MyElectronicProjects Documentation

Release 0.0.0

ponty

December 03, 2012

CONTENTS

MyElectronicProjects

Date December 03, 2012

PDF MyElectronicProjects.pdf

CHAPTER

ONE

ABOUT

Hobby electronic projects built by me.

Most of them are built on stripboard.

Links:

- home: https://github.com/ponty/MyElectronicProjects
- documentation: http://ponty.github.com/MyElectronicProjects

Design tool: EAGLE Light Edition

CHAPTER

TWO

STRIPBOARD DESIGN

Stripboard design representation in eagle:

- holes: copper should be cut or drilled here
- SMD: through-hole component, legs are drawn on top layer
- top layer: wires
- lines on documentation layer: wires
- bottom layer: original parallel strips of copper, only those are drawn, which are used for connection
- via: soldering points

Some components have no 3D view in the documentation.

CHAPTER

THREE

AC MAINS ALARM

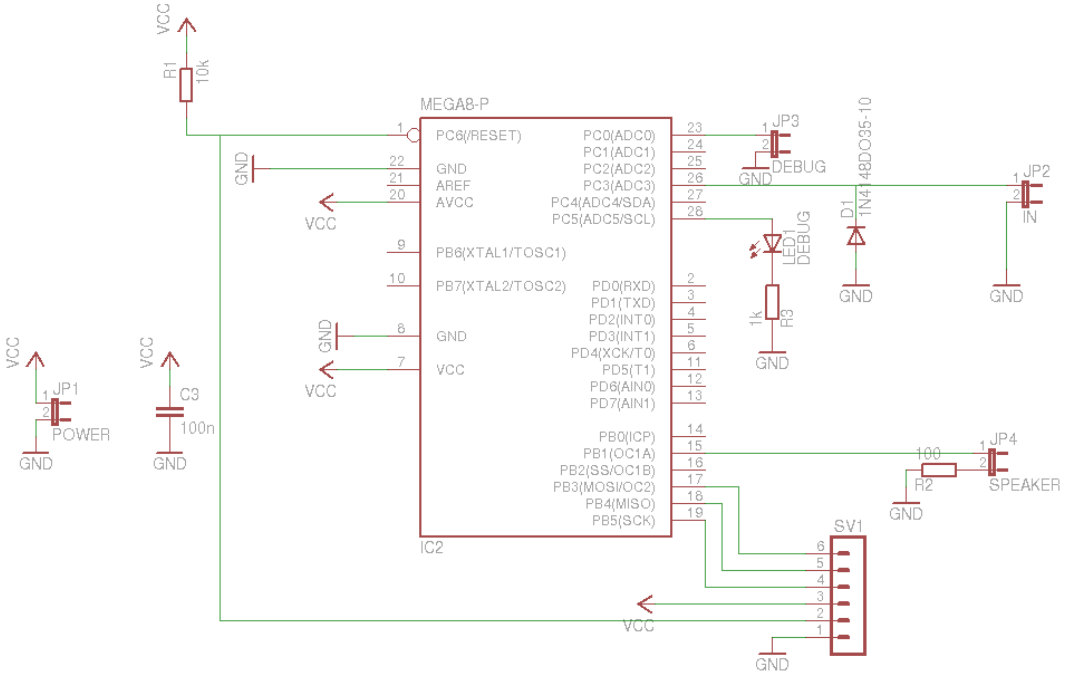
Status: OK

This device makes alarm sound periodically when the near cable has AC signal. In debug mode it can be used as AC mains detector.

3.1 Pins

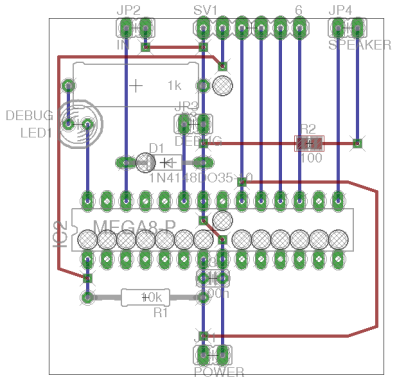
board pin	AVR pin	Arduino pin
in	PC3	A3
speaker	PB1	D9
debug in	PC0	A0
debug LED	PC5	A5

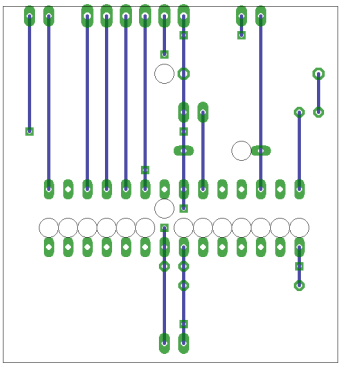
3.2 Schematic

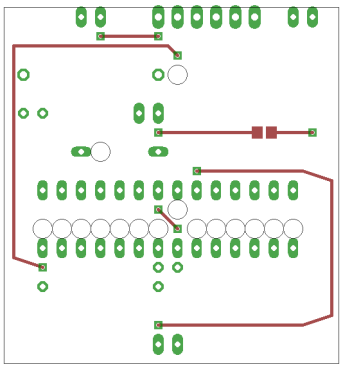


3.3 Board

Normal, bottom mirrored, wires only:







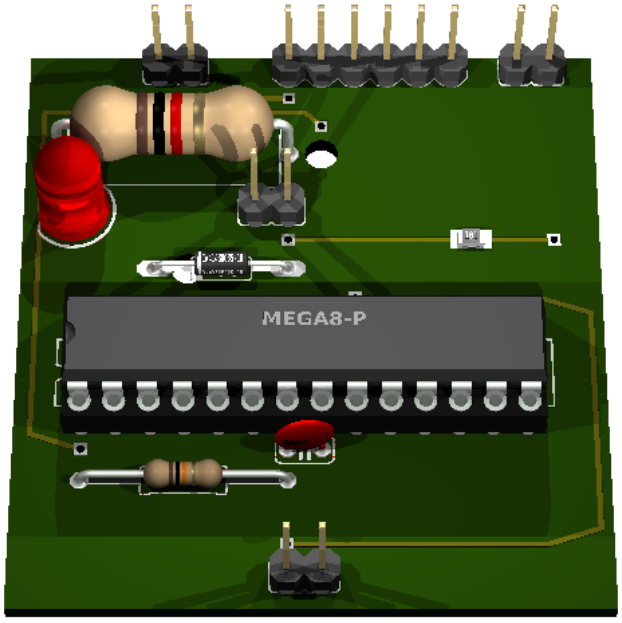
3.4 Partlist

Table 3.1:

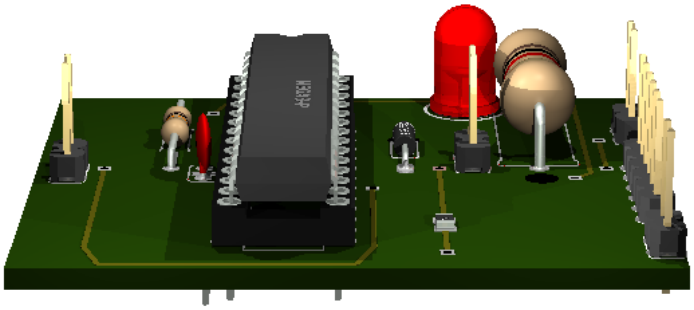
part	value	position
C3	100n	(1.45 0.4)
D1	1N4148DO35-10	(1.2 1)
IC2	MEGA8-P	(1.45 0.65)
JP1	POWER	(1.45 0)
JP2	IN	(1.05 1.7)
JP3	DEBUG	(1.35 1.2)
JP4	SPEAKER	(2.15 1.7)
LED1	DEBUG	(0.75 1.2)
R1	10k	(1.1 0.3)
R2	100	(1.95 1.1)
R3	1k	(1.05 1.4)
SV1		(1.65 1.7)

3.5 3D view

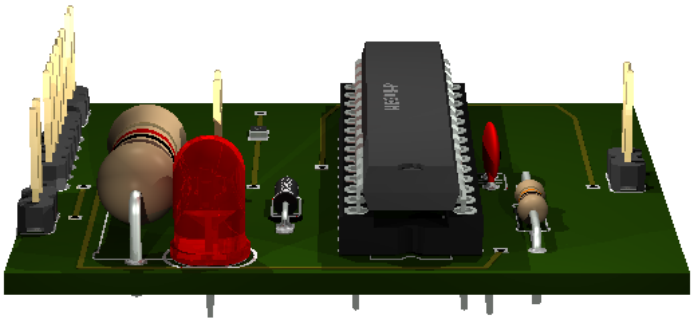
3.5.1 Front



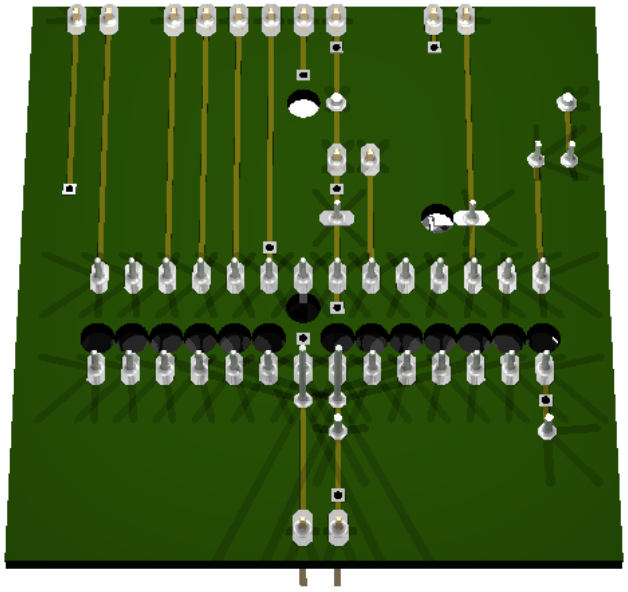
3.5.2 Right side



3.5.3 Left side



3.5.4 Bottom



3.6 Sources

Based on this design: http://www.edn.com/article/511304-Detect_live_ac_mains_lines.php

CHAPTER

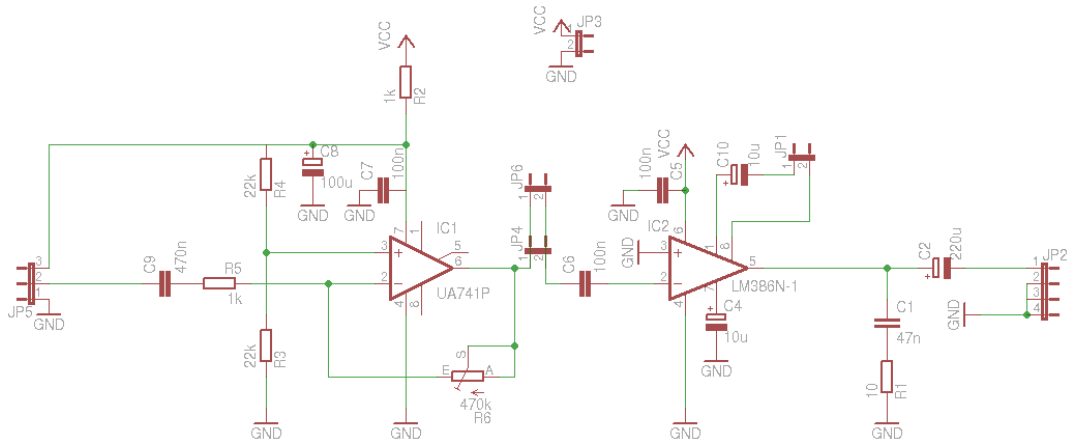
FOUR

AUDIO AMPLIFIER

Status: under construction

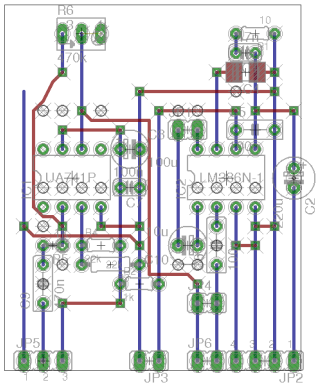
It is used for ...

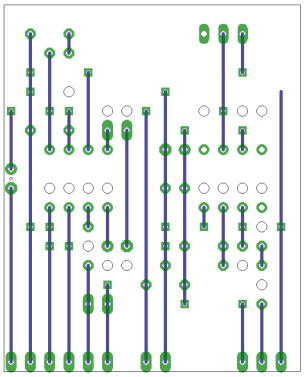
4.1 Schematic

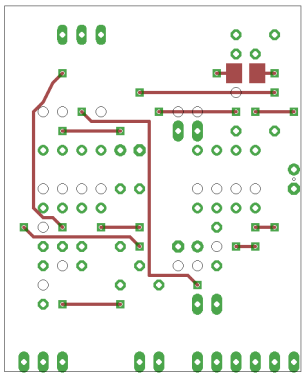


4.2 Board

Normal, bottom mirrored, wires only:







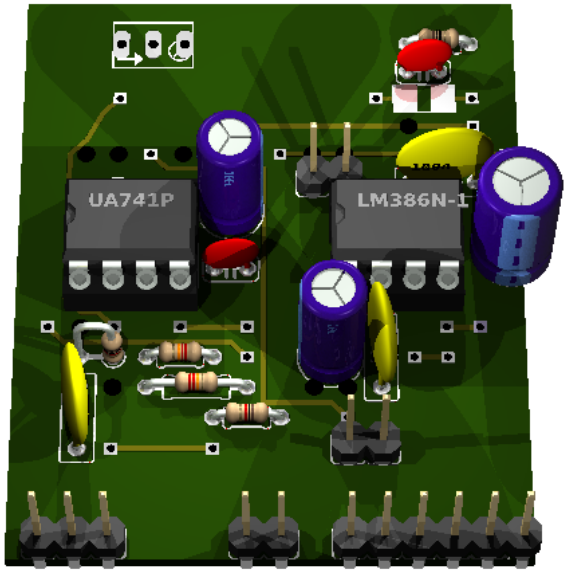
4.3 Partlist

Table 4.1:

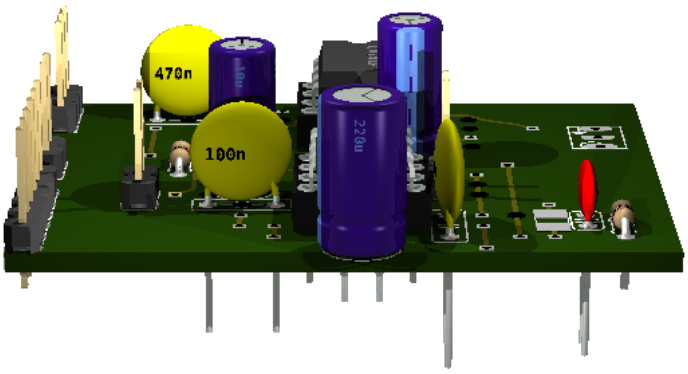
part	value	position
C1	47n	(1.45 1.7)
C2	220u	(1.7 1.05)
C4	10u	(1.45 1.6)
C5	100n	(1.5 1.3)
C6	100n	(1.3 0.7)
C7	100n	(0.85 1)
C8	100u	(0.85 1.2)
C9	470n	(0.4 0.5)
C10	10u	(1.15 0.7)
IC1	UA741P	(0.55 1.05)
IC2	LM386N-1	(1.35 1.05)
JP1		(1.15 1.3)
JP2		(1.55 0.1)
JP3		(0.95 0.1)
JP4		(1.25 0.4)
JP5		(0.4 0.1)
JP6		(1.25 0.1)
R1	10	(1.5 1.8)
R2	1k	(0.9 0.5)
R3	22k	(0.75 0.6)
R4	22k	(0.7 0.7)
R5	1k	(0.45 0.7)
R6	470k	(0.6 1.8)

4.4 3D view

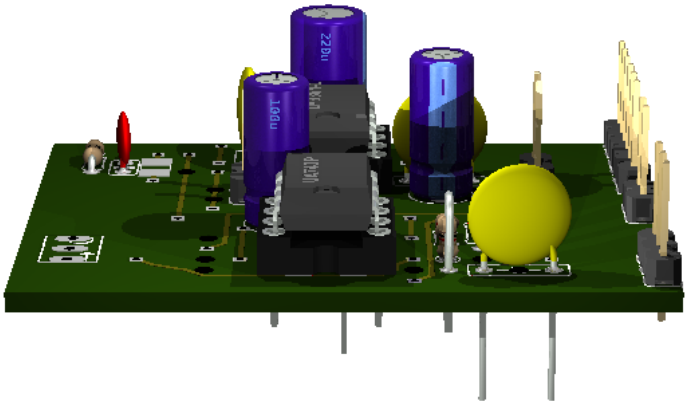
4.4.1 Front



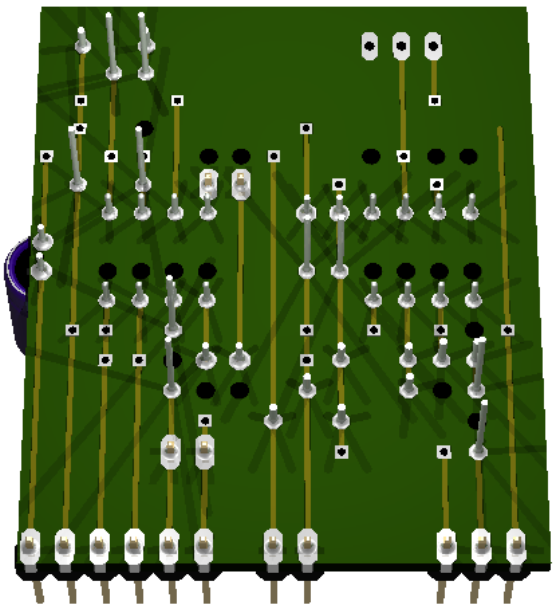
4.4.2 Right side



4.4.3 Left side



4.4.4 Bottom



4.5 Sources

original design

CHAPTER

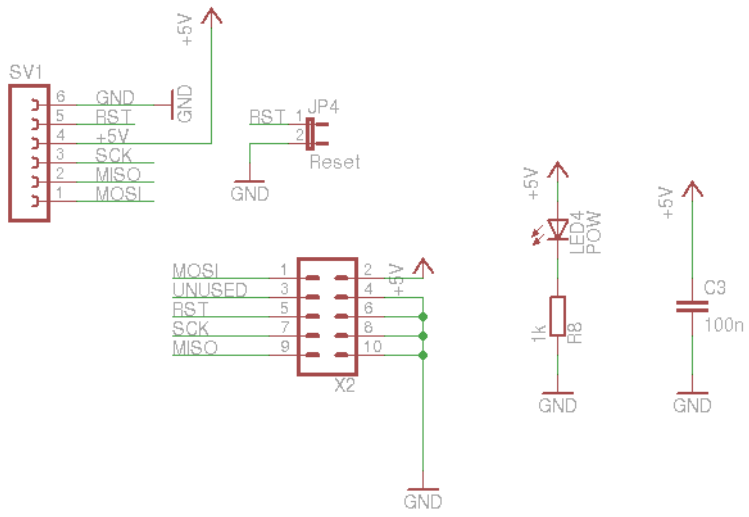
FIVE

AVR ISP HEADER ADAPTER

Status: OK

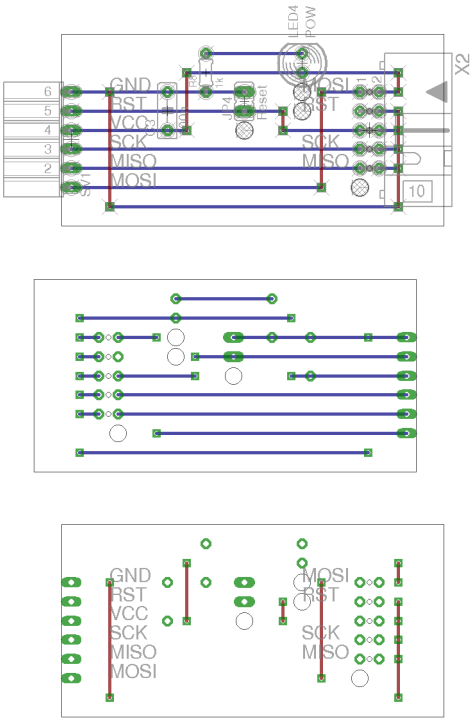
This board converts the AVR ISP 10 pin Header to a custom 6 pin header, which is easy to add to a stripboard with Atmega8 pin compatible microcontroller.

5.1 Schematic



5.2 Board

Normal, bottom mirrored, wires only:



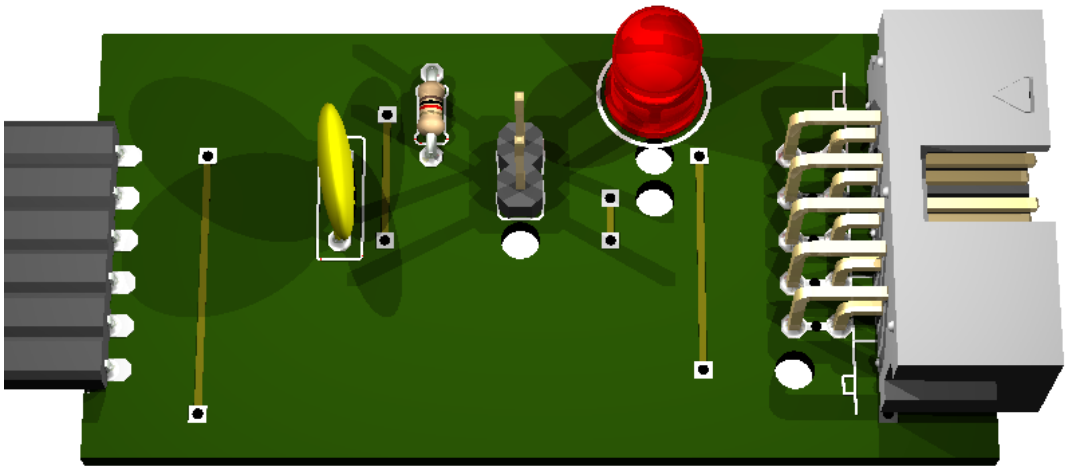
5.3 Partlist

Table 5.1:

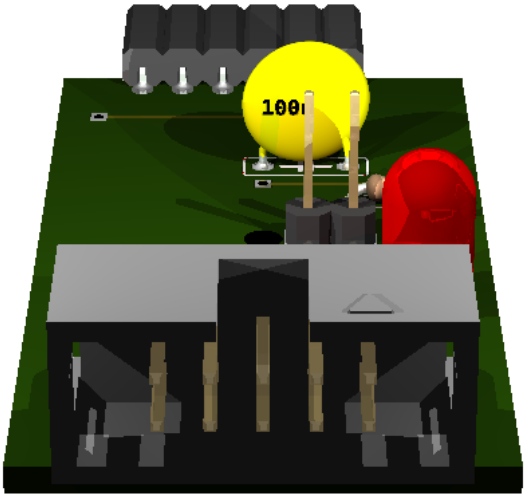
part	value	position
C3	100n	(0.6 0.7)
JP4	Reset	(1 0.75)
LED4	POW	(1.3 0.95)
R8	1k	(0.8 0.9)
SV1		(0.1 0.55)
X2		(1.65 0.6)

5.4 3D view

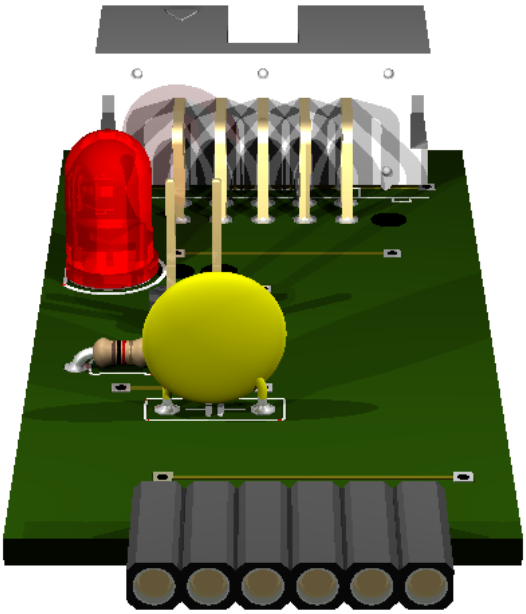
5.4.1 Front



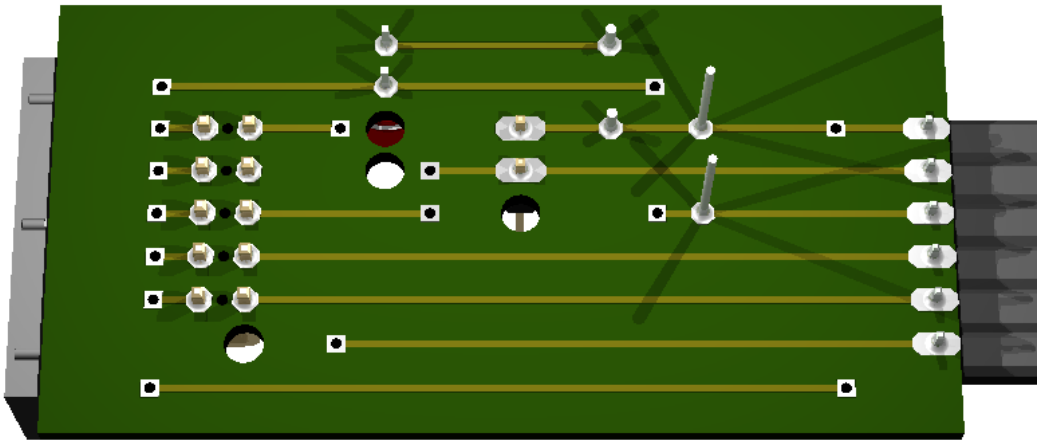
5.4.2 Right side



5.4.3 Left side



5.4.4 Bottom



5.5 Sources

AVR ISP Header Pinouts

CHAPTER

SIX

DAPA AVR PROGRAMMER

Status: OK

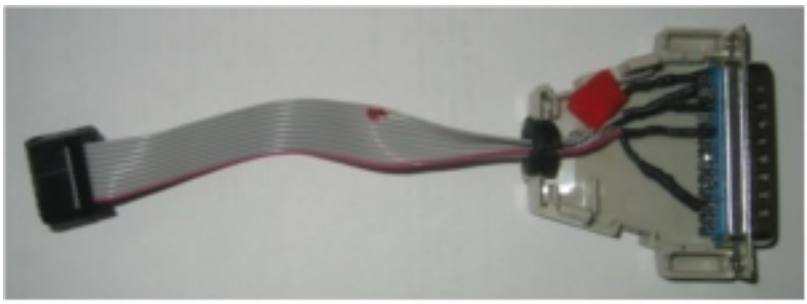
It is used for programming AVR controller and Arduino compatible boards using the parallel port.

