Decoder with FPGA for miniADSB receiver

Preliminary Manual

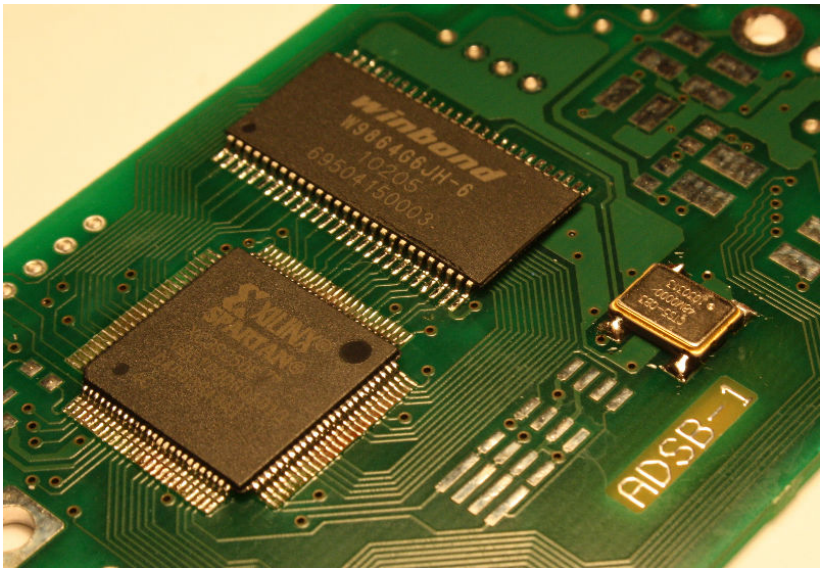

Table of Contents

1. ADS-B receiver overview

Complete system for receiving ADS-B consists of several parts:

- miniADSB receiver (miniadsb.web99.de)
- comparator (LM393 or similar), RS-485 transmitter (75176), 4V voltage source (transistor)
- FPGA decoder board
- A software in PC

Only the third (highlighted in green) part is matter of kit of decoder with FPGA.

1.1 miniADSB

Properly assembled and working miniADSB receiver has analog output in range 0,5 – 2 V. Follow instructions on web and forum of miniADSB receiver (miniadsb.web99.de). Oscilloscope is a need!

1.2 Data flow

A simplified diagram of data flow is on *Image 1*.

Because of loss of coax cable, miniADSB receiver should be close to antenna. Possible variant is to place it directly on the pole of the antenna. An output signal from miniADSB receiver is a relatively high-impedance analog signal 0,5 – 2,0 V. The comparator should be as close as possible (in one housing) to the receiver.

FPGA decoder has connector for classic ethernet cable. Ethernet cable is consisted of four twisted pairs. One twisted pair is used for RS-485 data from comparator, other three twisted pairs are used for 5 V power supply. MiniADSB receiver works from approx 4 V supply so even with very long cable there is sufficient margin for voltage drop over cable.

Thanks to RS-485 signalling, distance between decoder and comparator can be dozens of meters (while not counting voltage drop on cable, approx up to 100 meters).

