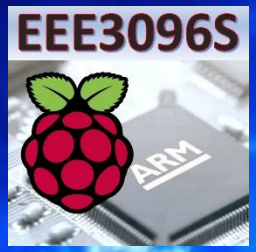
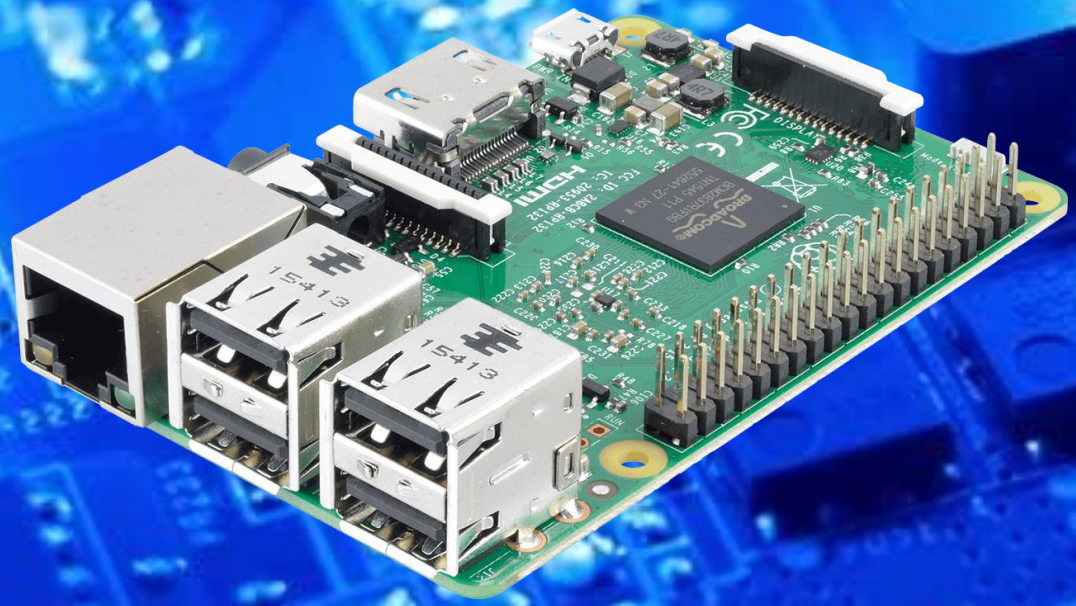




PRACTICE LECTURE P01

P1

Embedded Systems II

Dr Simon Winberg

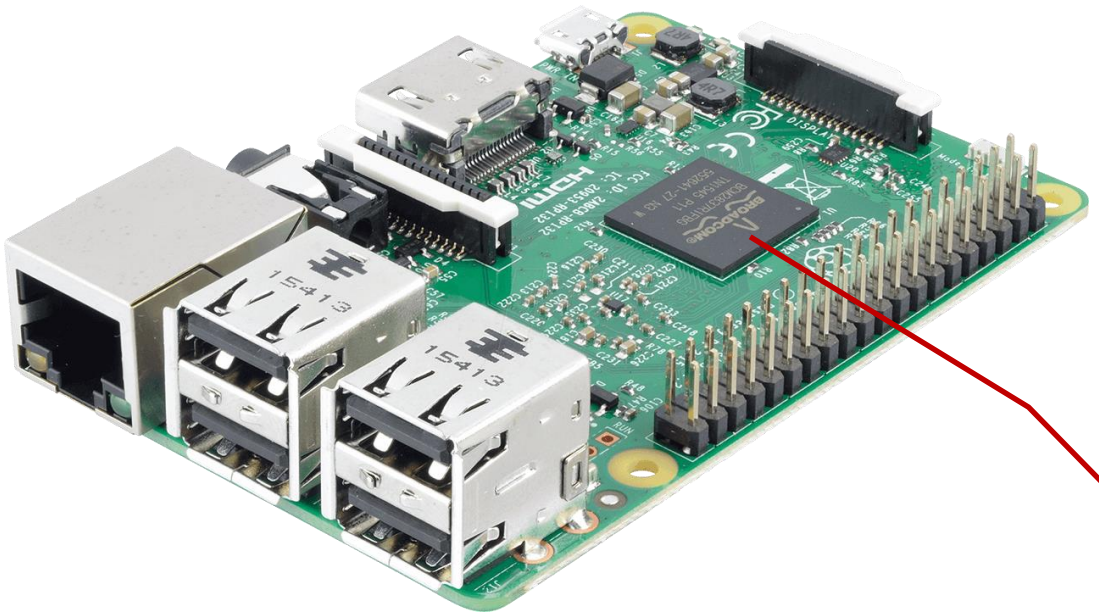
Electrical Engineering
University of Cape Town

Outline of Lecture P01

- Architecture of the Raspberry Pi (briefly)
- What can one do with the Raspberry Pi?
- Setting up the Raspberry Pi
- The boot process and getting it to work
- The headless Rpi
- Getting the RPi into Shape!

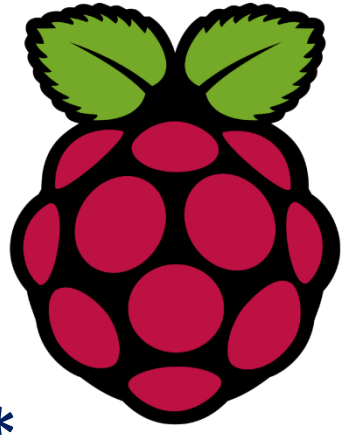
The Raspberry Pi 3

- In pracs (during 3rd term) we will be using the Raspberry Pi 3



It can be called a ‘single board computer’ (SBC) and can also double as an evaluation board for the BROADCOM BCM2837 1.2GHz QUAD Core System on Chip (SoC)

The Raspberry Pi



- *Classic definition:*
  - It's a 'Credit Card sized computer'
- Low priced – around R500 ⸗
- Over 14 Million of them sold to date*
- 50+ versions
- Designed for Education and Hobbiests
- Low Power usage
- RPi 3: 4 Core 64 Bit ARM-based CPU

⸗ For Pi3 *including* import duties and postage & packaging, could be closer to R400 without

* https://www.zdnet.com/article/14-million-raspberry-pis-sold-10-million-made-in-the-uk/

Power of the RPi

It is *more* powerful than the

CRAY supercomputer of 1980s

Aspect	***Cray***	***Raspberry Pi 3***
Price	$8m (1988)	$35*
CPUs	1	4-core
Word size	32 bits	64 bits
RAM	64MB	1GB
Cooling	Air cooled, heatsinks, fans	Air cooled, *no* heatsink or fan

* For basic Pi3 *excluding* import duties and postage & packaging

History of the RPi

- The first generation (Raspberry Pi 1 Model B) released Feb 2012
  - It was followed by a simpler and inexpensive Model A.
- 2014 board with improved design in Raspberry Pi 1 Model B+
  - These are the classic credit-card sized form-factor
- Improved A+ and B+ models were released I 2015
  - Compute module released in April 2014 for embedded applications
  - Raspberry Pi Zero released in Nov 2015 with smaller size and reduced input/output for US$5

History of the RPi

- All models feature a Broadcom system on a chip (SoC), includes an ARM compatible CPU and on-chip GPU (VideoCore IV)
- CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3
- On board memory range from 256 MB to 1 GB RAM
- Secure Digital (SD) cards are used to store the operating system and program memory in either the SDHC or MicroSDHC sizes

