D.Frederick

C:\\_Robotics\Raspberry\_Pi

**Notes on Building JAVA with SPI/I2C Raspberry PI**

**[May 25, 2016 – Update ]**

Download image from https://www.raspberrypi.org/downloads/raspbian/

Image: Raspbian\_Jessie\_Lite\_may\_2016 at https://downloads.raspberrypi.org/raspbian\_lite\_latest

**Unzip the image on your laptop**

**Create the SDCard Image**

Write the image to an Sdcard

Source:

https://sourceforge.net/projects/win32diskimager

Select the image file and the target Sdcard. Select "Write"

**Initial (one-time) configuration of the networking:**

Connect the RaspBerry Pi to a TV with an HDMI input

Connect a keyboard to the RaspBerry Pi

Power on the RaspBerry Pi

Log on using the default username 'pi' and password 'raspberry'

Run configuration program: sudo raspi-config

Select US keyboard

Enable SSH under advanced

Enable I2C

Enable SPI

Reboot the RaspBerry Pi

Configure the Network:

Log into the RaspBerry Pi using the account "pi" and the password just set

Configure the networking to allow a temporary configuration. (Delete all of the other lines)

See the explanation of "sudo" and a brief list of commands for the UNIX text editor futher below.

**sudo vi /etc/network/interfaces**

auto lo

iface lo inet loopback

auto eth0

iface eth0 inet static

address 10.18.95.100

netmask 255.255.255.0

gateway 10.18.95.1

Shutdown the RaspBerry Pi

**sudo shutdown –h 0**

Disconnect the RaspBerry Pi from the TV/Monitor and remove the keyboard

**Connect to the RaspBerry Pi using the following steps:**

Connect the RaspBerry Pi to your laptop. Either directly with a cross over cable or through a hub.

Start PuTTY on your Laptop (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

Configure PuTTY

Set the hostname/IP Address to 10.18.95.100

Set the connection type to SSH (Secure Shell)

Give the session a name and save the connection type for later use

Connect to the RaspBerry Pi with PuTTY (SSH)

You may receive an alert since this is the first time connection to the RaspBerry Pi.

Window titled "PuTTY Security Alert"

"Warning - Potental Security Breach!"

Select OK and continue

***Explanation:***

"**sudo**" = "Super User do" – Unix/Linux is a very secure operating system. Only the Super User (root) can edit important files. A normal user can ask the Super User to edit a file using the primative text editor vi.

**Vi basics**

Vi has two modes: Command mode and text entry mode. Press the escape key to enter command mode.

To exit a file without saving: **[Esc] :q!**

To exit a file with saving: **[Esc] :x!**

To delete one character: **x**

To replace one character: **r**

To delete a whole line: **dd**

To insert text: **[Esc] i ( This places vi in the text edit mode )**

To append text in the current line: **[Esc] A ( This places vi in the text edit mode )**

If you have a wireless USB connector, Configure the networking on the RaspBerry Pi to support both wireless and wired. Otherwise leave the networking as is:

Command to edit the interfaces file:

**sudo vi /etc/network/interfaces**

Notes on editing in **vi**:

The command above will put you in vi. Delete all of the lines of text by repeatedly entering dd. Then, select **[esc] i** to enter text enty mode. Copy and paste the text below by highlighting the text, then select [ctrl]-C, Switch windows to PuTTY and right-click into the SSH window. This will paste the text. Close the file by entering **[Esc] x!** and the file will be saved.

auto lo

iface lo inet loopback

auto eth0

iface eth0 inet static

address 10.18.95.100

netmask 255.255.255.0

#gateway 10.18.95.1

allow-hotplug wlan0

iface wlan0 inet manual

wpa-roam /etc/wpa\_supplicant/wpa\_supplicant.conf

iface default inet static

address 10.0.0.11

netmask 255.255.255.0

gateway 10.0.0.1

Command to edit the Wireless Network security configuration file:

**sudo vi /etc/wpa\_supplicant/wpa\_supplicant.conf**

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev

update\_config=1

network={

ssid="SIDD\_VALUE"

psk="9999999999999999999988888888888888888888"

proto=WPA

key\_mgmt=WPA-PSK

pairwise=TKIP

auth\_alg=OPEN

}

**NOTE: Put in your own Guest network information**

=================================

View and Update the DNS Resolver Configuration file to point to Googles DNS Server.

