MyElectronicProjects Documentation

Release 0.0.0

ponty

April 21, 2012

CONTENTS

MyElectronicProjects

Date April 21, 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

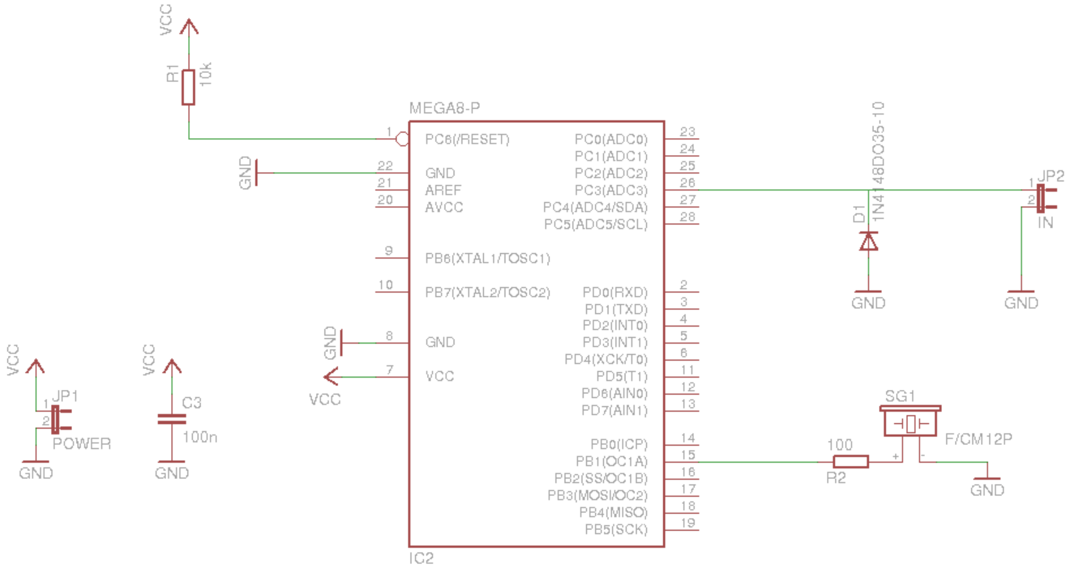
ALARM

Status: ?

3.1 Pins

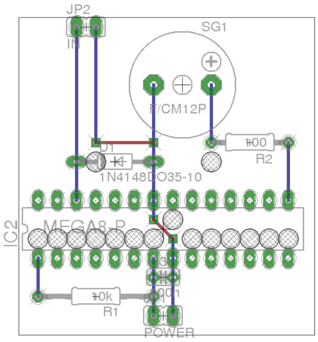
board pin	AVR pin	Arduino pin
in	PC3	A3
speaker	PB1	D9

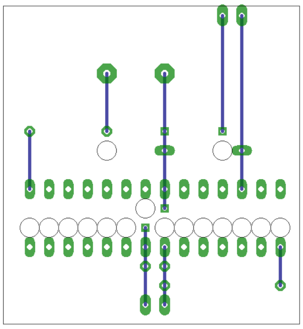
3.2 Schematic

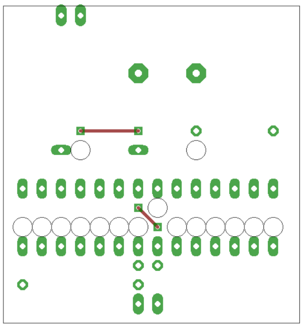


3.3 Board

Normal, bottom mirrored, wires only:







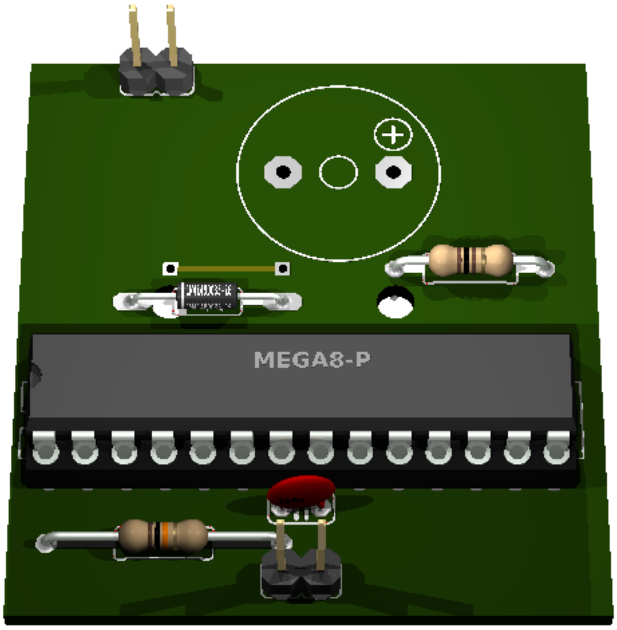
3.4 Partlist

Table 3.1:

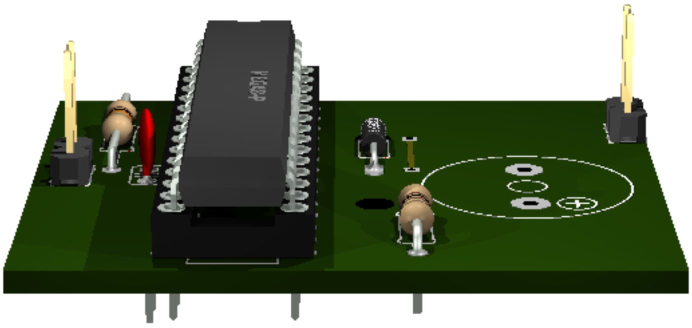
part	value	position
C3	100n	(1.45 0.4)
D1	1N4148DO35-10	(1.2 1)
IC2	MEGA8-P	(1.45 0.65)
JP1	POWER	(1.45 0.2)
JP2	IN	(1.05 1.7)
R1	10k	(1.1 0.3)
R2	100	(1.9 1.1)
SG1	F/CM12P	(1.55 1.4)

3.5 3D view

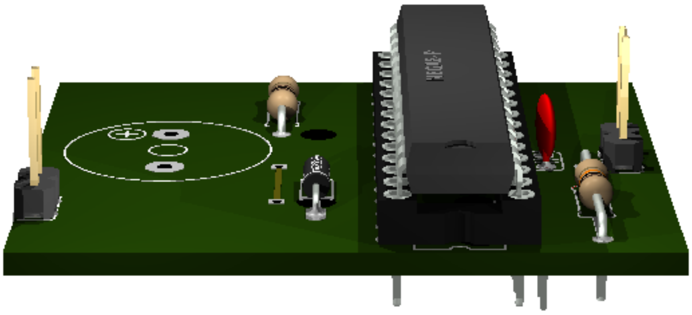
3.5.1 Front



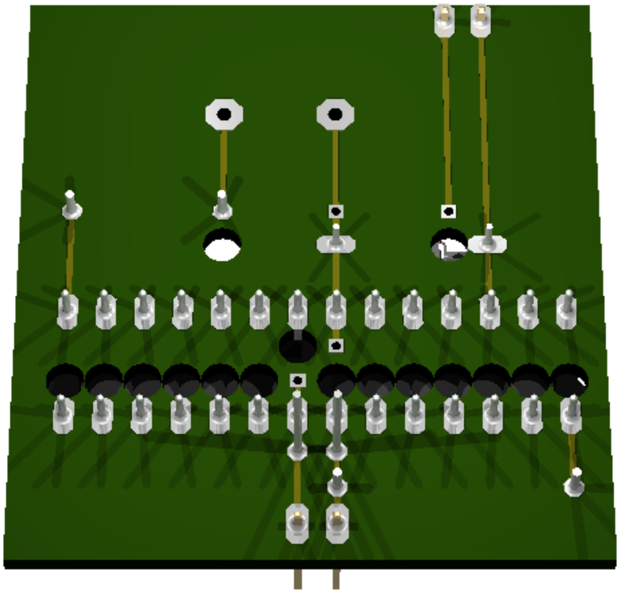
3.5.2 Right side



3.5.3 Left side



3.5.4 Bottom



CHAPTER

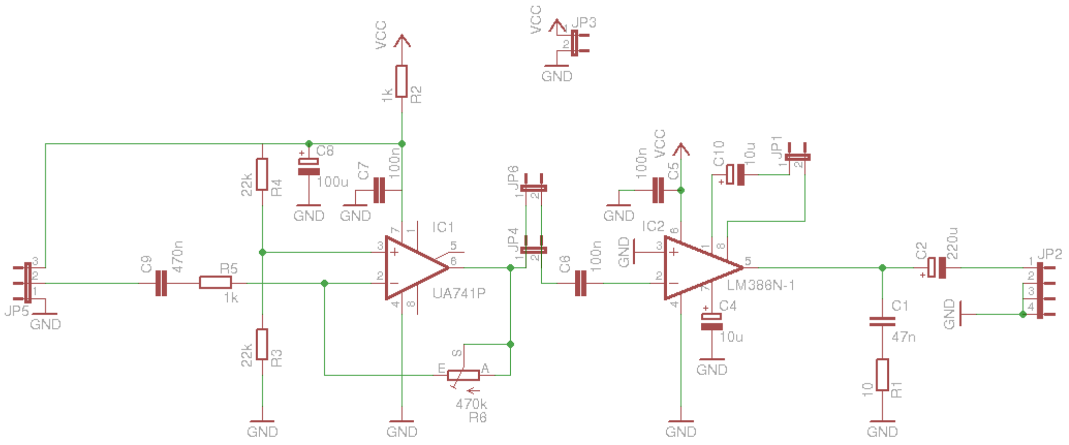
FOUR

AUDIO_AMPLIFIER

Status: under construction

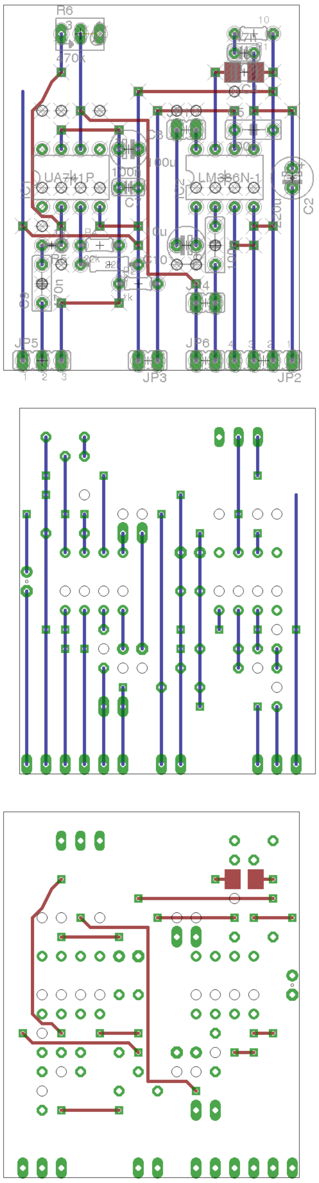
It is used for ...

4.1 Schematic



4.2 Board

Normal, bottom mirrored, wires only:



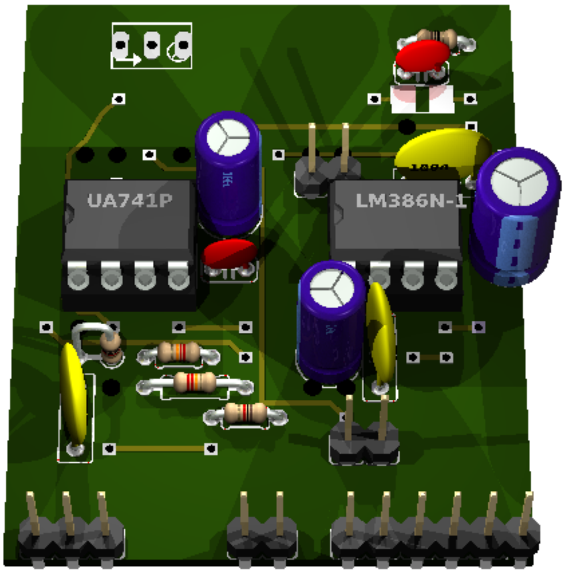
4.3 Partlist

Table 4.1:

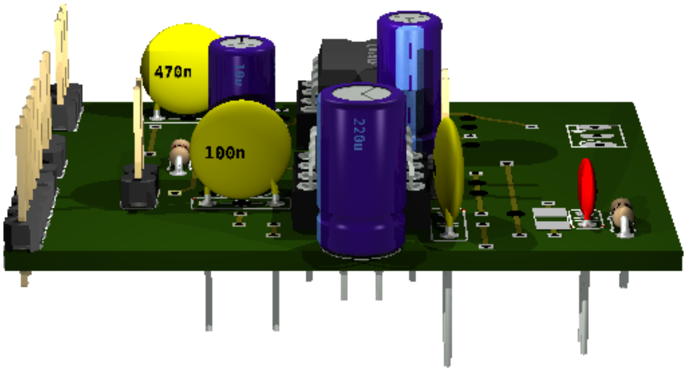
part	value	position
C1	47n	(1.45 1.7)
C2	220u	(1.7 1.05)
C4	10u	(1.45 1.6)
C5	100n	(1.5 1.3)
C6	100n	(1.3 0.7)
C7	100n	(0.85 1)
C8	100u	(0.85 1.2)
C9	470n	(0.4 0.5)
C10	10u	(1.15 0.7)
IC1	UA741P	(0.55 1.05)
IC2	LM386N-1	(1.35 1.05)
JP1		(1.15 1.3)
JP2		(1.55 0.1)
JP3		(0.95 0.1)
JP4		(1.25 0.4)
JP5		(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

