



RUGGEDPI (BETA2)

Updated 8 July 2014

POWER FLOW OVERVIEW

POWER (SOLAR, BATTERY, OR WALL PLUG (110/220) --->
ADAFRUIT SOLAR CHARGE CONTROLLER (ATTACHED TO POWER, BATTERY AND
LOAD; ATTACHED THERMISTOR SHUTS DOWN CHARGING OF BATTERY WHEN
TOO HOT) -->
VOLTAGE REGULATOR (UPCONVERTS INPUT VOLTAGE TO CONSTANT 5.35V FOR
PI) -->
ATXRASPI POWER CONTROLLER (BUTTON AND FULL SHUTDOWN) --->
RASPBERRY PI (LOADED WITH RACHEL/OTHER CONTENT)

HARDWARE COMPONENT LIST

CLICK HERE FOR THE LATEST ITEMIZED COST LIST:

<https://docs.google.com/spreadsheets/cc?key=0AiHH0xPhKBGadGd3NmFEjdjvSU85MIB2bTRKN21JZHc&usp=sharing>

TOOLS NEEDED:

- SCREWDRIVERS
- SOLDERING IRON
- "HELPING HANDS" FOR SOLDERING
- GLUE GUN
- DRILLS
- DRILL BITS: 1", 5/8" (16MM), 7/16"
- METAL FILES (TO SMOOTH LEXAN AND PLASTIC EDGES)
- MULTIMETER (THAT CAN READ DOWN TO HUNDRETHS)
- HEAT SHRINK WRAP
- HEAT GUN OR LIGHTER (FOR SHRINK WRAP)

RASPBERRY PI

(RASPBERRY PI) OVERVIEW

We use Version 2 of the Raspberry Pi.

Remember a few basic rules:

- Use a good quality USB cable if powering via USB
- Ensure you have at least 5V going to the Pi
- When using an high power USB, it is best to shunt power to it or you will be prone to USB cut outs. See the following picture for how to solder on a short wire to solve your problems.

(RASPBERRY PI) FIX USB POWER ISSUES

NOTE: This will not work on a Version 1. If the back of your Raspberry Pi doesn't look like the picture, please don't try this.

Solder on a wire to the back of your Raspberry Pi (Version 2) like the picture below:



SOLAR CHARGER

(SOLAR CHARGER) OVERVIEW

SOURCE: <http://www.adafruit.com/products/390>

DOCUMENTATION: <http://learn.adafruit.com/usb-dc-and-solar-lipoly-charger/using-the-charger>



1. Solder on capacitor (levels voltage when power from solar)
2. Solder on thermistor for over-heat protection
3. Solder on 1/8W 2.2K Ohm resistor to increase charge current to 1A

(SOLAR CHARGER) SOLDER ON CAPACITOR

PURPOSE: Stabilizes solar power (as sun changes position and voltage fluctuates)

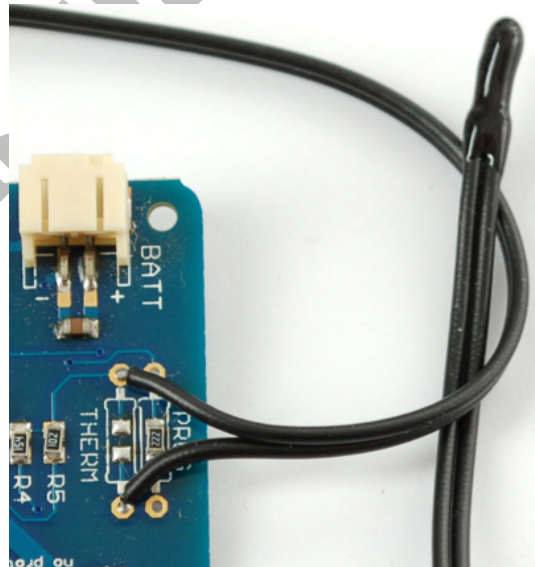
Nothing fancy, just match up the negative lead of the capacitor to the negative marking on the solar charger board. The legs of the capacitor can be shortened but I leave them long to allow it to be moved into an ideal position in the case.

NOTE: Do not lay the capacitor across and directly on the solar charger board as the chip may overheat due to the capacitor!

(SOLAR CHARGER) SOLDER ON THERMISTOR

PURPOSE: Prevents charging the battery when it is too hot and charging the battery will damage it.

Simply remove the 10K surface mount resistor from the **THERM** pads (or cut the trace going to it), and solder in a 10K NTC thermistor. Test out the system by trying to charge while you place the thermistor in a freezer or against some ice, as well as in a cup of > 50°C hot water. The charger should stop charging the battery. Once you are sure it is working, attach the sensing element (the epoxy bulb in this case) so it is resting against the battery. (from learn.adafruit.com)



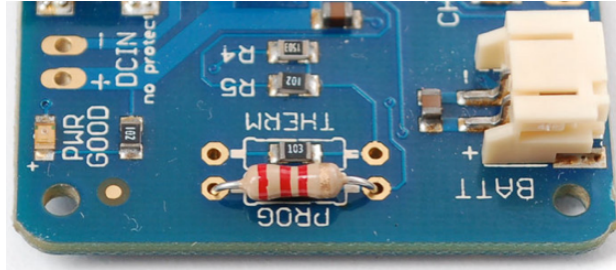
(SOLAR CHARGER) SOLDER ON RESISTOR

PURPOSE: Allows for increased charge current (1A)

Solder a 1/8w 2.2K Ohm resistor over top of the current 2K Ohm resistor.

The USB/Solar charger comes with a preset rate of 500mA which will work great for USB ports, USB wall adapters and solar panels up to 3 Watts. If you have a project that uses a larger panel, or perhaps some other sort of setup, you can easily adjust the current by soldering a resistor into the PROG pads.

SOLDER A 2.2K OHM RESISTOR OVER THE EXISTING RESISTOR "PROG"



WHY?

