

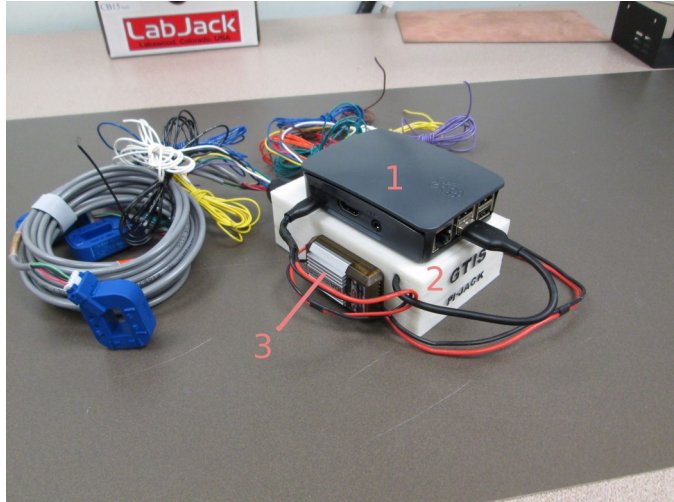
PiJack User's Manual

PiJack Overview

The PiJack data collection device built by SIL GTIS Power Systems in Waxhaw NC consists of 3 separate parts assembled together as a unit.

1. On the top is the Raspberry Pi in a case,
2. underneath is the LabJack U3 Data Acquisition Device (DAC) in a custom 3D printed case,
3. and on the side is a 60V DC-DC converter to provide 5V regulated power for the system.

The PiJack is designed for monitoring battery systems, and takes **DC only** readings and assumes the voltages and currents are not varying rapidly. Maximum voltage on any of the inputs is 60V. ***Exceeding 60V might damage the device!***



The standard configuration has 3 current inputs reading ± 150 amps using hall effect sensors, and 13 voltage inputs, 0-60V. A single voltage measurement is read once per minute from all 16 inputs and stored in a text file. At midnight each day, a new data file is started. A separate process compresses the data files and pushes them by ftp to the powermon.org web site where the graphs are generated by our server. You won't see today's data until tomorrow.

The PiJack expects either an ethernet connection or a wifi connection in order to push the data up to the powermon.org server. However, lacking an internet connection, it will accumulate data indefinitely. You can retrieve a copy of all the data files it has collected simply by inserting a FAT32 formatted USB flash drive into the Raspberry Pi. It will automatically create a folder named power on the USB drive and copy into it every data file that has not already been uploaded to the powermon server. This should take a few seconds up to a minute or two depending on how many days your pijack has been in operation. Just pull the drive out after the LED stops flashing or wait a minute or so if there is no LED.

Important:

Your Pi-Jack comes with a SD card installed, plus a second SD card which has a master copy of your configuration. **Keep the Master in a safe place and don't use it for anything else!** Some day the SD card inside the Pi will fail (it's gonna happen!) and you will need to take the Master and replicate it onto a new bootable SD card using the [Win32diskimager](#) program.

Installing your Pi-Jack

Your Pi-Jack should have been preconfigured by us before we shipped it to you, including your Wi-Fi settings if you gave them to us. This leaves only 3 steps to installing your working system:

1. Connecting the wires to the batteries you want to monitor
2. Power up the Pi-Jack
3. Use FileZilla to edit the ini file on our powermon.org server so your graphs will be labeled the way you want.

1 – Connecting the wires

The test wires are color-coded exit through two cable glands from the PiJack case, labeled E and F. The one black wire is the ground wire and should usually be connected to the negative terminal of your battery bank. Write down which color wires go where as you connect them up to various test points, including which is “E” and “F”. You will need this to setup the ini file and label your graphs. The current sensors are directional, but you can change the polarity in the .ini file, so don’t worry about which way is which when you install them.

The current sensors have a rectangular hole about 11mm x 19mm with a small corner cut out. The wire you’re measuring needs to fit through this hole. Cables over 1AWG may not fit, and large lugs won’t fit through. You may need to use a short copper busbar. 3/8” x 1/2” will fit nicely, and the 1/8” x 3/4” bus straps that come with the Outback 200RE batteries fit diagonally through the hole. A 1/4” x 3/4” bar might need just a little trimming with a file to make it fit.

For the voltage measurements you will need to come up with a way to attach the 24AWG wires to your test points. FYI: A standard sized computer screw will self-tap nicely in a 1/8” drilled hole in a copper busbar.