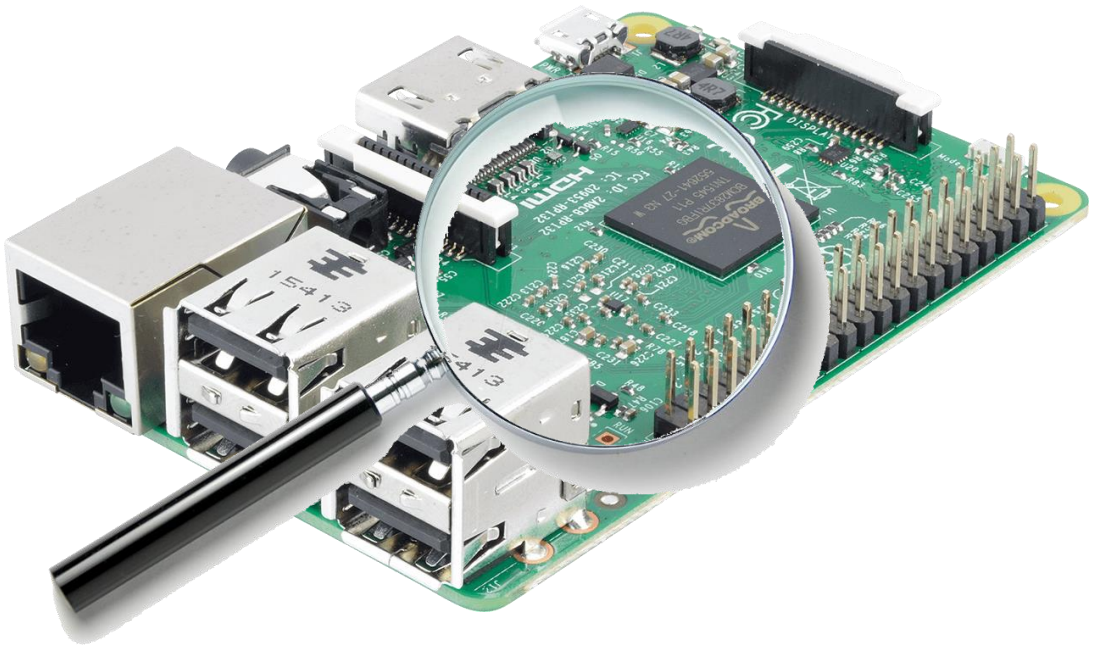
Supported Operating Systems

- The Raspberry Pi Foundation recommends using Raspbian (Debian-based Linux)
- Other third party operating systems are:
  - Ubuntu MATE
  - Snappy Ubuntu Core
  - Windows 10 IoT Core
  - RISC OS
  - Kodi media center

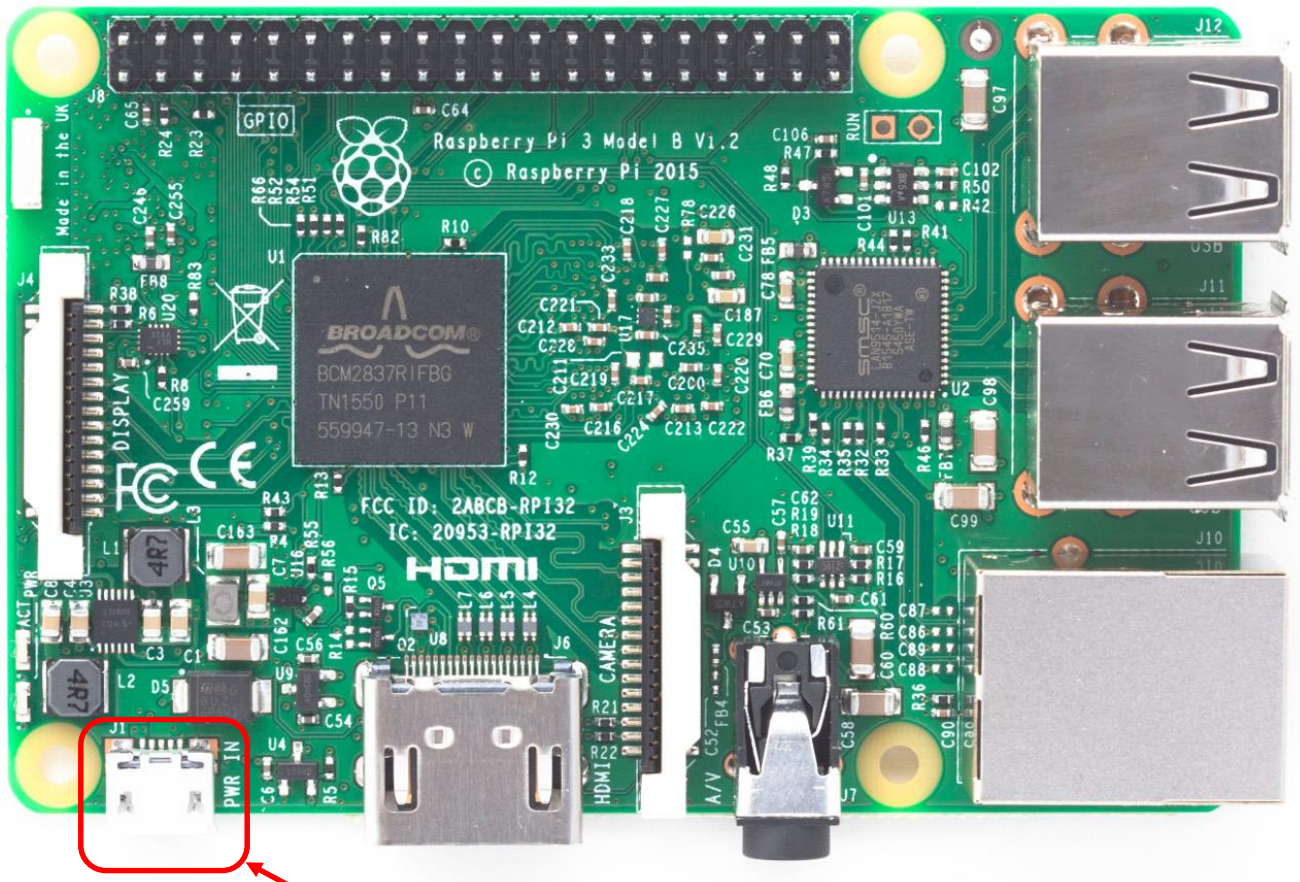
Raspberry Pi Community

- The Raspberry Pi community is one of the most exciting aspects of the Rpi
- The community developed a fanzine (magazine for fans) around the platform called The *MagPi* which in 2015, was handed over to the Raspberry Pi Foundation by its volunteers to be continued in-house.
- A series of community “Raspberry Jam” events have been held around the world