6.1 Test on Ubuntu

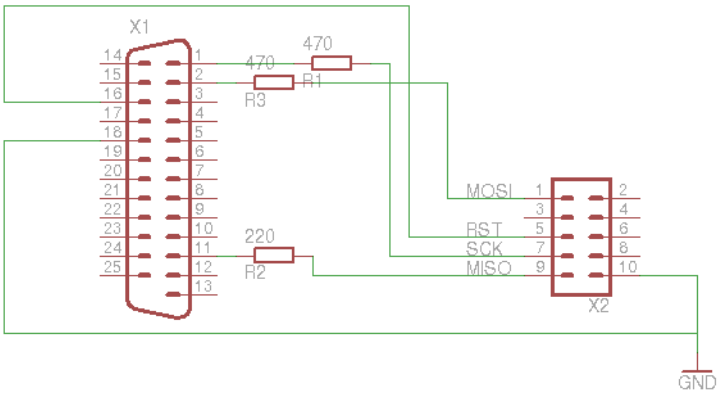
checking:

```
$ avrdude -patmega88 -cdapa

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e930a

avrdude: safemode: Fuses OK

avrdude done.  Thank you.
```

6.2 Image



6.3 Schematic



6.4 Partlist

Table 6.1:

part	value
R1	470
R2	220
R3	470
X1	
X2	

6.5 Sources

original design

Parallel port specification

AVR ISP Header Pinouts

CHAPTER

SEVEN

FM TRANSMITTER

Status: OK

source: http://rclindia.tripod.com/trans.html

CHAPTER

EIGHT

FTDI CABLE

Status: OK

Special cable.

connections:

FTDI pin	signal	color	6p4c (RJ14) pin
1	gnd	red	4
2	cts		
3	5v	green	3
4	rxd	yellow	2
5	txd	black	5
6	rts		

standard color code is reversed

8.1 Sources

RJ14 pinout

RJ14 wiring details

CHAPTER

NINE

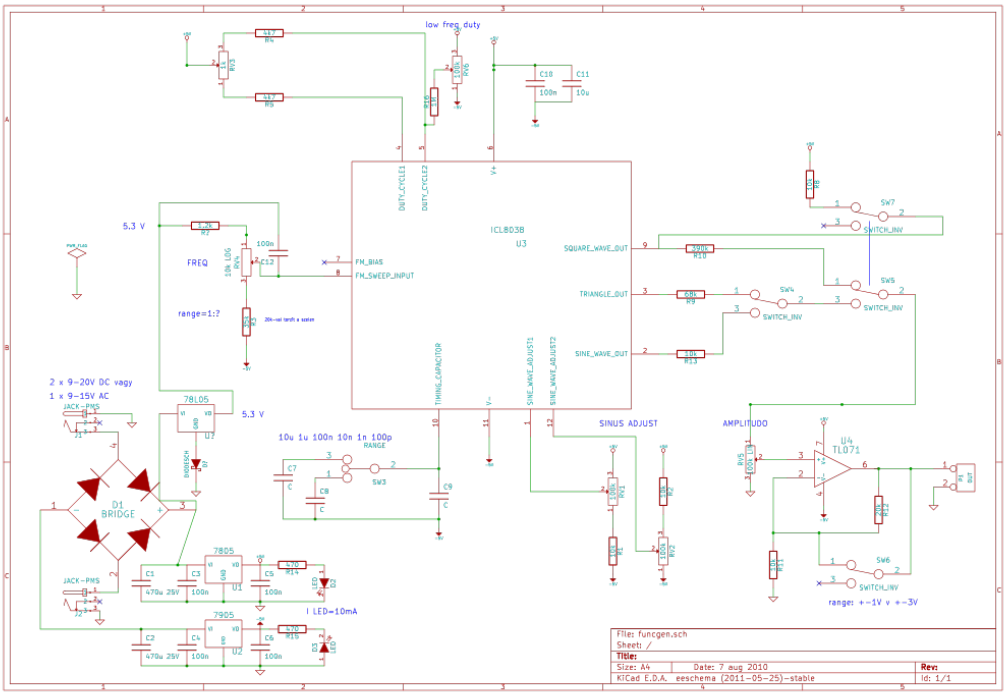
FUNCTION GENERATOR

Status: OK

Power input:

1x AC (9-15V) or 2x DC (9-20V)

9.1 Schematic



CHAPTER

TEN

GARMIN ETREX DATA CABLE

Status: OK

It is used for connecting Garmin eTrex to the serial port.

connections:

DB9 pin	garmin pin
3 (TxD)	2 (In)
2 (RxD)	3 (Out)
5 (GND)	4 (GND)

10.1 Images

10.2 Sources

original design

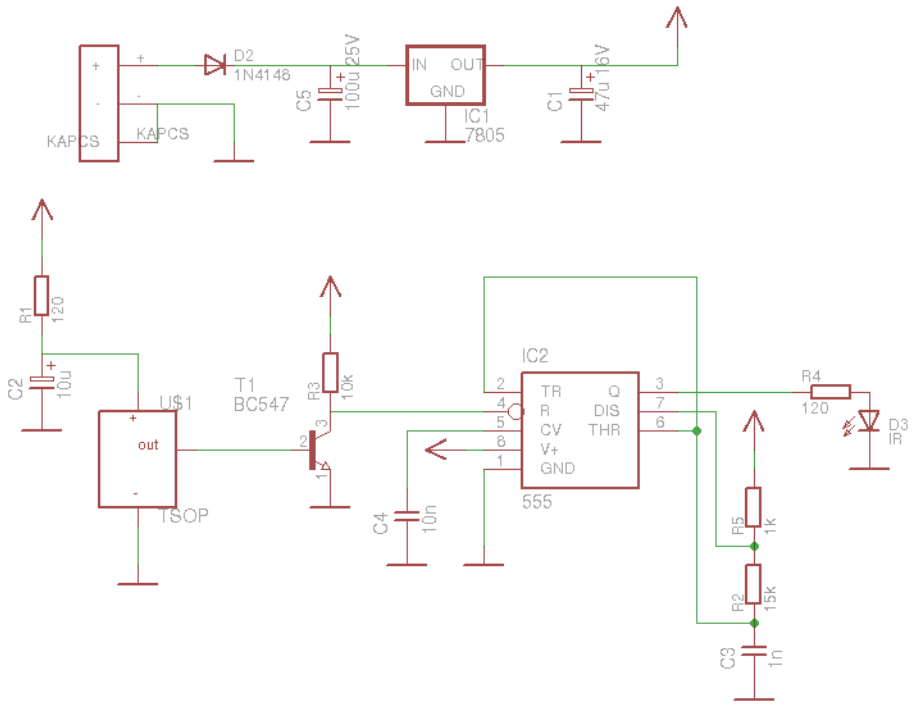
CHAPTER

ELEVEN

IR AMPLIFIER

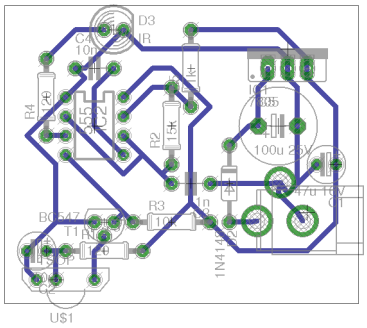
Status: OK

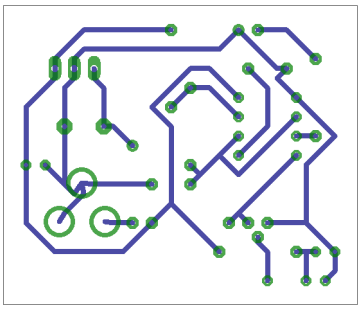
11.1 Schematic

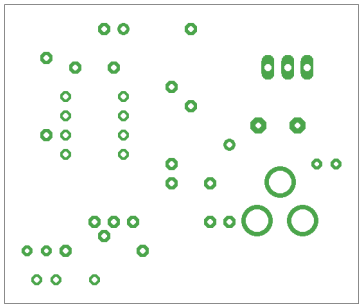


11.2 Board

Normal, bottom mirrored, wires only:







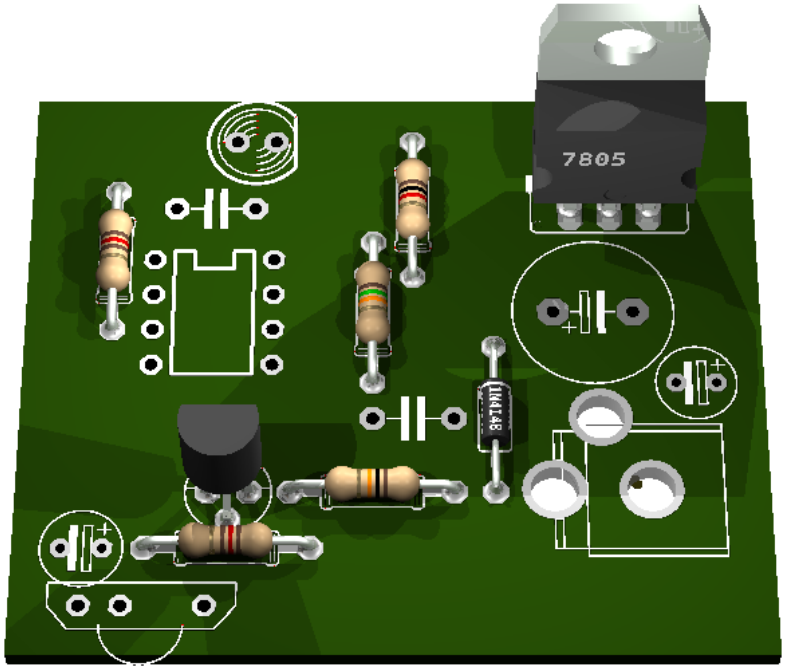
11.3 Partlist

Table 11.1:

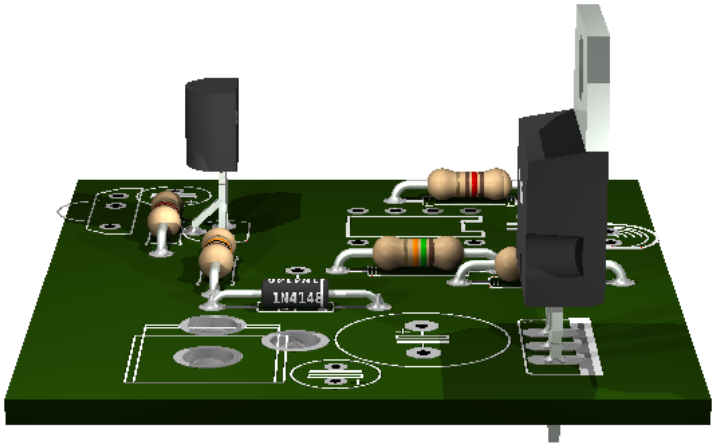
part	value	position
C1	47u 16V	(48.26 34.29)
C2	10u	(10.16 22.86)
C3	1n	(29.21 31.75)
C4	10n	(16.51 46.99)
C5	100u 25V	(40.64 39.37)
D2	1N4148	(34.29 31.75)
D3	IR	(19.05 52.07)
IC1	7805	(41.91 49.53)
IC2	555	(16.51 39.37)
R1	120	(17.78 22.86)
R2	15k	(26.67 39.37)
R3	10k	(26.67 26.67)
R4	120	(10.16 43.18)
R5	1k	(29.21 46.99)
T1	BC547	(17.78 26.67)
U$1	TSOP	(11.43 19.05)
U$2	DC ALJ	(44.45 27.94)

11.4 3D view

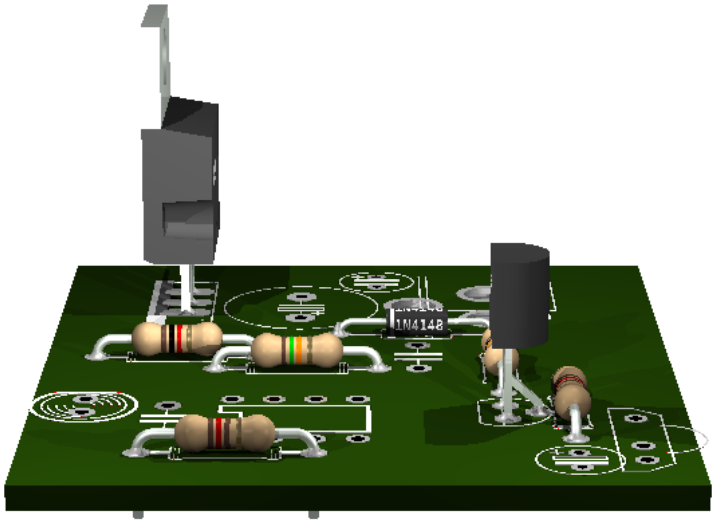
11.4.1 Front



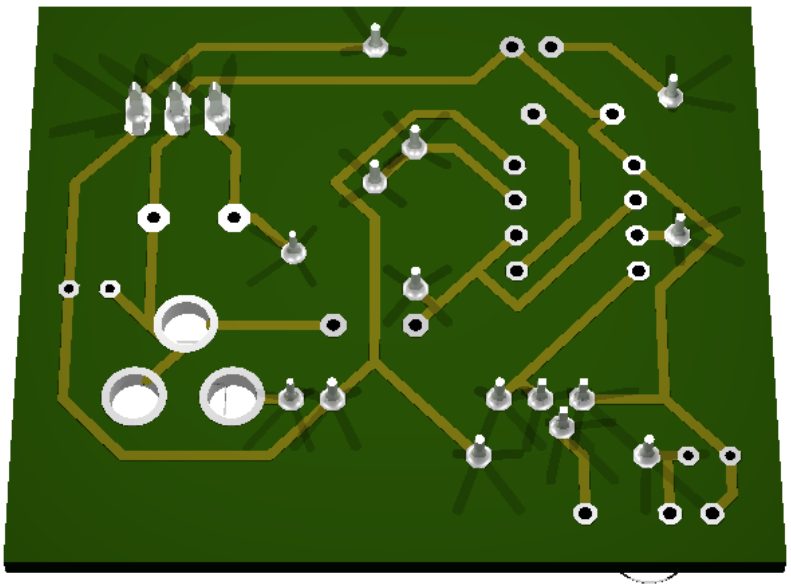
11.4.2 Right side



11.4.3 Left side



11.4.4 Bottom



CHAPTER

TWELVE

IRDA PORT

Status: OK

12.1 Schematic

source: http://www.infrarotport.de/ir-modul.HTML

12.2 Board

source: http://www.infrarotport.de/platine.HTML

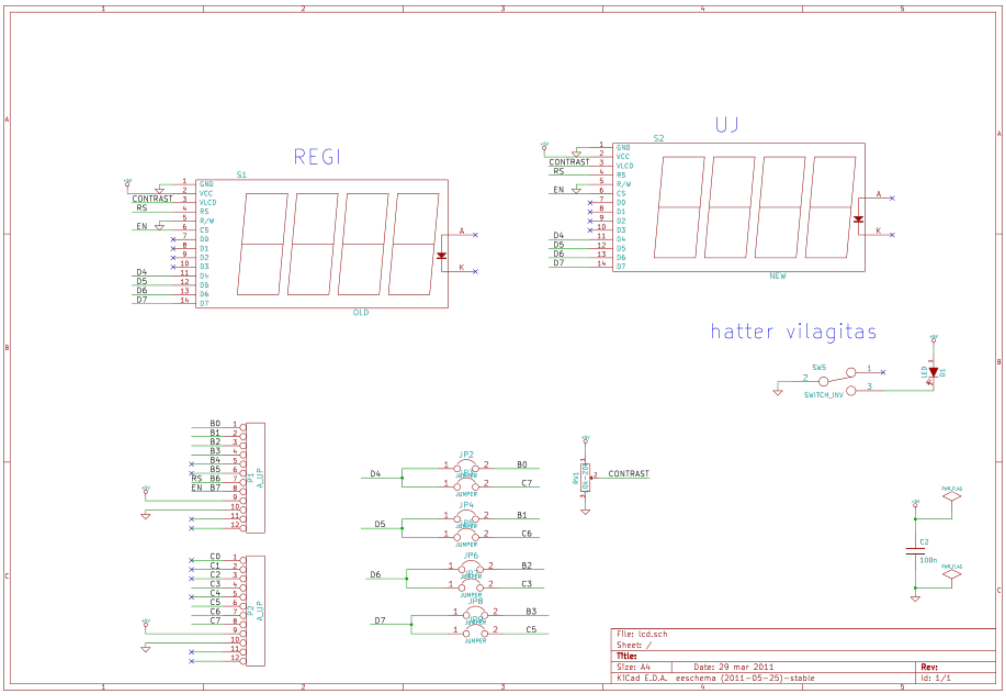
CHAPTER

THIRTEEN

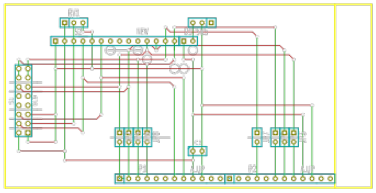
LCD MODULE

Status: OK

13.1 Schematic



13.2 Board



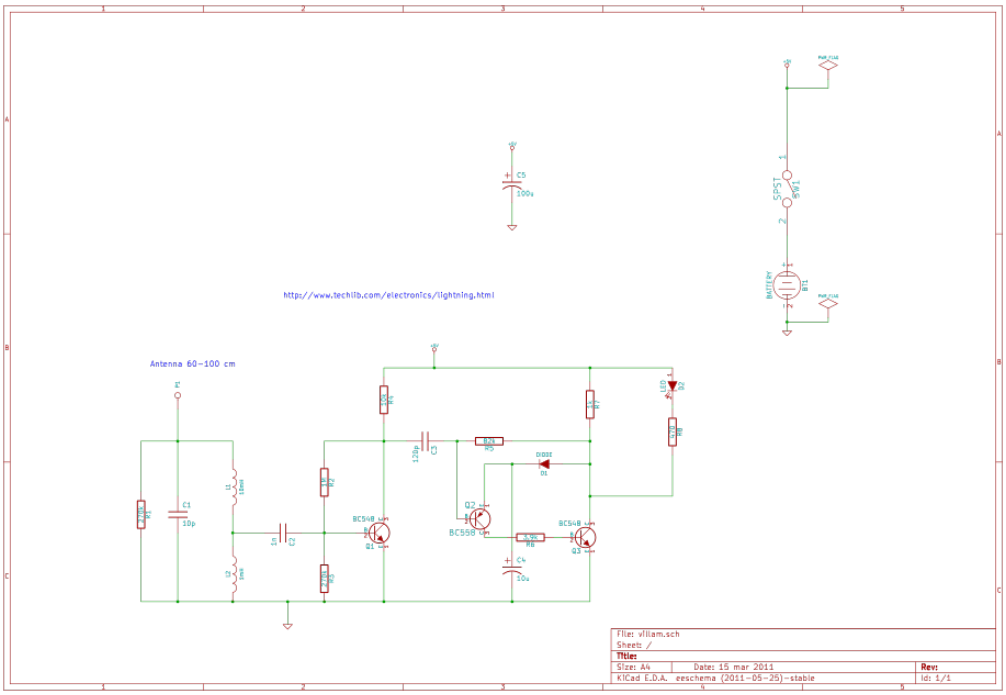
CHAPTER

FOURTEEN

LIGHNING DETECTOR

Status: OK

14.1 Schematic



14.2 Sources

based on: http://www.techlib.com/electronics/lightning.html

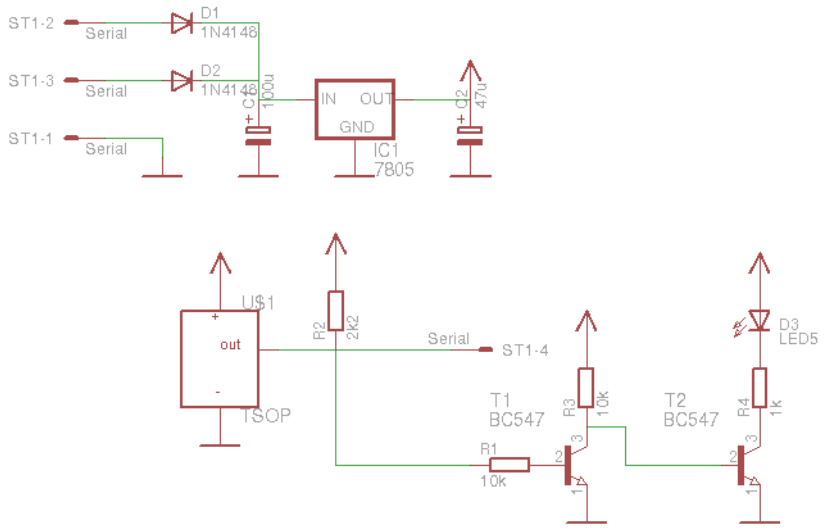
CHAPTER

FIFTEEN

LIRC SERIAL RECEIVER

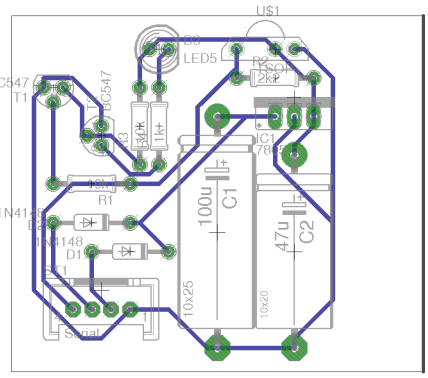
Status: OK

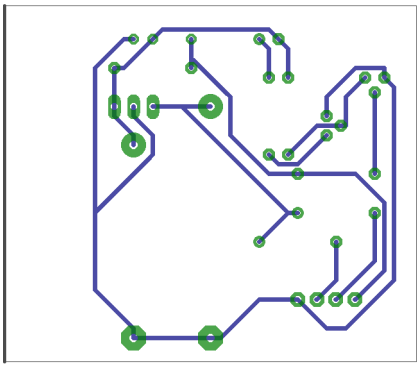
15.1 Schematic

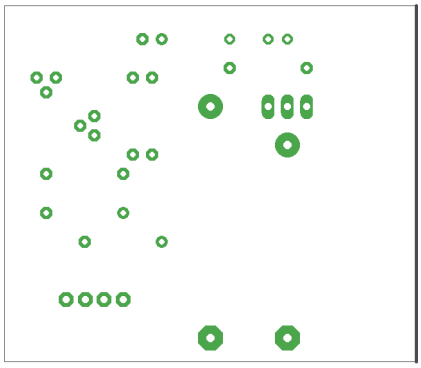


15.2 Board

Normal, bottom mirrored, wires only:







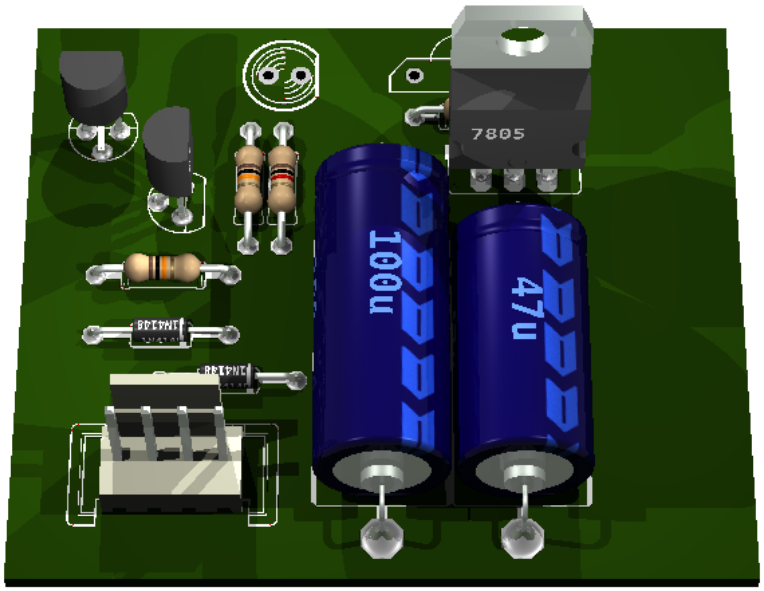
15.3 Partlist

Table 15.1:

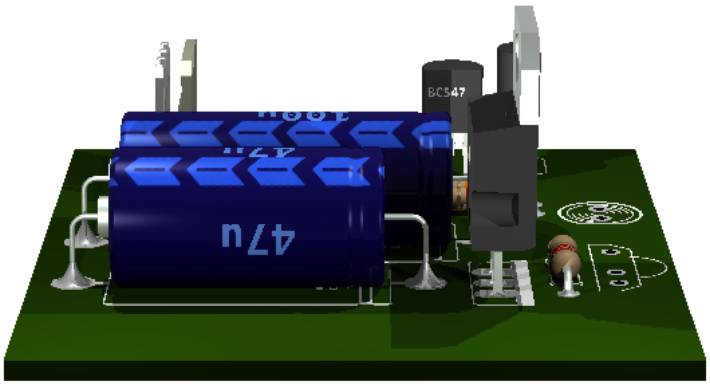
part	value	position
C1	100u	(1.2 1.85)
C2	47u	(1.6 1.75)
D1	1N4148	(0.75 1.75)
D2	1N4148	(0.55 1.9)
D3	LED5	(0.9 2.8)
IC1	7805	(1.6 2.55)
R1	10k	(0.55 2.1)
R2	2k2	(1.5 2.65)
R3	10k	(0.8 2.4)
R4	1k	(0.9 2.4)
ST1	Serial	(0.6 1.55)
T1	BC547	(0.35 2.6)
T2	BC547	(0.6 2.35)
U$1	TSOP	(1.5 2.8)

15.4 3D view

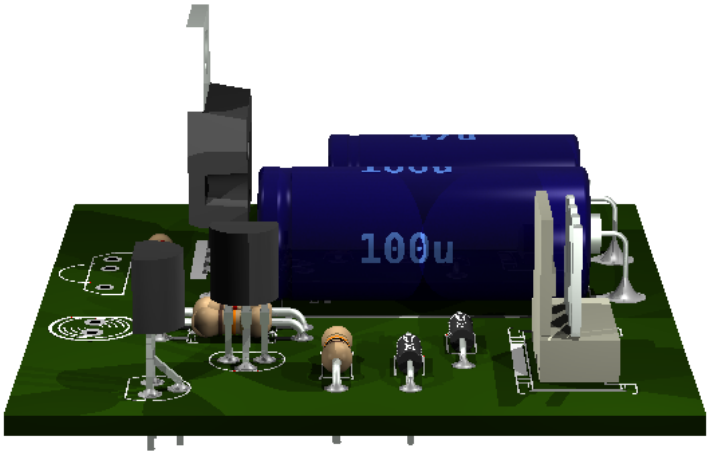
15.4.1 Front



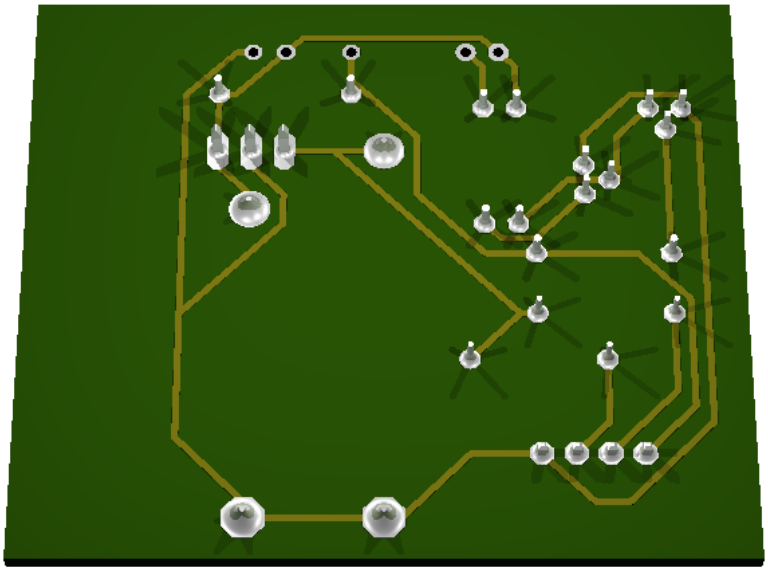
15.4.2 Right side



15.4.3 Left side



15.4.4 Bottom



CHAPTER

SIXTEEN

LOGIC PROBE

Status: OK

It is used for testing logic levels.

