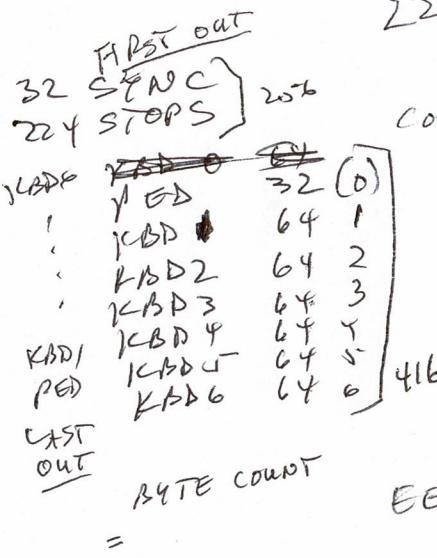


# SERIAL TO PIPE DRIVER

SYNC SYMBOL = DEADACE ~~F~~ (32 BITS) DATA CONFIG FOLLOWS  
~~DEADACE E~~

512 BITS KEYBOARD (8 KEYBOARDS) 0=OFF, 1=ON  
 224 BITS STOPS (28 bytes) 7:0, 0=LAST OUT 0=OFF, 1=ON  
 GROUPS OF 24 BITS MSB=0 (NO SYNC SYMBOL)

~~DATA~~ 8X 28 BYTES ~~XXXX XXX XXX XXX~~



## EEPROM STRUCTURE

BYTES ~~1:0~~ = BOARD ADDRESS  
~~2~~ = # BLOCKS IN THIS BOARD TO PROCESS

## COPPLER

BIT 7 =

NATIVE VS. COUPLED  
 1 = COUPLED  
 0 = NATIVE

~~DATA~~ BLOCK  
 BIT# 0:1 ~~0~~ = KEYBOARD # TO RESPOND TO  
 1 = ~~■~~ START BIT (0-63)  
 2:3 = END BIT (0-63)  
~~4:5~~ = OFFSET (0-63)  
~~6:7~~ = GATE BIT IN STOP FIELD  
 (WHEN 1, TURNS ON)  
 7:8 = AND GATE (SUB SUPER IN STOP FIELD)  
 9:8 = NOT AND GATE IN STOP FIELD  
 (UNISON OFF)

56KBaud  
 TX ON CHANGE  
 SCANNER=20ms

PIPE ON IF [KEYBOARD]

# BITS IN STOP

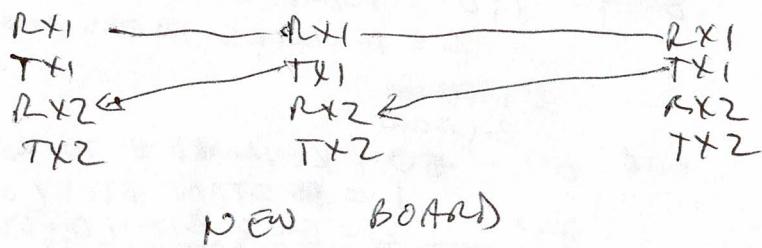
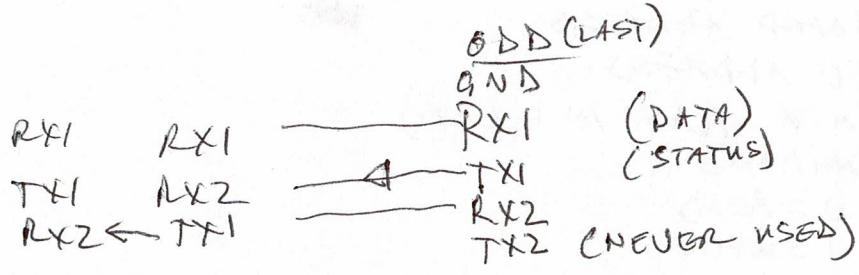
START

END

OFFSET

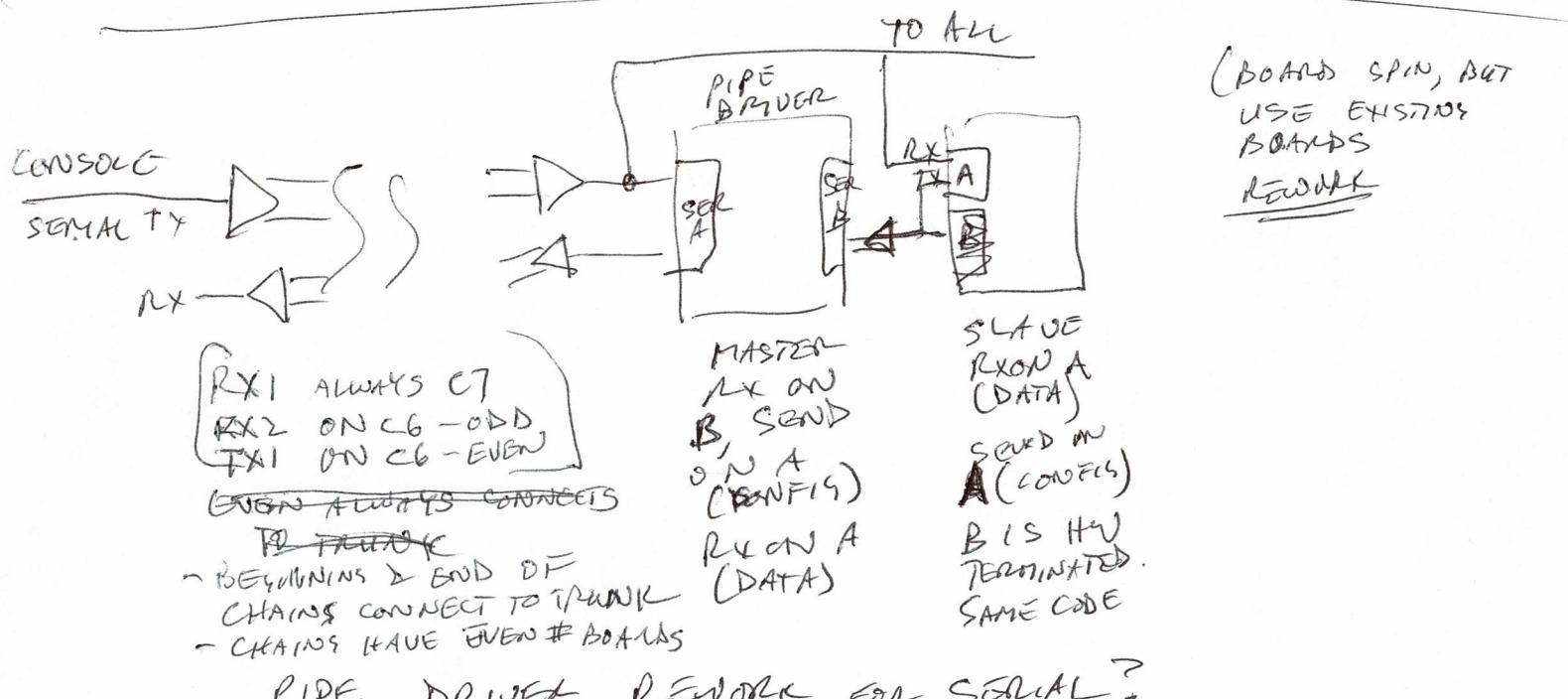
+ OFFSET  
 IF KEYBOARD BIT 1- START & END  
 + GATE & AND 8 ! NOT AND

GATED ! NOT AND  
 GATE & ! COUPLED || (GATE & COUPLED & AND)  
 = GATE & (! COUPLED || (COUPLED & AND))



# MODIFIED MIDI

1000 AAAA NOTE OFF, CH. A, DATA FOLLOWS (7BIT)  
 1001 AAAA NOTE ON, CH. A, DATA FOLLOWS (7BIT) (WE WILL INITIALLY SUPPORT ONLY 8 FOR MEM SAVINGS)  
 1010  
 1011 AAAA SHOW CHAN A, DATA FOLLOWS (7BIT)  
~~1100~~  
~~1101~~ BBBB = # OF 16 BIT BYTES TO FOLLOW (O=1, F=16)  
 1110 BBBB = # OF 16 BIT BYTES TO FOLLOW (O=1, F=16)  
 1111 BBBB = RESET  
 1101 AAAA AAAA = # OF BYTES TO FOLLOW (O=1, F=16)  
 AAAA FORMAT IS  
 110x AAAA MSB BOARD ADDRESS  
 LSB BOARD ADDRESS  
~~# BYTES~~  
 E/START ADDRESS  
~~# BYTES~~  
 DATA FOLLOWS, TO ~~16~~ BYTES ON READ