**cat /etc/resolv.conf**

**sudo vi /etc/resolv.conf**

nameserver 8.8.8.8

Reboot the host and verify you can get in with a wireless connection

=================================

===============================================

**DHCP client – No changes**

http://www.jeffgeerling.com/blog/2016/setting-static-ip-address-raspbian-jessie-lite-on-raspberry-pi

https://wiki.archlinux.org/index.php/dhcpcd

Raspbian Jessie Lite uses dhcpcd5

sudo /etc/dhcpcd.conf

In Raspbian Jessie Lite, there is a little additional configuration you need to provide for [dhcpcd](https://wiki.archlinux.org/index.php/dhcpcd), since Raspbian uses dhcpcd5 by default. If you want, you could disable/uninstall dhcpcd5 entirely, but it's simpler to provide the correct static IP configuration in dhcpcd's configuration file. In my case, I edited /etc/dhcpcd.conf

===========================================

**Enable Access Point**

**Before enabling the access point, get the JAVA Runtime Library installed. (Once you enable the Access Point, te PI cannot access the internet directly)**

<https://learn.adafruit.com/downloads/pdf/setting-up-a-raspberry-pi-as-a-wifi-access-point.pdf>

Enable normal WIFI to allow Pi to access packages

sudo apt-get update

sudo apt-get install hostapd isc-dhcp-server

Update: dhcpd.conf

sudo vi /etc/dhcp/dhcpd.conf

Comment out:

#option domain -name "example.org"

#option domain -name-servers ns1.example.org, ns2.example

**Enable: (Remove #)**

authoritative

**Add to the bottom:**

subnet 192.168.42.0 netmask 255.255.255.0 {

range 192.168.42.10 192.168.42.50;

option broadcast-address 192.168.42.255;

option routers **192.168.42.1**;

default-lease-time 600;

max-lease-time 7200;

option domain-name "local";

option domain-name-servers 8.8.8.8, 8.8.4.4;

}

sudo vi /etc/default/isc-dhcp-server

INTERFACES="" and update it to say INTERFACES="wlan0"

**Update Interfaces**

sudo vi /etc/network/interfaces

allow-hotplug wlan0

iface wlan0 inet static

address 192.168.42.1

netmask 255.255.255.0

**Update: hostapd**

sudo vi /etc/default/hostapd

Find the line

#DAEMON\_CONF=""

and edit it so it says

DAEMON\_CONF="/etc/hostapd/hostapd.conf"

**Update hostapd.conf**

sudo vi /etc/hostapd/hostapd.conf

interface=wlan0

driver=rtl871xdrv

ssid=Pi\_AP18

hw\_mode=g

channel=6

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

**Update**

sudo vi /etc/sysctl.conf

net.ipv4.ip\_forward=1

sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip\_forward"

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT

sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT

sudo iptables -t nat -S

sudo iptables -S

sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"

**sudo vi /etc/network/interfaces**

up iptables-restore < /etc/iptables.ipv4.nat

**wget** [**http://adafruit-download.s3.amazonaws.com/adafruit\_hostapd\_14128.zip**](http://adafruit-download.s3.amazonaws.com/adafruit_hostapd_14128.zip)

unzip adafruit\_hostapd\_14128.zip

sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.ORIG

sudo mv hostapd /usr/sbin

sudo chmod 755 /usr/sbin/hostapd

sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf

===

===

**Important files:**

sudo vi /etc/hostapd/hostapd.conf

sudo vi /etc/dhcp/dhcpd.conf

sudo vi /etc/network/interfaces

**/etc/hostapd/hostapd.conf**

interface=wlan0

driver=rtl871xdrv

ssid=Pi\_AP18

hw\_mode=g

channel=6

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

**/etc/dhcp/dhcpd.conf**

default-lease-time 600;

max-lease-time 7200;

subnet 192.168.42.0 netmask 255.255.255.0 {

range 192.168.42.10 192.168.42.50;

option broadcast-address 192.168.42.255;

option routers 192.168.42.1;

default-lease-time 600;

max-lease-time 7200;

option domain-name "local";

option domain-name-servers 8.8.8.8, 8.8.4.4;

