# Setup Guide, Power Meter Controller

NA6O Jan 2023

This is the initial setup guide for the NA6O power meter controller based on Raspberry Pi 3 B+ platform with software developed in LabVIEW Community ed. 2022.

**Initial Raspbian Image Installation**

Initial image installation is a download, then write the Raspbian image to a blank 8 GB microSD card. You will need an adapter to plug in that card. After this is done, follow all the instructions that follow to customize the OS. Then install and run the LabVIEW development system to download the LINX support and the target application. **Proper version:** **Debian V10 BUSTER** will allow LV2020+ to install properly. I had numerous failures getting LINX to load with later versions of the OS.

1. Get the OS image from

<https://www.raspberrypi.com/software/operating-systems/> Choose legacy OS V10 (Buster).

2. Use **Etcher** to write to SD card. <https://www.balena.io/etcher>

Boot up your system then go through all of the following setup items.

**Change hostname and password**

Preferences >> Raspberry Pi Conifiguration

System tab, at Hostname, set to **pwrpi**

System tab, click on Password, set to **your\_password**

**Enable interfaces**

Preferences >> Raspberry Pi Conifiguration

Enable SSH, VNC, I2C, Serial Port. All others disabled.

(Note: Enabling SPI seems to kill I2C)

**Set screen resolution to 1024 x 768 for VNC** (otherwise it's too small)

Preferences >> Raspberry Pi Conifiguration

Click on Set Resolution… and choose DMT mode 16 1024x768

**Set up serial UART**

sudo nano /boot/config.txt

If not already there, add a last line: enable\_uart=1

sudo nano /boot/cmdline.txt

(one long string) verify that it does NOT contain “console=serial0,115200”

# NOTE: If you run any of the Rpi configuration programs, the above items may be overwritten, so always check these settings last. Otherwise the LCD display won't work.

**Enable power-off output pin**

Add line to enable power-off output GPIO26 (pin 37):

sudo nano /boot/config.txt

Add another last line: dtoverlay=gpio-poweroff

If for some reason you want to use a different GPIO pin or change its polarity at power-off, the command has some options:

dtoverlay=gpio-poweroff,gpiopin=26,active\_low

Reference: <https://wiki.52pi.com/images/c/ca/1.How_to_use_1_GPIO_Pin_to_drive_RaspberryPi_Power-on-off-status_light.pdf>

**Disable WiFi and Bluetooth (Optional)**

sudo nano /etc/modprobe.d/raspi-blacklist.conf

Add the following lines (file is probably empty):

#wifi

blacklist brcmfmac

blacklist brcmutil

#bluetooth

blacklist btbcm

blacklist hci\_uart

**Install GPIO Info**

sudo apt install gpiod

This gives you a nice command gpioinfo that shows which pins are in use.

(Found that pin 29 is taken by an LED...)

**Set Nameserver**

sudo nano /etc/resolv.conf

domain

search

nameserver 10.59.100.1 (IP of your router)

**OPTIONAL: Set Fixed IP Address**

sudo nano /etc/network/interfaces

Comment out the following line, if present:

# iface eth0 inet manual

Add the following lines:

auto eth0

iface eth0 inet static

address 10.59.100.125 (Desired fixed IP)

netmask 255.255.255.0

network 10.59.100.1 (IP of your router)

broadcast 10.59.100.255 (.255 within your domain)

gateway 10.59.100.1 (IP of your router)

1. **SSH Notes**
2. For SSH on Windows, use Putty (port 22, default).
3. For SSH on Mac, from Terminal: ssh <IP adr> -l pi

# VNC Notes

No need to install anything custom. Defaults are fine. Some options available but not required:

Enable VNC in Raspberry Pi Configuration, Interfaces tab.

On taskbar, click on VNC icon and go thru setup.

Upper right corner menu, choose Options…

Security: set Encryption to Prefer On and Authentication UNIX password.

Connections: Check Allow direct VNC connections over TCP, set port to 5900.

VNC will auto-start at boot time. Connect with VNC Viewer available from. Reconnects when rebooted.

Login directly via VNC:

Hostname = **pwrpi** (or use the IP address)

User = **pi**  (lower case!!!)

Pass = **your\_password**

# I2C Tools

(This may be included with the OS, depending up the version)

Default I2C interface number is **1**.

Install some basic test tools

sudo apt-get install -y i2c-tools

Show available busses

i2cdetect –l

Print table of devices on bus 1 (i2c-1)

i2cdetect –y 1

Other commands:

i2cget

i2cset

i2cdump

The default ADS1115 ADC will appear with address 0x48. (0x27 was for something else.)

pi@pwrpi:~ $ i2cdetect -l

i2c-1 i2c bcm2835 (i2c@7e804000) I2C adapter

pi@pwrpi:~ $ i2cdetect -y 1

0 1 2 3 4 5 6 7 8 9 a b c d e f

00: -- -- -- -- -- -- -- -- -- -- -- -- --

10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

20: -- -- -- -- -- -- -- 27 -- -- -- -- -- -- -- --

30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

70: -- -- -- -- -- -- -- --

pi@pwrpi:~ $

lsmod | grep i2c\_

That will list all the modules starting with “i2c\_”. If it lists “i2c\_bcm2708” then the module is running correctly.

# LabVIEW Development System Installation

Download and install the LabVIEW Community Edition. 2022 Q3 was used for development but later versions should be ok.

<https://www.ni.com/en-us/shop/labview/select-edition/labview-community-edition.html>

# LabVIEW LINX and Application Installation

This is the Pi setup for LINX guide for the power meter. Note: It is MUCH better to cable up ethernet; WiFi fails all the time during code downloads.