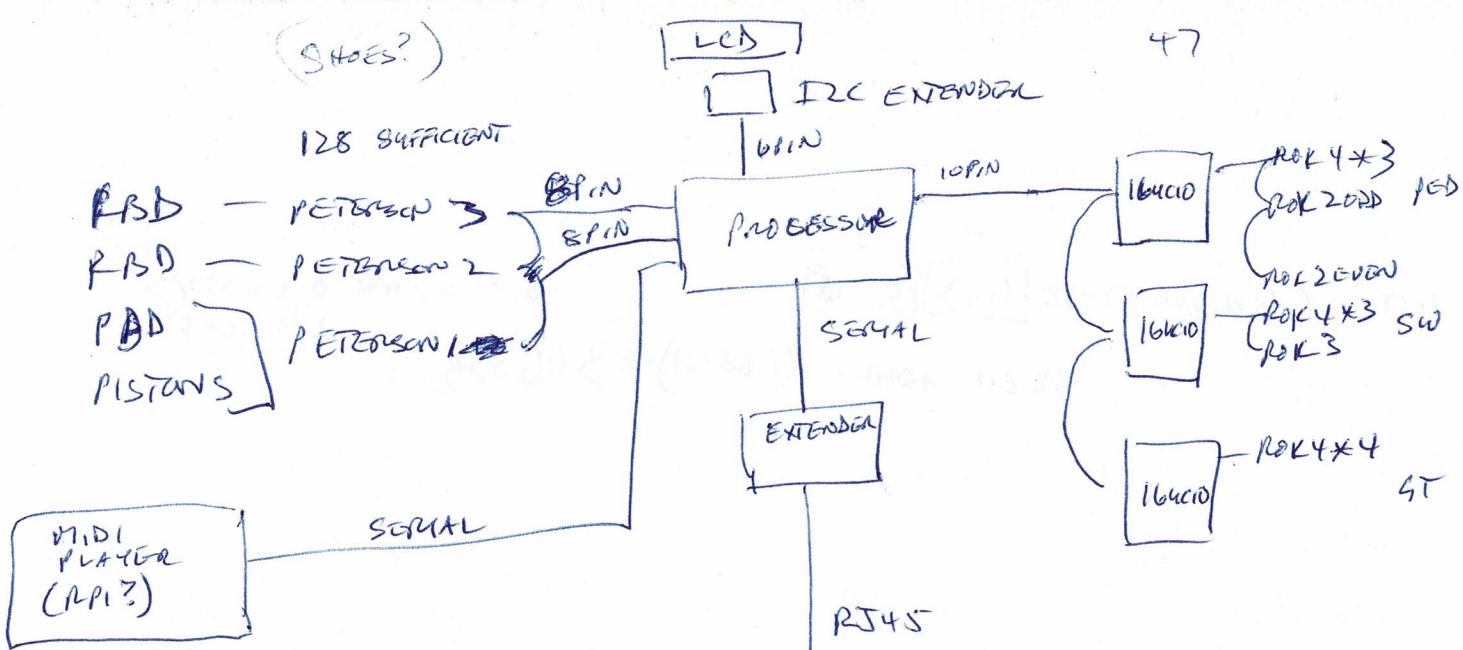


PIPE DRIVEN REWORK FOR SERIAL?

S/W DETERMINES EVEN U. ODD ADDRESS & SETS TX/RX POSITION (SCREENING THIS UP COULD KILL THE RESPONSE PART OF THE CHAIN, BUT THAT WILL ALL STILL HEAR)

(SHOES?)

47



SIGNAL = 5V  
FROM PIPE DRIVERS!



(NEW BOARD)  
PIPE TRUNK

BOURDON 1-97 PIANO 1

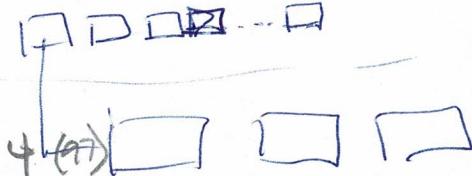


FIGURE TO USE ANOTHER BOARD

TRUMPET 1-85 PIANO 2  
P MUSICAL 1-73 PIANO 3

MIX1

MIX2

ROHL 1-73

NAGHT 1-61

SAL 1-61

VOX 1-49

PHONO 1-73

22/3

13/5

37 boards

$$\begin{aligned} & * 3 + \frac{1}{32} \\ & * 3 \quad 2 + \frac{2}{32} \\ & * 2 \quad 2 + \frac{9}{32} \\ & * 3 \quad 2 + \frac{9}{32} \\ & * 2 \\ & * 2 \quad 1 + \frac{17}{32} \\ & * 2 \quad 2 + \frac{9}{32} \\ & 2 \end{aligned}$$

- STOP SCAN IN IRQ
- MULTI TIMED UGH!!!
- REDUCE STOP SCAN TO 64, LOCAL YKBDS PER STOP CLOCK

MEASURE  
SCAN TIME = 2 KBD SCAN / STOP SCAN OR CONST KBD SCAN  
SEND TIME = 115200  
RECEIVE PRC TIME = 64MHz  
SHIFT OUT TIME ~ AFAP

29 BOARDS  
89's  
1-17  
1-22 (21+1)  
24-32

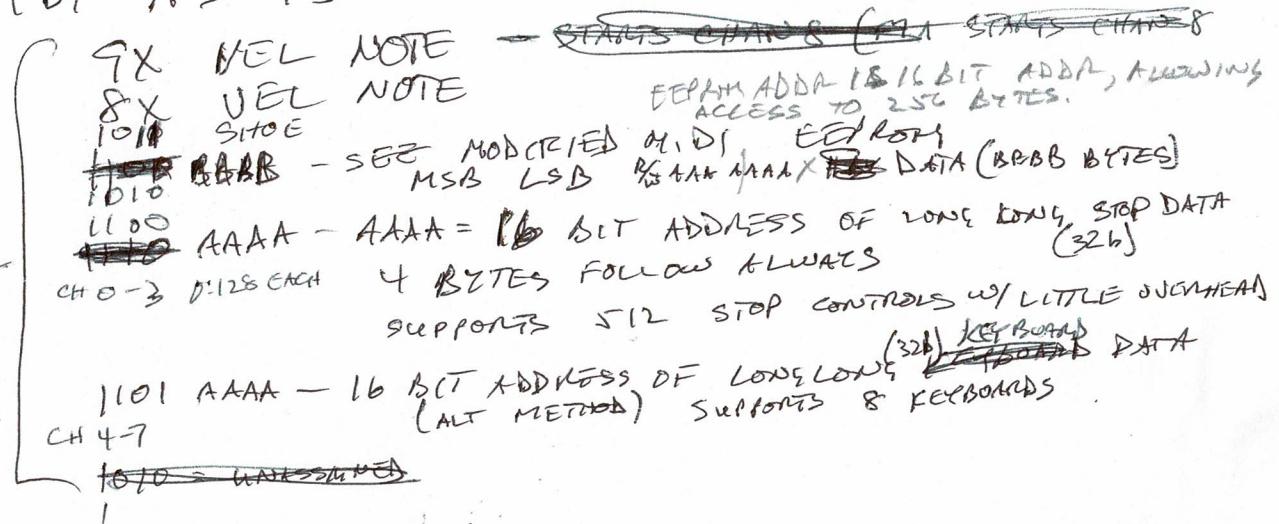
INST CURRENT STATE  $i \quad j$   
[0..3] [0..15]

(6 BITS EACH. 0,1 = STOPS  
2,3 = CCRS)

$$32\text{-BIT ADDR} = (i \& 0x01) \ll 3 || (j >> 1)$$

# 3RD VERSION

## MIDI AS IS



1000 AAAA RETURNS DATA

MSB  
LSB  
EE START ADDR (1AAA AAAA)

II BYTES  
DATA, 16 BYTES (0 = ACK)

FROM STOTS

1111 1111 = RESET

1111 1110 = ACK/CLEAR TO TRANSFER (ON CONFIG).