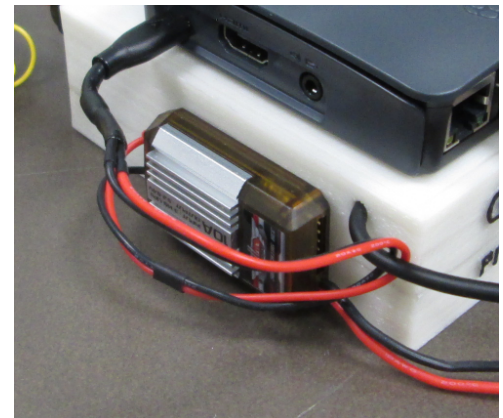


Current sensor and a voltage tap attached to a battery strap

2 – Power up the Pi-Jack

The power supply for the Pi-Jack is mounted on the side of the Pi-Jack. It can convert any DC voltage between 10V and 60V to the correct 5V the system needs. The output is the micro-USB jack which plugs into the raspberry-pi. The input is the long red and black wires with no connector on the end.

1. Unplug the micro-USB connector then hook up the red and black wires to the battery you will be monitoring. If you have several batteries in a 24V or 48V string, connect it to the top and bottom of the entire string, not just one of the 12V batteries.
2. Plug in the micro-USB connector and the system will boot and start collecting data.



3. To shut down, press the pushbutton sticking out of the side of the Raspberry-Pi case and wait 12 seconds for the system to do a clean shutdown before pulling the micro-USB plug to power it off.

How do I know it's working?

There are several ways to know.

1. Watch the blinking lights. It's collecting data once per minute and writing to the SD card, so the green LED on the Raspberry pi should blink a few times at least once each minute.
2. If you're patient you can wait until tomorrow and check the powermon.org website for a new data file and graphs.
3. Connect to the Pi over ssh or using a keyboard & DVI/HDMI monitor (see Appendix A) and issue the following commands:

```
cd /home/pi/power
ls -l
```

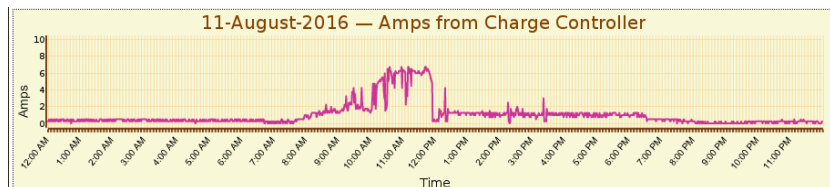
These commands will put you in the /home/pi/power folder and list all the files in that folder. You should see one starting with "La.xxx.xx.xx.csv" but having today's date where the x's are. You can watch the data being added to that file with the tail -f command like so:

```
tail -f La.2017.06.02.csv
```

(Substitute the correct filename) One line of data should be added each minute. Hit `ctrl-c` to exit the tail command.

3 – Use FileZilla to update your ini file

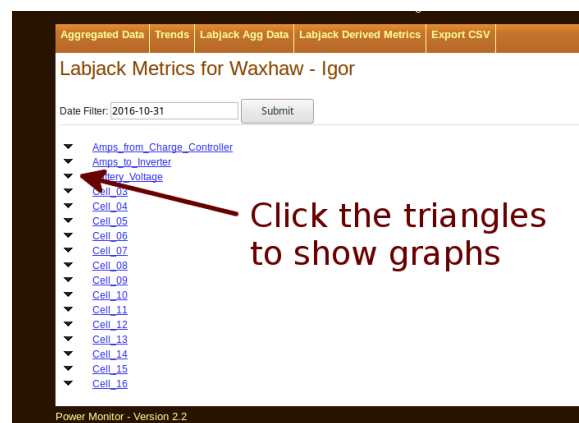
First things first – Understanding the Graphs



If your PiJack has already uploaded some data you should be able to view it immediately. Browse to <http://powermon.org/view/powermon/> and find your sitename. Click on the [labjack](#) link to see your graphs. (bookmark this page for easy access later.) You can also look at the data graphs for some of our other sites to get an idea how they ought to look.

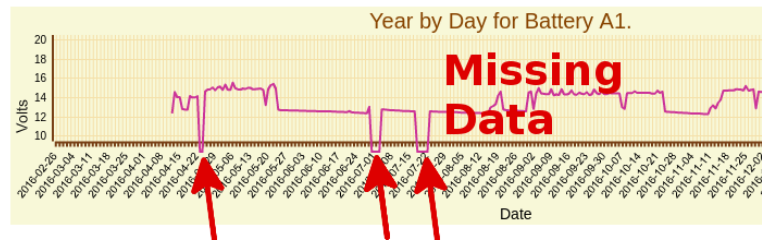
Note that the graph titles at first will simply be the color of the wires. You need to edit the ini file to give the graphs more meaningful titles. Instructions are a [few paragraphs down](#).

