### Ί Τ T $\bigcirc \bigcirc$ $\bigcirc$

### My Dot Matrix Display Clock

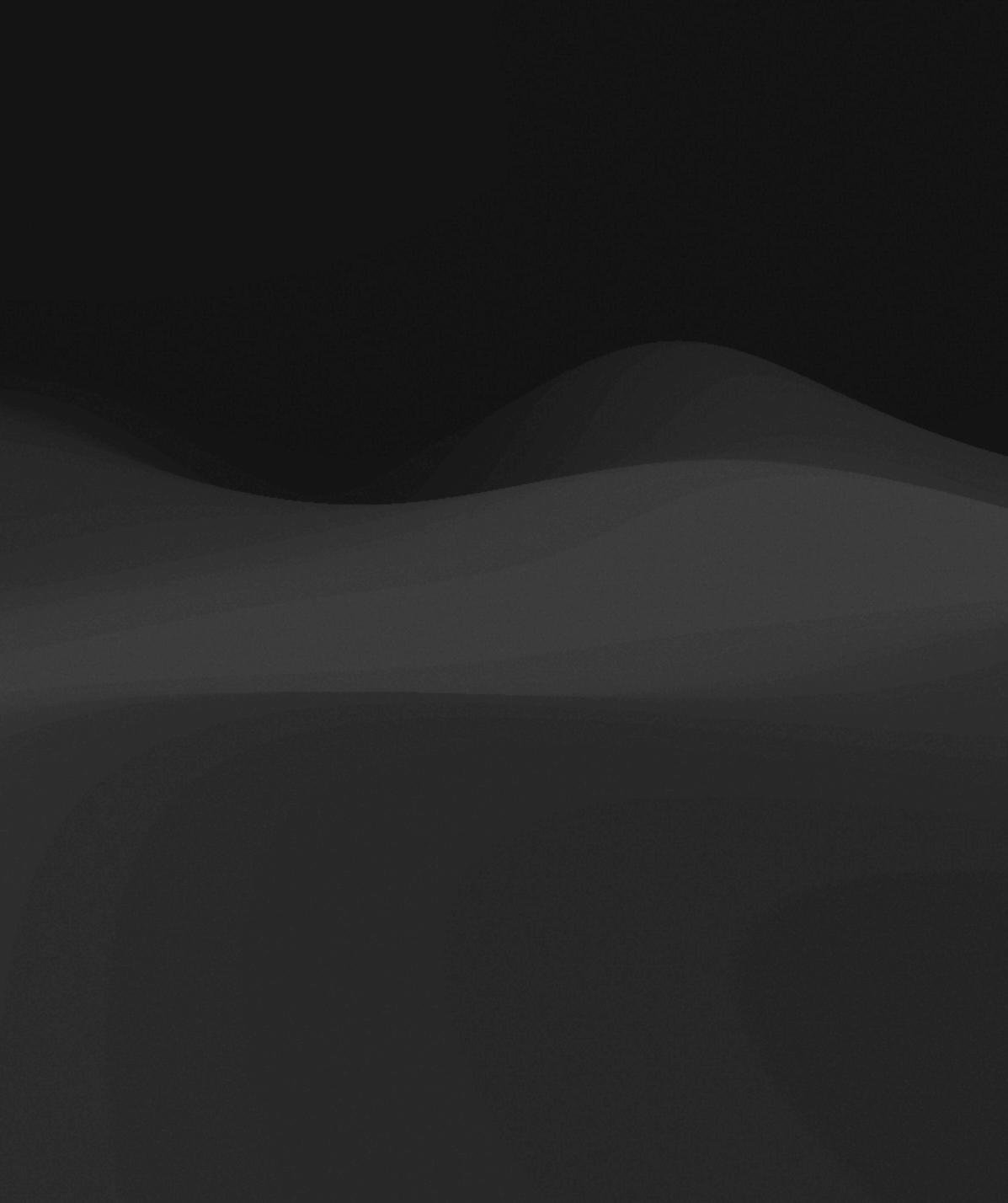
Chuck Rohs 2024-04

## Chuck Rohs

- Comp Sci from U of Calgary •
- Background in Compilers, Simulation, SCADA, Embedded, Security ullet
- Retired for almost five years
- Interest in embedded hardware, software and pinball •

# Agenda

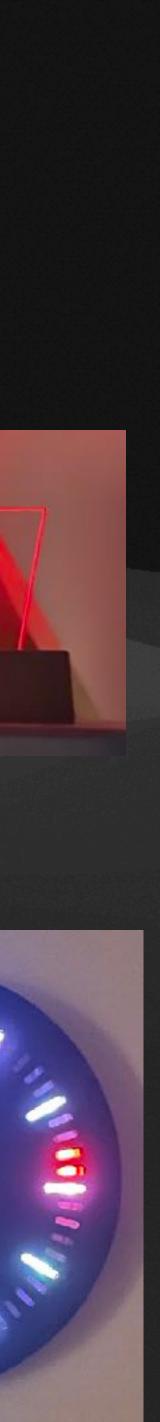
Clock Insanity Products Hub75 Interface Hardware Software Data Files Hub75 with esp8266



## Background

- I like clocks.
- It seems to be "geek" thing
- Examples: •
  - E-ink home assistant clock
  - Dot Matrix Heads up
  - DMD pinball clock
  - WS2812 strip clock  $\bigcirc$

15:18 Call In Market



## Background

- More Examples:
  - Unreadable WS 2812 strip clock
  - Pinball playfield clock
  - Seven segment home assistant display





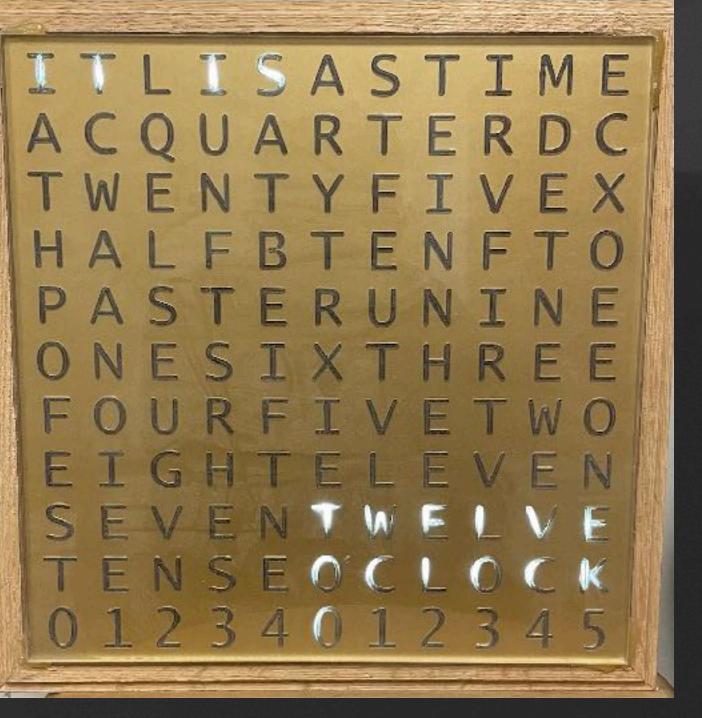
### JY-MCU JY-LKM1638 V12 C C JY-MCU JY-LKM1638 3868888888888888

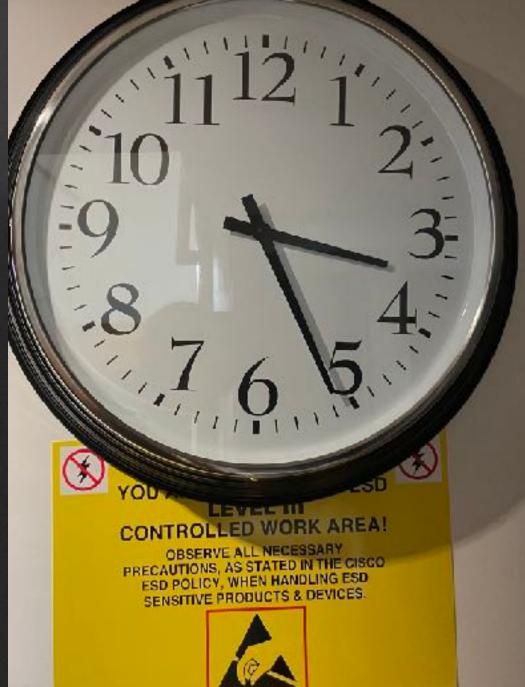


## Background

- More Examples:
  - Failed UV CNC clock
  - Word Clock
  - Persistence of vision clock, bought parts for, but never built
  - Normal clock, that is actually legible, to show me the time in the lab









### Overview

- 90s era pinball machines have 128 x 32 DM VF displays
- newer machine use led matrix displays
- There are a large number of animations made for this format of display

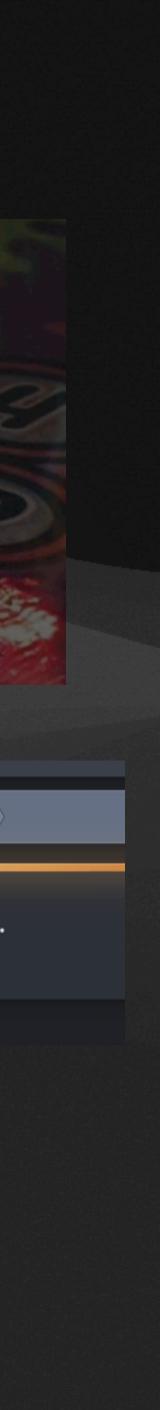


## History

- There are/were a number of these DMD clocks on the market
- Some have been shut down due to copyright violation
- Some "sort of" open source ones have come and gone <u>https://</u> <u>gitlab.com/modernhackerspace/</u> <u>dmdclock</u>

