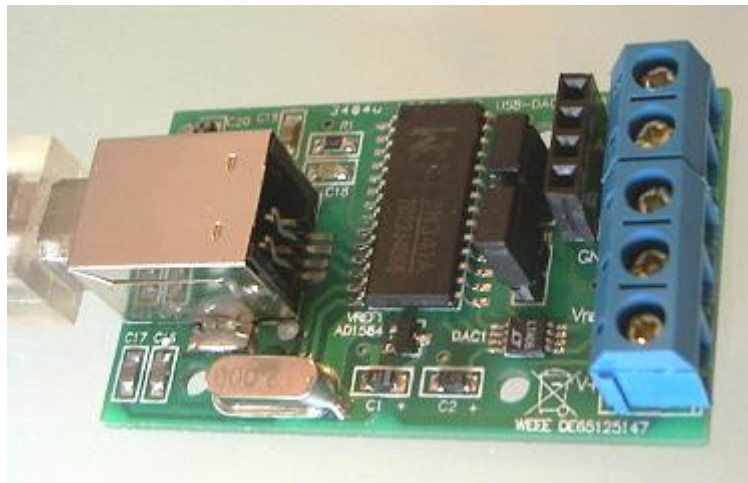


USB DAC



Features

- Compact USB DAC board for measurement and control applications
- Useful as ramp generator, LFO, etc.
- Two independent voltage outputs DAC A and DAC B
- Output voltage 0V ... 4.095V, software adjustable
- 50 Ohm outputs with short protection
- Resolution 12 bits (1mV per step)
- Approx. 400 samples/sec.
- Voltage reference internal or external
- Dimensions approx.. 30 x 50 x 20 mm
- Power supply (+5V) either USB or external
- Convenient screw-terminal connector
- USB chip set: CH341A
- Integrated I²C/TWI master interface
- System requirements XP, 2000, Vista 32
- Suitable for ProfiLab 4.0.software
- Easy programming Interface with ActiveX control (OCX)
- Programming examples in C++, Delphi und Visual Basic.

Download link for manual, driver and examples

http://www.abacom-online.de/div/setup_usb_dac.exe

Installation

- Before connecting the device to the USB download and install setup software please.
- After that connect the device to the USB and follow the Windows Plug & Play instructions.
- The device is now ready for use. Further information on how to use the board can be found in the manual installed with the software.

Screw terminals

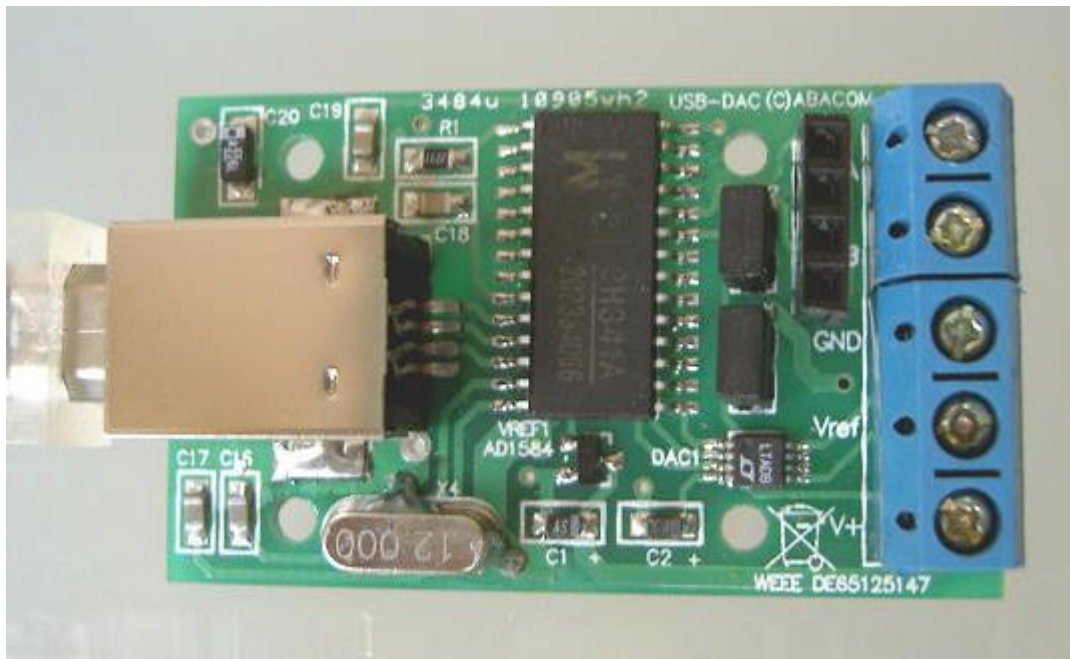
The module is equipped with five screw terminal contacts, which make it easy to connect peripheral circuits.

