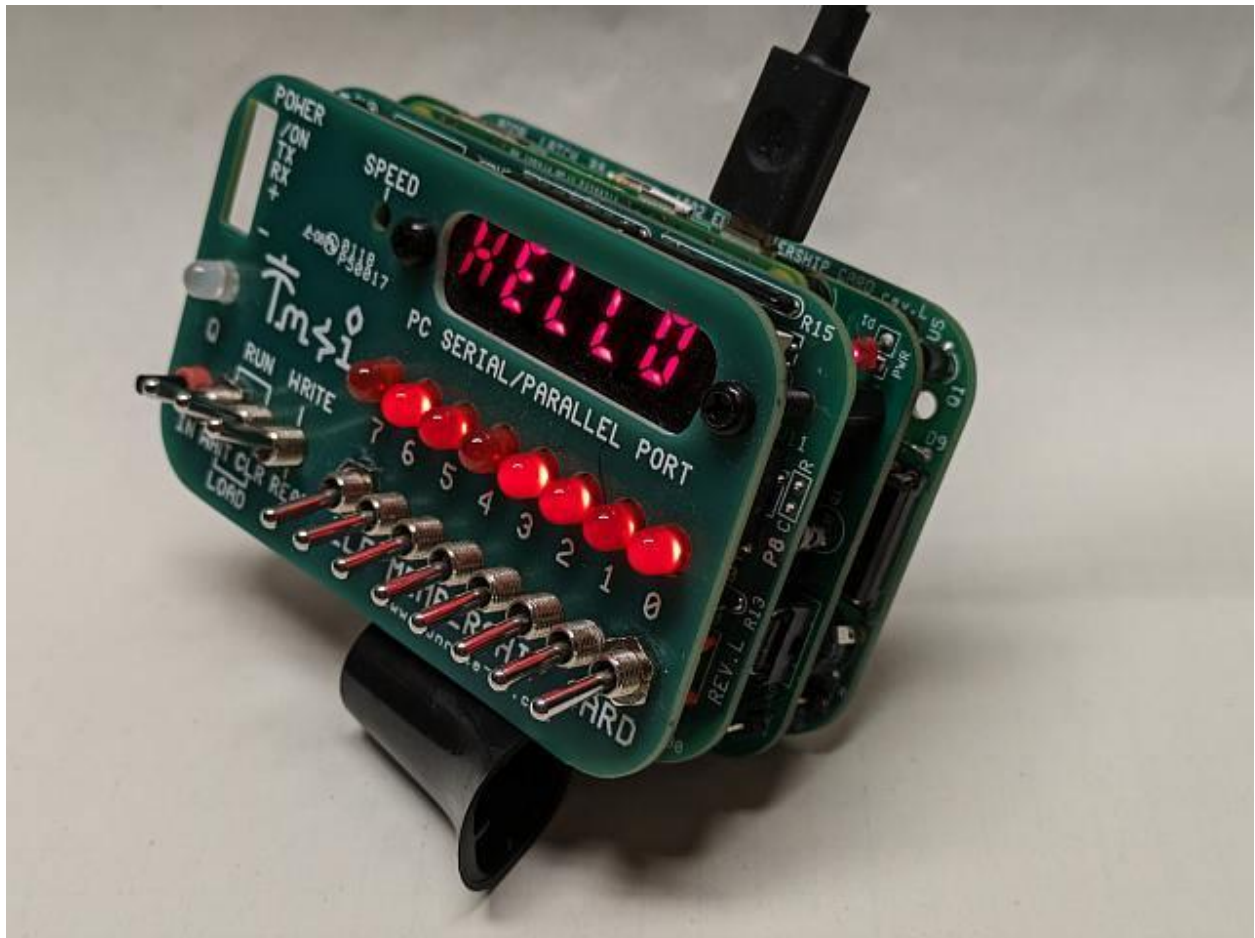


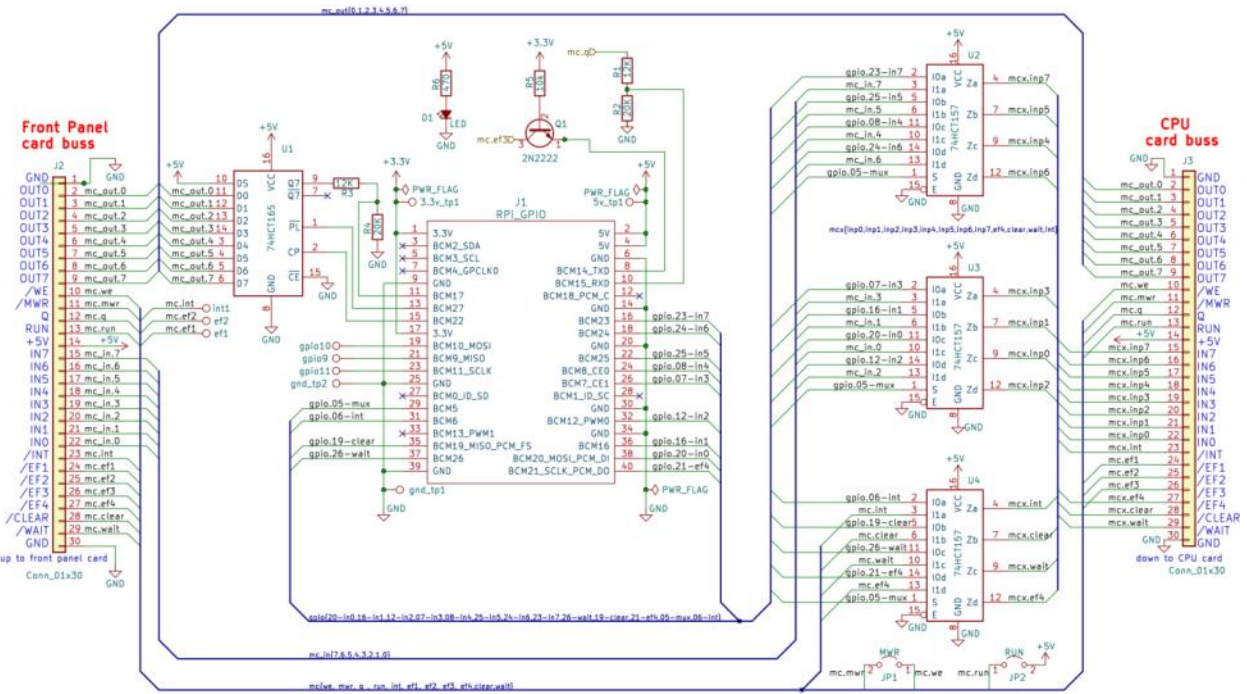
PiLoader Build Instructions

A PiZero Adapter for Lee Hart's Membership Card

BOM and Instruction Rev 2.0

Project Web Page : <http://anthonyhill.github.io>





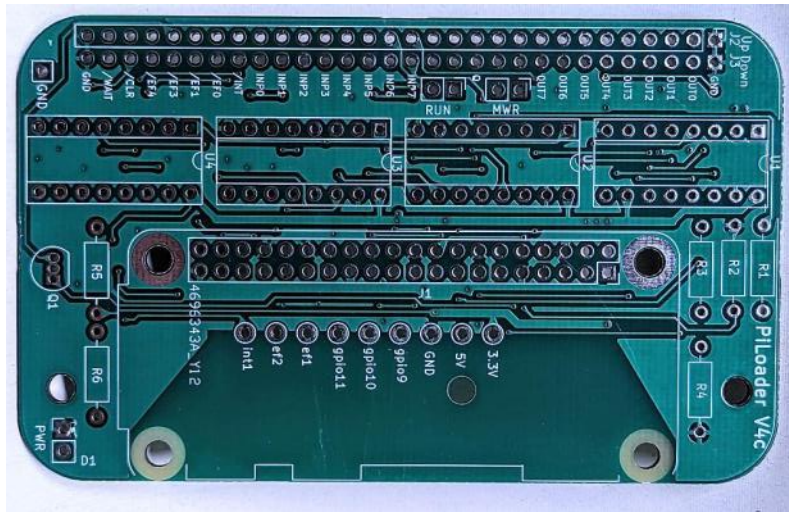
	Component	Manufacturers Part #	Description	Notes
1	J1	Adafruit part # 2243	Raspberry Pi GPIO female header	https://www.adafruit.com/product/2243
2	U1	74HCT165	8 bit shift register	must be HCT
3	U2, U3, U4	74HCT157	quad 2:1 mux	must be HCT
4	J2	1x30 pin header	0.1" spacing, 11.25mm height	up to front panel card
5	J3	Molex 22-18-2101	30 female connector	down to CPU card
6	Q1	2N2222	Small signal NPN transistor	could use 2N3094 or equivalent
7	R1,R3	12K ohm	Resistor – any wattage and precision	5V to 3.3V level shift
8	R2,R4	20K ohm	Resistor – any wattage and precision	5V to 3.3V level shift
9	R5	10K ohm	Resistor – any wattage and precision	3.3V to 5V level shift
10	R6	470 ohm 1/8 watt	Resistor – any wattage and precision	Optional – power LED current limit adjust value to change brightness
11	D1	T-1 $\frac{3}{4}$	LED Power indicator	Optional (with R6)
12	JP1, JP2	2.54mm 2 pin header	jumper	jumper for Front Panel Power Switch and Memory Protect
13	-	2x20 Pin Male Header	GPIO header for Raspberry Pi	see instructions before soldering
14	-	Nylon standoffs	Assorted 10mm M3 male & female	look for a Jucoan 380 piece kit
15	-	Raspberry Pi	PiZero, PiZeroW, PiZeroW2	without header pins preinstalled – see Note 2 in assembly instructions.

