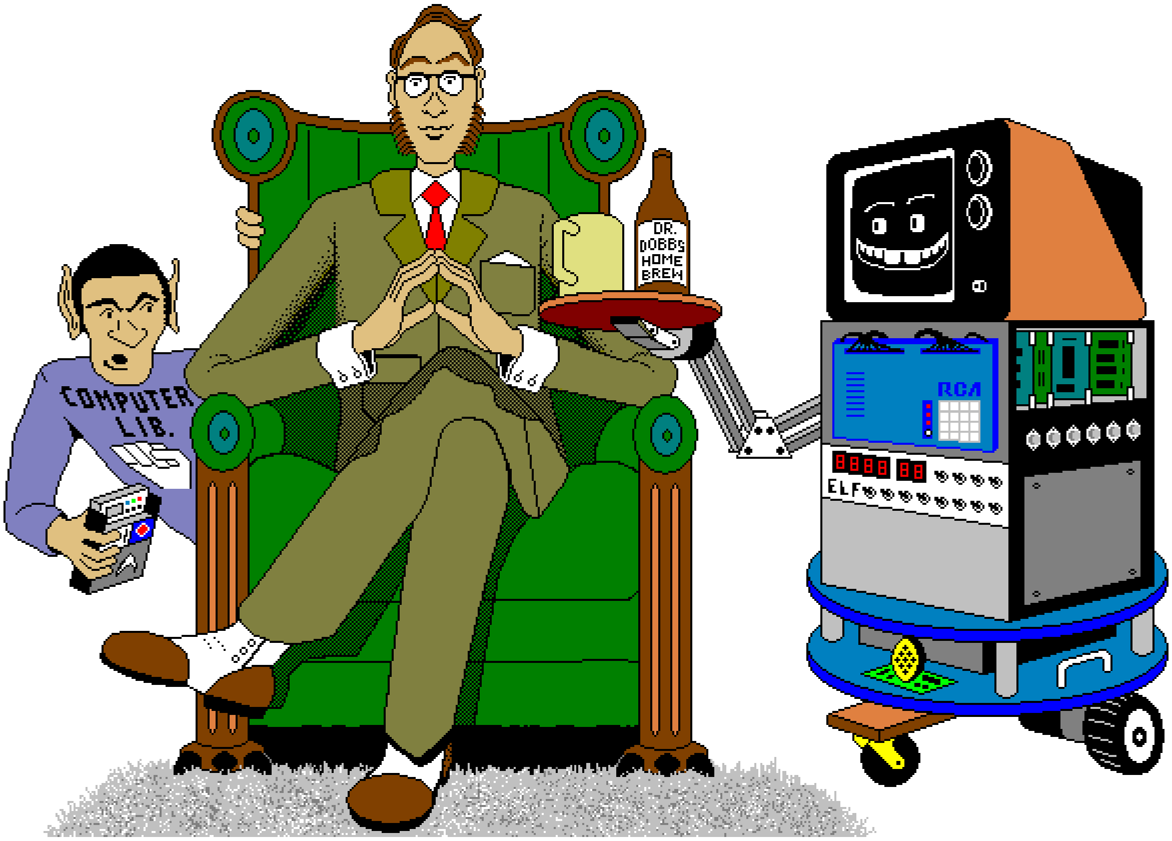
The 1802 Membership Card



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/1802.htm

Rev. K4 -- last revised 18 March 2022

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 at the dawn of the microcomputer revolution, by Joseph Weisbecker <https://en.wikipedia.org/wiki/Joseph_Weisbecker> of RCA Corporation. It used their new CMOS process, for very low power consumption and high noise immunity. It was intended for military and aerospace; applications too tough for other microcomputers to survive.

But Joe was a hacker at heart. His "Build the COSMAC ELF" article in Popular Electronics Aug 1976 described a simple low-cost 1802 computer. 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 -- just a bit slower.

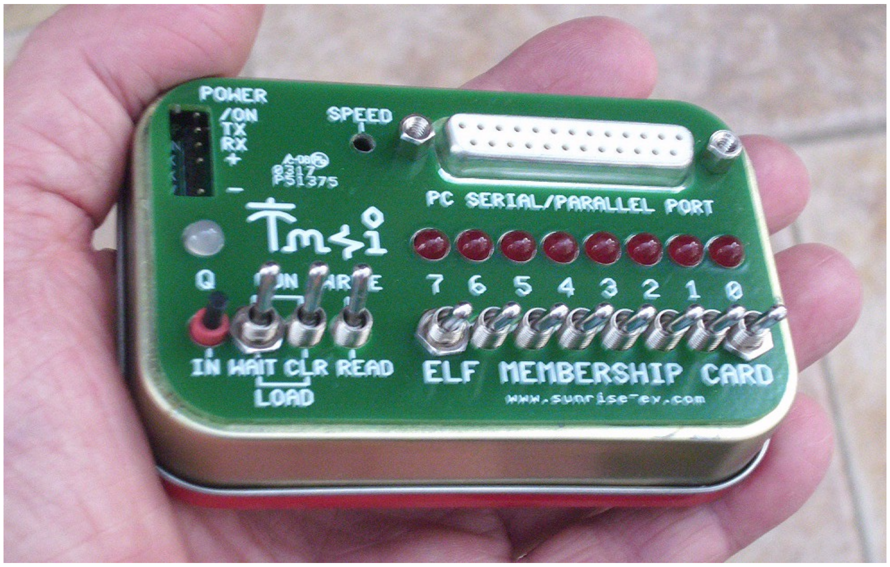
It was the ideal trainer. Hobbyists built thousands of ELFs, learning about computer design, construction, and programming in the process. It set me, and thousands more on a career in engineering. 1802's were designed into all sorts of amazing things; video games, music synthesizers, Chrysler engine computers, military weapon systems, and even spacecraft such as NASA's Galileo and the Space Shuttle. Eat stardust, x86 PCs!

What can you do with an 1802?

The Membership Card is a computer, like the Arduino and Parallax BASIC Stamp. It can run displays, read sensors, make annoying sounds, perform calculations, make decisions, and more. But the 1802 is easier to learn, and free assemblers, BASIC interpreters, and C compilers are available for really ambitious programs.

Unlike modern micros, the Membership Card is totally self-contained. No PC or special software is required. You can power it with a few AA cells or a small solar panel, and program it with nothing but the front panel switches and LEDs. If you need to compute on a desert island, this is the computer to have in your pocket!

So let's return to those thrilling days, when the heroic pioneers of the microcomputer revolution built their own machines from scratch, and programmed them to do incredible things, all for a tiny amount of money!



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

--
This manual was inspired by "Captain Cosmo's Whizbang" http://www.cosmacelf.com/publications/books/captain-cosmos-whizbang.pdf by Jeff Duntemann. Its cover cartoon is by Chris Cloutier, and the one on the back cover by Bill Higgins.

Specifications: What have we got here?

The Membership Card is a miniature version of the original Popular Electronics ELF, repackaged to fit in an Altoids(tm) tin. It's built entirely with vintage parts and techniques available back then (and still available today). It has the basics of every computer; a CPU, memory, and I/O.

CPU:	RCA CDP1802ACE microprocessor (the brains of this outfit).
Clock:	4 MHz ceramic resonator (that's MHz, not GHz).
Memory:	2K to 64K bytes, RAM and/or EPROM (and that's kilobytes, not megabytes). Comes with a 32K RAM at U8. Socket U2 adds another 32K of RAM or ROM. Supercapacitor holds programs and data 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 red LED. Four 1-bit flag inputs, one with a pushbutton switch, one with a green LED. One interrupt input.
Connectors:	6-pin power+serial I/O. A USB-serial adapter can supply both power and serial I/O. 25-pin DB25 connector with all I/O, control, and power signals on it. It can plug into a PC Parallel port or RS-232 serial port to operate the Membership Card directly from a PC.
Size:	3-1/2" x 2-1/8" x 3/4" (89 x 54 x 19 mm).
Power:	Voltage: 3.6v to 5v DC. Current: 5-10ma (depends on voltage and memory size), plus 1-2ma for each LED that is on.
Aroma:	A hint of curiously strong peppermint.

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 give them the thrill of saying, "I did it! It's alive! Bwoo-ha-ha-hah..."

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, or 60/40 is good. Lead-free electronics solder is OK, though it doesn't solder as well. It **must** 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.
- Small Philips screwdriver. For tightening mounting screws.
- A magnifying glass. My old eyeballs aren't good enough to read the markings on some parts, or see if a solder joint is done right or is shorted to the pad next to it. Your eyes may not be that good, either.

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

The Membership Card is your admission into the **COSMAC College of Computer Knowledge**. We'll start with an aptitude test. The next page is the "Parts List". If you bought a bare board, it's your "shopping list". If you bought a kit, it comes with everything in the list. Mark each box (X) as you find and identify it. If any parts are missing, let me know so I can send it out before you get bored and go back to watching TV. Ready? Let's get started!