- A: Voltage output DAC A
- B: Voltage output DAC B
- GND: Ground (connected to PC ground)
- Vref: reference voltage
- V+: USB supply voltage (5V)

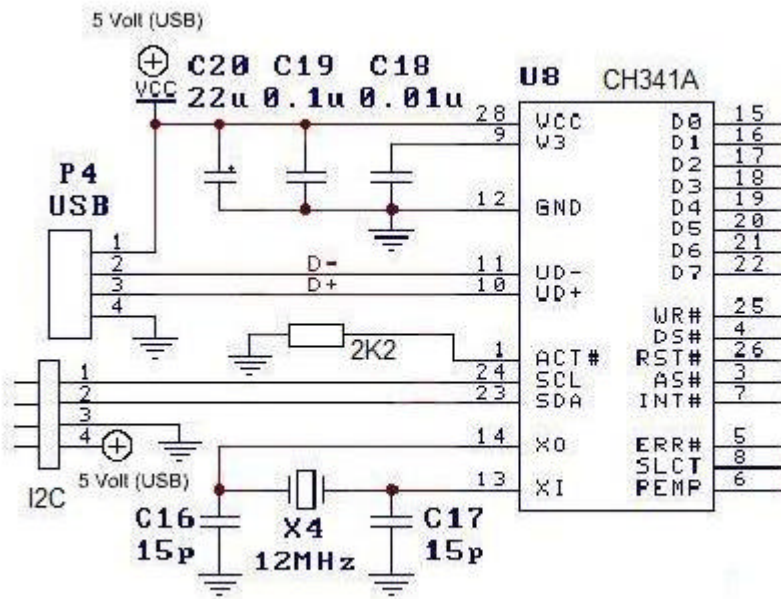
Screw terminal A and B are the DAC voltage output channels. Voltage at these outputs is controlled by software and ranges from 0V to 4.095V (Vref). Output voltages refer to PC ground (GND).

The internal reference voltage is available at terminal Vref, as long as jumper J1 is present. To feed in an alternative reference voltage through this terminal, jumper 1 must be removed.

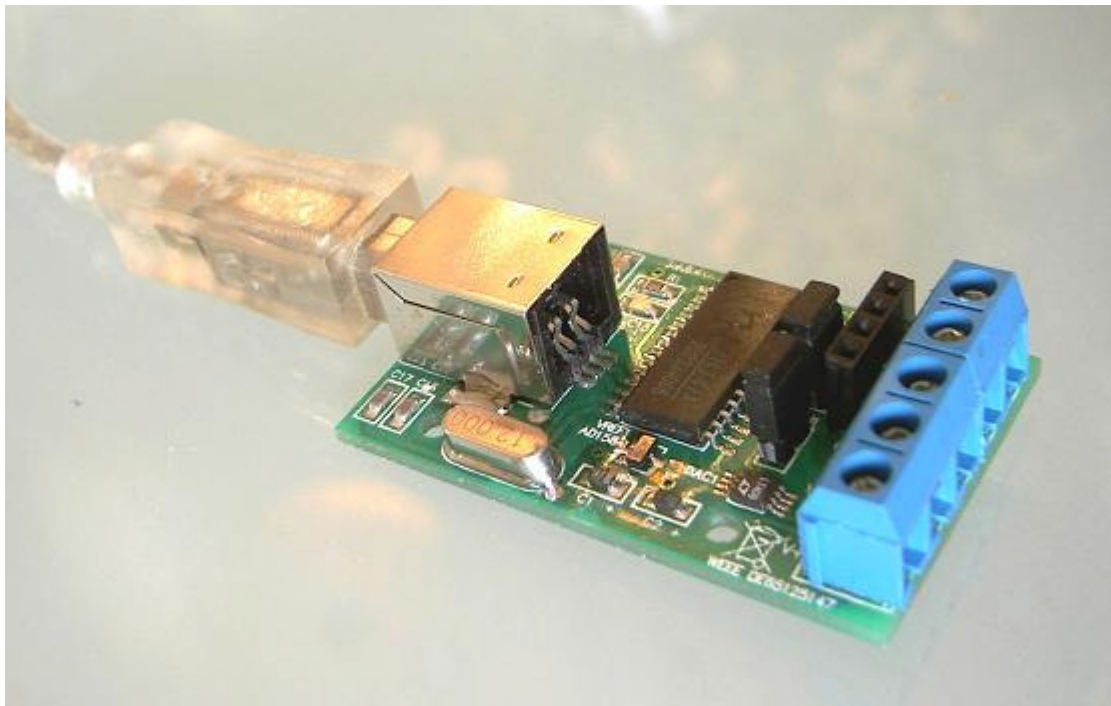
The USB supply voltage (+5V) is available at terminal V+, as long as jumper J2 is present. (Connecting your own circuits please remember that maximum current for USB ports is 500mA). If necessary the DAC can be supplied by a stabilized external 5V power supply through this terminal. In that case jumper J2 must be removed first.