**/etc/network/interfaces**

auto lo

iface lo inet loopback

auto eth0

iface eth0 inet static

address 10.18.95.100

netmask 255.255.255.0

#gateway 10.18.95.1

allow-hotplug wlan0

#iface wlan0 inet manual

iface wlan0 inet static

#wpa-roam /etc/wpa\_supplicant/wpa\_supplicant.conf

#iface default inet static

#address 10.0.0.16

address 192.168.42.1

netmask 255.255.255.0

#gateway 10.0.0.1

up iptables-restore < /etc/iptables.ipv4.nat

===========================================

Install JAVA JDK on pi

<< Install JAVA prior to turning in API

All future Raspbian images will ship with Oracle Java by default; existing users can install it by typing:

sudo apt-get update

sudo apt-get install oracle-java7-jdk

====================================

Install J4J

The simplest method to install Pi4J on your RaspberryPi is to execute the following command directly on your RaspberryPi.  
curl -s get.pi4j.com | sudo bash

sudo curl -s get.pi4j.com > pi4j\_install\_script

sudo bash

===========================================

**Save the current configuration by saving the current image**

Use the Win32DiskImager tool to save a copy of the image

Either:

RaspberryPi\_Wheezy\_with\_Network\_configured\_wired\_11\_wireless100.img

RaspberryPi\_occidentalis\_with\_Network\_configured\_wired\_11\_wireless100.img

**Restore disk image** (Only If you need to)

Starting with RaspberryPi\_Wheezy\_with\_Network\_configured\_wired\_11\_wireless100.img

Notes:

Has wireless and wired configured

Wired 10.18.95.10

Wireless: 10.0.0.11

Issues:

Need to figure out how to access the internet from wireless with wired configured

**Image Checks (Optional) and examples of Unix Commands:**

**View the network configuration (cat = concatenate [list file contents to screen] )**

**cat /etc/network/interfaces**

**View the network Interface configuration**

**ifconfig**

**View the DNS Server address**

**cat /etc/resolv.conf**

**Ping a network location**

**ping 10.18.95.100**

**Determine if JAVA is installed (Find a file called java searching from the root "/" of the file system)**

**sudo find / -name 'java' –print**

**Display the JAVA version**

**/usr/bin/java -version**

**Display the contents of the /usr/bin (User added binary files) in long format ordered by time reverse**

**ls -ltr /usr/bin**

**Display the current file system directory (Print working directory)**

**pwd**

***Lesson Learned: Cut and paste where possible. Typing leads to errors.***

**Install Pi4J (Connecting Java to the Raspberry Pi)**

Reference: <http://pi4j.com/>

<http://pi4j.com/install.html>

Determine if Pi4J is installed

**ls -ltr /opt/pi4j**

if you received the message "cat: /opt/pi4j: No such file or directory", then pi4j is not installed

Offline Instructions:

1. Download the Pi4J Debian/Raspian installer package (.deb) using your web browser at the following URL:

<http://get.pi4j.com/download/pi4j-1.0-SNAPSHOT.deb>

Download to a folder called:

C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\Pi4J

1. Transfer the download installer package over to your RaspberryPi. You can use any method you prefer to transfer the file (USB, SCP, FTP, etc.)

Copy to the RPI using SCP and place in the user "pi" home directory.

[ Run this command in a Command Window on the Windows computer]

**C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\PUTTY\pscp -pw pas1word C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\Pi4J\pi4j-1.0-SNAPSHOT.deb pi@10.0.0.11:/home/pi**

1. Once the installer package is available on your RaspberryPi, use the following command on the Pi to perform the installation:

[ Run this command on the Raspberry pi]

**sudo dpkg -i pi4j-1.0-SNAPSHOT.deb**

This will install the Pi4J libraries and example source files to:

/opt/pi4j/lib

/opt/pi4j/examples

**Reboot the Raspberry pi**

**sudo reboot**

Wait about 60 seconds and reconnect with PuTTY

Test the Installation:

Run the gpio command to check the installation:

**gpio -v**

**gpio readall**

**Enable the SPI Interface:**

Reference: <https://projects.drogon.net/understanding-spi-on-the-raspberry-pi/>

Installing the SPI kernel driver

The easiest way to do this is with the gpio program that’s part of wiringPi:

**gpio load spi**

That command will load the SPI driver and change ownership to the user running the command. Once we’ve done that, we can then run our SPI programs.

**Configure the I2C interface:**

Reference:

<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c>

Perform the following configurations

**sudo vi /etc/modprobe.d/raspi-blacklist.conf**

then add a # in from of the black list commands to enable the modules.

