own.

   a. Install RAM at U8. Load programs with the Front Panel. U2 socket can be empty.
   b. Or, put a 2k to 32k EPROM at U2, with programs that run in high memory (i.e. the Elf2K ROM). Use the Front Panel to load a LBR (address), then run it to jump to code in the EPROM.

( ) Jumper options A/B, Q0, Q1, Q2, Q3 are on the Front Panel card:

1. If you want to plug J2 into a PC parallel port, short A/B and leave Q0, Q1, Q2, and Q3 open. This multiplexes the eight OUT4 bits into 4 bits, so a PC parallel port can control the Membership Card.

2. If you want to use J2 as a general-purpose I/O port with all eight OUT4 bits on pins of J2, leave A/B open, and short Q0, Q1, Q2, and Q3.

( ) Re-install the Cover Card with #4 mounting hardware as shown on page 11. Plug the Membership Card and Front Panel cards together. Look in between, and make sure that you can see "daylight" between them. **NOTHING** should touch between the two boards except the 30-pin connector P1-J1, the top of trimpot R1, and the two hex standoffs. If anything else touches, rework your solder joints or trim the leads on the back of the Front Panel so there are no shorts.

( ) Put nuts on a few of the switches (like the ones at each end). There isn't room for nuts on all of them. They aren't really needed for such a small board. The hardware for J2 usually does the job.

( ) Finally, screw the Membership Card to the threaded standoffs in J2 with two #4 screws.

( ) The last page of this manual is a "cheat sheet" summary of operation. Cut out one vertical column, and fold it to fit inside the Altoids box. It's a handy reminder, and also keeps the pins on the back of the board from shorting to the metal case!

Power On!

Now it's time for the smoke test! We'll connect power, and try not to let any of the magic smoke out. (Old timers will tell you that electronic devices don't work if the smoke gets out.)

The Membership Card takes very little power; 3v to 5v DC at 1ma, plus a few ma for each LED that is lit. You can use three 1.5v batteries, an old cellphone charger that outputs 5vdc, or even the solar panel from a scrapped calculator for power. For example, a battery holder with three AA cells in a second Altoids tin works nicely.

R1 sets the clock frequency (12 turns end-to-end). Fully clockwise is very slow, but very low power. Fully counter-clockwise is maximum speed, which is OK at 5v but too fast with a 3v supply. C1 is a ceramic resonator; at 5v it will "lock in" to 1.8 MHz with R1 adjusted fully counter-clockwise, then clockwise 3/4 to 1 turn.

Zener diode D11 is an "idiot" diode. If power is connected backwards, or is over 5.1 volts, or is AC instead of DC, then D11 shorts it out to protect the rest of the board. At worst, D11 will get hot and fail shorted; but it's cheap and easy to replace.

P4 is the power connector. It is a common male header with four 0.025" square pins on 0.100" centers, located in the top left corner of the Front Panel board. There are many mating connectors. For example, the 4-pin cable used in PCs between the CDROM drive and sound card can be used. Other choices include:

-- Molex KK series: 22-01-3047 housing, and 08-55-0102 terminals (www.jameco.com 234819 and 234931). Common and easy to use, but not latched or keyed.
-- AMP latch series: 25403S-04 header, 25403H-04-R housing, 25403T terminals (www.jameco.com 152734, 152741, 181673). High quality, latched, keyed (if you use its header); hard to install terminals.

P4 pin	Name	Function
1	+	VDD (or VCC), the power supply positive; +3 to +5 volts DC.
2	RUN	Controls the 1802's clock oscillator. Connect to pin 1 to Run; leave pin 2 open to Stop.
3	LED	LED negative. Connect to pin 4 to enable the LEDs; leave pin 3 open to disable the LEDs.
4	-	VSS (or ground, common), the power supply negative.

Wire your power connector with positive (+) on P4 pins 1-2, and negative (-) on pins 3-4. To turn the Membership Card "on", plug the power connector onto the board. This makes RUN high (clock runs), and LED low (LEDs enabled). To turn it "off", unplug the connector; RUN goes low so the clock stops, and the LEDs are disabled to minimize power.

On Rev.D and later, D-connector J2 pin 18 can be used for the + power input. It has a diode in series (D12), so applying AC or reversing the power won't hurt anything. If you power your Membership Card it this way, use a jumper plug at P4 to short pins 1-2 and 3-4. Unplugging or plugging it in becomes your on/off switch.

Supercapacitor C4 will hold programs and data for many hours. To hold them longer, leave power connected to P4 pins 1 and 4. Use a DPST (double pole, single throw) switch to break the connections between pins 1-2 and 3-4. This switch provides "on" and "sleep" modes. "On" enables the clock and LEDs to work normally. "Sleep" holds memory and turns the LEDs off so current very low; a set of AA cells will maintain memory for years. Note: If the Front Panel is connected, it over-rides "Sleep" mode unless you also set the CLEAR switch and all data switches S0-S7 low.

Operation

Ok; so you connected power (and nothing smoked). Let's see if it works! The Front Panel selects the 1802's operating mode and shows its status. LED D8 shows the state of the 1802 "Q" register; it will be off or green at power-up. Data LEDs D0-D7 show the last data loaded or output by the 1802 program; they will initially be some random value. Data switches S0-S7 set the input data. S11 is the IN button; press it to load data.

S8 is the READ/WRITE switch; WRITE lets you read or write to memory. READ makes memory read-only. S9 and S10 are the mode switches, CLEAR and LOAD. They select the four operating modes as follows:

	S9	**S10**	
Mode	**CLEAR**	**LOAD**	**1802 Operation**
LOAD	down	down	Waits for the next memory read/write (next press of the IN button).
CLEAR	down	up	Reset the 1802 (sets registers Q, X, P, and R0 all to 0).
WAIT	up	down	Stops the program running in memory, and waits right where it is.
RUN	up	up	Runs (or resumes running) the program in memory.

Here's an example of how these switches and lights work.

1. Set switches S11-S0 as shown. "1" means the switch is up. "0" means the switch is down. "X" means push the IN button. "." means the switch position doesn't matter. I'll show the switch positions like this:

```
   Switch numbers and positions on the board
     S11  S10 S9 S8     S7 S6 S5 S4 S3 S2 S1 S0   Mode    Description
       .   1  0  .       .  .  .  .  .  .  .  .   CLEAR   Resets the 1802.
```

PROGRAM 1 -- BLINK Q FAST Let's load a simple program to blink the "Q" LED:

```
     S11  S10 S9 S8     S7 S6 S5 S4 S3 S2 S1 S0   Mode    Description
2.     .   1  0  .       .  .  .  .  .  .  .  .   CLEAR   Reset the 1802 (sets R0 to address 0000).
3.     .   1  0  1       .  .  .  .  .  .  .  .   WRITE   Set S8 up, so we can write to memory.
4.     .   0  0  1       .  .  .  .  .  .  .  .   LOAD    Set both S9 and S10 down.

       X   0  0  1       0  1  1  1  1  0  1  1        a. Set S7-S0 to "0111 1011", then press the IN button.
                         \___7___/  \___B___/             This loads hex 7B (the SEQ or "Set Q" instruction) into
                                                          memory address 0000, displays it in the LEDs, then
                                                          advances R0 from address 0000 to 0001.
```

	X 0 0 1	0 1 1 1 1 0 1 0 __7__/ __A__/	b.	Set S7-S0 to "0111 1010", then press IN. This loads hex 7A (the REQ or "Reset Q" instruction) into 0001, displays it, then advances R0 to 0002.
	X 0 0 1	0 0 1 1 0 0 0 0 __3__/ __0__/	c.	Set S7-S0 to "0011 0000", then press IN. This loads hex 30 (BR or "Branch Unconditionally") into 0002, displays it, then advances R0 to 0003.
	X 0 0 1	0 0 0 0 0 0 0 0 __0__/ __0__/	d.	Set S7-S0 to "0000 0000", then press IN. Loads hex 00 (tells "Branch Unconditionally" where to jump; in this case, back to 0), displays it, then advances R0 to 0004.

