FEZ TinyCLR SPWF04SA Update and Networking (31.12.2018)

Fez Site at GHI:
https://docs.ghielectronics.com/hardware/fez/intro.html#resources

FEZ Schematic:
file:///E:/GHI%20Codeshare%20Projects%20and%20Hardware/Boards_and_Modules/Board%20FEZ/FEZ%20T18%20Rev%20D%20Schematic.pdf

Here is a link to the GHI Forum. It shows how things have to be done:
https://forums.ghielectronics.com/t/network-samples/21609

Preparing the FEZ-Board:

1) **Open a virtual serial port (Bootloader Mode)**
 Hold down the BTN1 button and reset the board.
 This is needed to erase the firmware on the board.
 Press `E` and then enter to start the erasing procedure.

2) **Install GHI Bootloader (DFU Mode):**
 Put your board in DFU (Device Firmware Upgrade) mode by holding down the Boot0 button and resetting the board.
 Run the DfuSeDemo program.
 Near the bottom of the DfuSe Demo program window click on the `Choose...` button.
 Find the bootloader file, e.g. FEZCLR Bootloader v2.0.4
 Now click on the upgrade button.
 https://docs.ghielectronics.com/hardware/fez/tinyclr.html

If needed: Use GHI Utility **'TinyCLRConfig'** to update the firmware.

Before starting the **firmware of the SPWF04SA module should be updated:**
GHI Instructions: Updating the SPWF04SA Wi-Fi Module Firmware

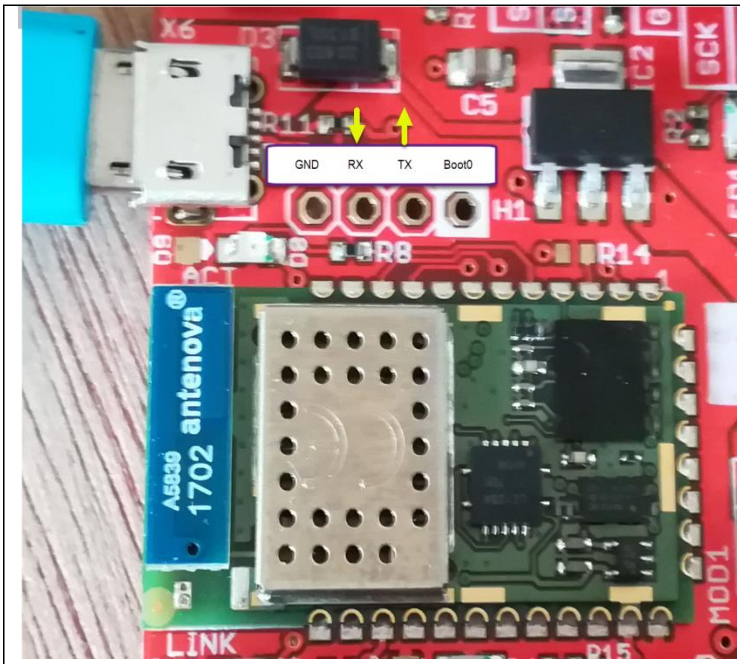
http://docs.ghielectronics.com/hardware/components/spwf04sa.html

Instead of erasing the firmware as suggested by GHI you can use this little Program to bring the FEZ Board in the desired condition:
'**FEZWiFiUpdatePreset**' (see GitHub Repository)

Here are some additional infos about updating the `SPWF04SA firmware:`

Before it works, the firmeware of the `SPWF04SA module has to be updated:`
Instructions from ST (See Instructions from GHI later on this page).
https://www.st.com/content/st_com/en/products/wireless-connectivity/wi-fi/spwf04sa.html#design-scroll

John_Brochue: I’d recommend going the UART way. The four-pin header near the Wi-Fi module and USB connector exposes the UART pins for the Wi-Fi module itself. That module has an ST processor inside that you can connect to using the various ST DFU tools to erase the chip and load the firmware. The firmware to load should be available in the download package from ST on the module page.