**1. Install support software on target Pi**

From LV2020+, Tools>>Hobbyist>>Target Configuration. Follow the NI directions to install the LINX support on the target Pi. Sometimes you need to try the whole process more than once.

May take a long time like 10 min to install LV on the target.

**If initial install fails,** it can be done manually. **This following procedure seems to give you the LV14 version, not a later one.** Then use the LV Tools>> Target Configuration to update it to the current version. <https://forums.ni.com/t5/Hobbyist-Toolkit/Raspberry-Pi-LINX-Installation-Failed/td-p/4154860> (Very tedious Linux command line procedure; not fun!).

**2. Other Support Software Required**

Using VI Package Manger, install the following:

* All of the OpenG utilities.
* LINX Raspberry Pi Addons (Author: MediaMongrels)
* A couple of subVIs in the ADS1015 driver must be replaced because a minor customization was required. In the Power Meter folder, there is a folder called LINX-ADS1015. Locate the folder of the same name in the following location, and replace it with the modified one. NOTE: You will have to change the destination folder to READ mode; it is read-only by default.

C:\Program Files (x86)\National Instruments\LabVIEW 2022\vi.lib\MediaMongrels\LINX Raspberry Pi Addons\LINX-ADS1015

**Connecting to the Pi**

Once the new Pi is basically configured as above, open the Power Meter Controller.lvproj project file. There will be a real-time target present called Power Meter Pi. Change the IP address and right-click on it to select Connect. After a pause, the connection indicator should turn green. At this point, any VI present under the target hierarchy can be run interactively. That includes the main Power Meter Controller VI.

**Set the LCD Baud Rate**

As delivered, the New Haven LCD is set for 9600 baud. 57,600 is required. To change baud rate, run the Change LCD Baud Rate Utility.vi. The LCD needs to be connected to the Pi. The resulting baud rate must be the same as the LCD Baud constant on the Power Meter Controller.vi diagram.

If you simply can’t change it, it’s also easy to change the LCD Baud constant on the Power Meter Controller.vi diagram to 9600. Display updates may be sluggish, however.

**Building and Running the Target Application**

A build specification is in the project under the target hierarchy.

1. Right-click on the Power Meter Controller build specification and select Build.
2. Right-click again and select Deploy.
3. Right-click again and select Run as Startup.
4. Reboot. This doesn’t always work from the dialog. You might need to use a VNC connection, or less desirably, cycle power.

From now on, the Pi will automatically run the controller program at power-on.

**Building and Running the GUI**

A build specification is in the project under the My Computer hierarchy.

1. Edit the Pwr Meter Settings.ini file and change the Controller IP to that of your target Pi.
2. Right-click on the Power Meter GUI build specification and select Build.
3. When it’s done, click Explore to see the resulting files.
4. Run the GUI (makes sense only if the target Pi is up and running the controller code.)

To deploy the GUI to another computer, copy the contents of the Build folder. It can be placed anywhere on the destination machine.

To run the executable GUI, the LabVIEW 2022 Q3 Runtime Engine (Windows, **32-bit**) is required on any computer that does not already have the LabVIEW development system installer.. Download and install from:

<https://www.ni.com/en-us/support/downloads/software-products/download.labview-runtime.html#460220>

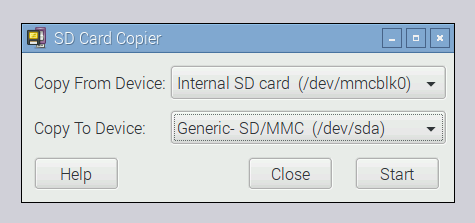
**Moving to a New Ethernet Domain**

If the entire system is moved to a new ethernet domain, you will need to find the IP address of the target Pi and edit that into the Pwr Meter Settngs.ini file. It would also have to be changed in the LabVIEW project before you can connect to the target.

If you left the Pi set as DHCP (the default), often you can try “Ping pwrpi” (or whatever hostname you gave it) and the return message will contain the IP. Otherwise, log into your router and see if you can find that hostname.

# Duplicating MicroSD Cards

Current version of Debian includes a card copying utility. From GUI, Accessories >> SD Card Copier. Plug your blank card into a USB card reader. In fact, get a tiny MicroSD adapter and leave the card in there permanently. Do this image copy and now you have a 100% cloned backup, ready for booting in case your primary boot drive gets corrupted. Also very useful when starting a new project.



**LabVIEW LINX Developer Notes**

Use regular VISA Serial, ASRL1::INSTR.

Regular TCP works.

Oddity: In interactive mode, if you modify a subVI then run the top level again, it does not download the modified code. Must force a recompile (control-Run Arrow at the top-level VI), then it will run normally.

Project utility >> restart doesn't always work. Reboot from a VNC session if it fails.

Super important regarding LV diagrams: There is one and only one LINX resource that you open. All VIs that access any kind of I/O must be in series, passing that resource along. Branching can result in race conditions, like having two parallel VIs accessing a single serial port. Test carefully!

A do-nothing loop with no timer will kill the GUI in interactive mode. Like regular RT.

Debian issue with TCP. If you connect to a port, then disconnect and close it, the OS will refuse any new connections to that port (error 60). This is with OR without an iptables setup to limit to a single connection to a port. That was done because otherwise more than one user can connect to the same port and then only one of them gets status messages. BUT… there seems to be a timeout on this, about 2 min, then you can connect again.

Solution:

1. On client side, do not use a fixed local port on TCP Open Connection.

2. On server side, set timeout on TCP Create Listener to 2000 ms (default is 25000).