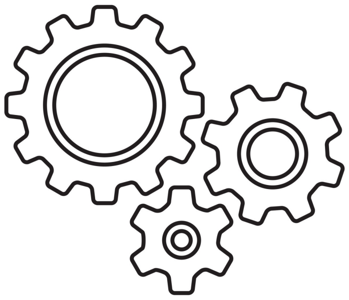
Assignment 1 - Configuration

Andrew N. Sloss

January 5, 2025



Abstract

This Assignment outlines the **Configuration** stage. The goal is to build a complete *Embedded Development Environment* for your *Raspberry Pi 3, 4 or 5.* This environment involves setting up the *Raspberry Pi* hardware platform and all the associated software. The aim is to have a *headless system* (a system that operates without a monitor, graphical user interface (GUI) or peripheral devices, such as keyboard and mouse). Later on, you can use all these features for the other assignments depending upon interests and goal.

NOTE: recommend noting all errors in the instructions. As procedures change over time.

Contents

1 Resources

- Raspberry Pi Getting Started Guide
- Download: Raspberry Pi Firmware Whitepaper

2 Objective

Configuration stage: The goal is to build a complete Embedded Development Environment for your *Raspberry Pi*. The goal involves setting up the hardware and all the associated software. Assignments 2 and 3 rely on a working platform. All the core components must be working properly. Use this assignment to become familiar with the board, i.e., understand the essential terms, hardware, and software.

2.1 Estimated Time

It should take between 2 to 3 hours to complete this assignment. Make sure you read the assignment carefully before setting up the board.

2.2 Terminology

Device	An external piece of hardware
Peripheral	An hardware function built either on the SoC or board
Distribution	A wrapped collection of Linux software, packages and management
HAT	Hardware Attached to Top, normally via the 40-pin header
HAB	Hardware Attached to Below, normally via PCIe + Raspberry Pi 5 only
Firmware	Boot firmware design to boot Raspberry OS

2.3 Nomenclature

$\mathbf{P}_x$	Pin X on the 40-pin Raspberry Pi header.
RPn	Raspberry Pi n = 3, 4, or 5.

3 Recommendations

This assignment starts the exploration process. Afterward, you will have a basic understanding of the hardware and software components. Attempt to record the procedures and observations. Use this information to write a short, structured scientific report outlining all the setup procedures; see Section 12.1. A helpful approach is to consider the report as *if you had to return to the board six months later* could you use the information to re-configure the device. In other words, what were the repeatable steps? Record the time spent on each activity and note any difficulties or gaps in understanding you encountered.

Make extensive use of the book *Exploring Raspberry Pi* by Derek Molloy. The book is getting old but contains some helpful information and advice. **Recommend ignoring Chapter 7 since online sources are more current and providing methods.**

Recommend that you make a list of useful Linux commands and references as you use them since you will likely revisit those commands frequently throughout the course. Optionally add those commands to your submitted report.

Finally, use the message board. We will provide a *Slack* group for everyone to communicate. This group is helpful if you have any issues or have questions. Even though the homework is individual, you can work as a team. Recommends using the group to help solve problems and, most importantly, have fun exploring :)

4 Feedback

If you see an error or anything that requires further explanation or is plain wrong, please let us know so that others may benefit from your observations. Engineering is about constant improvement.

5 Important

Don't pull the power from the device. The system is a Linux device; suddenly removing the power without informing the *Operating System* can potentially corrupt the filing system. Interesting aside, the ext2 file-system we will use is a filing system with journaling turned on, which has a measure of fault tolerance. The best practice is to force Linux to halt (see below) first and then pull the power. The halt forces Linux to be in a dormant safe state for power removal. Use this command before removing the power but always remember to wait until Linux has fully shutdown.

```
$ sudo halt
```

Be aware that the pins on the Raspberry Pi are susceptible. It is easy to damage the board if you connect the wrong electrical inputs to a pin or even static discharge. You might consider adding a **HAT** extension board to protect the *Raspberry Pi* against potential future damage.

6 Check-list

Make sure you have access to the following hardware components.

