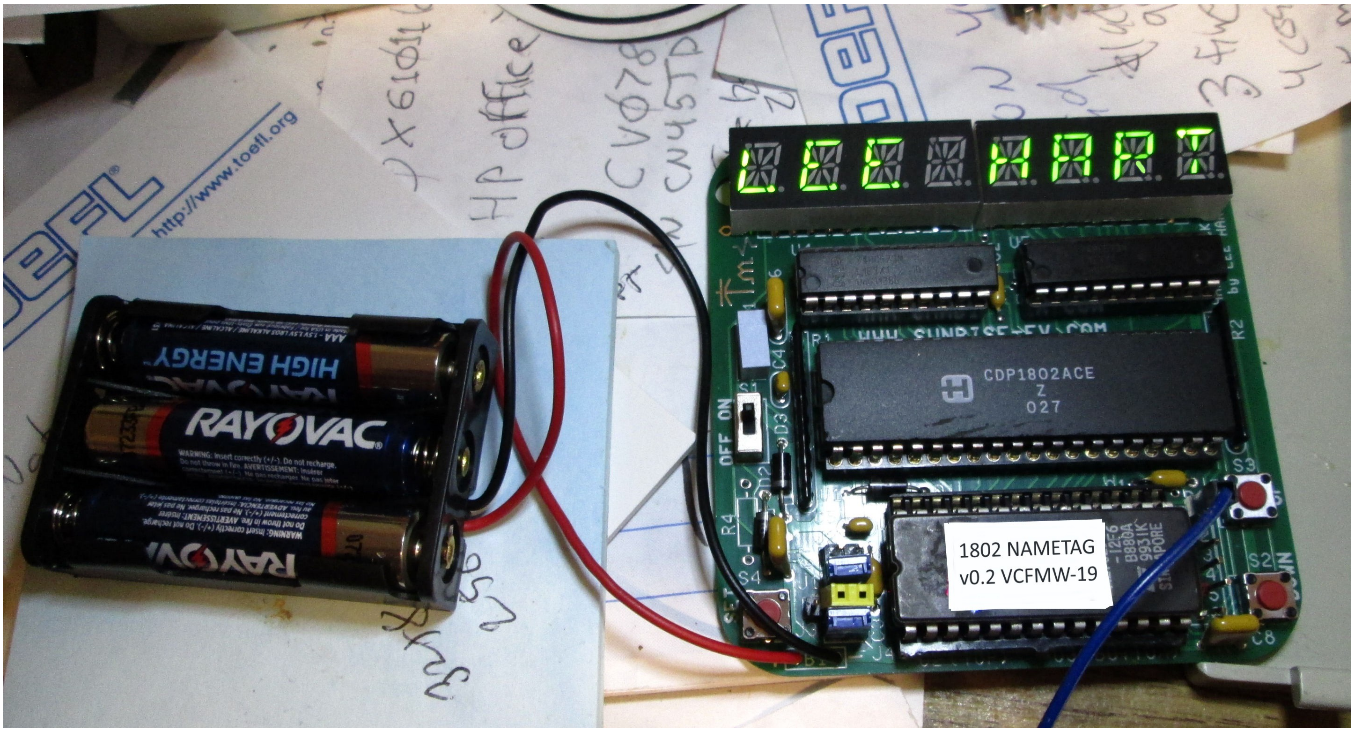
An 1802 BADGE

for the Vintage Computer Festival Midwest

by Lee Hart, David Madole, Chuck Yakym, and Herb Johnson – 10 Feb 2025



What is it?

- A cool retro nametag that displays a scrolling message of your choice
- Celebrates the 50th anniversary of the introduction of the microprocessor
- Classic user interface that is functional and educational
- Minimum size, cost, and parts count
- Maximum fun!

Description

The Badge is a complete working 1802 computer, built entirely with vintage parts and technology. Powered by batteries or a USB port, its LEDs display up to a 32-character scrolling alphanumeric message. On-board buttons let you set the message directly, or a serial port provides user interaction with any computer. Features:

- A CDP1802 microprocessor, running at 2 MHz
- 8K RAM (optional), with battery backup
- 8K EPROM (expandable up to 32K)
- An 8-digit 14-segment “starburst” LED display
- Software-driven 9600 baud TTL serial I/O port
- And, just a few components to tie it all together
- This is just rev.C – more to come!

For software, manual, and more information see **https://www.sunrise-ev.com/projects.htm#badge**

1802 Badge Parts List

QTY	ID#	Description	Source
3	A1,2,3	Nimh rechargeable AAA cells	tenergy ebay#278733418500
1	B1	battery holder for three AAA cells	greencell ebay#153299048475
3	C1,2,3	0.047uF 50v axial ceramic capacitor	jameco.com 2229811
1	C4	0.1uF 50v radial ceramic capacitor	jameco.com 2312439
1	C5	1000pF 50v ceramic capacitor	digikey.com BC5133-ND
3	C6,7,8	0.56uF (560nF) 50v ceramic capacitor	theelectronicgoldmine.com G18828
3	D1,2,3	1N5818 or 1N5819 Schottky diode	jameco.com 177957
	J1-J5	jumper wires (left over from cut leads)	
1	LED1,2	17-segment 4-digit LED display	(1) theelectronicgoldmine.com G21553
1	P1	6-pin stacking connector	samtec.com SSQ-106-03-G-S
1	R1	1meg x 5 SIP10 resistor network, isolated	mouser.com 652-4610X-2LF-1M
1	R2	5.6K x 4 SIP8 resistor network, isolated	mouser.com 652-4608X-2LF-5.6K
1	R3	47 ohm 5% 1/4w resistor	jameco.com 690742
1	S1	switch SPDT micro slide	lizardleds ebay#253329337174
3	S2,3,4	switch SPST tactile pushbutton	(1) theelectronicgoldmine.com G21644
1	U1	CDP1802 microprocessor	alltronics.com CDP1802ACE
1	U1a	40-pin IC socket for U1	jameco.com 41136
1	U2	27C64 programmed EPROM, labeled “1802 NAMETAG v0.2 VCFMW-10”	(2) order from me
2	U2a	14-pin IC socket strip	jameco.com 2125675
	U3	8K RAM 0.3" wide (not supplied in basic kit)	mouser.com 913-AS7C164-15PCN
1	U4	74HC541 octal buffer	jameco.com 46050
1	U5	74HC573 octal latch	jameco.com 46076
1	Y1	2 MHz ceramic resonator with capacitors	mouser.com 520-ZTT200MG
1	PCB	Badge printed circuit board, rev.C	# (also get it from me)
2	hardware	#4-40 screw, nut, and jackscrew	hardware store
1	HW-597	CH340G USB to TTL serial adapter	survy2014 ebay#201414990214
1	clip	to hang the Badge on your shirt etc.	scrounge from an old convention badge

Notes

Rev.A – Original release. LEDs were too close together, so their pins need to be bent to fit. RAM writes had unnecessary Wait states. LEDs interfered with RAM writes. (Don't use it with a RAM).

Rev.B – Corrected LED spacing. Fixed RAM writes so they don't have WAIT states. Added slot for clip to hang badge. Added jumper J5 to disable LEDs so you can write to an EEPROM at U2 for program development.

Rev.C – Same as B, but increased spacing on S1 to avoid shorts. Added slot for battery holder wires.

