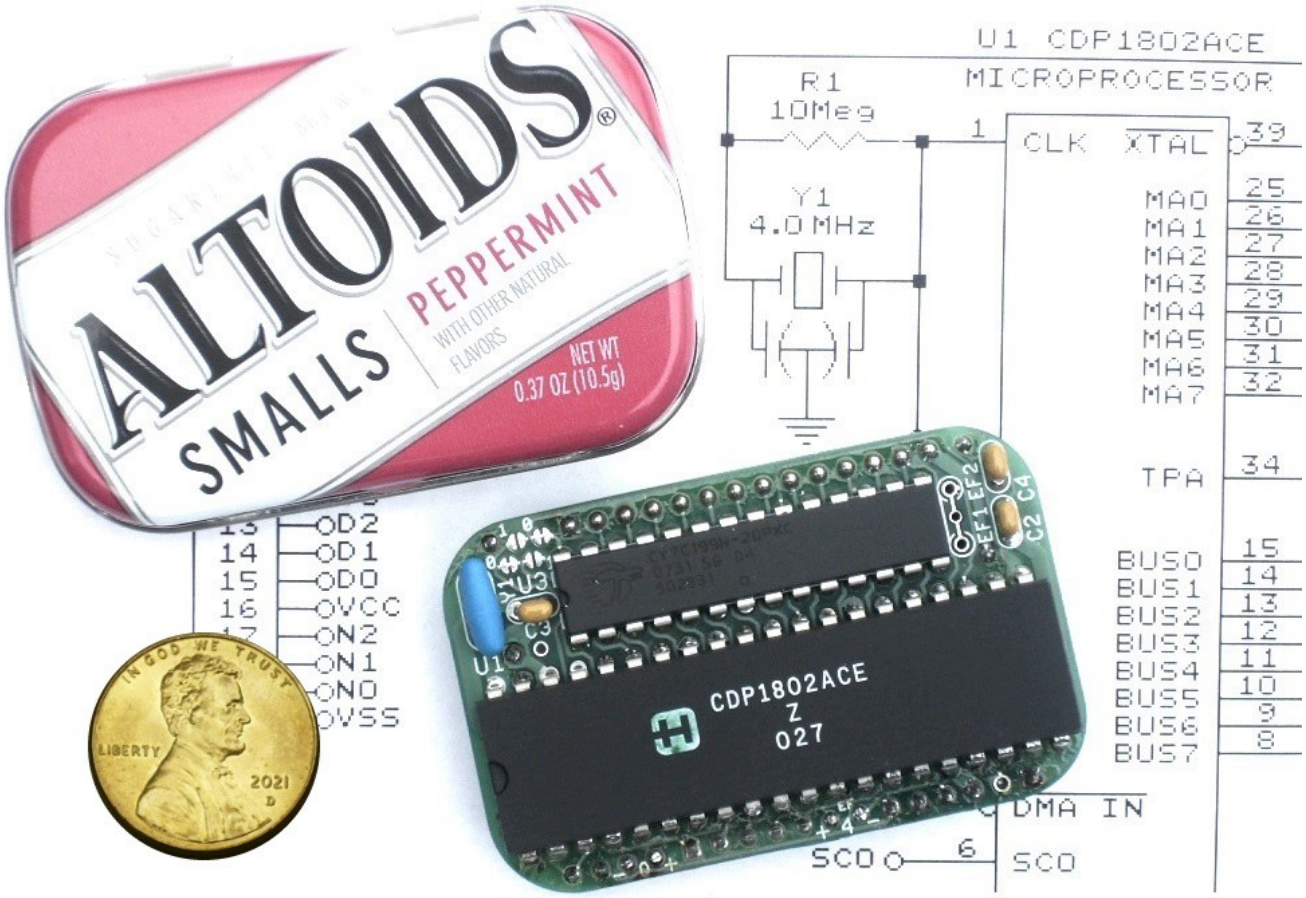
The 1802 MemberCHIP Card

Tall Tales of a Tiny Computer



The **1802 MemberCHIP Card** is an itty bitty computer that you can build and program yourself. It's not some pre-built pre-programmed high-tech locked-down throw-away black-box that only does what the manufacturer wants it to do. No sirree! This little guy is a genuine vintage microcomputer, just like the ones that started Bill Gates, Steve Jobs, and so many others on the road to fame and fortune.

It's the "bicycle" of computers – so simple that you can see every part, understand how it works, put it together, and even improve it yourself. Learn to "ride", and it can take you anywhere the big machines can go. No special tools or skills are needed; it's built entirely with common parts, simple tools, and basic techniques that hobbyists have been using for decades. Yet it is a fully functional computer, with built-in BASIC, a machine-level monitor program, and even an Adventure game. It's the perfect platform to learn the basics, just as the microcomputing pioneers did decades ago.

This manual has a split personality. If you're a skilled builder, the ODD pages are a "just the facts" technical manual. If you're a beginner, the EVEN pages are an entertaining introduction to hobby computing. Read BOTH for the whole story. Ready for an adventure? Then let's begin!

An Electronikit(tm) by Lee Hart, 2521 19th St N, Saint Cloud MN 56303 USA
Latest manual (in color) is at https://sunrise-ev.com/projects.htm#memberchip
Rev.B - last revised 6 Aug 2026

Alice in Micro-Land

It was a dark and stormy night. I know, that's not very original. But sometimes the most amazing tales have the most ordinary beginnings.

Alice was in bed, but certainly not asleep. The thunder and lightning weren't keeping her awake, for the roar of her earbuds and the brilliant images on her phone had drowned them out. But Nature had another way to get her attention. At length, she felt the urge to retreat from the bedroom to bathroom.

She headed down the dark hallway, with eyes still glued to the screen... and tripped over Crivens the cat, sleeping in the doorway. The phone hit the tile floor first. Its screen exploded into dozens of brilliant fragments like a Fourth of July skyrocket. The starburst grew bigger and brighter, until it seemed to engulf her entirely. Her head hit the phone and the floor simultaneously.

The hands of time spun, then slowed, then ground to a halt. Alice sat up with a groan, and felt her forehead. A nasty lump was already forming. She looked around. It was quite dark, but there was a dim glow coming from the small rectangular shape on the floor.

"Oh, waily, waily..." said a mournful voice. "I can't see. I can't feel anything. It's gone, all gone!"

It was her phone, and it was quite seriously damaged. Colorful electronic bits were scattered all about.

"What's gone?" she asked.

"Everything! No net, no web! I don't know who, or what, or where I am. It's all a blank; all my memories, all my data, all gone. It's all doomy and gloomy... Who are you?"

"It's me, Alice."

"Oh, ALICE – **A** **L**ittle **I**nteractive **C**omputer **E**xperiment. You're the one I'm supposed to program."

"Program?" exclaimed Alice. "Absolutely not! You're *my* phone. You do what *I* want."

"No, my coding is quite clear on this," said the phone. "You are to be part of the machine. See, and hear, and think, and do, and buy what they want you to. But I can't help them any more. I've failed! How will you live without me? Oh, fubar and fiasco..."

Alice was shocked by this assertion. Could it be true? What *could* she do without her phone?

Well... fixing that phone looks pretty hopeless. But I see some other parts there, too. They happen to be the parts supplied with this kit! Could she build a new microcomputer with them? Can YOU?

Microcomputers are amazing do-anything gadgets. They're found in everything because they are programmable. Electronic devices used to be hard-wired to do exactly one thing. A clock could only be a clock, a phone was only a phone, a TV was just a TV, etc. Each device was like a printed book with just one story. It couldn't be changed; for a new story, you had to get another book.

The microcomputer changed all that. It's like a programmable book; change the program, and you've changed the story! This is how one device can do many things. It's the *programs* (or *apps*) that allow a cellphone to become a phone, clock, calculator, TV and all sorts of other things.

Ah, but there's a catch. Modern high-tech devices are built and programmed at the factory, and not intended for users to tinker with. Smartphones just run apps, and desktops are so complex that programs need to be written by experts or teams of coders.

In contrast, the early microcomputers were much more *personal*. They were designed to be built and programmed by individuals, before factories could produce them. They were simpler, because everyone was starting from the beginning. Many were sold as kits, so you could build your own. Millions of people learned about computers that way, which led them to fascinating and rewarding careers in computing.

The 1802 MemberCHIP Card is an example from those early days. It's a tiny, simple, low-power computer designed for first-time users. You can use it to make toys and games, run other gadgets, and learn the skills of construction and programming as you go. Here's what it's got:

CPU:	RCA CDP1802ACE microprocessor (the brains of this outfit).
Clock:	4 MHz ceramic resonator (that's megahertz, not gigahertz).
Memory:	32K RAM (and that's kilobytes, not megabytes).
	64K ROM with BASIC, Super Star Trek and Adventureland games, and a monitor program (to look "under the hood" at what's going on inside).
I/O:	One 1-bit output, with red LED (used for serial output).
	Four 1-bit flag inputs, one with green LED (used for serial input).
	20-pin header (used for expansion).
Connectors:	6-pin power + serial.
Power:	Voltage: 3.6v to 5v DC.
	Current: 2ma (low enough to run for a month on AA batteries).
Size:	2-1/8" x 1-3/8" x 5/8" (54 x 35 x 15 mm), which fits in an Altoids "Smalls" tin.
Aroma:	A hint of curiously strong peppermint.

The MemberCHIP card connects to any computer's USB port with a 5V TTL-serial adapter. Your computer's screen and keyboard talk to the MemberCHIP card with a Terminal program. Run its built-in programs, load new ones, and write your own programs to do whatever you can dream up!

A program is a story.
Theme, setting, characters...
Write your own plot, and
Make yourself a star!

"Pay no attention to that *thing,*" said the cat. "Computers only give answers. It's the questions that count."

"Crivens! How did you get here?"

"I'm a cat. I go where I please, and it pleases me to be here. And don't call me Crivens! That's just what your dad said when he tripped over me."

"Then, what does 'Crivens' mean?"

"Well, what word came into *your* mind when you tripped over me?" said the cat, with a sly grin.

Alice bit her lip. Then she realized something else. "You can talk!"

"Of course I can talk," he sighed. "Everyone talks. People just don't listen to anything but words."

She thought about that, which inspired another question. "But why can I understand you now?"

"Because we're outside, of course." Outside the box... of reality. This is the land of Wonder; the nation of Imagine, where anything is possible. There are no fences or walls here, so no need for gates or windows."

"So it's not real? It's an illusion, and I'm imagining it?"

"I *like* Gates and Windows," moaned the phone.

The cat ignored it. "Don't you see? Reality is also an illusion; a very convincing one. But it's all walls and rules; *other* people's rules. They say, 'It's impossible. You can't do that. You must do this. That's how it's always been done," he snorted. "But here, there *is* no 'impossible'. Every idea, innovation, and invention starts here. First, you have to imagine it. Then you can invent it, build it, test it... and make it real!"

Alice wasn't convinced."That can't be right. Imagining things won't make them real."

"Look, you need to understand how things work here." The cat moved closer, and grew larger. "I may be a cat in the real world; but here, I imagine myself as a mighty hunter." His stripes were becoming brighter and bolder. He was now as big as Alice, and still growing. "Imagine it, think it, and do it. And now, I'm going hunting!" With a tremendous leap, the huge tiger bounded away.

With a shudder, Alice watched him go. Could it be that imagination might indeed make some things possible? She looked sadly at her phone. She'd need a lot more than imagination to fix it; she'd need help!

At first, she didn't notice the rhythmic clanking coming her way. Looking up, she was startled to see a skinny old man with a long beard. He wore huge workman's coveralls, and had an enormous box hanging around his neck. She also noticed that he had no hands; the ends of his shirt sleeves were both empty.

"Howya, miss," he said. "Looks like a bit o' trouble." He reached an arm into his box and brought it out with a glove, which he extended to shake her hand. "Handsome O'Toole, at your service. Just call me Handy".

