

ZPI

Retrofitting a broken Zeos Notebook 386 with a raspberry pi and custom mainboard

- [DEV - Development Prototypes](#)
 - [DEV KBD+RTC](#)
- [Modules](#)
 - [KBD - Keyboard](#)
 - [RTC - Realtime Clock](#)
 - [PWR - Power Supply](#)
 - [CPU - Computer](#)
 - [DSK - Floppy Disk Drive](#)
 - [LCD - Built-in Display](#)
 - [VGA - Video Output](#)
 - [PCI - Parallel and Serial Ports](#)
 - [MDM - Bluetooth Modem](#)
 - [USB - USB Hub and Devices](#)

DEV - Development Prototypes

DEV - Development Prototypes

DEV KBD+RTC

Modules

Planned modules for the main board, to replicate all of the functionality of the original device.

KBD - Keyboard

Duties

Convert the stock keyboard into USB

Parts

- Teensy 4.1
- Original Keyboard

References

Keyboard Matrix

@ Connector	Top Down
B D F H J	2 4 6 8 X
A C E G I K	1 3 5 7 X X
-----	-----
1 3 5 7 X X	A C E G I K
2 4 6 8 X	B D F H J

	A	B	C	D	E	F	G	H	I	J	K
1	` ~	F1	F3	5 %	F5	7 &	F10	F8	Rshft	LArr	-
2	Esc	LCtrl	F2	F4	Spc	F6	F9	F7	\	DArr	-
3	1 !	Fn	3 #	4 \$	6 ^	8 *	9 (Rctl	-	UArr	-
4	Tab	Z	LAlt	T	Ent	U	0)	RAlt	+ =	N	Del
5	Fn?	LShft	E	R	M	I	O	[{	' "	RArr	-
6	Cap	2 @	D	F	B	. >	P	F11	- _	Num	Scl
7	Q	W	X	G	Y	K	; :	H	F12	Back	Ins
8	A	S	C	V	J	L	/ ?	, <] }	PrSc	Pas

RTC - Realtime Clock

Duties

- Emulate a DS1307 real time clock chip
- Optionally implement extra ram set/recall
- Optionally implement DS3231 alarms, temperature

Parts

- Reuse the Teensy 4.1 that's handling Keyboard input
- CR1220 Battery (recycled from 380XD repair)

PWR - Power Supply

Duties

- Provide +5v power to the computer
- Provide charging current for the Ni-Cd battery
- LED outputs for Power on, Battery Present, Charging, Charge complete

Parts

- LM2576 (TI?, 3A?) [lm2576hv.pdf](#)
- ATtiny84 Ni-Cd Monitor [ATtiny24A-44A-84A-DataSheet-DS40002269A.pdf](#)
- Based on Freescale MC68HC908QY4A Example [AN2679.pdf](#) [AN3392.pdf](#)