1. The LED display and pushbutton switch are out of stock at thelectronicgoldmine.com. There may be other sources, but you can order them directly from me; Lee Hart leeahart@earthlink.net

2. The programmed EPROM and PCB are special parts. Order from Lee Hart leeahart@earthlink.net

3. IC sockets are only supplied for U1 and U2. You can socket the others ICs if you like. For RAM IC U4, use socket pins so it will fit under U2 (digikey.com ED5037-ND or mouser.com 575-055210).

Assembly

Check the parts list to be sure you have all the parts. Mount all parts on the TOP (printed) side and solder them on the BOTTOM side. Mark each box (■) as you install each part. Note: ICs come with their pins bent slightly outward. To fix this, stand the IC on its side on the table, and tip it slightly inward so the pins are straight and will fit into the holes on the board.

( ) LED1: 4-digit LED display. Install it so the side with printed text is on the bottom.

( ) LED2: 4-digit LED display. Install it the same way as LED1.

( ) U4: 74HC541. Install it so lettering is right side up.

( ) C2: capacitor. 0.047uF, yellow, marked “R5C 473”.

( ) U5: 74HC573. Install it so lettering is right side up.

( ) J5: Nothing needs to be installed here.

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

( ) C6: capacitor. 0.56uF, yellow, marked “RMC .56K” and “X7R 200V”.

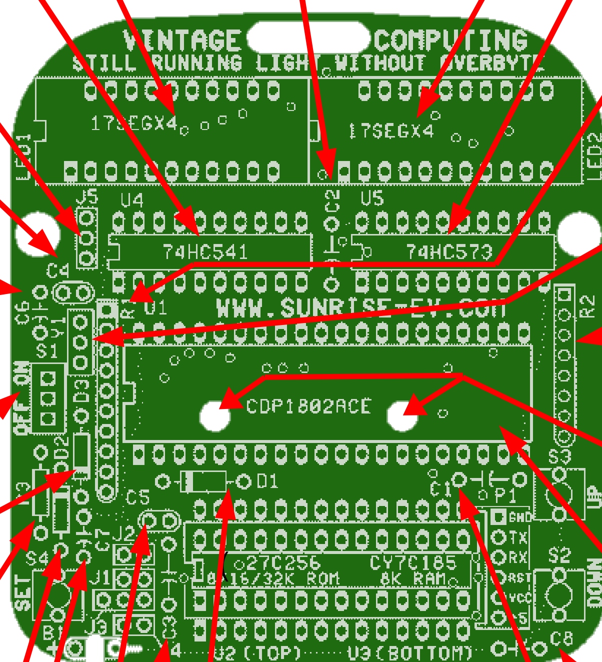
( ) S1: Slide switch.

( ) D3: 1N5818 diode. Banded end <u>down</u>.

( ) R3: 47ohm resistor. Only install to use with nimh batteries!

( ) D2: 1N5818 diode. Banded end <u>up</u> (↑).

( ) C7: 0.56uF capacitor. Yellow, marked “RMC .56K” and and “X7R 200V”.



( ) D1: 1N5818 diode. Banded end on <u>left</u>.

( ) C3: 0.047uF capacitor. Yellow, marked “R5C 473”.

( ) C5: 1000pF capacitor. Yellow, marked “102 C5K”.

( ) R1: 10-pin resistor network, marked “B105G”. End with dot goes on <u>top</u>.

( ) Y1: 2MHz resonator Blue, marked 2000A

( ) R2: 8-pin resistor network, marked “B562G”. End with dot goes on <u>top</u>.

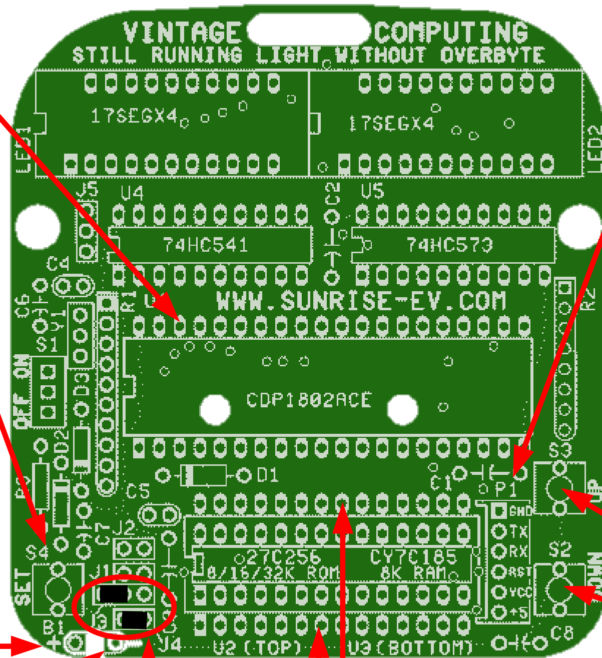
( ) #4 hardware: Mount jackscrews on bottom with nuts on top.

( ) U1a: 40-pin IC socket. Cut out the center two bars so they don't hit the #4 nuts.

( ) C8: capacitor. 0.56uF, yellow, marked “R5C 56K” and “X7R 200V”.

( ) C1: capacitor. 0.047uF, yellow, marked “R5C 473”.

Assembly (continued)



( ) U1: CDP1802ACE microprocessor. Plug into socket so the lettering is right side up.

( ) S4: pushbutton.

( ) Battery holder: Cover the bare wire on the back with a couple layers of tape to prevent shorts.

( ) B1: Battery holder:

a. Solder RED wire into the "+" hole.
b. Solder BLACK wire into the "–" hole.
c. Route wires through slot to back of board.
d. Mount battery holder to jackscrews with #4 flat-head screws.

( ) P1: stacking header Put the black body on <u>top</u> of the board, and solder the pins on the bottom. Don't cut off the extra pin length; they can be used to plug in USB serial adapters with sockets.

( ) S3: pushbutton.

( ) S2: pushbutton.

( ) U2: EPROM marked "1802 Nametag v0.2".

a. Place the two 14-pin socket strips in the <u>outer</u> row of holes. Do not solder them yet.
b. Plug the EPROM into the socket strips with the text right side up, and notch on the left end.
c. Now solder the socket strips to the board. (This insures the sockets are in the right places).

Jumper Options

There are four jumpers to select the size and type of ROM at U2, and one to enable/disable the LED display when writing to an EEPROM at U2. The current kit comes with an 8K EPROM at U2 (and no RAM at U3), so set the jumpers as follows. See schematic for a chart showing the jumper settings for other parts at U2.

Use scrap pieces of wire for jumpers. Or, you can install 0.025" square pin headers (not supplied), and use shorting jumpers if you want to change them.

( ) J1: Leave it open (no jumper).

( ) J2: Leave it open (no jumper)

( ) J3: Short the <u>center</u> pin to the <u>left</u> pin (closest to S4).

( ) J4: <u>Short</u> the two pins.

( ) J5: Leave it open (no jumper).

There is already a foil jumper shorting the top two pins of J5 to enable the LEDs for writes to 0-32K. If you want to use an EEPROM at U2 and write to it in-circuit, cut this jumper, and short the center pin to the lower pin to disable the LEDs.

Let's See It Work!

Now for the big moment. The batteries supplied will probably need to be charged before use, as it's safer to ship them that way. So use the USB-serial adapter (supplied) to power the badge. **BE CAREFUL NOT TO PLUG IT IN BACKWARDS!** The bottom of the adapter labels its 5V and GND pins; plug it in so the +5V and GND pins match the labels on the board. (See Appendix B for information on installing and using this adapter.)

You should be rewarded with a (random) scrolling message!

Note: The 1000 mAH claim on the batteries is a lie; they are more like 500 mAH (but that's still 8-10 hours of running time).