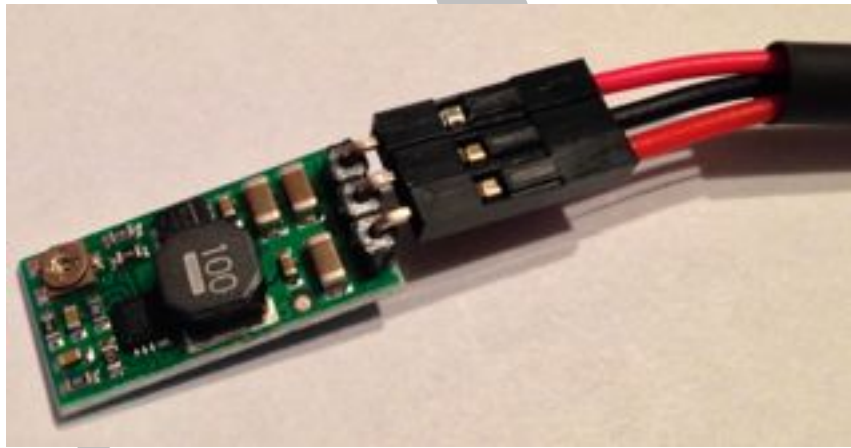
The current is set by $1000/R_{PROG}$ Amps, where R_{PROG} is the resistance. So for 2 Kohms, that would make it $1000/2000 = 0.5 \text{ A} = 500 \text{ mA}$. If you want 1A, you would use a 1K ohm resistor. If you want to increase the current, you need to *decrease the resistance*, **so you can just solder over the existing 2K. So for example, soldering another 2K resistor into R_{PROG} will give you 1K total resistance and 1000 mA current draw. See above for a 2.2K resistor soldered for about 950mA of max current draw.** If you want to set the max current draw lower, you'll need to remove the 2K resistor.

VOLTAGE REGULATOR

(VOLTAGE REGULATOR) OVERVIEW

PURPOSE: Increases voltage to 5.35V. Without it, when running on battery, the charger will pass through 3.7V to the Pi. This causes unstable performance.

I soldered the headers as shown but you can solder them different if you want. Far pin is voltage out, center pin is ground, and near pin is voltage in.



(VOLTAGE REGULATOR) ADJUST THE POWER OUTPUT

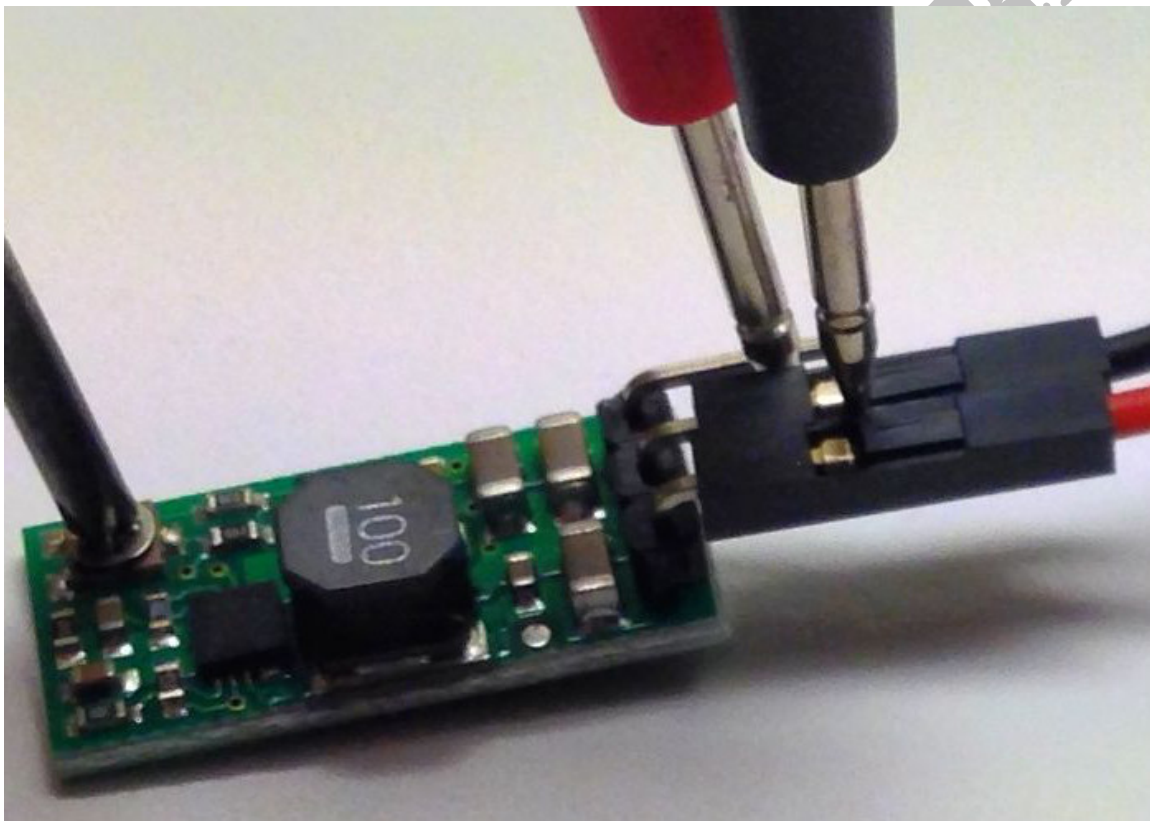
IMPORTANT: This is an extremely sensitive operation. The voltage regulator should be tuned by adding an input power of two AA batteries. This is to ensure you are starting at a very low input

power. The regulator CANNOT down regulate and will be damaged by higher input voltages than output.

I used two AA batteries to provide ~3V:



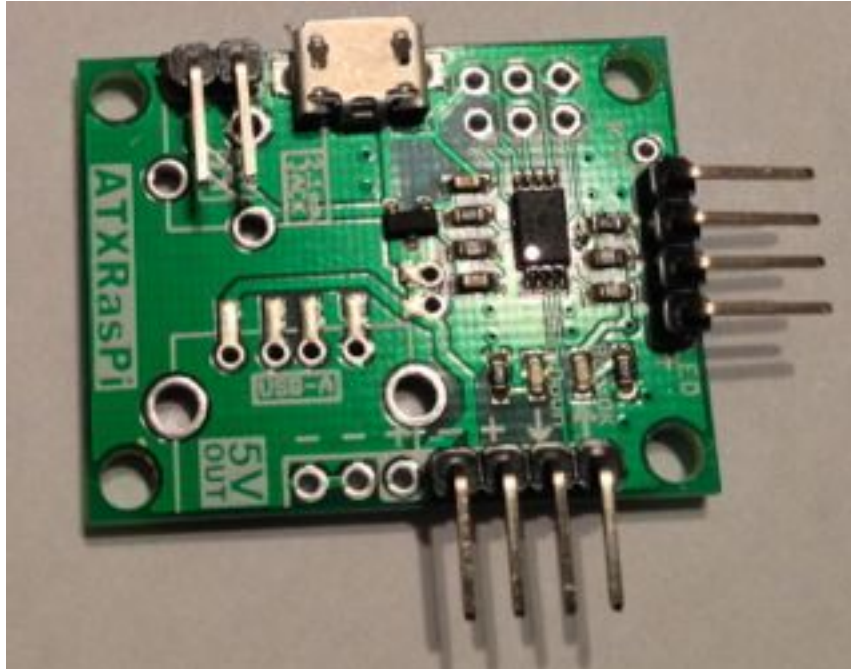
Closeup of multimeter lead placement while adjusting screwdriver:



ATXRASPI V2

(ATXRASPI V2) OVERVIEW

I soldered the headers as shown but you can solder them different if you want to create a different layout in the Pelican case. Right angle headers allow for a closer mounting profile.