Closer Look at the RPi

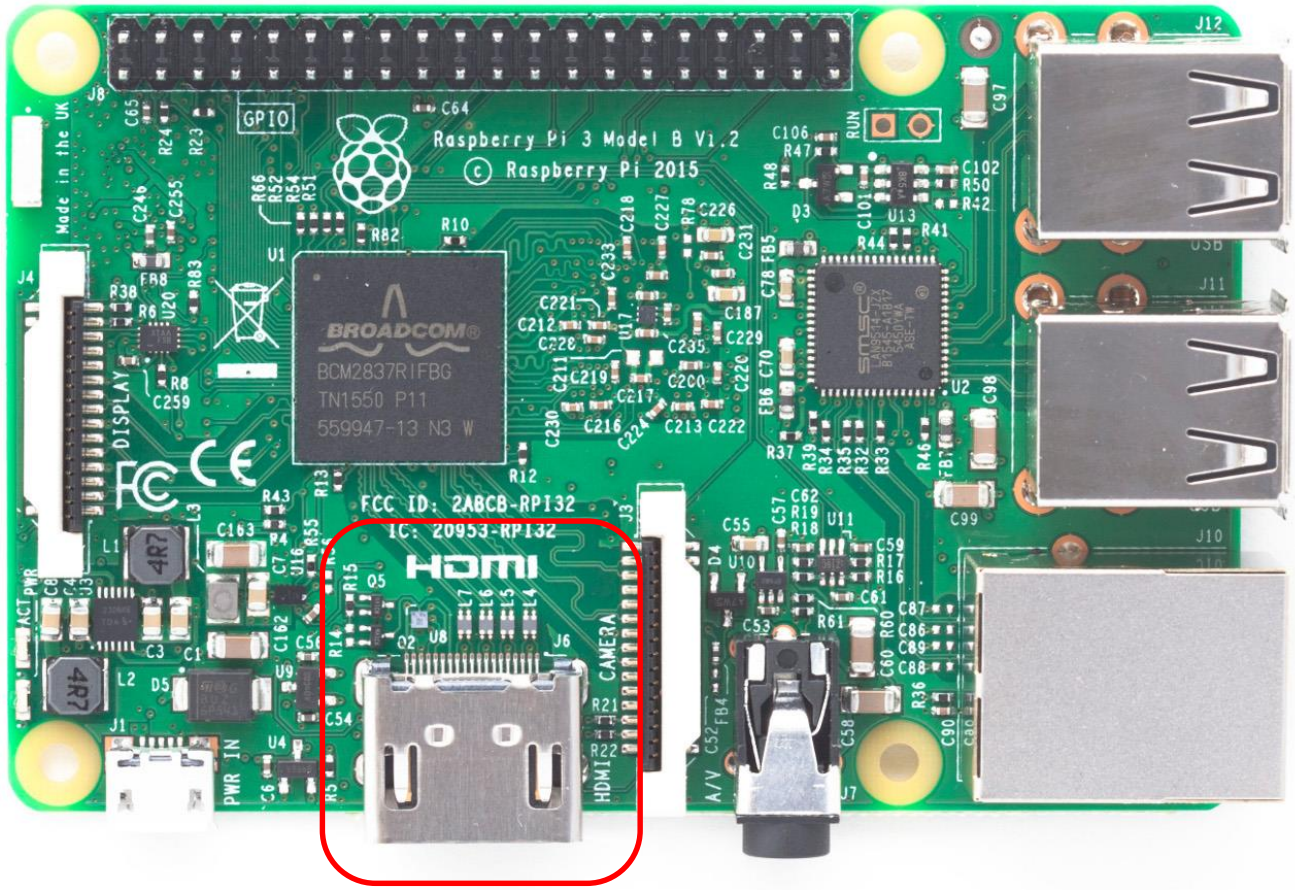


The Basics - Power



5v micro USB power connection, just like your mobile phone!