Operation

To set the 32-character message with the buttons:

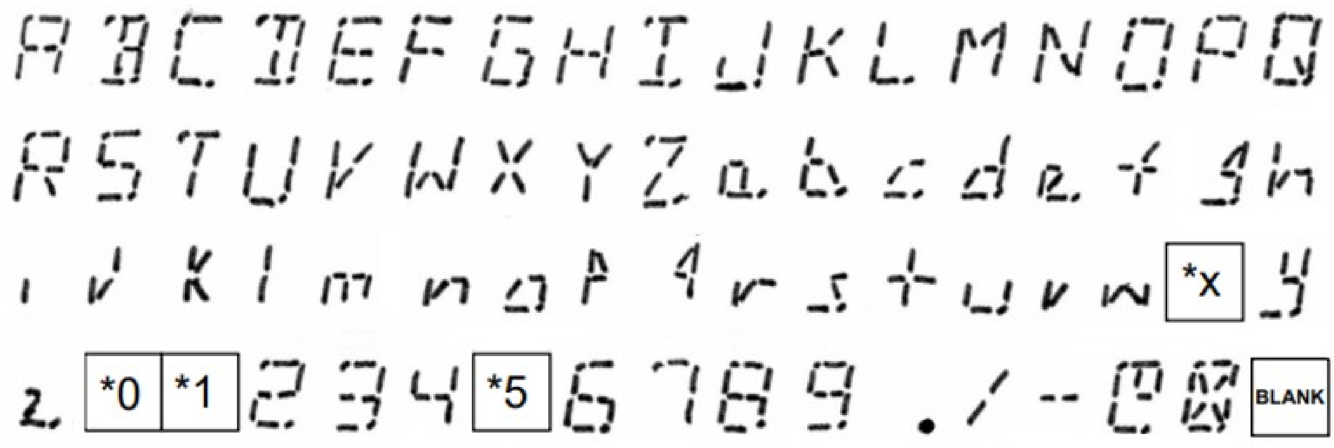
1. Turn the Badge "off" with S1. Press and hold SET, then turn S1 "on". This stops the display from scrolling, and displays the next character of the current message in the rightmost digit.
2. Press both UP and DOWN together (while holding down SET) to clear the display to all blanks. This insures that you can start your new message at the beginning, rather than somewhere in the middle.
3. Press UP or DOWN to display the desired first character. There are 64 characters to choose from (A-Z, a-z, 0-9, and some punctuation). The character set is shown on the next page. Use a capital "O" for digit "0", lowercase "l" for digit "1", capital "S" for digit "5", and capital "X" for "x". You can hold a button down to auto-repeat. When it gets to the last character in the set, it loops back to the other end of the set.
4. When the desired character is shown, release the SET button momentarily to step to the next character.

To edit a message already loaded:

1. Press and hold the SET button when the character you want to change appears in the right-most LED character location.
2. Press UP or DOWN to change it to a different character.
3. Release the SET button momentarily to step to the next character, then press and hold SET again to stop on it.
4. Release the SET button when finished.

(Yeah, it's a bit awkward. Give us time. The software was barely finished in time for the show! We're still working to improve it.)

Badge Font



*x	There is no lowercase x. Use an uppercase X.
*0	There is no number 0. Use an uppercase O.
*1	There is no number 1. Use a lowercase l.
*5	There is no number 5. Use an uppercase S.

Serial Port

The serial port is a TTL asynchronous data transfer type using Transmit Data, Receive Data and a common ground. It is similar to the RS-232 standard, but uses 5V and 0V logic levels and non-inverted data (Idle = 5V = logic 1). You can use a TTL to RS-232 adapter to talk to any traditional terminal or computer with an RS-232 serial port. Or, you can use a TTL-to-USB serial adapter which are supported by most modern computers and operating systems. As a bonus, the 5V power from the USB port is used to power the board and charge the batteries (if you have rechargeable batteries and R3 is installed).

Connector P1 is to connect a USB adapter. Pin 1 = GND, pin 2 = TX (serial data out from the Badge), pin 3 = RX (serial data in to the Badge), pin 4 = RST (reset the 1802), pin 5 = VCC, and pin 6 = +5V.

On many adapters, the DTR or RTS signal is available as an output. These are normally high when the Terminal is off-line or disconnected, and go low when the Terminal is on-line or connected. The Badge can use the falling edge of this signal to reset the 1802. Alas, the adapter supplied does not bring this signal out to a pin. But if you're adventurous, you can add it (see Appendix B). Or, just use an external pushbutton to ground pin 4 to reset the 1802.

Configure your serial port for 9600,N,8,1 (9600 baud, No parity, 8 data, and 1 Stop bit).

To set the scrolling message with the serial port:

1. Turn the Badge off. Unplug the USB adapter from the Badge. Remove one of the batteries if your USB adapter cannot control DTR or RTS (i.e. the supplied adapter).

2. Hold down <u>both</u> the UP and DOWN buttons.

3. Plug in the USB adapter. The Badge will automatically turn on, and display the following message on your terminal:

```
1802 Name Tag V.0 by Lee Hart & David Madole
TAB to skip ahead, RETURN to blank to end.
>                                 <
```

4. Type your message between the > and <. You can use A-Z, a-z, 2,3,4,6,7,8,9, period, dash, slash, and at-symbol. Use uppercase "O" for digit 0, lowercase "l" for digit 1, and uppercase "S" for digit 5. Characters not available (see Badge Font chart above) are ignored. TAB skips ahead one character, and ENTER returns to the start of the line.

5. When you have entered the last character, hit the ENTER key to save it. Replace the battery, unplug the USB adapter, and turn the Badge on. You should see your message!

Software Description

The program produces a 32-character scrolling display on the LED digits. The message is random at first start-up, but messages you load are retained by the batteries even with S1 "off". "Off" power consumption is tiny (microamps), so the batteries will last for years.

The program was written by David Madole. Source (and a neat animated GIF) is on his github page at https://github.com/dmadole/Nametag. There's some curious stuff going on in the code if you like puzzles.

Notice that there is no RAM! Unlike most vintage microprocessors, the 1802 can function perfectly well without RAM. The 32-character message is stored in the 1802's sixteen 16-bit registers, using 6 bits per character, with 8 characters packed into each group of three 16-bit registers. It thus uses 12 registers for the message, and 4 for program execution.

Since the 1802's address pins (A0-7) have less drive than the 74HC573 (A8-A15), the software leaves them on twice as long to balance the brightness.

The serial port is bit-banged in software, using the 1802's EF3 flag pin for input and the Q pin for output.

Hardware Description

The hardware was designed by Lee Hart. It follows Einstein's dictum to "keep things as simple as possible, but not too simple".

The 1802's internal oscillator uses ceramic resonator Y1 and two resistors in R1 to produce a 2 MHz clock. The 1802 has a multiplexed address bus, so U5 latches the upper bits (A8-A15) at each TPA pulse. The lower address bits (A0-A7) are used directly.

C4 and one resistor in R1 form the reset circuit, to momentarily pull the 1802 /CLEAR pin low on power-up. C6 and R2C also allow a negative-going pulse on P1 pin 4 to reset the 1802. This can be generated with an external pushbutton, or from the DTR or RTS signal from a USB-TTL serial adapter.

A0-A14 select a location in EPROM U2 at any address from 0-32K (but an 8K EPROM only uses A0-A12, so it is partially decoded). A15=0 chip-selects U2, and /MRD=0 enables it to put the selected location's contents on the data bus for the 1802 to read and execute.