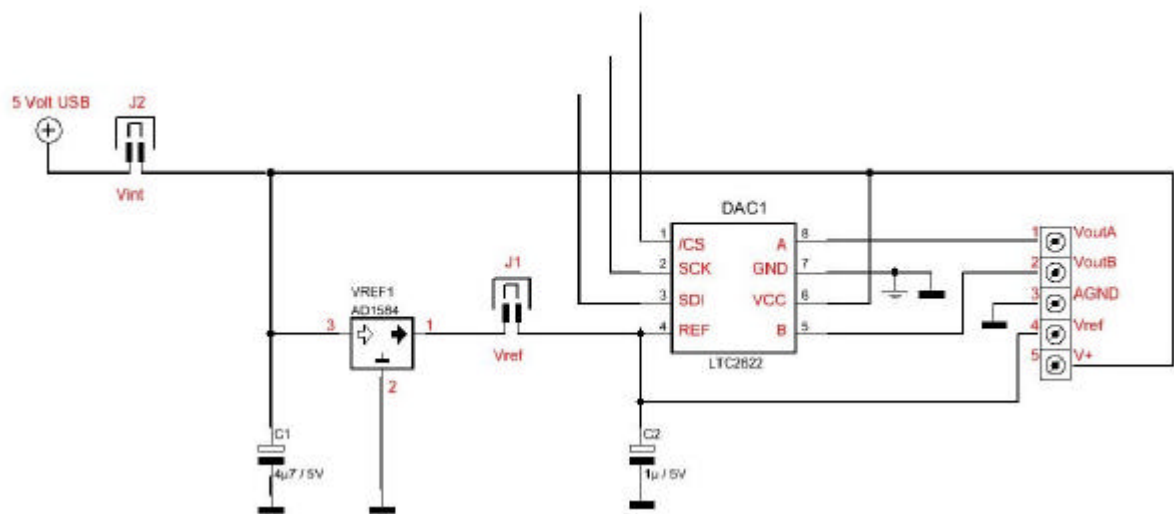


USB interface



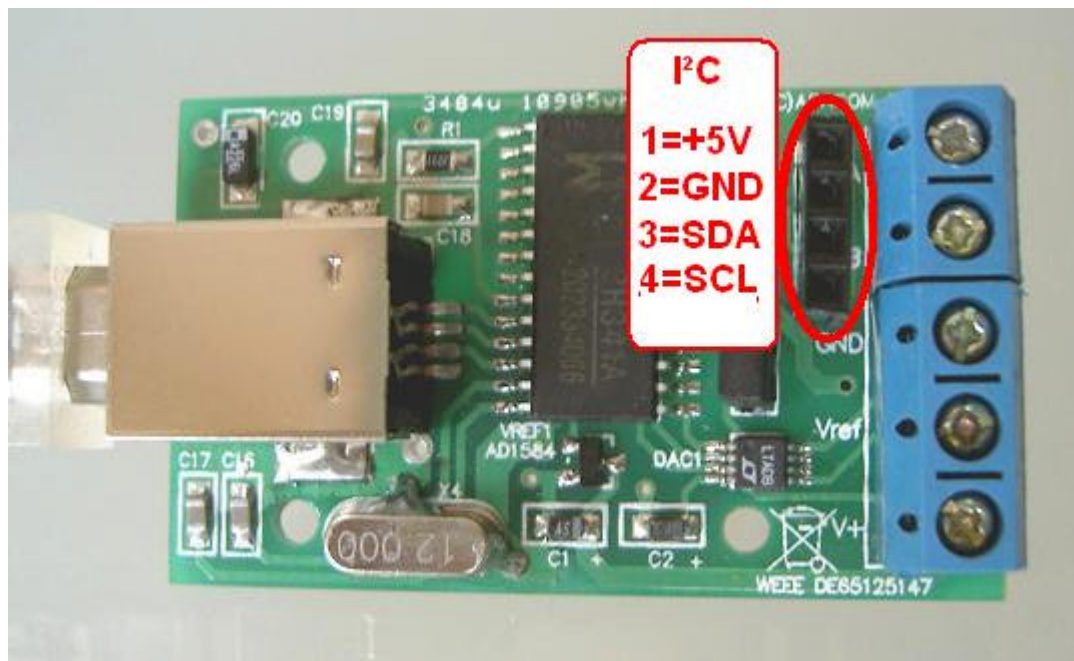
The DAC is internally controlled through a SPI (Serial Peripheral Interface). A CH341A chip is used as USB-to-SPI bridge. This chip also provides the I²C (TWI) master interface.





