

APPENDIX A

Generic Test Point Placement

Test coverage

When designing with testing in mind, it is always important to try and get as much coverage as possible on all signals on your board. As such, there are a couple of rules to keep in mind when testing:

- Try and break your board up into “blocks” of functionality. If you have for example a power supply – you can see it as 1 “block”, that will either be working or not working depending on the presence of the output voltage assuming that there is an input voltage injected into it. Using the power supply example, you should place 3 test points in that case – 1 for the common ground that both input and output should be referenced to, one for the input injection point, and one for the output voltage to be measured. If you have more than 1 “block” that shares a common ground, then 1 test point is sufficient for the reference to both.
- When you are running through a test sequence, it is important to only rely on previously tested parts of your circuit when testing a second part of your circuit. For example, if you have a processor that uses the above power supply, then you can only test that processor once you have established that the power supply is in fact working. However, in situations where you cannot use the previously tested part of the circuit to power the following part (if for example you need to have a connector in between your power supply and processor), then it is important to place test points on both the output of the one circuit as well as the input to the following circuit – even if while in normal operation the first circuit will power the second.
- There are two distinct test stages in the testing process – pre-programming and post programming. Pre-programming tests are physical that involve measuring of voltages and available signals (like pulse sources) that would be input into a processor (if any), and ensuring that those signals are all at the correct expected levels. Post-

programming tests are run after the pre-programming tests have verified all signals are at the correct level and the processor has been programmed. These tests generally involve either running diagnostic checks with the help of the processor (for example reading and writing to a memory chip) or injecting signals into input ports of the processor, and ensuring that it reads correctly (like for example injecting a 1.5V signal into the processor, and ensuring that the ADC measurements are correct)

- When you have completed the placing of your test points, every component on your board (with the exception of through hole, wave soldered connectors, these can be visually inspected) should have to have participated in at least 1 test. Components can be either tested pre- or post-programming. A pre-programming example would be the first “power supply” example - if you have tested the power supply's output after injecting an appropriate input and it passes, then all of the components in the power supply circuit can be assumed to be working. A post programming example would be the ADC voltage measurement example – if 1.5V is injected into the board, and the processor measures correctly then it can be assumed that the series resistor protecting the ADC pin and the ADC external reference can be assumed to be working.
- It is important when testing complex components with multiple parts to ensure that all of the parts are tested. For example, when testing a quad amplifier ic that has 4 op-amps in it, each of the op amps that are used in the circuit need to be tested individually before it can be assumed that the IC is working.

Test Point Checklist

There are a few components that generally need to checked for on most boards. This is not an exhaustive list and the above guidelines should still be followed when placing test points even if all of the components on the check list are ticked off:

Tick	Section
	The unregulated input from a mains supplied transformer should be tested to ensure it supplies power to the individual supplies in the correct range (so even though your power supply may work if the voltage is twice as high as it should be, failure conditions need to be taken into account)
	All voltages that are supplied from the individual supplies need to be tested (5V, 3.3V, 12V etc)
	Any signal sources that will inject into the processor need to be tested. This includes things like opto couplers that are isolating signal from off board or the SAMES chip on the MCU board that will supply pulses. Remember to the full chain for each of these signal sources should be tested, so in the case of the opto coupler, there should be test points on the input and the output to the processor, and on the SAMES chip, test points should be placed so that a power source can be simulated to push output pulses from the chip
	Any devices connected exclusively to the processor, that use its internal peripherals for communication do not need test points placed, as these should be tested by the processor to ensure that the functionality is tested completely in the format that it will be used in future. This includes components like FRAM memory, EEPROMs and dataflash memory, oscillators etc.
	Off board communication that would go off board via a connector needs to have test points placed right next to the connector pins. Through hole wave soldered connectors are not generally tested as the automated optical inspection test is considered sufficient, but in the case of specialised connectors, special provision would need to be made (for example, a specialised SMD connector used to connect the an LCD display would need to be tested by placing a camera to ensure that the segments work correctly, testing the full chain)
	Whatever programming method is used would need to be added for all of the signals required (like JTAG programming for example)
	Debug port (if available) needs to be taken through the whole chain (for example, if in normal operation the debug port is a RS232 port that goes through a driver chip, then

Tick	Section
	the test points need to be placed after the RS232 chip, not before on the processor)
	If there is a backup supply (like a battery or supercap), then that would need test points placed on it so that the backup capabilities can be tested
	Any components that are placed on the board where they offer no functionality to the board itself, need to be tested for whatever purpose they are meant to be used for (for example, in a product with two PCBs, where one PCB holds the filter front for a PLC modem on the second PCB)
	Components that cause a change in some external medium (like for example buzzers that make sound or LEDs that create light) should be tested to ensure that they do act as expected in those mediums on the test jig but test points should still be placed before and after them to check that their voltage drops are correct.

Table 15: Test Point Check List

Specifications on placement.

- The test points themselves need to preferably be a through hole that is plated on both sides of the PCB, with a 0.8mm drill hole and a square pad of 1.5x1.5mm.
- The test points need to be placed a minimum of 2mm away from any physical components on the PCB on the side that the test point will be accessed from, to ensure that the test pins don't get blocked by those components.
- Test points should be placed a minimum of 3mm apart on test jigs that need to be cleaned once a week. If there is no other choice, the test points can be placed an absolute minimum of 2.54mm apart on boards that are not high volume, as then the test jigs will need to be cleaned daily.
- In situations where there is not space whatsoever to place test points, the pins of through connectors/components can be used as test points as well, taking into account that these test pins will need to be specially ordered and as such will increase lead

times on test jig creation and repair. This should only be used in low volume situations

- Close to the center of the PCB, a landing position for the presence switch should be made available (and marked on the silkscreen layer if possible), of about 5mm in diameter.
- The test jig is clamped from above, and as such space needs to be left on the four corners for the test clamp to either press onto the bare PCB, or onto very rigid components that are shock resistant (like a transformer or connector), to ensure that the board can be pressed down with sufficient force to counteract the test pin springs.
- Test points and through hole component leads should be placed a minimum of 1.5mm away from the edge of the PCB to ensure that the board is easy to place in the test jig.

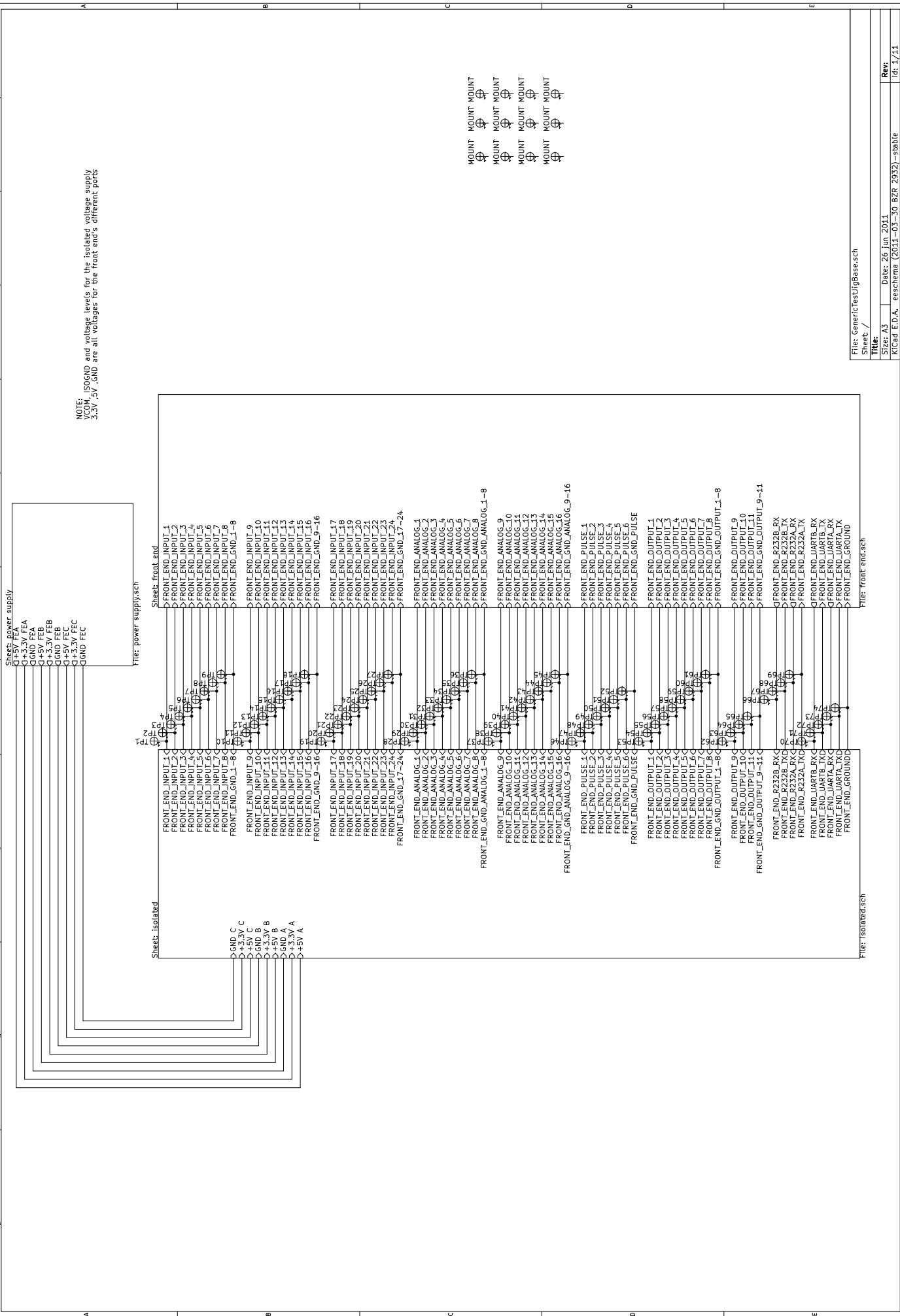
APPENDIX B.1

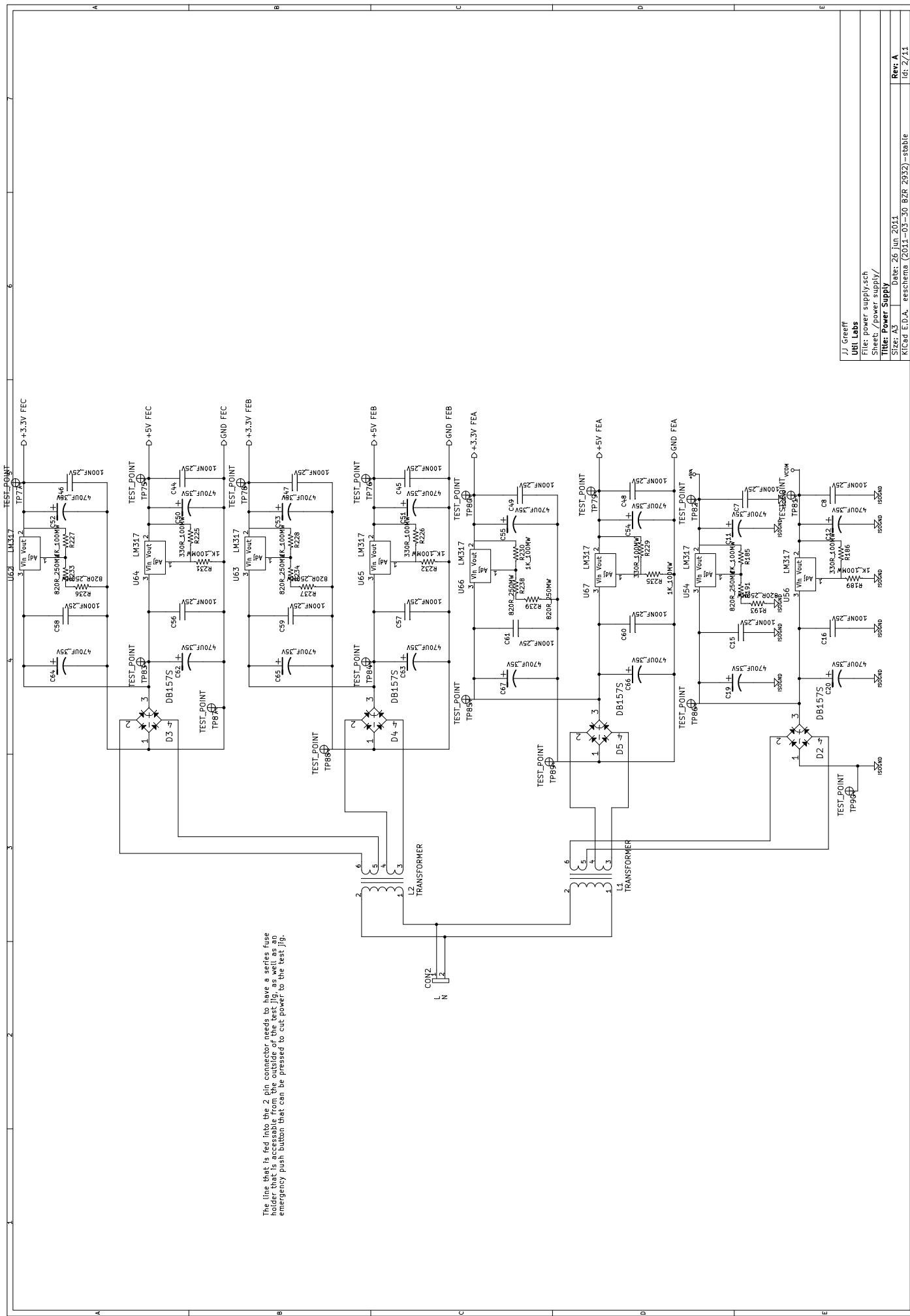
This appendix contains the hardware design files for the prototype. The following schematics are contained in this appendix:

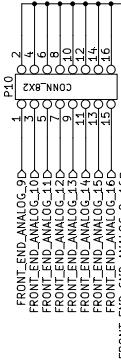
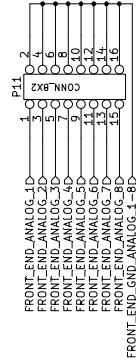
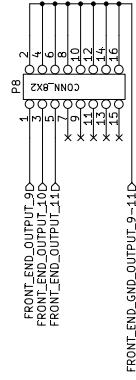
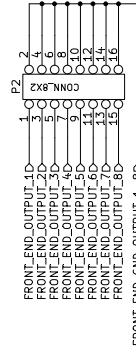
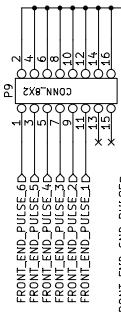
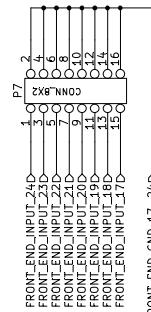
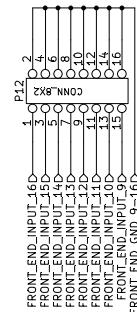
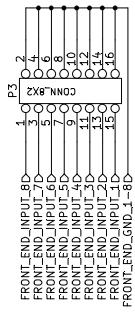
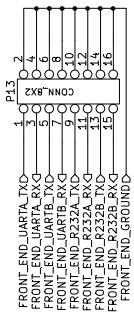
- Generic Test Jig Base Board version 1 (11 pages)
- Generic Bed Of Nails Base Version 1 (2 pages)
- Generic Bed Of Nails Base Version 2 (1 page)

The gerbers for the designs are available in:

- On project disk /DesignFiles/Hardware/Prototype/GenericBaseV1Gerbers.zip
- On project disk /DesignFiles/Hardware/Prototype/GenericBedV1Gerbers.zip
- The second version of the generic bed in this shape was never made into a pcb.



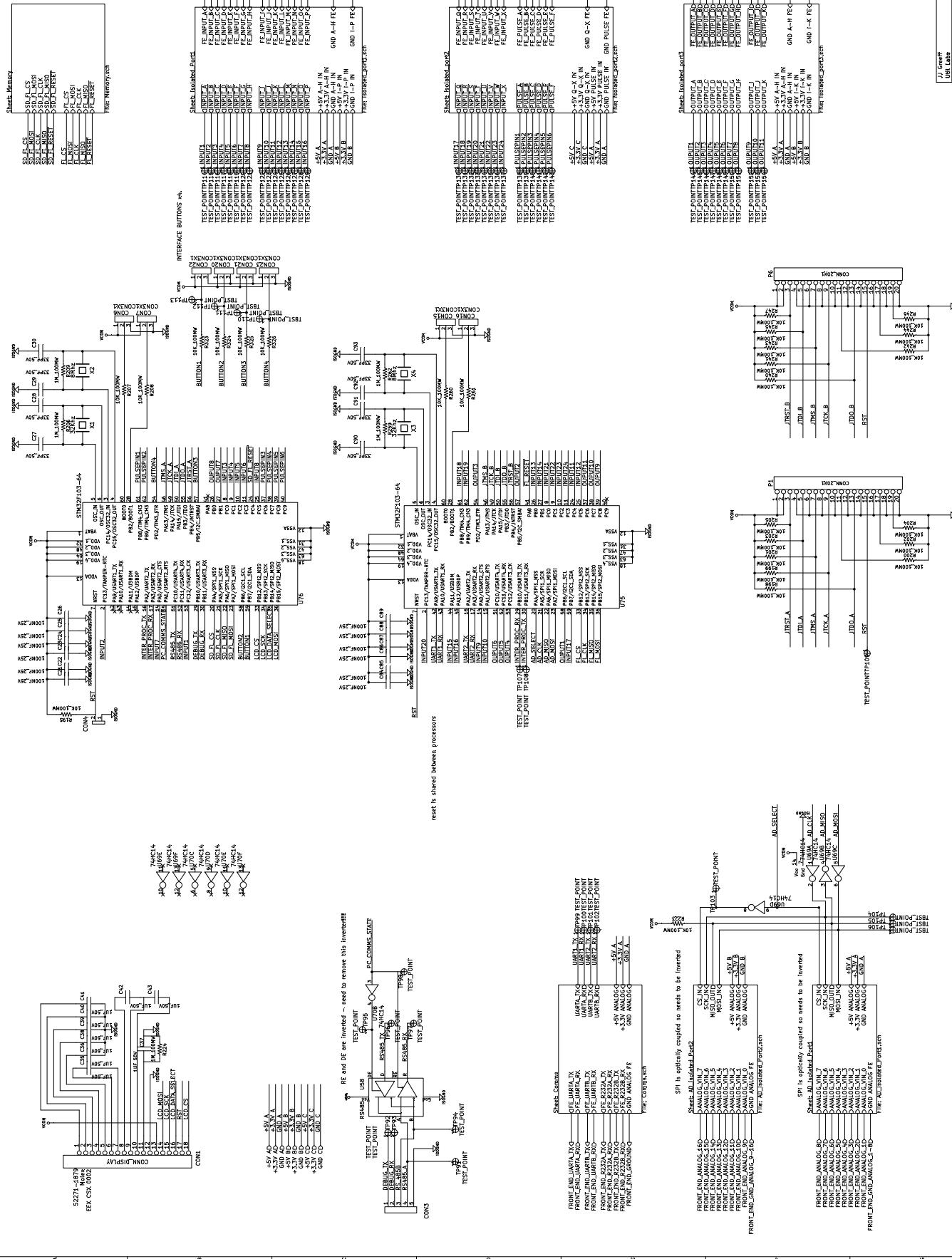


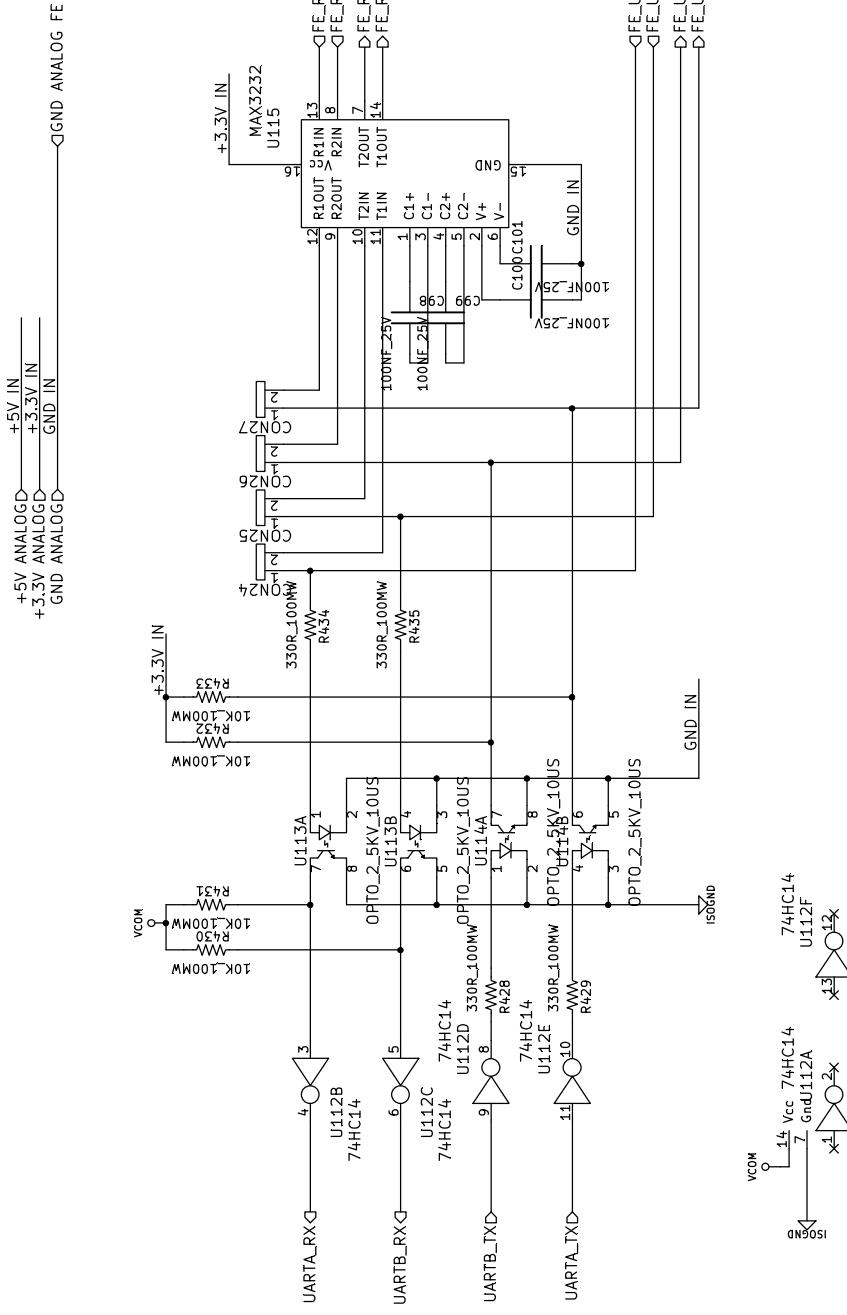


NOTE:
 ANALOG PORT 1 AND DIGITAL PORT 1 SHARE A GROUND,
 ANALOG PORT 2 AND DIGITAL PORT 2 SHARE A GROUND,
 PULSE INPUT AND DIGITAL PORT 3 SHARE A GROUND

J.J. Greeff	Rev. A
UH! Labs	
Title: Front end /	
Sheet: / front end /	
Title: Front End	Date: 26 Jun 2011
Size: A3	eeschema (2011-03-30 BZR 29.22)-stable
KICad EDA.	
	Id: 3/11

THE JOURNAL OF CLIMATE





JJ Greeff
Util Labs

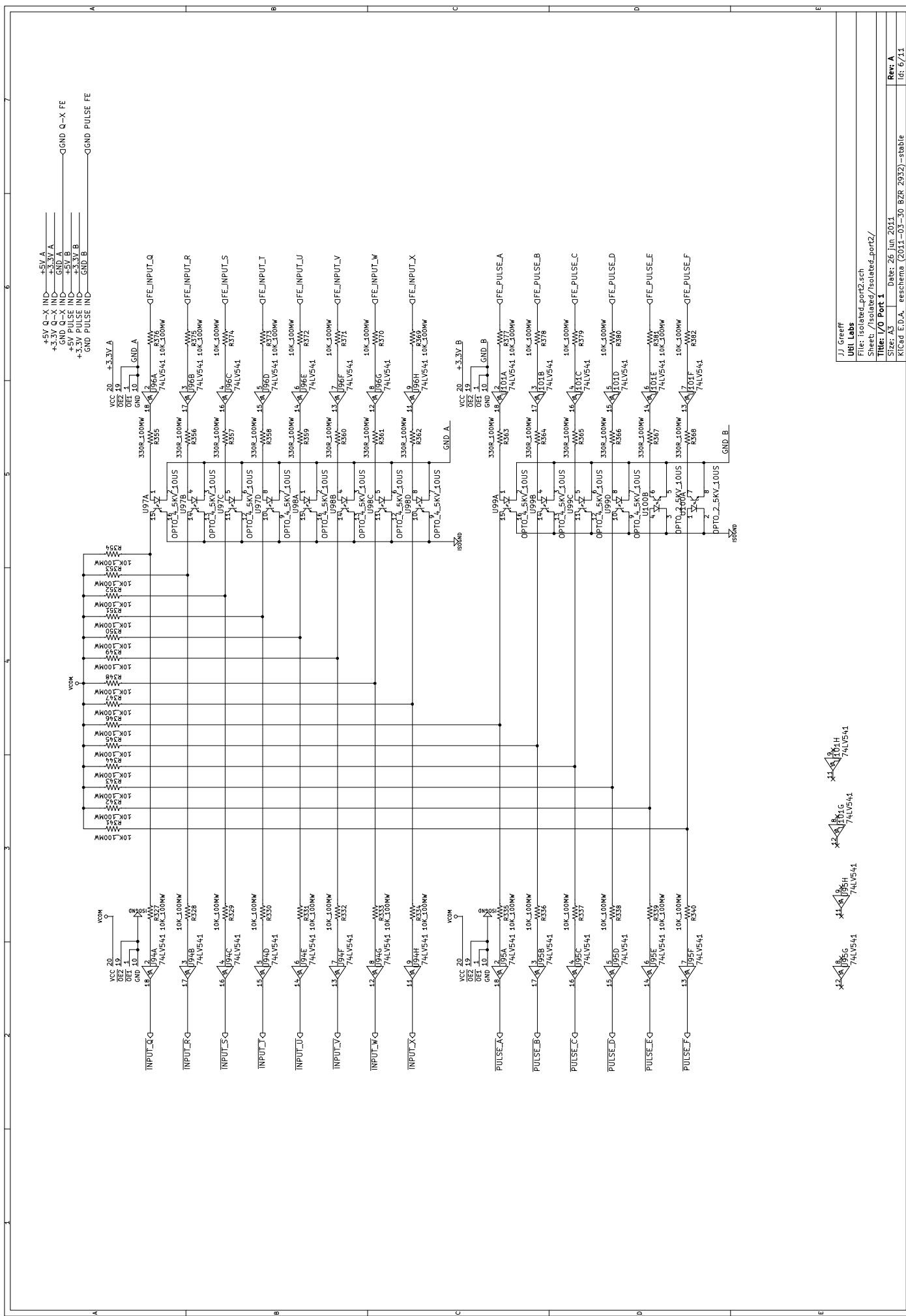
Sheet: /isolated/Comms/
Title: Comms

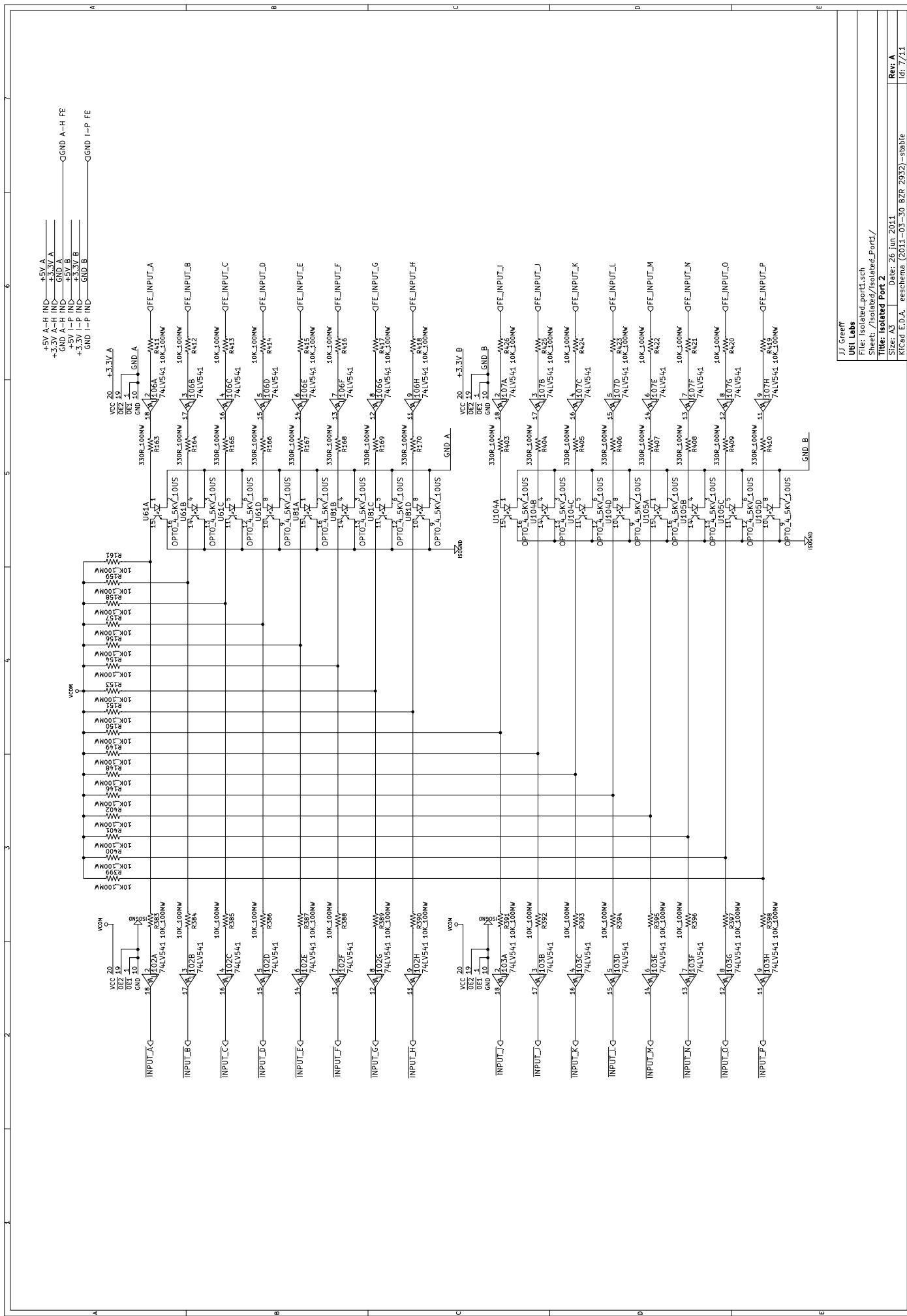
Title: Comments Date: 26

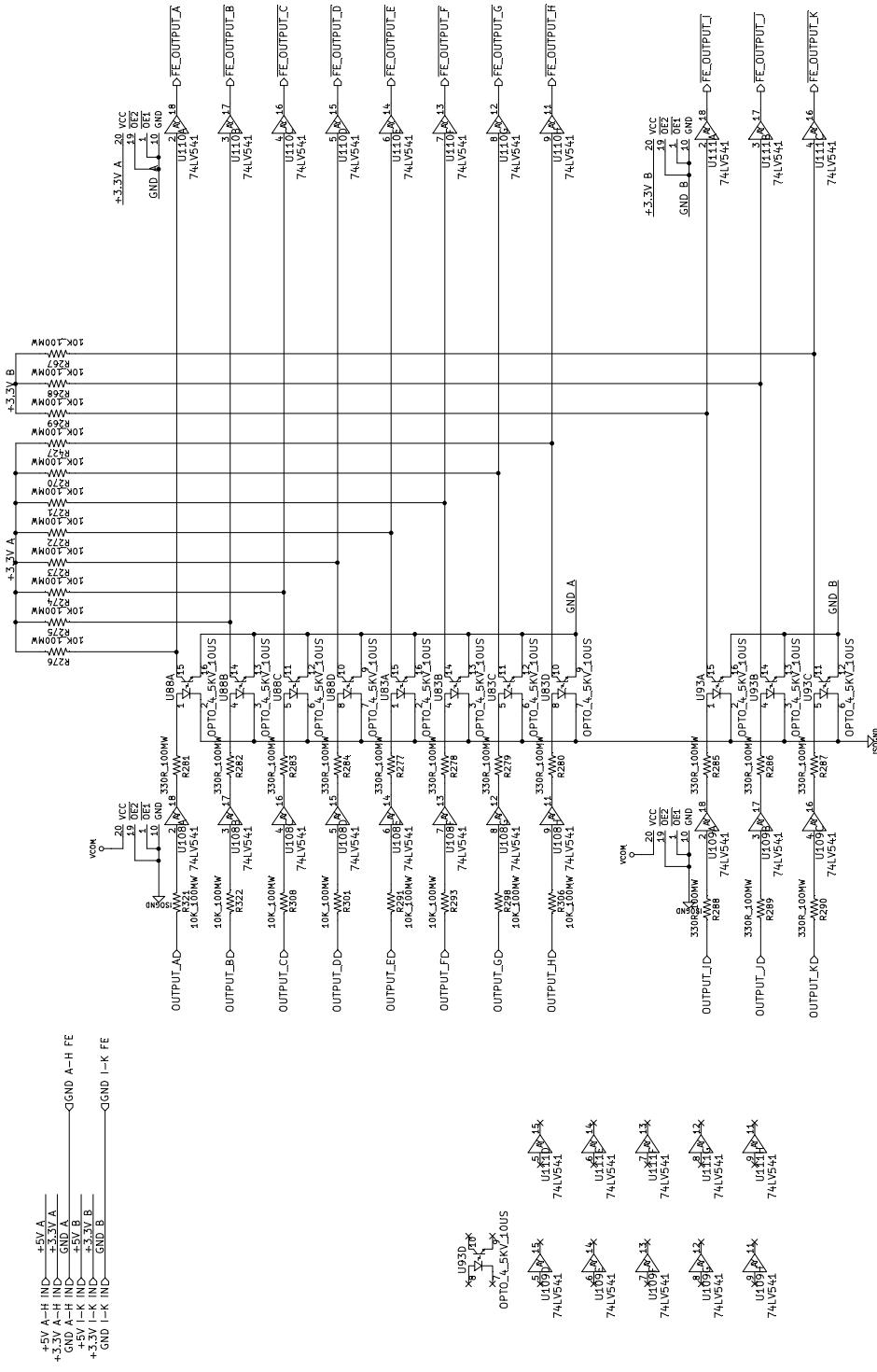
KisCad EDA eschewa (2)