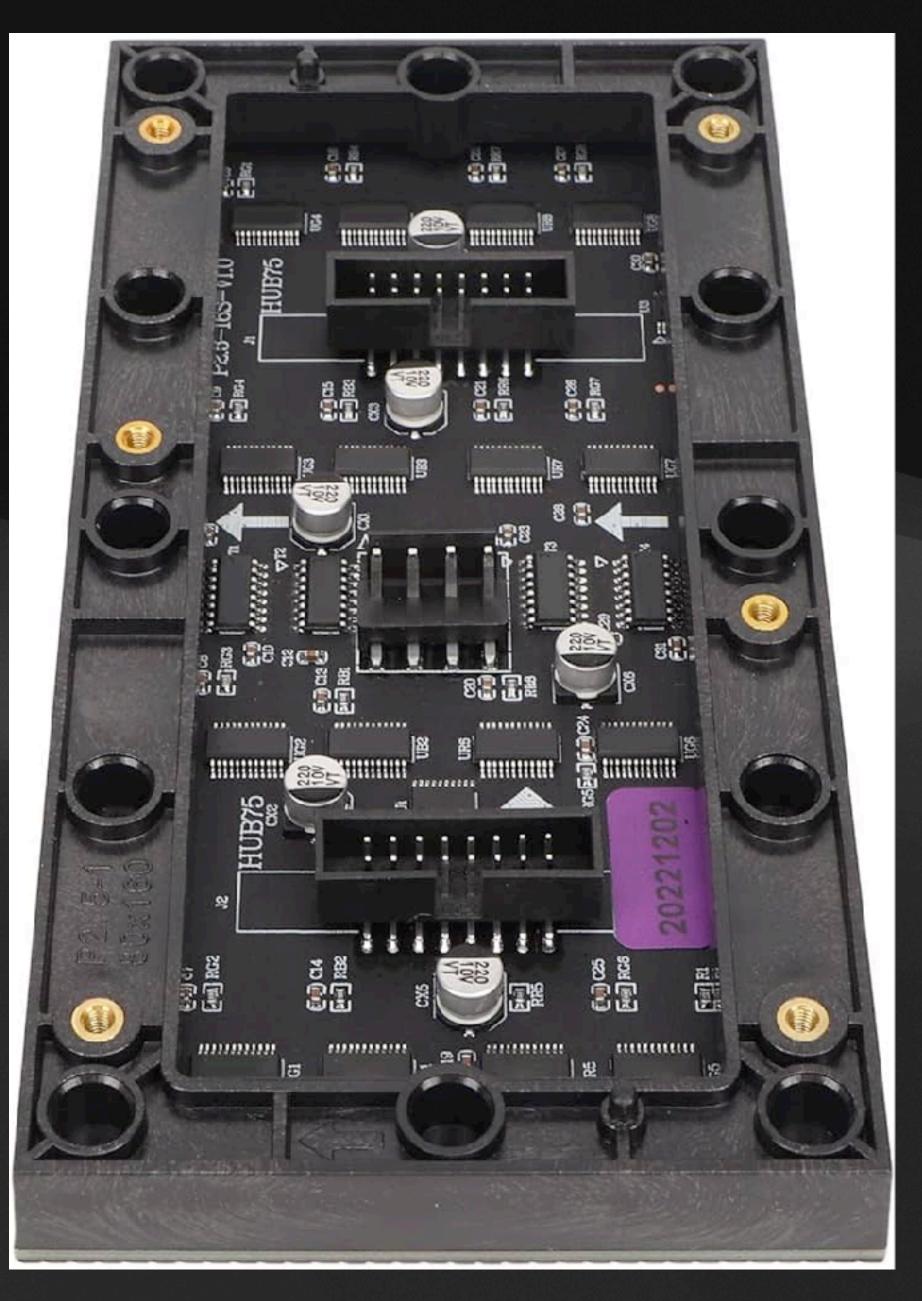
The Run-DMD shop is closed for the indefinite future. A big thank you to all customers for your support over the years. If we decide to produce more units in the future, the shop will reopen.





## History

- Initially I was going to buy one, but they are all in the \$500 range if you could find any.
- I was using these HUB75 displays in my pinball machine.
- I thought I'd build one.
- code to drive these from a raspberry pi has been around a while <u>https://github.com/hzeller/rpi-</u> rgb-led-matrix



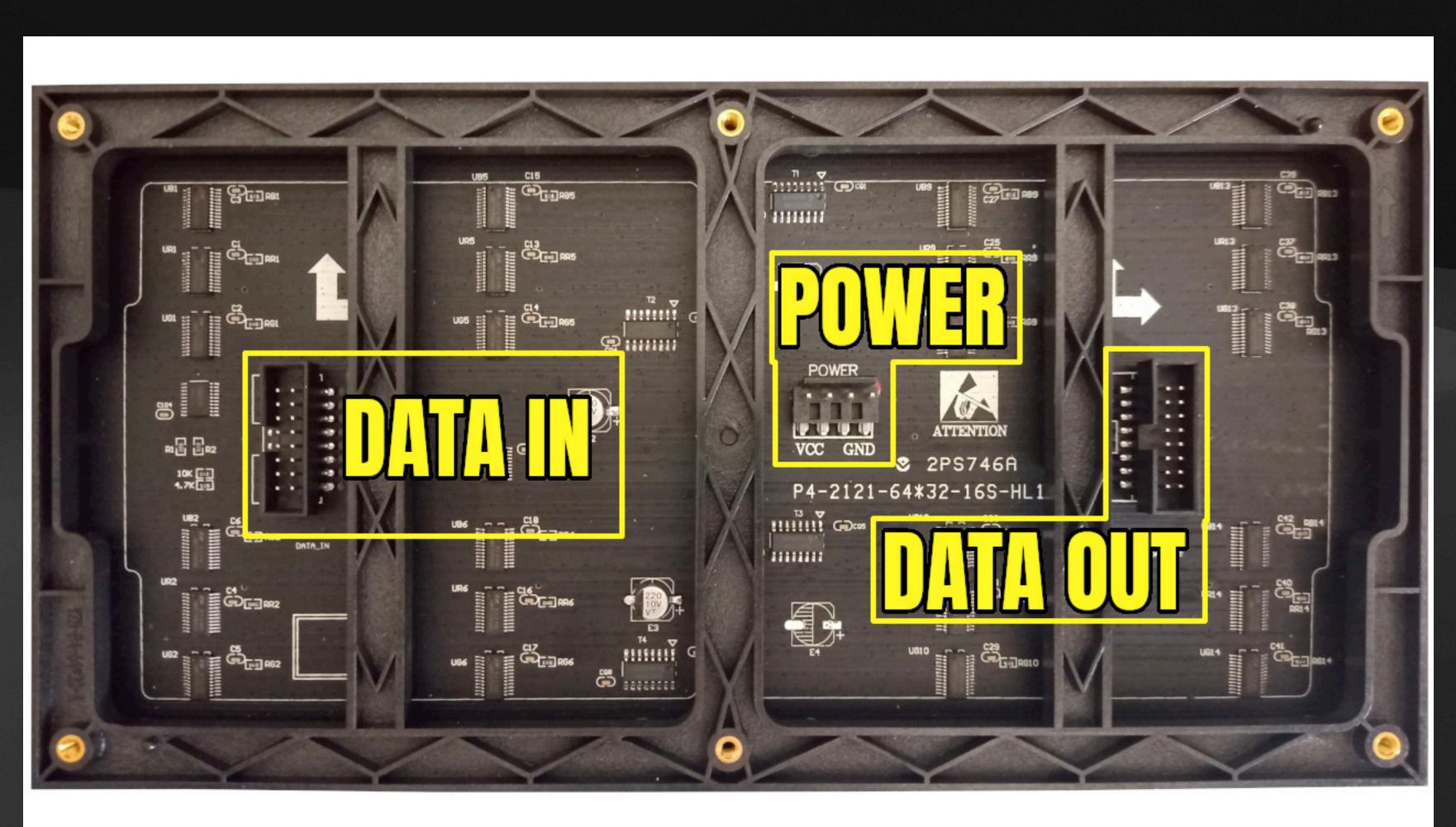
## Hub75 Panels

- Used in signage
- Come in various dimensions: 64X32 64x64 32x8
- Panels will daisy chain
- They come in various sizes (pitches)
- p2.5 is the "pinball size"
- p5 are twice as big
- "p" is the spacing in mm



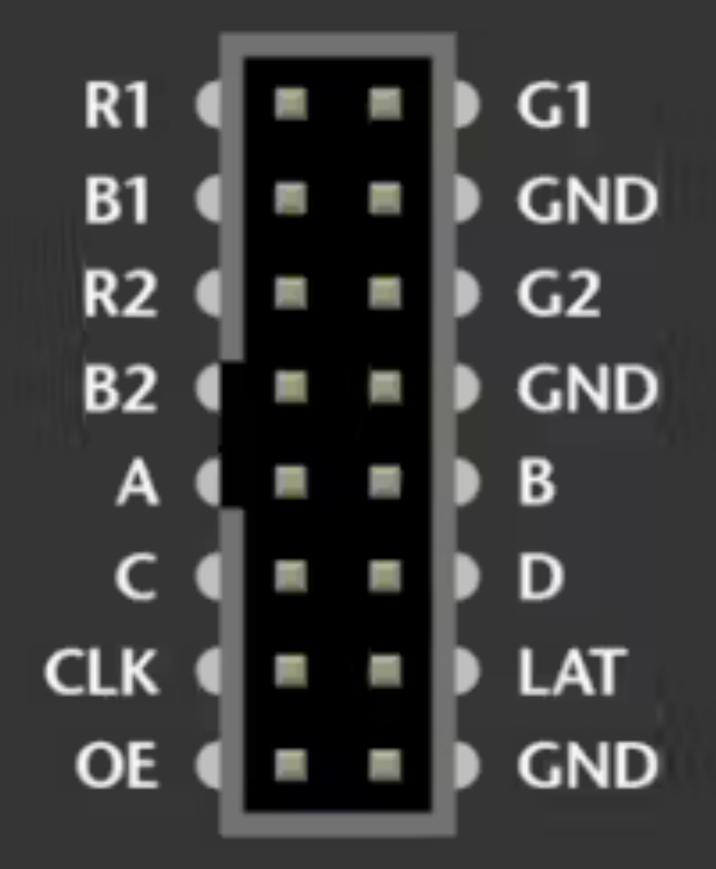
# Hub75 Panels

- Hub75 connections
- Panels can be chained together
- Separate 5V power can draw many amps