Vcc=3-15V

Source: http://www.sentex.net/~mec1995/circ/probe2.htm

Differences:

- R2=4M7
- R3=1k
- R4=1k
- R5=1k
- LED + 1k for Power

CHAPTER

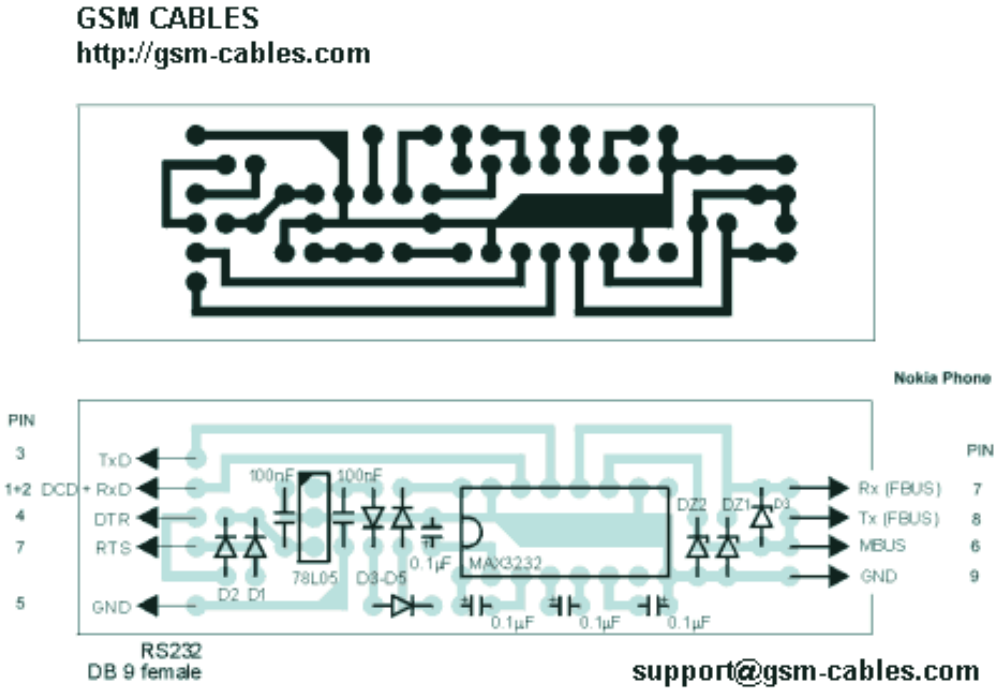
SEVENTEEN

NOKIA ADAPTER

Status: OK

17.1 Sources

original design



CHAPTER

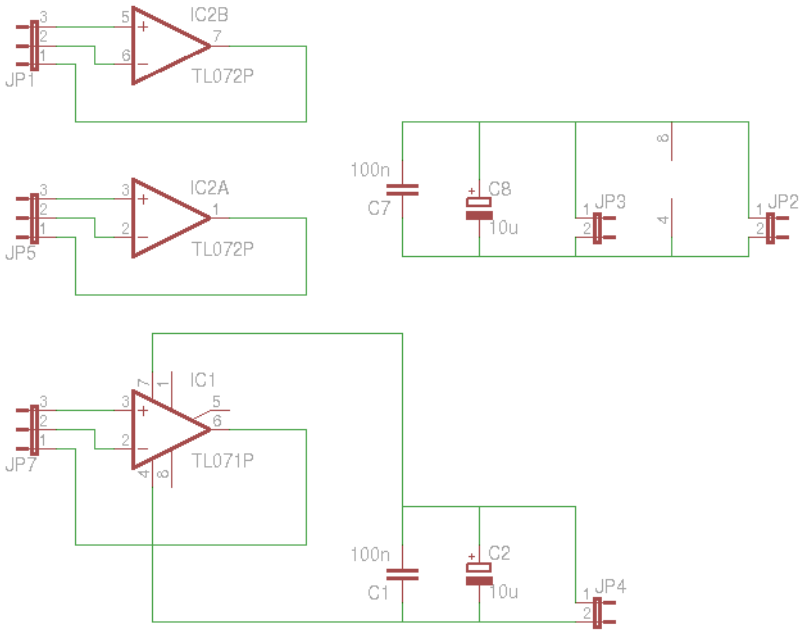
EIGHTEEN

OP-AMP MODULE

Status: OK

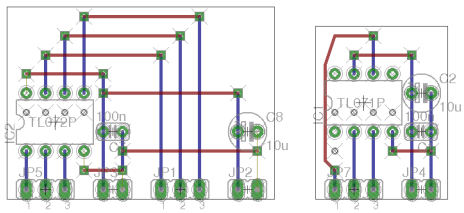
It is used for op-amps in breadboard.

18.1 Schematic

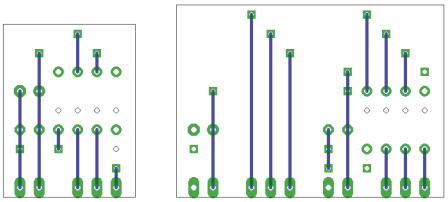


18.2 Board

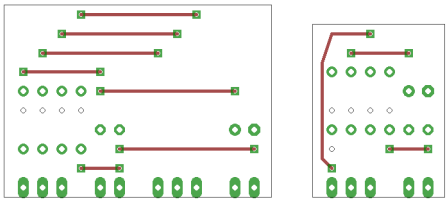
top



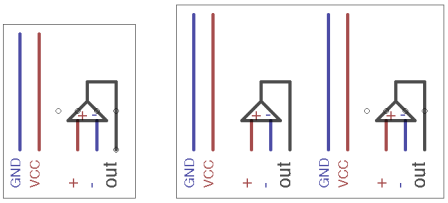
bottom mirrored



wires only



document



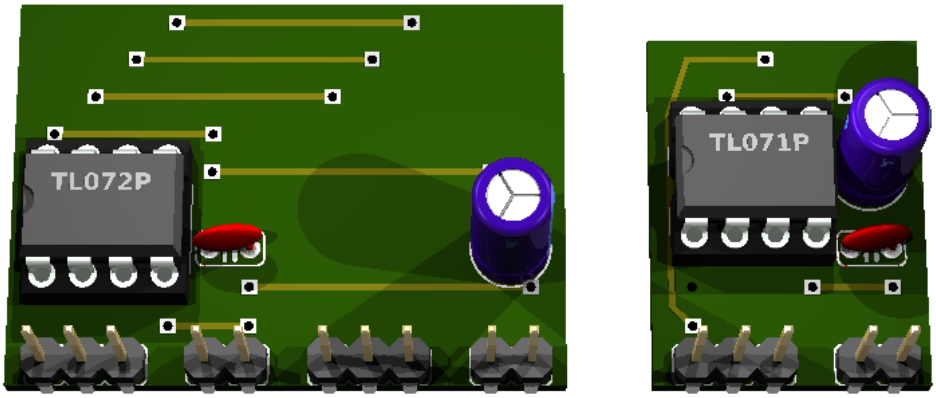
18.3 Partlist

Table 18.1:

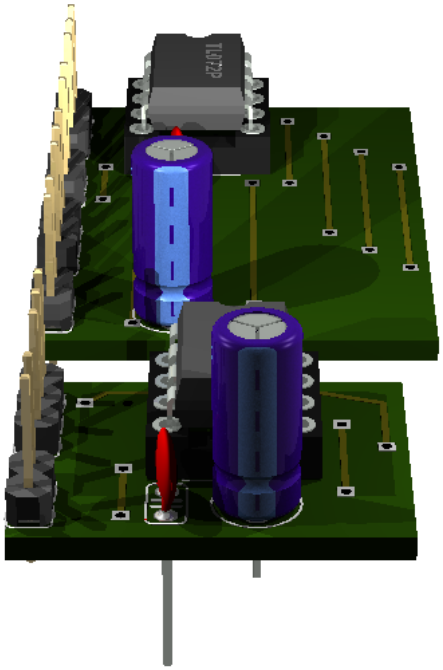
part	value	position
C1	100n	(2.35 0.4)
C2	10u	(2.35 0.6)
C7	100n	(0.75 0.4)
C8	10u	(1.45 0.4)
IC1	TL071P	(2.05 0.55)
IC2	TL072P	(0.45 0.45)
JP1		(1.1 0.1)
JP2		(1.45 0.1)
JP3		(0.75 0.1)
JP4		(2.35 0.1)
JP5		(0.4 0.1)
JP7		(2 0.1)

18.4 3D view

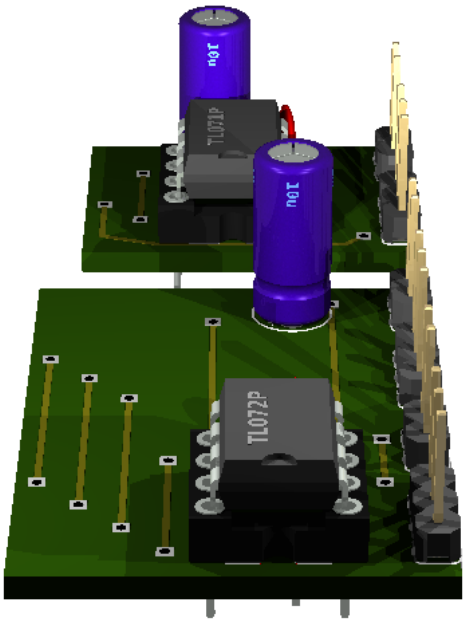
18.4.1 Front



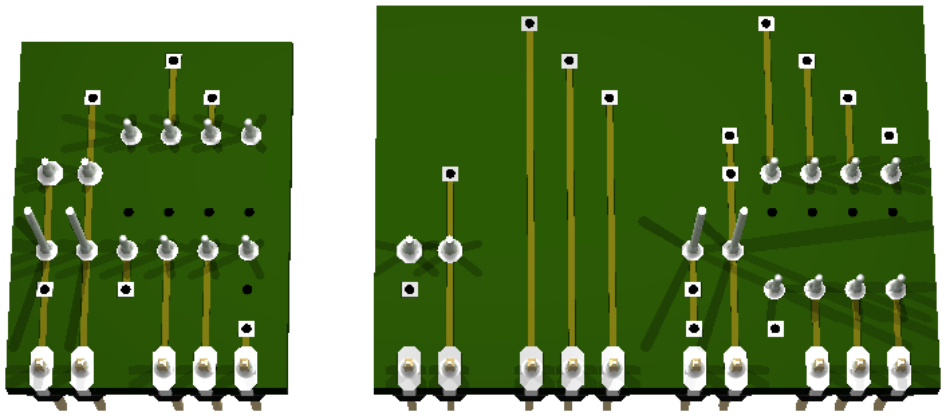
18.4.2 Right side



18.4.3 Left side



18.4.4 Bottom



CHAPTER

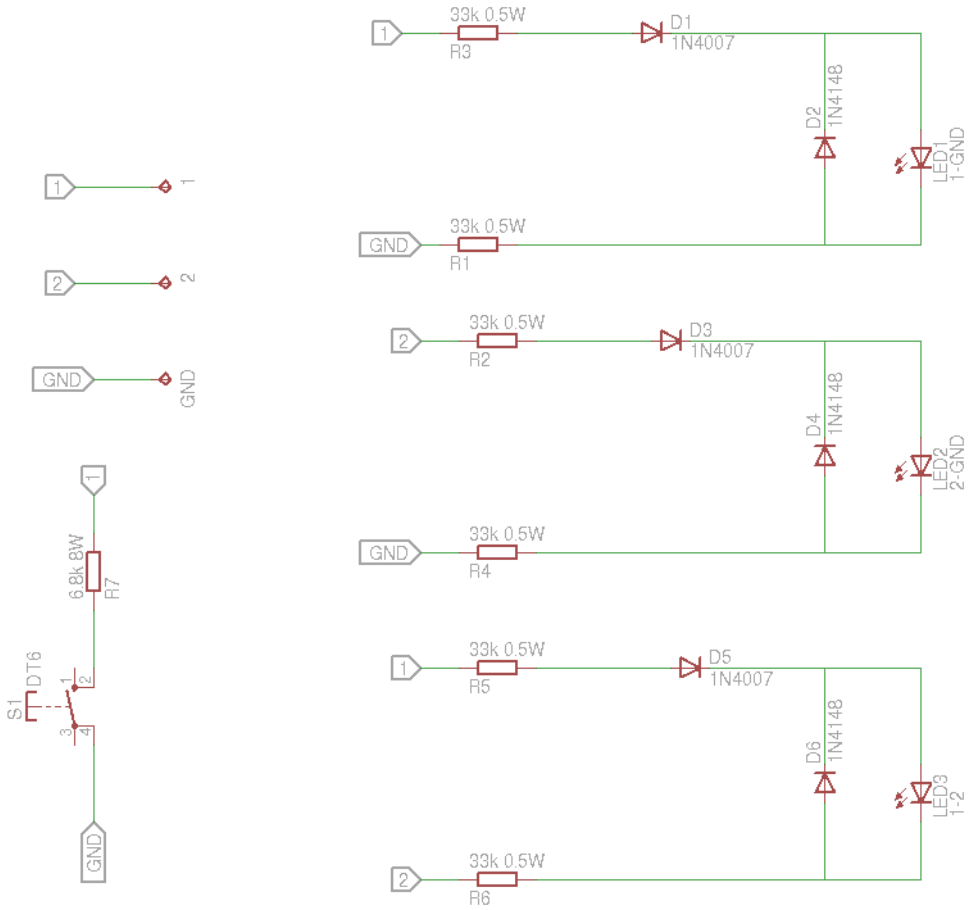
NINETEEN

OUTLET TESTER

Status: ??

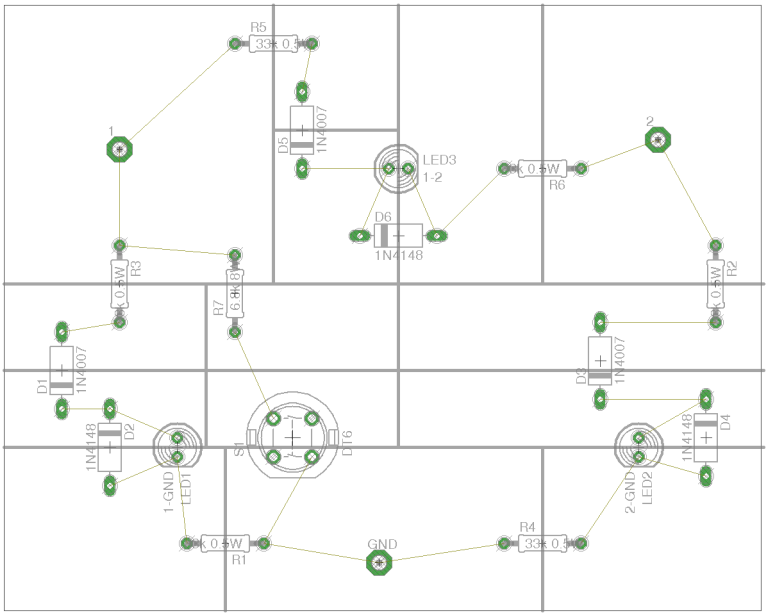
This device is used to verify that an 230V AC wall outlet is wired properly.

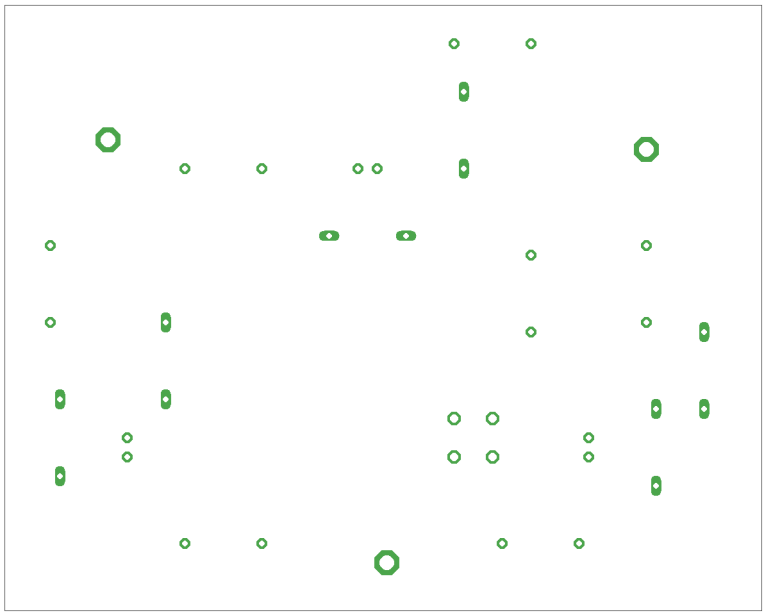
19.1 Schematic

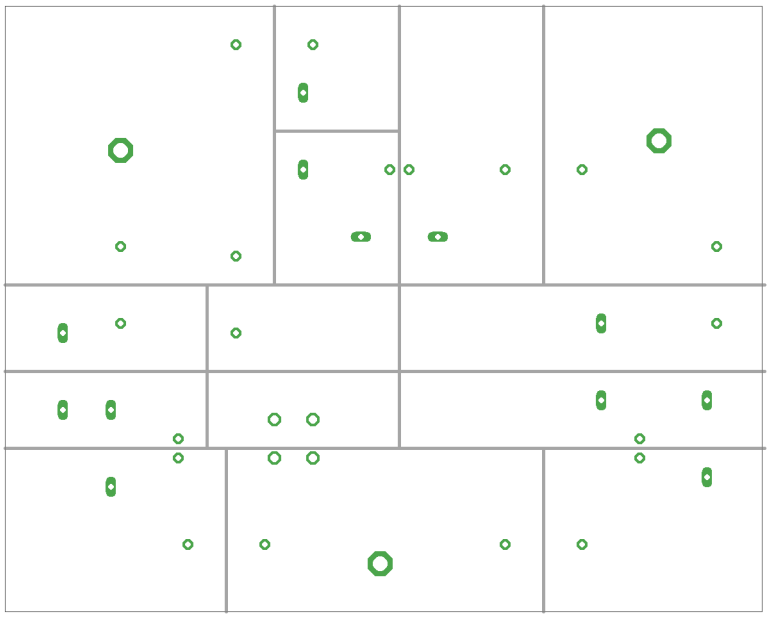


19.2 Board

Normal, bottom mirrored, wires only:







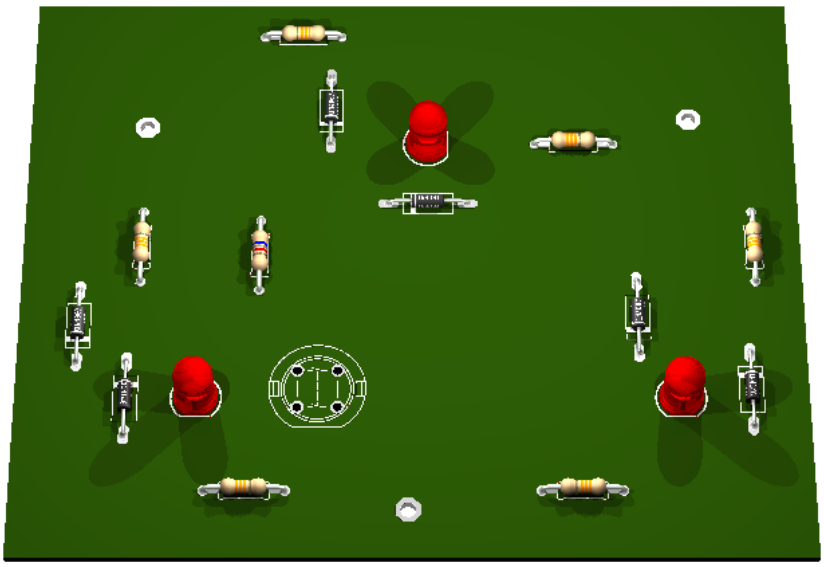
19.3 Partlist

Table 19.1:

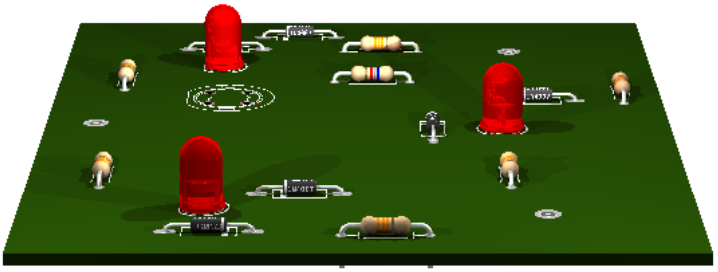
part	value	position
1	TPPAD1-20	(0.6 2.4)
2	TPPAD1-20	(3.4 2.45)
D1	1N4007	(0.3 1.25)
D2	1N4148	(0.55 0.85)
D3	1N4007	(3.1 1.3)
D4	1N4148	(3.65 0.9)
D5	1N4007	(1.55 2.5)
D6	1N4148	(2.05 1.95)
GND	TPPAD1-20	(1.95 0.25)
LED1	1-GND	(0.9 0.85)
LED2	2-GND	(3.3 0.85)
LED3	1-2	(2.05 2.3)
R1	33k 0.5W	(1.15 0.35)
R2	33k 0.5W	(3.7 1.7)
R3	33k 0.5W	(0.6 1.7)
R4	33k 0.5W	(2.8 0.35)
R5	33k 0.5W	(1.4 2.95)
R6	33k 0.5W	(2.8 2.3)
R7	6.8k 8W	(1.2 1.65)
S1	DT6	(1.5 0.9)