ALCABU E.U.A. EESTIKEELIA (Z
4

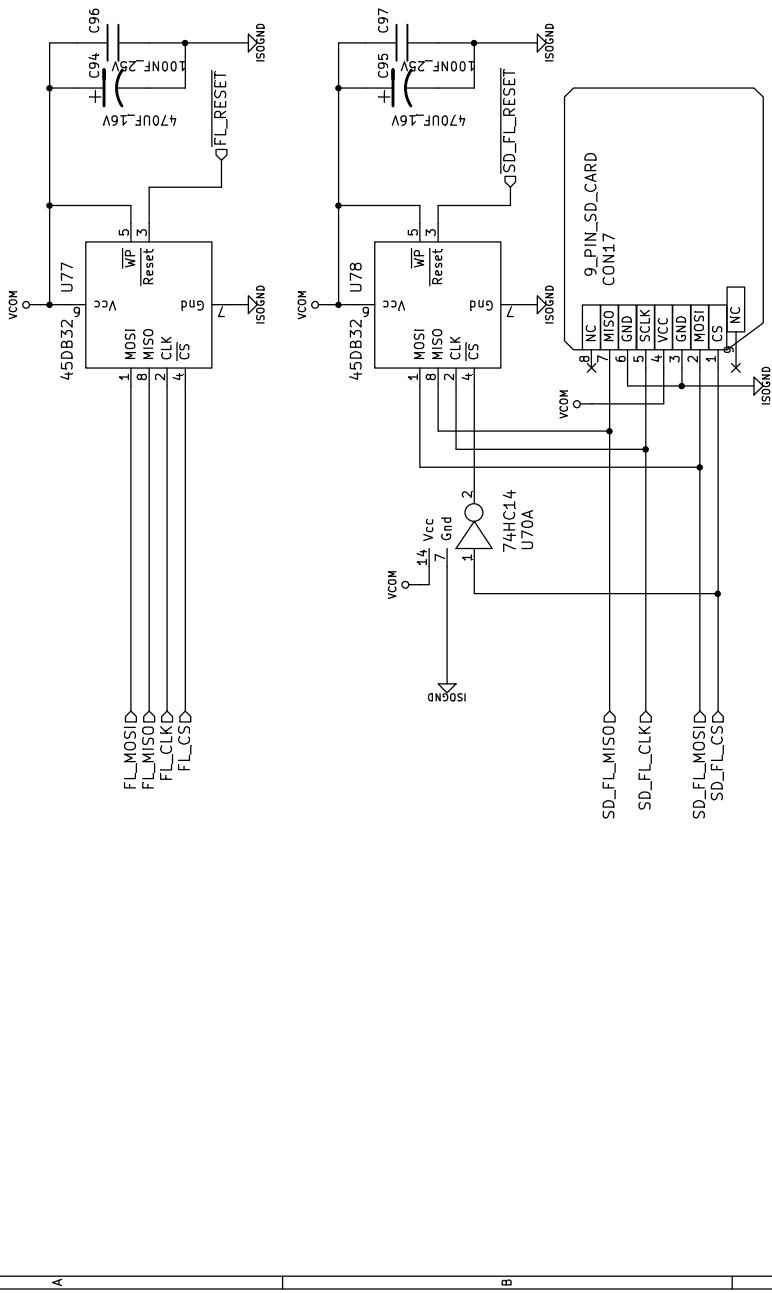
卷之三



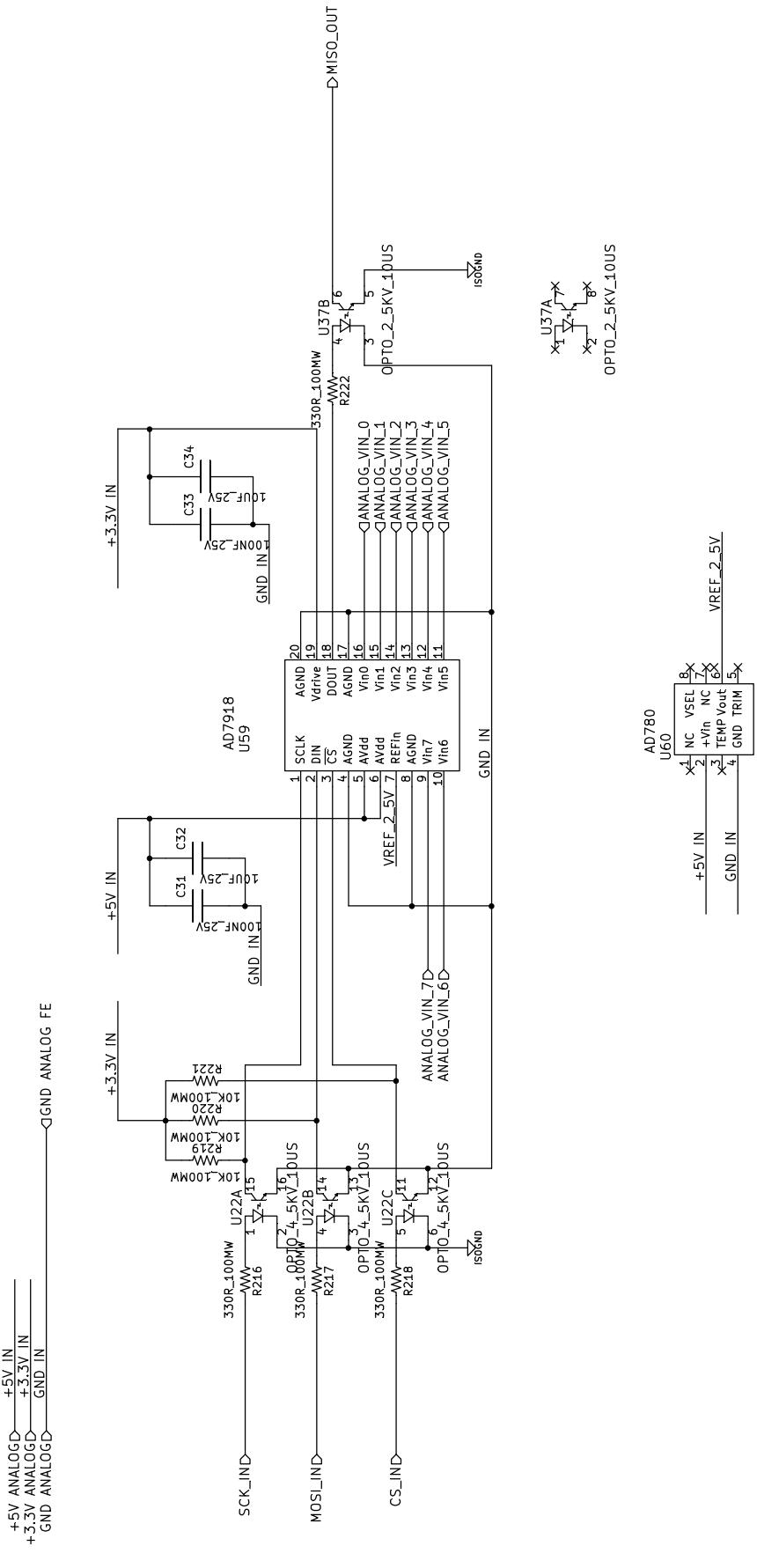




JJ Greeff
Util Labs
File: Isolate
Sheet: /1
Title: Isol
Size: A3
KICcad E D



JJ Greff		
Util Labs		
File: Memory.sch		
Sheet: /Isolated/Memory/		
Title: Memory		
Size: A4	Date: 26 Jun 2011	Rev: A
Kitcad E.D.A.	eescinema (2011-03-30 BZR 2932)-stable	Id: 9/11



NOTE: The Temp output is 560mV @ 25°C and the temperature coefficient is $2mV/^\circ C$. Output voltage is 2.5V with VSEL NC and 3V if VSEL = GND

JJ Greeff
Hilfe Lektorat

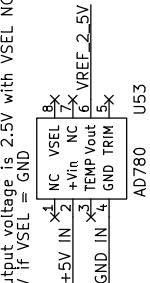
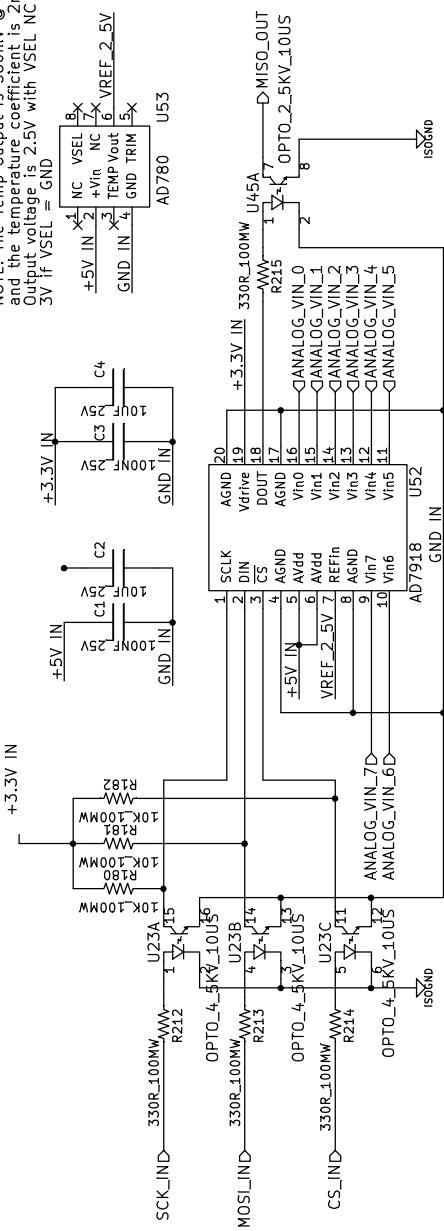
Util Labs
File: AD_Isolated_Port2.sch
Sheet: /isolated/AD_Isolated_Port2/

Title: AD Port 2

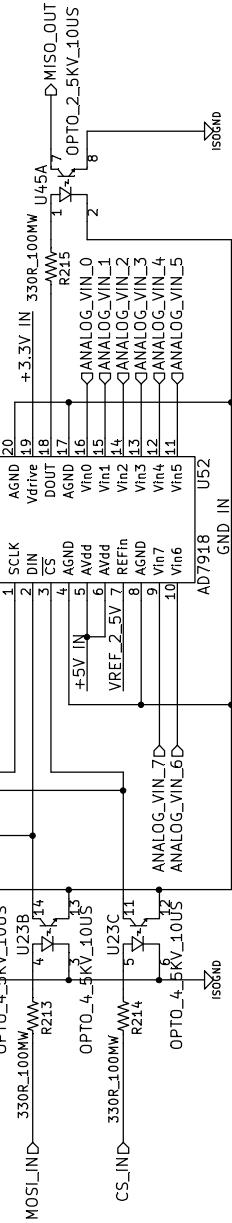
thema (2011-03-30 BZR 2932)-stable | id: 10/1

2

+5V ANALOG
+5V IN
+3.3V ANALOG
GND IN
GND ANALOG FE



NOTE: The Temp output is 560mV @ 25°C and the temperature coefficient is 2mV/°C. Output voltage is 2.5V with VSEL NC and 3V if VSEL = GND.

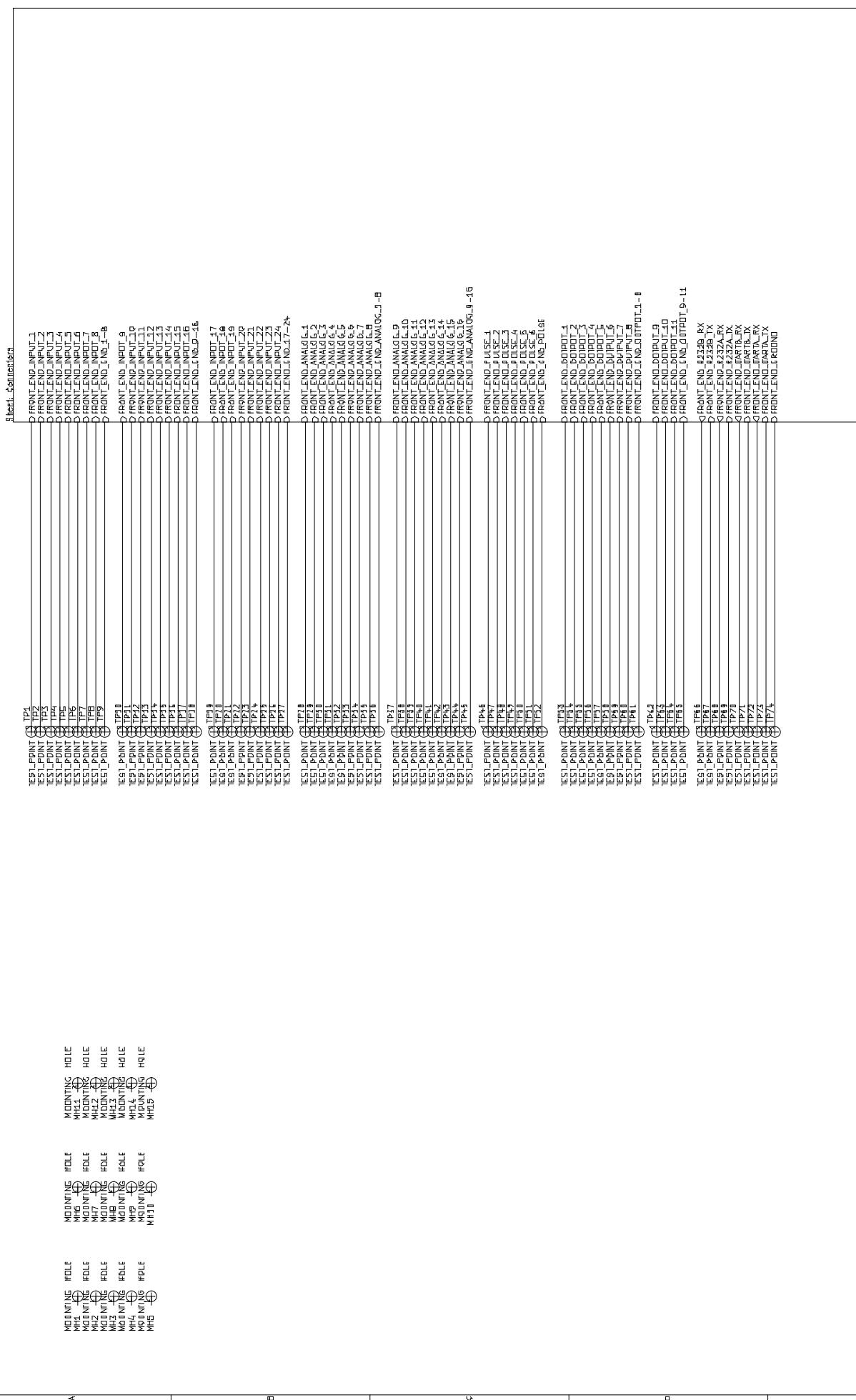


U45B
X₄
X₅
X₆
X₇
X₈
U23D
OPTO_4_5KV_10US

JJ Greff
Util Labs
File: AD_Isolated_Port1.sch
Sheet: /Isolated/AD_Isolated_Port1/
Title: A/D port 1
Size: A4 Date: 26 Jun 2011
KiCad E.D.A. eescinema (2014-03-30 BZR 2932)-stable
Rev: A
Id: 11/11

U45A
R215
OPTO_2_5KV_10US

1 2 3 4 5 6 7 8



הניב העממי

Orilabs
File: Bedofnails.sch
Sheet: /
Title: Bed Of Nails -
Sizes A3 **B**
Klond E.D.A. **esch**

7

6

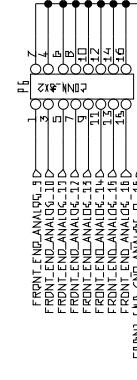
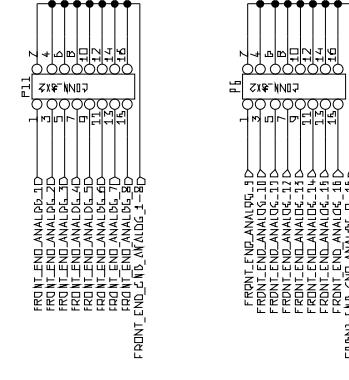
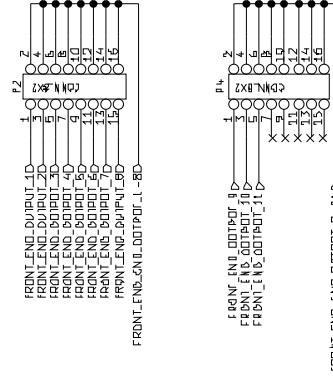
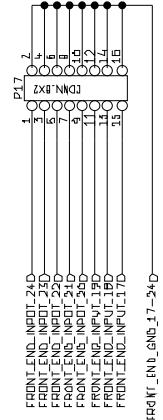
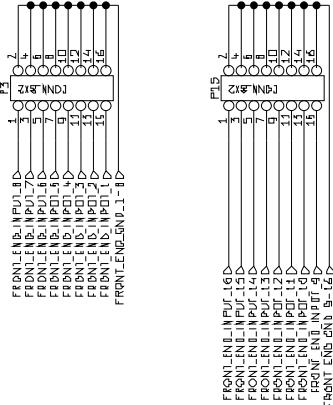
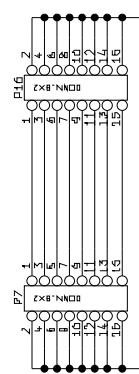
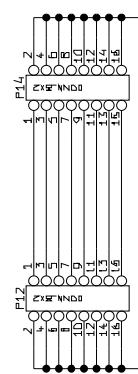
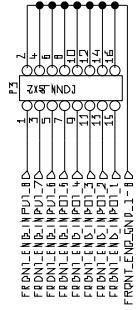
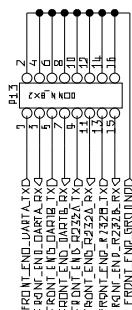
5

4

3

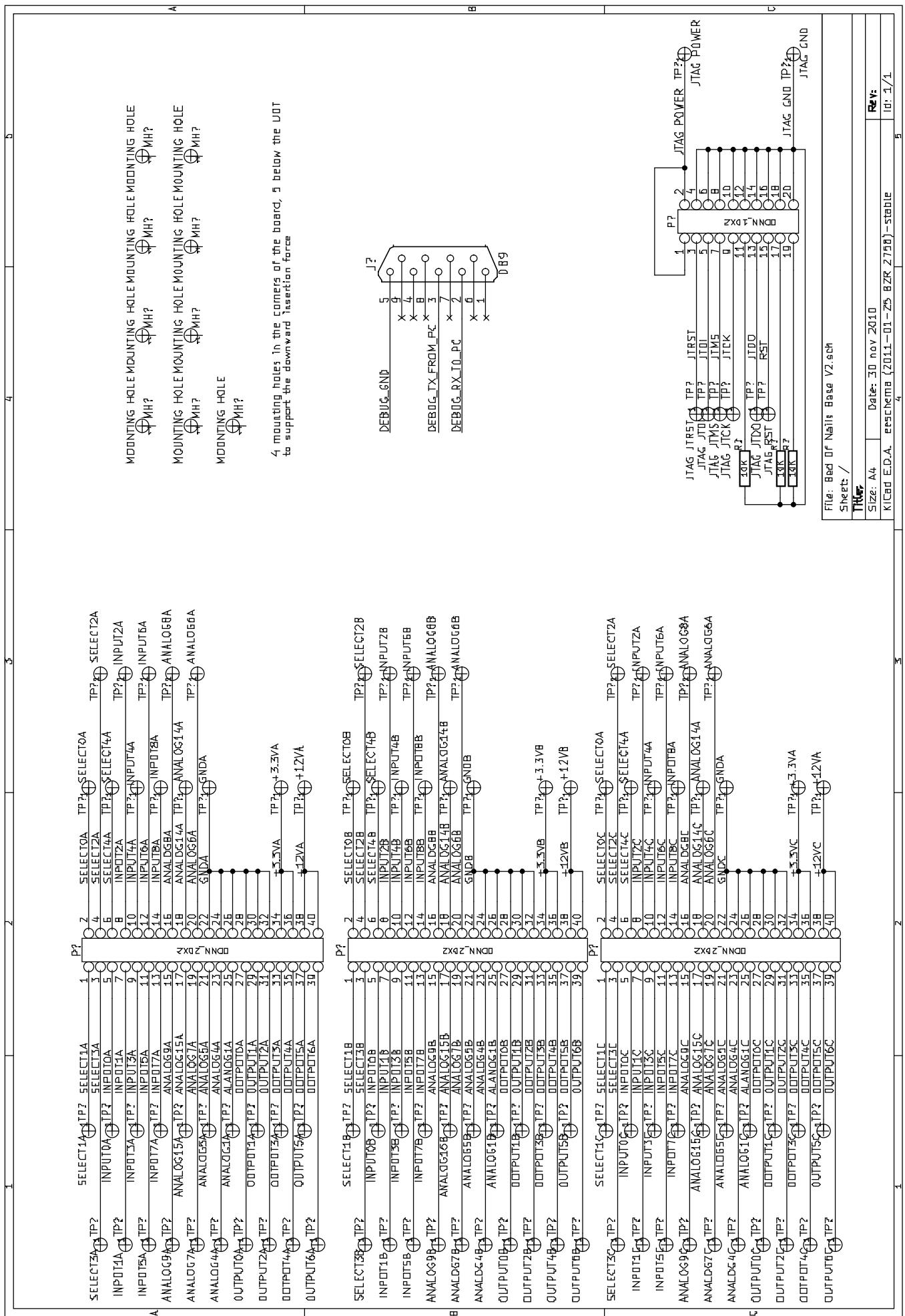
2

1



NOTE:
ANALOG PORT 1 AND DIGITAL PORT 1 SHARE A GND.
ANALOG PORT 2 AND DIGITAL PORT 2 SHARE A GND.
PULSE INPUT AND DIGITAL PORT 3 SHARE A GND.

Rev A	Rev B
Front End Board	Front End Board
File: Connectors.kch	File: Connectors.kch
Sheet: Connectors	Sheet: Connectors
Block: Front End	Block: Front End
Date: 30 Oct 2010	Date: 30 Oct 2010
KIT Ref: EDA Revision: 0	KIT Ref: EDA Revision: 0
Ver: A	Ver: A
Design Date: [2011-01-29] 27581 - date file	Design Date: [2011-01-29] 27581 - date file
Rev A	Rev A
12 / 2 /	7



APPENDIX B.2

This Appendix contains the firmware snippets for the designs in the prototype. The following files are used:

- drv_Rs485.h and drv_Rs485.cpp
- brg_Rs485.h and brg_Rs485.cpp
- brg_processor.h and brg_processor.cpp
- brg_processorMaster.h and brg_processorMaster.cpp
- brg_processorSlave.h and brg_processorSlave.cpp

All of these files are available on the cd in:

[`/DesignFiles/Firmware/Prototype/PrototypeFirmware.zip`](#)

APPENDIX C

This appendix contains the hardware design files for Iteration 2. The following schematics are contained in this appendix:

- Generic Test Jig Base Board Rev B Schematic (4 pages)
- Mcu Bed Of Nails Base Rev A Schematic (6 pages)

The gerbers for the designs are available in:

- On project disk /DesignFiles/Hardware/Iteration 2/GenericBaseRevBGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 2/McuBedBaseRevAGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 2/McuBedStopperRevAGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 2/McuBedGuideRevAGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 2/McuBedOutlineRevAGerbers.zip

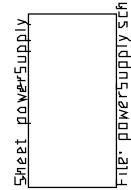
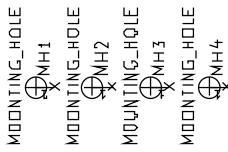
3

4

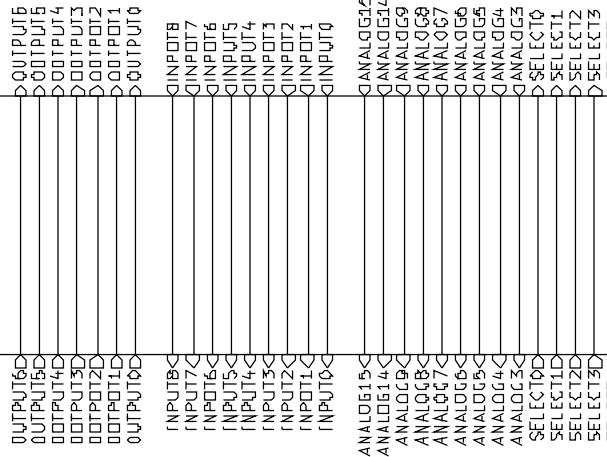
5

2

1

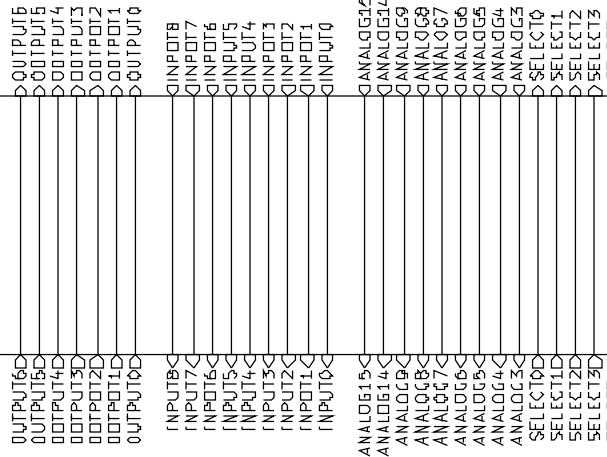


Sheet: processor



Title: processor.sch

Sheet: frontEnd



Title: frontEnd.sch

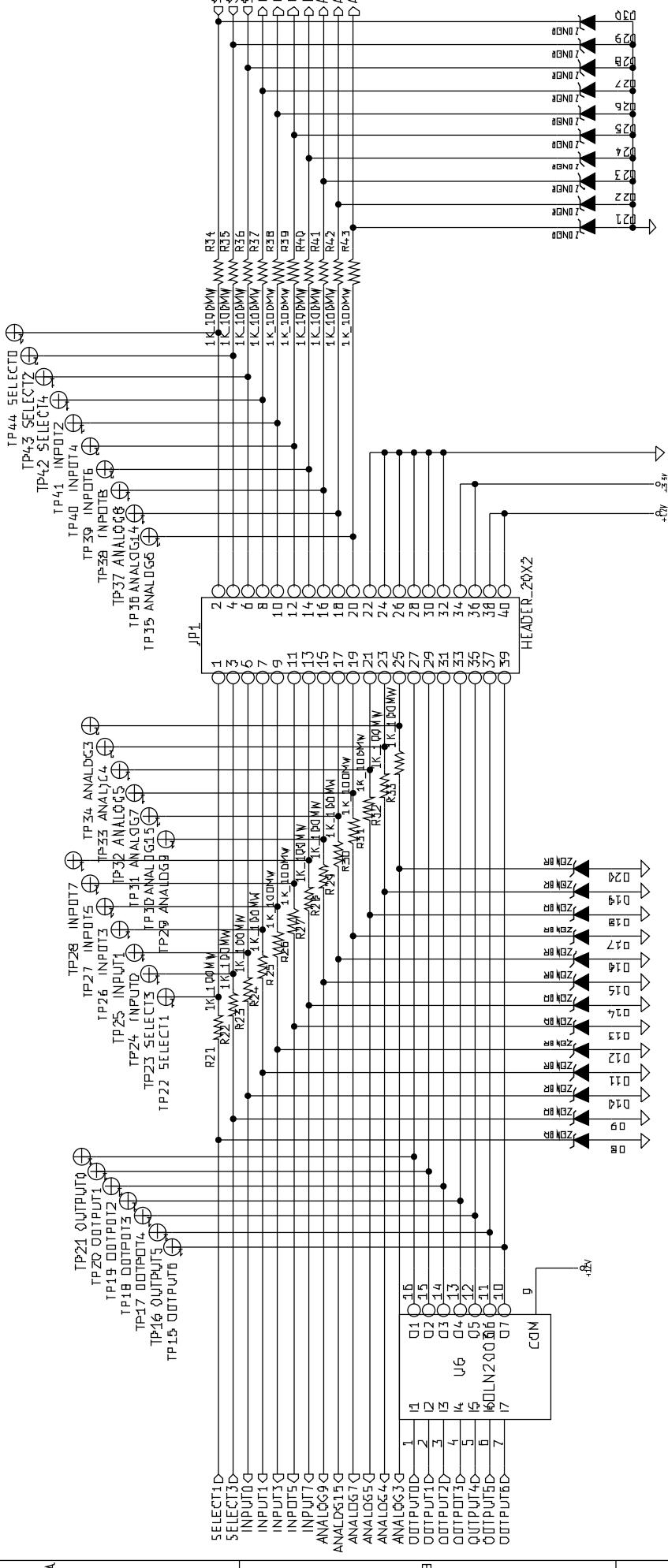
File: genericTestingV2.sch
Sheet: /
Title:

Size A4	Date: 5 Jan 2011	Rev: 1
KICAD E.O.A.	escheno (2011-07-D3)-testing	4

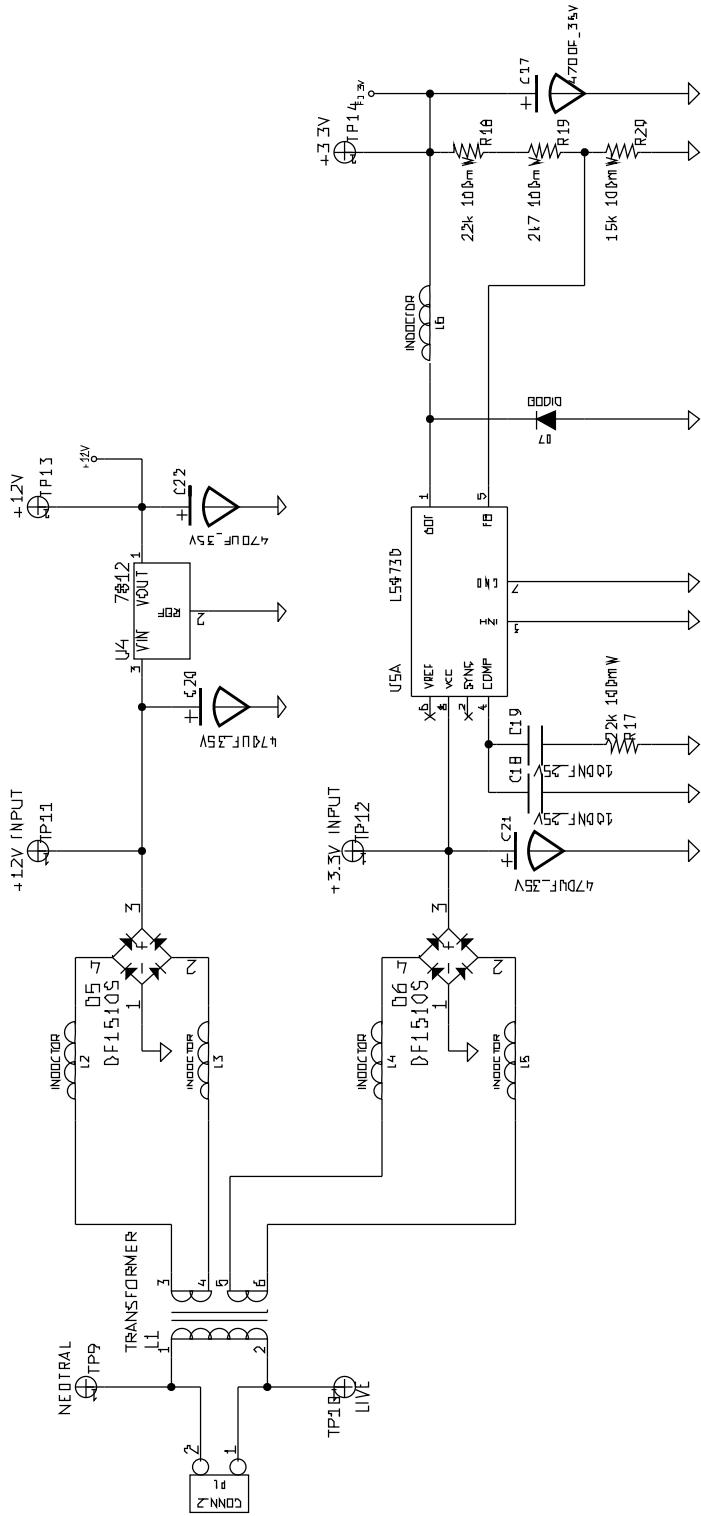
A

B

C



Re: V:	IG	Z/4
Title:		
Size A4	Date: 6 Jan 2014	
KICD ED.A. EPERM 4	PERM 2014-07-D3 -terling	E