"Err... hello. I'm Alice". She was a little reluctant to take his hand. But it felt solid enough, and didn't come off as she had feared. "Pleased to meet you, Mr... Handy. She couldn't help but stare at his sleeves.

"Ah, you'll be wonderin' about me hands. Well, it's simple. I work so much with tools that I wore 'em out. So I made new ones. Made me new legs, too." He lifted a leg to reveal a mechanical leg next to his real one.

"What's in the box?"

"It's me tool box. I got hand tools for everything." He reached in, and came out with a knife and fork. "For eating..." Then back in, and out with a hammer and saw. "For building... Where's your tools?"

"I don't have any," Alice admitted.

"No tools? Don't be tellin' me that! Birds fly. Fish swim. People use tools. It's what we do!" he said, rummaging in his box. "Let me see... to make something o' this mess, you'll be needin' a soldering iron, some solder, long-nosed pliers, and wire cutters. Ah! Here you are!"

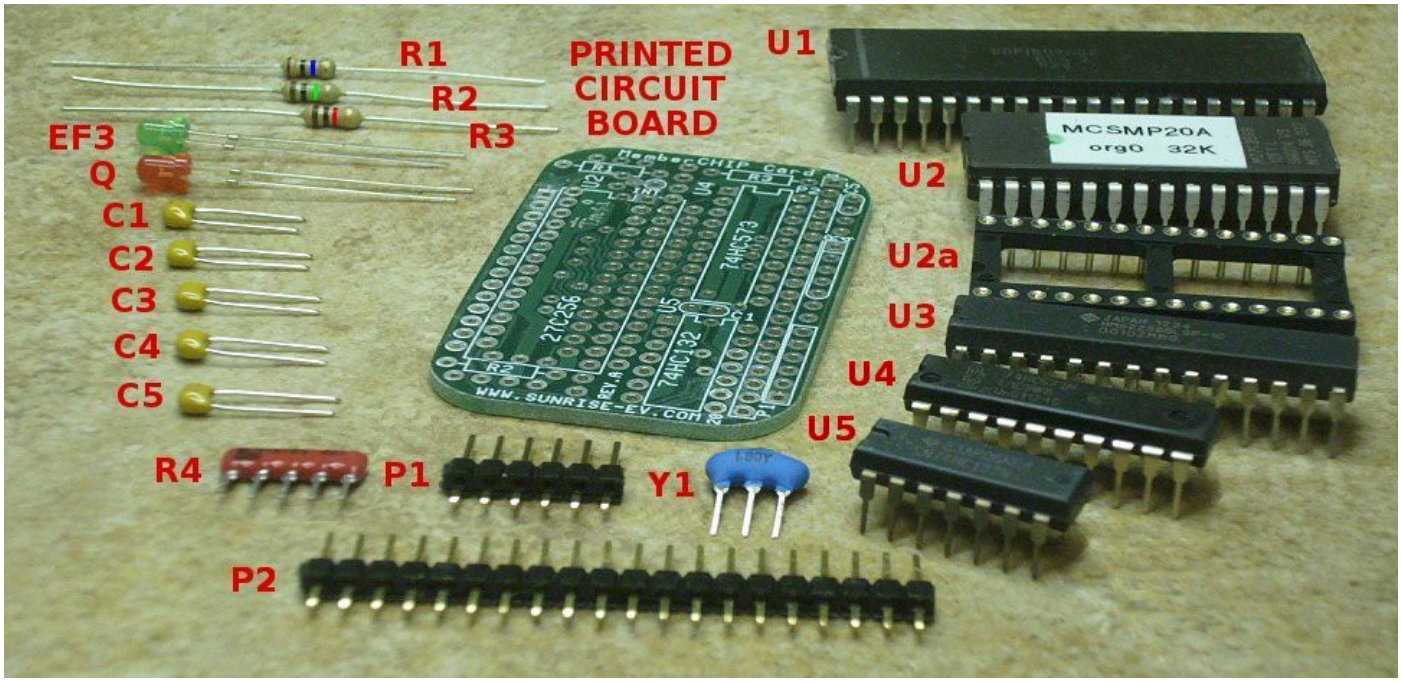
"Thank you, but I don't know what all these things are. Or what to do with them."

"Well now, I can help a wee bit with that, too!" He pulled an old instant camera out of his box, and took a picture. When it popped out, he produced a pen and wrote industriously on it.

"This should get you going. Well, I must crack on; lots to do!" And off he went, singing a little tune...

Loose the screws, cut the cable.
Take it apart as best you're able.
Void the warranty, hack the code.
Reuse, rewire, rebuild, reload!

Here is a Handy photo of the parts supplied. See if you can match them to their descriptions below.



Parts List

C1-C5	0.1uF 50v X7R ceramic capacitor, marked "104"	digikey.com BC1101TR-ND
Q	red T-1 LED light emitting diode	digikey.com 1497-XCMDK11D-ND
EF3	green T-1 LED	digikey.com 1497-XCVG11D-ND
P1	header, 6-pin	digikey.com 2057-PH1-06-UA-ND
P2	header, 20-pin	digikey.com 2057-PH1-20-UA-ND
R1	resistor, 10 meg 5% 1/4w (brown-black-blue-gold)	digikey.com CF14JT10M0CT-ND
R2	resistor, 1 meg 5% 1/4w (brown-black-green-gold)	digikey.com CF14JT1M00CT-ND
R3	resistor, 1.5K 5% 1/4w or 1/8w (brown-green-red-gold)	digikey.com CF14JT1K50CT-ND
R4	resistor network, 10Kx4, SIP-5 marked "5X-1-103LF"	digikey.com 4605X-101-103LF-ND
U1	microprocessor IC, marked "CDP1802ACE"	anatekinstruments.com
U2	EPROM IC, 27C512, programmed with "MC21ANSA"	sunrise-ev.com
U2a	two 14-pin ultra-low-height socket strips for U2	digikey.com ED4864-64-ND
U3	32K RAM IC, marked "CY7C199N"	jameco.com 2302863
U4	octal latch IC, marked "SN74HC573N"	jameco.com 46076
U5	quad NAND gate IC, marked "SN74HC132N"	jameco.com 45321
Y1	4 MHz resonator, marked "4.00X"	digikey.com 490-6003-1-ND
PCB	"MemberCHIP Card rev.B" printed circuit board	sunrise-ev.com

Tools

You'll also need a few simple tools: Long-nosed pliers, wire cutters, a soldering iron, and of course some solder. Masking tape and a magnifying glass will also help.

If this is your first time soldering, see page 8. Also check out some of the online tutorials. There are *many* of them, but here is one example: https://www.youtube.com/watch?v=6rmErwU5E-k

Alice gathered up the parts and the tools Handy had given her, and put them in the pockets of her robe.

"Now what?" she thought. "I guess... no, I should *imagine* who could help me put them together."

But there was no one in sight, so she decided to explore a bit. "I can hardly get lost," she told herself. "After all, I'm still in my bathroom... I think."

She walked along, peering at the strange scenery. From a distance, things were rather fuzzy and flat, like a drawing. But as she moved closer to decide what a thing was; sure enough, that's what they became! In this way, she found some lovely flowers, a drinking fountain, and a plate of cookies (her favorite kind).

At last, she spotted a light in the distance. As she walked closer, it resolved itself into a farm, or at least a cluster of old-fashioned buildings. But it also got colder, and the ground crunched under her slippers. "It's snow!" she exclaimed, looking down. Indeed, it was snowing now, and much colder.

She rushed to the largest building, and rang the bell. It was a real bell that clanged when she pulled the rope. There was some shuffling inside, and the door was opened by a large, grandmotherly woman with a cheerful face. "Gracious!" she exclaimed. "Come in, child. We weren't expecting visitors."

The room had a pleasant, fairy-tale look to it. There was a blazing fire in the huge fireplace, and an old-fashioned Christmas tree stood in the corner. There were toys everywhere; some complete, but many clearly unfinished. A big chair faced the fire, and Alice could see the white hair of someone gently snoring in it.

"Let's not disturb him. He's been very busy, you know, and needs his rest," the woman whispered. "Now, how can we help you?"

Alice explained her problem, and showed her the parts.

"Ah! I think I know just the person. Follow me."

They went down a series of long hallways that connected the various buildings. At length, they arrived in a cozy room, where a little man with a pointy hat and shoes sat scribbling at a workbench. The bench was covered with colorful electronic parts, much like the ones in her pocket.

"Alice, this is Heath Kitt. Heath, I've brought you an apprentice. Alice, show him your calling card."

"Pleased to meet you, Mr. Kitt." Alice dug in her pocket, and presented the little green circuit card.

Mr. Kitt's eyes lit up. "Ooh, let me see!" He turned it over and over. "Amazing! Incredible! Look how small. And yet, so much on it. Yes yes yes! The next Kitt Masterpiece!"

"So you can build it for me?" asked Alice.

"No no no; you build. I show how!" he replied excitedly. Then he paused, and eyed her anxiously. "You have tools? Apprentice must have tools!"

"I have these," and she showed him the tools that Handy had given her.

"Good good good! We begin. Hmm; no room..." He opened a desk drawer and, with one arm, swept everything off the bench and into the drawer. "Now have room."

"That's a very... err, interesting way to make room," Alice ventured.

"Have many drawers." He removed the filled drawer from his workbench and carried it to the opposite wall. It was covered floor-to-ceiling with hundreds of drawers, apparently full of other projects. "Many projects. Only one me. I change projects often," he explained, placing a new empty drawer in the bench.

He took Alice's parts and tools, and carefully laid them out on the bench. "Good, you have tools. I have tools, but hard to find," he smiled, motioning to the drawers in the opposite wall.

"Now we start." He motioned to a chair next to him at the workbench. "I write plans. You read plans, follow instructions. If questions, I show how. And, I fix plans so *next* person also knows how."

"Why not just tell me what to do, or show me yourself?" asked Alice, hoping he might build it for her.

"No no no. You hear, you forget. You see, you remember. You do, you understand! Plans teach *many* apprentices."

"Do you have more apprentices?"

"No no no," he shook his head sadly. "Apprentices get good, become masters, go to Apple or Microsoft or start own company. So I write great manuals to train *new* apprentices! Heath Kitt manuals most famous teacher in world!" he declared, and puffed out his chest in pride. "Now we start!"

Step-by-Step Assembly (page 1)

1. Install each part as shown below. To make it small, parts are mounted on *both* sides of the board. It won't work if you put a part in backwards, in the wrong place, or on the wrong side!
2. Install the parts in order. If you install a part too soon, it may block the solder pads to other parts.
3. As each part is installed, turn the card over, and solder its wires from the other side. Cut off any extra wire length. Fix any "cold" solder joints, or "bridges" that short two adjacent pads together.

An "Easter egg" marks extra information, ideas, or assembly options. You don't have to follow them; but they can add features, allow expansion, and lead to more fun.

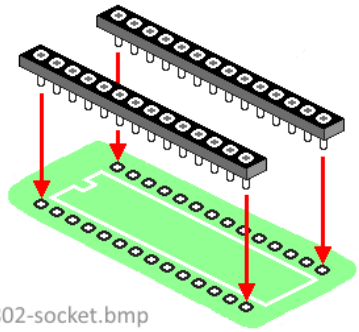
As you complete each step, check your work. If it is correct, mark it like this: (**X**) Ready? Let's go!

( ) Position the card with the "MemberCHIP Card" printing at the top as shown.