The LED displays are multiplexed, with 8 active-high digit selects (one common anode per digit), and 15 active-low segment-selects (one cathode per segment). To avoid output latches for all these bits, 1802 address bits A0-A14 select a segment, and data bits D0-D7 select a digit. U4 is a buffer for extra drive, since a digit may have up to 15 segments all active at once.

The 1802 displays a character by “writing” to the EPROM memory space at 0-32K. This sets /MWR=0 (to enable anode buffer U4), sets one Data bit high (to select the desired digit), and puts the desired segment pattern on the Address bus. /MWR=0 also pulses the 1802 /WAIT input low for 1 mSec via C5 and R1. The WAIT keeps the LEDs on long enough to be visible. The program spends about 10% of its time figuring out what to display in the next digit, then 90% of its time displaying it.

EPROMs ignore the Write cycle (of course). But it is possible to set the Jumpers to use an EEPROM at U2, so you **can** write to it. To do this, cut the foil jumper at J5 and reconnect the center pin to the lower pin to disable the LEDs while writing to an EEPROM. Otherwise, the LEDs will load down the address and data buses and you'll get bad writes.

EEPROMs can't be read immediately after writing; but the 1 mSec WAIT generator will automatically keep the 1802 program from continuing until the Write cycle is finished. (Check the data sheet for your EEPROM to be sure 1 mSec is long enough to write to it.)

Pushbuttons S2, S3, and S4; and pullup resistor in R2 control the 1802's EF1, EF2, and EF4 flag inputs. The program reads them to control the scrolling message in the 1802 registers.

The 1802 Q pin is used for serial output, and the EF3 pin is used for serial input. R2 provides a pullup resistor, in case no external serial input device is connected. The serial input is also connected to the 1802 /INT pin so a serial input can interrupt the program; but the software does not currently use this capability.

If A15=1, diode D1 holds /WAIT high, so writes to an optional RAM at U3 do not have WAIT states. But the rev.A board does not disable the LEDs during writes to RAM; so it is not practical to add a RAM unless you find a way to disable the LEDs during RAM writes (such as by disabling U4 by jumpering pin 1 from /MWR to VCC). The rev.B board fixed this by changing U4 to a 74HC541, which has two enables; one to /MWR, the other to A15).

An optional 0.3” wide 8K RAM can be installed at U3. Sample part#s are Alliance AS7C164AL, Hitachi HM6264ALSP, Sony CXK5864BSPL, Toshiba TC5565L, Cypress CY7C185L, and Alliance AS6C6264L. An “S” in the suffix specifies a 0.3” wide package (for parts available in both 0.3” and 0.6” wide versions), and “L” means it is low power (needed for battery backup). Millions of these parts were used as cache memories in 486-class PCs, so they are available from the usual offshore scrap dealers. The Alliance AS7C164 is currently available from Mouser.com (913-AS7C164-15PCN) but it is not a low-power version (it draws ~1mA standby).

The batteries continuously supply power via Schottky diode D2 to 1802 VDD pin 40 and the (optional) RAM. This maintains the contents of the 1802's registers (and RAM). With S1 “off”, VCC has no source of power and so will be ~0v. This insures the RAM's active-high chip-select CS1 pin 20 will be low, putting the RAM in standby mode. Power drain when S1 is “off” is thus tiny (just a few microamps), so fully-charged batteries will last for years.

Switch S1 connects VCC to VDD, to power everything else. If you connect an external source of 5V power (such as a USB-serial adapter) Schottky diode D3 will power VDD (and optional RAM) even if dead or no batteries are installed. Resistor R4 (if installed) will also charge any rechargeable batteries at B1.

The Badge is all CMOS, so power consumption when “on” is quite low; 10-20 mA (depending on the EPROM type and how many LEDs are on). Adding a bigger EPROM, EEPROM, or RAM will increase power consumption.

“The plan” is to continue working on the Badge. If we find a source of good low-power RAMs we can add a monitor, BASIC, and other features.

Hints and Kinks

It's kind of awkward to hold down the SET button while programming the Badge. It's a bit easier if you install a toggle switch in place of the SET pushbutton. Set it “off” to run, “on” to program.

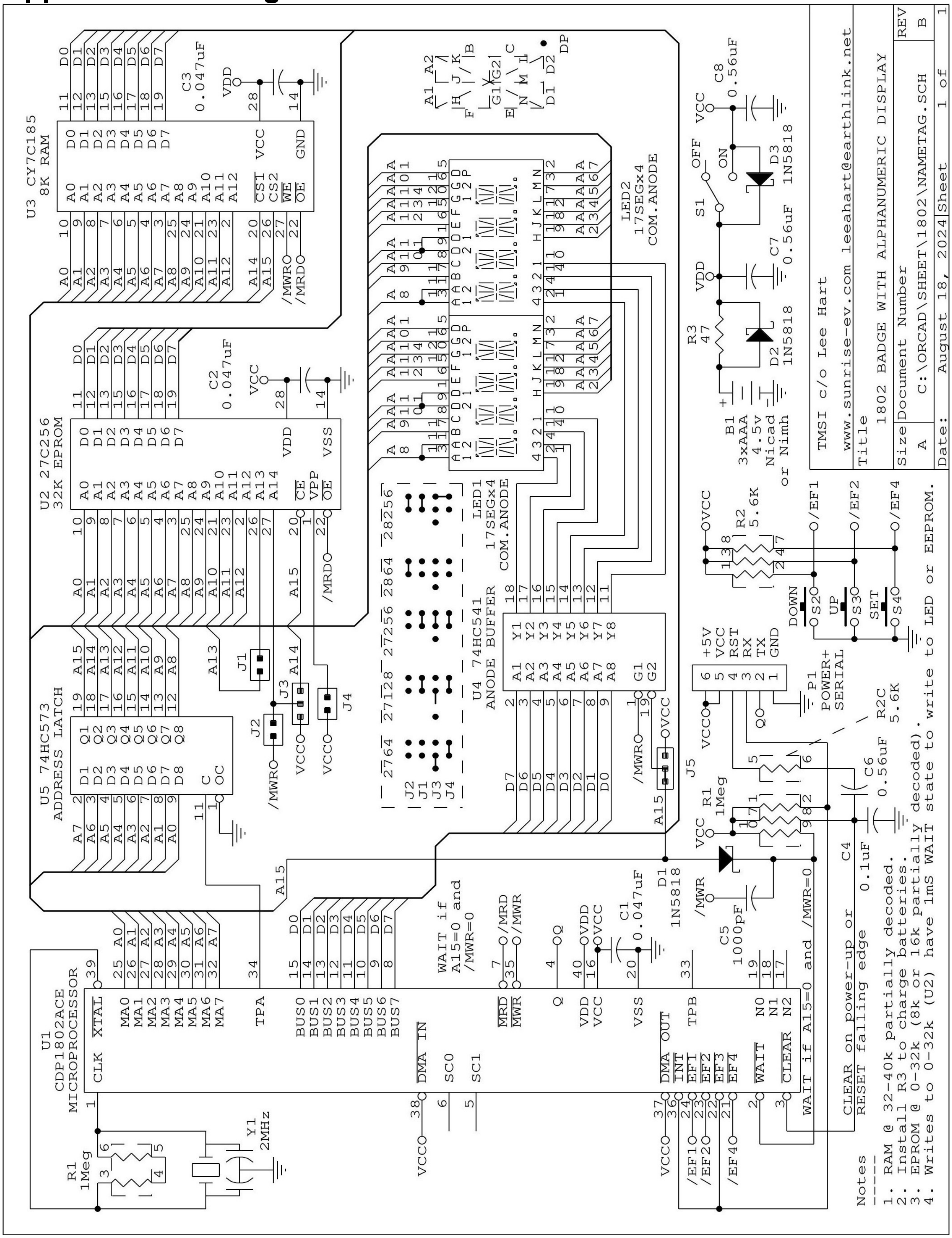
Display brightness is low for good battery life. You can add a green filter (such as Digikey.com PRD360G-ND, $2.60) to improve contrast and readability.

