Raspberry Pi Workshop

Daniel Axtens
daniel@axtens.net
@daxtens

4 October 2013

What we'll cover

Introduction

Applications

- A go-anywhere computer
- A smart, simple electronics controller

Conclusions

Introduction

- What is a Raspberry Pi?
  - Credit card sized, $35 computer.
  - Not a ‘conventional’ computer—ARM processor (like many phones). Cannot run Windows. Usually runs Linux.
  - Otherwise full-featured: HDMI, USB, ethernet, etc.
  - GPIO (General Purpose I/O): easy to connect to electronics.
- How do I use it?
  - Topic of this presentation!
  - Huge amounts of resources online. (Including this presentatation, slides and all source code: TODO).
- Why should I care?
  - Enables simple development a whole class of applications and devices that would otherwise be an enormous amount of work and expense.

Welcome to Raspberry Pi

- Plugging in
  - Requires SD Card to do *anything*. No beeps/information if the SD card is stuffed.
    - **If it doesn't work, suspect the SD card.**
  - Power is via Micro-USB.
    - No data pins — cannot communicate over the SD port.
    - Surprisingly voltage-sensitive.
- Booting up
  - You'll see scrolling text, and then either a command line or a graphical user interface.
  - Your machines have been set up with the **Raspbian** Linux distribution.

Setup: `raspi-config`

- Keyboard Layout - set it to US not UK or # won't work.
- You have a choice of booting into GUI or CLI.
- The username / password is `pi` / `raspberry`, unless you changed it.
- Reboot when done.
- If in CLI, `startx` should take you to GUI.

Explore!

- Things have different names to what you might be used to.
- Things are a bit slow: RPis do not make good general purpose computers!
- If you're googling something and you don't get any results for Raspbian, try your search terms with Debian instead of Raspbian.
  - Debian is the base system for Raspbian.
  - Versatile and widely used Linux distribution.

Getting on the internet

- Ethernet - should automatically pick up an address.
- At ANU, proxy will make your life hard.
  - `iccache.anu.edu.au` port 8080 (?)
  - Several places this may need to be set.
- Some 3G modems will work. You will need (at least) the packages `ppp` and either `pppconfig` or a graphical tool.
- Packages can be installed with `apt-get` or some graphical tools. Good resources on the internet. Debian Linux based OS: large range of existing cross-platform software, much can be trivially ported.

Applications I: a go-anywhere computer

When might I want a computer? Examples:

- Displays that aren't stupidly frustrating to program and impossible to change once deployed.
- Telemetry over 3G.
- Anything you want to program without being restricted to C or PICBASIC or some other horrible language.
- Upgradability. (Even over-the-air upgrades!)
- Interface with standard USB peripherals without having to write drivers!
- Give your users traditional keyboard/mouse/screen interface without paying for a proper computer.

Applications I: a go-anywhere computer

My group's project: Car-park occupancy

- Take photo
- (Display on screen for debugging purposes)
- Send to server for processing

Packages:

- `streamer` (input), `feh` (display) , `ppp`, `pppconfig` for 3G modem.

Applications II: A smart, simple electronics controller

- RPis can make electronics slightly less painful.
- GPIO (General Purpose I/O) pins allow connection to the outside world. Can be interfaced with programming language of your choice.
- Be careful—very few failsafes/buffers/protections built in.
- See demo!
- So very very much easier than most embedded systems!

Useful resources for electronics

Just a little bit harder to Google than the software side of things...

- Good pin-out diagram: `http://www.hobbytronics.co.uk/raspberry-pi-gpio-pinout`
- Code samples `http://elinux.org/Rpi_Low-level_peripherals`
- Seriously consider a helper library/app, e.g.:
  - RPIO: `http://pythonhosted.org/RPIO/`
  - webiopi: `https://code.google.com/p/webiopi/`

Conclusions

- For ~$50, you can have a computer behind your project.
- Massive increase in flexibility, massive increase in prototyping speed, massive increase in built-in capability.
- Standardised inputs and outputs - 5V USB. 3.3V GPIO.
- Huge volume of resources on the internet.
- These slides, associated source code, circuit diagrams will be available from `http://daxtens.github.io` (from sometime this weekend).