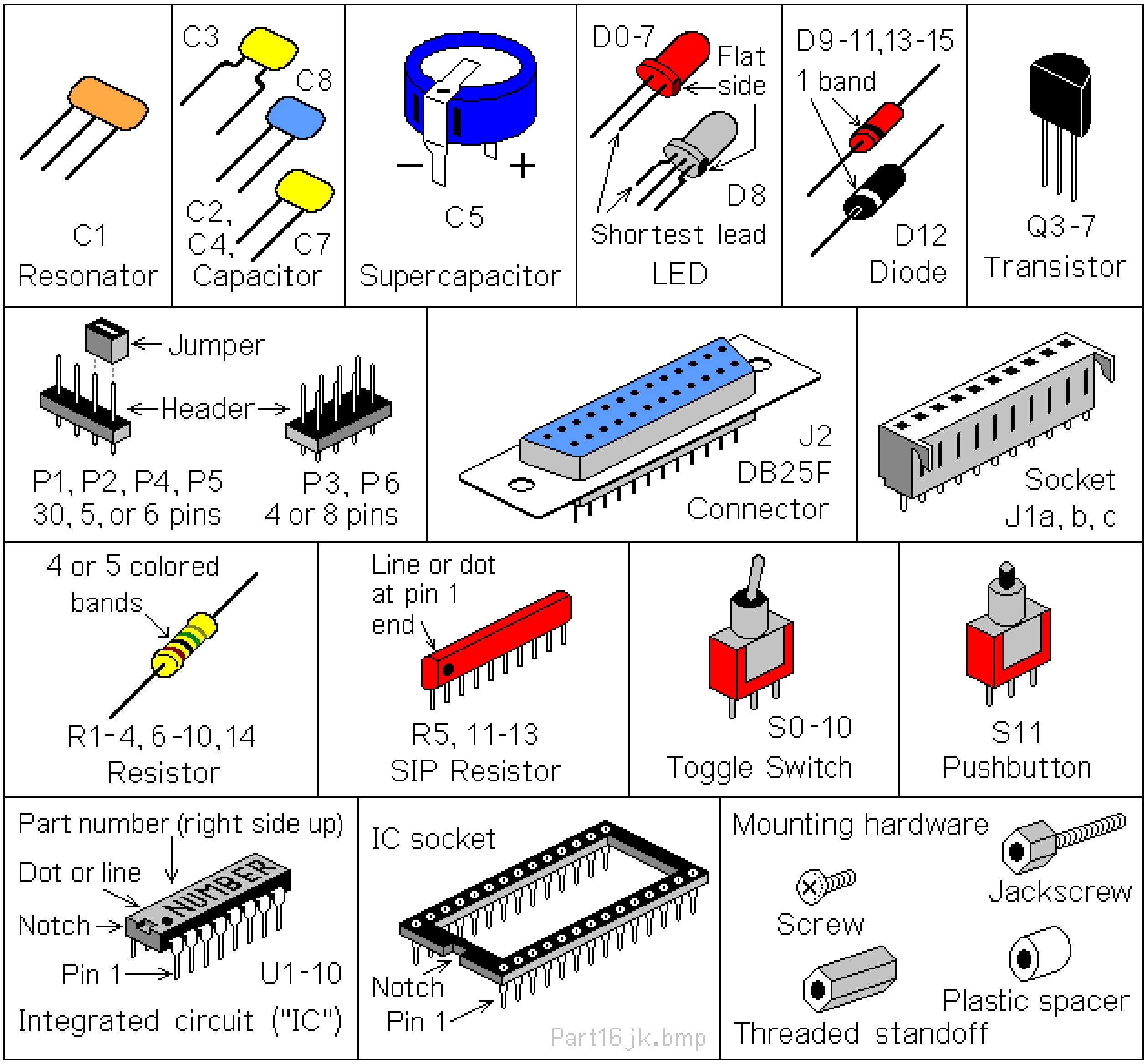
Parts List

Quantity	Identifier	Description	Source for replacement parts
( ) 1	C1	ceramic resonator, 4 MHz (tan, marked Z4.00M)	Mouser 81-CSTS0400MG03
( ) 3	C2, C4, C7	capacitor, 0.1uF X7R ceramic 0.1" wide (yellow, marked 104)	Jameco 1570161
( ) 1	C3	capacitor, 0.1uF X7R ceramic 0.2" wide (yellow, marked 104)	Jameco 544921
( ) 1	C5	supercapacitor, 0.22F, 5.5Vdc (black disk, 0.4" diameter)	Mouser 504-KR-5R5H224-R
( ) 1	C8	capacitor, 4.7uF ceramic (blue, marked 475)	Mouser 810-FK14X7R1E475K
( ) 8	D0-D7	LED, T1-3/4, red	Jameco 253753
( ) 1	D8	LED, T1-3/4, red/green, common cathode (white, 3 leads)	Digikey 754-1886-ND
( ) 5	D9,10,13-15	diode, 1N4148 (smaller reddish glass tube)	Jameco 36038
( ) 1	D11	diode, 1N4734A 5.6v zener (larger reddish glass tube)	Jameco 36118
( ) 1	D12	diode, 1N5818 Schottky (black tube)	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" square pins on 0.1" centers	Jameco 103342
( ) 1	P2	header, 5-pin, with 0.025" square pins on 0.1" centers	Jameco 2076789
( ) 1	P3	header, 4x2 pin, with 0.025" square pins on 0.1" centers	Jameco 109517
( ) 2	P4, P5	header, 6-pin, with 0.025" square pins on 0.1" centers	Jameco 153700
( ) 1	P6	header, 2x2 pin, with 0.025" square pins on 0.1" centers	Jameco 115027
( ) 3	Q3, Q4, Q6	transistor, FJN4303 PNP with base resistors (marked R4303)	Arrow FJN4303RTA
( ) 2	Q5, Q7	transistor, FJN3303 NPN with base resistors (marked R3303)	Digikey 2368-NTE2358-ND
( ) 1	R1	resistor, 10meg 1/4w (brown-black-blue-gold)	Jameco 691817
( ) 1	R2	resistor, 1k 1/4w (brown-black-red-gold)	Jameco 690865
( ) 2	R3, R6	resistor, 100k 1/4w (brown-black-black-orange-brown)	Jameco 691340
( ) 1	R4	resistor, 1 meg 1/4w (brown-black-green-gold)	Jameco 691585
( ) 3	R5,R12,R13	SIP resistor, 8-pin, 7 x 100k (black, marked MEC A 104G)	Digikey 4308R-101-823LF-ND
( ) 3	R8, R9, R10	resistor, 470k 1/4w (yellow-violet-yellow-gold)	Jameco 691500
( ) 1	R11	SIP resistor, 10-pin, 9 x 2.2k (yellow, marked 10A1-222G)	Mouser 652-4610X-1LF-2.2K
( ) 2	R7, R14	resistor, 3.3k 1/4w (orange-orange-red-gold)	Jameco 690988
( ) 11	S0-S10	toggle switch, subminiature SPDT	Digikey CKN1091-ND
( ) 3	nuts	for the toggle switches	(comes with the switches)
( ) 1	S11	pushbutton switch, subminiature SPDT	Digikey CKN1740-ND
( ) 1	U1	1802 microprocessor	Alltronics.com CDP1802ACE
( ) 1	U1a socket	40-pin, ultra-low height, Mill-Max 115-43-640-41-003000	Digikey ED90220-ND
( ) 1	U2a 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	74HC00 quad 2-input NAND gate	Jameco 45161
( ) 1	U5	4013 dual D flip-flop	Jameco 893443
( ) 1	U6	74HC541 or 74HCT541 octal buffer	Jameco 46050
( ) 1	U7	74HC273 or 74HCT273 octal D flip-flop	Jameco 45743
( ) 1	U8	32k RAM, 0.3" wide; CY7C199, CXK58257, HM62256, etc.	Jameco 242376
( ) 1	U9	4071 quad 2-input OR	Jameco 13274
( ) 1	U10	74HC157 or 74HCT157 quad data selector	Jameco 272170
( ) 1	PCB	Membership Card (rev.K4) and Front Panel Card (rev.J or K)	TMSI (that's me :-)
( ) 2	screw	#4-40 x 3/16" round head machine screw	Fastenal 1128626
( ) 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
( ) 8	jumper	jumpers for P2, P3, and P6 headers	Mouser 737-MSB-G
( ) 1	case	Altoids (tm) or equivalent tin candy box	grocery or candy store

Optional:

( ) 1	PCB	Cover Card; a decorative front panel	TMSI (also from me)
( ) 1	U2	Memory upgrade RAM or EPROM with choice of software	TMSI

Did you find them all? Here are some hints:



Comments on Components

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 usually have their value in uF (Micro Farads) or pF (Pico Farads). For example, "104" means 10 with four zeroes after it; that's 100,000pF (picoFarads). For more fun, this may also be written as 0.1uF. 1 uF is a millionth of a Farad, and 1 pF is a millionth of a uF.

ICs have room for a part number, but it's usually obscured with extra letters. For example, the 1802 is marked "CDP1802ACE".

ICs and static electricity: (Old pros and young fools can skip this paragraph.) ICs are easy to damage with 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. 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.

Membership Card Assembly

All parts go on the side of the board with the white lettering (the top, or "component" side). All soldering is done on the other side (the bottom, or "solder" side). The only exception is the Front Panel J1 connector (and I'll remind you when we get to it).

Some parts are POLARIZED; they have to be installed the right way around. Be sure the end with the band or dot or "+" and "–" signs are positioned exactly as described.

Check off the steps as you go (X), in case you get interrupted and have to come back to it later.

( ) R1: 10 megohm resistor (tan body, with brown-black-blue-gold color bands). Bend the wire leads close to the body, and place it on the board at the location marked "R1". Bend the leads outward slightly to hold it in place. Turn the board over, solder each wire, and cut off the excess as short as possible.

Install the rest of the parts the same way.

( ) R4: 1 meg. Tan body, with brown-black-green-gold bands.

( ) D10: 1N4148. Red glass body, marked "4148". Install so the banded end is on the <u>LEFT</u> as shown.

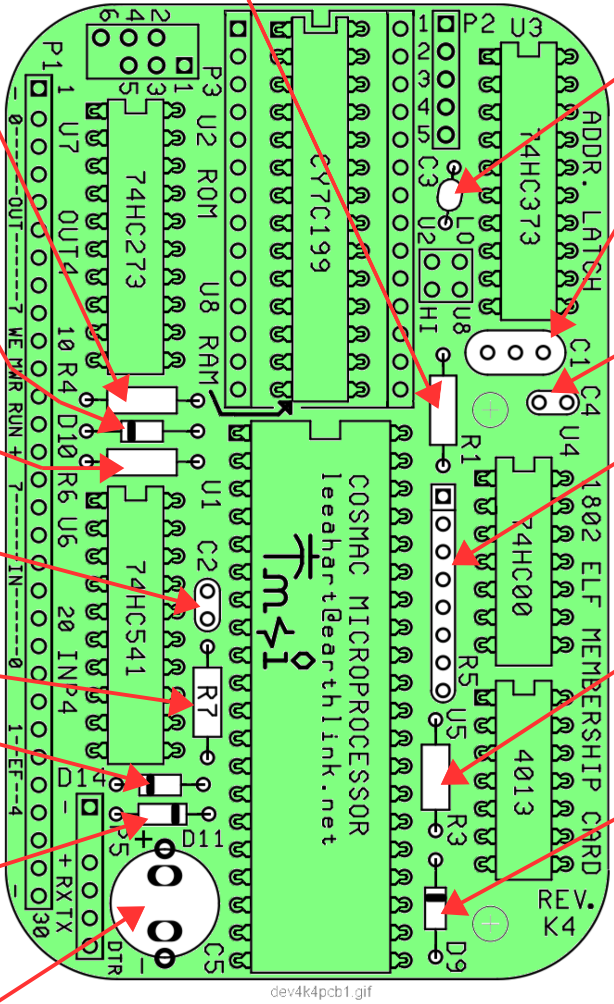
( ) R6: 100k. Brown body, with brown-black-black-orange-brown bands.

( ) C2: 0.1uF. Yellow body marked "104".

( ) R7: 3.3k. Tan body, orange-orange-red-gold.

( ) D14: 1N4148. Red glass body, marked "4148". Banded end on <u>LEFT</u>.

( ) D11: 1N4734A. A larger reddish tube, marked "4734A". Banded end on <u>RIGHT</u>.



( ) C3: 0.1uF. Yellow body 0.2" wide, marked "104"

( ) C1: 4.0 MHz resonator. 3 pins, with tan body, marked "Z4.00M".

( ) C4: 0.1uF. Yellow body marked "104".

( ) R5: 100k x 7. 8-pin SIP, marked "MEC A 104G". The printed side must be on the <u>LEFT</u>, so the end with the dot is on <u>TOP</u>.

( ) R3: 100k. Brown body, with brown-black-black-orange-brown bands.

( ) D9: 1N4148. Red glass body, marked "4148". Banded end on <u>TOP</u>.

( ) C5: 0.22F. A black disk, marked "5.5V" and "0.22F". The lead on top has a "–" sign, and goes in the hole near the "–" and "C5" markings, closest to the corner of the board. Hint: Slip a scrap of paper under C5 when you solder it, so the bottom "+" tab won't short to the "–" pad! Remove the paper when done.

Before you continue...

There are a few decisions you need to make. First, do you want to keep it as small as possible, so it will fit in the Altoids tin? Second, do you want to use IC sockets?

