



techFX microDAC

Copyright 2009 The Silicon Horizon Inc.

Visit our webpage for tech support and forums

<http://www.TheSiliconHorizon.com/>

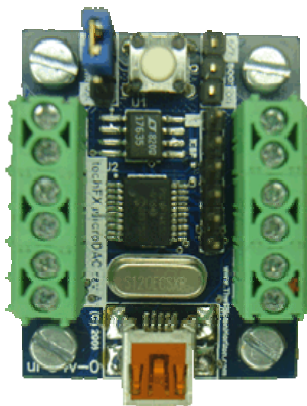


Table of Contents

Warning and user agreement of liability	3
System requirements:.....	4
Operating system.....	4
Cables and power supply	5
TechFX microDAC controller overview	6
Hardware features:	6
IO ports and pins diagram.....	7
IO pins and ports descriptions	8
Vsource select jumper	8
ICSP programming port.....	8
TX, RX pins.....	8
RC0,1,2,5 pins.....	8
DAC1,2 pins	8
Description of firmware options.....	9
HID firmware for windows.....	9
LIBusb firmware for linux.....	9
Programming the right firmware onto your device.....	10
Using techFX microDAC in Windows XP and Vista.....	11
Using the techFX microDAC in Linux.....	12
Device Characteristics	14

Warning and user agreement of liability

This product, techFX microDAC controller can cause damage when wired incorrectly, or not used in its intended ranges / uses. By using this device, you agree to not hold us accountable for an harm that may result to yourself through bodily injury, shock, and distress. You also agree not to hold us responsible for damages to equipment, computers and other such electronics due to the miswiring or misuse of this device. You assume all liability and responsibility for the use of this device.

System requirements:

Operating system

Windows

You must have NET 2.0 framework distribution installed to run this software. You can do a web search for "dotnetfx.exe" or use the following link:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en>

You should download and install the latest version of Microsoft Visual Studio Express edition 2005 or 2008.

Linux

Ubuntu 9.04 distribution is recommended as that is the testing platform.

The recommended LIB's are:

G++ (Gcc)
LIBusb
QT3
QT4
QT3-developer
QT3toQT4
QT4-developer

Sufficient permission is necessary to connect to USB devices. UDEV should be modified such that the USB device can be accessed accordingly when programs are run. If not possible, then a temporary elevation of permissions through the use of "sudo sh" can be used.

Cables and power supply

You will need a free USB 2.0 port, mini-USB cable.

If using USB power, then the accuracy of the DAC and ADC may be off by as much as 8-10 mV (this is due to the USB voltage being above 5 volts sometimes by as much as 30 mV).

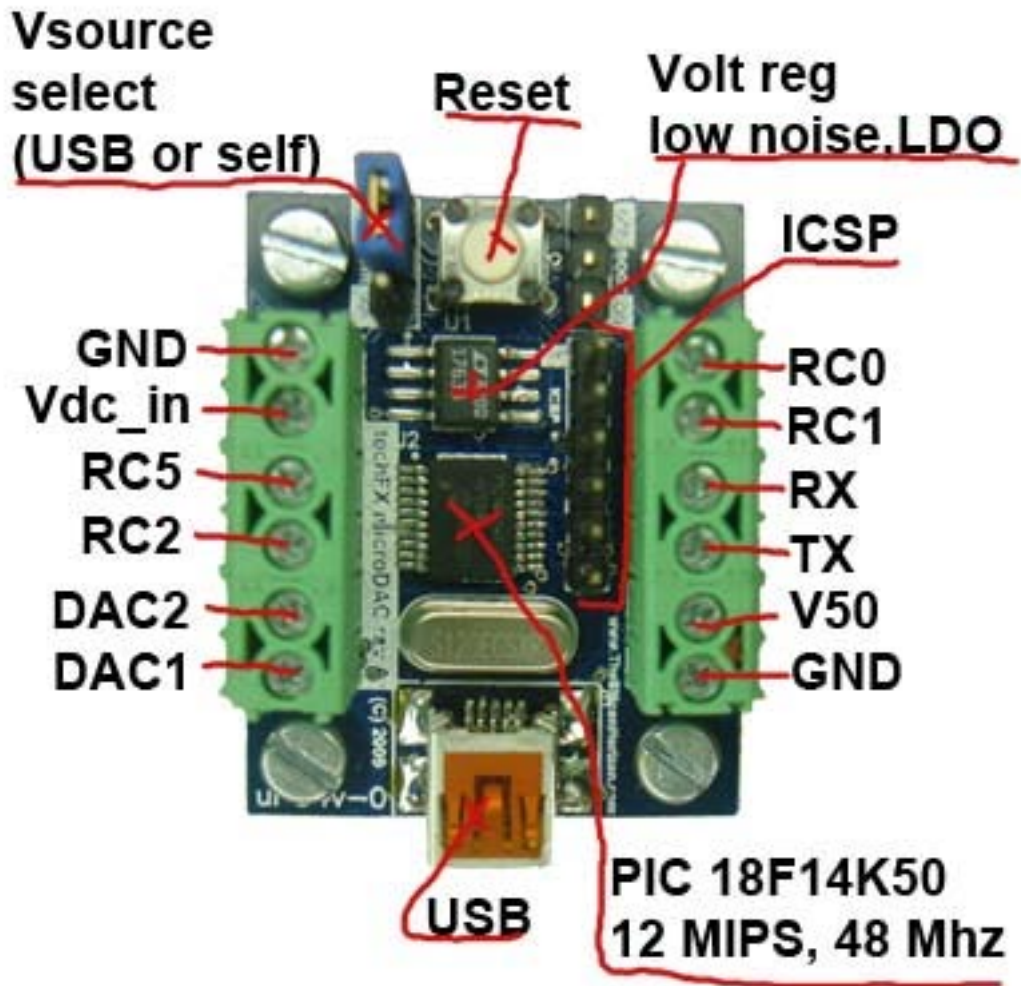
If using the DC power input pin "Vdc_in" then it is suggested a good switching wall adapter from 6-9 Vdc at least 500 mA rating.

