

# Teuthida Technologies Home

All about embedded systems engineering.

## Project: Sequencer – Part 1 [Updated]

Posted on [October 5, 2012](#)

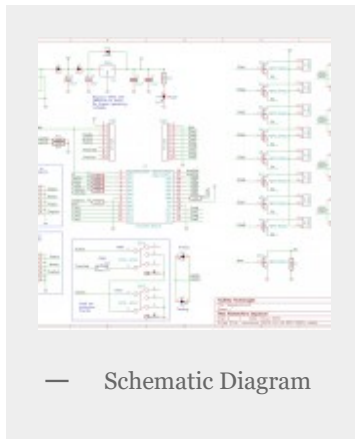
This is the first in a number of posts covering the development of a fairly simple sequencer board. This board is designed to be easy to put together, and could possibly be the basis for a simple kit. It is intended from the outset, that this be an open source hardware and software development.



This is not my first crack at designing a sequencer board. See the picture to the left; In 1999, Bret Walters and I came up with our first sequencer design for a fireworks show. This early design reflected our desire to keep things simple. I wrote the firmware for the PIC16F84 in CCS C, and the PC based, scripting and control language was developed by me, but I'll be darned if I can remember much beyond that.

This development was quite a learning experience. A lot of time has passed, so my troubles with the CCS compiler should not color your view of it now. What I really do recall was that my grasp of the requirements of a sequencer left a lot to be desired. For instance, the 1999 board lacked a continuity check capability. This crucial feature allows the operator to determine if all of the circuits are connected properly to the controller. We left this out on the assumption that it was simpler to just wire things up correctly in the first place. Wrong! On a recent Discovery Channel Show called "[Pyros](#)", professional pyrotechnicians showed time and time again how important it is for a show to pass its continuity tests. We noticed as much when our test show pretty much suffered a massive failure to launch.

This new board addresses this serious failing. The 1999 board also used two power supplies. A 12 volt supply for the micro (regulated on board down to 5 volts) and a 48 volt supply for the ignitors. The new board has a single, flexible supply to meet multiple needs. Other changes are that this board is designed using the open source [Kicad](#) package which did not exist way back then and the PIC will be programmed with Microchip's brand new [XC8](#) compiler.



Enough of the old and onto the new! To the left is the Kicad generated schematic of the new sequencer. Using Kicad for schematic capture was easy, with one caveat. The parts library is the sum of what developers wanted. They (unlike me) where crazy about Atmel, less so on Microchip. The result is that if you use a more modern part, prepare to create your own symbols.

The design itself is fairly typical of simple PIC designs. A micro-chip surrounded by I/O circuitry of all sorts. In spite of this, there are some aspects of the design that should be highlighted:

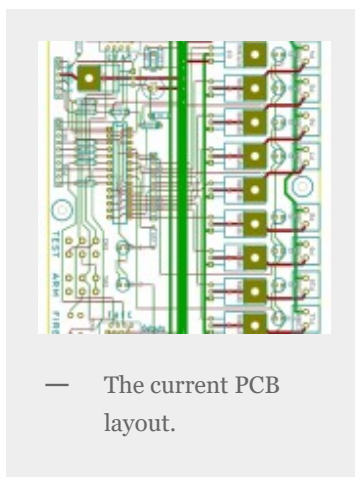
- Unlike the 1999 version, there's no need for a crystal oscillator and its tricky layout and twitchy capacitors. Many modern PIC chips supply a robust internal oscillator.
- The debug signals, Vpp, IcsDat, and IcsClk are dedicated to their role in debugging.
- The 8 outputs are from Q2 through Q9. These are all tied to Q10 which is in parallel with R2, a 22K resistor. If Q10 is off, it does not conduct and firing current must flow through R2. This current is low enough to NOT fire the end effect, but can still be measured to confirm continuity. When Q10 is on, a large amount of current may flow, firing the end effect. Thus Q10 is the "Arming" transistor.
- Like classic TV's channel 1, Q1 was removed and the other transistors were not renumbered. Thus there is no Q1.
- The ten LEDs are connected to six PIC I/O lines in a simplified [Complimentary Drive](#) configuration using a single current limiting resistor.
- The Test, Arm, and Fire switches have hardware interlocks to prevent a fire signal from triggering an end effect in the wrong mode. This is in addition to software checks to prevent this from happening. When firing things off, you can never be too careful.
- The board is designed to be daisy chained to allow for more than 8 end effects. Various parts of the design may be omitted based on where in the chain the board is installed (at the head, or in the tail).
- The power is derived from a simple 7805, 5 volt voltage regulator. This is perfectly adequate for the standard 12 volt input. For higher input voltages however, the 7805 is replaced by a drop in, buck regulator module that is more efficient.

Once the schematic was captured and verified carefully, a netlist was generated. Then, a footprint was selected for each part. As before in the parts library, the footprint library is a bit chaotic. Several custom footprints had to be created even for this simple design.



Once this done, the data was imported into the pcb layout software. Well, that is a bit of a simplification. The reality was that the custom footprints required several passes and a lot of painstaking scrutiny to get right. This is most certainly an error prone step that must not be rushed.

Still, eventually the import was fully completed and the focus shifted to the layout and how the board would be used in the field. Connectors were moved, parts dragged about until a harmonious arrangement was found. Then reviews of the routing of analog signals, power, and control signals were corrected and verified.



Finally there was a final stage of proofreading and checking. The true proof will be in the testing, but that has yet to happen, The next step is to order prototype PCBs and populate, program and test them. That should be easy, right? 😊

As always, your comments and suggestions are most welcomed.

Peter Camilleri (aka Squidly Jones)

**[Updated]** The sharp eyed will have noticed that the schematic does not quite match the PCB. That's because that schematic was just a bit out of date. For an up-to-date schematic click on [Pyro Seq Schematic](#).

---

**SHARE THIS:**



---

**LIKE THIS:**

Loading...

This entry was posted in [Projects](#) by [Peter Camilleri](#). Bookmark the [permalink \[http://teuthida-technologies.com/?p=1410\]](#) .