Our program is loaded. Let's read it back to see if it is correct.

	S11 S10 S9 S8	S7 S6 S5 S4 S3 S2 S1 S0	Mode	Description
5.	. 1 0 .	. . . .	CLEAR	Reset the 1802 (sets R0 to address 0000).
6.	. 1 0 0	. . . .	READ	Set S8 down, so we can read memory.
7.	. 0 0 0	. . . .	LOAD	Set both S9 and S10 down.
	X 0 0 0	0 1 1 1 1 0 1 1 __7__/ __B__/	a.	Press the IN button. This reads memory address 0000, displays its contents in the LEDs ("0111 1011" which is hex 7B), then advances R0 from address 0000 to 0001.
	X 0 0 0	0 1 1 1 1 0 1 0 __7__/ __A__/	b.	Press IN again. Displays "0111 1010" = hex 7A from address 0001, then advances R0 from 0001 to 0002.
	X 0 0 0	0 0 1 1 0 0 0 0 __3__/ __0__/	c.	Press IN again. Displays "0011 0000" = hex 30 from address 0002, then advances R0 to 0003.
	X 0 0 0	0 0 0 0 0 0 0 0 __0__/ __0__/	d.	Press IN again. Displays "0000 0000" = hex 00 from address 0003, then advances to 0004.

If our program is correct, now we can run it!

8.	. 1 0 .	. . . .	CLEAR	Reset the 1802 (set R0 back to address 0000).
9.	. 1 1 .	. . . .	RUN	Both S9 and S10 up. The 1802 begins running the program starting at address 0000.

This program is very simple; it tells the 1802 to turn the Q LED on (red), then off (green), and repeat forever. But it's doing it so fast that the LED looks orange. To prove that it's really going on and off, connect a pair of headphones or a small speaker between J2 pin 15 (Q) and pin 20 (ground); you'll hear an audio tone! Adjusting R1 (the SPEED trimpot) changes the pitch.

Or, we can use the WAIT mode to temporarily stop the program.

10.	. 0 1 .	. . . .	WAIT	Stop right where you are! The 1802 "freezes" where it is in the program. The Q bit may be caught set, or reset.

Flipping S10 up and down will RUN and WAIT the program, sometimes catching Q set (red), sometimes reset (green). WAIT is handy for debugging; you can stop at any time, and check any point in the circuit with a meter or logic probe to see what is going on, then continue execution.

Here is a bit longer program. It does the same thing (blinks Q), but much s-l-o-w-e-r. Use the same sequence of switch flipping as above. Let's simplify the description so it's not so wordy. See if you can figure out how to enter it. (Hint: There's a "cheat sheet" at the end of the manual).

PROGRAM 2 -- BLINK Q SLOW Blink the Q LED slowly.

address	machine code			
(R0)	Hex	Binary	Mnemonic	Human readable comments
0000	F8	1111 1000	LDI	LoaD Immediately...
0001	08	0000 1000	8	...8
0002	B2	1011 0010	PHI R2	Put it in the HI half of register 2
0003	22	0010 0010	DEC R2	DECrement register 2
0004	92	1001 0010	GHI R2	Get the HI half of register 2
0005	3A	0011 1010	BNZ	Branch if it is Not Zero...
0006	03	0000 0011	3	...to address 3 (loops 8 x 256 = 2048 times)
0007	CD	1100 1101	LSQ	Long Skip over next 2 instructions if Q=1...
0008	7B	0111 1011	SEQ	...if Q was 0, then SEt Q=1
0009	38	0011 1000	SKP	...and SKIP next instruction
000A	7A	0111 1010	REQ	else Q was 1, so REset Q=0
000B	30	0011 0000	BR	BRanch unconditionally...
000C	00	0000 0000	0	...to address 0 (to begin again)

The instructions from 0000 to 0002 set up a 16-bit counter in register 2, and set it to hex 08xx. Instructions 0003 to 0006 are a loop; so register 2 counts down until it gets to hex 00FF. The high byte is then 00 (and the low byte is left at FF); so the BNZ instruction stops looping and the program continues at address 0007.

The LSQ instruction tests Q; if Q=1, it skips ahead 2 and resets Q to 0. If Q=0, it continues to set Q to 1. The result is to "toggle" Q on/off with each pass. Finally, the Branch instruction at 000B jumps back to the beginning to repeat the whole thing forever. You can control the speed by changing the value at 0001, or by adjusting trimpot R1.

PROGRAM 3 -- READ SWITCHES AND DISPLAY VALUE IN LEDS

This program is a bit more complex. It reads the 8 data switches, and displays their settings on the 8 LEDs. It tests the Membership Card's ability to read and write to the switches and lights. Run it with S8 up (write).

address	machine code			
(R0)	Hex	Binary	Mnemonic	Comments
0000	E1	1110 0001	SEX 1	Set X register to 1 (OMG! 1802's have sex instructions)
0001	90	1001 0000	GHI 0	Get HIgh byte of register 0 in D (sets D=0)
0002	B1	1011 0001	PHI 1	Put D in HIgh byte of register 1 (so R1=00xx)
0003	F8	1111 1000	LDI	Load D Immediately with...
0004	10	0001 0000	10h	...10 hex
0005	A1	1010 0001	PLO 1	Put D in the LOw half of R1 (so R1 is now 0010)
0006	6C	0110 1100	INP 4	INPut port 4 (front panel switches) & write it to memory at (R1)
0007	64	0110 0100	OUT 4	OUTput to port 4 (front panel LEDs) contents of memory at (R1)
0008	30	0011 0000	BR	BRanch unconditionally...
0009	00	0000 0000	0	...to address 0000

More Programming

Toggling in programs with the Front Panel gets old fast, doesn't it? It's really just there for testing and debugging. For bigger programs, you'll want to use either the Parallel or Serial port to download programs from your PC.

If your PC has a parallel port, all you need is a standard DB-25 male-to-male cable. Download the program at http://www.sunrise-ev.com/MembershipCard/ELF-LINK.BAS. Set Read/Write switch S8 to READ (down), all other switches UP, and run the ELF-LINK program. This is a Microsoft QuickBASIC program that can completely control the Membership Card; it can operate all the switches, monitor all the LEDs, load and save programs, etc. Even if you don't like BASIC, the program is posted in plain ASCII so you can write a version in your favorite programming language. Now, who will be the first to translate it into C?

If you don't have a parallel port, there are many other ways to automate control and program loading. See the web page at http://www.sunrise-ev.com/membershipcard.htm#projects for some of the ways to do it.

The current Rev.G Membership Card has a new way. It adds an RS-232/TTL serial interface, and a second memory chip socket. It holds an EPROM with software to talk to a PC's RS-232 port (or USB port with a serial-to-USB adapter). So far, we know it works with the Elf2K EPROM from Spare Time Gizmos (http://www.sparetimegizmos.com/Hardware/Elf2K.htm). I'm exploring other options as well. I plan to offer an expansion kit with a preprogrammed EPROM and the extra RAM for U8 (as soon as I get a round tuit).

In Case of Difficulty...

The usual problems are bad solder joints. Either a pin hasn't been soldered, or there is too much solder so it shorts to another pin.

Next, look for something too tall on the Membership Card that is shorting to the back of the Front Panel. Likely candidates are the pin headers P2 and P3.

Look for parts installed backwards (diodes or ICs or SIP resistors), or in the wrong place (like resistors).

If all this looks right, make sure you have +3v to +5v power to the board. The voltage on U4 pin 12 (RUN) should be high to enable the oscillator. U4 pin 11 will be 1/2 the supply voltage if the oscillator is running.