PELICAN CASE EXTERNAL CONNECTIONS

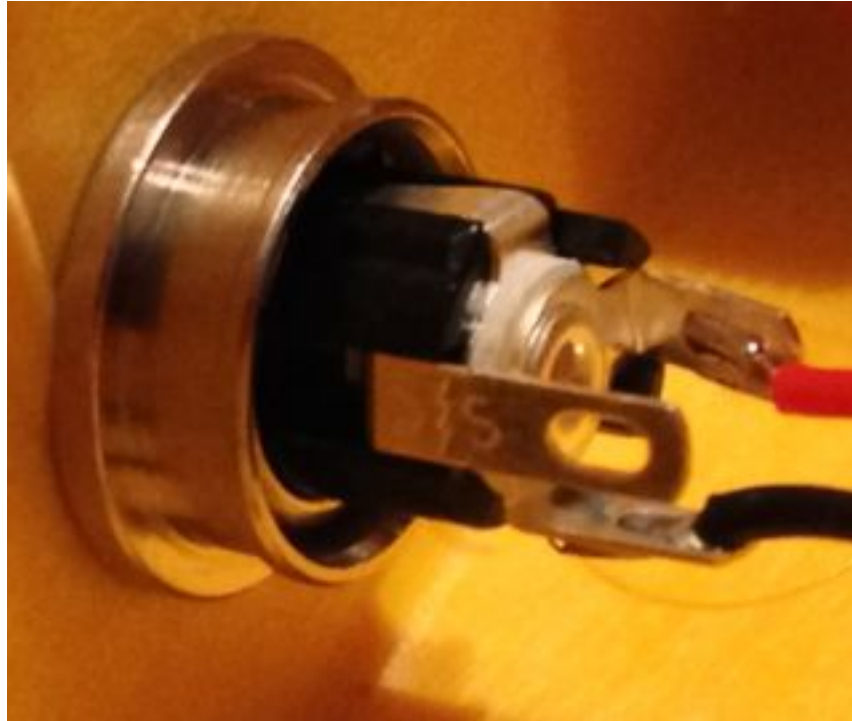
(PELICAN CASE) POWER JACK (WATERPROOF)

POWER JACK OVERVIEW

View of the assembly of the external power plug:



View shows internal view of power plug as it passes through Pelican case to the outside. (+) lead is attached to center and (-) lead is attached to outside (NOT the one marked C/S)



(PELICAN CASE) POWER SUPPLY PLUG (WATERPROOF)

CHANGE THE EXTERNAL POWER ADAPTER PLUG TO ONE THAT FITS

IMPORTANT: Add the black shield BEFORE you solder on the plug!

Soldered (+) lead to inside connector and (-) lead to outside:



Added hot glue to the connection to seal and add strength:

Hot glue the connection so that positive and negative don't short circuit.



Finished product.



(PELICAN CASE) POWER BUTTON (WATERPROOF)

POWER BUTTON ASSEMBLY

To SHUT DOWN: Briefly push the button.

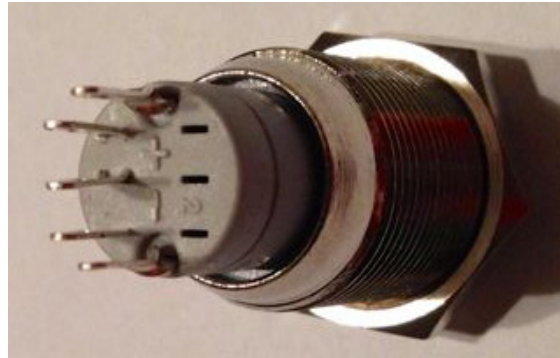
To SHUT DOWN: Push the button for around 4 seconds (until the light starts fading in/out). The light will go out when the Pi is shutdown. Push the button for ~10 seconds and it will hard shut down the Pi.

The light will be on when the Pelican is on and running. The light will be off when the Pelican is off. The light will fade in/out while shutting down.

The button has 5 solder points. The two outside provide power to the light that indicates power status (connect to the Status LED pins on the ATXRaspi). The center point is Common Ground. In 2nd picture below, the point left of center is Normally Open. The point right of center is Normally Closed (NOT USED).

The center and point left of center are used for momentary power during on/off (connect to the ATXRaspi Power Button connection pins).

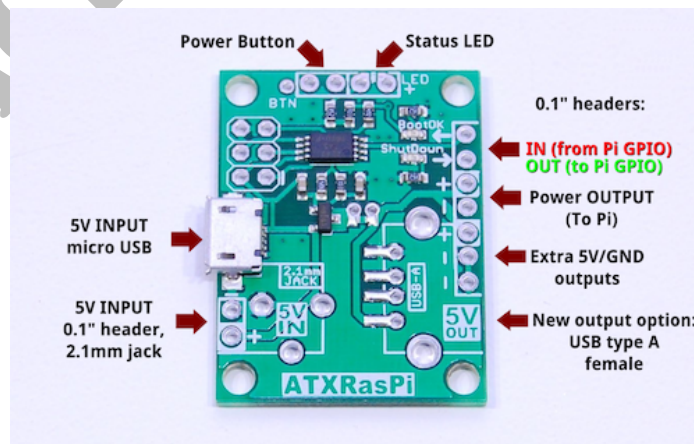
View of the assembly of the external power plug.



Soldered with the four wires to 4-wire quick disconnect (that goes to ATXRASPI).



ATXRaspi connection points.



Finished product running.



