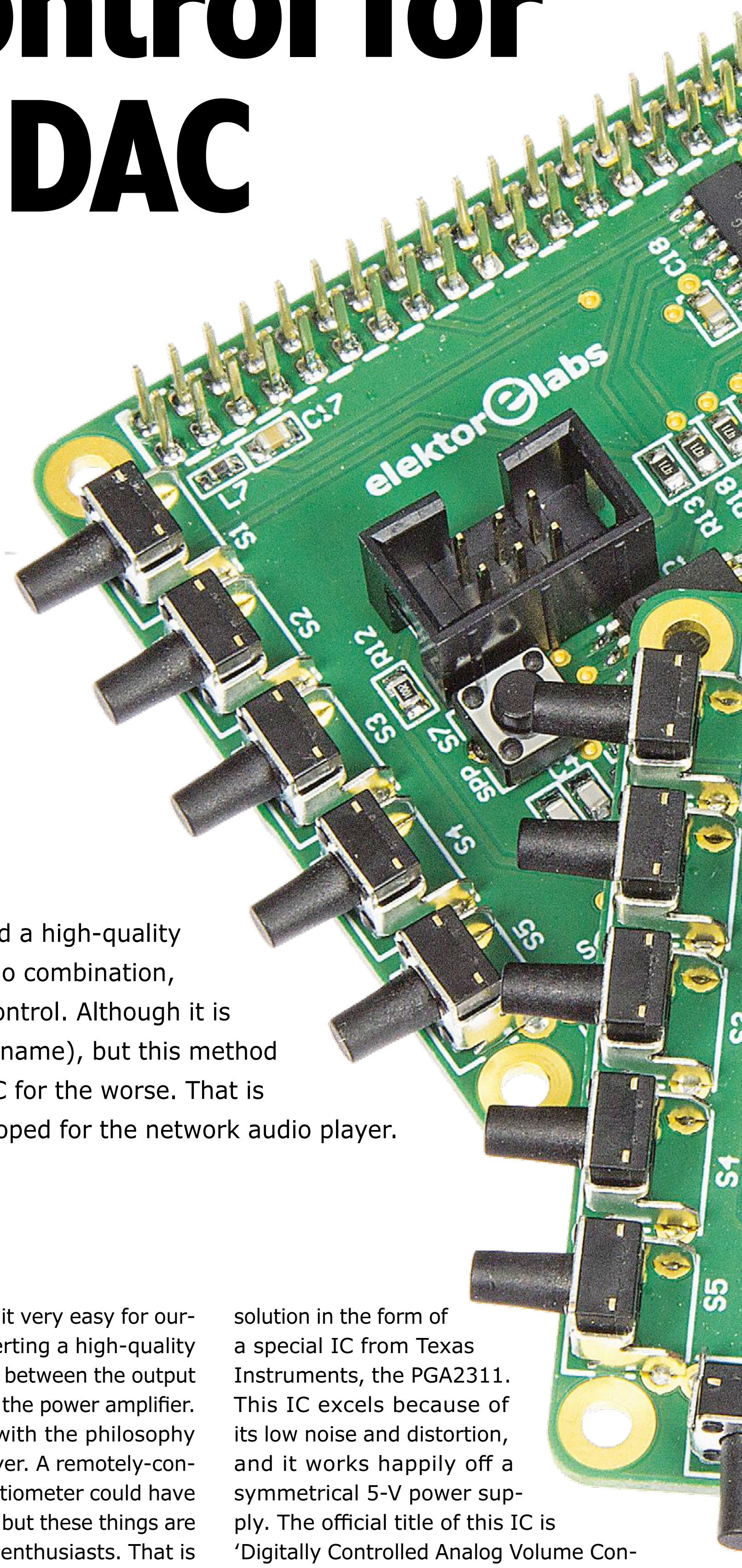
eLABS PROJECT

Volume Control for RPi Audio DAC

loud and quiet in high quality



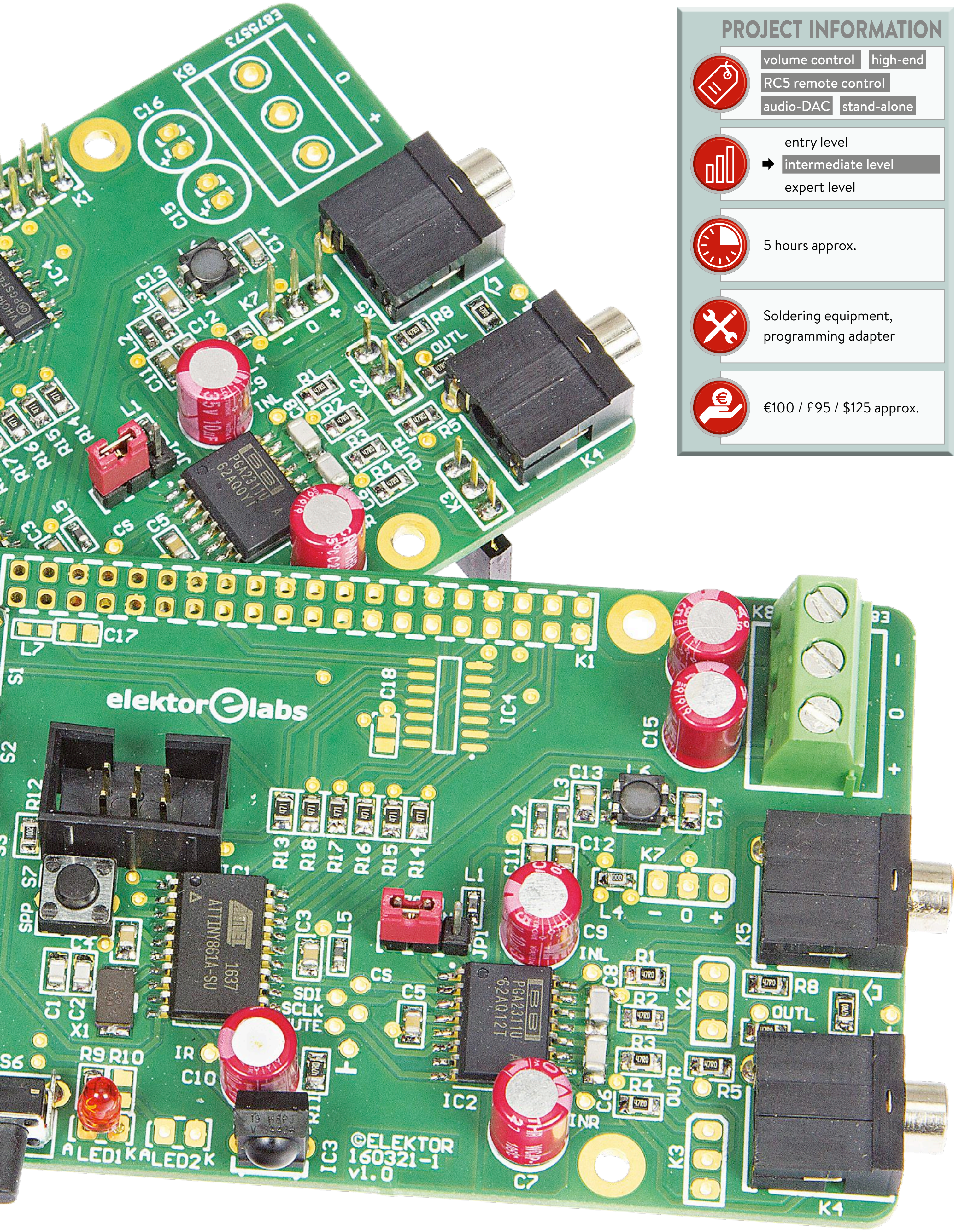
By **Ton Giesberts** (Elektor Labs)