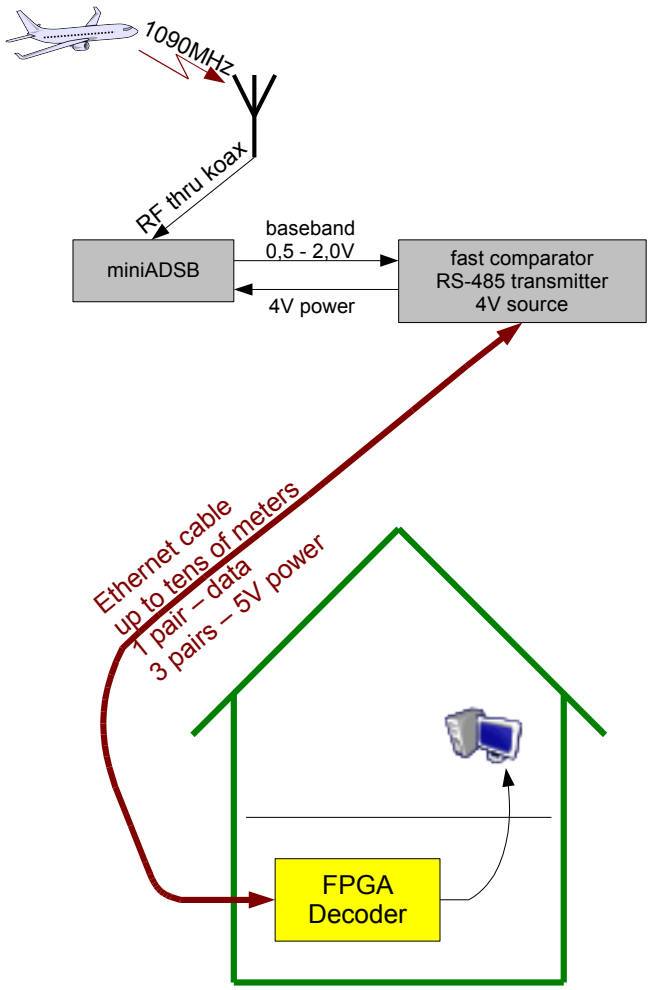

Image 1: Data flow illustration

1.3 Comparator

Comparator **is not** part of this decoder. A circuit similar to *Image 2* must be somewhere close to the miniADSB receiver.

Many other users of miniADSB receiver claims that they use LM393 or similar as receiver comparator. In my opinion LM393 is too slow. I have used LT1011 (quite obsolete, but was available at local store).

Time constant of R1-C1 pair should be around 100 ns or may be completely omitted (only for filtering). Time constant of R2-C2 pair should be around 2-5 µs. It must not be too small (decreasing sensitivity) or too big (slow response for airplanes nearby). Do few experiments to see which results in best reception. There are many ways how to connect R5-R6-R7 network. Its purpose is to increase somehow voltage on negative input of comparator to reduce noise while nothing is being received.

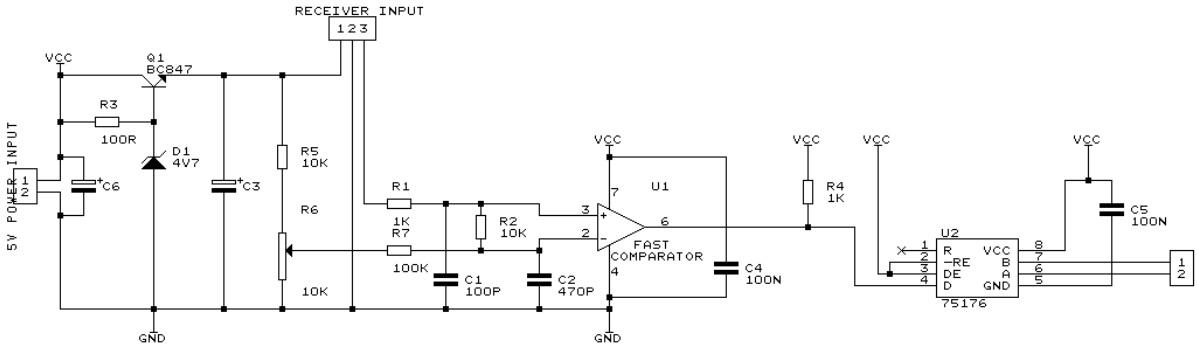

Image 2: Comparator with RS-485 transmitter

Decoded has option to invert received data (can be set using web interface), so there is no demand to precisely check wheather data are not inverted (swapped data lines, swapped positive and negative input of comparator, ...). If you see there is odd number of inversions on data path, simply add one more inversion in the decoder.

2. FPGA internals

There is a MicroBlaze processor inside the FPGA running at 48 MHz. MicroBlaze is 32 bit processor with almost 1 MIPS/MHz with three operand insructions.