(PELICAN CASE) ETHERNET PASS-THRU (WATERPROOF)

ADD EXTERNAL ETHERNET PORT (WATERPROOF)

Use a 10-12" Ethernet patch cable to connect the inside to the Pi.



CABLING

(CABLING) POWER BUTTON TO ATXRASPI

PROVIDES ON/OFF AND LED POWER ON BUTTON



It is hard to see...here are some tips. In the example above and below, the blue/green wires provide momentary power for start/shutdown. They should be connected to the power button via the 4-wire quick disconnect cable. Solder one of the blue/green (doesn't matter which one) to the power button center pin (CG, Common Ground) and solder the other one to the left of center pin (NO, Normally Open). Close-up of wires laid flat to show how they are run.



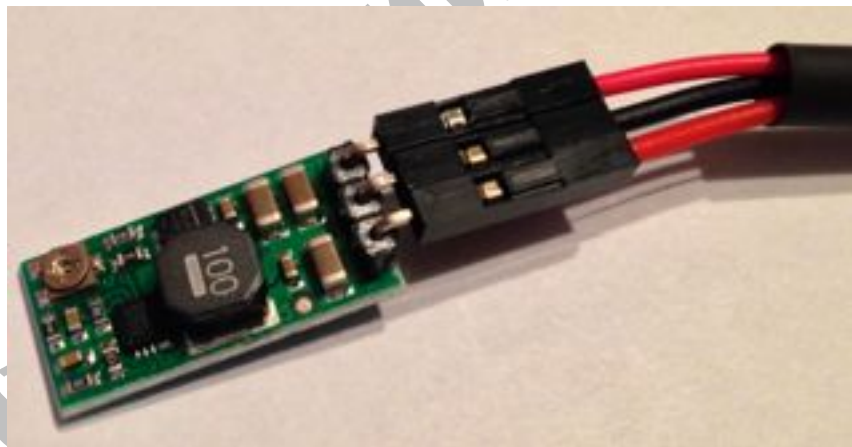
(CABLING) SOLAR CHARGER TO VOLTAGE REGULATOR TO ATXRASPI

3-WAY POWER CABLE

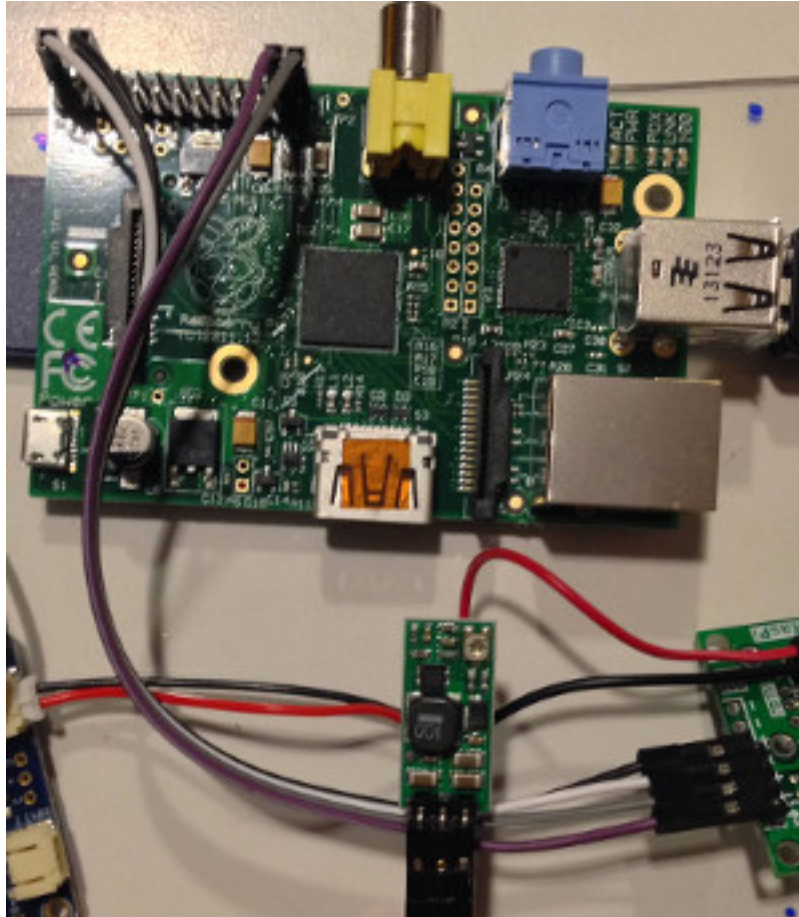


Probably the most difficult cable to create as it takes a little creativity. This cable takes power from the solar charge controller and passes it to the voltage regulator. Using the same ground but different output pin, the voltage regulator passes power to the ATXRaspi.

(ATXRASPI V2) CABLING ATTACHMENT TO REGULATOR (SEEN ON THE LEFT SIDE IN THE PICTURE ABOVE AND THEN BELOW ATTACHED TO REGULATOR)



(ATXRASPI V2) CABLING ATTACHMENT TO PI



The last cable requires no soldering. The four pins coming off the ATXRaspi feed the Raspberry Pi power and monitor shutdown). The black/white cables (in the example above) provide power while the purple/grey cables monitor the Pi for power state.

EASY SOFTWARE INSTALL

DOWNLOAD THE IMAGE AND COPY TO AN SD CARD

DOWNLOAD THE BIT TORRENT SYNC CLIENT

We use Bit Torrent Sync to transfer data. It is as fast as your internet connection and restarts where it left off if disconnected. It also will continually monitor the sync data for updates from the source. Download it from this site (it runs on almost all operating systems): <http://getsync.com/>

DOWNLOAD THE IMAGE – NORMAL AND RECOMMENDED METHOD

Use this appropriate “secret” below to download the image or images that suit your needs. You will need to ensure that you have enough space on your local folder to store the image being transferred.

SD cards are not created equal in size. I have seen 64GB SD cards that range in size from 62.5 to 67.1. It is all in the manufacturing process.

These are the folders that can be synced and their secrets:

- ENGLISH – 59GB - Full Image (easiest):
BWVIWEJY74CKTOALRI4H4NHEJZTDSULQT
- SPANISH – 59GB - Full Image (easiest):
B7NOX3DCZ5SPDENXNMACFNATDG27FN3BM

DOWNLOAD THE IMAGE – ALTERNATE METHOD

NOTE: If you are bandwidth impaired, you might benefit from this method of install. It uses Clonezilla BootCD and installs the image onto the SD card in about 1 hour. Instructions are included in the sync folder. It is also very beneficial if your SD card is smaller in size than 62GB.

