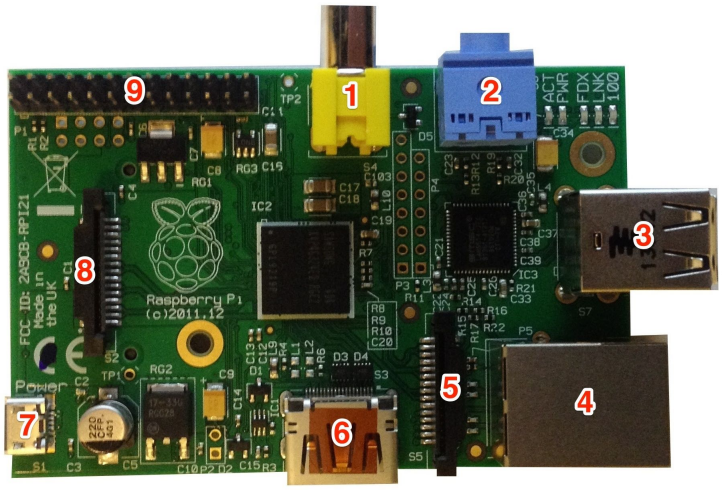
EECS 113 Assignment 6:
Raspberry Pi Setup Guide/Report

Ryan M. Morrison

General Setup

Introduction to the Hardware



1. Composite Video RCA
2. 3.5mm Audio Out
3. USB 2.0 (2 ports)
4. Ethernet RJ45
5. Camera CSI
6. HDMI
7. Micro-USB Power
8. Display DSI
9. GPIO

Figure 1: Raspberry Pi unit with itemized list of peripheral connection ports.

1. ACT: Green = SD Card Access or Activity
2. PWR: Red = Unit is Powered with 3.3V
3. FDX: Green = Full Duplex (LAN) Connected
4. LNK: Green = Link/Activity (LAN)
5. 100: Yellow = 100Mbit (LAN) Connected

Figure 2: Status LEDs: located on the board near 3.5 audio out (see top left of Figure 1 image).

In order to set up the Raspberry Pi according to this guide, the following items are required:

- SD card loaded with NOOBS installer.
- USB keyboard and mouse. It is possible to set up with just a keyboard, but a mouse simplifies things.
- Monitor with either HDMI or USB-to-serial cable.
- One of the two cables mentioned above (whichever is appropriate for your monitor).
- Micro USB power supply.
- USB wi-fi dongle if using wireless internet, or ethernet cable to connect to router.

The unit and accessories used for this guide are bundled into a kit available at the following online retailer: http://www.oddwires.com/raspberry-pi-starter-kit-pro/

Note. For simplicity, the Raspberry Pi unit will be referred to as RPi for the duration of this guide.

Note. Whenever a terminal command is shown in this guide, it will be prefaced by a dollar sign and displayed in a typewriter font: `$ this is how terminal commands look`.

Hardware Connections for Viewing Raspberry Pi Output

To interact with the RPi GUI or issue commands with the terminal, the unit can be connected to a monitor, television, or accessed remotely. This section shows connections to a television and monitor; remote access is discussed later.

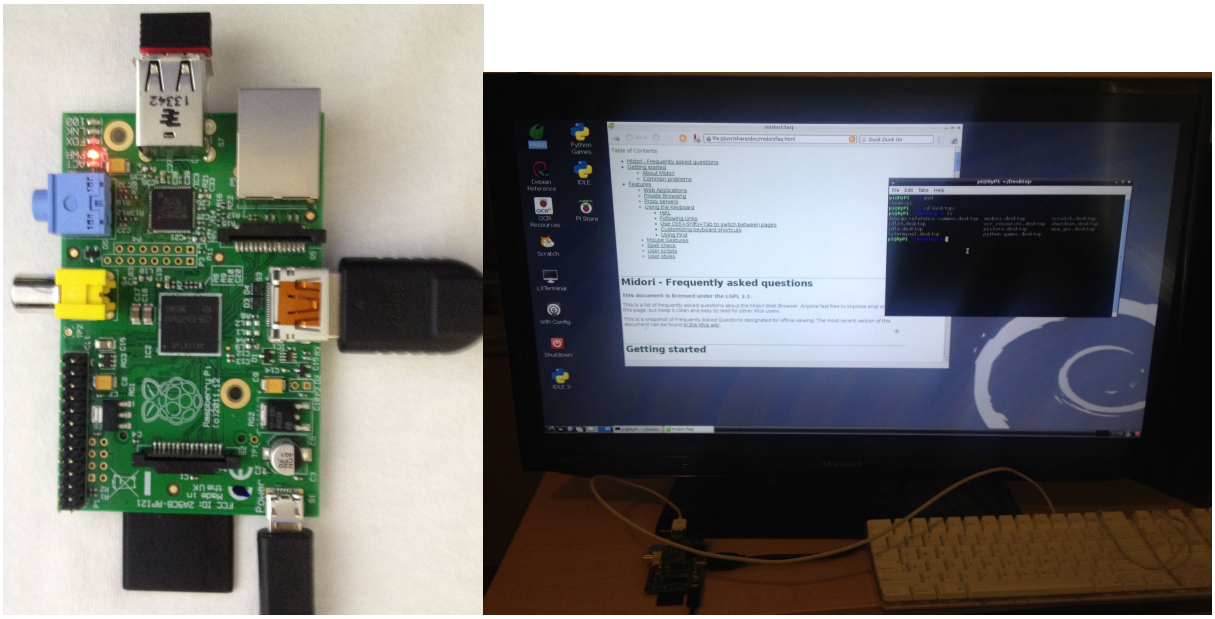


Figure 3: RPi unit with HDMI cable connected (left) and the unit connected to a television (right).

Figure 4: RPi unit connected to PC via serial-to-usb.

Initial Setup with Raspbian OS

It should be mentioned that whenever the micro USB power is connected to the RPi, the user should not simply pull the cord to power down the unit. Whenever shutting down the RP is desired, always use the Shutdown option via the GUI, or enter `$ sudo poweroff`, or `$ sudo halt` in the terminal. This will avoid the possibility of SD card corruption.

1. Prior to powering on the RPi: insert SD card and connect monitor, keyboard, and mouse. Also connect either an ethernet cable or USB wi-fi dongle.
2. Plug micro USB power adapter into source and then connect to RPi unit.
3. The NOOBS installer will boot into a screen similar to the one below:

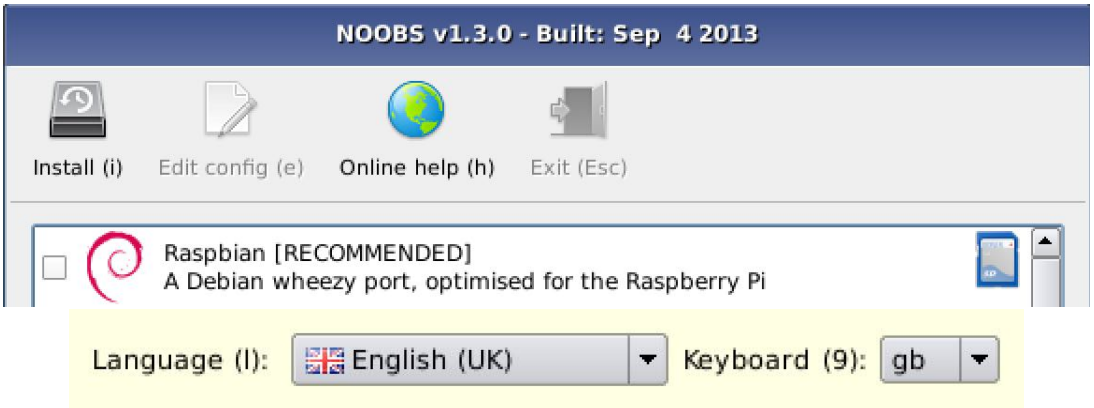


Figure 5: Relevant NOOBS installer menu options.