6.1 Course equipment

1. Raspberry Pi board
2. Power supply compatible with the Raspberry Pi model
3. microSD Card recommend at least 16GB (method of programming)

6.2 Standard Equipment

1. Wireless router (plus Ethernet ports) connected to the internet
2. Ethernet cable
3. HDMI monitor and Cable
4. Laptop/desktop with Ubuntu, Windows or Mac OS
5. USB Keyboard and mouse

6.3 Procedure

1. Create boot/installation media drive
2. Install an Operating System
3. Set Console Auto Login
4. Update and upgrade all the software
5. Configure the Wi-Fi
6. Change target name
7. Install ssh

7 Initialization

Install the *Raspberry Pi Imager* (link provided below), and follow the guidance on how to install the *Raspberry Pi OS* onto a microSD card. Install the latest *Raspberry Pi OS 32-bit ”Bullseye”* version. This should make installation a lot easier. You may like to take some time looking at the other images available.

https://www.raspberrypi.com/software/

This method allows you to skip many stages.

7.1 Install an Operating System

At this point you should have **Firmware plus OS** on an microSD card. Before connecting up the power-supply to the Raspberry Pi, place the firmware installation microSD Card into the SD slot on the board (make sure it fits correctly, can be a little tricky) and make sure you have a working Ethernet cable attached or WiFi. The ethernet cable is critical for a faster updates. Connect the USB keyboard and mouse and finally use the micro HDMI cable to connect a display monitor.

Supply power to your Raspberry Pi; you should see the firmware boot screen on your monitor. Follow the procedures to install the OS. Select English (US) and US layout keyboard (this can be re-configured later, but it is easier to do it at this stage). The installer will now install the OS onto the microSD Card. This process normally takes up to 10 minutes, depending upon the size of the microSD card chosen. Once complete, the Raspberry Pi will reboot into a full GUI screen. Complete the install procedure in the GUI.

7.1.1 Set Console Auto login

You should now have a complete working Linux operating system running on your target hardware. Next is to remove the desktop GUI so it can operate as a headless Embedded Device, still in the GUI bring-up a terminal window and type in the following command:

```
$ sudo raspi-config
```

Note: you can also do this through the Desktop configuration tool.

This will bring-up another window with a set of options, recommend exploring the options. Choose the *"Boot Options"* and then choose *"Console auto login"*. You can choose just *"Console"* without auto login but that slows everything down because you will have to login in each time you reboot or power-on the board. Remember to save the modifications. If you choose to use login the default is username *pi* and password *raspberry*.

For any reason you need to go back into the GUI (e.g. use *raspi-config*) use the following command. Make sure you have a keyboard, mouse and monitor connected.

```
$ sudo startx
```

Now reboot the board from the desktop menu option.

7.1.2 Update and Upgrade all Software

You should now see the Raspberry Pi rebooting and if auto login is set then system will take you straight into a terminal command prompt. Now we need to update and upgrade the software.

Recommend searching for how to *update* and *upgrade* the software. Worth noting the differences. Also there is a way to update the boot firmware.

7.2 Communication

You should now have fully updated software running on the target hardware. The Raspberry Pi 1 and 2 were more complicated to setup since they didn't include builtin WiFi hardware. For Raspberry Pi 3 on-wards, the WiFi should be already configured for the most part. Communication stage is all about setting up the WiFi and target board name. WiFi allows you freedom to place the board in different locations without requiring a physical connection which makes it easy to work with.

7.2.1 Configure the WiFi

You can configure through the Desktop UI (User Interface) or follow the instructions below. Make sure that the USB Wi-Fi is working. Check that you can see that the hardware is connected. Using the command line:

```
$ dmesg | more
```

This command lists all the devices available on the target hardware. Once you have found that the WiFi is available, configure the software by modifying the *interfaces* file using the following command (it is also possible to configure this file using the desktop GUI and connecting to a WiFi network):

```
$ sudo nano /etc/network/interfaces
```

The *nano* editor is a simple and easy way to edit files on the Raspberry Pi (note *vi* is also available). The following worked for me but it can be fiddly:

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
auto wlan0

iface wlan0 inet dhcp
   wpa-ssid "Your Network SSID"
   wpa-psk "Your Password"
```

NOTE: *if you can work out an easy method to connect to multiple WiFi networks, I would be interested.*

For more details, recommend following the following link:

http://raspberrypihq.com/how-to-add-wifi-to-the-raspberry-pi/

7.2.2 Changing target name

Recommend that you change the name of your device. This is good practice since you may want to have more that one device on the network.

