Hi,

I have several questions about the ESP32 and Fujinet

Things I think are true

* For Fujinet purposes H8 and H89 are synonymous, though the hardware implementations will be different.
* My solution refers to <https://github.com/DarrellH89/H89_ESP32>
* The ESP32 runs FreeRTOS by default. My solution never specifies a core, so it is running on one core only.
* If a process is specified to run on a specific core, then all code called by it runs on the same core
* The H89 writes to port 0x7E to start a command sequence, which generates an ESP32 interrupt function.
* The H89 looks for status H89\_READ\_OK. It then reads from port 0x7C which generates an ESP32 interrupt function to set the status to ESP\_BUSY. The ESP code can then load another byte into the data latch and set the status to H89\_READ\_OK.
* The H89 checks for status H89\_WRITE\_OK and writes data to port 0x7C which generates an ESP32 interrupt to process the data as a command or data as appropriate.
* The H89 runs CP/M natively and doesn’t need this Fujinet function unless speed is an issue.
* I need to run software on the H89 that supports the Fujinet disk emulation. From a disk implementation point of view, IMG files make the most sense for the H89. The FlashFloppy config files will be useful here.
* I should change my communication design to use Tasks and Queues to eliminate my buffers and perhaps some overhead

Things I’m not sure about:

* My solution stores network information (webpage user & password, SSID, Network password) in the EEPROM NVM. It looks like this might conflict with Fujinet?
* Since the main task is set to run on Core 1, can I set my communication tasks to use Core 0?
* My use of a local webpage to transfer files is conflict with Fujinet?
* Why isn’t OTA included? Or is that only useful for developers?
* Do we need to support the Fujinet buttons in our hardware? Current discussion was to handle this via extra bits available on the 8255 design Joe is working on.