I²C master connector

The USB power (Vcc, GND) and the I²C signals (SCA and SCL) are available at a female pin connector. Make sure that the overall current from USB does not exceed 500mA. The I²C pull-up resistors (usually 2K2) must be provided by the user circuitry.



Software

The software setup copies the following files to your hard-disc:

API

This directory contains files and examples made for some programming languages.

DOC

Contains some useful datasheets.

DRIVER:

This folder contains files necessary for Windows plug & play installation. Connecting the device to the USB these driver files for CH341A chipset installed. The device is then ready for use. In software applications the device is addressed with a device number. This is a consecutive number of available CH341A chips starting from zero. Applications open a data channel to the CH341A chip that will then execute the data transport to the DAC. In most applications the device no. is adjustable.

TEST

A simple test program can be found in this folder.

Functional test

For a simple device test and to find out the device no. use the USB_DAC_TEST.EXE from the \TEST folder.

ProfiLab Expert

This folder contains examples for ProfiLab Expert. Please make sure to have latest update from our Internet site installed.

Programming interface (API)

Programming examples for some languages can be found in the \API directory of the software installation.

The DAC and the I²C interface is programmed using a ActiveX control (file ABACOM_USB_DAC.OCX). The ActiveX control is registered as "ABACOM USB DAC" during software installation process and can be imported with any programming language that support ActiveX controls. Please refer to the manual of your programming language how ActiveX controls are handled.

The following code snippets are meant as illustration for the functional principles and may fail to compile. Remember the code examples.

To program the board an object "ABACOM USB DAC" (USB DACX) must be created. This could be done dynamically at run-time or with the IDE's form designer.

```
TUSB DACX *DAC ;

DAC = new TUSB DACX(this);
DAC->Parent = this;
DAC->Left = 24;
DAC->Top = 40;
```

The DAC object defines some properties and methods for general control administration (left, top, etc.) and specialized ones for DAC and I²C control., like

```
DAC->DAC1_STATUS = ...
```

or

```
DAC->I2C_Write(...)
```

Methods and Properties are named with leading "DAC_" and "I2C_".corresponding to their purpose.

DAC programming

Both DAC channels have a corresponding property DAC_A and DAC_B which control the output voltages:

```
DAC->DAC_A = 1024; // DAC A = 1.024 V
DAC->DAC_B = 4095; // DAC B = 4.095 V
```

For a synchronized write to both DAC channels the DAC_SYNC property can be set. In that case DAC A will not change its output value, before DAC B is written:

```
DAC_SYNC = true; // Activate SYNC
DAC->DAC_A = 1000; // Prepare DAC A
DAC->DAC_B = 2000; // Update both channels
```

In the need to use more than one board, additional DAC objects must be created. To address a device the device number must be set:

```
DAC1->DAC_DeviceNo = 0; //1st device
DAC2->DAC_DeviceNo = 1; //second device
etc.

DACX->DAC_DeviceNo = -1 //Device disabled (default)
```

I²C (TWI) programming

The device is equipped with an integrated I²C(TWI) master interface, which makes it possible to connect additional peripheral chips and control them with PC software. Some basic knowledge in I²C technology and understanding of chips datasheets is necessary for that.

The programming examples are based on a ready-made I²C demonstration board from MicroChip (TM), which can be purchased separately from our web-shop.
(http://www.electronic-software-shop.com/product_info.php?pName=microchip-ic-board-p-17&cName=hardware-ic-twi-c-2_13)

Instead of that, any other I²C circuitry could be connected and programmed.