The default view lists all the data graphs from the most recent data file on the server, probably yesterday. However, it doesn't draw any of the graphs in order to save bandwidth. Display the graphs you are interested in by clicking on the adjacent small black triangle to show or hide a particular graph. Your web browser will remember



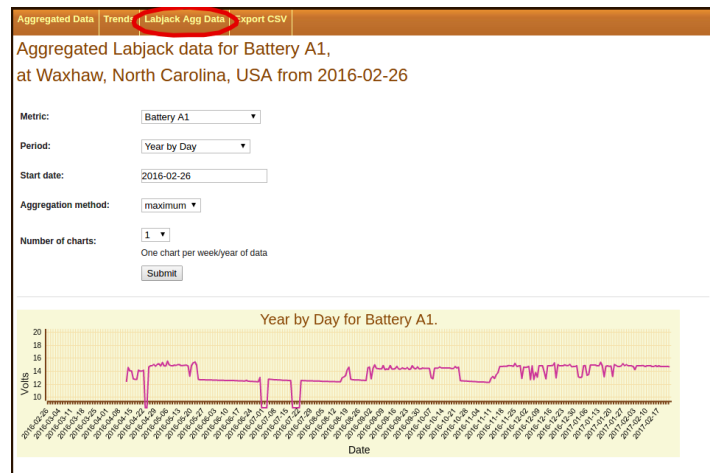
which graphs you had open next time you visit the powermon.org site. The “Date Filter” at the top of the page allows you to see the graphs from any particular day you wish. Choose a date from the drop-down calendar, then hit the submit button.

Clicking on any of the graph names will display that particular graph, going back in time for 7 days. Again, there is a “Date Filter” at the top of the page so you can choose to see 7 days worth of data from any past date you wish. Note that if the plotted data ever drops below the bottom of the Y-axis, that indicates mission data points for that time period.



Weekly & Yearly graphs

In addition to the daily graphs, the powermon website can draw you graphs for a week at a time or year at a time. These long term graphs can be very useful for seeing how your batteries are holding up. Click the Link “Labjack Agg Data” in the orange bar at the top of the page. You can choose to view a week at a time or a year at a time of data. The year-by-day charts are most useful. You can choose to “aggregate” a day’s worth of data in different ways. Choose



“maximum” if you want to see the highest voltage or current point for each day. (it plots only one point per day.) This is good for seeing how often your batteries reach a full charge. Choose minimum if you want to see how far run down the batteries are getting each day at different times of the year. Average might be useful if you’re looking at current; it will average all the data points for each day together.

Setting up the ini file for your PiJack device on powermon.org

The .ini file controls how the data from your PiJack will be plotted on the powermon.org website using our Django based graphing system. You need to label each chart so it makes sense to you.

Use [FileZilla](#) or a similar ftp program to login to <ftp://ftp.powermon.org> using your ftp account name. Your ftp name will be something like ecglisiro@powermon.org. If you loose your ftp name or password, email power_systems@sil.org for help.

Inside your home folder should be any files your labjack has uploaded, and a subfolder named “labjack” In there should be an ini file named something like La.2016.04.08.ini You need to edit this file to change the titles on the graphs.

Note that the ini file takes effect on the date embedded in the file name, and there can be multiple ini files. This allows you to change the physical configuration of your PiJack wiring at a later date and, by uploading a new ini file, display both the old and new data correctly.

.ini overview

Edit the ini file with a plain text editor. Be careful not to remove or add any commas to the file. Line 1 is ignored, and each of the following lines define a data graph.