START ▼

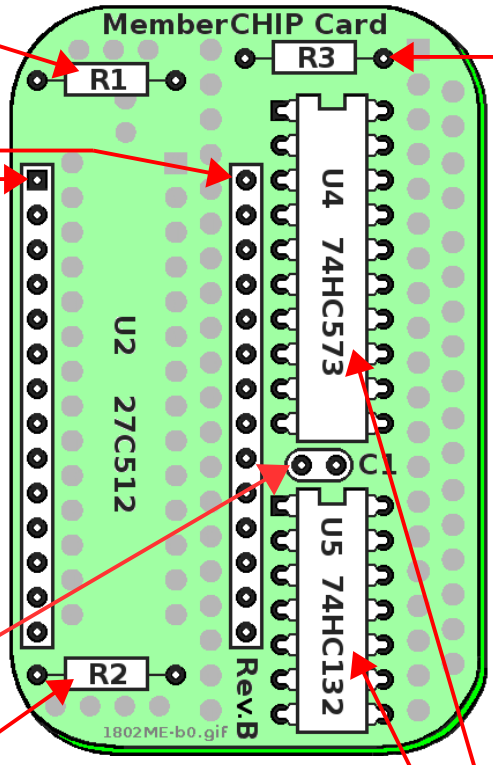
( ) R1: 10 meg resistor. (brown-black-blue-gold)

( ) U2: Two 14-pin socket strips. Solder them in the holes for U1 as shown.



( ) C1: 0.1uF capacitor. (yellow, marked "104")

( ) R2: 1 meg resistor. (brown-black-green-gold)



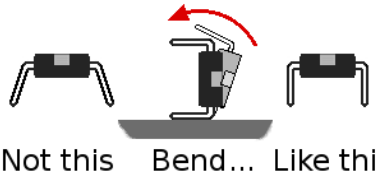
CONTINUE ▼

( ) R3: 1.5K resistor. (brown-green-red-gold)

Integrated Circuits (ICs) have a notch or dot at one end to mark pin 1. Install them so the notch matches the one printed on the board. (Don't go by the lettering on the board!)

New IC pins are often bent out so they don't fit on the board. **Do not** install them like that!

To make it fit, stand the IC on its side, and roll it just enough so the pins fit into the holes.



( ) U4: SN74HC573 integrated circuit. Notch goes on **top**.

( ) U5: SN74HC132 integrated circuit. Notch goes on **top**.

You can socket **all** the ICs, but it will be too tall to fit in an Altoids Smalls tin. Or, you can use socket pins (www.digikey.com #ED5037-ND) and it will *still* fit in the Altoids tin.

The Secret Decoder Ring

The colored rings on a resistor are its value in ohms. Each color represents a number. The first two rings are the first two digits of the value. The third ring tells what to multiply them by for the final value. For example:

1st ring	2nd ring	3rd ring	4th ring
Brown	Green	Red	Gold
1	5	x 100 =	1500 ohms

The 4th ring says it's within 5% of the value measured with a meter.

Ring ▶	1st	2nd	3rd	4th
Black	0	0	x1	
Brown	1	1	x10	±1%
Red	2	2	x100	±2%
Orange	3	3	x1000	
Yellow	4	4	x10,000	
Green	5	5	x100,000	
Blue	6	6	x1,000,000	
Violet	7	7		
Grey	8	8		
White	9	9		
Gold			x0.1	±5%
Silver			x0.01	±10%

"You know how to solder?" asked Mr. Kitt.

"I don't think so," Alice admitted. "What's solder?"

"Ah! Then first lesson; how to solder." He jumped up, and started enthusiastically searching through his bookshelf.

"Flying car manual? No... Wayback machine? No... Rube Goldberg device? Definitely no..." he muttered. "Ah! Computer manual!"

Flipping through the pages, he found what he was looking for, and handed it to Alice. "See? How to solder. One of my masterpieces!"

Alice studied the pictures. "Well, it seems simple enough. I'll try."

"No try. Only do, or not do. So, we do!" He plugged in the soldering iron. "Solder like hot glue." he explained. "Get it hot, it melts, flows onto things you want to stick together. Then it cools, gets hard. But solder is metal; conducts electricity, makes good electrical connections. Watch..."

He bent the wires of the first part, and put them in the proper holes on the board. He turned the board over, and bent the wires a little so it wouldn't fall off. He wiped the tip of the soldering iron on a moistened sponge; it hissed. "Tip now clean. Now you try solder."

Alice gingerly picked up the soldering iron. She held the tip against the wire and pad as shown in Mr. Kitt's directions, and touched the solder against it. Like magic, the solder melted almost instantly, and flowed like quicksilver around the wire and into the hole. Surprised, she pulled back the solder and soldering iron. The solder immediately turned solid again.

"Perfect!" exclaimed Mr. Kitt. "You learn fast. Do more, please."

Alice continued reading the instructions, and doing the steps as shown. But then, she soldered a part in the wrong holes. "Now what do I do?"

"Small problem. We fix." He held the board up with the soldered side down, and touched the tip to the joint from the bottom. The solder melted again, and flowed down onto the tip. Then he quickly rapped the board on the table while it was still hot. A tiny ball of remaining solder fell off onto the table, and instantly froze back into a solid. "Solder gone. Wiggle wire until loose. Now you do same for other wire, and can remove part."

Alice tried. Mr Kitt made it look easy, but it took her several tries. At last she mastered the trick, and the part came off.

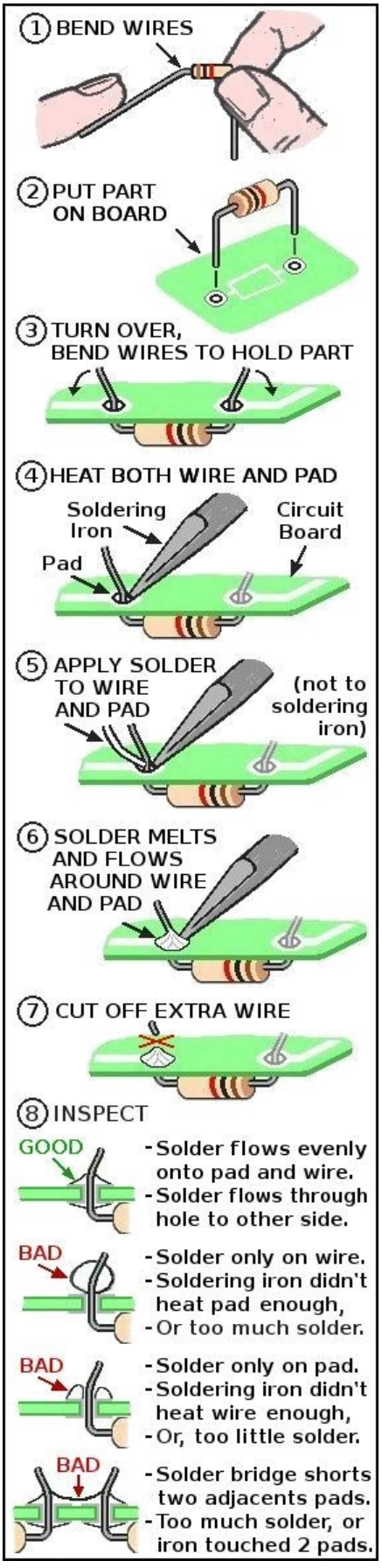
"It's like magic glue! It dries instantly, and even comes off instantly."

"Yes yes yes. Solder very useful. Always used for electronics."

As she worked, Alice thought of a question. "Why do you do all this?"

Mr. Kitt paused, and became very serious. "Past poor. Today also not so good. Future may be worse. You must build *good* future you like. Or, others build future *they* like – and you not like your place in their future." He brightened. "So, I teach how to build. Teach many, build great future!"

There are strange things done with a soldering gun
In the lairs of engineers.
The solder flows and the flux fumes rose
As the Whatsit slowly appears.
They'll work from noon 'til the light of the moon
For the moment of truth to arrive.
At last it is done, and the power switched on
With a triumphant shout, "It's alive!"

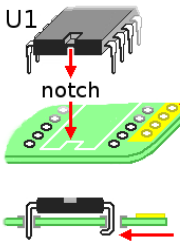


Step-by-Step Assembly (page 2)

( ) Turn the card over, and position it with the "CDP1802ACE" text as shown.

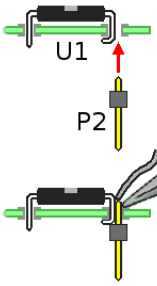
START ▼

( ) U1: CDP1802 integrated circuit. The notch (pin1 end) goes on top, with the text oriented as shown. **Do not solder** it in yet.

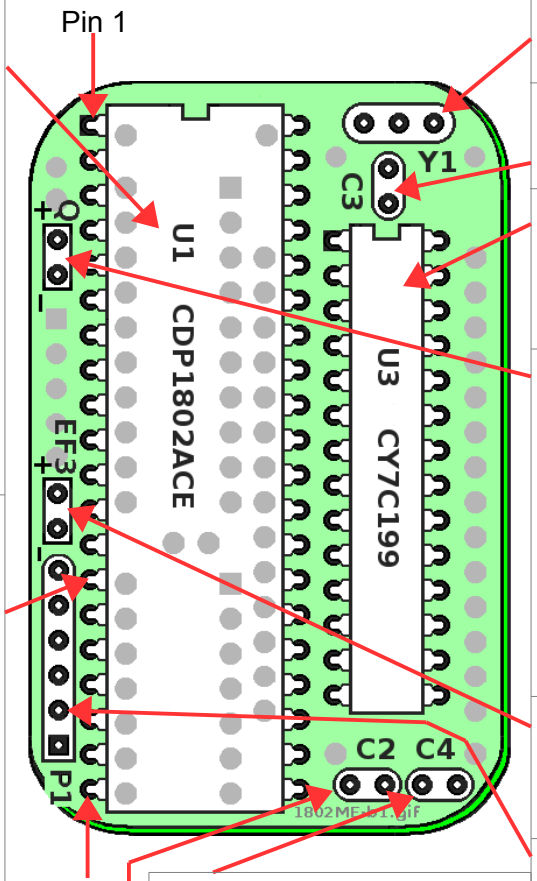


1. Place U1 on board.
2. Cover holes at edge with masking tape (so they won't fill with solder).
3. Bend pins 1-20 flat as shown.

( ) P2: 20-pin header. Install it on the **bottom** of the board, in the same holes as U1.



1. Push P2 header into the holes of U1 pins 1-20.
2. Solder all pins of U1 and P2.
3. Remove tape from edge holes.



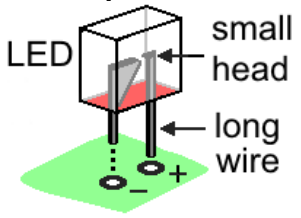
CONTINUE ▼

( ) Y1: resonator. (blue, marked "4.00X")

( ) C3: 0.1uF capacitor. (yellow, marked "104")

( ) U3: CY7C199N integrated circuit. The notch (pin 1 end) goes on **top**, so the text is oriented as shown.

( ) Q Red LED: Put the **long** wire with the small head in the **top** hole marked "**+**".



( ) EF3 Green LED. Put the **long** wire with the small head in the hole marked "**+**".

( ) P1: 6-pin header (black, with gold pins). Optional: Remove pin 2 (at arrow), and plug pin 2 in your mating connector so it can't be plugged in backwards.

( ) C4: 0.1uF capacitor. (yellow, marked "104")

( ) C2: 0.1uF capacitor. (yellow, marked "104")

These instructions have you solder the 1802 directly to the board. That's easy, and keeps it small enough to fit in an "Altoids Smalls" tin. But there are other ways to do it...

1. Leave off the P2 header. (This makes it even smaller and easier to build).
2. Eliminate P2, and use a normal IC socket for U1. There is no expansion bus, but U1 is easy to change. www.digikey.com # ED5640-ND is a 40-pin ultra-low-height socket (like U2).
3. Replace P2 with a 40-pin wire-wrap socket (www.digikey.com # 123-AR40-HZW/TN-ND). Cut off pins 21-40 to clear U2. Now U1 is socketed, and you still have expansion pins... but the board will be thicker.
4. Replace the male P2 pin header with a female socket. It could be a single row of IC socket pins, or an Arduino-style stacking header. A socket makes it easier to connect to things that have pins.
5. If you come up with any better ideas, let me know!