To the delight of many, the previous issue of Elektor had a high-quality network audio player based on the Raspberry Pi/Volumio combination, which however was not provided with a 'real' volume control. Although it is possible to control the volume via Volumio (what's in a name), but this method seriously affects the high-end characteristics of the DAC for the worse. That is why we now describe a volume control especially developed for the network audio player.

Characteristics

- High-end volume control
- Suitable for Raspberry Pi + audio DAC
- Compact sandwich construction
- Uses an RC5 remote control
- Also suitable as stand-alone volume control

We could have made it very easy for ourselves by simply inserting a high-quality stereo potentiometer between the output of the audio DAC and the power amplifier. But that doesn't fit with the philosophy behind the audio player. A remotely-controllable motor potentiometer could have been a solution here, but these things are scarcely available to enthusiasts. That is why we chose a high-quality, solid-state solution in the form of a special IC from Texas Instruments, the PGA2311. This IC excels because of its low noise and distortion, and it works happily off a symmetrical 5-V power supply. The official title of this IC is 'Digitally Controlled Analog Volume Control'; the control of this IC takes place



PROJECT INFORMATION

volume control · high-end · RC5 remote control · audio-DAC · stand-alone

- entry level
- ➨ **intermediate level**
- expert level

5 hours approx.

Soldering equipment, programming adapter

€100 / £95 / $125 approx.

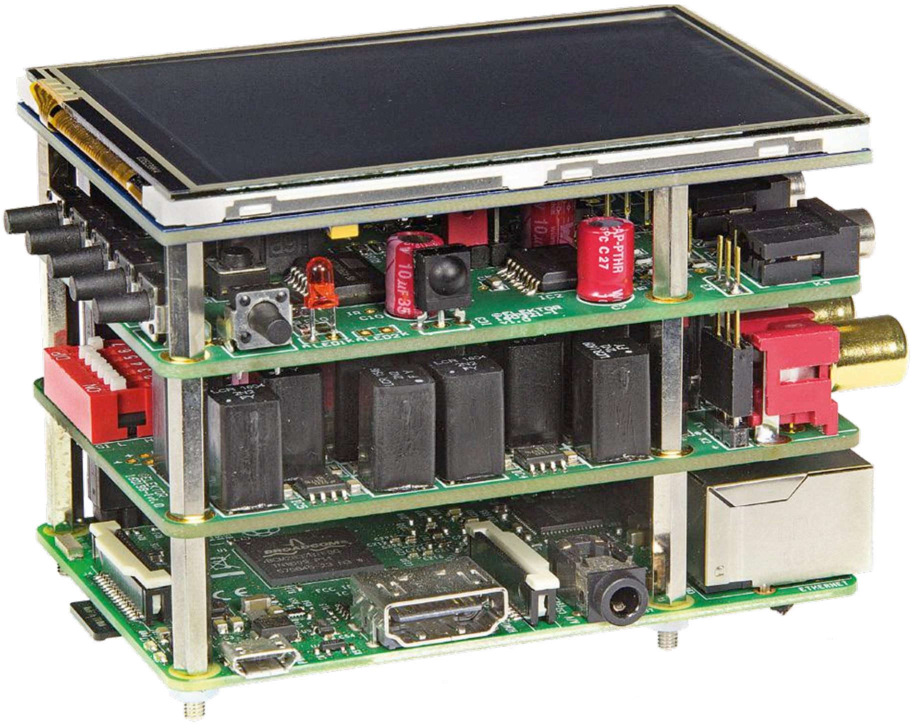


Figure 1. The ‘Big DAC’ in full glory.

across a serial interface connection. Additional features of this IC include a zero-crossing detector and a mute function. For the control we use a microcontroller from Atmel’s ATtiny series.

General design

Because this volume control was designed, in the first instance, to be used with the audio-DAC from the previous edition, its circuit board has been designed such that it can simply be plugged into the DAC circuit board. This way a thick sandwich of Raspberry Pi, audio-DAC, volume control and display is created (disrespectfully called a ‘Big DAC’, after a well-known product from the fast-food chain with the Scottish name), as can be clearly seen in **Figure 1**.

With the aid of half a dozen pushbuttons (a seventh button is used to store the default volume setting), Volumio and/or another application that is running on the RPi can be controlled. In addition, there is (obviously) provision for using an IR-remote control that uses the standard RC5 protocol.

And now then the good news for all those readers who are searching for an ‘independent’ usable volume control (that is, without RPi and audio-DAC): that is perfectly possible with the design described here! You can even omit a few parts...