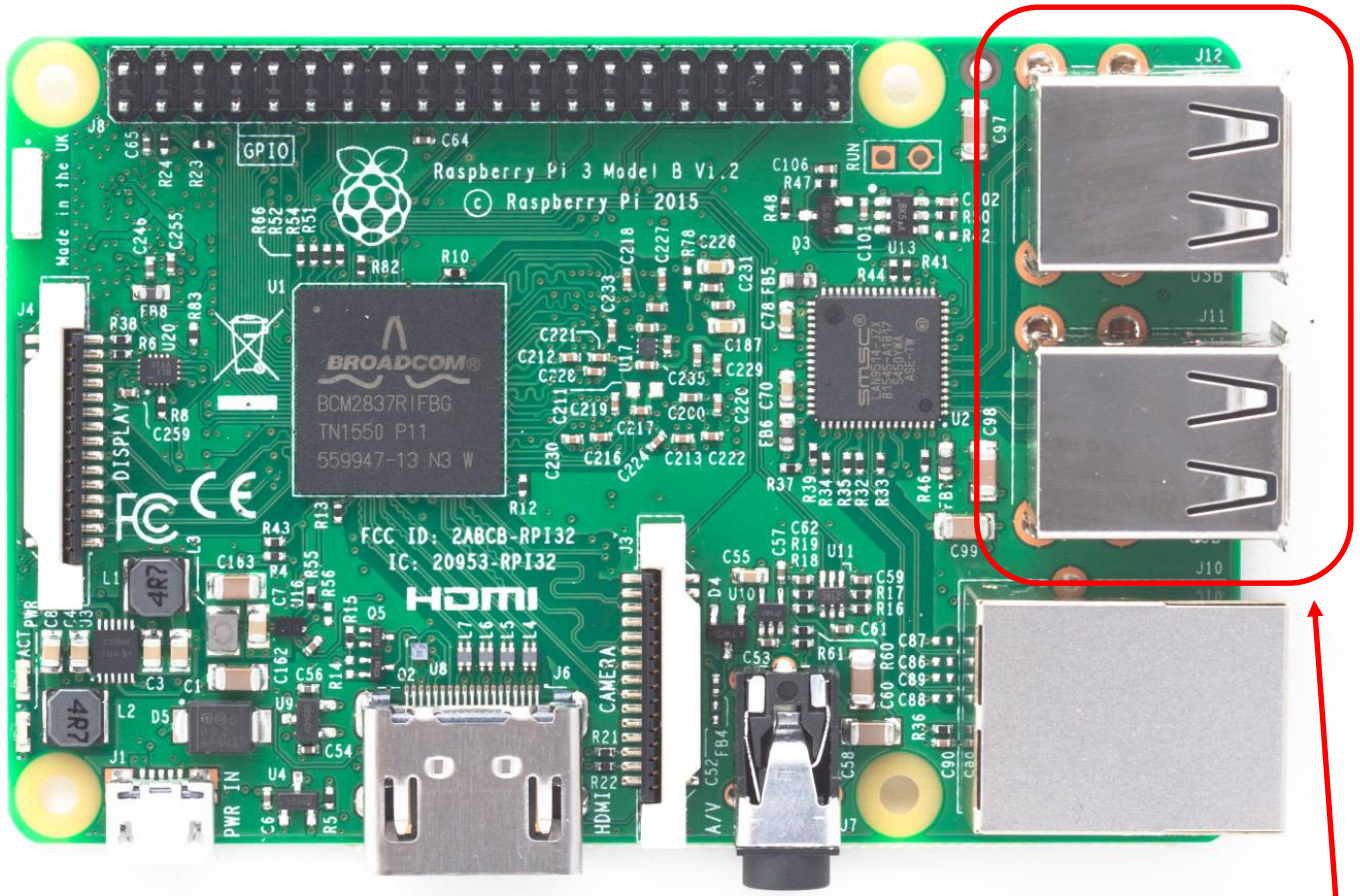
The Basics – Audio Visual Port



We will use this as a backup in case default settings aren't right

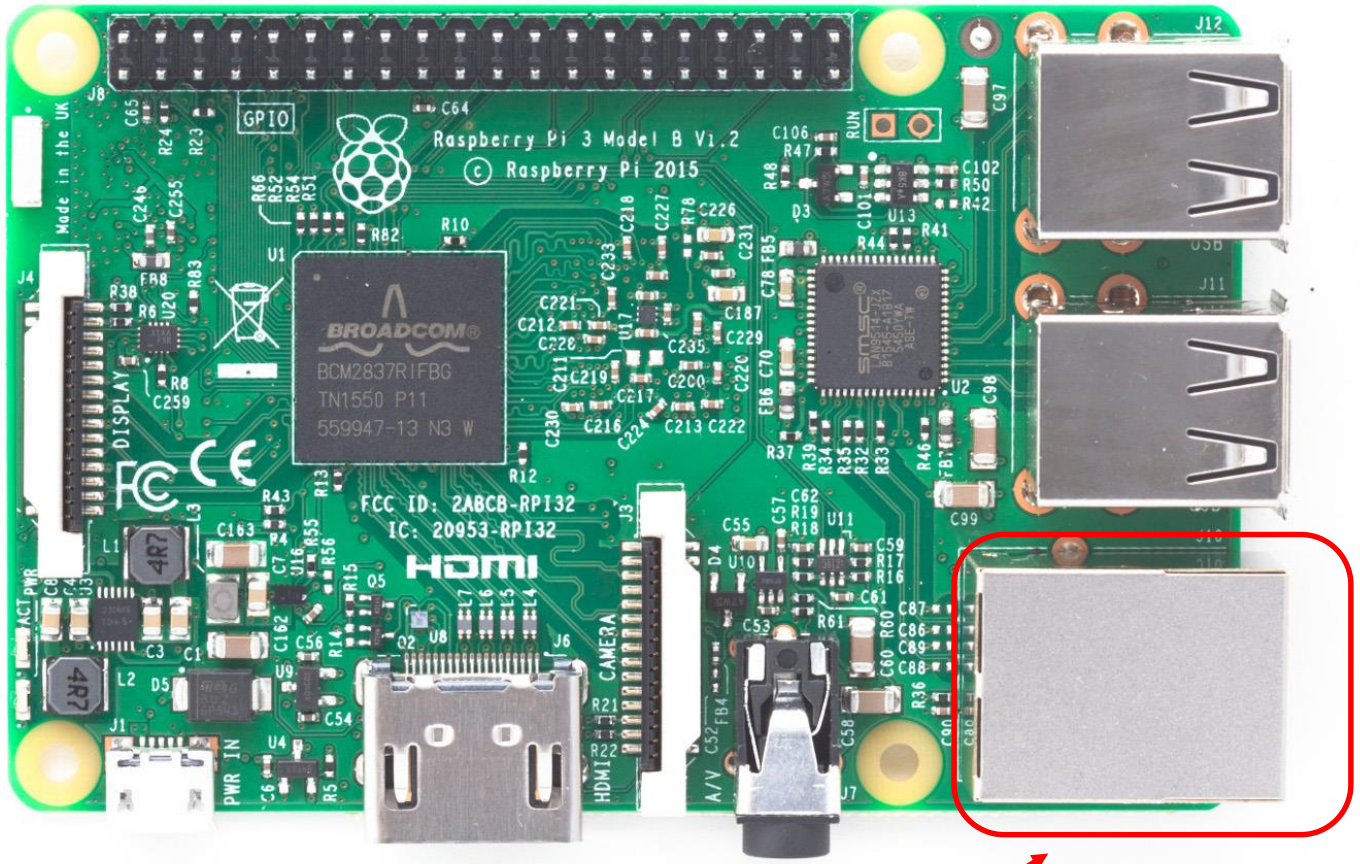
HDMI - Video & Audio - Works with new monitors / TVs