#blacklist spi-bcm2708

#blacklist i2c-bcm2708

**sudo vi /etc/modules**

Add the following two lines to the end of the file:

**i2c-bcm2708**

**i2c-dev**

**Now install the i2c-tools package by:**

Reference:[**https://packages.debian.org/sid/utils/i2c-tools**](https://packages.debian.org/sid/utils/i2c-tools)

Copy to Raspberry pi using secure copy (SCP)

[ Run this command in a Command Window on the Windows computer]

**C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\PUTTY\pscp -pw pas1word C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\I2C-Tools\i2c-tools\_3.1.1-1\_armhf.deb pi@10.0.0.11:/home/pi**

Install the package:

**sudo dpkg -i i2c-tools\_3.1.1-1\_armhf.deb**

On-Line process:

**sudo apt-get install i2c-tools**

Now add a new user to the i2c group:

**sudo adduser pi i2c**

Reboot the machine by:

**sudo shutdown -r now**

After the reboot test to see any device connected by:

**sudo i2cdetect -y 1**

**sudo i2cdetect -y 0**

================================================

**Initial JAVA Demonstration**

Create a small JAVA program on the RaspBerry Pi. Compile the program and run it.

Move to your home directory using the following command:

**cd /home/pi**

Create a simple program.

**vi HelloWorld.java**

Highlight the following text and paste it into vi. (Press "i" to get vi into the insert mode, then right click)

**public class HelloWorld {**

**public static void main(String args[]) {**

**java.lang.System.out.println("Hello World!");**

**}**

**}**

Compile the JAVA source.

**javac HelloWorld.java**

Run the JAVA Program

**java -cp . HelloWorld**

==================================================

**Writing the same JAVA program in NetBeans**

Create the project using the commands:

File -> New Project -> JAVA Application

Define Project Name NetBeansHelloWorld

Define Project Location: C:\\_Robotics\Raspberry\_Pi\NetBeans\NetBeansProjects

Add the one line of code after the TODO line:

java.lang.System.out.println("Hello World! From NetBeans");

Run -> Build Main Project

Copy the JAVA application JAR file (java archive) to the Raspberry Pi

**C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\PUTTY\pscp -pw pas1word C:\\_Robotics\Raspberry\_Pi\NetBeans\NetBeansProjects\NetBeansHelloWorld\dist\NetBeansHelloWorld.jar pi@10.0.0.11:/home/pi**

**Note:** Sometimes its easier to copy the Windows command above to notepad to make sure the command is all on the same line and easier to see.

Run the command on the RaspBerry Pi using the following command:

**java -jar "NetBeansHelloWorld.jar"**

====================================

**Configuring NetBeans to Run JAVA programs that use Input/Outputs of WiringPi**

When JAVA applications need to use features of the **WiringPi,** the JAVA compiler (NetBeans) must have access to the WiringPi Library. This is a JAR file installed on the RaspBerry Pi.

Copy the WiringPi Library to the Laptop using the following command:

**C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\PUTTY\pscp -pw pas1word pi@10.0.0.11:/opt/pi4j/lib/pi4j-core.jar C:\\_Robotics\Raspberry\_Pi\NetBeans\NetBeansProjects**

For the specific JAVA Project use the following command sequence to link in the WiringPi Library

In Project, Right click on the name of the Project the select Properties.

Next select libraries, select import JAR/Folder, browse to:

C:\\_Robotics\Raspberry\_Pi\NetBeans\NetBeansProjects\pi4j-core.jar

**Testing the LEDs Scripts**

**This java program reads a text file which drives the colors of the LEDS**

**Complete project:**

C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\Example\_LED\_Code\Complete\_Project

**Key files (Source and compiled):**

**C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\PUTTY\pscp -pw pas1word C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\Example\_LED\_Code\Complete\_Project\Example\dist\Example.jar pi@10.0.0.11:/opt/pi4j/examples**

**C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\PUTTY\pscp -pw pas1word C:\\_Robotics\Raspberry\_Pi\RaspBerry\_pi\_Build-up\_SW\Example\_LED\_Code\input.txt pi@10.0.0.11:/opt/pi4j/examples**

**Run the program on the RaspBerry Pi (Need to figure out how to better reference the library)**

**sudo java -cp Example.jar:.:classes:/opt/pi4j/lib/'\*' example.Example**