- The first entry on each line (a string) is the cryptic name that the Labjack calls that data line. “EIO0” is the brown wire coming from port “E” on the PiJack box. “FIO7” is the White wire coming from Port “F” on the PiJack box.
- The second entry on each line (a number) is the gain to multiply the raw data by to get correct voltage, and
- the third entry (a number) is a negative offset, (explained below).
- The fourth entry on each line is the color of the wire, which isn’t used in the graphs, just for your reference so don’t change it.
- The fifth entry is the Y axis label on the graph, you shouldn’t modify that either. The sixth and seventh entries are not used, but
- the eighth entry is the title used on the graph, and you will want to modify that.

```
*La-template-.yyyy.mm.dd.ini (~/.Desktop) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Cut Copy Paste Find
*La-template-.yyyy.mm.dd.ini x
1 La Point,Gain,negOffset,Color of Wire,Y-Axis Label,String
  Letter,Position in said string (H=High L=Low; l=Voltage of
  battery closest to ground 4=Voltage of four batteries in
  string),Title,Nominal voltage of battery, Battery Type,Comment
2 EIO0,26.452,0,Brown,Volts,,E Brown wire,,Graph
3 EIO1,26.452,0,Red,Volts,,E Red wire,,,
4 EIO2,26.452,0,Orange,Volts,,,E Orange wire,,,
5 EIO3,26.452,0,Yellow,Volts,,,E Yellow wire,,,
6 EIO4,26.452,0,Green,Volts,,,E Green wire,,,
7 EIO5,26.452,0,Blue,Volts,,,E Blue wire,,,
8 EIO6,26.452,0,Purple,Volts,,,E Purple wire,,,
9 EIO7,26.452,0,White,Volts,,,E White wire,,,
10 FIO0,125,1.25,Brown,Amps,,,Brown Amps,,,
11 FIO1,125,1.25,Red,Amps,,,Red Amps,,,
12 FIO2,125,1.25,Orange,Amps,,,Orange Amps,,,
13 FIO3,26.452,0,Yellow,Volts,,,F Yellow wire,,,
14 FIO4,26.452,0,Green,Volts,,,F Green wire,,,
15 FIO5,26.452,0,Blue,Volts,,,F Blue wire,,,
16 FIO6,26.452,0,Purple,Volts,,,F Purple wire,,,
17 FIO7,26.452,0,White,Volts,,,F White wire,,,
18 Temp Sensor,1,0,,Spare,,,,,
.ini Tab Width: 4 Ln 17, Col 27 INS
```

Things to edit

1. The first thing to do is change the names that will appear on the graphs so that they are meaningful to you. In the default ini file, the graph titles are simply “E” and the color of the wire or “F” and the color of the wire. You’ll want to change this to whatever you call the battery you have attached it to, such as “Bank A Battery 1” **Caution: no dashes or special characters are allowed, just use letters, digits or spaces. Special characters crash the plots!**

The graphs on powermon.org will be sorted alphabetically based on this field, NOT in the order they appear in this file. By cleverly choosing what names to give your graphs, you can control which ones appear at the top & bottom.

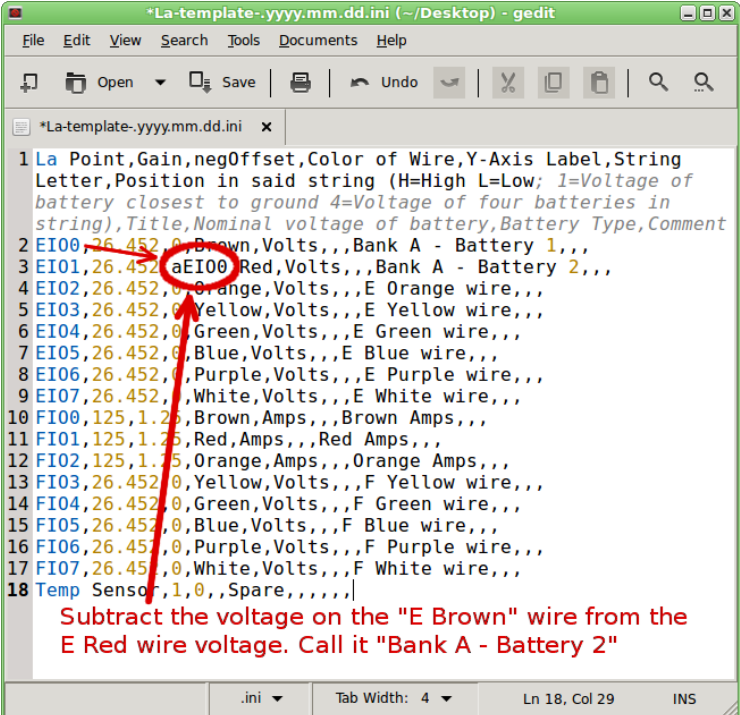
2. You can view the graphs with you web browser while editing the ini file with FileZilla. Save the .ini file using FileZilla and after saving it hit the refresh button on your web browser (or F5)

It's possible to have two Labjack devices connected to one Raspberry Pi in order to collect up to 32 different measurements. If you have two LabJack devices, the second one will have its own ini file beginning with Lb instead of La. Likewise the data files collected and uploaded by the second Labjack will start with Lb.

and the graphs will immediately be redrawn with the changes you made.