```
$ sudo nano /etc/hosts
```

Within the file change the line which has **127.0.1.1 raspberrypi** to something like **127.0.1.1 ALICE0**. Remember to save and exit. Then carry out a similar procedure by modifying the *hostname* file.

```
$ sudo nano /etc/hostname
```

Within hostname file change **raspberrypi** to **ALICE0** and then save, exit, and reboot.

```
$ sudo reboot
```

After reboot the target should now be operational with the new name. For more information on changing the target name refer to the following link:

http://www.howtogeek.com/167195/how-to-change-your-raspberry-pi-or-other-linux-devices-hostname/

7.3 Applications

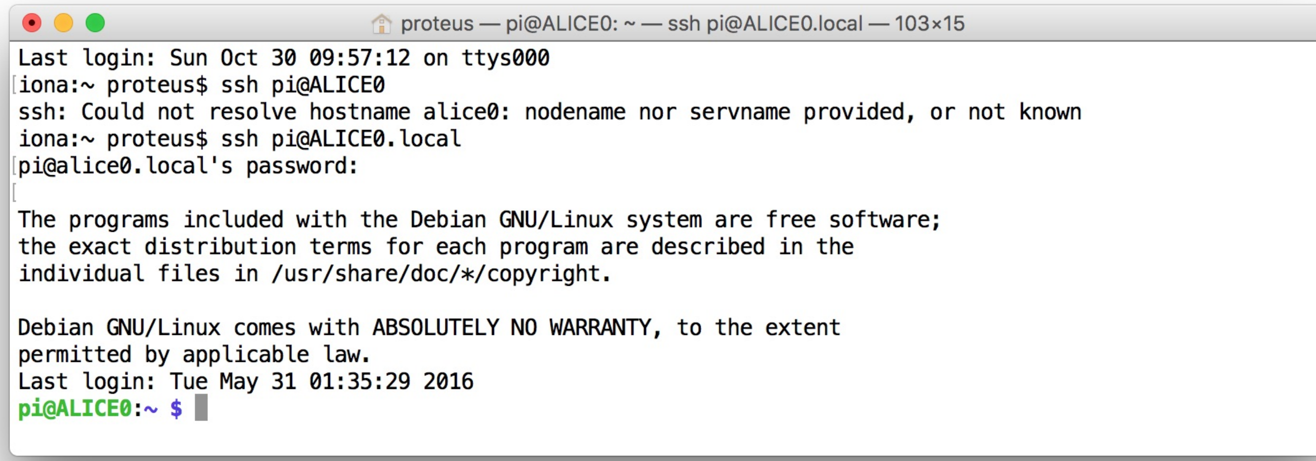
The Application stage is more about getting the various software pieces in place to move files to/from the *Raspberry Pi* to the host-computer (your development machine. Traditionally the focus has been on *Telnet* and another tool called *Samba* but today *ssh* is becoming more important and the standard method. Just as a background, *Telnet* allows for remote login which is really useful when configuring and compiling code. And *Samba* is really useful for moving and copying files over to an embedded target. Both useful fundamental tools but both replced by *ssh*.

7.3.1 Install ssh

The *ssh* service should already be running on your *Raspberry Pi*. You can double check by listing out the services running on the board. For *Microsoft Windows* there are many client options but the two most popular clients are **PuTTY** and **WinSCP**. These clients allow you to use *ssh* and transfer files between host-machine and target-board. Even though the examples are with a Mac they should work on a *Windows* just as well.