Switch to RUN, and look for signals on the 1802 TPA, TPB, /MRD, and SC0 pins. Even if it is executing nonsense (no program), these pins will still be pulsing high/low as the 1802 tries to read memory. Also check to see that all the 1802 MA0-7 and BUS0-7 pins are going high and low (so none are open or shorted).

Check the voltage on 1802 /EF4 input. It should be high, and go low when you push the IN button. If it's the opposite, you have the pushbutton in backwards. Here is how LOAD mode works:

- Press IN. The 1802 /EF4 pin and flip-flop U5B pin 11 go low.
- Release IN. /EF4 goes high. U5B sets, so its /Q output pin 12 goes low. This makes 1802 /DMA-IN go low.
- The 1802 does a DMA in cycle. It puts an address on MA0-7, and pulses /MWR low to write to memory.
- During a Write cycle, /MRD is high. N2.or.LOAD is also high as we are in LOAD mode, so U4C pin 10 is low. This enables U6 to put the 8 DATA switches on BUS0-BUS7, where they get written into memory.
- 1802 SC1 is high during a DMA cycle; this resets U5B. The 1802 then does a read cycle (/MRD low) to read the byte just written to memory. /MRD low lets U5A set when TPB goes high, to latch the byte in U7.

Still doesn't work? Email me for help at leeahart@earthlink.net. Failing that, send it to me and I'll fix it! :-)

Last Writes

I'm still working on this manual, to add better assembly directions, instructions for troubleshooting, and ways to load and save programs from your PC without all that flippin' switch flipping. I hope what I have so far is enough to get you going. If not, please contact me with comments, corrections, questions, or ways to improve it. My contact information is on the first page. Watch the websites listed below for the latest details!

Changes along the way:

July 2010: Rev.A board: Changed R8-R10 from 100k to 470k.

Aug 2010: Rev.B: Added Q1 and C6. Changed I/O port from 5/7 to 4/5/6/7 to match Elf. Added RUN to P4.

Nov 2010: Changed R4 and R8-10 from 470k to 499k, and C6 from 82pf to 100pf (for expediency; I ran out of 470k and 82pf parts, and had lots of 499k and 100pf).

Jul 2011: Rev.C: Fixed silkscreen I/O port names; IN5/7 now INP4, OUT5/OUT7 now OUT4. (IN and OUT ports still respond to any port number from 4-7). Added "Cover Card" for a more finished appearance.

Aug 2011: Corrected P2 jumper chart on schematic. For 6116, 2716, 2732; P2 was 2-3, changed to 3-4. For 2764; P2 was 3-4, changed to 2-3. For 27128 and 27256; P2 was 1-2 and 3-4, changed to 2-3 and 4-5.

Sept 2011: Added power supply info.

Jan 2012: Added photos of finished boards.

Mar 2012: Added "cheat sheet" summary page.

Apr 2012: R1 changed from 1meg to 500k (I ran out of 1meg pots).

June 2012: Rev.D board: R1 went back to 1meg. R2 changed to 5.6k (to raise maximum clock frequency). Added D12 (so power is available on DB25 connector J2). Added R14 (to use J2 pin 1 for serial input). Added jumpers A/B, O0, O1, O2, and O2 (to use J2 for general purpose I/O instead of PC parallel I/O).

Aug 2012: Ran out of parts; so changed C5 from 0.047F to 0.022F, and R2 and R14 from 5.6k to 6.8k.

Jan 2013: Rev.E: Made room for a bigger supercapacitor at C5 (now 0.10F or 0.22F).

May 2013: Rev.F: Changed C1 to ceramic resonator (more stable frequency). Removed R2 to make room.

Feb 2014: Rev.G: Added 32k RAM (U8; under U2); can have 32k RAM plus 32k EPROM! Add serial I/O using Q and EF3 to D-connector (Q2, Q3, C8, D13, R15). Add 2-color LED at D8 (shows Q and EF3).

Mar 2015: Rev.H: Changed Q1 from 2N7000 to FJN3301, and R5 from 100k to 10k for faster /A15 rise time. Added D14 for power-on CLEAR even if front panel switches are set to RUN.

Links for more about the 1802 and ELF computers:

http://www.sunrise-ev.com/membershipcard.htm
This is my website, with ordering information, manual updates, schematics, cheat sheets, and more.

http://incolor.inebraska.com/bill_r/elf/html/elf-1-33.htm
This is the Aug 1976 Popular Electronics article that introduced the ELF to hobbyists. Most of it applies directly to the Membership Card.

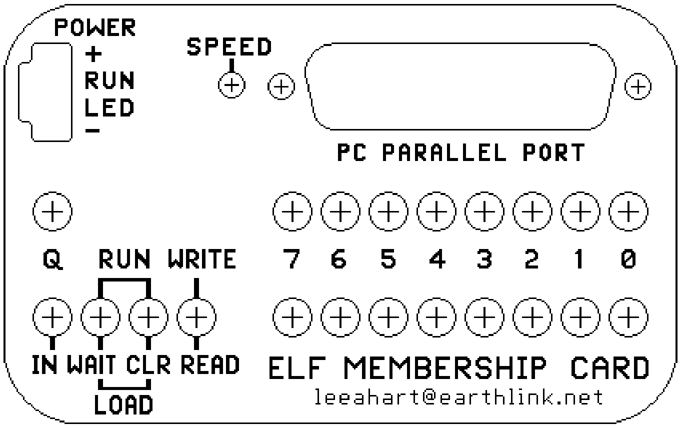
http://datasheets.chipdb.org/RCA/MPM-201B_CDP1802_Users_Manual_Nov77.pdf
An online copy of RCA's User Manual for the 1802. "Must read" reference material!

http://www.ittybittycomputers.com/IttyBitty/ShortCor.htm
"A Short Course in Programming" by Tom Pittman. An excellent introduction to programming the 1802.

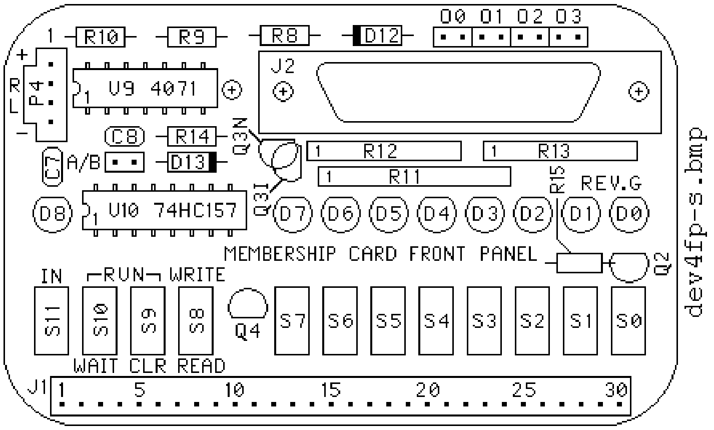
http://www.cosmacelf.com
The COSMAC ELF "fan club", with lots of information on the many commercial and hobbyist variants.

http://www.retrotechnology.com/memship/memship.html
Herb Johnson's website on the Membership board. He built one of the first ones, and has done a great job documenting it (many thanks, Herb)! There are loads of design notes, history, software examples, etc.