ALL UNITS SEE 1100 AND HOLD XFER  
EXCEPT FOR ADDRESSED UNIT UNTIL FE ACK

HOST IS [SEND ARRAY] [RECEIVE EX ARRAY] {SEND ACK}

NOTE THIS IS PROGRAMMING, SO STOP  
STUFF CAN BE IGNORED

```

        }

        while (1)
    {
        // Datastructure
        // Keyboard and Stop arrays are 16 long long words (512 bits).
        // EEPROM configuration. PIC18F46K42 has 1K of EEPROM, 4K data RAM
        // 0= Board Address MSB
        // 1= Board Address LSB
        // 2= Number of valid blocks (Up to 32)
        // 6= Validity signal for EEPROM, if EEPROM is valid, =0x5A
        // 7= Validity signal for EEPROM, if EEPROM is valid, =0x96
        // Configuration blocks (n start at 1)
        // 8n+0 = Block Config
        // Bit 0 = bit 9 of Start field
        // Bit 1 = bit 9 of End field
        // Bit 2 = bit 9 of Gate field
        // Bit 3 = bit 9 of And field
        // Bit 4 = bit 9 of Nand field
        // Bit 7 = 1=Coupler 0=Stop
        // 8n+1 = Start bit - 512 bit number of the Keyboard bit to start
        // the pipe drivers. This always maps to PIPE01
        // 8n+2 = End bit - 512 bit number of the Keyboard bit to end
        // the pipe drivers
        // 8n+3 = 256 bit Offset from keyboard
        // 8n+4 = Gate bit - 512 bit number of the Stop bit to gate on.
        // This reflects the stop in question
        // 8n+5 = AND gate - 512 bit number of the Stop bit to AND with
        // the gate bit. Used when the Coupler bit =1
        // 8n+6 = NAND gate - 512 bit number of the Stop bit to NAND with
        // the gate. Used for Unison Off.
        // Activation formula is
        // if (keyboardbit+offset) is between Start and End. So
        // start loop (x) at (Start+offset to End+offset)
        // Drive bit[x]=KeyboardBit[start+offset+x] & (Gate[n] &
        // (!Coupler&!NAND) | (Coupler&AND))
        // Calculation is done on successive commands
    }
}

```

```

    // Receive from UART1
    // command 0xAn. SysConfigure n=number of bytes (0-15, 0 on read) .
    // Byte order following: BoardMSB // BoardLSB // Address (bit 7=0
    // for write, 1 for read) // n bytes of data (if Write)
    // If it isn't mine, CTS=false for me until 0xFF.
    // command 0xBn. MIDI CC command, n=channel. Byte order following:
    // Controller number (0-0x7F), Value (0-0x7F)
    // command 0b110r aaaa. r:0=stopData, l=KeyData. aaaa=address of long
    // long. Byte order following: 4 bytes of data, MSB first in time.
    // command 0xFE = ACK/CTS. Hold serial output until ACK if the EEPROM
    // 0xAn command isn't mine. Host process [Send An array]
    // [Receive En array] [Send 0xFE]
    // command 0xFF = Reset
    //

    if (UART1_is_rx_ready()) {
        TempBuffer=UART1_Read();
    }

    // 0xAn EEPROM sys configure
    if ((TempBuffer&0xF0)==0xA0) {
        while (!UART1_is_rx_ready()) {}
        TempData1=UART1_Read();
        while (!UART1_is_rx_ready()) {}
        TempData2=UART1_Read();
        if ((TempData1<<8|TempData2)==BoardAddress) Mine= true; else Mine=false;
        while (!UART1_is_rx_ready()) {}

        EEBase=UART1_Read();
        ByteCount=TempBuffer&0x0F;

        if (Mine) {
            if ((TempBuffer&0x0F)>0 && (EEBase&0x80)==0) {//write
                for (i=1; i <= (ByteCount&0x7F); i++) {
                    while (!UART1_is_rx_ready()) {}
                }
            }
        }
    }
}