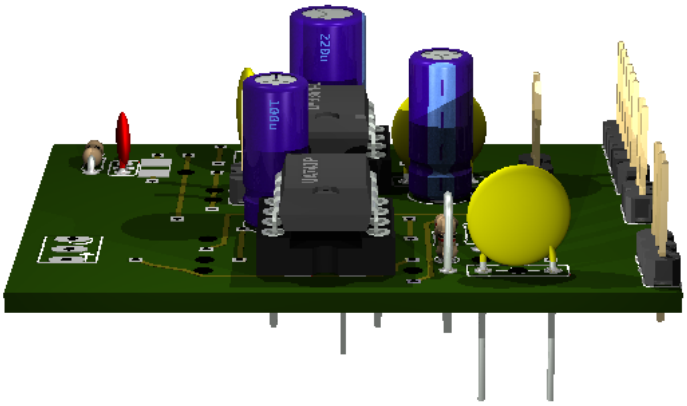
4.4.1 Front



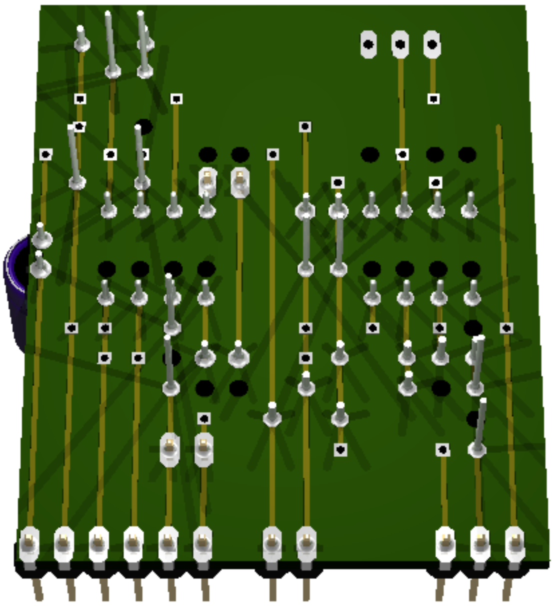
4.4.2 Right side



4.4.3 Left side



4.4.4 Bottom



4.5 Sources

original design

CHAPTER

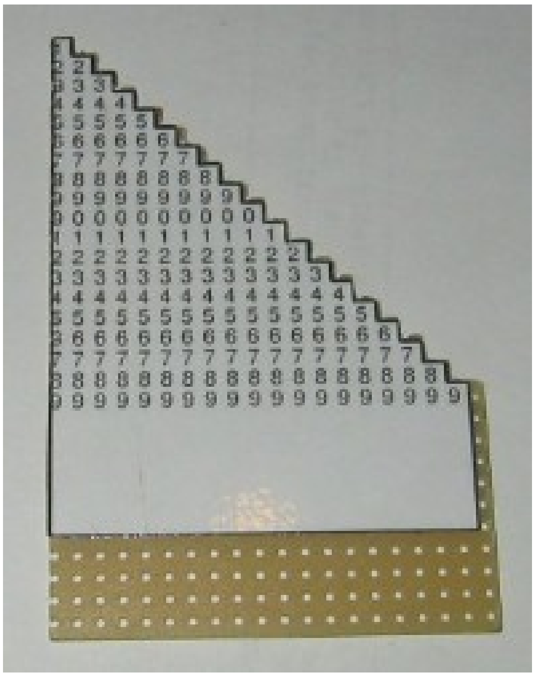
FIVE

WIRE BENDING TOOL

Status: OK

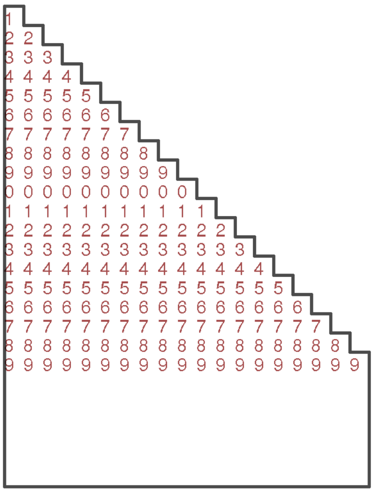
It is used for bending wires.

5.1 Image



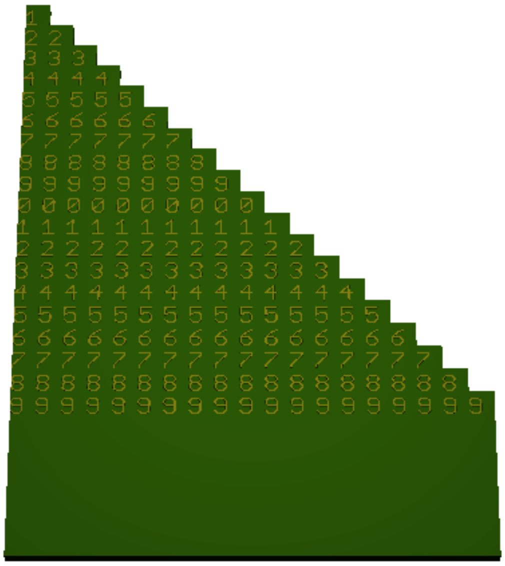
5.2 Board

Normal:



5.3 3D view

5.3.1 Front



5.3.2 Right side

CHAPTER

SIX

DAPA AVR PROGRAMMER

Status: OK

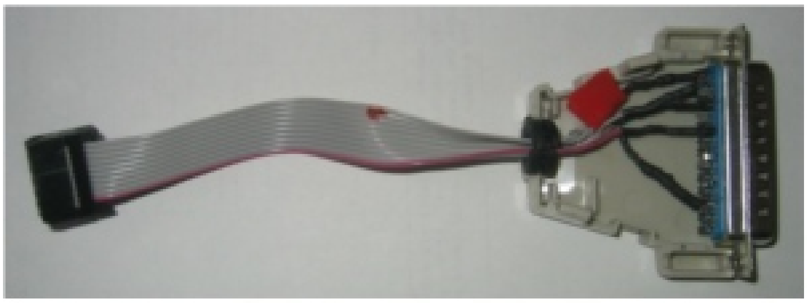
It is used for programming AVR controller and Arduino compatible boards using the parallel port.