Inside FPGA, some of the decoding work is made using logic. Preprocessed data are put into FIFO from which they are picked up by firmware.

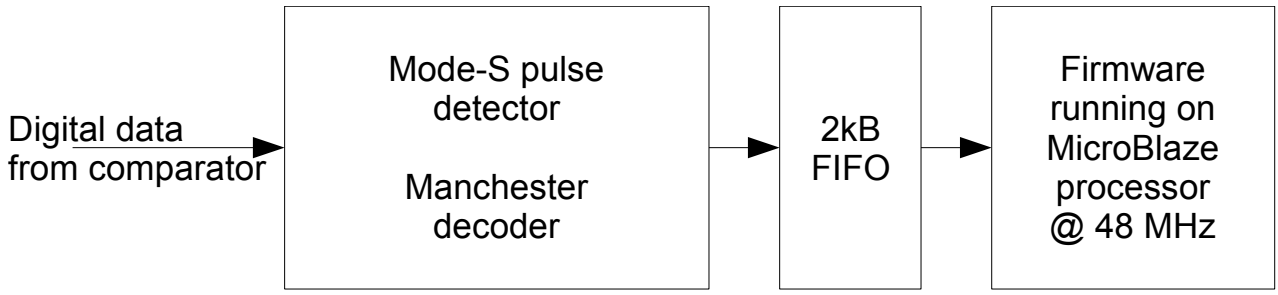

Image 3: Data flow inside FPGA

Program runs completely from SDRAM memory and the program is loaded into SDRAM upon startup from Flash by bootloader. MicroBlaze processor has cache mamory to speedup memory accesses.

3. Web interface

3.1 Web interface capabilities

User can set many parameters of decoder (mostly access rights) by web interface. Web interface is procected by password, which can be changed. The authentication method is known as HTTP Basic Authentication. **The password is transmitted over network in plaintext** equivalent. So use web interface only over on trusted network or anyone between you and receiver will see your password.

3.2 Setup related to TCP/IP

There are several settings related to TCP/IP setup, like in many other equipments which are using ethernet connectivity (like network printers, wifi routers, ...). The most important is IP adress. The decoder can use DHCP assigned address or static IP adress.

When flash memory is blank or upon reset (using reset jumper) the decoder has IP adress of 192.168.1.100 and password is set to 1234 (username admin). First thing you should do is to set another address and password.

3.3 Mac address

Important. Every device connected to ethernet must have unique MAC address assigned by manufacturer. Each manufacturer must buy its own MAC address range. And you are manufacturer. :-) Don't worry, there are several (namely, every fourth) MAC addresses known as 'locally administered'. If you will assign MAC address from this range and ensure that there won't be two devices with same address connected to one network, everything will be working fine.

Locally administered MAC adresses must begin with 02, 06, 0A, 0E, 12, 16, 1A, 1E etc...

So for example MAC address `52:33:11:AA:00:01` is fine.

When flash memory of receiver is blank (and after reset), MAC address is created from random number, which is computed from noise from receiver. If you will have two decoder boards and you will power up them without connected receiver, they will probbably have same MAC address and you network will crash (not mentioning problem that both will have same preprogrammed IP addresses).

4. Data formats

The decoder knows several data formats, all of them are derived from basic „AVR Format“ known by many other decoders. The data can be accessed by serial port, by TCP stream or by UDP datagram. Not all interfaces supports all formats.

	Basic AVR Format	Timestamped AVR Format	Binary AVR Format	Compressed AVR Format
Serial port	in future	supported	supported	not available[1]
TCP stream	in future	supported	supported	supported
Web interface	not planned	supported	not planned	not planned
UDP datagrams	in future	in future	not planned	not available[1]

Accessing data by web interface is meant only as auxiliary method, mainly for installation and debuging purposes.