- ENGLISH – 43.4GB - Image in two parts (safest for unknown card sizes but slightly more difficult):
BGBYEPKQ4IY4V3PNV4JFUPSXGOI3NQZDF
- SPANISH - __GB - Image in two parts (safest for unknown card sizes but slightly more difficult):
BGJO54LZNXRJEOPDBUPC72NY5BQPHTADM

If you image using this method, run the following command from the command line.

```
cd /root
wget https://raw.githubusercontent.com/dweeber/rpiwiggle/master/rpi-wiggle
chmod +x rpi-wiggle
sudo ./rpi-wiggle
```

When the script finishes, (this part is very quick)... Press Enter to have the system reboot.
When the system reboots, you can watch the screen. There will come a point where the following text will show:

Starting resize2fs_once, THIS WILL TAKE A FEW MINUTES

DO NOT INTERRUPT THE PROCESS... just let it run.

This will take a while depending on the size of the SDcard you are expanding to. A 16GB card was taking a good 7+ minutes. A 32GB hard will take significantly more.

INSTALL THE IMAGE TO YOUR SD CARD

For img files, use multiple methods listed here: http://elinux.org/RPi_Easy_SD_Card_Setup

If specific instructions are needed, they are included in the transfer folders.

**** IMPORTANT ** SETUP KA-LITE**

Decide how you will operate KA-Lite

If you wish to track multiple KA-Lite Servers, then please follow these instructions:

1. Single Server, One Site, No Internet
 - a. Run the following installation script
2. Single Server, One Site, With Internet
 - a. Run the following installation script
 - b. After install, you will need to login to the website <http://192.168.10.1:8008> as the admin user (the one you identify during the install script below)
 - c. You will setup up on online KA-Lite Central Server account at: <https://kalite.learningequality.org/accounts/login/>
 - d. Once you finish the installation instructions, your KA-Lite install will automatically check in to the Central Server and report statistics
3. Multi-Server, One Site, With Internet
 - a. Same as Single Server, One Site, With Internet (above)
 - b. **** Highly important **** To ensure that you know which server is which, you need to change the name of the server to something meaningful (DURING the install), since it will report using that name. The installation script will ask you to name the server.
 - c. On the Central Server website, you can set up different facilities for the KA-Lite installs to report under.
 - d. Additionally, if you have multiple KA-Lite installs under the same sharing network (as seen on the Central Server website), students can login with their same credentials onto any of the installs on the same sharing network and see their personalized information.

Run the installation script

Inside the ka-lite directory (that you cloned above) you should find a script called `setup_linux.sh`

Run this script as pi (NOT as root)

```
sudo pi  
/home/pi/ka-lite/setup_linux.sh
```

(1) This will create a new public/private key pair for syncing

(2) Allows the admin to change their admin user/password

(3) Probably most important, sets a unique name for the install so you can ID it properly when it syncs to the online central server at <http://kalite.adhocsync.com/accounts/login/> (i.e. you know what Pi is calling home and from where, if named it appropriately)

NOTE: Highly recommend that you select YES for “Optimize server”

The server will be started and scheduled to run at boot.

RUN THE SERVER

(If you're installing the server to test/develop, rather than deploy, follow the [development instructions](#) instead.)

```
sudo service kalite stop
sudo service kalite start
```

If the automatic background option was not chosen, start the server by running `sudo ./start.sh` in the ka-lite directory.

CHECK THAT KA LITE IS NOW ACCESSIBLE

The local KA Lite website should now be accessible at <http://192.168.10.1:8008>

To access it from another computer connected to the same network, replace 192.168.10.1 with your computer's external IP address or domain.

UPDATE THE INDEX.HTML TO SHOW KA-LITE CONTENT

See the last section called, "UPDATE INDEX.HTML FILES FOR ADDED CONTENT"

(OPTIONAL) ADD SUBTITLES AND CHECK FOR VIDEO UPDATES

Login to the server the web interface at <http://192.168.10.1:8008/securesync/login/>

User: root

Password: rachel

From this screen, you can:

- Add students
- Add teachers
- Update content
- Update language packs for closed captioning
- Update the server time/date
- Manage users (move them between groups)

(OPTIONAL) CHECK FOR UPDATES TO KA-LITE

Login to the server via ssh and run:

```
python /home/pi/ka-lite/kalite/manage.py update
```

CHANGE THE DEFAULT PASSWORD FOR ROOT AND PI

Change the default password from “rachel” to something more secure.

While ssh'd into the Pi, execute the following command and follow instructions:

```
passwd pi
```

and

```
sudo passwd root
```

FINALLY...ALMOST DONE...

Follow any OPTIONAL guidance towards the end of this document.

MANUAL (AKA “THE HARD WAY”) SOFTWARE INSTALL

INSTALL RACHEL (OS + RACHEL INITIAL CONTENT)

NOTE: THE FOLLOWING COMMANDS WERE RUN FROM OSX

<http://pi.worldpossible.org/howto.html>

DOWNLOAD RACHEL

Download the file from the FTP source (instructions at link above)

CAUTION: Use FILEZILLA to download or you may have issues with corrupt downloads.

CHECK MD5

Find the md5 of the downloaded file

```
md5 rachel_pi_wsearch_ap_30gb.img.zip
```

and compare against the sum (EXAMPLE)

```
cat rachel_pi_wsearch_ap_30gb.img.zip.md5  
f8f04ca97f27a8246da7fa80576a8931  rachel_pi_wsearch_ap_30gb.img.zip
```

INSTALL IMAGE ONTO SD CARD (INSTRUCTIONS BELOW FOR OSX)

Unzip archive using your favorite archive tool (7z, The Unarchiver, Winzip, etc)

Find the device name of your SD card

```
sudo diskutil list
```

Copy image to SD card. In the following example, the drive was listed as `/dev/disk1`. However, for `dd`, you need to use the raw device so use `/dev/rdisk1` below.

```
sudo dd if=/<change to pi image location>\  
/rachel_pi_wsearch_ap_30gb.img of=/dev/rdisk1 bs=1m
```

UPDATE THE RASPBERRY PI OPERATING SYSTEM

BOOT AND CONNECT

START YOUR PI with your SD card installed and internet plugged into the Ethernet port

Connect to the RPI access point

Connect with SSH to the Pi:

```
ssh pi@192.168.10.1  
      (password:  rachel)
```

UPDATE YOUR SOURCE PACKAGES AND UPGRADE THE PI

```
sudo apt-get update && sudo apt-get upgrade  
sudo rpi-update
```

You might have to reboot if there are updates after `rpi-update` executes.