Paul Schmidt has made a 3D-printed case for his Badge. He also added a toggle switch in parallel with the SET button to save his poor fingers during programming. A ZIP file with his STL files and instructions is at www.serpentwebsite.com/fileshare.htm.

That's all, folks! Let us know what you think of the Badge, and if you have any questions or ideas for improvements. :-)

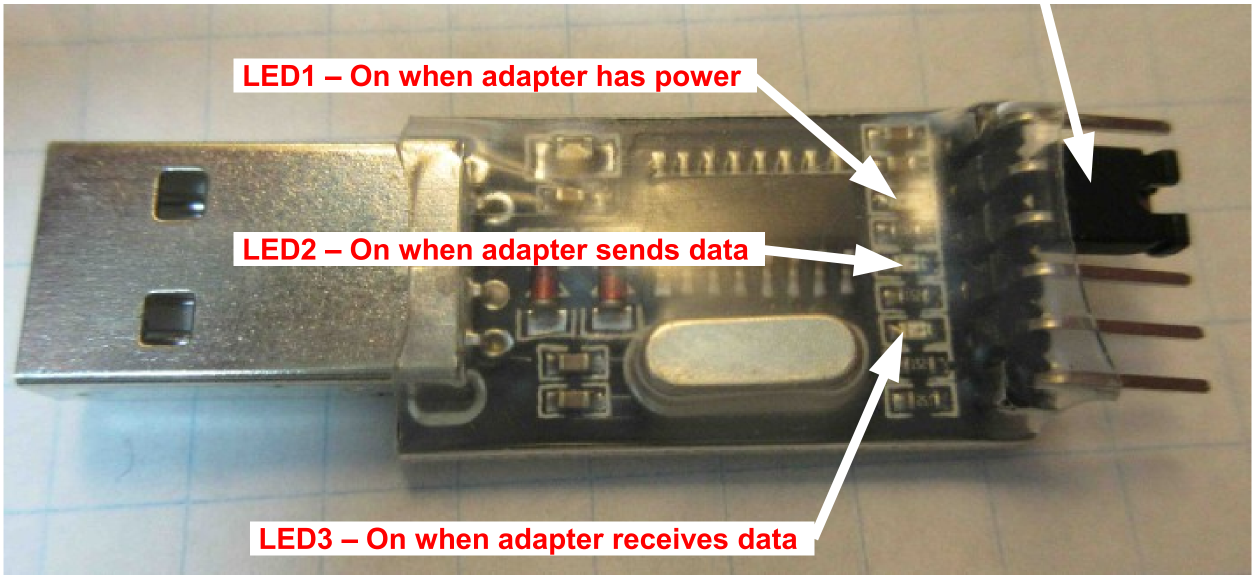
Lee Hart, leeahart@earthlink.net, 320-656-9574, www.sunrise-ev.com

Appendix A -- Badge Schematic



Appendix B - CH340G USB-TTL Adapter

This device converts a modern computer's USB port into a vintage serial port. The (included) jumper configures it for 3.3V (short VCC to 3V3) or 5V (short VCC to 5V) output levels. The Badge uses 5V levels, so the VCC-5V connection is already made on the Badge PC board, so **REMOVE** this jumper.

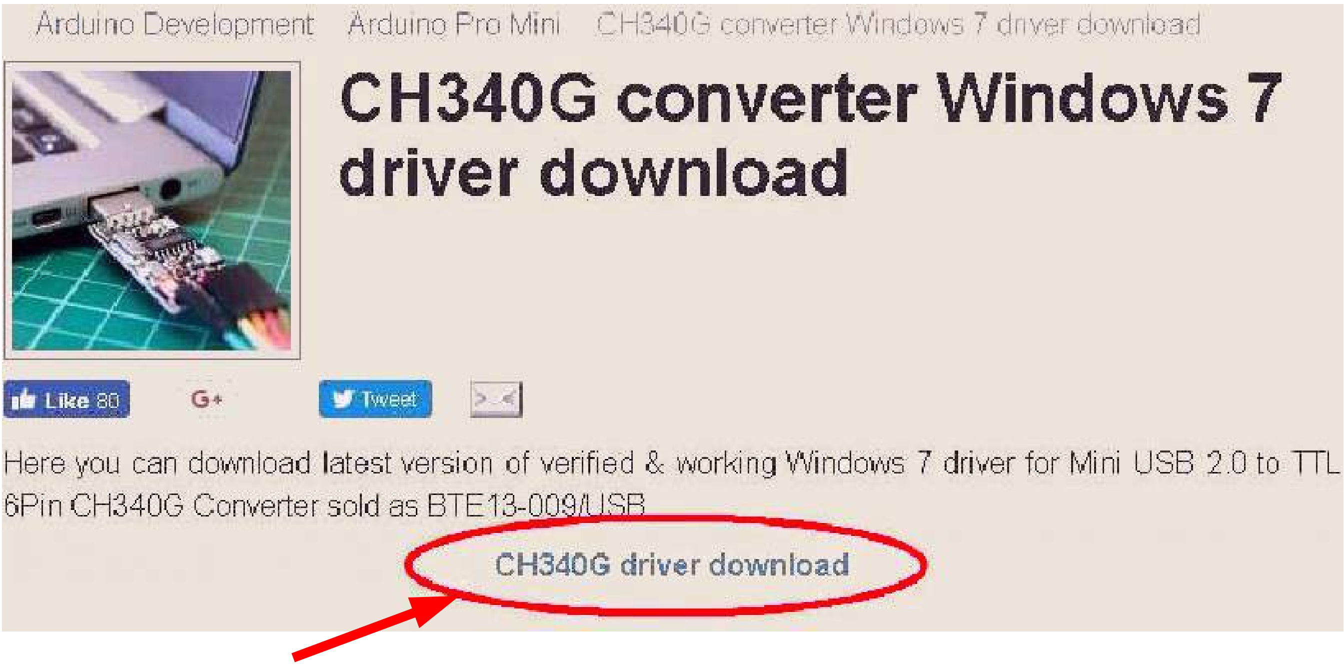


It's a typical modern "no documentation or support" gadget. We do not support it for them; so use these notes at your own risk. This is just an unofficial and independent description of how we got it to work with Windows 7. It **should** work with other version of Windows and other operating systems; but you'll have to find the drivers, and the key pokes and mouse strokes to install it will be a little different.

A driver **must** be downloaded and installed manually **BEFORE** you plug in the module! For Windows 7, one source is http://www.arduined.eu/ch340g-converter-windows-7-driver-download/. This is an Arduino site, and believed to be legitimate and less likely to contain malware or viruses. If this URL is no longer valid, or you need one for a different operating system, search for "CH340G driver". The manufacturer's own driver is at http://www.wch.cn/downloads/CH341SER_ZIP.html (Google can translate it for you).

As an example, here are step-by-step instructions for installing the driver for Windows 7:

1. Find a website that lists a driver for your operating system and version. For Win7, we'll use the one on the Arduino page http://www.arduined.eu/ch340g-converter-windows-7-driver-download/.