Pin Headers

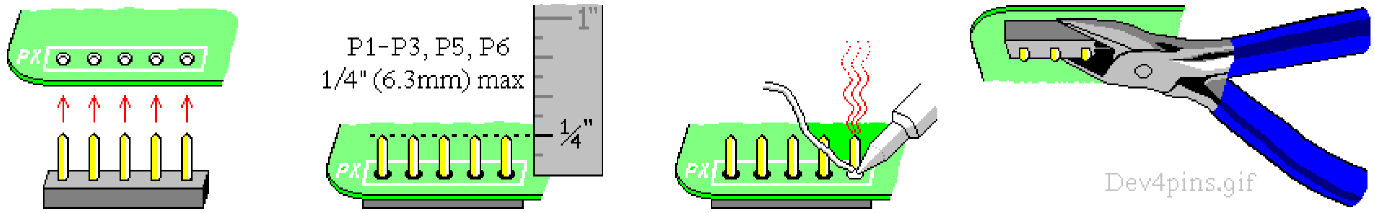
Pin headers P1-P3 and P5-P6 can be installed the EASY way, or the HARD way. Pick ONE method (A or B):

A. **The EASY way:** Do it this way if you do NOT plan to fit both cards into the Altoids tin. Just install the headers as supplied. The black plastic body will sit on the TOP of the board, and you solder the pins from the bottom.

B. **The HARD way:** This way keeps the height as low as possible, so both cards fit inside the Altoids tin. The header pins will be installed WITHOUT their plastic body, so the cards sit 0.1" closer together. See the illustration below, and do the following steps as you install each header P1-P3, and P5-P6:

- Push or tap the **short** end of each pin so it is flush with the plastic body.
- Insert the pins with the plastic body against the **bottom** of the board (see drawing below).
  Be sure the pin are no more than 1/4" (6.2mm) high, or they will short to the Front Panel!
- Solder the pins on the **top** side. Don't use too much solder!
- Cut off the plastic body and pins on the bottom as short as possible.

Hint: To remove excess solder on top, temporarily plug a 10-pin female connector (like J1) on top of the pins to hold them in place. Touch your soldering iron to the pin from underneath, so the excess solder flows down onto the iron. Then unplug the female connector.



Insert from bottom

check height

solder from top

cut off pin and body under board

IC Sockets

IC sockets make troubleshooting and chip replacements easier. But they add cost, and make the board taller so it won't fit in the Altoids tin. They are also the least reliable part of the whole computer, especially if you use el-cheapo sockets!

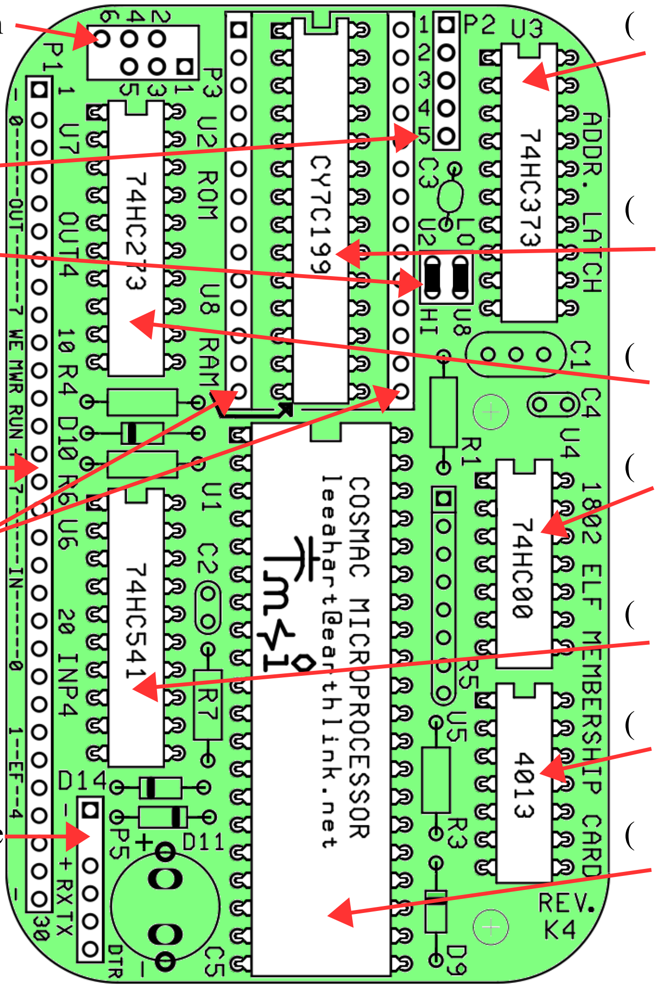
I supply high-quality very-low-height sockets for U1 and U2 (the 1802, and expansion memory chip). You can add sockets for the rest if you watch out for height and quality issues. If you want to socket everything, I suggest Mill-Max 115-43-3##-41-003000 where ## is the number of pins (digikey.com and mouser.com carry them). Or, use individual socket pins for the lowest possible height; Mill-Max #0552-1-15-01-11-27-10-0 (digikey.com ED5037-ND, mouser.com 575-055210). Socket pins are the best option for RAM chip U8, as it is mounted under U2.

Made up your mind? Then let's continue...

Membership Card Assembly (continued)

ICs are polarity sensitive; they **must** be installed with the pins in the correct holes. 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 5 to find pin 1.

ICs usually come with the pins bent outward a little bit. To fix this, stand each IC on its side on the table. Press down, and tip the IC slightly inward to bend the pins 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.



( ) P3: 6-pin header. An 8-pin part is supplied; remove the 2 extra corner pins.

( ) P2: 5-pin header.

( ) P6: 4-pin header.

( ) P6: Shorting jumpers. Plug 2 onto P6 to short U2-HI, and U8-LO.

( ) P1: 30-pin header.

( ) U2a: 28-pin IC socket. Melt the bars with your soldering iron to make two 14-pin rows. File or sand the sides smooth. Solder each half into the holes for U2.

( ) P5: 6-pin header. Remove pin 2 to act as a "key", so you can't plug the mating connector on backwards.

( ) U3: 74HC373 latch IC. (or IC socket if desired). Pin 1 is on TOP, so the lettering faces as shown.

( ) U8: 32k RAM (or socket pins if desired). Pin 1 on TOP.

( ) U7: 74HC273 latch (or socket). Pin 1 on TOP.

( ) U4: 74HC00 gate IC (or socket). Pin 1 on TOP. soldering iron to make

( ) U6: 74HC541 IC (or socket). Pin 1 on TOP.

( ) U5: 4013 flip-flop (or socket). Pin 1 on TOP.

( ) U1a: 40-pin IC socket. Notched end on TOP.

( ) U1: 1802 (marked "CDP1802ACE"). Plug it into the socket, with the notched pin 1 end on TOP. Note: The socket is **tight!** You have to push pretty hard to fully seat U1 in its socket. When it is fully inserted, the skinny part of the pin will go all the way into the socket.

( ) U2 Expansion Memory (Optional): To install a pre-programmed Expansion Memory EPROM at U2, follow the directions that came with it. To install your own chip (RAM or ROM), see **TABLE 1** on the schematic at the end of this manual. For example, to install a 27C256 32k EPROM at U2, install shorting jumpers between P2 pins 2-3 and 4-5, and between P3 pins 1-3 and 4-6.

( ) P6 Memory Map Jumpers: For a standard ELF, short U8-LO (RAM at 0-32k) and U2-HI (U2 at 32-64k). If you install an optional EPROM at U2, its programs should be org 8000h (32-64k). If you install an EPROM with org 0h programs, then short U2-LO (ROM at 0-32k) and U8-HI (RAM at 32-64k).

Front Panel Card Assembly

Install the parts on the Front Panel card the same way you did on the Membership Card. Remember, the ICs, diodes, and transistors are polarity sensitive; they **must** be installed with their leads in the correct holes.

( ) U10: 74HC157 multiplexer. Pin 1 on TOP, so the lettering faces as shown.

( ) Q7: FJN3303 NPN transistor, marked "R3303". Install it with the flat side on the LEFT, and its 3 leads in their respective holes. Wiggle it down so it is no more than 0.3" (7.5mm) high. Solder the leads, and cut off the excess.

( ) Q6: **FJN4303 PNP** transistor, marked "**R4303**". Install it like Q7, but with its flat side on the RIGHT. Note: Q6 is a different part number than Q7!

( ) Q4: **FJN4303 PNP** transistor, marked "**R4303**". Flat side on the RIGHT.

( ) Q3: **FJN4303 PNP** transistor (marked "**R4303**"). Flat side faces DOWN.

( ) R11: 2.2Kx9 10-pin SIP. Yellow, "10A1-222G". Printed side on LEFT.

( ) R2: 1k. Tan body, brown-black-red-gold.

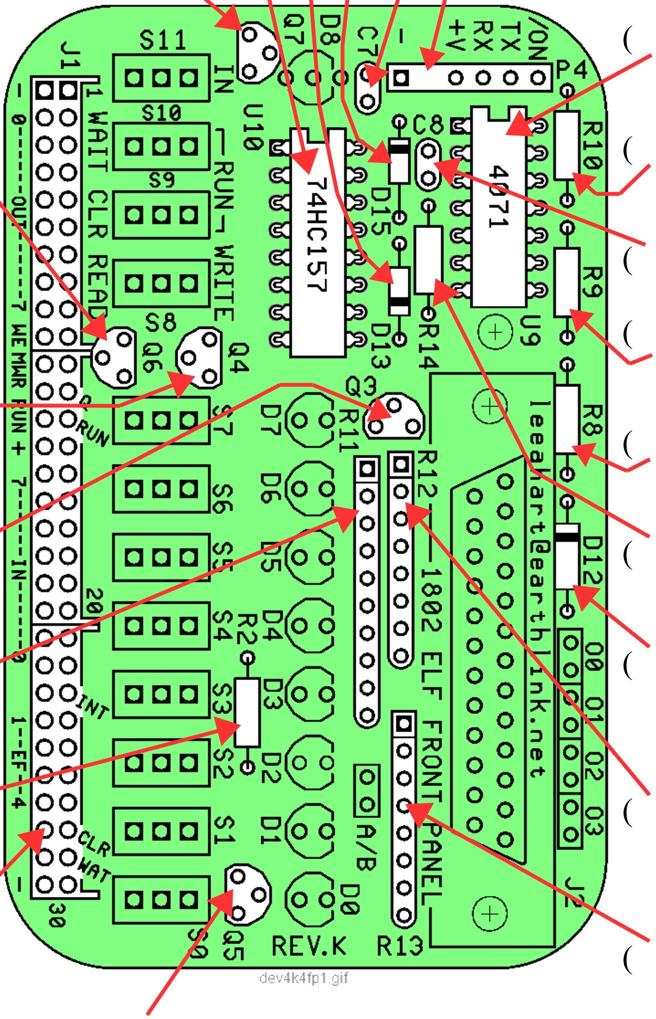
( ) J1a,b,c: 30-pin socket. (3 10-pin parts supplied). Mount on the BOTTOM with the "ears" toward the center, and their pins in the INNER row of holes. Solder on the top, and cut off the "ears".

( ) D13: 1N4148, marked "4148". Band on BOTTOM.

( ) D15: 1N4148, marked "4148". Band on TOP.

( ) C7: 0.1uF. Yellow, marked "104".

( ) P4: 6-pin header. Remove pin 2 as a key. Install it with its plastic body.



( ) U9: 4071 quad OR gate. Pin 1 on TOP.