The Basics – USB Connections



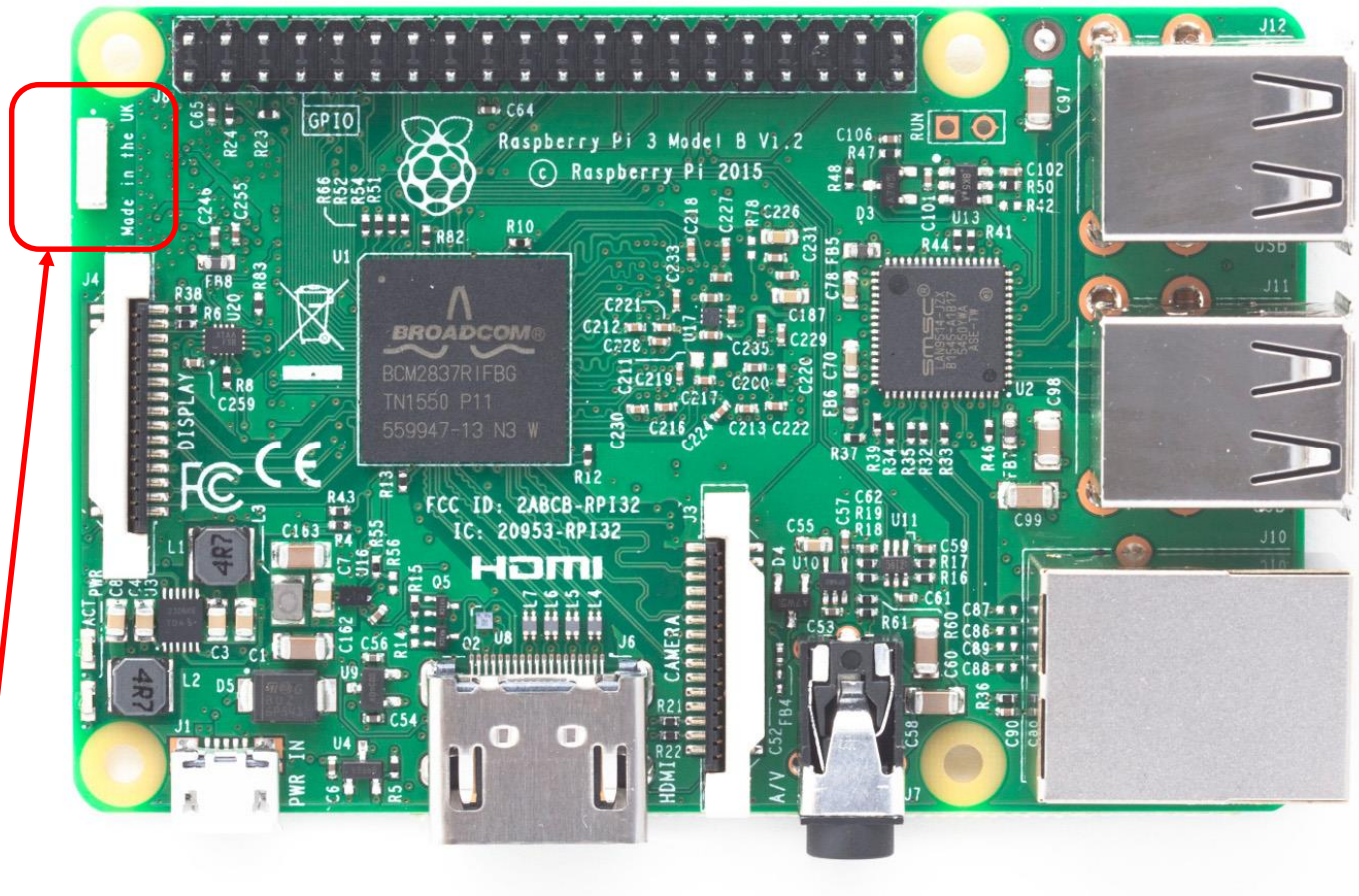
4 x USB 2.0 ports

The Basics – 1x Eth Connections



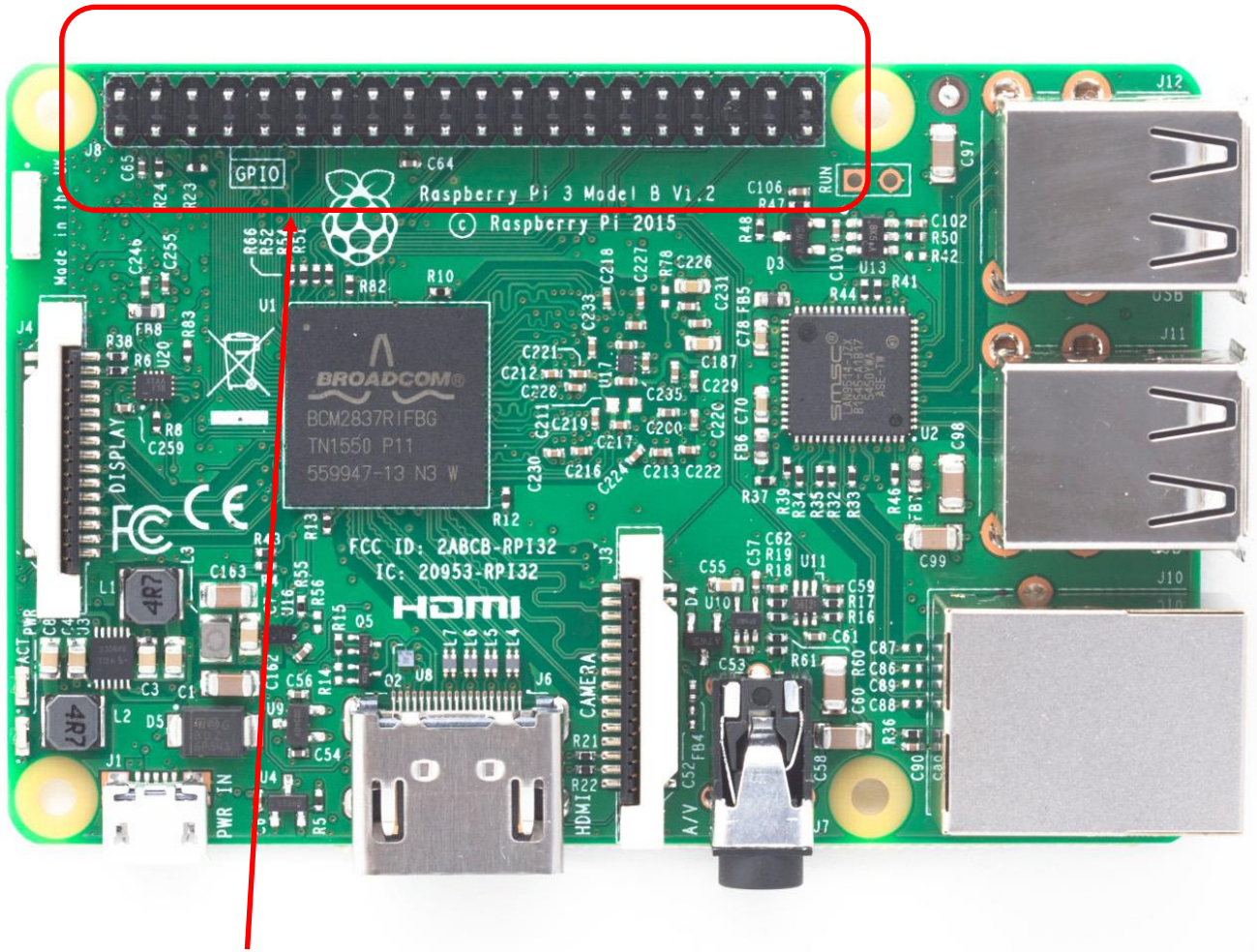
10/100 Ethernet connectivity RJ45

The Basics – Connections



WiFi (2.4GHz 802.11n) and Bluetooth 4.1 wireless

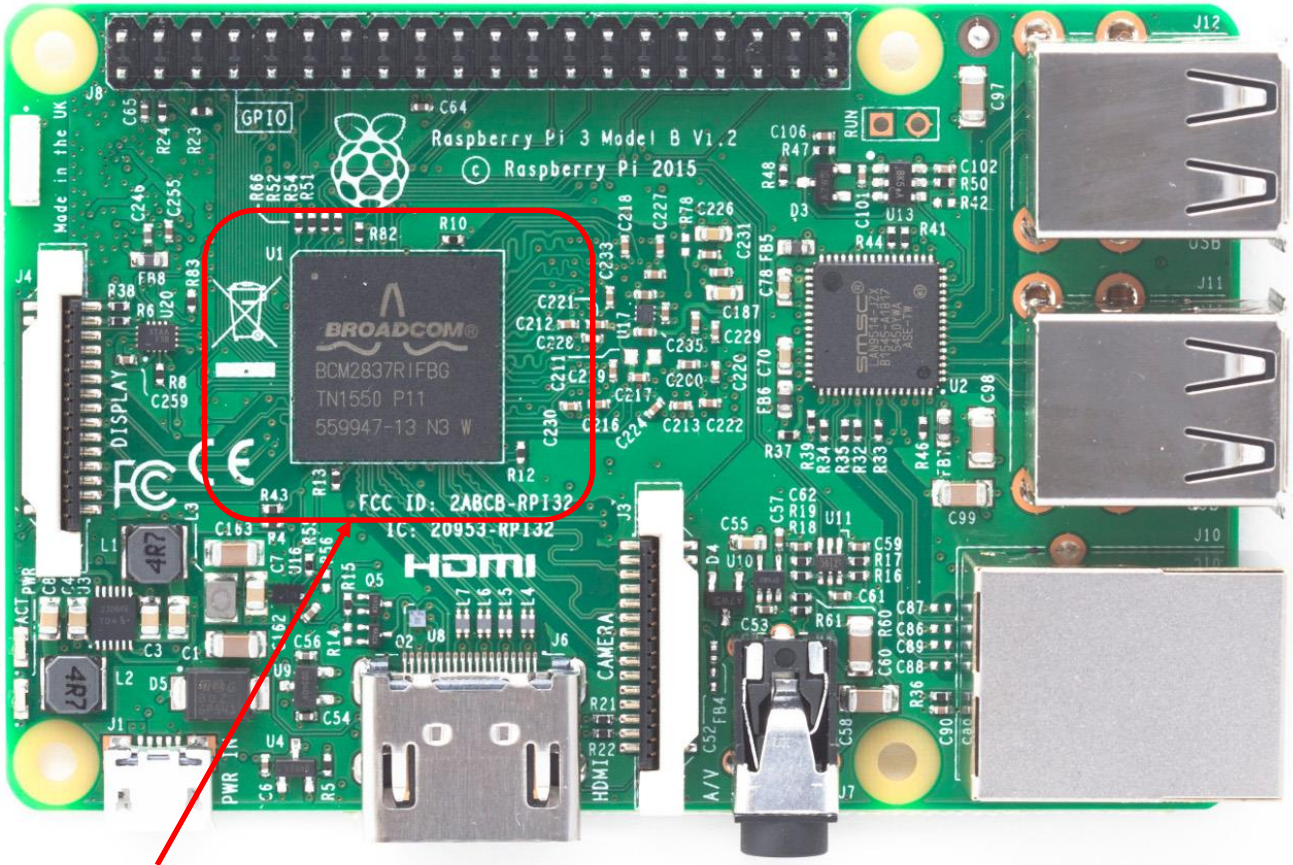
The Basics – Connections



Your biggest friend for the pracs…

40-pin General-Purpose Input Output (GPIO) header