TechFX microDAC controller overview

Hardware features:

- 12 bit DAC 2 channel
- 10 bit ADC on configurable IO
- Ultra low power / USB powered / Self powered options
- Low noise / LDO LT1763 voltage regulator 5 Vdc
- 48 Mhz 12 MIPS PIC18F14K50 MCU
- 256K SPI eeprom
- IO configurable as TTL / PWM / Analog in / INT pin
- Serial UART
- USB port
- Small 1.5" x 1.2" size on quality 4 layer pcb.

IO ports and pins diagram



IO pins and ports descriptions

Vsource select jumper

This jumper selects between USB power and Self power through the Vdc_in pin.

If using USB power, move the jumper to the outside of the board to select USB power.

If using Self power, move the jumper to the inside of the board to select Self power. Then connect the wall adapter or battery + to the Vdc_in pin, and the ground to the GND pin.

DO NOT CONNECT OR MOVE JUMPER AND EXTERNAL POWER WHEN THE DEVICE IS POWERED

ICSP programming port

The ICSP programming port allows you to program the device using a PICKIT 2, PICKIT 3, ICD2 or ICD3 programmer. You may find them on www.microchip.com and on ebay as well.

TX, RX pins

The TX,RX pins may be configured as other types of IO (see 18f14k50 datasheet for info on setting it up). However it is primarily intended for the serial UART port.

The TX terminal goes to the TX pin of the MCU, and the RX terminal goes to the RX pin of the MCU. They are not reversed. Therefore, if you are hooking up an LCD, you would hook up the RX of the LCD to the TX terminal of the microDAC and so on (reversing them).

RC0,1,2,5 pins

The RC pins are configurable as many types of IO including TTL input, TTL output, Analog input, some are PWM, and 1 of them is an Interrupt pin which may receive an external interrupt signal. Please consult with the 18f14k50 datasheet on how to setup the registers to use the configuration you wish to use.

DAC1,2 pins

The DAC pins are channel A, channel B from the MCP4822 DAC chip. The output can go from 0 to 4.096 volts and is more accurate when on self power than when using USB since USB may be a little over 5 volts.

Description of firmware options

HID firmware for windows

The HID firmware allows you to connect a device (similar to a mouse) and not have to load a driver for them because the driver is built into the operating system. Therefore, you connect the HID programmed microDAC and windows will automatically recognize it as a HID device.

LIBusb firmware for linux

The LIBusb firmware operates in a similar fashion In linux with operating systems that have usb support built into the kernel (the newer kernels such as the one included with Ubuntu 9.04 distribution).

You need to have sufficient permissions to connect to linux USB devices, and editing UDEV configuration files and assigning them to a group may be necessary. As an alternative to that, you can execute a “sudo sh” on Ubuntu distributions to elevate yourself to root and run the binary that way. It will then be able to access the microDAC USB device.