( ) R10: 470k. Tan, yellow-violet-yellow-gold bands

( ) C8: 4.7uF. Blue, "475".

( ) R9: 470k. Tan, yellow-violet-yellow-gold bands

( ) R8: 470k. Tan, yellow-violet-yellow-gold bands

( ) R14: 3.3k. Tan, orange-orange-red-gold bands.

( ) D12: Schottky diode. Black,marked"1N5818". Banded end on TOP.

( ) R12: 100Kx7 8-pin SIP. Black, "MEC A 104G". Printed side on LEFT.

( ) R13: 100Kx7 8-pin SIP. Black, "MEC A 104G". Printed side on LEFT.

( ) Q5: FJN3303 NPN transistor, marked "R3303". The flat side faces LEFT.

Front Panel Card Assembly (continued)

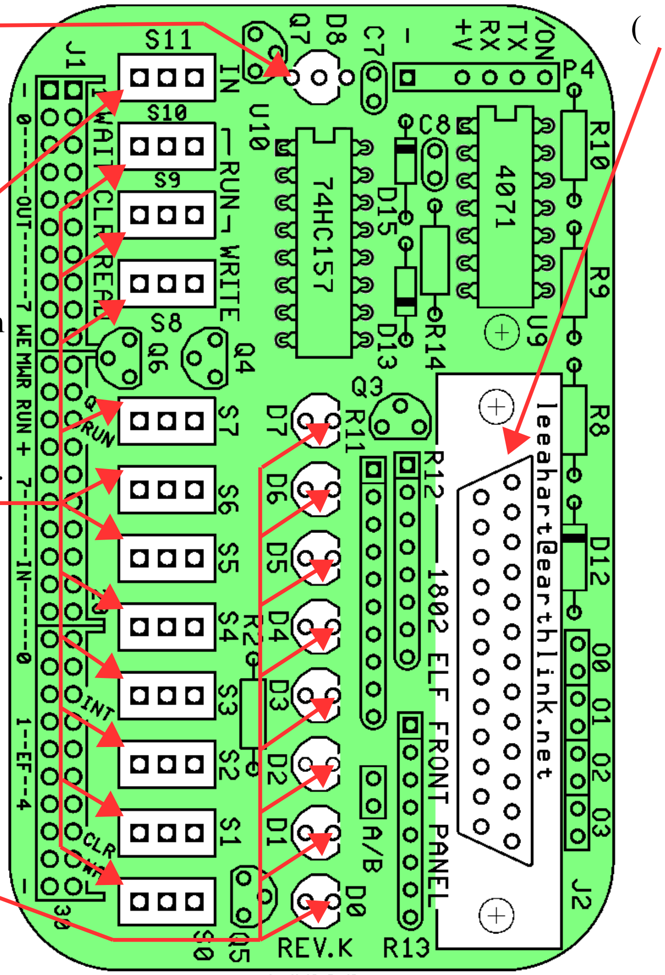
DO NOT SOLDER the following parts yet. We'll use the Cover Card to position them, and solder them later.

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

( ) S11: Pushbutton switch. Place it on the board with its printed side UP. Bend a pin so it won't fall off.

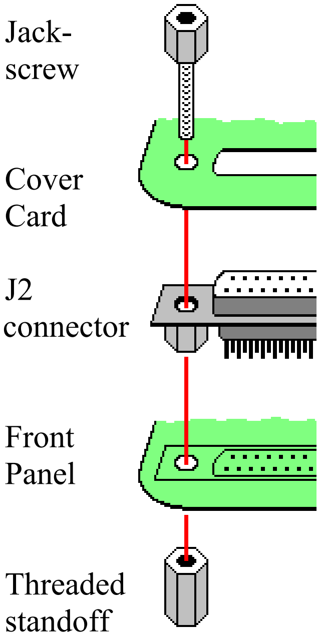
( ) S0-S10: Toggle switches. Place them on the board with the side marked “1A 120V" on TOP. Bend a pin on each one to hold it in place.

( ) D0-D7: Red LEDs. Place the short lead in the right hole, so the flat side is on the right. Bend a lead to hold them on the board.



dev4k4fp2.gif

( ) J2: 25-pin D-connector. Mount it loosely to the Front Panel and Cover Card (see Note 1) with #4 hardware as shown:



( ) Wiggle the LEDs and switches so they fit neatly into the holes in the Cover Card (or your front panel). Work the board as close as you can. Careful! The switches can break if the holes in your front panel don't match well enough. Hint: A 1/8" (3mm) strip of cardboard can space the LEDs up a bit for a better appearance. **NOW** solder the parts to the board, with the Cover Card holding everything in position.

( ) Remove the mounting hardware from J2. Be sure you can easily remove and re-install the Cover Card. If necessary, adjust parts or enlarge holes in your front panel so it fits 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! :-)

( ) Jumper options A/B, OØ, O1, O2, O3:
To connect J2 to a classic PC parallel port: Short A/B, and leave OØ, O1, O2, and O3 open.
To use J2 as a general-purpose I/O port: Leave A/B open, and short OØ, O1, O2, and O3.

Note 1: The optional Cover Card is a decorative panel with all the labels and holes made for you (see photo on page 2). It is available at http://www.sunrise-ev.com/1802.htm and makes assembly easy; just remove the cover from the Altoids tin, and drop the finished unit into it. Or, use a nibbling tool to remove the bottom of the tin, and solder the Cover Card in its place (**after** you finish assembly, so you know everything fits).

If you don't have a Cover Card, you can make your own panel (using the Front Panel as a template). Or, just leave it “naked”, with no enclosure or cover.

Final Assembly

( ) Re-install the Cover Card with #4 mounting hardware as shown on page 10. Plug the Membership Card and Front Panel cards together. Look between them to be sure that **NOTHING** touches between the two boards except the 30-pin connector P1-J1, 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. Note: If you left the plastic bodies on the headers, add #4 nuts under the hex standoffs for the extra height.

( ) Put nuts on a few of the switches (like the ones at each end). There isn't room for nuts on all of them, and they aren't all needed anyway. A few nuts and the hardware for J2 will do 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, trim the corners, 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; 3.6-5Vdc at a few ma, plus 1-2 ma for each LED that is lit. A battery holder with three 1.5v AA cells in a second Altoids tin works nicely. Or you can use a USB cable to a computer, an old cellphone charger that outputs 5Vdc, a single 3.6v lithium cell, or even a little solar panel.

Connecting Power

P4 and P5 are power connectors; P4 is on the Front Panel, and P5 on the CPU card. They are standard 6-pin headers, with 0.025" square pins on 0.100" centers (with pin 2 removed for keying). There are many mating connectors. Example: Molex KK series: 22-01-2061 housing, and 08-55-0102 terminals (www.digikey.com WM1579-ND and WM2312-ND respectively). It is common, inexpensive, and easy to assemble.

P4 pin	Name	Function
1	–	VSS. Ground, common; the power supply negative.
2		key (remove the pin, and plug the hole in your mating connector).
3	+	VDD. Power supply positive; +3.6 to +5 volts DC (battery, USB supply, etc.)
4	RX	Serial TTL data input (connects to PC TX output).
5	TX	Serial TTL data output (connects to PC RX input).
6	/ON	Off/On control. Ground or low for ON; open or high for OFF (standby).

If you're the impatient type, connect your positive supply to pin 3, and ground to pins 1 and 6. Plug/unplug it for on/off. Supercapacitor C5 will hold memory for hours even when off. Then go to the **Operation** section.

If your power supply is a battery, it can maintain memory indefinitely. Wire an SPST switch between P4 pins 1-6. Off is "Standby" (minimum power consumption); memory is retained, the clock stops, the Front Panel LEDs are disabled, and the 1802 is reset. On is "Run"; the clock and LEDs are enabled, and the 1802 runs programs. (Pin 6 combines the function of the RUN and LED pins on rev.H2 and earlier Membership Cards.)

For a more permanent setup, get the Sparkfun #9716 FTDI 5V USB-serial Basic adapter (a naked PCB) or #9718 FTDI 5V USB-serial Cable (packaged, with cable) at www.sparkfun.com. Block pin 2 (CTS, the brown wire) with a toothpick or something as a key. Then plug it into P4 (black to –, green to /ON) to provide power and serial I/O from a USB port. Note: You still have to ground pin 6 to turn the 1802MC "on" (either with a piece of wire, or a switch, or by telling your Terminal program to go "on line" or "connect").

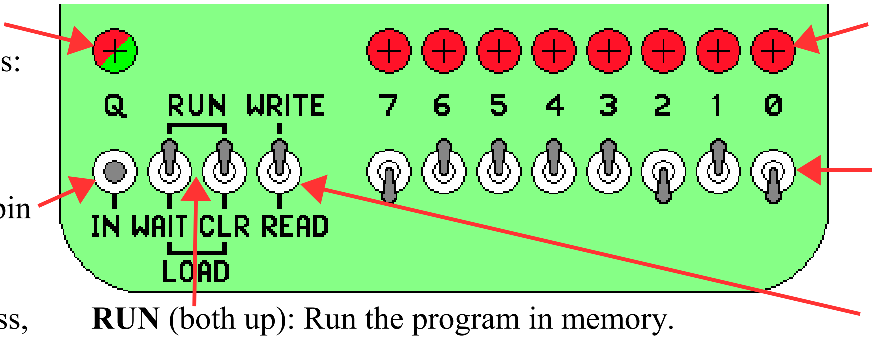
Notes: - Remember, /ON (P4 pin 6) needs to be grounded (P4 pin 1) to enable the 1802, clock, and LEDs.
- The Front Panel switches over-ride "Standby" unless you set the WAIT switch up.
- Standby power at 4v is under 50uA with the chips supplied; so three AA cells will last for years.
- If you install any fast modern EPROMs or RAM upgrades, the standby power will be higher.

Zener diode D11 is an "idiot" diode. If power is over 5.6v, or connected backwards, or is AC instead of DC, 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. (Note: D11 actually begins to conduct above 4.5v, which increases supply current.)

DB25 connector J2 pin 18 can also be used for the + power input. It has a diode in series (D12), so applying AC or reversed polarity DC to it won't hurt anything. You can also connect a backup battery between J2 pin 18 (+) and pin 19 (GND), and it will “take over” when power is removed from P4.

Operation with Front Panel

OK; so you connected power (and nothing smoked). Let's see if it works! The Front Panel controls the 1802's operation, and shows its status.



Q LED shows status of the Q and EF3 pins:
RED if Q=1
GREEN if EF3=1

IN button sets EF4 pin low (in RUN mode), and in LOAD mode:
- write Data to address,
- read Data to verify,
- advance to next address.

Data LEDs show output data bits: RED=1, off=0.

Data switches set input data bits. Up=1, Down=0.

Read/Write switch
Up=Write enable
Down=Read only

RUN (both up): Run the program in memory.
WAIT (Wait down): Pause program right where it is.
CLEAR (CLR down): Reset Q, X, P, and R0 all to 0.
LOAD (both down): Read/write Data when IN is pressed.

Let's load a program and see if it works! For each step, set the switches as shown. "1" means the switch is up. "0" means the switch is down. "X" means push and release the IN button. "." means the switch position doesn't matter. I'll show the switch positions like this:

IN	WAIT	CLR	READ	7	6	5	4	3	2	1	0	Mode	Description
.	1	0	.	.	.	.	.	.	.	.	.	CLEAR	Resets the 1802.

