Setting up Bluetooth serial connectivity from your Pi

Configure the btserial.sh as described in section "Editing the Image" from the document: https://hacks.mozilla.org/2017/02/headless-raspberry-pi-configuration-over-bluetooth/

My findings are that you'll only ever need to run the btserial script once as root (sudo ./btserial.sh). This will create or update and enable the required service to run and from then on every time on bootup.

Also, you don't need to do this on an image, you can do it straight away on a mounted file system if you have access somehow already.

The screen stuff won't work in Cygwin, but you can use puTTY. You can connect to a serial port in puTTY. Just figure out which COM port (in Bluetooth settings) is used for the outgoing connection to the Pi's Bluetooth connection, then configure puTTY to use this port. The port you're connecting to on the Pi is 115200 (as configured in the btserial.sh).

Connect via puTTY, then you can use sudo raspi-config to configure the network, as described here: https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md

Connecting this way with Patrick's version doesn't require you use a password, so it's really wide open. I opted to remove the credentials from my version of btserial.sh, so you'll need to enter your username and password every time you log on. A method which I much prefer.

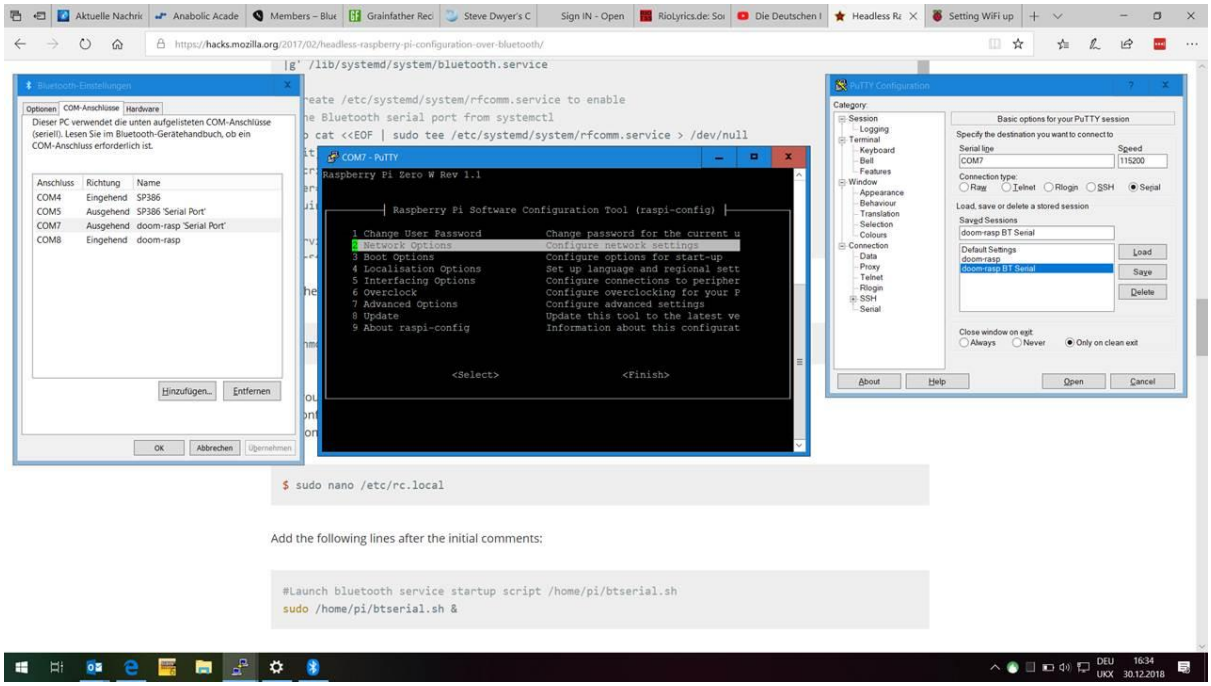
If you're still paranoid about having logins possible over Bluetooth other than your own, to disallow users connecting over the Bluetooth serial port altogether, you'll need to disable the rfcomm service (sudo systemctl disable rfcomm). This will stop users connecting over the Bluetooth serial port, and will stay disabled after subsequent reboots. To be able to log back in over Bluetooth serial, re-enable the service (sudo systemctl enable rfcomm) while gaining access some other way, such via ssh or VNC.

To get serial Bluetooth working from an Android phone, fork https://github.com/mjarvisal/termux-app, build that in Android Studio and deploy the APK built during the build to your Android.

Basically, wherever you are with your Pi, you won't need a screen & keyboard to get it set up. With the Bluetooth you can obviously set up the wifi wherever you are, if that's available, with sudo raspi-config, or at least query your ip address with ifconfig.

Once you have an IP address, you can obviously use VNC. That's available for Android but of course is much easier to work with on a laptop. But if there is no wifi available, you can perform all the cli tasks that you might need to over Bluetooth, with either puTTY or Termux on Android (not available on the Play Store release, you'll have to build this yourself).

This does enable you however to be able to connect to any Pi from your laptop over Bluetooth in order to set up the wifi, as long as the btserial stuff has been set up first, meaning you'll never have to lug around a screen, keyboard & mouse anywhere ever again.



Disclaimer:
These instructions don't cover how to get the rfcomm service running on an image prior to installing Linux on your Pi, they simply assume that this has been done and you've gained access to your Pi some other way prior to going out on the road with it.