19.4 3D view

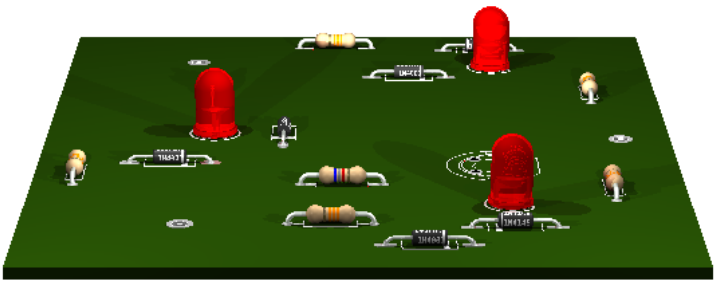
19.4.1 Front



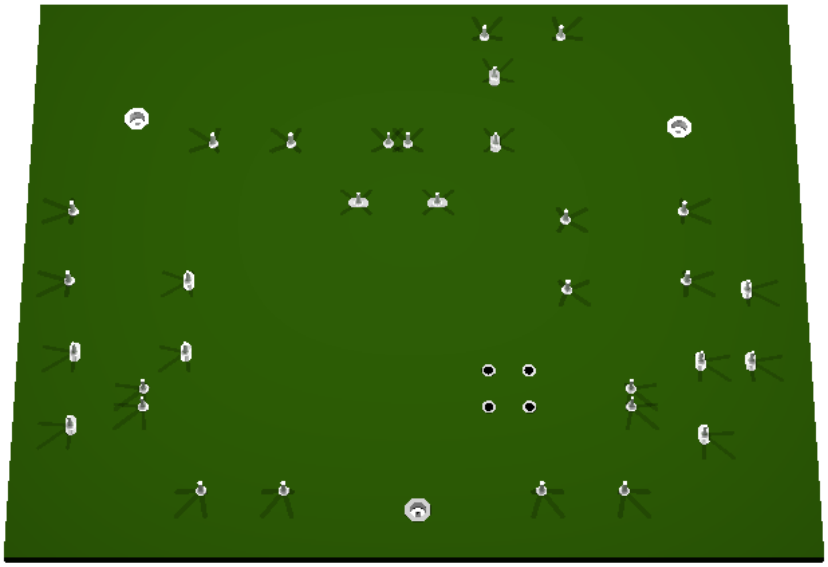
19.4.2 Right side



19.4.3 Left side



19.4.4 Bottom



19.5 Sources

http://en.wikipedia.org/wiki/Receptacle_tester

CHAPTER

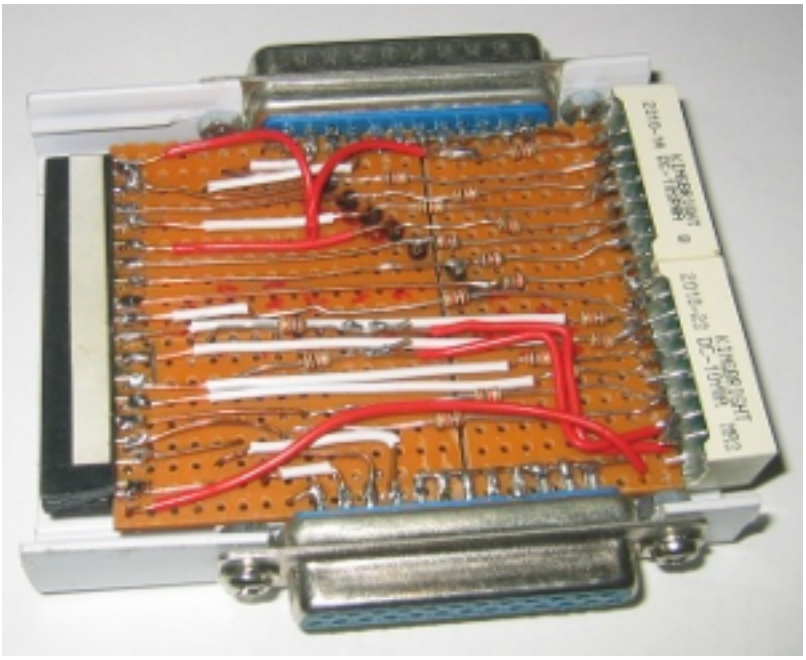
TWENTY

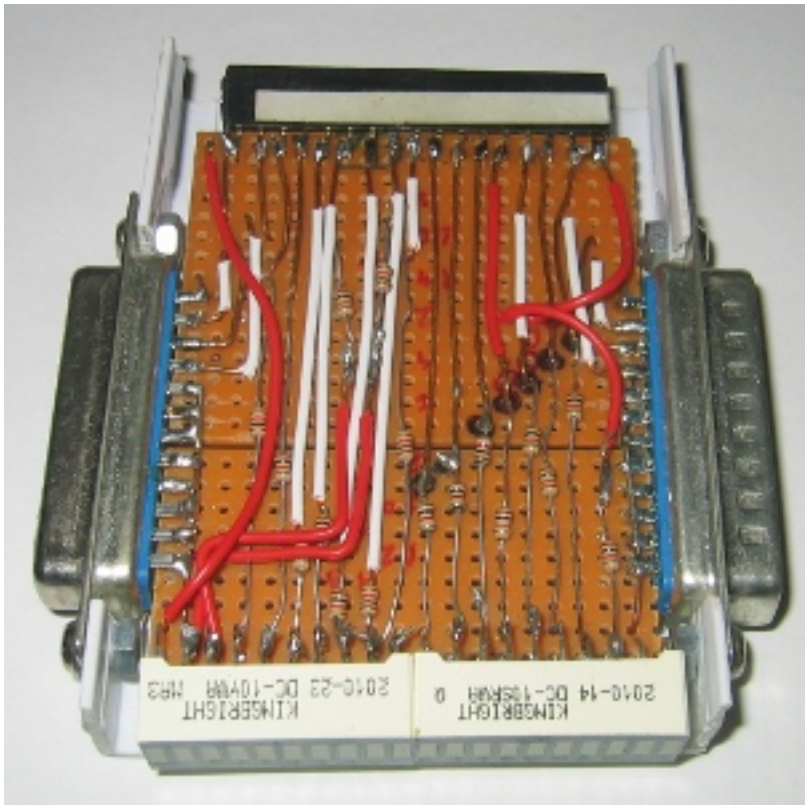
PARALLEL PORT MONITOR

Status: OK

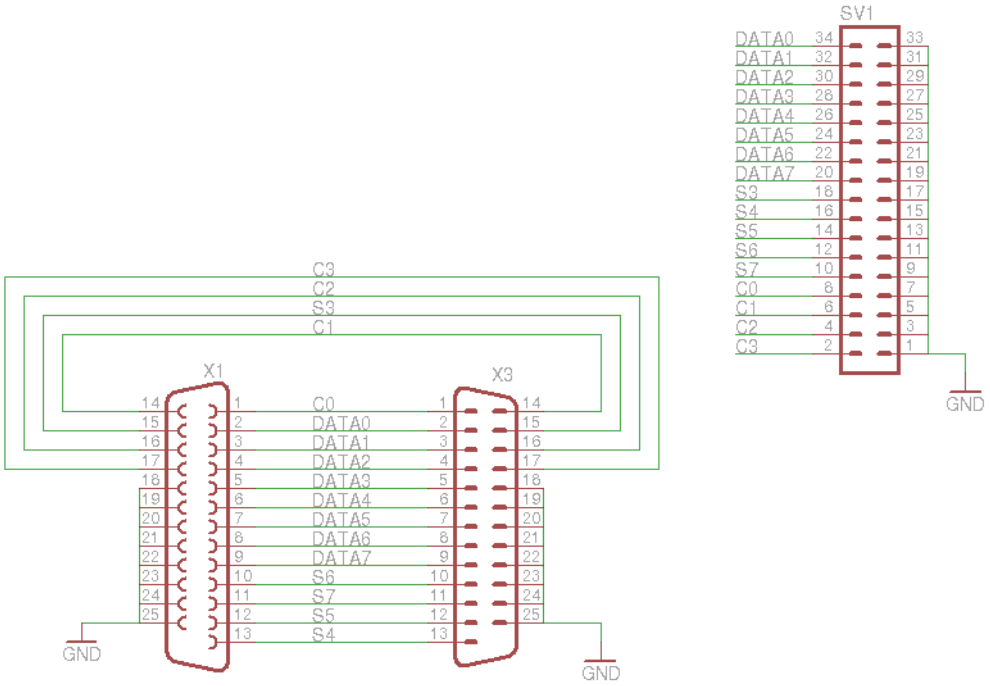
It is used for monitoring the parallel port signals.

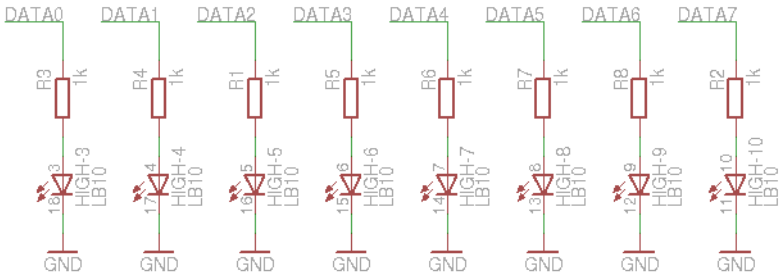
20.1 Images

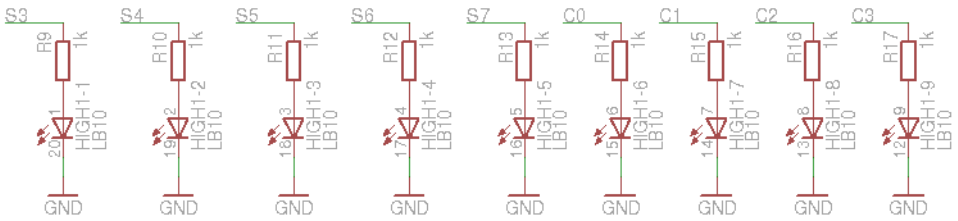




20.2 Schematic

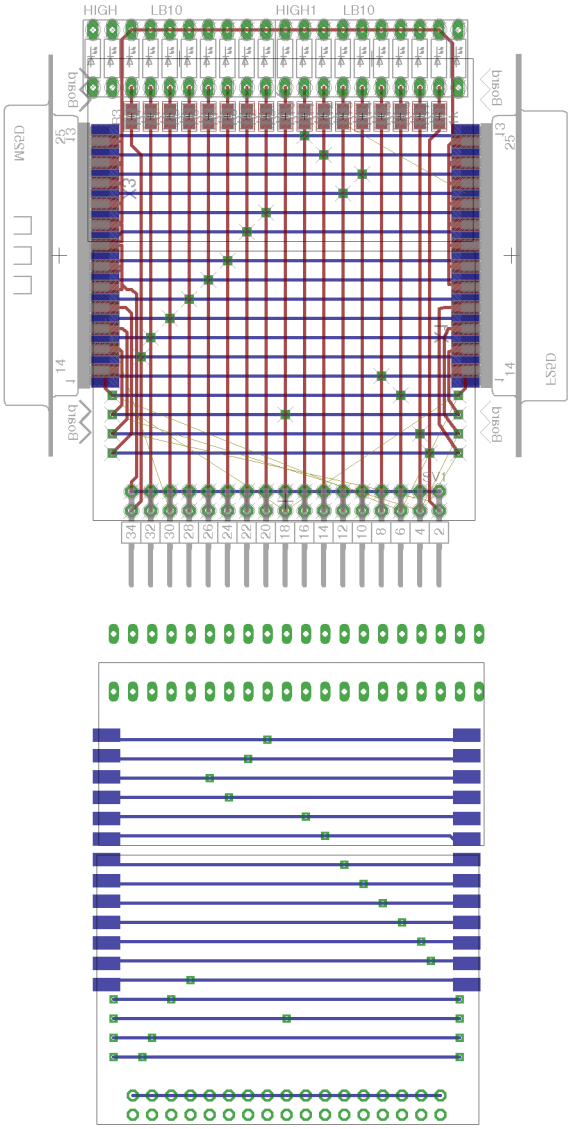


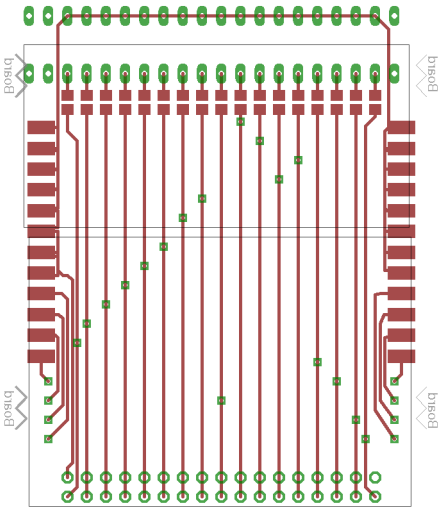




20.3 Board

Normal, bottom mirrored, wires only:





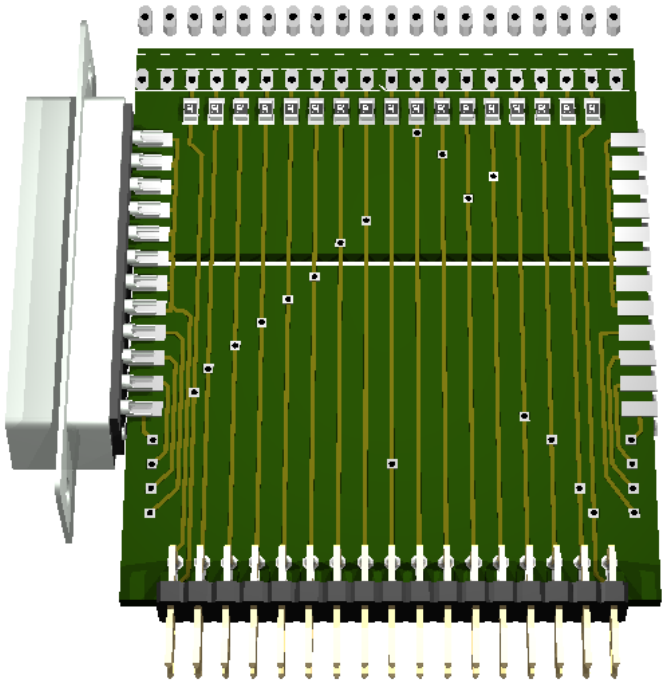
20.4 Partlist

Table 20.1:

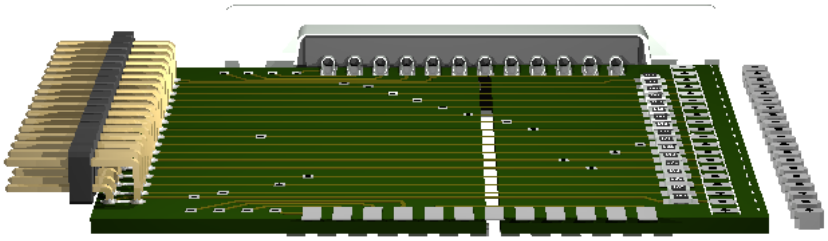
part	value	position
HIGH	LB10	(0.45 2.55)
HIGH1	LB10	(1.45 2.55)
R1	1k	(0.4 2.25)
R2	1k	(0.9 2.25)
R3	1k	(0.2 2.25)
R4	1k	(0.3 2.25)
R5	1k	(0.5 2.25)
R6	1k	(0.6 2.25)
R7	1k	(0.7 2.25)
R8	1k	(0.8 2.25)
R9	1k	(1 2.25)
R10	1k	(1.1 2.25)
R11	1k	(1.2 2.25)
R12	1k	(1.3 2.25)
R13	1k	(1.4 2.25)
R14	1k	(1.5 2.25)
R15	1k	(1.6 2.25)
R16	1k	(1.7 2.25)
R17	1k	(1.8 2.25)
SV1		(1 0.25)
X1		(2.175 1.525)
X3		(-0.175 1.525)

20.5 3D view

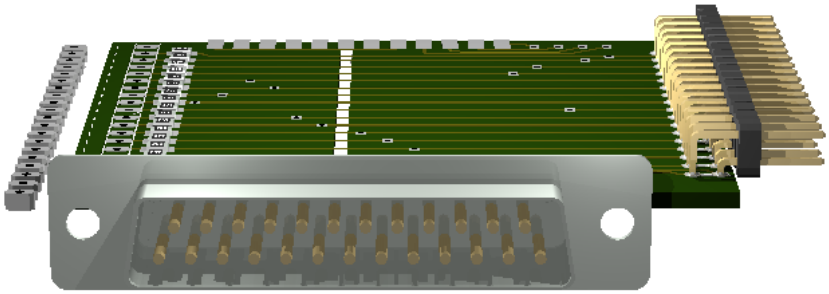
20.5.1 Front



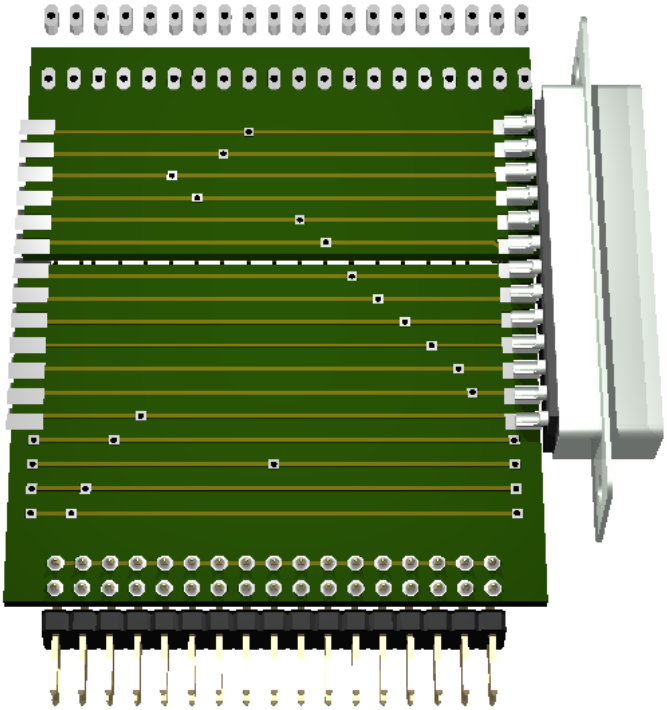
20.5.2 Right side



20.5.3 Left side



20.5.4 Bottom



20.6 Sources

original idea

CHAPTER

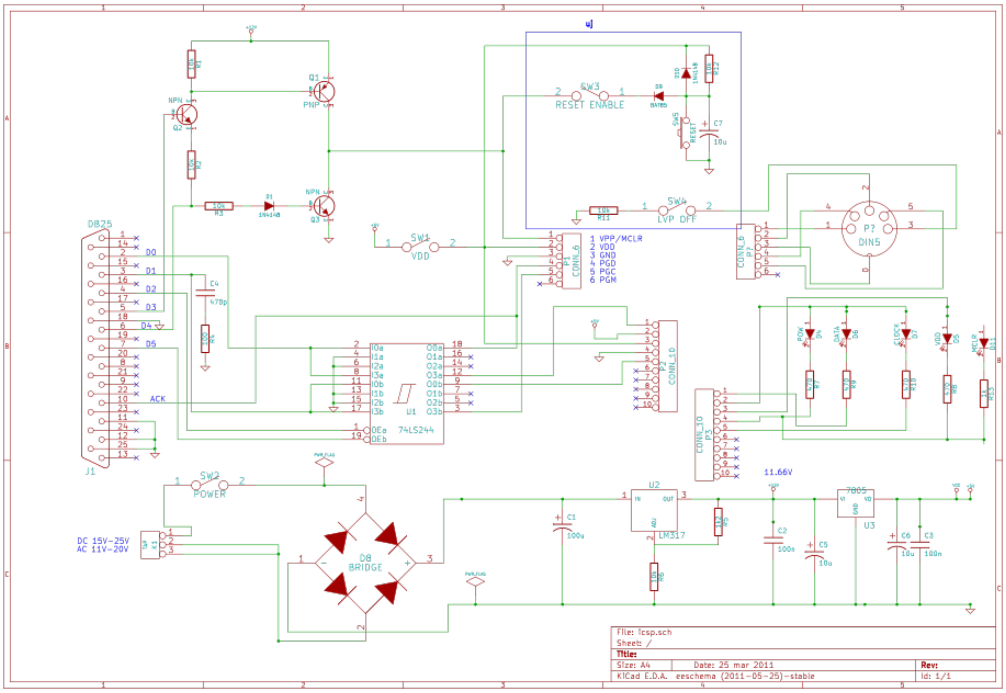
TWENTYONE

PIC ICSP PROGRAMMER

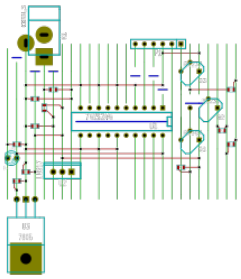
Status: OK

It is used for programming PIC using the parallel port.

21.1 Schematic



21.2 Board



21.3 Sources

original design: http://www.best-microcontroller-projects.com/pic-programmer-circuit.html

which is based on: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en011060

CHAPTER

TWENTYTWO

PONG

Status: OK

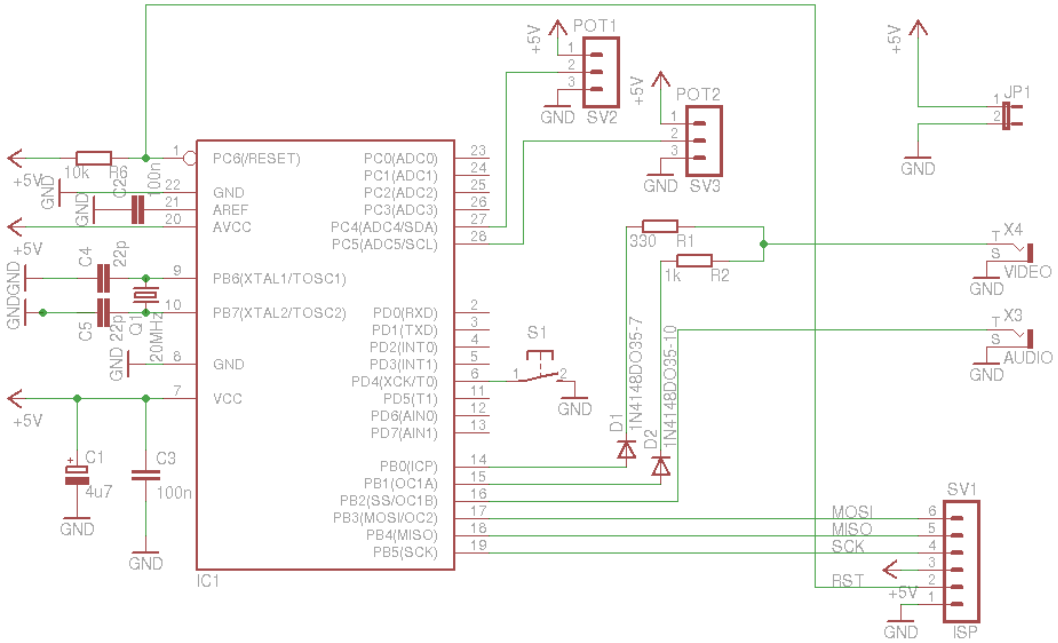
Arduino based pong console

Pong firmware: https://github.com/ponty/arduino-pong

program config:

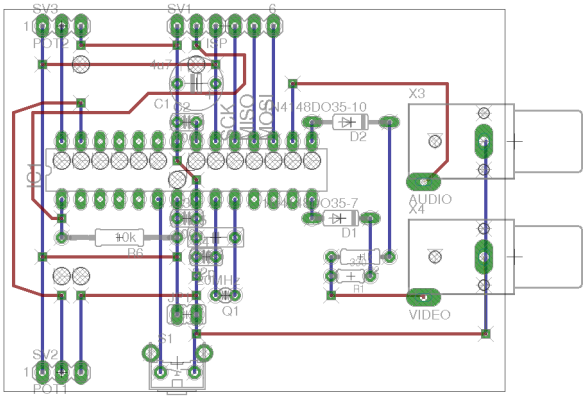
```
const int pin_wheel1 = 4; // analog
const int pin_wheel2 = 5; // analog
const int pin_button1 = 4;
const int pin_audio = 10;
```

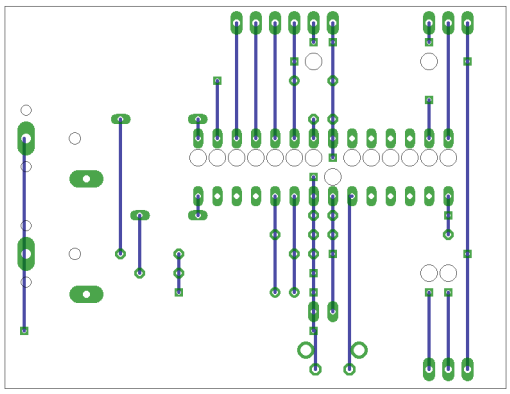
22.1 Schematic

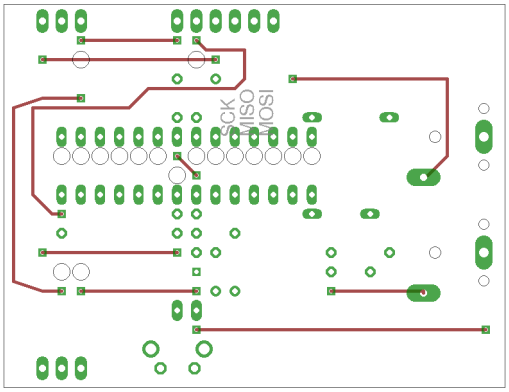


22.2 Board

Normal, bottom mirrored, wires only:







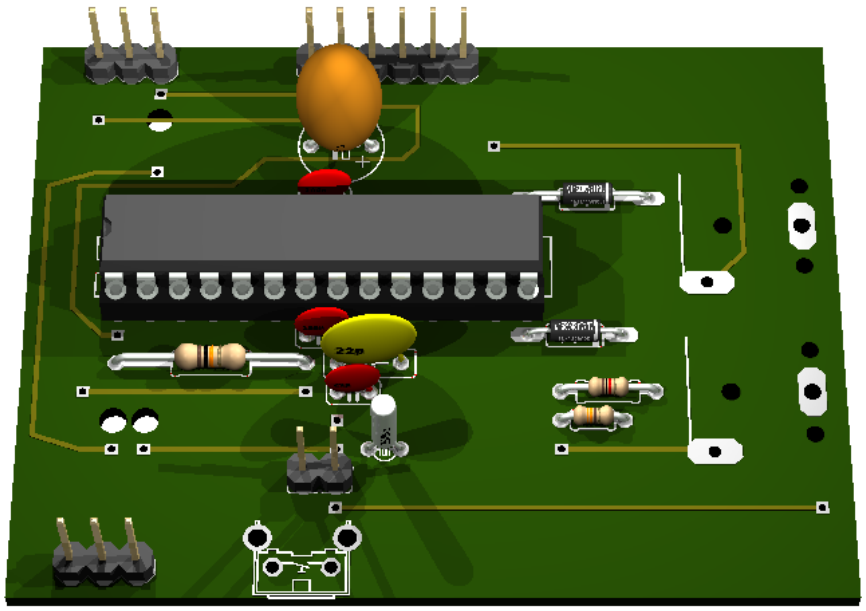
22.3 Partlist

Table 22.1:

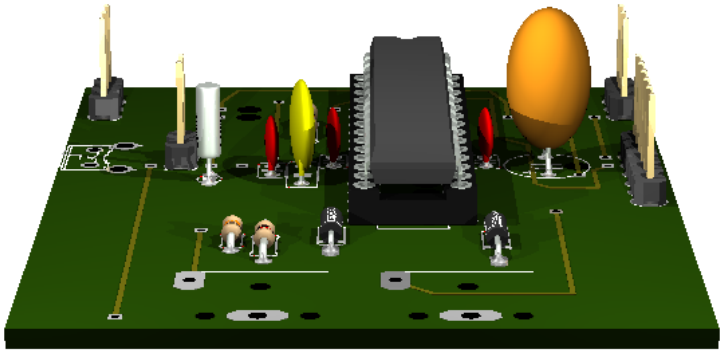
part	value	position
C1	4u7	(1.1 1.7)
C2	100n	(1.05 1.5)
C3	100n	(1.05 1)
C4	22p	(1.15 0.8)
C5	22p	(1.2 0.9)
D1	1N4148DO35-7	(1.85 1)
D2	1N4148DO35-10	(1.9 1.5)
IC1		(1.05 1.25)
JP1		(1.05 0.5)
Q1	20MHz	(1.25 0.6)
R1	330	(1.9 0.7)
R2	1k	(1.95 0.8)
R6	10k	(0.7 0.9)
S1		(1 0.25)
SV1	ISP	(1.25 2)
SV2	POT1	(0.4 0.2)
SV3	POT2	(0.4 2)
X3	AUDIO	(2.75 1.4)
X4	VIDEO	(2.75 0.8)

22.4 3D view

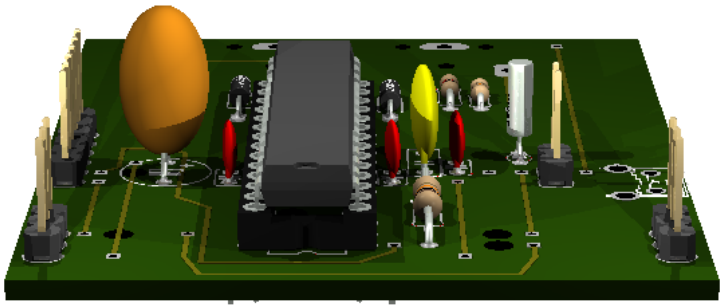
22.4.1 Front



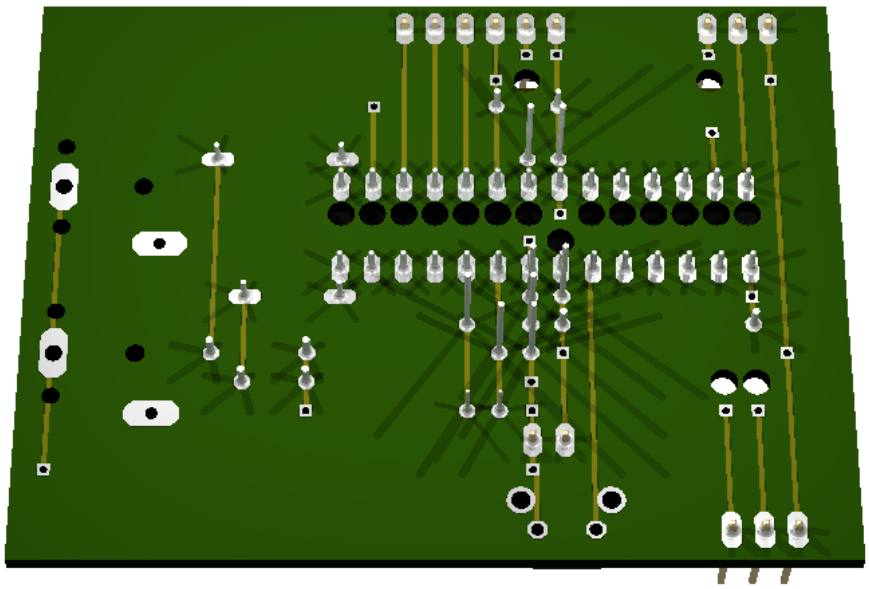
22.4.2 Right side



22.4.3 Left side



22.4.4 Bottom



22.5 Sources

similar projects:

- http://www.instructables.com/id/Ardu-pong-the-Arduino-based-pong-console/
- http://www.instructables.com/id/2-player-Pong-using-Arduino/
- http://blog.makezine.com/2007/08/22/arduino-pong/

CHAPTER

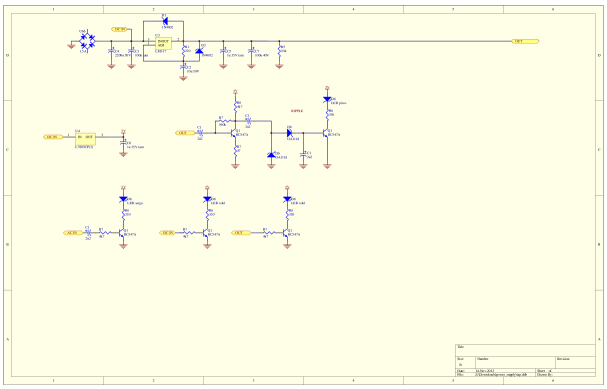
TWENTYTHREE

POWER SUPPLY

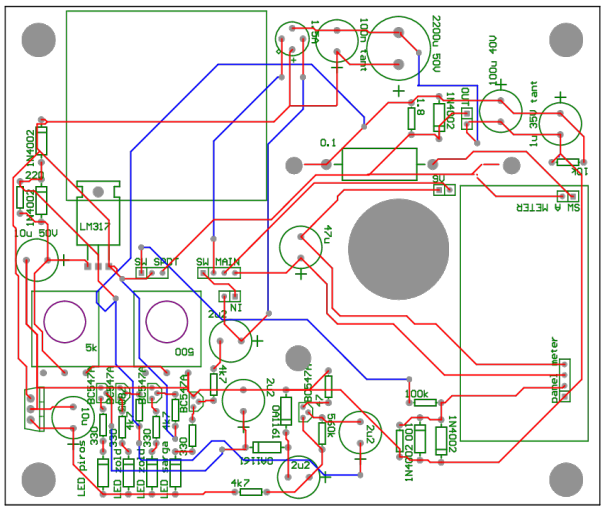
Status: OK

LM317 based power supply.

23.1 Schematic



23.2 Board



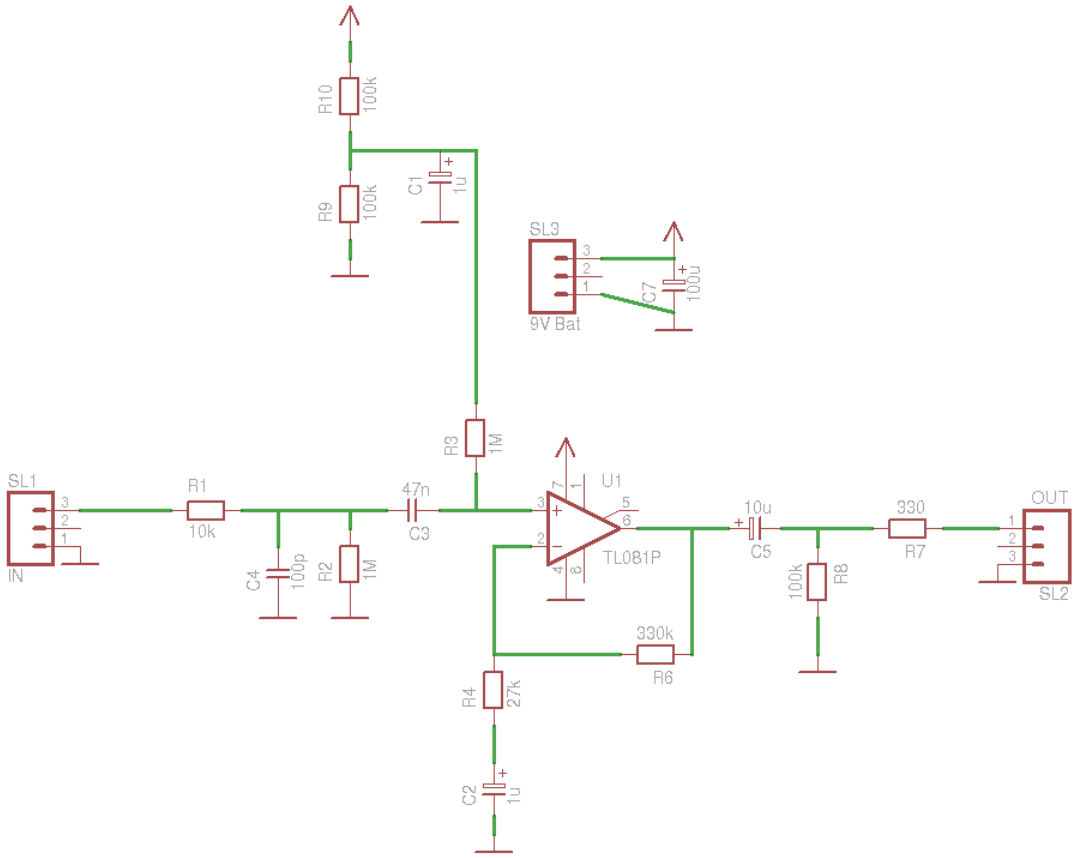
CHAPTER

TWENTYFOUR

PREAMPLIFIER

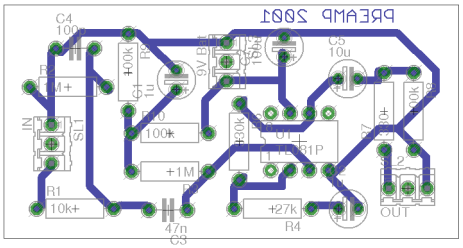
Status:

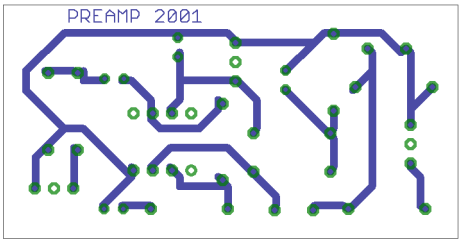
24.1 Schematic

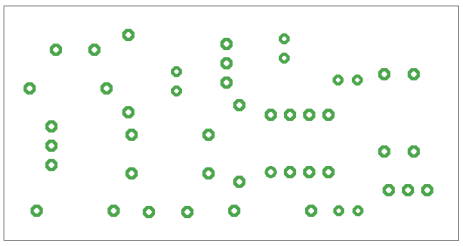


24.2 Board

Normal, bottom mirrored, wires only:







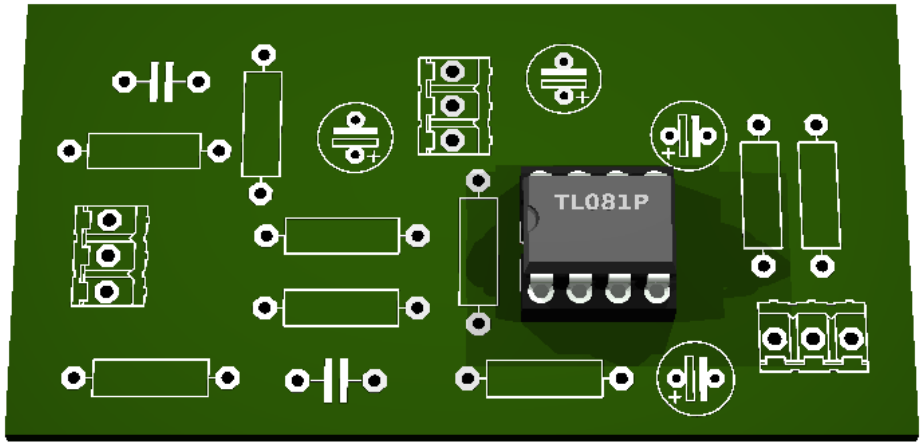
24.3 Partlist

Table 24.1:

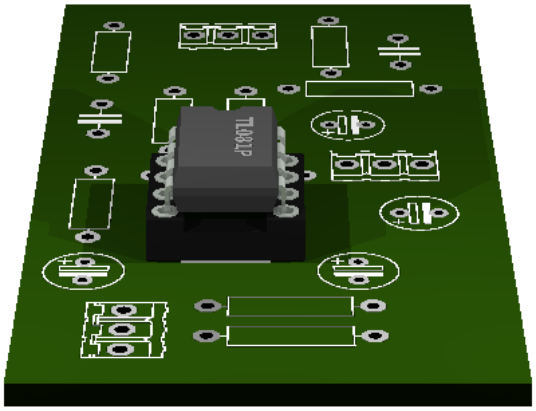
part	value	position
C1	1u	(20.28 19.21)
C2	1u	(41.69 3.37)
C3	47n	(19.21 3.21)
C4	100p	(6.97 24.58)
C5	10u	(41.66 20.61)
C7	100u	(34.5 23.53)
R1	10k	(6.97 3.42)
R2	1M	(5.97 19.5)
R3	1M	(19.48 8.29)
R4	27k	(32.99 3.37)
R6	330k	(28.64 12.26)
R7	330	(47.72 16.31)
R8	100k	(51.64 16.31)
R9	100k	(13.93 21.48)
R10	100k	(19.48 13.37)
SL1	IN	(3.81 11.96)
SL2	OUT	(50.82 6.12)
SL3	9V Bat	(26.88 22.8)
U1	TL081P	(36.53 12.26)

24.4 3D view

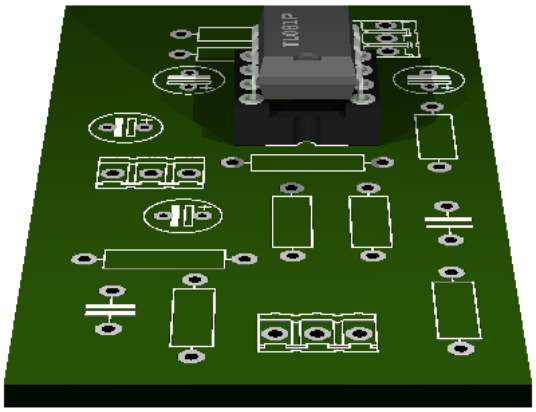
24.4.1 Front



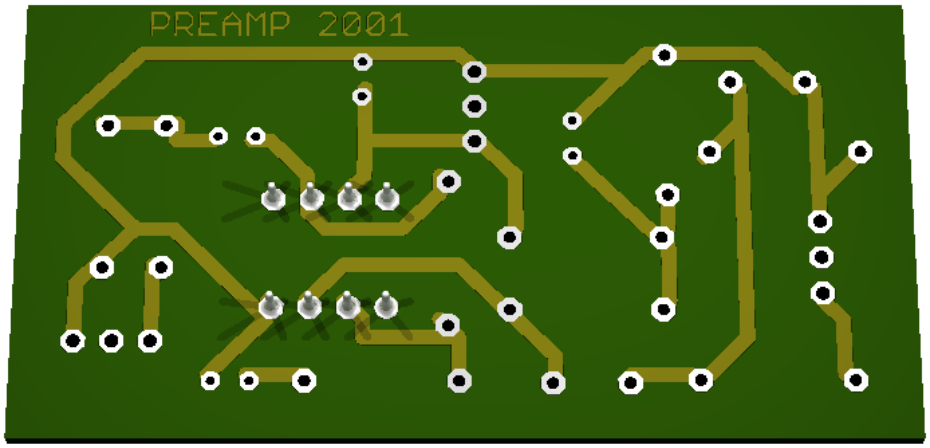
24.4.2 Right side



24.4.3 Left side



24.4.4 Bottom



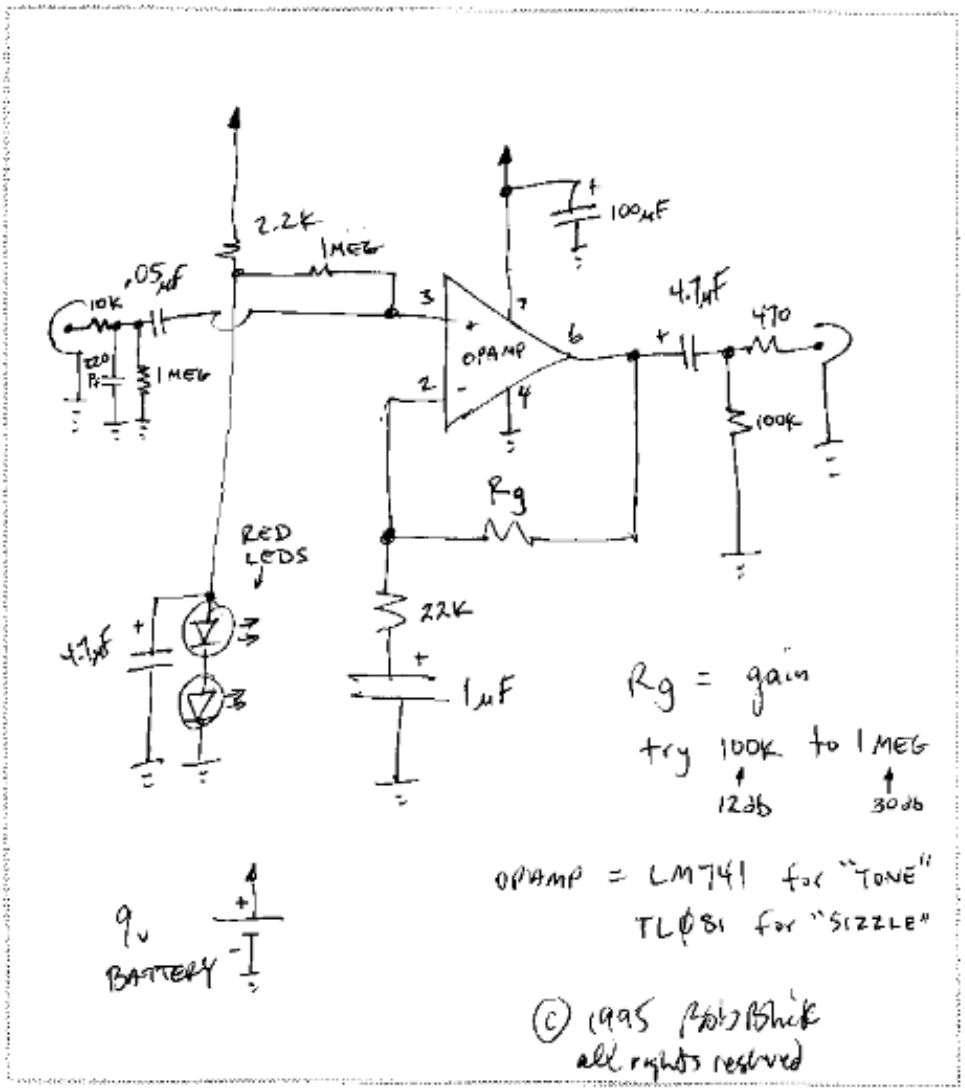
24.5 Sources

original design

BOB BLICK CALCULATION SHEET

BY _____ DATE 10/31/95 SUBJECT Guitar Preamp SHEET NO. ____ OF ____

CHKD _____ DATE _____ JOB NO. _____



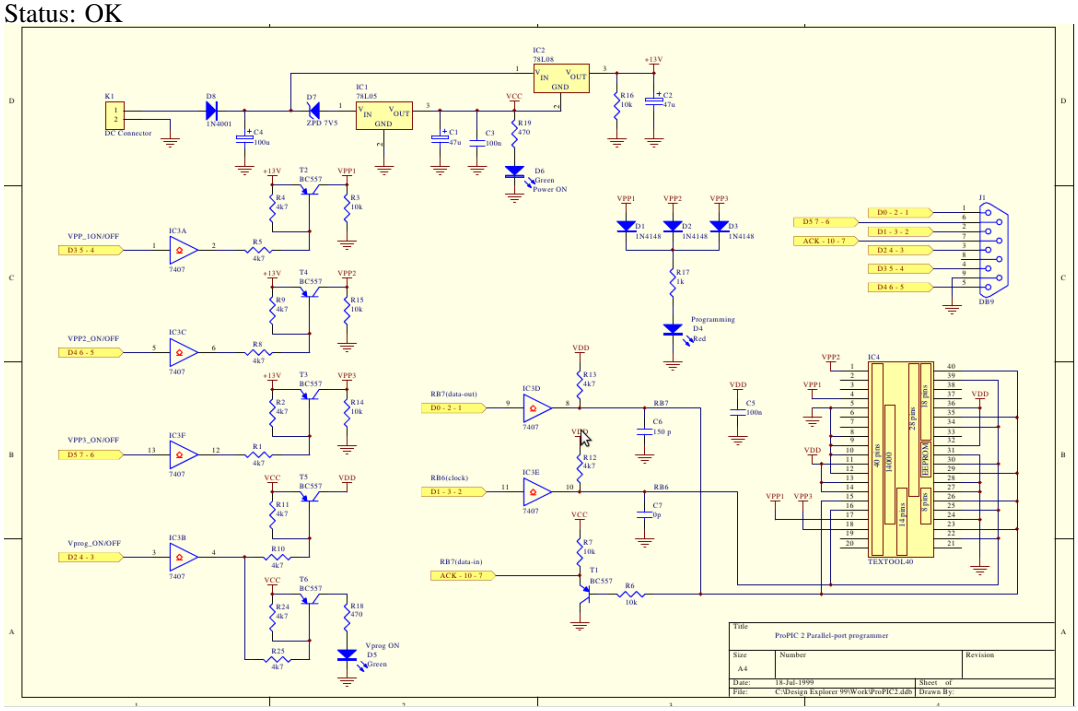
S ENG-7 REV. 6/91

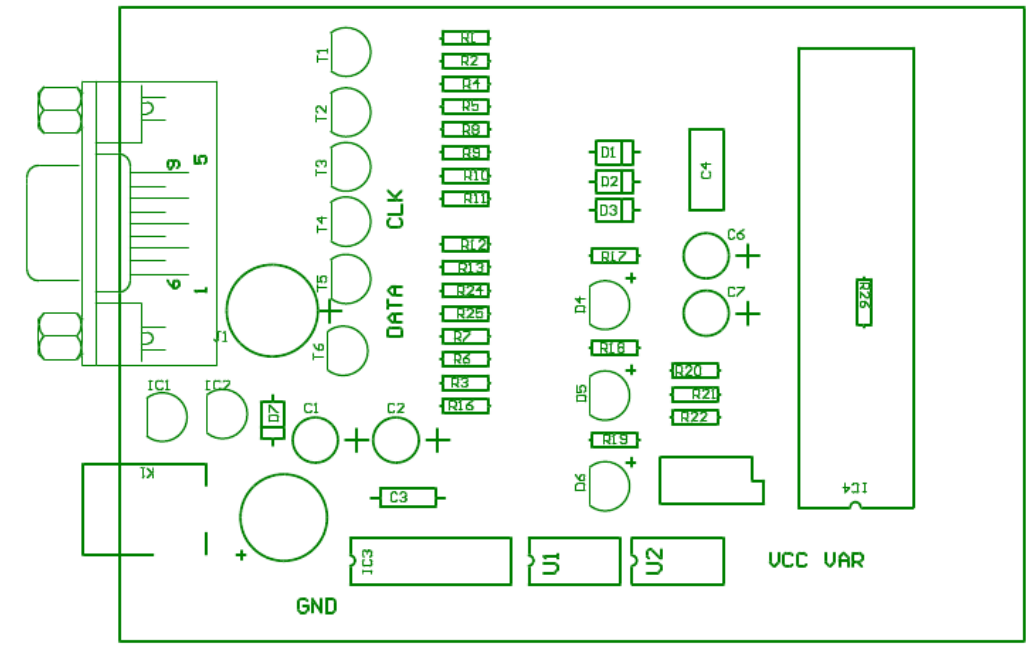
CHAPTER

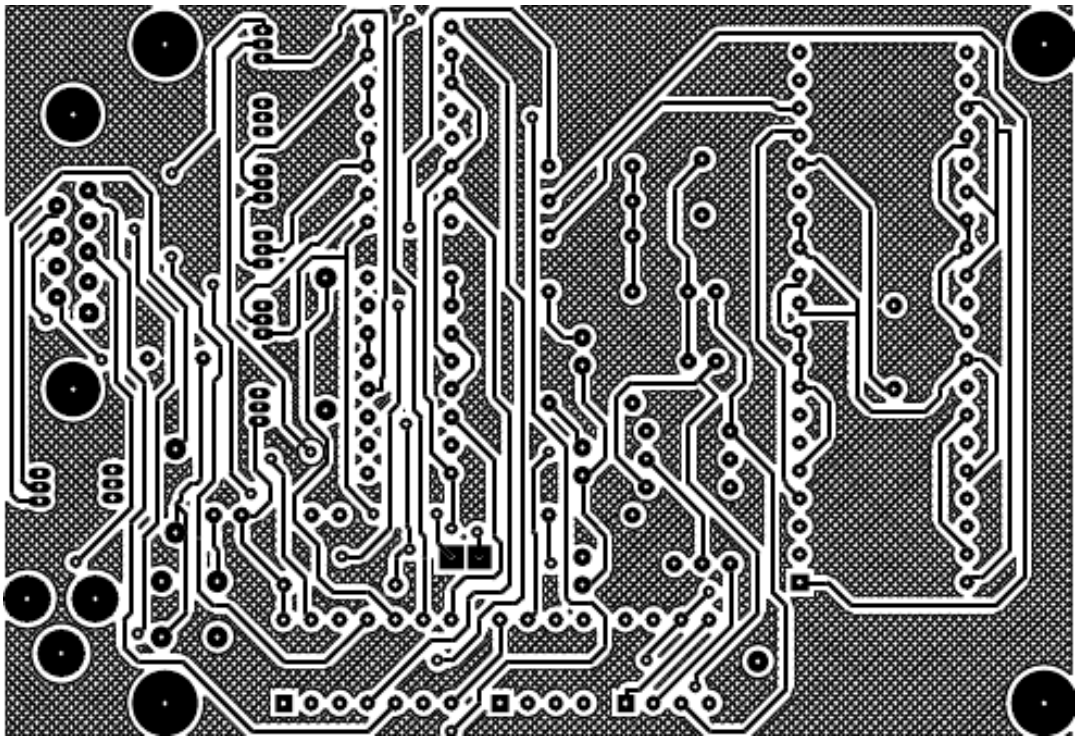
TWENTYFIVE

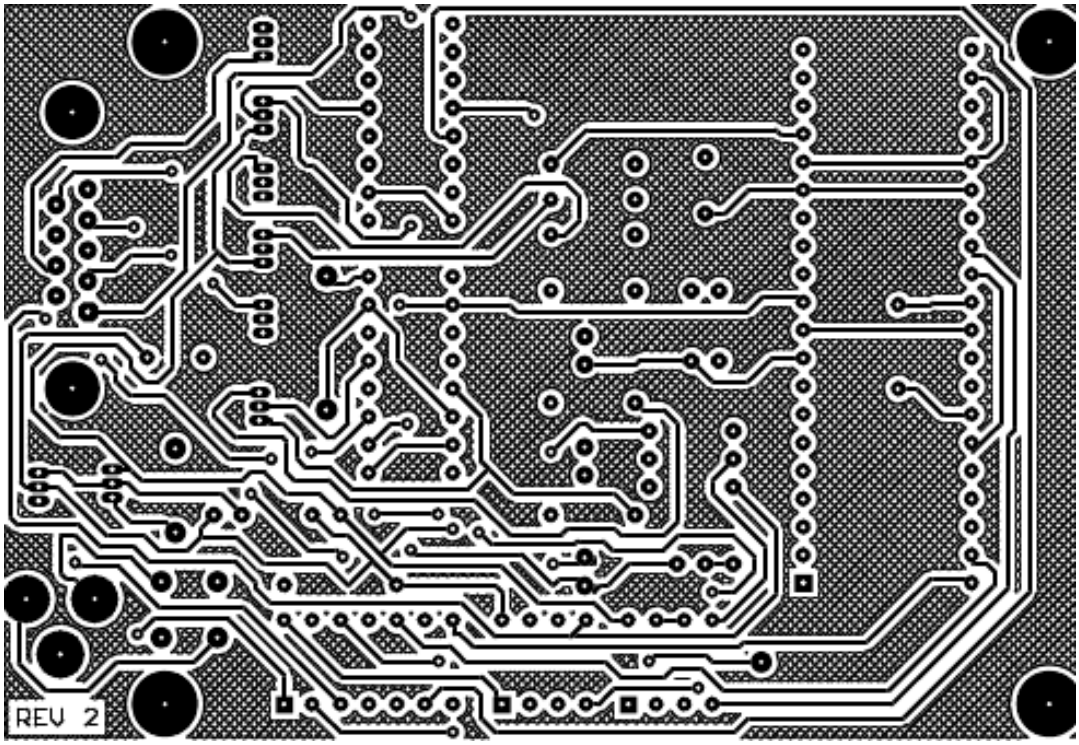
PROPIC2 PIC PROGRAMMER

Status: OK




T1
T2
T3
T4
T5
T6
CLK
DATA
R1
R2
R4
R5
R8
R9
R10
R11
R12
R13
R24
R25
R7
R6
R3
R16
D1
D2
D3
C4
C6
C7
R17
R18
R19
R20
R21
R22
R26
D4
D5
D6
IC1
IC2
D7
C1
C2
C3
K1
J1
IC3
U1
U2
IC4
VCC VAR
GND





Links:

- http://www.mpu51.com/propic/propic2.htm

CHAPTER

TWENTYSIX

SERIAL PORT 1WIRE ADAPTER

Status: OK

original design: http://owfs.org/index.php?page=com-ds9097-passive

CHAPTER

TWENTYSEVEN

SERIAL PORT LOOPBACK PLUG

Status: OK

It is used for testing the serial port.