Program 1 -- Blink Q Fast

This is a simple program to blink the "Q" LED:

	IN	WAIT	CLR	READ	7	6	5	4	3	2	1	0	Mode	Description
2.	.	1	0	.	.	.	.	.	.	.	.	.	CLEAR	Reset the 1802 (sets R0 to address 0000).
3.	.	1	0	1	.	.	.	.	.	.	.	.	WRITE	Set WRITE up, so we can write to memory.
4.	.	0	0	1	.	.	.	.	.	.	.	.	LOAD	Set WAIT down (for LOAD mode).
	X	0	0	1	0	1	1	1	1	0	1	1		a. Set Data to "0111 1011", then press IN. 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.
					___7___/				___B___/					
	X	0	0	1	0	1	1	1	1	0	1	0		b. Set Data 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.
					___7___/				___A___/					

```
     X 0 0 1      0 0 1 1 0 0 0 0       c. Set Data to "0011 0000", then press IN. This loads
                  \___3__/  \___0__/       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       d. Set Data to "0000 0000", then press IN. Loads hex 00
                  \___0__/  \___0__/       (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.

```
IN WAIT CLR READ  7 6 5 4 3 2 1 0  Mode    Description
5.   .  1  0 .    . . . . . . . .  CLEAR   Set WAIT up to reset the 1802 (set R0 to address 0000).
6.   .  1  0 0    . . . . . . . .  READ    Set READ down, so we can read memory.
7.   .  0  0 0    . . . . . . . .  LOAD    Set WAIT down, for LOAD mode.
     X 0 0 0      0 1 1 1 1 0 1 1       a. Press the IN button. This reads memory address 0000,
                  \___7__/  \___B__/       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       b. Press IN again. Displays "0111 1010" = hex 7A from
                  \___7__/  \___A__/       address 0001, then advances R0 from 0001 to 0002.
     X 0 0 0      0 0 1 1 0 0 0 0       c. Press IN again. Displays "0011 0000" = hex 30 from
                  \___3__/  \___0__/       address 0002, then advances R0 to 0003.
     X 0 0 0      0 0 0 0 0 0 0 0       d. Press IN again. Displays "0000 0000" = hex 00 from
                  \___0__/  \___0__/       address 0003, then advances to 0004.
```

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

