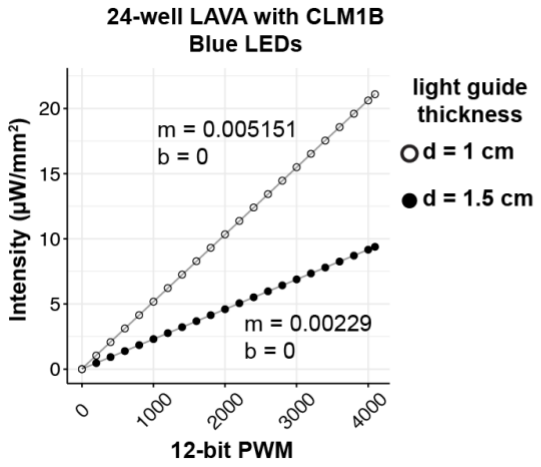
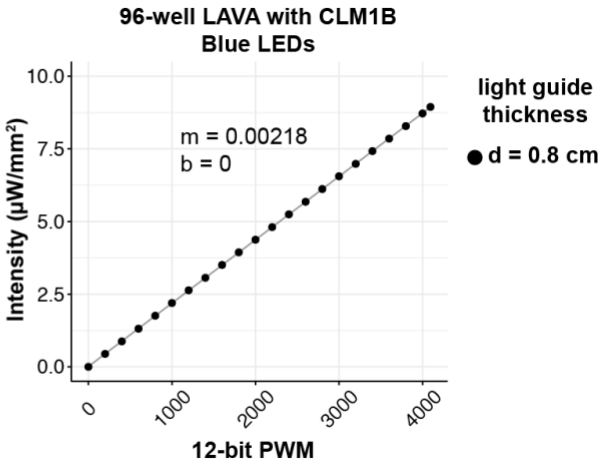
After assembling the LAVA device hardware, follow the following steps to configure the Raspberry Pi and set up wireless pattern upload from a personal computer.

We highly recommend measuring the power output of the LEDs with a power meter (such as PM100D, Thorlabs) in order to determine power output of each well in units of µW mm^{-2} (see Step C). This calibration curve will then be used to set the calibrated values *m* and *b* in the LAVA board GUI, JarJarBlinks. If using CLM1B Blue Cree LEDs as specified in the bill of materials, the following m and b values can be used without calibration:

m	b	LAVA configuration	Light guide thickness (d)
0.005151	0	24-well	1 cm
0.00229	0	24-well	1.5 cm
0.00218	0	96-well	0.8 cm





A) SD card flash instructions

(using Terminal on MacOSX; if need to use Windows for this process can google search for equivalent methods for SD card flashing in Windows)

- Download provided disk image (raspberrypi.dmg, 7.95GB file size)
- Insert blank, used, or formatted 8GB SD card into the SD reader on your Mac
- Open Terminal and locate the SD card in by entering

```
diskutil list
```

- Identify the SD Card (look under NAME and SIZE to verify correct volume). On my computer, the SD Card was /dev/disk2.
- Unmount the SD card by entering (Modify text in magenta to match your disk name)

```
diskutil unmountDisk /dev/disk2
```

- Format the SD card as FAT32

```
sudo newfs_msdos -F 32 /dev/disk2
```

- Flash SD card by entering code below. Modify text in magenta to match your disk name and disk image location:

```
sudo dd if=~/Downloads/raspberrypi.dmg of=/dev/disk2
```

Be aware that restoring the disc image to the SD Card can take some time, and you will not be shown any progress in Terminal while the SD Card is being written! My Macbook Pro took ~2hrs for an 8GB microSDHC card

- Once complete, eject card and insert into Raspberry Pi
- Power LED board and wait ~1min for Pi to turn on. LED light sequence will change (ex. B01 blinks) and you will see APPLE_PI Wifi network appear. If not, SD card flash was not successful.

B) OPTIONAL: Change name of Wifi network (default is APPLE_PI)

- Power LED board and connect to APPLE_PI Wifi network (pass: raspberry)
- Wirelessly SSH into Pi:
    - Mac: in Terminal, enter `ssh pi@192.168.4.1`
    - Windows: in PuTTY, connect to `pi@192.168.4.1`
    - Password: raspberry
- Configure the access point host software (hostapd) by entering:

```
sudo nano /etc/hostapd/hostapd.conf
```

you will see the text below:

```
interface=wlan0
driver=nl80211
ssid=APPLE_PI
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=raspberry
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

- Change **APPLE_PI** to desired Wifi name. DO NOT modify any other lines.
- use keyboard combination ctrl-O then 'enter' to save the file, then crtl-X to exit the nano text editor
- Reboot the Pi and you will see new Wifi name appear

```
sudo reboot
```

C) LED board calibration instructions

- Power LED board and connect to APPLE_PI Wifi network (pass: raspberry)
- Wirelessly SSH into Pi:
    - Mac: in Terminal, enter `ssh pi@192.168.4.1`
    - Windows: in PuTTY, connect to `pi@192.168.4.1`
    - Password: raspberry
- In new Terminal window (no SSH), cd to folder containing LED_calibration.py file
- Move the LED_calibration.py file to the Pi home directory by typing

```
scp LED_calibration.py pi@192.168.4.1:
```

- Now, we need to stop the LED_driver process on the Pi to be able to control the LEDs through the python script. To do this, return to the SSH Terminal window and view the list of running processes on the Pi by entering

```
htop
```

- One of the processes should correspond to the LED driver (see 'Command' column). Write down the PID corresponding to this process (in our case it was 376).
- Exit htop by typing q
- Kill the LED_driver process by entering code below, replacing PID with the number you recorded

```
sudo kill PID
```

- Run the python calibration code by entering

```
python LED_calibration.py
```

- You can view or edit the python code on your personal computer and re-upload it to Pi, or open and edit it on Pi directly with the built-in nano text editor.
- Once you restart the LED board, all settings will go back to normal

D) Pi Wifi network connection and pattern upload

These steps needs to be performed only once:

- Windows only: download PuTTY ssh client
- Power LED board (wait ~1min for boot to complete) and on personal computer connect to Wifi network APPLE_PI (password: raspberry)
- Wirelessly SSH into Pi:
    - Mac: in Terminal, enter `ssh pi@192.168.4.1`
    - Windows: in PuTTY, connect to `pi@192.168.4.1`
    - Password: raspberry
- If prompted, accept host key
- Ensure SSH was successful by entering `ls` (should see Pi file directory)
- Close Terminal/PuTTY

From now on, to upload light patterns:

- Connect to APPLE_PI Wifi network (no need for Terminal/PuTTY)
- Open JarJar_Blinks.jar GUI, enter m and b values and desired light patterns – click Upload. Pattern will be updated within 5 sec.

E) OPTIONAL: Update LED driver executable

- Power LED board and connect to APPLE_PI Wifi network (pass: raspberry)
- Wirelessly SSH into Pi:
    - Mac: in Terminal, enter `ssh pi@192.168.4.1`
    - Windows: in PuTTY, connect to `pi@192.168.4.1`
    - Password: raspberry
- Remove current LED driver version:

```
rm -r LED_Driver/
```

- In new Terminal window (no SSH), cd to folder containing the new LED_Driver folder

```
scp -r LED_Driver/ pi@192.168.4.1:
```

- Back in the Pi Terminal window, reboot the Pi

```
sudo reboot
```