

USB-DAC



Features

- Kompakter USB-DAC (LTC2622) für die Mess- und Steuerungstechnik
- Einsetzbar z.B. als Rampengenerator, LFO, etc.
- Zwei unabhängige Spannungsausgänge DAC A und DAC B
- Ausgangsspannung: 0V...+4,095V
- Auflösung 12 Bit (entsprechend 1 mV pro Schritt)
- ca. 400 Samples/Sek.
- Referenzspannung intern (4,095V) oder extern
- Abmessungen ca. 30 x 50 x 20 mm
- Betriebsspannung (+5V) wahlweise vom USB oder extern
- komfortabler Anschluss über Schraubklemmen
- USB-Chipsatz: CH341A
- Integrierte I²C/TWI-Master-Schnittstelle
- Systemvoraussetzungen: XP, 2000, Vista, WIN 7, 32/64 Bit
- Einsetzbar mit ProfiLab 4.0.
- Einfache Programmierung per ActiveX-Steuerelement (OCX)
- Programmbeispiele in C++, Delphi und Visual Basic.

Download-Link für Anleitung, Treiber, Beispiele

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

Installation

- Bevor Sie die Karte mit dem USB verbinden, laden Sie bitte unbedingt die zugehörige Software herunter und führen Sie das Installationsprogramm aus.
- Danach können Sie die Karte mit dem USB verbinden und die gewohnte Windows-Plug&Play-Installation durchführen.
- Nun ist die Karte einsatzbereit. Weitere Informationen zur Verwendung der Karte finden Sie jetzt in der Anleitung und in den weiteren Dateien, die das Installationsprogramm installiert hat.

Anschlussklemmen

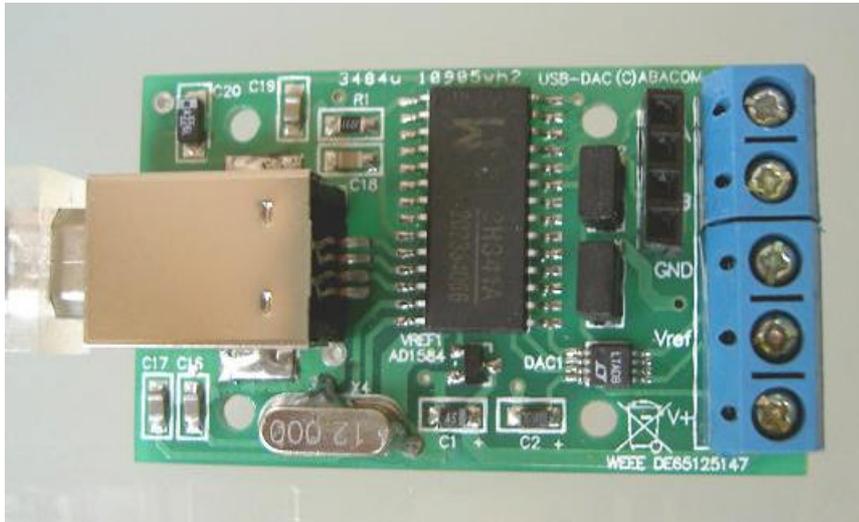
Das Modul stellt fünf Anschlussklemmen für den bequemen Anschluss Ihrer Peripherieschaltung bereit:

- A: Spannungsausgang DAC A
- B: Spannungsausgang DAC B
- GND: Masse (verbunden mit PC-Masse)
- Vref: Referenzspannung
- V+: USB-Versorgungsspannung (5V)

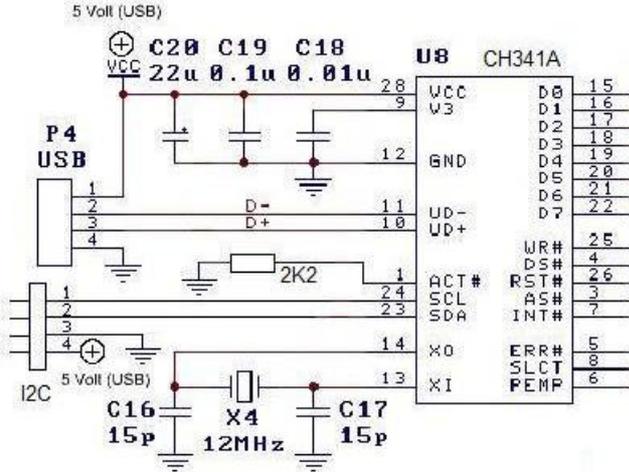
Die Klemmen A und B stellen die Spannungsausgänge des DAC. Die Ausgangsspannung dieser Ausgänge kann per Software zwischen 0 Volt und der Referenzspannung (Vref) eingestellt werden. Die Ausgangsspannung ist auf die PC-Masse bezogen (GND)

Der Anschluss Vref liefert die interne Referenzspannung (4,095V), solange die Steckbrücke (Jumper J1) aufgesteckt ist. Soll eine andere Referenzspannung verwendet werden, so muss die Steckbrücke J1 zu nächst entfernt werden. Danach kann eine eigene Referenzspannung über die Klemme Vref eingespeist werden.