The Basics – Internals

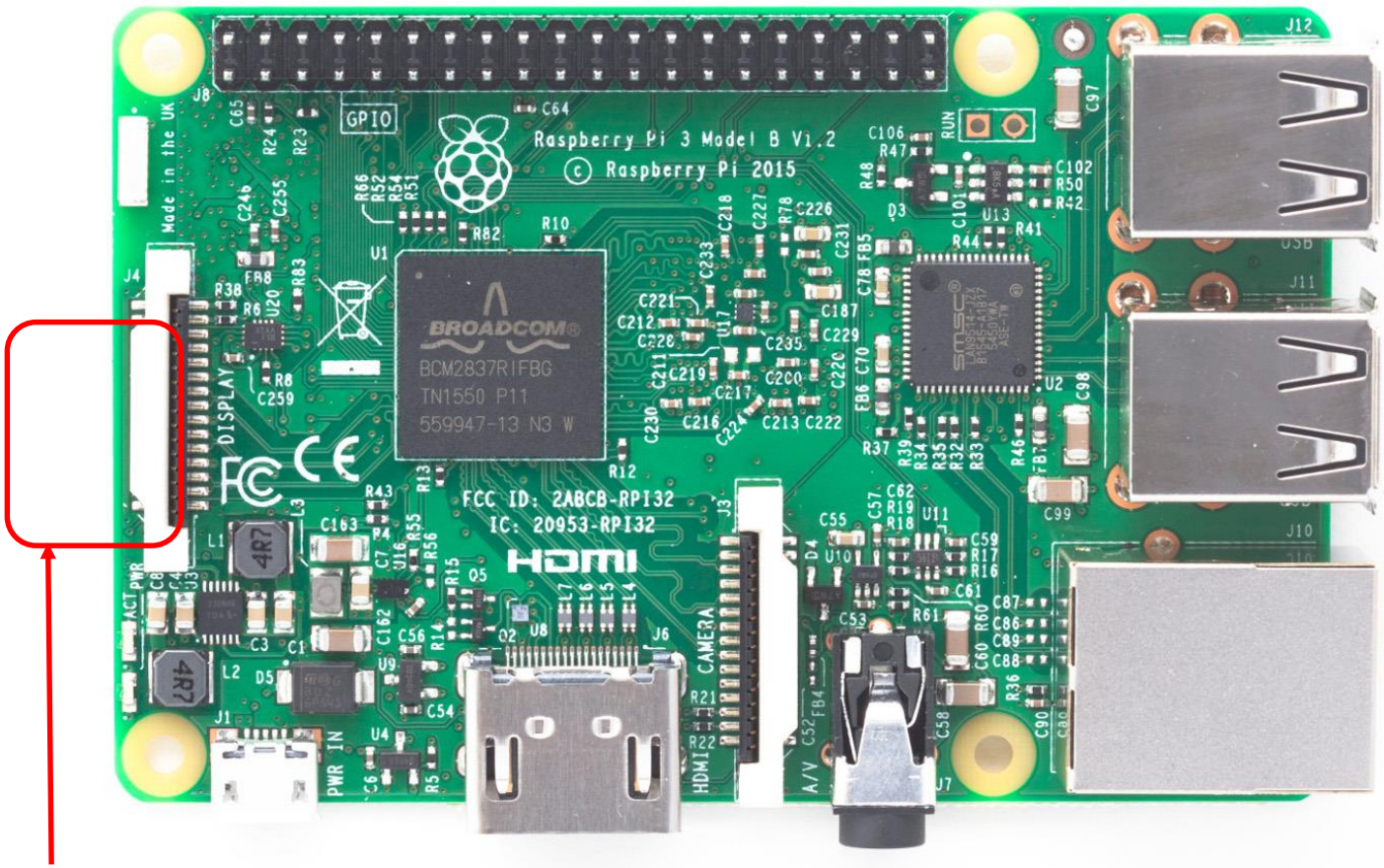


The Serious Business

System on Chip (SoC) Broadcom BCM2837

ARM 4-Core ARM Cortex-A53 at 1.2GHz – GPU -1GB Ram

The Basics – Storage



Micro SD card – 8GB recommend up to 64GB Class 10

We will boot the embedded Operating System from here

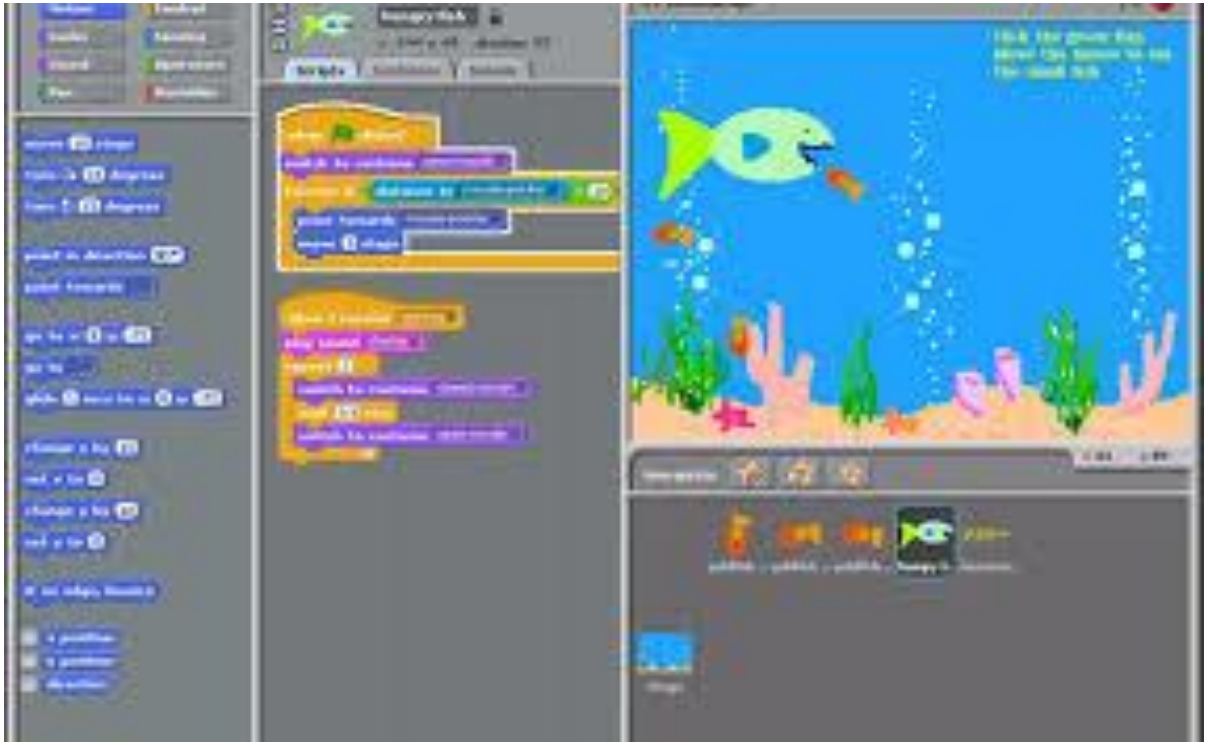
What can I do with the Raspberry Pi

What can I do with a Pi?

