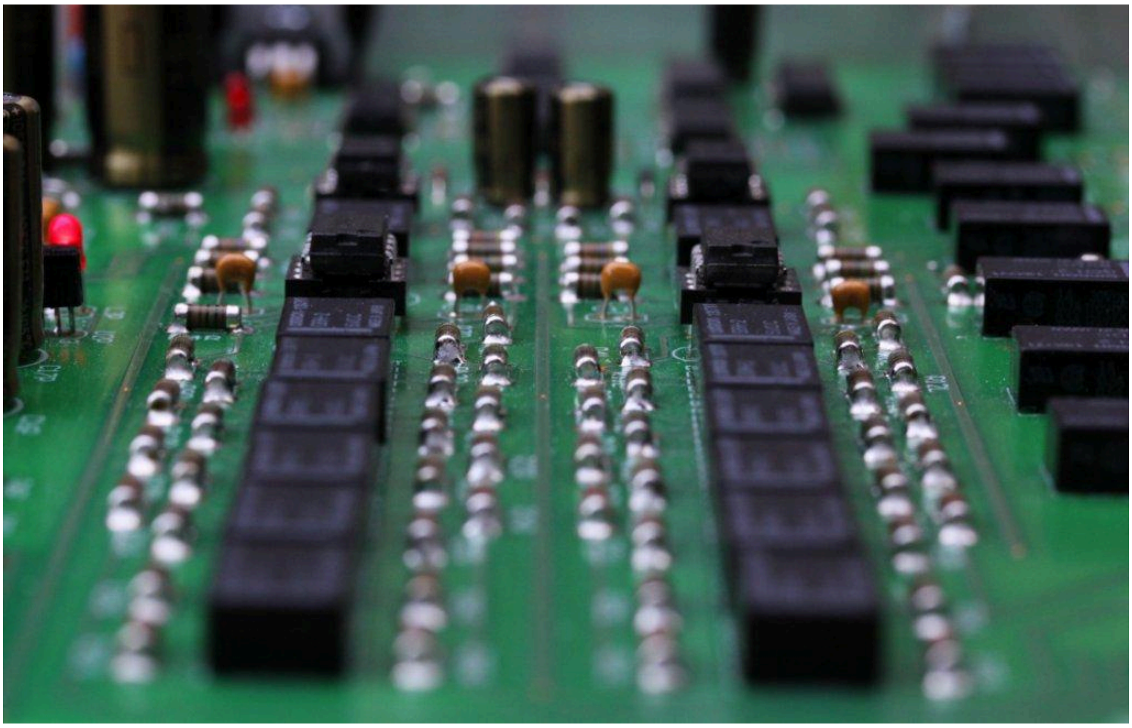
RelaiXed2

User Manual

Revision 2016-09-12

Jos van Eijndhoven



Recent updates to this manual:
12/09/2016: Add a section on connecting a 2x16 character OLED display
16/05/2016: Updated the smd-mosfet type number, added photos of rotary switch wiring.
01/06/2013: Clarified the setting of jumper J3C, clarified the mosfet mounting
13/12/2012: Updated the URL links to the SourceForge site with the project software
03/12/2012: Added a comment on adapter-sockets for the metal-can opamps.
31/3/2012: Added paragraph on standard versus high-end component selection,
mounting instructions for the BS250 mosfets, and a line about the Mouser transformers.
08/2/2012: Add some comments and photos for the metal-can LME49720HA audio opamp,
add comment on installing missing libraries when the PC-windows relaixed application doesn't start up.

Contents

Introduction

The ‘relaiXed2’ is an audio pre-amplifier, aiming at high-end audio enthousiasts, designed for DIY (Do-It-Yourself) electronic assembly. The ‘RelaiXed2’ design is the successor of the original ‘RelaiXed’ which was first published in December 2007. After two small updates in 2008 and 2009, this RelaiXed2 is a totally renewed design who’s schematics has hardly anything in common with the previous versions. The overall intention and spirit remained, but implementation details are chosen differently. Next to this manual, other design documentation is available from the RelaiXed2 website at: http://vaneijndhoven.net/jos/relaixed2/index.html

This preamplifier is designed to accept balanced stereo signals, usually fed through XLR connectors, next to a couple of standard ‘single ended’ inputs through ‘cinch’ connectors. Its preamplifier outputs are balanced connectors. This complements a number of power-amplifiers that recently appeared in the DIY world preferring balanced input signals, such as the ExtremA (http://www.hardwareanalysis.com) and commercial designs such as the Class-D modules from http://hypex.nl.

The new Relaixed2 maintains several design aspects from its predecessor:

- A 6-stage relay-based attenuator for 64 volume steps, and relay-based input selection.
- IR remote control and a 2-digit 7-segment display for user feedback
- A compact high-quality opamp-based gain stage powered from a low-noise power-supply.
- An identical PCB form factor and connector mounting positions.

However the design is updated in many aspects:

- It now provides left-right balance control next to volume control.
- The attenuator is located AFTER the opamp stage, providing cleaner noise-free sound and a facility to suppress power-on and power-off output signal bumps.
- The design contains only one microcontroller. The PCB board is now operated from a I2C-based ‘output expander’ in stead of a second microcontroller in the older design.
- The microcontroller now features a USB connector, supporting easy in-circuit updates of its embedded software (firmware).
- The source-code of the project is all rewritten and made available as open-source. The code (svn-)repository is available at: http://sourceforge.net/projects/relaixed2/

The preamp features infra-red remote control for all its functions. Volume control and channel selection are executed through small-signal relays. Two series of six relays implement a 4-channel (balanced stereo) 64-step logarithmic attenuator. The combination of high-quality contact relays with prime quality resistors leads to superior volume control, audibly far better than conventional potentiometers employing a sliding contact over a resistive layer. The sealed relays maintain their contact quality over a practically unlimited time span, unlike (expensive, high-end) rotary switches which degrade over time.

The main ('relay') PCB carries the input selection relays, the attenuator, the active output stage, the power supply stabilization section and part of the digital control. With its on-board XLR connectors, this PCB must be mounted close to the chassis rear panel. An additional small 'display' PCB is designed for front panel mounting, and accommodates the IR-reception, a 2-digit 7-segment display for visual feedback, and an optional manual control switch. Note that no audio signals are routed to the front panel. The professional quality gold-plated PCB not only looks really beautiful but is also very easy to solder, and some people believe that such gold finish positively affects sound quality.

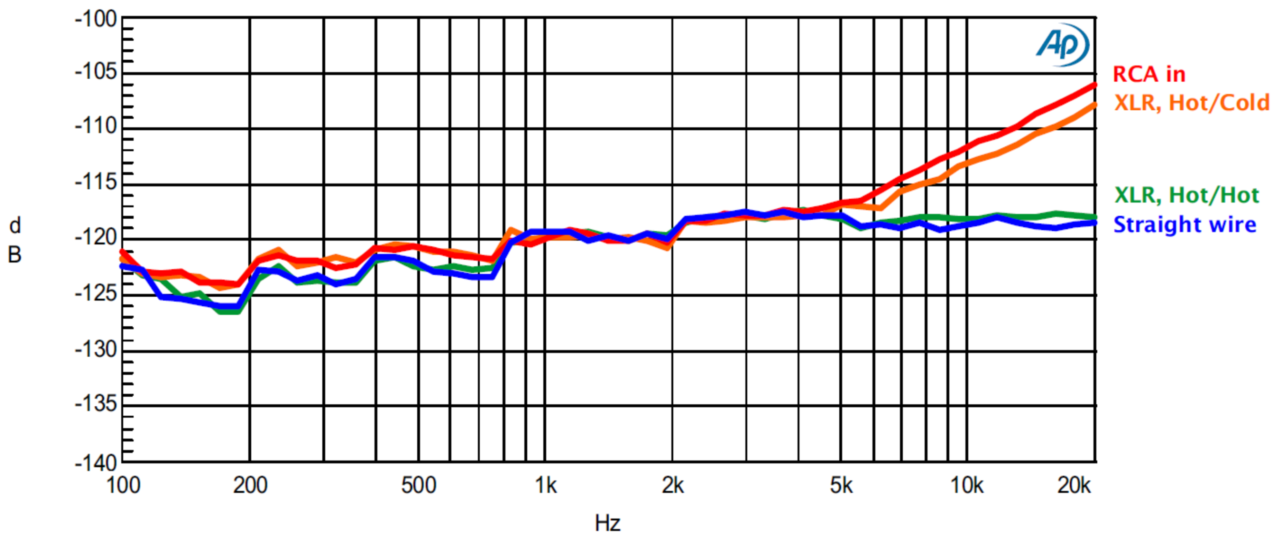
The preamp has a volume range from 00 up to and including 64. Volume levels 01 to 64 span a dynamic range of 60dB with 0.95 dB steps. A 60-dB range in power corresponds to a range of for instance 0.1mW to 100W, depending on your power amplifier. Volume level 00 disables all input signals giving zero output.

The RelaiXed2 has been designed to provide a balanced audio path from input to output. However, many users will also want to connect conventional 'single ended' (cinch-based) input signals. The preamp supports that, by converting such inputs to balanced output signals. Six selectable inputs are provided. The design was made to support two configurations: 4 single-ended input channels and 2 balanced input channels, or 3 single-ended and 3 balanced ones. The RelaiXed has no dedicated IR remote control unit but meets different existing remote control protocols: the Philips RC5 and RC6 as well as the Sony SIRC protocol. As also other brands meet these protocols, it should be easy to find a compatible control device. In fact, any 'generic' IR remote control unit will be fine. Within these protocols, there is still much freedom to link specific buttons to functions and to activate the RelaiXed only for a specifically selected device code. To meet such flexibility the preamp can 'learn' button code signals. This is accomplished through a set-up procedure explained later in this document.

Specifications

Output connectivity:	one pair of XLR male connectors for stereo balanced output
Input connectivity:	6 stereo inputs: 3 pair cinch connectors + 3 pair XLR connectors -or- 4 pair cinch connectors + 2 pair XLR connectors
Output impedance:	250 - 380 ohm, varies with volume selection
Input impedance:	47Kohm
Max. gain, cinch input:	3x (1 : 1.5+1.5) at max. volume (is +9.5dB voltage gain)
Max. gain, XLR input:	2x (1+1 : 2+2) at max. volume (is +6dB voltage gain)
Frequency range:	DC to 320 kHz (-3dB) (Limited by input filter. Without filter: 2MHz for 2x1V output)
L/R channel separation:	on cinch inputs: 128dB on XLR inputs: 137dB (at 1kHz)
Signal-to-Noise ratio:	123 dB at 2Vrms nominal output, max. volume setting, shorted input
Output noise voltage:	10nV/√Hz at max. volume and shorted input
Distortion at 1kHz	-120dB or 0.0001%, limited by measurement equipment, see fig below
Maximum input voltage:	6Vrms (cinch) or 2x5Vrms (XLR)
Volume steps:	63 steps of nominal 0.95dB, is 60dB range
Balance control:	+9 to -9 volume step difference between the L and R channel
Power consumption:	standby mode 0.5Watt, full power < 8 Watt.
Switched power output:	max. 10A x 250V

RelaiXed distortion vs frequency + instrument limit (blue)
THD *without* noise
All graphs taken at 5V in, 5V out, volume setting adjusted accordingly



Collecting Components

A separate document is provided with a component table, that lists in detail component types, the component count to be ordered, and their part numbers in a leading web-shop. As one can order most components easily from there, I do not intend to provide complete kits including all the components.