## Hub75 Panels

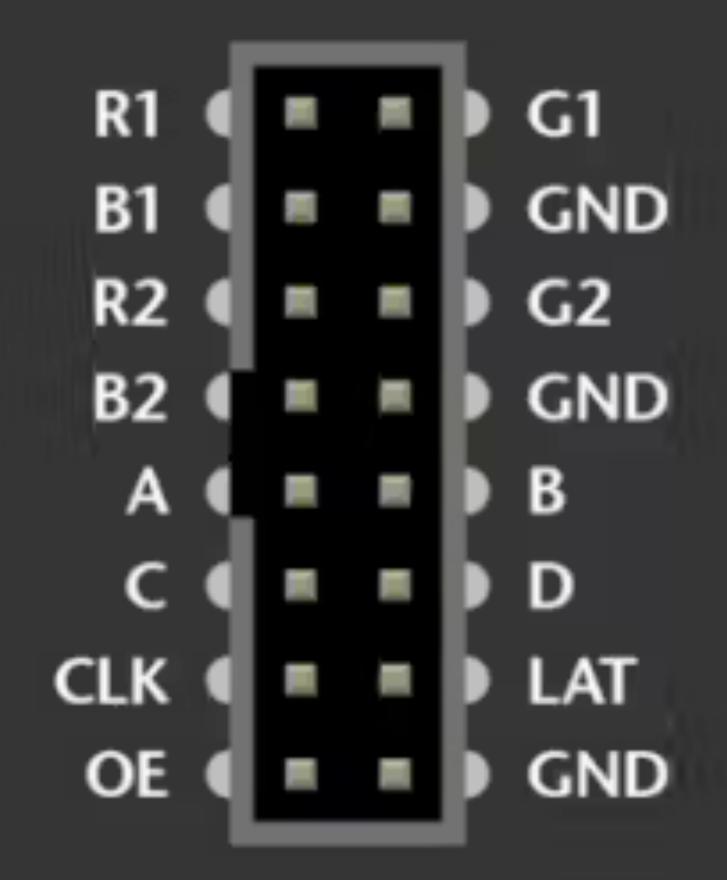
- Separate 5V power (in a separate connector)
- In a 32 line panel, only 2 lines display at a time, one in the top half and one in the bottom
- r1 g1 b1 are data for the top line, r2 g2 b2 are data for bottom line
- 16 addresses required in this case set by the four address lines: A,B,C,D
- CLK is the clock pin to clock in the serial R G B data
- OE is output enable, and LAT is Latch





## Hub75 Panels Continued

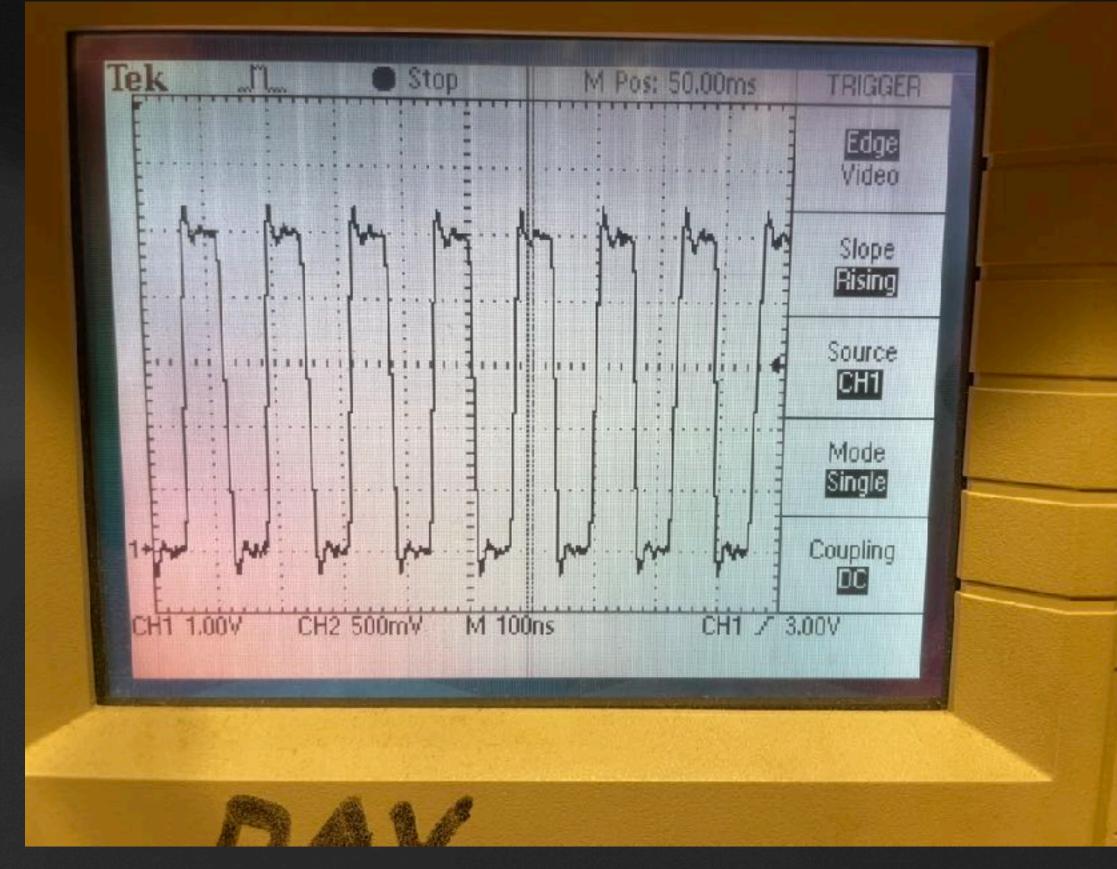
- Leds are on or off.
- To achieve any kind of intensity differences, it must be done using PWM in software.
- I.E. Super high frame rate.
- Eg 5Mhz / 2048 "pixels per field" = 2440 FPS 2400 / 30 = 80 updates per frame. ~6 bpp of color depth.
- Panel hardware can handle 75 MHz





### Hub75 Panels Continued

Scope on the pi pico clock





# Driving the Displays

- Lots of options for capable microcontrollers/driverboards: •
  - RaspberryPi lots of software for this •
  - Esp8266 works but not enough I/O for effective framerate (PWM)
  - Teensy + smartled shield more expensive works well
  - Pi Pico (Rp2040) works great

## Requirements:

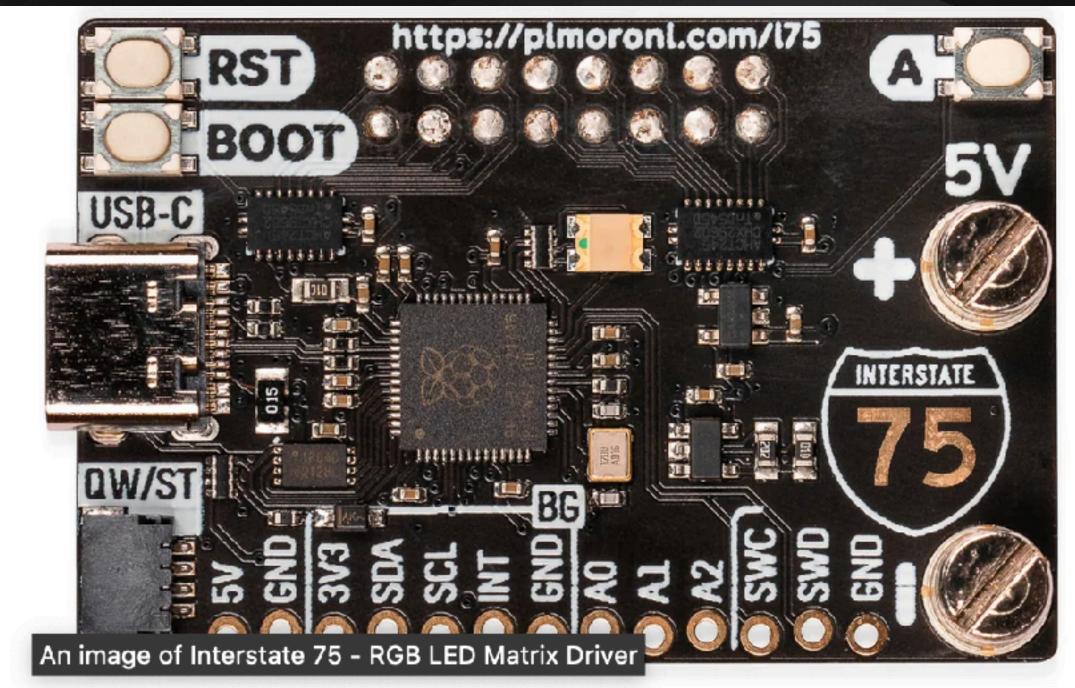
- machine. (not required for the clock)
- This really limited me to the Teensy and the rp2040
- PWM brightness.
- Also needed:
  - RTC
  - µSD
  - Buttons / rotary encoder for settings