- For some of your measurements you will want to set an offset number. For example, if you have two 12V batteries in a series arrangement the first will measure 12.7V and the second will measure 25.4V; the combined voltage of the two. But you can display the individual battery voltage of the second battery by subtracting the voltage of the battery below it in the stack. The third number in each line is the offset, by default zero except for the current sensors on lines on FIO0 through 2. (The current sensors read 1.25V when there is zero current.) In the third entry on the line, replace the zero with a lower case "a" followed by the input you want to subtract. This is case sensitive! Examples of valid entries are:

aEIO0
aFIO3
aFIO7
aEIO5



```
*La-template-yyyymm.dd.ini (~/Desktop) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Cut Copy Paste Find
*La-template-yyyymm.dd.ini x
1 La Point,Gain,negOffset,Color of Wire,Y-Axis Label,String
  Letter,Position in said string (H=High L=Low; 1=Voltage of
  battery closest to ground 4=Voltage of four batteries in
  string),Title,Nominal voltage of battery,Battery Type,Comment
2 EI00,26.452,0,Brown,Volts,,,Bank A - Battery 1,,,
3 EI01,26.452,aEIO0,Red,Volts,,,Bank A - Battery 2,,,
4 EI02,26.452,0,Orange,Volts,,,E Orange wire,,,
5 EI03,26.452,0,Yellow,Volts,,,E Yellow wire,,,
6 EI04,26.452,0,Green,Volts,,,E Green wire,,,
7 EI05,26.452,0,Blue,Volts,,,E Blue wire,,,
8 EI06,26.452,0,Purple,Volts,,,E Purple wire,,,
9 EI07,26.452,0,White,Volts,,,E White wire,,,
10 FIO0,125,1.25,Brown,Amps,,,Brown Amps,,,
11 FIO1,125,1.25,Red,Amps,,,Red Amps,,,
12 FIO2,125,1.25,Orange,Amps,,,Orange Amps,,,
13 FIO3,26.452,0,Yellow,Volts,,,F Yellow wire,,,
14 FIO4,26.452,0,Green,Volts,,,F Green wire,,,
15 FIO5,26.452,0,Blue,Volts,,,F Blue wire,,,
16 FIO6,26.452,0,Purple,Volts,,,F Purple wire,,,
17 FIO7,26.452,0,White,Volts,,,F White wire,,,
18 Temp Sensor,1,0,,Spare,,,,,
Subtract the voltage on the "E Brown" wire from the
E Red wire voltage. Call it "Bank A - Battery 2"
.ini Tab Width: 4 Ln 18, Col 29 INS
```

- Possibly you might want to modify the gain and offset of the current sensors. (FIO0, FIO1 & FIO2)

The Hall effect current sensors we've chosen are designed to measure high currents over 100A and aren't very sensitive or accurate with current less than an amp. If you are using small wires, you can loop the wire through the donut hole in the sensor more than once, increasing the sensitivity and reducing the error. Divide 125 by the number of times the wire passes through the donut hole and replace the multiplier number in the .ini file lines 10, 11, or 12.

Example: for the brown current sensor I wrapped 3 loops around the sensor, with the wire passing through the center 4 times. The second number in the .ini file line #10 is changed from 125 to 31.25. (divide 125 by 4 = 31.25)

After you've collected some data you may notice the amp readings don't measure exactly zero at times when you know there was zero current flowing. Changing the offset for the amp sensors in the thousandths place (like 1.252 or 1.249) should get them accurately zeroed.

Appendix A – Gaining Console access to the Pi operating system

If everything is working correctly, you won't need this, but if your PiJack isn't collecting data properly, you might want to try troubleshooting from the raspberry pi console.

Direct Physical connection

The Raspberry Pi is running the Raspbian Operating System, a special version of Debian Linux. If you plug in a keyboard and monitor you can login to the Raspberry Pi command prompt.

The username is: pi

Password is unique to your PiJack and should have been given to you when you bought it. Contact power_systems@sil.org if you lost your password.

There is no GUI installed on the PiJack, it has just a linux command line terminal. (Remember DOS? Like that only different commands.)

SSH access over the network

You can not use the Pi username and password to connect over SSH, it is blocked for improved security since the pagekite program is opening a backdoor past your firewall into the PiJack. However, you can connect to the PiJack over SSH if you first on your own computer create an RSA Public key/private key pair and then copy the public key to the authorized_keys file on the PiJack.