File PowerSupply.sch
Sheet /PowerSupply/
Title

Size A4	Date 5 Jan 2011	Rev 10-3/4
KICD E.O.A. Revision (Z011-07-D3)-testing	4	5

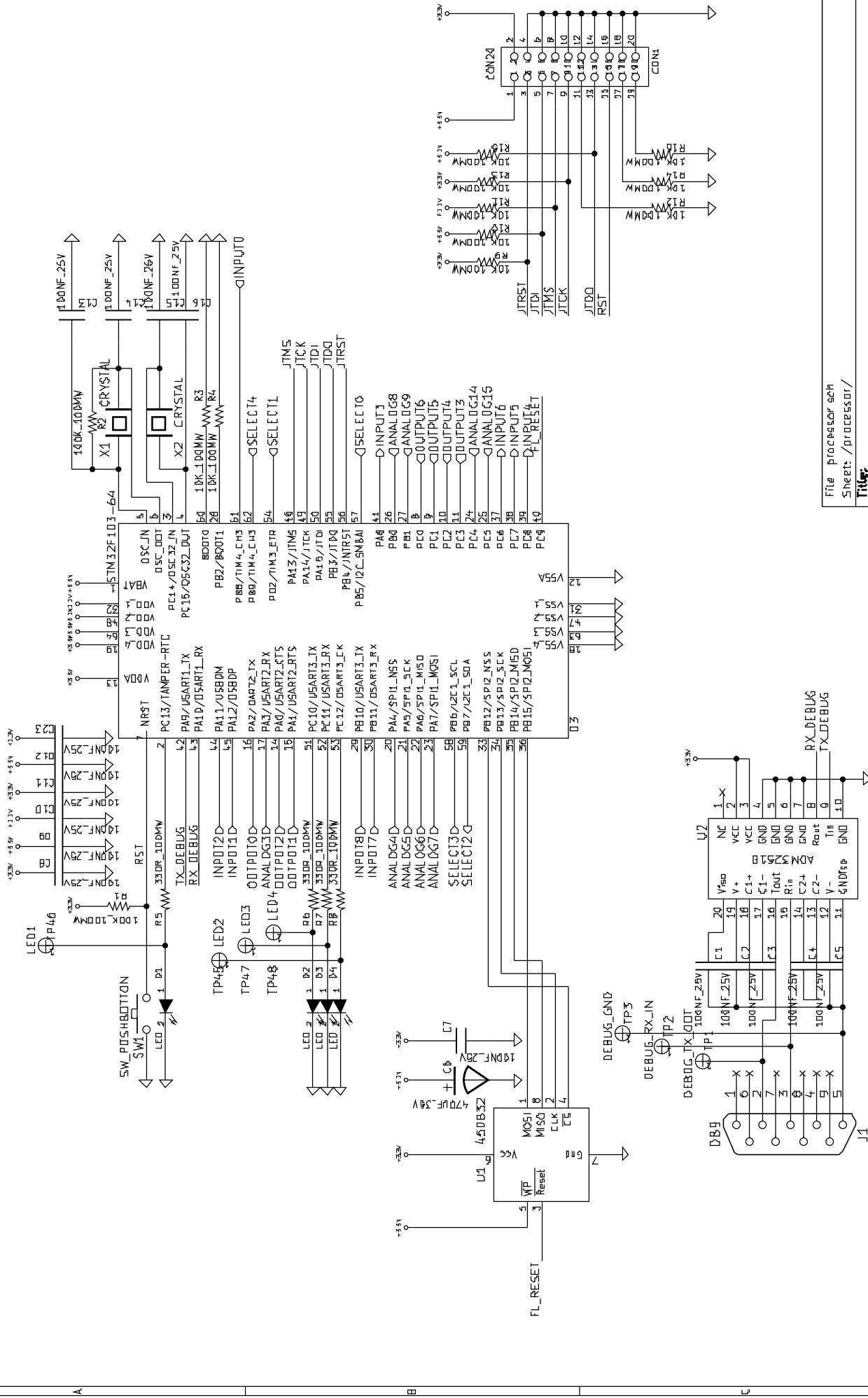
3

4

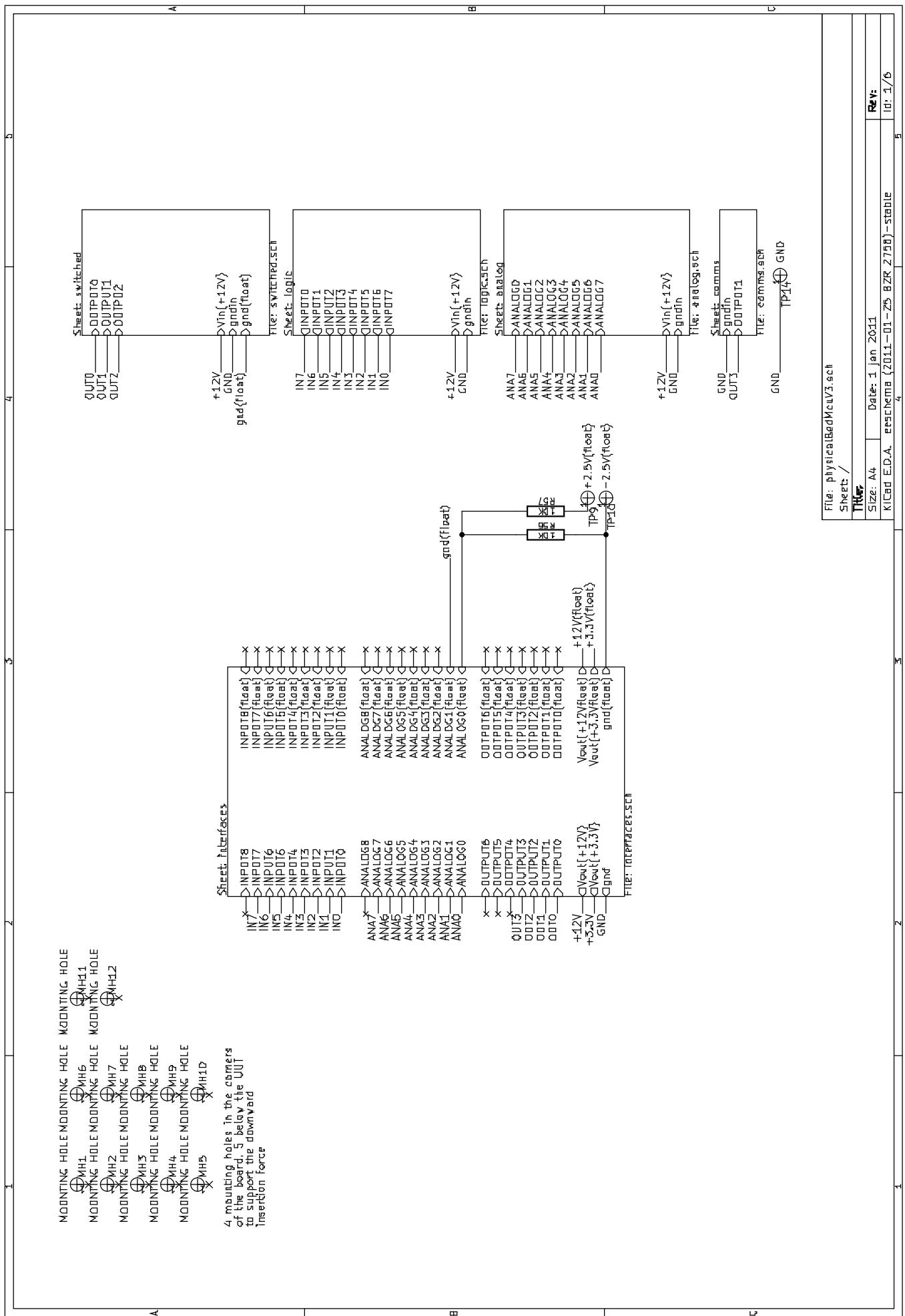
5

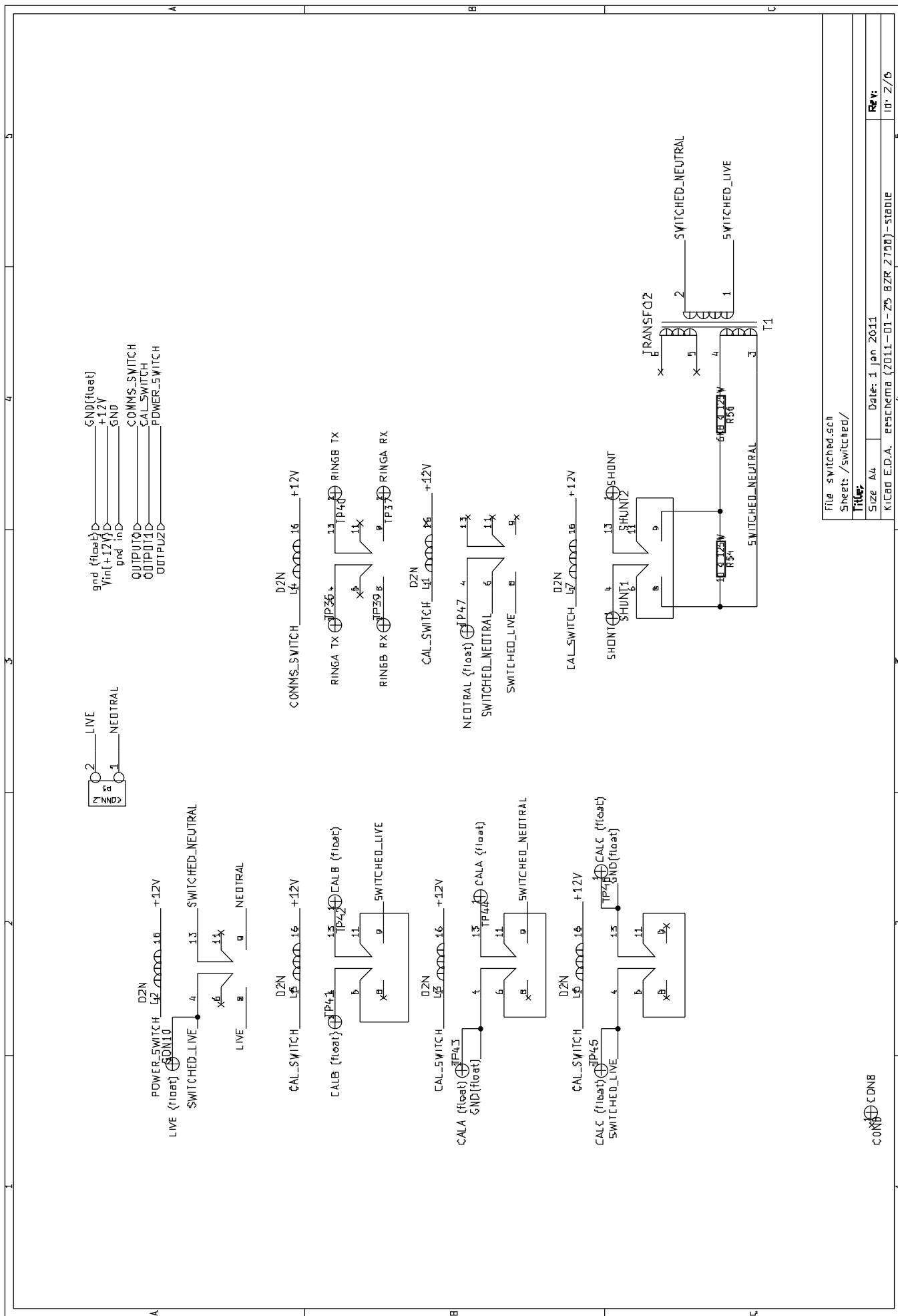
2

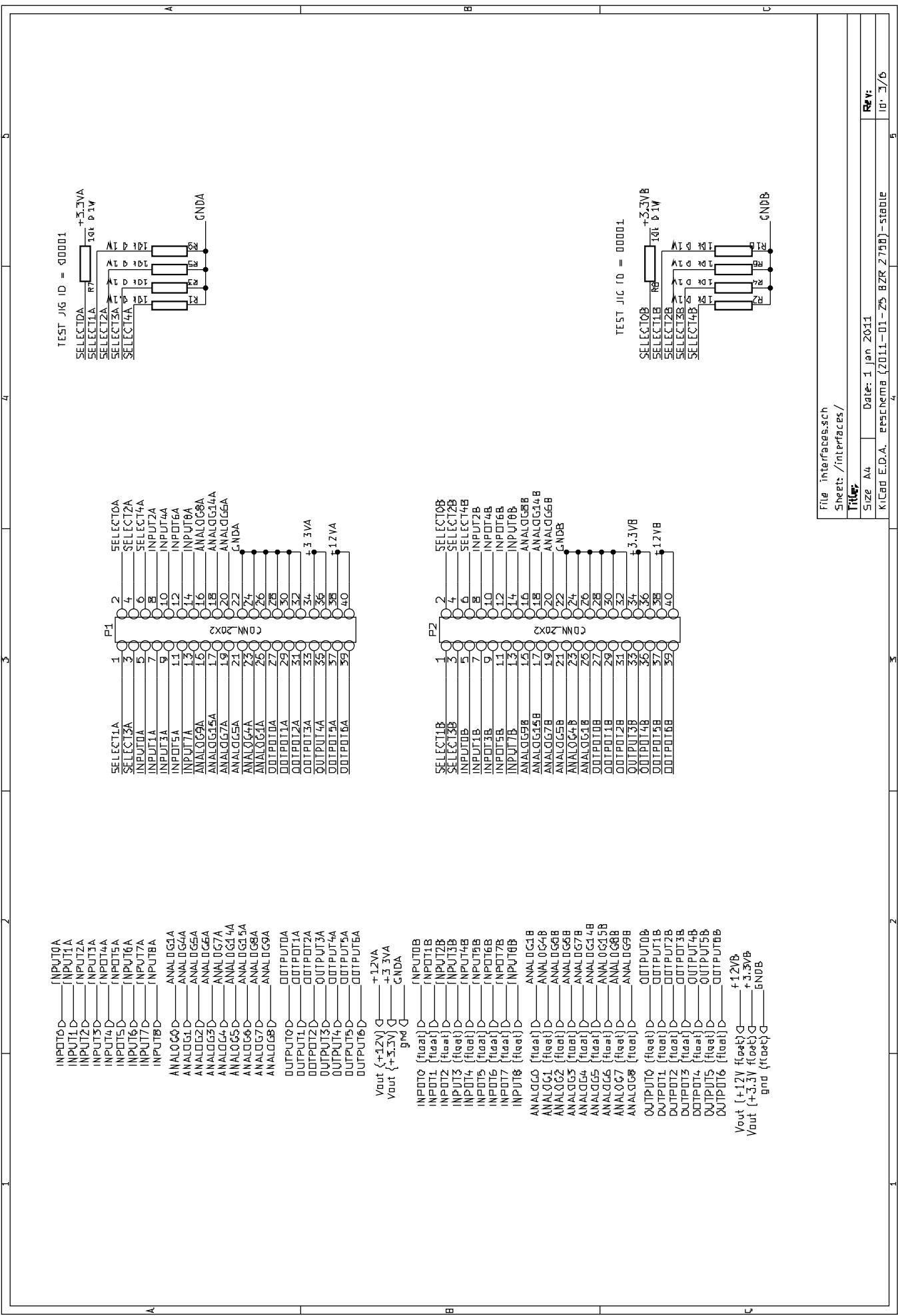
1

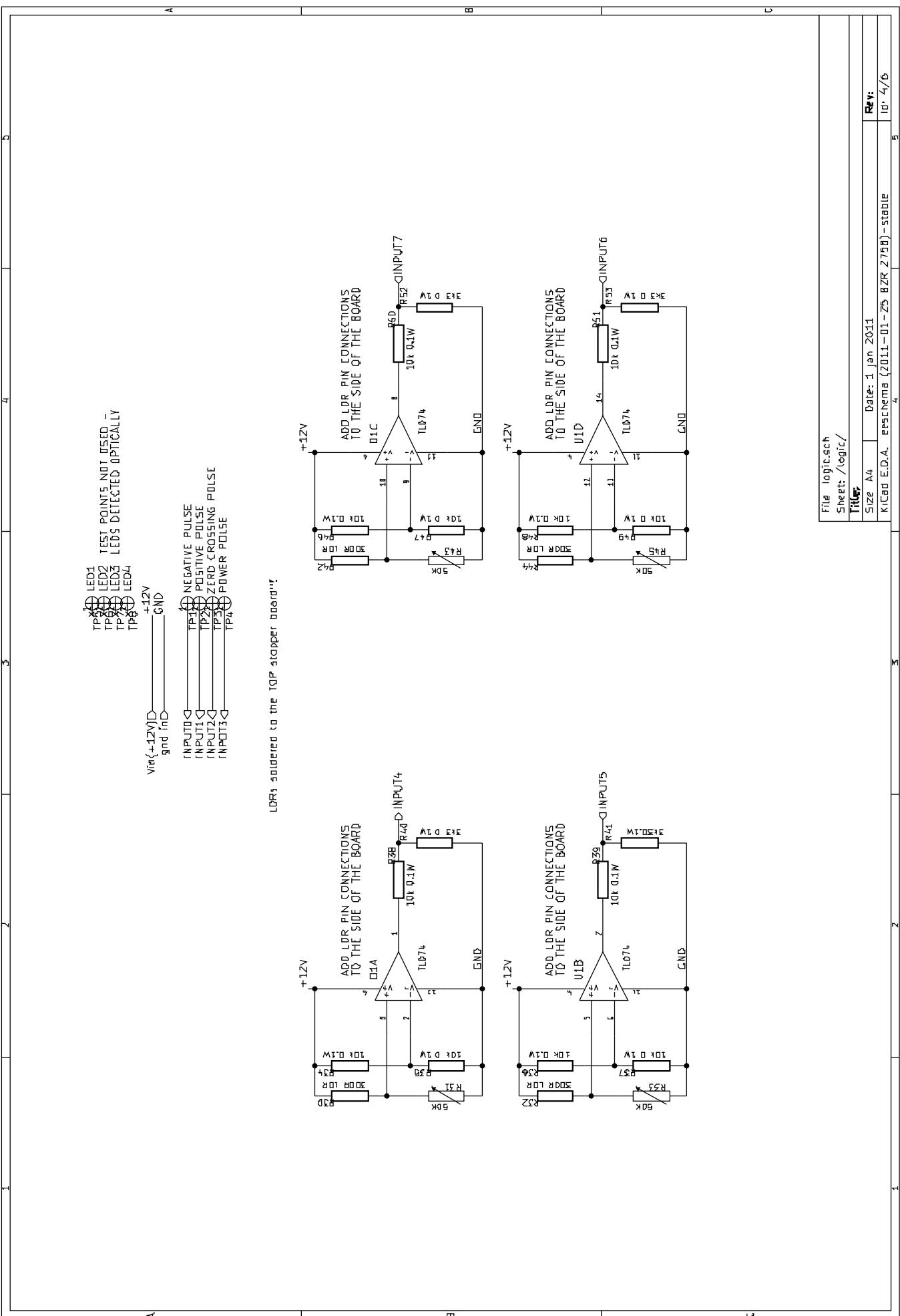


File: KICD_E.O.A_Schematic_01.dwg
Sheet: /01022501/
Title: KICD_E.O.A.
Size: A4
Date: 5 Jan 2011
Eraschen (2011-07-D3)-testing
Rev: 10-1/4





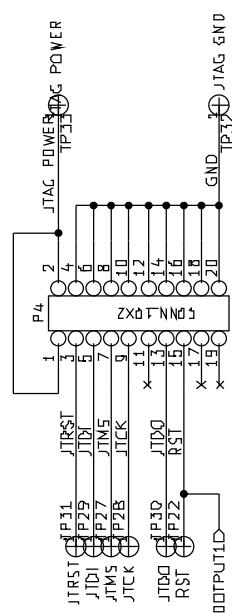




1 2 3 4 5

A

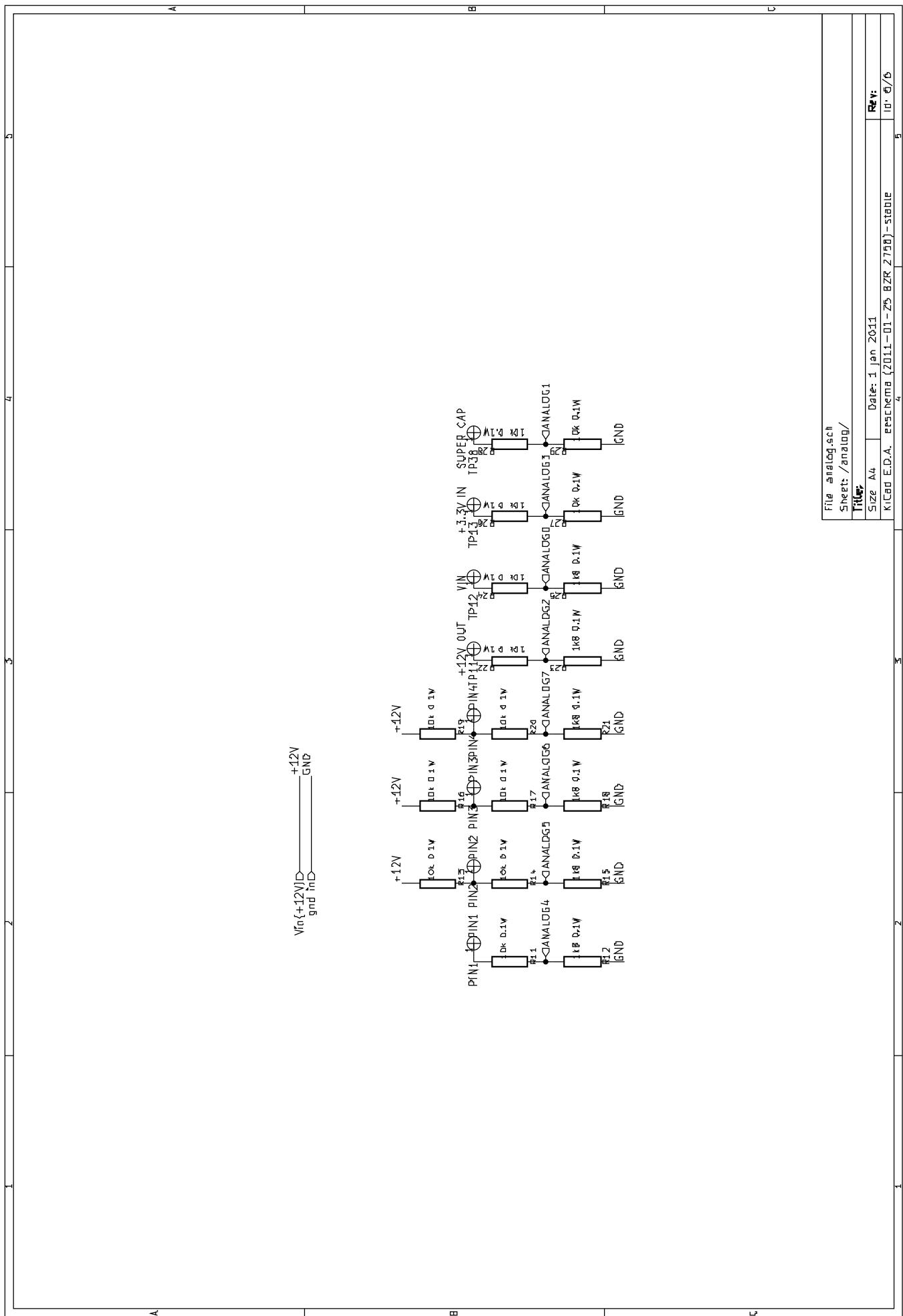
GND IN GND



A B C

B

File: comments.scl			
Sheet: /comments/			
Title:			
Size: A4	Date: 1 Jan 2011		Rev:
KICad EDA_EF5L1embo (2011-01-25 07:58) - stable			11-5/6
	4	5	
	2		
	3		
	1		



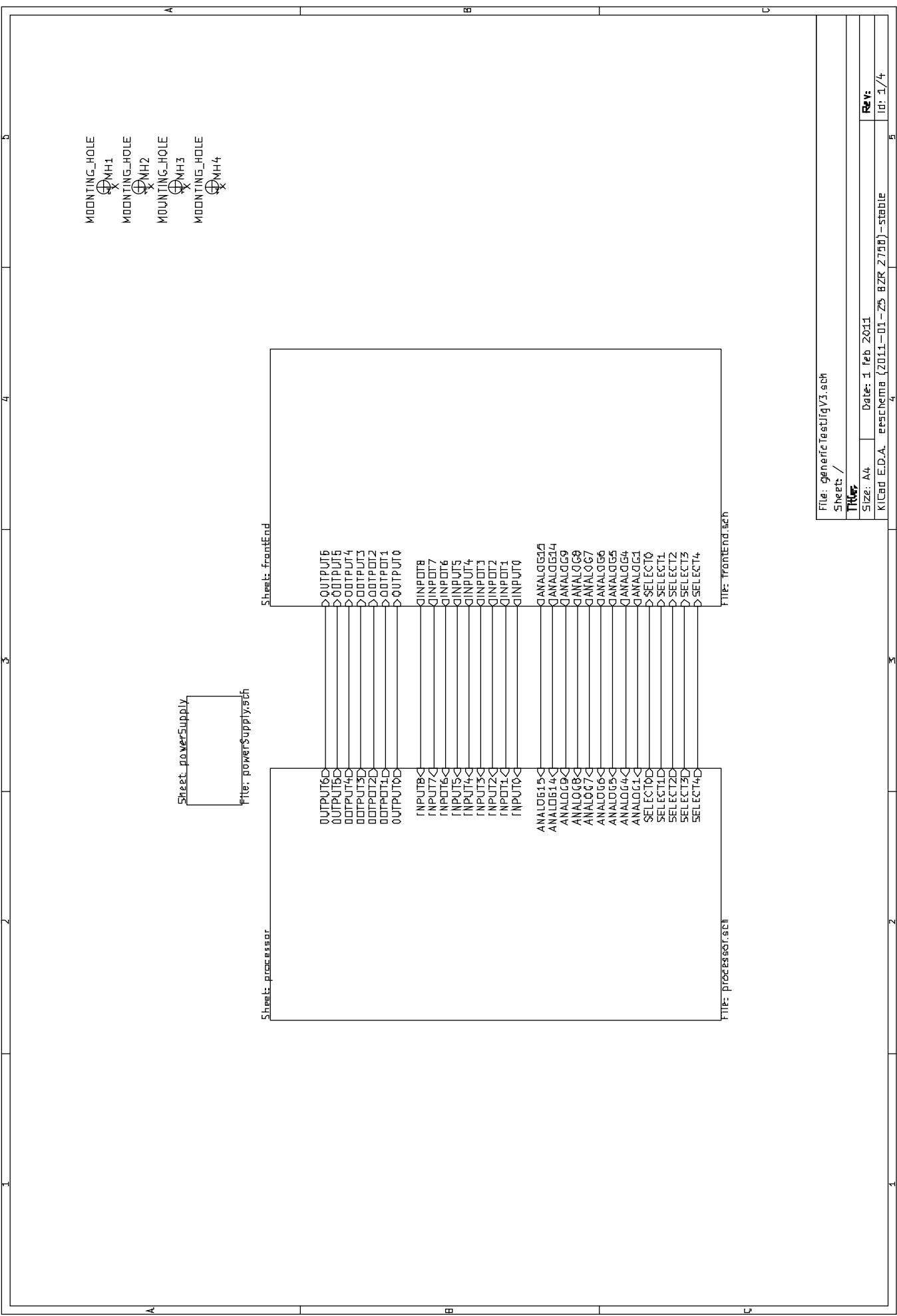
APPENDIX D

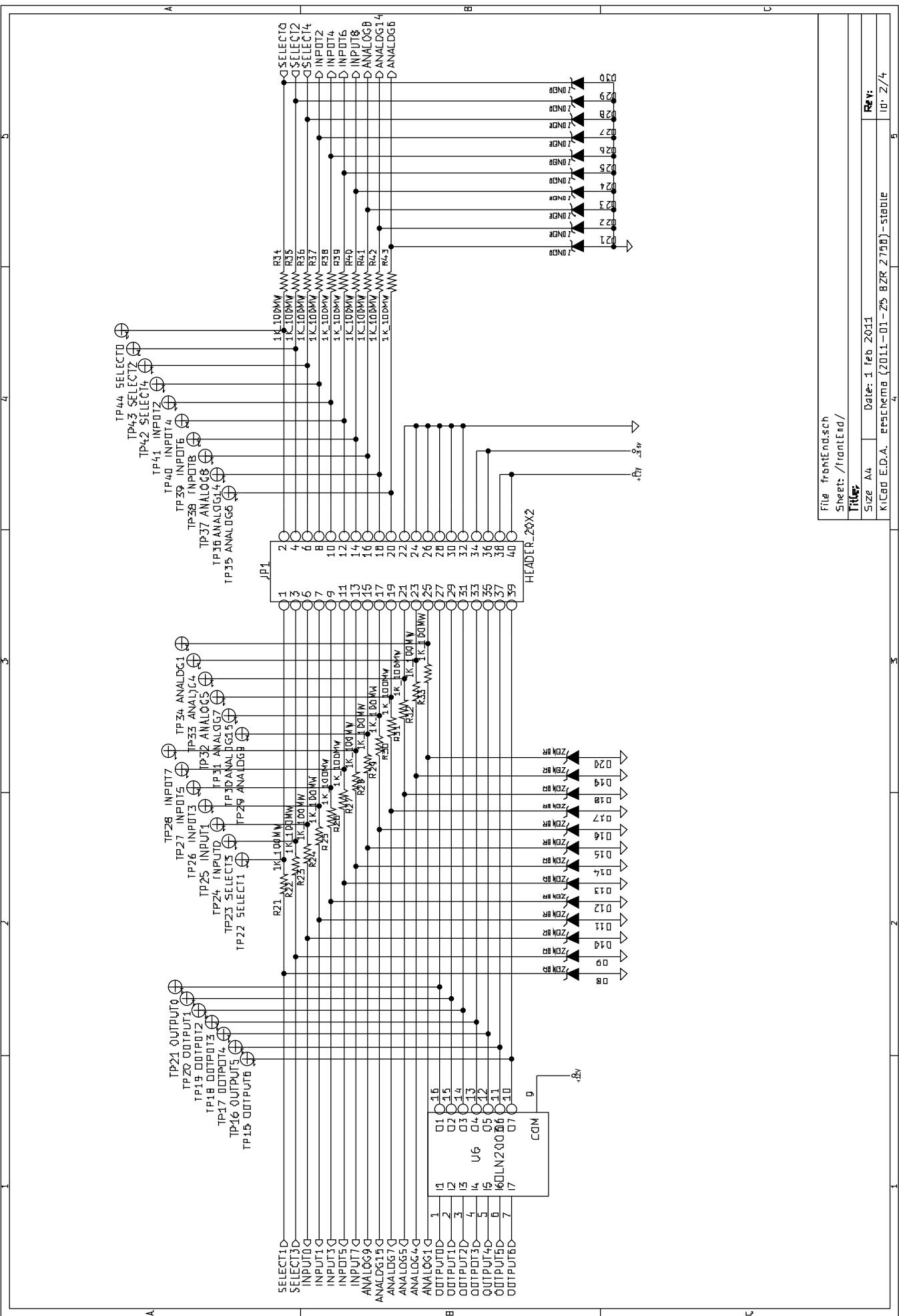
This appendix contains the hardware design files for Iteration 3. The following schematics are contained in this appendix:

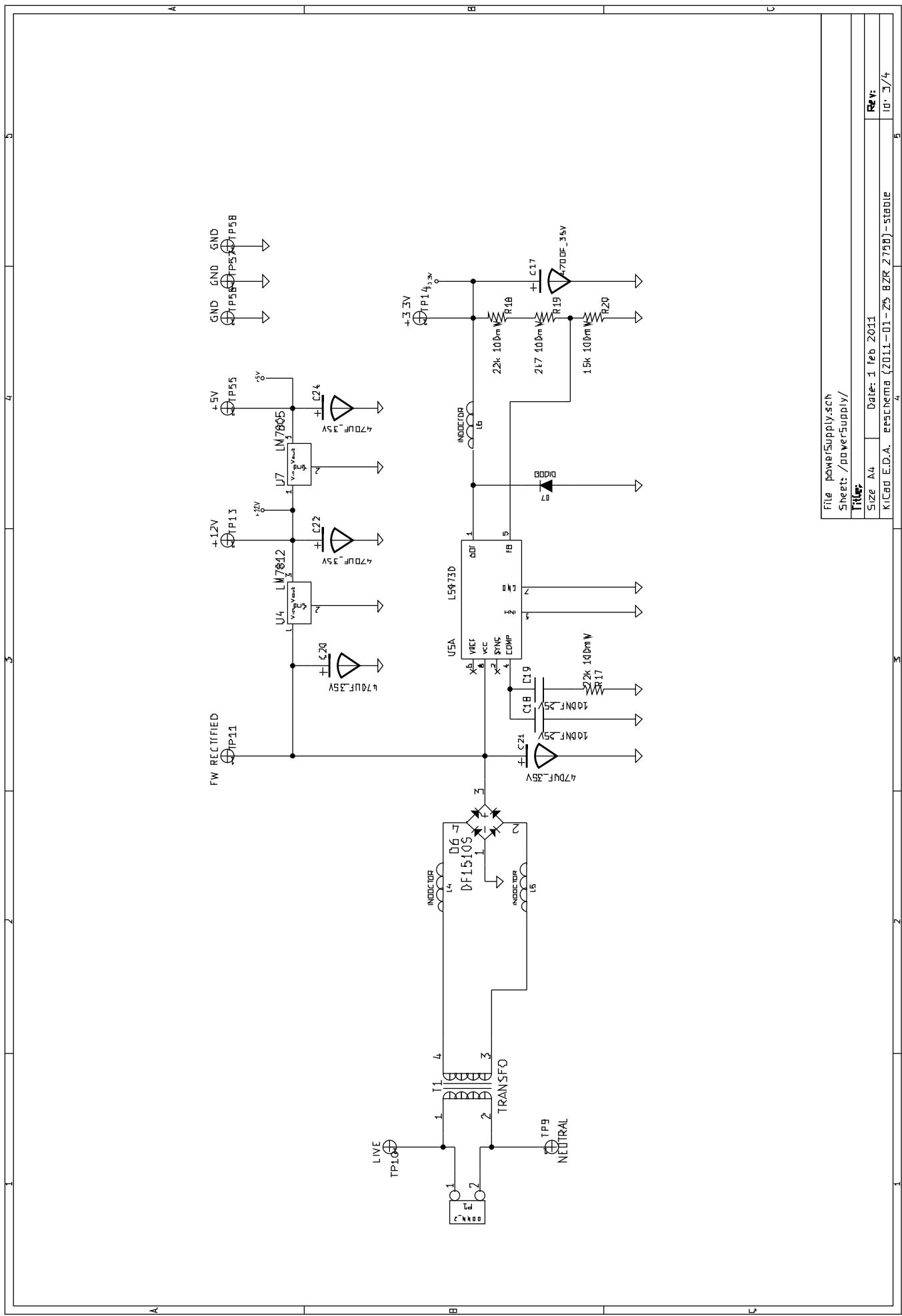
- Generic Test Jig Base Board Rev C Schematic (4 pages)
- Mcu Bed Of Nails Base Rev B Schematic (4 pages)
- Mcu Bed Of Nails Stopper Rev B Schematic (3 pages)

The gerbers for the designs are available in:

- On project disk /DesignFiles/Hardware/Iteration 3/GenericBaseRevCGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 3/McuBedBaseRevBGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 3/McuBedStopperRevBGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 3/McuBedGuideRevBGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 3/McuBedOutlineRevBGerbers.zip