Modules

CPU - Computer

Duties

Actually handle the computer parts of the computer

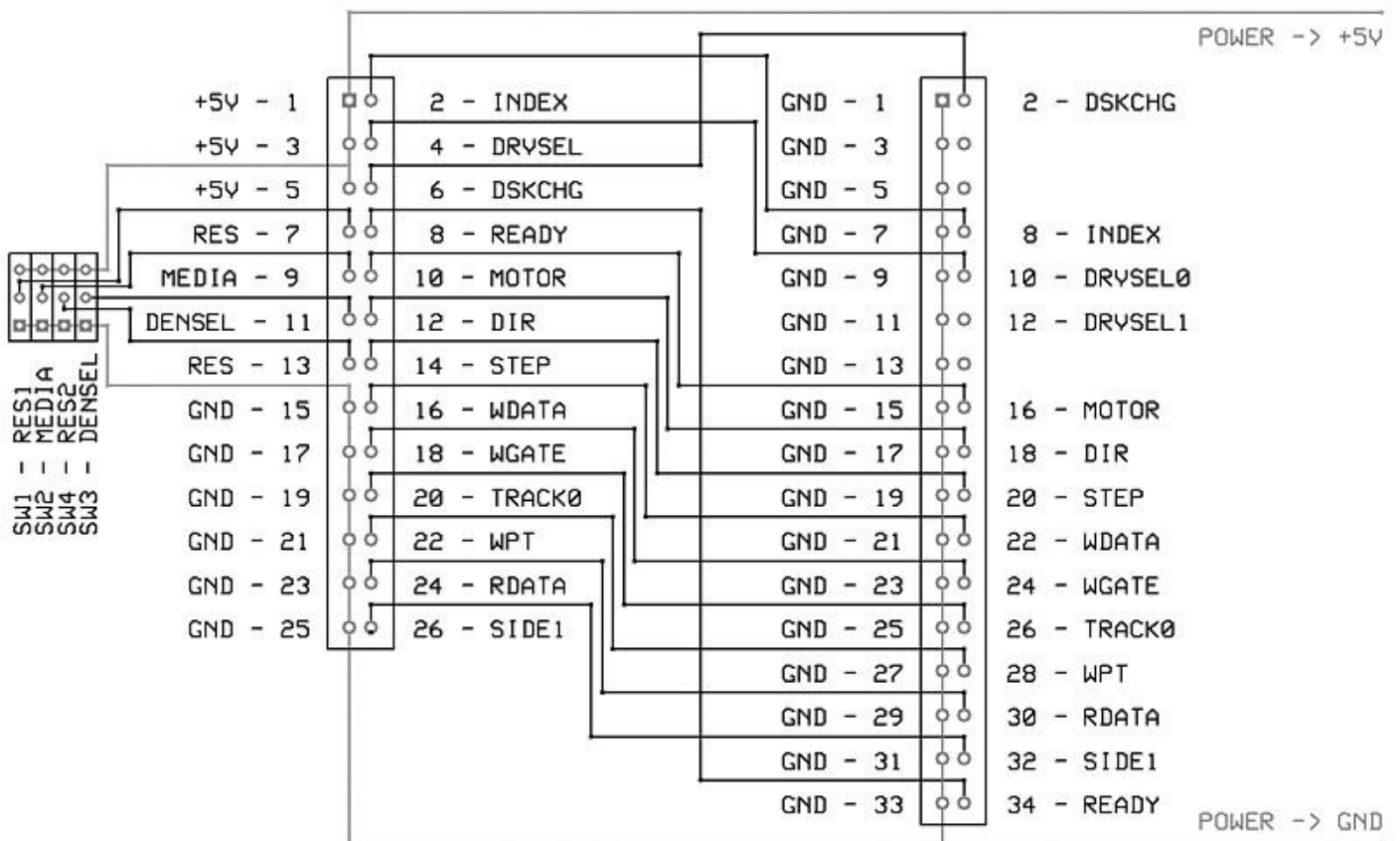
Parts

Either (Depending on need for HDMI and PCI)

- Raspberry Pi Compute Module 3
- Raspberry Pi Compute Module 4

DSK - Floppy Disk Drive

Pin Nos.	Signals	Pin Nos.	Signals
1	+5V	2	INDEX
3	+5V	4	DRIVE SELECT
5	+5V	6	DISK CHANGE
7	NC	8	READY
9	HD OUT (HD at HIGH level)	10	MOTOR ON
11	NC	12	DIRECTION SELECT
13	NC	14	STEP
15	0V	16	WRITE DATA
17	0V	18	WRITE GATE
19	NC	20	TRACK 00
21	NC	22	WRITE PROTECT
23	0V	24	READ DATA
25	0V	26	SIDE ONE SELECT



```

#elif defined(ARDUINO_RASPBERRY_PI_PICO)
#define DENSITY_PIN 2 // IDC 2
#define INDEX_PIN 3 // IDC 8
#define SELECT_PIN 4 // IDC 12
#define MOTOR_PIN 5 // IDC 16
#define DIR_PIN 6 // IDC 18
#define STEP_PIN 7 // IDC 20
#define WRDATA_PIN 8 // IDC 22 (not used during read)
#define WRGATE_PIN 9 // IDC 24 (not used during read)
#define TRK0_PIN 10 // IDC 26
#define PROT_PIN 11 // IDC 28
#define READ_PIN 12 // IDC 30
#define SIDE_PIN 13 // IDC 32
#define READY_PIN 14 // IDC 34
#endif USE_TINYUSB

```

Duties

Map the physical floppy drive to usb

Parts

- TEAC FD-335HF 26-pin floppy drive
- Arduino Compatible Microcontroller

Drive References

Pin	Signal	Pin	Signal
1	+5V	2	Index
3	+5V	2	Drive Select
5	+5V	2	Disk Change
7	NC	2	Ready
9	HD OUT	2	Motor On
11	NC	2	Direction Select
13	NC	2	Step
15	GND	2	Write Data
17	GND	2	Write Gate
19	NC	2	Track 00

Pin	Signal	Pin	Signal
21	NC	2	Write Protect
23	GND	2	Read Data
25	GND	2	Side One Select

LCD - Built-in Display

Path One: Keep the current LCD

Duties

If recycling the current LCD, will need to ensure proper power is available and do one of two things:

- Convert the video signal from HDMI to 4-bit monochrome LCD signal
- Write a driver that is similar in structure to SPI/I2C framebuffer to output compatible signal

Pros

- Less hardware to purchase
- Keep original aesthetic

Cons

- Difficult to get awkward and high voltages to run CFL backlight
- Have to reverse engineer old 4-bit signal standard (Documentation is minimal, but looks to be comparable to VGA)
- Writing a *god damn* video driver

Path Two: Use a new LCD

Duties

If using a new off-the-shelf LCD, will need to find something that will fit inside of the current case.

- Current screen: 4:3 ratio, 9.5" diagonal, 7.6" width, 5.7" height
- 16:9 screen: 11.6" diagonal, 10.1" width, 5.7" height
- 16:10 screen: 10.8" diagonal, 9.1" width, 5.7" height

Pros

- Much easier to just use HDMI/VGA and 12v power
- Brighter, easier to see
- Color!

Cons

- Damn near *impossible* to find modern LCDs that are the proper size
- Will need to make custom mounting hardware for the screen
- Price of a new LCD is *slightly* more than I'd like to spend

Modules

VGA - Video Output

PCI - Parallel and Serial Ports

Duties

Make available the original type of ports the original device had, in roughly the same positions

These ports consist of:

- 1 Parallel Port
- 1 External 9-pin Serial Port
- 1 Internal Serial Port, for the modem

Parts

While still undecided, some candidates are available.

- ASIX MCS9901 PCIe to Dual Serial and Single Parallel Controller
- Prolific PL2303GC USB to Full UART IC
- Prolific PL-2303HX USB-Serial Bridge
- WCH CH340 USB-Serial chip

MDM - Bluetooth Modem

Duties

Have a way to allow the computer to "dial in" to a network using bluetooth, and a OpenWRT "base station". Optionally, have the board be designed to use a similar connector, layout, and size of the original modem, so that this part can be used by others with compatible boards.

Parts

- HC-05 Bluetooth to Serial Port module

Modules

USB - USB Hub and Devices