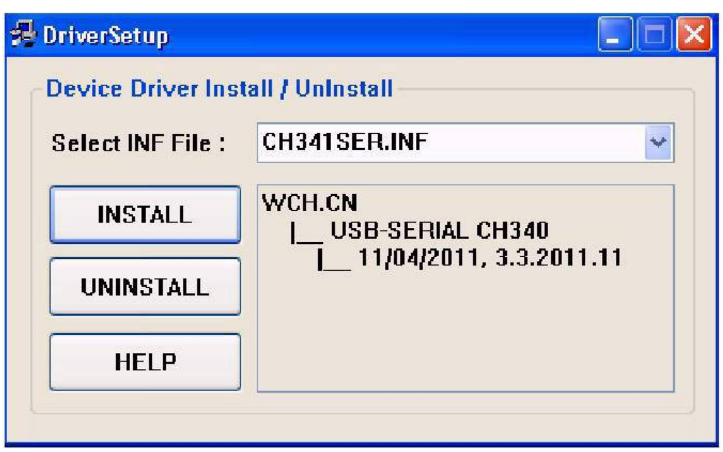


2. Click on the link to download the driver. This downloads **CH341SER.zip** which is a ZIP file. It must be "unzipped" to extract all the files inside it.

3. Open your **Downloads** folder (or wherever you or your computer puts downloaded files).

4. Right-click on the **CH341SER.zip** file, then click "Extract All..." This creates a new folder named **CH341SER** in your Downloads folder with all the new files and subfolders unzipped inside it.

5. Open (double-click) the new **CH341SER** folder. Inside it, you will see *another* "CH341SER" folder, and an "INSTALL" folder.



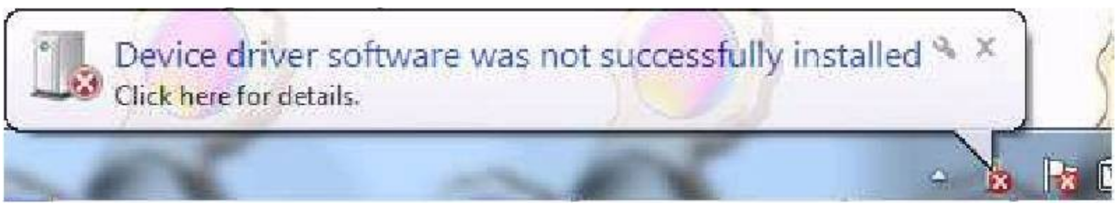
6. Open (double-click) this inside "CH341SER" folder. Inside it are a bunch of files including a **SETUP.EXE** program. Aha! Run (double-click) the SETUP program to display this dialog box...

7. Click **INSTALL**. You should get the following message box. (Please forgive poor Engrish.)

8. **Now** you can plug the USB adapter into your computer. Windows should detect the new USB device, and look for the driver for it. It **might** find the new driver all by itself, in which case you should now have a new COMn port.

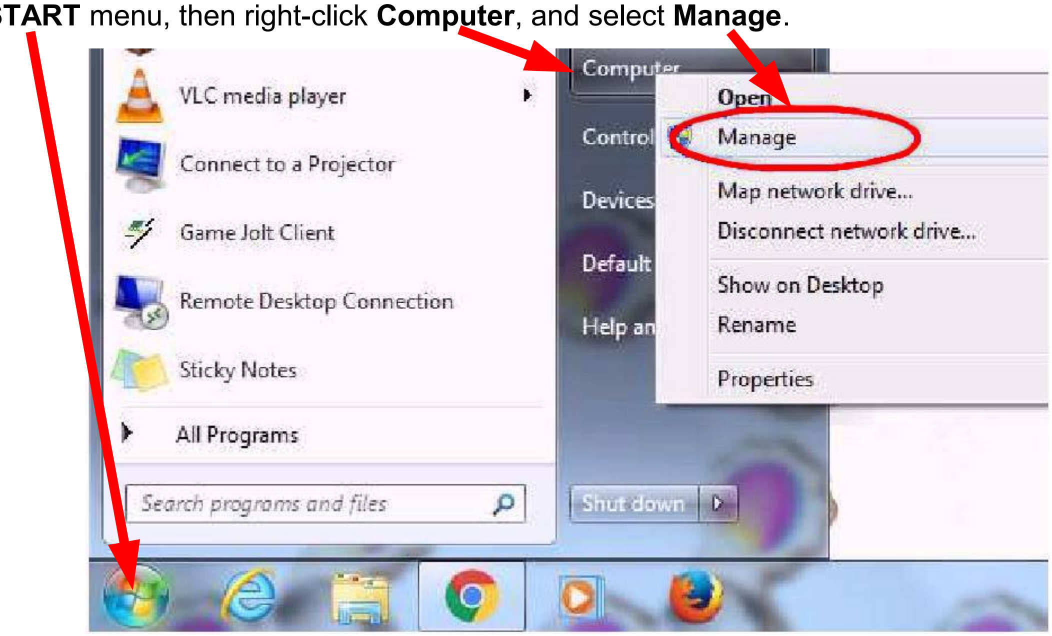
9. If this doesn't work, you have to manually guide Windows to new driver. When you plugged in the USB adapter, Windows may ask you where find the driver for it. Try telling it to look in the new CH341SER folder you created.

10. If you plugged in the USB adapter before you installed the driver, or if Windows can't find the downloaded driver, you'll get an error message like...

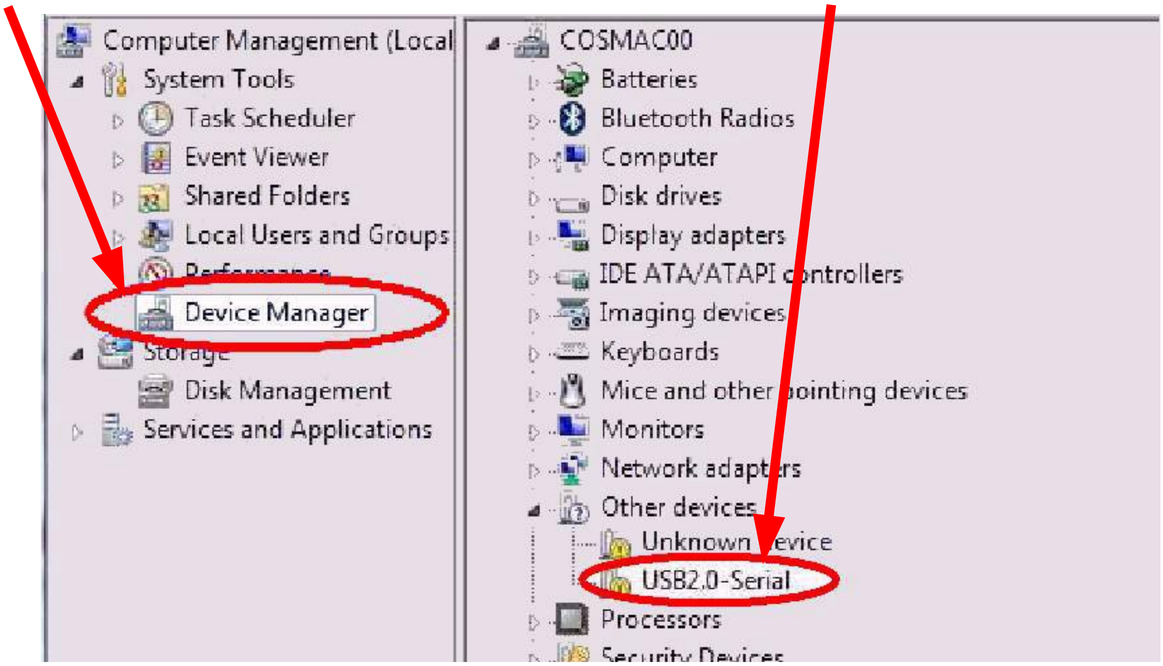


In this case, you have to manually guide Windows to find and install the driver. Briefly, you go to the **Device Manager**, right-click to update the driver, then select "Choose my own path", and point that path to the CH341SER folder you created above. Here is the procedure in detail...