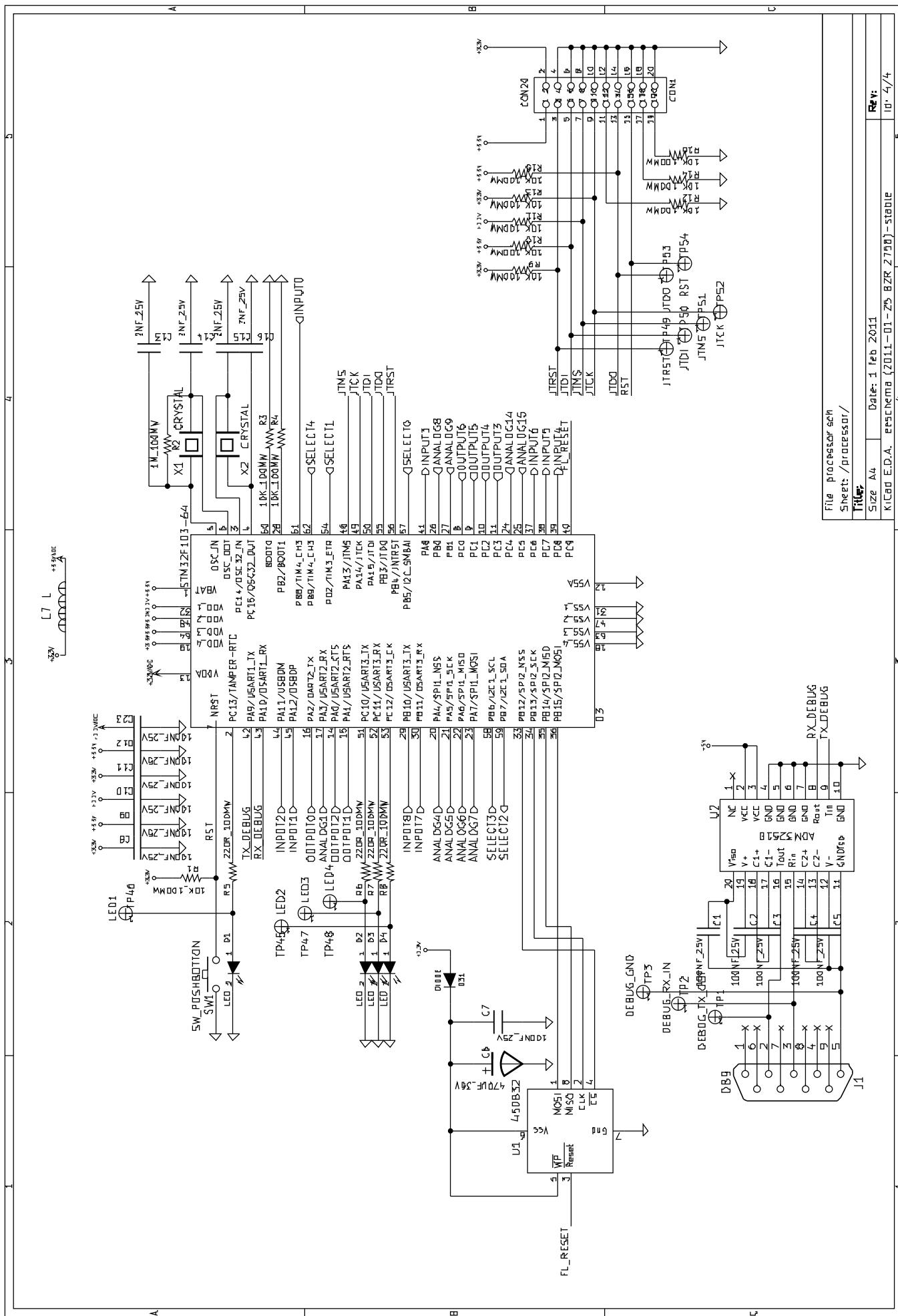


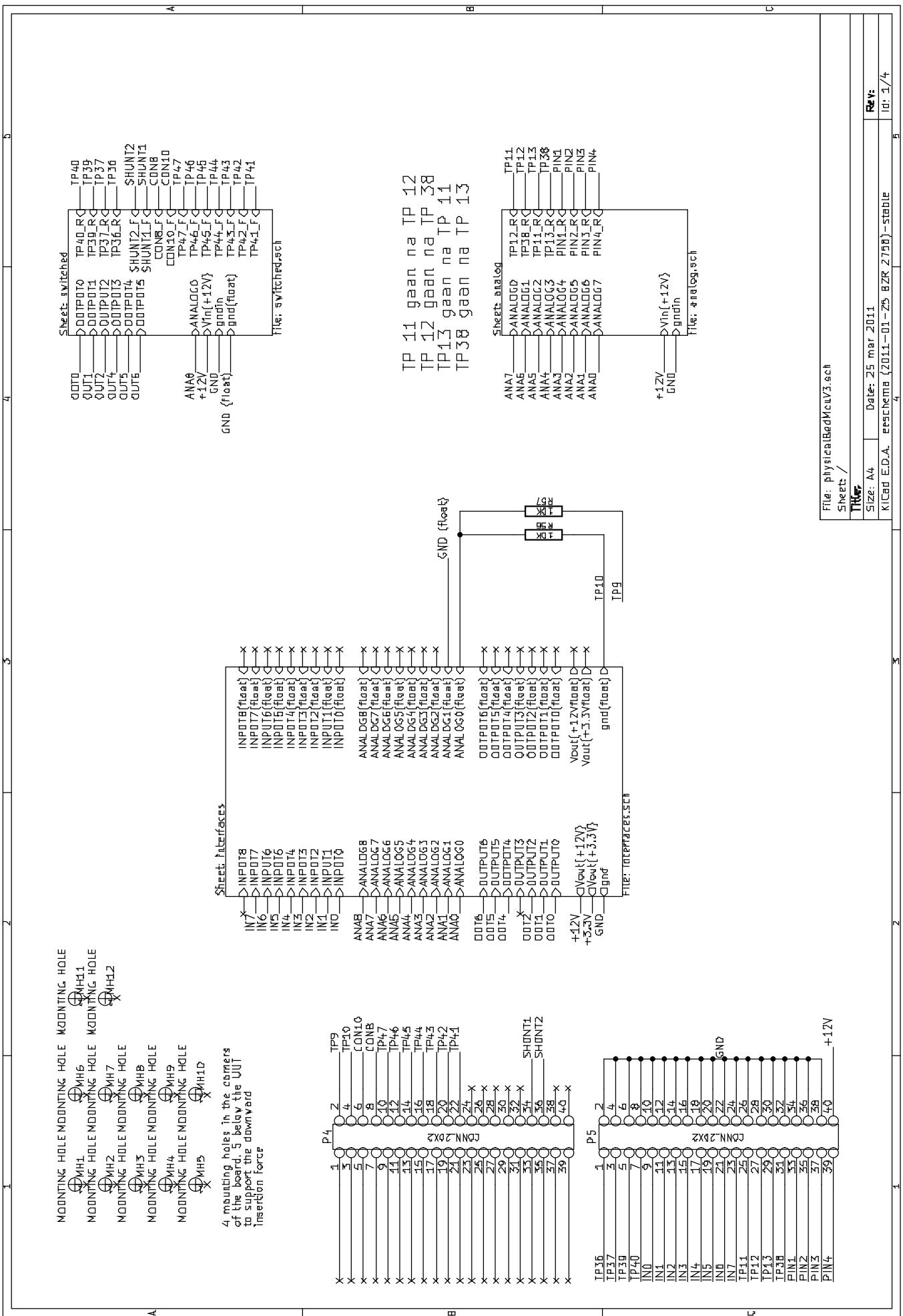


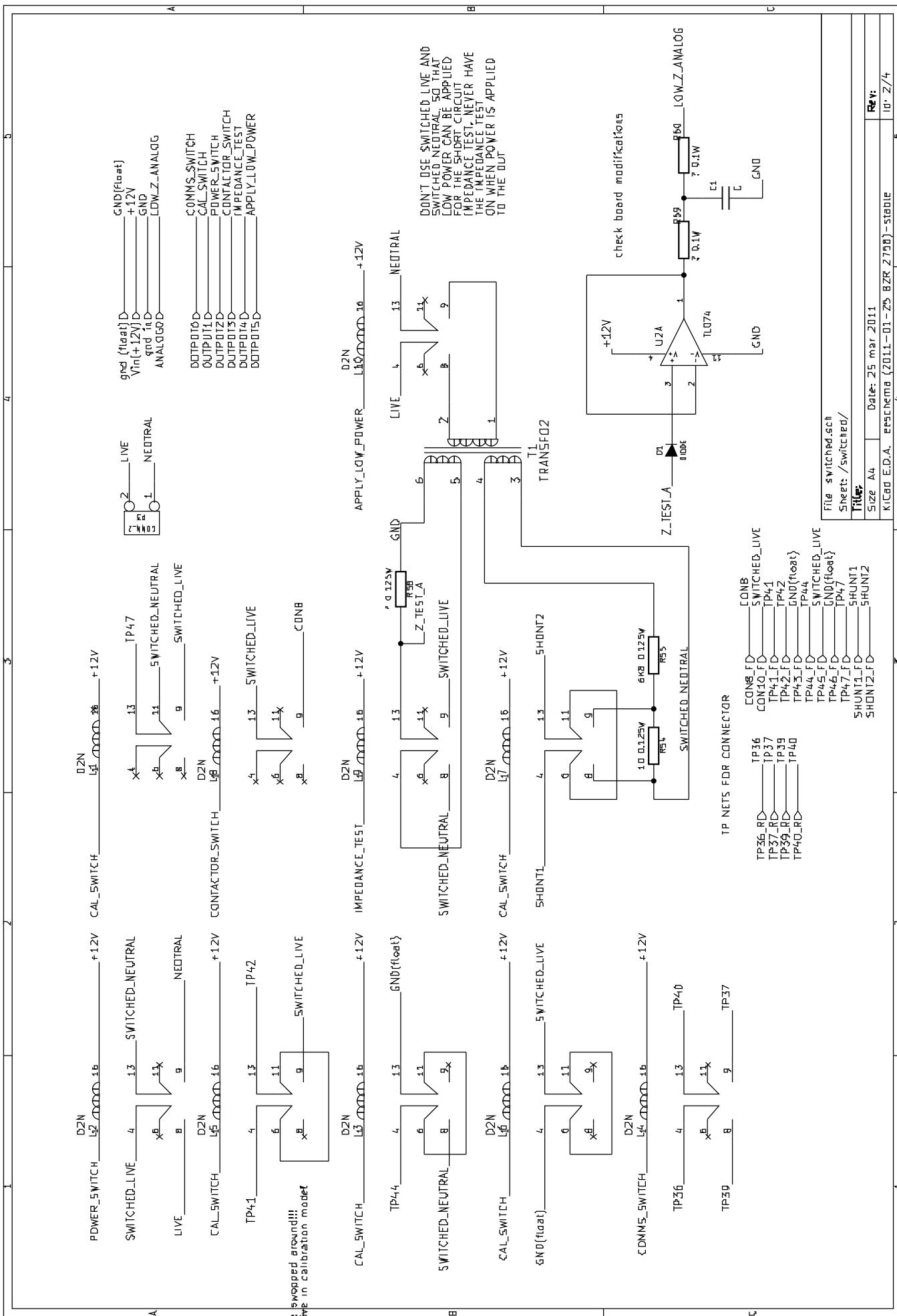


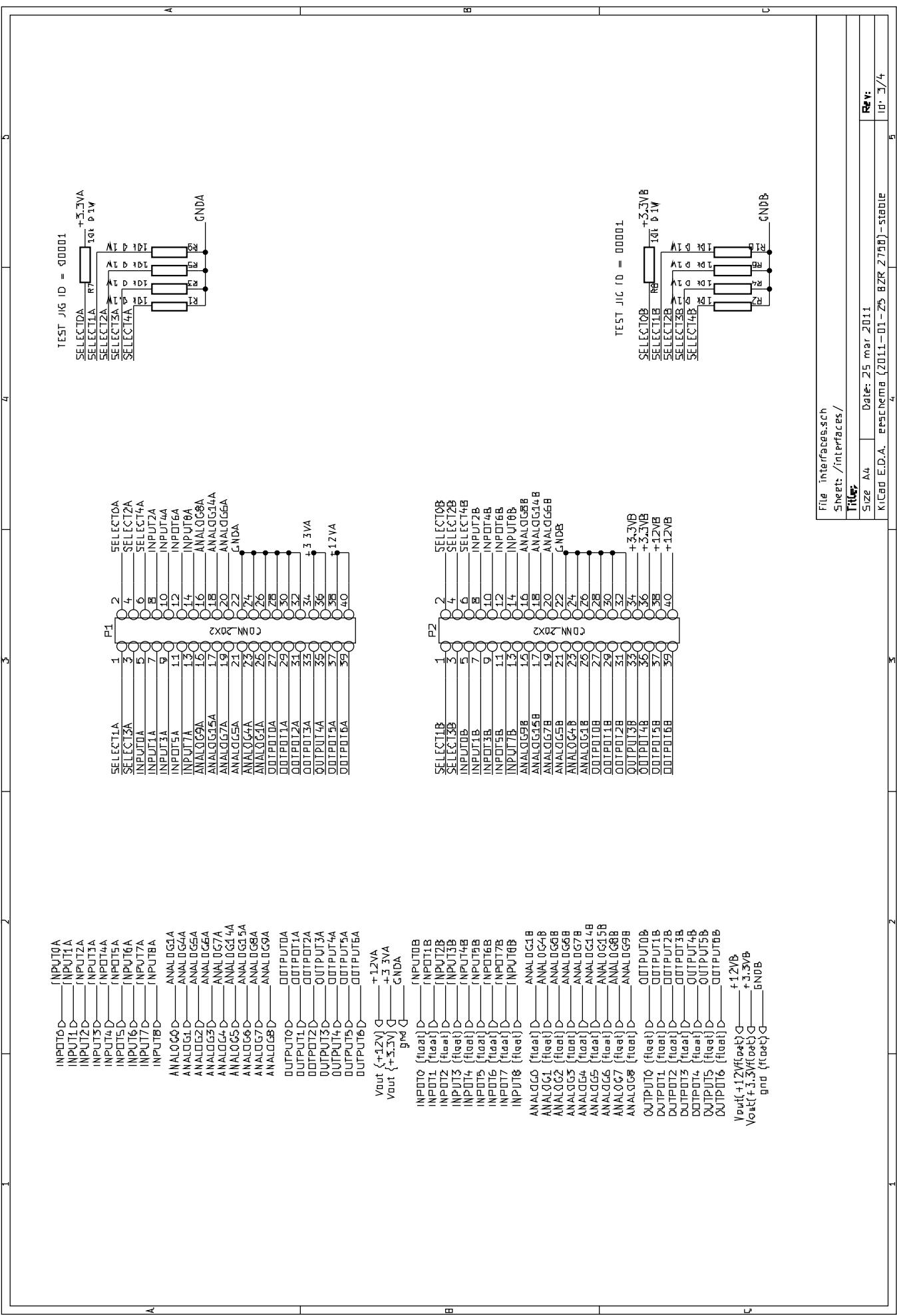
File: powerSupply.sch
Sheet: /powerSupply/
Title:
Size: A4 Date: 1 feb 2011
KICad E.O.A. EFS-Lembo {2011-01-25_BZR 2750}-stabile

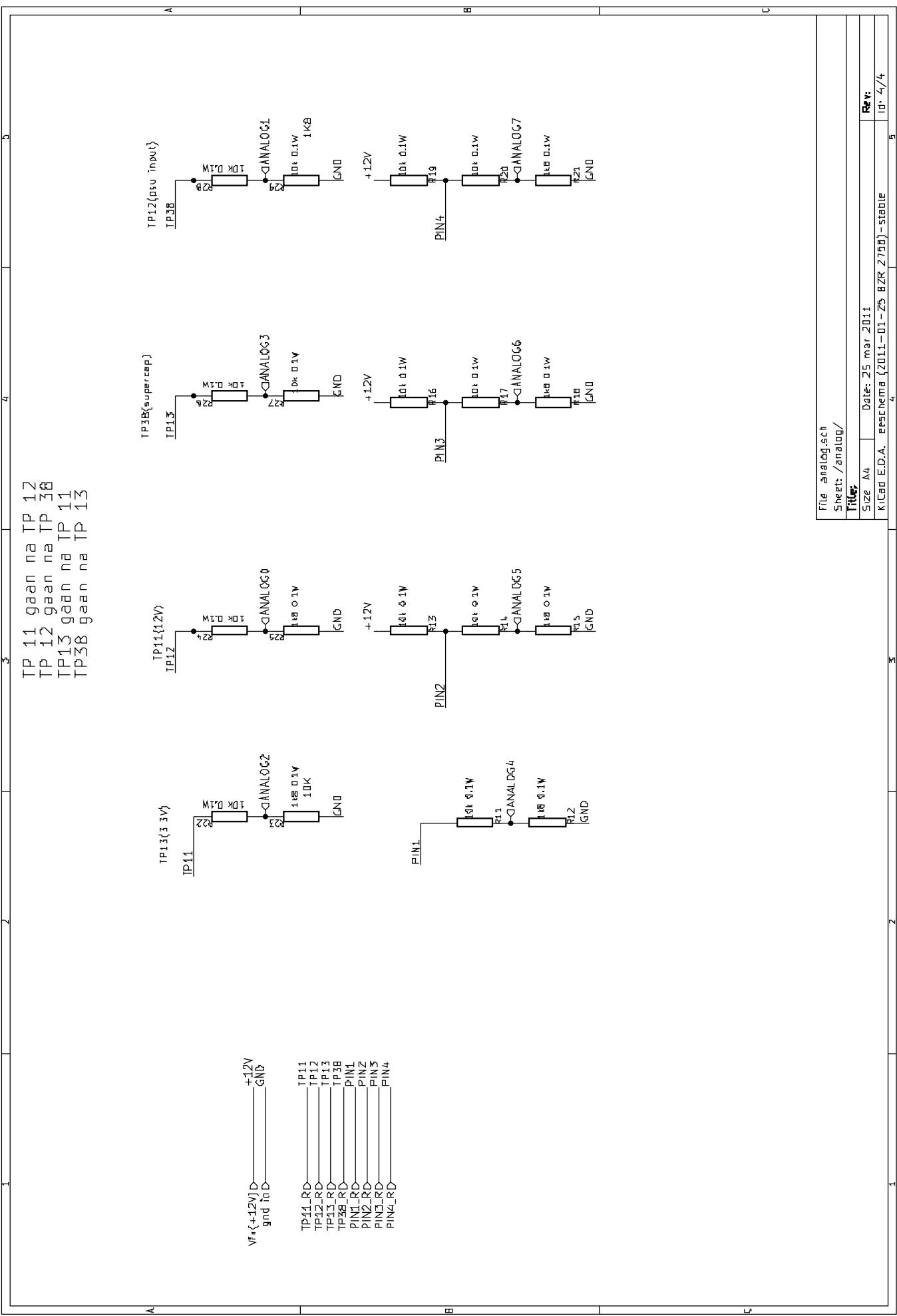
Rev: 1/4











4

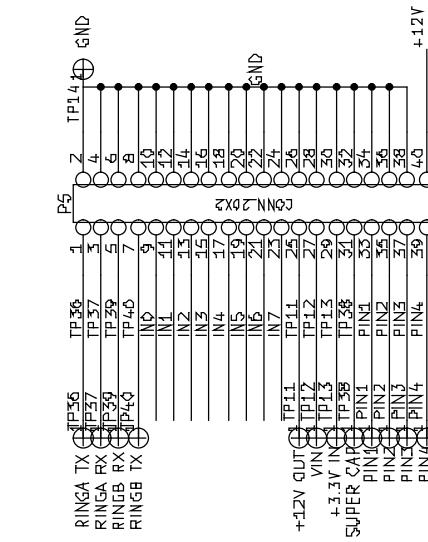
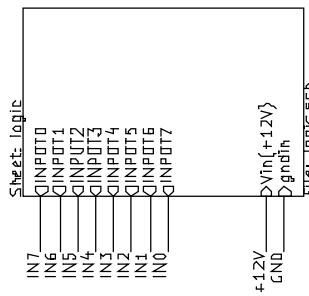
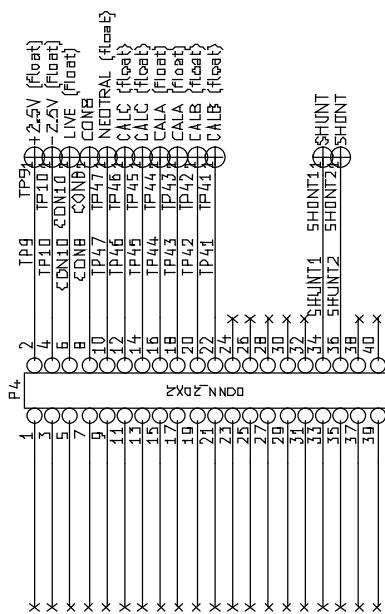
3

2

1

4 mounting holes in the corners
of the board, 5 below the JUT
to support the downward
insertion force

Mounting hole
MH1
MH2
MH3
MH4
MH5
MH6
MH7
MH8
MH9
MH10
MH11
MH12



File: stopper.sch
Sheet: /
Title:

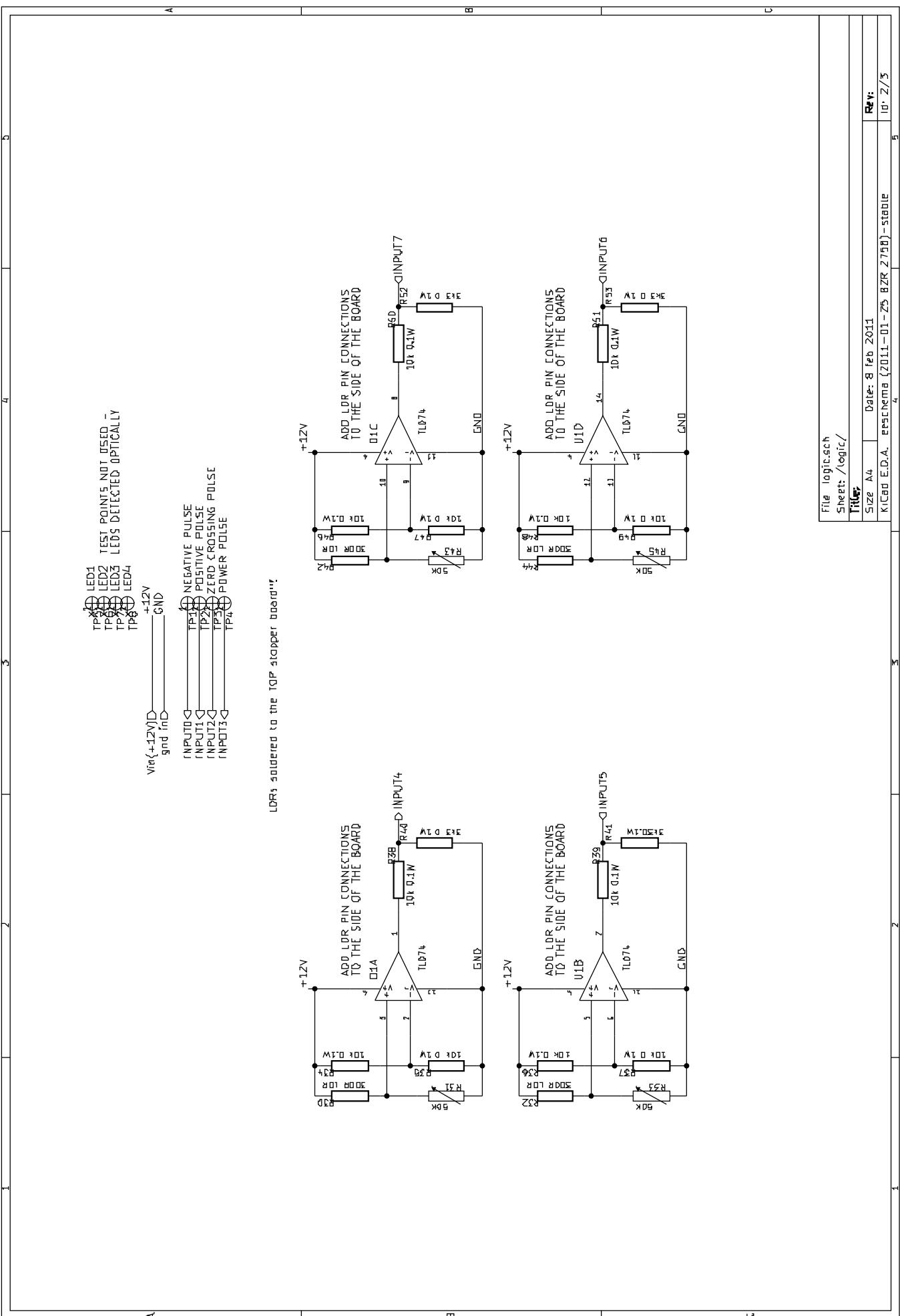
Size: A4 Date: 8 feb 2011
KICad EDA Editor (2011-01-25 07:58 -stable)
Rev: 1/1/3

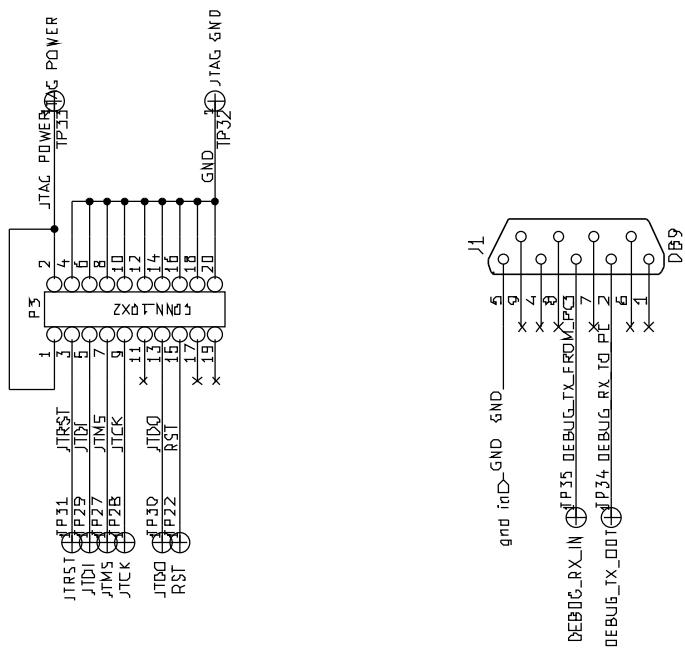
File: eqmms.sch
Sheet: /
Title:

Size: A4 Date: 8 feb 2011
KICad EDA Editor (2011-01-25 07:58 -stable)
Rev: 1/1/3

File: eqmms.sch
Sheet: /
Title:

Size: A4 Date: 8 feb 2011
KICad EDA Editor (2011-01-25 07:58 -stable)
Rev: 1/1/3





File → comm → Sht
Sheet: /comm/s/

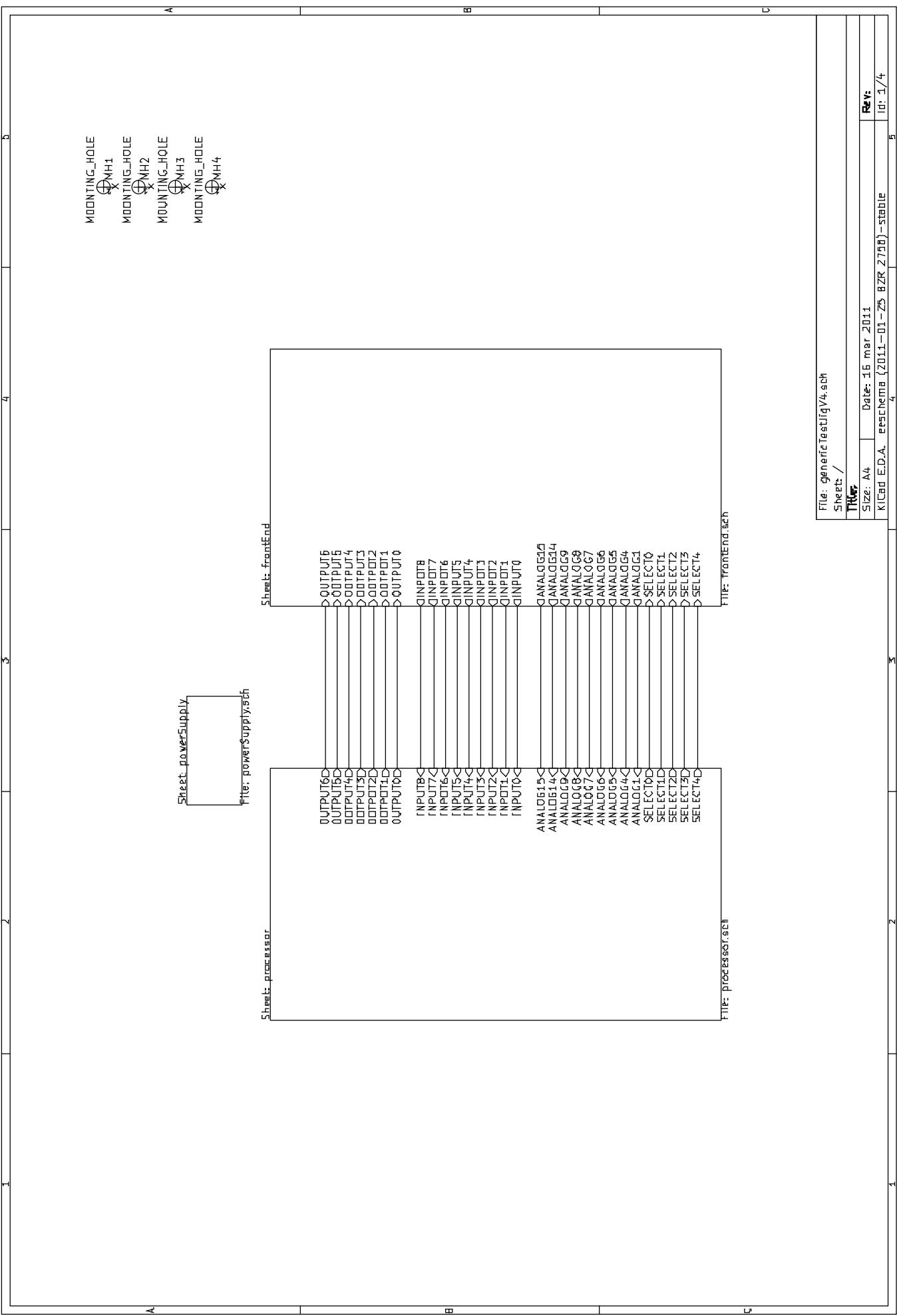
APPENDIX E

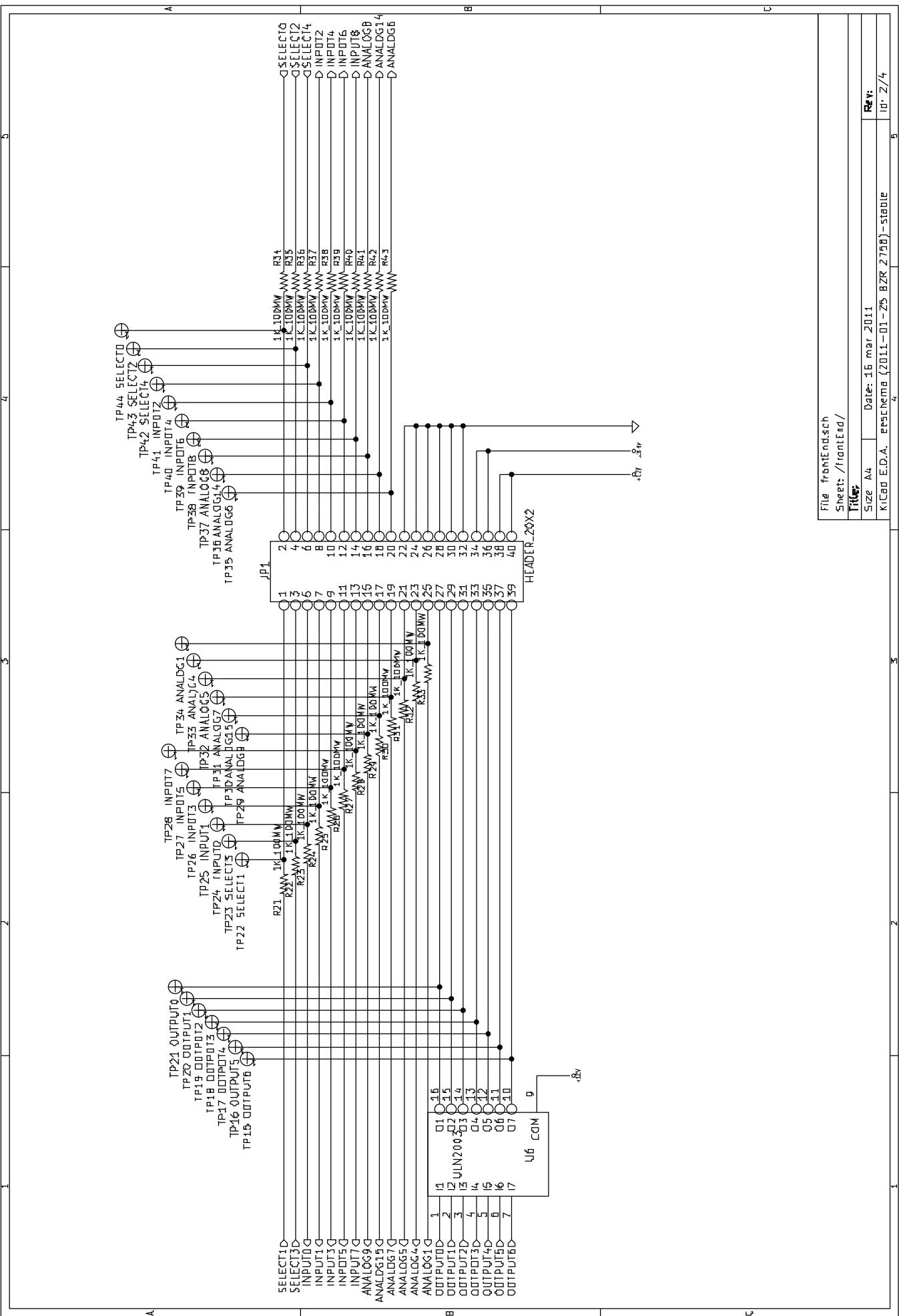
This appendix contains the hardware design files for Iteration 4. The following schematics are contained in this appendix:

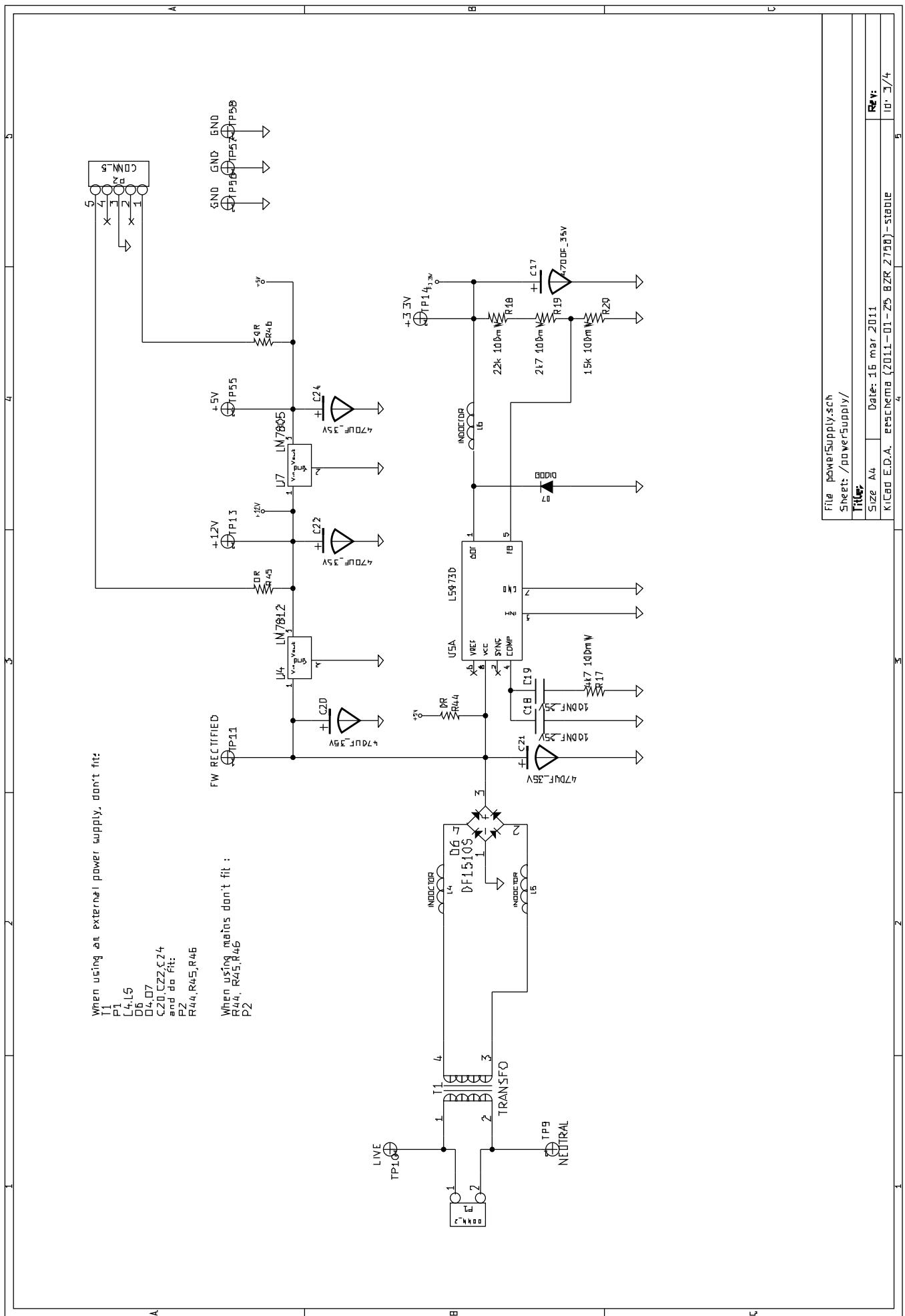
- Generic Test Jig Base Board Rev D Schematic (4 pages)
- Mcu Bed Of Nails Base Rev C Schematic (4 pages)
- Mcu Bed Of Nails Stopper Rev C Schematic (3 pages)
- Display Bed Of Nails Base Rev A Schematic (2 pages)
- Display Bed Of Nails Stopper Rev A Schematic (3 pages)
- Base Template Rev A (2 pages)
- Stopper Template Rev A (2 pages)

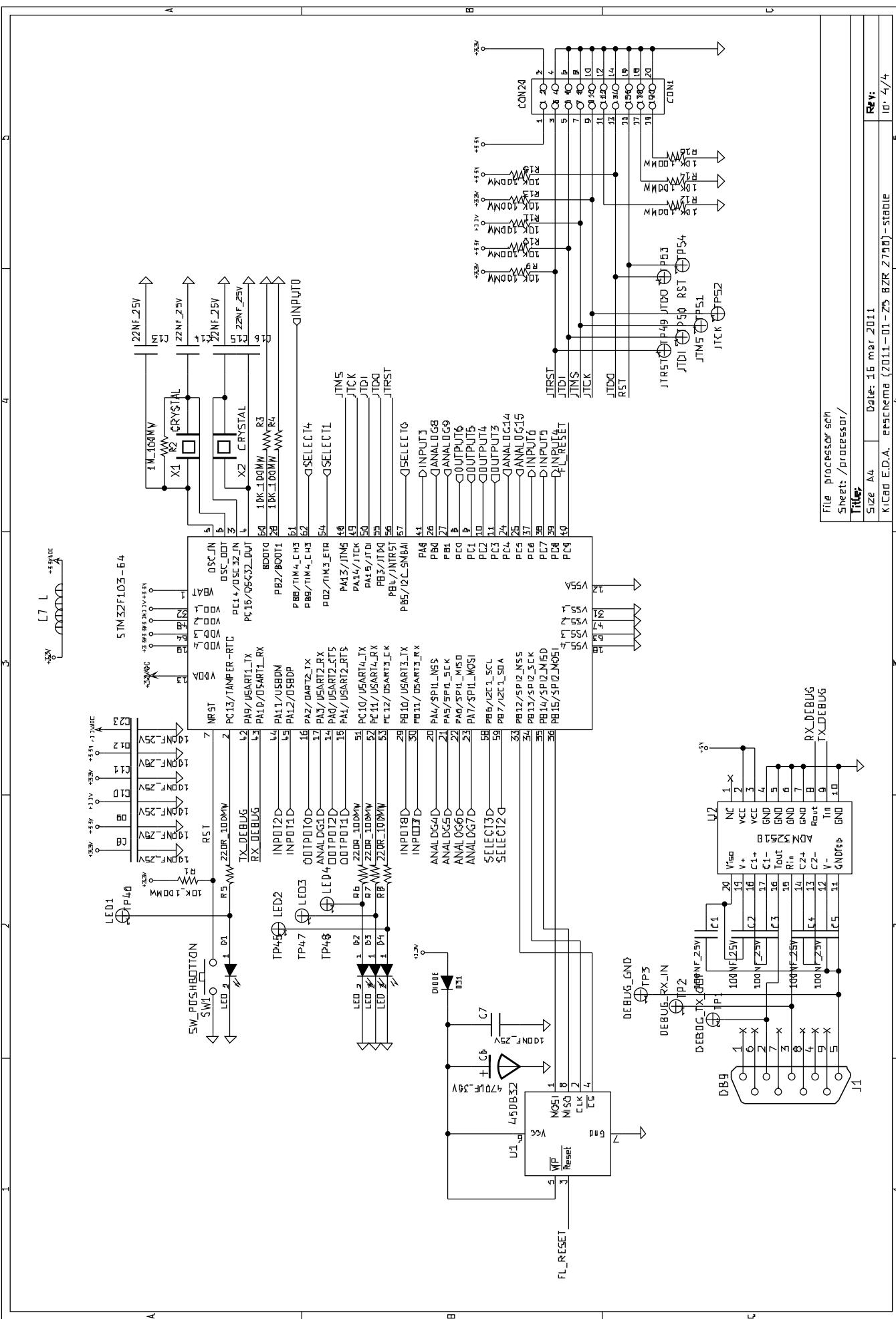
The gerbers for the designs are available in:

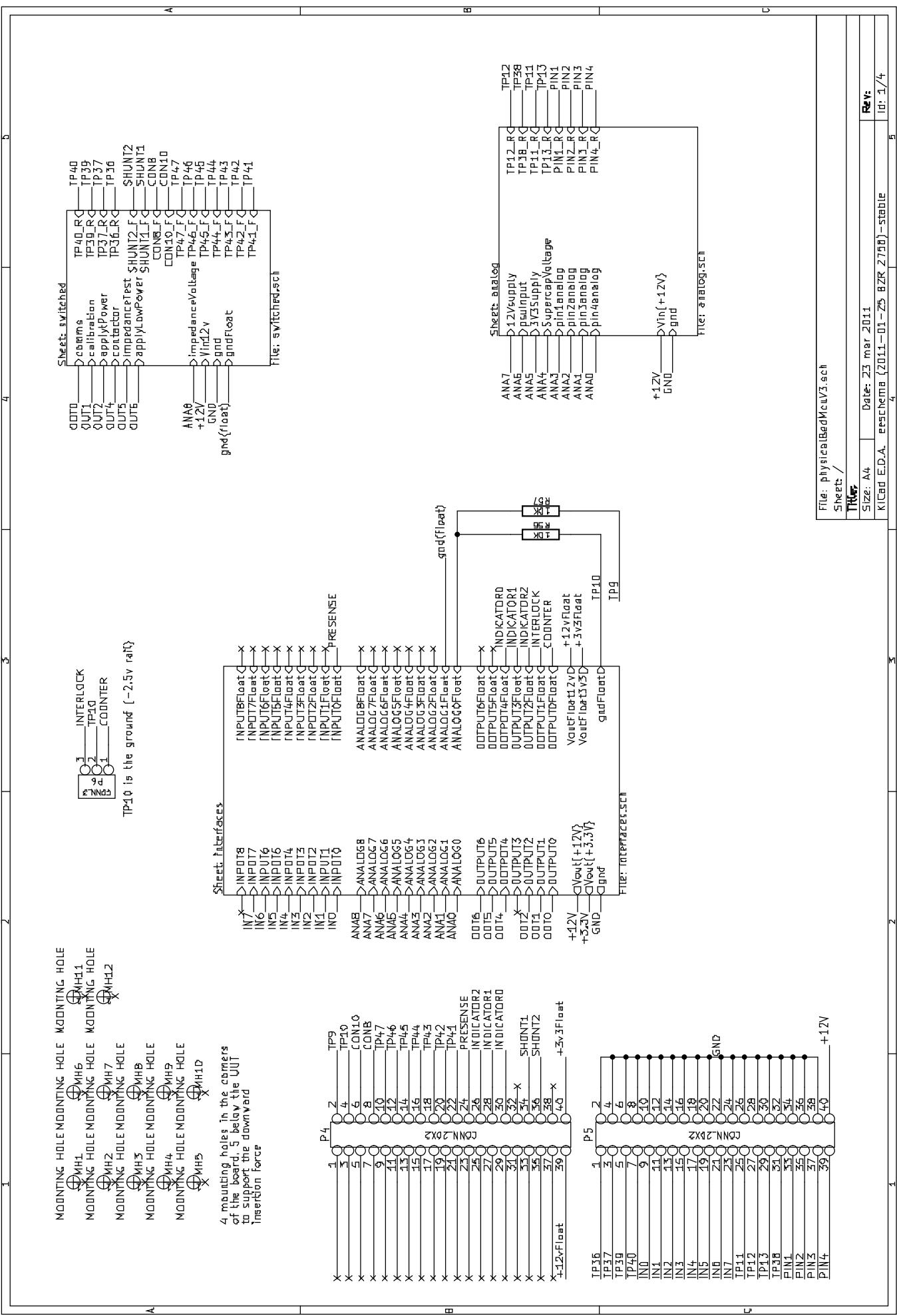
- On project disk /DesignFiles/Hardware/Iteration 4/GenericBaseRevDGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 4/McuBedBaseRevCGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 4/McuBedStopperRevCGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 4/McuBedGuideRevCGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 4/McuBedOutlineRevCGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 4/DisplayBedBaseRevAGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 4/DisplayBedStopperRevAGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 4/DisplayBedGuideRevAGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 4/DisplayBedOutlineRevAGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 4/BaseTemplateRevA.zip
- On project disk /DesignFiles/Hardware/Iteration 4/StopperTemplateRevA.zip