6.1 Test on Ubuntu

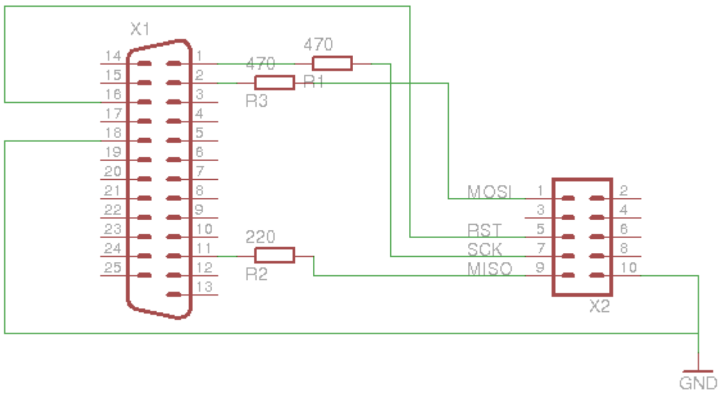
checking:

```
$ 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

ISP pinout

CHAPTER

SEVEN

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

7.1 Sources

RJ14 pinout

RJ14 wiring details

CHAPTER

EIGHT

GARMIN ETREX DATA CABLE

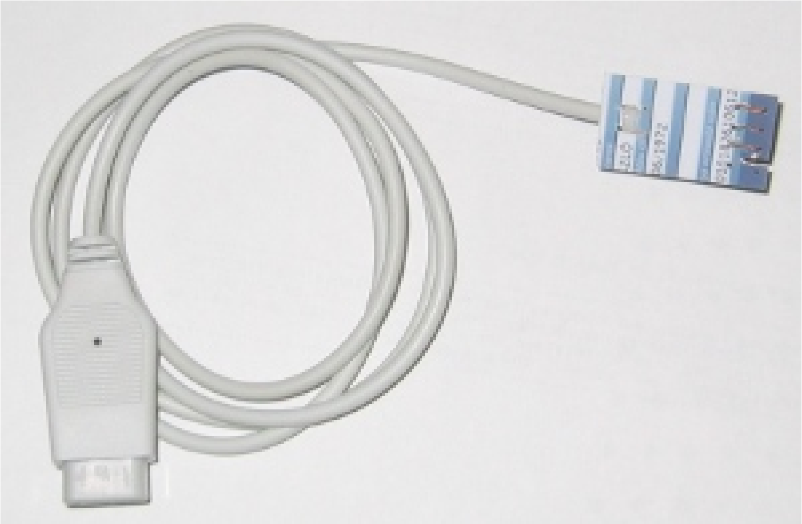
Status: OK

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)

8.1 Images



8.2 Sources

original design

CHAPTER

NINE

SERIAL PORT LOOPBACK

Status: OK

It is used for testing the serial port.

Connected pins:

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

9.1 Images

9.2 Sources

original design

Serial port pinout

CHAPTER

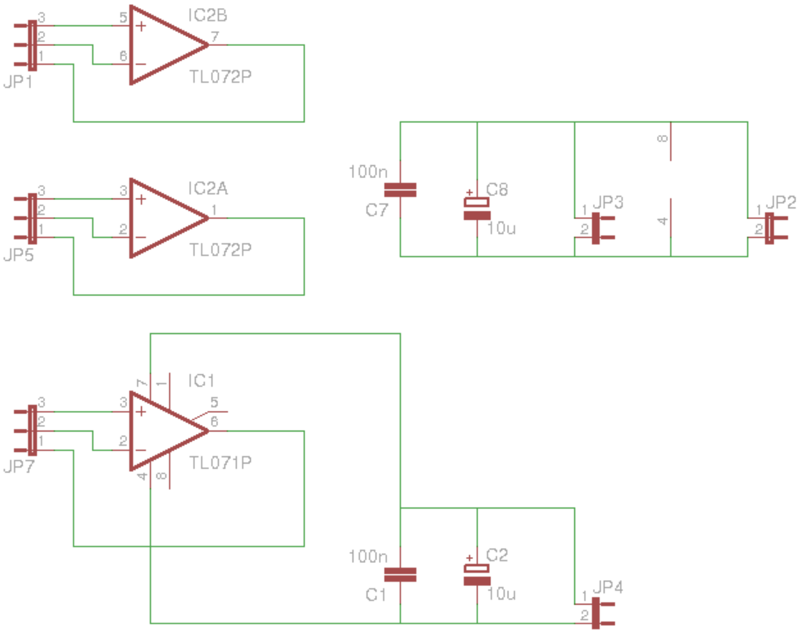
TEN

OP-AMP MODULE

Status: OK

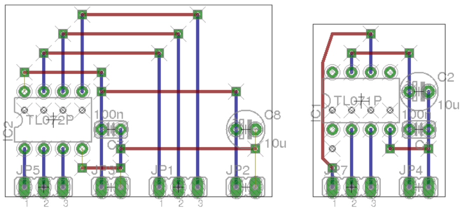
It is used for op-amps in breadboard.

10.1 Schematic

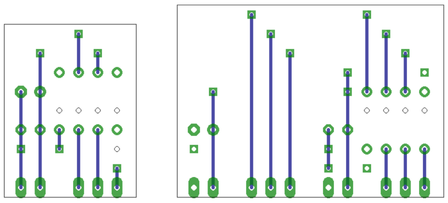


10.2 Board

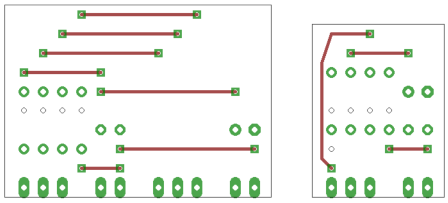
top



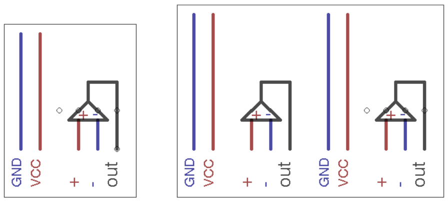
bottom mirrored



wires only



document



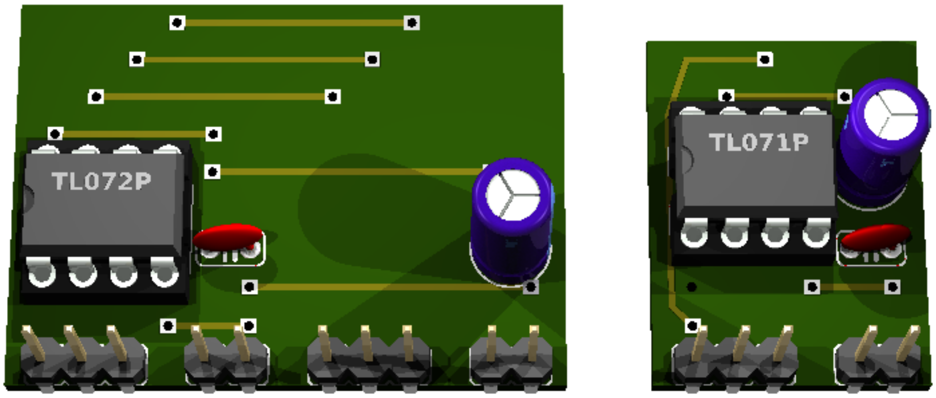
10.3 Partlist

Table 10.1:

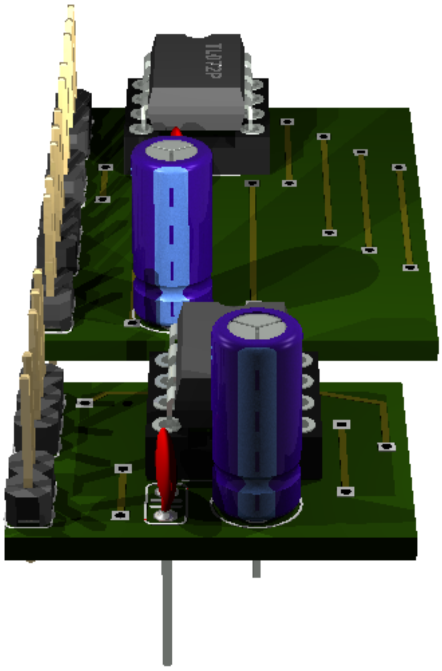
part	value	position
C1	100n	(2.35 0.4)
C2	10u	(2.35 0.6)
C7	100n	(0.75 0.4)
C8	10u	(1.45 0.4)
IC1	TL071P	(2.05 0.55)
IC2	TL072P	(0.45 0.45)
JP1		(1.1 0.1)
JP2		(1.45 0.1)
JP3		(0.75 0.1)
JP4		(2.35 0.1)
JP5		(0.4 0.1)
JP7		(2 0.1)

10.4 3D view

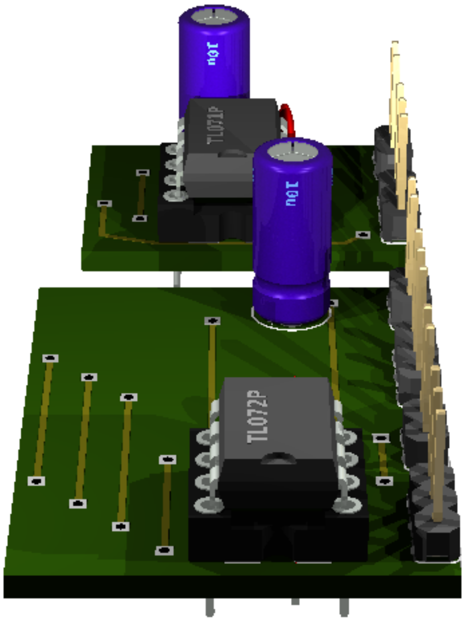
10.4.1 Front



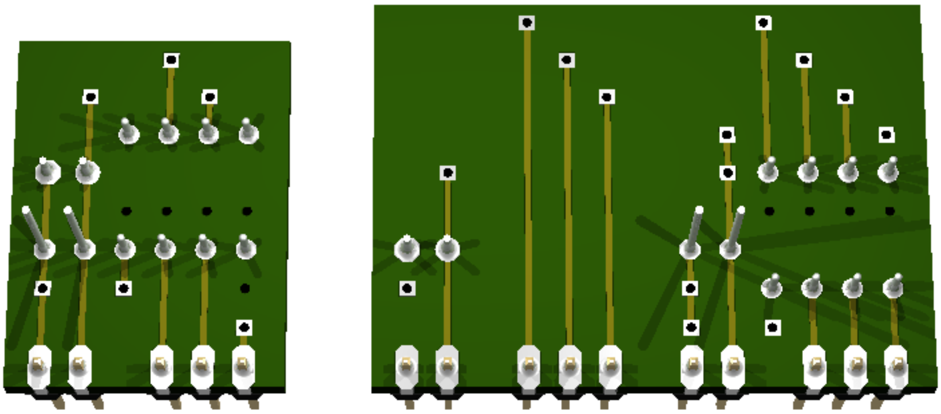
10.4.2 Right side



10.4.3 Left side



10.4.4 Bottom



CHAPTER

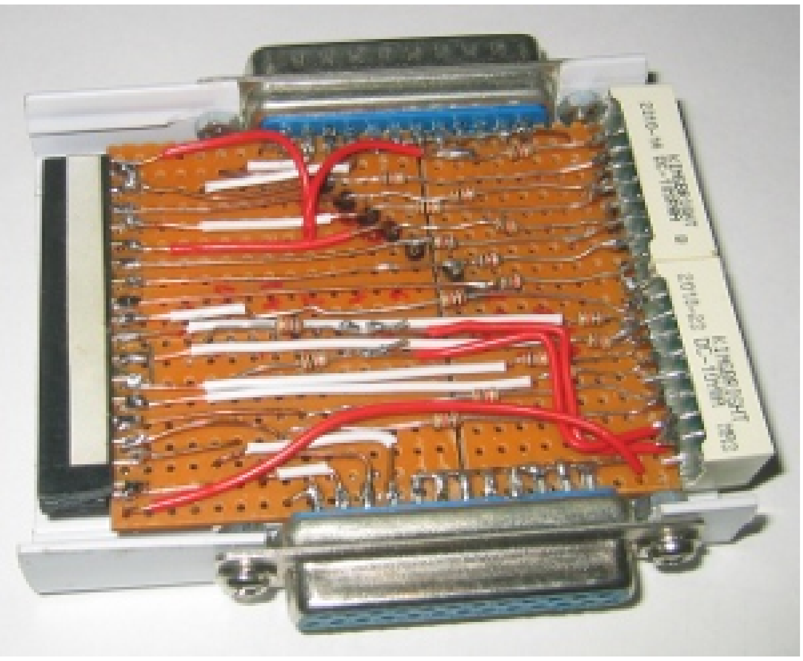
ELEVEN