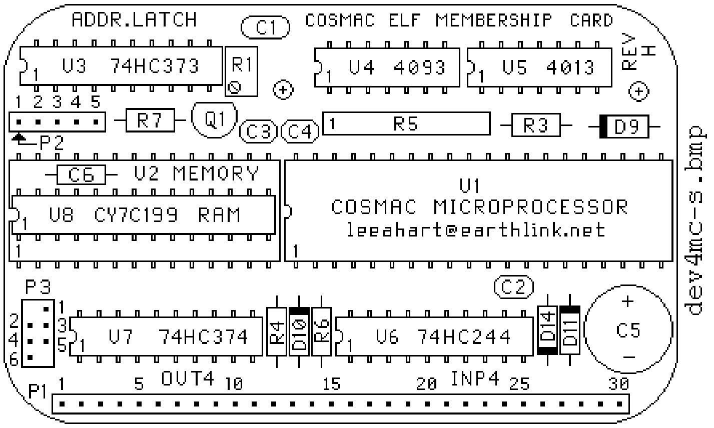
"Mugshots" of the Elf Membership Card 3-board Set



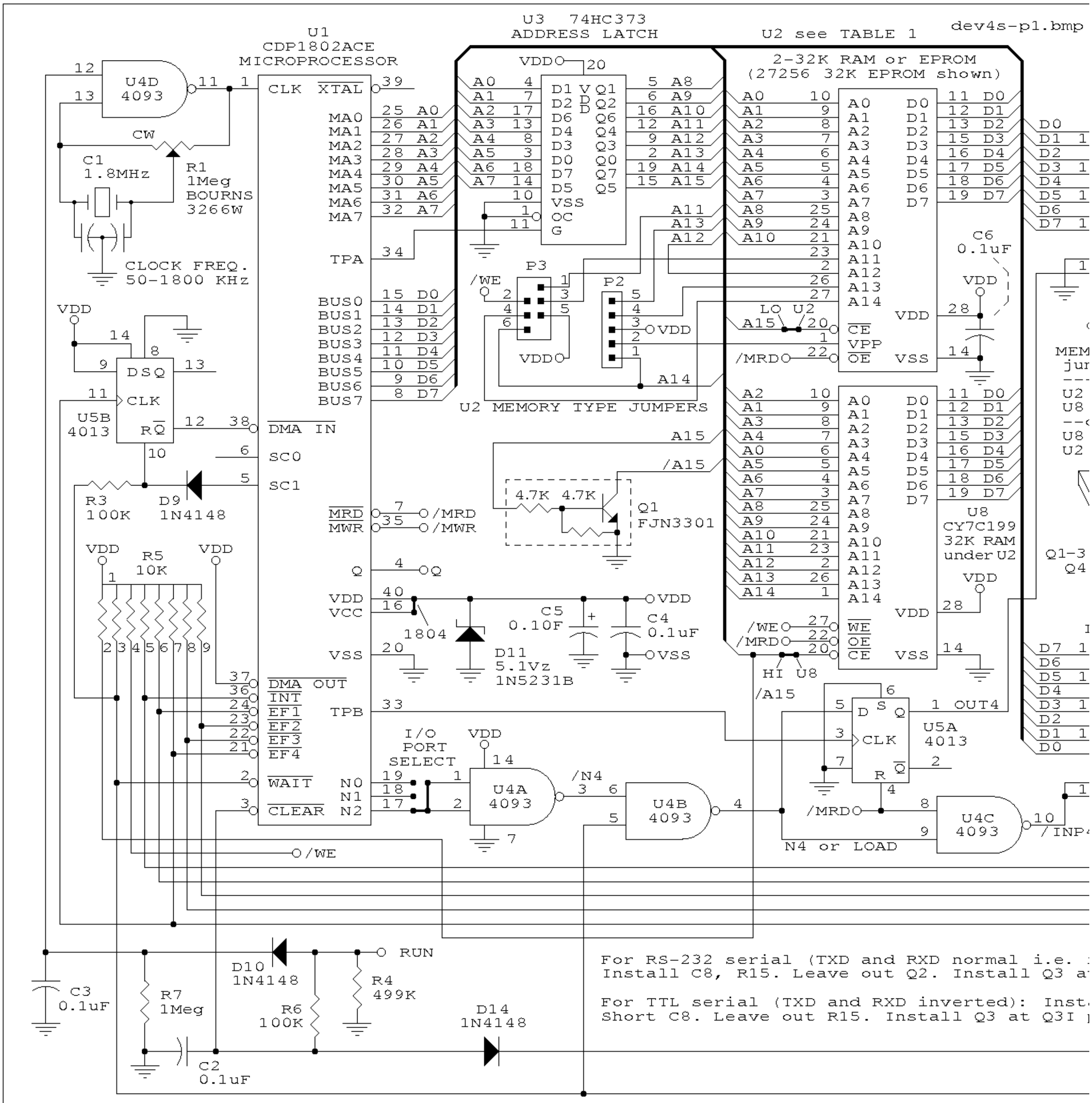
Cover Card (1:1 scale): Can use as a drilling template.



Front Panel Card (1:1 scale): Part location and placement guide.



Membership Card (1:1 scale): Part location and placement guide.



For RS-232 serial (TXD and RXD normal i.e.
Install C8, R15. Leave out Q2. Install Q3 a

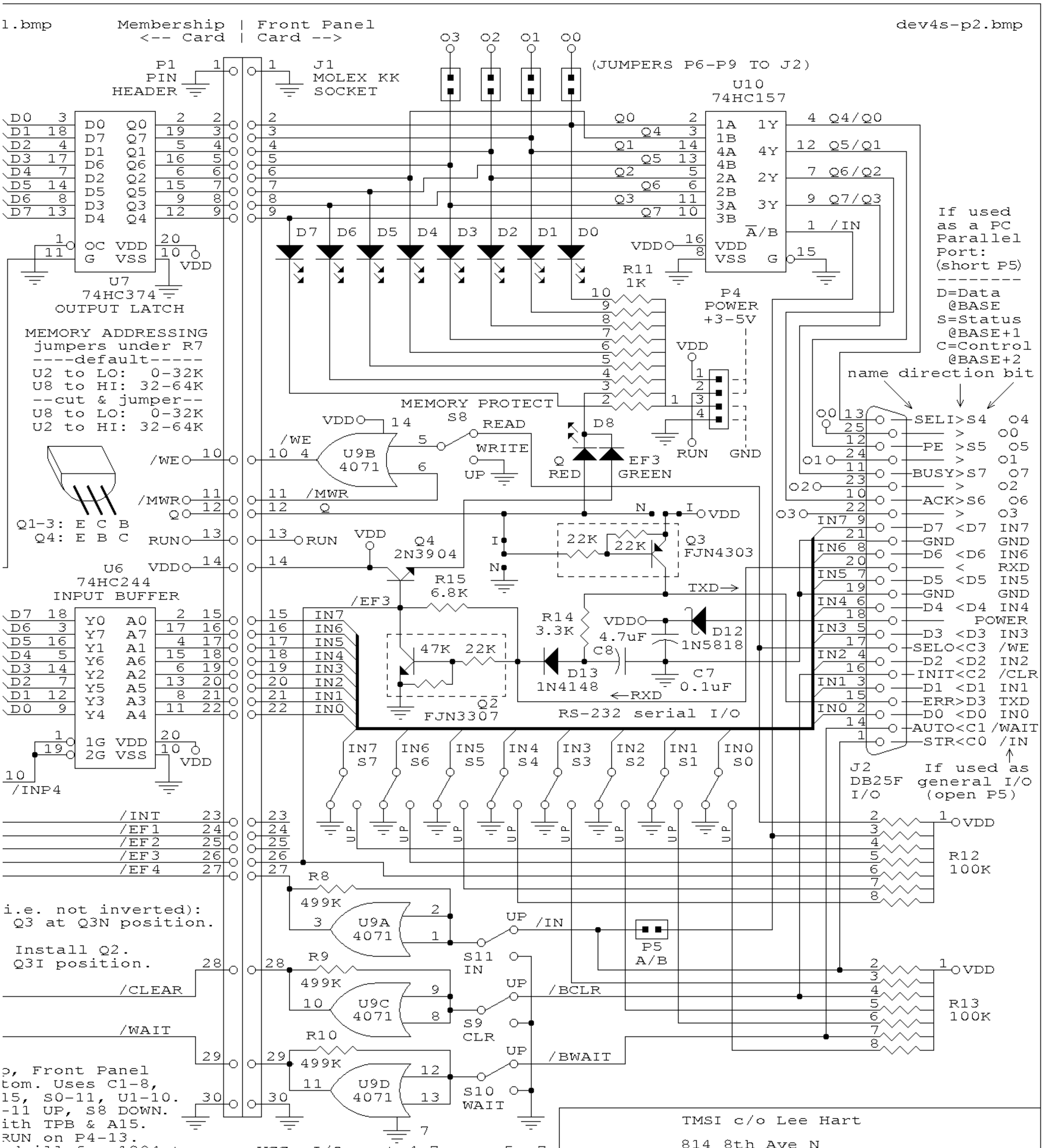
For TTL serial (TXD and RXD inverted): Inst
Short C8. Leave out R15. Install Q3 at Q3I