4.1 Basic AVR Format

Format supported by many decoders and PC applications. Each Mode-S packet is one line, consisting of asterisk '*', 14 or 28 hexadecimal digits holding 7 or 14 bytes of data and a semicolon ';'. End of line is Windows-Style, two bytes 13 10 (decimal).

```
*8D3C484899045DAE0807FFD23686;
*02E617B0D9F419;
*5D504DE20CDCC9;
*02A4853445E148;
*5D774100258F2C;
*02A185347F640D;
*2800171CD666DF;
*5D02912996CFE1;
*8D02912958CD84AF2E9F84A045FF;
```

4.2 Timestamped AVR Format

This format is simply Basic AVR Format extended by 48 bit timestamp. The timestamp is free running counter running at 12 MHz, or 83,333 ns. To distinguish between timestamped and basic format, line starts with '@' character.

1 Decompressor must know the event of sliding window flush, where it can start decompressing of the stream. This combination is not possible unless special synchronization techniques are used to detect the event.

```
@01D7F90185AE8D3C484899045DAE0807FFD23686;
@01D7F9029C1302E617B0D9F419;
@01D7F90634285D504DE20CDCC9;
@01D7F906F64202A4853445E148;
@01D7F908FBF65D774100258F2C;
@01D7F9096CD502A185347F640D;
@01D7F90B485D2800171CD666DF;
@01D7F90D8CED5D02912996CFE1;
@01D7F90F5AD48D02912958CD84AF2E9F84A045FF;
```

4.3 Binary AVR Format

To save traffic (over slow networks), Binary AVR Format sends data directly in binary format instead of hexadecimal. Special escape characters (0xFF, 0xFE and 0xFD) are included to tag indicate start of packet data and timestamp. Data can be easily synchronized on serial line, a TCP stream after connect always starts at packet beginning..

On miniadsbrx yahoo group is available utility bin2adsb and adsb2bin for with compiled binaries for Linux.

4.4 Compressed Binary AVR Format

To save even more space, Binary AVR Format, as is, is compressed by slightly modified gzip (zlib (same compression used by zip, png etc.).

There are two simple modifications to current zip format:

- Block type is only 2 bits long, not 3 (bit indicating end-of-file is omitted, as data are endless)
- Block is always aligned to byte boundary so each block can be immediately flushed to byte-oriented stream. Block type bits (first bits of new block) always start at byte boundary.

Compressor parameters are fixed to compression method known as „fastest“ (gzip -1).

Using Compressed Binary AVR Format saves about 70% of trafic compared to Timestamped AVR Format.

On miniadsbrx yahoo group is available utility sungz available for Linux. Obrataining Timestamped AVR Format from Compressed assuming receiver has IP address 192.168.1.100 and port 7779 can be done using command `nc 192.168.1.100 7779 | sungz | bin2adsb`.

5. Interfacing to a PC

5.1 Any application

Access to TCP stream can be done using standard utilities found both on Windows and Linux like `nc` or `telnet`. Assuming receiver has IP adress 192.168.1.100 and TCP service is at port 7777, you can access data by typing `nc 192.168.1.100 7777` or `telnet 192.168.1.100 7777`. Using `nc` is cleaner than `telnet`.

Data are also accessible over http protocol and can be accessed for example using `wget` with command `wget http://192.168.1.100/rx -O - 2>/dev/null`. Note that the address is `/rx`.

5.2 PlanePlotter

PlanePlotter can directly cope with Timestamped AVR Format data, but it can receive data only from COM port, so you will need some virtual COM port utility, like com0com. This utility can create two COM ports and emulate null modem cable between them. Data from receiver can be redirected directly into first COM port, while PlanePlotter is instructed to receive data from the second COM port of the null modem cable.

Using Timestamped AVR Format PlanePlotter software can do MLATs (planeplotter.pbworks.com/w/page/17117304/MLAT-Introduction), like with hacked SBS-1.

Important: PlanePlotter is not free, but costs €25. Of course you can crack it, but you will not be able to share data with other users and do MLATs and this is not so much fun.

5.3 Sprut's adsbScope

This software (www.sprut.de/electronic/pic/projekte/adsb/adsb.htm) can access directly TCP streams. It is free and opensource so I hope it will sooner or later supersede PlanePlotter.