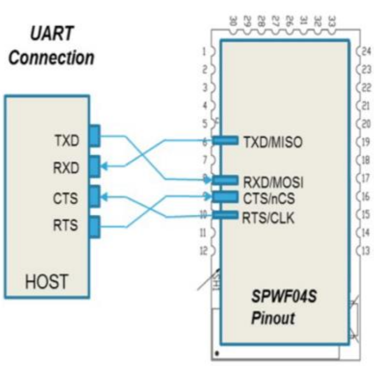
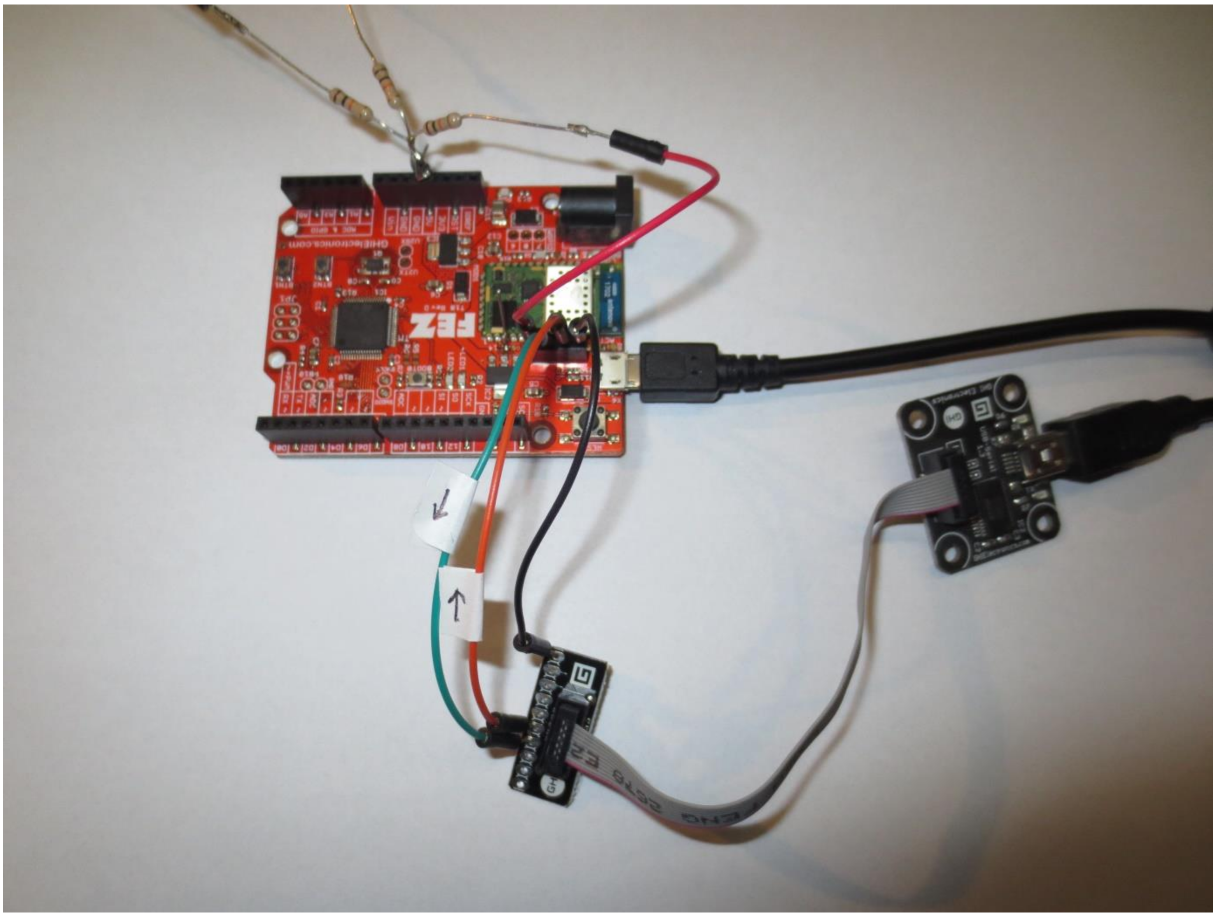
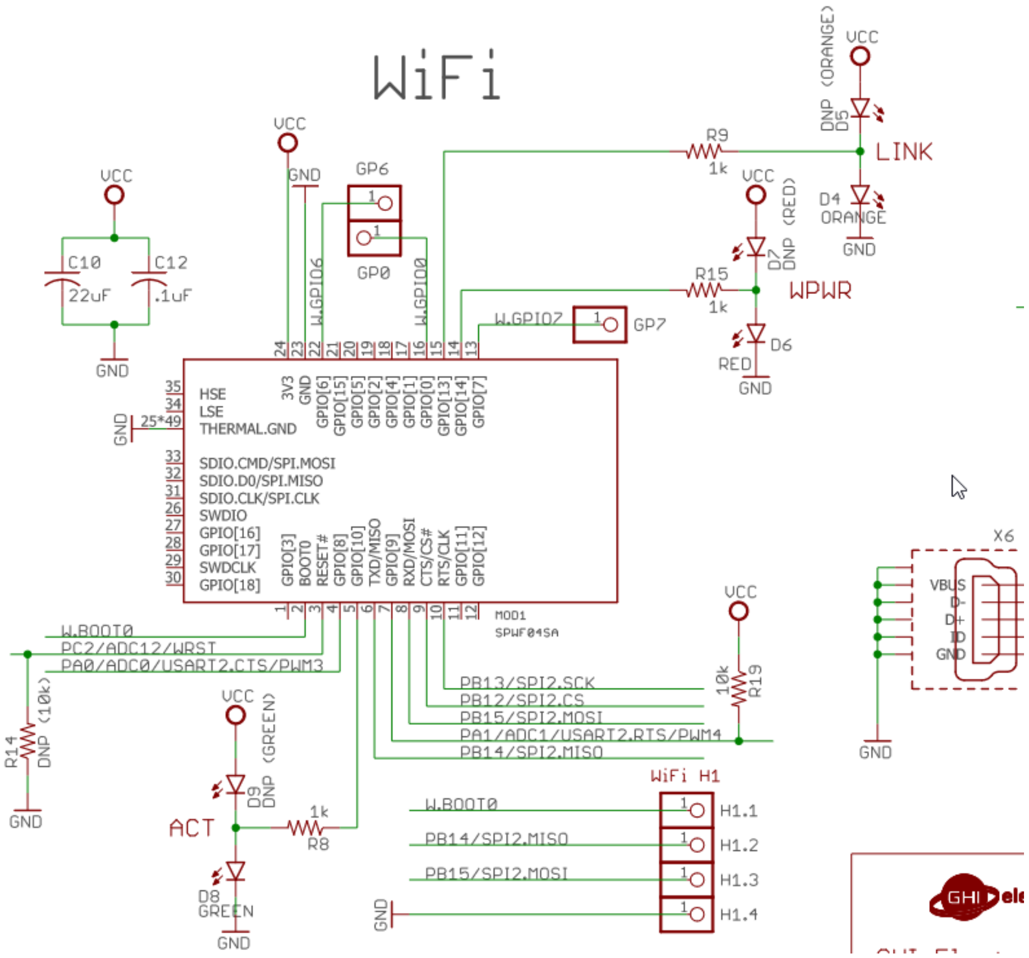
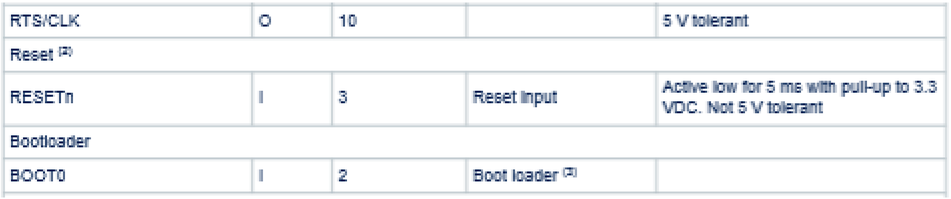