4. If a mouse is not being used, use the keyboard (arrow keys, numeric keys, tab key, return key, etc.) to navigate menus and make selections.
5. Here, we are installing the Raspbian OS which is best for the RPi. Depending on the version being installed, there may be options at the bottom of the screen (see Figure 3 above) for the system and keyboard language; if it is there, choose the preferred languages now, otherwise this setting can be selected in step 7.

   Select the Raspbian option and select 'Install' at the top of the screen. Select 'Yes' for any verification prompts and wait for the install to finish (it will possibly take considerable time).
6. After the install is finished, select 'OK' and wait for the system to reboot.
7. After reboot, the raspi-config screen will appear with a number of configuration options:

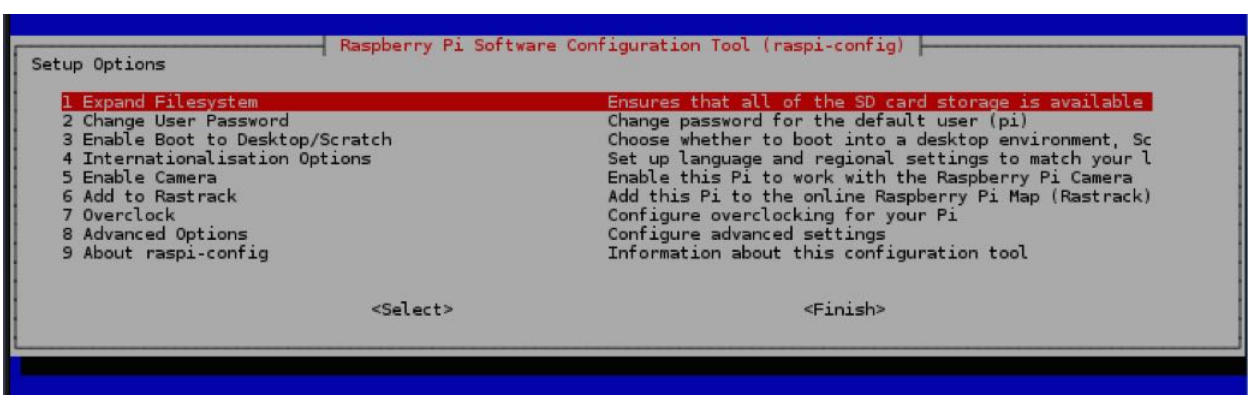


Figure 6: Raspberry Pi Software Configuration Tool (raspi-config).

8. The following explanations are found at http://www.raspberrypi.org/documentation/configuration/raspi-config.md. Here is a walkthrough of the configuration options:

(a) **Expand Filesystem**: Assuming Raspian install is performed with the NOOBS installer, this option is not needed. Otherwise, select to expand installation to entire SD card.

(b) **Change User Password**: Default user is `pi` and the password is `raspberry`. This option is used to change those values.

(c) **Enable Boot to Desktop/Scratch**: This option selects if the RPi boots into terminal or desktop. Choose boot into desktop.

(d) **Internationalisation Options**: If keyboard/language settings were not selected in step 5, use this option to make those changes now. Additionally, this option is used to select the time zone.

(e) **Enable Camera**: This option enables the RPi camera module. For the purposes of this guide, it is not needed.

(f) **Add to Rastrack**: This option adds the location of the RPi unit to a Google map that tracks Pi users around the world. For the purposes of this guide, it is not needed.

(g) **Overclock**: This option will overclock the RPi unit's CPU. Selecting this produces mixed results and possible system instability. Is a subject for another guide and not needed for the purposes of this guide.

(h) **Advanced Options**: See step 9 below for detailed explanation of this.

(i) **About raspi-config**: Self-explanatory info dialog.

9. Selecting 'Advanced Options' in the RPi configuration tool presents the following screen:

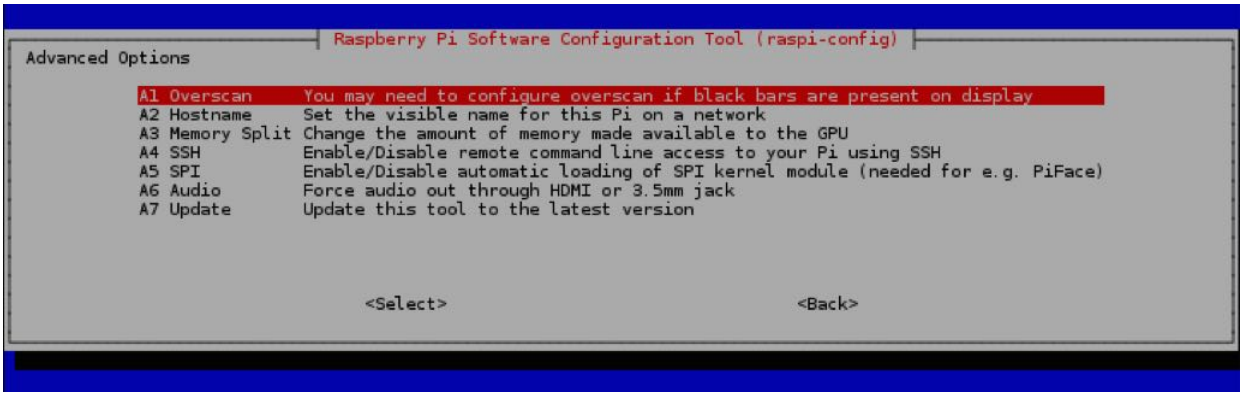


Figure 7: Advanced Options sub-menu in the raspi-config tool.

10. The following explanations are found at http://www.raspberrypi.org/documentation/configuration/raspi-config.md. Here is a walkthrough of the configuration options:

(a) **Overscan**: If some of the screen is cropped at the edges on the television/monitor being used, enable overscan to rectify this.

(b) **Hostname**: This option is used to set the visible name of the RPi unit on the network. Screenshots in this guide have the hostname set as `RyPi`.

(c) **Memory Split**: This option is used to allocate memory to the GPU. For the purposes of this guide, it is not needed.

(d) **SSH**: In this guide, ssh will be used to access the RPi filesystem remotely. This will allow the unit to be untethered from a display and, thus, make working with the unit easier. Enable this feature and make sure to change the default password (step 8b) to something secure.

(e) **SPI**: Setting for automatic loading of SPI kernel module. Not needed for this guide.

(f) **Audio**: Setting for audio out options (HDMI or 3.5mm jack). Not needed for this guide.

(g) **Update**: Update raspi-config tool to latest version. Not needed for this guide.

11. Once all the necessary options have been configured, select 'Finish' on the main configure screen and reboot the system.

UCI Network Configuration

A useful terminal command that will be used later in this guide, and probably many times in the future is `$ ifconfig`. Open the terminal application and type that command at the prompt. A screen similar to the one below should appear:

Figure 8: Output of `ifconfig` command.