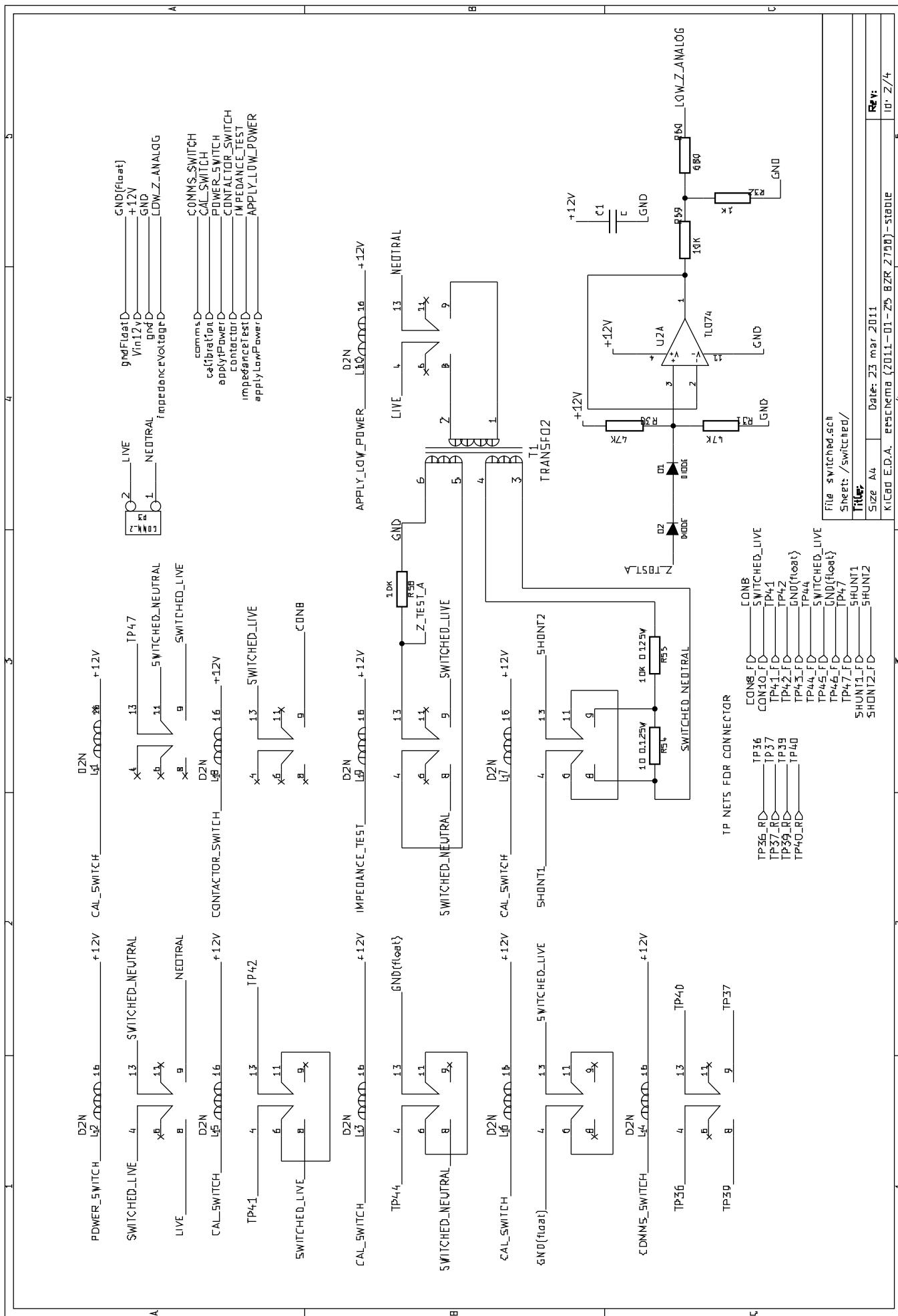


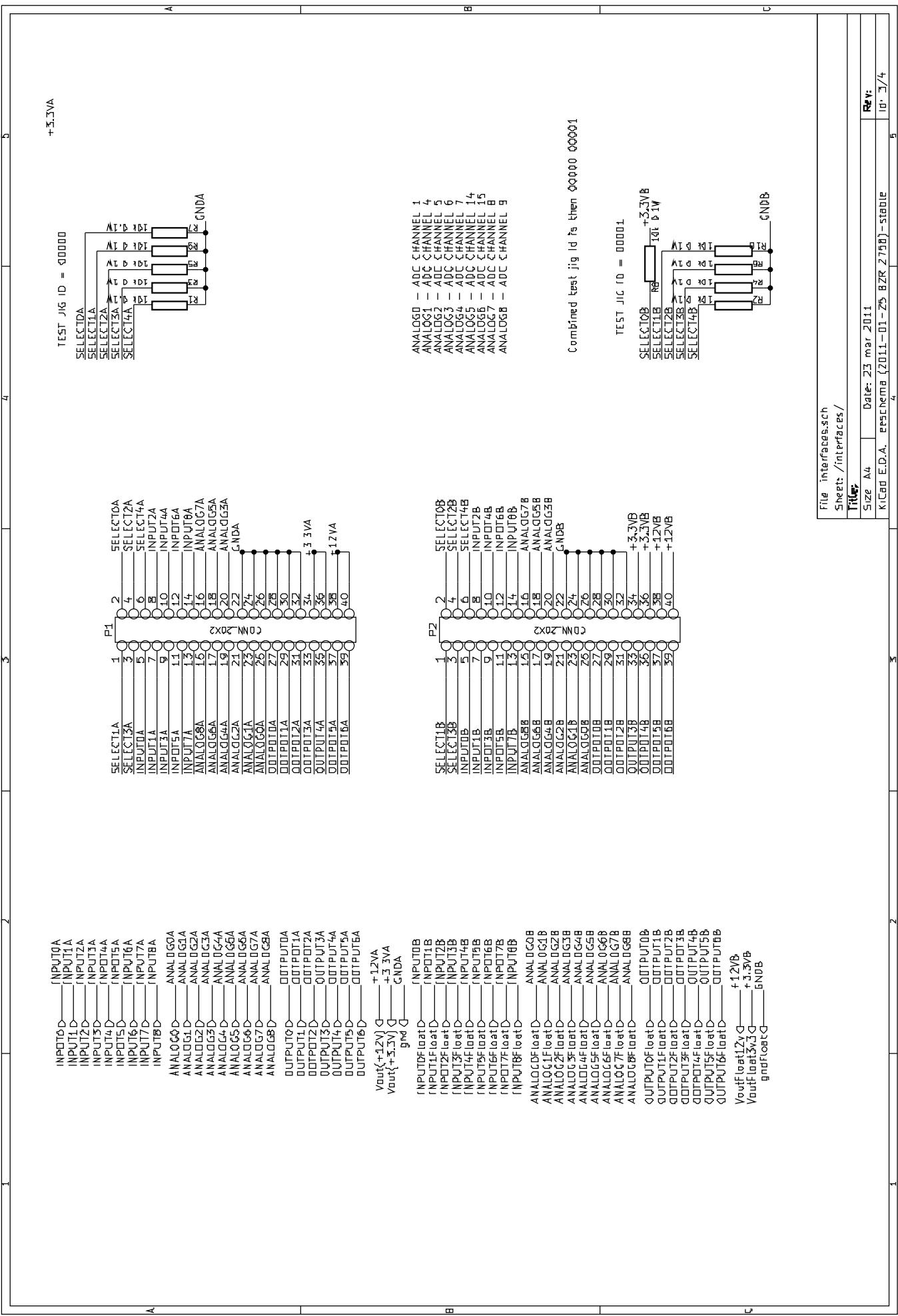


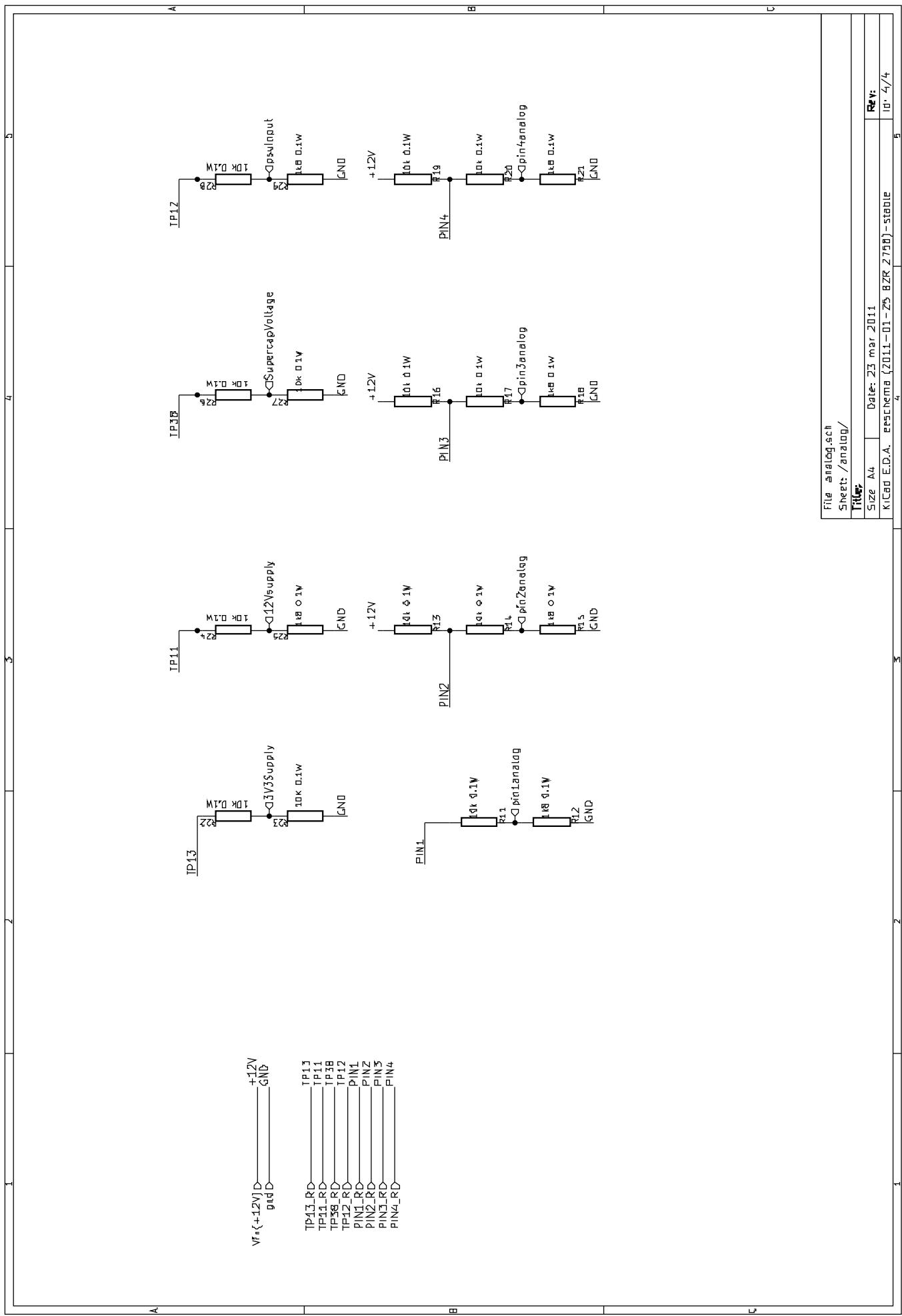


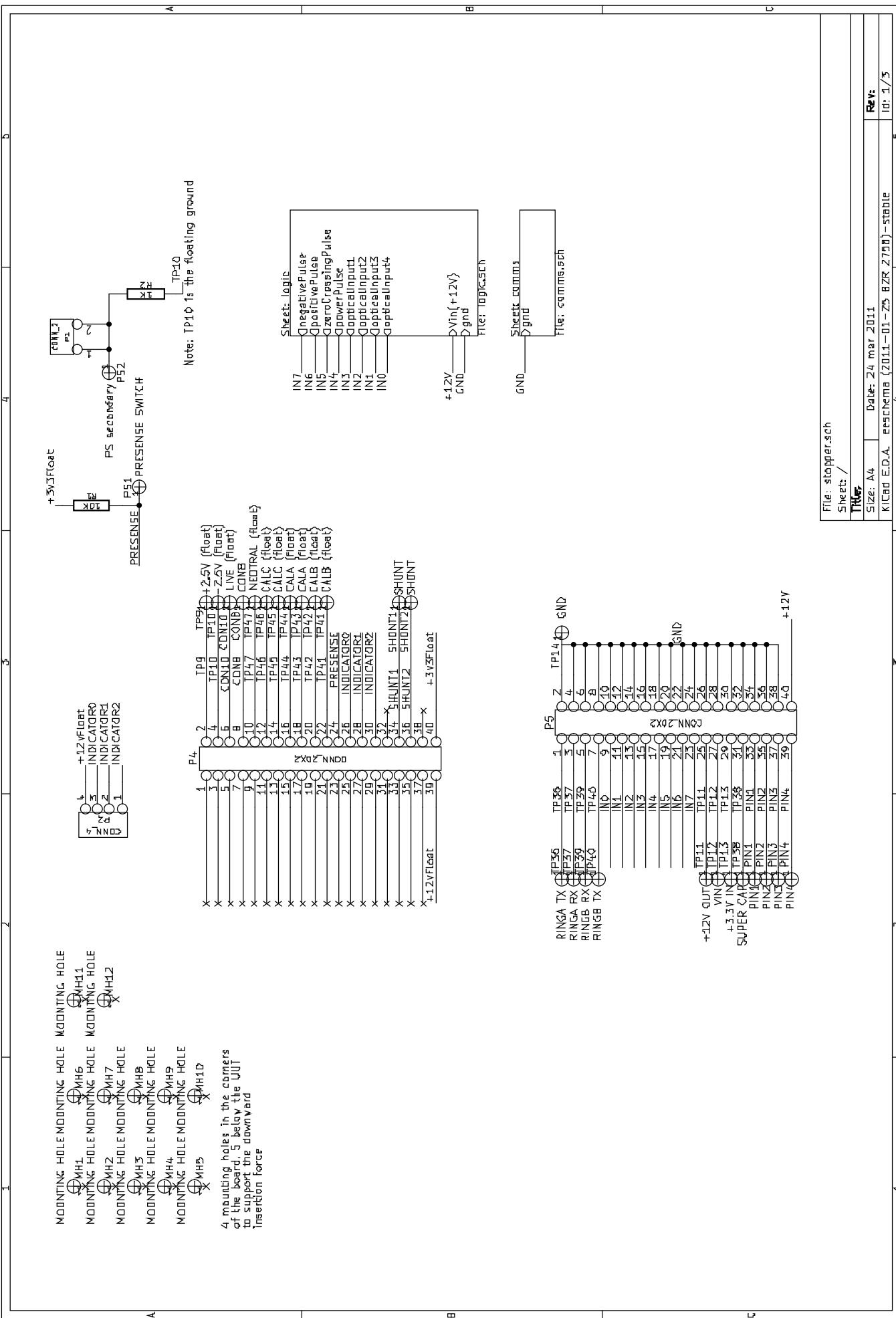


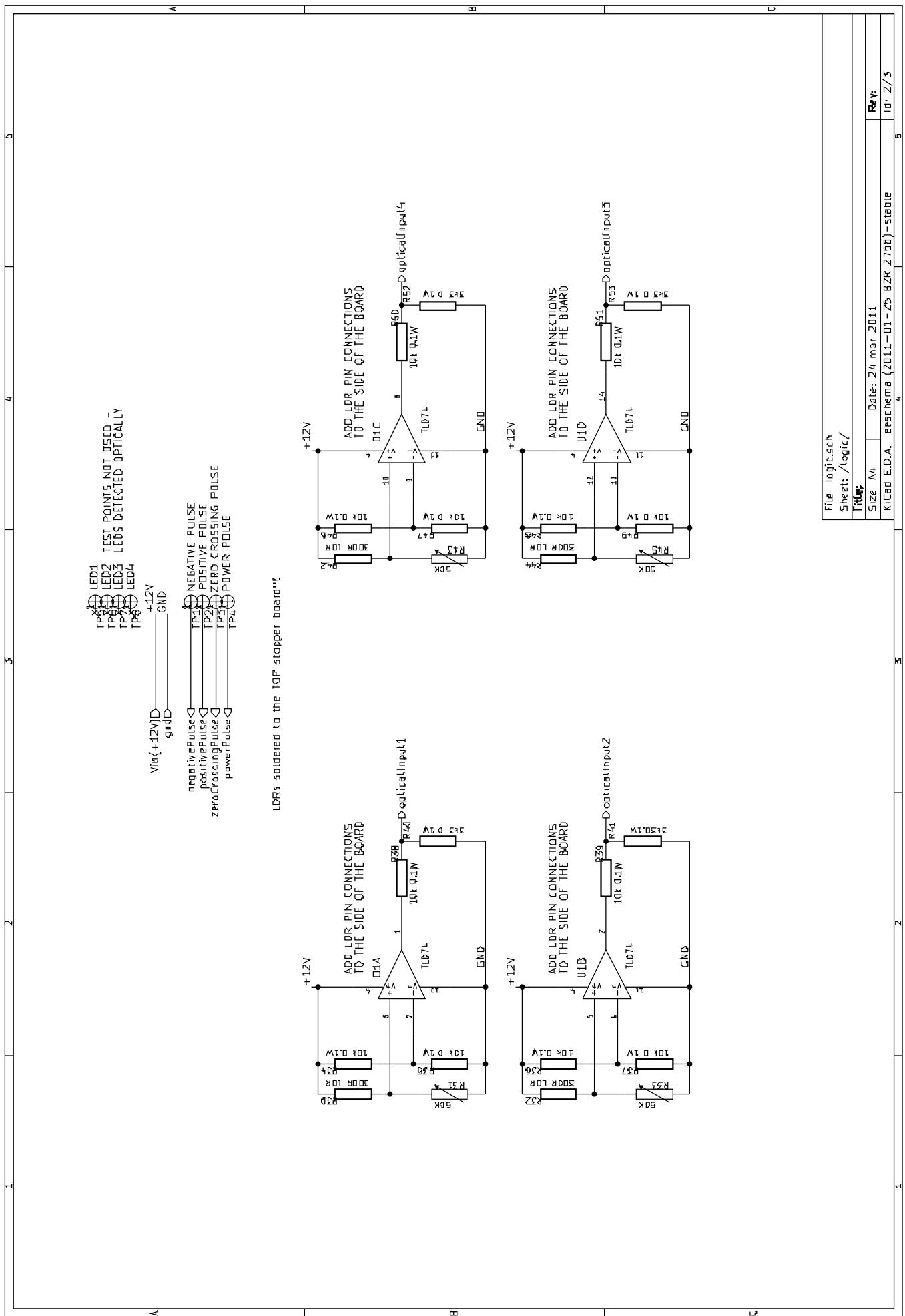


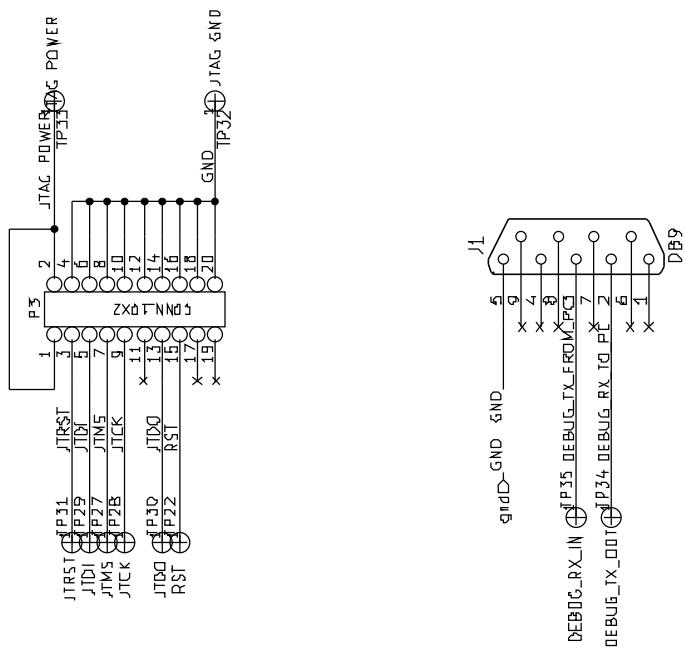




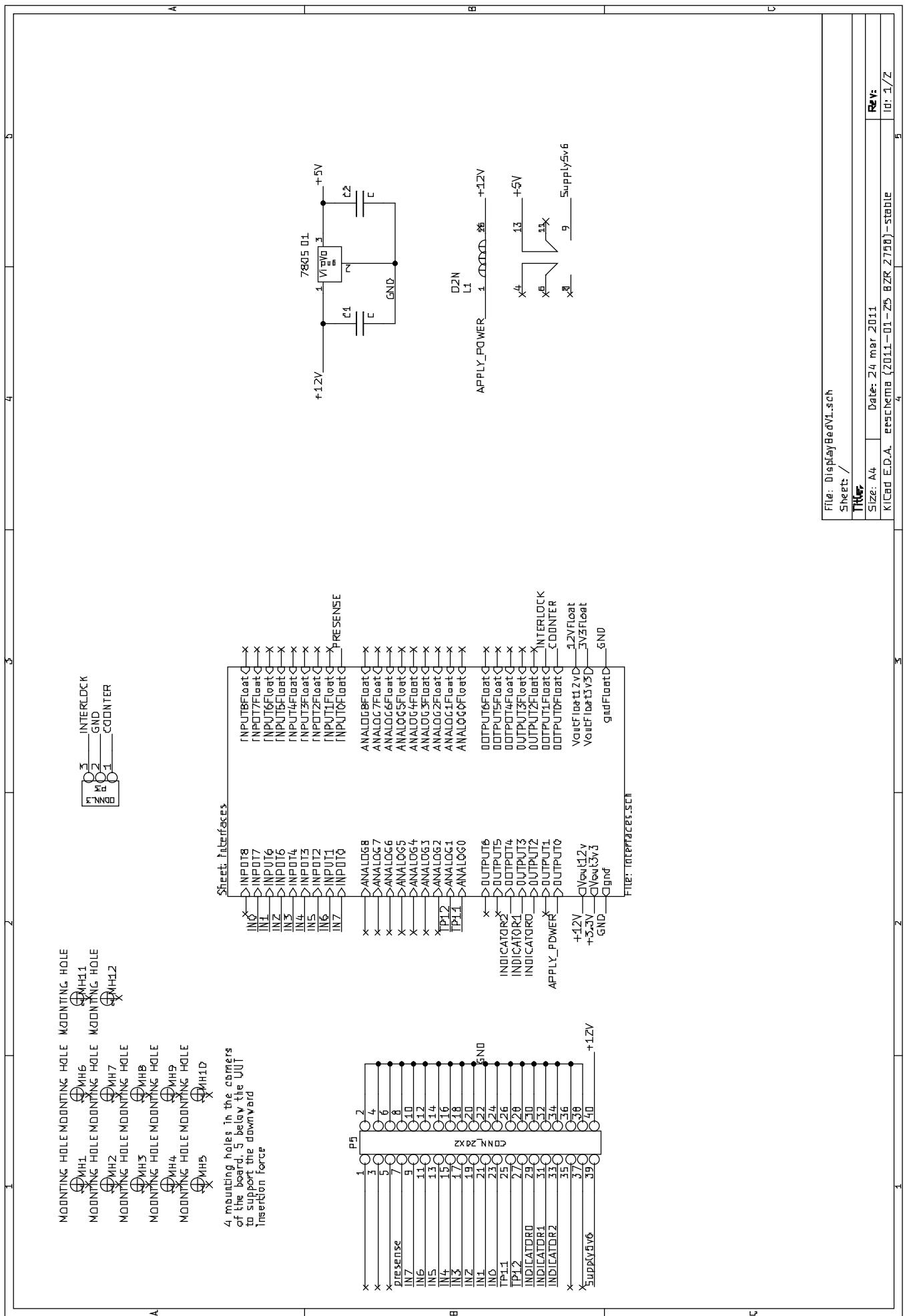


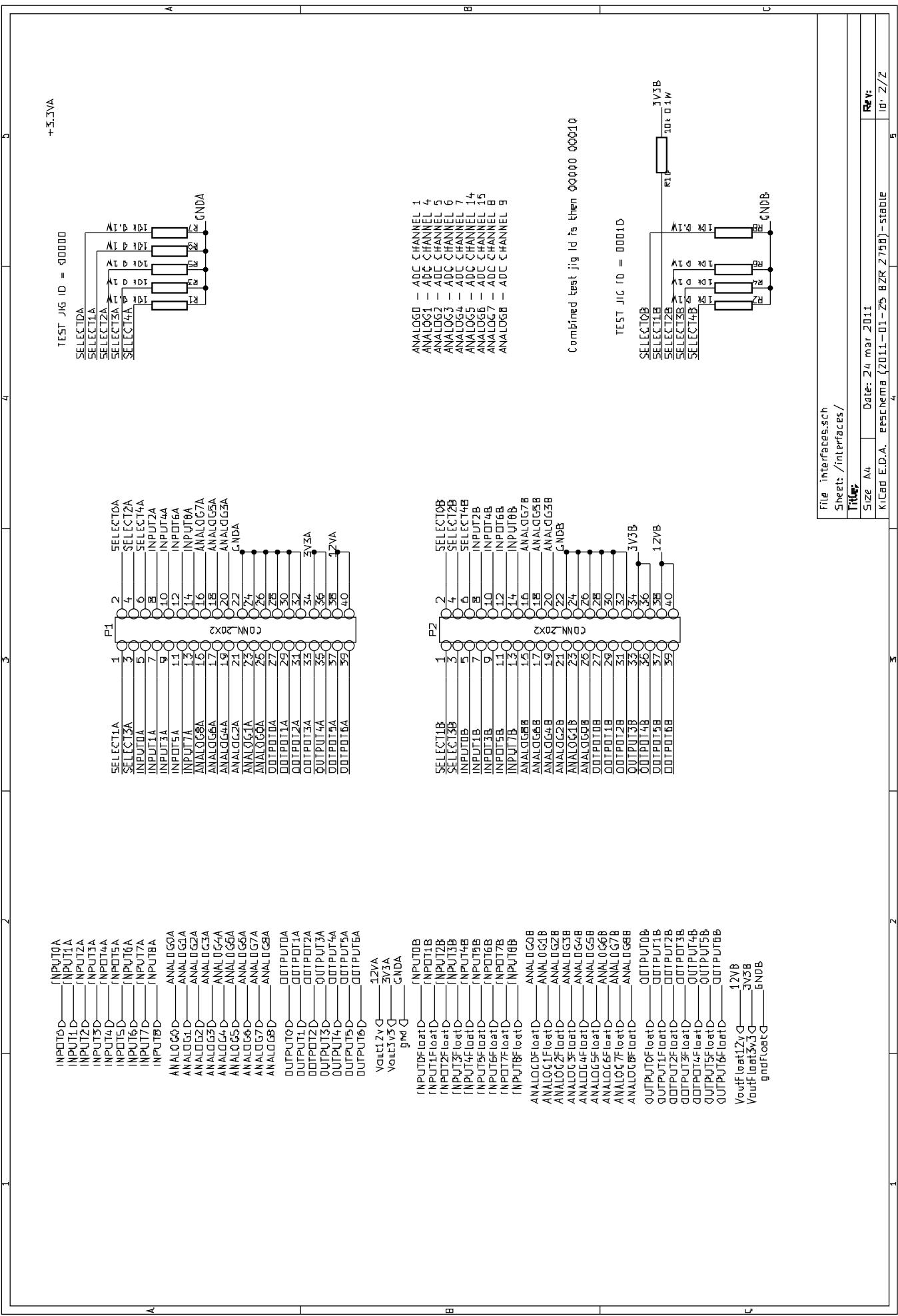


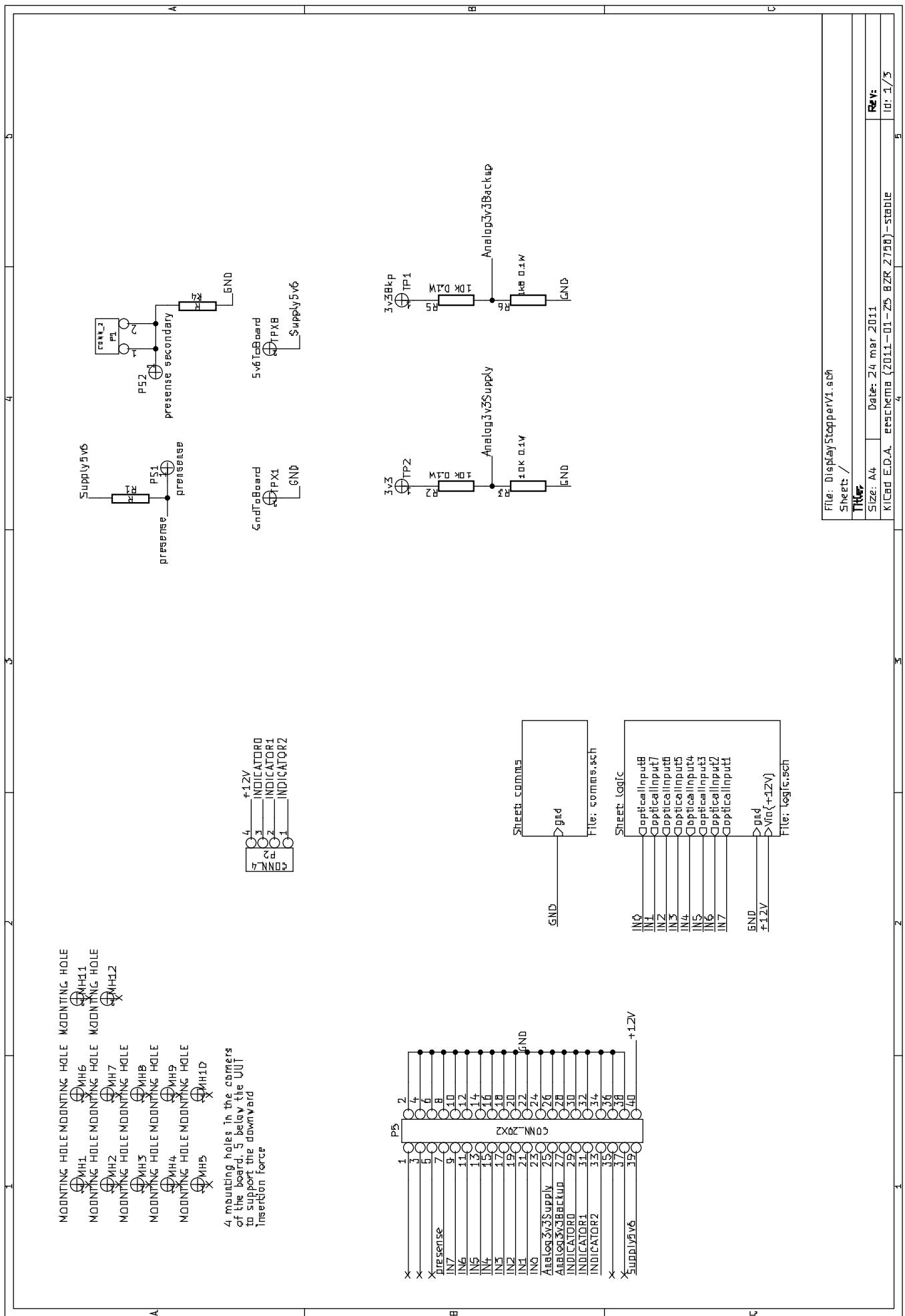


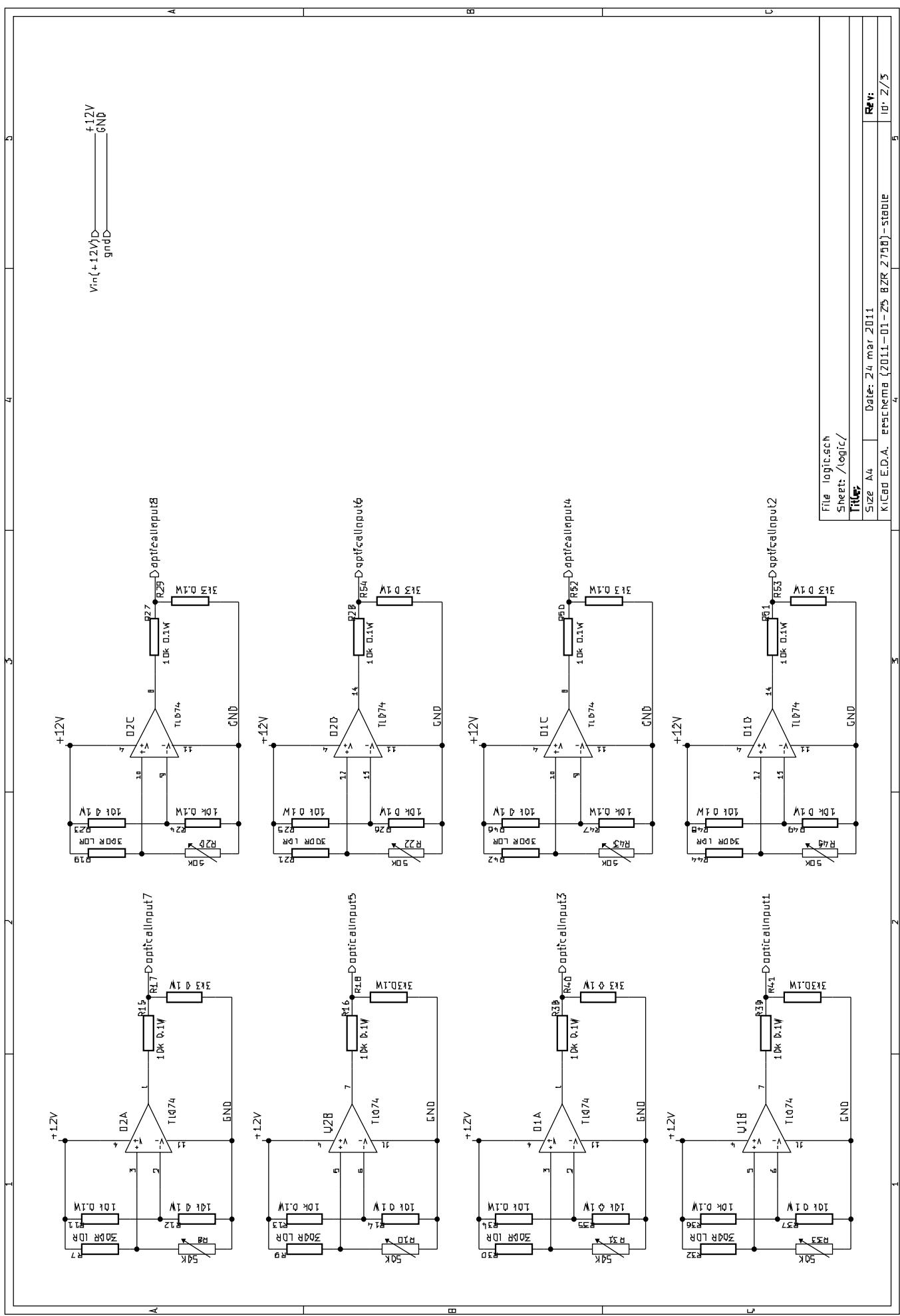


File > comm & Sheet /comms/







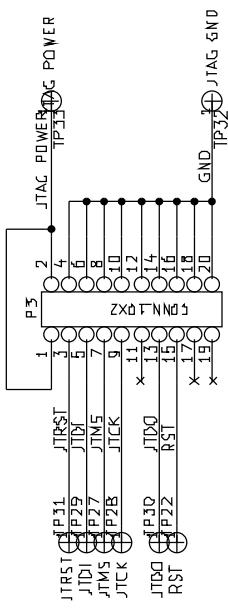


4

5

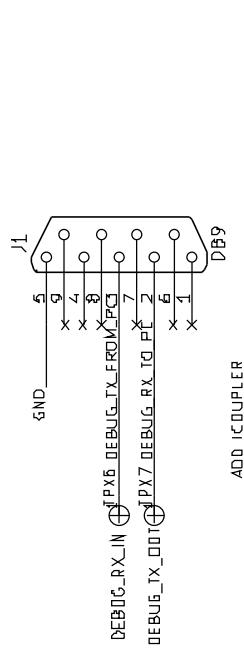
2

1



GND

B



ADD IC COUPLER

File:	comms.sch
Sheet:	/comms/
Title:	
Size:	A4
Date:	24 mar 2011
Rev:	5
KICad EDA_EasyLib [2011-01-25_BZR 2750]-stable	4

A	B	C
1	2	3
3	4	5
5	6	7
7	8	9

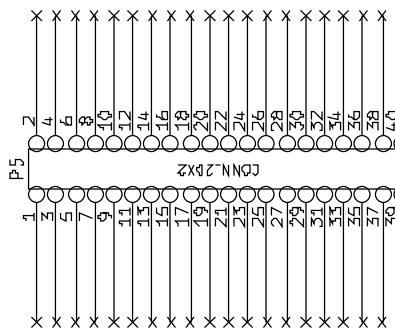
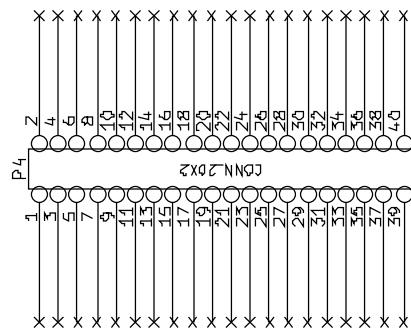
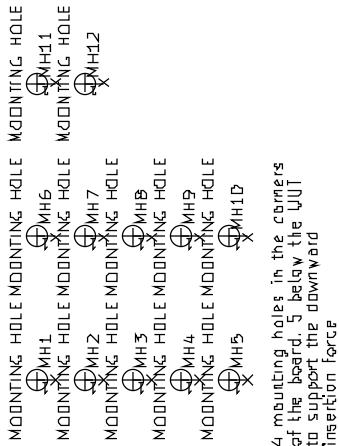
3

4

5

2

1



Sheet: Interfaces

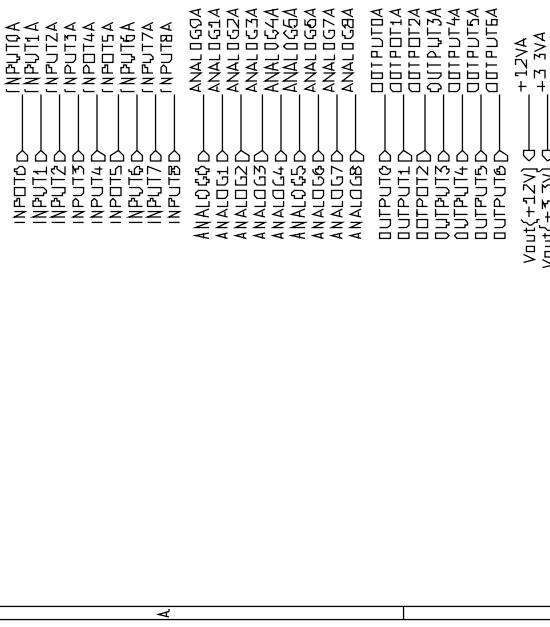
X	INPUT8	X
X	INPUT7	X
X	INPUT6	X
X	INPUT5	X
X	INPUT4	X
X	INPUT3	X
X	INPUT2	X
X	INPUT1	X
X	INPUT0	X
X	DANALOG8	X
X	DANALOG7	X
X	DANALOG6	X
X	DANALOG5	X
X	DANALOG4	X
X	DANALOG3	X
X	DANALOG2	X
X	DANALOG1	X
X	DANALOG0	X
X	DOUPUT16	X
X	DOUPUT15	X
X	DOUPUT14	X
X	DOUPUT13	X
X	DOUPUT12	X
X	DOUPUT11	X
X	DOUPUT10	X
X	Vout12vD	X
X	Vout12v3D	X
X	GndFloatD	X

File: Interface.sch

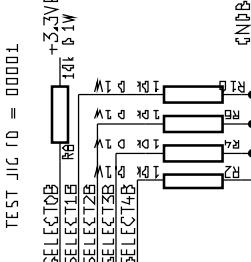
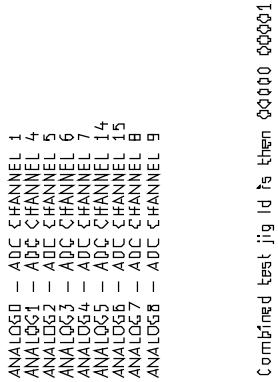
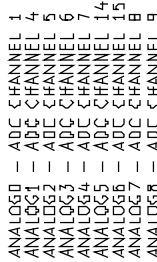
File: BaseTemplate.sch
Sheet: /

Title:

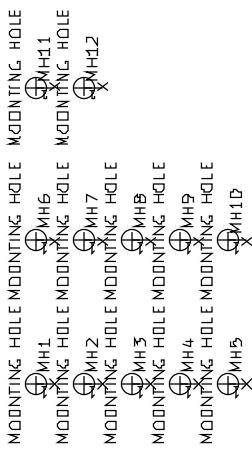
Size: A4	Date: 30 mar 2011	Rev: 1
KICAD E.O.A. schematic (2011-07-D3)-testing	4	10: 1/2



NOTE: ON THIS SCHEMATIC ONLY MODIFY THE ADDRESS!

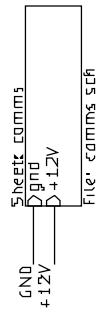
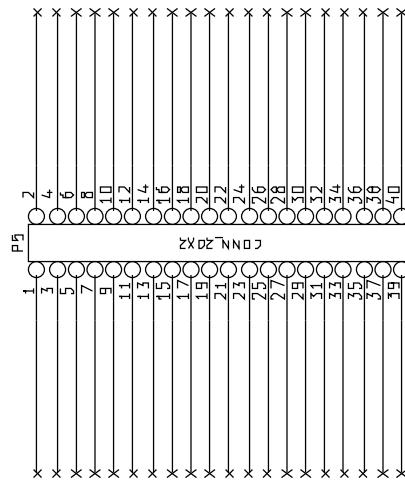
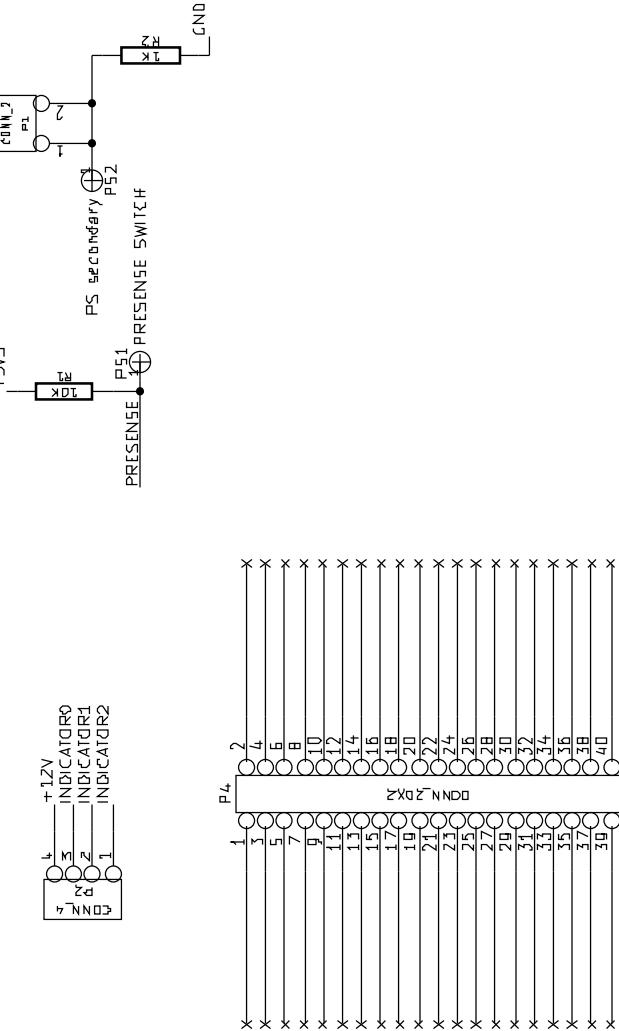


File Interfaces.sch
Sheet /Interfaces/
Title: KICD E.O.A. Testbench (2011-07-D3)-testing
Size: A4 Date: 30 mar 2011
Rev: 10-2/z



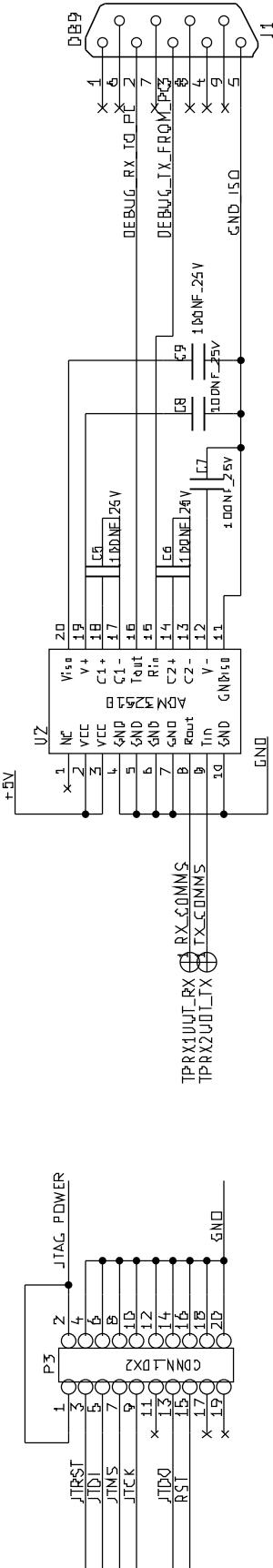
Note 4 mounting holes in the corners of the board. At least 5 below the UUT to support the downward insertion force.

NOTE at least one +12V pin needs to be brought up through the ribbon cable for the ICQD1er PS0 and indicator LEDs. +2v pin needs to be brought up through the ribbon cable for the presence switch CND needs to be brought up. Output lines need to be brought up for the 3 indicator LEDs.

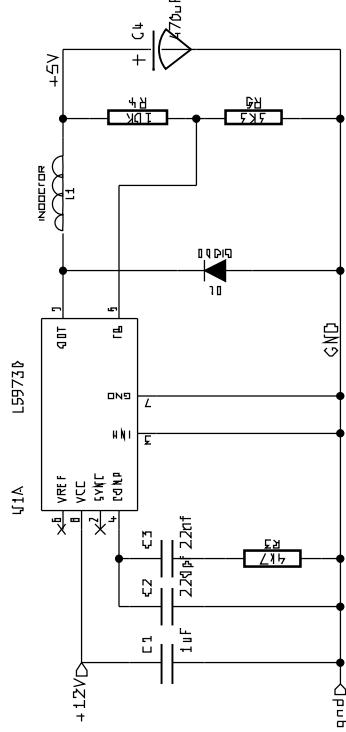


File StopperTemplate.sch
Sheet /
Title:

Size A4	Date: 5 Jul 2011	Rev: 5
KICad E.O.A.	Printed (2011-07-D3)-testing	10-1/2
4	4	5



Notes:
The +5V power supply for the JTAG converter can either come from the U1 circuit or be brought up from the base, in which case the U1 circuit can be removed.
U1_RX goes to the DEBUG_RX test point on the UUT
U1_TX goes to the DEBUG_TX test point on the UUT
JTAG pins each go to the corresponding JTAG test points



File: common.sch
Sheet: /common/
Title: **L5973D**

Size	A4	Date	5 Jul 2011	Rev:
KICED E.O.A.	eschematic (2011-07-D3)-testing	4	10-2/2	5

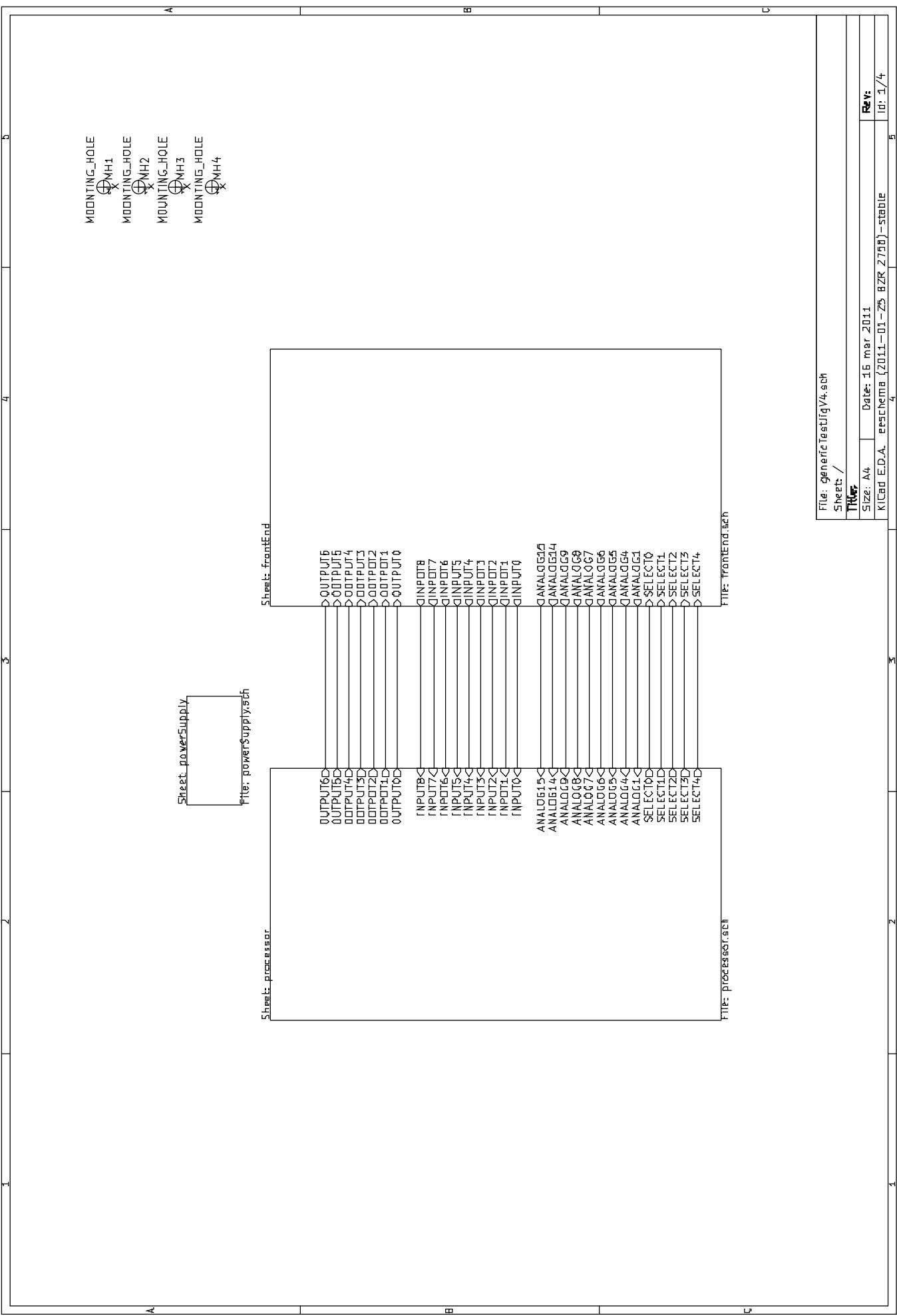
APPENDIX F.1

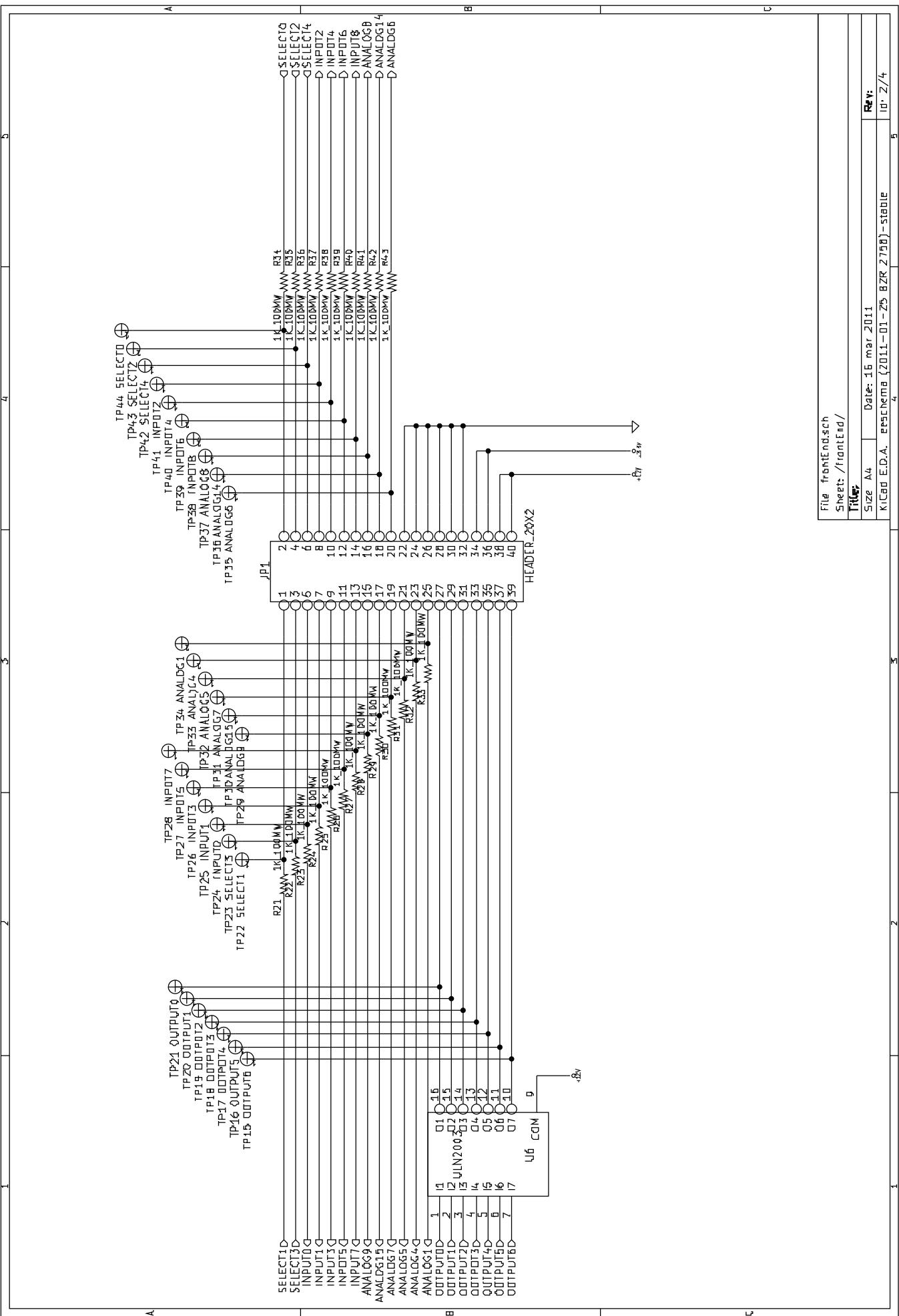
This appendix contains the hardware design files for Iteration 5. The following schematics are contained in this appendix:

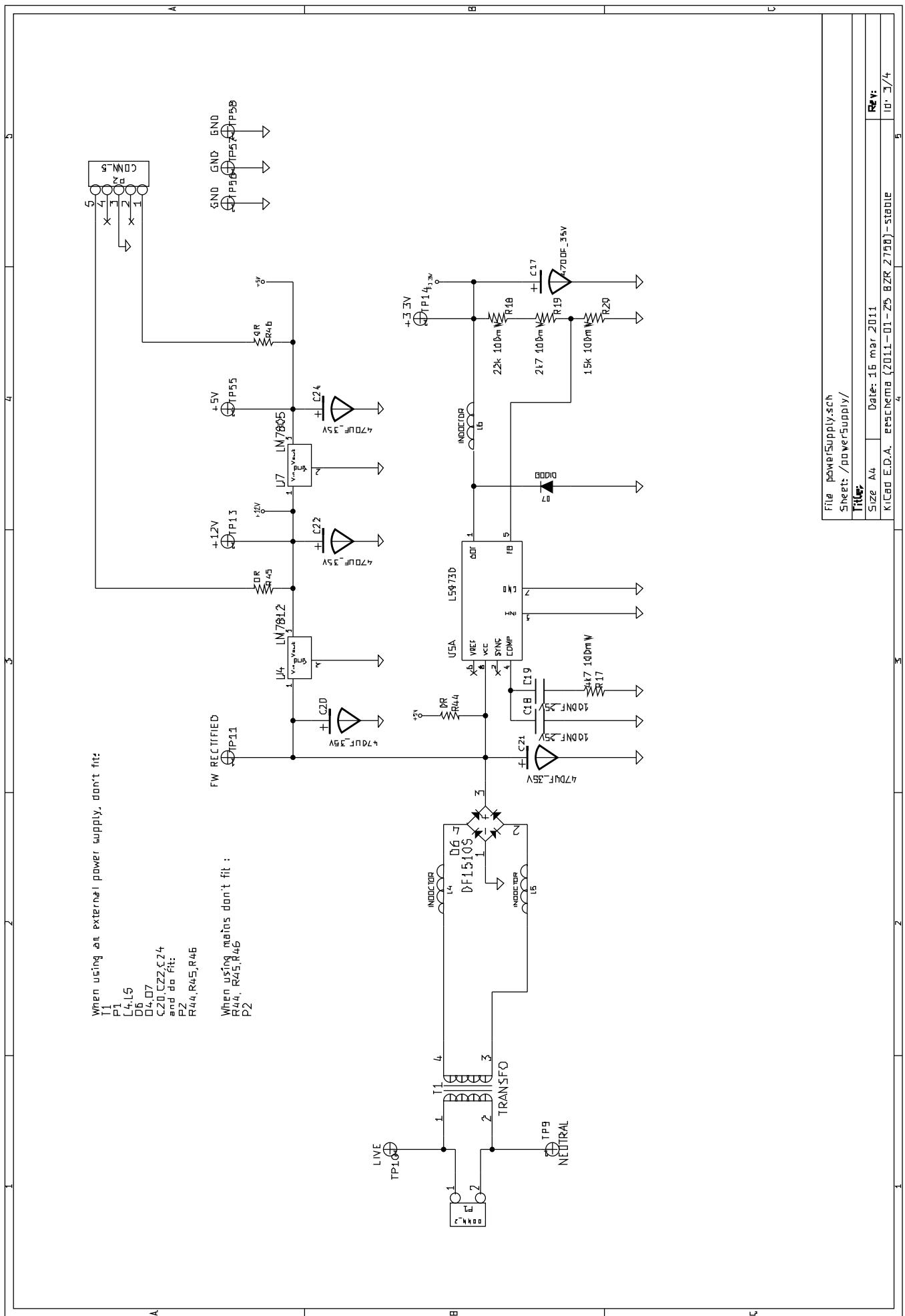
- Generic Test Jig Base Board Rev D Schematic (4 pages)
- Mcu (RevD) Bed Of Nails Base Rev C Schematic (4 pages)
- Mcu (RevD) Bed Of Nails Stopper Rev B Schematic (3 pages)
- Base Template Rev B (2 pages)
- Stopper Template Rev B (2 pages)

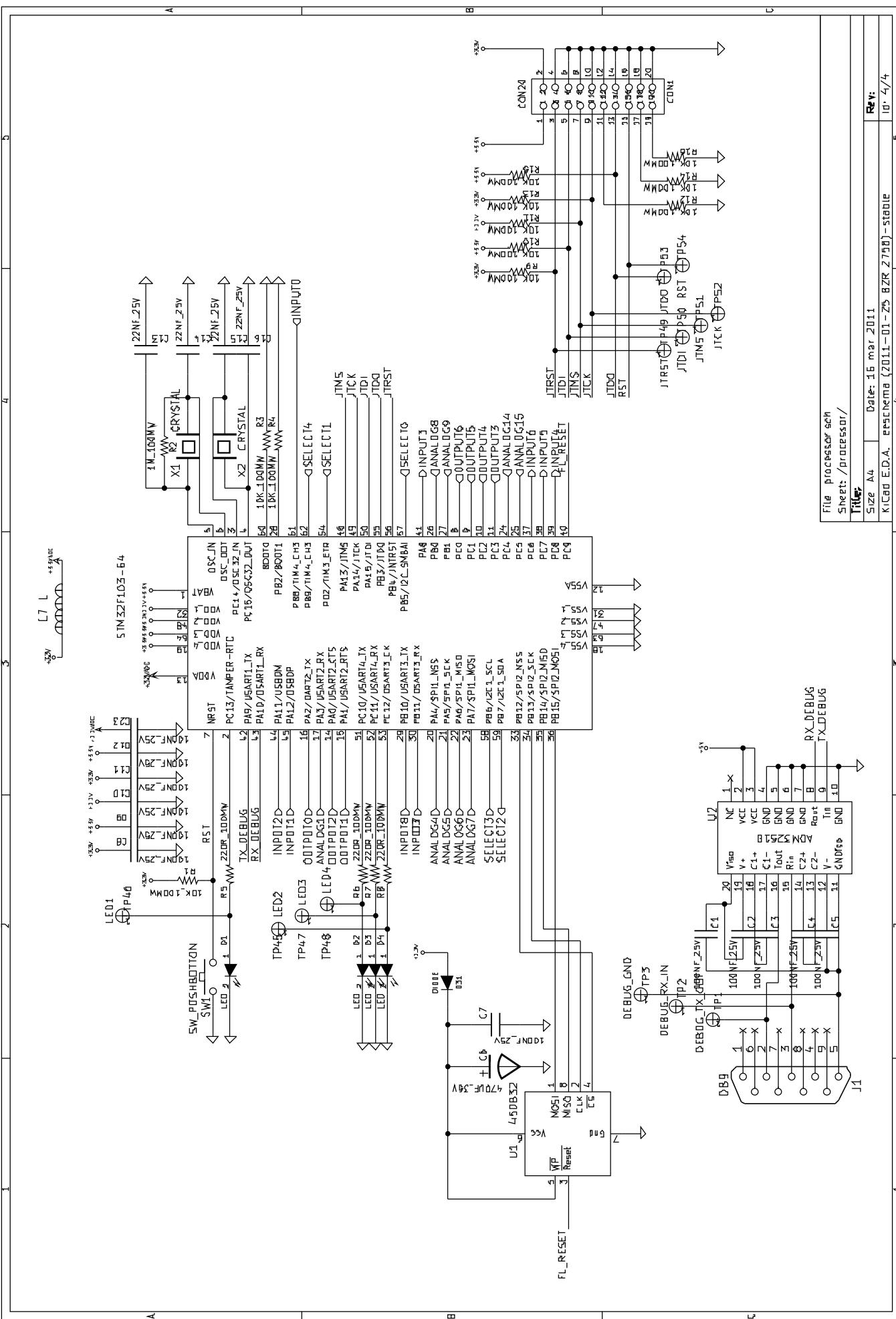
The gerbers for the designs are available in:

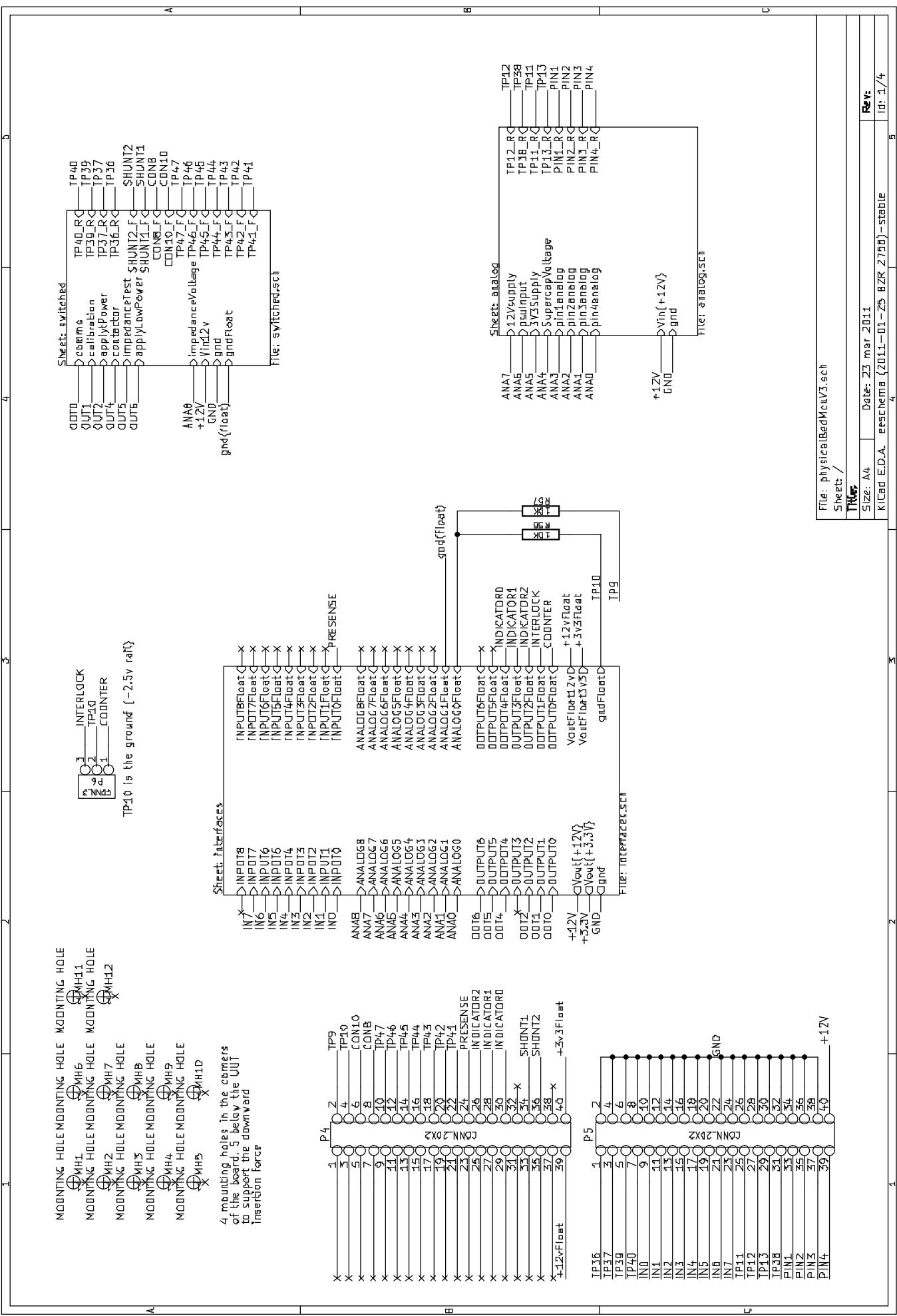
- On project disk /DesignFiles/Hardware/Iteration 5/GenericBaseRevDGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/Mcu(D)BedBaseRevCGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/Mcu(D)BedStopperRevBGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/Mcu(D)BedGuideRevAGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/Mcu(D)BedOutlineRevCGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/Mcu(D)BedBackRevAGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/BackTemplateRevBGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/BaseTemplateRevBGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/StopperTemplateRevBGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/GuideTemplateRevBGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/OutlineTemplateRevBGerbers.zip
- On project disk /DesignFiles/Hardware/Iteration 5/PositionerRevAGerbers.zip

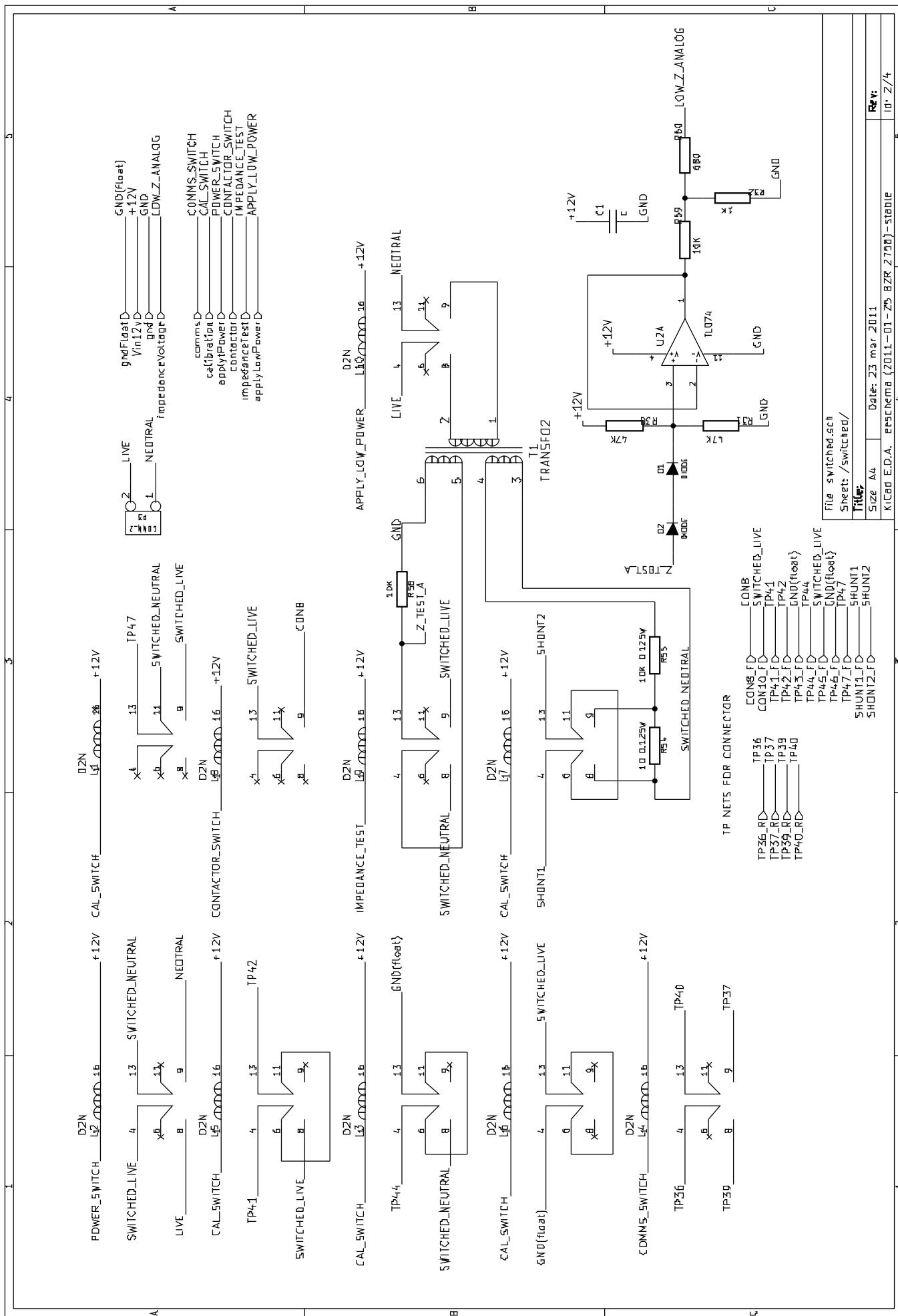


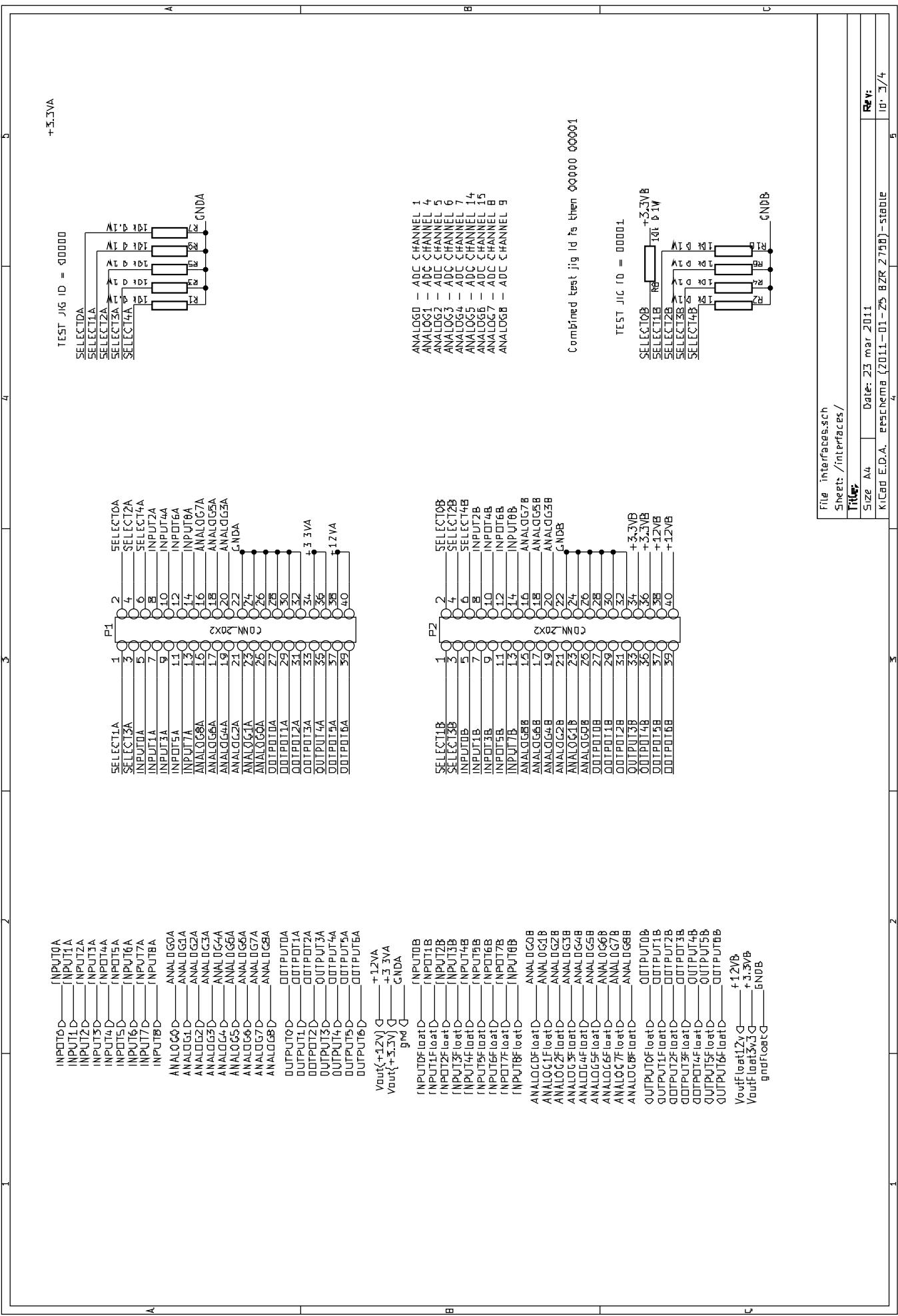


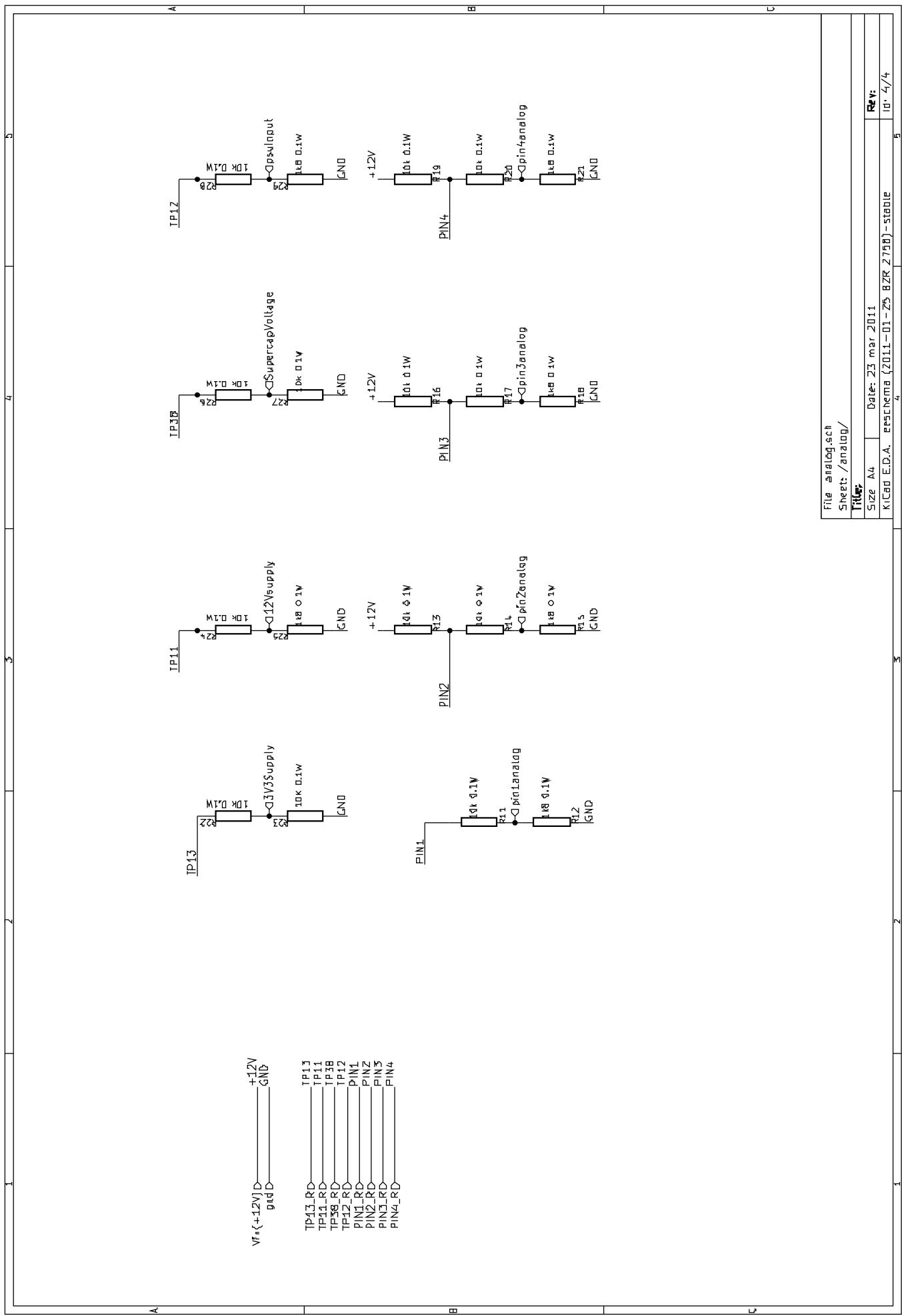




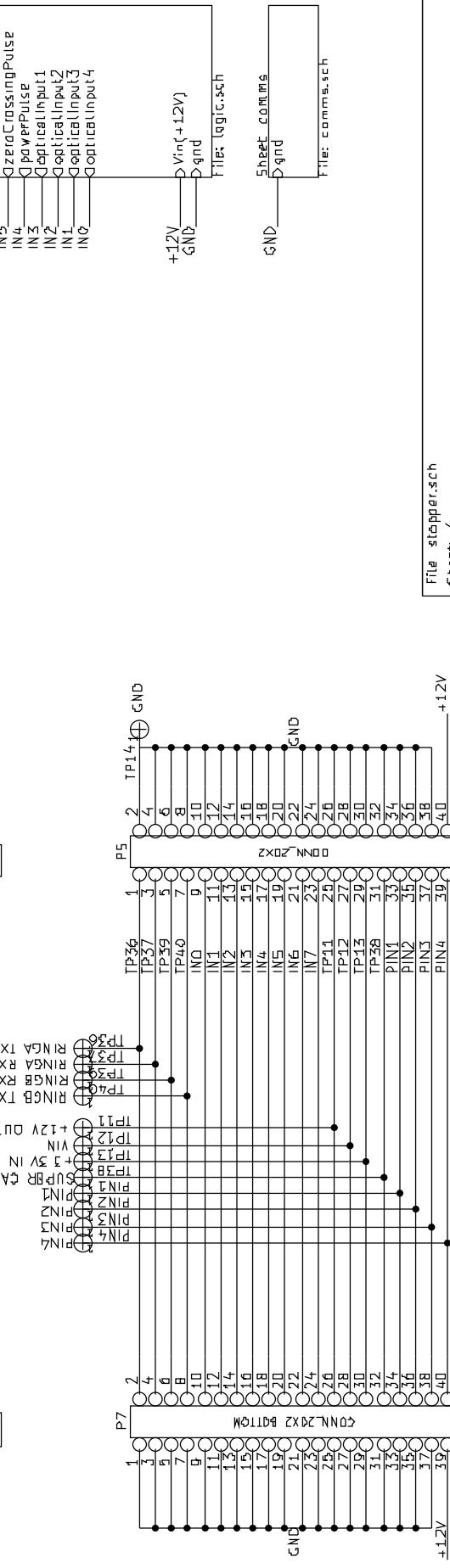
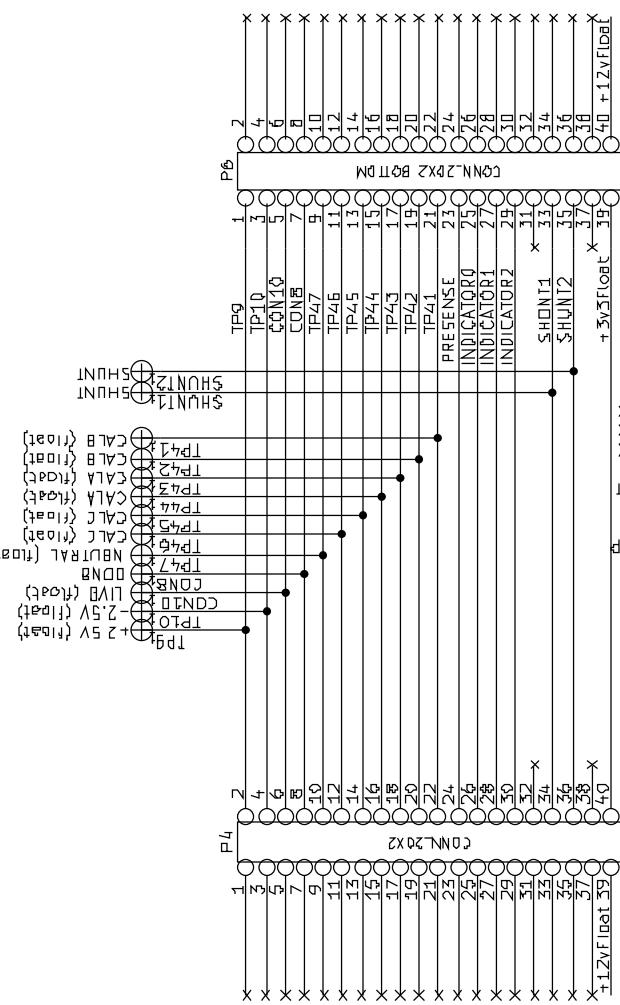


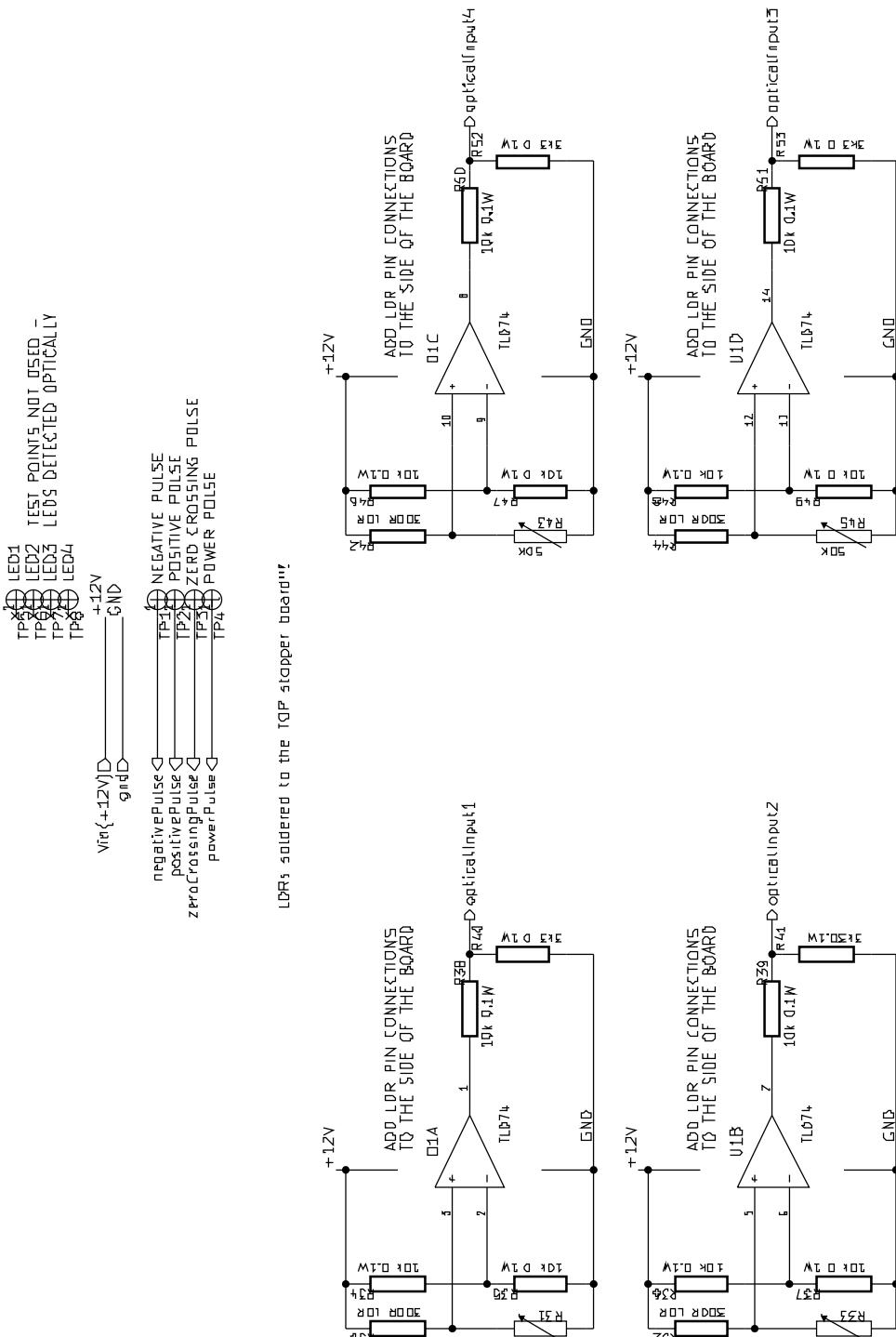
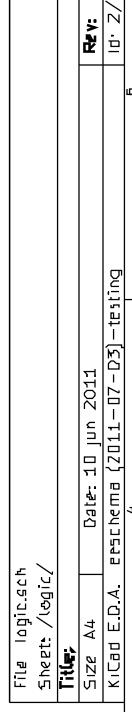




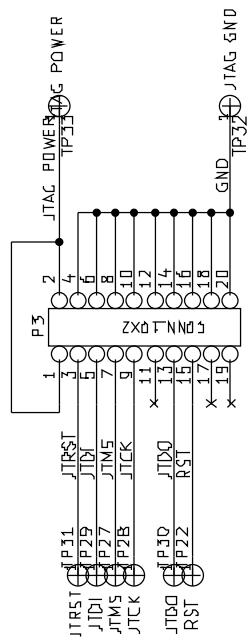
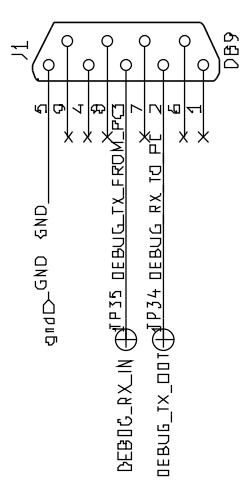


44 mounting holes in the corners of the board. S below the UDT to support the downward insertion force





File: Command.sch
 Sheet: /clmtrans/
Title:
 Size: A4 Date: 11 Jun 2011
 KiCad EDA, Revision: (2011-07-D3)-testing Rev: 10-3/3
 4



1 2 3 4 5

A

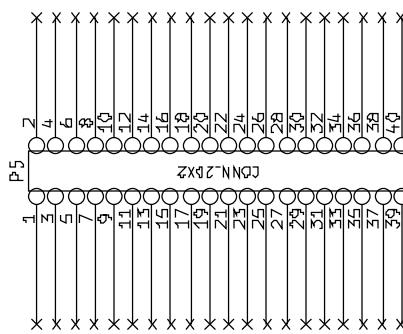
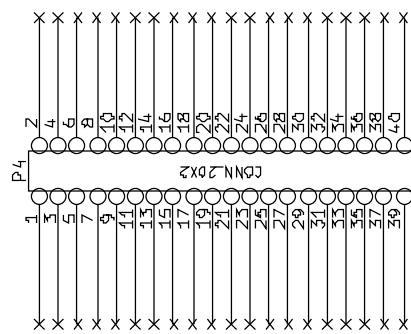
B

C



4 mounting holes in the corners
of the board, 5 below the I₂C
to support the downward
insertion force

*Note: 4 mounting holes in the corners
of the board, 5 below the I₂C
to support the downward
insertion force*



*Note: The INTERLOCK and COUNTER lines need to be routed to output
lines that are connected to the same +12V rail (and ground)*

M1 M2 M3 M4 M5

Sheet: Interfaces

X	INPUT8	X	INPUT16	X	INPUT24	X	INPUT32	X	INPUT40	X	ANALOG8	X	ANALOG16	X	ANALOG24	X	ANALOG32	X	ANALOG40	X
X	INPUT7	X	INPUT15	X	INPUT23	X	INPUT31	X	INPUT39	X	ANALOG7	X	ANALOG15	X	ANALOG23	X	ANALOG31	X	ANALOG39	X
X	INPUT6	X	INPUT14	X	INPUT22	X	INPUT30	X	INPUT38	X	ANALOG6	X	ANALOG14	X	ANALOG22	X	ANALOG30	X	ANALOG38	X
X	INPUT5	X	INPUT13	X	INPUT21	X	INPUT29	X	INPUT37	X	ANALOG5	X	ANALOG13	X	ANALOG21	X	ANALOG29	X	ANALOG37	X
X	INPUT4	X	INPUT12	X	INPUT20	X	INPUT28	X	INPUT36	X	ANALOG4	X	ANALOG12	X	ANALOG20	X	ANALOG28	X	ANALOG36	X
X	INPUT3	X	INPUT11	X	INPUT19	X	INPUT27	X	INPUT35	X	ANALOG3	X	ANALOG11	X	ANALOG19	X	ANALOG27	X	ANALOG35	X
X	INPUT2	X	INPUT10	X	INPUT18	X	INPUT26	X	INPUT34	X	ANALOG2	X	ANALOG10	X	ANALOG18	X	ANALOG26	X	ANALOG34	X
X	INPUT1	X	INPUT9	X	INPUT17	X	INPUT25	X	INPUT33	X	ANALOG1	X	ANALOG9	X	ANALOG17	X	ANALOG25	X	ANALOG33	X

File: interface.sch

File Base SCh
Sheet /

Title:

KICad E.O.A.