```
$ sudo service --status-all
```

To get *ssh* operational you have to enable the host side. Example below is getting ssh working on Mac OS, Windows should have a similar procedure. This is a one time setup since a secure certificate is exchanged.

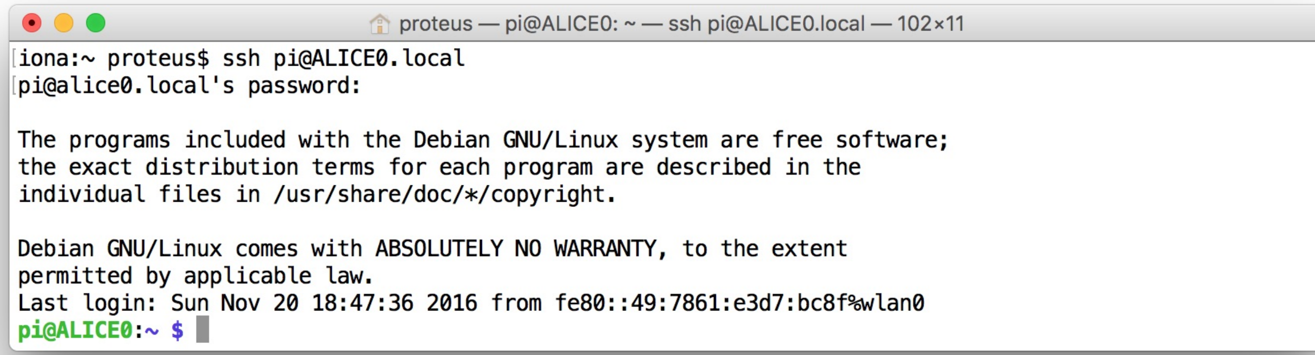


```
Last login: Sun Oct 30 09:57:12 on ttys000
iona:~ proteus$ ssh pi@ALICE0
ssh: Could not resolve hostname alice0: nodename nor servname provided, or not known
iona:~ proteus$ ssh pi@ALICE0.local
pi@alice0.local's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 31 01:35:29 2016
pi@ALICE0:~ $
```

If you exit from the *ssh* connection and repeat the procedure. You should be able to log back into you board and see something similar to the below image. Notice I used *pi@ALICE0.local* as the username:



```
iona:~ proteus$ ssh pi@ALICE0.local
pi@alice0.local's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Nov 20 18:47:36 2016 from fe80::49:7861:e3d7:bc8f%wlan0
pi@ALICE0:~ $
```

You should now have a fulling working *ssh* environment which allows you to directly login into the target board from your host machine. For more information take a look at:

https://en.wikipedia.org/wiki/Secure_Shell

8 Verification

Verification is the process to discover the truth, accuracy and validity. Create a file on you host PC call it something like *hellow.c* (basically a standard hello world test program), write the following code:

```
#include <stdio.h>

int main(void)
{
printf("hello world\n");
return 1;
}
```

From you host PC use samba to copy over the files to the Raspberry Pi. Once the file is safely across, log into the Raspberry Pi over telnet and go to the directory where the file was placed and compile *hellow.c* using the following command.

```
$ cc hellow.c
```

If no errors have occurred this should create an image called *a.out* which can then be executed using the following command:

```
$ ./a.out
```

Check that the program runs. If there are any errors or issues, try to correct them. Once you have satisfied yourself that everything is functional then this assignment is complete.

```
iona:~ proteus$ ssh pi@ALICE0.local
pi@alice0.local's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Nov 20 19:29:18 2016
pi@ALICE0:~ $ nano hellow.c
```