One of the things I needed was processing data at 5 M Baud from my pinball

RP2040 has PIO that I wanted to play with, and allowed better control over

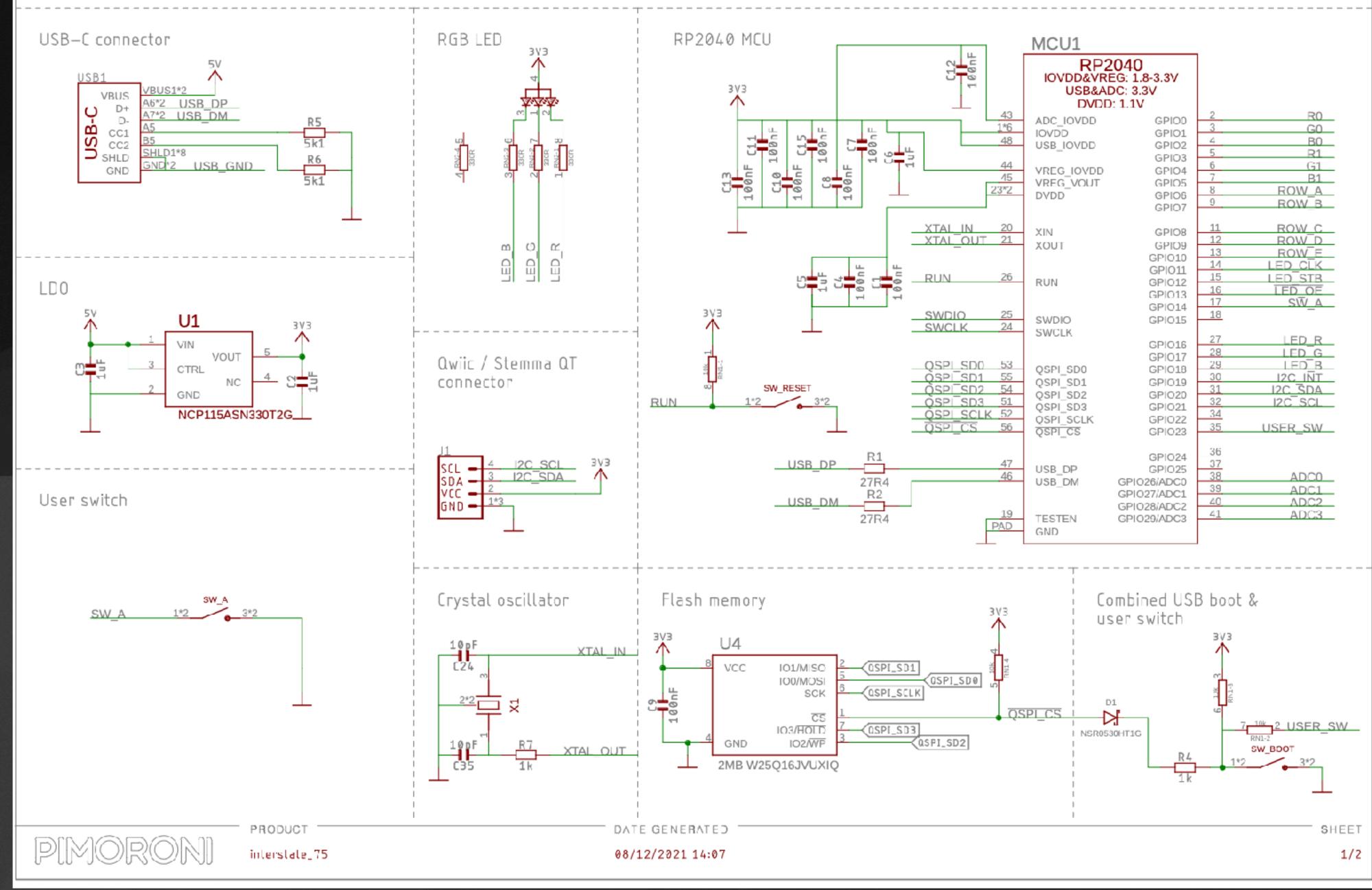
### Hardware:

- Pimoroni already had decent hardware and software for this purpose - interstate75
- This was a good starting point for the design



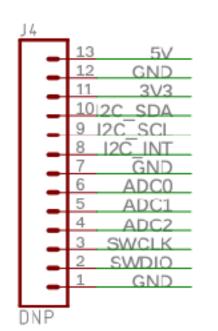


### Interstate 75 (HUB75 LED Matrix Driver), PIM584



### Interstate 75 (HUB75 LED Matrix Driver), PIM584

### Expansion header



5V screw terminals

P\$1

P\$1

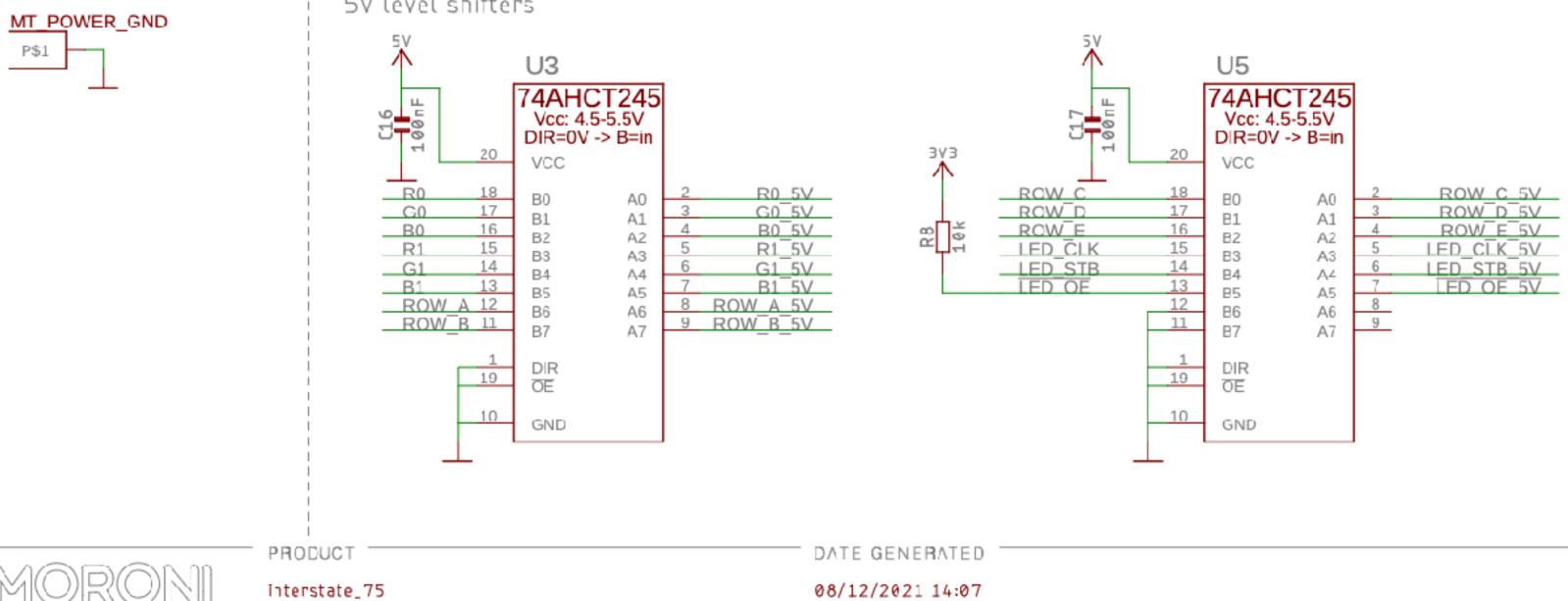
PIMO

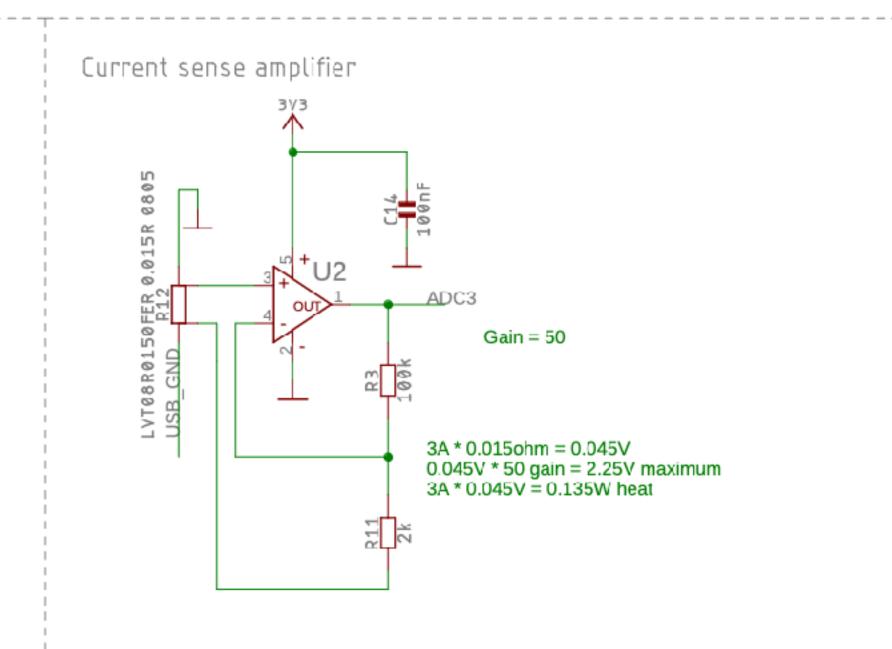
MT\_POWER\_5V A

LED Matrix header (top view of driver side connector)