The microcontroller (PIC18F25J50-I/P) has an internal flash memory to contain its software image.

- Be aware that a microcontroller directly from the shop is 'empty', and the RelaiXed preamp does not operate on that.
- An empty microcontroller can only be programmed through a real programmer device, such as Microchip's PICKIT2 or PICKIT3. I would recommend acquiring one of these only if you want to modify the project source-code and create your own software version. For more information on this, see the RelaiXed software manual.
- Once the microcontroller is programmed with its software image, the software can easily be updated. The USB connector can connect to a PC through a regular USB cable, and a new downloaded image can be written into the microcontroller with the help of a utility on the PC. The USB connector is not operational on an 'empty' microcontroller.

This leads to the following shopping:

- The two PCBs can be obtained/bought from me, as indicated on the project web-page.
- This also holds for a pre-programmed PIC microcontroller.
- The other components can be obtained from a web-shop. A detailed component table is provided as separate document. Stock-numbers are provided for Farnell and for Mouser. Farnell is European, but closely cooperates with the US Newark. For my Mousr shopping-list I included both European and US prices. All prices are excl sales tax. As these prices can change daily, my list obviously provides a global impression only. In Europe, Farnell seems to have a policy to discourage direct sales to private persons, although they still handle that. Still, Mouser is more convenient for private buyers.
- I can deliver a dedicated front-panel, but the rest of the cabinet is to be bought in the Hifi2000 shop.

Standard versus high-end configuration

The LM4562 audio opamps (U1L and U1R in the schematics) represent the state-of-the-art in Audio opamps, and are an important contributor to the sonic quality of the RelaiXed. They combine low noise with high bandwidth and low distortion. The component table indicates their 'normal' dual-in-line plastic package. The LME49720 seems to be exactly the same silicon component with a more recent type number. These opamps create a very clean and highly detailled sound. However, I must confirm claims on internet that the 'metal can' (-HA) version really sounds better, but those are considerably more expensive (such as LME49720HA). It would probably depend on your other audio equipment, but this metal-can version creates a more 'open' sound with more spatial resolution. To me it is a serious improvement, and worth looking at ebay for a good price for these parts. Of course you are free to experiment with other (dual) audio opamp types, such as the OPA2134 or OPA2604: the ciruit design is fine for that. If you really want to use a particular 'single opamp' type, such as the OPA627, AD797, or LME49990, you have use those in combination with a single-to-dual adapter.

The design has HF filter input capacitors (C1L/R, C2L/R) that limit the amplifier frequency range. The proposed standard type, with C0G dielectricum, are first-class industrial types. Nevertheless, the high-quality 'silver-mica' types as denoted in the component table might bring you still sonic improvements. Unfortunately, the silver-mica types are somewhat larger then footprint provides, and sometimes even have leads that are just a fraction too thick for the provided PCB holes. :-(

For a 'standard' versus 'high-end' component selection I would recommend:

	Standard	**High-end**
Audio opamp U1R, U1L	LM4562 or LME49720 (PDIP)	LME49720-HA (Metal can)
Powersupply opamp U3P, U4P	NE5532	LM4562
Input filter C1R C2R C1L C2L	C0G type	Silver-Mica type

Please note that the 'high-end' versus 'standard' differentiation is based on just listening. In all standard measurements both versions are truly identical: in noise, bandwidth, speed, distortion... For the metal-can opamps you might want to acquire little cooling heatsinks and socket adapters, as shown further down in the mounting description.

Comments on other components

The selected choice of components of course leaves some freedom for variations. In general I prefer to select first-grade industrial quality types. Maybe you could save some money with a cheaper selection, or experiment with a few more high-end components. The selection of the following components deserves some extra comment:

- The MELF resistors from Vishay-Beyslag are professional quality types, selected for their low 'current-noise'. Note that 'compatible types' from other manufacturers mostly don't provide this level of noise performance (0.01uV/V).
- The TQ2-5V relays are high-quality sealed relays, which will maintain contact quality over a long life-time. They represent a considerable part of the overall component cost. It is certainly worthwhile to search (for instance) ebay.com for a sharply-priced offer. The Omron G6H relays are fine replacements, if you can more easily obtain those.
- The power-supply also uses opamps (U3 and U4) to create finely stabilized supply voltages for the audio opamps. The indicated NE5532 types are good for that purpose, with proper low-noise behavior and low output impedance. Alternatively, the fore-mentioned LM4562 perform slightly better on my test-bench as well as in a sonic test, also for this powersupply application.
- The tiny toroid power transformers proposed in the component table from Farnell are high-quality types: they spread little magnetic field around, are physically silent, have low

internal losses (low power consumption), and are easy to mount on a single screw. Unfortunately, these are somewhat expensive. Alternatively, the component table shows relatively cheap 'block' transformers from Mouser. These have the advantage that the primary and secundary winding are not on top of each other, and therefor exhibit a very low parasitic coupling capacitance. This is a really nice property! Note also that the indicated 10VA 2x6.3V transformer from Mouser will only barely fit in my example 'Hifi-2000' cabinet with its internal height of exactly 4cm.

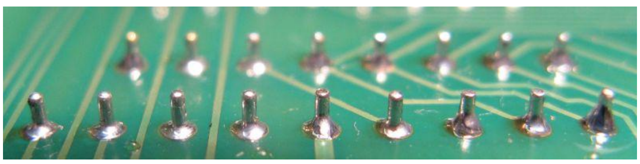
- The output power relay, Omron G6RN-1A, is used to switch-off the analog (+/-15V) power transformer when the digital circuit enters stand-by mode. The relay has enough power to also switch a nearby power-amplifier if that would be convenient in your system. The chosen relay type has a relative sensitive (low-current) coil. Please check that if you search for a replacement type (A Tyco RYA3 would also be OK). If you don't feel the need to switch the analog power-supply, you might omit this relay entirely.
- The small power-supply decoupling capacitors C3L/R and C4L/R are merely placeholders. For now, these are not mounted/used in building the RelaiXed.
- I would recommend to mount the four opamps in sockets: that allows for easy replacement later, maybe to upgrade to newer opamp types. For all other (digital) ICs, mounting in sockets seems somewhat overkill to me. You can do so if you are uncertain of your soldering skills and are afraid to burn some ICs too hot.
- The power-supply elco's are good-quality reasonably low-ESR types, with a preference for the 'high temperature' components that are guaranteed to 105 degrees Celcius. This high temperature is definitely not expected to occur, but such elco's typically have a much better lifetime.
- The USB connector is (by current standards) a somewhat old-fashioned 'B' type connector. The more modern mini- or micro- USB connectors are not suitable for manual soldering on the PCB.
- The schematics indicate a pair of BS250 mosfet transistors as display driver, which have a classic 'TO-92' through-hole footprint on the PCB. Unfortunately, these transistors have become less easy to obtain. You might ask me for a few from my own stock. The more recent version of the display-PCB (labelled July 2011) provides an extra footprint, which allows a small SOT23 (SMD-type) footprint as alternative. This allows the selection of alternative components such as a FDV304P or BSH201. *Check my mounting instructions!*

Assembly

The RelaiXed was designed to make DIY assembly relatively easy. The use of tiny SMD components was (mostly) avoided. Nevertheless, if this would be your very first soldering experience, you might try a simpler/smaller pet-project first: some experience might be beneficial for the quality of the final result. The description below aims to provide tips for non-experienced users, and reduce the number of potential mistakes.

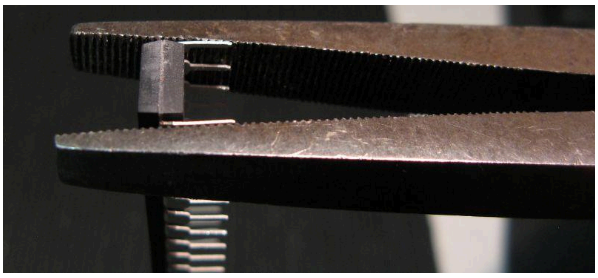
Soldering instructions

Of course, remarks about basic soldering techniques are not needed, as you already have good skills in soldering... This is how the bottom side of my PCB looks like. A nice smooth and shiny



soldering joint is made by having your soldering wire close to the soldering pad, so that it melts on the pad. If you melt the solder on the tip of your soldering-iron, and wait a few seconds before applying it to the pad&component, it will not attach nicely. If you do a faster job, you get a better result, and reduce the risc of overheating your components. Traditional solder wire has a tin-lead metal mixture with about 40% lead. This the easiest type to work with. For environmental reasons, lead is currently being avoided for soldering. A beautiful alternative is the tin-copper-silver mixture. The (typically 3%) silver content is by some audio enthousiasts appraised for giving better sounding results. As all lead-free types, this has a somewhat higher melting temperature, making it slightly less easy to work with. Also, this type of solder cools down to a grayisch appearance, not so shiny. Finally, thin soldering wire like 0.5 to 0.7 mm, is strongly preferred for this kind of PCB assembly.

The (digital) ICs have a conventional DIL (dual-in-line) housing. By default their pins do not point straight down, but are slightly slanted outwards. This makes them somewhat difficult to fit in their pin-holes in the PCB. This mounting is simplified if you bend the pins to point straight down. Some persons do that by manually pressing the IC against the table surface. I prefer to bend them by gently pressing with a ‘flat nose pliers’ as shown in the picture below:



Assembly order

In general I recommend two rules to choose component assembly order, which however do not always agree with each other:

- To ease the mounting and soldering of conventional through-hole components, you mount the lowest-profile (lowest height) components first. These components are plugged-in at the top side of the PCB, and soldered at the bottom side. For soldering, you turn the PCB upside down on your desk. This ordering ensures that the components to be soldered are mounted neatly against the PCB, as they are supported by the desk surface: they don’t just hang in the air below the PCB.
- The on-board power-supply and power-regulation is finished and tested with a multi-meter, before soldering-in the other ICs (digital logic, audio opamps). This prevents damaging components in case of some error.