Programming the right firmware onto your device

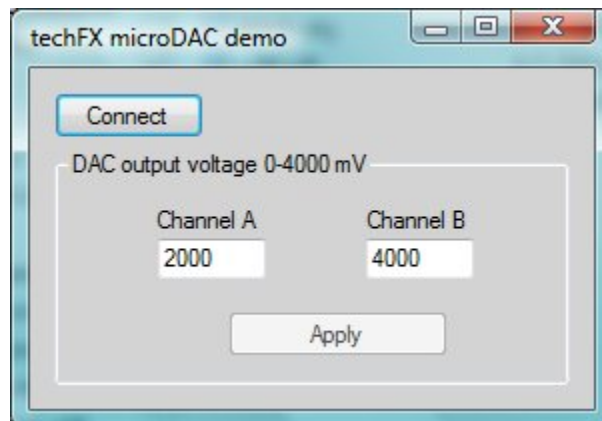
You will need to download from www.microchip.com MPLAB IDE 8.30 or newer and MPLAB C18 Compiler 3.3 or newer to program the sourcecode if you are using an ICD 2 or 3 programmer (or making changes).

If you just want to upload the premade firmwares, then you can do that with a PICKIT 2 or PICKIT 3 programmer since it uses a separate windows programming utility.

Using techFX microDAC in Windows XP and Vista

(you must have the HID firmware programmed on the controller)

1. Set Power jumper pin to “USB” for USB powered device. Set to “Self” for Self powered device through Vdc_in pin. (This is your preference, for best DAC /ADC accuracy do not use USB power). **USB power on the jumper is when it is set in a position closest to the edge of the board. Setting the jumper inward is Vdc_in pin power from a wall adapter or other source.**
2. If not using USB power, connect DC wall adapter 6-9 Vdc to Vdc_in pin.
3. Connect mini-USB cable to controller and a free USB port on your computer.
4. Observe that a USB connect sound occurred or go to system in control panel and observe a new “HID” device when you connect the controller.
5. Install Microsoft Visual Studio 2005 Express edition (free) and Net 2.0 + updates.
6. Run the windows demo from the CDRom.
7. Pressing the “connect” button will activate the apply button if it can connect to the controller.
8. Setting the 2 DAC channels (in millivolts) and pressing apply sends that to the controller, and those 2 output pins will be at that voltage. Note that the max voltage level with gain=2 (default in firmware) is 4.096 volts—sending a value higher than 4096 will result in zero output at that pin).



Using the techFX microDAC in Linux

(you must have the LIBusb firmware programmed onto your controller)

(this guide will assume Ubuntu linux 9.04, please adjust your LIBs etc accordingly to your distribution)

Install necessary LIB's such as: G++, LIBusb, QT3, QT4, QT3TO4, QT3-developer. You must have the LIBusb firmware programmed on the techFX microDAC to use it under Linux.

1. Set Power jumper pin to "USB" for USB powered device. Set to "Self" for Self powered device through Vdc_in pin. (This is your preference, for best DAC /ADC accuracy do not use USB power). **USB power on the jumper is when it is set in a position closest to the edge of the board. Setting the jumper inward is Vdc_in pin power from a wall adapter or other source.**
2. If not using USB power, connect DC wall adapter 6-9 Vdc to Vdc_in pin.
3. Connect mini-USB cable to controller and a free USB port on your computer.
4. Goto the SYSTEM/LOGS/SYSLOG in ADMIN and see if the controller has enumerated under USB. If you cant see where it is, you can look at the log and unplug it—observe—then plug it in again.