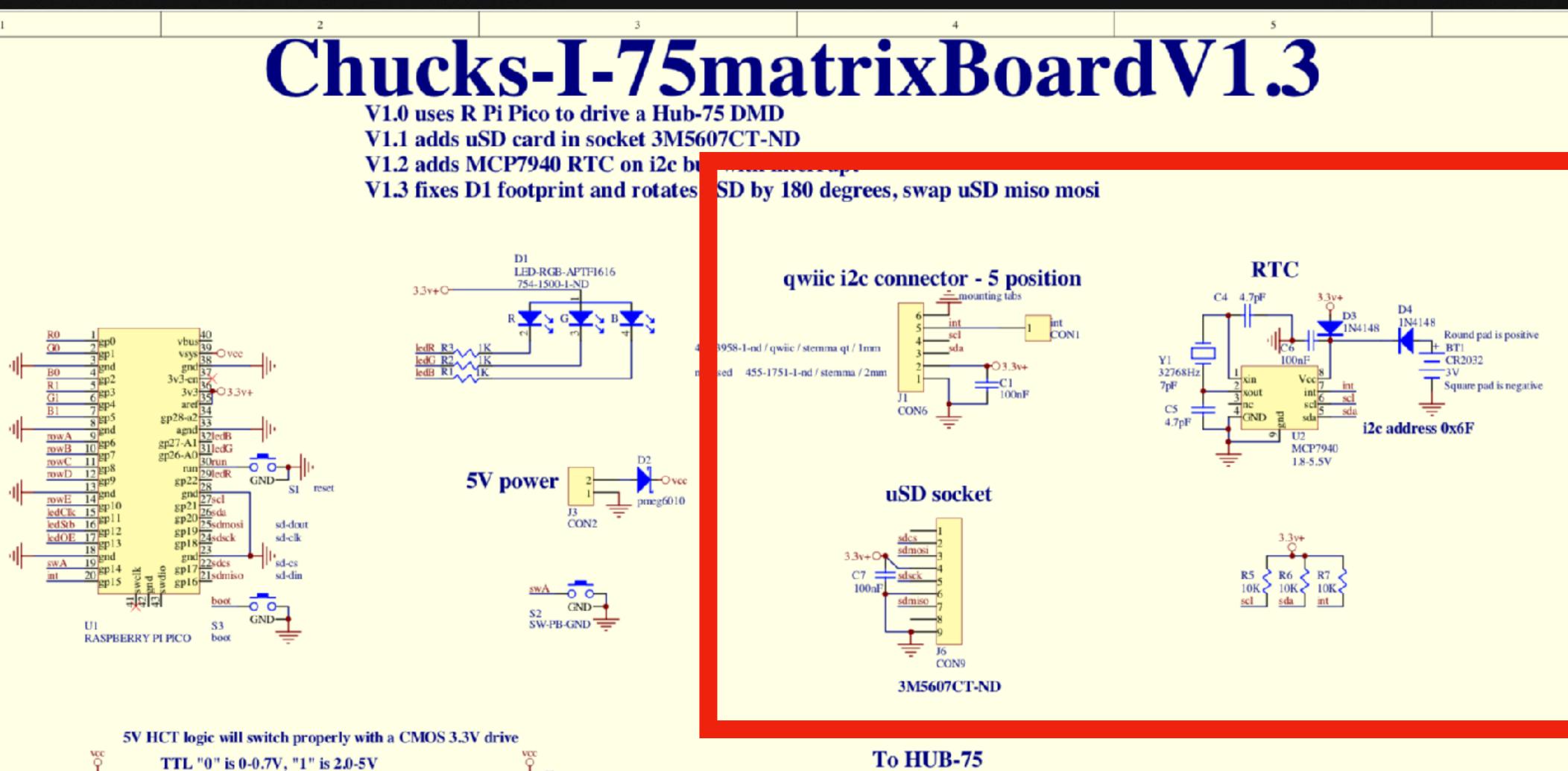
		J2		
LED OE 5V	1	_	2	GND
LED CLK 5V	3		4	LED STB 5V
ROW C 5V	5		6	ROW D 5V
ROW A 5V	7		8	ROW B 5V
B1 5V	g		10	ROW E 5V
R1 5V	11		12	G1_5V
B0_5V	13		14	GND
R0_5V	15		16	G0 5V
_		_		_