Connected pins:

- 1-6-4
- 2-3
- 7-8

27.1 Images

27.2 Sources

original design

Serial port pinout

CHAPTER

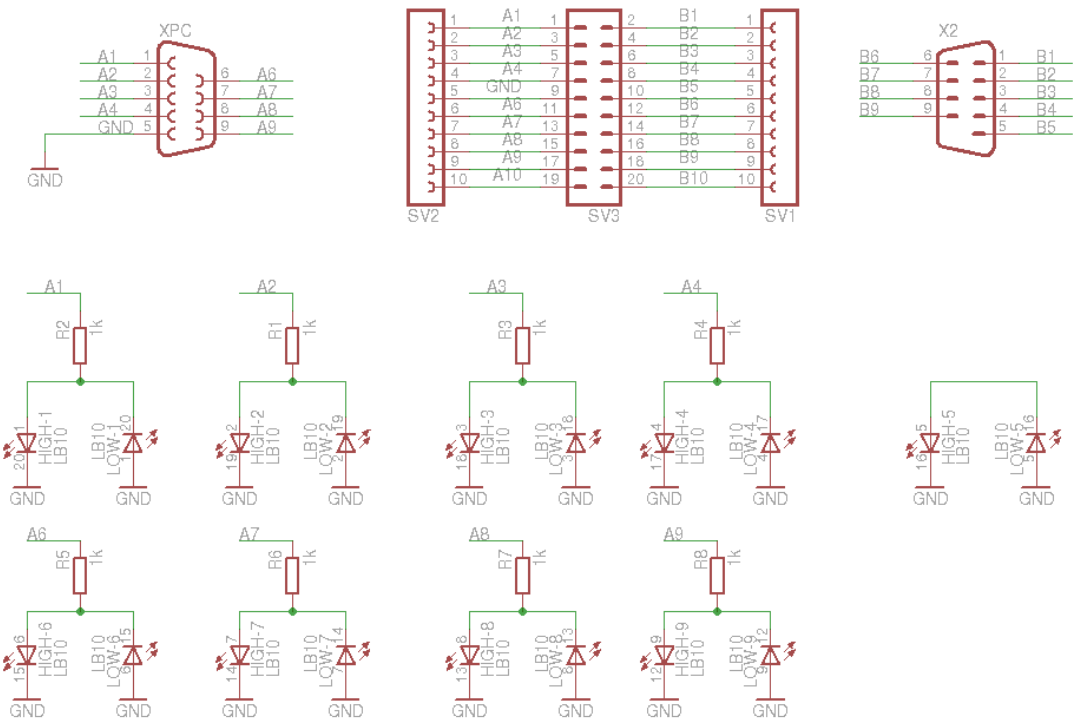
TWENTYEIGHT

SERIAL PORT MONITOR

Status: OK

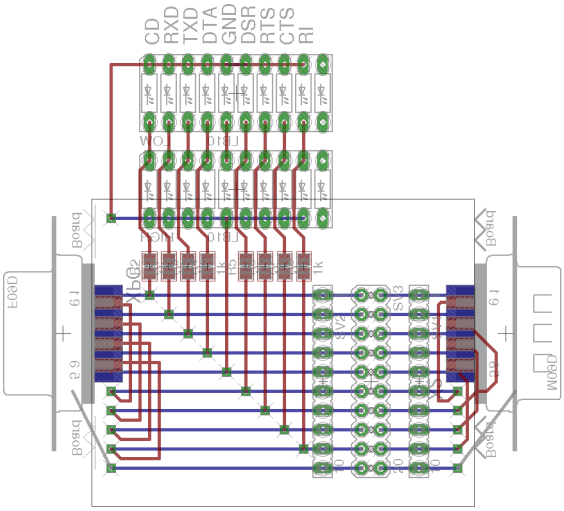
On each signal there is one LED for positive and one LED for negative voltage. It is easy to change connections or connect external parts. Examples: Loop-Back, Null Modem,..

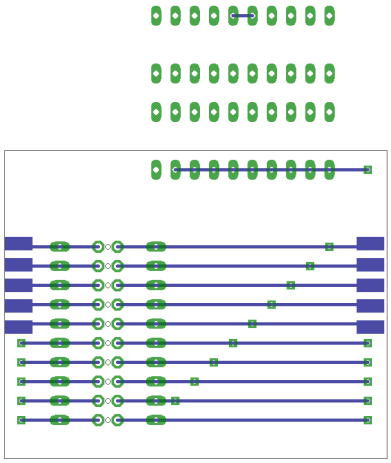
28.1 Schematic

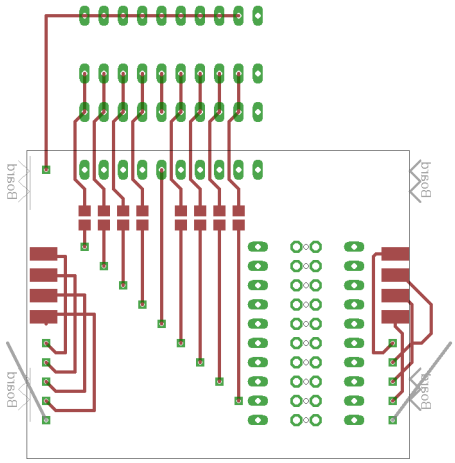


28.2 Board

Normal, bottom mirrored, wires only:







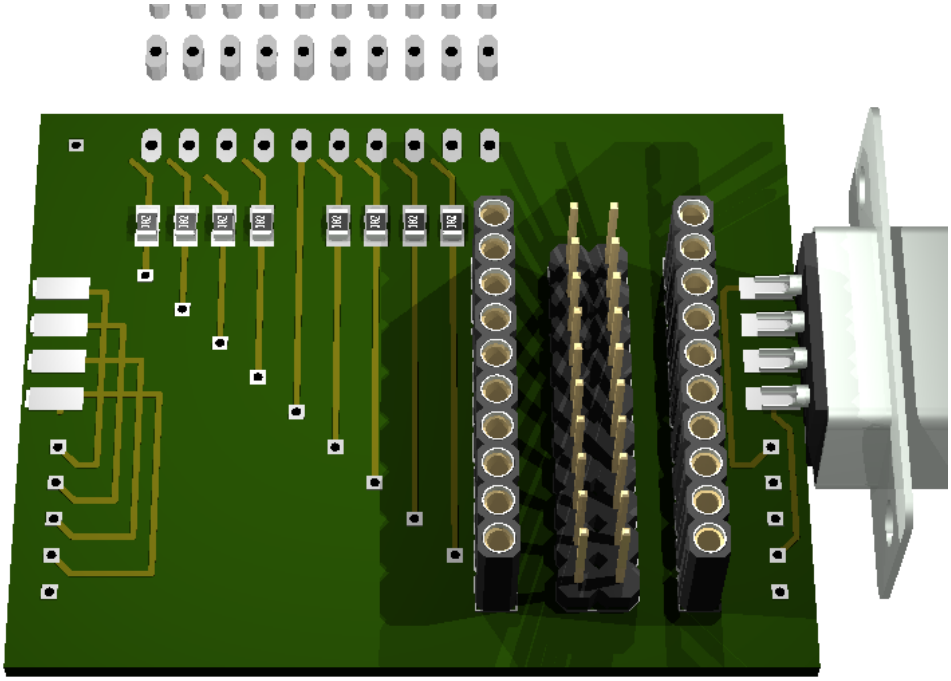
28.3 Partlist

Table 28.1:

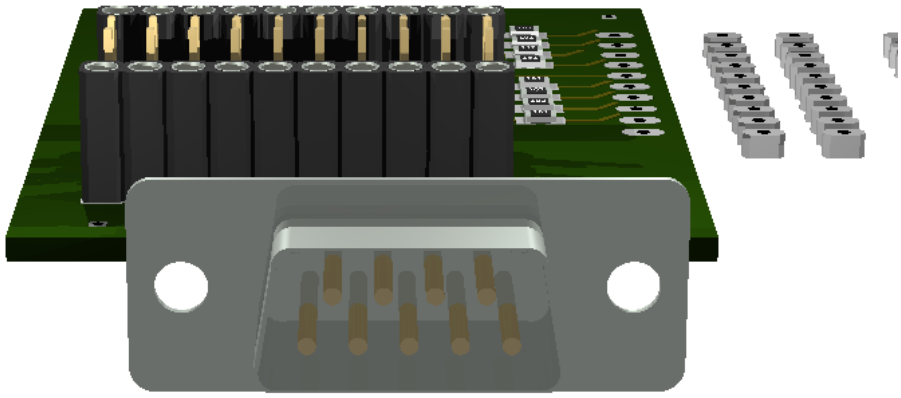
part	value	position
HIGH	LB10	(0.75 1.65)
LOW	LB10	(0.75 2.15)
R1	1k	(0.4 1.25)
R2	1k	(0.3 1.25)
R3	1k	(0.5 1.25)
R4	1k	(0.6 1.25)
R5	1k	(0.8 1.25)
R6	1k	(0.9 1.25)
R7	1k	(1 1.25)
R8	1k	(1.1 1.25)
SV1		(1.7 0.65)
SV2		(1.2 0.65)
SV3		(1.45 0.65)
X2		(2.15 0.9)
XPC		(-0.15 0.9)

28.4 3D view

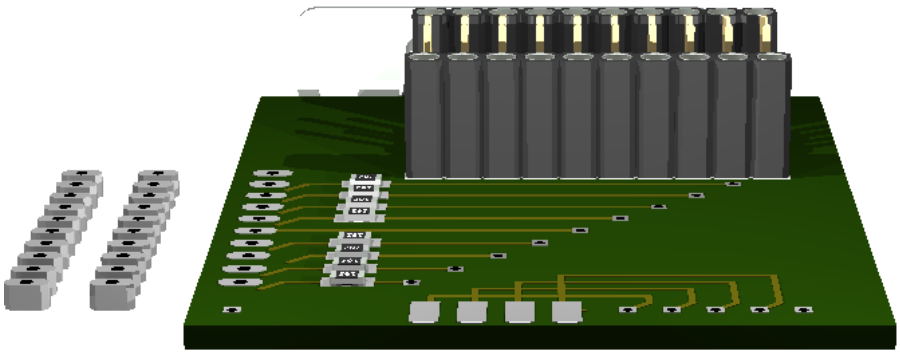
28.4.1 Front



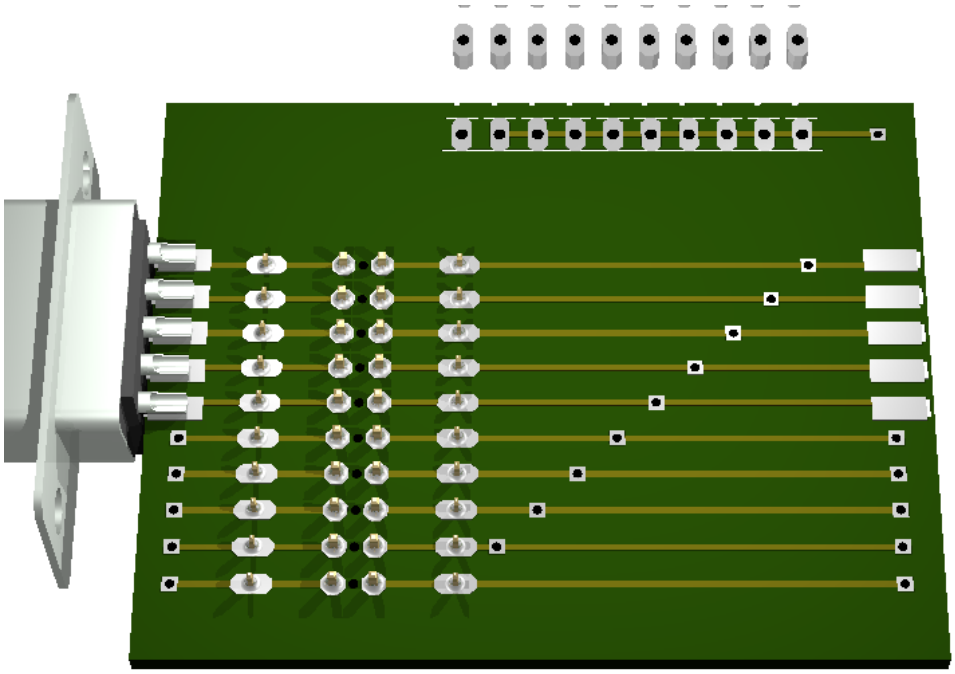
28.4.2 Right side



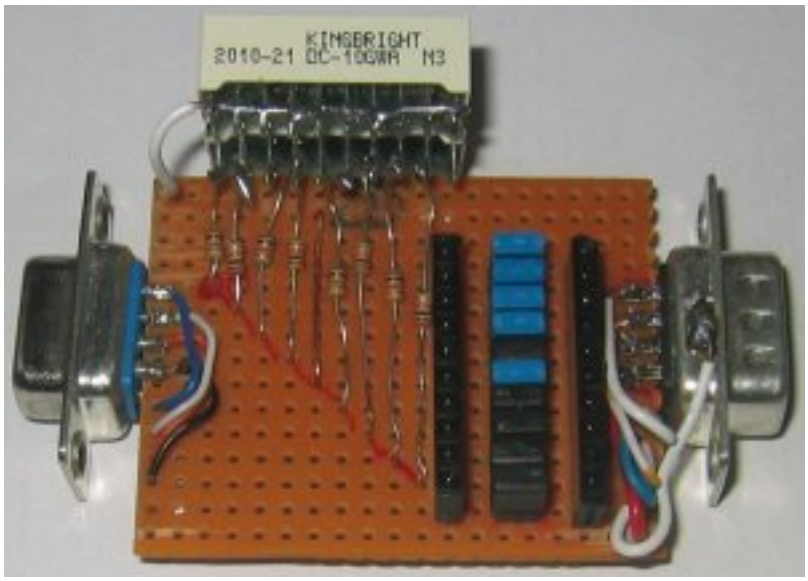
28.4.3 Left side

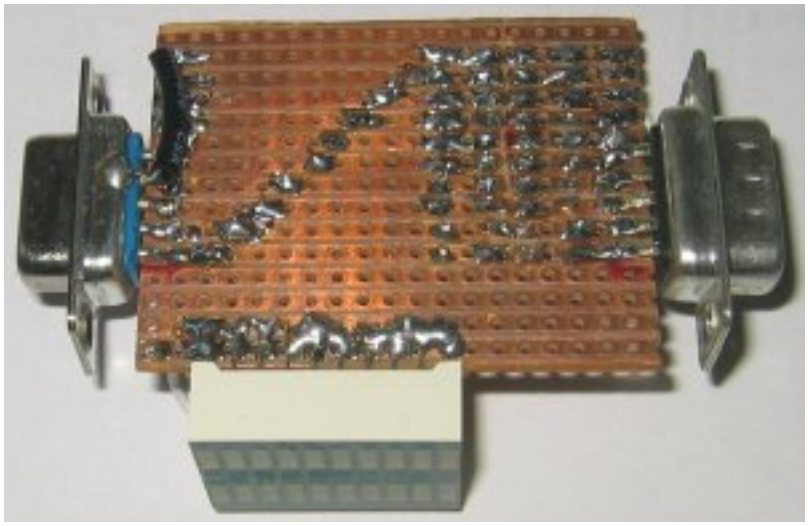


28.4.4 Bottom



28.5 Images





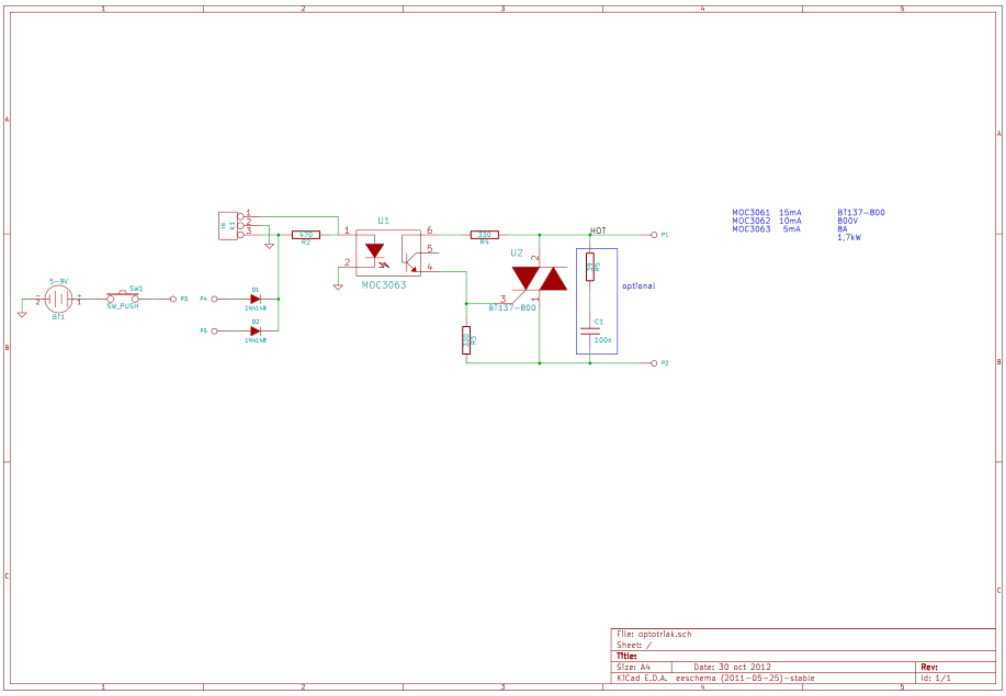
CHAPTER

TWENTYNINE

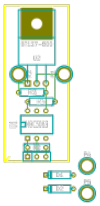
STANDBY KILLER

Status: OK

29.1 Schematic



29.2 Board



29.3 Sources

based on: http://www.extremecircuits.net/2010/05/usb-operated-home-appliances.html

CHAPTER

THIRTY

STK200 AVR PROGRAMMER

Status: OK

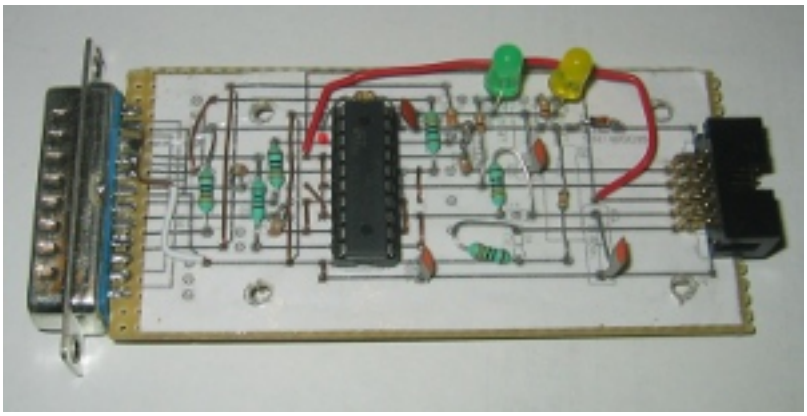
It is used for programming AVR controller and Arduino compatible boards using the parallel port.