Der Anschluss V+ liefert die USB-Spannungsversorgung (+5V), solange die Steckbrücke (Jumper J2) aufgesteckt ist. (Bitte beachten Sie beim Anschluss eigener Schaltungen den max. zulässigen Strom von 500 mA des USB-Port!). Bei Bedarf kann der DAC durch eine eigene stabilisierte Versorgungsspannung (+5V) über die Klemme V+ versorgt werden. In diesem Fall muss unbedingt Jumper J2 entfernt werden.



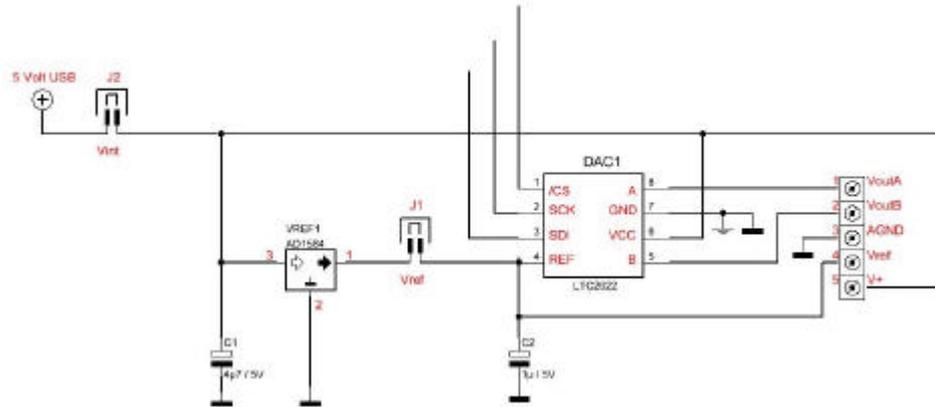
USB-Interface



Die Ansteuerung des DAC erfolgt intern über eine serielle Datenverbindung (SPI)
 Die Umsetzung des USB auf SPI übernimmt ein CH341A-Interface-Chip. Darüber hinaus stellt der CH341A die I2C-Master-Schnittstelle bereit.



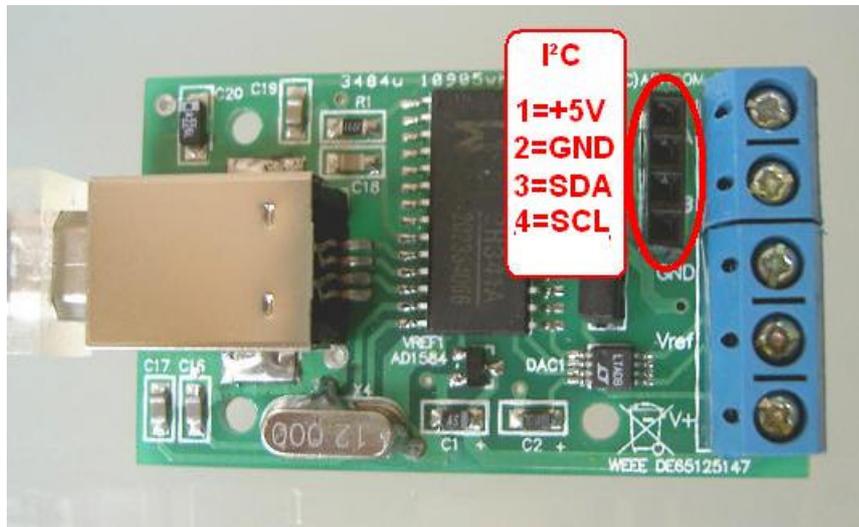
DAC und Referenzspannung



Der Analogteil des Moduls besteht aus einem hochwertigen Digital-Analog-Konverter (Linear Technologies LTC2622) und einer 4,095V Spannungsreferenz (Analog Devices AD1584).

I²C-Master-Anschluss

Eine vierpolige Buchsenleiste stellt die I²C-Signale und die USB-Versorgungsspannung für eigene Schaltungserweiterungen bereit. Dabei darf die Gesamtstromentnahme aus dem USB-Port 500mA nicht übersteigen. Die I²C-Pullup-Widerstände (meist 2K2) müssen in der Anwenderschaltung vorgesehen werden.