(OPTIONAL) YOU MAY NEED TO ESCALATE TO ROOT PRIVS FOR SOME CMDS

```
sudo su
```

ADDING THE ATXRASPI V2 POWER CONTROLLER

<http://lowpowerlab.com/atxraspi/>

(NOT REQ FOR RACHEL PI BUILD) UPDATE PACKAGES AND INSTALL GIT

```
sudo apt-get install git-core
```

(NOT REQ FOR RACHEL PI BUILD) CHECK THAT PYTHON IS INSTALLED

```
python -V
```

It should be 2.6 or 2.7...if not, use the following to install:

```
sudo apt-get install python
```

INSTALL WIRINGPI

Download WiringPi from github and build it:

```
cd /home/pi/  
git clone git://git.drogon.net/wiringPi  
cd wiringPi  
./build
```

**COPY THE SHUTDOWNCHECK BASH SCRIPT TO YOUR HOME DIRECTORY - /HOME/PI/
AND ADD THE PROPER EXECUTE RIGHTS:**

```
cd ~  
wget https://raw.githubusercontent.com/LowPowerLab/ATX-Raspi/master/shutdowncheck  
sudo chmod 755 shutdowncheck
```

SETUP SCRIPT TO RUN AT BOOT

```
sudo nano /etc/rc.local
```

Add the following line marked in red – the lines in black are already part of the /etc/rc.local file:

```
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
# ADD THIS LINE – it will start WiringPi on boot
(cd /home/pi && exec ./shutdowncheck) &
```

NOTE: Be sure to add the line BEFORE the line “exit 0”

ADDING KA-LITE

<http://kalitewiki.learningequality.org/installation/linux-installation>

INSTALL REQUIRED SOFTWARE

Install Python

Almost all popular versions of Linux come with Python already installed. To ensure that it is a usable version, run `python -V` from the command line, and ensure that the version number starts with 2.6, or 2.7.

If Python is not installed, install it by running `sudo apt-get install python` or the equivalent command in your distribution's package manager.

Install M2Crypto and psutil

You can install this package to make KA Lite run faster (especially on devices with low-powered CPUs), by running:

```
sudo apt-get install python-m2crypto python-psutil
```

DOWNLOAD KA LITE

Clone the repository into a directory of your choice. Use `cd` to navigate into the target directory, and then run the command below (the files will be put into a subdirectory of your current directory named `ka-lite`):

```
cd /home/pi/  
git clone https://github.com/learningequality/ka-lite.git
```

OPTIMIZE KA LITE

KA-Lite should be optimized to run on the Raspberry Pi. Run the following scripts one line at a time

```
sudo nano /home/pi/ka-lite/kalite/local_settings.py  
  
sudo echo 'CONFIG_PACKAGE = "RPi"' > /home/pi/ka-lite/kalite/local_settings.py  
  
sudo echo 'CONTENT_ROOT="/home/pi/khan-content/"' >> /home/pi/ka-lite/kalite/local_settings.py  
  
sudo apt-get install nginx  
  
sudo rm /etc/nginx/sites-enabled/default  
  
sudo touch /etc/nginx/sites-enabled/kalite  
  
sudo sh -c '/home/pi/ka-lite/kalite/manage.py nginxconfig > /etc/nginx/sites-enabled/kalite'  
  
sudo rm /etc/nginx/nginx.conf  
  
sudo touch /etc/nginx/nginx.conf  
  
sudo echo "user www-data;" >> /etc/nginx/nginx.conf  
  
sudo su  
# for whatever reason, the cat cmd will not work  
# with sudo to build the following file  
  
# copy/paste the entire following section from the  
# next line (starting with "cat") and ending with "EOF"  
  
cat > /etc/nginx/nginx.conf << EOF  
user www-data;  
pid /var/run/nginx.pid;  
  
###
```

```
# we have 1 cpu so only need 1 worker process
worker_processes 1;

events {
    ###
    # good overall speed on RPi with this setting
    worker_connections 1536;

    ###
    # Activate the optimised polling for linux
    use epoll;

    ###
    # Keep multi_accept off - RPi+KA Lite is slowed if "on"
    multi_accept off;
}

http {
    ###
    # RPi+KA Lite is faster with sendfile "off"
    sendfile off;
    tcp_nopush off;

    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ###
    # Speed up landing page by caching open file descriptors
    open_file_cache max=2048;

    ##
    # Logging Settings
    # don't log, we don't need to know normally
    access_log off;
    error_log off;
```

```
##
# Gzip Settings
gzip on;
gzip_disable "msie6";

##
# Virtual Host Configs
include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}
EOF

exit
```

CONSOLIDATE KA-LITE VIDEOS TO ONE LOCATION

Move all Khan videos to one folder that can be referenced by different programs and copy updated content to KA-Lite.

(These two scripts, `khanlink2.py` and `khanlink3.py` were written by Manny Ackerman, Developer for RACHEL, World Possible)

This is designed to run on a RPi on which Rachel and KA-Lite have been loaded (but not the KA-Lite videos--that is the next step). It is assumed that ka-lite is loaded at `/home/pi/ka-lite`. If you loaded it somewhere else, then search for the location in the program and change it. (It only appears once in the program.) However, do not put the ka-lite directory anywhere beneath the `www` directory. Django does not forbid that, but the documentation repeatedly says that this would be a potential cause of security problems. The program also assumes that you have the KAOS portion of RACHEL loaded at `/var/www/ka`. Again, if it is in a different place, fix the program, the directory only appears once.

This program moves all of the KAOS videos to the KA-Lite video directory, and replaces them with links in the KAOS video directories. There is some redundancy between the various KAOS directories, and this will make it so that each video is only stored once. (Collisions are handled intelligently.)

This program prints each file name as it is processed, and runs for quite a while. (Processing many gigabytes of data on an RPi takes a while.)