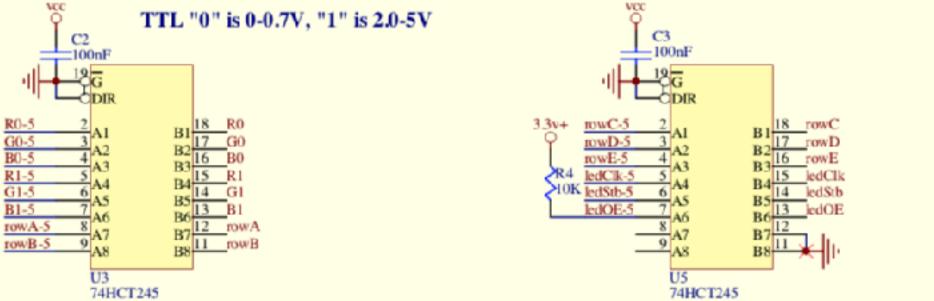




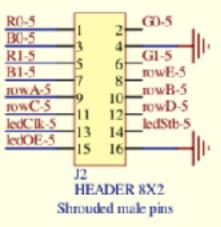


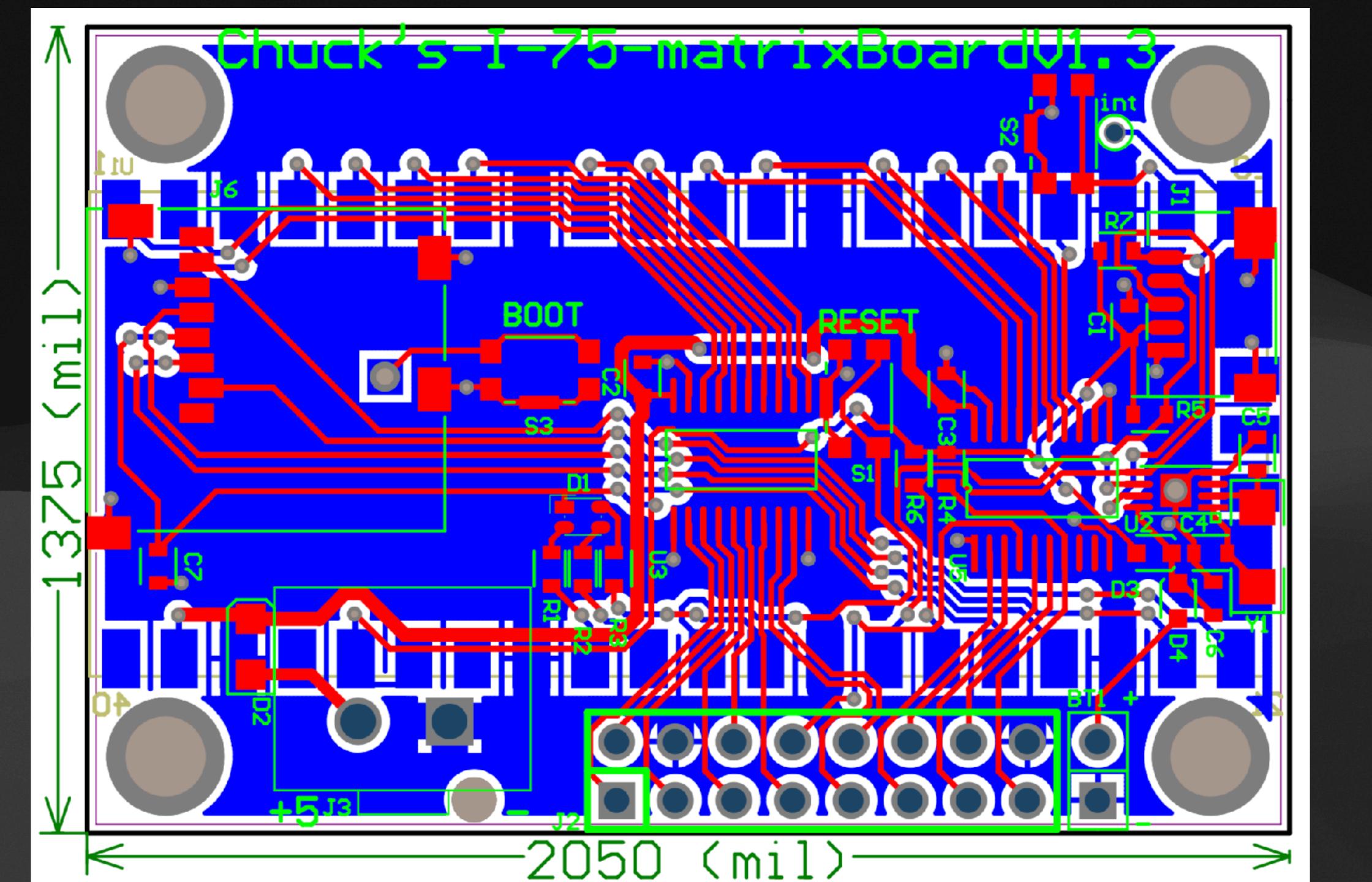
08/12/2021 14:07

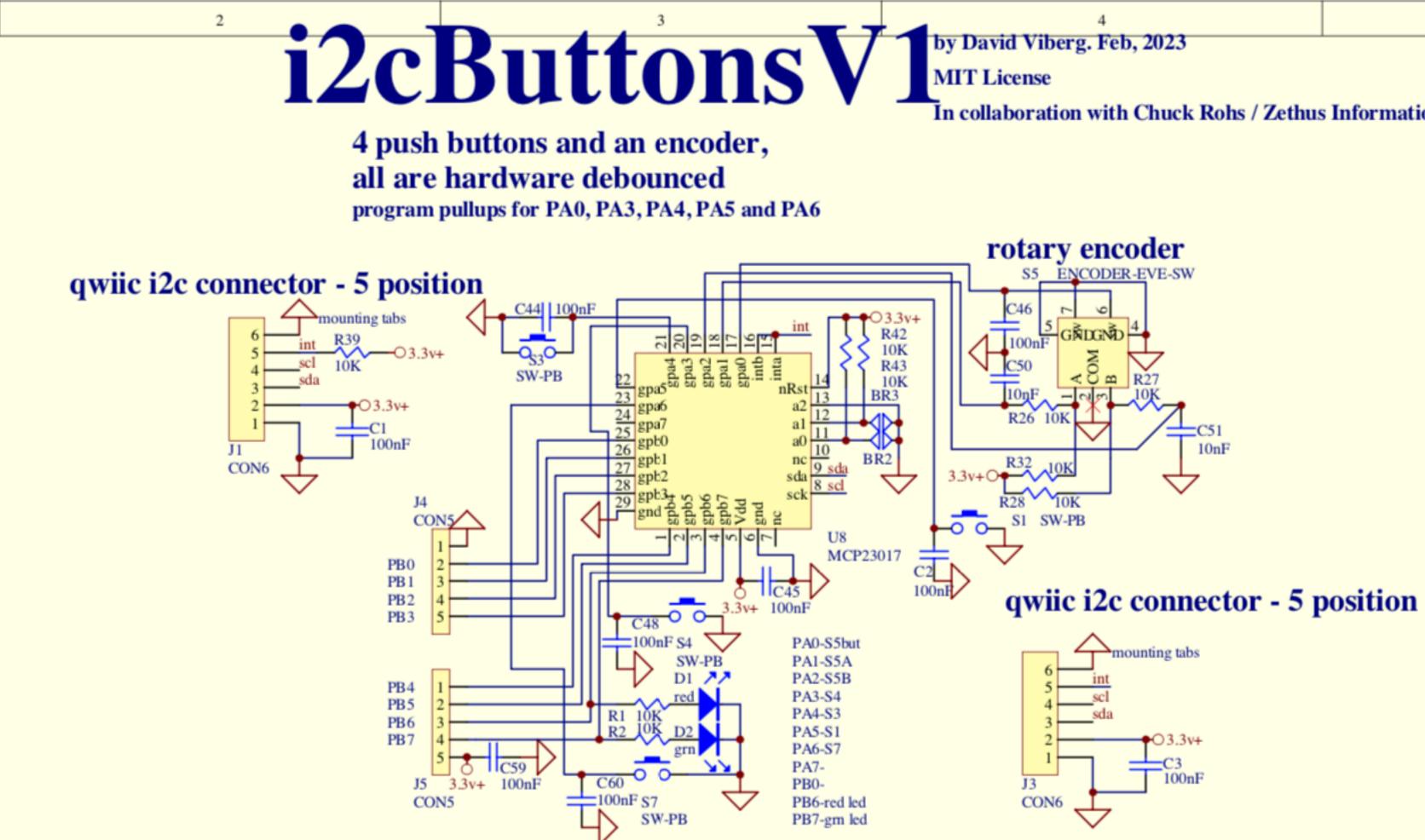


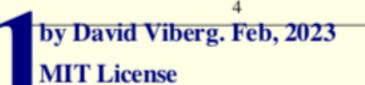


rowC-5



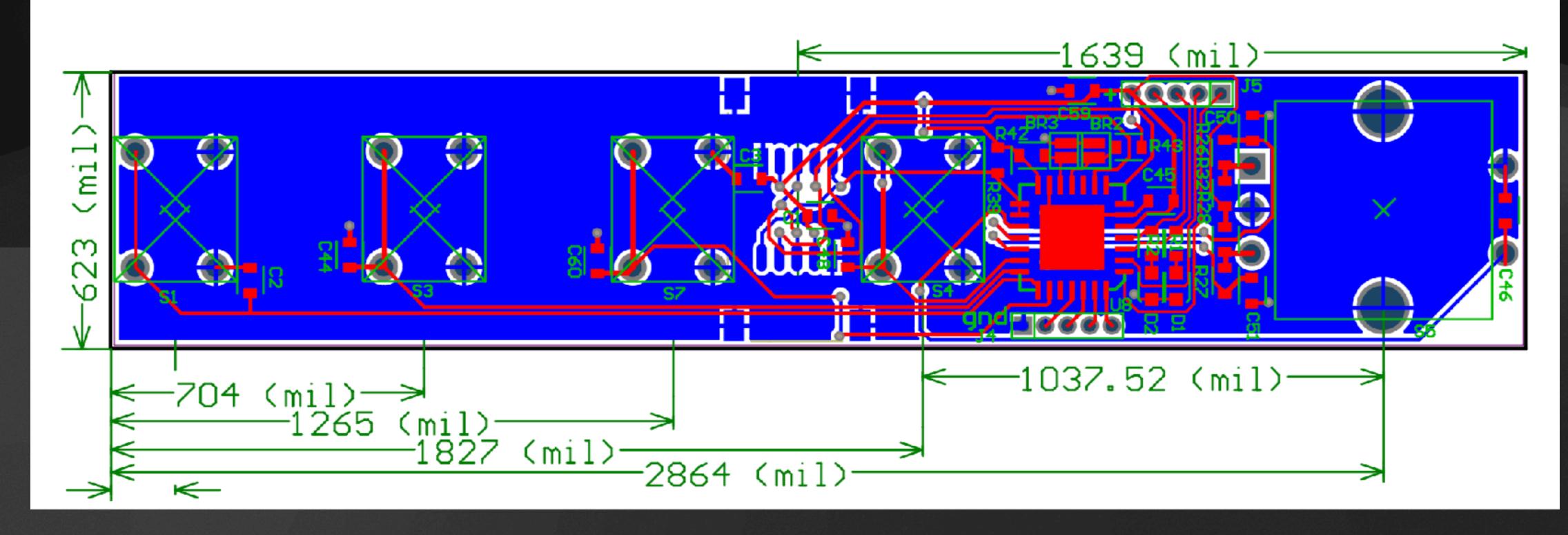


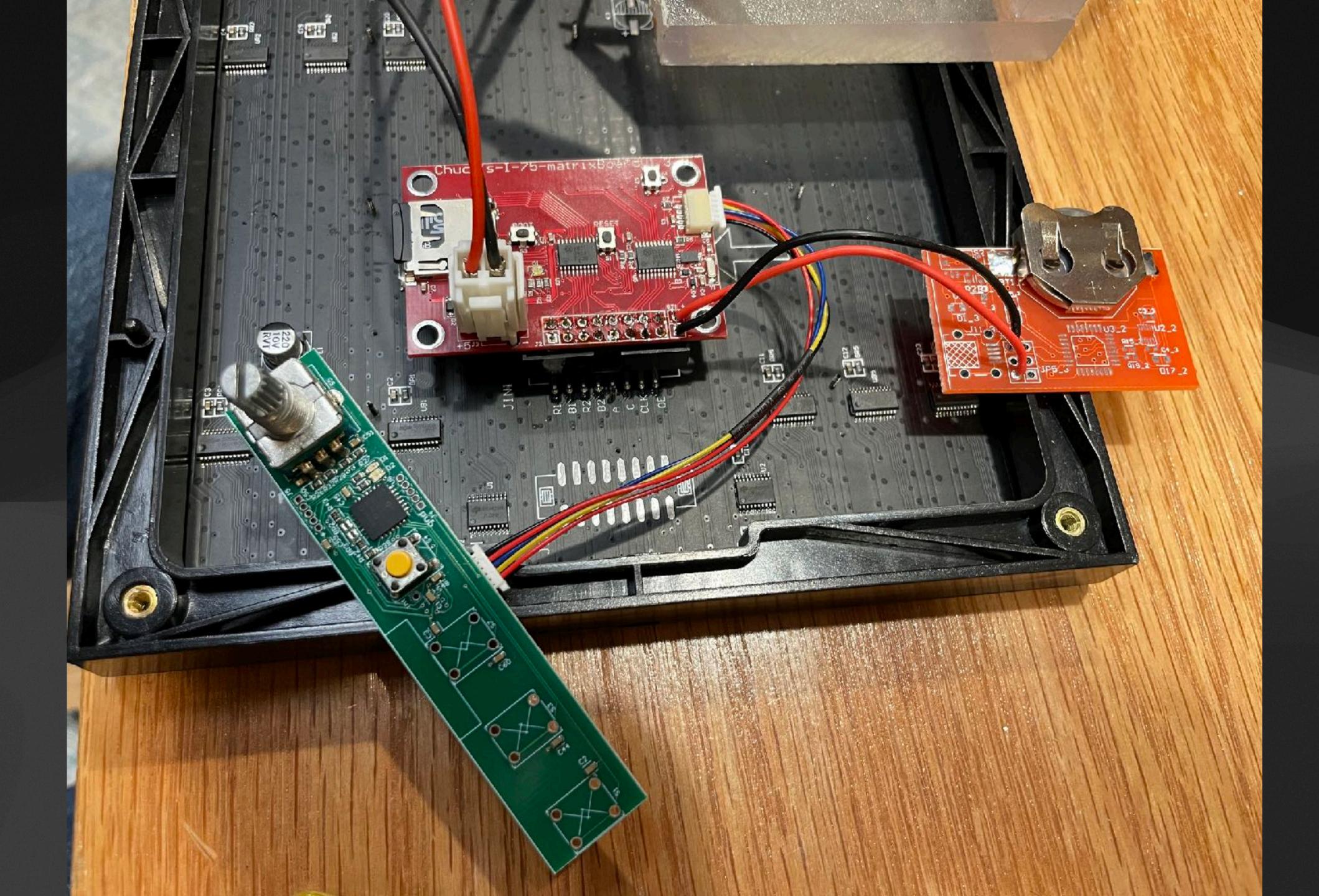


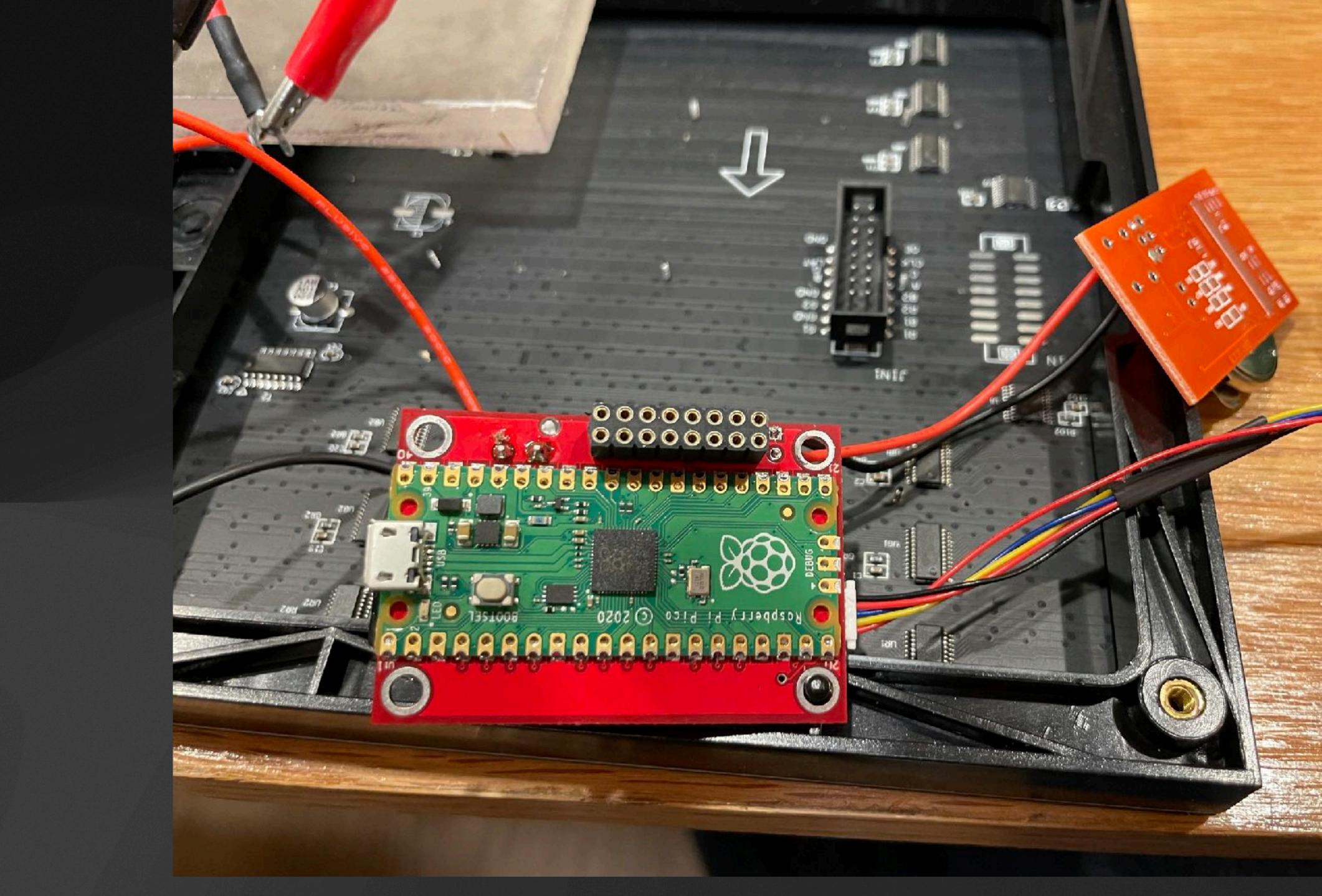


In collaboration with Chuck Rohs / Zethus Information Systems

5









### Software

- At a high level this software is a mashup of:
  - Teensey pixelmatrix code from mission pinball for serial port "framebuffer" streaming
  - Interstate 75 code for pi pico pio/hub75 driver
  - custom code for:
    - ROM image decode
    - Real Time Clock
    - Menus, Time Display, Second indicator, Font handling

## Dev Environment

- VSCode
- CMake
- appears as a mass storage device over USB.

• Programming is done by coping a "UF2" file to the pico in bootloader mode. It

- I investigated a few sources for these
- encoding
- can be turned on or off in the menu.

Original roms and other software images had proprietary run length

 The RUN-DMD image had "raw" images, but the frame sequencing and timing, needed to be reverse engineered, with a lot of "burn and learn"

Animations are broken down by machine, and each machine animation

 I spent lots of time in HexFiend trying to decode their Roms for my V1

·		<u> 1969 - 1989 - 1988 - 1986</u>		<u> - 1965 - 1967 - 196</u>									<u></u>	100000000000000000000000000000000000000	*****		XXXXX					
								80800800				run_dmd.										
	1088	0024669D DDDD840	02 20046666	65005DDD	46610001	66903000					00000000					00000000	fù»Ň	ffe ])¶a	fŭØ Vm	Ä	f}∕f@	&mP f
		GADDDDDD DDDDDD9		00404000		DDDD5001					00000000						1>>>>>	1 1 1 C	>>>P >>>		>>>>P	לאֿכר≠
	1216		07 2007D0DD	00204000				30300300			10030030	_	07DDDDDDC		DDDDDDDDD	DDA10300	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>				:>>>>/	so to
	1280		00 10090000	00202000				00000000		DDDDDDDDD		DDDDDDDØ	-		DDDDDDDD	DDDC1000	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>					
								00000000		DDDDDDDDD		DDDDDDDD			DDDDDDDDD	DDDDA000		89			and the second se	
	1344	the second se															0>>>>>>		>>>->>>			>>>>
	1408	ADDDDD38 Ø3DDDDD		D7600DDD				00000000				DDDDDDDD4			DDDDDDDD	DDDDD680	*>>0>>>	e 110 114	*****	-		· · · ·
	1472	ADDDDD13 01DDDDD		D4600DDD				00000000					_	600008DD	DDDC69DD	DDDDDD10	*>> >>>	·	>>>< >>>		;>>∆ù>>>−	· · · · ·
		ADDDDD03 03DDDDD			DDD806DD			30300300			DDD40039	and the second s				DDDDDD40	<b>≭&gt;&gt;</b> >>∢	: M>>" >>ÿ	>>>< >>>	5	;>> <u>†</u> ≠>>'	>>> 0M>
	1600	and the second se		DeceesDD				90990990						D90038DD	DD880032	DDDDDD60	waxa gara,	] >> >>/		5	;>>A ->>÷	→ YM>
	1664	ADDDDD5D DDDDDDA	\8 807DDDDD	C00007DD	DDDC06DD	DDDDDDD41						DODDDDDD	70000000				≠>>]>>>†	· }››¿ ››‹	>>>>A>>>	ç	;>>Ä >>ÿ	→>>>}
	1728	ADDDDDDD DDDDDC1	LC 20ADDDD	700005DD	DDDDØ6DD	DDDDDD91	DDDDDD20	00000000	608DDDDD	70 <b>00</b> 38DD	DDD7803D	DODDDDDD	CDDDDDDD	D40038DD	DD76636	8DDDDD70	*>>>>>	•••••	anne inn	s	>>p >>0	0-
	1792	ADDDDDDD DDDDA10	32 2000000D	506004DD	DDDD16DD	DDDDDDD2	DDDDDD35	DODDDDDD	20800000	600029DD	DDD5005D	DODDDDDD	DODDDDDD	D70008DD	DDD60030	90000050	≠>>>>>°	>>> P >>>	>>>>**>>	>>>> ç	>>` >>'	]
	1856	ADDDDDDD DDDA800	33 31DDDDDD	400004DD	DDDD16DD	DDDDDDAD6	DDDDDC84	DODDDDDD	20800000	63003CDD	DDD4667D	DODCDDDD	DODDDDDD	D90608DD	DDD60000	CDDDDD40	#>>>>/	>>>@	>>>/+>><	>>>> ç	>>` >>`	30000
	1920	ADDDDDDD DDDD300	38 83DDDDDD	100002DD	DDDD26DD	DDDA5ADD	DDDDDA84	DODDDDDD	20800000	60804DDD	DDD166CD	DDD6CDDD	DODCDDDD	DD9698DD	DDD686034	DDDDDD10	#>>>>>@	>>> >>>	833/Z333/	>>>> c <c< td=""><td>;&gt;&gt;1 M&gt;&gt;-</td><td>Ö&gt;÷Ö&gt;&gt; «</td></c<>	;>>1 M>>-	Ö>÷Ö>> «
	1984	ADDCDDDD DDDDC10	38 84DCDDDD	300001DD	CDDD36DD	D967D8DD	DDDDDA84	DODDDDDD	208DDDDD	6301CDDD	DD8301DD	DDD44668	DDD89DDD	DD3628DD	DDD6001C	DDDDD800	≠c>>>>i	(3) (3)	6» Ÿgÿ>>>/	>>>> 0	>>` Ō>>Ā	»> <sup>∙</sup> Fh≥ÿ
	2048	ADD78DDD DDDDDA	02 25D8D0DD	200200287	DDDD46DD	DCCDD9CD	DCDDDA8B	DODDDDDD	20800066	602CDDDD	DD200466	DDD346DC	DDD67DDD		D66602CD	DDDDD290	#00>>>>/		F> (ÖŸÖ c>/		of	fo"Fo÷