Software

Laden Sie die Software von unseren Internetseiten herunter. Das Setup-Programm kopiert folgende Ordner auf Ihre Festplatte:

API

Dieser Ordner enthält Dateien und Beispiele für die eigene Programmierung der Karte mit verschiedenen Programmiersprachen.

DOC

Enthält einige Datenblätter und evtl. nützliche Zusatzinformationen.

DRIVER:

Dieser Ordner enthält die USB-Treiber die Sie für die Windows Plug&Play-Installation benötigen.

Nach dem Anstecken am USB muss die Treiberinstallation für den CH341A-Interfacechip erfolgen. Den Treiber finden Sie im Ordner DRIVER. Danach ist das Gerät einsetzbar. Die Adressierung der Karte erfolgt über eine fortlaufende Gerätenummer des CH341A-Interface-Chips. Alle am System angeschlossenen CH341A-Chips erhalten automatisch eine fortlaufende Nummer beginnend mit 0.

Programme die Daten an das Gerät senden wollen, öffnen zunächst anhand der Gerätenummer einen Datenkanal zum CH341A-Chip, über den dann die Datenübertragung zum DAC erfolgt. In der Anwendungssoftware muss in der Regel nur die richtige Gerätenummer eingestellt werden, um den DAC anzusprechen.

TEST

enthält ein einfaches Testprogramm für einen ersten Funktionstest.

Funktionstest

Für einen ersten Funktionstest und zum Ermitteln der richtigen Gerätenummer finden Sie ein Testprogramm im Ordner \TEST.

ProfiLab Expert

In diesem Ordner finden Sie Beispiele für ProfiLab Expert. Bitte verwenden Sie das neueste Internet-Update für die ProfiLab-Software.

Programmierschnittstelle

Programmierbeispiele für verschiedene Programmiersprachen finden Sie im Ordner VAPI der Softwareinstallation.

Die Programmierung des DAC und der I²C-Schnittstelle erfolgt mit Hilfe eines ActiveX-Steurelements. Dieses befindet sich in der Datei ABACOM_USB_DAC.OCX. Das Steuerelement wird bei der Installation der Software automatisch unter dem Namen "ABACOM USB DAC" registriert und kann mit allen Programmiersprachen verwendet werden, die den Import von ActiveX-Steurelementen unterstützen. Grundlegende Informationen zum Import von ActiveX-Steurelementen entnehmen Sie bitte der Anleitung zu Ihrer Programmiersprache. Dies nachstehenden Programmauszüge stellen keinen lauffähigen Programmcode dar. Sie dienen nur zur Erläuterung des Funktionsprinzips. Bitte beachten Sie auch die mitgelieferten Programmbeispiele.

Für die Programmierung erzeugt man zunächst ein "ABACOM USB DAC" – Objekt vom Typ USBDACX. Dies kann entweder dynamisch zur Laufzeit oder mit Hilfe des Formular-Designers der Programmiersprache geschehen.

```
TUSBDACX *DAC ;  
  
DAC = new TUSBDACX(this);  
DAC->Parent = this;  
DAC->Left = 24;  
DAC->Top = 40;
```

Neben den gewohnten Eigenschaften zur Verwaltung des Steurelements, wie .z.B. LEFT, TOP, etc. verfügt das Steuerelement über Methoden und Eigenschaften, die speziell der Programmierung des DAC und der I²C-Schnittstelle dienen, wie z.B.

```
DAC->DAC_A = ...
```

oder

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

Diese Namen dieser Eigenschaften und Methoden beginnen entsprechend Ihrer Verwendung mit "DAC_" bzw. "I2C_".

Programmierung des DAC

Den beiden Kanälen des DAC sind Eigenschaften (Properties) DAC_A und DAC_B zugeordnet, die es erlauben die Ausgangsspannung der Kanäle zu setzen:

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

Ein gleichzeitiges (synchrones) Einstellen beider Kanäle ist durch die Eigenschaft DAC_Sync möglich. Ist diese Eigenschaft aktiviert, so ändert sich der Spannungsausgang von DAC A erst dann, wenn auch DAC B neu eingestellt wird.

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

Für die Verwendung mehrerer Karten legt man ggf. mehrere Objekte an und setzt die Gerätenummer DAC_DeviceNo entsprechend:

```
DAC1->DAC_DeviceNo = 0; //erste Karte
DAC2->DAC_DeviceNo = 1; //zweite Karte
usw.
DACX->DAC_DeviceNo = -1 // Disabled device
```

Programmierung der I²C/TWI-Schnittstelle

Das Karte besitzt eine I²C-Master-Schnittstelle, die es erlaubt zusätzliche I²C-Peripherie-Chips am PC zu betreiben. Dazu sind Kenntnisse der I²C-Technik und das Verständnis der Datenblätter der verwendeten Chips unerlässlich.