Assembly - phase 1

According to above two points, you can mount everything except

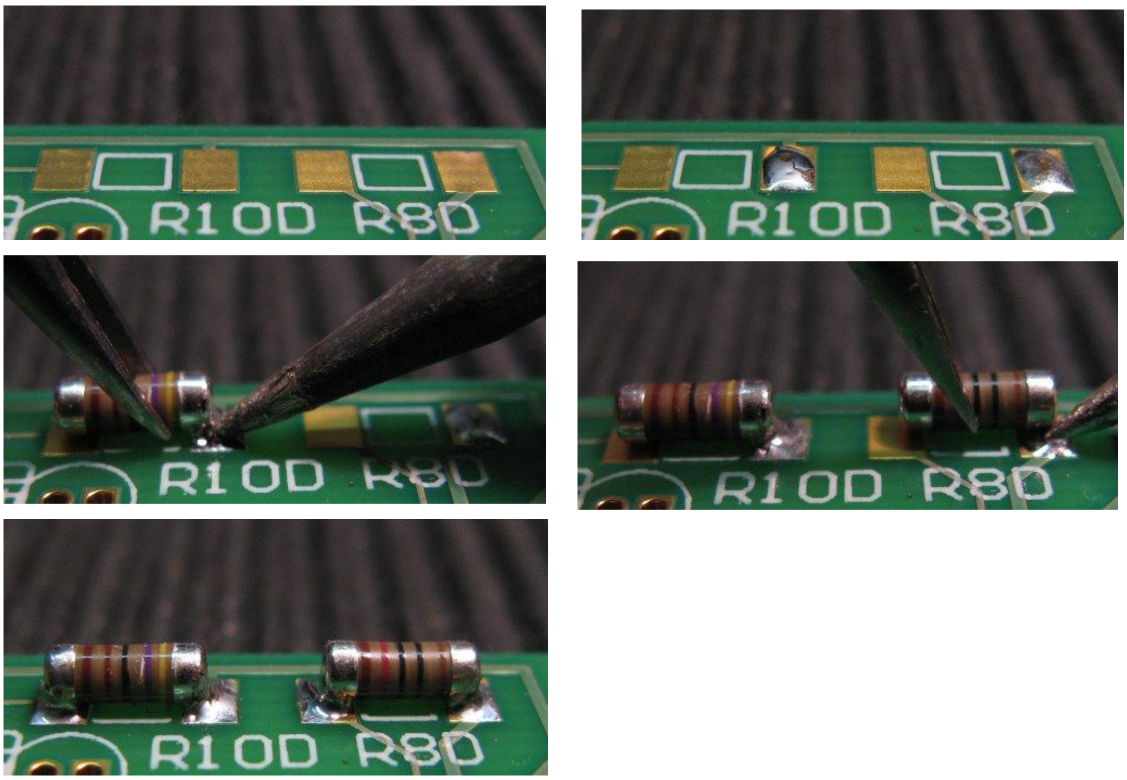
- All four opamps (U3, U4, U1L, U1R)
- Digital control ICs on the main PCB: U1C, U2C, U3C, U4C, U5C, U6C
- The microcontroller and IR receiver on the Display (front) board: U1D, U3D

Mounting the MELF resistors

If you don’t have experience with soldering SMD components (components without wires), these MMB0207 resistors are an easy start: they are still relatively large and actually mount quicker than traditional resistors with wire ends. You mount them in three steps:

1. Apply a little solder on one of its two PCB pads. You can efficiently do this for many resistors at once. Be careful to apply just very little solder, and do so quickly: the ‘raisin’ in the solder should not all evaporate!
2. Pick the resistor with a tweezers, and hold the resistor in place. With your other hand, heat the pre-soldered pad with the resistor on top, such that the resistor ‘sinks’ in the solder, ending neatly flat on the PCB. If done OK (quickly enough) the solder will nicely spread between the resistor and the PCB pad.
3. Solder the other end of the resistor to its pad on the PCB. This is now easy, since the resistor is already fixed in place.
4. If needed, you might apply some extra solder to the first resistor terminal.

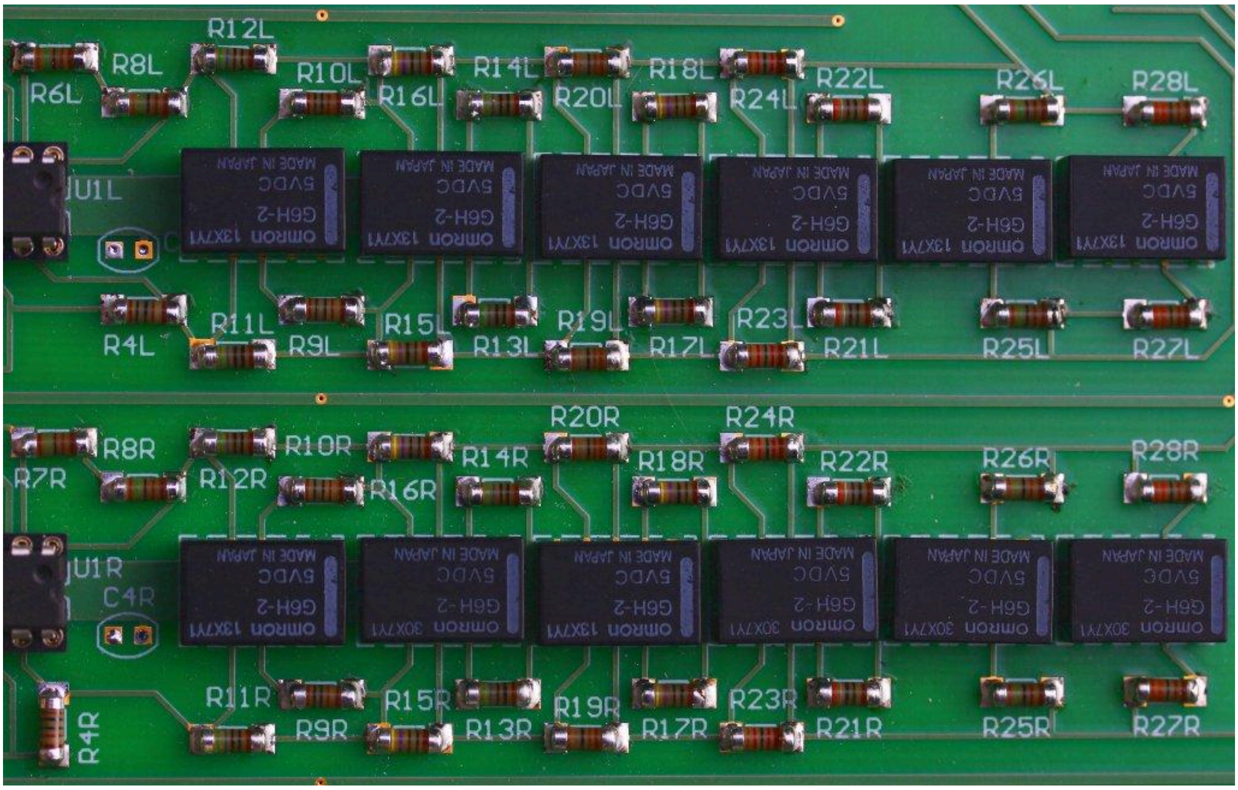
As example, steps 1 to 3 are shown for 2 resistors in the pictures below:



Note that all resistors have colored rings that indicate their value. The relation between resistor value and ring colors is explained on e.g. http://en.wikipedia.org/wiki/Electronic_color_code. The slightly wider brown ring, left-most in above fotos, indicates a 1% tolerance on the resistor value.

Resistor R17P is indicated in the schematics with a value of 1 ohm, but should better be replaced with a small wire-bridge. This also saves you from buying 10 of these :-)

Just as confirmation, the photo below shows the colors of all resistors of the attenuator. Note in this picture that every set of 4 resistors that is located ‘above’ each other shares the same resistance value. Note that, unfortunately, R6 and R7 have changed their identifiers in left versus right.



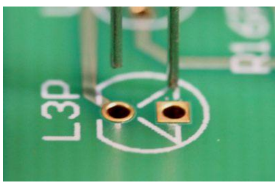
Mounting sockets for the opamps

For the 4 opamps, I would advise to mount 8-pin DIL mounting sockets. This allows you to later easily update the board with newer/other opamp types for improved sonic quality. For all other (digital) ICs, I would not use mounting sockets. If you feel unsure about your soldering skills, such sockets might give you an ease-of-mind, avoiding the danger to overheat your IC's. This might hold in particular for the (preprogrammed) PIC microcontroller.

When building a 'high-end' version of the relaixed, the metal-can version of the opamp can be plugged into the standard DIL-socket. The optional/alternative 'silver-mica' 470pF capacitors in the component list have a larger size than the default 'C0G' type capacitor, and can be mounted rotated as in a foto further down, showing these yellow caps near the metal-can opamps..

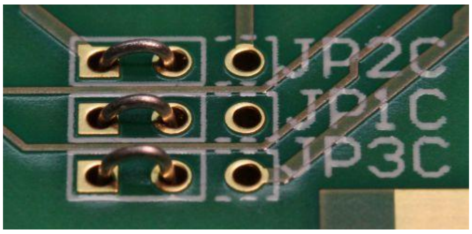
Mounting the LEDs

The three LEDs on the main board have an orientation, do NOT solder them wrongly!
The longer of the two 'legs' is the anode, the mounting orientation is shown below:



Mounting the jumpers

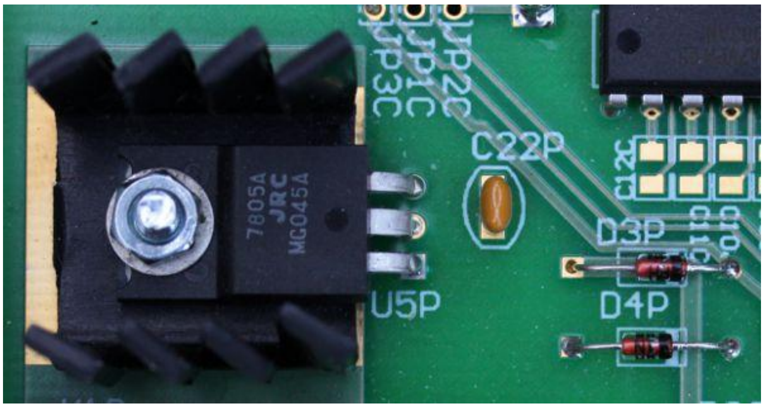
The main board contains three jumpers to set the mode of operation. Normally, you would not use actual jumpers. but merely solder small wire-bridges for the desired setting. Jumpers JP1C and JP2C are used to identify each relay-board in case more then one board is connected to a single front display board. In the ‘normal’ setup, with a single relay board, these jumpers are set to GND (pin 1). Jumper JP3C selects whether input 4 will be used as cinch (single-ended) input or as XLR (balanced) input. To choose the single-ended mode for input 4, set the jumper to pin 1, matching the drawn connection in the schematics. This default for the three jumpers is shown below:



To use input 4 in balanced (XLR) mode, create the JP3C connection between the center and the right hole in above picture (between pin2 and pin 3).

Mounting the 5V regulator

The ‘7805’ 5V regulator (U5P) is mounted on the PCB with a small heatsink for cooling. The photo below shows an IC with a black tab for the screw, typically this is just a metal tab. The IC can be directly attached to the heatsink and the bare-metal area on the PCB, no isolation is needed. (But, the screw should NOT contact the metal of your chassis!) As the IC will become only hand-warm, no cooling paste is needed for heat guidance.