Use it as a Desktop Computer if you really wanted to

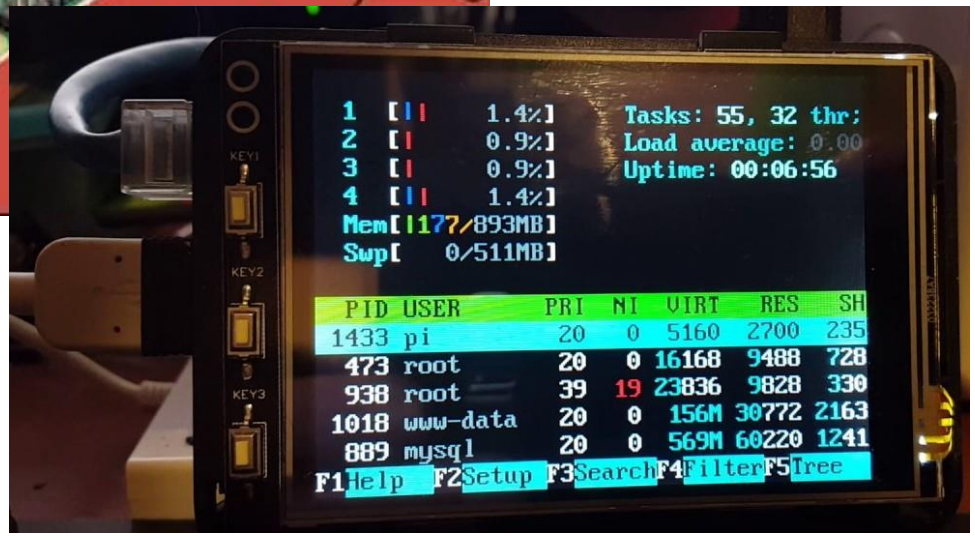
What can I do with a Pi?

Run learning programs (in an unlikely event)



What can I do with a Pi?

Host a website…



What can I do with a Pi?

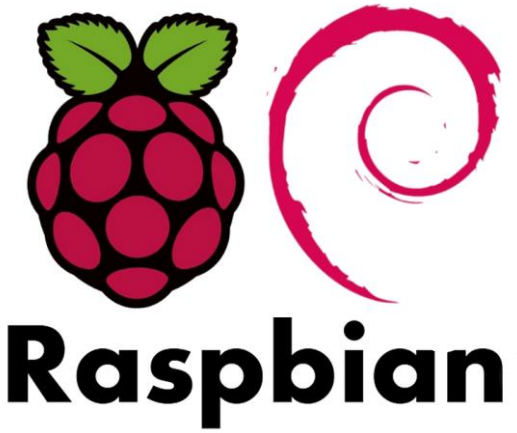
I think it's quite obvious by now…

Basically you can pretty much do anything you can do with a regular computer, and then some…

Especially: Use it as an embedded platform because it is small and has convenient GPIO and other ports that are needed for many embedded platforms

Setting up the Raspberry Pi

Needed Software for the RPi



- It runs Embedded Linux, the distro is called “Raspbian”
- You can download a disk image to burn onto a MicroSD card here: https://www.raspberrypi.org/downloads/
- Get NOOBS (NOOBS_latest.zip) the precompile disk image, as well as the Raspian sources disk (raspbian_lite_latest.zip)
- Get the Win32 Disk Imager for writing images to USB sticks or SD cards from: http://sourceforge.net/projects/win32diskimager/

Software will be available in the lab. A copy of the disk imager is on Vula.

Booting the Pi

- Experimenting with Linux commands
- Finding out how to copy files from the Rpi to the PC and vice versa
- Basically figure out how to do thing you will likely need to use in the later pracs in terms of e.g. getting code files that may need to be submitted

Understanding the Boot Sequence

- Microcontrollers /microprocessors are different
  - Different types of CPU
  - Difference peripherals
  - Different address map, etc.
- Linux tries to provide a generic structure that caters for a broad range of processors
- But as a consequence …
  - The hardware has to be configured for the software
  - The software needs to be configured for the hardware
- The main configuring of the hardware is really about setting up the memory and the boot method.

The Generic Boot Sequence

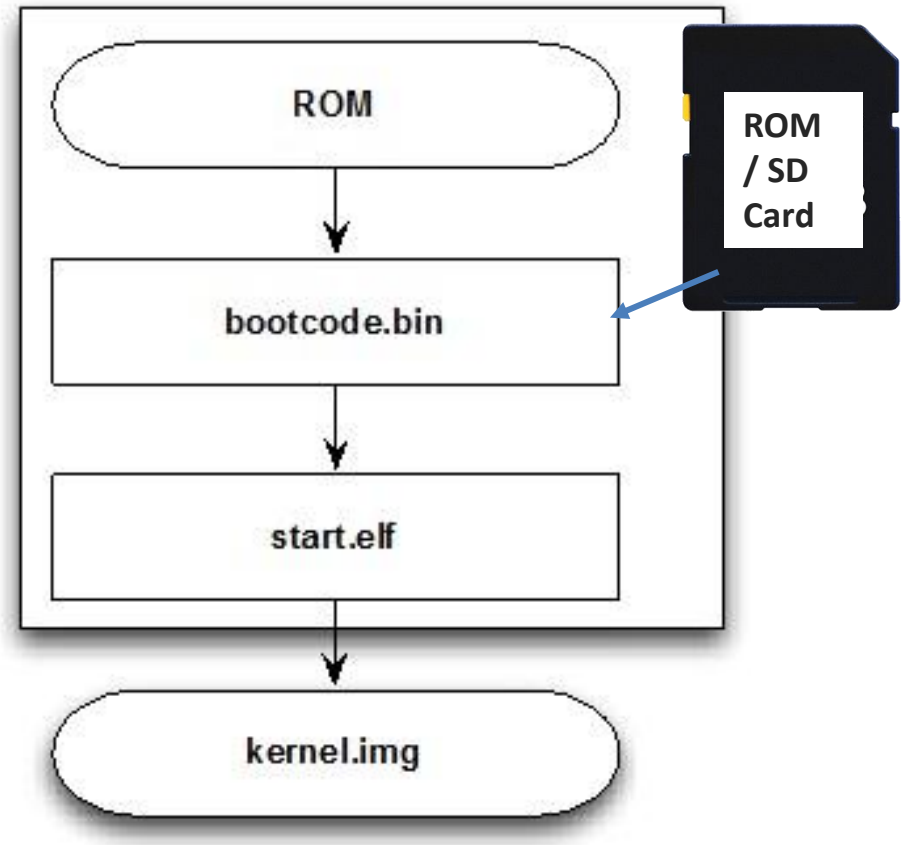
- The generic boot sequence for an embedded platform is as follows:
  1. Hardware power-on / boot initialization
  2. Bootloader activation
  3. Linux kernel boot and loading the rest of the operating system

Raspberry Pi boot sequence

The RPi follows a similar fashion to the generic boot sequence:

There is a bootloader sitting in internal flash/ROM which then attempts to load in O/S loader (bootcode.bin for Raspbian) from the flash or alternate non-volatile memory device.