You will need to download and install 3 free tools on MSWindows in order to create your encryption keys and install them on the PiJack. (Linux users can do all this with built-in tools, but I won't explain how here.)

Download and install the free 10-day trial Paragon Extfs program which allows you to mount and read the linux ext4 formatted SD card on your windows machine: <https://www.paragon-software.com/home/extfs-windows/>

Most windows editors will change the line endings of linux files to match the Windows style, totally messing things up on linux. However, you can edit a linux text file without screwing up the line endings using notepad++. Get the free notepad++ text editor here: <https://notepad-plus-plus.org/>

Download and install the full MSI windows installer package for PuTTY from here: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

After installing PuTTY, run PuTTYgen

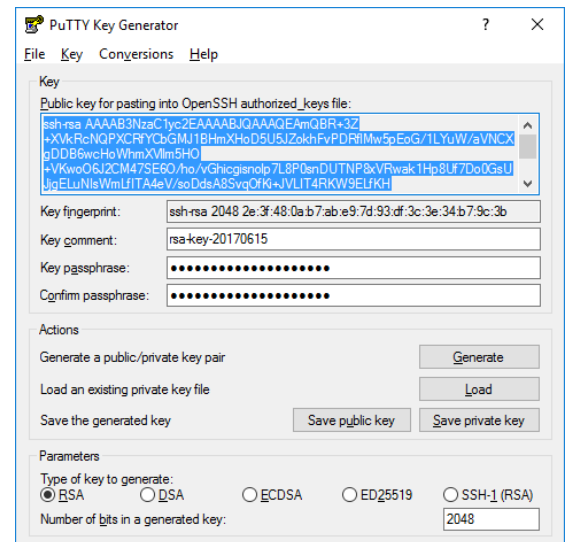
For the "Type of Key to generate" choose **RSA** and leaves the bits at **2048**. Click the **generate** button and wiggle the mouse around until it finishes.

I recommend you enter a passphrase in the "Key Passphrase" and "Confirm passphrase" boxes. If your private key every gets stolen, this will slow down the attackers from using it, they will have to crack this passphrase first, which takes time. The key can not be used without knowing the passphrase.

Click the "Save private key" button, name the key "pijack-echo.ppk" and save the key to a place on your hard disk only you have access to. You will need to configure putty to use this file when connecting to your pijack.

You need to copy the public key to the correct place on the PiJack's SD card. This is the tricky part. Shutdown the PiJack and remove the SD card. After installing the paragon Extfs program on windows, insert the SD card in your Windows computer. Windows will offer to "Scan and Fix" the drive, refuse. Windows should mount the 2 partitions on the SD card. Ignore the small one named "Boot" and use file explorer to browse the other, probably mounted as drive "F."

Browse to the folder /home/pi/.ssh/ and use notepad++ to open the file authorized_keys, it should have 4 lines in it already. You want to add a 5th line by copying from the PuTTYgen program into notepad++. There is a text box in PuTTYgen titled "Public key for pasting into OpenSSH authorized keys_file." Copy the entire contents (use scroll bar to make sure you get it all) of that text box into a single line in notepad++ then save and exit notepad++. Re-open notepad++ and the authorized_keys file to verify it actually wrote the extra line to the file. (Sometimes it doesn't!) Close and safely eject the SD card from your windows machine and put it back into the PiJack and boot.



Use Putty to connect by SSH to your PiJack

Now that you've done the hard part of installing your own key, it will be easy to connect to the PiJack from now on using PuTTY.

If possible set your DHCP server to always give the PiJack the same IP number. On the pi, you can find it's current IP number by typing the command ifconfig. It's network name should be raspberrypi, but sometimes this isn't visible to other computers on the network, so connecting by ip number is preferred.

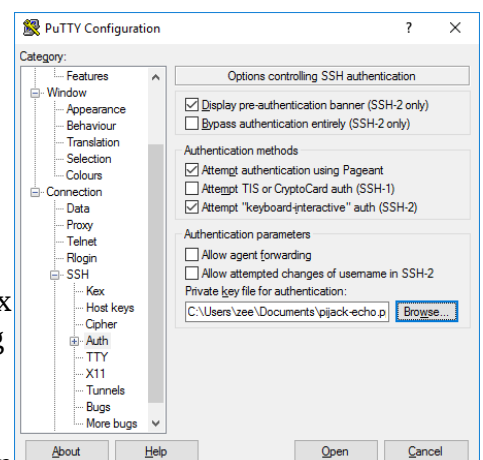
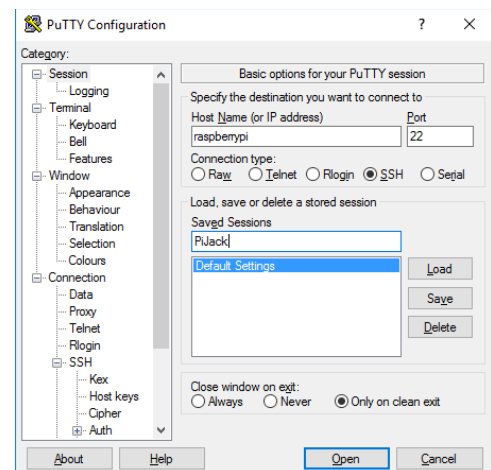
Start the PuTTY desktop app from the windows start menu.