6. Decoder connectors

6.1 Power supply

The decoder board requires **stabized** 5 V power supply. Such power supply is MW0513SZ. Receiver, comparator and decoder, all running from one power supply requires current up to 400 mA, using 500 mA power supply should be sufficient.

6.2 Reset jumper

If you forgot password etc... you can reset **all** settings by powering up the decoder with installed reset jumper. Note: the jumper is available only on PCB version ADSB-2, not ADSB-1. During the process of reset, all four front panel LEDs will blink about three times.

6.3 RJ-45

Both ethernet and receiver connectors are RJ-45. Take care not to swap them as this effectively short circuits the 5V power supply. The ethernet connector is the longer one, because it contains magnetics.

6.4 LEDs

There are four LEDs on front panel and two LEDs on each RJ-45 connector (the receiver connector has dual-color LEDs, but they are connected in way that only one color is used). It is up to you what colors you will use on front panel.

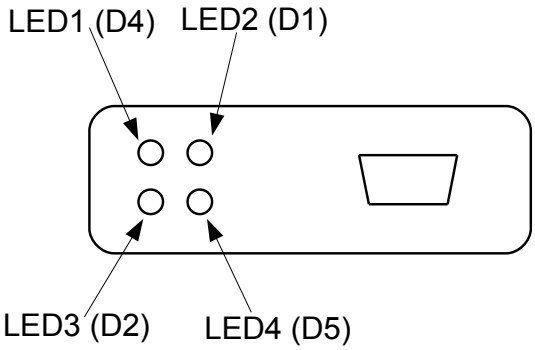

Image 4: Front panel LEDs

LED	Function
LED1	Data chain ok (PC connection ok and some data are being received) Blinks on every packet, but not more than apporox 10 times per second
LED2	No receive data There are no Mode-S data from receiver

LED	**Function**
LED3	Receiver overflow (too many messages) Blinks for approx. 1 second
LED4	No PC connection (can be configured to watch one particular IP address)
Ethernet connector green LED	Ethernet link status
Ethernet connector orange LED	Ethnernet packet receiver or transmit (this LED is independent on firmware)
Receiver connector green LED	Noise receive – cable connected (this LED is independent on firmware)
Receiver connector orange LED	Correct packet receive Blinks on every packet, but not more than approx 10 timer per second (this LED is driven by firmware)

7. Flash programming / preloading

7.1 Flash memory

There is a flash memory on the PCB containing FPGA configuration and firmware. It cannot be programmed directly, but can be programmed thru FPGA using its JTAG port. The FPGA must be configured before it can programme Flash via JTAG port (using USER1 JTAG instruction). In normal operation, configuration is downloaded from Flash memory, when upgrading only, no special action must be taken. When programming blank Flash memory, FPGA must be first configured via JTAG port too. All needed files (configration, programming, image for upgrading using web interface) are included in the provided firmware (see files section on miniadsbrx yahoo group).

7.2 Flash memory types

Because of shortage of Flash memories on market, there are footprints on the PCB for several different kinds of memories. Unfortunately, Atmel and Thomson (ST, Numonyx, ...) have different pinout, you must double check where to solder your memory.

Atmel and Thomson memories have same command for reading, so FPGA configuration and bootloader is same for both memories. Unforotunately, they are erased and programmed using completely different commands, so firmware must be patched for each memory (or detect which memory is used). Currently, I have enough M25P80 8Mbit (1 MB) memories, to firmware is counting only with this one type.

7.3 Memory preloading

I can **upon request** preload flash memory for you, so programming using JTAG cable will not be necessary. But I don't want to carry any responsibility that your flash memory will be erased and misprogrammed during firmware flashing over web and your decoder won't boot. You will need JTAG cable to make the decoder live again.