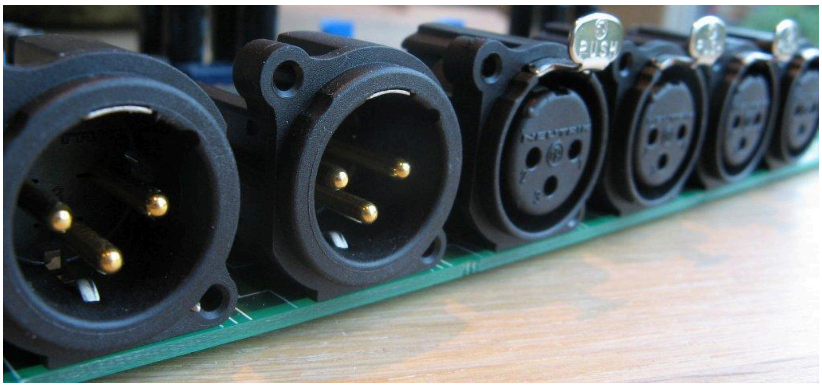
Note also the orientation of the diodes D3P and D4P!

Mounting the elco’s

The elco’s C13P to C21P are probably the last components to mount, and also have an orientation: Be sure to mount them with their ‘+’ terminal in pin 1 (the square pad), also denoted with a ‘+’ on the PCB.

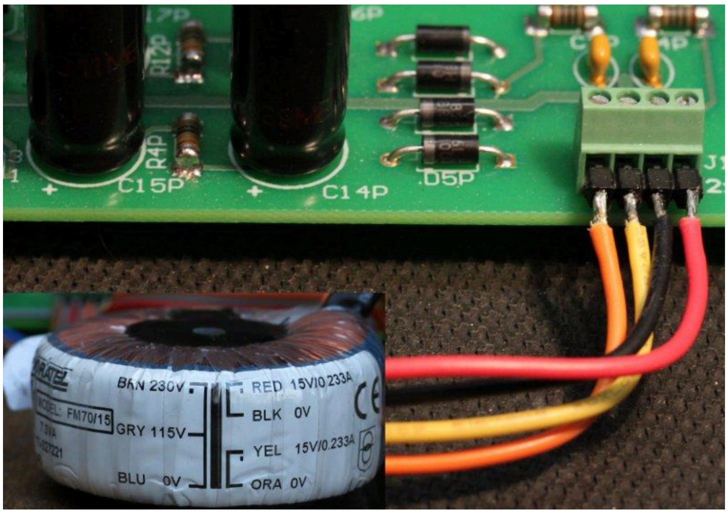
Mounting the XLR connectors

The picture below show the XLR connectors, with the male connectors for the audio output. The types on this photo have golden contacts, according to the component list, which is not common for XLRs. These types are best matched with guilded cable plugs, as mentioned in the component table.



Connecting the transformers

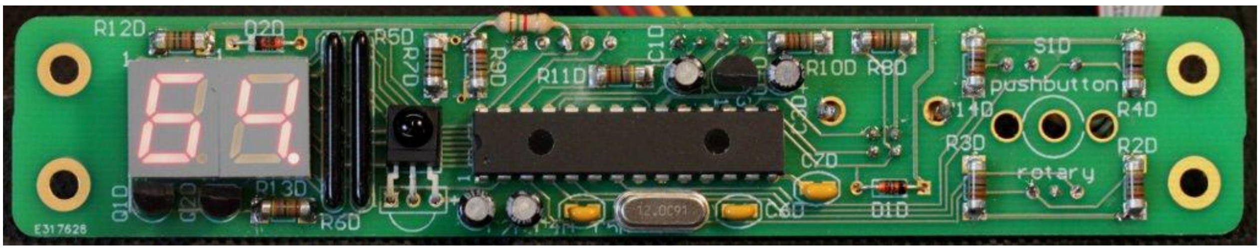
To connect the transformers to the PCB connectors (J1P and J2P), a correct wiring order must be chosen. If the transformer has separated secundary windings (such as the proposed toroids), the ‘lower’ side of one winding and the ‘upper’ side of the second winding must be connected to the center two pins of the connector. This is illustrated with the picture below, note the color markings on the transfomer:



(But you might mirror/reverse all 4 wires together, or just the orange and red in above picture.)

Mounting the LED displays

For mounting the LED display, please double-check that they are not mounted upside-down: their decimal dot must be below. You would not be the first nor the second to do this wrong... Also see the IR receiver orientation and the microcontroller IC (although those two should be mounted only AFTER doing the first tests on powersupply...) The black bars next to the LED displays are the resistor arrays R5D and R6D. These have no dedicated mounting orientation.

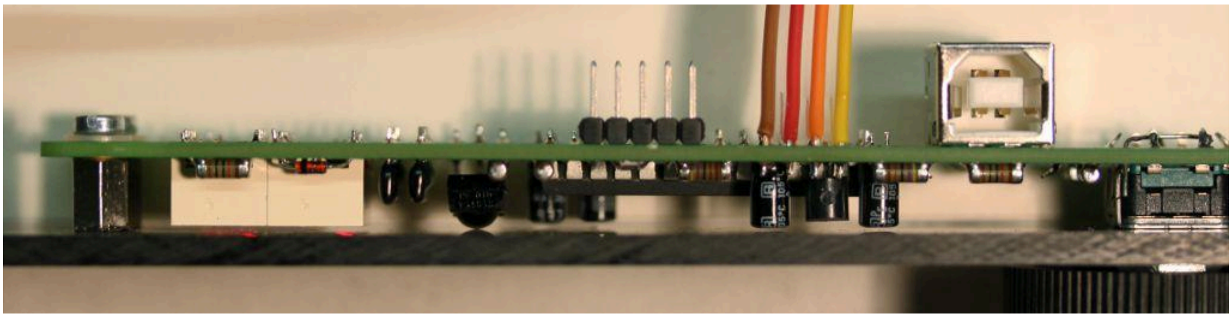


Mounting one extra resistor

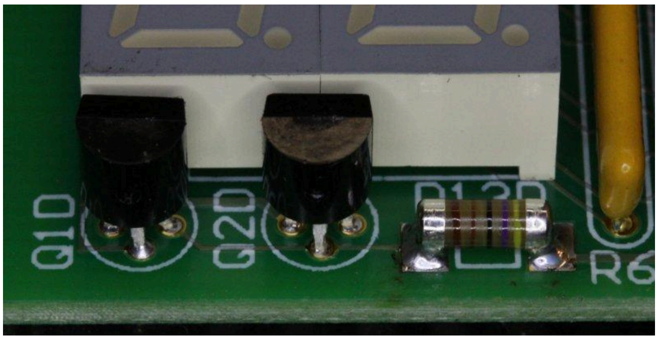
The first version of RelaiXed2 unfortunately had one error in its design: the display board must draw some extra current from the 3.3V stabilizer to function correctly. To repair for this, the schematics got a late update to include an extra ‘R99’ which is not accomodated on the first version PCB. That is the resistor you see along the top side of the PCB in above foto. You can choose a 1Kohm to 1.5Kohm resistor, and mount that anywhere on the PCB between the 3.3V and GND. The later display-PCB labelled ‘July 2011’ does provide a standard (MELF) footprint for this resistor.

Keep a low profile

Most components on the front (display) side of the display print should be mounted relatively close to the PCB surface. That allows the PCB to be mounted with 7mm spacers against a (transparant) front plate, as shown in the top-view picture below:

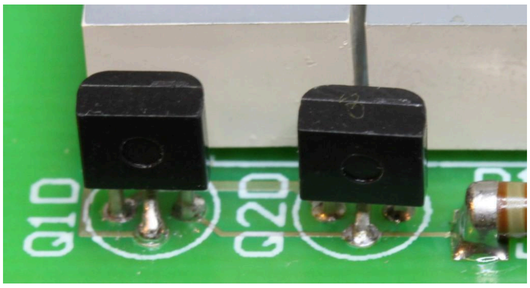


Mounting the BS250 mosfets



Please check this text carefully!! This component is THE only one -for now- that led to non-functional displays at first attempts...
There are several manufacturers of this type of mosfet, but they apparently use different pin alignments :-(

- Philips BS250, Vishay Siliconix BS250KL, Vishay Siliconix BS250, Diodes BS250: These are fine, they are mounted with their flat side towards the displays, according to the drawing on the PCB, and above foto.
- Diodes/Zetex BS250P, as provided by Mouser and RS Components, currently in the Relaixed proposed component list. This one has a **reversed pin order**. Its ‘round’ side is flattened with rounded edges only, and prints ‘BS250’ on this curved side, without its trailing ‘P’, for added confusion. You should mount this one with its flat side facing away from the LED display, its printed text side towards the display. See photo below:

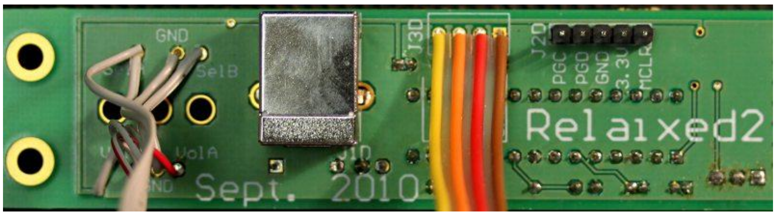


If you are uncertain about the mounting of your BS250, I would advise to simply not yet solder them in. You can try to power-up the PCB with these things 'loose' in place, and see whether it works, or needs rotation. A momentary bad contact will not destroy anything. (But unpower the board when plugging the device in or out.)

The front display PCB version marked as "July 2011" or later provides alternative mounting positions for FDV304P mosfets in a small SMD SOT-23 footprint (denoted in the schematics with Q3D, Q4D). These are mounted on the PCB backside, and they do not have confusing differences in pin order. (Alternatively, the BS250FTA type is OK as well, but is very sensitive to static handling and seems to easily fail.)

Mounting the display back-side

The display back side is oriented towards the inside of your chassis, after mounting the display board against your front panel. It contains the USB connector oriented towards the top side, providing access with the top cover of your preamp removed. (The USB connection is -with my current software- not used in normal operation, it is intended for software upgrade or software development purposes only.) A 'type B' USB connector is used, which is a somewhat old-fashioned type: today the mini- or micro-USB connectors are more popular. Those smaller connector types were not selected for this design because those are difficult to solder manually. Right next to the USB connector is the 4-lane ribbon wire that connects to the relay board. I did solder this wire directly to the PCB, and have it in a detachable connector on the relay-board. With some searching I found this nice colored 4-lane ribbon with 0.1" (2.54mm) pitch. Other 4-thread wiring would be fine as well: the circuit doesn't pose particular restrictions.

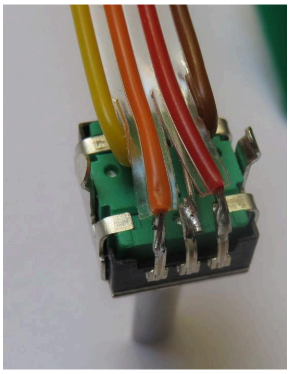
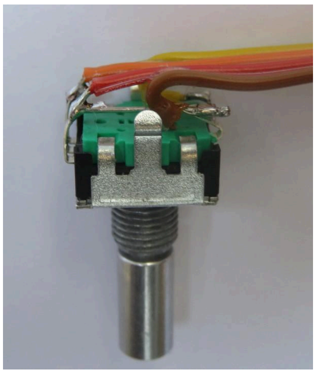


To the right from this wiring, a 5-pin header is mounted to allow connection with a PIC programmer, such as the PIC-KIT 2 or 3-series. You would only need this connector if you do your own software development, or when you buy an empty PIC and 'fill' that through your own programmer.