```
IN WAIT CLR READ  7 6 5 4 3 2 1 0  Mode    Description
8.   .  1  0 .    . . . . . . . .  CLEAR   Reset the 1802 (sets R0 back to address 0000).
9.   .  1  1 .    . . . . . . . .  RUN     Set WAIT up, for RUN mode. The 1802 begins running
                                           the program starting at address 0000.
```

This program is very simple; it tells the 1802 to turn the red "Q" LED on, then off, and repeat forever. But it's doing it too fast to see the individual blinks (like 83 KHz)! To prove that it's really going on and off, 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 the WAIT switch up and down will RUN and WAIT the program, sometimes catching Q set (red), sometimes reset (off). 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.

Program 2 -- Blink Q Slow

Blink the Q LED slowly.

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 hex-binary "cheat sheet" at the end of this manual).

address (R0)	machine code Hex	Binary	Mnemonic	Human readable comments
0000	F8	1111 1000	LDI	LoaD Immediately...
0001	A2	1010 0010	162	...162 decimal, which is A2 hex (check it for yourself!)
0002	B2	1011 0010	PHI R2	Put it in the HI half of register 2

0003	22	0010 0010	DEC R2	DECrement (i.e. subtract 1 from) register 2
0004	92	1001 0010	GHI R2	Get the HI half of register 2
0005	3A	0011 1010	BNZ	Branch if Not Zero...
0006	03	0000 0011	3	...to address 3 (loops (162-1) x 256 = 41216 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 at 0000 to 0002 set up a 16-bit counter in register 2, and sets it to hex A2xx. 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 which 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. The A2h in location 0001 was chosen to blink Q exactly once per second at 4 MHz (the speed with the resonator supplied).

Change the value at 0001 to control the speed. How fast can it go? Set it to 01. Connect a pair of headphones or a small speaker between J2 pin 15 (TXD, controlled by Q) and pin 19 (ground). You'll hear an audio tone!

Program 3 -- Read Switches and Display Value in LEDs

This program is a bit more complex. It reads the 8 data switches, displays their settings on the 8 LEDs, and pulses Q at a rate set by the switches. It tests the 1802's ability to read the switches and write to the lights. If you connect a speaker as described above, you'll hear a tone whose frequency is set by the switches. See if you can enter it in hex, without the binary values. Note that some instructions have TWO bytes on the same line; they go in consecutive memory addresses. Set S8 to WRITE (up) to run this program.

Address	Hex opcode	Mnemonic	Comments
0000	E1	SEX 1	Set X register to 1 (OMG! The 1802 has sex!)
0001	90	GHI 0	Get HI byte of register 0 in D (which is 0; so this sets D=0)
0002	B1	PHI 1	Put D in HI byte of register 1 (so R1=00xx)
0003	F8 13	LDI 13h	Load D Immediately with 13 hex (TWO bytes in this instruction)
0005	A1	PLO 1	Put D in the LOw half of R1 (so R1=0010)
0006	6C	INP 4	INPut port 4 (front panel switches) & write to D and memory at (R1)
0007	64	OUT 4	OUTput to port 4 (front panel LEDs) contents of memory at (R1)
0008	CD	LSQ	Long Skip over next 2 instructions if Q=1...
0009	7B	SEQ	...if Q was 0, then Set Q=1
000A	38	SKP	...and SKIP next instruction
000B	7A	REQ	else Q was 1, so Reset Q=0
			(At this point, D = the value read from the switches)
000C	FF 01	SMI 1	Subtract Memory Immediately from D (this means D=D-1)
000E	3A 0C	BNZ 0Ch	Branch if Not Zero to address 000C (so this loops "switch" times)
0010	7A	REQ	Reset Q
0011	30 00	BR 0	BRanch unconditionally back to address 0000
0013	xx		(address used by INP and OUT for the switch and LED values)

More Programming

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

Parallel I/O

If your PC has a parallel port, all you need is a standard DB25 male-to-male cable. Download the program at http://www.sunrise-ev.com/MembershipCard/ELF-LINK.exe. Set S8 to READ (down), all other switches UP, and run ELF-LINK. This is a Microsoft QuickBASIC program to control the Membership Card switches and LEDs to load, save, and run programs. If you don't like BASIC, the .BAS source is in plain ASCII so you can re-write it in your favorite programming language. Now, who will be the first to translate it into C?

Serial I/O

Serial I/O uses the 1802's EF3 and Q pins. TTL levels are +3.3v to +5v idle, and 0v to +0.5v active. It is available on P4 and P6, whose pinouts match the Sparkfun #9718 USB-serial cable, which provides both serial I/O and power (see https://www.sparkfun.com/products/9718).

RS-232 serial is on J2 (the DB25 connector) on the Front Panel card. RS-232 levels are -5v to -12v idle, and +5v to +12v active (i.e. standard RS-232 signals are inverted compared to TTL).

D8 is a 2-color LED. It is red when the Q output pin is high (active), green when the EF3 input pin is low (active), and yellow if both are active at once. Since Q and EF3 are used for serial I/O, D8 is red when sending data, and green when receiving data. If Q is low and no serial input is connected, D8 will be off.

Serial I/O needs a program. You can "toggle in" a simple serial loader. Or, add an EPROM with a serial I/O program in U2. Then you can communicate with a "terminal" program such as Hyperterm, RealTerm, or TeraTerm with a PC's RS-232 port, or USB port with a USB-to-serial adapter. Here are some examples:

- Herb Johnson's IDIOT monitor http://www.retrotechnology.com/memship/idiot_ramrom.html
- Chuck Yakym's monitor 2.0+BASIC ROM http://www.sunrise-ev.com/MembershipCard/MCSMP20J.bin
- Spare Time Gizmo's Elf2K ROM http://www.sparetimegizmos.com/Downloads/v88.hex
- Mike Riley's Diskless Elf/OS ROM http://www.sunrise-ev.com/MembershipCard/disklessElfOS.hex

Note: Front Panels before rev.J did not invert Q. Front Panels from rev.J to present invert Q. Your software needs to use the right level to communicate. http://www.retrotechnology.com/memship/elf2k_mship.html

Serial I/O can be tricky to set up. See http://www.retrotechnology.com/memship/mem_rom_serial.html for help. Also see http://www.sunrise-ev.com/1802.htm#projects for other ways to load programs.

Operation without a Front Panel (CPU card only)

You can use the 1802MC CPU card all by itself by installing an optional programmed EPROM memory IC at U2. It must be an "org 0h" program, so it runs automatically at power-on or reset. I'll use MS20ANSA as an example (see http://www.sunrise-ev.com/1802.htm for other programs and ordering information).

( ) Install your 32k EPROM at U2. (Naturally, pin 1 and the notch must match the outline on the board.)

( ) Install shorting jumpers at P2 between pins 2-3 and 4-5, and between P3 pins 1-3 and 4-6. These configure U2 for a 27256 or 27C256 EPROM. (If you're installing a different part, see TABLE 1 on the schematic on page 20).

( ) Install shorting jumpers at P6 between U2-LO and U8-HI. This addresses the EPROM at org 0h, and the RAM at 8000h.

( ) Install a shorting jumper between RUN and VDD (P1 pins 13-14). When RUN is connected to VDD, the 1802 runs. When you open RUN, it goes into Standby; power consumption falls to microamps and the 1802 clock stops; but data is retained in the registers and memory. (You can use an SPST switch to connect RUN and VDD to provide an ON-Standby function.)

( ) Install a shorting jumper between /WE and /MWR (P1 pins 10-11). This enables programs to write to RAM. (Opening this jumper write-protects memory.)

( ) Connect your USB-serial adapter to your PC (see "Serial I/O" on page 15). Don't connect the 1802MC yet; let's get the serial adapter working first. If it's the Sparkfun cable, it's easy. If it's something else, you may have to experiment.

( ) Start your Terminal program (HyperTerm, TeraTerm, RealTerm, etc.). Configure it as follows:

– 4800 baud, 1 Start, 8 data, no parity, 1 Stop, Full Duplex
– No hardware or software handshaking.
– Set the Pacing or Transmit Delay to 10 msec/char and 250 msec/line.
– Set the ENTER key to send only an ASCII <CR>.

( ) Short your adapter's TX to RX. When you type on your keyboard, it should appear on the screen! If so, you were successful. If not, keep "fiddling" until it works.

( ) Now you can connect your USB adapter to P5 on the 1802MC. Tell your Terminal program to go "on-line" or "connect". (If your USB adapter has the DTR or RTS signal on P5 pin 6, then going "on-line" or "connecting" will reset the 1802 and start its program running).

( ) The first key you type must be ENTER (to send the ASCII <CR> code). The MCSMP20 program will figure out the baud rate, and display its sign-on message. If so, you're in business!

See the online MCSMP20 and BASIC3 manuals at http://www.sunrise-ev.com/1802.htm to use the monitor and BASIC. Now, have fun! :-)

In Case of Difficulty...

The most common problems are poor soldering, and parts located in the wrong places. Look for bad solder joints: A pin that's not soldered, or one with too much solder so it shorts to another pin.

Look for something on the Membership Card that is shorting to the back of the Front Panel. Good candidates are pin headers P2 and P3. You should be able to freely slide a piece of paper between the two boards.

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

Power: Check for +3.6 to +5 Vdc between GND (P1 pin 1 or 30) and VDD (P1 pin 14).

RUN: Check to see that RUN (P1 pin 13) is high to enable the oscillator. If the oscillator is running, the DC voltage on U1 pin 39 should be about 1/2 the supply voltage (it's actually a 4 MHz sine/triangle wave).

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 the 1802 /EF4 input. It should be high, and go low when you push the IN button. If it's the opposite, you have pushbutton S11 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 sets 1802 /DMA-IN 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.

The two boards can be checked separately. The Front Panel is just a set of independent switch outputs and LED inputs. To test it, unplug the Membership Card. Connect +3.6 to +5 Vdc to VDD (P4 pin 3), negative to GND (P4 pin1). Ground /ON (P4 pin 6). Now use a piece of wire in J1 to jumper any output to any input. Operate the corresponding switch and confirm it's working with the LED. For example, jumper Q (J1 pin 12) to /EF4 (J1 pin 27); when the IN button is up, the red Q LED is on; when IN is down, the Q LED is off.

To check the CPU board by itself, connect +power to VDD (P1 pin 14), and GND (P1 pin 1 or 30). Jumper P1 pin 13 to pin 14 (RUN). The 1802 will "run" the program in U2. If there is no program (because you haven't loaded one), remove U2 and use eight resistors or two SIP networks to pull all the data bus pins in socket U2 high (pins 11-13 and 15-19). This is FFh, the "Subtract immediate" instruction. The 1802 will march through memory, incrementing the address bus and doing a read cycle (/MRD low) at each address. Other instructions (like the NOP instruction C4h) can be forced the same way. Use an oscilloscope to check the address, data, and control signals to be sure they are all working, and producing digital logic level (a short between two traces will generally cause voltage that is halfway between VDD and GND as the two outputs "fight" each other).

If you don't have an oscilloscope, the 1802 will run at "zero" clock speed. You can jumper /EF4 to 1802 pin 1, and "clock" the 1802 with the IN button. Now you can single-step through each bus cycle to see what the 1802 is doing. The RCA 1802 datasheet or MPM-201B manual (link below) describe what it should happen.

Still doesn't work? Email me for help. Failing that, send it to me at Lee Hart, 814 8th Ave N, Sartell MN 56377 USA and I'll try to fix it! :-)

Links for more information about the 1802 and ELF computers:

http://www.sunrise-ev.com/1802.htm

My website, with ordering information, manual updates, schematics, cheat sheets, and more.

https://billr.incolor.com/elf/html/elf-1-33.htm

The Aug 1976 Popular Electronics article that introduced the Elf. Most of it applies directly to the 1802MC.

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 Membership Card "home" page, with tons of design notes, history, software, and info.

Last Writes

I'm still improving the manual, so please contact me with corrections and improvements. Also watch my web pages for updates! Here's a list of changes along the way:

Jul 2010 Rev.A: R8-10 was 100k, now 470k.

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

Nov 2010 R4,8-10 was 470k now 499k. C6 was 82pF, now 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, now 3-4. For 2764; P2 was 3-4 now 2-3. For 27128 and 27256; P2 was 1-2 and 3-4, now 2-3 and 4-5.

Jun 2012 Rev.D: R2 was 6.8k, now 5.6k. Added D12 so power is available on J2. Added R14 (so J2 pin 1 is serial input). Added jumpers A/B, O0, O1, O2, and O2 (to provide 9 instead of 5 output bits on J2).

Aug 2012 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, and changed it to 0.10F.

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: Q1 changed 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. Add P6 to Front Panel.

May 2015 Rev.H2 CPU: Replaced Q1 with Q1+Q2 to make a proper inverter for fastest /A15 rise time.

Jun 2016 Rev.H2 Front Panel: Added R2 to improve RS-232 voltage levels.

Aug 2016 Upgraded C5 from 0.1F to 0.22F. Added jumper options for EEPROMs to schematic.

Feb 2017 Rev.I: Added TTL serial I/O to power connector P4. Combined RUN and LED into /ON signal. Added Q6 and R15. Moved D15 to be across C8. Deleted the confusing serial jumper options.

Jun 2017 Rev.J: Delete R15. Add Q7 to invert serial output, so red Q LED is off (instead of on) when idle.

Aug 2017 R5, R12, R13 were 100k, now 82k (I ran out of 100k). R11 was 1k, now 2.2k (saves power).

May 2018 Rev.K: U4 was 4093, now 74HC00 (faster). Delete C6, Q1, Q2, R7. R1 was trimpot, now 10meg.

July 2018 I ran out of 1 meg resistors, so R4 is changed to 1.5 meg.

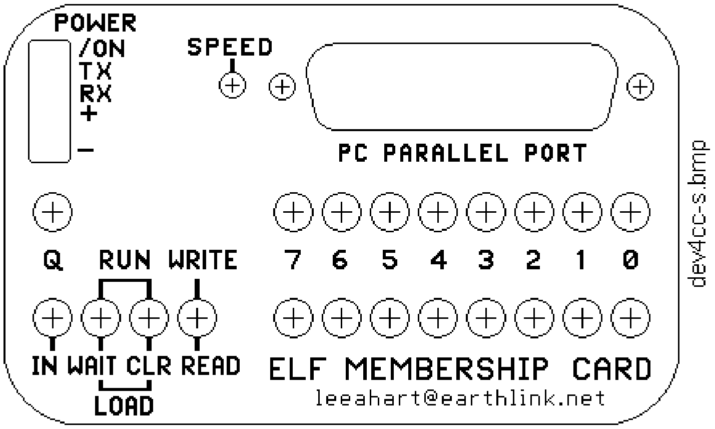
Oct 2018 Changed R4 back to 1 meg. R5, R12, R13 were 82k, now 100k.

Apr 2019 Rev.K2: U6 was 74HC244, now 74HC541. R8-10 was 499k, now 470k. D11 was 1N5231 (5.1v), now 1N4734 (5.6v) to reduce supply current at 5Vdc. C5 was 0.22F, now 0.33F (same $, more Farads).

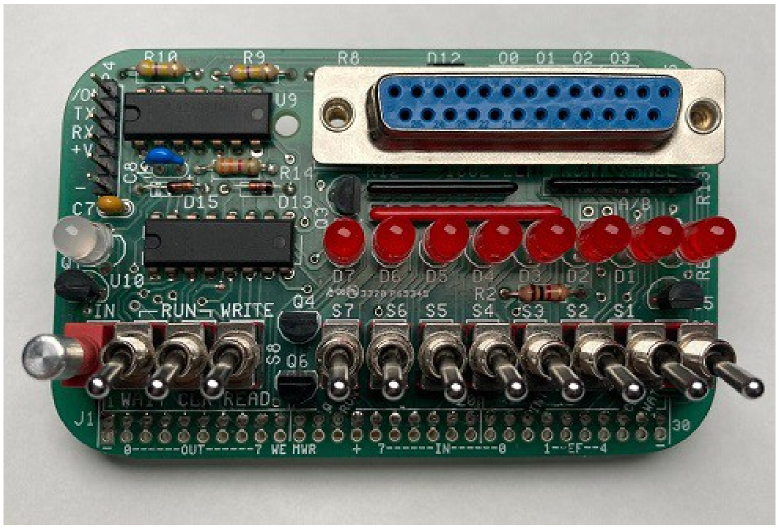
Sep 2020 Rev.K3: U7 was 74HC374, now 74HC273. C5 was 0.33F, now 0.22F.

Jun 2021 Rev.K4: Added P5 and R7 (to use CPU card without Front Panel). Added P6 (for address jumpers). Replaced 0.6” wide RAM at U2 with 0.3” wide RAM at U8 (makes adding memory upgrade easier).

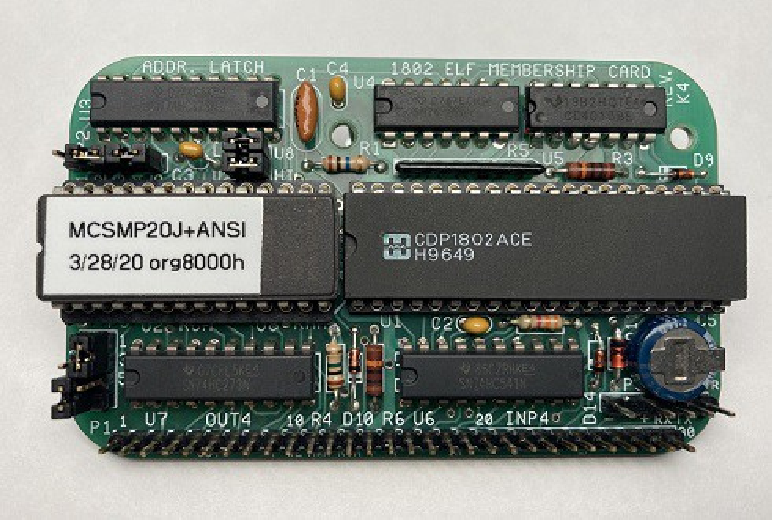
"Mugshots" of the Elf Membership Card 3-board Set



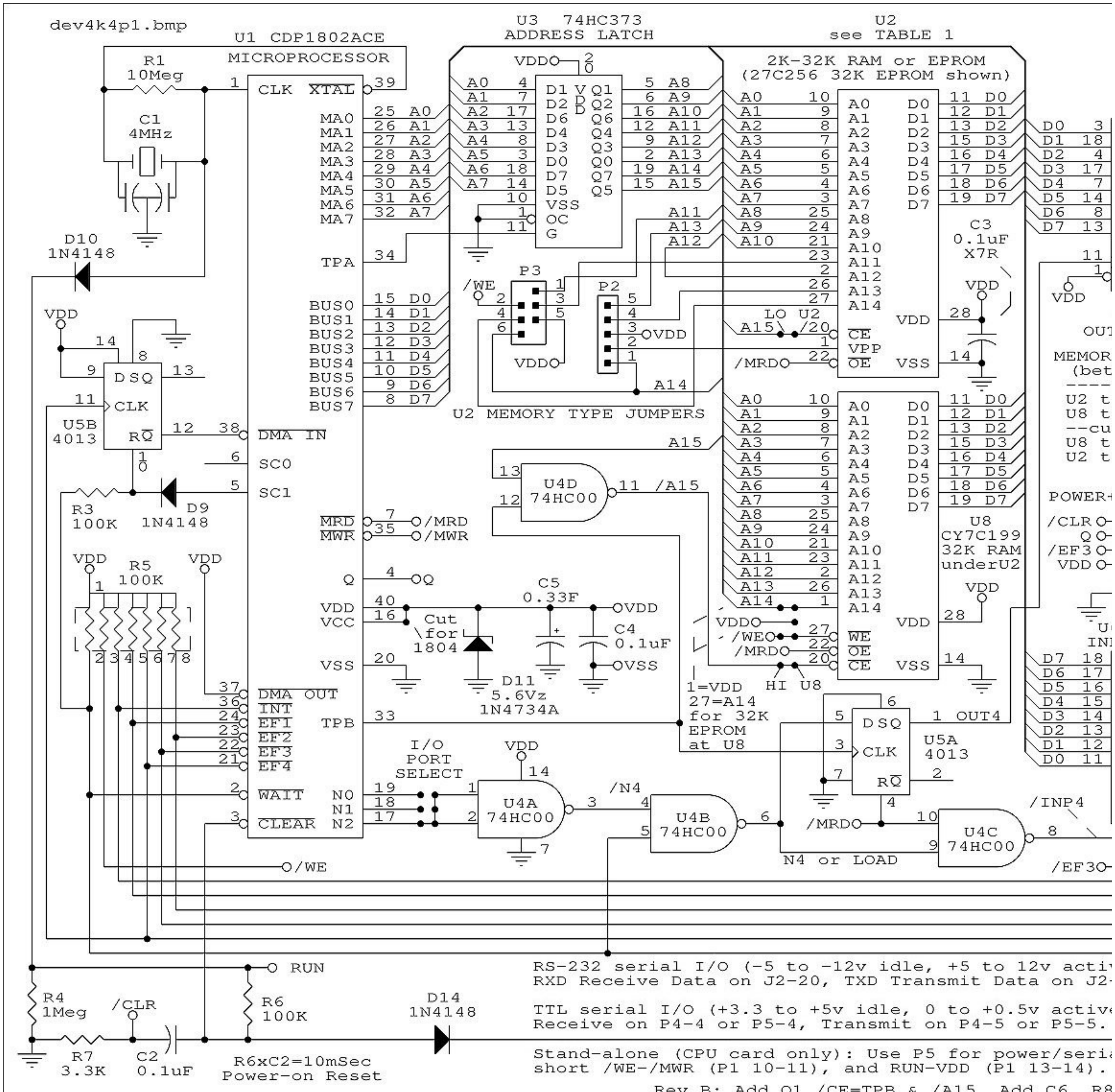
Cover Card (1:1 scale): Can use as a drilling template.



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



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



RS-232 serial I/O (-5 to -12v idle, +5 to 12v activ
RXD Receive Data on J2-20, TXD Transmit Data on J2-

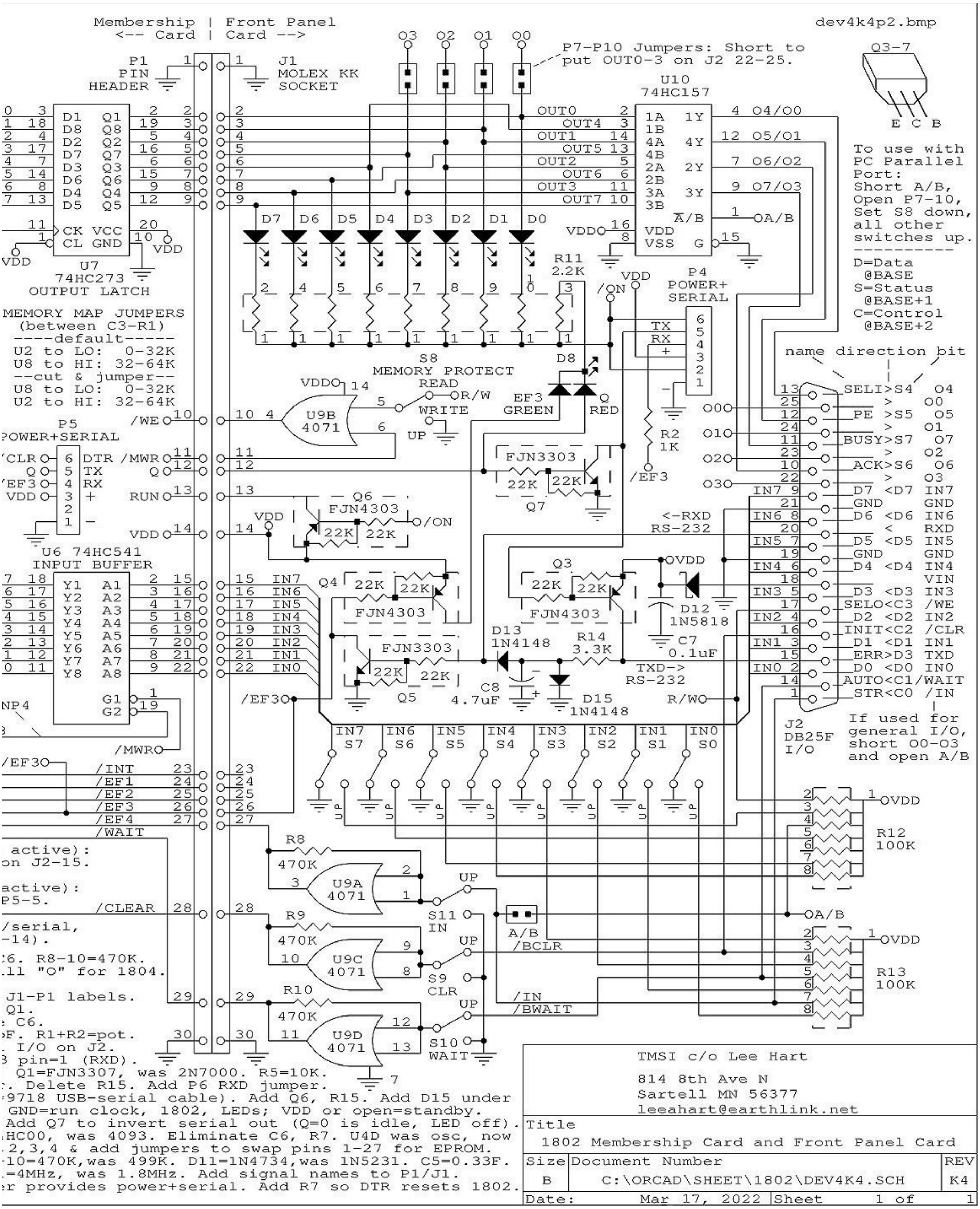
TTL serial I/O (+3.3 to +5v idle, 0 to +0.5v active
Receive on P4-4 or P5-4, Transmit on P4-5 or P5-5.

Stand-alone (CPU card only): Use P5 for power/seria
short /WE-/MWR (P1 10-11), and RUN-VDD (P1 13-14).

TABLE 1			Pin Numbers				Jumpers	
U2 part#		type	1	23	26	27	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
28C256	32K	EEPROM	A14	A11	A13	/WE	1-2,4-5	1-3,2-4
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
28C64	8K	EEPROM	VDD	A11	NC	/WE	2-3	1-3,2-4
* 2732	4K	EPROM		A11	VDD		3-4	1-3
* 2716	2K	EPROM		VDD	VDD		3-4	3-5