11. Click on the **START** menu, then right-click **Computer**, and select **Manage**.

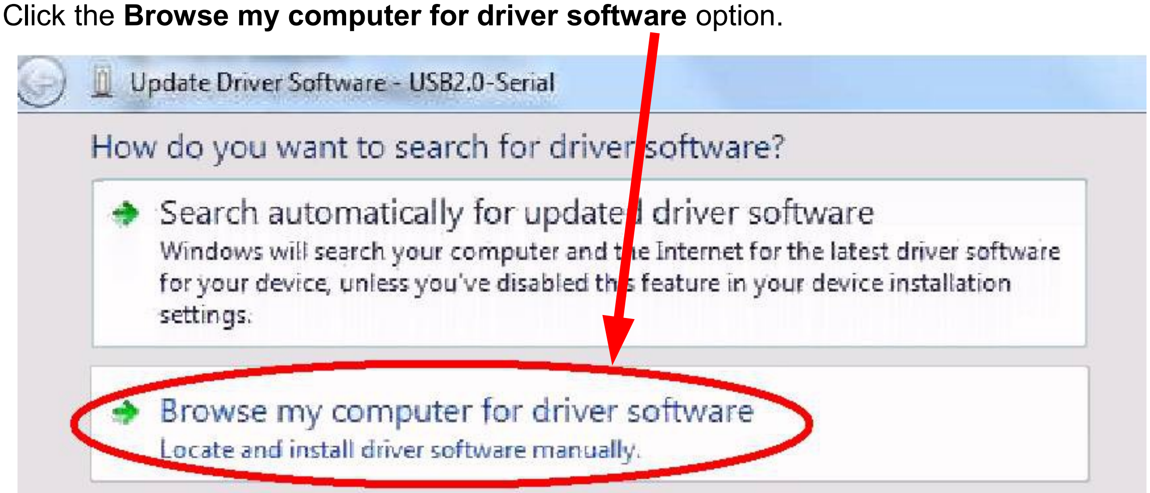


12. This opens the Computer Management menu as shown below. Under “System Tools”, click the **Device Manager**, then under “Other Devices” look for the **USB2.0-Serial** device.



13. Right-click the **USB2.0-Serial** device, and then select the **Update Driver Software...** option.

14. Click the **Browse my computer for driver software** option.



15. (Installing the driver... continued...) The following dialog box (or its equivalent) should pop up. Click the **Browse...** button, find the **CH341SER** folder with the driver files, and click on it so it appears in the "Search for driver software in this location:" box. Be sure the "Include subfolders" box is checked. Then click the **Next** button to install it.

16. Remove, and then re-insert your CH340G USB-serial adapter. Now Windows should find it, and use the correct driver. (You should no longer get the error message in step 10.)

Testing the USB-TTL Adapter

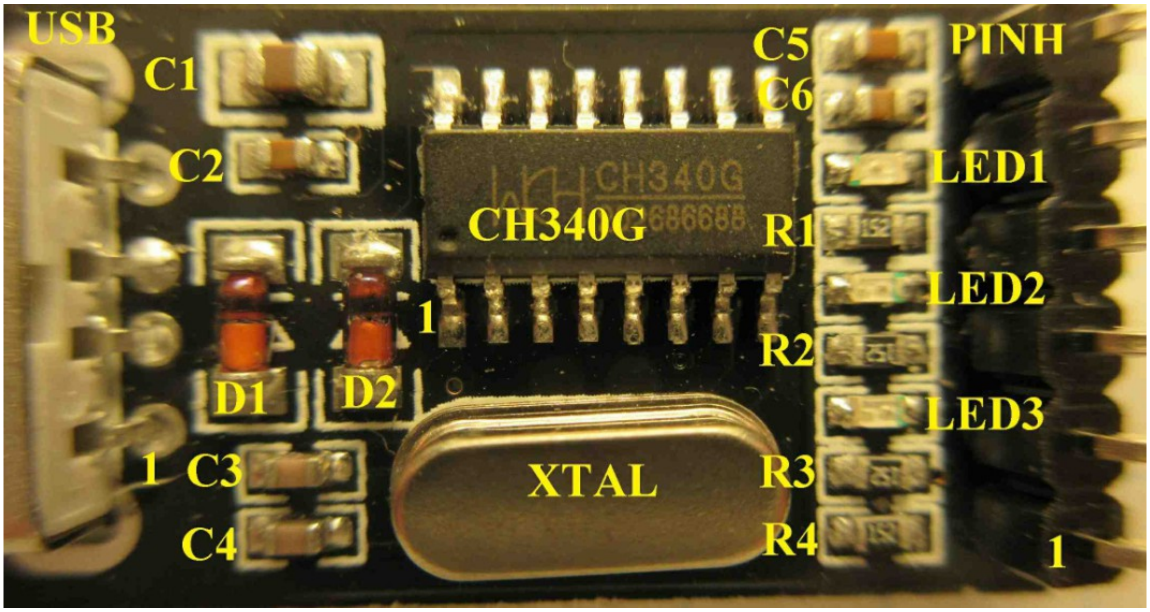
TEST the Adapter before you plug it into your Badge. Use the jumper (on the adapter) to short TXD to RXD, and plug it into your PC. Run your Terminal program (Hyperterm, RealTerm, TeraTerm etc.). Configure it for 9600N8 (9600 baud, No parity, 8 Data bits), and No hardware or software handshaking. When you tell it to "connect", any keys you type on the keyboard should be echoed and appear on the screen. TXD and RXD are normally high, and go low on Start bit and zero data bits.

You'll need some kind of Terminal program. There are dozens of free Terminal programs, for every computer and operating system. They usually come as part of the operating system. For decades, Windows came with "Hyperterm", but they've stopped providing it. It's not very good; but it worked and was common and free to download.