Mounting the rotary switch

All the way on the left in above picture you see some (ugly temporary) wiring that I tested to connect to the front rotary switch. That switch can be mounted directly on this PCB (on the front side!), or connected through some wiring for more freedom to choose another location in your front panel. Note that once the switch is soldered directly into the PCB it would be almost impossible to take it out again for mounting elsewhere. Although the switch has 5 contacts, a 4-lane wiring to the switch is be sufficient, connecting to 3 contacts (vol.A, vol.B, and Sel.B) and a shared GND (connect your wire to both pins). The Sel.A connection on the PCB is not used in today's version of the software. If you decide to solder the switch directly on the PCB of the first version (labelled 'Sept. 2010'), you need to add a small wire to connect the Sel.A and Sel.B pin locations.

The photos below show my hand-soldering of a 4-wire strip to the back of the rotary switch. The blank wire that connects from the middle contact on one side to the brown (ground) wire at the other side is not so easy to spot. I assume you can figure out yourself how to connect these four wires to the PCB into the original pin-holes of the switch. On the PCB side, you do not need the extra wire to interconnect both ground connections.

Test 1: +5V supply

Connect the 2x7V transformer to the main board (J2P), do not connect the display board, and apply primary power. Have a multi-meter ready to check some DC voltages:

- The LED L1P should light up.
- The output of U5P (pin 3, right pin) should be at 5.7V above gnd. On the PCB top side, you can pick-up 'gnd' at the right-side pin (kathode) of D4P or at the front-side pin of R20.
- On each side of R17P you should measure about 11V (this value is still high because there is no load.)
- On pin 14 of (the not yet mounted) U3C you measure about 5.0V.
- You did not hear any relay to click to 'on' state.

Test 2: +/- 20V analog supply

Disconnect/unpower the 5V supply. Connect the 2x15V transformer to the board (J1P), and apply primary power to this transformer (do not yet connect the transformer primary to the board relay output).

- LEDs L2P and L3P should light up.
- Both ends of R4P give a reading of about +24V. Both ends of R12P show about -24V.
- Pin 3 of the (not yet mounted) U3P is at +7.5V.
- Pin 3 of the (not yet mounted) U4P is at -7.5V.

Test 3: +3.3V supply

Connect the 2x7V transformer to the main board, disconnect the 2x15V transformer.
Connect the display-board through its 4-wire interface with the main board (connecting J3D with J1C). Power-up the transformer.

- The decimal dot of the right 7-segment display lights up. All other led segments remain dark.
- Pins 14 and 20 of the (not yet mounted) U1D show +3.3V. Same for pin 3 (right-most pin) of U3D.
- Pin 3 of J3D shows the power-supply voltage that is input to this board at about +5.0V.

Assembly - phase 2

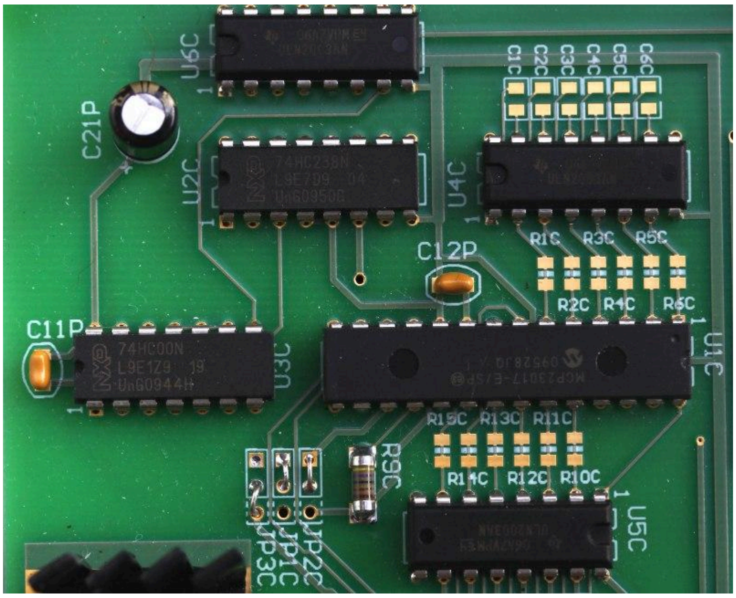
Disconnect all power, disconnect the display board.
Insert opamps U3 and U4 in its sockets. Connect the 2x15V transformer and apply its primary power.

Test 4: +/-15V supply

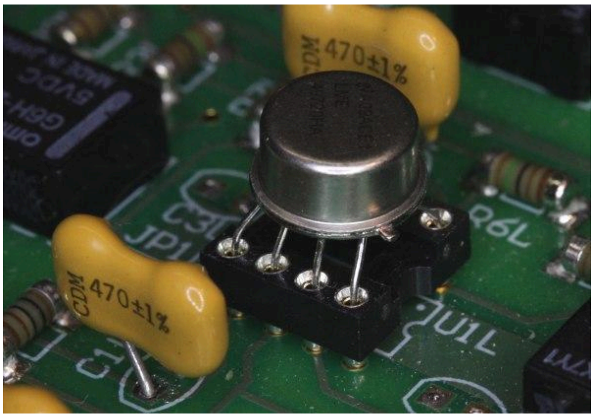
- LEDs L2P and L3P light up.
- Pins 1 and 7 of U3P show +15V.
- Pins 1 and 7 of U4P show -15V.

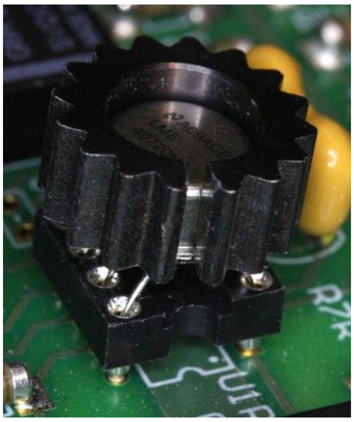
Assembly - phase 3

Now all power supplies are checked to operate as expected, we can safely continue with mounting the other ICs. Disconnect all power, disconnect the display board.
Mount all missing IC's. BE CAREFULL about their orientation!! Once soldered-in, you will not be able to get them out without destroying them. (The easiest way to remove a wrongly mounted digital IC is to first cut-off all its pins, then remove the pins one-by-one from the board with your soldering iron and a pincet.) The digital ICs on the main PCB are mounted as in the foto below:

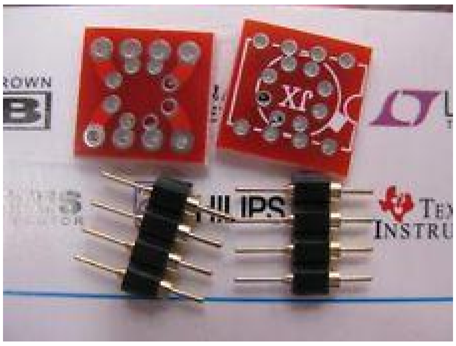


The metal-can opamp can also be plugged into a DIL-8 socket, as shown below. They tend to run quite warm, so I preferred to use a 'TO92' cooler on top of these.





However, inserting these metal-can opamps in a DIL8 socket is somewhat tricky, not to be done often. if you think you might exchange opamps once in a while for testing/experimenting, I would recommend to mount them in an adapter socket, which are easily found on ebay. (You might also look for the opamps themselves and cooling fins on ebay...)



Now connect both transformers and power them up.

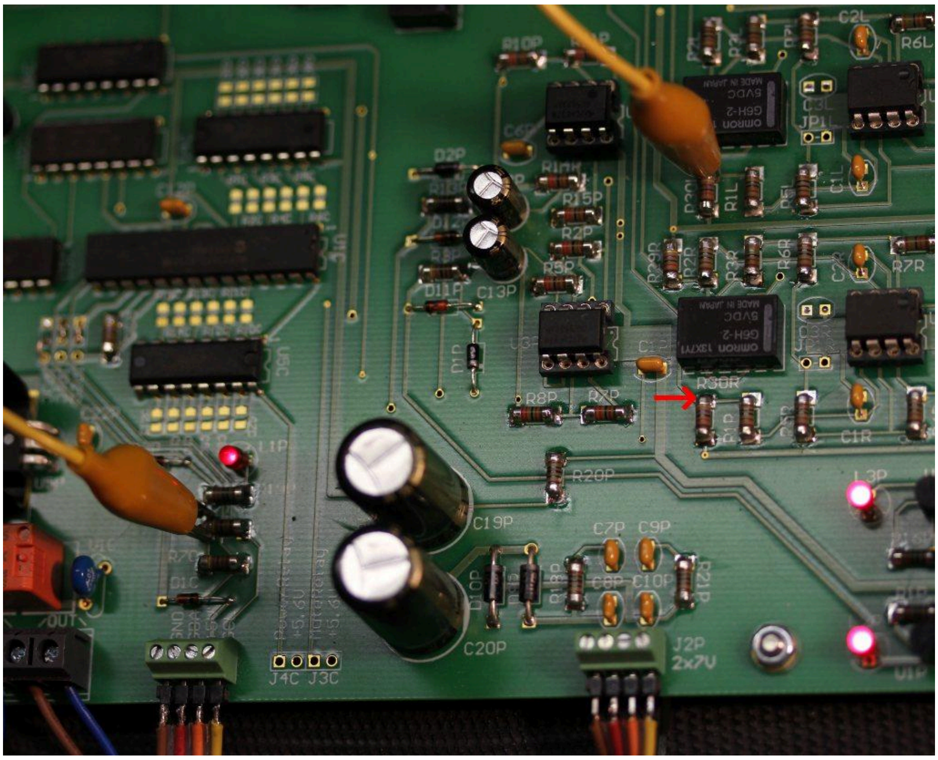
Test 5: final supply-voltage and temperature test

- The three power-LEDs light up. The display shows ‘00.’.
- Re-check the stabilized power supplies (+5V, +/-15V, +3.3V) as above.
- If you have already connected an input rotary switch (S1D) you might turn it, and you hear relays clicking.
- All 4 opamps will run just hand-warm. The 5V stabilizer (U5P) will run just hand-warm if multiple relays are switched to ‘on’ state.
- The voltages at R17P, R4P, R12P give a somewhat lower lower value as in the earlier (unloaded) measurements, like +8V and +/-20V.

Test 6: Attenuator chain

To test appropriate mounting of all resistor values, a +5V DC input voltage is applied to the cinch (single-ended) input of the amplifier, and the output voltage is measured at different volume selections.
The relaiXed is powered with both transformers, the front-panel is connected and assumed to be operational: input channel-selection and volume selection are supposed to work.