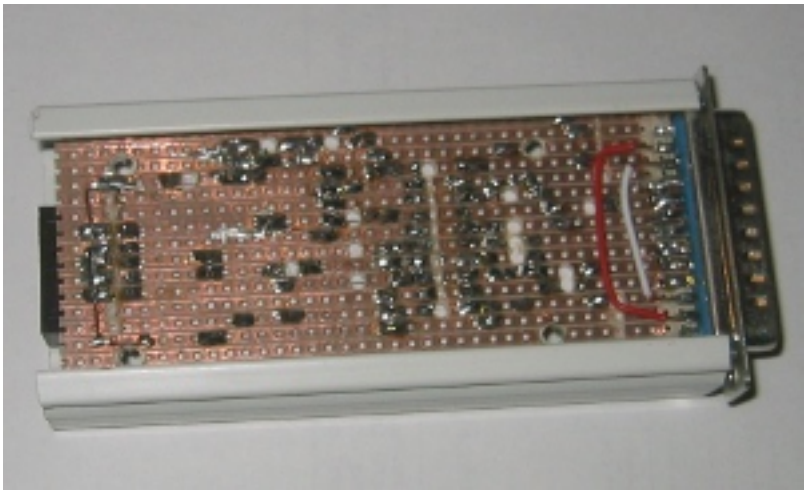
30.1 Test on Ubuntu

checking:

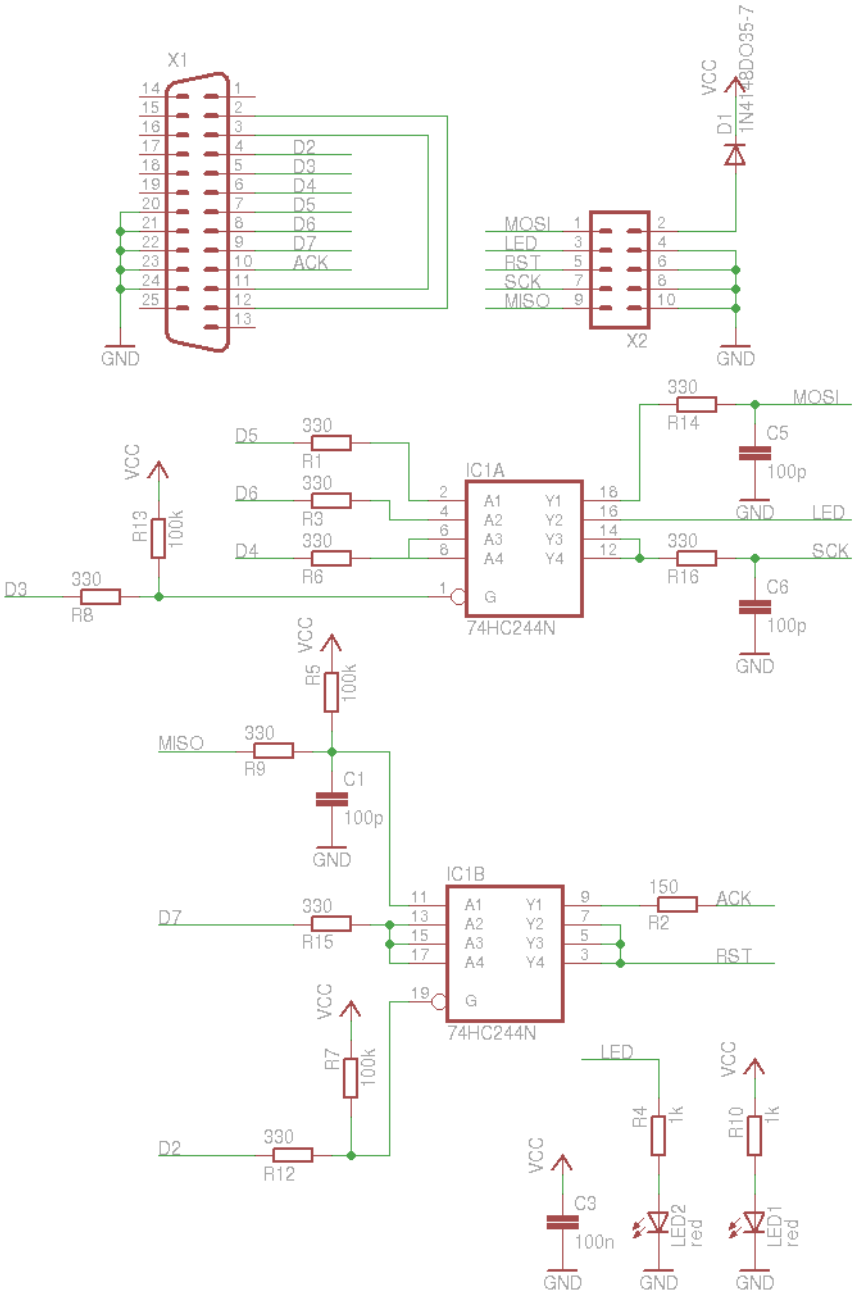
```
$ avrdude -patmega88 -cstk200

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e930a

avrdude: safemode: Fuses OK

avrdude done.  Thank you.
```

30.2 Image



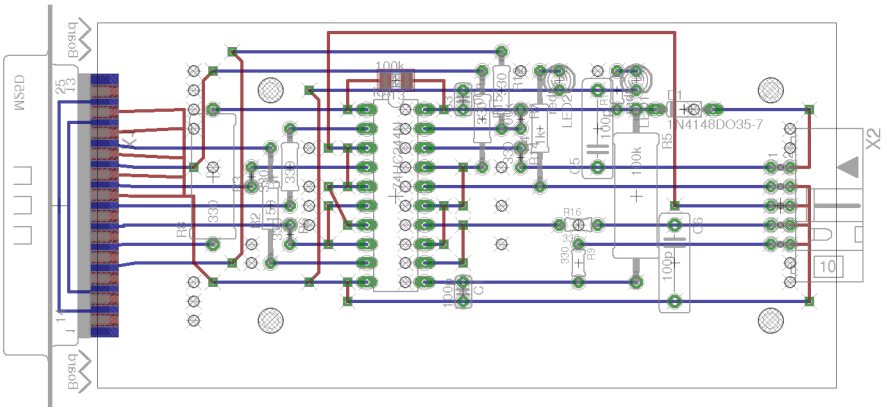


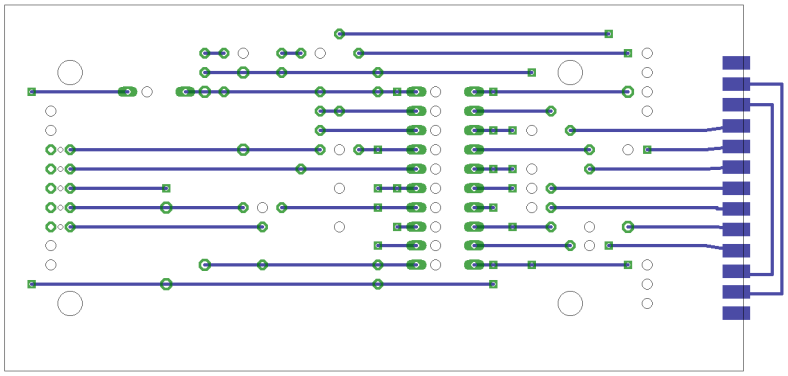
30.3 Schematic

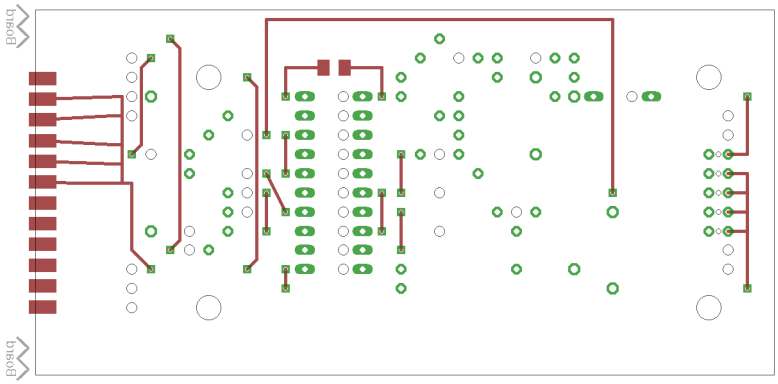


30.4 Board

Normal, bottom mirrored, wires only:







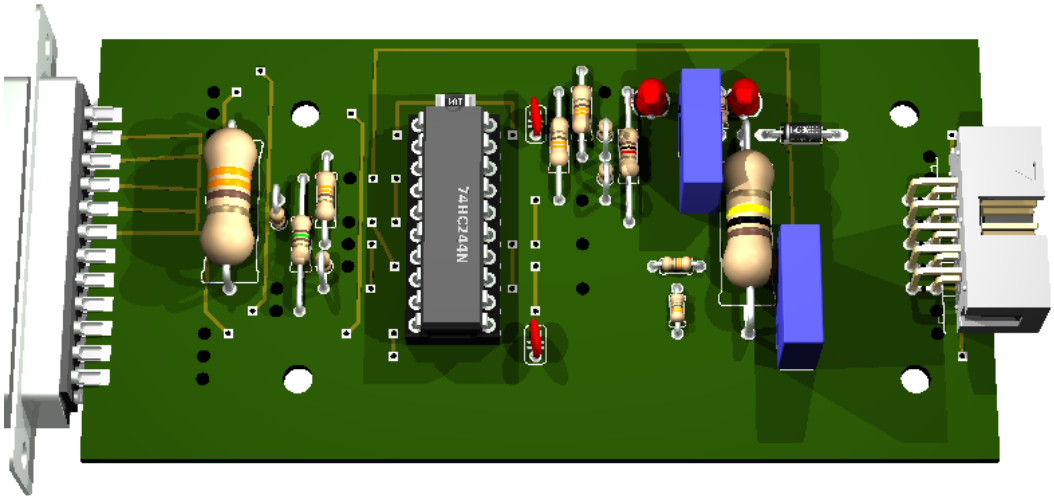
30.5 Partlist

Table 30.1:

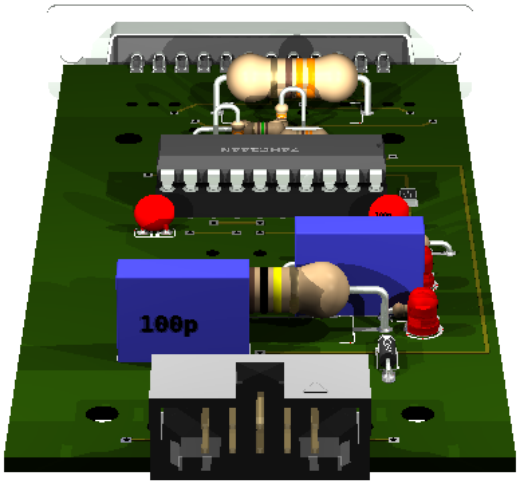
part	value	position
C1	100p	(2.3 0.95)
C3	100n	(2.3 1.95)
C5	100p	(3 1.8)
C6	100p	(3.4 1.1)
D1	1N4148DO35-7	(3.45 1.9)
IC1	74HC244N	(1.95 1.45)
LED1	red	(3.2 2.05)
LED2	red	(2.8 2.05)
R1	330	(1.4 1.6)
R2	150	(1.3 1.4)
R3	330	(1.2 1.55)
R4	1k	(2.7 1.8)
R5	100k	(3.2 1.45)
R6	330	(1.4 1.25)
R7	100k	(2.6 1.85)
R8	330	(1 1.55)
R9	330	(2.9 1.1)
R10	1k	(3.1 2)
R12	330	(2.5 2)
R13	100k	(1.95 2.05)
R14	330	(2.6 1.65)
R15	330	(2.4 1.85)
R16	330	(2.9 1.3)
X1		(0.2 1.4)
X2		(3.95 1.4)

30.6 3D view

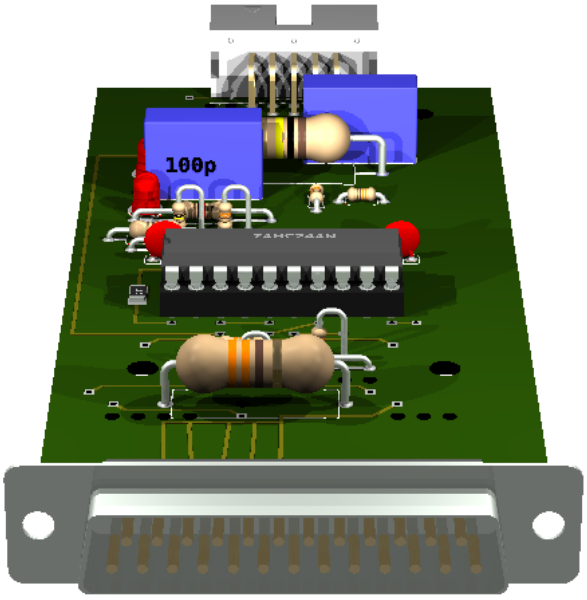
30.6.1 Front



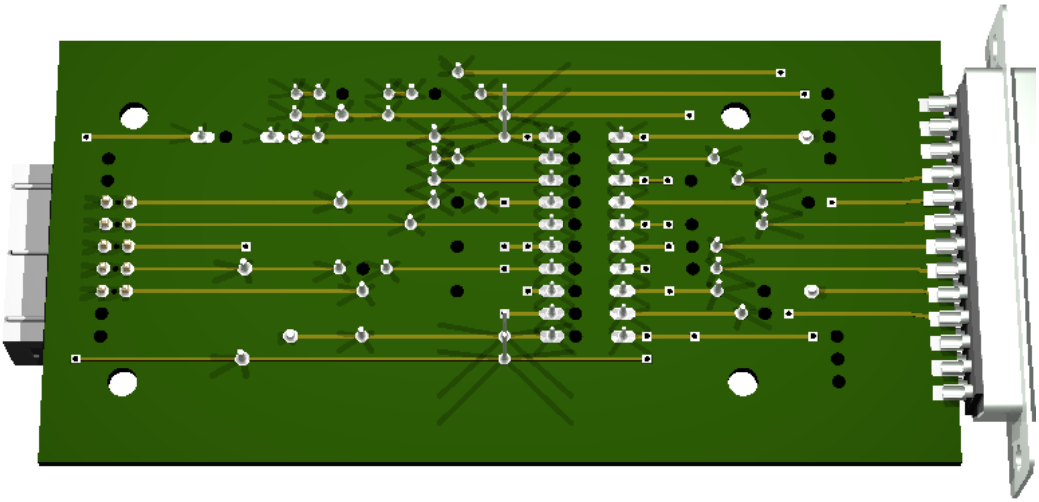
30.6.2 Right side



30.6.3 Left side



30.6.4 Bottom



30.7 Sources

original design

Parallel port specification

AVR ISP Header Pinouts

similar designs:

- http://www.sbprojects.com/projects/stk200/

CHAPTER

THIRTYONE

USB 1WIRE HUB

Status: OK

Low speed USB device which can handle multiple 1wire buses. Example program: onewire_demo.py under softusbduino

Based on V-USB hardware.

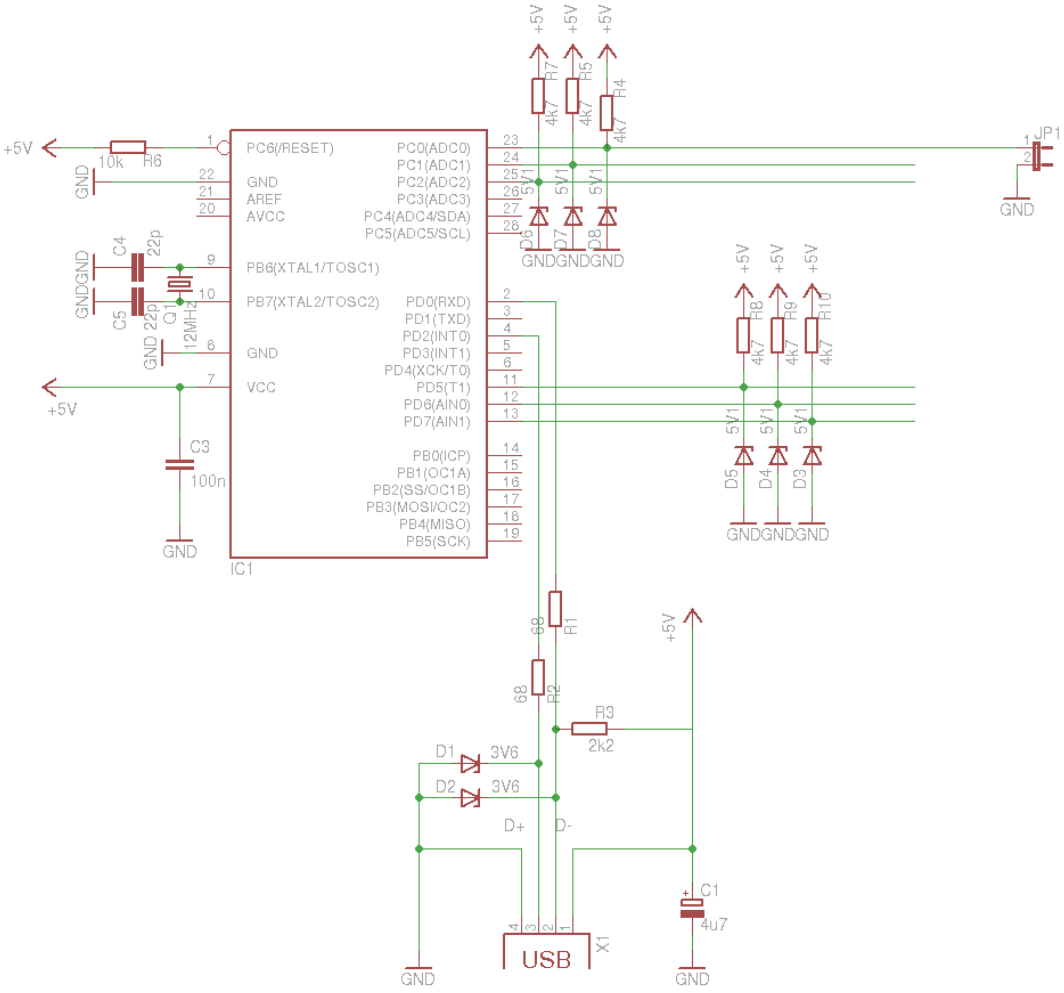
connections:

function	AVR pin	Arduino pin
1wire	PC0	A0
1wire	PC1	A1
1wire	PC2	A2
1wire	PD5	D5
1wire	PD6	D6
1wire	PD7	D7
USB D-	PD0	D0
USB D+	PD2	D2

V-USB defines:

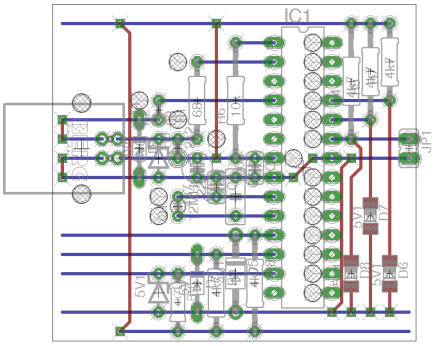
```
#define USB_CFG_IOPORTNAME      D
#define USB_CFG_DMINUS_BIT      0
#define USB_CFG_DPLUS_BIT       2
```

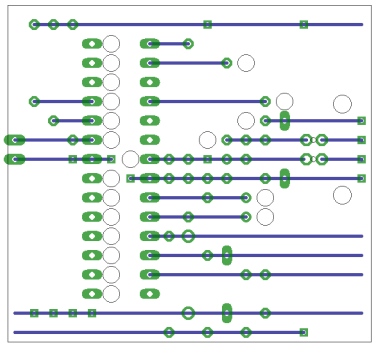
31.1 Schematic

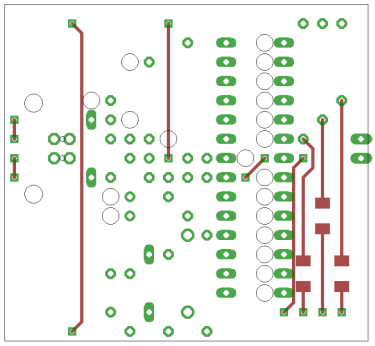


31.2 Board

Normal, bottom mirrored, wires only:







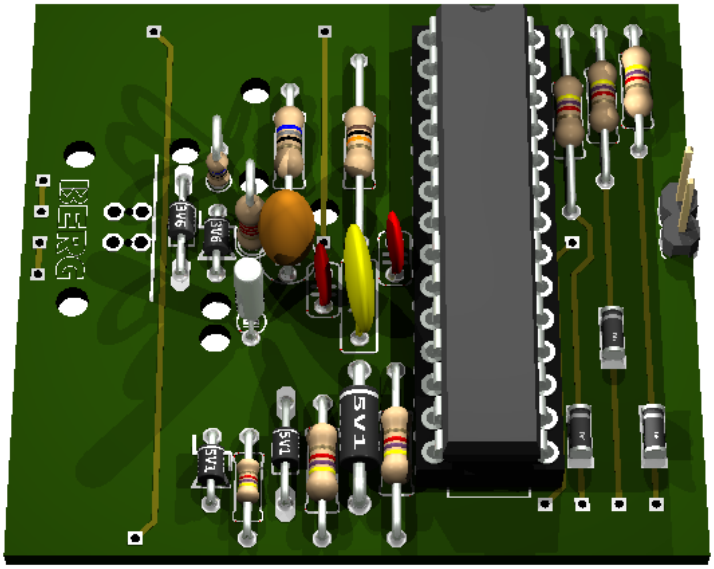
31.3 Partlist

Table 31.1:

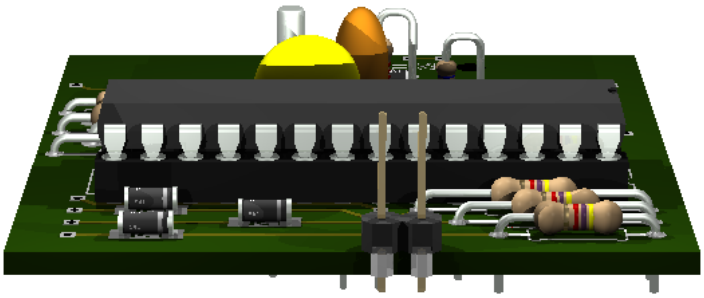
part	value	position
C1	4u7	(0.9 1.05)
C3	100n	(1.2 1.05)
C4	22p	(1 0.95)
C5	22p	(1.1 0.9)
D1	3V6	(0.6 1.15)
D2	3V6	(0.7 1.1)
D3	5V1	(0.7 0.4)
D4	5V1	(0.9 0.45)
D5	5V1	(1.1 0.5)
D6	5V1	(1.9 0.5)
D7	5V1	(1.8 0.8)
D8	5V1	(1.7 0.5)
IC1		(1.45 1.05)
JP1		(2 1.15)
Q1	12MHz	(0.8 0.85)
R1	68	(0.9 1.4)
R2	68	(0.7 1.35)
R3	2k2	(0.8 1.15)
R4	4k7	(1.7 1.5)
R5	4k7	(1.8 1.55)
R6	10k	(1.1 1.4)
R7	4k7	(1.9 1.6)
R8	4k7	(1.2 0.45)
R9	4k7	(1 0.4)
R10	4k7	(0.8 0.35)
X1		(0.3 1.15)

31.4 3D view

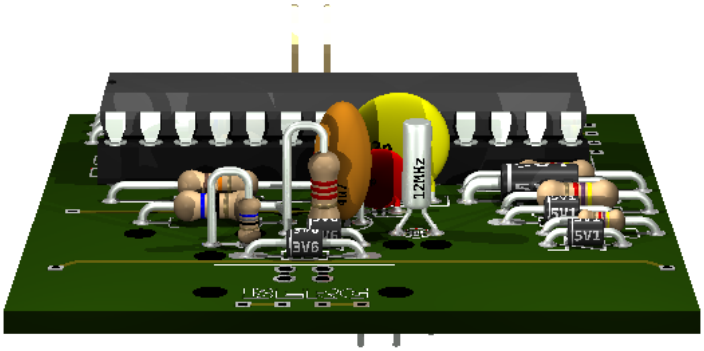
31.4.1 Front



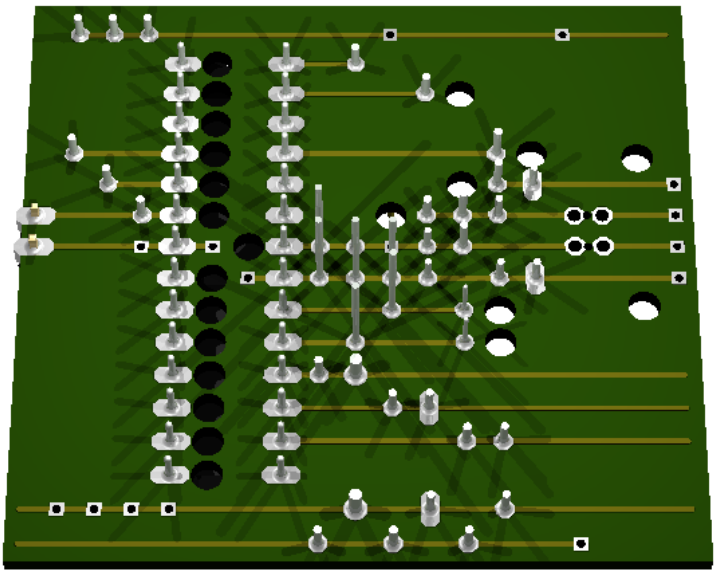
31.4.2 Right side



31.4.3 Left side



31.4.4 Bottom



CHAPTER

THIRTYTWO

USB TRICKLE CHARGER

Status: OK

Charging current: 50 mA

USB port power is max 500 mA -> 9 charger can be connected to one USB port

32.1 Power dissipation

LED

- current = 15 mA
- power = 30 mW

R4

- current = 35 mA
- power = 60 mW

Transistor

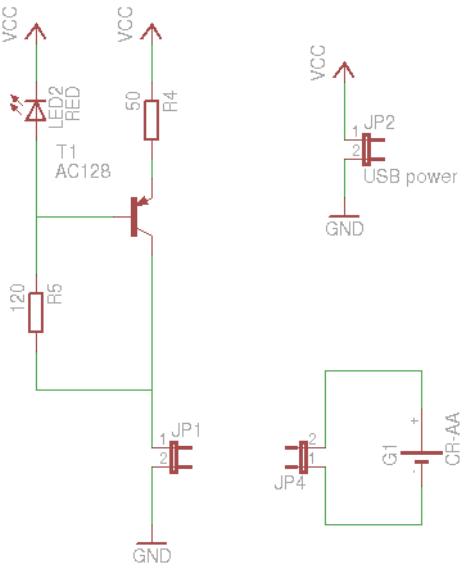
- current = 50 mA
- power = 100 mW

32.2 Charge time

T = C * 1.5 / 50mA

Capacity	current	time(h)	time(d)
1000 mAh	C/20	30h	1.25d
2000 mAh	C/40	60h	2.5d
3000 mAh	C/60	90h	3.75d

32.3 Schematic



32.4 Sources

Based on this design: http://www.extremecircuits.net/2009/10/low-cost-universal-battery-charger.html

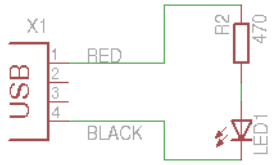
CHAPTER

THIRTYTHREE

USB LED

Status: OK

It is used for testing USB power.



CHAPTER

THIRTYFOUR

USBASP AVR PROGRAMMER

Status: OK

It is used for programming AVR controller and Arduino compatible boards using the USB port.

firmware, design: http://www.fischl.de/usbasp/

USBasp is based on V-USB (http://www.obdev.at/products/vusb/index.html)

34.1 V-USB hardware recommendation

only difference to USBasp: 1.5 kΩ pull-up resistor

http://vusb.wikidot.com/hardware

“Solution B: Level conversion on D+ and D- Level conversion with Zener diodes.

Instead of reducing the AVR’s power supply, we can limit the output voltage on D+ and D- with Zener diodes. We recommend 3.6 V low power types, those that look like 1N4148 (usually 500 mW or less). Low power types are required because they have less capacitance and thus cause less distortion on the data lines. And 3.6 V is better than 3.3 V because 3.3 V diodes yield only ca. 2.7 V in conjunction with an 1.5 kΩ (or more exactly 10 kΩ) pull-up resistor. With 3.3 V diodes, the device may not be detected reliably.

If you use Zener diodes for level conversion, please measure the voltage levels to make sure that the diodes you have chosen match the requirements.

Advantages of the Zener diode approach:

- Low cost.
- Easy to obtain.
- Entire design can be at 5 V.
- AVR can be clocked at high rates.

Disadvantages:

- Not a clean solution, a compromise between all parameters must be found.
- Zener diodes come with a broad range of characteristics, especially at low currents, results may not be reproducible.
- High currents when sending high-level.
- High level is different for signaling and in idle state because signaling uses high currents to drive the diodes while idle state is driven by a 1.5 kΩ pull-up resistor.”