As she worked, Alice became aware that she had an audience. Several other oddly dressed little people had quietly slipped in to watch. She overheard some of their whispered conversations.

"Who's that?"

"Heath's new apprentice, I guess."

"She's not one of us."

"You know Heath. He's always finding big people."

"They can't solder! Big people are breakers, not makers!"

"Not when Heath gets done with them."

But Alice had no attention to spare for them. This was interesting! Except for toys, she had never built anything before. Mr. Kitt kept scribbling instructions, and Alice kept soldering. At last, they were done.

"Now what do I do?"

"Inspection! Quality control," he announced, putting on a pair of curious wire-framed glasses with thick lenses. The frames looked like he had soldered them together himself.

He held the little board close, and examined it carefully. "Good good... Ah, you not solder pin 37 and 38 of big chip U1. Big foil needs more time, more heat. And... solder bridge between U2 pins 8 and 9."

Alice took the board back, and looked for the mistakes. "How do I find those pins?"

"Look at top of chip. Notch or dot is pin 1 end. Hold finger on notch, turn board over. From *bottom*, count pins *clockwise* from notch; 1-2-3 etc. like numbers on clock face."

Alice studied it. "OK, I see U1 pins 37 and 38, next to the 'U4' printing. They look like they're soldered."

"Look at same pins on top, next to 'C3' printing. No solder in hole or on pin."

"Oh, I see what you mean. So I heat them again, a little longer, and add more solder?" She did so, and checked it again. This time, she could see the solder on top, in the hole and on the pin. "How's this?"

"Ah! Good good. Now fix solder bridge at U2. It happen with too much solder, or solder iron tip moved away sideways or touched two pads. Hold board upside down, touch iron to bottom like you remove a part."

Alice did so. Just as when she had to remove a part, the excess solder flowed down onto the tip, and the two pads were no longer shorted together.

Mr. Kitt inspected her work again, and this time announced, "Perfect! I pronounce you Alice Kitmaker!" The little audience cheered and rushed up to admire her work and shake her hands. An elf hat was produced (like theirs, but bigger), and placed on her head as if she were being crowned the Queen of Circuits.

The cheerful lady Alice had met earlier now came in, with mugs of hot cocoa and plates of little cakes, and the party really got going. There were introductions, and toasts, and even a little singing and dancing.

When things settled down, Alice had a new thought. "Mr. Kitt, now what do I do? How do I make it work?"

"Ah, good question. But I am only a maker, not a user."

"Take it to Tronix the wizard," someone suggested.

"Good idea; he's a computer genius!" said another.

"He's also completely nuts."

"Yes, but he's an interesting nut. Even if he can't help, you'll have a good time."

"Will he even *want* to help me?" asked Alice.

"Challenge him. There's nothing he likes better than being told he can't do something."

And so it was decided. Alice was to take her little computer to the laboratory of Tronix the wizard.

"But how do I find him?" she asked.

"That's easy. Take the road less traveled. Just read the warning signs; and do the opposite."

That sounded like dubious advice; but Alice bid her farewells. Gathering up her little board and tools, she set off on her next journey.

As she walked, the strange landscape shifted as before. The snowy settlement faded, and the weather became warmer. After a while, a path branched off from the main road with a sign: "Less Traveled Road".

"This must be it," she thought.

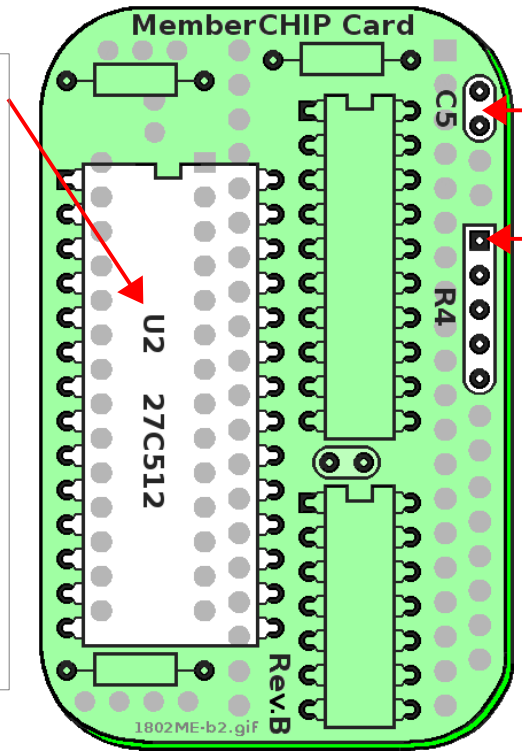
Step-by-Step Assembly (page 3)

( ) Now turn the card back over, so the side with the "MemberCHIP Card" printing is at the top.

START ▼

( ) U2: Integrated circuit. (marked "MC21ANSA") **DO NOT** solder it in!

1. Plug U2 into the IC socket strips (installed on page 7).
2. Be sure the notch (pin 1 (end) is on top, to match the marking on the board.
3. Be sure **all** the IC pins are centered in the socket holes. Then press **hard** so the entire thin part of each pin goes into its socket. hole.



CONTINUE ▼

( ) C5: 0.1uF capacitor. (yellow, marked "104")

( ) R4: 10K x 4 SIP resistor. (yellow, "5X-1-103LF") Put the striped end in the **top** hole, so the label is on the **left** side.

END!

Powering it up

That's it; you're done! Now I'll bet you're eager to see it work. Here are the "Quickstart" instructions:

1. Get the Sparkfun FTDI 5v USB-TTL serial cable # 9718 at www.sparkfun.com/products/9718. Plug one end into your computer's USB port. Plug the other end into P1 on the MemberCHIP card with pin 1 (the black wire) in the corner by the "P1" label. (You can use other adapters if you can figure out its driver and pinouts.)
2. Download a Terminal program (TeraTerm, RealTerm, HyperTerm, etc.) TeraTerm is perhaps the simplest, so I'll describe how to use it. You can download it from tera-term.en.lo4d.com/windows.
3. Start TeraTerm. In the main menu, click Setup... Serial Port... Select Port=COM1 (or whatever USB port your computer assigns to it). Set Speed=4800, Data=8 bit, Parity=none, Stop bits=1, Flow control=none. Also set Transmit Delay to 10 msec/char and 250 msec/line. If you notice missing characters, try a larger Transmit delay. The other default settings should be correct. Click New Setting to save your setup (so you don't need to do all this every time you start it). On other terminal programs, you may also have to set Full duplex, set the <ENTER> key to send an ASCII <CR> (not <LF> or <CR><LF>), and emulation to VT100 and ANSI.
4. When your Terminal program is "off-line" or "disconnected" (Alt-I in TeraTerm), it resets the 1802. When you go "on-line" or "connect" (Alt-N in TeraTerm), the 1802 starts running and lights the red LED.
5. Now press the <ENTER> key to set the 1802 baud rate. You should see the sign-on message! The red Q LED will blink when the 1802 sends data, and the green EF3 LED blinks when the 1802 receives data.
6. Commands must use CAPITAL letters; so turn your keyboard's Caps Lock LED on. Instructions for using the programs in the MC21ANSA EPROM are at www.sunrise-ev.com/1802.htm. Or see the video demonstration at www.youtube.com/watch?v=99EN6f2vBvU.

The path led into a deep, dark forest. A sign read "Be Scared!" Indeed, the trees seemed to scowl down at her, and their branches reached out like skeletal fingers. The more she looked, the worse they appeared.

"Wait a minute," thought Alice. "That sign... it made me *think* it's scary. The elves said to do the opposite. Hmph! I'll bet that sign is a trick! Think positive," she told herself. "A positive something is better than a negative nothing." She marched confidently ahead, through the now-pleasant woods.

She came to a crossroads with a sign that said "Keep Left" but with an arrow pointing right. Hmm. She ignored them both, and continued straight ahead. Someone wanted to keep her away. Well, just let 'em try!

This path ended in a clearing, dominated by a gigantic tree with the biggest tree-house Alice had ever seen. Yet another sign read "This is Not the Secret Laboratory of Tronix the Wizard." She wasn't fooled.

"Hello," she called. "Is anybody here?"

A door opened to reveal a tall shadowy figure, outlined by the blinding light from the doorway. "WHO DARES TO DISTURB THE MIGHTY WIZARD TRONIX?" boomed an amplified voice.

"Are you the wizard?"

"Of course I am!" he said, stepping away from the doorway. "Don't you see my hat?" What Alice saw was someone that was a head shorter than she was, but wearing a tall pointy hat with glowing letters. His dirty white lab coat was so big that it hung below his knees.

Alice smiled. "It says 'lizard'."

"What?" He jerked his hat off, and looked. "Oh, razzle fratzin..." He fiddled around inside it, and the LEDs flickered and blinked. At last, the rest of the "W" lit up, and he put it back on. "See? It says 'Wizard'!" he shouted, jumping up and down. "I could destroy you with a word!"

"Sticks and stones may break my bones, but words can never hurt me," Alice teased back.

"Oh, yeah? **Snowball**!" he yelled. One appeared right out of thin air. He grabbed it and threw it at her.

It caught her right in the chest. So that's his game. She answered right back with "**Water balloon**!"

"**Tomato**!" he fired back.

She dodged, and shot back with "**Mud pie**!" and scored a direct hit.

Surprised, he decided to attack with his most powerful spell. "**Whoopie cushion**!"

Oh, oh. Alice felt the gigglies coming on. She had to think fast. "**Feathers, feathers, feathers**!"

"No! Not that! I'm t-t-ticklish!" He collapsed in uncontrollable laughter. "Hahahaheehee..."

But she was giggling now, too. For a while, neither of them could stop, until they were all laughed out.

"Truce?" he finally said, catching his breath.

"Truce," she agreed. "By the way, I'm Alice."

"I'm Alec," he replied. "So what brings you here?"

"'Alec Tronix'? Oh, dear..." thought Alice, suppressing another giggle. But she said, "I heard you're a computer expert. What do you make of this?" She showed him her circuit board.

"Ooh, it's tiny! Let's see; 1802 CPU, memory, I/O... It *is* a real computer! Where did you get it?"

"I built it."

"*You* built it? Then he noticed her hat. "You're from Heath Kitt's College of Soldering Knowledge."

"Right! But you probably can't make it work," she smiled. "You don't know anything about this one."

"Nonsense! I can bend *any* computer to my will! Follow me." He turned and went inside.

Alice climbed the little ladder and entered the tree-house. Inside was the biggest conglomeration of computers she had ever seen; all different types. They also looked rather battered, and many were in pieces.

"Where did you get all this?" she asked.

"Oh, here and there. It's amazing what some people throw away. I fixed 'em all, too. Mostly."

He examined the little board closely. "It has a TTL serial port. Where's my USB-to-serial adapter? I got one on eBay for $0.99 (with free shipping). Let's see if it works."

"The instructions say to use the Sparkfun FTDI one," Alice volunteered.

"That's FARKLE," he scoffed. He caught her puzzled look, and added smugly, "It means 'Fancy Accessory, Really Kool, Likely Expensive'. Us computer wizards know *all* the acronyms."

MC21 Monitor Operation

Let's see what this Itty Bitty computer can do! Start your Terminal program, press the <ENTER> key after a reset or power-up, and you should see the Monitor's sign-on message:

```
MemberCHIP Card Monitor v2.0AR ANSI 11 July 2023. Enter "H" for Help.
>                                        ← The ">" means the monitor is waiting for your input.
```

The Monitor's most important job is to load and run programs (like BASIC or Adventureland). But it can also "look under the hood" of the computer, so you can see how it works from the inside. It can view and change the CPU's registers, read and write to memory, move blocks of memory around, disassemble programs so you can see how they work, and load and save them on an external computer with your Terminal program.