```

ACSO MIXTURE

1-25

26-37

38-49

50-61

2

2 $\frac{2}{3}$

4

8

1 $\frac{1}{3}$

2

2 $\frac{2}{3}$

4

1

1 $\frac{1}{3}$

2

2 $\frac{2}{3}$

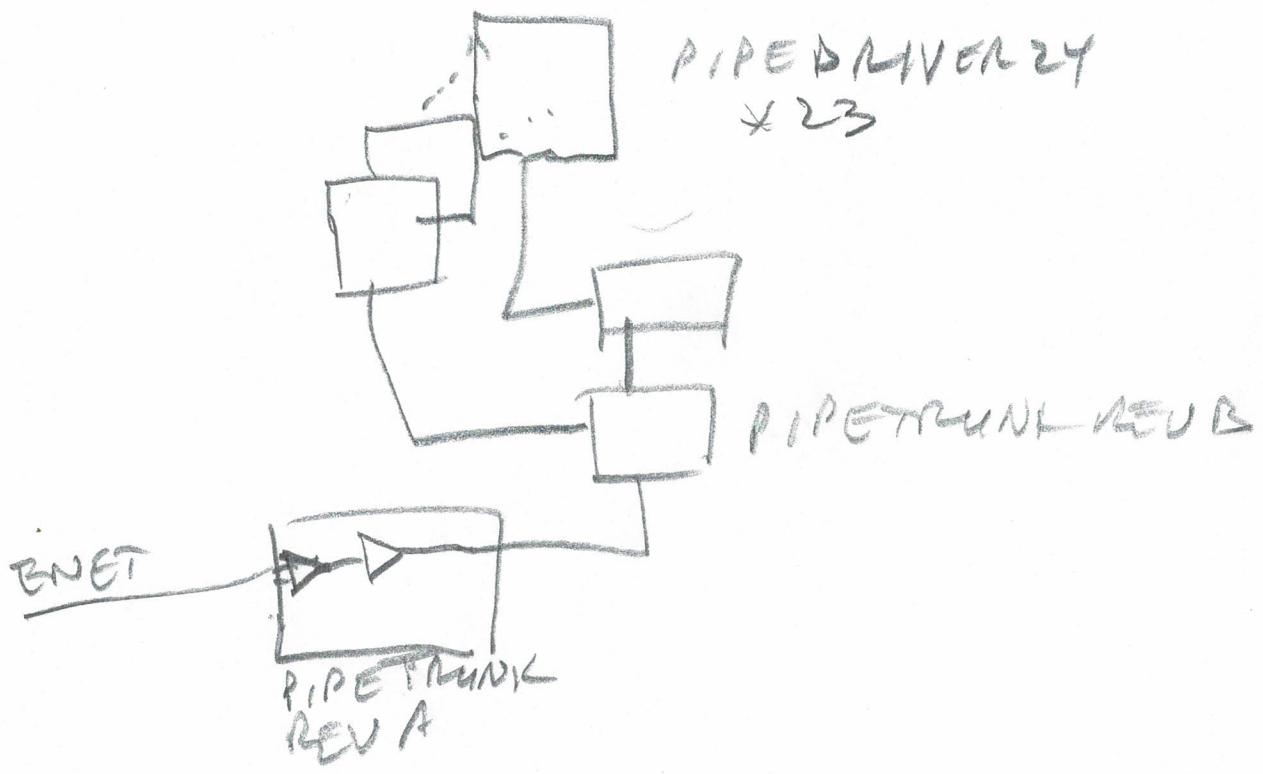
2 $\frac{2}{3}$

1

1 $\frac{1}{3}$

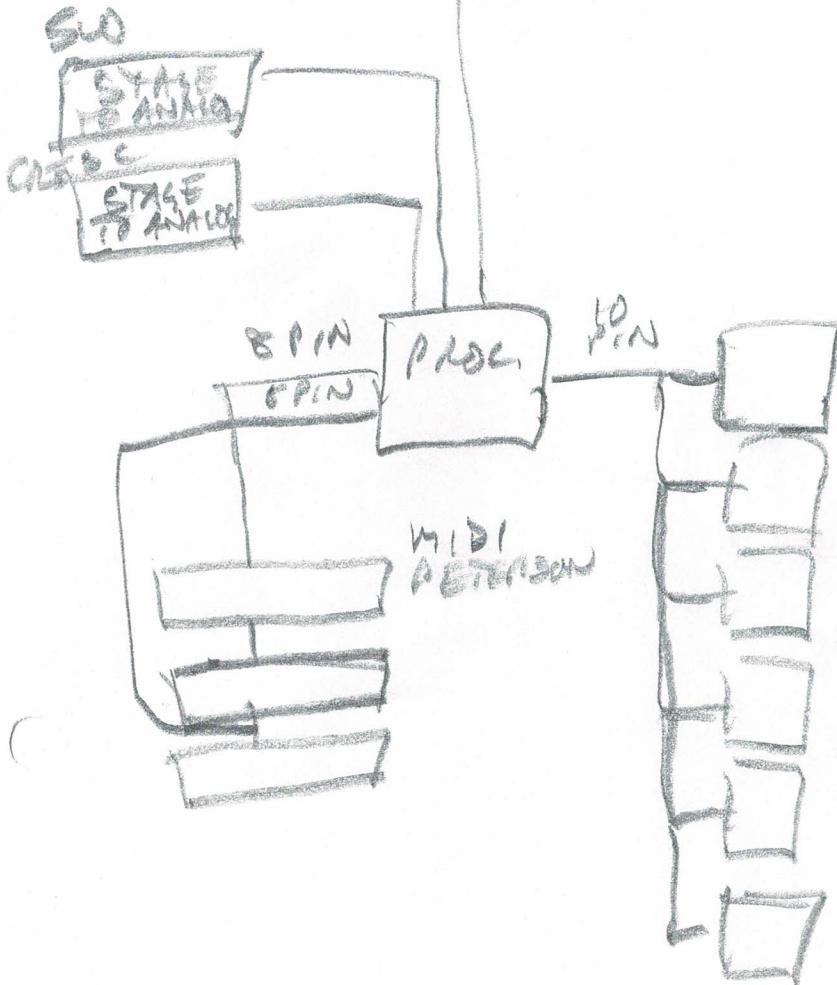
2

CHAMBER



CONSOLE

TO CHAMBER  
(DIFFERENTIAL ON  
CAT 5)



8 HARRIS LOCKER REV A

$$\begin{aligned}
 P\text{C} &= 192 - 223 \\
 S\text{W} &= 384 - 444 \\
 G\text{T} &= 448 - 508
 \end{aligned}$$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27		
Rustant 16'	Bourdon 16'	Volute 16' (SN)	Octave 8'	Bourdon 8'	Volute 8' (SN)	Cabretta 4'	Bassoon 4' (SN)	Great 16' (SW)																				
Resonant 32'	Bourdon 16'	Volute 16' (SN)	Octave 8'	Bourdon 8'	Volute 8' (SN)	Cabretta 4'	Bassoon 4' (SN)	Great 16' (SW)																				
								Principal 8'																				
								Octave 4'																				
								Flute 8'																				
								Clarinet 8'																				
								Trombone 4' (SW)																				
								Trumpet 4' (SW)	Trumpet 4' (SW)																			
								Quint 11/3	Quint 11/3																			
								Terz 2'	Terz 2'	Terz 2'																		
								Nazard 2/3	Nazard 2/3	Nazard 2/3																		
								Principal 4'	Principal 4'	Principal 4'	Principal 4'																	
								Violin 8'	Violin 8'	Violin 8'	Violin 8'																	
								Clarinet 8'	Clarinet 8'	Clarinet 8'	Clarinet 8'																	
								Flute 8'	Flute 8'	Flute 8'	Flute 8'																	
								Violin 16'	Violin 16'	Violin 16'	Violin 16'																	
								Resonant 32'	Resonant 32'	Resonant 32'	Resonant 32'																	

14. *U* *U*  
 15. *U* *U*  
 16. *U* *U*  
 17. *U* *U*  
 18. *U* *U*  
 19. *U* *U*  
 20. *U* *U*  
 21. *U* *U*  
 22. *U* *U*  
 23. *U* *U*  
 24. *U* *U*  
 25. *U* *U*  
 26. *U* *U*  
 27. *U* *U*

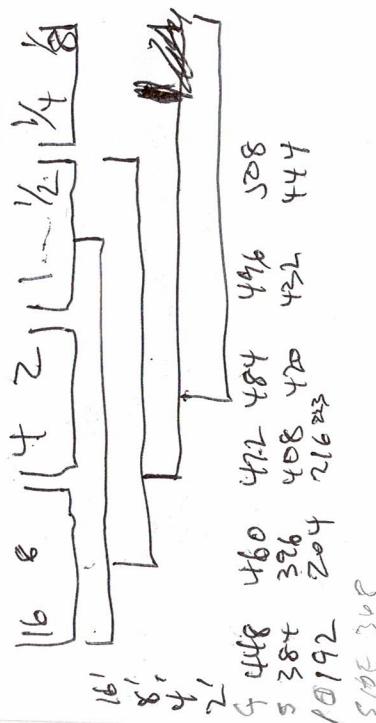
The Church of Jesus Christ of Latter-day Saints, Aliso Viejo Stake Center Wicks Organ Summer 2020  
 951-285-1245

14. *U* *U*  
 15. *U* *U*  
 16. *U* *U*  
 17. *U* *U*  
 18. *U* *U*  
 19. *U* *U*  
 20. *U* *U*  
 21. *U* *U*  
 22. *U* *U*  
 23. *U* *U*  
 24. *U* *U*  
 25. *U* *U*  
 26. *U* *U*  
 27. *U* *U*

14. *U* *U*  
 15. *U* *U*  
 16. *U* *U*  
 17. *U* *U*  
 18. *U* *U*  
 19. *U* *U*  
 20. *U* *U*  
 21. *U* *U*  
 22. *U* *U*  
 23. *U* *U*  
 24. *U* *U*  
 25. *U* *U*  
 26. *U* *U*  
 27. *U* *U*

14. *U* *U*  
 15. *U* *U*  
 16. *U* *U*  
 17. *U* *U*  
 18. *U* *U*  
 19. *U* *U*  
 20. *U* *U*  
 21. *U* *U*  
 22. *U* *U*  
 23. *U* *U*  
 24. *U* *U*  
 25. *U* *U*  
 26. *U* *U*  
 27. *U* *U*