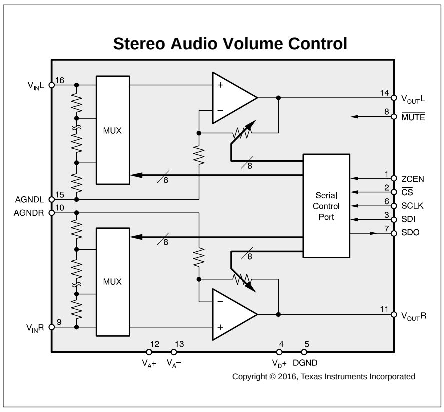


Figure 3. Amplifiers with variable gain are used for the volume control.

The schematic

We’ve drawn the complete schematic for the volume control in **Figure 2**. You may have expected us to describe the circuit as a block diagram first, but that is not necessary — to the untrained eye the schematic looks complicated, but appearances can be deceiving.

We attempt to deal with the description systematically, and start with the beginning: the power supply (after all, without a power supply even the most artistically assembled circuit board is useless).

The power supply

If the volume control is used as an ‘add-on’ to the audio-DAC then a separate power supply is not necessary. For the volume controller, the required (analog) +4.74 V and –4.81 V (abbreviated in the schematic to +4V7_A and –4V8_A) are supplied by the DAC board; the same is true for the +4.7 V for the microcontroller (+4V7_D in the schematic). These power supply voltages enter via connector K7, which is soldered to the bottom side of the circuit board (refer later on) and connects with K6 on the DAC board. The +3.3 V, which is only required for the inverters (IC4) that drive the GPIO pins of the expansion connector, is generated by the Raspberry Pi (pin 1 of the expansion connector).

The power supply voltage for the PGA2311 is filtered by the common-mode filter coil L6 and inductors L2, L3 and L4. Filter inductor L5 filters the digital power supply voltage for the microcontroller (IC1) while L7 assumes this task for the 3.3-V power supply. Inductor L1 ensures sufficient separation between the digital and analog power supply voltages for the PGA2311. Inductor L4, finally, has a very important job: it prevents the creation of an RF ground-loop, because both the power supply ground (K7) and the ground connections of K2 and K3 are already connected together on the DAC board.

When used as an add-on to the DAC board, connector K8 and capacitors C15 and C16 do not need to be fitted.

On the other hand, when using the volume control for other applications, the components IC4, C17, C18, L7, K2, K3 and K7 do not need to be mounted. Additionally, in this case it is also best to replace L4 with a 0-Ω resistor. Note: for stand-alone use, a separate, symmetric 5-V power supply is required!

The PGA2311

For those who are interested, in **Figure 3** we have drawn the internals of the PGA2311. It is immediately clear that we have two (because it is stereo) amplifiers, the amplification of which are set by a set of (serial) control inputs. The control signals ($\overline{CS}$, SCLK, SDI and $\overline{MUTE}$) come from microcontroller IC1.

A brief remark about the connection of ZCEN (Zero Crossing Enable). The PGA2311 has a zero-crossing detector; the idea behind this is (when this function is enabled) that the change in volume only takes effect after the next positive-going zero-crossing of the audio input signal. In this way spurious audio artifacts as a consequence of the change in volume can be minimized. On the circuit board ZCEN can be connected via a jumper to either +4.7 V or ground, so that you have the choice whether to enable or disable this functionality.

For a more detailed description of the inner workings of the PGA2311 we refer you to the datasheet [1]. Between the microcontroller and the PGA2311 we provided test points on the circuit board for the serial control signals (~~CS~~, SDI, SCLK and ~~MUTE~~).

