DE0\_Nano\_DAC repository contents

*DAC\_12bitController\_Eagle* contains the Eagle files for the Breakout Board schematic and layout. This is so that a user can have more boards printed out or edit the board if additional features are necessary. The schematic shows the electronic components necessary to stuff the board. The FPGA code included in this repository with a single DE0\_Nano board is designed to be used with one such daughter board.

*DAC\_control\_files* contains both some python code and a complete C++ control program written in the Visual Studio 2013 environment. The python code is written so that the file *convert\_voltages\_to\_waveform\_line.py* can automate writing a waveform file compatible with the DAC system. The C++ executable is in the “Debug” subfolder. To run the executable, the Visual Studio runtime environment must be downloaded and installed. To compile the code yourself, it requires Visual Studio 2013 or later.

*DE0\_12bitDAC\_controller* contains the FPGA code. It is written in VHDL in the Altera Quartus Prime development environment. This code defines the logic for the FPGA to handle digital signal analysis, for processes including communication with the computer, logic pulse sequencing, signal processing to read ADC values, and output DAC values.

Device Information and Instructions

The FPGA controller is built on the DE0-Nano FPGA evaluation/educational board by Terasic, with information found here: [DE0-Nano Cyclone IV](http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=139&No=593)

Terasic's DE0-Nano board is billed as an educational evaluation board. It includes a large amount of input/output logic channels, an 8-channel ADC, an accelerometer, SDRAM with huge amounts of data storage space, and several LEDs, buttons, and switches. It is built as something that is general-purpose and easy to use. Our custom DAC and Logic expansion module makes use of many features of this board, including status LED lights, logic for DAC control and logic sequencing, ADC for processing external voltage levels, and logic inputs that are used to handle receiving data from a control computer. Any of Terasic's breakout boards that are listed as having 40-pin header expansions are all pin-compatible, so any such board can be compatible with the designs explained here if future versions require updated FPGA boards.

The USB computer communication and ADC are interfaced through a solderless breadboard, to allow adaptability to adding several extra logic I/O and using different ADC channels. The actual communication is handled by the UM245R evaluation board. This board has a USB connection, a jumper to set its logic to 3.3V, and logic lines that can be connected to the 26 pin connector on the DE0-Nano board. The ADC inputs are also on the 26 pin connector, and the value range is 0V to 3.3V. A 3.3V Zener diode with 200 Ohm regulator circuit is highly recommended for each ADC input.

DAC Breakout Board

The breakout board is designed with a pair of AD7945B 12-bit DACs which allow up to 2 MHz update rates and the option to put in whichever op-amp is necessary to achieve precision or timing requirements. The board also includes a logic buffer chip to convert the DE0-Nano logic standard of 3.3V CMOS to drive cables, or to convert incoming logic to 3.3V CMOS inputs to the FPGA (since our labs typically use 5V TTL, this chip ensures that we can safely drive logic into the FPGA). The DAC board is designed to be powered by +/-12V, and via a jumper can also power the FPGA board. The FPGA board can alternatively be powered by its own +5V supply or via USB cable.

Each logic input/output and DAC output is in a 2-pin header, designed to be connected to twisted-pair wires. This mitigates pickup noise on each line, and can be soldered onto a BNC through-hole connector on the front plate of your enclosure for the system.

Computer communication

After wiring together all the components, and soldering all components onto the DAC breakout board, computer communication is achieved through the UM245R evaluation board. When plugged into a computer, the FT\_PROG program is necessary to configure the UM245R board, found here: [FT PROG](http://www.ftdichip.com/Support/Utilities.htm#FT_PROG)

Set the device ID to any of the following: DACBRD00, DACBRD01, DACBRD02, or DACBRD03. The C++ control software looks for USB devices with these device IDs, and if found, opens the connection. If multiple devices of the same ID are connected, it will cause errors when transmitting data via the control program.

Defining Logic and Waveform files

Logic files can be any file-type extension, but are space or tab delimited. The format is designed to be

(1st time interval) i d1 d0 l3 l2 l1 l0  
(2nd time interval) i d1 d0 l3 l2 l1 l0  
etc…

where the time interval is in milliseconds to denote how long that logic state should be held before proceeding to the next logic state in the file. The tokens listed represent certain logic being “on” if present, and “off” if absent. i denotes activating the interlock, d0 triggering Dac0, d1 triggering Dac1, and l3, l2, l1, l0 referring the Logic 3, … etc logic output lines. At the end of the logic sequence, the logic outputs retain their state until the next logic sequence is triggered.

Waveform files can be any file-type extension, but are space or tab delimited. The format is designed to be

(time\_1) start end  
(time\_2) start end  
etc…

where in this case, the times are an ongoing time stamp for the waveform. Each line has a duration of (time\_n – time\_(n-1)) and the final waveform time listed will be the total duration for the entire waveform. The logic file uses the token d0 or d1 to internally trigger the next waveform segment, or the external logic input for the DAC can be used.