Then, assuming a Linux operating system, the os loader searches for the start program, which is somewhere within the boot partition, and then it puts the kernel image into memory and continue to load subsequence drivers, etc., that makes up the operating system.



Raspberry Pi boot sequence

How you tell what stage of the boot sequence you are in?...

Why would you want to know?...

It can be very useful to know because it can help you determine how the system failed and identify what aspect of the system to fix … otherwise you are working totally in the dark!

Linux outputs to the screen (cout) to say what it is doing.
Example output of the early boot phase:

```
Uncompressing Linux... done, booting the kernel.
[  0.000000] Booting Linux on physical CPU 0
[  0.000000] Initializing subsys cpu
[  0.000000] Linux version 3.6.1 (gcc version 4.6.3)
[  0.000000] CPU: ARMv8-compatible processor [420eb365]
             revision 6 (ARMv8), cr=00c5387d
…
```

Raspberry Pi boot sequence

The last stage of the boot sequence is the **init** operations, (which we will find out more about later) …

The init stage essentially starts services (or ‘daemons’ as they are generally called by Linux gurus)

Basically if you get to the **init** stage, you have succeeded in getting the hardware and boot memories properly set up to load linux. You will most probably be able to log in to the operating system and start using it.

Don’t expect too much at this stage, you might get just this basic login:

```
Raspbian GNU/Linux 7 raspberrypi

raspberrypi login:
```

The image configured for the labs actually has auto-login activated, you won’t need to log in.

Default Login for Raspberry Pi

The default root user login for Raspbian is:

Username: pi

Password: raspberry

Using Linux

A handout of useful Linux commands has been provided

I thought hardcopies would be useful as you might not have access to the internet when using the Rpi with a screen attached

raspi-config

If you are logging in with a monitor, then open a terminal. Otherwise if via ssh just enter the command:

sudo raspi-config

From this program you can change lots of settings. I suggest changing your password from the default. Note that in this tool you also have the option to change what environment you want you the Pi to boot in to: either Desktop GUI or Command Line

Creating users

- In the possible even you might share your RPi (e.g. in project later to test comms) you may want to create separate users, use:

sudo useradd –m <newusername> -G sudo

Leave out the –G sudo if you don't want to grant them superuser privileges (e.g. if you are just letting them use programs to run e.g. network socket and don't want to let them access your code)

Editing text files on the RPi

- Use nano or vim
- If you are booting with a monitor or exporting the display you can just use an editor like Leafpad or IDLE.

Raspberry Pi Booting

If you have a HDMI monitor or VGA monitor and HDMI→VGA convertor then try hooking it up to your RPi and boot it with the monitor (and keyboard and mouse) attached. Have a look at the various boot logs displayed.

but for the lab…

Introducing…

The Headless Pi

Running Headless

- In the labs we have certain limitations:
  - Each workstation has just one monitor, one keyboard, and one mouse.
  - So if we wanted to run the RPi as a desktop computer, there are resource limitations.
- The solution is to run it in headless mode…

However… there are a number of spare keyboards, computer mice, and HDMI→VGA connectors that can be used if there is a problem, e.g. if your RPi is not connecting via the network. These is just for debugging purposes.

Running Headless

- Get hold of the disk image and use the image burning tool to write the OS image over to the SD card provided with your RPi
- Note:
  - You cannot simply copy the files (e.g. using Windows Explorer) since the bootcode image that loads the rest of the operating system has to go into the correct sectors of the SD card.
  - This is because the bootloader in ROM is too basic to read the FAT etc. of the SD card*.

* Technically things are a bit more sophisticated, it has another program that can read the OS install program packaged within the OS that can be used to prepare and restructure the SD so that it can boot linux from the card.

Booting while headless

- The lab image has been configured to boot into the Desktop GUI and to start sshd (the ssh daemon).
- The IP has also been configured to be static, set to 192.168.0.10, netmask 255.255.255.0
- The lab machines all have a 2nd network port, which should already be configured to 192.168.0.1, but if not please change the network settings for that card
- Once you can ping 192.168.0.10 you should be able to run an ssh client (like putty) to connect up to your RPi on that address.

Getting the RPi into Shape!

Recommended

- Prac 1 is really about getting your RPi ‘comfortably’ set up, once you’ve familiarised yourself with the usual Linux tools.
- It is quite nice using the GUI Desktop on the RPi, e.g. using IDLE right on the machine instead of copying files across all the time.
- These are things you may want to try getting set up on your RPi, and why…

Consider getting these set up on the RPi

- Xwindows
  - If you are using Bluelab you could boot into Linux and then on the RPi export your Xwindows display to the PC display

Consider getting these set up on the RPi

- Xwindows
  - Do as follows:
    - From a bluelab terminal you should be able to
      ssh pi@192.168.0.10 -X
      Where the –X is used to share the display.
      Then when you have logged in to the RPi type lxterm to start a new terminal, or IDLE to start the Python IDE that will not pop up on the monitor of the PC.

If the above ssh command does not work then you may need to find what the IP address is of the bluelab machine, on the bluelab terminal type xhost - then ssh in to the RPi and in the RPi terminal type export DISPLAY=machineip:0 and now start a terminal or other X program and it should open in the bluelab PC's window.

Consider getting these set up on the RPi

- Xwindows
- VNC
  - VNC you can pretty much get the same functionality as the Xwindows export, essentially it is a tool that copies the RPi's display to a window on the PC.
    Get hold of tightvnc or similar which may likely be on the lab machine already. Start the vncserver on the Rpi and use the vncviewer on the PC to connect to the vncserver display on the RPi.

Consider getting these set up on the RPi

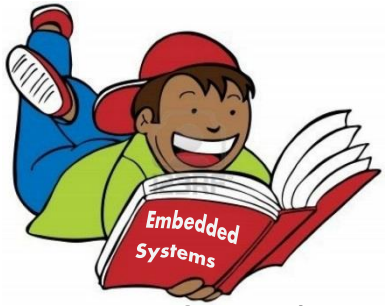
- You can by all means try to connect the Rpi up to the internet to get further tools installed on it. You may need to bridge your network connection to do this, or set up the Rpi network to use DHCP.
- Once you're got the desired tools set up, like your favourite Python and C editors you probably don't need to worry about using the RPi again for this course at least.
- For all the pracs the code development tools already provided with Raspbian will be sufficient.

You’re now all set for Prac1 !!

The Next Episode…

Lecture L02

(CH1) Challenges in embedded systems design. Understanding the ‘design flow’ (or design lifecycle) of an embedded system.



Reminder: please try to read (at least briefly) over chapter 1 of the textbook if you manage to get hold of it or can borrow a copy to read over.