TABLE 1: U2 part#	pin#s:	Pin Numbers 1	23	26	27	Jumpers P2	P3
62256	32K RAM	A14	A11	A13	/WE	1-2,4-5	1-3,2-4
6264	8K RAM		A11	VDD	/WE	3-4	1-3,2-4
* 6116	2K RAM		/WE	VDD		3-4	2-3
27256	32K EPROM	VDD	A11	A13	A14	2-3,4-5	1-3,4-6
27128	16K EPROM	VDD	A11	A13	VDD	2-3,4-5	1-3,4-5
2764	8K EPROM	VDD	A11	NC	VDD	2-3	1-3,4-5
* 2732	4K EPROM		A11	VDD		3-4	1-3
* 2716	2K EPROM		VDD	VDD		3-4	3-5

* is a 24-pin chip: Put its pin 1 into socket pin 3.

P4 power pin 1 VDD: +3 to +5 VDC (3 AA cells etc.)
connector pin 2 RUN: to pin 1 to run, open stops osc.
pin 3 LED: to pin 4 enables LEDs, open=off.
pin 4 COMMON for power (GND, VSS).

Notes:
1. Boards stack, component side up, Fr
on top, Membership Card on bottom.
D0-14, J1-2, P1-9, Q1-4, R1,3-15, S
2. PC Parallel Port: Set S0-7, S9-11 U
3. Rev.B: Add Q1 to gate U2 /CE with T
R8-10 was 100K, now 470K. Add RUN o
Add C6=82pF. Add feedthru "O"; dril
4. Rev.C: R4 and R8-10 were 470K, now
5. Rev.D: R2 was 15K, now 6.8K. Add R1
names. Swap P1-J1 designators. Add
Add N0-N2 & Q1 options to stack 2 M
6. Rev.E: Add room for larger C5 (up t
7. Rev.F: Remove R2. Replace C1 with 1
8. Rev.G: Add U8. Change C6 to bypass
D8 is red for Q=1 (TXD active), gre
9. Rev.H: Add D14, move R4. R5 was 100



i.e. not inverted):
Q3 at Q3N position.

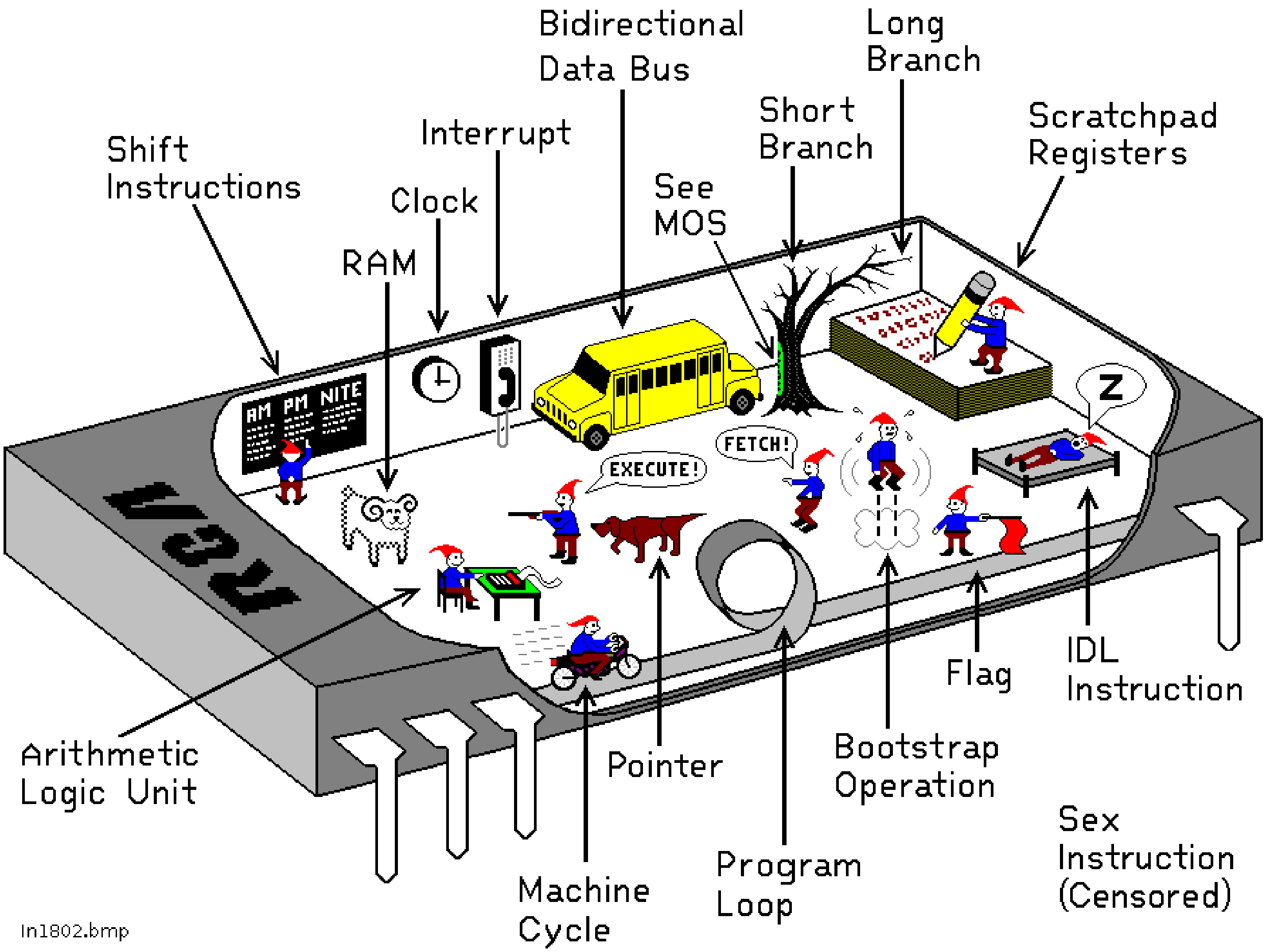
Install Q2.
Q3I position.

ɔ, Front Panel
tom. Uses C1-8,
15, S0-11, U1-10.
-11 UP, S8 DOWN.
ith TPB & A15.
RUN on P4-13.
drill for 1804 to open VCC. I/O port 4-7, was 5or7.
now 499K. C6 was 82pF, now 100pF.
dd R14 for RS-232 input. Add D12. Add J2 signal
Add jumpers A/B, Q0-Q3 for 8 outputs on J2.
< 2 Membership Cards w/o memory or I/O conflict.
(up to 0.22F).
ith 1.8 MHz ceramic resonator.
pass cap. Add serial I/O to J2 using Q and EF3.
, green for EF3 pin=1 (RXD active).
s 100K, now 10K. Replace "O" with jumper; open=1804.

TMSI c/o Lee Hart
814 8th Ave N
Sartell MN 56377
leeahart@earthlink.net

Title 1802 Membership Card and Front Panel Card		
Size B	Document Number C:\ORCAD\SHEET\1802\DEV4H.SCH	REV H
Date: March 29, 2015	Sheet 1 of 1	

The Inside Story...



How the 1802 in an Elf computer REALLY works!