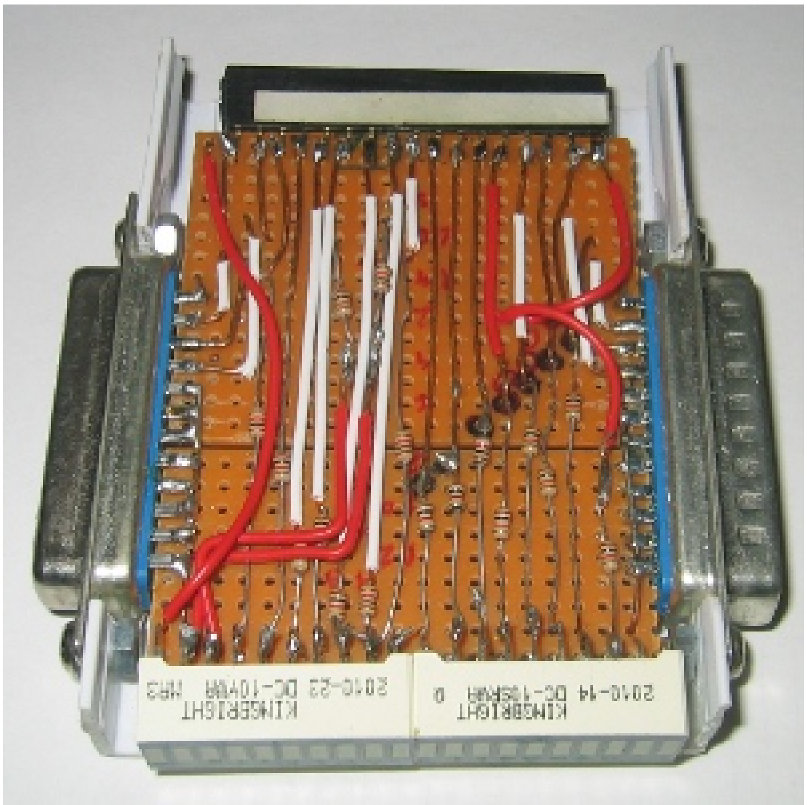
PARALLEL PORT MONITOR

Status: OK

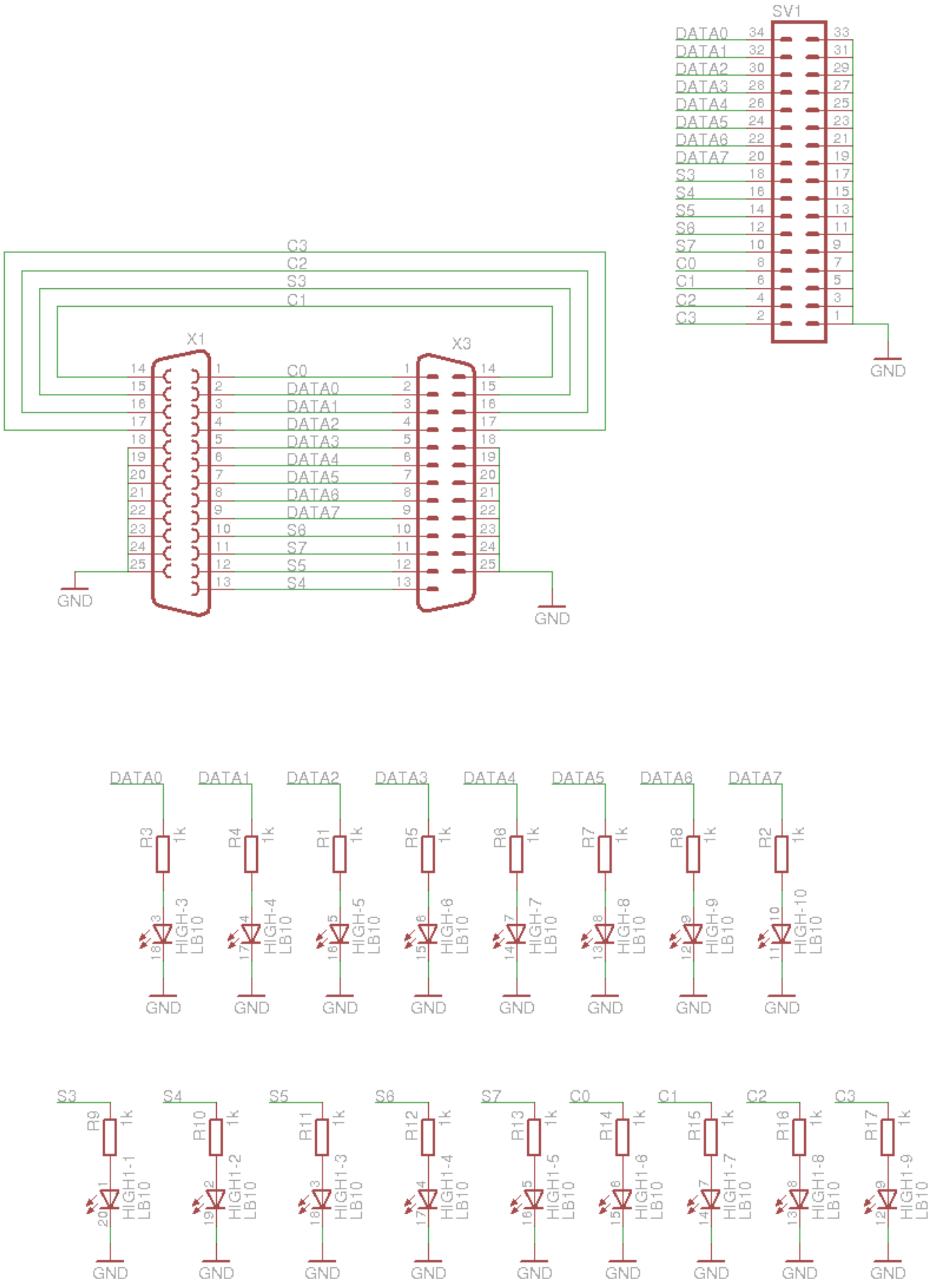
It is used for monitoring the parallel port signals.

11.1 Images



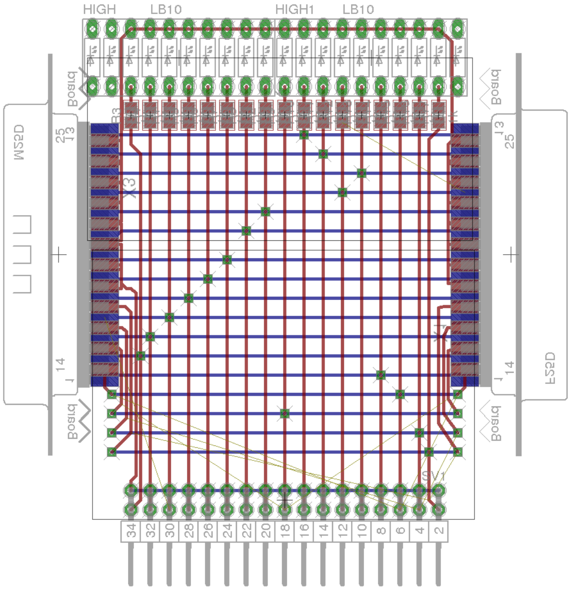

KINGBRIGHT
KINGBRIGHT

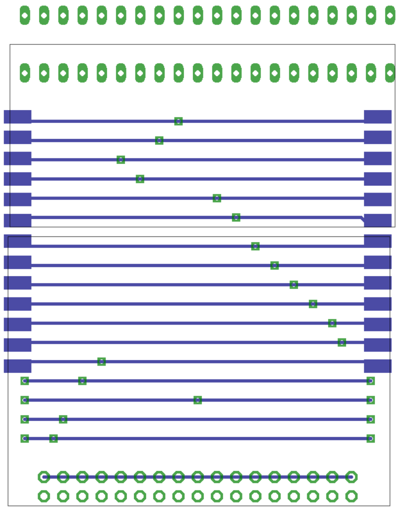
11.2 Schematic

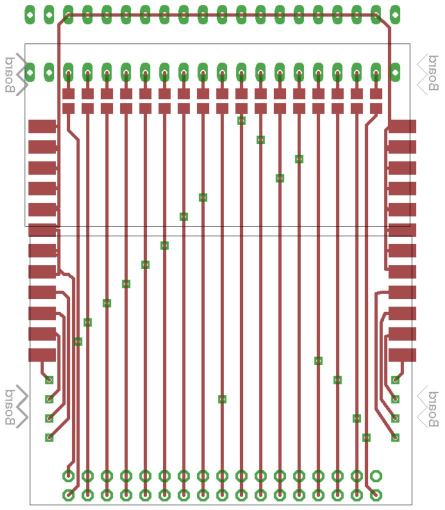

SV1
DATA0 34 33
DATA1 32 31
DATA2 30 29
DATA3 28 27
DATA4 26 25
DATA5 24 23
DATA6 22 21
DATA7 20 19
S3 18 17
S4 16 15
S5 14 13
S6 12 11
S7 10 9
C0 8 7
C1 6 5
C2 4 3
C3 2 1
GND
C3
C2
S3
C1
X1
X3
C0
DATA0
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
S6
S7
S5
S4
GND
GND
DATA0 DATA1 DATA2 DATA3 DATA4 DATA5 DATA6 DATA7
R3 1k R4 1k R1 1k R5 1k R6 1k R7 1k R8 1k R2 1k
HIGH-3 LB10 HIGH-4 LB10 HIGH-5 LB10 HIGH-6 LB10 HIGH-7 LB10 HIGH-8 LB10 HIGH-9 LB10 HIGH-10 LB10
GND GND GND GND GND GND GND GND
S3 S4 S5 S6 S7 C0 C1 C2 C3
R9 1k R10 1k R11 1k R12 1k R13 1k R14 1k R15 1k R16 1k R17 1k
HIGH1-1 LB10 HIGH1-2 LB10 HIGH1-3 LB10 HIGH1-4 LB10 HIGH1-5 LB10 HIGH1-6 LB10 HIGH1-7 LB10 HIGH1-8 LB10 HIGH1-9 LB10
GND GND GND GND GND GND GND GND GND

11.3 Board

Normal, bottom mirrored, wires only:







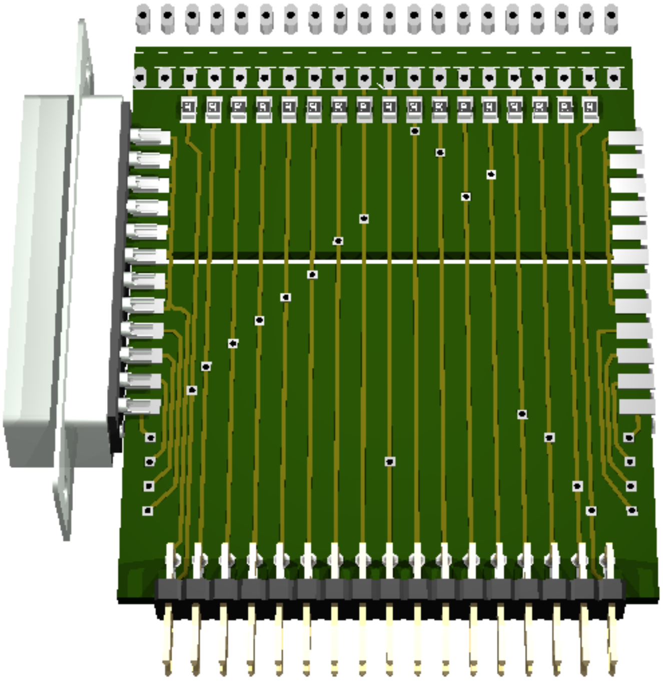
11.4 Partlist

Table 11.1:

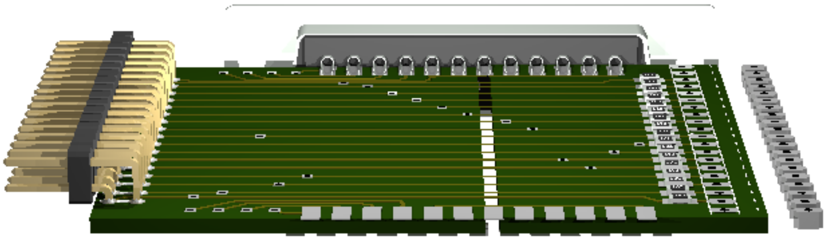
part	value	position
HIGH	LB10	(0.45 2.55)
HIGH1	LB10	(1.45 2.55)
R1	1k	(0.4 2.25)
R2	1k	(0.9 2.25)
R3	1k	(0.2 2.25)
R4	1k	(0.3 2.25)
R5	1k	(0.5 2.25)
R6	1k	(0.6 2.25)
R7	1k	(0.7 2.25)
R8	1k	(0.8 2.25)
R9	1k	(1 2.25)
R10	1k	(1.1 2.25)
R11	1k	(1.2 2.25)
R12	1k	(1.3 2.25)
R13	1k	(1.4 2.25)
R14	1k	(1.5 2.25)
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)

11.5 3D view