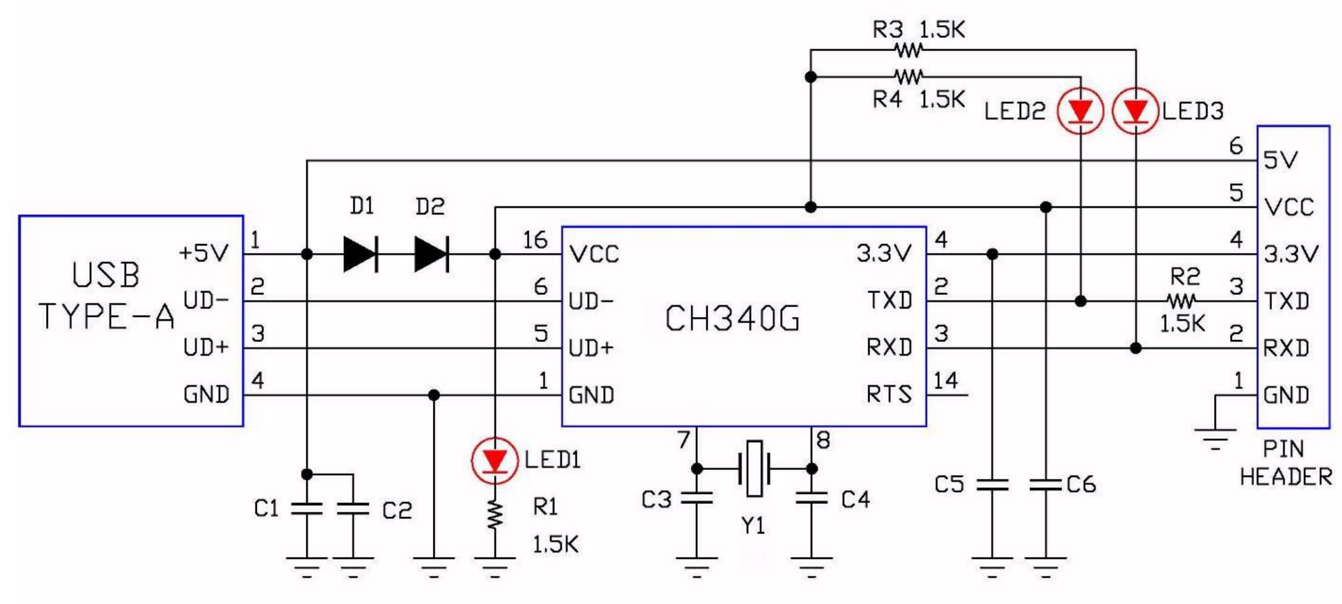
Once this works, remove the jumper and plug the Adapter into connector P1 on the Badge. The "GND" pin of the adapter goes in the Badge socket at the top marked "**–**" and is the one closest to the edge of the board. **BE SURE NOT TO PLUG IT IN BACKWARDS !** If you do, it reverses the +5v and GND connections, and can destroy the Badge!

A good plan is to "key" the adapter so it can't be plugged in backwards. To do this, **CUT OFF** the adapter pin labeled "3V3". Plug the mating hole on the 1802 or 6502 Badge (P1 pin 4, labeled "RES") with a piece of a toothpick etc. This will also keep the 3V3 signal from preventing resets on the Badge.

If you want to use the adapter with some other gadget, you won't want to cut off the 3V3 pin. It is used to set the adapter's serial output logic level to either 5v or 3.3v. The 5V pin provides +5v power from the USB port to power things (like the 1802 or 6502 Badge). The 3V3 pin provides a small amount of 3.3v power from the CH340G chip, and is not able to drive any significant load.



Top of the Adapter board, with shrink-wrap removed and parts labeled.

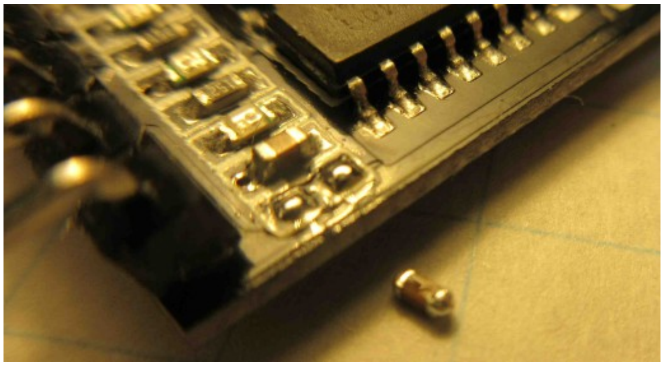


Schematic

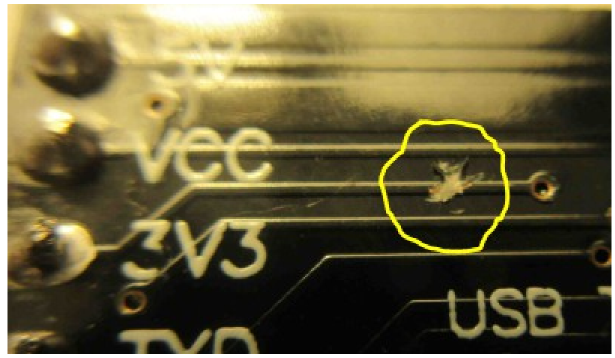
Modification to add RTS (Optional – Not for mere mortals!)

Most terminal programs set RTS high initially, or when you use their “hang up” or “disconnect” command; and set RTS low when you use their “on-line” or “connect” command. RTS can thus be used to reset the Badge (and other devices). RTS is available on pin 14 of the CH340G chip, but is not brought out to the 6-pin header. This modification replaces the 3V3 pin with the RTS signal.

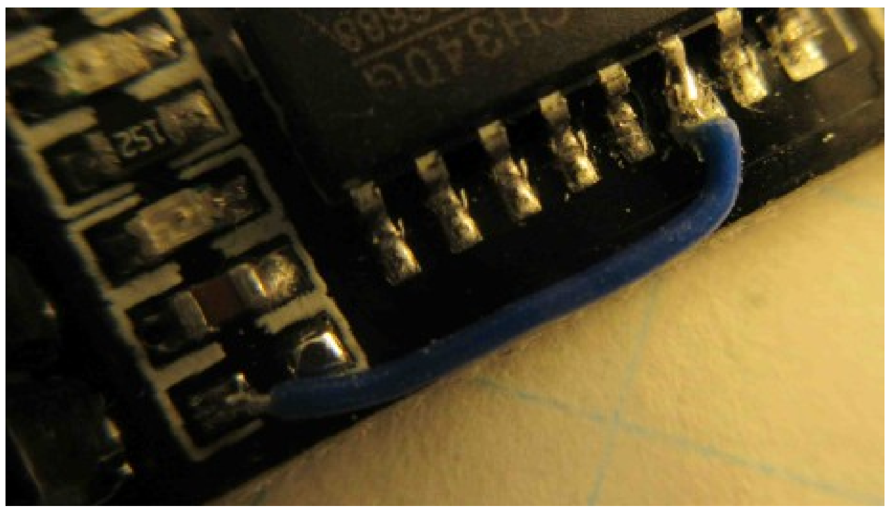
1. Remove C5.



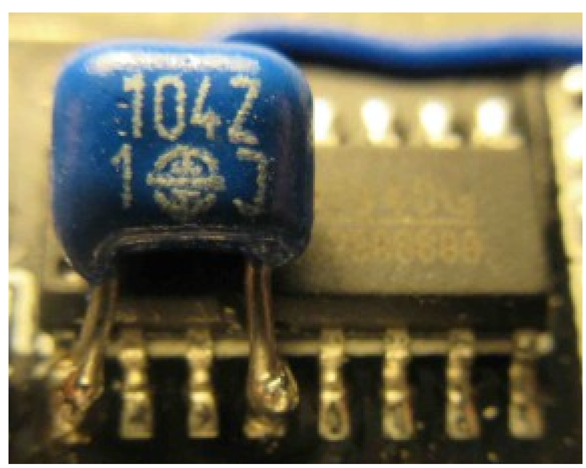
2. Cut Trace to 3V3 pin.



3. Add a jumper wire from CH340G pin 14 to the outer pad of C5.



4. Add a new 0.1uF decoupling capacitor between pin 1 and pin 4. (This replaces the one removed in step 1.)



Your terminal program's connect/disconnect commands should now control the level on the 3V3 pin of the USB adapter. When plugged into the 1802 or 6502 Badge, switching from “off-line” to “connect” will reset the Badge, and then let its program run.