Membership Card Operating Summary

©2014 by TMSI, 814 8th Ave N, Sartell MN 56377 <leeahart@earthlink.net>

Register Summary

D	8 bit	D register (Accumulator)	DF	1 bit	Data flag (ALU carry)
Rn	16 bit	1 of 16 registers (n=0-F)	P	4 bit	Program register select
X	4 bit	Data register select	T	8 bit	Saved X,P after interrupt
IE	1 bit	Interrupt Enable	Q	1 bit	Output flip-flop

Interrupt: Finish executing current instruction, then save X and P in T, then set P=1, X=2 and IE=0 (Inhibit). Execution continues at M(R1).

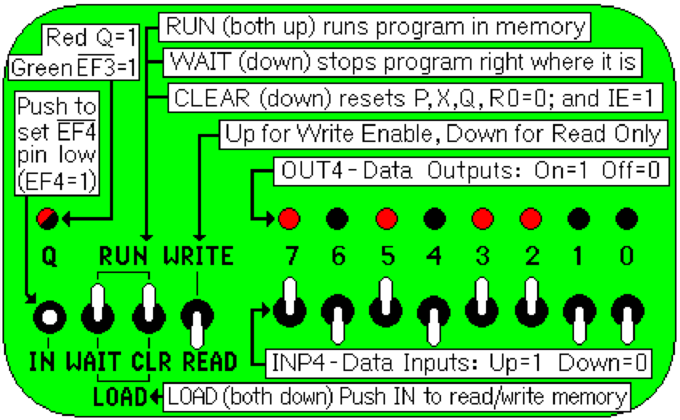
DMA: Finish executing current instruction, then use R0 as memory address for data transfer. DMA-IN writes data from bus to memory, or DMA-OUT reads data from memory to Bus. Then increment R0.

If simultaneous, DMA-IN has highest priority, then DMA-OUT, then Interrupt.

EF1-4: External 1-bit flags that can be tested by Branch instructions.

Q: External 1-bit output that can be set, reset, and tested by Branch and Skip instructions.

http://www.sunrise-ev.com/membershipcard.htm


Red Q=1
Green EF3=1
Push to set EF4 pin low (EF4=1)
RUN (both up) runs program in memory
WAIT (down) stops program right where it is
CLEAR (down) resets P,X,Q,R0=0; and IE=1
Up for Write Enable, Down for Read Only
OUT4 - Data Outputs: On=1 Off=0
Q
RUN
WRITE
7 6 5 4 3 2 1 0
IN WAIT CLR READ
INP4 - Data Inputs: Up=1 Down=0
LOAD
LOAD (both down) Push IN to read/write memory

Register Operations

1n .. INC Rn	Rn=Rn+1
2n .. DEC Rn	Rn=Rn−1
60 .. IRX Rx	Rx=Rx+1
8n .. GLO Rn	D=Rn(low8)
9n .. GHI Rn	D=Rn(high8)
An .. PLO Rn	Rn(low8)=D
Bn .. PHI Rn	Rn(high8)=D

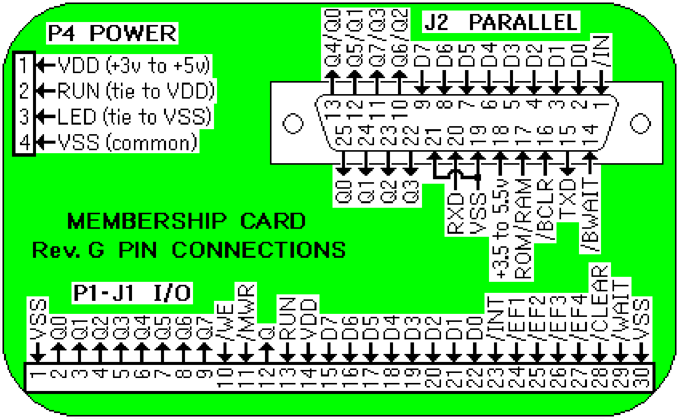
Memory Load / Store

0n .. LDN Rn	D=M(Rn) for n not 0
4n .. LDA Rn	D=M(Rn), then inc Rn
F0 .. LDX	D=M(Rx)
72 .. LDXA	D=M(Rx), then inc Rx
F8 nn .. LDI nn	D=M(Rp)
5n .. STR Rn	M(Rn)=D
73 .. STXD	M(Rx)=D, then dec Rx

Short Branch Instructions (nn is low byte of address on same page)

30 nn..BR nn	branch to nn	38 nn..NBR nn	no branch (skip nn)
31 nn..BQ nn	branch if Q=1	39 nn..BNQ nn	branch if Q=0
32 nn..BZ nn	branch if D=0	3A nn..BNZ nn	branch if D not 0
33 nn..BDF nn	branch if DF=1,	3B nn..BNF nn	branch if DF=0,
or BPZ nn	br. if positive or 0,	or BM nn	branch if minus,
or BGE nn	br.if equal or greater	or BL nn	branch if less than
34 nn..B1 nn	branch if EF1=1	3C nn..BN1 nn	branch if EF1=0
35 nn..B2 nn	branch if EF2=1	3D nn..BN2 nn	branch if EF2=0
36 nn..B3 nn	branch if EF3=1	3E nn..BN3 nn	branch if EF3=0
37 nn..B4 nn	branch if EF4=1	3F nn..BN4 nn	branch if EF4=0

name↑ dots=bus cycles opcode↑ ↑2nd byte ↑description



Long Branch & Skip (hhll=address high byte, low byte)

C0 hhll...LBR hhll	branch to hhll	C8 hhll...NLBR hhll	no br.(skip 2)
C1 hhll...LBQ hhll	branch if Q=1	C9 hhll...LBNQ hhll	branch if Q=0
C2 hhll...LBZ hhll	branch if D=0	CA hhll...LBNZ hhll	branch if D not 0
C3 hhll...LBDF hhll	branch if DF=1	CB hhll...LBNF hhll	branch if DF=0
C4 ...NOP	no operation	CC ...LSIE	skip 2 if IE=1
C5 ...LSNQ	skip 2 if Q=0	CD ...LSQ	skip 2 if Q=1
C6 ...LSNZ	skip 2 if Dnot0	CE ...LSZ	skip 2 if D=0
C7 ...LSNF	skip 2 if DF=0	CF ...LSDF	skip 2 if DF=1

Logic, Arithmetic, and Shift (result in D, carry/borrow/shift in DF)

F1 .. OR	D=D or M(Rx)	F9 nn .. ORI nn	D=D or nn
F2 .. AND	D=D and M(Rx)	FA nn .. ANI nn	D=D and nn
F3 .. XOR	D=D xor M(Rx)	FB nn .. XRI nn	D=D xor nn
F4 .. ADD	D=D+M(Rx)	FC nn .. ADI nn	D=D+nn
74 .. ADC	D=D+M(Rx)+DF	7C nn .. ADCI nn	D=D+nn+DF
F5 .. SD	D=M(Rx)−D	FD nn .. SDI nn	D=nn−D
75 .. SDB	D=M(Rx)−D−not DF	7D nn .. SDBI nn	D=nn−D−not DF
F7 .. SM	D=D−M(Rx)	FF nn .. SMI nn	D=D−nn
77 .. SMB	D=D−M(Rx)−not DF	7F nn .. SMBI nn	D=D−nn−not DF

only add, subtract, and shift instructions change DF

Hex-Binary-Decimal Conversion

Hex	Binary	Dec
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Hex-Binary-ASCII Conversion