7.4 Programming over JTAG: SVF files

My compile outputs are so called SVF files. So you can use any JTAG cable which understands SVF files. One cable like this one is ASIX PRESTO programmer (tools.asix.net/prg_presto.htm) (please don't blame me, I was hardware developer of this programmer). The JTAG connector on PCB has 1:1 wiring to PRESTO (ADSB-2 omits not used VPP and key pin to save space). You can also make your own JTAG cable to LPT port, web is full of these cables and software for them.

8. PCB documentation, assembling notes

8.1 Schematics

See schema_ADSB-2.pdf on miniadsbrx yahoo group. Texts are not lines, but fonts, so it is searchable. ADSB-1 is not actual, two pieces of PCB were made and I am keeping both of them.

8.2 Soldering notes

See postup_osazovani.pdf (in Czech) for notes about soldering small SMD chips – FPGA and SDRAM. See pokladacka.pdf on miniadsbrx yahoo group for parts layout.

8.3 Testing and debugging

After assembling all parts, connect power supply to board. Programme the supply to 5.00 V and current limit to 400 mA. Power consumption with disconnected miniADSB receiver should be below 300 mA. Check that all power sources are working correctly and providing 1.2 V, 2.5 V and 3.3 V.

Now you can configure FPGA and programme the Flash memory. If Flash memory is already programmed, decoder should boot up, raising power approx one second after powerup by about 20 mA.

8.4 Firmware

See firmware_0-00.zip on miniadsbrx yahoo group and its readme. New versions, feature requests, bug reports etc... should be posted to the yahoo group.

8.5 Partlist

- Green parts: Available at most local stores
- Grey parts: There may be some problem obtaining these parts
- Red parts: Key parts which are usually not available at local stores
- Violet parts: Ethernet connection – without them is still serial cable connection possible