	2112	9A5A7C5D DCDDDD1	0 20024664	20000208	DDDD46DD			_					00044000		49DA8DDD		öZI] (>>		F>>]>m/Ö/		Tù©>>>÷	ōc-M/> <sup>s</sup>
		3700060A D6000C6		_			the second se									DDD96366	73++34		F>iE>Fj≠/		ic>≠>>>ê	>mĀ=ÿ> '
		84000601 6200040													DDDDDDCA	DDA10000	No+ bo fo	M16 131	0ò.;f./Fd	_	:M>=<=/	006 "Si
			D 44DD4CDD			CDD41884	_						D8910917		80056608	D9000000	AN 50	nsl stfasis	Ö NBFd		à>P cê	un oüä
		85A69402 1549DCL										_			6D94ADD9		Öllik Tas	"+1>ò>dM>a			.uvr çe	_vp eye
																					+TJ2L	I DA
		4CD42918 073DD96					_										L'J -Tg	g@}ĭ>M∕mAíÇ			/åe+ 1	/ 26 -
		6D99AD28 011CD6D		_								_					mo≠ ++	· LM· · ·	DA DA		l>0"g∉ _	JOT Dd-
		4ACDA420 0060052															JOP	fillemof	>'∨ <'à	p 8	in6A -	−ÿ0 BC†
		41440303 0104010															AD -	di bN	ùH'A DP	D	X	e '`
		45002400 0410500										and the second second					Е* Р	Fd	d Ç		Т_р	ä` dP
	2752	00413010 0011144	12 20020021	8110024C	00000671	02640000	14014100	00000000	000341182	21000000	000000000	20000000	04102049	00000041	18221000	00000000	A0 6	P A L	qd A		ÇI	
	2816	00374303 0316500	00 00000016	50034000	00300003	03003003	30074030	30300300	00303003	03003030	00000410	00200200	00007450	00000000	00000000	00000000	e ì	· @	9			
	2880	00154000 002A000	00 00000019	40001000	00000000	00340000	00000000	00000000	000000000	00000000	00000001	90000000	0000006Z	00000000	00000000	00000000	@ *	6	4			ê
	2944	00300303 0300300	00000000 56	30600300	00300003	03003009	30030030	30300300	00303003	016666640	30030030	047DDDD6	10000000	60002569	D000468D	DD100000					ff@	}>÷
	3008	00200202 0200200	00000000 00	00000000	00000000	000000000	00000000	00000000	00000000	70000000	60030031	90000000	D4003003	Ø84DDDDD	D2020000	DD 300000					}>>>`	ů>>> *
	3072	00200202 0200200	00000000 00	00000000	00000000	00000000	20020020	00000000	00000009	DDDDDDDD	DA10001C	DODDDDDD	DD6600000	634DDDDD	D4010000	DD400300					>>>>/	· · · · · · ·
		00300303 0300300									DDD100AD			6870D000							ù>>>>-	***********
		0000000 0000000												68900000							>>>>>è	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
		0000000 0000000									DC102DDD				_							
		00200202 0200200											Sector 1									çnainn
		00200202 0200200																			ù>>' IP	
		00200236 8751200															6áQ					,,,e ,,
		00204899 8888531															HôààS					
																	©S4Vài	ia				»»/ »»
		00264853 3456888																				»» <sup>*</sup> »»
		00595111 1111479																àA Hà D				»»" »»
		01961111 1111113															n 6	ööR				»»– »»
		03931111 1111111													_		1	5yàc		36R		··- ··
		05711111 1111111						_				_					9	hàu		čwwd	>>9 :	>>- ()
		04611111 1111111		and the second se													a	gàŪ		`¢ww@		ss- ds
		05511111 1111111															Q	6000		°ÿàáÕ	the second se	ù»÷ Éi
		02611111 1111111															a	6800		<b>πôôò</b>	∡≈>Ÿò>— [	
		00931111 1111111															i	yáA		ÏМИКТАТА	ç'>/>>Ä İ	ÑF>fi}'>
	4160	00361111 1111111	11 11111111	11111113	69985208	03003003	2004AAAA	AAAAAAAB	40302004	ADD8DDDD	4A10CD29	DDD6DD7D	D4DD4004	D9798100	64CD56@8	72CDDD10	6	ið	R M	имик. 6	-' <i>~</i> 9>>J (	ð)>+>}'>
	4224	00055111 1111111	11 111111111	111111111	13579962	00000000	00016AAA	AAAAAAA	103036D7	0201DDD9	4DC14DA1	CDD7CD46	64DC0038	6CCD5002	64005201	484DDD20	Q	N N	lőb j	MININES **		M°Ö≎ÖFd∢
	4288	00005631 1111111	11 11111111	111111111	11114788	63000000	00002488	8888884D	00202051	20082020	9D9209D7	3A40444D	D4D38864	23DDA083	D6034401	DD499140	V1		Gàc \$	ààààM	Q−ÿ ùé	¢:@DM**
		00000376 5421111												_			vT!			#332u	LŸBA/	FpLFm¢@
		00200201 4786542															GÜT		6ys	i i	à÷ù≎ë	'≠âút
		00200202 0214788												_			׾		ys	3	FØ	JY B
		00000000 0000013				_									_			aeC	Yb	≠Q	qœ	UD
		00200202 0200200																Wàe1	10	-4	4	D D
		00200202 0200200																#gád!	u hP		nc	10
		00200202 0200200		_														#gaa! \$hŪS	6P		DC	!Q %
																					6	20
		00200202 0200200			_													\$xeA				
	4864	00000000 0000000	000000000000000000000000000000000000000	20000001	25766431	mmn	20000000	20200200	000000000	00000000	000000000	000000000	00003003	00000000	000000000	000000000		%v	ат х			