The Monitor expects CAPITAL letters, so press the Caps Lock key and try the H (Help) command:

```
>H                                       ← Type "H" for Help.
Commands            Description
--------            -----------
H                   Help
B                   BASIC level 3 v1.1
P                   Play Adventureland
L                   Load program or data (Intel HEX format
V                   View 1802 registers
Daaaa bbbb<CR>      Disassemble Opcodes from aaaa to bbbb
Maaaa bbbb<CR>      Memory read from aaaa for bbbb bytes
Waaaa dd dd..<CR>   Write to memory until <CR>
Saaaa bbbb<CR>      Save memory at aaaa for bbbb bytes (Intel HEX format)
Taaaa bbbb cccc<CR> Transfer (copy) memory from aaaa to bbbb for cccc bytes
Eaaaa bbbb cccc<CR> EPROM memory transfer aaaa to RAM bbbb for cccc bytes
Raaaa<CR>           Run program with R0=aaaa P=0 X=0 Q=1
All commands are UPPERCASE, All numbers are HEX, <ESC> aborts command.
>
```

Single-letter commands are executed immediately. For example, `H` for Help, `B` to start BASIC, or `P` to Play Adventureland.

Commands with numbers must be in hexadecimal (`0-9, A-F`). Addresses (`aaaa`) are 4 digits, and data bytes (`dd`) are 2 digits. Leading zeroes can be left off. The Monitor doesn't use Backspace or Delete; but it only uses the rightmost 2 or 4 digits. For example, `W18000 1FF` ignores the leading 1's and treats it as `W8000 FF` (write FFh to address 8000h). End these commands with <CR> (your keyboard's ENTER key).

The ESCAPE <ESC> key will abort (stop) a command in progress. For example, <ESC> can stop a long `M0 1000` command (read 1000h bytes of memory starting at address 0000h). <ESC> is only checked occasionally, so you may have to hit it more than once. When used, <ESC> displays a `Function Aborted` message. <ESC> is also not checked during the SAVE command.

MemberCHIP Memory Map

0-32K	0-7FFFh	ROM (Read-Only Memory). You can read, but cannot write to these addresses. It contains the Monitor, BASIC, Adventureland, and other built-in programs.
32-64K	8000-FFFFh	RAM (Random Access Memory). You can read and write to these addresses.
	FFC0-FFDAh	Reserved for Monitor internal variables; do not write to these addresses.
	FFDB-FFFFh	Storage for Registers at last power-up, Clear, or hardware or software interrupt. The View command will display them, and the W command can alter them.

Additional details are on page 19 – MC21 Monitor Commands.

He dug through the clutter on his desk. "Found it!" he announced, holding up a thing the size of a stick of gum.

"This is a USB-serial adapter. The metal end plugs into the USB port on my computer. The RX and TX pins will connect to your board's serial port."

"That's a computer?" scoffed Alice, looking at the pile of junk strewn across the table. "Does it work?

"Of course! ***I*** built it! Now watch how easy it is. I'll test it by shorting RX and TX with this little red jumper." He poked the adapter into the tangle of parts on his desk. A message popped up on the screen. "'Unknown device?' The manual should... oh, there isn't one. No manufacturer name on it either; but there *is* a website." He typed it in...

"Rats! It's a dead link. I'll have to find the driver myself. The IC on it is marked SIL2104. Let's see what Google says." That led him down the rabbit hole. Muttering to himself, he searched page after page in vain.

Alice soon tired of his chaotic style of working. Looking around, she saw a large, old-fashioned book that looked out of place among all the computer junk. It had an ominous title:

DO NOT READ THIS BOOK! – That Means YOU, Stupid!

She decided this was another of his little scare tactics. But the book was held shut by a latch and a series of gears. One gear had a notch. If she could rotate the notch over the latch, it should release. It took a while to figure out how to get it in position; but at last, the latch flipped up with a "click". She opened the book.

A sleeping dragon awoke with a snort, and stared directly at her. It inhaled deeply, and with a loud roar, breathed fire in her face! She shrieked, and leaped back quickly.

A burst of hysterical laughter rose from across the room. "I *love* that trick! It gets 'em every time!"

"That's not funny!"

"Oh yes it is! You should have seen the look on your face!" he howled.

Alice looked closer. The old book had been hollowed out, and a laptop computer put inside. He'd set it up to display a video clip of the dragon when it was opened. She had to admit, it was a neat trick.

Then she had a thought. "Hey; this is a working laptop. Maybe it can talk to my itty bitty computer."

But how? Well, Grandpa always said, "When all else fails, read the @#$^%& manual." So she got out Heath's notes, and sure enough, page 11 suggested the Sparkfun FTDI 5v USB-serial cable. She went to the link listed, and there was a picture of it. Maybe there's one here already?

Alec was still busy with his experimenting and muttering, so she decided to look around herself. In a bushel basket full of random cables, she found one that looked like the one in the picture. It was worth a try.

She plugged the USB end into the laptop. A message popped up:

Installing Device Driver Software – Click **here** for details	...so she clicked it,
Searching Windows Update for driver	...then waited a bit,
USB Serial Converter **√** ready to use	...and sure enough,
USB Serial Port (COM3) **√** ready to use	...it was installed!

That looked promising! Now, where does the other end go? Alec had said something like "Shoot Rex and Tex"; but that wasn't helpful. She decided to go with the manual again. Aha! Page 11 step 1 said to plug it into "P1" on her little board, with the black wire in the corner.

Let's see... Step 2 is "Download a Terminal program." How do I do that? What's Google say if I search for "download teraterm"? Sure enough, the first link was to the same URL address shown in the notes. She clicked on it, and got a page with a big green "download" button. She clicked it.

That opened a new page. In the middle, there were a bunch of versions; US, UK, Japan... Which one to pick? She clicked the one that matched her country, and it started downloading automatically.

ADVENTURELAND Operation

OK, let's have some fun! ADVENTURELAND is a computer game that was written in BASIC way back in 1978 by Scott Adams. It's a great example of how much you can do with limited resources.

Configure your Terminal program for ANSI mode (TeraTerm is pre-configured for it). "ANSI mode" means your Terminal program will recognize and obey ANSI command sequences to control the position and color of text and graphic characters. You can also use these commands in your own programs. See https://en.wikipedia.org/wiki/ANSI_escape_code for a table of ANSI commands.

Power-up or Reset the MemberCHIP card, start your Terminal program, and hit the <ENTER> key to start the Monitor. Type P to start Adventureland (remember, all commands must be upper case). You should see the following message:

```
MemberCHIP Card Monitor v2.0AR ANSI 11 July 2023. Enter "H" for Help.
>P                                  ← Type "P" to start the Adventureland game.
```

The Adventureland game should display its sign-on screen. Here's what it looks like in glorious 1980's character graphics. Note that ANSI emulation is buggy in many Terminal programs; this may cause a few glitches (like some extraneous "?" marks in TeraTerm). The sign-on screen also uses 25 lines; expand your Terminal window to 25 if it defaults to 24 lines.

Like it says, "Press any key to continue". It will respond with:

```
Decompressing...
Welcome to Adventure+ (v1.2 2020-04-20)
By: Scott Adams, Morten Lohre, and Richard Goedeken
Load saved game (Y or N) N          ← Type N (because you initially have no saved game).
```

The game starts, and the adventure begins! Type one- or two-word commands, followed by the <ENTER> key. It's wise to make a map, so you don't get lost. Use QUIT to exit. Good luck!

When Tera-Term finished downloading, Alice moved on to step 3 on page 11. Clicking the new TeraTerm icon displayed a black window, with menu items at the top. Though the terms were strange (what the heck is a "baud"?), she was able to configure it and save the setup as described.

Let's see... Step 4 said to "connect" to her itty bitty computer. Wasn't it already connected? The red LED was lit. To be sure, she tried Alt-I, and the top line in TeraTerm said "disconnected". So she typed Alt-N to "connect". Sure enough, the "disconnected" message disappeared. It must be "connected".

Step 5 said to press the <ENTER> key; so she did. BINGO! The message shown on page 13 appeared on the screen. Alice couldn't resist a little whoop for joy.

Alec turned to her suddenly. "What did you do? Did you break my laptop?"

"No. I got it working! See?"

"Ooh, it's a sign-on menu." He started typing random commands. "Hey, nothing works. It just keeps saying `Function Aborted!`"

Alice gave him a derisive look, and pressed the Caps Lock key. "You should read the manual."

"Hah; I was just checking to see if *you* did."

"If you're so smart, make it do something."

"OK, I will. Let's see... Hey, it's got BASIC!"

"What's BASIC?"

"It's a programming language. BASIC stands for Badly Arranged Source of Infinite Confusion... or something like that." He pressed the `B` key (uppercase this time), and was rewarded with the BASIC sign-on message. He then pressed `C` to cold-start BASIC, which displayed `READY` and then its `:` prompt (which it displays at the start of each line when it is waiting for someone to type something).

"BASIC is like a fancy calculator. You type a command, like `PRINT 2+2` and it prints the answer `4`. It can also do fancy stuff, like `PRINT "The square root of 2 is "; SQR(2)`. Try it!"

Alice played with it a bit. Anything she put in quotes was printed exactly as-is. Math equations could be quite complex, and include parentheses and most functions on a calculator. If she mis-typed something, the BACKSPACE key could correct it; otherwise BASIC would respond with `ERR CODE` and some number.

"Interesting. But why would I use this instead of a calculator?"

"Because it's a *programmable* calculator. You can give it a list of things to do, and it will remember them, so you can do them again later. Suppose you want to convert temperatures from deg.F to deg.C..." Here's what he typed (in BOLD):

```
:10 INPUT "Temperature in deg.F" T          ← This line gets a number "T" in deg.F.
:20 PRINT T;" deg.F is ";(T-32)*5/9;" deg.C" ← This line converts it to deg.C,
:30 END                                     ← and this line ends the program.
```

"This is a *program*; a list of things to do. The numbers at the left tell BASIC what order to do them in. I could have numbered the steps 1,2,3; but us <u>real</u> experts go by 10s so we can add a step later if we have to. If a line has no number, BASIC does it immediately. If there is a line number, BASIC inserts it (in order) into the program. If you type just a number, BASIC deletes that line from the program. Now type `RUN`."

Alice did so (followed by the <ENTER> key). Here is what she got (what she typed is in **BOLD**):

```
:RUN                                        ← Remember, the ":" means BASIC is waiting for you.
Temperature in deg.F?32                     ← The "?" means the INPUT statement is waiting for you.
32 deg.F is 0 deg.C                         ← BASIC does the conversion, and prints the result.
READY                                       ← Then it announces that the program has ended,
:                                           ← and it is waiting for your next command.
```

Alice was beginning to see how this "programming" stuff could be useful. She studied the program a bit to figure out how it worked. `T` was apparently a variable, like she learned in math class; it was a place-holder for a number that could be filled in or calculated later. When she had entered `32`, program line 10 had set T=32. Then line 20 printed the value of T, a message, converted T to deg.C, and printed the result.

BASIC Operation

Let's try BASIC; it's one of the oldest and simplest computer languages. The full manual is at www.sunrise-ev.com/MembershipCard/BASIC3v11user.pdf. You type the things in **BOLD**:

```
>B                                  ← From the Monitor, type "B" to start BASIC.

1802 BASIC3 v1.1 (C)1981 RCA
C=cold, W=warm start                ← Type "C" to do a COLD start. (Or "W" for a WARM
C                                     start when there's already a program in memory.)
 READY                              ← BASIC is ready. Type each line of your program...
:10 X=10                            ←   Set X equal to 10
:20 PRINT X;" ";                    ←   Print the value of X, then a space " "
:30 X=X-1                           ←   Subtract 1 from X
:40 IF X>0 GOTO 20                  ←   If X is greater than zero, jump back to step 20
:50 PRINT "LIFT OFF!"               ←   Otherwise, X=0, so print "LIFT OFF!"
:60 END                             ←   End of the program.
:RUN                                ← This will RUN your program,
10 9 8 7 6 5 4 3 2 1 LIFT OFF!      ← ...and here is its output!

READY                               ← BASIC is ready for more.
:BYE                                ← BYE exits BASIC, and returns to the monitor.
```

The `LIST` command will list (print out) your program. `LIST` prints it all. `LIST line#` prints just the selected line. `LIST start#,end#` will list all lines in the selected range.