Put the name raspberrypi or the IP number in the "host name" box and leave the port 22 and connection type SSH. In the "Category" tree structure on the left pane of the PuTTY window choose – Connection – SSH – Auth

In this window you can hit the browse button and find your private key which you previously saved to your hard disk. I named mine pijack-echo.ppk

Now go back to the tree and choose "session" at the top of the tree. From this home page you can enter a name in the saved sessions box (PiJack) and hit the save button. Next time you can recall the setting by choosing PiJack – Load then Open to launch a terminal session.

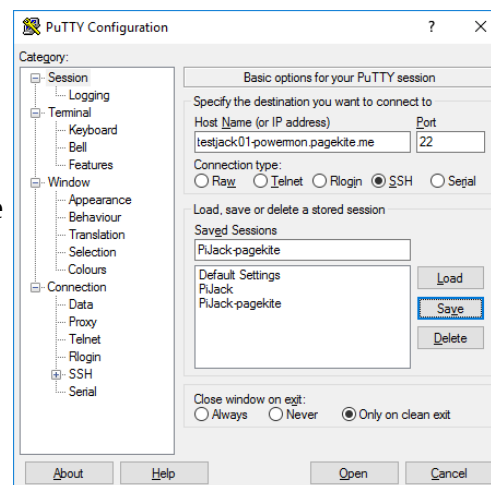
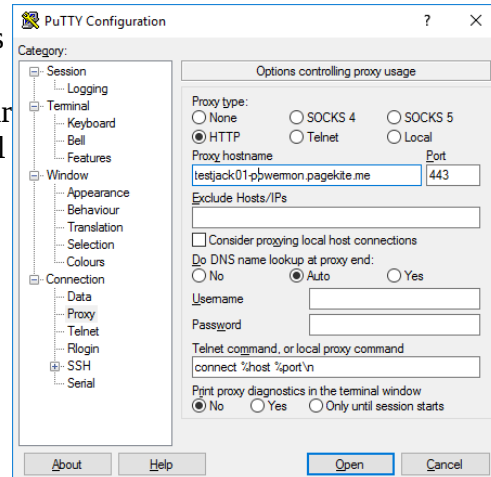
After this you should be able to "Load" the PiJack settings from the Saved Sessions list and hit the "Open" box to start a terminal session to the PiJack. The username is "pi" (lower case) and you will be prompted for the passphrase you created when you generated your private key. Try it.



SSH access worldwide using Pagekite

Your PiJack has a remote access feature that gives us access from JAARS to your PiJack even when it has a private IP number behind a NAT firewall. If you need to access to your PiJack from off site contact power_systems@sil.org and tell us what IP number or range you will be trying to access from. For enhanced security, the pagekite system is setup to only allow certain IP numbers in, and we need to add yours to the list. We will email you your pagekite host name used below.

Once we've done that for you you can reconfigure PuTTY to get access through the pagekite proxy. Open PuTTY and load the profile you use to access your PiJack from your network. You need to change the "host name" to your pagekite name – something like ecgIsiro-powermon.pagekite.me. In this example I'm using testjack01-powermon.pagekite.me. Then you need to go in the category tree to – Connection – Proxy and choose a proxy type of HTTP, fill in the proxy hostname with the same hostname, and set the port to 443. Then go back to the Category tree – Session and make up a new Saved Session name and hit the Save button. Try it out by hitting the Open button.