The input and outputs go via resistors

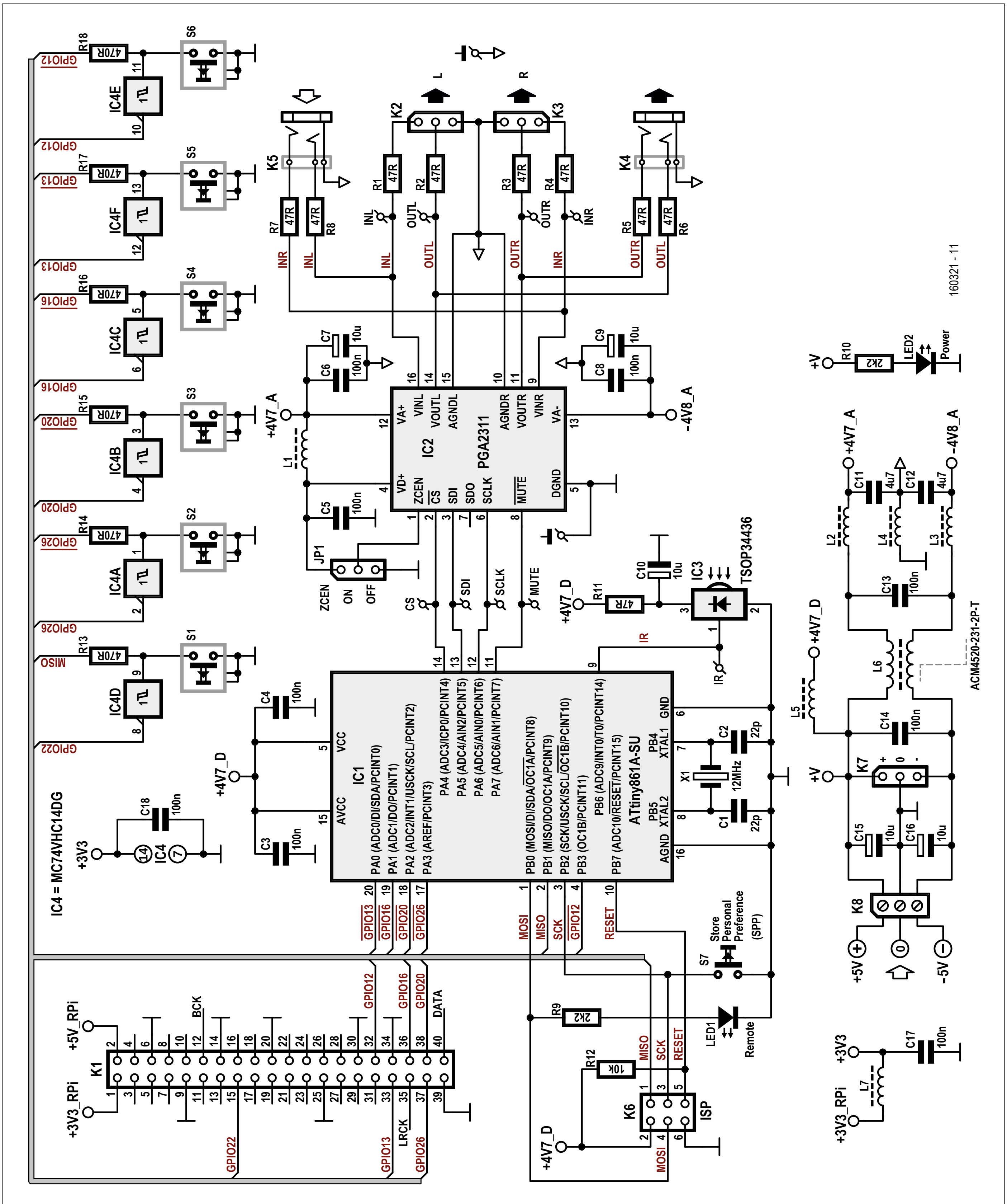


Figure 2. Unsurprisingly, at the heart of the circuit sits a microcontroller.

of 47 Ω (R1 through R8) to two different connectors. A pair of so-called stack-through-connectors (K2 and K3) connect the inputs and outputs of the DAC and the volume control (in the same was as is done for the power supply). The inputs and outputs of IC2 also go to two 3.5-mm audio connectors (K4 and K5) on the circuit board. This means that the unprocessed DAC signal is available using a 3.5-mm jack (K5) or via the cinch connectors on the DAC board. The output signal from the volume control is available on K4.

When the board is used as an 'independent' volume control, K5 is the input for the PGA231 and K4 is the output. The 47-Ω resistors protect the outputs from

COMPONENT LIST

Resistors
R1,R2,R3,R4,R5,R6,R7,R8,R11 = 47Ω, 125mW, 1%, SMD 0805
R9,R10 = 2.2kΩ, 100mW, 5%, SMD 0805
R12 = 10kΩ, 100mW, 5%, SMD 0805
R13,R14,R15,R16,R17,R18 = 470Ω, 100mW, 5%, SMD 0805

Capacitors
C1,C2 = 22pF, 50V, 5%, C0G/NP0, SMD 0805
C3,C4,C5,C13,C14,C17,C18 = 100nF, 50V, 10%, X7R, SMD 0805
C6,C8 = 100nF, 25V, 5%, C0G/NP0, SMD 1206
C7,C9,C10,C15,C16 = 10µF, 35 V, 0.04Ω, 6.3mm max. diam., pitch 2mm or 2.5mm, Würth Elektronik 870055673001 (WCAP-PTHR Series)
C11,C12 = 4.7µF 25V, 20%, X5R, SMD 0805

Inductors
L1,L2,L3,L4,L5,L7 = 600Ω @ 100MHz, 0.15Ω, 1.3A, SMD 0603, Murata BLM18KG601SN1D
L6 = ACM4520-231-2P-T (TDK), 2 x 0.05Ω, 230Ω @ 100MHz, 2.6 A, SMD

Semiconductors
LED1 = red, T-1, 3mm
LED2 = green, low-power, SMD 0805
IC1 = ATtiny861A-SU, SOIC-20, programmed with firmware # 160321-11 (or 160321-12)
IC2 = PGA2311UA, SOIC-16
IC3 = TSOP34436, 3-pin epoxy package
IC4 = M74VHC14DG, SMD SOIC-14 (ON Semiconductor only)

Miscellaneous
K1 = 40-way (2x20) GPIO stacking socket (female), extra long
K2,K3,K7 = 3-way bus strip, vertical, 0.1" pitch (mounted on the bottom side of the board), with long stack-through pins
K4,K5 = 3-way stereo jack, 3.5mm, PCB mount, e.g. Lumberg KLBR 4
K6 = 6-way (2x3) boxheader, vertical, 0.1" pitch
K8 = 3-way PCB screw terminal block, 0.2" pitch, 630V
JP1 = 3-pin pinheader, vertical, 0.1" pitch
LED1 = 2-pin pinheader, vertical, 0.1" pitch
JP1 = jumper, 0.1" pitch
S1,S2,S3,S4,S5,S6 = pushbutton, right angle, SPST, FSMRA4JH TE Connectivity
S7 = pushbutton, SPST, FSM4JRT TE Connectivity
X1 = 12MHz miniature quartz crystal, 5x3.2mm, SMD, Cload 18pF
4 pcs 17mm M2.5 spacer (05.12.173 Ettinger)
4 pcs 6mm M2.5 bolt
4 pcs M2.5 nut
PCB 160321-1 v1.1

Optional
To support the LC display (instead of 4 bolts): 4 pcs 14mm M2.5 spacer (Ettinger 05.12.143)