The following I²C methods are usable for that purpose:

Read/write bytes from/to chip registers

The method

```
DAC->I2C_Write(0x20,0x09,LEDStatus)
```

Writes a byte (here LEDStatus) to the chip (slave) address 0x20 into a chip register (here 0x09).

```
DAC->I2C_Read(0x20,0x09,InData)
```

Read a register (here 0x09) of a chip, with chip address 0x20. The return value InData contains a readable ASCII string (HEX dump; e.g. "A7")

Direct read/write data over i²C bus

The following stream function allows direct read and/or write over the I²C bus:

```
OutData = "92 00";  
DAC->I2C_Stream(outData, 2, inData);
```

The parameters (inData; outData) are readable ASCII strings (Hex dumps). The example writes two bytes 0x92 and 0x00 out on the bus. The number of bytes to write is implied with the the length OutData hex dump.

The number of bytes to read must be given in the second parameter (here: two). The return value is a readable hex dump string, like "FE B6" of the read bytes.

This function writes/reads bytes – without certain protocol – over the I²C bus and is therefore usable with almost any peripheral chip type.

EEPROM

Read/write to EEPROM can be done with

```
DAC->I2C_WriteEEPROM(eprom24C02, 0, "FF FE");  
DAC->I2C_ReadEEPROM(eprom24C02, 0, 7, Data);
```

For write operation the EEPROM type, the memory (start) address and the write data must be fed. For read operations the number of bytes to read must be given in addition (here: seven). Read and write data are readable hex dump strings again.

Clock frequency

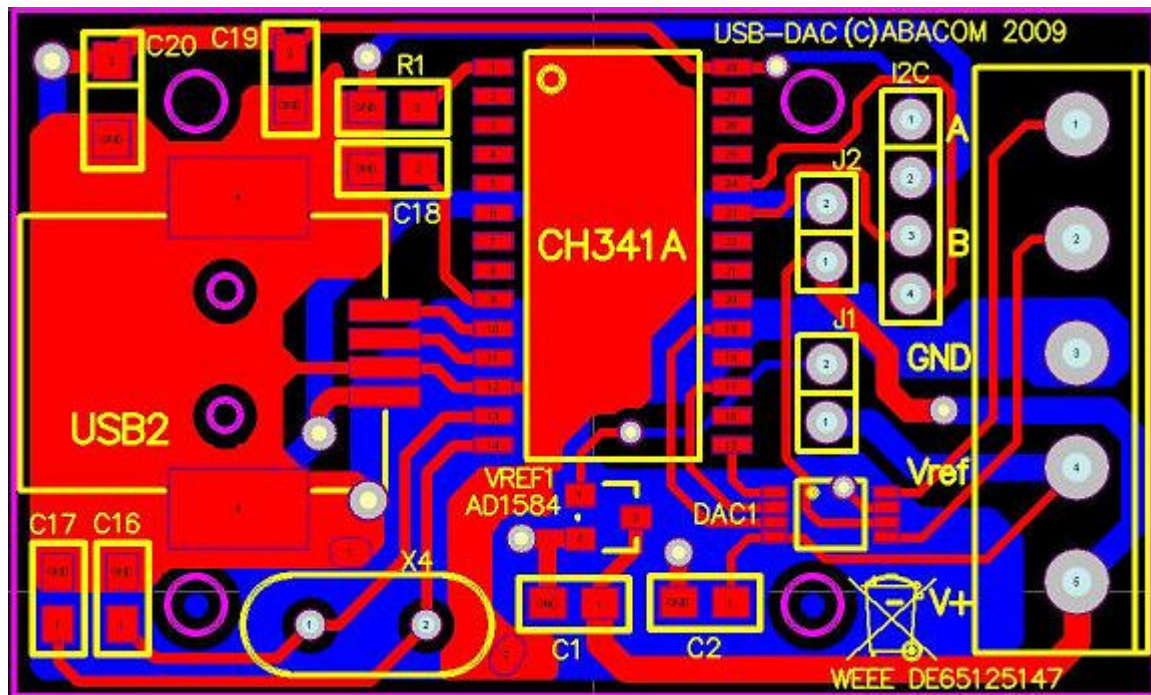
The I²C speed can be adjusted in four steps:

```
DAC->I2C_Speed = slow_20kHz  
DAC->I2C_Speed = normal_100kHz  
DAC->I2C_Speed = fast_400kHz  
DAC->I2C_Speed = high_750kHz
```

Fast and high clock rates may not be supported by all I²C chips.

Dimensions

30 x 50 x 20mm



© 2009 – ABACOM-Ingenieurgesellschaft