The sections `eth0` and `wlan0` are of interest here. In the screenshot above, eth0 shows no IP address, whereas wlan0 does—this indicates that this particular unit is connected to a wireless network. Setting up for use on the UCI network requires the assignment of an IP address to the unit prior to connecting to the network.

1. Note the `HWaddr` for the unit (in the figure above, it is `b8:27:eb:ba:d4:f0`).
2. On another computer, navigate to UCI's Mobile Access Registration website: http://www.oit.uci.edu/mobile/registration.html
3. Use the address noted in step 1 (without colons) and register it as a new MAC address on the site.
4. Wait a few minutes, and then reenter `$ ifconfig`. The IP address, which is needed in future sections, should now be available (in the figure above, it is `10.0.1.20`).

Hardware Connections for Internet Access

Figure 9: RPi unit with USB wi-fi dongle and `ifconfig` output.

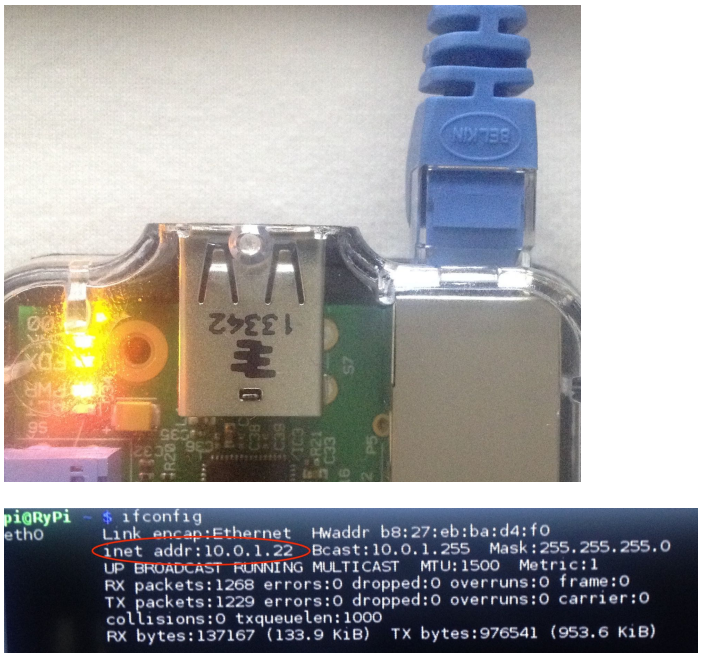


Figure 10: RPi unit with ethernet connected and `ifconfig` output.

Updating and Upgrading

Now that the RPi unit is set up and it is necessary to perform some steps to get everything up to date. Updating and upgrading the RPi should be done periodically in order to keep all the installed software and packages current.

1. In the terminal, enter `$ sudo apt-get update`. This updates the package lists from the repositories with information on the newest packages and their dependencies. Contrary to what many believe, this doesn't actually update any software, but it is a necessary precursor to actually update the installed software/packages.

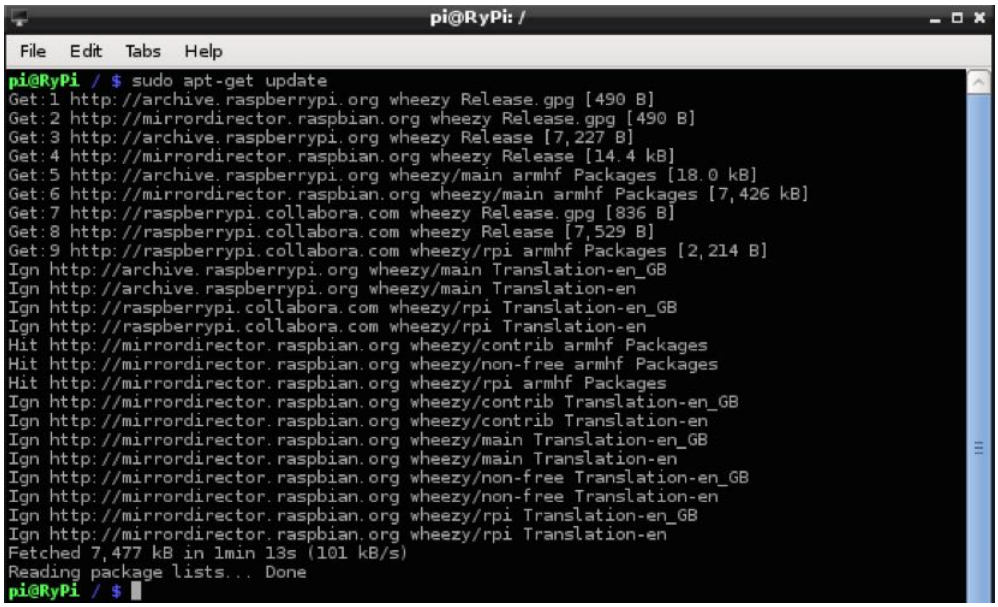


Figure 11: Output for `sudo apt-get update` command.

2. After updating the package lists, enter `$ sudo apt-get upgrade`. This actually reads the updated package lists and updates any packages that are not current. At some point the terminal will prompt for yes/no answer to proceeding with the upgrade process; type `y` and press enter.

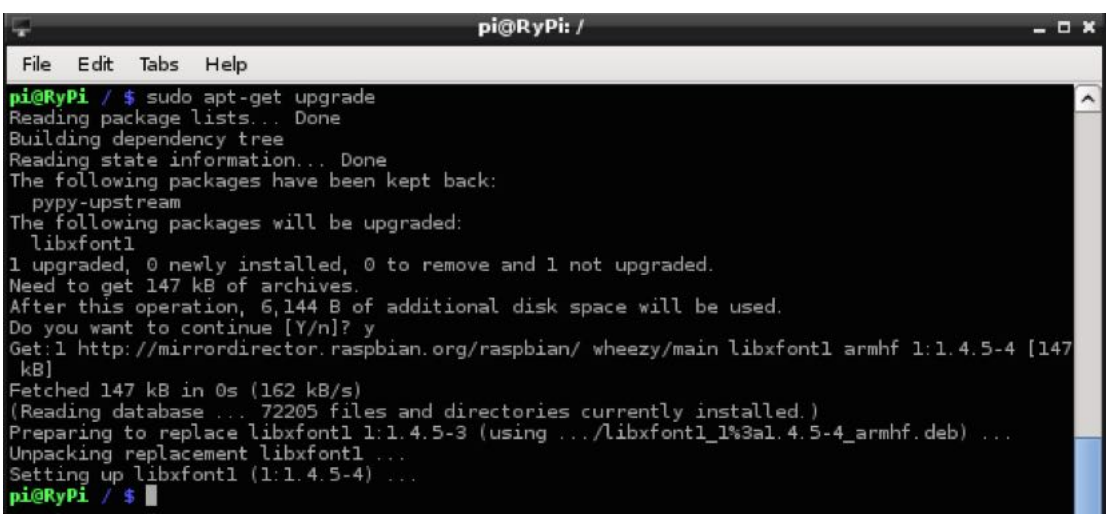


Figure 12: Output for the `sudo apt-get upgrade` command.

3. The first time updating and upgrading is done, it may take a considerable amount of time. Remember to update and upgrade periodically.

File System Access with SSH

Setting up SSH on the RPi will establish the unit as a host server, whereupon clients with access (RPi user password) can view files and folders, install packages, execute code, etc. remotely. This is convenient, and suggested here, for the purpose of freeing the RPi from being physically connected to a monitor, mouse, and keyboard. Additionally, the user is benefitted by gaining access to the USB ports for other peripherals necessary when prototyping. A working knowledge of Linux terminal commands is needed for this to be effective, as the interaction with the RPi will be strictly terminal-based (no GUI).