A voltage of +5V is applied as input by selecting ‘C1’ (the first input) and choosing maximum volume (64). A temporary connection is made to get the 5V from the power-supply, and inject that into the audio input, for instance according to the foto below:



The 5V is taken from the left side of R8C (or R7C or R19P) and hooked up to the top side of R30L. (For the next measurement of the ‘Right’ channel, you make the connection to the top of R30R, indicated by the red arrow). Alternatively, you can also connect any 5V source directly to the actual cinch input.

At full volume, you measure at the Left XLR output: the ‘+’ Left output shows +7.5V and the ‘-Left output shows -7.5V. When the volume selection is stepped down, the output voltage decreases approximately according to the table below:

Vol	voltage	Vol	voltage	Vol	voltage	Vol	voltage
64	7.50	48	1.33	32	0.23	16	0.041
63	6.74	47	1.20	31	0.21	15	0.037
62	6.03	46	1.06	30	0.183	14	0.033
61	5.55	45	0.97	29	0.167	13	0.030
60	4.79	44	0.86	28	0.146	12	0.026
59	4.32	43	0.77	27	0.132	11	0.024
58	3.87	42	0.68	26	0.117	10	0.021
57	3.56	41	0.62	25	0.107	9	0.019
56	3.01	40	0.54	24	0.092	8	0.017
55	2.74	39	0.49	23	0.083	7	0.015
54	2.47	38	0.44	22	0.074	6	0.013

53	2.28	37	0.40	21	0.068	5	0.012
52	1.93	36	0.35	20	0.059	4	0.011
51	1.75	35	0.31	19	0.053	3	0.010
50	1.58	34	0.28	18	0.047	2	0.009
49	1.46	33	0.25	17	0.043	1	0.008

The above table shows the logarithmic nature of the volume control: every step the voltage reduces with 0.95dB, which is approximately 11%. The actual reduction per step can be slightly larger or smaller due to rounding of ideal resistance values to the standard E12 series.

Repeat these measurements for the '-' Left channel, and the '+' and '-' Right channels.

Test 7: USB operation

You can connect the display-print to a computer with Microsoft Windows installed. (I'm sorry, I don't like Windows, actually I am an Ubuntu fan...) This USB connection is valid with either:

1. the display print only, detached from the relay PCB, or,
2. the display print connected to the relay PCB with all its power-supplies off, or,
3. the display print connected to the relay PCB with the power-supply enabled (with primary power on).

For this experiment you could probably do 1: the display-print only. If you connect it to the PC, you will notice, if everything is OK:

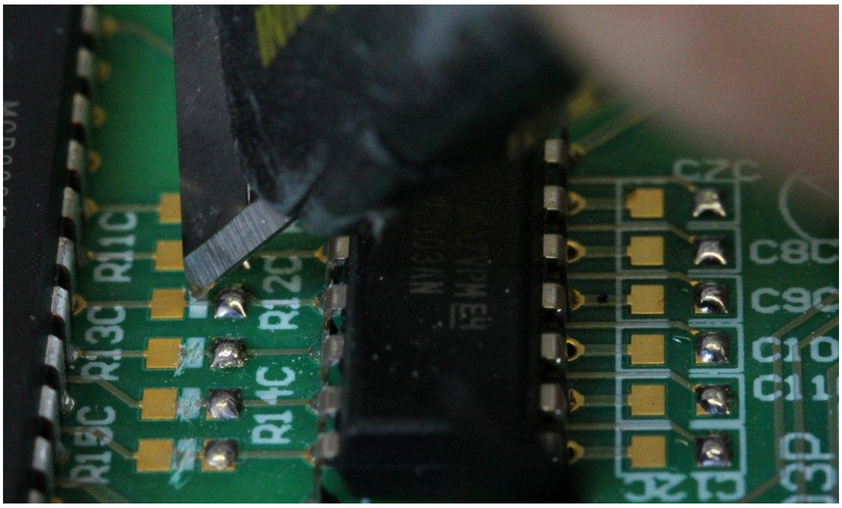
1. The display-print powers-up through the USB cable, the display shows life (like a '00' volume' or some other volume of a previous test).
2. The Windows operating system recognizes a new USB device. After some seconds it pops-up a small window, showing the 'RelaiXed2' device name. Note that this occurs only once, for a new device. Later, when re-connecting again, this pop-up is not shown anymore.
3. If you want to look further, you can check in "Control Panel->Device Manager->Human Interface Devices". A new 'USB Input Device' is added to the list. If you double-click such a device, a 'Device Properties' window opens. In the 'Details' tab of the proper device, you can find the USB 'Hardware Id' granted to the RelaiXed project as VID_04D8 ('vendor Id') and PID_FB29 ('product Id'). In Windows-7 its 'relaixed2' name is hidden under the property 'Bus reported device description'. Note that you do not need to install Windows device drivers to interface with the relaiXed board.
4. You can simply and safely unplug the USB cable again, in a running system.
5. During USB power-up, the display-print quickly and shortly showed a message 'E1'. This is an error message, indicating that it was not able to communicate with the (unconnected/unpowered) relay-PCB.
6. If you would re-do this PC connection experiment with an attached and powered relay-board, a short message 'U1' would be temporarily shown, indicating a working USB link. If you detach the USB cable (while powering the display through the relay board) it would shortly show a message 'U0' to denote losing the USB connection.

To actually communicate between this RelaiXed display PCB and your PC, a dedicated Windows application program was made. This application can be downloaded through the project website. You don't need this application for normal Relaixed operation. However, if you would (later) like to update the software (firmware) of the PIC microcontroller, you can use this application program to upload the new version in the PIC: see a later chapter in this manual how to do that. For more technical information on the software developed for this RelaiXed project, you can check the software manual, available separately through the website.

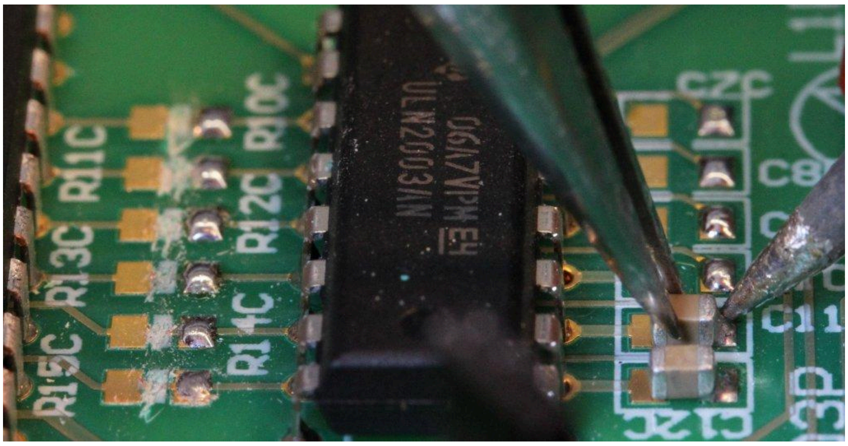
Assembly - optional phase 4

The schematic page 'Digital Control' shows RC-circuits that are optional to mount (R1C-R6C, R10C-R15C, C1C-C12C). This extension reduces the steepness of the voltage-transient on the driver lines of the volume-switch relays. Reducing those transients suppresses the undesired effect that small remnants of those edges appear on the audio outputs as noticeable clicks. In this Relaixed2 design, I took some extra care to minimize such parasitic coupling, but it is not yet truly absent. Audible clicks probably still occur if you attach a high-power amplifier. You could first build your relaiXed and experiment a while to see if this effect bothers you, before deciding to install these extra RC filters. Of course, if you are not afraid to handle these small components, you can mount these among the first components (in phase 1), which might be a more convenient order.

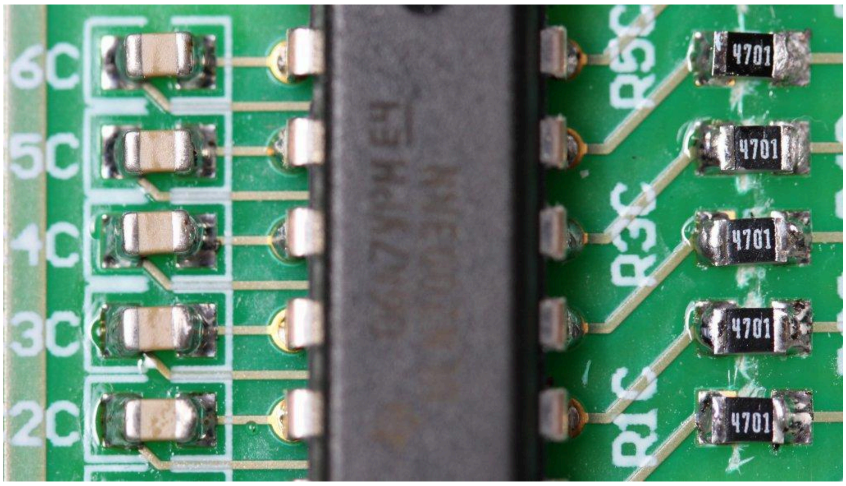
These RC filters are built with smaller SMD components ('0805' form factor), as I thought many of you would not need them, and I didn't want to sacrifice much PCB space and end-up with a less nice component layout. BEFORE installing the resistors, you MUST scratch/remove a short-cut wire that is present on the PCB beneath the resistor footprint. (This connection exists to have a working circuit when these RC filters are not installed.) See my pictures below:



Apply solder on one side of each SMD component. Scratch the tiny connection underneath the resistor footprint, painted white on the PCB. (Don't scratch anything on the capacitors!) Probably you want to check with a multimeter that the short-circuit underneath the resistor is really gone.



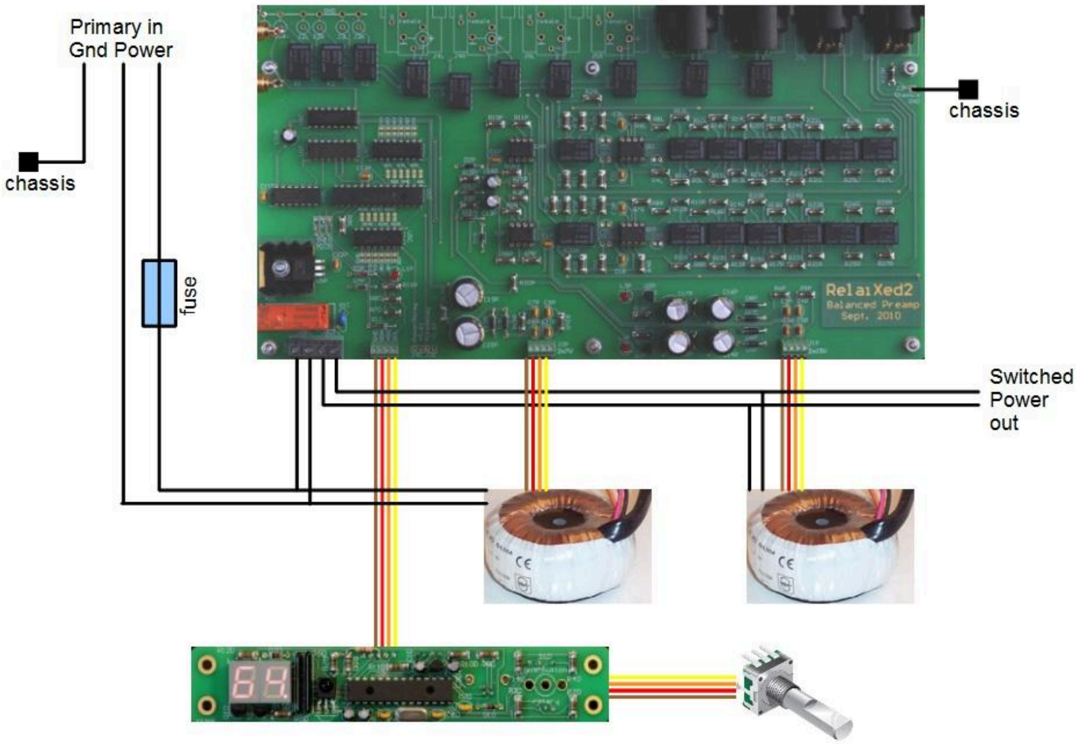
Insert the components: hold in place, reheat existing solder, press-down the component to obtain a flat mounting. Afterwards, add solder to the other side of the components.