## BOARDS

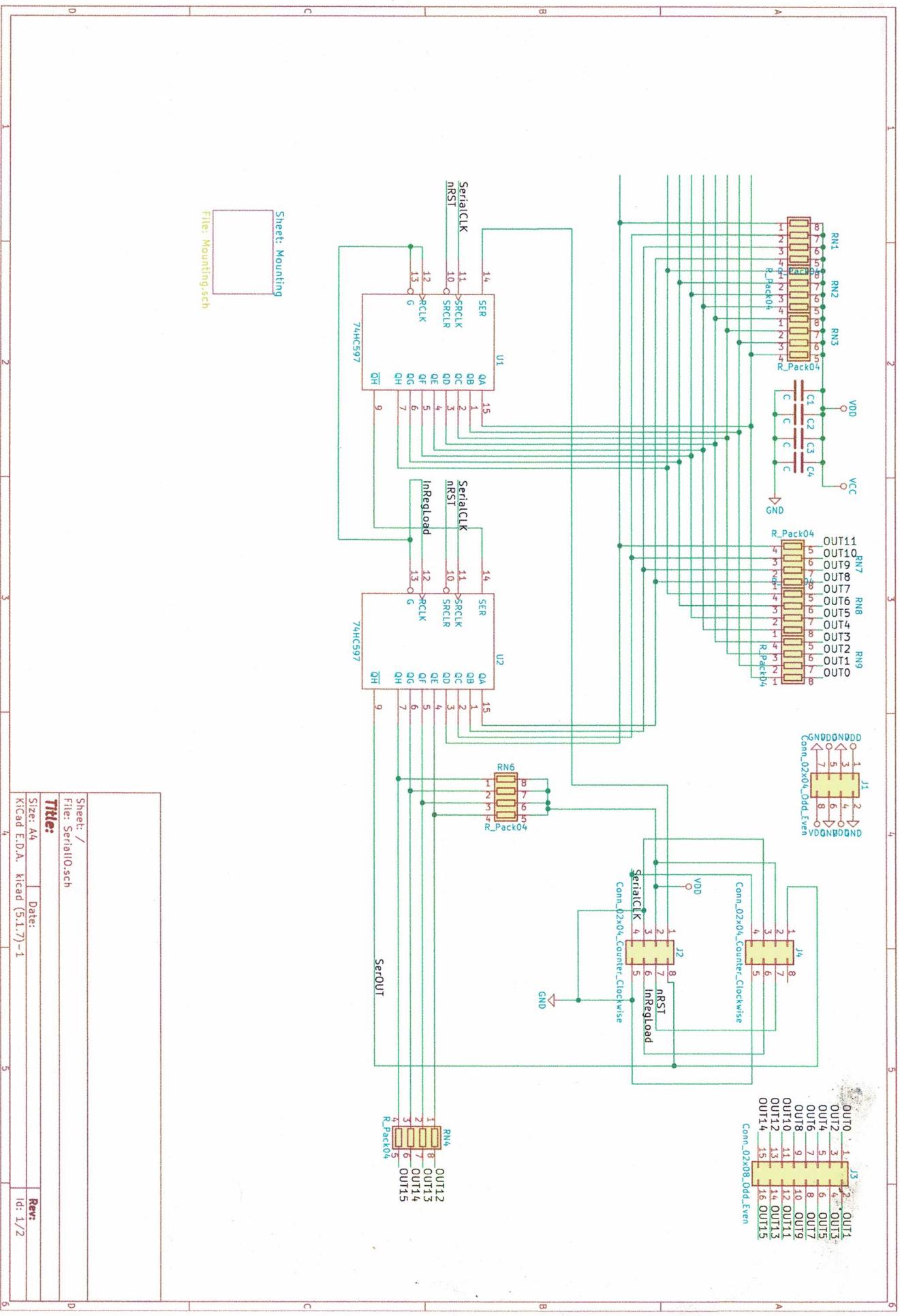


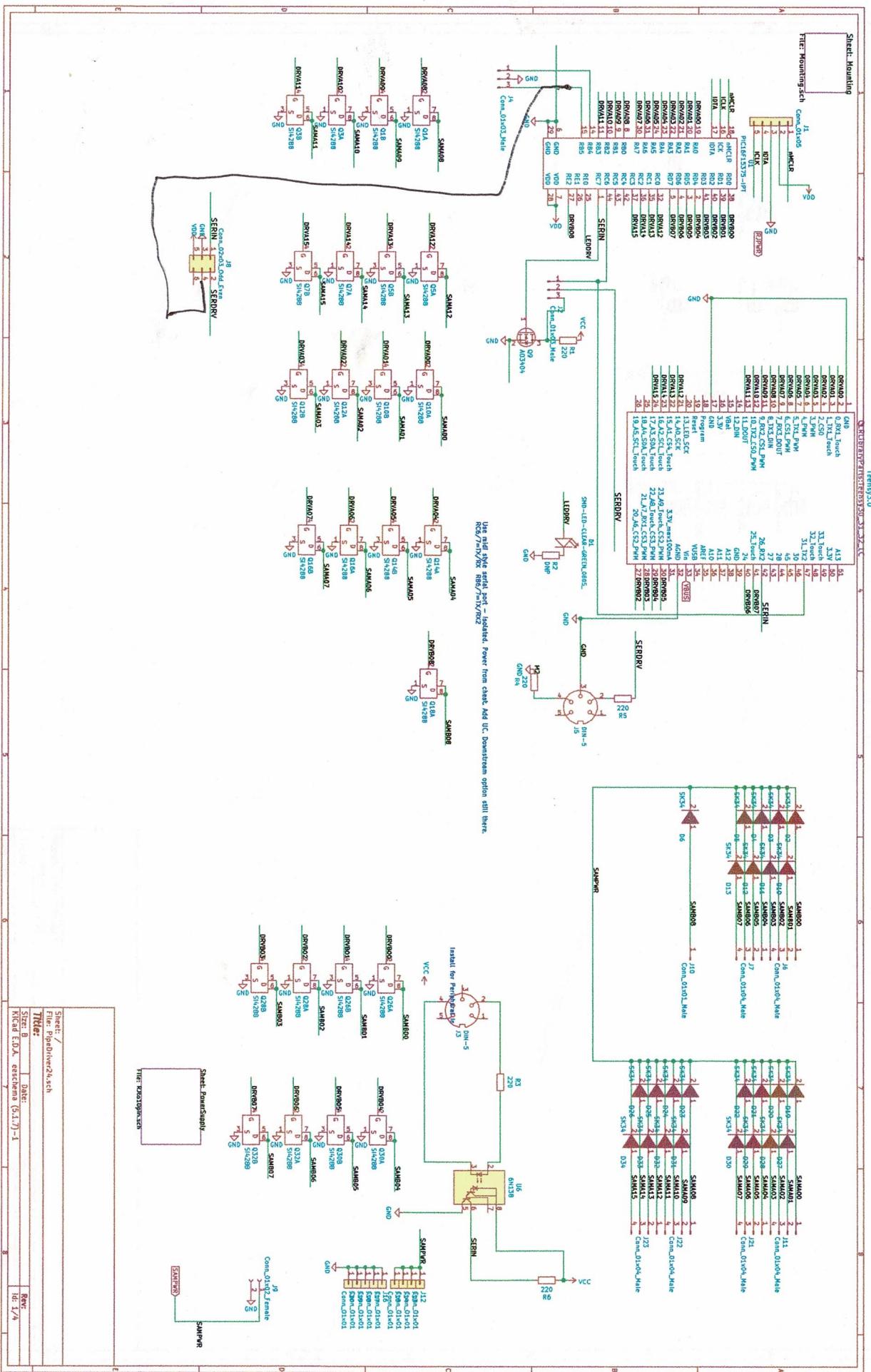
0/0 = Always on

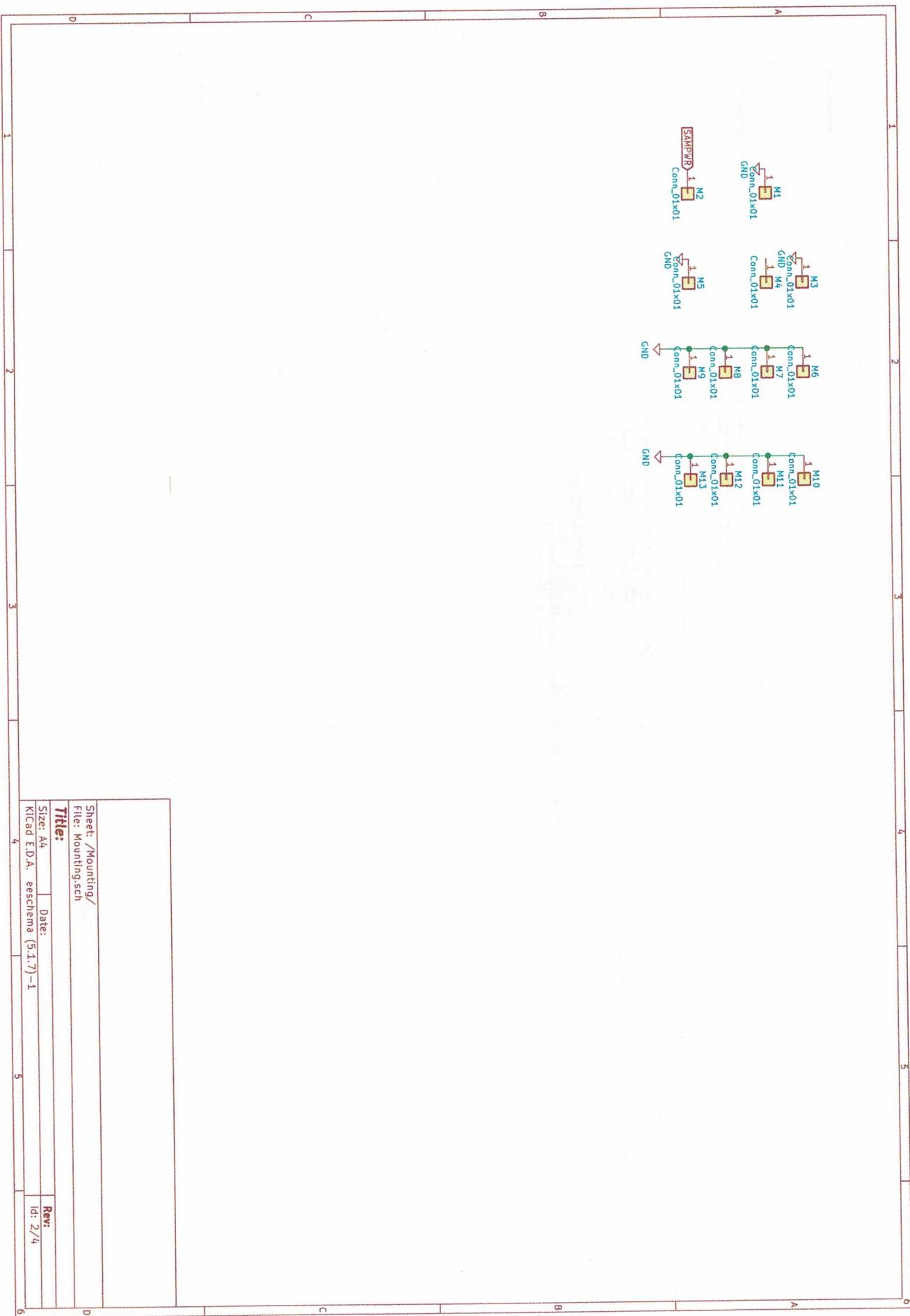
For ~~site~~, site

KEY ~~511~~ (get last) =  
Always on (then)