SSH Pagekite access from Linux

On linux you need to setup a proxy service using corkscrew. For all connections to pagekite.me the corkscrew program will proxy them over port 443. The private key, named "alpha" in this example, needs to be in your ~/.ssh/ folder. On ubuntu or debian based linux do this:

```
sudo apt-get install corkscrew
```

```
sudo nano ~/.ssh/config
```

add these lines at the bottom:

```
Host *.pagekite.me
  CheckHostIP no
  ProxyCommand corkscrew %h 443 %h %p
  IdentityFile ~/.ssh/alpha
  User pi
```

Now you should be able to connect as the user pi using the alpha key like so:
`ssh ecgIsiro-powermon.pagekite.me`

If your having trouble connecting, add -v to get hints on why it's failing.

Appendix B – changing the Wi-Fi and upload settings

WiFi settings

If your PiJack isn't connecting to your wifi network or you want to change the SSID and password the PiJack uses, you will need to do this.

The WiFi settings are in the file `/etc/wpa_supplicant/wpa_supplicant.conf`. Only the root user can edit the wifi settings file. If you remove the SD card from the Pi and mount it on another computer you can edit the file, or from a terminal on the running Pi do this:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
    set the correct ssid and passphrase
```

The system should automatically recognize any changes you made and reconnect to the access point within a minute or two. But if you're setting up wifi for the first time (e.g. you were previously using a wired ethernet connection) you will have to do this also:

```
sudo nano /etc/cron.d/powermon
    remove the "#" from the line about wifi_up.sh
```

The upload script

If you need to have the ftp uploads to powermon.org run at a specific time of day or simply attempt more often you will need to do this.

By default the upload program runs every 4 hours attempting to upload all .csv files in `/home/pi/power/` that have not already been uploaded. The way it tells which files have been uploaded is a simple hack; the script sets the executable "x" file permission on the files that have been uploaded. The already uploaded files will also have a .bz2 extension indicating they have been compressed. The script that does the uploading is `/usr/local/bin/ftppush.pl` and it is launched by the system cron daemon. To change the times when it is launched, from a terminal, edit the powermon file in `/etc/cron.d/` like so:

```
sudo nano /etc/cron.d/powermon
```

The second line launches the `ftppush.pl` script and you can change when it launches by modifying the first part of that line. The way those asterisks and numbers work makes sense only to a geek programmer, and is explained pretty well on wikipedia here: <https://en.wikipedia.org/wiki/Cron>

The default setting tries uploading 15 minutes after the hour, every 4 hours and looks like this:

```
15 */4 * * * pi /usr/local/bin/ftppush.pl
```

If you want it to try once every hour, at 10 minutes after the hour, change it to:

```
10 * * * * pi /usr/local/bin/ftppush.pl
```

Appendix C – Troubleshooting

After you've installed a public key on the Pi for your particular workstation PC (See appendix A), you will be able to ssh to the Pi or browse the filesystem using the sftp protocol and FileZilla.

Your data files are created and stored on the raspberry pi in the /home/pi/power/ folder.

The data files end with .csv, but just before they are uploaded to the powermon.org website they are compressed and a .bz2 is added to the filename. There is a program on the powermon.org server that runs hourly and looks for newly uploaded files and decompresses them, so that for every file on the Pi in the /home/pi/power/ folder with a .bz2 extension there should be a copy on the powermon.org server without the bz2 extension.

You can login by ftp to <ftp://ftp.powermon.org> and verify which files have been uploaded and decompressed.

Troubleshooting steps:

If you can't ssh to the Raspberry Pi, check the lights on it.

No LEDs lit – no power! Check the power supply.

A single solid red led means you're getting power but not running. Power cycle it to reboot.

Still no blinking green LED? – dead or corrupt SD card. Pull it out and check it.

Buy a new 16Gb+ SD card and use the MASTER SD card to create a new bootable SD card.

Verify your Pi is actually collecting data by browsing to the /home/pi/power/ folder and looking for a file with today's date. It should get a little bit larger every minute as data is added.

From a terminal you can view the file with the command:

```
tail /home/pi/power/La.yyyy.mm.dd.csv
```

(Ctrl-c to exit)

Verify the pi knows the correct time & date by typing **date**

If the Pi is having trouble talking to the labjack, it will create a file named errorlog.yyyy.mm.dd.csv

If the errorlog file exists, unplug and replug the full size USB cable to the labjack and wait a minute to see if it starts collecting data properly.

You can try running the data collection manually from the terminal and watch for error messages. Try /home/pi/LJFuse/monitorLJ.sh (Ctrl-c to exit)

If files are accumulating in the /home/pi/power folder, but not making it to the powermon.org server try manually running the upload program and watching for error messages. From a terminal type:

```
/usr/local/bin/ftppush.pl
```

It should login to powermon.org and upload any files that are waiting. Watch for error messages and contact power_systems@sil.org for help.