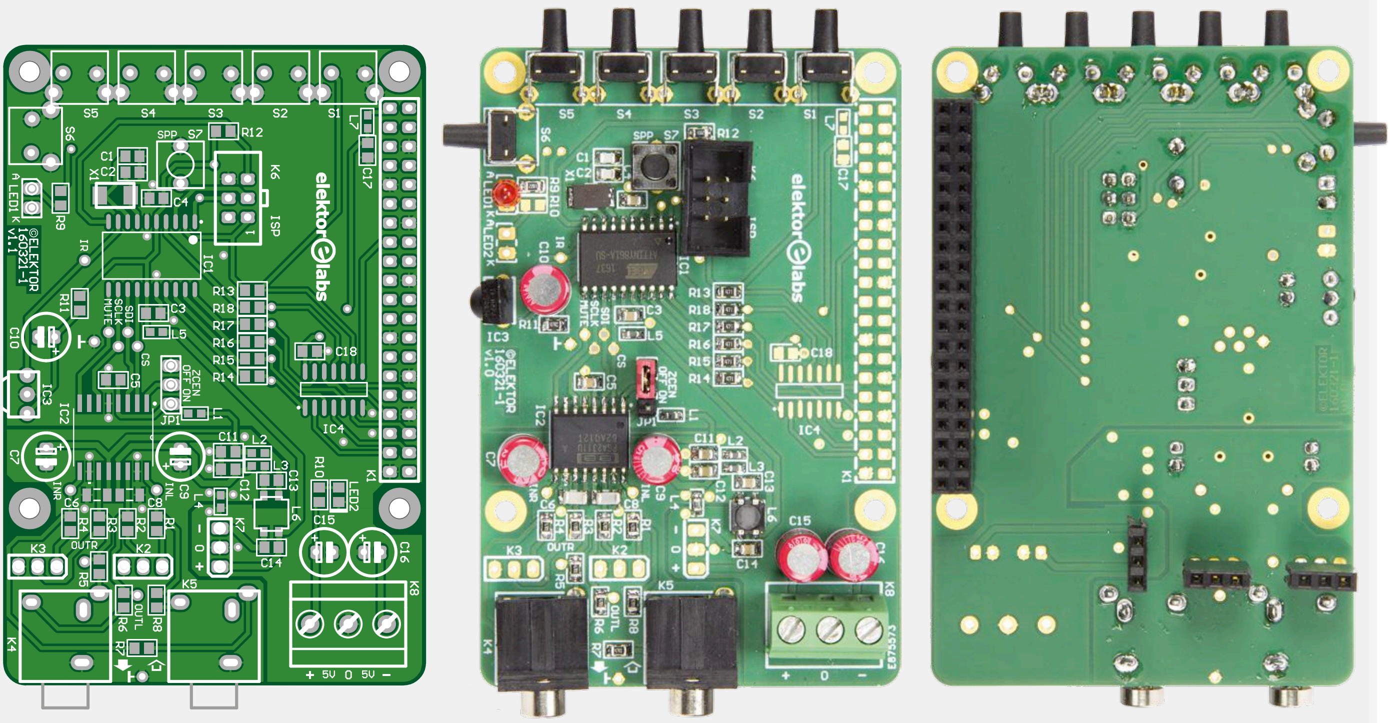


Figure 4. The circuit board for the volume control fits perfectly on the circuit board for the audio DAC.

capacitive loads and create a separation between the various connectors.

Construction

For the volume control we designed a circuit board with dimensions that are identical to those of the Raspberry Pi and the audio DAC (**Figure 4**). The board can be sandwiched with the RPi and the audio DAC; after that the display can be plugged into the entire assembly. **Figure 5** gives an impression of the complete assembly.

Note: for connectors K2, K3 and K7 (the so called stack-through connectors), parts with extra long pins are used; these have to placed some distance away from the circuit board, otherwise they will not fit into the corresponding headers of the DAC board. And, naturally, you have to make sure that the connector bodies are fitted nice and parallel to the circuit board. Mutatis mutandis this is also true for the expansion connector K1, which is also fitted on the bottom side of the circuit board.
The circuit boards are firmly attached together using male/female standoffs; the same applies to the display.

The microcontroller

In order to get the PGA2311 to do what it needs to do, we need a microcontroller that sends the appropriate data to the serial interface of the volume controller. Here we use an ATtiny861A — a 20-pin IC with just enough I/O for the serial interface, seven pushbuttons, the IR receiver (for the remote control) and an LED. The latter (LED1) is a through-hole version that, if desired (using a PCB header on the board and a couple of wires), can potentially be mounted somewhere else. All the ISP I/O lines are also in use (except Reset); when you are programming the microcontroller (via connector K6) you have to make sure that your fingers are away from the pushbuttons! For the IR receiver we used a read-made solution in the form of the TSOP34436 (IC3); the output of which is also available as a test point. The IR protocol is RC5, for which there is an extremely handy command in Bascom (which is what we used when writing the firmware):

```
Getrc5(Address, Command)
```

Two version of the firmware are available, depending on the version of Volumio that you are using. Both versions are part of the free download [3]. We will describe both versions below.

Firmware 160321-12

This version of the firmware was developed with Volumio 2 in mind [2]. It is provided with a plug-in for the GPIO buttons (**Figure 6**). Nearly all GPIOs can be linked to one for the following functions: play/pause, volume+/–, previous, next and shutdown. Because our DAC does not have its own volume control, four functions are used to operate Volumio; however for the benefit of other DACs that have a volume control we have also connected the functions volume+ and volume–. For these functions nearly any of the GPIO can be freely selected – provided that they are not already used for I^2S or the 3.5" touch-sensitive display.

The GPIOs are provided with pull-downs by default. Our Volumio version (2.041 from 12.12.2016) does not (yet?) support the 3.5" LCD. In **Table 1** we summarize the functions of all the pins of the RPi expansion connector for you.

We use the RC5 commands from **Table 2**.

For the GPIOs logic '1' (High) corresponds to 3.3 V. We therefore need to use an MC74VHC14DG (IC4, hex Schmitt-trigger inverter — note: from ON Semiconductor) as a level shifter. The inputs of this IC can withstand voltages up to 7 V — perfect for our purpose. The corresponding I/Os of the microcontroller have to be High in the inactive state. Consequently the two I/Os that are configured as the inputs for the volume control (PA0/PA1), the pullups have to be turned on; the outputs that are configured for the four functions (PA2/PA3/PB1/PB3) have to be High in the inactive state.

Figure 5. Here you can clearly see how the sandwich is put together.

High-end volume control on a small surface area

Table 1: Raspberry Pi Expansion Connector Pin Function				
K1	**RPi (v3)**	**160321-12**	**I²S**	**3,5" LCD**
1	3,3 V	3,3 V		3,3 V
2	5 V			5 V
3	GPIO2 (SDA1)			NC
4	5 V			5 V
5	GPIO3 (SCL1)			NC
6	GND	GND	GND	GND
7	GPIO4 (GPIO_GCLK)			NC
8	GPIO14 (TXD0)			NC
9	GND	GND	GND	GND
10	GPIO15 (RXD0)			NC
11	GPIO17 (GPIO_GEN0)			TP_IRQ
12	GPIO18 (GPIO_GEN1)		BCK	NC
13	GPIO27 (GPIO_GEN2)			NC
14	GND	GND	GND	GND
15	GPIO22 (GPIO_GEN3)	S1/Play/Pause		NC
16	GPIO23 (GPIO_GEN4)			NC
17	3,3 V			3,3 V
18	GPIO24 (GPIO_GEN5)			LCD_RS
19	GPIO10 (SPI_MOSI)			LCD_SI/TP_SCK
20	GND	GND	GND	GND
21	GPIO9 (SPI_MISO)			TP_SO
22	GPIO25 (GPIO_GEN6)			RST
23	GPIO11 (SPI_SCLK)			LCD_SCK/TP_SCK
24	GPIO8 (SPI_CE0_N)			LCD_CS
25	GND	GND	GND	GND
26	GPIO7 (SPI_CE1_N)			TP_CS
27	ID_SD			
28	ID_SC			
29	GPIO5			
30	GND	GND	GND	
31	GPIO6			
32	GPIO12	S6/Shutdown		
33	GPIO13	S5/Volume up		
34	GND	GND	GND	
35	GPIO19		LRCK	
36	GPIO16	S4/Volume down		
37	GPIO26	S2/Previous track		
38	GPIO20	S3/Next track		
39	GND	GND	GND	
40	GPIO21		DATA	