MSD / LSD	0x 000	1x 001	2x 010	3x 011	4x 100	5x 101	6x 110	7x 111
x0 0000	NUL	DLE	(sp)	0	@	P	`	p
x1 0001	SOH	DC1	!	1	A	Q	a	q
x2 0010	STX	DC2	"	2	B	R	b	r
x3 0011	ETX	DC3	#	3	C	S	c	s
x4 0100	EOT	DC4	$	4	D	T	d	t
x5 0101	ENQ	NAK	%	5	E	U	e	u
x6 0110	ACK	SYN	&	6	F	V	f	v
x7 0111	BEL	ETB	'	7	G	W	g	w
x8 1000	BS	CAN	(	8	H	X	h	x
x9 1001	HT	EM	)	9	I	Y	i	y
xA 1010	LF	SUB	*	:	J	Z	j	z
xB 1011	VT	ESC	+	;	K	[	k	{
xC 1100	FF	FS	,	<	L	\	l	¦
xD 1101	CR	GS	−	=	M	]	m	}
xE 1110	SO	RS	.	>	N	^	n	~
xF 1111	SI	US	/	?	O	_	o	DEL

Logic, Arithmetic, and Shift Instructions (continued)

F6 .. SHR	shift D right;
	D(msb)=0, DF=D(lsb)
		76 .. SHRC
		or RSHR
			shift D right with carry;
			D(msb)=DF, DF=D(lsb)

FE .. SHL	shift D left;
	D(lsb)=0, DF=D(msb)
		7E .. SHLC
		or RSHL
			shift D left with carry;
			D(lsb)=DF, DF=D(msb)

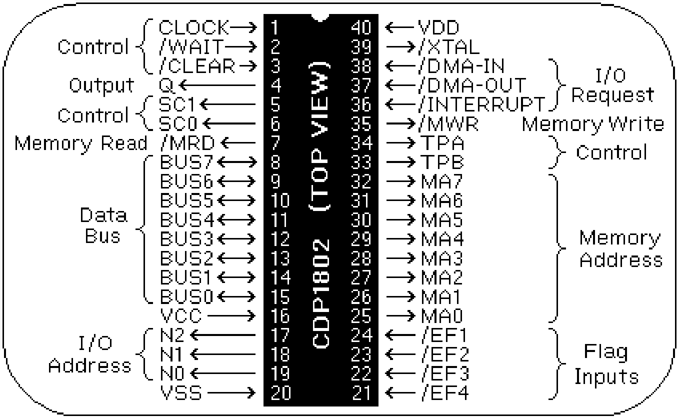
Control Instructions

00... IDL	wait for INT or DMA	78 .. SAVE	M(Rx)=T
68 ..	reserved for 1804/5/6	7A .. REQ	reset Q=0
70 .. RET	X,P=M(Rx); inc Rx; IE=1	7B .. SEQ	set Q=1
71 .. DIS	X,P=M(Rx); inc Rx; IE=0	Dn .. SEP n	set P=n
79 .. MARK	T=X,P; M(R2)=X,P; X=P; dec R2	En .. SEX n	set X=n

Input/Output Instructions (with N output on pins N0, N1, and N2)

61 .. OUT1	bus=M(Rx), N=1; inc Rx	69 .. INP1	M(Rx)=D=bus, N=1
62 .. OUT2	bus=M(Rx), N=2; inc Rx	6A .. INP2	M(Rx)=D=bus, N=2
63 .. OUT3	bus=M(Rx), N=3; inc Rx	6B .. INP3	M(Rx)=D=bus, N=3
64 .. OUT4	bus=M(Rx), N=4; inc Rx	6C .. INP4	M(Rx)=D=bus, N=4
65 .. OUT5	bus=M(Rx), N=5; inc Rx	6D .. INP5	M(Rx)=D=bus, N=5
66 .. OUT6	bus=M(Rx), N=6; inc Rx	6E .. INP6	M(Rx)=D=bus, N=6
67 .. OUT7	bus=M(Rx), N=7; inc Rx	6F .. INP7	M(Rx)=D=bus, N=7

Add: DF=1 if carry. Subtract: DF=0 if borrow (negative).



Membership Card Summary: Keep the left half in the manual, in case you need to make a copy later.

Cut the right half out, fold, and put in the Altoids case as a quick reference (& so board won't short to case).

Membership Card Operating Summary

©2014 by TMSI, 814 8th Ave N, Sartell MN 56377<leeahart@earthlink.net>

Register Summary

D	8 bit	D register (Accumulator)	DF	1 bit	Data flag (ALU carry)
Rn	16 bit	1 of 16 registers (n=0-F)	P	4 bit	Program register select
X	4 bit	Data register select	T	8 bit	Saved X,P after interrupt
IE	1 bit	Interrupt Enable	Q	1 bit	Output flip-flop

Interrupt: Finish executing current instruction, then save X and P in T, then set P=1, X=2 and IE=0 (Inhibit). Execution continues at M(R1).

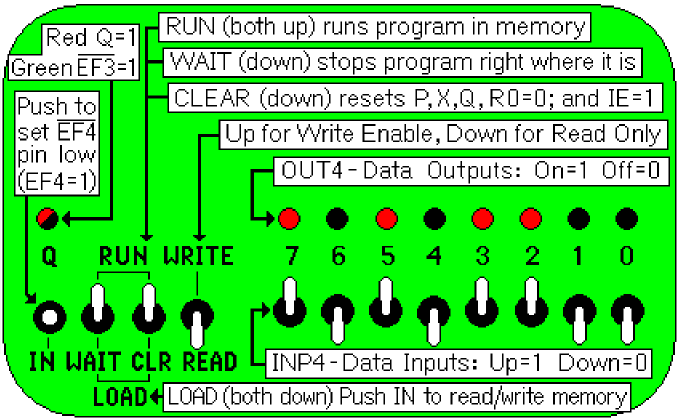
DMA: Finish executing current instruction, then use R0 as memory address for data transfer. DMA-IN writes data from bus to memory, or DMA-OUT reads data from memory to Bus. Then increment R0.

If simultaneous, DMA-IN has highest priority, then DMA-OUT, then Interrupt.

EF1-4: External 1-bit flags that can be tested by Branch instructions.

Q: External 1-bit output that can be set, reset, and tested by Branch and Skip instructions.

http://www.sunrise-ev.com/membershipcard.htm


Red Q=1
Green EF3=1
Push to set EF4 pin low (EF4=1)
RUN (both up) runs program in memory
WAIT (down) stops program right where it is
CLEAR (down) resets P,X,Q, R0=0; and IE=1
Up for Write Enable, Down for Read Only
OUT4 - Data Outputs: On=1 Off=0
Q
RUN
WRITE
7
6
5
4
3
2
1
0
IN
WAIT
CLR
READ
INP4 - Data Inputs: Up=1 Down=0
LOAD
LOAD (both down) Push IN to read/write memory

Register Operations

1n .. INC Rn		Rn=Rn+1
2n .. DEC Rn		Rn=Rn−1
60 .. IRX Rx		Rx=Rx+1
8n .. GLO Rn		D=Rn(low8)
9n .. GHI Rn		D=Rn(high8)
An .. PLO Rn		Rn(low8)=D
Bn .. PHI Rn		Rn(high8)=D

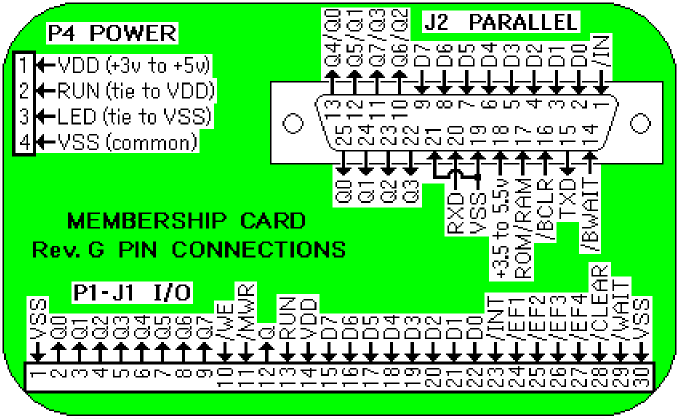
Memory Load / Store

0n	.. LDN Rn	D=M(Rn) for n not 0
4n	.. LDA Rn	D=M(Rn), then inc Rn
F0	.. LDX	D=M(Rx)
72	.. LDXA	D=M(Rx), then inc Rx
F8 nn	.. LDI nn	D=M(Rp)
5n	.. STR Rn	M(Rn)=D
73	.. STXD	M(Rx)=D, then dec Rx