Die Programmbeispiele beziehen sich auf eine I²C-Demonstrationsschaltung von MicroChip(TM), die anschlussfertig als Zubehör bei uns erworben werden kann.
http://www.electronic-software-shop.com/product_info.php?pName=microchip-ic-board-p-17&cName=hardware-ic-twi-c-2_13

Ebenso lassen sich - bei entsprechender Programmierung - natürlich auch andere I²C-Schaltungen an der Schnittstelle betreiben.

Für die Kommunikation mit I²C-Chips stehen folgende Funktionen zur Verfügung:

Byteweises Schreiben und Lesen Chip-Registern

Die Methode

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

schreibt ein Byte (hier LEDStatus) an die Chip-(Slave)-Adresse 0x20 in ein Chip-Register (hier 0x09).

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

Liest ein Register (hier 0x09) eines I²C-Chips, dessen Chip-Adresse 0x20 ist. Der Rückgabewert InData enthält den gelesenen Byte-Wert als lesbare Zeichenkette (HEX-Dump; z.B. "A7")

Direktes Schreiben und/oder Lesen von Busdaten

Ein direktes Schreiben und Lesen von Datenbytes über den I²C-Bus ist mit der Funktion STREAM-Funktion möglich:

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

Die Datenübergabe (inData; outData) erfolgt als lesbare Zeichenkette (Hex-Dump). Das Beispiel schreibt zwei Bytes 0x92 und 0x00 auf den Bus. Die Anzahl der zu schreibenden Bytes wird aus der Länge des Hex-Dumps (OutData) implizit ermittelt.

Die Anzahl der zu lesenden Bytes muss explizit angegeben werden (hier zwei). Der Rückgabewert inData liefert wiederum eine lesbare Zeichenkette der eingelesenen Bytes, wie z.B. "FE B6".

Die Funktion schreibt/liest die Daten wie gesagt – ohne jede eigene Logik – direkt über den I²C-Bus und ist damit sehr universell einsetzbar.

EEPROM

Das Schreiben und Lesen eines Standard-EEPROMS wird durch die folgende Funktionen ermöglicht:

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

Der Funktion wird zum Schreiben der EEPROM-Typ, die Speicher-(Start-)Adresse und die Daten als lesbarer Hex-Dump übergeben. Zum Lesen muss zusätzlich die Anzahl der zu lesenden Bytes (hier: sieben) übergeben werden.

Taktfrequenz

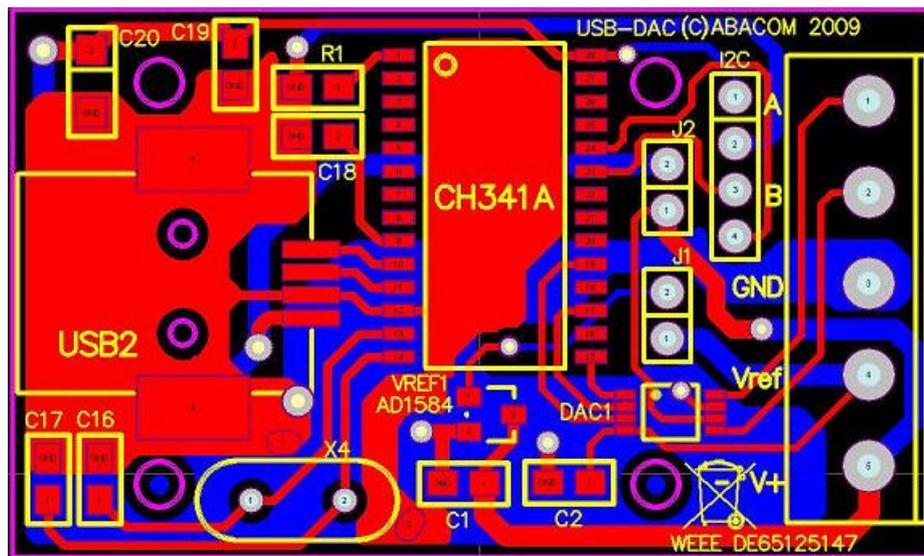
Die I²C-Übertragungsgeschwindigkeit, d.h. die Taktfrequenz ist in vier Stufen einstellbar:

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

Hohe Taktraten sind nicht unbedingt für alle I²C-Chips geeignet.

Abmessungen

30 x 50 x 20mm



© 2009 – ABACOM-Ingenieurgesellschaft