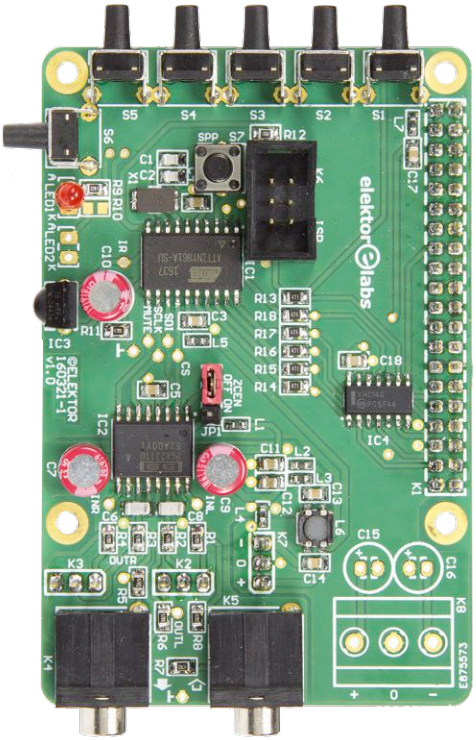
Table 2: RC5 commands	
Play/pause	RC5 command 1 (Digit entry)
Previous track	RC5 command 2 (Digit entry)
Next track	RC5 command 3 (Digit entry)
Shutdown	RC5 command 12 (Standby)
Mute/de-mute	RC5 command 13
Personal preference settings	RC5 command 14
Increase sound volume	RC5 command 16
Decrease sound volume	RC5 command 17
Shift sound balance to the right	RC5 command 26
Shift sound balance to the left	RC5 command 27

The six pushbuttons (S1 through S6) are connected via resistors of 470 Ω to the I/Os of the microcontroller. When one of these buttons is pushed the resistors prevent a short-circuit at the output of the µC, while at the same time the corresponding inverter is activated by making its input Low. The value of the resistors is low enough to make the inputs of the µC logic low.
When pushbutton S7 is pressed the volume level that is currently set will be stored as a personal preference. This button is not with the other six near the edge of the circuit board because it is likely it will be used only sporadically. LED1 flashes briefly four times when S7 is pushed, to indicate the present setting of the volume level is being written to EEPROM. After switching on this setting is also used as the initial value.
When a new microcontroller is used for the first time, a default volume level of -20 dB is set. This can be 'surprisingly' loud — you are warned!
LED1 indicates that an RC5 command has been received — but that does not necessarily have to be one of the commands actually supported by the volume control. The IR receiver (a Vishay type suitable for TC5) is connected to PB6 because the Getrc5 command uses TIMER0 and the TIMER0 interrupt. If you are going to develop your own software for another IR protocol, then do make sure you use the correct IR receiver. In any case, the connections for the receiver are fairly common.
While we were developing the software we noticed something strange: when the 'divide clock by 8' is turned off, the Getrc5 statement doesn't work very well. This means that the 'ckdev8' fuse has to be enabled when programming the fuses in the microcontroller (see **Figure 7**)! For this reason we have increased the crystal frequency for the microcontroller to 12 MHz (X1 in the schematic).

Firmware 160321-11

This version of the firmware was initially developed for use with Volumio 1.55 and our Raspberry Pi Audio DAC (ref Elektor July & August 2017). However, this version can also be used with Volumio 2 — or for entirely other applications. Note: when you use the volume control as a stand-alone device, you need a separate symmetrical 5-V power supply! In order to prevent ground loops we strongly suggest using an isolated power supply;

additionally L4 has to be replaced with a 0-Ω resistor (0603) or a wire link.
If K4 and K5 are fitted and the appropriate cables are used, the volume control can be inserted into any analog audio signal path (V_{max} amounts to 2 V). In this way you can provide any amplifier with a remote volume control.
It is also possible to use connectors K2 and K3, instead of the 3.5-mm audio connectors K4 and K5. In that case circuit board headers have to fitted on the board (or the connecting wires can be soldered directly to the circuit board).
Pushbuttons S1 through S6 are used only for changing the volume settings; these functions are, together with the RC5 commands you can find summarized in **Table 3**.



Just as with the other firmware version, here S7 is also used to store the current volume setting as a personal preference in EEPROM.

When the mute function (IC2 pin 8) is activated (via software or hardware), the internal preamplifiers in the PGA2311 are disconnected from the outputs and VOUTL and VOUTR are connected to ground via 10-kΩ resistors. This can increase the noise in a connected amplifier. Also when the mute function is activated an internal offset calibration is carried out.

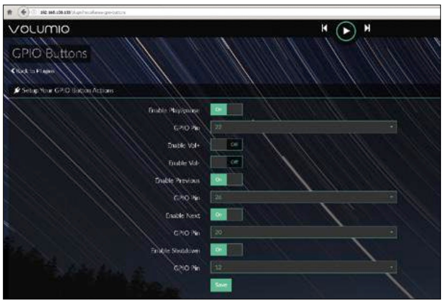


Figure 6. The Volumio GPIO plugin.

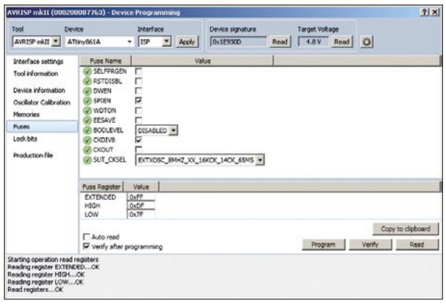


Figure 7. Must-know: the programming of the fuses.