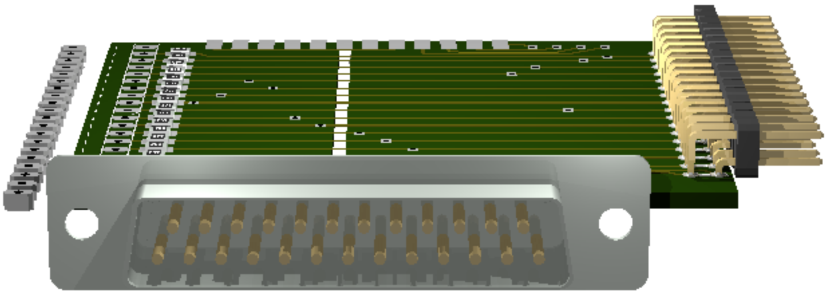
11.5.1 Front



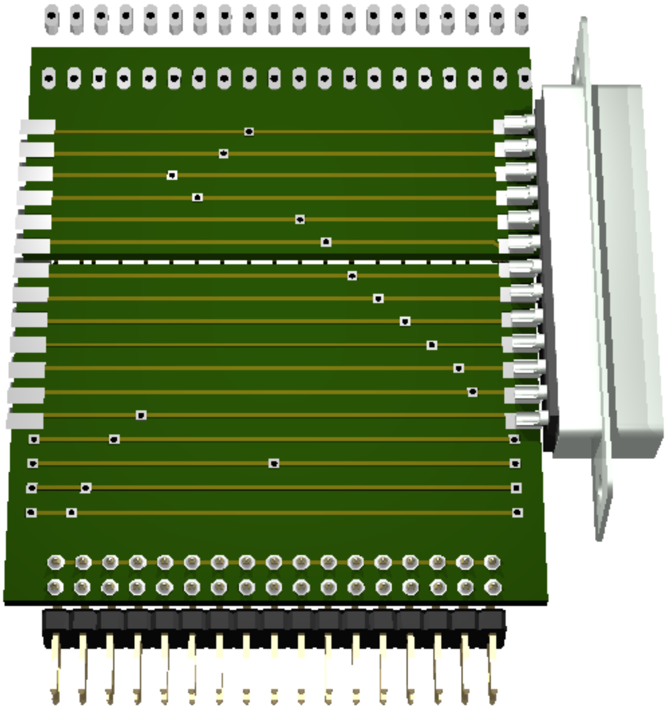
11.5.2 Right side



11.5.3 Left side



11.5.4 Bottom



11.6 Sources

original idea

CHAPTER

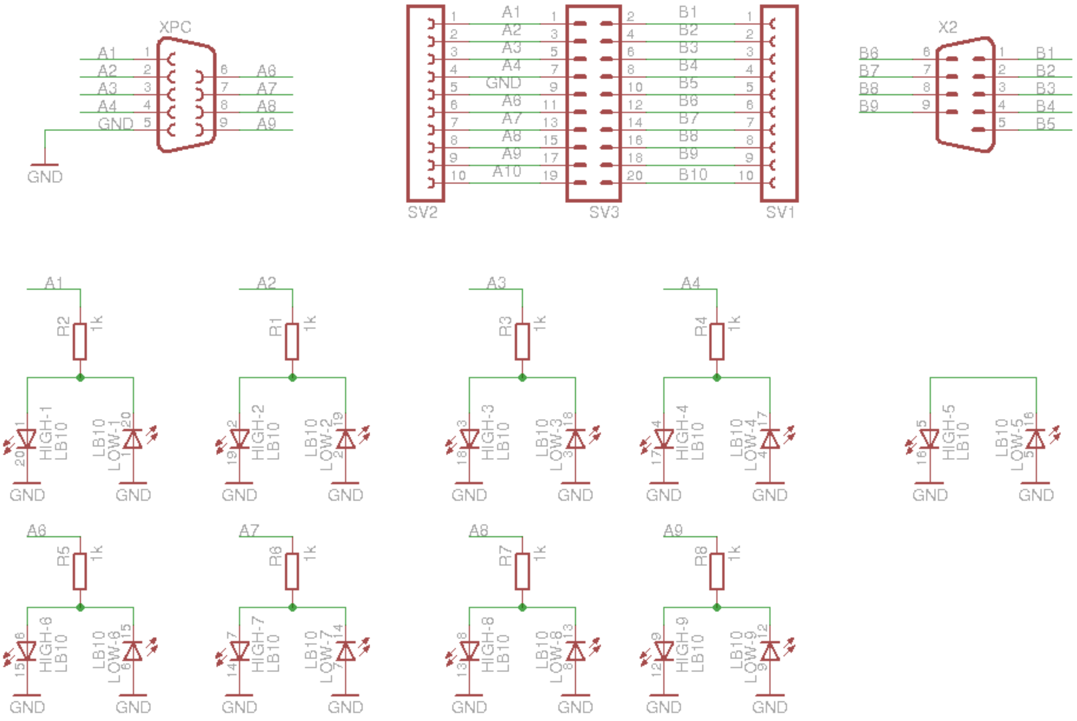
TWELVE

SERIAL PORT MONITOR

Status: OK

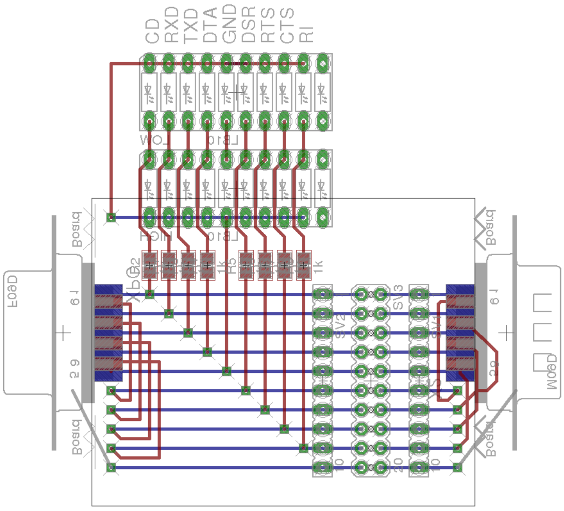
On each signal there is one LED for positive and one LED for negative voltage. It is easy to change connections or connect external parts. Examples: Loop-Back, Null Modem,..

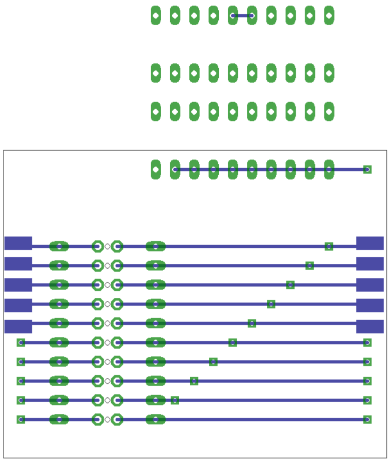
12.1 Schematic

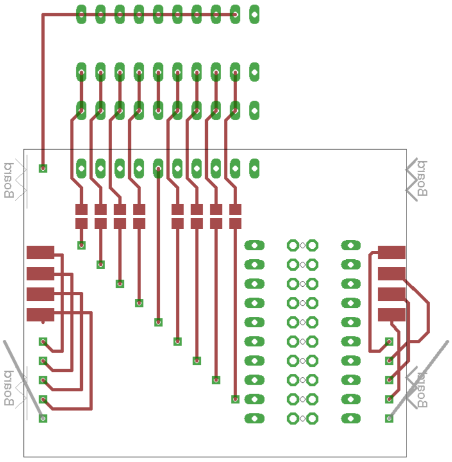


12.2 Board

Normal, bottom mirrored, wires only:







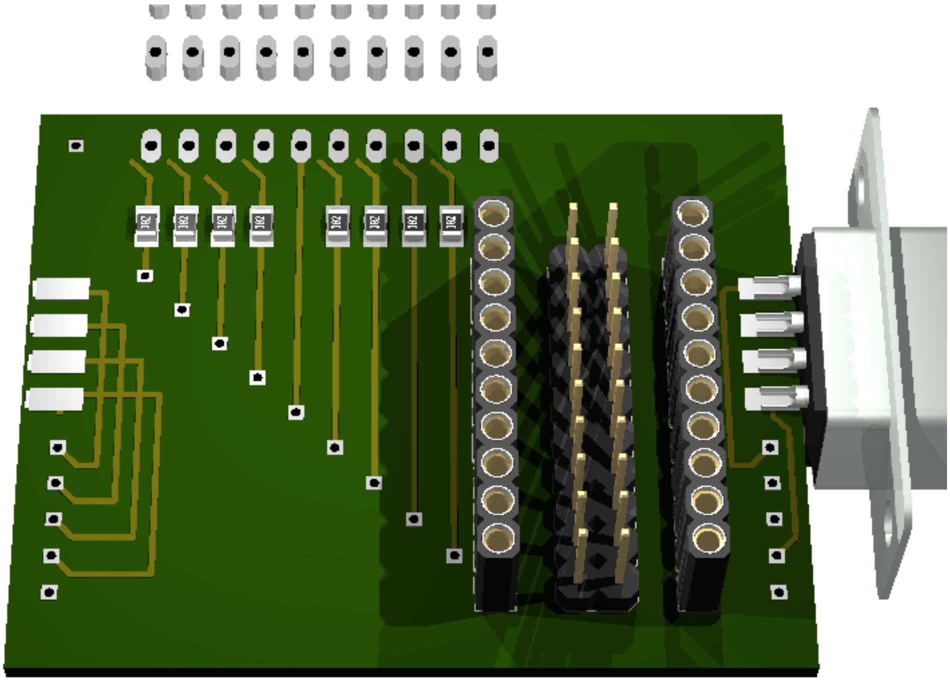
12.3 Partlist

Table 12.1:

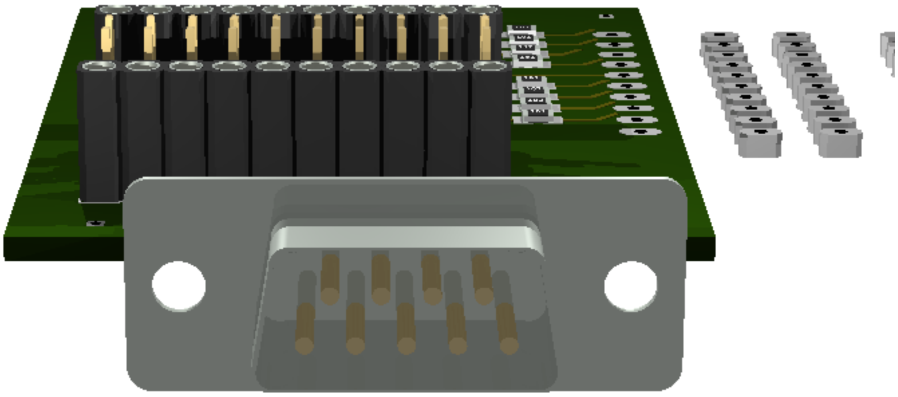
part	value	position
HIGH	LB10	(0.75 1.65)
LOW	LB10	(0.75 2.15)
R1	1k	(0.4 1.25)
R2	1k	(0.3 1.25)
R3	1k	(0.5 1.25)
R4	1k	(0.6 1.25)
R5	1k	(0.8 1.25)
R6	1k	(0.9 1.25)
R7	1k	(1 1.25)
R8	1k	(1.1 1.25)
SV1		(1.7 0.65)
SV2		(1.2 0.65)
SV3		(1.45 0.65)
X2		(2.15 0.9)
XPC		(-0.15 0.9)

12.4 3D view

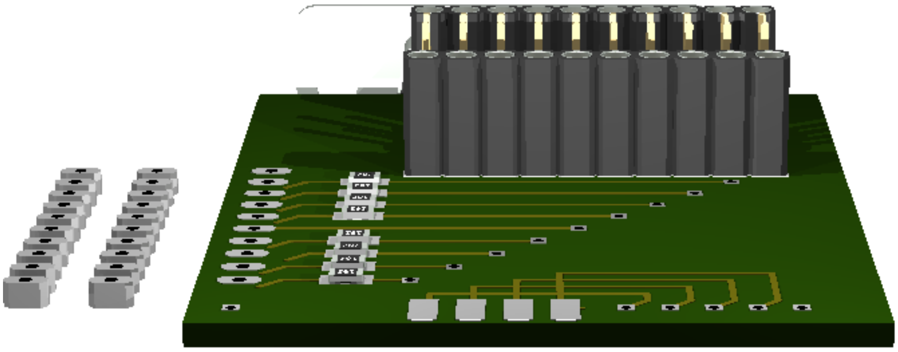
12.4.1 Front



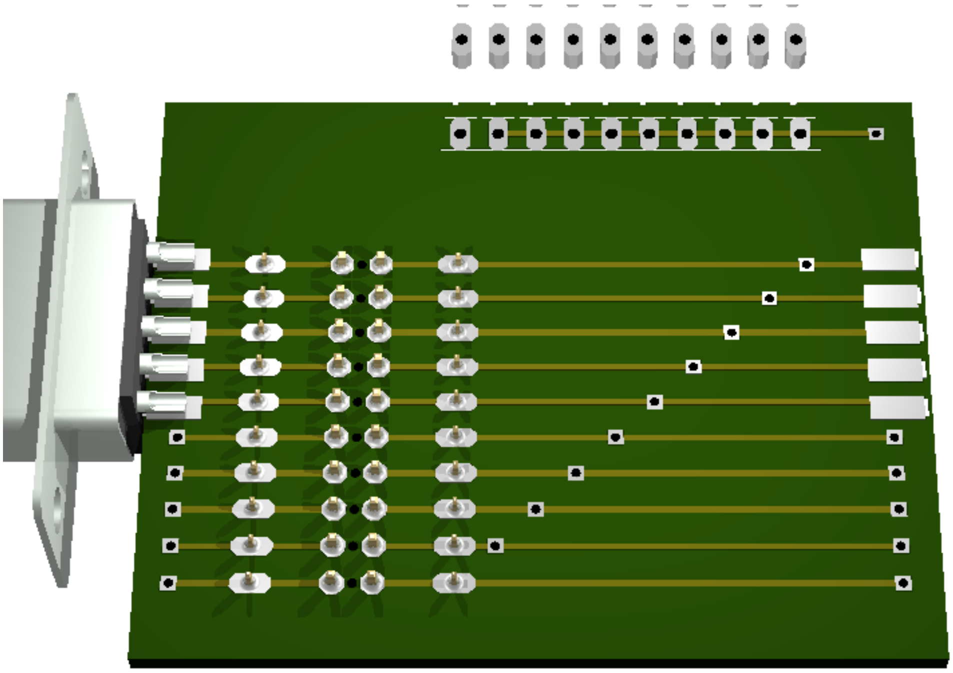
12.4.2 Right side



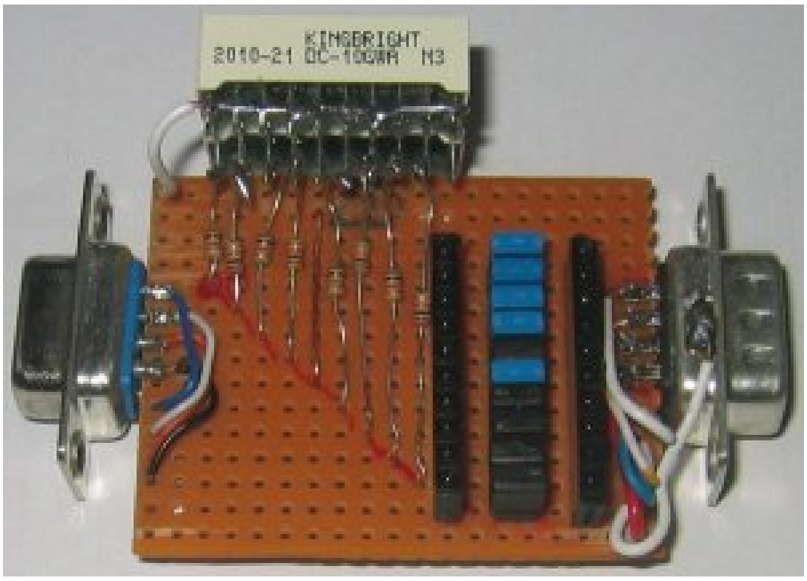
12.4.3 Left side

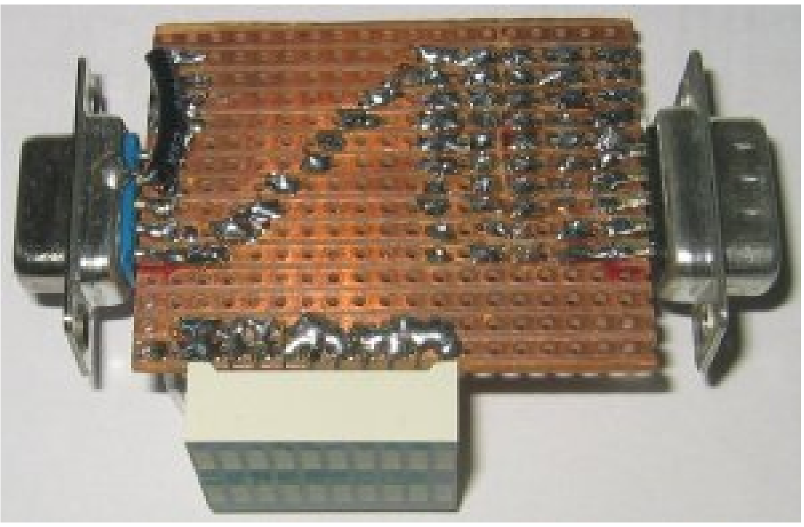


12.4.4 Bottom



12.5 Images





CHAPTER

THIRTEEN

STK200 AVR PROGRAMMER

Status: OK

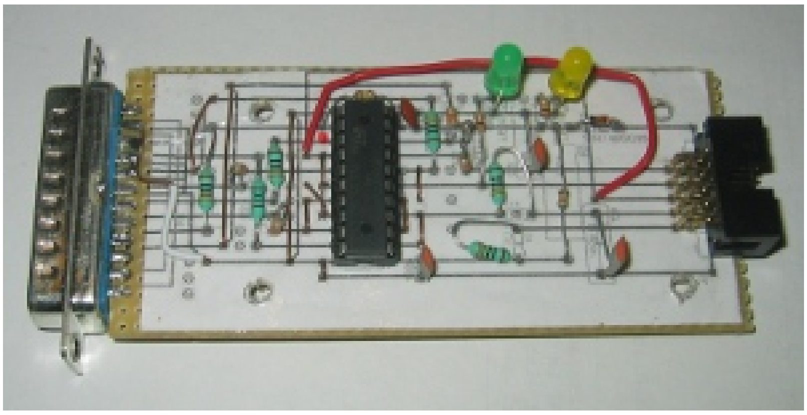
It is used for programming AVR controller and Arduino compatible boards using the parallel port.

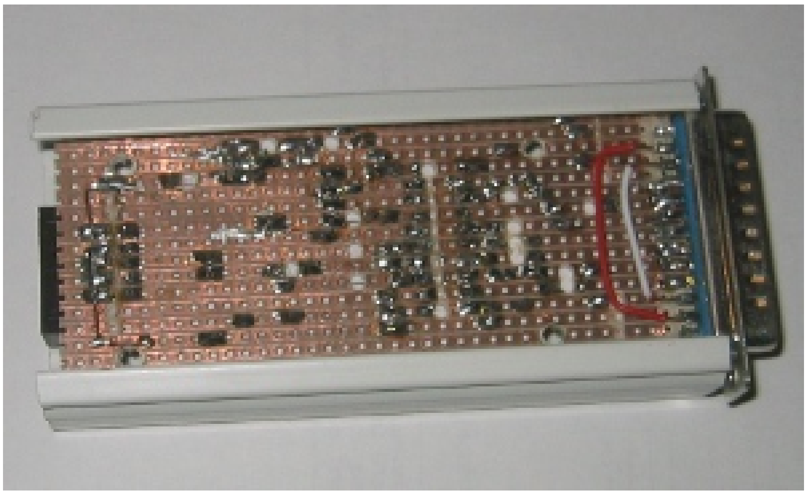
13.1 Test on Ubuntu

checking:

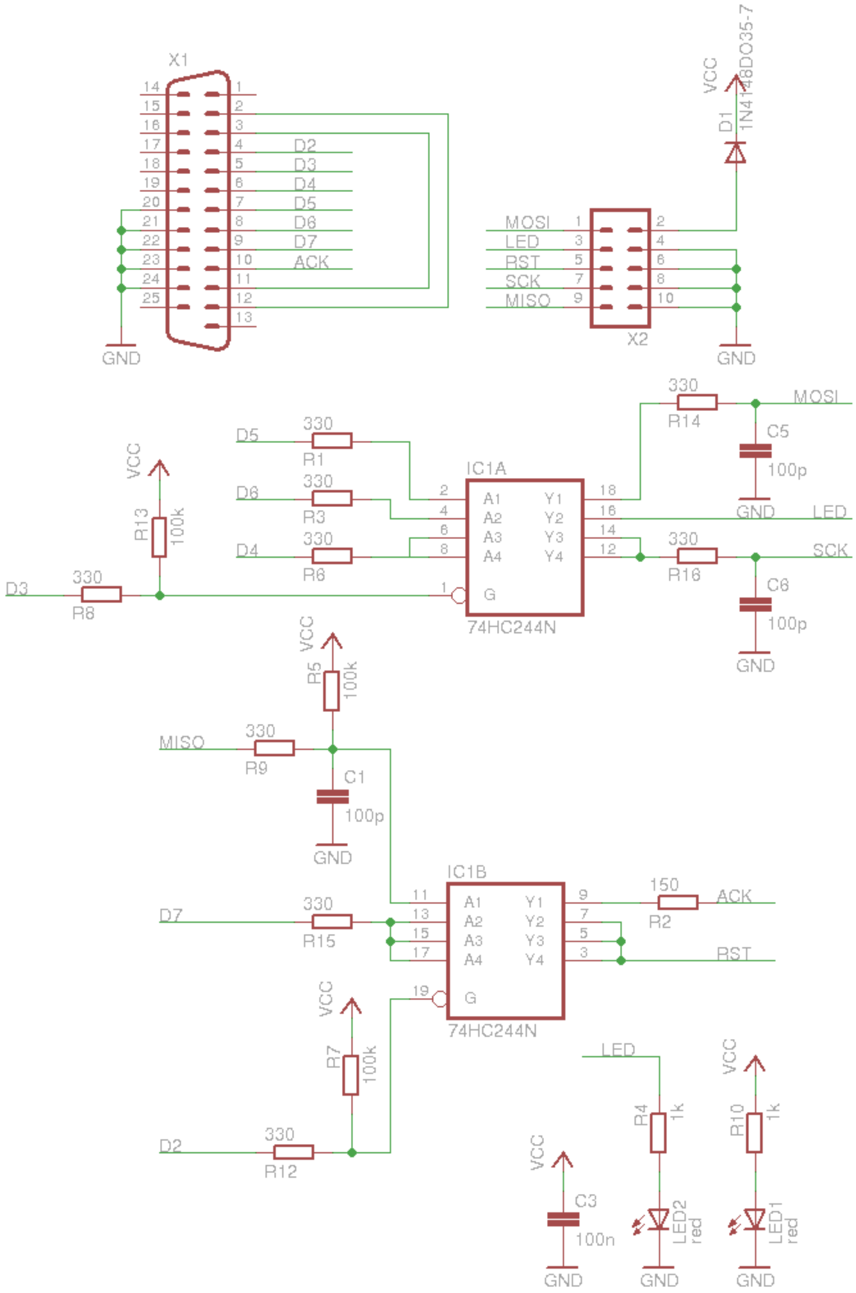
```
$ avrdude -patmega88 -cstk200