```
pi@ALICE0:~ $ cc hellow.c
pi@ALICE0:~ $ ls
a.out     Developer  experimentsl  noworkload.dat  Public        Videos
Desktop   Documents  hellow.c      oldconffiles    python_games
Dev       Downloads  Music         Pictures        Templates
pi@ALICE0:~ $ ./a.out
Hello World
pi@ALICE0:~ $
```

Listing 1: File: strlen.c

After running *a.out* successfully, recommend replacing *cc* line with *gcc hellow.c -o hellow*. This will name the executable output *hellow* instead of the standard *a.out*. Execute *hellow*, you should see exactly the same output.

Once complete get familiar and comfortable with some of the other commands. Explore both *ls -l* and *ls -lh* list directory information. And other commands such as *df*, *df -h*, *du*, and *du -chs*. Note *du -chs* is powerful command to get detailed folder information.

9 Experiment

This year there is an experiment. You will have to change some code to get the code to work. Each board has a manufacturer ID. You will need to add the Raspberry Pi code for your board. Recommend working together to solve this problem.

The function that needs updating is $void\,prototype\_translate\_information(char * model)$

- Download *ee_course.zip* from Canvas: located $files/Assignments/ee\_course.zip$
- Copy the zip file to the target board, e.g., *scp ee_course.zip pi@"target".local* : .
- Logon to the Raspberry Pi, create a new directory, i.e., *mkdir experiment*
- Move the zip file to *experiment* directory, i.e., *mv ee_course.zip experiment/.*
- Change directory, i.e., *cd experiment*
- Unzip, i.e., *unzip ee_course.zip*
- Install Gnuplot, i.e., *sudo apt install gnuplot*
- Execute the shell script, i.e., *./run.sh* (should produce some graphs after 10 to 15 minutes)
- Note: you will need to add some code to *prototype.c* to support your board (work with the group)

10 Submission

Submit an engineering report which includes the objective, the procedures you used to configure your board (with window captures as proof of working configuration), and finally, a conclusion with references. Recommend listing down all the challenges and difficulties as well as the time-taken to achieve the activity.

For more a stretched goal, try to get an IDE working through *ssh*, and set up a source control system.

10.1 Completion

On completion you should have the following:

- A working *Raspberry Pi* booting in headless mode.
- A cross-development environment.
- Ability to transfer code + data to and from the *Raspberry Pi* and a host machine.
- Transfer the experiment to the target machine, correct the code, and build + run.
- Optional (recommended), IDE on the host machine.
- Optional (recommended), a source control system to save your work.
- A report recording all your discoveries and process.

11 Assessment

Total marks for this assignment is **(10)**.

The assignment will be assessed as shown below.

- **(0)** Optional, provide a short paragraph on your Embedded Systems experience. This additional piece of information can help assist us.
- **(3)** Points given for completing the exercise.
- **(3)** Points given for the quality of the report. Make sure you leave enough time to complete the report. The report should include a clear objective, a list of tasks carried out (including difficulties encountered), a list of useful instructions, and a conclusion with all the sources you used as references.
- **(2)** Points given for executing the Experiment (see Section 9). Adding the graphs to your report.
- **(2)** Points given for improving the programming environment. For example, getting an IDE to work with your target board. Setup an *IDE* such as *Microsoft's Visual Studio*, *atom* or *Eclipse C/C++*. Or add a version control system such as like *GitHub* or *Bitbucket*.

12 Hints-and-Tips

A list of useful *hints-and-tips*:

1. Any doubts use the *Slack channel* to ask questions.
2. Setup *GitHub* or *Bitbucket* account as soon as possible once you have a stable environment. I don't want any loss of work.
3. If the system is acting strange, recommend rebooting the entire board, e.g., *sudo reboot*
4. If you consistently encounter strange problems or corruption, recommend reformatting the SD card and starting again.
5. If *Raspberry Pi* is unstable, check that the power supply is correct. A wrong power supplies cause the board to reboot randomly.
6. Be careful with the pins; if wrongly configured, the Pins can destroy the board.
7. At the very start, ensure the keyboard and language are *English US* and *US layout.*
8. Always use $ *sudo halt* to shutdown the *Raspberry Pi.* Make sure the software has completed the shutdown cycle before pulling the power.

12.1 Example, report structure

Provided just for reference. This is generic report structure.

1. Title Page:
   - Title of the report
   - Author(s) names and affiliations
   - Contact information for corresponding author
   - Date of submission
2. Abstract:
   - Brief summary of the study, including the research question, methods, results, and conclusion
   - Usually limited to 150-250 words
3. Introduction:
   - Background and context of the study
   - Statement of the research problem or question
   - Objectives or hypotheses
   - Brief literature review to provide context
4. Methods:
   - Detailed description of the research design, methodology, and materials

- Clear and reproducible procedures
- Statistical methods, if applicable

5. Results:

   - Presentation of the data, usually in the form of text, tables, and figures
   - Avoid interpretation at this stage; focus on presenting the facts

6. Discussion:

   - Interpretation of the results and their implications
   - Comparison with existing literature
   - Limitations of the study
   - Suggestions for future research

7. Conclusion:

   - Summarize the main findings
   - Emphasize the significance of the results

8. References:

   - Cite all the sources referred to in the report using a consistent citation style (e.g., APA, MLA, Chicago)
   - Include books, articles, and other relevant sources

9. Acknowledgments:

   - Recognize individuals, organizations, or funding agencies that contributed to the research

10. Appendices:

    - Include additional information that is too detailed for the main body of the report
    - Supplementary data, questionnaires, code, etc.

11. Figures and Tables:

    - Numbered and labeled appropriately
    - Captions should be informative and self-contained