V8CE-WFD Hardware Construction

These are the construction details of my prototype "Spacewar" peripheral for the PiDP-8. AFAIK, this is one of the first non-IoT (Internet of Things) applications of the ESP8266.

Schematic and build notes:

This is the first time in 40+ years of goofing around with electronics that I have ever used a CAD program to do a schematic - they usually were done in my head, or on paper. So:

1. Have mercy on me.
2. There COULD be a mistake; check pins with spec sheets and use common sense.

The power comes from a USB wall wart of 1A capacity with expected good +5 regulation. Do not attempt to power this from a USB port from a computer, because the ESP8266 is very "peaky" in current demand and this design overclocks the ESP8266 on top of that. The +5 is dropped down to 3.3volts by an 1117. An addtional L4931-33 is with additional filtering used for local regulation (theoretically more precise) to the TLV5617 and its voltage reference - whether this was needed or not is not certain. You might be able to get away without it and use the main 3.3V directly. An ICL8069 volt 1.2v precision voltage source is used at the TLV5617. You could cheat and use a divider or a Zener here (the TO92 contains basically a fancy, precise Zener). But, I wanted a rock-solid display, and remember we are dealing with 2.5 volts /1024 here (the TLV5617 multiplies the reference by two).

The input ports are provided by the HC165 as an input port expander for the paddle buttons. Pullups are used at the button inputs. The inverted output at the HC165 is used (could have been done in software). Clock/data lines are shared between the HC165 and TLV5617 because I wanted to be conservative with port usage. So, they cannot be operated simultaneously (probably not an issue).
I selected the TLV5617 (not cheap or common) because of its good settling time. You MIGHT be able to get away with a less expensive D/A with possibly some redesign and recoding. Resistors pull to ground and are at the output in order to operate in spec, to prevent ringing upon output voltage changes, and some output protection. You might be able to live without them, maybe not.

---------->IMPORTANT NOTE:!!!!!<----------

The TLC5617 referred to in the schematic is no longer in production; but the TLV5617A is an exact equivalent in production (I believe that TI standardized its naming conventions).

The serial I/O is used for initial programming and debugging, and is not important in normal operation. It is usually connected to a USB-to-serial adapter. The USB devices run at +5 volts and the ESP8366 is a 3.3V device, so a resistor network for input and a resistor for output are provided. If you have a 3.3V USB serial device (not common at this time), you may want to bypass these. But keep in mind that the ESP8266 may not be very 5V tolerant and it could be damaged if used this way with a +5 adapter.

BEWARE of cheap cloned USB serial adapters! (read on the Internet about this issue). They may or may not work with them with stock Windows drivers. Save yourself a lot of trouble and get a genuine Prolific or FTDI adapter.
The "Z" axis is provided and is recommended, but might not be necessary, depending on your output device.

Physical Construction:

Surface-mount adapters were used for the ESP-07 and TLV5617 so that it could exist in my ancient, wire-wrapped world. Your construction techniques may be different. I suspect that had this been made on a surface-mount PCB, it could probably be done on a 4x2" or even smaller PCB. There isn't that much to it.
I put the serial USB to a 1/8" stereo jack. This is not really good practice, but it is what I had laying around. If you remove it while the computer is ON, you will short the output and the USB chip in your computer will shut down and error out, probably requiring a power-off computer restart. As I understand it, the protection circuitry in the computer is designed to do this a FINITE number of times. SO, if you build it this way (NOT recommended), remove the USB from the computer FIRST. Ask me how I know this.

Even if you can get the ESP-07 programmed by someone else, implementing the programming/serial port to USB and the programming switch is recommended. There will be upgrades/software changes and you can then solder in the ESP-07 permanently (mine is socketed). For those in the know, there wasn't enough room for over-the-air software upgrades although by using the larger-capacity flash of the slightly more expensive and harder-to-find ESP-12F, this would be possible and will probably be used in the future.
The BNC connectors and cables were overkill, but keep in mind that you are working with high-speed analog signals operating at only up to about 2.5 volts. Many X-Y displays, including mine, have a DB-25 connector on the back for X,Y,Z input.

But, Noise is Bad.

The ESP-07 is used, but as mentioned, the -12 could also have been used (it has a cheaper PCB antenna, the -07 has a chip antenna and an antenna output). Other variants may not have enough I/O. Operating in the same room or a small house, an external antenna is really not needed. If you are bread boarding, an adapter for the odd footprint is recommended (I made mine, but it's cheaper and easier to buy). Be aware that a pre-made adapter may have some pull-ups for the GPIO's already installed.

---------->IMPORTANT NOTE:!!!!!<----------

There is a great deal of confusion about GPIO4 and GPIO5 on the ESP-07. There were, and are, many units out there (including mine) where the silkscreen is INCORRECT. There are pinouts on the Internet that are also INCORRECT. The pin that I used is the THIRD DOWN from the TOP of the chip (pin14, GPIO5). Use this regardless of what your ESP-07 silkscreen says.

Paddles

I deliberately made the paddles look “prototyped” and used cheap switches. A DB-9 was used to connect them because, again, that’s what I had to work with. You may have a better connector available.