Size:

A4

Date:

5 Jul 2011

Version:

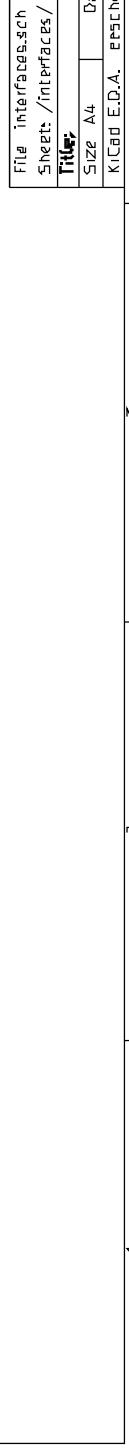
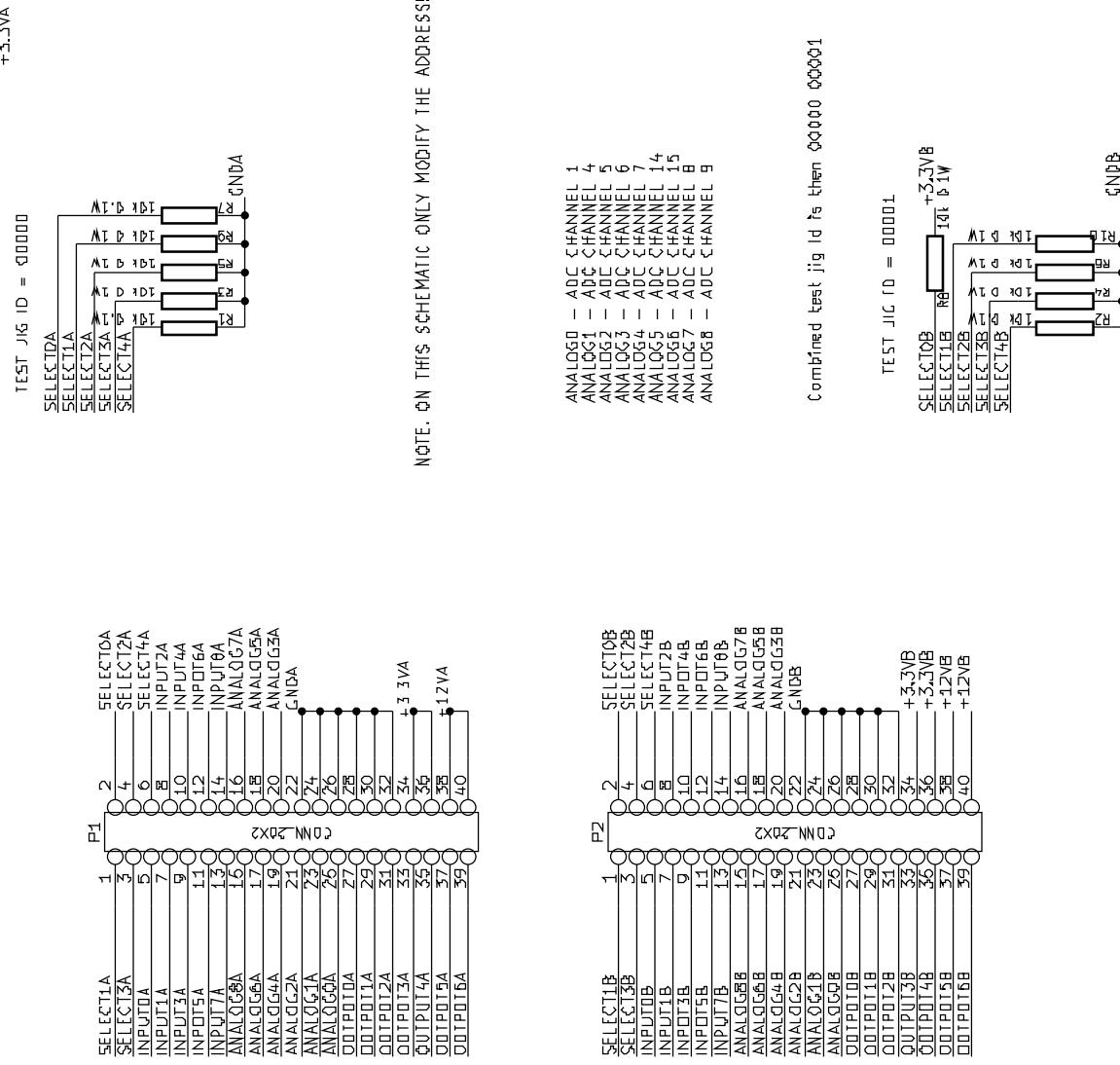
4

Rev:

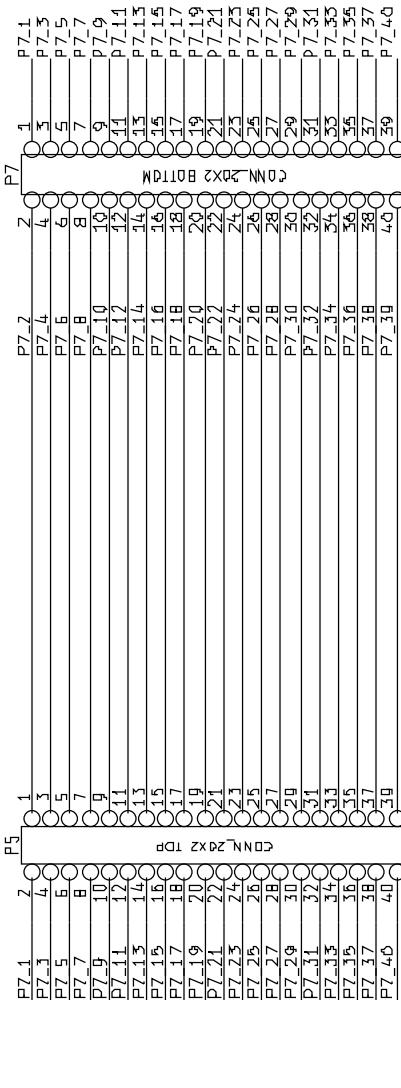
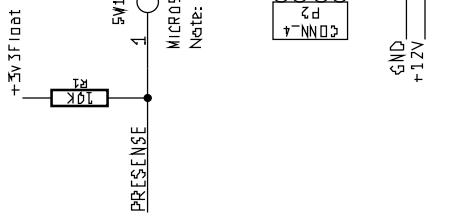
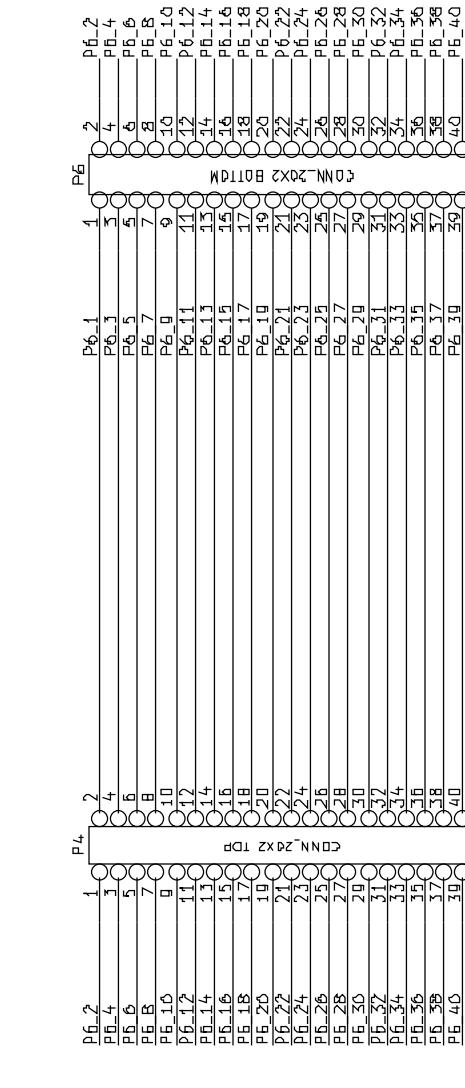
10-1/2

Page:

5



4 mounting holes in the corners of the board, 5 below the QFP to support the downward insertion force



Note
The indicator lines need to have lines assigned to them in the place of the
Net names INDICATOR-0-2. The presence switch needs to have an input line
assigned to it. instead of the PRESENCE net name. a 12V power line that
on the same ground as the INDICATOR lines needs to be brought up and a
3.3V and GND line needs to be brought up for the presence switch from the
base.

File stoppers.sch

Sheet /

Title

Size A4

Date: 5 Jul 2011

Rev:

10-1/2

File commes.sch

Sheet /

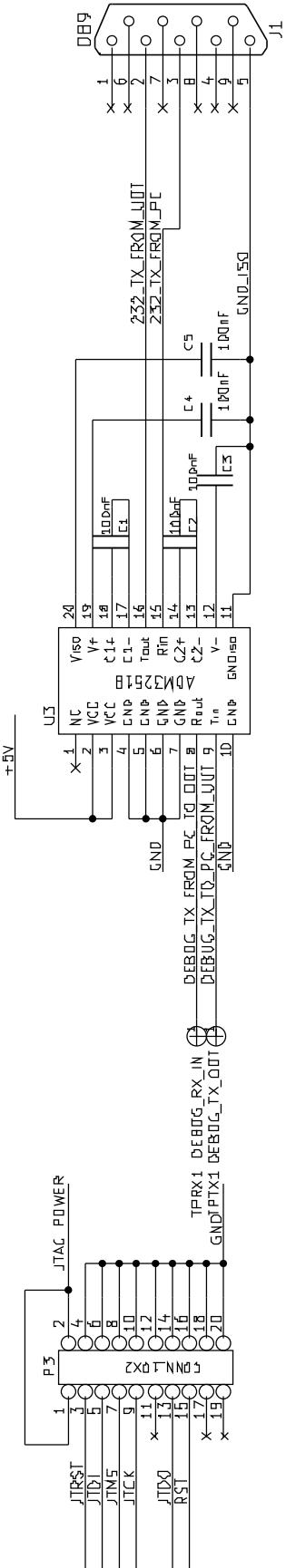
Title

Size A4

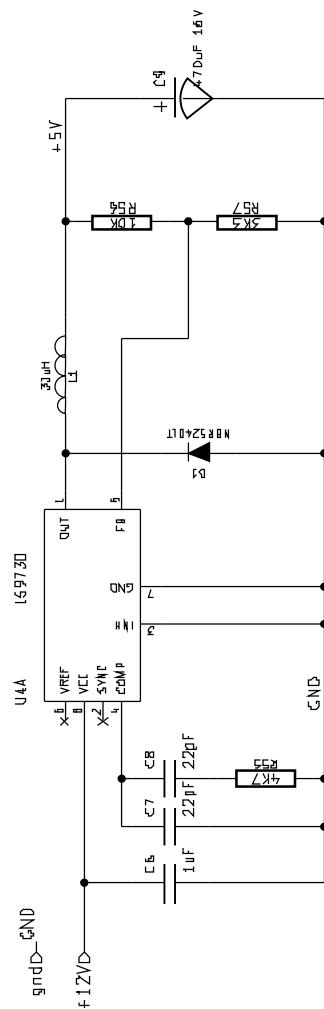
Date: 5 Jul 2011

Rev:

5



Note:
The +5V power supply for the Jcoupler can either come from the U1 circuit or be brought up from the base2, in which case the D1 circuit can be removed.
U1_RX goes to the DEBUG_RX test point on the DUT
D1_RX goes to the DEBUG_TX test point on the DUT
JTAG goes to the corresponding JTAG test points



File Comments		Sheet /Comments/
Title:		Date: 5 Jul 2011
Size: A4	Date: 5 Jul 2011	Rev: 10. 2/2
KICAD E.O.A.	Design: (2011-07-D3)-testing	
		4
		1
		2
		3
		4
		5

APPENDIX F.2

This Appendix contains the firmware snippets for the designs in Iteration 5. The following files are used:

- drv_Rs485.h and drv_Rs485.cpp
- brg_Rs485.h and brg_Rs485.cpp
- brg_processor.h and brg_processor.cpp
- brg_processorMaster.h and brg_processorMaster.cpp
- brg_processorSlave.h and brg_processorSlave.cpp

All of these files are available on the cd in:

</DesignFiles/Firmware/Prototype/PrototypeFirmware.zip>

The Java Application also uses the following files:

- DetectorJig.java
- genericTestboard.java
- GenericTestJig.java
- JigImplementation.java
- McuJigImplementation.java
- McuTestBoard0.java
- McuTestBoard1.java
- PulseDataBlock.java

All of these files are available on the cd in:

</DesignFiles/Application/ApplicationFiles.zip>

APPENDIX G

The EOS operating system was developed in house to deal with the specific architectural requirements of the Util Labs system. EOS devices sit in a wider System designed around the concept of a central messaging bus that allows each device to communicate with each other device in the system based on a standard routing protocol. Exact details of the system are available on written request from Util Labs. This section will just give a general overview of the system and the flow of information to give some context to the design of the generic test jig framework as the generic test jig boards use the EOS operating system and most of the boards that will be tested will use it as well.

System architecture overview

All information sent on the virtual message bus is sent or received in the form of messages. Each message contains a message ID that defines what kind of information it contains, the data itself, some error checking information and routing information. Data can be routed UP or DOWN the message bus.

The following systems archetypes are discussed:

- **Server**

The server is the central repository of data and settings. It consists of a number of data , communications and web servers which are used to store, correlate and report on all data that is collected by the wider system. The server is connected to the rest of the system via GPRS connections to each of the concentrators.

- **Concentrators**

Concentrators connect large numbers of local concentrators into cohesive networks. Concentrators use a GPRS connection to communicate to the server, and a mesh network implementation on PLC to connect to the Local concentrators below it. Each concentrator contains all of the information required and contained in its network.

- **Local Concentrators**

Local concentrators are masters of small local area networks that are constrained by their physical size as all of the nodes in the local network are connected to the local

concentrator via a serial connection. They connect to Concentrators via a mesh network on PLC. Each local concentrator contains all of the information required and contained in its local network.

- **Local Nodes**

Nodes are the generators of information, and can be measurement devices, user terminals or any other device that generates data.

Information flow in the system is different depending on which direction the information is traveling. For an UPWARDS request/Response, the destination of a message will always be the server.

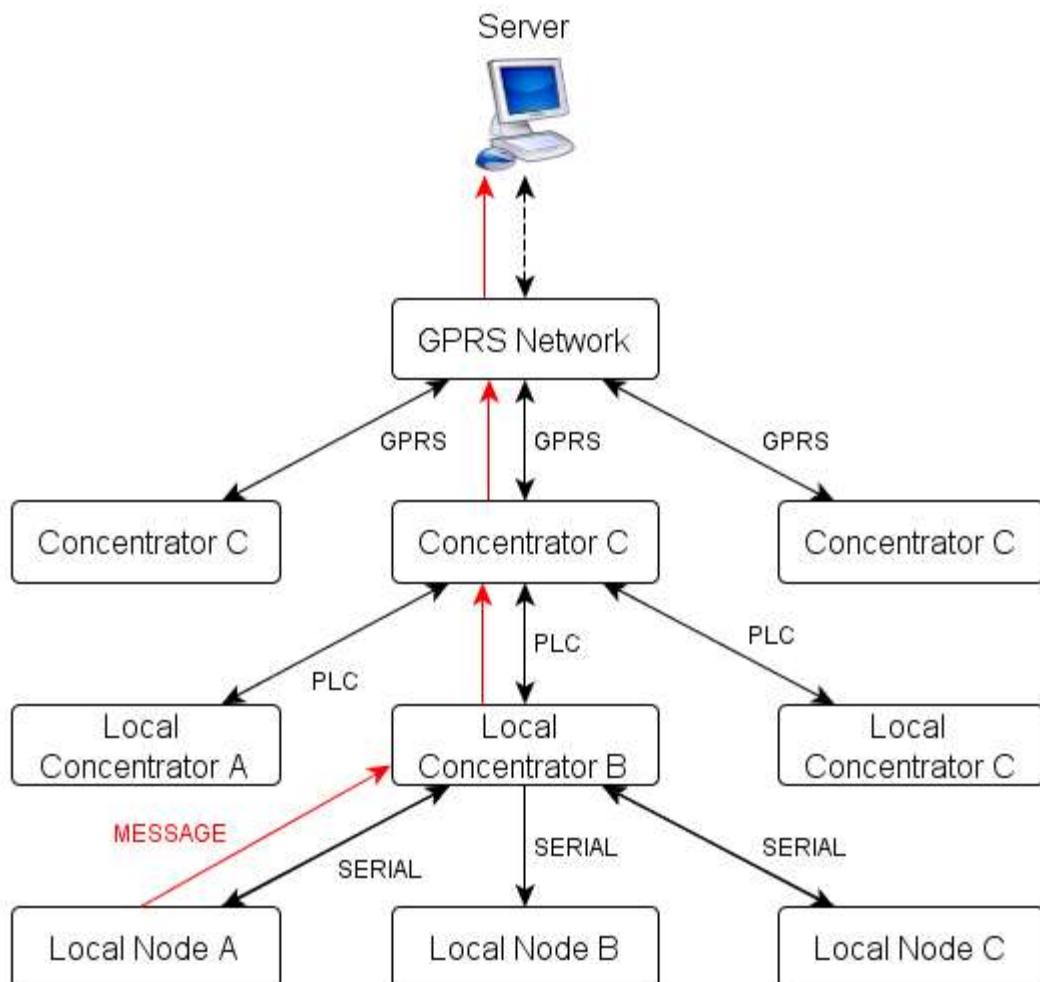


Illustration 58: Virtual Message Bus Showing An UPWARDS Routed Message

For example, if a local node generates some measurement, for example an hourly temperature measurement, this value will get sent via its serial port with its routing destination set to the server. This message will be received by the local concentrator and get buffered into a file on the local Concentrator that contains all of the temperature measurements of all of the local nodes on the local network, and then every 6 hours these temperature measurements are packed into a message that is then sent from the Local concentrator via the PLC network to the Concentrator, which collects similar temperature messages from all of the other Local Concentrators and stores them in a file. Every 24 hours these stored values are packed into a message by the Concentrator and forwarded up to the server which will collect these packed messages from all of the other Concentrators. So even though the final destination of the generated data was the server, the data will take 24 hours to reach the server databanks. Depending on the priority of the data being collected, this data can be collected and not packaged but sent upwards directly. For example in the event of an over temperature alarm, the message is given priority and sent out of the serial port immediately to the local concentrator that acknowledges that the node on its local network is in alarm state, and sends an immediate network alarm to the Concentrator which then acknowledges that there is a local network inside of its network that has an active alarm and sends a message to the server that will notify an operator via email that some action needs to be taken. However, in the event of a request from a lower level, for example, if a local node requests the time from the server, then the request will be sent to the server but will first travel to the local concentrator, which will respond directly to the request with the time since it contains all of the information required by its local network. For a DOWNWARDS request/Response then, the server/device always sends to a specific unit, as it knows about all of the devices below it in the hierarchy. So in effect, each device “sees” on the message bus that it is directly connected to the server above it with a number of slave devices below it based on its level in the network hierarchy.

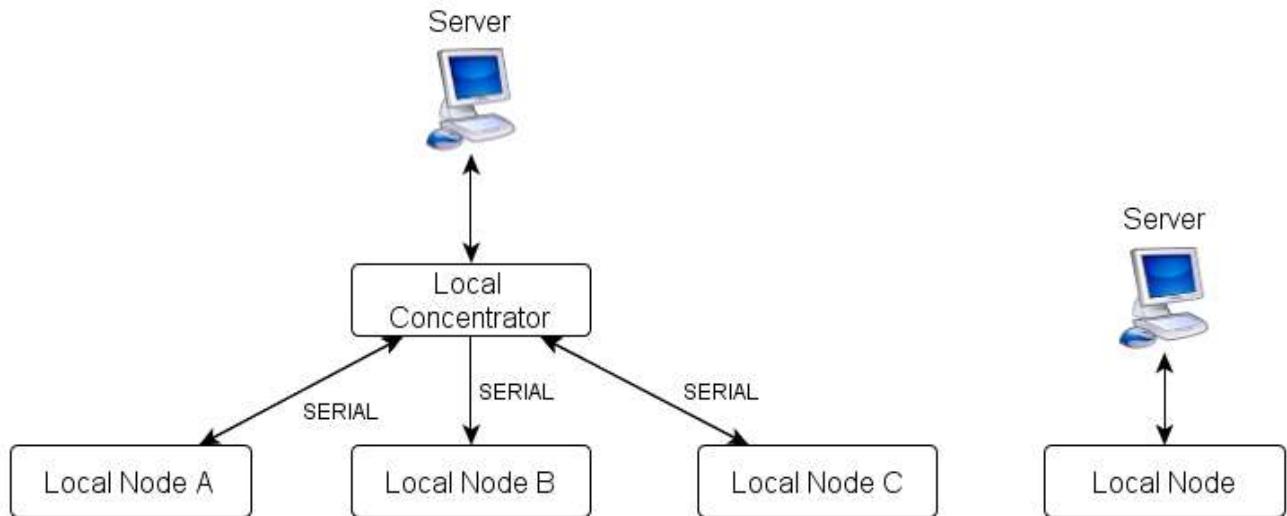
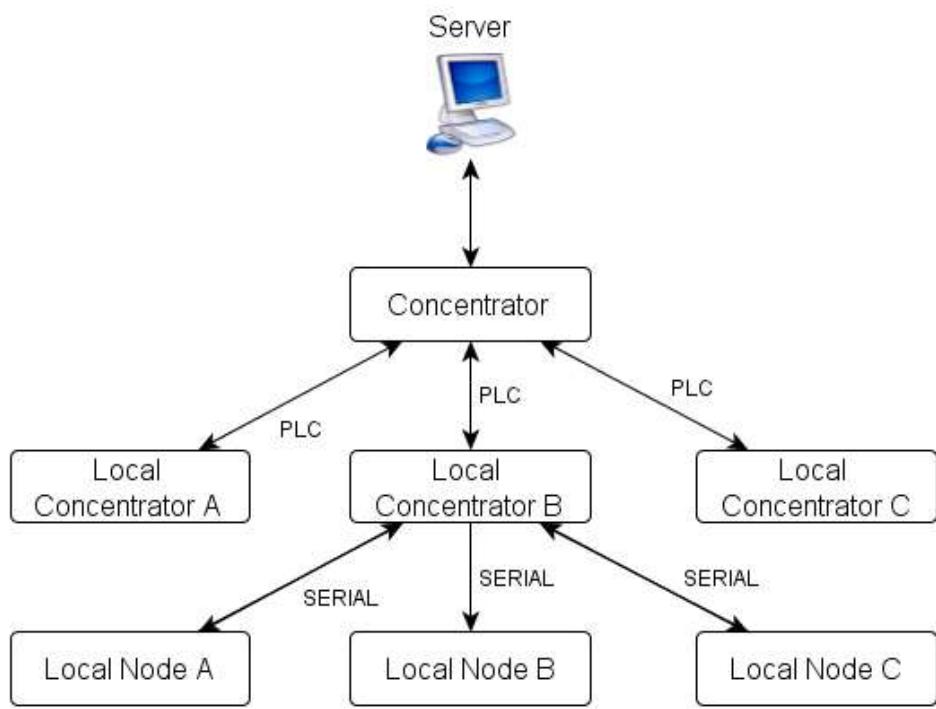


Illustration 59: What Each Device Connected To The Virtual Message Bus "Sees"

EOS Architecture

This message bus architecture between devices is continued inside each individual device running the EOS operating system as well.

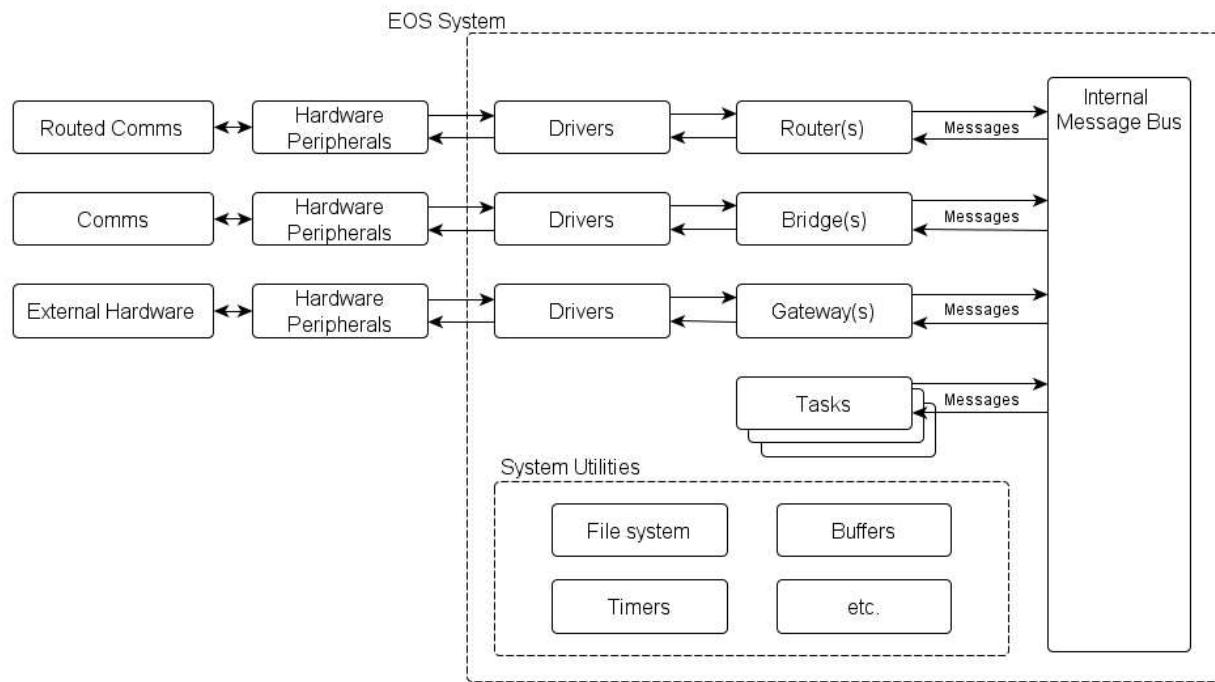


Illustration 60: EOS Operating System Modules Interactions

The internal message bus is connected to individual Tasks, Gateways, Bridges and Routers, each of which send and receive messages.

- **Tasks**

Tasks are the work horses in the EOS operating system and they are responsible for all data processing in the system. Tasks track the state of Device, store and retrieve data and schedule actions. In the Generic Test Jig system, the InputStateTracker is a task that gathers information about the states of pins on the system and tracks the current input state of the device, and responds to requests against this state.

- **Gateways**

Gates are interfaces between the external hardware peripherals and the message bus. They create virtual modules to represent external hardware on the device. For example, on the Generic Test Jig system the ADC peripheral is controlled via a gateway, that accepts message requests for voltage measurements. When such a request is received, the required number of measurements are done, and once the data has been collected, the gateway will send a response message onto the message bus that contains the collected data. Gateways are always associated with drivers

since these drivers allow gateways to be used generically. For the ADC example above, the gateway will be identical regardless of whether the on board processor ADC peripheral is used or if the SPI bus is used to take a measurement from an external ADC chip. Provided the driver maintains the same interface to the Gateway, the code remains identical.

- **Bridges**

Bridges allow external hardware busses to be connected directly to the internal message message bus of the device. For example, on the generic test jig board, the debug bridge allows the host application to directly inject requests to the for state, voltage etc. All messages received by the bridge from the internal message bus will also be pushed out on to the hardware bus, allowing the external system to receive information as well. The difference between a bridge and a router is that a bridge generally deals with unrouted information. In other words ALL message received from the internal message bus will be sent to the external bus. This characteristic of the bridge was exploited in the prototype to create the inter-processor bus and allow 2 processors housing different tasks and gateways to act as 1 processor by effectively joining their message busses together. However some filtering had to be added into the inter processor bus so that similar tasks and gateways on the 2 processor didn't end up responding to their own messages.

- **Routers**

Routers are similar to bridges, except that they deal with routed messages. These routed messages allow for the upwards and downwards sending of messages discussed in the system architecture above. Routers tie the internal and external messaging busses together. There are no routers on the generic test jig board.

Since the system architecture and the internal architecture is tied together through the routers, any message can be sent to any device in the network provided it follows the flow of messages described in the system architecture above. For example it is possible to light a local LED on a device from the server, purely by sending a message from the server with that device's serial number in the routing field and the underlying virtual message bus will ensure that message arrives at its intended destination.

The internal system also has a number of utilities that each of the above modules can use that perform certain generic functions, like for example storing and retrieving files on a Flash File System, scheduling and getting notified by timers etc.

The Port of the operating system to a new device involves making sure that all of the device drivers are available for that devices peripherals and then ensuring that gateways map correctly onto its peripheral drivers. This porting is done by adding any drivers that have been created and then setting up the mappings in the sys_platform.h file. The port information for the generic test jig boards is shown below:

- The routing device type is set (DEVICE_TYPE_MASTER_CONTROLLER) , which would put the generic test jig on the level of a Concentrator in the system architecture above.
- The processor type is then chosen as STM32F103 for the ST arm cortex processor.
- The system clock is set to 72Mhz which will allow the underlying operating system to use the PLL on the 8Mhz clock.
- Individual peripherals are then enabled or disabled on the advanced peripheral busses
- The file system information is added (the discussion of the underlying flash file system is beyond the scope of this document, but any required information can be requested in writing from Util Labs.), including the mappings for SPI port of the dataflash chip where the flash file system data is stored.
- The pins connected to the ADC peripheral are all set to high impedance inputs.
- The GPIO pins are then created as instances of InputPin and OutputPin objects with their associated mapping objects so that they can be used by they GPIO gateway.
- Any GPIO input pins that will be required to be detect pulses faster than 20ms can be set up as fastInputs as well. In the case of the generic test jig, all of the inputs need to have this functionality.

Setting the above information in the plt_system.h file (as can be seen on the accompanying CD in the directory /firmware/Include/platform/plt_system.h) will allow the eos operating system to be run on the generic test jig hardware.

REFERENCES

- [1] "ISO9000", Wikidepia (last updated 13 June 2011)
available online:
http://en.wikipedia.org/wiki/ISO_9000 (retrieved 19 June 2011)
- [2] "Java SE Desktop Overview", Oracle Sun Developers Network (2010)
available online:
<http://java.sun.com/javase/technologies/desktop/> (retrieved 19 June 2011)
- [3] Seica Systems, "Virtuous ICT test and virtual bed of nails."
available online:
<http://www.seica.com/di/c/cd/Redazionali%20Home/PCB%20marzo%202010ENG.pdf>
- [4] "Fixtureless in-circuit test", Wikidepia (last updated 30 November 2009)
available online:
http://en.wikipedia.org/wiki/Fixtureless_in-circuit_test (retrieved 24 June 2010)
- [5] "Boundary scan", Wikidepia (last updated 21 May 2010)
available online:
http://en.wikipedia.org/wiki/Boundary_Scan (retrieved 24 June 2010)
- [6] "USB Crossconnect for ARM", Rowley (last updated 30 June 2010)
available online:
<http://www.rowley.co.uk/arm/CrossConnect.htm> (retrieved 24 June 2010)
- [7] "Built-in self-test", Wikidepia (last updated 15 June 2010)
available online:
http://en.wikipedia.org/wiki/Built-in_self-test (retrieved 24 June 2010)
- [8] Scheiber, S.F., "Building a successful board test strategy" Newnes Publishers, USA 2001
- [9] STMicroelectronics , "Application Note AN2606 : STM32TM microcontroller system memory boot mode ", STMicroelectronics , USA, 2010
available online:
<http://www.st.com> (retreived 24 June 2010)

[10] STMicroelectronics , “Reference Manual RM0008: STM32F101xx, STM32F102xx, STM32F103xx,STM32F105xx and STM32F107xx advanced ARM-based 32-bit MCUs”
STMicroelectronics , USA, 2011

available online:

<http://www.st.com> (retrieved 24 June 2010)

[11] “Pogo Pin”, Wikidepia (last updated 15 June 2010)

available online:

http://en.wikipedia.org/wiki/Pogo_pin (retrieved 24 June 2010)

[12] Ingun Prüfmittelbau GmbH,“Test Probes Catalog 2010/2011”,Ingun, Germany 2010

available online:

<http://www.ingun.com> (retreived 23 June 2010)

[13] SEGGER Microcontroller GmbH & Co ., “User Manual UM08010-R3: J-Link JTAG Isolator ”, SEGGER, Germany 2008

available online:

<http://www.segger.com> (retrieved 24 June 2010)

[14] J. Ben Hadj Slama et al, “Study and Modelling of Optocouplers ageing”, Journal of Automations and Systems Engineering, 2008 Volume 2

[15] Analog Devices, “ICoupler Digital Isolation – Unparalleled Performance and Integration”, Analog Devices, Boston USA

available online:

http://www.analog.com/en/interface/digital-isolators/products/CU_over_iCoupler_Digital_Isolation/fca.html?ref=ASC-PR-296-a
(retrieved 24 June 2010)

[16] Baoxing Chen, “iCoupler ® Products with isoPower™ Technology: Signal and Power Transfer Across Isolation Barrier Using Microtransformers”, Analog Devices, Boston, USA

available online:

<http://www.analog.com/static/imported-files/overviews/isoPower.pdf>

- [17] "Decimation (Signal Processing)", Wikidepia (last updated 27 May 2010) available online:
http://en.wikipedia.org/wiki/Decimation_%28signal_processing%29
(retrieved 24 June 2010)
- [18] On project disk /Datasheets/74LV541.pdf
- [19] On project disk /Datasheets/STM32F103RE.pdf
- [20] On project disk /Datasheets/CNY74-4.pdf
- [21] On project disk /Datasheets/AD7908_AD7918/AD7928.pdf
- [22] On project disk /Datasheets/AD780.pdf
- [23] On project disk /Datasheets/MAX485.pdf
- [24] On project disk /Datasheets/ADM3251E.pdf
- [25] On project disk /Datasheets/2EDGK-75.pdf
- [26] On project disk /Datasheets/2EDGRC-75.pdf
- [27] On project disk /Datasheets/LDR9001.pdf
- [28] On project disk /Datasheets/L5973D.pdf
- [29] On project disk /Datasheets/LM78xx.pdf
- [30] On project disk /Datasheets/ULN2003A-D.pdf
- [31] On project disk /Datasheets/D2N-RELAY.pdf
- [32] On project disk /Datasheets/TL074.pdf