Final mounting result. Note that these SMD resistors are marked with their value 4.7K ($470x10^1$), many types don't have that anymore.

Assembly - Chassis wiring

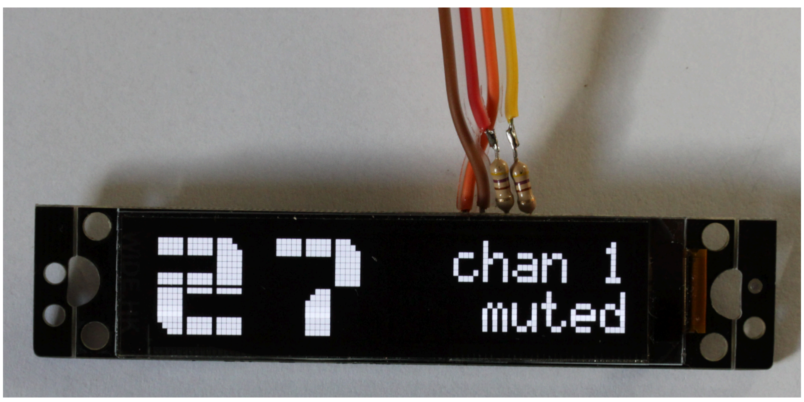
Below is a schematic overview of the inter-board routing:



I prefer (recommend) to use audio in/out connectors that are isolated from the chassis back-panel. For safety-reasons, your chassis is supposed to be connected to the ground that comes with the mains plug. To avoid direct ground-loops between various devices in the audio set through power-gnd and audio-gnd, I would not create a direct ‘hard’ connection between audio-gnd and the chassis, but instead make this a ‘soft’ connection through a resistor. That is indicated in the top-right of above picture. Note that XLR connectors often provide a chassis-connection as fourth pin (next to audio-plus, audio-minus and gnd). The PCB does not connect that chassis-pin to audio-gnd. If you prefer otherwise, you are free to add a small wire on the bottom of the PCB to connect the XLR chassis and gnd pins.

The ‘switched power out’ can be optionally used to connect other equipment, such as your power amplifier. Alternatively, you can take the 5V relay control signal to outside of the preamplifier chassis, and somewhere externally use a second power relay to switch other equipment.

Adding an OLED character display



Rather a long time time ago, I had started with another display, as alternative to the 2-digit 7-segment display that is present on the front controller PCB.
I made a choice for this 2x16 character OLED display sold here:
http://www.ebay.com/itm/IIC-I2C-1602-16x2-OLED-Module-Display-For-Arduino-PIC-AVR-ARM-/161162845021
Reasons for choosing this display are:

- OLED displays have a marvellous contrast.
- The white-on-black text is nice for its bare-bones simplicity and readability
- Its i2c serial interface connects easily to the Relaixed without modifications.

Although an early version for software-support of this display was made more than a year ago, the results were not offered because the communication to this display appeared to be unreliable. With some new effort, this got resolved now, and an extra software update to drive this display was released. You would need to upgrade the firmware in your PIC controller to (at least) the release of 2016-09-10, or svn83. You can download this release as 'RelaixedApp-20160910.hex' from the project sourceforge site. This release will work with either one of the displays alone, or with both displays together.

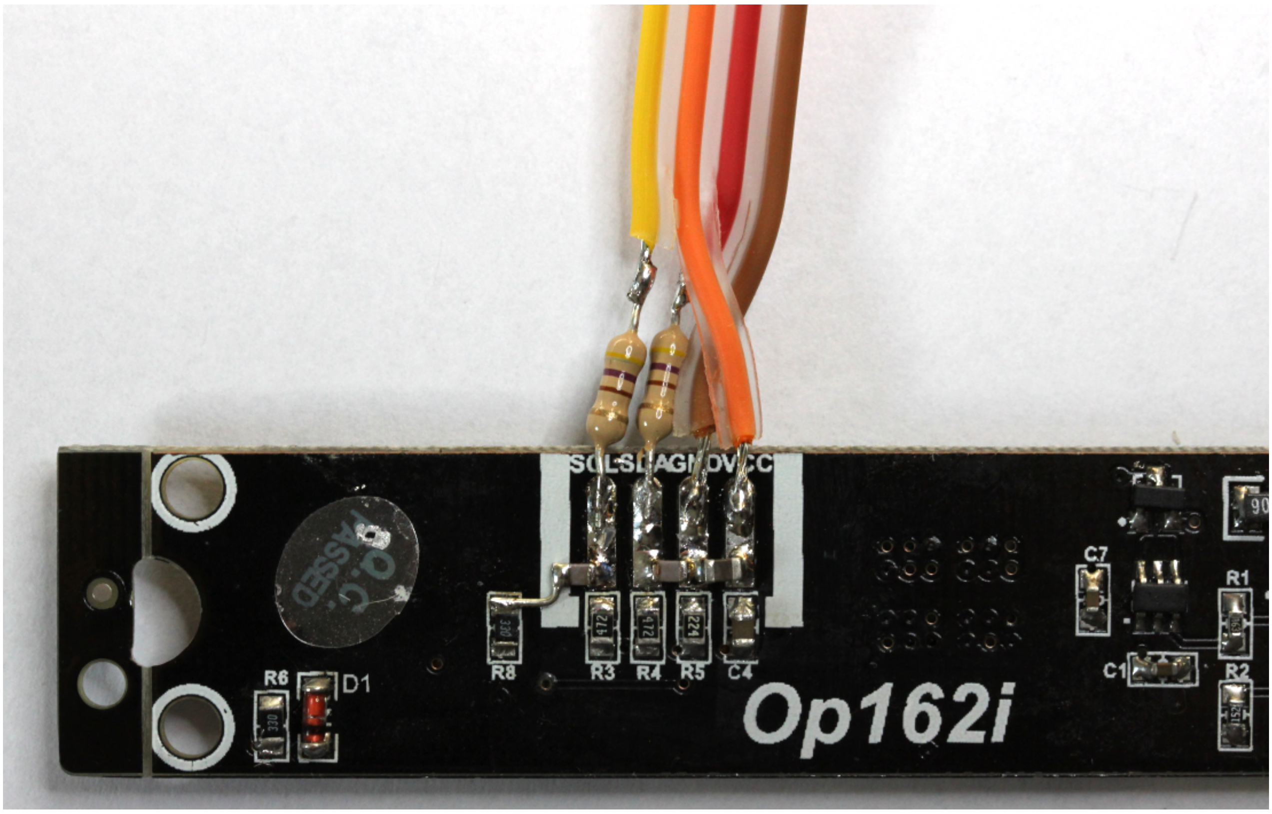
On the RelaixedSMD preamplifier, or when you use two Relaixed-Passive relay boards on a single controller, the left-right balance control is also indicated in the display: on the location where the upper photo shows 'muted'.

To obtain a reliable display operation, it seems required to add a few components to the display interface. I added on the back of the display connector:

- A 10uF capacitor between its VCC and GND (same type as used elsewhere in the RelaixedPassive)
- Two 100pF capacitors, one on each of the i2c clock and data lines (SCL and SDA), with their other pin to Gnd.
- Two 470ohm resistors, in series with the clock and data lines.

This was the original backside of the display:

This is how it looks after my extra components and 4-wire bus connections:



Note the three small (0805) capacitors that are mounted horizontally across the solder terminals: from left-to-right first the two 100pF caps, then the 10uF cap. Obviously, you might create a different layout. After twisting the wires in the 4-wire flat-cable as in above photo, they are in proper order to simply connect into the Relaixed i2c header, with the brown wire at position 1 of the connector (pin 1 is GND). It is connected in parallel to the other cable that connects the front-PCB to the main PCB.

Operating instructions

Using the rotary switch

This RelaiXed design is operated through a single rotary switch that has an additional ‘push button’ control built-in. Merging multiple control functions into a single knob was chosen as result of my personal preference: It keeps the front-panel visually simple and sleek, and it relieves me from doing a more complex mechanical job which is not my expertise...
Note that this new design has an additional switch control connection on the PCB which would allow operation with two rotary switches, however the current embedded software does not yet support that.

The RelaiXed is in one of two major operating modes:
Standby mode:
In standby mode, the display is off, except for one ‘decimal dot’ in the display which remains on. The analog power-supply is switched off, all relays are off (unpowered).

The RelaiXed does not react on turning the volume-control. It ‘wakes-up’ to operational mode by clicking the knob (press and release), or through the remote control power-button.

Operational mode:
When entering ‘operational mode’, the analog power-supply is powered-up. The audio output remains in ‘mute’ mode for about two seconds. After these initial two seconds, the unit becomes operational.

- Turning the control-knob updates the volume. The volume values range from ‘00’ to ‘64’.
- Clicking the control-knob (quick press-and-release) moves the audio input selector to the next channel, with a cyclic wrap around. The display shortly shows ‘C1’ to ‘C6’ and then reverts back to displaying the current volume.
- Pushing and turning the control-knob adjusts the left-right balance. The display shows ‘-9’ to ‘ 0’ to ‘ 9’. This corresponds to the volume-difference between the left and right channel. Releasing the knob reverts back to showing the volume.
- Pushing the knob and holding it still and pressed for about two seconds puts the RelaiXed in stand-by mode. The display first shows ‘00’ denoting mute mode, and then switches itself off.

The last chosen volume and input channel are stored in the microcontroller internal flash memory. At the next power-up the RelaiXed comes back to this previous mode.
You might directly switch the RelaiXed primary power input off, instead of merely switching back to ‘standby’ mode. If you switch the primary power off, the microcontroller will quickly enter ‘mute’ mode to silence the audio output just before the analog power-supply is faded-out, ensuring a graceful power-down.

