Setting up Raspberry Pi for Tensile Tester operation

OS version : raspios bullseye

1. Follow the usual steps to create a RPi image and boot
    1. It is easiest to use the dedicated application, *Raspberry Pi Imager*, that creates the SD card for your RPi
    2. Choose to customize the installed image
    3. Set the hostname to `tensilepi`
    4. Set the username to `tensile` and the password to `t3ns1le`
    5. You can try to set the wireless nework to UWS and your password, but usually this doesn't work because UWS is, well, UWS :-(
    6. Set the wireless LAN country to CA
    7. Set the locale to Edmonton and leave the keyboard a US
    8. Change to the Services section and enable SSH. Select "Use password authentication"
    9. Save and say yes to using the custom settings, then yes to start formating the image
2. When the Imager is finished, pop the SD card into the RPi and boot
3. The very first boot will take some time as the OS configures itself.
4. Connect to Wifi
    1. It appears that UWS at the UofA will not work
    2. An option is to tether to a cellphone, which does work well.
5. Update the system via the terminal; `sudo apt update`, then `sudo apt upgrade`
    1. This may take a while
6. Enable VNC
    1. `sudo raspi-config`
    2. Navigate to Interface Option and hit enter
    3. Navigate to VNC and hit enter
    4. Set the VNC server to be yes, hit enter
    5. exit `raspi-config`

6. reboot

7. Test the VNC connection using your laptop

    1. On a Mac, install `tigervnc-viewer`, for example
    2. Get the RPi IP address
    3. Connect using the VNC viewer

8. Install and configure `node-red` using this guide

    1. Execute in a terminal the following

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

    - Install the RPi specific nodes
    - Don't set up user security
    - Don't set up the Projects feature
    - Name the flows file `flows_tensilepi.json`
    - Use `t3ns1le` for the catchphrase
    - Choose the default colour scheme
    - Select the default font, monaco
    - Allow Function nodes to load external modules

    2. Enable node-red to start at boot

```
sudo systemctl enable nodered.service
```

    3. Check if it's running by `ps ax | grep node` and if not, start using

```
sudo systemctl start nodered.service
```

    4. Now you can bring up the node-red flow editor on a computer on the same network as the `tensilepi` by connecting to `http://tensilepi.local:1880` in a browser

9. Install the node-red dashboard and git packages

```
cd ~/.node-red
npm i node-red-dashboard
npm i npm install node-red-contrib-git-nodes
```

10. Install and configure pigpio
`sudo systemctl enable pigpiod`

11. Restart node-red on the RPi

    ```
    node-red-restart
    ```

12. Once restarted, you can bring up the node-red dashboard on a computer on the same network as the `tensilepi` by connecting to `http://tensilepi.local:1880/ui` in a browser.

    - changes made in the flow editor will be reflected in the UI once they are "deployed"

13. The flow editor can be accessed by connecting to `http://tensilepi.local:1880`
14. We need a couple of contributed nodes. They are installed from the node-red root directory, so `cd ~/.node-red`
15. Install the PID node; `npm install node-red-contrib-pid`
16. Install the UI List node; `npm install node-red-node-ui-list`
17. Restart node-red; `node-red-restart`
18. Now, `PID` will show up under the function group and `list` under the dashboard group
19. The next thing to do is import the `flows_tensilepi.json` file to get back all our control UI and algorithms.
20. Using the "stacked" aka "hamburger" menu at the top right, choose Import.
21. Select the `flows_tensilepi.json` file from the repo and import.
22. The Flow should appear as a tab.
23. Deploy it and check that the UI is now there.
24. Last, build the C modules

    ```
    cd tensile-tester-flow/software
    chmod +x build.sh # just in case
    ./build.sh
    ```

25. End