Volumio Mk. 2
At the time of writing, the small 3.5” touch screen is not (yet) supported in Volumio 2. When the plugin for supporting the original Raspberry Pi display is installed, a standard monitor can be connected to the HDMI connected of the RPi. Another hint related to the installation of this plugin: do not connect a keyboard to the RPi. If you are using a dongle for

Table 3: Pushbuttons and RC5 commands (firmware 160321-11)		
S1	personal preference	RC5 command 14
S2	balance to the left	RC5 command 27
S3	balance to the right	RC5 command 26
S4	volume down	RC5 command 17
S5	volume up	RC5 command 16
S6	mute	RC5 command 13

Table 4: Selected measurement results	
Power supply current	
LED1 off	11 mA
LED1 on	12 mA
Maximum output voltage (1 kHz, THD = 0.1%)	
0 dB gain	2.3 V
20 dB gain	2.6 V
–20 dB gain, (3.9 V in)	0.39 V
THD+N	
1 kHz, B = 22 kHz	0.0003% (0 dB, 2 V out)
1 kHz, B = 80 kHz	0.00042% (0 dB, 2 V out)
20 kHz, B = 80 kHz	0.0022% (0 dB, 2 V out)
1 kHz, B = 22 kHz	0.0016% (20 dB, 2 V out)
1 kHz, B = 80 kHz	0.0025% (20 dB, 2 V out)
20 kHz, B = 80 kHz	0.0035% (20 dB, 2 V out)
IMD (50 Hz : 7 kHz = 4:1)	
0 dB gain, 2 V in	0.0007%
20 dB gain, 200 mV in	0.0038%
DIM (3.15 kHz square wave + 15 kHz sine)	
0 dB gain, 2 V in	0.0009%
20 dB gain, 200 mV in	0.0008%
Bandwidth (–3 dB)	
0 dB, output triangle	1.1 MHz
20 dB	690 kHz
Crosstalk	
1 kHz	< –100 dB
20 kHz	< –90 dB

the mouse/keyboard combo it is better to remove that too. It is recommended to start the installation from within the WebUI.

Measurements

Just as with the audio DAC for the RPi we have performed a few measurements on our volume control; the most important results are summarized in **Table 4**.

We have also made a few plots with our Audio Precision Analyzer, both for the audio DAC as well as the volume control. These are further explained in the sidebar.

(160321)

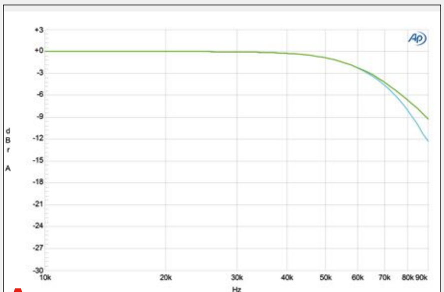
Web Links

[1] Datasheet PGA2311: www.ti.com/lit/ds/symlink/pga2311.pdf

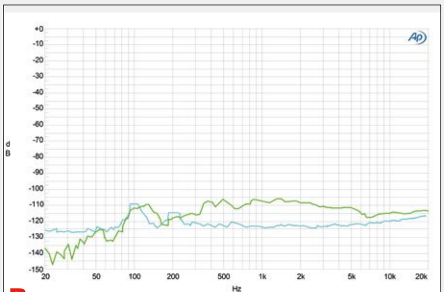
[2] Volumio: https://volumio.org

[3] Download: ww.elektormagazine.com/160321

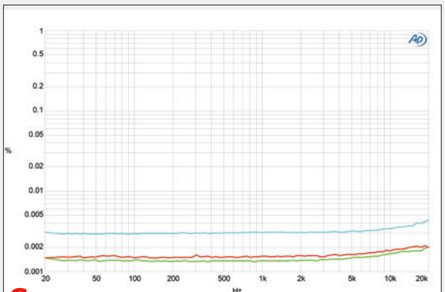
Measurements on the Big DAC



A



B



C

One of the showpieces of our former audio lab in Holland is a full blown Audio Precision Analyzer — which fortunately has survived the recent move to Aachen intact. Naturally we have tested the 'Big DAC' using our analyzer; below you can see some of the results.

Audio-DAC

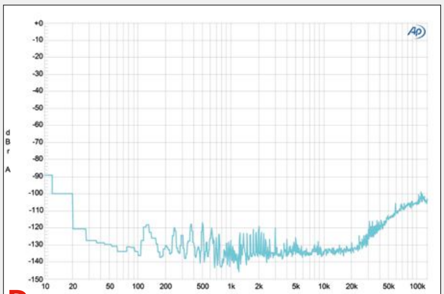
Plot A shows the amplitude as a function of the frequency with a stepped sine signal (70 frequencies, full scale) at a sampling frequency of 192 kHz. The top (green) curve was measured with a steep output filter; the bottom (blue) curve with is less steep filter. The corner frequency amounts to 63.5 kHz (64.5 kHz in theory). At 90 kHz the amplitude is -9.2 dB (green) and -12.3 dB (blue) respectively. At 45 kHz and 51 kHz both curves are equal (-0.5 dB and -1 dB respectively).

Plot B gives an impression of the channel separation as a function of frequency. The top (green) curve is the crosstalk from the left channel to the right channel. In this direction the crosstalk is slightly larger than the other way around. This is because the length of the signal traces on the circuit board to the output connectors are not the same. The trace for the right channel is longer, with the consequence of more crosstalk from left to right. Nevertheless, this is a good result for such a compact design.

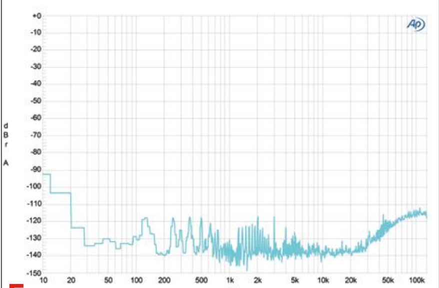
Plot C shows the total harmonic distortion plus noise (THD+N) as a function of frequency at sample frequencies of 48, 96 and 192 kHz. In all cases the bandwidth is 80 kHz. Incidentally, at a bandwidth of 22 kHz the THD+N is much lower when the sampling frequency is 48 kHz.