Using the IR control

The RelaiXed IR receiver, in the current software release, can decode the following IR signals:

- The Philips RC5, RC5X, and RC6 protocols
- The Sony SIRC protocol in its 12-bit and 15-bit version

This flexibility allows the RelaiXed to be used with a wide range of IR control handhelds, among which all multi-brand controllers.

Within each of these protocols, received commands have a device-code and a button-code. By default (when the PIC is newly programmed), the RelaiXed reacts on each protocol and each device code, interpreting the button-code only. This allows you to quickly find-out if your IR handheld is compatible with the RelaiXed receiver. To avoid that the RelaiXed reacts on too many devices, and interprets commands destined for another of your devices, see the next section.

To support interoperability with various types of IR controllers, the RelaiXed functions are activated by various button codes. The table below lists the active button-codes:

Function	Activated by buttons
Volume up	Volume up, Cursor up
Volume down	Volume down, Cursor down

Balance left	Balance left, Cursor left
Balance Right	Balance Right, Cursor right
Input select Up	Channel/Program up
Input select down	Channel/Program down
Input select 1..6	Digit 1 .. 6
Mute	Mute, Pause, Stop
Unmute	Play, Volume up, Volume Down
Power down	Power, Standby
Power up	Power

Restricting IR device target

By default, loaded with a fresh firmware, the Relaixed IR receiver can decode and handle various manufacturer IR protocols. As result, it will react on everything that is recognized. A special initialization sequence will restrict the RelaiXed receiver to react on one chosen protocol and target device code only. If you have several IR-controlled devices around, this restriction is probably needed to prevent the Relaixed from acting upon commands that you intended for one of your other devices.

- Within the first two seconds of power-up, the display shows '00'. If you press 'volume up' followed by 'volume down' inside this two-second period, RelaiXed will restrict itself to react only on the IR protocol and target device ID that came with these buttons. To confirm successful restriction, the display will shortly show 'rc'.
  Note that this activation only occurs at real power-up from primary power, not on power-up from standby. (Or, in other words, only after the PIC comes out of reset state.)
- To reprogram the RelaiXed to bind itself to a different IR controller and device ID, the above sequence can be repeated with the new controller: the power-up initialization sequence is not restricted to an earlier bound controller.
- Note that you cannot go back to the mode where RelaiXed will fully operate on any IR controller without filtering on a device ID, except by re-programming its firmware.

Upgrading to a new software revision

The Relaixed project software consists of three components, available for download at http://sourceforge.net/projects/relaixed2/files/:

1. The 'RelaixedApp-*.hex' embedded software ('firmware') that is stored internally in the PIC microcontroller that controls the RelaiXed operation. It is stored in the PIC flash memory at addresses above 2000(hex).
2. The PIC 'RelaixedHIDBootloader-*.hex' embedded software. This software operates the PIC USB connection and contains a self-programmer function to allow re-programming the 'RelaixedApp' through a USB connection to a PC. It is stored in the PIC flash memory at addresses below 2000(hex).

3. The PC-bound software application ‘Relaixed-*.exe’ that is used to communicate with the Relaixed-PIC, mainly for reprogramming the ‘RelaixedApp’. Next for programming, it has an additional feature to support relaixed application programmers: it will do logging of run-time Relaixed debug messages.

If you order a Relaixed microcontroller from me (typically with the PCBs), the microcontroller will contain both the upper 1. and 2. software components, and is ready to use.

If you later want to upgrade your RelaiXed to a newer firmware version, you need to download 3. and (the newer version of) 1. Through a USB connection, you can now install 1. inside the PIC.

If you bought a blank microcontroller from the shop, you will need to use a real programmer device to first ‘program’ 2. into the PIC chip. After doing so, you can install 1. through the use of 3.

All the above mentioned software is available in source-code form. You can browse through the code at the following link: https://sourceforge.net/p/relaixed2/code If you want to help me with co-development of this code, you should download the source-code while maintaining the code ‘SVN’ revision system. For further information see the Relaixed software development manual.

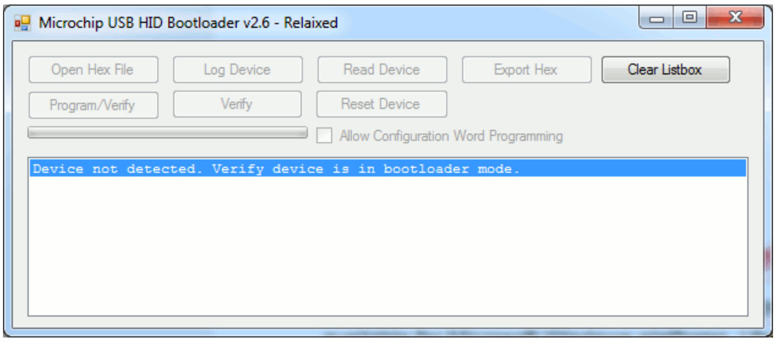
Installing the PC-based Relaixed interface application

Currently, the PC-based application to communicate with your RelaiXed is only available for Microsoft Windows platforms. I feel really sorry about that....
You can download the application from this link:
http://sourceforge.net/projects/relaixed2/files/Relaixed-1.1.exe/download

You can store this ‘.exe’ file anywhere on you PC, and launch it by clicking it: it does not need further install scripts to run. It will run on various Windows versions, I tried both ‘XP’ and ‘Windows-7’. It is itself a 32-bit application, but will run fine also in a 64-bit Windows version. Its main prerequisite is the availability of the Windows .Net runtime environment. This is available by default in Windows-7. In older Windows you could install it from e.g.:
http://www.microsoft.com/download/en/details.aspx?id=17851

On some PCs with an older Windows version, the Relaixed application will not start-up, not show the window as below, but immediately crash with an error. This is due to some missing code-libraries that this application assumes present in the Windows installation folders. If the application does not start-up in your PC, you should install the missing libraries from:
http://www.microsoft.com/download/en/details.aspx?id=5555

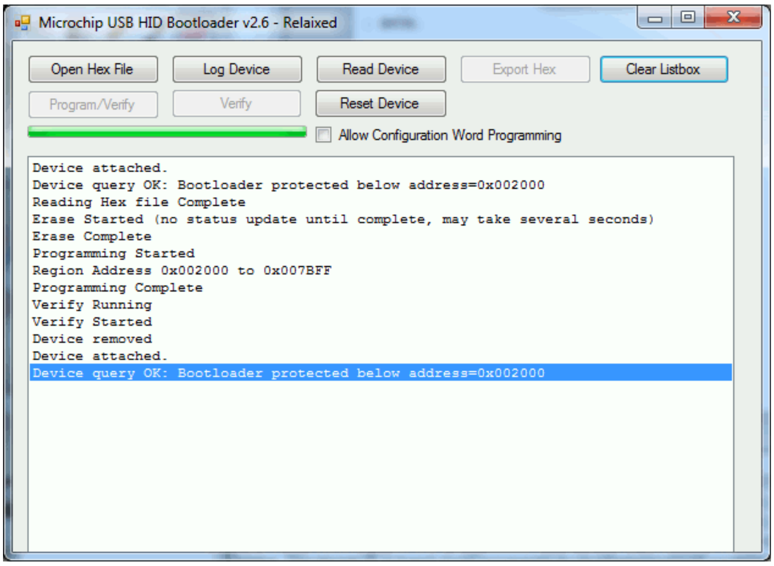
If you launch the application without any (USB-) connection to a RelaiXed frontpanel, it will come up with a window such as:



So it complains (rightfully) that the RelaiXed device is not detected. The 'is in bootloader mode' message is actually not appropriate, and is a remnant from the original Microchip software: the RelaiXed does not have such mode.

Updating your microcontroller

- You download the new firmware version from http://sourceforge.net/projects/relaixed2/files/, named as RelaixedApp-20110803.hex (or with some newer date in its name). Store this file somewhere on your PC.
- You connect the Relaixed frontpanel with a USB cable to your PC. You can leave the frontpanel connected into the rest of your pre-amplifier. Preferably you do not power-up the pre-amplifier. Its display will light-up from USB power. The device should be automatically/silently recognised by Windows as USB HID device, as described in the Assembly chapter, 'test 7 - USB operation'.
- You launch the forementioned RelaiXed interface application. Its text window should now show messages 'Device Attached' and 'Device query OK'. It is also OK to first launch the application and then plug-in the USB.
- In the application you now click 'Open Hex file' and provide it the just downloaded 'RelaixedApp-*.hex' file.
- You click the 'Program/Verify' button. If the reprogramming works fine, you end up with a screen like:



The log shows that the PIC passes through the following phases: erase, program, verify, restart. This indicates that you are done, and can use your RelaiXed with the new firmware version.

Unfortunately, this re-program process is somewhat unreliable. You would have a reasonable chance that it does not pass through all those four phases and becomes stuck earlier...
I am not yet sure whether that is due to errors in the original Microchip code, or due to updates I made in this code, but the original source code already contained comments regarding risc of deadlock. That source code from Microchip is really badly written and structured, and seems to me in terrible shape, beyond repair. I have some plans to rewrite a totally new and cleaner application as replacement, that would also not be bound to just Windows. However, that might take a while....

The following text helps you to recover from a premature deadlock during programming:

Deadlock during 'Verify'

Normally this means that the programming went fine. I have never seen that the programming itself completed but was somehow faulty, causing a verification error. So you might just try and use the result. The deadlock itself is removed by detaching the USB cable, unpowering the Relaixed front panel, and closing the bootloader PC application. You can then reconnect the USB and restart the app. You might want to try now 'Open Hex File', followed by 'Verify' to check the content of the embedded flash.

Deadlock during 'Programming'

Luckily, this situation seems more rare. You should definitely retry: detach the USB, unpowering the frontpanel, close the PC application, and start again. In a few cases, the mis-programming leads to a faulty flash content where the PIC refuses to make a valid USB connection with the PC: the device is not recognized anymore. If this occurs, you cannot simply redo the same procedure.

Failsafe bootloader access

To recover from such a faulty flash content, I provided the PIC bootloader with a **failsafe restart** method: Your pre-amp is unpowered, you keep the control-knob (switch) pressed, you plug in the USB, and only then release the control knob. The display now comes up with a single LED-segment lighted up. The PC application will now definitely connect to the Relaixed again, irrespective of earlier errors, and you can safely redo the programming with a 100% chance of success.

Programming an empty microcontroller

If you bought an empty microcontroller, or if you want to replace the microcontroller bootloader itself, you need to attach a programmer device. The current software configuration has reserved the lower 2000(hex) bytes in the on-chip flash for a USB HID bootloader, which is plenty of space. The standard bootloader from Microchip was modified by me for different reasons, explained in the software manual. This modified bootloader is available from the project sourceforge website:
http://sourceforge.net/projects/relaixed2/files/RelaixedHIDBootloader-20110525.hex/download
(or a newer date in this name).
You provide this file to the programmer software, and connect the programmer to the front-panel as in the picture below. Note that the front-panel has a standard 5-pin programmer interface, but this Pickit2 has an extended 6-pin interface. The required power can be provided by the programmer itself, there is no need for powering-up the front panel.


MICROCHIP
PICkit™ 2
Power
Target
Busy