R, R ADD + TPT super -  
R, 497 -

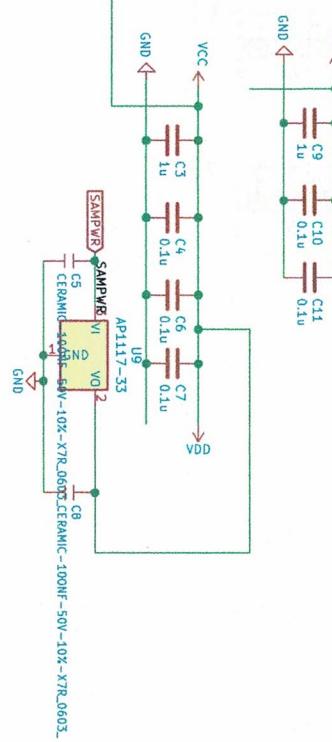






Sheet Mounting

File: Mounting.sch



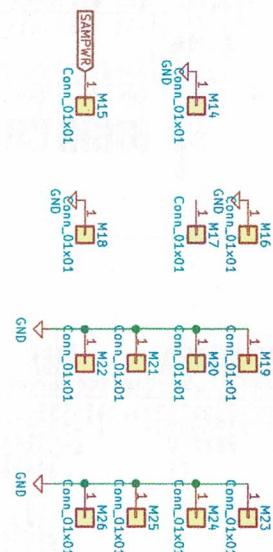
Install for Host Out

Install for Peripheral In

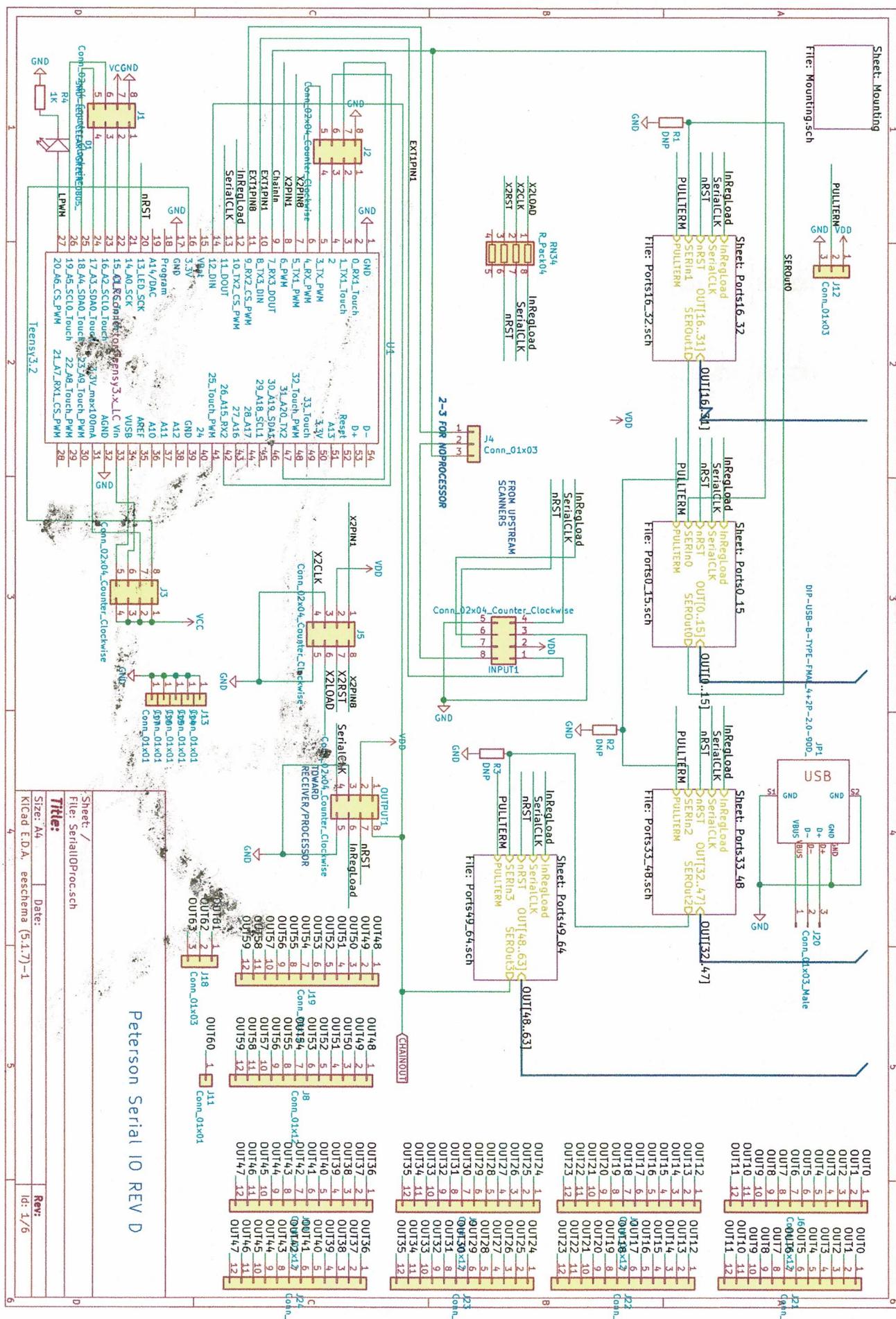
Sheet: /PowerSupply/  
File: Roto10pin.sch

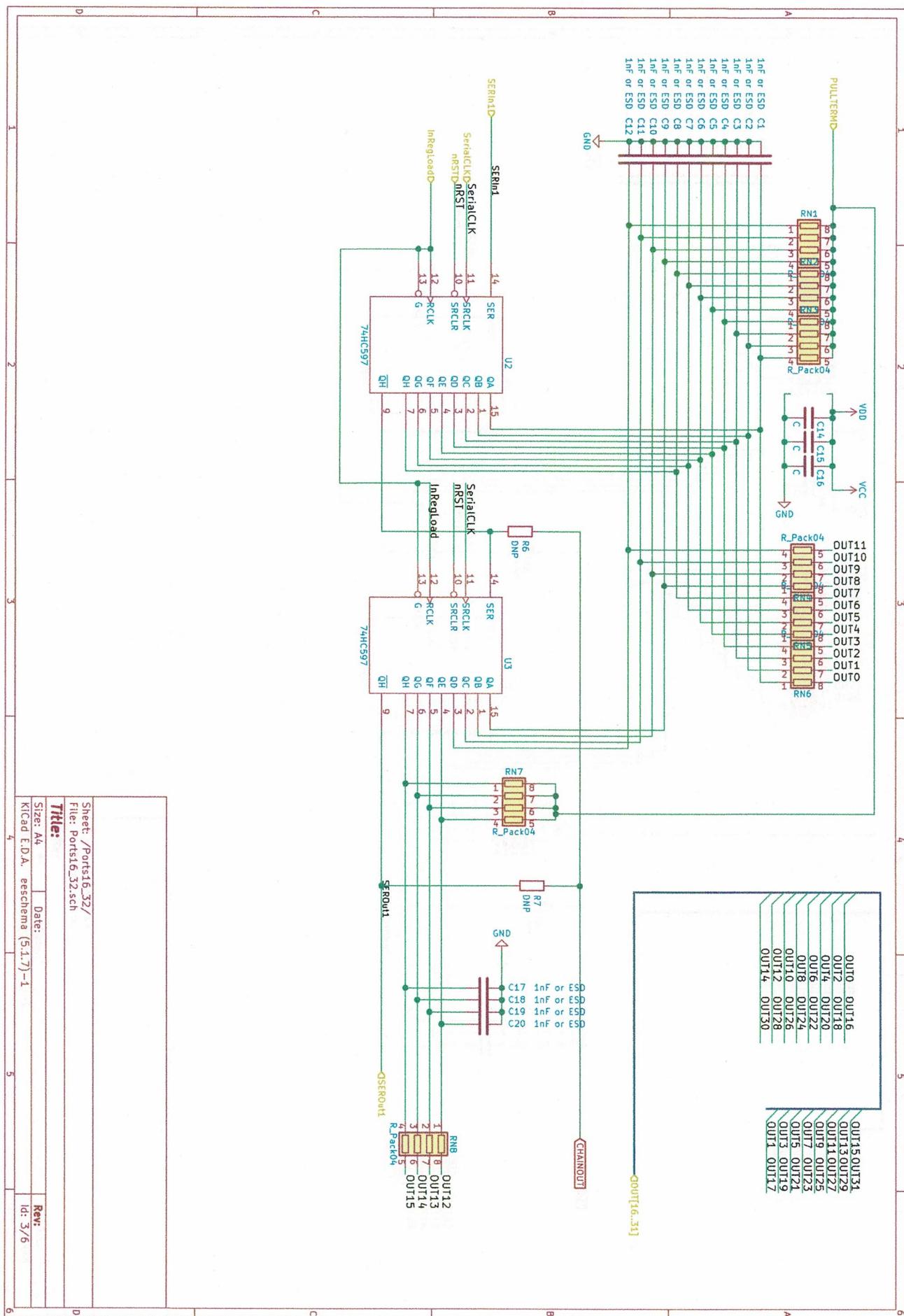
Title:

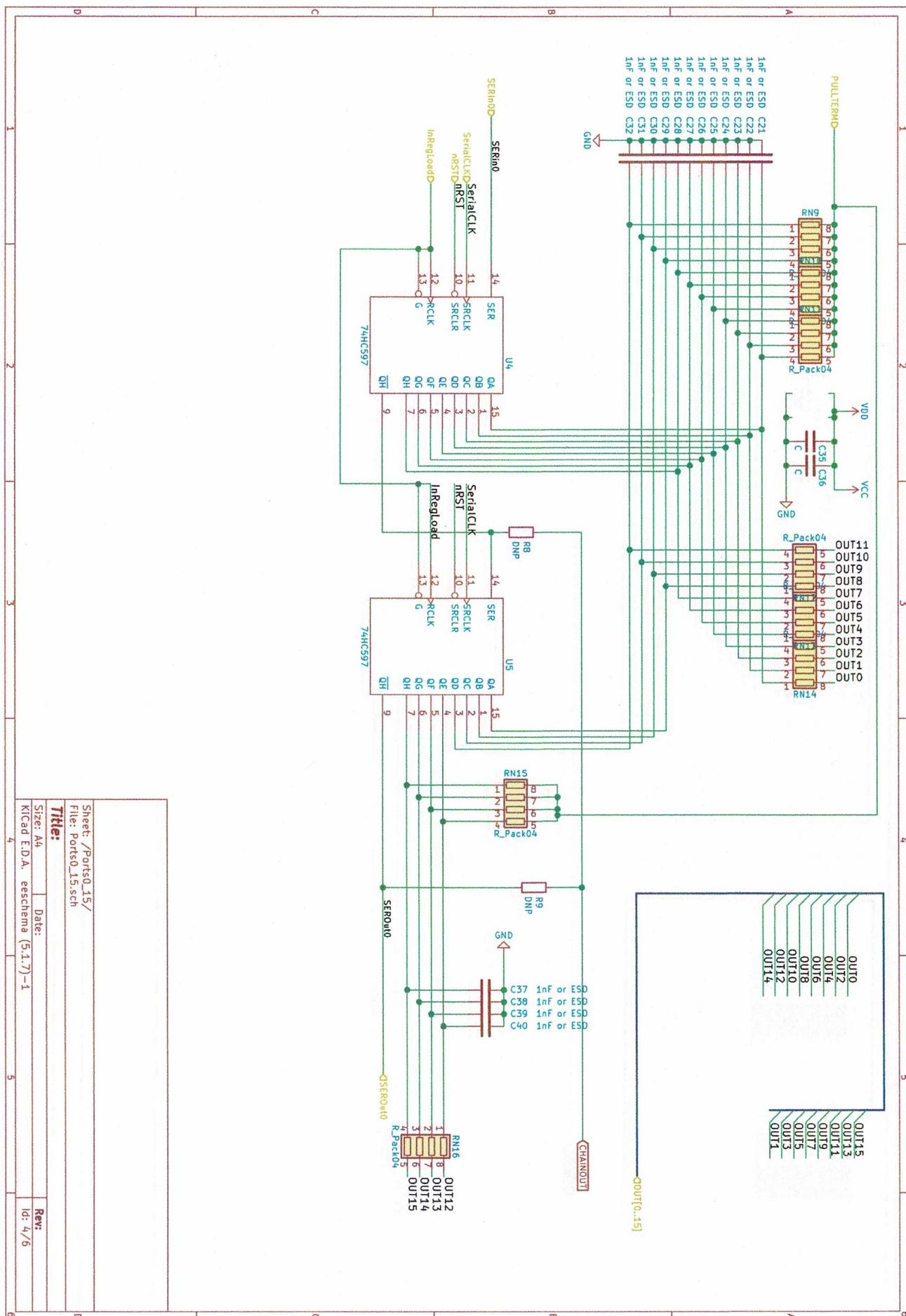
Size: A4	Date:	Rev:
KICad E.D.A. beschema (5.1.7)-1		Id: 3/4