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.
```

13.2 Image



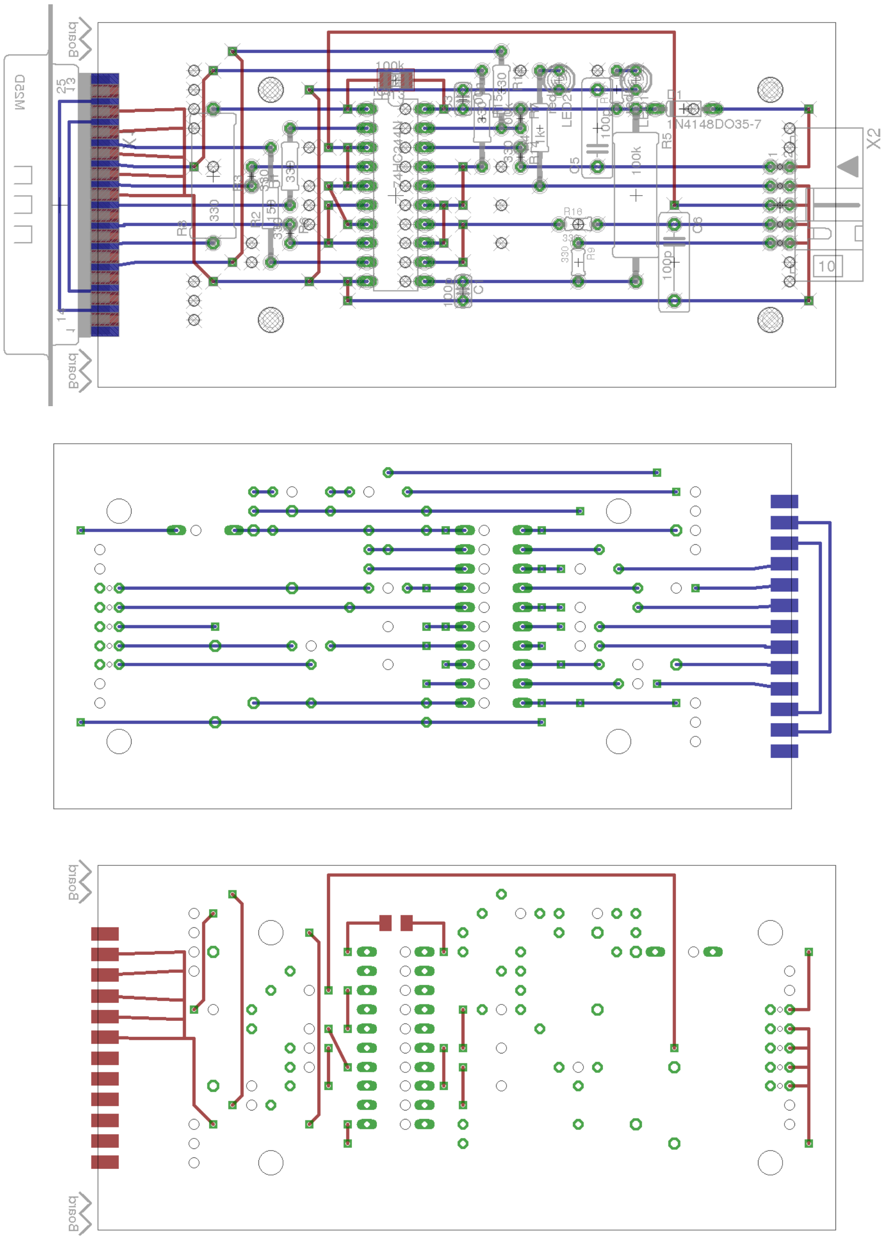


13.3 Schematic



13.4 Board

Normal, bottom mirrored, wires only:



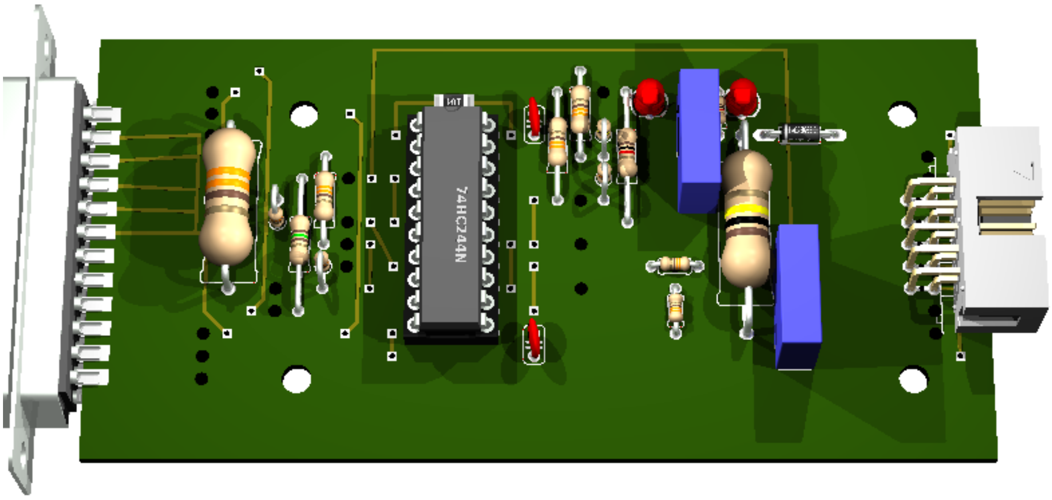
13.5 Partlist

Table 13.1:

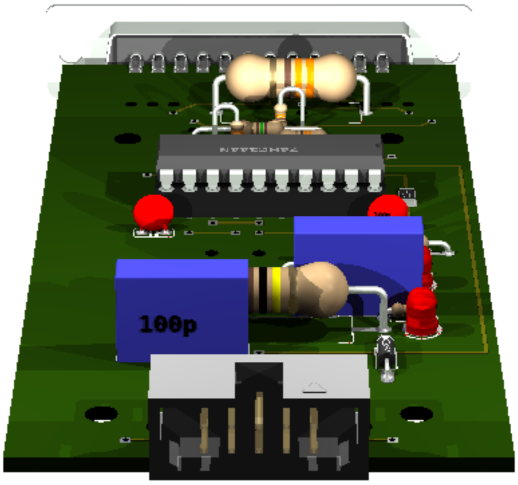
part	value	position
C1	100p	(2.3 0.95)
C3	100n	(2.3 1.95)
C5	100p	(3 1.8)
C6	100p	(3.4 1.1)
D1	1N4148DO35-7	(3.45 1.9)
IC1	74HC244N	(1.95 1.45)
LED1	red	(3.2 2.05)
LED2	red	(2.8 2.05)
R1	330	(1.4 1.6)
R2	150	(1.3 1.4)
R3	330	(1.2 1.55)
R4	1k	(2.7 1.8)
R5	100k	(3.2 1.45)
R6	330	(1.4 1.25)
R7	100k	(2.6 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)

13.6 3D view

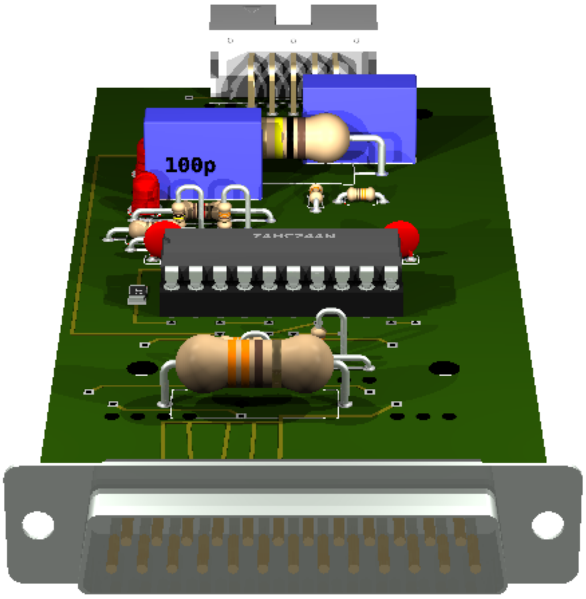
13.6.1 Front



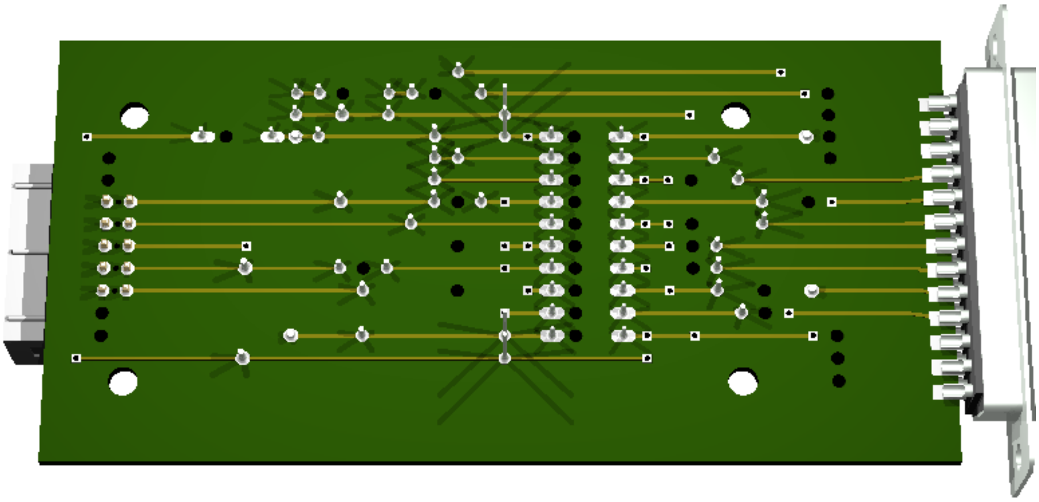
13.6.2 Right side



13.6.3 Left side



13.6.4 Bottom



13.7 Sources

original design

Parallel port specification

ISP pinout

similar designs:

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

CHAPTER

FOURTEEN

USB1WIRE

Status: OK

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

Based on V-USB hardware.

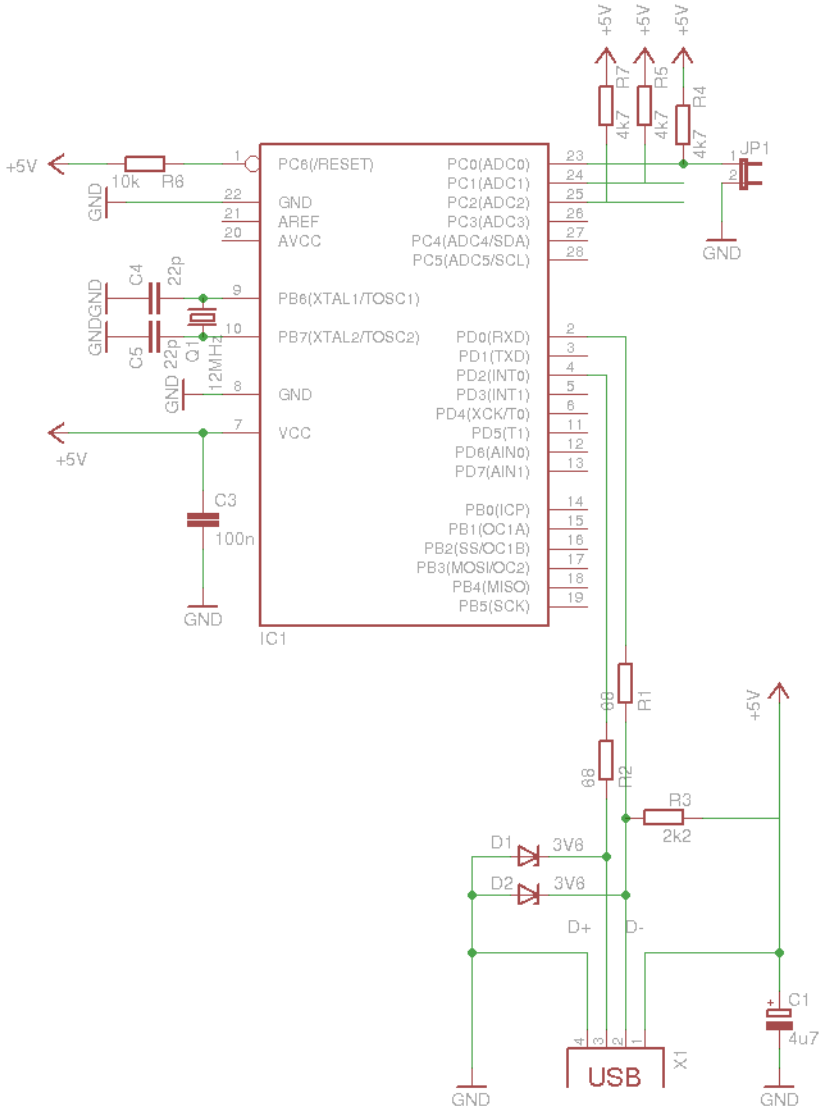
connections:

function	AVR pin	Arduino pin
1wire	PC0	A0
1wire	PC1	A1
1wire	PC2	A2
USB D-	PD0	D0
USB D+	PD2	D2

V-USB defines:

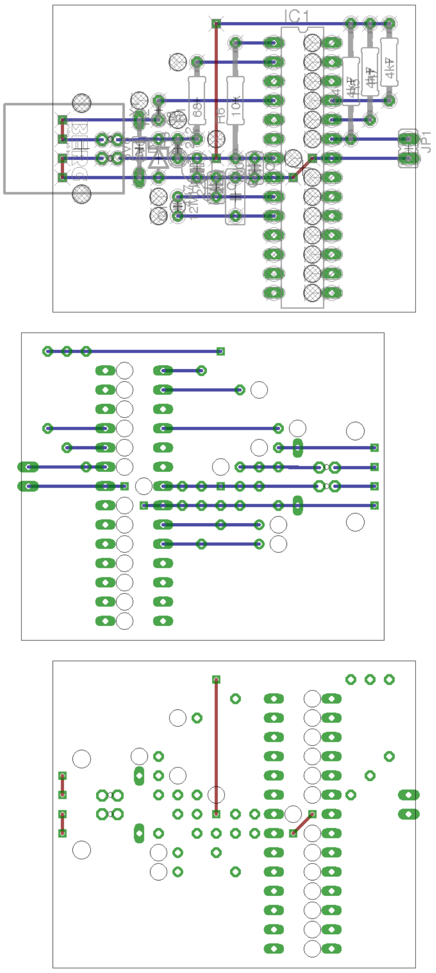
```
#define USB_CFG_IOPORTNAME      D
#define USB_CFG_DMINUS_BIT      0
#define USB_CFG_DPLUS_BIT       2
```

14.1 Schematic



14.2 Board

Normal, bottom mirrored, wires only:



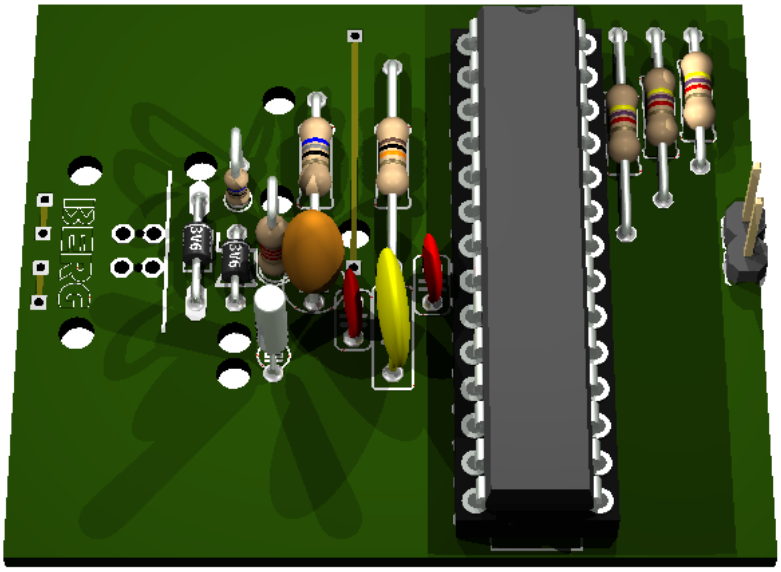
14.3 Partlist

Table 14.1:

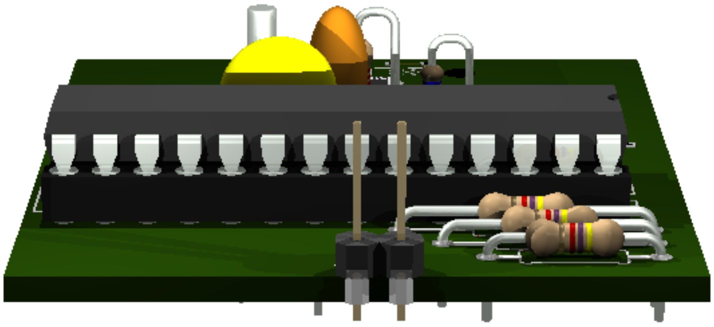
part	value	position
C1	4u7	(1.4 0.85)
C3	100n	(1.7 0.85)
C4	22p	(1.5 0.75)
C5	22p	(1.6 0.7)
D1	3V6	(1.1 0.95)
D2	3V6	(1.2 0.9)
IC1		(1.95 0.85)
JP1		(2.5 0.95)
Q1	12MHz	(1.3 0.65)
R1	68	(1.4 1.2)
R2	68	(1.2 1.15)
R3	2k2	(1.3 0.95)
R4	4k7	(2.2 1.3)
R5	4k7	(2.3 1.35)
R6	10k	(1.6 1.2)
R7	4k7	(2.4 1.4)
X1		(0.8 0.95)

14.4 3D view

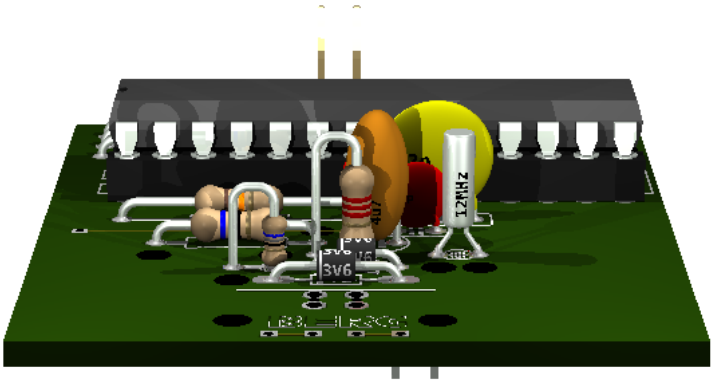
14.4.1 Front



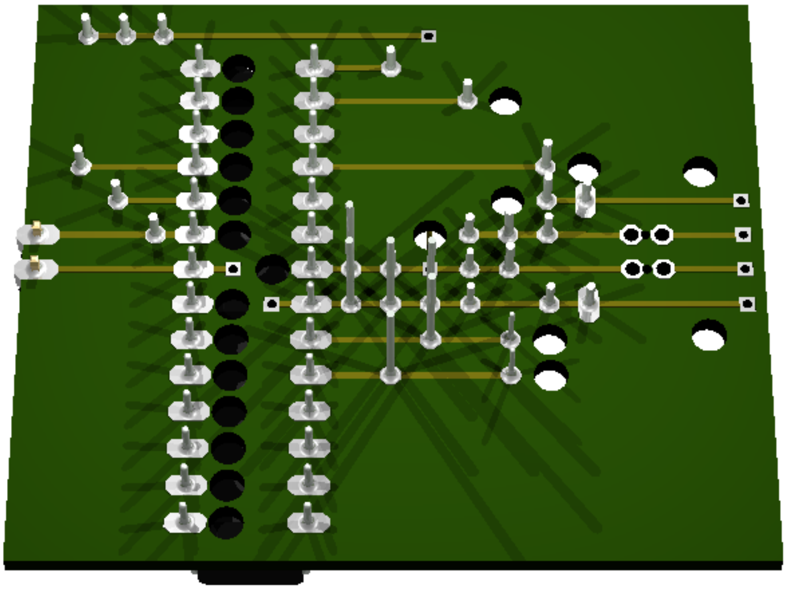
14.4.2 Right side



14.4.3 Left side



14.4.4 Bottom



CHAPTER

FIFTEEN

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)

15.1 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.”

15.2 Makefile

Tested with atmega88. Makefile settings:

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

15.3 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 r
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:

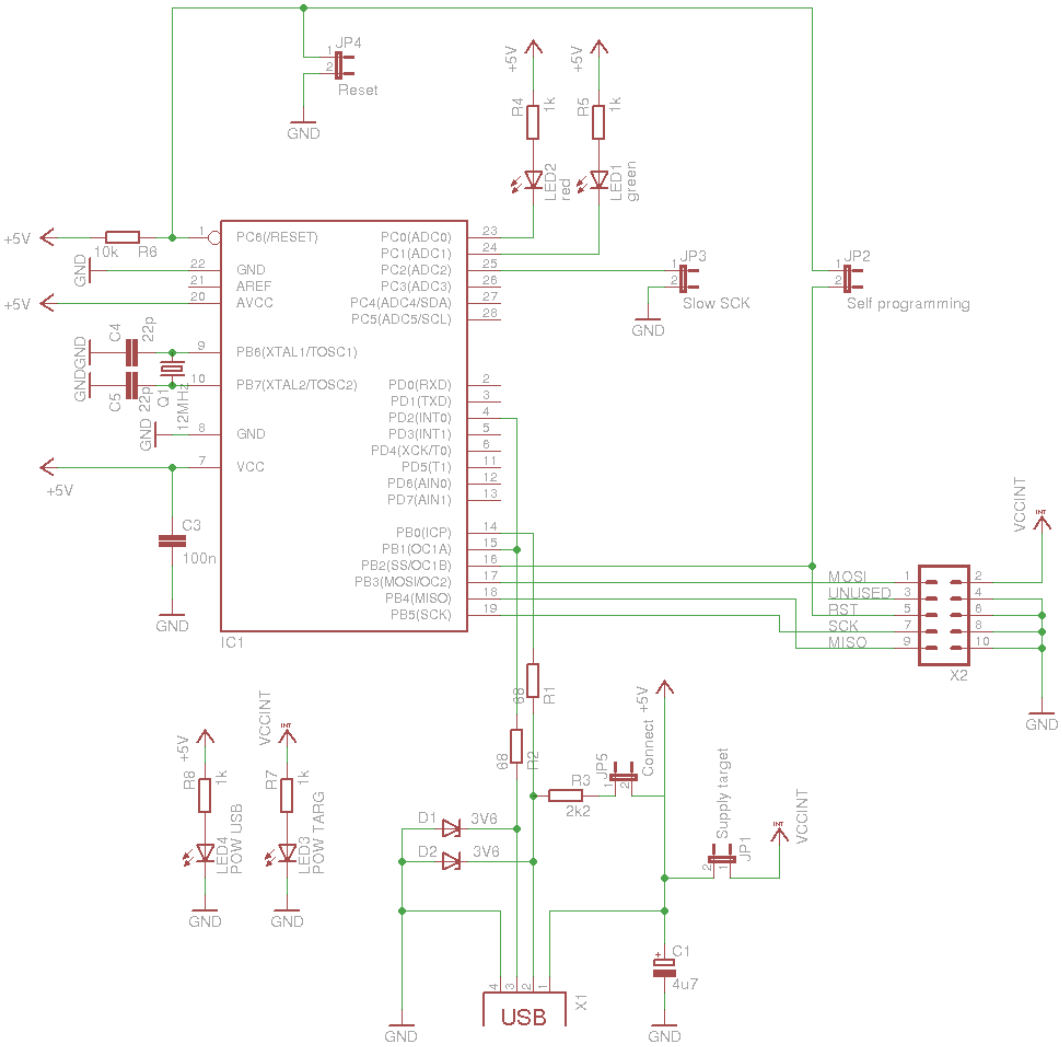
```
$ ls -l /dev/bus/usb/003/006
crw-rw-rw- 1 root users 189, 261 2011-11-05 10:33 /dev/bus/usb/003/006