1. If the user password was not changed in step 8b of 'Initial Setup with Raspbian OS', refer to that section and change the user password in the raspi-config tool. This is crucial, because if the password is left as default, anybody can gain access to the RPi filesystem.

2. If SSH was not enabled in step 10d of 'Initial Setup with Raspbian OS', refer to that section and enable SSH by selecting Advanced Options→SSH→Enable in the raspi-config tool.

3. In the RPi terminal, enter the following command:
   `$ sudo rm /etc/ssh/ssh_host_* && sudo dpkg-reconfigure openssh-server`.
   This will generate unique host public/private keys which will further secure the RPi from intruder attacks.

Figure 13: Generating unique host public/private keys on the RPi.

4. There are some differences in setting up a remote client depending on the client platform. This guide is written under Mac OS X, and therefore provides instructions respective of that platform. Linux follows the same procedure as Mac OS X, so it is omitted. Windows requires an additional program, Putty, to be installed. Instructions for Windows users can be found here: http://elinux.org/RPi_Remote_Access.

5. To connect to the RPi with SSH, open the terminal application in Mac OS X and type:
   `$ ssh -lpi <IP address of RPi>`
   A prompt will ask for the RPi user password—enter the password and the client should connect to the RPi server.

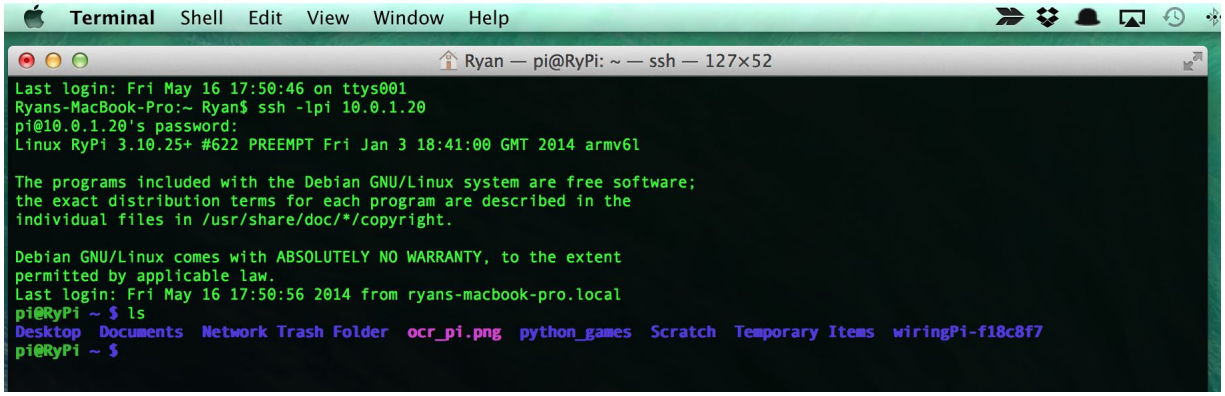


Figure 14: Remote connection to RPi with SSH on Mac OS X.

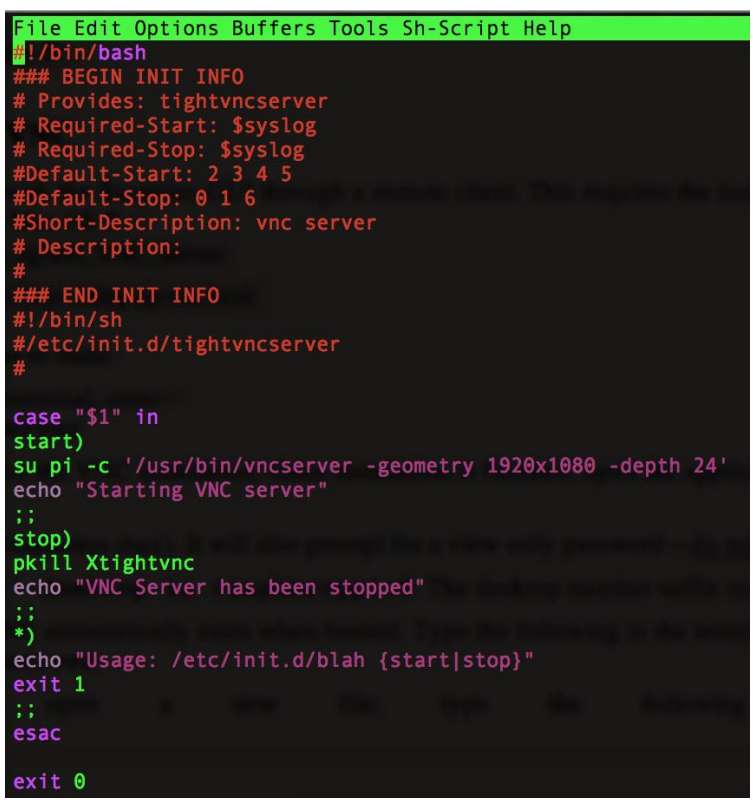
6. Lastly, if connecting remotely to the RPi is going to be a frequent occurrence, it is suggested that a static IP address is created for the RPi. Otherwise, it is possible that a new IP address will be assigned to the RPi each time it is rebooted—this would require hooking up the RPi to a monitor and keyboard to find the new IP address. Directions for this can be found here: http://www.suntimebox.com/raspberry-pi-tutorial-course/week-3/day-5/.

Remote Access and Desktop Control with VNC

It is also possible to connect to the RPi and interact with the Raspbian GUI through a remote client. This requires the installation of a VNC server on the RPi and a VNC client on the other machine. Again, this guide provides instructions for Mac OS X.
For Linux instructions, look here: http://elinux.org/RPi_VNC_Server

For Windows instructions, look here: http://myraspberrypiexperience.blogspot.com/p/setting-up-vnc.html

1. As suggested in step 6 above, set the IP address to static.
2. Install the VNC server on the RPi—in the RPi terminal, enter:
   `$ sudo apt-get install tightvncserver`
3. Follow and agree to any subsequent prompts for the VNC installation. Once installation is finished, open the application by typing:
   `$ vncserver`
4. It will prompt for a password to be created (8 characters max). It will also prompt for a view only password—do not select this (choose no).
5. A message will be returned that says: `New 'X' desktop is raspberrypi:0`. The desktop number suffix may be different, i.e. 1, 2, etc.—make note of it.
6. A startup script needs to be created so that VNC automatically starts when booted. Type the following in the RPi terminal:
   `$ sudo nano /etc/init.d/tightvncserver`
7. The text editor nano will open a new file; type the following in the file and then save/close it:



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
### BEGIN INIT INFO
# Provides: tightvncserver
# Required-Start: $syslog
# Required-Stop: $syslog
#Default-Start: 2 3 4 5
#Default-Stop: 0 1 6
#Short-Description: vnc server
# Description:
#
### END INIT INFO
#!/bin/sh
#/etc/init.d/tightvncserver
#

case "$1" in
start)
su pi -c '/usr/bin/vncserver -geometry 1920x1080 -depth 24'
echo "Starting VNC server"
;;
stop)
pkill Xtightvnc
echo "VNC Server has been stopped"
;;
*)
echo "Usage: /etc/init.d/blah {start|stop}"
exit 1
;;
esac

exit 0
```

Figure 15: Startup script to allow VNC to start at RPi boot.

8. In Figure 13, the line that references geometry and depth can be configured differently according to screen resolution—lesser values will present faster network response, but the desktop appearance will be degraded.

9. Provide the script with executable permissions: `$ sudo chmod 755/etc/init.d/tightvncserver`

10. Enter the following to allow automatic starting of VNC upon booting:
    `$ sudo update-rc.d tightvncserver defaults`

11. The following commands will start/stop the VNC server:
    `$ sudo /etc/init.d/tightvncserver start`
    `$ sudo /etc/init.d/tightvncserver stop`

12. To connect to the RPi with VNC, open the Finder in Mac OS X and find the RPi in the sidebar:

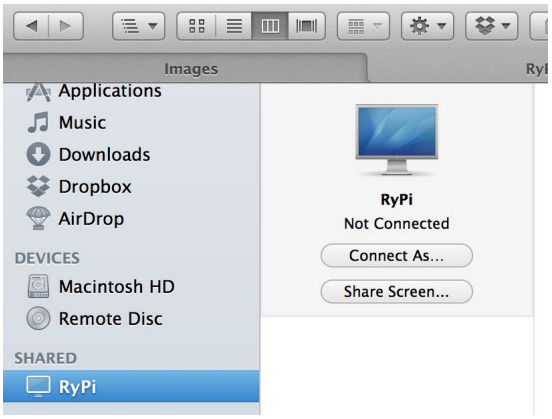


Figure 16: Mac OS X Finder with the RPi server 'RyPi' in the sidebar.

13. Click 'Share Screen' and the screen share application should open with the RPi desktop available:

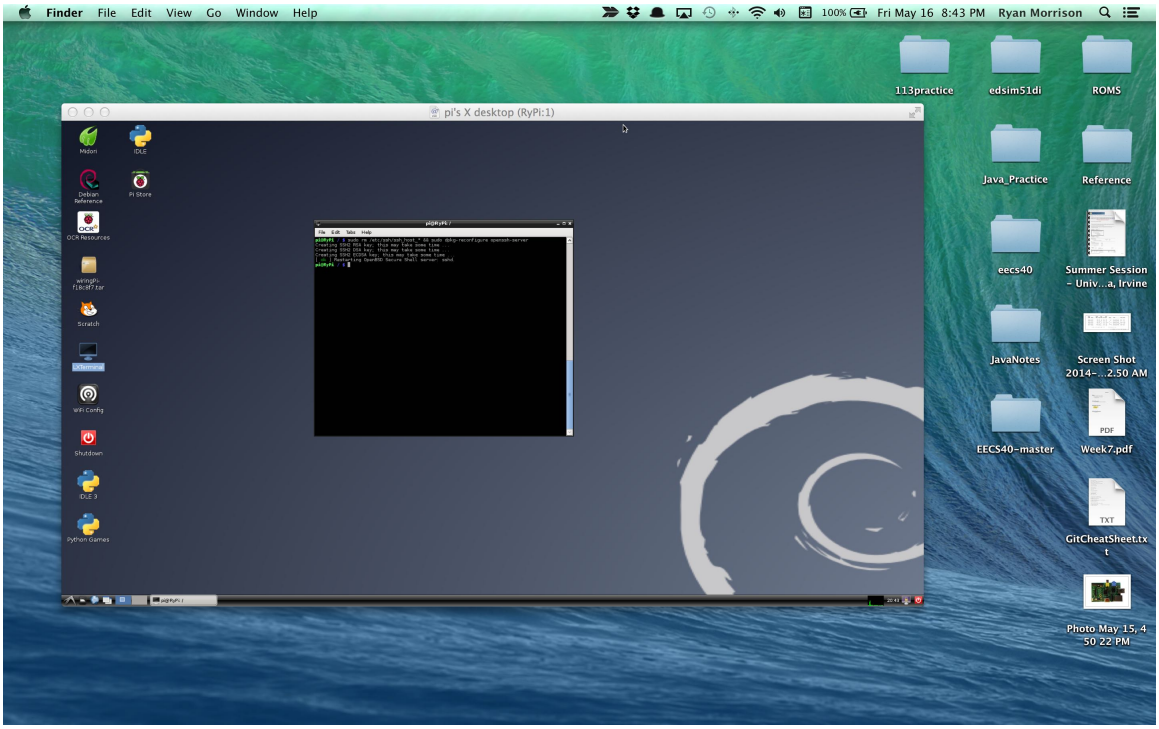


Figure 17: Mac OS X remotely connected to RPi desktop with VNC.

14. Note: depending on network performance and graphics settings, there may be considerable 'lag' with UI effects such as resizing windows, cursor movement, etc. For this reason, remote connection to the RPi is usually best with SSH.

Interfacing LCD with Raspberry Pi

In this section, a package will be installed that allows programs executed on the RPi to interface with the GPIO pins. Then an LCD will be connected to the RPi and this connection will be verified by running a sample program from the wiringPi package.

Installing WiringPi Package