```
 * is a 24-pin IC: put its pin 1 into socket pin 3.
             /pin 6 /ON: GND to Run, Open for Standby.
P4 power /  pin 5  TX: TTL serial out.
+ serial<   pin 4  RX: TTL serial in.
connector\  pin 3 VDD: +3 to +5 VDC (3 AA cells etc)
            \pin 1 GND: Common for power (VSS).
```

```
Rev.B: Add Q1 /CE=TPB & /A15. Add C6. R8
 Add RUN on P4-2. I/O port=4-7. Drill "C
Rev.C: R4,8-10=499K, C6=100pF.
Rev.D: R2=6.8K. Add R14, D12. Swap J1-P1
 Add jumpers for Q0-Q3, A/B, N0-2, Q1.
Rev.E: C5=0.22F, was 0.047F. Delete C6.
Rev.F: C1=1.8MHz resonator, was 20pF. R1
Rev.G: Add U8, C6. Add Q/EF3 serial I/O
 D8: red if Q=1 (TXD), green if EF3 pin=
Rev.H: Add D14. R4=GND, was /CLEAR. Q1=F
Rev.H2: Add Q1+Q2 for /A15 inverter. Del
Rev.I: P4 now 6-pin (use Sparkfun #9718
 PCB across C8. P4 /ON is RUN+LED: GND=r
Rev.J: Fix Q4 silkscreen. Add D15. Add Q
Rev.K: R1=10meg, was trimpot. U4=74HC00,
 controls U8 /CE. U8 unscramble A0,2,3,4
Rev.K2: U6=74HC541,was 74HC244. R8-10=47
Rev.K3: U7=74HC273, was 74HC374. C1=4MHz
Rev.K4: Add P5 to CPU so USB adapter pro
```



MEMORY MAP JUMPERS
(between C3-R1)
----default-----
U2 to LO: 0-32K
U8 to HI: 32-64K
--cut & jumper--
U8 to LO: 0-32K
U2 to HI: 32-64K

To use with PC Parallel Port: Short A/B, Open P7-10, Set S8 down, all other switches up.

D=Data @BASE
S=Status @BASE+1
C=Control @BASE+2

If used for general I/O, short O0-O3 and open A/B

active):
on J2-15.

active):
P5-5.

/serial,
-14).

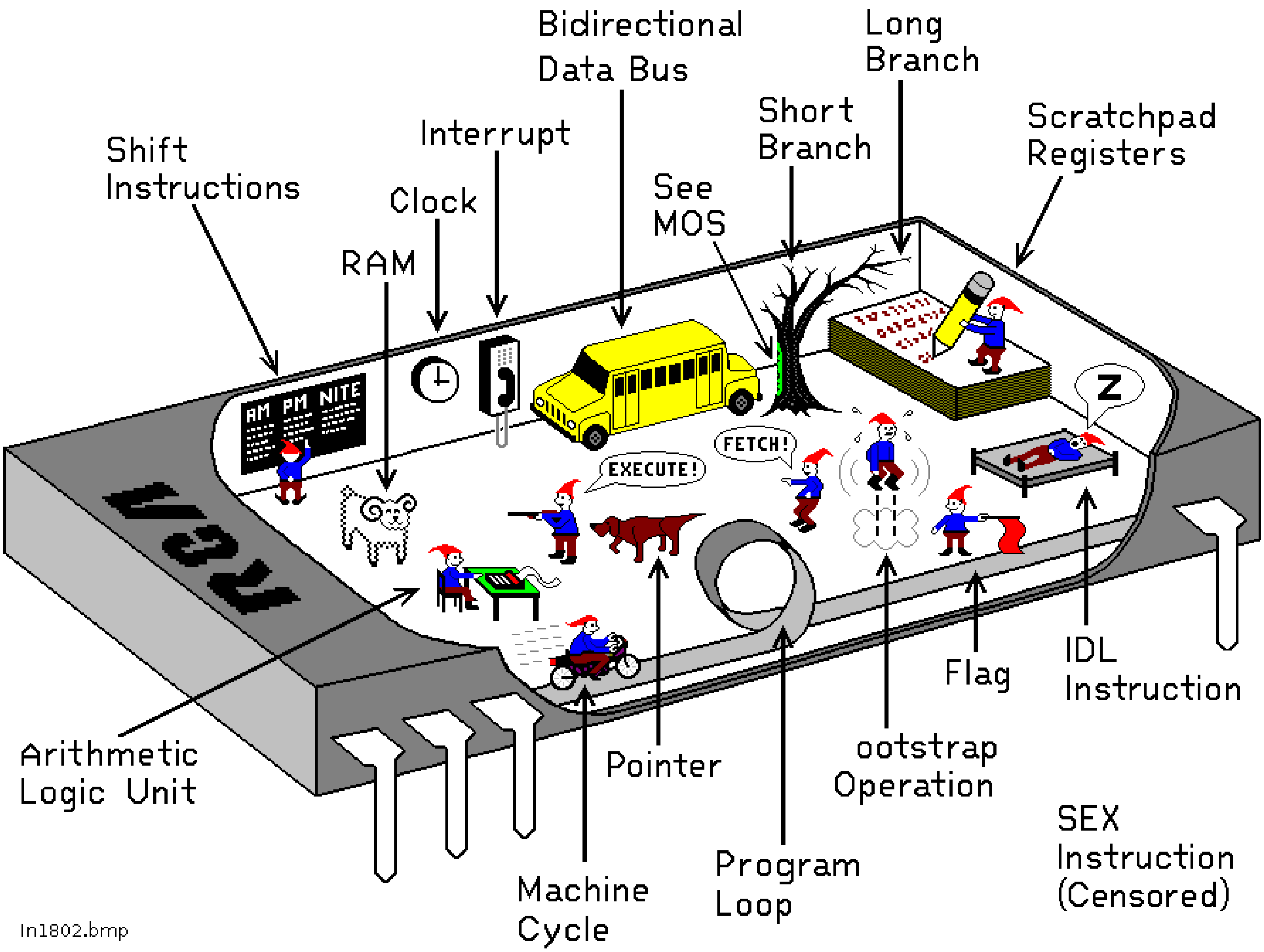
C6. R8-10=470K.
ll "O" for 1804.

J1-P1 labels.
Q1.
C6.
F. R1+R2=pot.
. I/O on J2.
3 pin=1 (RXD).
Q1=FJN3307, was 2N7000. R5=10K.
. Delete R15. Add P6 RXD jumper.
9718 USB-serial cable). Add Q6, R15. Add D15 under
GND=run clock, 1802, LEDs; VDD or open=standby.
Add Q7 to invert serial out (Q=0 is idle, LED off).
HC00, was 4093. Eliminate C6, R7. U4D was osc, now
2,3,4 & add jumpers to swap pins 1-27 for EPROM.
-10=470K,was 499K. D11=1N4734,was 1N5231. C5=0.33F.
=4MHz, was 1.8MHz. Add signal names to P1/J1.
r provides power+serial. Add R7 so DTR resets 1802.

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

Title		
1802 Membership Card and Front Panel Card		
Size	Document Number	REV
B	C:\ORCAD\SHEET\1802\DEV4K4.SCH	K4
Date: Mar 17, 2022	Sheet 1 of 1	

The Inside Story...



How the 1802 in an Elf computer REALLY works!

1802 Membership Card Summary

©2019 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/1802.htm

Register Operations		Memory Load / Store	
1n .. INC Rn	Rn=Rn+1	0n .. LDN Rn	D=M(Rn) for n not 0
2n .. DEC Rn	Rn=Rn−1	4n .. LDA Rn	D=M(Rn), then inc Rn
60 .. IRX Rx	Rx=Rx+1	F0 .. LDX	D=M(Rx)
8n .. GLO Rn	D=Rn(low8)	72 .. LDXA	D=M(Rx), then inc Rx
9n .. GHI Rn	D=Rn(high8)	F8 nn .. LDI nn	D=M(Rp)
An .. PLO Rn	Rn(low8)=D	5n .. STR Rn	M(Rn)=D
Bn .. PHI Rn	Rn(high8)=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 hhll)
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

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

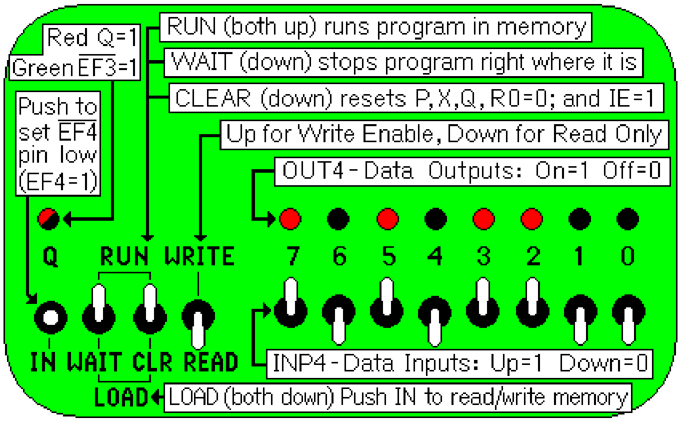
00.... IDL	wait for INT or DMA	78 .. SAV		M(Rx)=T
68 ..	reserved for 1804/5/6	7A .. REQ		reset Q=0
70 .. RET	X,P=M(Rx); inc Rx; IE=1	7B .. SEQ		set Q=1
71 .. DIS	X,P=M(Rx); inc Rx; IE=0	Dn .. SEP	n	set P=n
79 .. MARK	T=X,P; M(R2)=X,P; X=P; dec R2	En .. SEX	n	set X=n

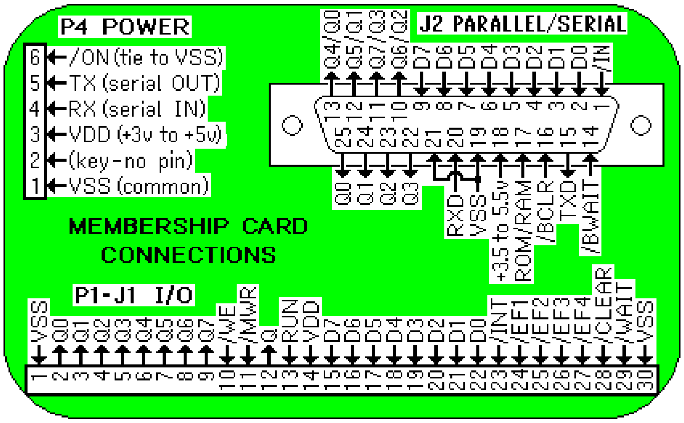
Input/Output Instructions (with N output on pins N0, N1, and N2)

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.


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

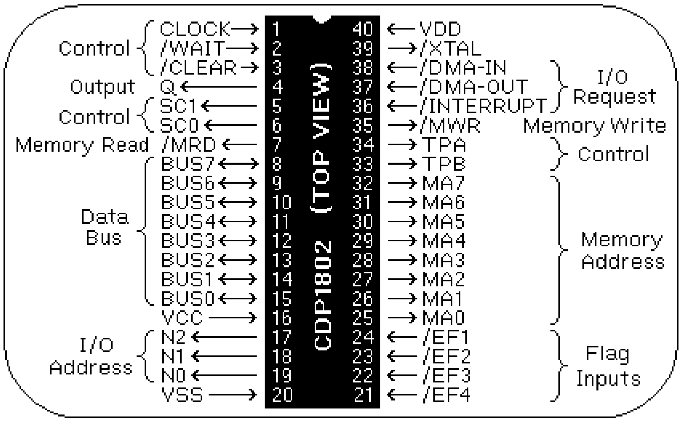


Hex-Binary-Decimal Conversion

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	0x	1x	2x	3x	4x	5x	6x	7x
LSD	000	001	010	011	100	101	110	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



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

1802 Membership Card Summary

©2019 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).

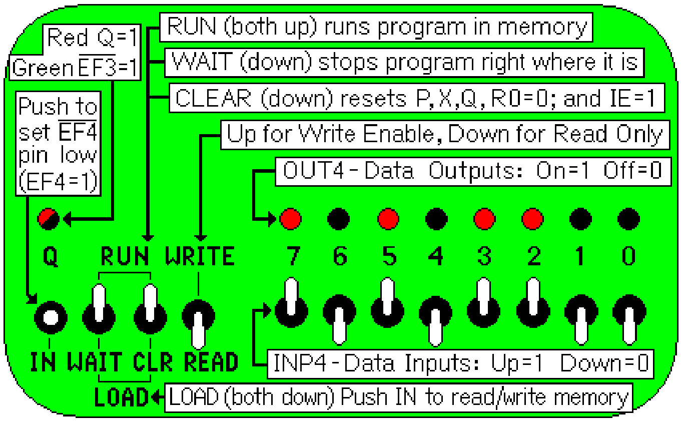
DMA: Finish executing current instruction, then use R0 as memory address for data transfer. DMA-IN writes data from bus to memory, or DMA-OUT reads data from memory to Bus. Then increment R0.

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/1802.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

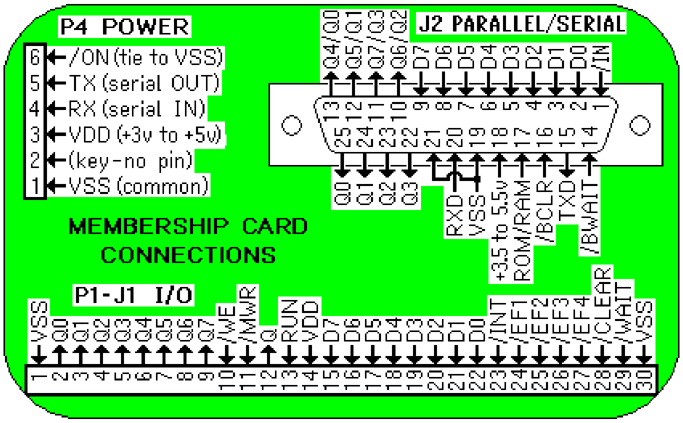
Memory Load / Store

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 hhll)
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	0x	1x	2x	3x	4x	5x	6x	7x
LSD	000	001	010	011	100	101	110	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)

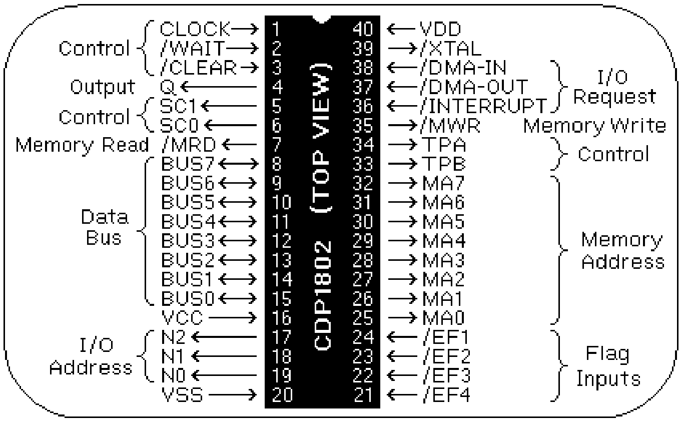
Control Instructions

00.... IDL	wait for INT or DMA	78 .. SAV		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).


CDP1802 (TOP VIEW)
Control
CLOCK→ 1
/WAIT→ 2
/CLEAR→ 3
Output
Q← 4
Control
SC1← 5
SC0← 6
Memory Read
/MRD← 7
Data Bus
BUS7↔ 8
BUS6↔ 9
BUS5↔ 10
BUS4↔ 11
BUS3↔ 12
BUS2↔ 13
BUS1↔ 14
BUS0↔ 15
VCC→ 16
I/O Address
N2← 17
N1← 18
N0← 19
VSS→ 20
40 ←VDD
39 →/XTAL
38 ←/DMA-IN
37 ←/DMA-OUT
36 ←/INTERRUPT
I/O Request
35 →/MWR
Memory Write
34 →TPA
33 →TPB
Control
32 →MA7
31 →MA6
30 →MA5
29 →MA4
28 →MA3
27 →MA2
26 →MA1
25 →MA0
Memory Address
24 ←/EF1
23 ←/EF2
22 ←/EF3
21 ←/EF4
Flag Inputs