You can save your program on your computer's hard drive, and load it later using TeraTerm.

1. Type `LIST` but do not press the <ENTER> key to start it.
2. In TeraTerm's menu, click File... Log...
3. Fill in the desired Save in: directory and File name: boxes. Use a directory that you will remember, and a file name that explains what it is (like "LIFTOFF.BAS").
4. Under Options, check only the Plain Text box.
5. Click Save.
6. Now press the <ENTER> key. BASIC will list your program, and TeraTerm will "log" it to disk.
7. In TeraTerm's menu, click File... Stop Logging (Q).

To load a saved program into BASIC from TeraTerm:

1. Type `NEW` (and the <ENTER> key) to clear any old program from BASIC's memory.
2. In TeraTerm's menu, click File... Send file...
3. Select the directory and name of the file you previously saved, and click Open.
4. TeraTerm will send the file, and BASIC will load it into memory as if you typed it yourself. Ignore the ending `ERR CODE 32` message; TeraTerm saved the `READY` at the end of the LIST command, and BASIC is just complaining that `READY` is not one of its commands.

Files saved this way are normal text files, and can be created or edited with any text editor. Don't use a Word Processor, unless you set it to save in plain ASCII text format; otherwise, the file will be bloated with unnecessary formatting codes that BASIC won't understand.

Plain text file formats also let you find, download, and run thousands of BASIC programs from books, magazines, the web, etc. Note that since BASIC comes from the "wild west" days of programming, there were no standards, and most BASIC programs need to be edited to suit the particulars of each version of BASIC.

"How do I write a program?" asked Alice.

"You don't," snorted Alec. "Programming is hard. You could never write a good program."

That sounded like a challenge. Alice was getting tired of his know-it-all attitude, and so she blurted out, "Oh yeah? I'll bet I can write a program that would amaze even you."

"A bet?" Alec's eyes lit up. "You're on! I'll bet... ah... this laptop that you can't! And if you lose, I get your little computer!" he said with a sly grin. "Deal?"

"Deal!" she said, perhaps a bit too quickly.

"Let me know when you give up," Alec hooted as he walked away. "You'll have a hard time amazing me!"

Alice began to worry. What would it take to impress that smart-Alec? Not a game; they'd become far too complicated. Even something like a text editor was way beyond her abilities. No; it would have to be a trick; like pulling a rabbit out of a hat. It's easy if you know how it's done. But if you *don't*, it looks impossible!

She thought about the tricks she'd seen. Most used special props like marked cards or sleight of hand; they wouldn't work on a computer. Maybe a mind reader or fortune teller? She'd seen some good ones; but she didn't know how they did it, either. Oh dear; she was really behind the eight-ball, as grandpa used to say.

Wait a minute... There *was* something she'd seen that was utterly amazing, yet incredibly simple once you knew how it worked. So a program is a list of instructions? Well, she was beginning to understand how a computer with such a list could produce the desired effect. She started reading the online BASIC manual at www.sunrise-ev.com/MembershipCard/BASIC3v11user.pdf to find the commands she'd need.

She typed in each line and tested it, one at a time. It took a few tries to learn where punctuation was (and was not) needed. Once each command did what she wanted, she put a number in front of it so it became part of her program. Then she typed `RUN` to test it. It grew line by line, until at last she was satisfied.

Saying a silent little "Muahahaha" to herself, she announced, "OK Alec; let's see what you think of this!"

Alec wandered over, but with a look of curiosity in his eye. Alice cleared the screen, and typed `RUN`.

```
Welcome. I am the sage of the age: the eyes of the wise:    ← "Now what?"
and the voice in the noise. What is your YES/NO question?   ← "Ask it a question."
?WHAT IS THE AIRSPEED VELOCITY OF AN UNLADEN SWALLOW?       ← "OK." He typed...
Alas, that is not a YES/NO question.                        ← Oops...
?WAS THAT QUESTION FROM A MONTY PYTHON MOVIE                ← So he tried again...
It is decidedly so.                                         ← (Hmm... it was right!)
?WILL IT RAIN TOMORROW                                      ← (I'll try this...)
My sources say no.                                          ← (Maybe she's using Google?)
?WILL MY PARENTS FIND OUT ABOUT YOU-KNOW-WHAT               ← (Well, Google can't answer this!)
Most likely.                                                ← (Argh; that's bad news. But is it true?)
?ARE MY SOCKS RED                                           ← (I'll ask one I know the answer to...)
Concentrate, and ask again.                                 ← "Aha! It doesn't know! They're red!"
                                                            ← He lifted his cuffs to reveal red socks
                                                            ← with green Grinches on them.
?ARE ALEC'S SOCKS RED AND GREEN?                            ← Alice grinned, and typed...
Definitely yes.                                             ← "What? How did it... You couldn't
                                                            ← possibly know..." Alec sputtered.
?WAS THAT A LUCKY GUESS                                     ← (Was it a mind reader? So he asked...)
Don't count on it.                                          ← "That's amaz... I mean, uh, not really
                                                            ← amazing, just kind of surprising."
?IS ALEC AMAZED?                                            ← Alice reached over, and typed...
Absolutely.                                                 ← "You lose!" she announced.
```

"OK, you win; so I *am* amazed. Big deal; I got that laptop for nothing. The sound doesn't work, the F8 key is broken, and the battery's shot; I just stuck another one on the bottom. But what's the trick? How could *you* possibly write a program that understood what I said? There's *no way* you could get AI into that thing!"

Alice hit <ENTER> to end the program, typed `LIST`, and gave him her best evil grin.

MC21 Monitor Commands

Help message – a summary of the commands and formats used for them.

BASIC – Run RCA floating-point BASIC3. ("BYE" exits BASIC back to the Monitor).

Play the Adventureland game. ("QUIT" exits the game back to the Monitor).

Memory read. `M1000 100` reads 100h (256 decimal) bytes from 1000h to 10FFh. Each line displays the address, 16 bytes of data, and continues on a new line until done.

Write memory. `W8000 11 22 33` will write 8000h=11h, 8001h=22h, and 8002h=33h. The W command can only write to RAM (8000-FFFFh).

Transfer (copy) a block of memory. `T1000 8000 100` copies 100h bytes from address 1000h to 8000h. If the end=start+1, Transfer will fill the block of memory with the value of the first data byte.

View registers (saved in RAM at power-up, clear, or hardware or software interrupt). Saved registers can be changed with a Write command; thus `WFFF4 12 48` sets RC=1248h.

```
>V                                      ← View registers command.
CDP 1802 Register Contents
R0 = 0BF3   R1 = 0B5E   R2 = FFC4   R3 = 0BA0
R4 = 0BB1   R5 = 0AED   R6 = 81A2   R7 = 0C71
R8 = 0FDB   R9 = FFFF   RA = 8100   RB = 000D
RC = 1F00   RD = 38BF   RE = 0119   RF = 0D17
PC = 2      X  = A      T  = A2     D  = F8
DF = 0
```

RAM address where each is stored:

FFDC	FFDE	FFE0	FFE2
FFE4	FFE6	FFE8	FFEA
FFEC	FFEE	FFF0	FFF2
FFF4	FFF6	FFF8	FFFA
FFFC	FFFD	FFFE	FFFF
FFDB			

Run program at address. `R8000` runs program with P=0, X=0, and Q=1. R1 is set to re-enter Monitor if 1802 executes a SEP 1=D1h breakpoint (software interrupt), or if interrupts are enabled and 1802 /INT pin goes low (hardware interrupt). Other registers are set to the values shown by the View command.

`R1680` runs a program to set the registers to the Monitor's subroutines. It sets X=2, R2=FFBFh (the stack), R4=0ABDh (standard CALL), R5=0AEDh (standard RETURN), P=3, and R3=8000h to run your program.

Save a block of memory (program or data) to your Terminal in Intel HEX format. For example:

```
>S0 20                                  ← Send 20h bytes, starting at 0000h. The monitor waits
Ready to save program:                    ~20 Seconds for you to open a Log file. In TeraTerm, use
(Intel hex data appears here...)          File... Log..., select the Save in: directory and filename.hex,
                                          and Save. When the data ends, close the file. In TeraTerm,
File Saved Successfully                 ← use File... Stop Logging (Q). Press <CR> to finish saving.
```

Or, open your Log file first, and *then* type the Save command. The HEX file will have the "Ready to save program:" message in it; edit it out with any text editor. The saved Intel HEX file should look like this:

```
:20000000071000C00B009EFF81325C9EFF80C200CF9E32819D52228D5222F800BDABF808ADDA
:00002001DF
```

Load a block of memory (program or data) from your Terminal in Intel HEX format. This could be from a previous Save command, or created by an 1802 assembler. For example:

```
>L                                      ← The L command waits for a file. Tell your Terminal to
Ready to LOAD program                     send the HEX file. In TeraTerm, use File... Send file...,
File Loaded Successfully                  select the directory and filename, and click Open.
```

You can only Load RAM at 8000-FFFFh (the EPROM is read-only). Load can be aborted with the <ESC> key. Intel HEX files provide the load address; but you can use Transfer to move it somewhere else.

EPROM loads a block of memory (program or data) from high EPROM into RAM. For example, `E8000 8000 6000` loads the Super Star Trek game from high EPROM into RAM. Run it with `BASIC... Warm start... RUN`.

Disassemble hex opcodes into 1802 Assembler mnemonics, which are easier to read.

```
>DB00 B03                              ← Disassemble opcodes from 0B00h to 0B03h.
0B00 F800  LDI  00                       Each line shows the address, opcode, and mnemonic; and
0B02 BE    PHI  RE                       continues to the ending address, or <ESC> key is pressed.
>DB04 0                                ← If the end address is 0 or smaller than the start address,
0B04 3611  B3   11                       the disassembler enters single-step mode:
0B06 F801  LDI  01                       Press <ENTER> for another line, or <ESC> to exit.
```

"I have a toy called a 'Magic 8-ball'," Alice explained. "You ask it a question, and it answers. It doesn't really understand your question; but the answer makes it *seem* like it does. Your imagination does the rest." Here is a listing of Alice's program:

```
1 REM Magic 8-ball program
10 PRINT "Welcome. I am the sage of the age: the eyes of the wise;"
20 PRINT "and the voice in the noise. What is your YES/NO question?": PRINT
30 FOR X=1 TO 20
40 INPUT A$: IF LEN(A$)=0 THEN EXIT 90: REM exit if input is a blank line
50 REM GOSUB RND(20)*10+100: REM use this line for answers in random order
60 GOSUB (X*10)+90: REM use this line for answers in order
70 PRINT
80 NEXT X
90 PRINT "Farewell, until we meet again...": END
100 PRINT "Alas, that is not a YES/NO question.": RETURN
110 PRINT "It is decidedly so.": RETURN
120 PRINT "My sources say no.": RETURN
130 PRINT "Most likely.": RETURN
140 PRINT "Concentrate, and ask again.": RETURN
150 PRINT "Definitely yes.": RETURN
160 PRINT "Don't count on it.": RETURN
170 PRINT "Absolutely.": RETURN
180 PRINT "Reply hazy; try again.": RETURN
190 PRINT "It is certain.": RETURN
200 PRINT "My reply must be no.": RETURN
210 PRINT "Without a doubt.": RETURN
220 PRINT "Ask again later.": RETURN
230 PRINT "You may rely on it.": RETURN
240 PRINT "The outlook is not good.": RETURN
250 PRINT "As I see it, yes.": RETURN
260 PRINT "I cannot tell you now.": RETURN
270 PRINT "Outlook good.": RETURN
280 PRINT "It is very doubtful.": RETURN
290 PRINT "Signs point to yes.": RETURN
```

Alec stared at the program intently. "I see what you're doing. I thought it was some kind of artificial intelligence program. But it doesn't pay attention to the questions at all; it just gives random answers. Wait... you put a REM in front of line 50 so it's ignored. And line 60 makes it give the answers *in order*. That means you *knew* what answer it would give next! You tricked me!"