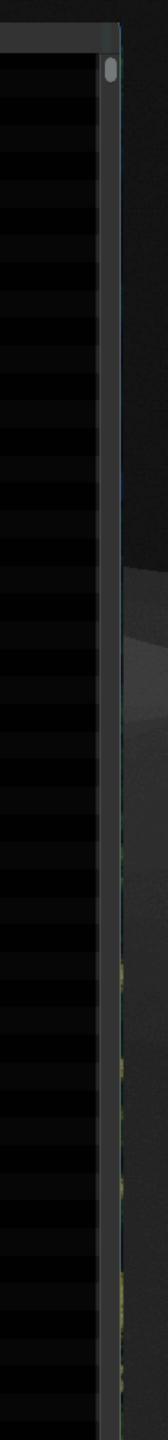


- Each animation has a list of frai
- Each frame points to an "image" and has a delay.
- There is a mask for clock transparency
- Lots of pinball animations are just two or three frames cycling over and over

		0
	E	5
		$\sim$

00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	1
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	0
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	1
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	2
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	3
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
00000000	00000000	
aaaaaaaa	00000000	





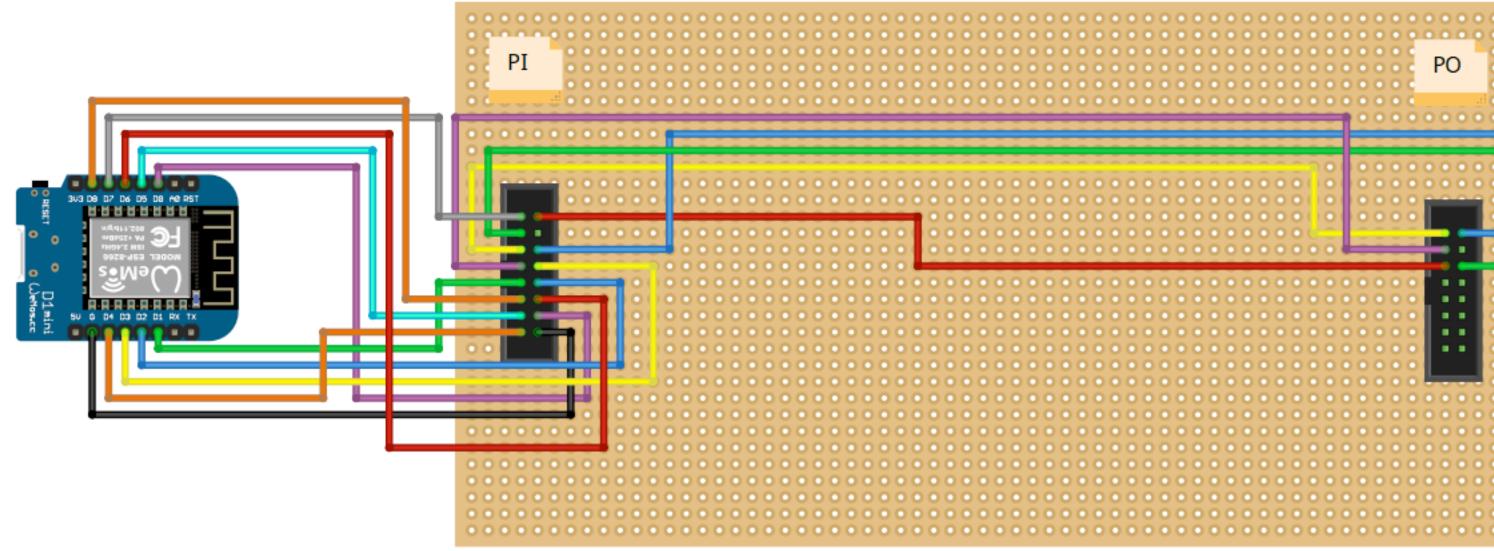
- After doing all the reverse engineering, I stumbled on this:
  - https://github.com/sigmafx/DotClk-Resources
- ~2000 animations in separate files
- Makes it easy to add your own
- Has tools to extract animations from ROMs

• I rewrote my parser and menu code at this point.

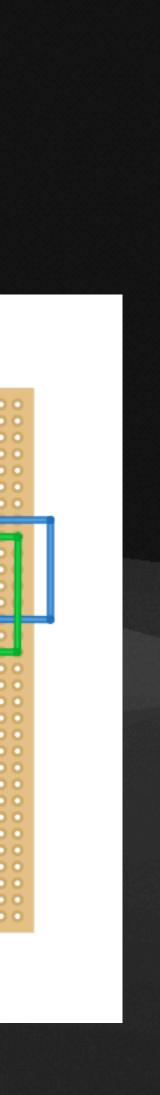
## **ESP8266 and HUB 75**

If you don't have enough I/O you can daisy chain the outputs back into the inputs

In this example 1 data line is used to drive all six (RGB1 and RGB2) datelines









# Demo video here

## Thanks for your interest!

Questions?