Assembly Instructions

1. Read & follow all construction recommendations in the Membership Card manual (see note 1).
2. The vertical spacing between the three PCB's and the Raspberry PI is very tight. Do trial assemblies to assess fit prior to soldering to make sure the connectors you are using will allow everything to fit and not touch or otherwise short out.
3. Attach a 2x20 pin header to Raspberry PI (see note 2).
4. Solder the 30 pin male header (J2) to the PiLoader PCB (see note 3).
5. Solder the 2 pin headers JP1 and JP2 to PiLoader PCB
6. Solder the three 10 pin female headers to the bottom of the PiLoader PCB (see note 3).
7. Solder four 16 pin DIP sockets for U1 to U4 to PiLoader PCB (see note 4)
8. Solder J1 to PiLoader PCB.
9. Solder R1 to R5 to PiLoader PCB.
10. Solder Q1 to the PiLoader PCB.
11. Optional: solder D1 & R6 to PiLoader PCB if you want a power indicator.
12. Insert U1 to U4 into their sockets (if you did not solder them directly to the PiLoader PCB)
13. Connect the Raspberry Pi Zero to J1 ("upside down")
14. (Optional) Attach a jumper to the MWR pin header (beside U2) if you want to bypass the front panel Read/Write switch and always use Write functionality.
15. (Optional) Attach a jumper to the RUN pin header (beside U2) if you want to bypass the front panel /ON power switch function and a let the MC processor card always run whenever power is connected. Note that on front panel revisions L and later you need to also jumper /ON to ground on P4 to enable the data LEDs and 7 segment displays.
16. Assemble the PiLoader PCB in between the MC Front Panel and CPU card. Nylon 10mm M3 standoffs are useful here to secure the three PCB's together.
17. Plug a 5V USB power supply (wall wart) with a microUSB connector into the USB port closest to the PiLoader's power LED (see photo below).

Notes:

1. Follow these assembly instructions in order. It will make clearances for soldering easier.
2. To ensure that the Raspberry Pi fits between the PiLoader card and the Front Panel card, the GPIO 2x20 pin header soldered to the Raspberry Pi should be inserted from the rear so that only 6 mm extends in front. Trim the plastic carrier and excess pin flush with the back of the Raspberry Pi. Note that his technique is described in recent Membership Card instructions (rev K or later) as "**B. The HARD way:**"
3. Follow the recent (rev K or later) Membership Card assembly instructions for installing card interconnection header pins and sockets. Use "**A. The EASY way:**" instructions for the male pins - do **NOT** install the male pins by following the "**B. The HARD way:**" instructions. Note that it does not matter which instructions you followed when you built your CPU card.
4. If you are brave, solder U1 to U4 directly to card. The height doesn't matter either way. I prefer sockets for testing and troubleshooting and have never had a problem using the cheap ones. But I'm very careful with how I lead form my IC's and how I insert them into a socket and then check for bent pins. YMMV.



Software

Raspberry PI Operating System

There are many options for Raspberry Pi operating systems but the recommendation for the PiLoader is Raspberry Pi OS Lite (32-bit). The Lite version does not include a desktop environment (GUI) as it tends to be slow running on a Raspberry Pi Zero.

Raspberry PI Program to Membership Card Interface Programs

There are two simple support programs written in standard C provided. The first is a program for loading assembled code binary files from the Pi to the 1802's memory. The second program is a demonstration of how to read the data displayed on the 1802 Front Panel card LED's from the Pi.

In addition, there are few small utility programs for manipulating the GPIO pins and some 1802 example code.

Note : for reference there are currently at least seven libraries available to access GPIO pins from a C program running on a Raspberry PI. See https://elinux.org/RPi_GPIO_Code_Samples

Building the Interface Programs

Currently the PIGPIO library is used for accessing the PI GPIOs by the Loader code.

You will probably need to install the package :

```
➤ sudo apt install pigpio
```

To compile, use

```
➤ cc -Wall -o 1802load 1802load.c -lpigpio
➤ cc -Wall -o 1802scan 1802scan.c -lpigpio
```

To run, use:

```
➤ sudo ./1802load [-h -x -g] <binary file> [load_address] [execute_address]
    - there are various options available - see the README.TXT file
➤ sudo ./1802scan
```

Notes :

1. There is a small **README.TXT** file in the software directory that describes what's in that folder and how to use it.
2. Do not cut and paste the above commands from this PDF. The “-“ sign in the compile commands gets converted to some other symbol that won't work. Re-type them.