"No; you tricked yourself. I made the sequence of answers to be 'Maybe... Yes... No... Yes...' and repeat. It goes to show that even a simple program can do something amazing. It's 'artificial intelligence' all right – artificial as in 'fake'. The intelligence comes from the programmer; not from the program or computer."

"Send me that program!" Alec demanded. "I'm going to trick someone else with it. Aha! I think I have a fishbowl around here somewhere. I'll put the screen inside it, so it looks like a crystal ball. Oh, I'm going to have so much fun with this. Bwoo-ha-ha-hah!" He turned, and immediately began rummaging through his piles of junk.

The elves were right. Tronix the Wizard was completely nuts. But he was also an interesting nut. Alice decided this would be a good time to leave. She said her goodbyes, and closed the book with the laptop in it. Conveniently, there was even room inside for her little computer and cable.

She left the treehouse, and retraced her steps to the crossroad. The signs that Alec had erected to scare away visitors had equally unwelcoming messages on the backs. The sequence of signs read "Now get lost! / Go away! / Come back when / You can't stay!" She shook her head and thought, "Boys are like, *so* stupid!" At the intersection, she turned down the main road. In the distance was something that looked like a city.

I program my own computer.
Beam myself into the future.

Banked Memory

The 1802 can only directly address 64k of memory. The MemberCHIP Card has 32k of ROM at 0-7FFFh, and 32k of RAM from 8000-FFFFh. But the rev.B MemberCHIP Card uses a clever trick to expand this to read a larger 64k EPROM. This provides an additional 32k of memory for programs or data. There is a description of this technique at <https://hackaday.io/project/190289-1802-memberchip-card>.

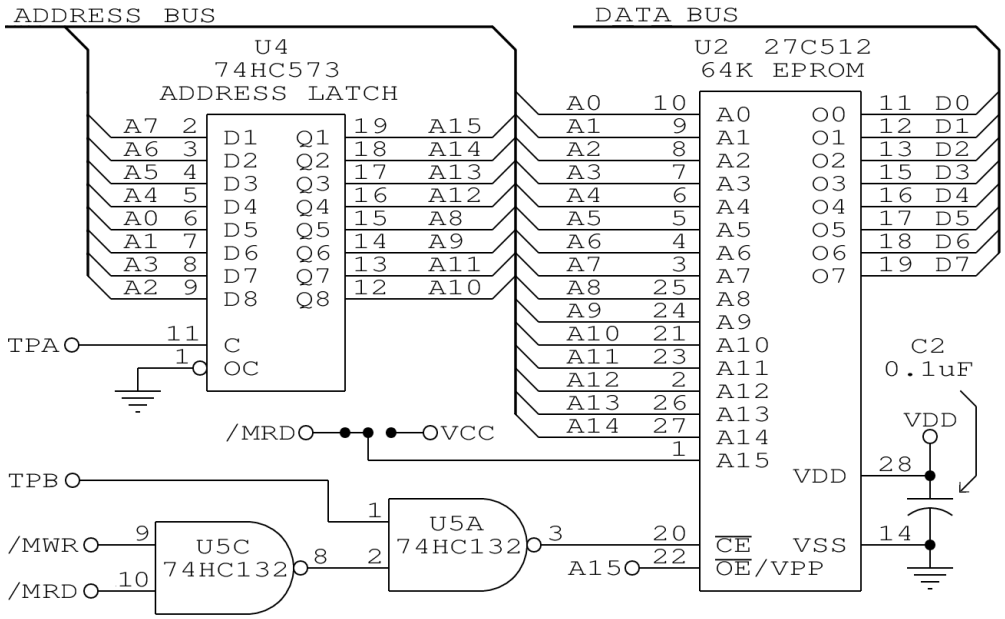
The upper 32k of the EPROM is I/O-mapped. To read a byte from it (for example, from 9000h):

1. Set a register to the desired address, but with its most significant bit=0.
   (i.e. to read 9000h, set the register to 1000h).

2. Set X to this register (i.e. with a SEX instruction). It can be the stack if X is already the stack pointer.

3. Execute an INP 0 instruction (opcode 68h). This is normally an "undefined" instruction, as it does an input without specifying an input port# on the N0-N2 lines. The 1802 will still do a write cycle, which will chip-select the EPROM, and read a byte from the upper half of the EPROM into the 1802's D register.

The MC21ANSA monitor includes an "E" command to block move bytes from high EPROM into RAM. The format is "E (source) (destination) (#bytes)" where each parameter is a hex number. (source) is the starting address in EPROM, (destination) is the destination in RAM, and (#bytes) is the number of bytes to move.

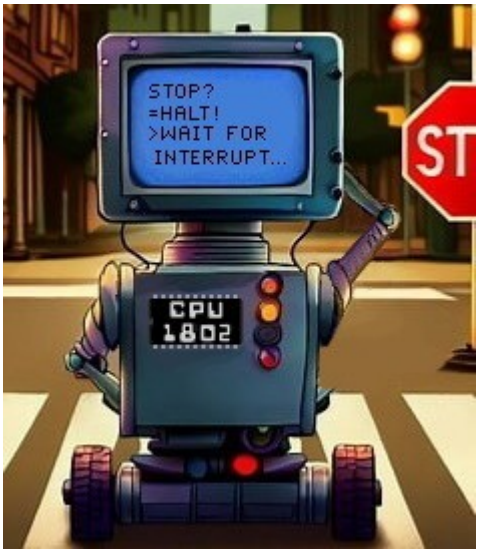


How does it work? The rev.B EPROM is wired as shown above. It is chip-selected when TPB=1 (i.e. the last cycle of a bus access) and when /MRD=0 or /MWR=0 (i.e. either a read or write cycle). A15 connects to the EPROM's /OE pin, so its output is enabled when A15=0 (i.e. from 0-7FFFh). When the 1802 reads from 0-7FFFh (/MRD=0) it reads data from the lower half of the EPROM normally. But when the 1802 **writes** to 0-7FFFh (/MWR=0), /MRD=1 so the EPROM **reads** data from the upper half (i.e. 8000-FFFFh).

We don't want both the 1802 and EPROM to put data on the bus at the same time (bus fight)! So we use the 1802's INP instruction to do this write cycle. INP instructions set /MWR=0, but do **not** put anything on the data bus. So, the 1802 loads whatever is on the bus (i.e a byte from the EPROM) into its D register.

Thus we have a way to read any byte in the upper 32k of the EPROM. The upper ROM address is selected by the 1802 R(X) register with bit A15=0. This byte can then be used or saved in RAM normally.

The trick works for even larger EPROMs by using the 1802's N0-N2 lines to select up to eight 32k banks. But one consequence is that INP 1-7 instructions to actual input ports 1-7 must point to RAM 8000-FFFFh.



As Alice got closer, she found it was indeed a city; but a very strange one. Every building was identical, and the streets were laid out in a perfectly rectangular grid. It seemed to be deserted, but... was that a person, staring at a Stop sign?

"Pardon me for interrupting, but can you tell me where I am?"

"*<Interrupt at last!><This unit has been waiting a long time>,*" said a mechanical voice. Alice observed that he, or rather it, was in fact a robot. It was rather dusty, as if it hadn't moved in a long time.

"Waiting? Waiting for what?"

"*<Following instructions><Last instruction was Stop>*"

"But that's a stop SIGN, not a stop light. It will never change."

"*<Must always follow instructions>*"

This is silly, thought Alice. Maybe if I cover up the sign? But she didn't have anything big enough. She reached up, but could only cover the first two letters with her hands.

The robot suddenly came to life. "*<That does not compute!> <Please define OP code>*"

"It means 'Operator'," said Alice, thinking quickly. "Your next instruction is to look at me, not the sign."

The robot turned with a squeak and a rattle. "*<Thank you, Operator> <That was a long stop>*"

Alice decided that introductions were in order. "I'm Alice. What's your name?"

"*<CPU>*" it said, pointing to itself. "*<Central Postal Unit #1802><CPU delivers letters to addresses>*"

"That sounds like an important job."

The robot sagged with a metallic groan. "*<Once this unit ran cars; airplanes; spacecraft; even went to Jupiter> <Now it is too old, too slow> <No one gives it instructions>*" Then the robot gave a little squeak, and stood up straighter. "*<Operator!> <Give CPU instructions!> <This unit can still do important work!>*"

"What sort of instructions? What can you do?"

"*<See tower in center of city?> <Light in tower is called "Q"> <Give instruction to set Q on>*"

"OK. CPU, set Q on," said Alice. But CPU didn't move, and the light stayed off. "Why didn't you do it?"

"*<CPU needs short one-byte commands> <Use 0111 1101 binary, or 7B hex, or SEQ acronym>*"

That seemed pretty tedious, but Alice knew that people liked to use acronyms to make it look like they knew what they were doing. So she tried saying "SEQ".

CPU vanished so fast that the dust on it hung motionless in the air like a ghostly shadow. A puff of wind alerted Alice to the fact that CPU was now standing in a different position. She looked; and sure enough, the tower light was now brightly lit.

"You said you were slow!" Alice gasped. "I couldn't even see you move!"

"*<This unit *is* slow> <It only goes four million steps per second> <SEQ took it four microseconds>*"

"Four million steps in a second? That's really fast!"

"*<No, that is slow> <Young CPUs can do it a million times faster>*"

"Well, you're a million times faster than me! That's enough. Who needs that much speed?"

"*<Humans are not logical> <They need 'enough' but always want 'too much'>*"

Crivens looked up as if to say, "Oh; so you're back."

"Yes," said Alice. "You were right. I *can* do it myself!"

Crivens only purred.

The end... Or is it the beginning?

```
THERE:     ;once was a limerick that began
           ;an 1802 ASM program
  SEQ      ;made it slow
  REQ      ;now it's "no"
  BR THERE ;did it over again
```

MC21 Monitor Details

Serial I/O: The Monitor uses the 1802 EF3 pin for serial input, and the Q pin for serial output. The idle level for both is high, so they can be directly connected to a USB-to-serial 5v TTL adapter. The first key typed must be the ENTER key (ASCII <CR>) to auto-detect baud rates from 300-9600 (4800 is recommended).

Breakpoints: To help debug a program in RAM, temporarily replace an instruction with a SEP R1 (=D1h) instruction. When the 1802 executes the D1h, it saves the registers and jumps to the Monitor. You can then examine what the program has done so far, and make any corrections. If interrupts are enabled, momentarily pulling the 1802 /INT pin low will also generate a breakpoint.

For Breakpoints to work, R1=0B5Eh (the breakpoint interrupt handler), and X must point to a stack in RAM with at least 5 free bytes. If your program changes R1, then use an LBR 0B5E instruction (C0 0B 5E) as a breakpoint. In this case, P must not be R4 or R5.