Plot D shows the FFT analysis of a 1 kHz full-scale sine at a sampling frequency of 32 kHz. Remarkable is the increase of the noise just above the audio bandwidth. This could be noise shaping. At a bandwidth of 22 kHz the THD+N amounts to 0.0007% (at a bandwidth of 80 kHz that is 0.012%). A large number of noise spikes are visible, mainly caused by the RPi, which is just below the DAC. The second and third harmonics of the sample frequency are just visible.

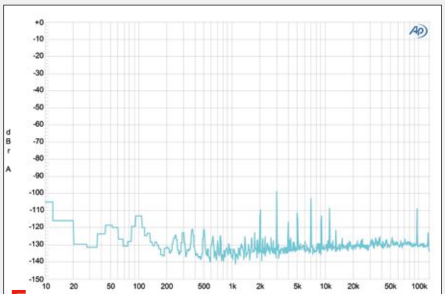
Plot E shows the same as Plot D, but then at a sampling frequency of 44.1 kHz. The high-frequency noise is here less when compared to the previous plot. The second and third harmonics of the sample frequency are a little bit more pronounced here; nevertheless the harmonic distor-



D



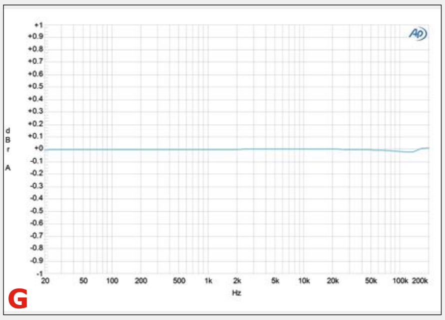
E

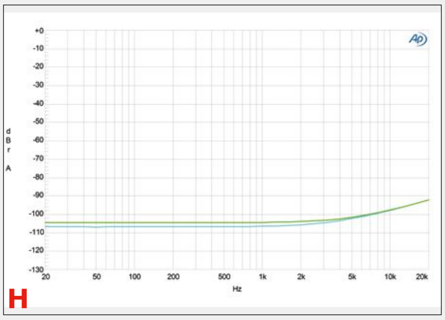


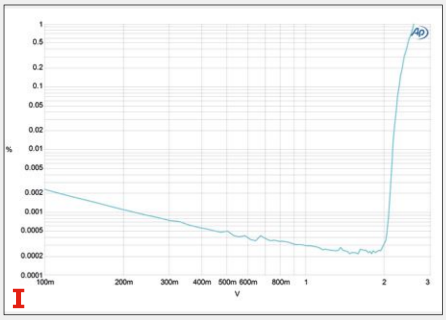
F

FROM THE STORE

- ➞160321-1
  Bare printed circuit board
- ➞160321-91
  Assembled module
  (headers supplied separately —
  DIY soldering)
- ➞160198-91
  Assembled Audio DAC module
  (headers supplied separately —
  DIY soldering)
- ➞SKU 17631
  Raspberry Pi 3 (model B)


G


H


I

tion (without noise) amounts here to only 0.00019%.

Plot F finally shows the FFT analysis of a 16-bit full-scale sine of 1 kHz at a sampling frequency of 32 kHz, 'on the fly' converted to 24-bit/96-kHz (as is possible in Volumio 1.55). Remarkable is the absence of the high-frequency noise that is visible in Plot D.

Volume control

Plot G again shows the amplitude as a function of frequency at a gain of 0 dB and an output level of 2 V. The maximum frequency of the generator in the Audio Precision Analyzer goes up to 200 kHz. The small 'bump' around 100 kHz is not significant (note the vertical scale!).

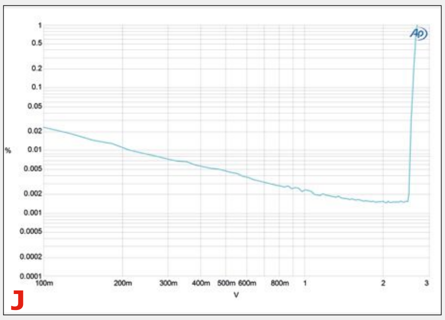
Plot H shows the crosstalk as a function of frequency. The difference between the two curves amounts to about 2 dB — the circuit therefore behaves almost completely symmetrically.

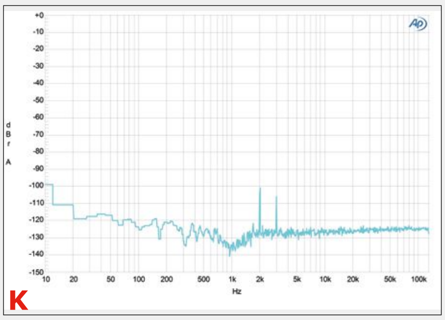
Plot I makes the THD+N visible as a function of output level at a gain of 0 dB. The bandwidth is here reduced to 22 kHz to better show where the circuit begins to clip. At this low power supply voltage the maximum undistorted output voltage is about 1.9 V. Above this level the THD increases quickly (0.1% at 2.3 V and 1% at 2.6 V).

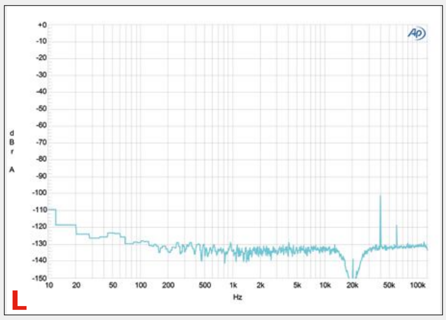
Plot J is the same is Plot I, but now at a gain of 20 dB. The maximum undistorted output voltage is now surprisingly much higher, at 2.48 V (THD = 0.0015%) the circuit begins to clip. Apparently at lower gain this occurs earlier in the input stage because the input voltage is then higher.

Plot K shows the FFT of a 1-kHz signal at a gain of 20 dB and an output voltage of 2 V. Here only the second and third harmonics are visible, corresponding to a THD (without noise) of 0.001%. (THD+N at 22 kHz bandwidth is 0.0016%, and 0.0025% at 80 kHz bandwidth).

Plot L finally is the same as Plot K, but now for a 20-kHz signal. Here too the second and third harmonics are visible. THD = 0.00086%; THD+N at 80 kHz bandwidth is 0.0035%.


J


K


L