NOTE: To execute, copy/paste the entire script at one time (from “`sudo...`” to “`EOF`”)

1. `khanlink2.py`


```

sudo cat > /home/pi/khanlink2.py << EOF
import os

def dirs(MyDir):
    if os.path.isdir(MyDir):
        for f in os.listdir(MyDir):
            kaosname = os.path.join(MyDir,f)
            if os.path.isdir(kaosname):
                dirs(kaosname)
            else:
                if os.path.isfile(kaosname):
                    ext = os.path.splitext(f)[1]
                    if ext == ".mp4":
                        print kaosname
                        kalname = os.path.join("/home/pi/ka-lite/content",f)
                        if os.path.exists(kalname):
                            if os.path.islink(kalname):
                                os.unlink(kalname)
                                os.rename(kaosname,kalname)
                                os.chmod(kalname, 0755)
                            elif os.path.isfile(kalname):
                                os.unlink(kaosname)
                        else:
                            os.rename(kaosname,kalname)
                            os.symlink(kalname,kaosname)
                            os.chmod(kaosname, 0755)
                        print kalname
    return

if __name__ == "__main__":
    import sys
    if len(sys.argv) > 1:
        dirs(sys.argv[1])
    else:
        dirs("/var/www/ka")
EOF

```

Execute the script by running:

```
sudo python ./khanlink2.py
```

2. khanlink3.py

This program is designed to run after khanlink2.py. It is again assumed that ka-lite is loaded at /home/pi/ka-lite and If you loaded it somewhere else, then search for my location in the program and change it. (It only appears once in the program.) It is also assume that you have downloaded ka-lite video torrent ("ka-lite-content-resized"), and that the contents of the torrent are on a network attached drive which is mounted on the file system. Manny mounted it at /home/pi/nas (it was not at the root of my nas), and the full path to the video files was /home/pi/nas/RPi/ka-lite-content-resized/content. If you mounted it somewhere else, you can change the program but you don't have to – Manny set it up so it can be entered as a parameter.

This program looks at all the video files in the torrent copies any that are not already present in the KA-Lite video directory on the RPi.

If you have the KA-Lite video torrent at /home/pi/nas/RPi/ka-lite-content-resized/content, you may run this program at a command prompt:

```
sudo python ./khanlink3.py
```

If you have the KA-Lite video torrent somewhere else, then give the full path of the torrent video directory as a parameter:

```
sudo python ./khanlink3.py <video path>
```

This program also prints each file name as it is processed, and runs for hours.

Decide how you will operate KA-Lite

If you wish to track multiple KA-Lite Servers, then please follow these instructions:

1. Single Server, One Site, No Internet
 - a. Run the following installation script
2. Single Server, One Site, With Internet
 - a. Run the following installation script
 - b. After install, you will need to login to the website <http://192.168.10.1:8008> as the admin user (the one you identify during the install script below)

- c. You will setup up on online KA-Lite Central Server account at:
<https://kalite.learningequality.org/accounts/login/>
 - d. Once you finish the installation instructions, your KA-Lite install will automatically check in to the Central Server and report statistics
- 3. Multi-Server, One Site, With Internet
 - a. Same as Singe Server, One Site, With Internet (above)
 - b. **** Highly important **** To ensure that you know which server is which, you need to change the name of the server to something meaningful (DURING the install), since it will report using that name. The installation script will ask you to name the server.
 - c. On the Central Server website, you can set up different facilities for the KA-Lite installs to report under.
 - d. Additionally, if you have multiple KA-Lite installs under the same sharing network (as seen on the Central Server website), students can login with their same credentials onto any of the installs on the same sharing network and see their personalized information.

Run the installation script

Inside the ka-lite directory (that you cloned above) you should find a script called setup_linux.sh

Run this script as pi (NOT as root)

```
sudo pi  
/home/pi/ka-lite/setup_linux.sh
```

(1) This will create a new public/private key pair for syncing

(2) Allows the admin to change their admin user/password

(3) Probably most important, sets a unique name for the install so you can ID it properly when it syncs to the online central server at <http://kalite.adhocsync.com/accounts/login/> (i.e. you know what Pi is calling home and from where, if named it appropriately)

NOTE: Highly recommend that you select YES for “Optimize server”

The server will be started and scheduled to run at boot.

RUN THE SERVER

(If you're installing the server to test/develop, rather than deploy, follow the [development instructions](#) instead.)

```
sudo service kalite stop  
sudo service kalite start
```

If the automatic background option was not chosen, start the server by running `sudo ./start.sh` in the ka-lite directory.

CHECK THAT KA LITE IS NOW ACCESSIBLE

The local KA Lite website should now be accessible at <http://192.168.10.1:8008>

To access it from another computer connected to the same network, replace 192.168.10.1 with your computer's external IP address or domain.

UPDATE THE INDEX.HTML TO SHOW KA-LITE CONTENT

See the last section called, "UPDATE INDEX.HTML FILES FOR ADDED CONTENT"

(OPTIONAL) ADD SUBTITLES AND CHECK FOR VIDEO UPDATES

Login to the server the web interface at <http://192.168.10.1:8008/seuresync/login/>

User: root

Password: rachel

From this screen, you can:

- Add students
- Add teachers
- Update content
- Update language packs for closed captioning
- Update the server time/date
- Manage users (move them between groups)

(OPTIONAL) CHECK FOR UPDATES TO KA-LITE

Login to the server via ssh and run:

```
python /home/pi/ka-lite/kalite/manage.py update
```

CHANGE THE DEFAULT PASSWORD FOR ROOT AND PI

Change the default password from "rachel" to something more secure.

While ssh'd into the Pi, execute the following command and follow instructions:

```
passwd pi
```

and

```
sudo passwd root
```

(OPTIONAL) CHANGE THE LOCALE AND TIME

SETTING LOCALE

```
sudo dpkg-reconfigure locales
```

SETTING TIME

```
sudo date -s "Jul 5 08:10" <-- change this to your current local time
sudo dpkg-reconfigure tzdata
```

(OPTIONAL) BACKING UP AND RESTORING THE SD IMAGE

BACKUP (EXAMPLE FOR OSX)

List the drives, identify your SD card, start the backup image

```
sudo diskutil list <-- my drive was /dev/disk1; use /dev/rdisk1 below
sudo dd if=/dev/rdisk1 of=/Users/sd/Desktop/RPI-RACHEL-SD-64GB-5Jan2014.dd bs=1m
```