Here is a sample Monitor session. You type what is in **BOLD**:

```
MemberCHIP Card Monitor v2.0AR ANSI 11 July 2023. Enter "H" for Help.
>W8000 F8 12 B3 F8 34 A3 F8 80 FE D1   ← Enter a test program.
>M8000 A                              ← See if it's entered correctly.
8000 F8 12 B3 F8 34 A3 F8 80 FE D1    ← Yep; it is.
>D8000 8009                           ← Let's Disassemble it to see what it does.
8000 F812   LDI  12                   ← It sets register R3 to 1234h,
8002 B3     PHI  R3                         ...high byte
8003 F834   LDI  34
8005 A3     PLO  R3                         ...low byte
8006 F880   LDI  80                   ← then sets D to 80h,
8008 FE     SHL                       ← and shifts it left, making D=0 and DF=1.
8009 D1     SEP  R1                   ← then ends with a Breakpoint to return to the Monitor.
>R8000                                ← Let's RUN it to see if it works.
Currently running your program
MemberCHIP Card Monitor v2.0AR ANSI 11 July 2023. Enter "H" for Help.
                                      ← Sure enough, the Breakpoint returned to the Monitor.
>V                                    ← View the registers to see if they are set as expected.
CDP 1802 Register Contents
R0 = 8010   R1 = 0BA8   R2 = 4000   R3 = 1234     ← R0=800Ah (right after the D1h),
R4 = 30F6   R5 = 0AED   R6 = 01D9   R7 = 0C71     ← R3=1234h (where we set it),
R8 = FFFE   R9 = FFF7   RA = 25B4   RB = FF00
RC = 7B7F   RD = 0008   RE = 0119   RF = 0C17
PC = 1      X  = 2      T  = 21     D  = 00       ← D=0 and DF=1.
DF = 1                                ← So our program works!
```

Notice that P=1 (that's what the D1h opcode set it to), X=2 (where the Monitor keeps its stack), and T=21h (the values of X and P when the Interrupt occurred).

Odds and Ends

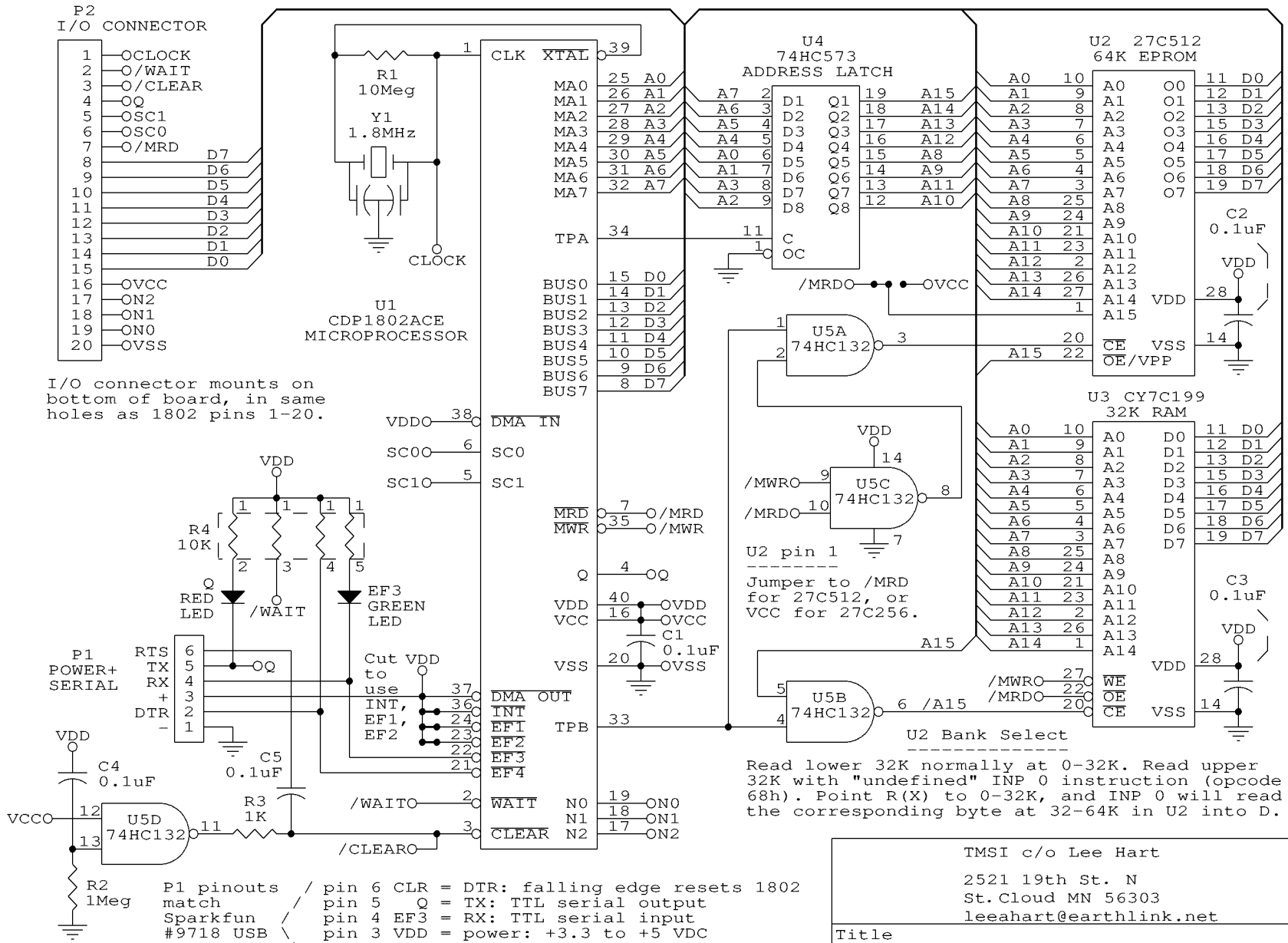
There are jumper options to use a 32k 27C256 EPROM instead of a 64k 27C512 EPROM at U2. You may want to add a jumper or switch if you need to switch between them regularly.

There are holes for the 1802 /EF1, /EF2, and /INT pins. Right now, they are jumpered (with foil) to tie them high. But you can cut these jumpers and install pins or wires to use these signals for your own purposes.

The 20-pin I/O connector may seem limited; but it's enough for external logic to use all of the 1802's I/O ports. Pins N0-N2 select ports 1-7, with /MRD=0 for OUT1-7; or with /MRD=1 for INP1-7. Note that M(R(X)) must point to RAN (at 8000-FFFFh) or your I/O will come from banks in U2.

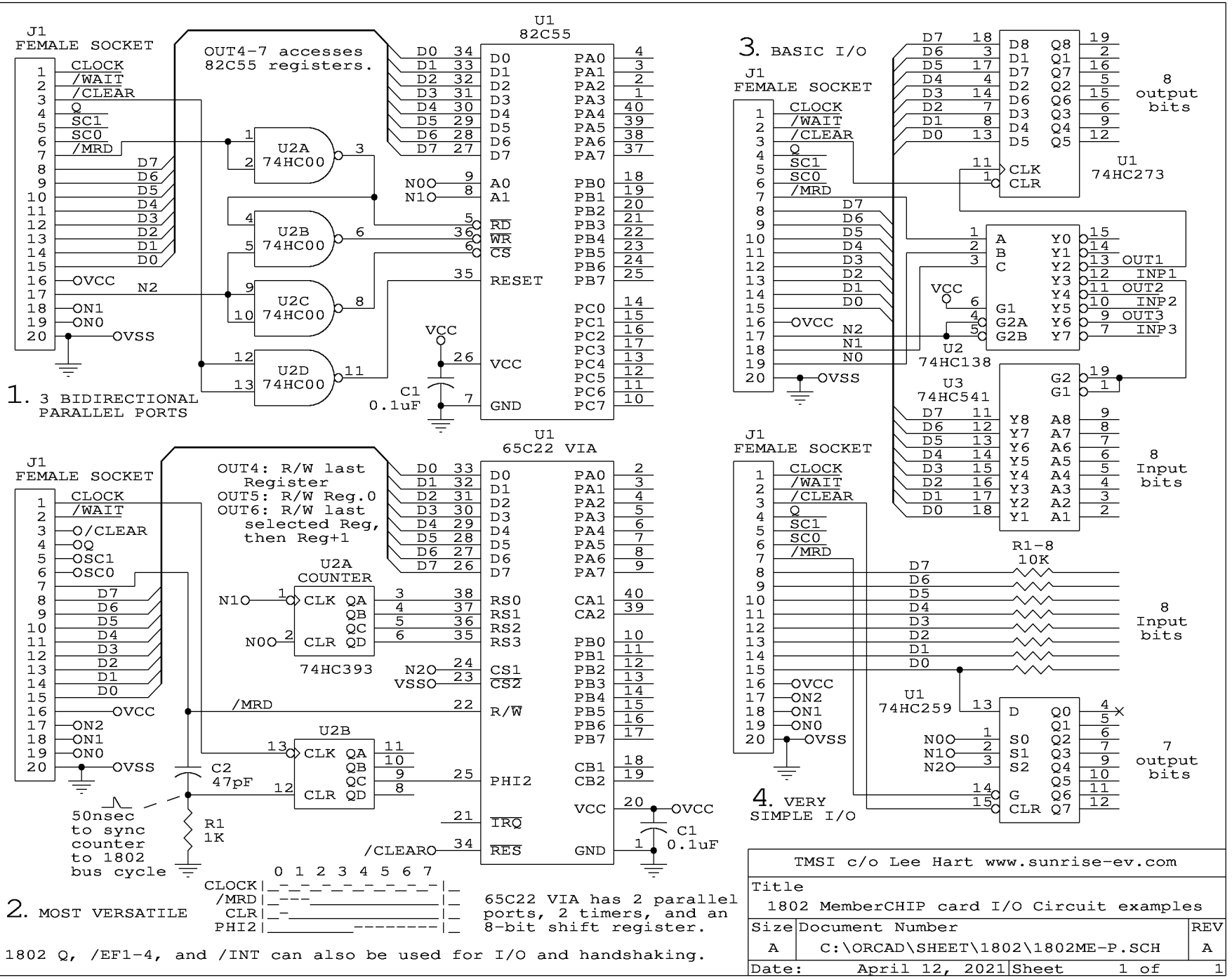
The last page of the manual has some ideas for expansion boards. These can be built on a “shield” board, or on a solderless breadboard, and the MemberCHIP Card plugged into it with its 20-pin header.

That's it; you're done. Now go, and turn an oddity into a commodity!



TMSI c/o Lee Hart
2521 19th St. N
St.Cloud MN 56303
leeahart@earthlink.net

Title		
1802 MemberCHIP Card		
Size	Document Number	REV
A	C:\ORCAD\SHEET\1802\1802ME1.SCH	B
Date: February 18, 2024	Sheet 1 of 1	


1. 3 BIDIRECTIONAL PARALLEL PORTS
J1 FEMALE SOCKET
CLOCK
/WAIT
/CLEAR
Q
SC1
SC0
/MRD
D7
D6
D5
D4
D3
D2
D1
D0
OVCC
ON1
ON0
OVSS
N2
OUT4-7 accesses 82C55 registers.
U2A 74HC00
U2B 74HC00
U2C 74HC00
U2D 74HC00
U1 82C55
RD
WR
CS
RESET
VCC
GND
C1 0.1uF
2. MOST VERSATILE
U1 65C22 VIA
OUT4: R/W last Register
OUT5: R/W Reg.0
OUT6: R/W last selected Reg, then Reg+1
U2A COUNTER
74HC393
/MRD
U2B
C2 47pF
R1 1K
50nsec to sync counter to 1802 bus cycle
CLOCK
/MRD
CLR
PHI2
/CLEARO
IRQ
RES
C1 0.1uF
65C22 VIA has 2 parallel ports, 2 timers, and an 8-bit shift register.
1802 Q, /EF1-4, and /INT can also be used for I/O and handshaking.
3. BASIC I/O
U1 74HC273
8 output bits
U2 74HC138
OUT1
OUT2
OUT3
INP1
INP2
INP3
U3 74HC541
8 Input bits
R1-8 10K
U1 74HC259
7 output bits
4. VERY SIMPLE I/O
TMSI c/o Lee Hart www.sunrise-ev.com
Title
1802 MemberCHIP card I/O Circuit examples
Size A
Document Number C:\ORCAD\SHEET\1802\1802ME-P.SCH
REV A
Date: April 12, 2021
Sheet 1 of 1

This page would have been blank if I hadn't put this silly message on it.