Figure 2. UART connection with host device







RTS/CLK	O	10		5 V tolerant
Reset [2]				
RESETn	I	3	Reset input	Active low for 5 ms with pull-up to 3.3 VDC. Not 5 V tolerant
Bootloader				
BOOT0	I	2	Boot loader [3]	

X-NUCLEO-IDW04A1 Expansion Board
https://www.st.com/resource/en/data_brief/x-nucleo-idw04a1.pdf

FEZ:
https://github.com/ghi-electronics/TinyCLR-Drivers/issues/28
Since the firmwares now default pins to input-pullup, the Wi-Fi module is put into python mode by default. We don't want to change the firmware for this one pin because it's very specific and not every use of the firmware will have Wi-Fi. Adding a constructor parameter for the pin is probably the best way to go, perhaps call it mode. Will need to add to the pins library as appropriate. For now the pin can manually be set to low if needed.

SPWF04Sx:

https://www.st.com/resource/en/datasheet/spwf04sa.pdf

Main function is set at the module initialization when "factory" configuration is used. The GPIOs are all set as input pulldown with the exception of GPIO7 that is set as input pull-up

To enable the firmware download, pin BOOT0 needs to be high during **power-up**. RESETn needs to be pulled low at least 5 ms to initiate the firmware download sequence.

When GPIO[8] is high at the **start-up** the module enters the MicroPython mode.

To select the SPI as host interface, this GPIO[9] must be driven high at the reset time.

Here is an example by GHI to test the functions of `SPWF04SA module`.

https://gist.github.com/Arke64/d5439537c76ba67bd28609f31ada4572

Actually GHI's example doesn't work as is. The following Commands have to be added to prevent the board from working in python mode.

```
private static GpioPin _pinPyton;
….
….
_pinPyton = cont.OpenPin(FEZCLR.GpioPin.PA0);
_pinPyton.SetDriveMode(GpioPinDriveMode.InputPullDown);
```

This is the name of my working test project **'FezWiFiTest'.**

STM-AT-Commands:
https://www.st.com/content/ccc/resource/technical/document/user_manual/group0/69/bf/26/08/53/62/49/af/DM00329655/files/DM00329655.pdf/jcr:content/translations/en.DM00329655.pdf