1. Per the TA instructions, this package must be downloaded on a separate machine and then transferred to the RPi with FTP. Reasons for this are unclear, as it seems that directly downloading the package with the RPi should be possible. However, this guid will follow the indirect approach, but, instead of using FTP which requires additional software, this guide will utilize the SSH and VNC features that are already set up and available to use.
   For the sake of completeness, FTP clients suggested by the TA are WinSCP and FileZilla.

2. Establish remote SSH and VNC connections to the RPi as instructed in 'Viewing and Accessing Raspberry Pi OS'.

3. On the client machine, download the latest WiringPi snapshot from here:
   https://git.drogon.net/?p=wiringPi;a=summary

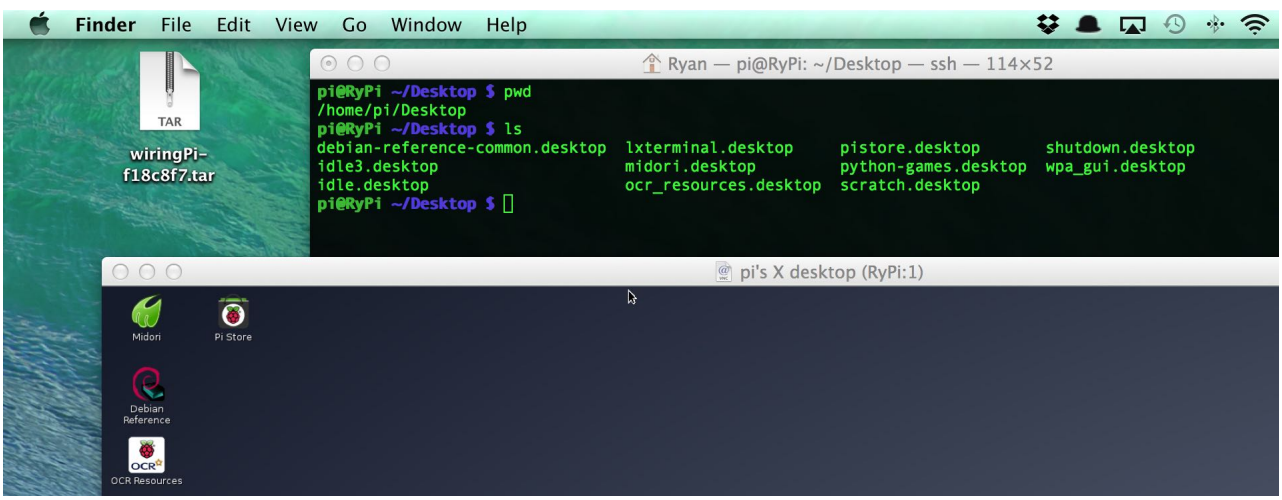


Figure 18: Mac OS X desktop with the downloaded wiringPi package and RPi terminal/screen share sessions.

4. There are two options available: dragging and dropping the wiringPi tar file from the client to the RPi, or using a terminal command.

   (a) For drag and drop, open the Finder in Mac OS X and open two tabs: one for RPi desktop and one for client desktop. Drag the package from the client to the RPi desktop.

   (b) For file transfer with the terminal, enter the following command:
   `$ scp wiringPi-xxxxxxx.tar <user>@<IP address>:/path/to/desired/location`
   The x's will be substituted for the particular package build you have. User is the RPi user name, IP address is the one for the RPi, and the path is the destination for the file on the RPi.

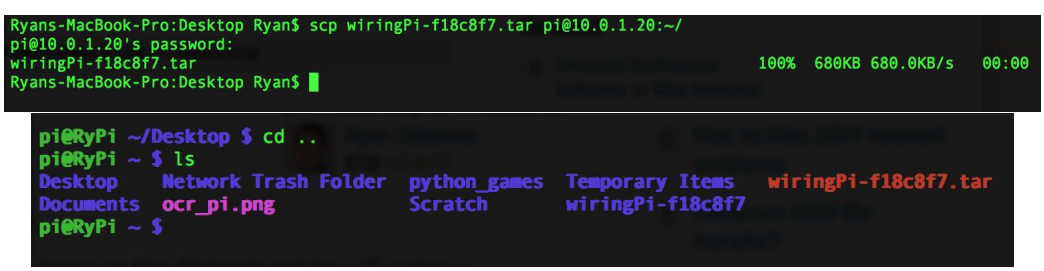


Figure 19: Terminal command to transfer file from client (top) to RPi server (bottom).

5. Now that the tar file is on the RPi in the home directory, extract it by typing:
   `$ tar xfz wiringPi-xxxxxxx.tar`

6. Install I^{2}C library by typing: `$ sudo apt-get install libi2c-dev`

7. Navigate into the extracted wiringPi directory and build the library by typing: `$ ./build`

8. Verify everything is installed properly:

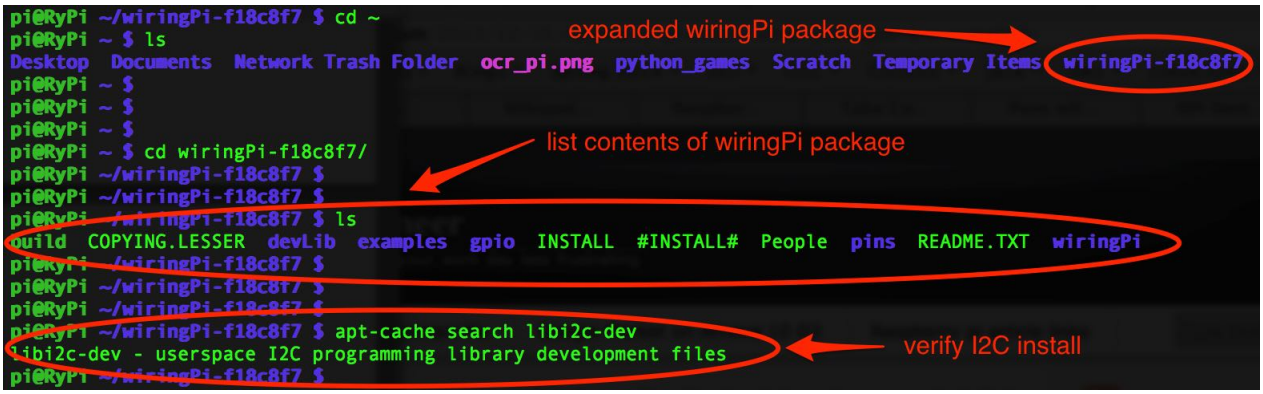


Figure 20: Install verification of wiringPi and I^{2}C libraries.

9. Load and test I^{2}C (from within wiringPi directory) by typing:
   `$ gpio load i2c`
   `$ gpio i2cd`

Figure 21: Load and test I^{2}C.

10. To make the wiringPi examples, change to the examples directory: `$ cd examples/`.
    Choose to either make all examples or selective ones: `$ make really-all` or `$ make <example name>`

Figure 22: Output after building all wiringPi examples.

Connecting LCM1602A LCD to Raspberry Pi