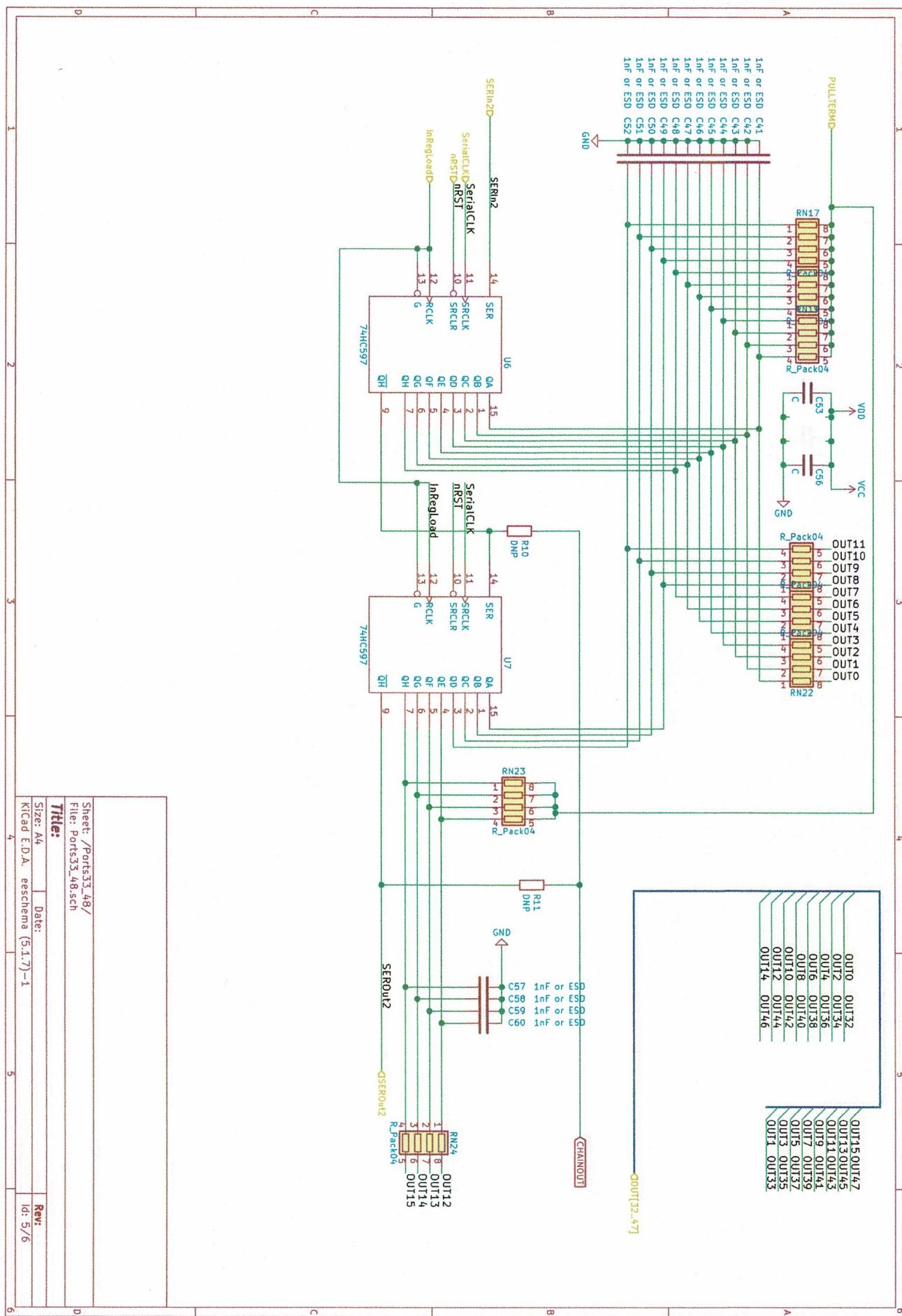


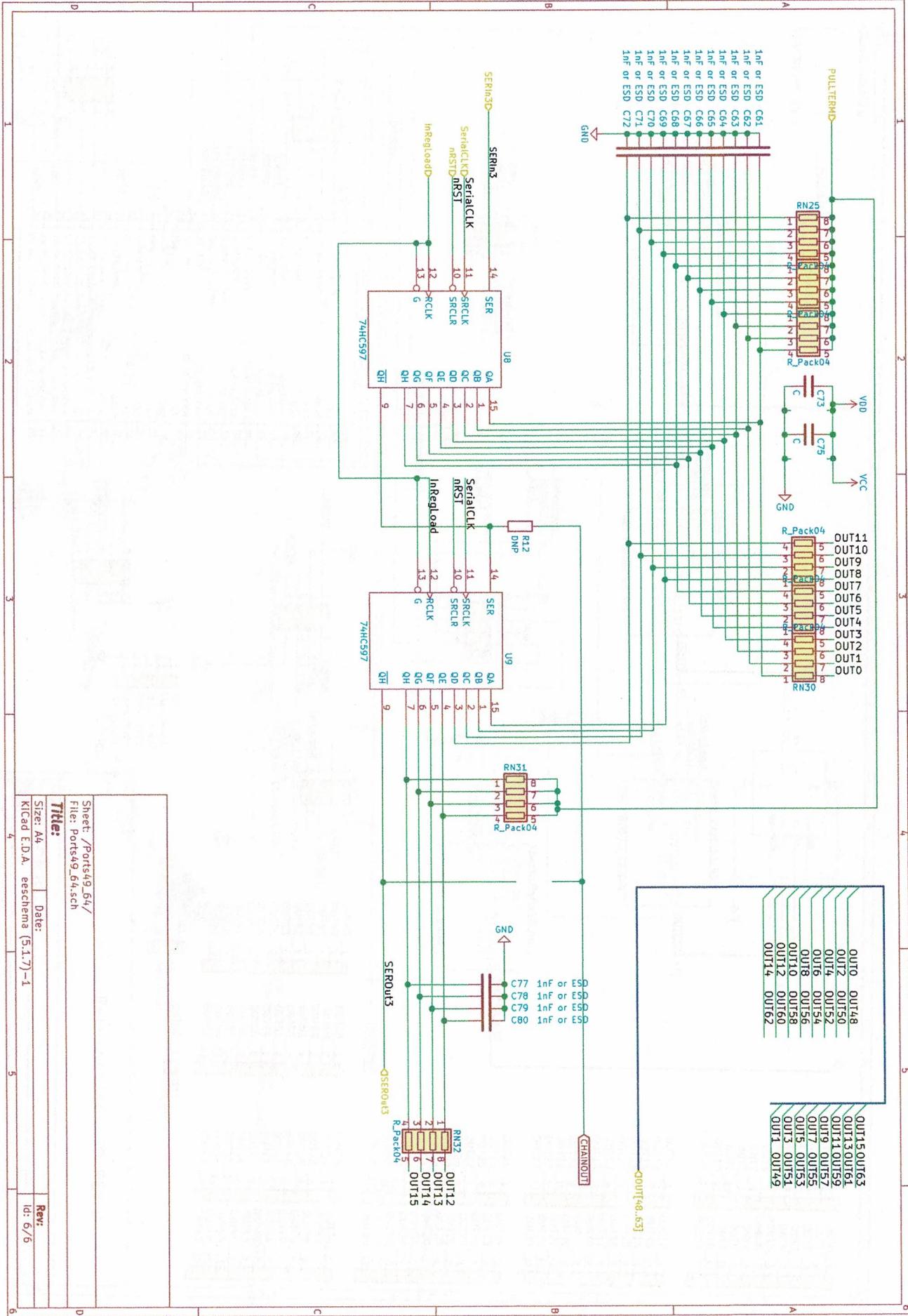
Sheet: /PowerSupply/Mounting/			
File: Mounting.sch			
<b>Title:</b>			
Size: A4	Date:	Rev:	
KICad E.D.A.	eeschema (5.1.7)-1	Id: 4/4	
1	2	3	4
5	6	7	8











# AUDIO DEV TASKS

CRESCENDO

TUTTI

MIDI IN

CONSOLE CONTROL  
MIDI OUT ON MIDI CONN  
COUPLERS REFLECTED IN MIDI OUT

RECORD/PLAY BACK

UI DESIGN

DATABASE PLAY BACK

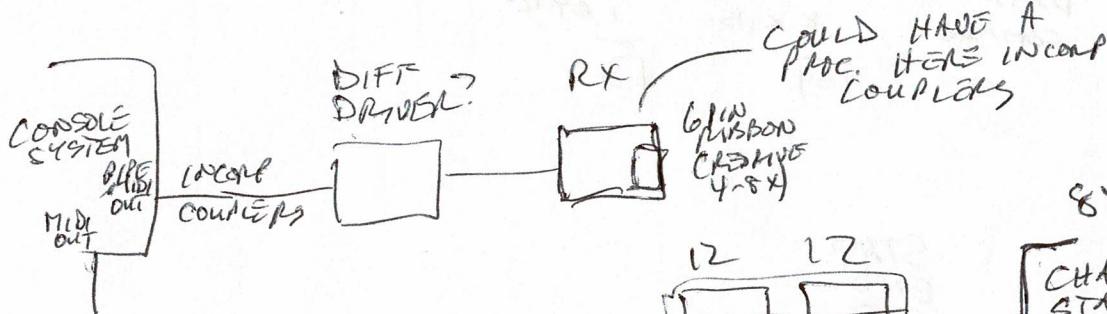
PIPE CONTROL

CHECK PINOUTS

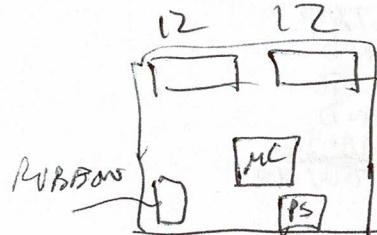
NEW BOARD

FIRMWARE ARCHITECTURE.

TO PIPE DRIVERS - SERIAL STREAM, MIDI, INTEGRATES  
~~ALL CONTROL ACTION BEFORE IT SETS THESE.~~  
 CONFIG MESSAGES OFP0 ~~E7E~~ XXE F7



BOARD  
 14 BIT ADDRESS BLOCK  
~~7 BIT BE ADDRESS~~  
~~2 MSG DATA FIELD~~  
~~(1)~~  
 7 BITS DATA FIELD  
 6 BITS COMMAND FIELD  
 00 READ  
 10 WRITE  
 01 YES  
 11 EXIT



CHANNEL - 0-15 / 0-15  
 START - 0-7F  
 END - 0-7F  
 GATE - (CH#, #)  
~~OFFSET~~ (0-15)  
 AND GATE (CH#) ~~AND GATE (CH#)~~  
 NOT AND GATE (CH#)  
 START FIRST NOTE (0-7F)

(0 BYTES) USE ALL  
 256 BYTES = 16 Blocks (15)

BLOCK 0:  
 BOARD ADDR (2 BYTES)  
~~#~~ BLOCKS (1 BYTE, 0-15)

CHANNELS

- P0 - 1
- AT(I) - 2
- SW(I) - 3
- CH(I) - 4
- EU(I) - 5
- PS(I) - 6

## DATA

0x DEADACER	SYNC	32 BITS	4 BYTES	
0x 0nnn nnnn	STOP	8x32 BITS	32 BYTES	$\frac{234}{192}$ CONIXOS
	KEYBOARDS			
	KBD5		8 BYTES	
	4		8 BYTES	
	3		8 BYTES	
	2		8 BYTES	
	1		8 BYTES	
	PERML		4 BYTES	
12.85 ms			(40 BITS)	
56KB and 9 BITS			80 BYTES	
			76 BYTES AFTER SYNC	

## CONTROL

0x DEADACE1	SYNC	32 BITS	4 BYTES	
	BOARD ADDR	16 BITS	2 BYTES	
	EE ADDR	16 BITS	2 BYTES	
	DATA	64 BITS	4 BYTES	
	COMMAND	8 BITS	1 BYTES	
		104	13	

START  
END  
GATE  
AND  
NAND  
~~COFFLER~~