		Package	Mnf.	Manufacturer Part Number	Other Mnfs.	
C2, C3, C17, C18, C27, C28, C29, C30, C31, C32, C33, C48, C49, C50 C4, C5, C6, C7, C8, C9, C11, C12, C13, C14, C15, C16, C19, C20, C21, C22, C26, C34, C35, C36, C37, C38, C39, C41, C42, C44, C45, C47, C53	14 Ceramic 10nF	SMD 0603	Kemet	C0603C103K5RACTU	many...	Generic, X7R, >=16V
	29 Ceramic 100nF	SMD 0603	Kemet	C0603C104K3RACTU	many...	Generic, X7R, >=16V
C23, C24, C25	3 Ceramic 100nF	SMD 0805	Kemet	C0805C104K5RACTU	many...	Generic, X7R, >=50V
D1, D2, D5	3 Generic Red LED	LED 3mm				Generic LEDs
D4	1 Generic Yellow LED	LED 3mm		China :-)		Generic LEDs
D8	1 Generic Red LED	SMD 1206				Generic LEDs
P1	1 CAN9 90° Female	Canon 9	Tyco	1734354-1	many...	
R2, R3, R11, R12, R13, R14, R16, R17	8 Resistor 330R	SMD 0805	Rohm	MCR10EZHF3300	many...	
R10	1 Resistor 100R	0207	Vishay	MRS25000C1000FRP00	many...	
R18	1 Resistor 1K	SMD 0603	Rohm	MCR03EZPFX1001	many...	
C1, C52	2 Tantalum 47µF/6V	Size B	Vishay	293D476X96R3B2TE3	many...	
C43, C46	2 Tantulum 100µF/4V	Size B	Vishay	293D107X0004B2T	many...	
C40	1 Ceramic 2.2µF/6V	SMD 0805	Panas.	ECJ-2YB0J225K	many...	May be for 10V or 16V
C51	1 Ceramic 10µF/6V	SMD 0805	Panas.	ECJ-2FB0J106M	many...	May be for 10V or 16V
D9	1 BAT54WX	SOD-523	ST	BAT54K	MicroComm	
J2	1 DJK02			unknown		power connector
L1	1 BLM18BA100SN1D	SMD 0603	Murata	BLM18BA100SN1D		ferrite chip, replacement possible
L2	1 2.7µH/2.3A	SMT43	Matsuta	SMT43-2R7	unknown	
U3	1 ST3232CD	SO16	ST	ST3232CD	TI, MAX, ...	3.3V variant of MAX232
U10	1 75176	DIL8	TI	SN75176BP	NI, LIN, ...	RS-485 interface
D6, D7	2 ESDA6V1L	SOT23	ST	ESDA6V1L	unknown	transil, 6.1V, can be ommited
J3	1 RJHS-538A	RJ-45 conn.	Amph.	RJHSE-538A	unknown	RJ-45 without magnetics, 2 LEDs
U4 / U11	1 AT45DB161D-SU	many*	Atmel	AT45DBxxxD-SU	ST*	2Mx8 Flash; * - not pin compatible!
U5	1 MT48LC4M16A2P-7E	54-TSOP II	Winb.	W9864G6JH-6	Micron, ...	4Mx16 SDRAM, >=133MHz
U6	1 LM3671-1.2	SOT23-5	NS	LM3671MF-1.2/NOPB		600mA switched 1.2V
U7	1 MCP809T-315I/TT	SOT23	Mchp.	MCP809T-315I/TT	none	watchdog, can be ommited
U8	1 KF33BD	SO8	ST	KF33BD		500mA linear 3.3V
U9	1 KF25BD	SO8	ST	KF25BD		500mA linear 2.5V
U12	1 CB3LV-3C-48M	SMD	CTS	CB3LV-3C-48M0000		Oscillator, 48MHz
U13	1 XC3S250E-4VQ100C	VQ100	XIL	XC3S250E-4VQ100C	none	FPGA
U1	1 PU1S041A-34-LF	RJ-45 conn.	Bothand	PU1S041A-34 LF	unknown	RJ-45 with magnetics, 2 LEDs
U2	1 ENC28J60/SS	TSSOP28	Mchp.	ENC28J60/SS	none	10BASE-T Ethernet PHY
R1	1 Resistor 0603 2K32	0603	Vishay	CRCW06032K32FKEA	many...	can be replaced with 2K2
R4, R5, R6, R7	4 Resistor 0603 49R9	0603	Rohm	MCR03EZPFX49R9	many...	can be replaced with 47R
	1 Enclosure		Hamm	1455C801 or 802		Hammong mfg.
	1 Laser cut ends		Plehaso			Plehaso, 1.6mm Al

Table 1: Partlist

The strange part DJK02 is power connector according to *Image 5*. Can be obtained at GMe (www.gme.cz) with ordering number 806-049 or at ECOM (www.ecom.cz) with ordering number 08160. Many other source possibly exists.

Image 5: DJK02

8.6 Enclosure

Enclosure is hammond standard aluminium enclosure with custom laser cut (from different material – it looks a bit different) ends. The enclosure looks really good, but it is also quite expensive.

Image 6: Enclosure

The enclosure is anodized, but the ends not. You can try anodize them your own. I tried it with unsatisfactionary results.

9. Kit ordering

Prices effective from Dec 2010:

PCB	12€
Red parts	40€
Grey parts (cannot be ordered without red parts)	6€
Violet parts (cannot be ordered without red parts)	8€
Enclosure + Laser cut ends	15€
Everything (PCB, red and violet parts, enclosure)	74€
Everything (PCB, red, grey, green and violet parts, enclosure)	**79.5€**

Preliminary postage – if everything fits to 500g – package ensured to 20€ – for more ensurance, the postage would cost must more :-(

Czech Republic	2€
Europe	**5.5€**
Other	7€

The price may change due to USD/EUR and CZK/EUR changes!

The price already includes fees of PayPal transfer.

For orders within Czech Republic, I accept bank transfers within Czech Republic in CZK, and the price is lower by 4%, PayPal fee (3.9% + some static).

10. Document version history

Date	Version
13.11.2010	First version
14.11.2010	Added power supply, DJK02 ordering
27.11.2010	Added firmware note Added chapter controls
1.12.2010	Corrections and kit price determination