1. The LCD must first be soldered before it can be used. See the 'References' section for a link on soldering tutorials.

2. Once the LCD is soldered, connect the LCD to the RPi according to the following diagram and the comment notes from lcd.c file in wiringPi examples.

wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin
–	–	3.3v	1 | 2	5v	–	–
8	R1:0/R2:2	SDA	3 | 4	5v	–	–
9	R1:1/R2:3	SCL	5 | 6	0v	–	–
7	4	GPIO7	7 | 8	TxD	14	15
–	–	0v	9 | 10	RxD	15	16
0	17	GPIO0	11 | 12	GPIO1	18	1
2	R1:21/R2:27	GPIO2	13 | 14	0v	–	–
3	22	GPIO3	15 | 16	GPIO4	23	4
–	–	3.3v	17 | 18	GPIO5	24	5
12	10	MOSI	19 | 20	0v	–	–
13	9	MISO	21 | 22	GPIO6	25	6
14	11	SCLK	23 | 24	CE0	8	10
–	–	0v	25 | 26	CE1	7	11
wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin

8-bit Interface
GPIO 0-7 connected to display data pins 0-7
GPIO 11 is the RS pin
GPIO 10 is the Strobe/E pin

4-bit Interface
GPIO 4-7 is connected to display data pins 4-7
GPIO 11 is the RS pin
GPIO 10 is the Strobe/E pin

Figure 23: GPIO pin diagram from wiringPi website (left) and the pin connections as listed in lcd.c (right).

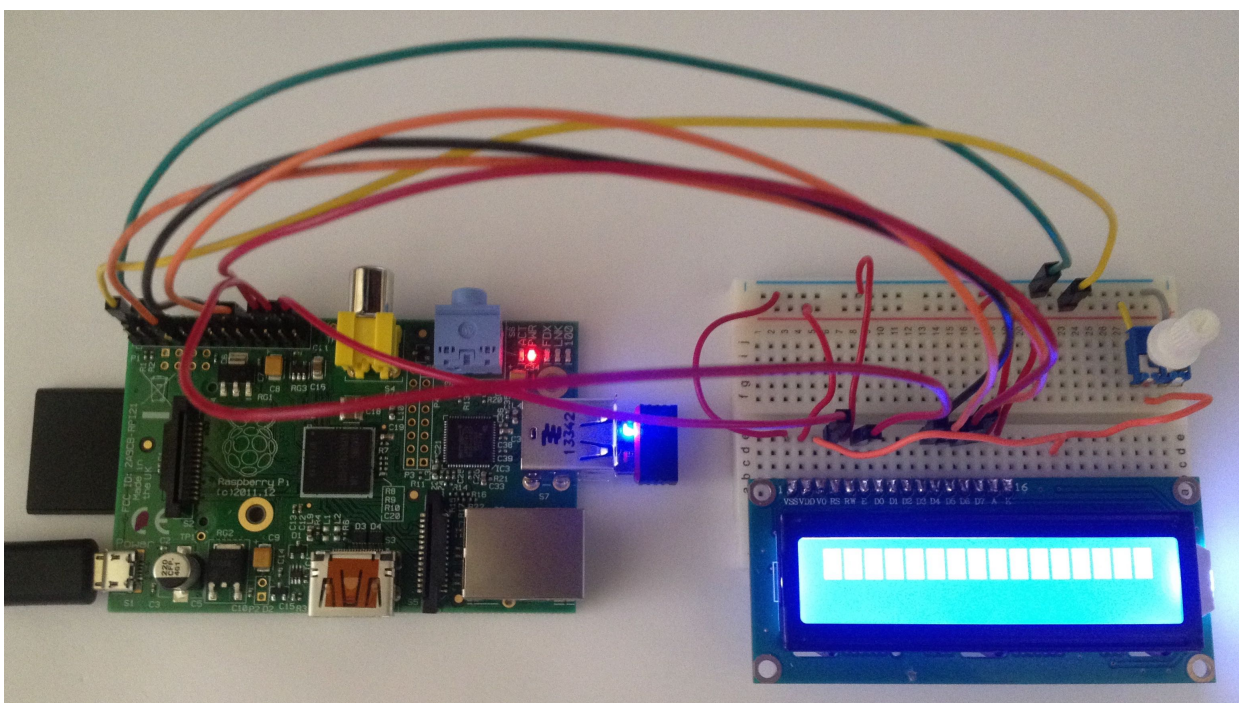


Figure 24: RPi and LCD connected and powered up.

3. The potentiometer on the right is used as a contrast control. Nothing will display on the LCD except for squares until some test code is run on it.

Running LCD Sample Program from WiringPi

1. On the RPi, navigate to the wiringPi examples folder: `$ cd ~/wiringPi-xxxxxxx/examples`
2. Make sure that the examples have been built (see section 'Installing WiringPi Package' step 10).
3. Run the lcd.c test code by typing: `$ sudo ./lcd 4 16 2`
4. The test code will issue terminal prompts for user to press enter a series of times. The three test steps are shown below.

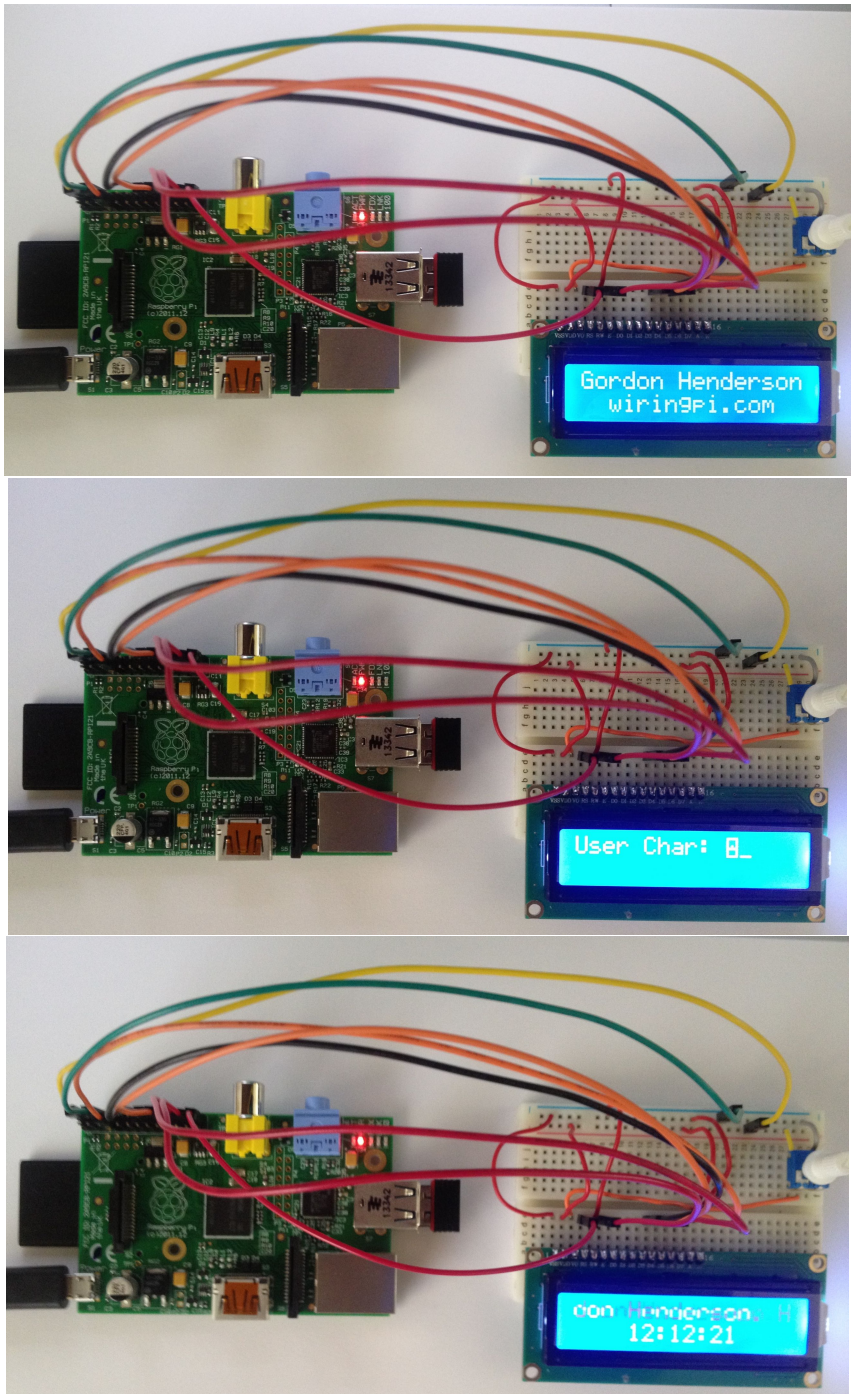


Figure 25: Test code lcd.c running on LCD: top is initialization, middle is user character input, and bottom is left-scroll text with static clock.

5. Video for the terminal activity and LCD output can be viewed here: https://www.youtube.com/watch?v=ELeGJlYxqrI

Modify LCD Program to Accept Push Button Input

1. Next, the lcd.c source file is modified to take a push button instead of key entry input. The circuit constructed above is augmented with a push button circuit, where the state of the push button is connected to GPIO at wiringPi pin number 1.

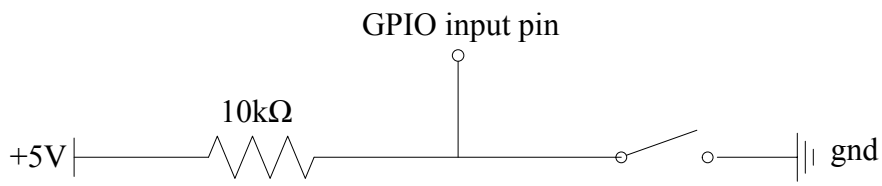