$ 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:

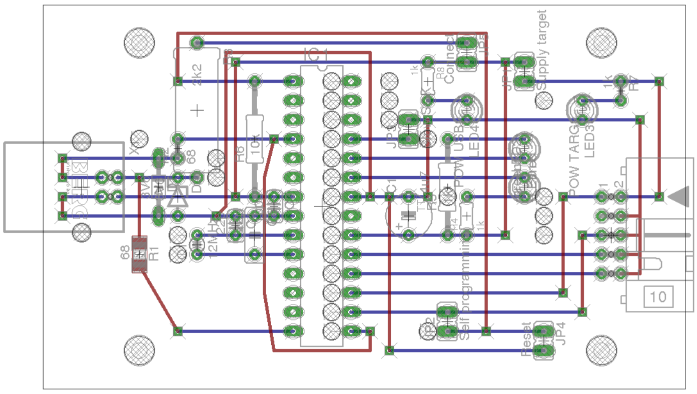
```
$ avrdude -patmega88 -cusbasp

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.
```

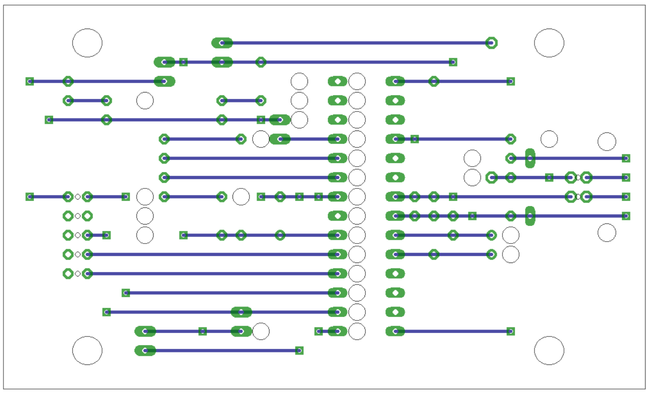
15.4 Schematic

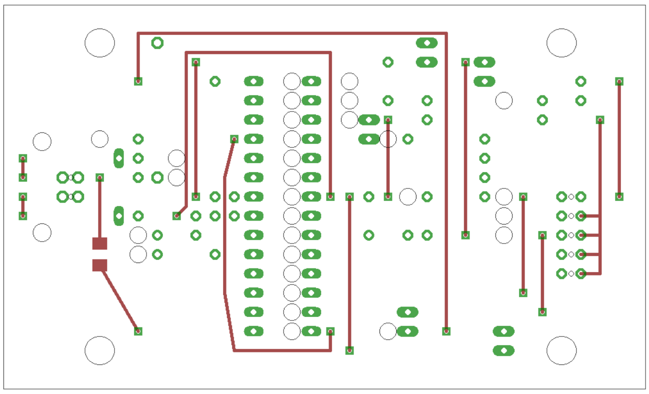


15.5 Board

Normal, bottom mirrored, wires only:







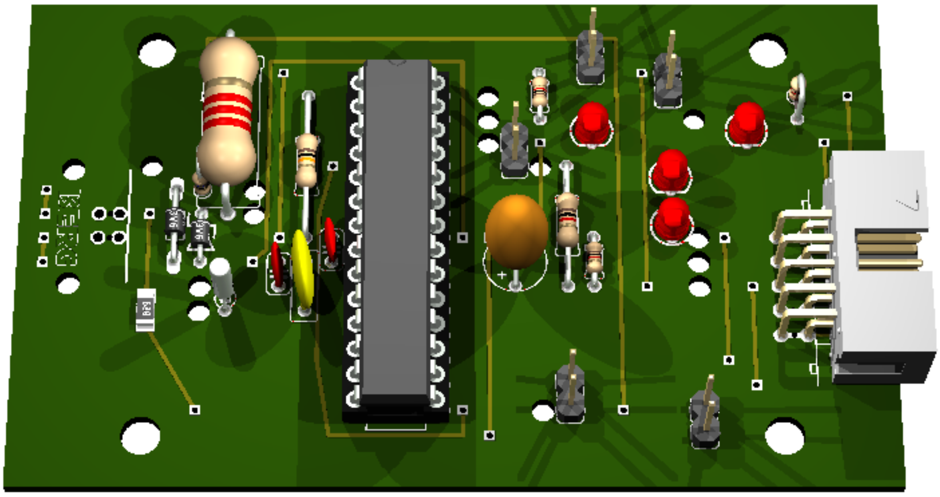
15.6 Partlist

Table 15.1:

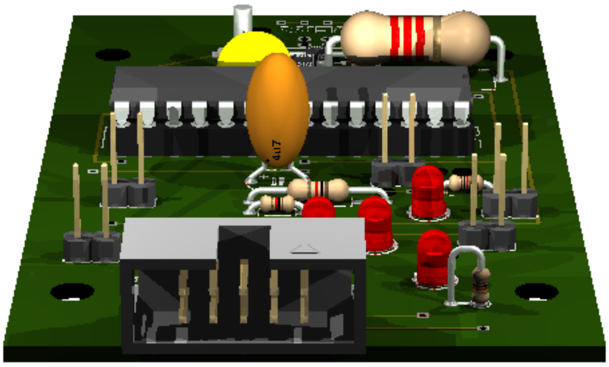
part	value	position
C1	4u7	(2.4 0.8)
C3	100n	(1.7 0.85)
C4	22p	(1.5 0.75)
C5	22p	(1.6 0.7)
D1	3V6	(1.1 0.95)
D2	3V6	(1.2 0.9)
IC1		(1.95 0.85)
JP1	Supply target	(3 1.55)
JP2	Self programming	(2.6 0.25)
JP3	Slow SCK	(2.4 1.25)
JP4	Reset	(3.1 0.15)
JP5	Connect	(2.7 1.65)
LED1	green	(3 1.15)
LED2	red	(3 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)
X1		(0.7 0.95)
X2		(3.45 0.7)

15.7 3D view

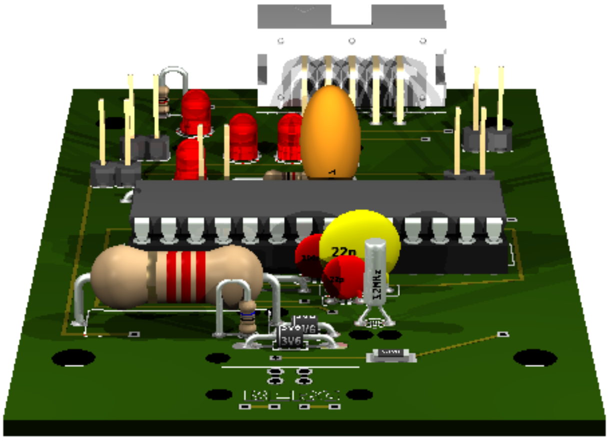
15.7.1 Front



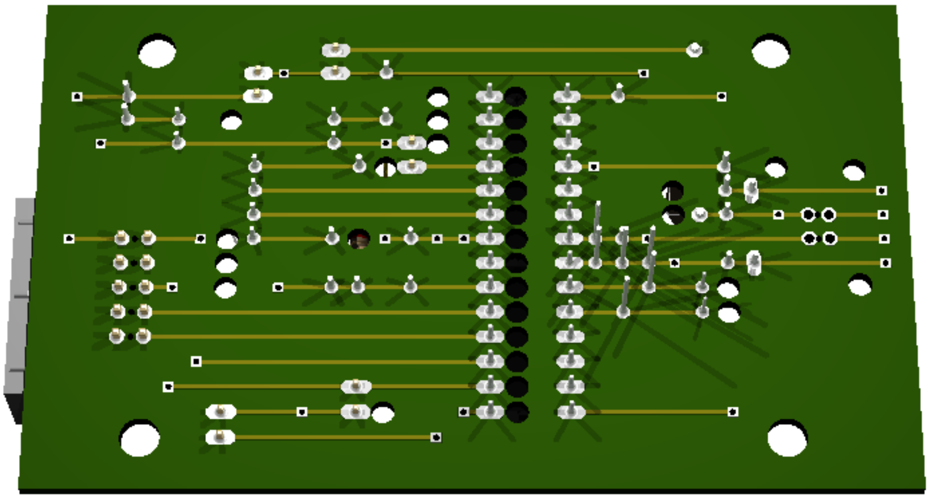
15.7.2 Right side



15.7.3 Left side



15.7.4 Bottom



15.8 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)
```

15.9 Sources

original design

ISP pinout

similar projects:

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

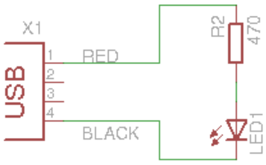
CHAPTER

SIXTEEN

USB LED

Status: OK

It is used for testing USB power.



CHAPTER

SEVENTEEN

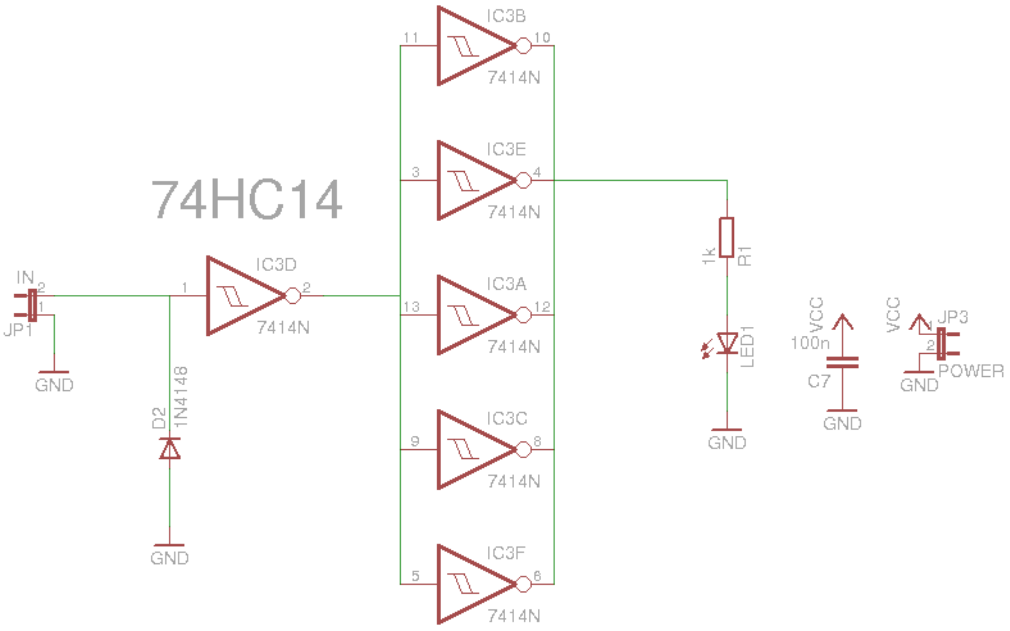
WIRE DETECTOR

Status: OK

It is used for detecting mains wire.

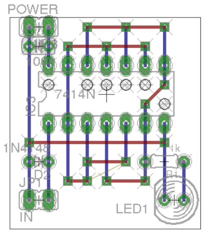
Based on this design: http://www.edn.com/article/511304-Detect_live_ac_mains_lines.php

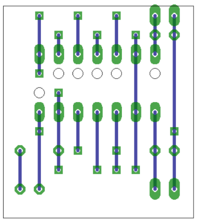
17.1 Schematic

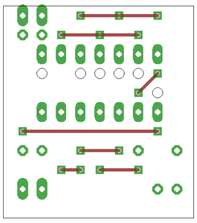


17.2 Board

Normal, bottom mirrored, wires only:







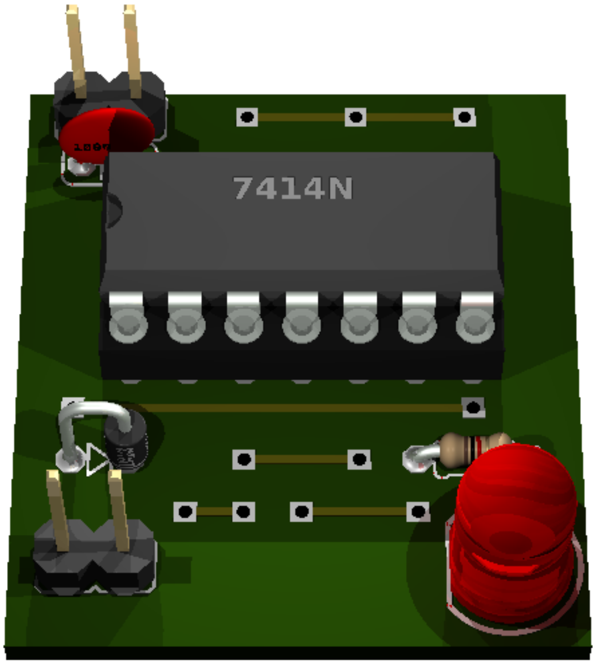
17.3 Partlist

Table 17.1:

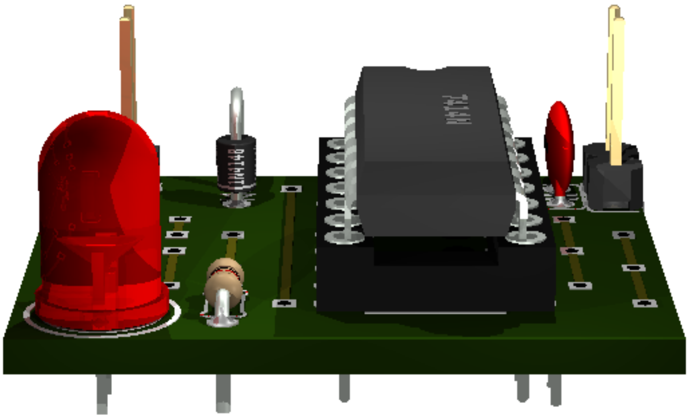
part	value	position
C7	100n	(0.85 1.9)
D2	1N4148	(0.85 1.3)
IC3	7414N	(1.2 1.65)
JP1	IN	(0.85 1.1)
JP3	POWER	(0.85 2)
LED1		(1.55 1.1)
R1	1k	(1.5 1.3)

17.4 3D view

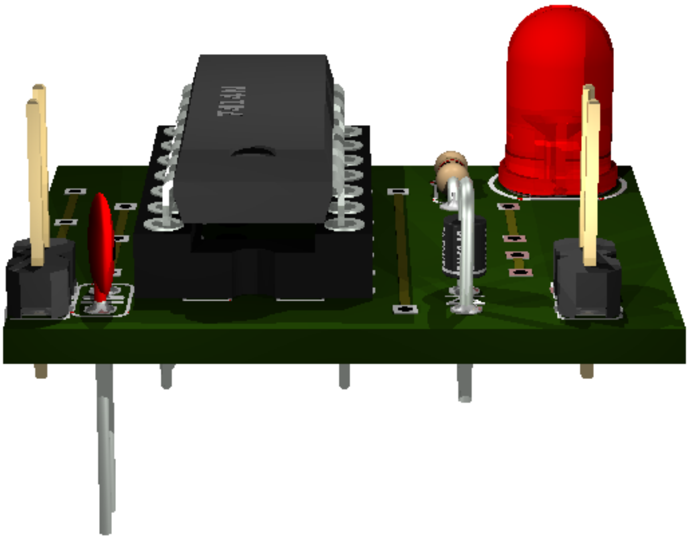
17.4.1 Front



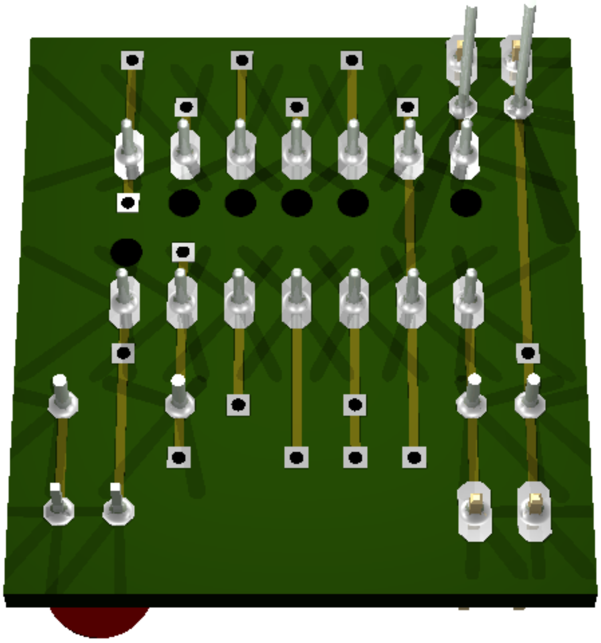
17.4.2 Right side



17.4.3 Left side



17.4.4 Bottom



17.5 Sources

original design