34.2 Makefile

Tested with atmega88. Makefile settings:

```
TARGET=atmega88
HFUSE=0xdd
LFUSE=0xef
```

34.3 Test on Ubuntu

checking:

```
$ lsusb |grep -i 16c0:05dc
Bus 003 Device 006: ID 16c0:05dc VOTI shared ID for use with libusb

$ ls -l /dev/bus/usb/003/006
crw-rw-r-- 1 root root 189, 261 2011-11-05 10:31 /dev/bus/usb/003/006

$ avrdude -patmega88 -cusbasp
avrdude: Warning: cannot query manufacturer for device: error sending control message: Operation
avrdude: error: could not find USB device "USBasp" with vid=0x16c0 pid=0x5dc
```

The permission should be changed:

```
$sudo nano /etc/udev/rules.d/60-objdev.rules
```

add this line:

```
ATTRS{idVendor}=="16c0", ATTRS{idProduct}=="05dc", GROUP="users", MODE="0666"
```

update rules:

```
$sudo udevadm trigger
```

checking again:

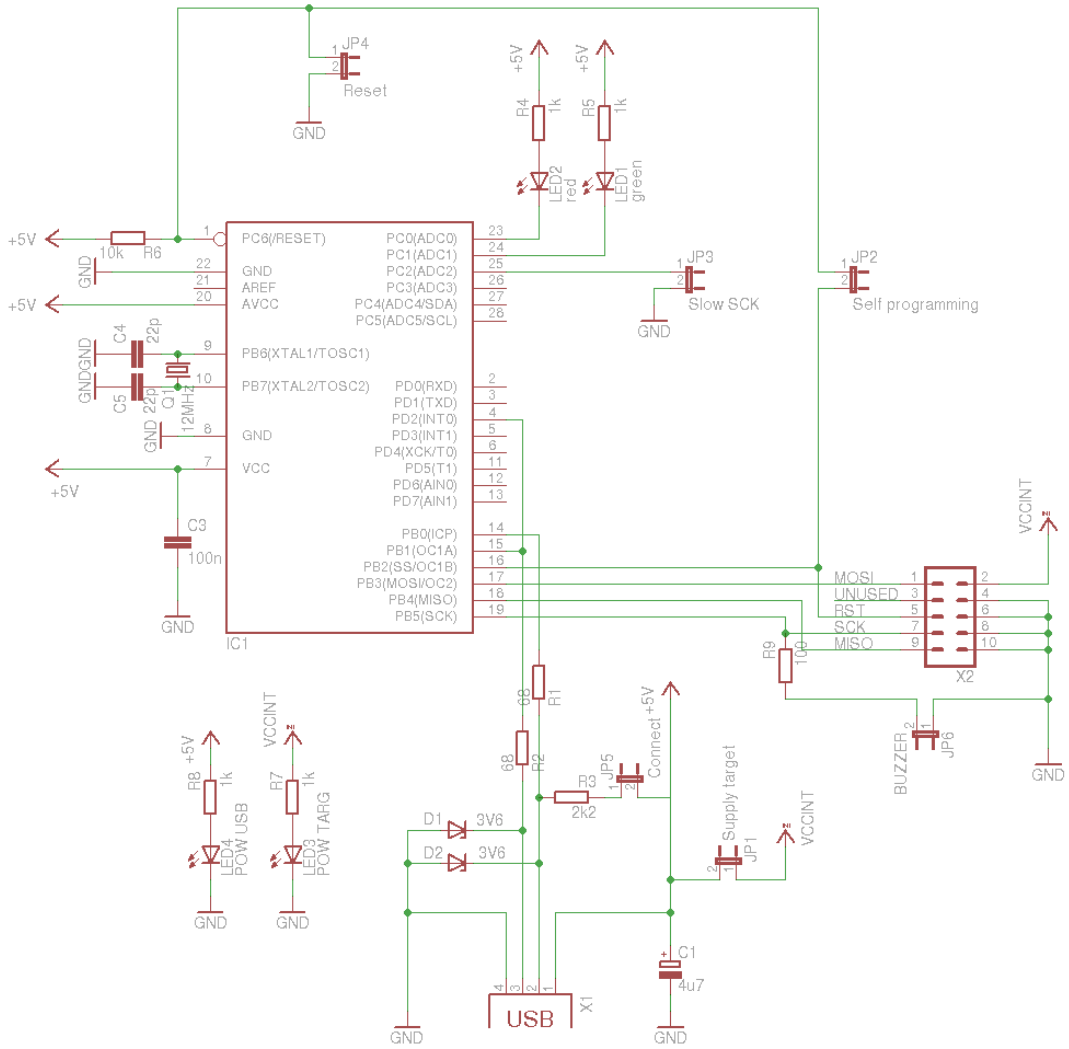
```
$ ls -l /dev/bus/usb/003/006
crw-rw-rw- 1 root users 189, 261 2011-11-05 10:33 /dev/bus/usb/003/006

$ avrdude -patmega88 -cusbasp
avrdude: error: programm enable: target doesn't answer. 1
avrdude: initialization failed, rc=-1
         Double check connections and try again, or use -F to override
         this check.
avrdude done.  Thank you.
```

Permission is OK now.

Testing with connected controller:

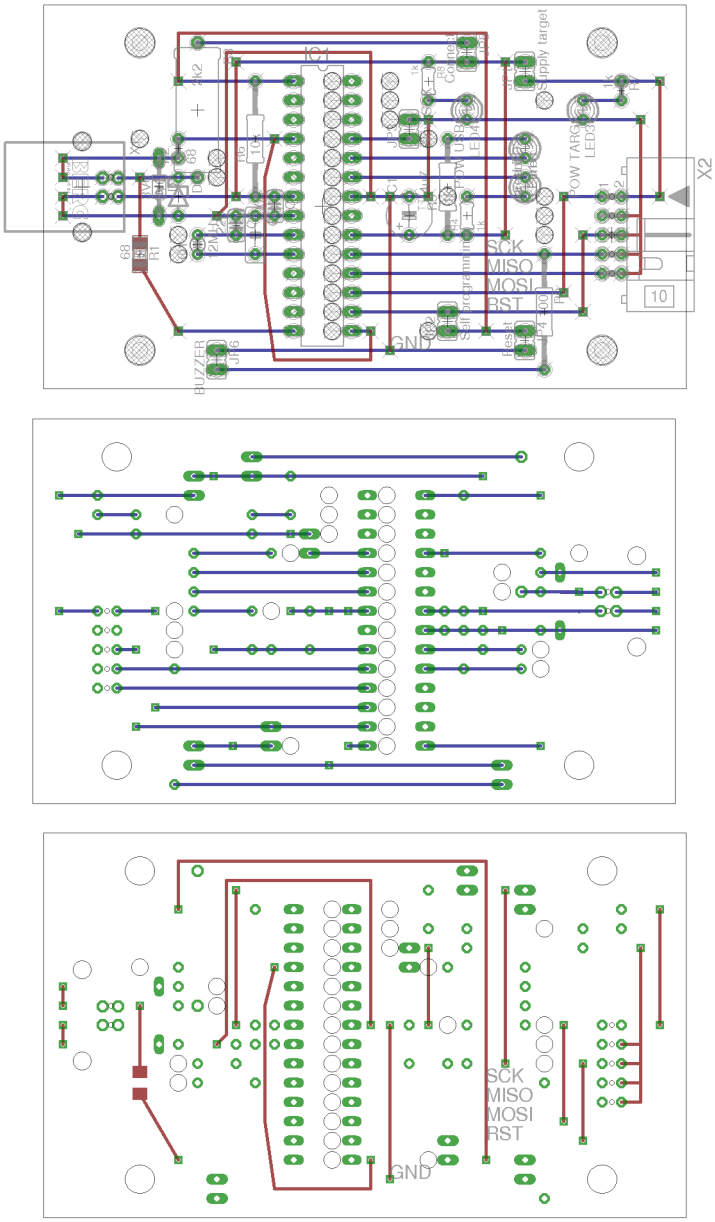
```
$ avrdude -patmega88 -cusbasp

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.01s

avrdude: Device signature = 0x1e930a

avrdude: safemode: Fuses OK

avrdude done.  Thank you.
```

34.4 Schematic



34.5 Board

Normal, bottom mirrored, wires only:



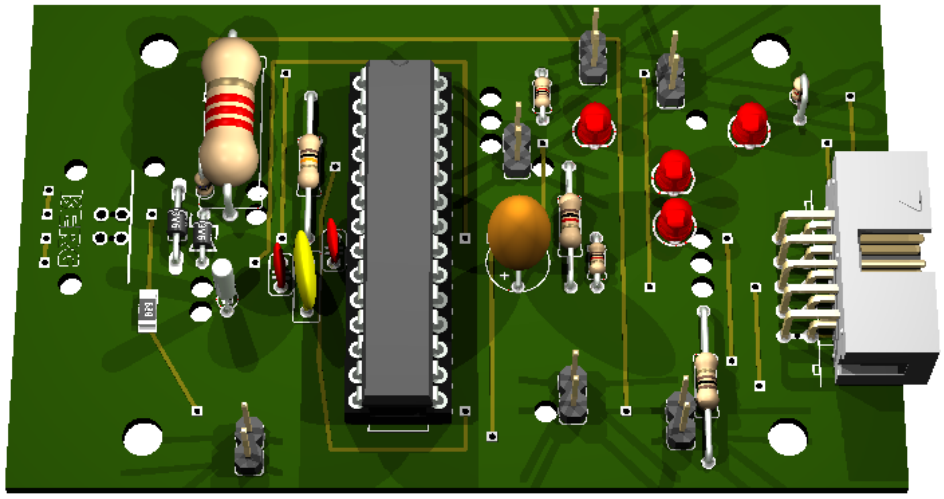
34.6 Partlist

Table 34.1:

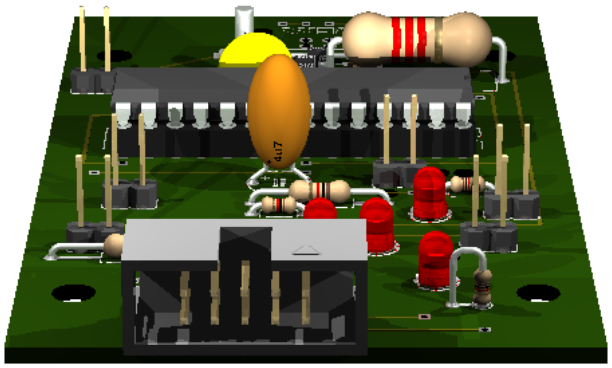
part	value	position
C1	4u7	(2.4 0.8)
C3	100n	(1.7 0.85)
C4	22p	(1.5 0.75)
C5	22p	(1.6 0.7)
D1	3V6	(1.1 0.95)
D2	3V6	(1.2 0.9)
IC1		(1.95 0.85)
JP1	Supply target	(3 1.55)
JP2	Self programming	(2.6 0.25)
JP3	Slow SCK	(2.4 1.25)
JP4	Reset	(3 0.15)
JP5	Connect	(2.7 1.65)
JP6	BUZZER	(1.4 0.05)
LED1	green	(3 1.15)
LED2	red	(3 0.95)
LED3	POW TARG	(3.3 1.35)
LED4	POW USB	(2.7 1.35)
Q1	12MHz	(1.3 0.65)
R1	68	(1 0.6)
R2	68	(1.2 1.15)
R3	2k2	(1.3 1.35)
R4	1k	(2.7 0.8)
R5	1k	(2.6 0.95)
R6	10k	(1.6 1.2)
R7	1k	(3.5 1.45)
R8	1k	(2.5 1.5)
R9	100	(3.1 0.3)
X1		(0.7 0.95)
X2		(3.45 0.7)

34.7 3D view

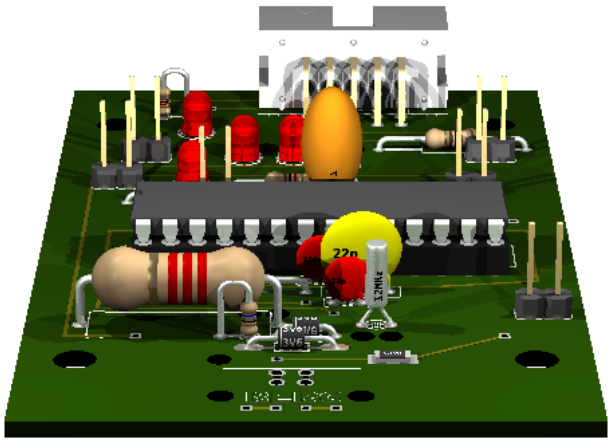
34.7.1 Front



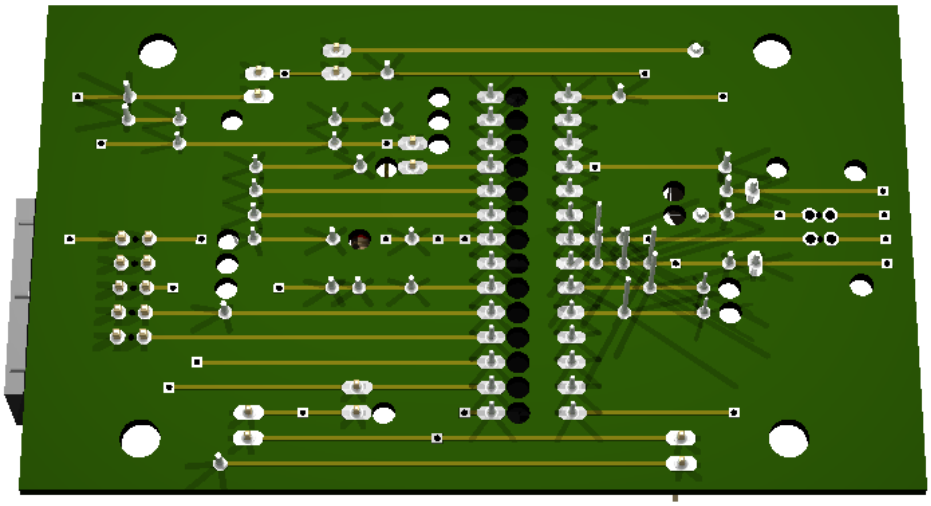
34.7.2 Right side



34.7.3 Left side



34.7.4 Bottom



34.8 Reset

To reset on Ubuntu:

```
#!/usr/bin/env python
import logging
import usb.core
logging.basicConfig(level=logging.DEBUG)
import fcntl

ID_VENDOR = 0x16c0
ID_PRODUCT = 0x05dc
USBDEVFS_RESET = 21780


def find():
    print("searching for device (%x:%x)" % (ID_VENDOR, ID_PRODUCT))
    dev = usb.core.find(idVendor=ID_VENDOR,
                        idProduct=ID_PRODUCT,
                        )
    if not dev:
        print("device not found")
    return dev


def usbstr(i):
```

```
    s = str(i)
    s = '000'[0:3 - len(s)] + s
    return s


def usbfs_filename(dev):
    return '/dev/bus/usb/%s/%s' % (usbstr(dev.bus), usbstr(dev.address))


def reset1(dev):
    fname = usbfs_filename(dev)
    print("Resetting USB device %s" % fname)
    with open(fname, 'w') as fd:
        rc = fcntl.ioctl(fd, USBDEVFS_RESET, 0)
        if (rc < 0):
            print("Error in ioctl")
    print("OK")


def reset2(dev):
    dev.reset()  # not working

dev = find()
if dev:
    reset1(dev)
```

34.9 Sources

original design

AVR ISP Header Pinouts

similar projects:

- http://lategahn.2log.de/index.php?USBASP-Stripboard-layout

CHAPTER

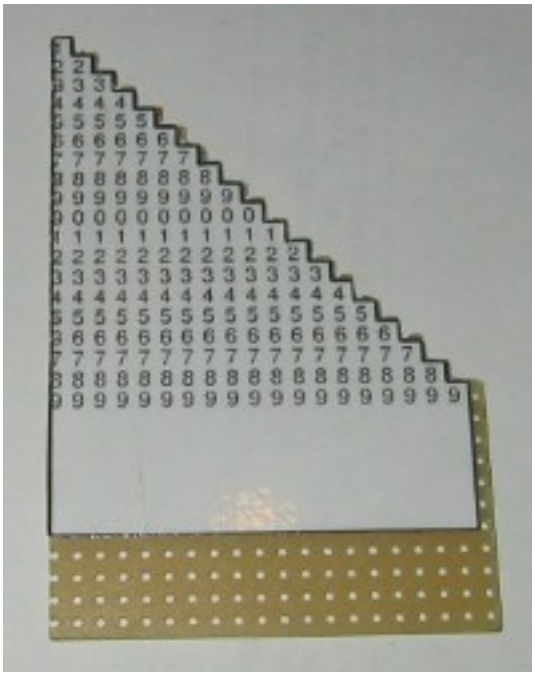
THIRTYFIVE

WIRE BENDING TOOL

Status: OK

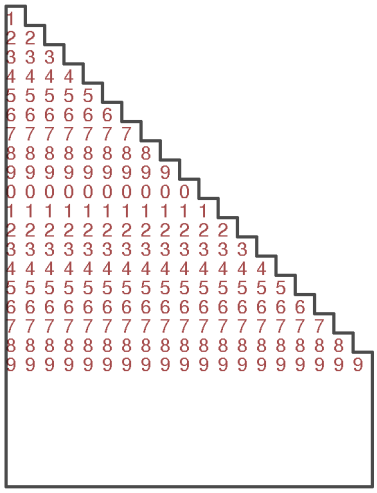
It is used for bending wires.

35.1 Image



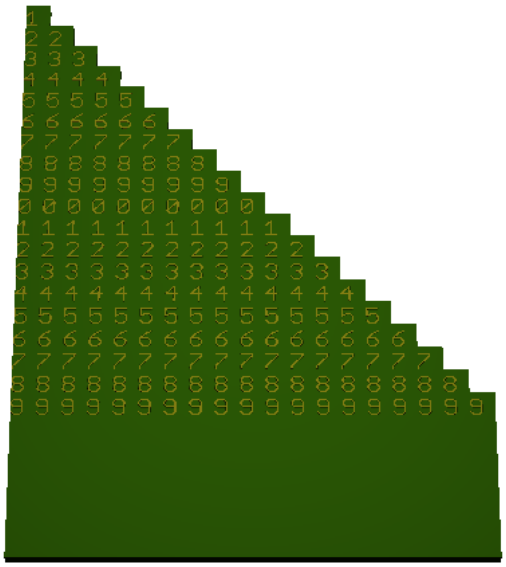
35.2 Board

Normal:



35.3 3D view

35.3.1 Front



35.3.2 Right side

CHAPTER

THIRTYSIX

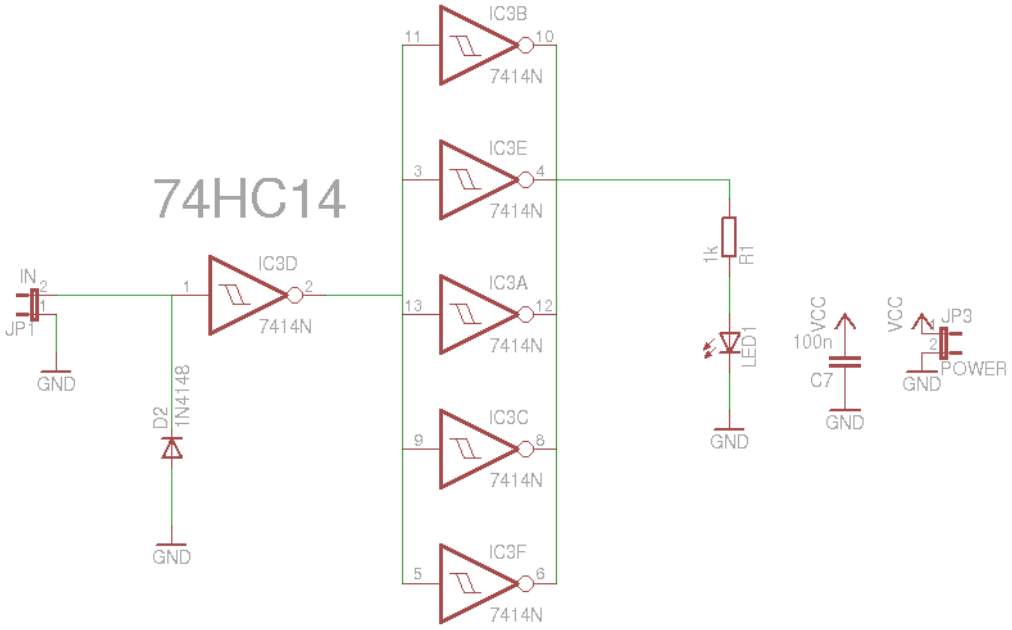
WIRE DETECTOR

Status: OK

It is used for detecting mains wire.

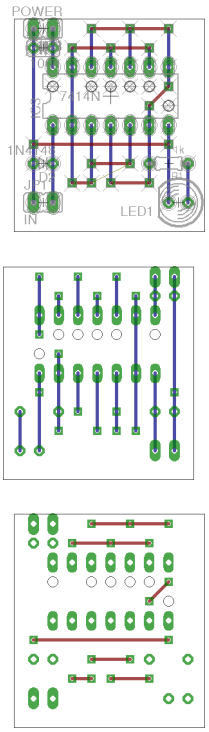
Based on this design: http://www.edn.com/article/511304-Detect_live_ac_mains_lines.php

36.1 Schematic



36.2 Board

Normal, bottom mirrored, wires only:



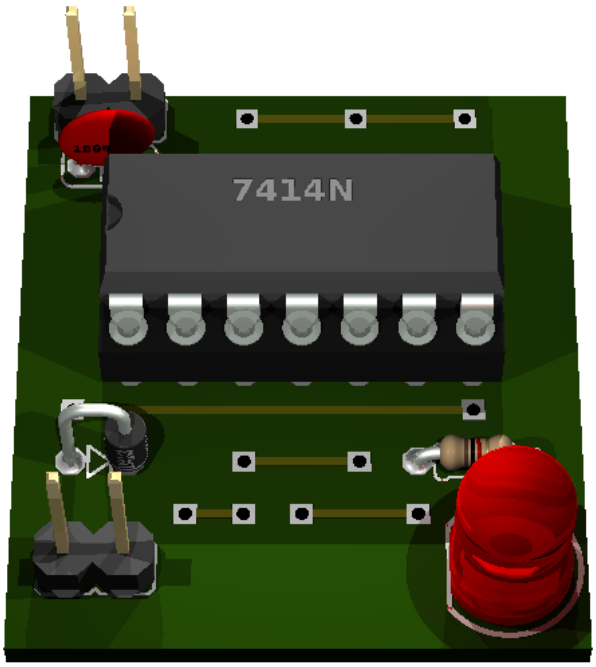
36.3 Partlist

Table 36.1:

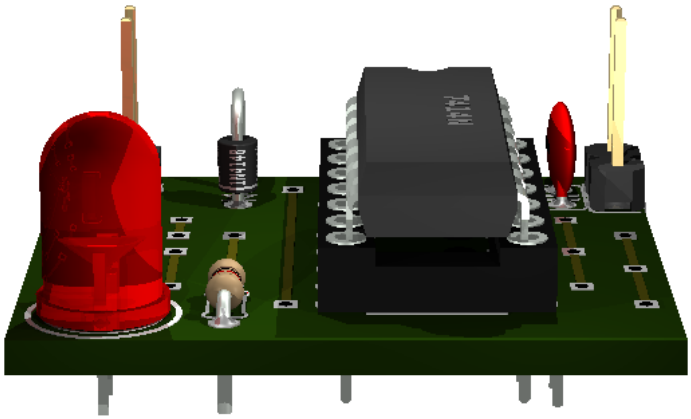
part	value	position
C7	100n	(0.85 1.9)
D2	1N4148	(0.85 1.3)
IC3	7414N	(1.2 1.65)
JP1	IN	(0.85 1.1)
JP3	POWER	(0.85 2)
LED1		(1.55 1.1)
R1	1k	(1.5 1.3)

36.4 3D view

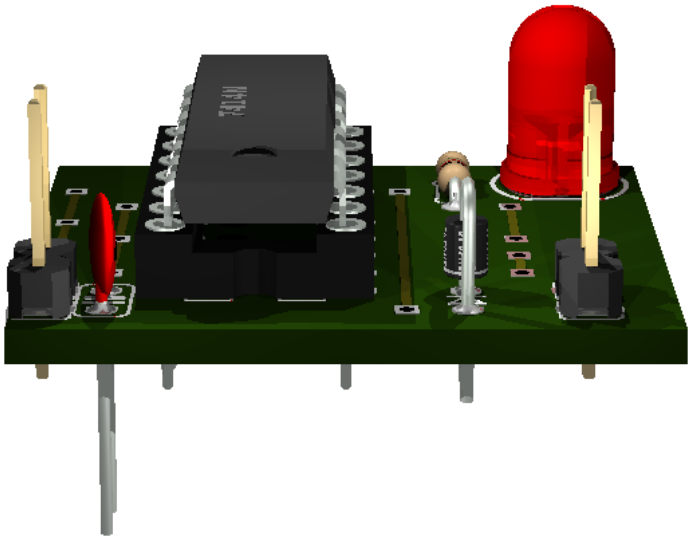
36.4.1 Front



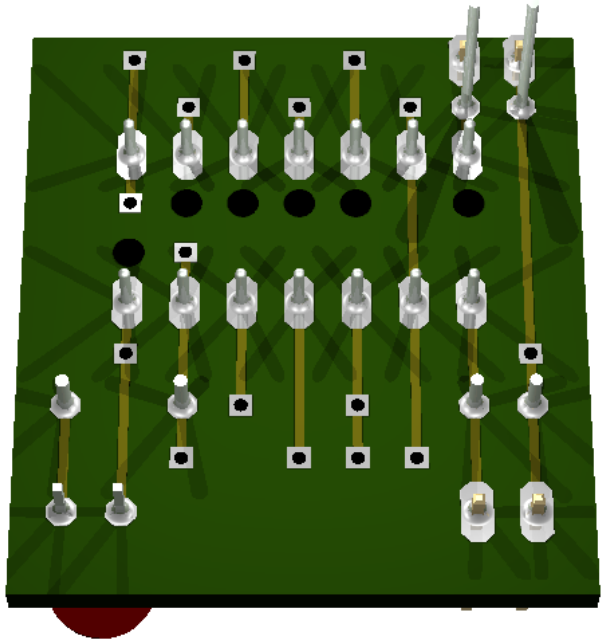
36.4.2 Right side



36.4.3 Left side



36.4.4 Bottom



36.5 Sources

original design