Figure 26: Circuit schematic for push button connection to RPi GPIO pin.

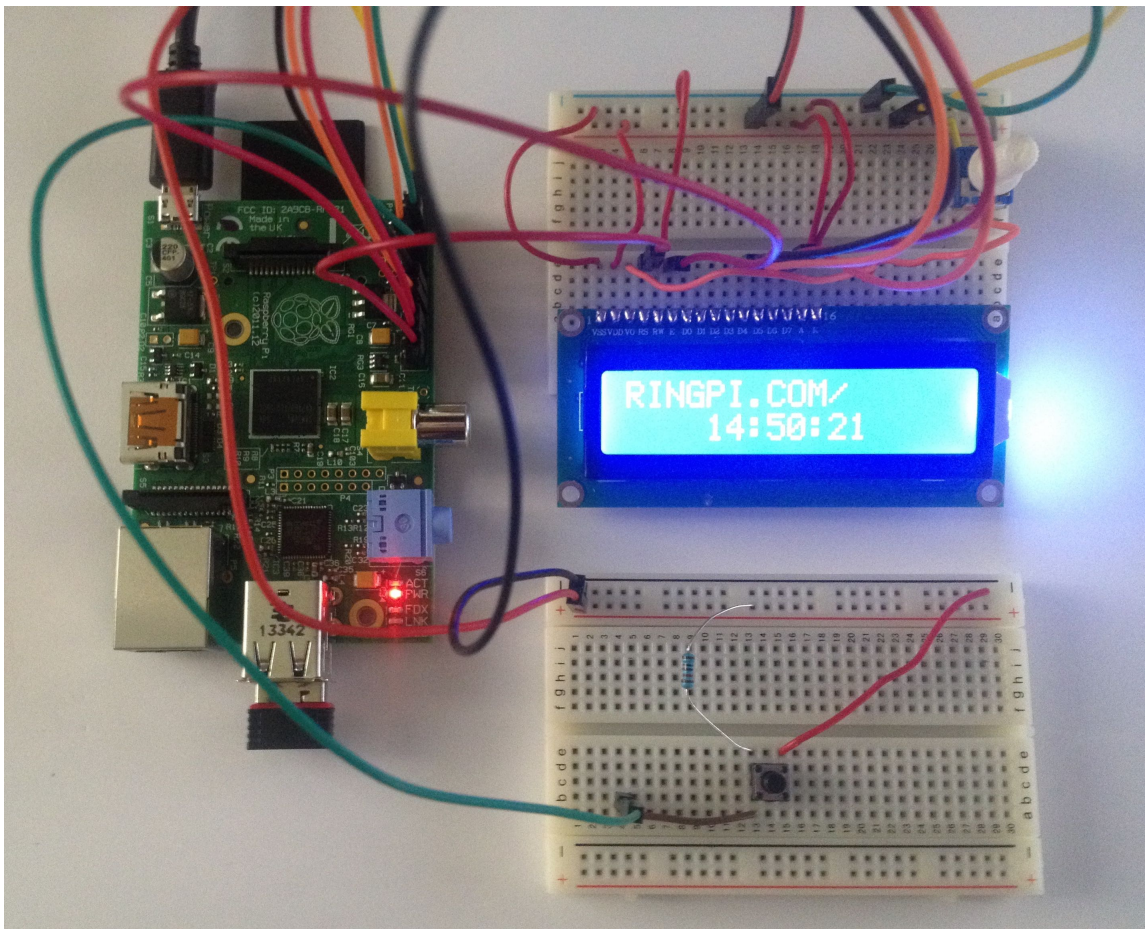


Figure 27: LCD circuit with push button connected.

2. The LCD output for this section is identical to the previous section; in order to see the actual interaction of the button with the circuit, please see the video at the following link:
http://youtu.be/WJRnKehPsfg

Modify LCD Program to Output Student Names to Display

1. In this section, the lcd.c source file is modified to not only accept push button input, but the output to the LCD is changed to show information about the group members. This data left-scrolls across the top row, and the time is statically positioned at the bottom row.
2. Again, the physical circuit is exactly the same as the one in Figure 27.
3. For viewing the output, please see the video at the following link: http://youtu.be/xCc53KH8jmc

Connecting Ultrasonic Sensor

1. As a last exercise, the ultrasonic sensor is connected to the RPi and the sonar.c test file is run.

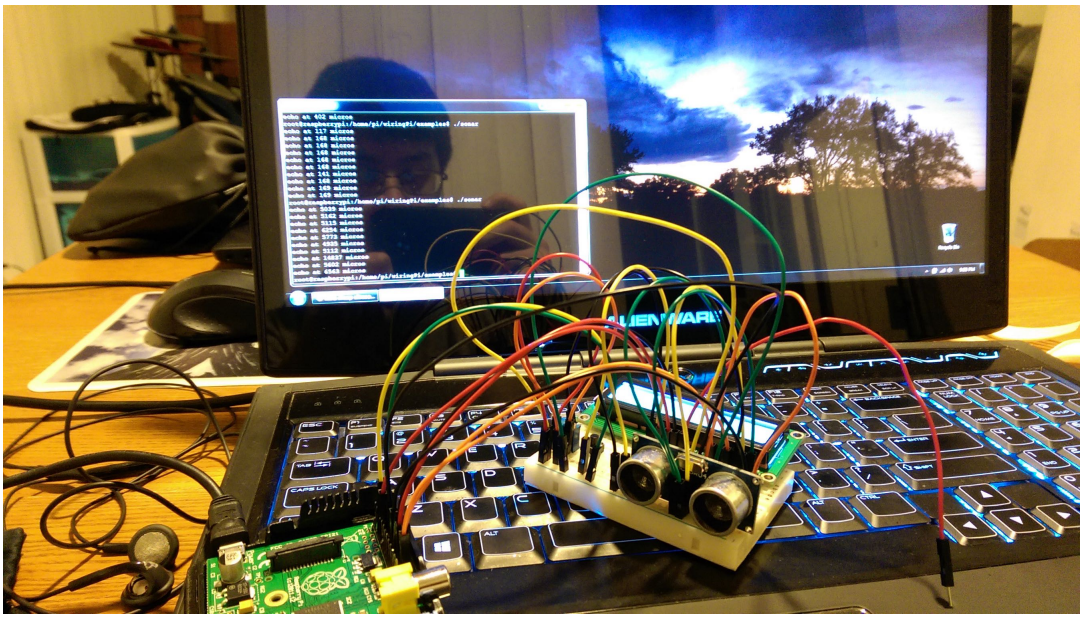


Figure 28: RPi with ultrasonic sensor connected and sonar test executable running.

Final Project Proposals

Ryan Morrison

1. 16 band audio spectrum analyzer. Connect LCD and microphone to RPi and use code that implements FFT to display a dynamic graphic display of sound presented to mic.
2. Digital compass. Connect LCD and compass module to RPi and use code that dynamically outputs the orientation on the display.

Jonathan Lam

1. Binary calculator. Use LED arrays and push buttons to illuminate binary patterns and perform logic functions such as AND, OR, XOR, etc.

Esteban Gomez

1. Computer user recognition system. Use RFID/NFC chips connected to RPi which reads a user chip and authenticates/logs the user in with settings, open programs, themes, etc. specific to that users profile.

References

Raspberry Pi Website: http://www.raspberrypi.org

Info on RPi and Remote Access: http://elinux.org/RPi_Remote_Access

Info on RPi and VNC: http://elinux.org/RPi_VNC_Server

Soldering Tutorial: http://www.aaroncake.net/electronics/solder.htm

WiringPi Website: http://wiringpi.com

LCM1602A LCD Data Sheet: http://www.adrirobot.it/datasheet/LCD/PDF/LCM1602A.pdf

Ultrasonic Sensor: http://rasathus.blogspot.co.uk/2012/09/ultra-cheap-ultrasonics-with-hy-srf05_27.html

Ultrasonic Sensor Data Sheet: http://www.micropik.com/PDF/HCSR04.pdf