RESTORE

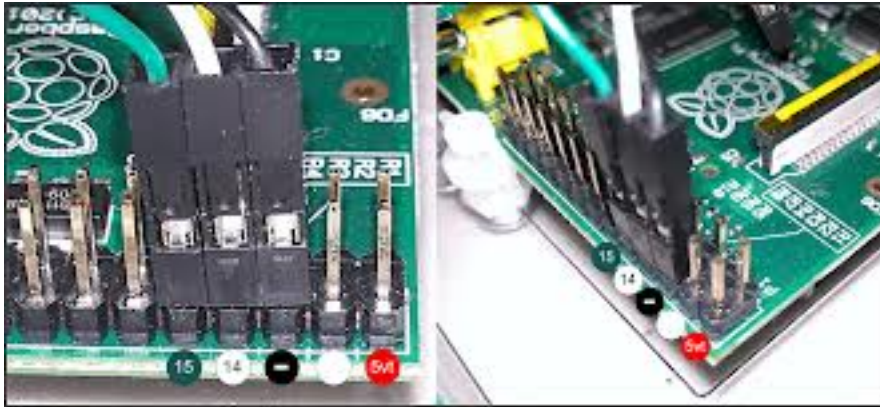
```
sudo diskutil list <-- my drive was /dev/disk1; use /dev/rdisk1 below
sudo dd of=/dev/rdisk1 if=/Users/sd/Desktop/RPI-RACHEL-SD-64GB-5Jan2014.dd bs=1m
```

(OPTIONAL) CONNECT TO PI USING A USB CONSOLE CABLE

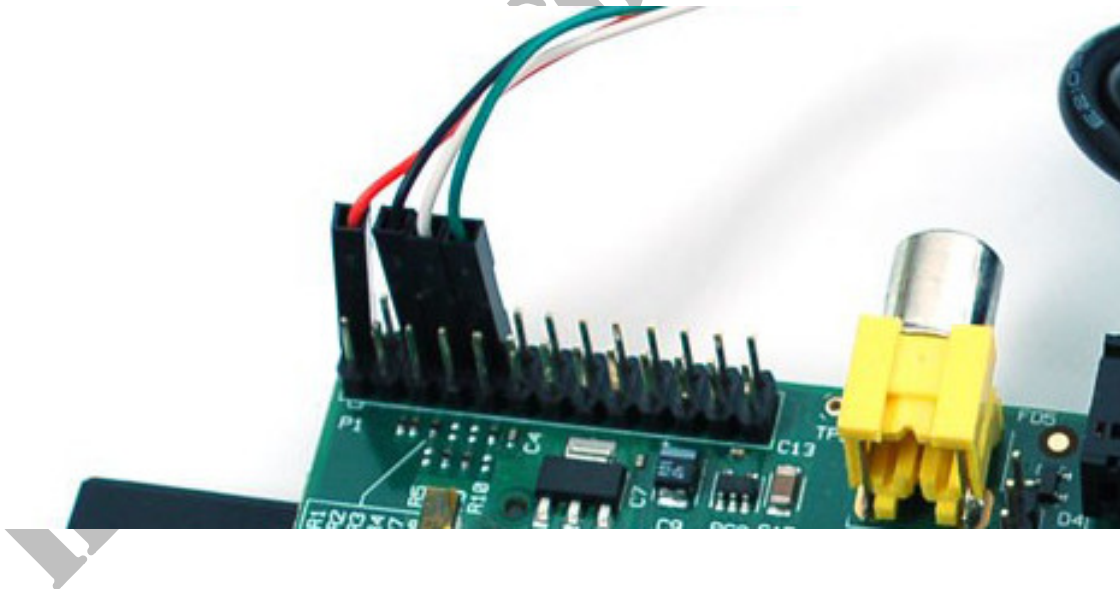
<http://learn.adafruit.com/adafruits-raspberry-pi-lesson-5-using-a-console-cable>

OUR SCENARIO - CONNECT THE ADAFRUIT USB SERIAL CABLE (NO POWER PROVIDED – YOU ARE POWERING THE PI USING GPIO PINS FROM THE ATXRASPI)

CONNECT THE ADAFRUIT USB SERIAL CABLE (NO POWER PROVIDED – YOU ARE USING USB OR OTHER MEANS TO POWER THE PI)



CONNECT THE ADAFRUIT USB SERIAL CABLE (USB SERIAL CABLE PROVIDES POWER TO PI – **DO NOT CONNECT POWER TO USB**)



From the OSX terminal type

```
screen /dev/cu.PL
```


and then hit the TAB key and it will fill in the appropriate serial device, once filled, type 115200 as below and click ENTER

THIS IS AN EXAMPLE (your numbers after PL may be different!)

```
screen /dev/cu.PL2303-000012FD 115200
```

You should be connect via serial console to the device (you should see the login prompt)

(OPTIONAL) ADDING ADDITIONAL CONTENT

ADDING GOODWILL COMMUNITY FOUNDATION CONTENT

You can request a USB drive with the content from the Goodwill Community Foundation by following the FAQ here (<http://www.gcflearnfree.org/educators/faq>). When you receive the USB stick with the latest GCF content, copy over the following three files and place them in /var/www/

1. "GCFLearnFree.Org Offline.html"
2. "Terms Of Use.html"
3. "content" folder

Change the owner to pi:pi:

```
sudo chown pi:pi /var/www/content /var/www/Terms* /var/www/GCFLearn*
```

ADDING PHET CONTENT

PhET is simulation software for various STEM courses. The product doesn't work 100% but they are updating their simulations to use HTML5 so that it will work from any modern web browser.

You can download it from here:

<https://phet.colorado.edu/>

Copy the content to /var/www/PhET

UPDATE INDEX.HTML FILES FOR ADDED CONTENT

index-23Feb2014.html

<https://drive.google.com/file/d/0ByHHOxPhKBGaVEVPQWJHcG0zOG8/edit?usp=sharing>

ENGLISH BUILD WITH KA-LITE AND GCF

index-GCF-16Feb2014.html

<https://drive.google.com/file/d/0ByHHOxPhKBGaTUJ6ejlvNHhLTFk/edit?usp=sharing>

ENGLISH BUILD WITH KA-LITE, GCF AND PHET

index-GCF-PhET-16Feb2014.html

<https://drive.google.com/file/d/0ByHHOxPhKBGaWHh6dS1Scml0QIE/edit?usp=sharing>

SPANISH BUILD - WITH KA-LITE AND GCF CONTENT

index-es-22Feb2014.html

<https://drive.google.com/file/d/0ByHHOxPhKBGaUmlXSW12cmI5TGc/edit?usp=sharing>

HACKERS FOR CHARITY