Short Branch Instructions (nn is low byte of address on same page)

30 nn..BR	nn	branch to nn	38 nn..NBR	nn	no branch (skip nn)
31 nn..BQ	nn	branch if Q=1	39 nn..BNQ	nn	branch if Q=0
32 nn..BZ	nn	branch if D=0	3A nn..BNZ	nn	branch if D not 0
33 nn..BDF	nn	branch if DF=1,	3B nn..BNF	nn	branch if DF=0,
or BPZ	nn	br. if positive or 0,	or BM	nn	branch if minus,
or BGE	nn	br.if equal or greater	or BL	nn	branch if less than
34 nn..B1	nn	branch if EF1=1	3C nn..BN1	nn	branch if EF1=0
35 nn..B2	nn	branch if EF2=1	3D nn..BN2	nn	branch if EF2=0
36 nn..B3	nn	branch if EF3=1	3E nn..BN3	nn	branch if EF3=0
37 nn..B4	nn	branch if EF4=1	3F nn..BN4	nn	branch if EF4=0

name↑ dots=bus cycles opcode↑ ↑2nd byte ↑description



Long Branch & Skip (hhll=address high byte, low byte)

C0 hhll...LBR	hhll	branch to hhll	C8 hhll...NLBR	hhll	no br.(skip 2)
C1 hhll...LBQ	hhll	branch if Q=1	C9 hhll...LBNQ	hhll	branch if Q=0
C2 hhll...LBZ	hhll	branch if D=0	CA hhll...LBNZ	hhll	branch if D not 0
C3 hhll...LBDF	hhll	branch if DF=1	CB hhll...LBNF	hhll	branch if DF=0
C4 ...NOP		no operation	CC ...LSIE		skip 2 if IE=1
C5 ...LSNQ		skip 2 if Q=0	CD ...LSQ		skip 2 if Q=1
C6 ...LSNZ		skip 2 if D not 0	CE ...LSZ		skip 2 if D=0
C7 ...LSNF		skip 2 if DF=0	CF ...LSDF		skip 2 if DF=1

Logic, Arithmetic, and Shift (result in D, carry/borrow/shift in DF)

F1 .. OR	D=D or M(Rx)	F9 nn .. ORI	nn	D=D or nn
F2 .. AND	D=D and M(Rx)	FA nn .. ANI	nn	D=D and nn
F3 .. XOR	D=D xor M(Rx)	FB nn .. XRI	nn	D=D xor nn
F4 .. ADD	D=D+M(Rx)	FC nn .. ADI	nn	D=D+nn
74 .. ADC	D=D+M(Rx)+DF	7C nn .. ADCI	nn	D=D+nn+DF
F5 .. SD	D=M(Rx)−D	FD nn .. SDI	nn	D=nn−D
75 .. SDB	D=M(Rx)−D−not DF	7D nn .. SDBI	nn	D=nn−D−not DF
F7 .. SM	D=D−M(Rx)	FF nn .. SMI	nn	D=D−nn
77 .. SMB	D=D−M(Rx)−not DF	7F nn .. SMBI	nn	D=D−nn−not DF

only add, subtract, and shift instructions change DF

Hex-Binary-Decimal Conversion

Hex	Binary	Dec
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Hex-Binary-ASCII Conversion

MSD / LSD	0x 000	1x 001	2x 010	3x 011	4x 100	5x 101	6x 110	7x 111
x0 0000	NUL	DLE	(sp)	0	@	P	`	p
x1 0001	SOH	DC1	!	1	A	Q	a	q
x2 0010	STX	DC2	"	2	B	R	b	r
x3 0011	ETX	DC3	#	3	C	S	c	s
x4 0100	EOT	DC4	$	4	D	T	d	t
x5 0101	ENQ	NAK	%	5	E	U	e	u
x6 0110	ACK	SYN	&	6	F	V	f	v
x7 0111	BEL	ETB	'	7	G	W	g	w
x8 1000	BS	CAN	(	8	H	X	h	x
x9 1001	HT	EM	)	9	I	Y	i	y
xA 1010	LF	SUB	*	:	J	Z	j	z
xB 1011	VT	ESC	+	;	K	[	k	{
xC 1100	FF	FS	,	<	L	\	l	¦
xD 1101	CR	GS	-	=	M	]	m	}
xE 1110	SO	RS	.	>	N	^	n	~
xF 1111	SI	US	/	?	O	_	o	DEL

Logic, Arithmetic, and Shift Instructions (continued)

F6 .. SHR	shift D right;
	D(msb)=0, DF=D(lsb)
		76 .. SHRC
		or RSHR
			shift D right with carry;
			D(msb)=DF, DF=D(lsb)

FE .. SHL	shift D left;
	D(lsb)=0, DF=D(msb)
		7E .. SHLC
		or RSHL
			shift D left with carry;
			D(lsb)=DF, DF=D(msb)

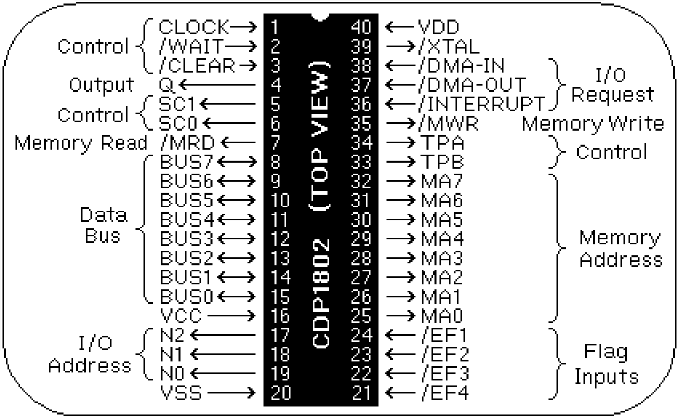
Control Instructions

00.... IDL	wait for INT or DMA	78 .. SAVE		M(Rx)=T
68 ..	reserved for 1804/5/6	7A .. REQ		reset Q=0
70 .. RET	X,P=M(Rx); inc Rx; IE=1	7B .. SEQ		set Q=1
71 .. DIS	X,P=M(Rx); inc Rx; IE=0	Dn .. SEP	n	set P=n
79 .. MARK	T=X,P; M(R2)=X,P; X=P; dec R2	En .. SEX	n	set X=n

Input/Output Instructions (with N output on pins N0, N1, and N2)

61 .. OUT1	bus=M(Rx), N=1; inc Rx	69 .. INP1	M(Rx)=D=bus, N=1
62 .. OUT2	bus=M(Rx), N=2; inc Rx	6A .. INP2	M(Rx)=D=bus, N=2
63 .. OUT3	bus=M(Rx), N=3; inc Rx	6B .. INP3	M(Rx)=D=bus, N=3
64 .. OUT4	bus=M(Rx), N=4; inc Rx	6C .. INP4	M(Rx)=D=bus, N=4
65 .. OUT5	bus=M(Rx), N=5; inc Rx	6D .. INP5	M(Rx)=D=bus, N=5
66 .. OUT6	bus=M(Rx), N=6; inc Rx	6E .. INP6	M(Rx)=D=bus, N=6
67 .. OUT7	bus=M(Rx), N=7; inc Rx	6F .. INP7	M(Rx)=D=bus, N=7

Add: DF=1 if carry. Subtract: DF=0 if borrow (negative).



Membership Card Summary: Keep the left half in the manual, in case you need to make a copy later.

Cut the right half out, fold, and put in the Altoids case as a quick reference (& so board won't short to case).