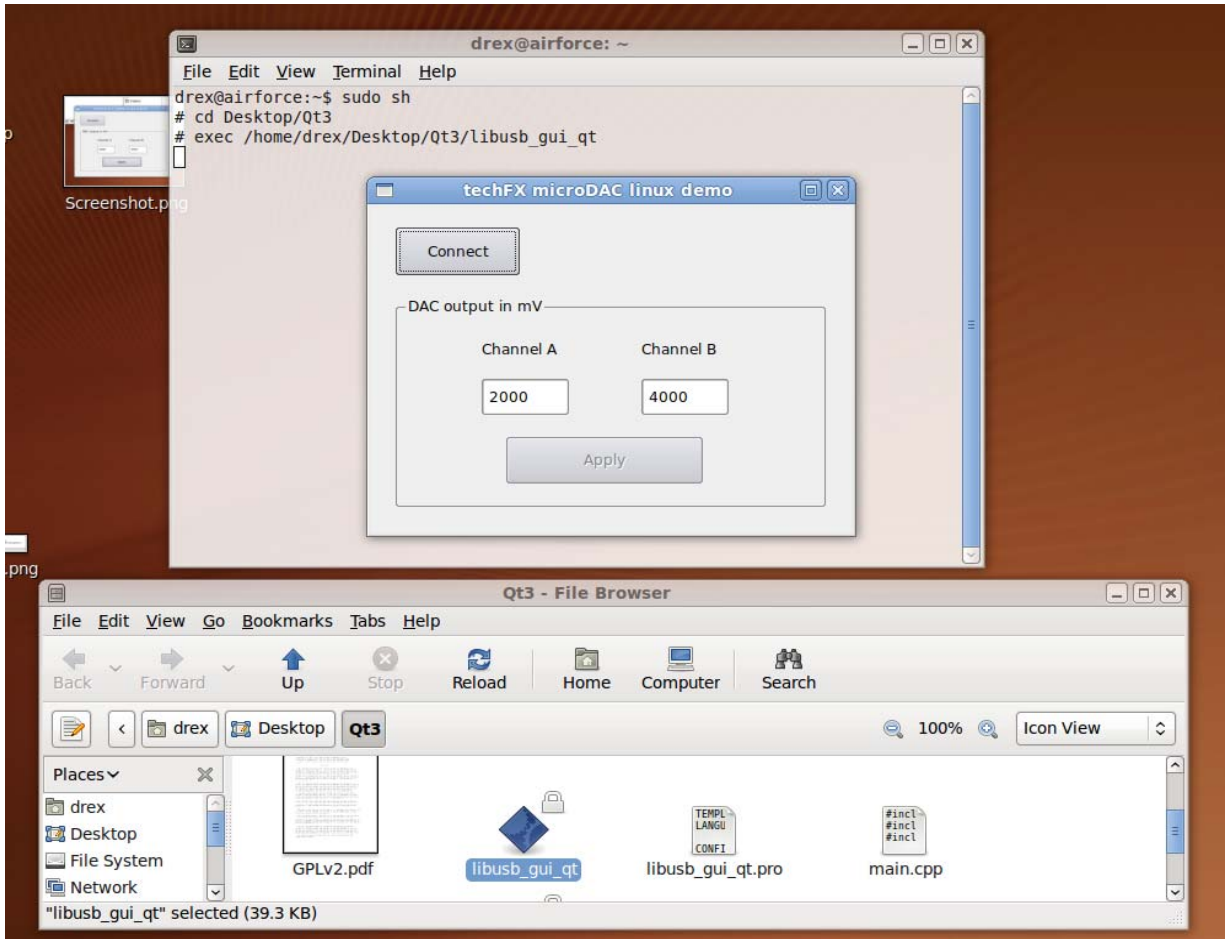


```
Jun  1 19:47:01 airforce kernel: [ 29.696514] eth0: no IPv6 routers present
Jun  1 19:47:46 airforce kernel: [ 74.500012] usb 5-3: USB disconnect, address 2
Jun  1 19:47:49 airforce kernel: [ 78.000040] Clocksource tsc unstable (delta = -327436797 ns)
Jun  1 19:47:52 airforce kernel: [ 80.832033] usb 5-3: new full speed USB device using ohci_hcd and address 3
Jun  1 19:47:52 airforce kernel: [ 81.005905] usb 5-3: configuration #1 chosen from 1 choice
```

last update: Mon Jun 1 19:47:01 2009

5. Now that the USB is connected, start a terminal session. To connect to the USB device we need to have permissions. This can be done with UDEV but we will do it simply by elevating to root in the terminal window. "sudo sh" is the command you type. Now goto the directory where you copied the Linux source from CD. You can then do the "exec /path/libusb_gui_qt" command to run it, and the box will show. When you press connect, the apply button will become clickable if it worked (that sets the DAC output voltages).

(see following picture)



Device Characteristics

Vdc_in input voltage	6-9 Vdc
Max amperage rating	500 mA
Max amperage per pin	25 mA
Max amperage total pins	200 mA
DAC output voltage range	0-4.096 V
Max input voltage at pin	5 V
Temperature range	0-85 deg C