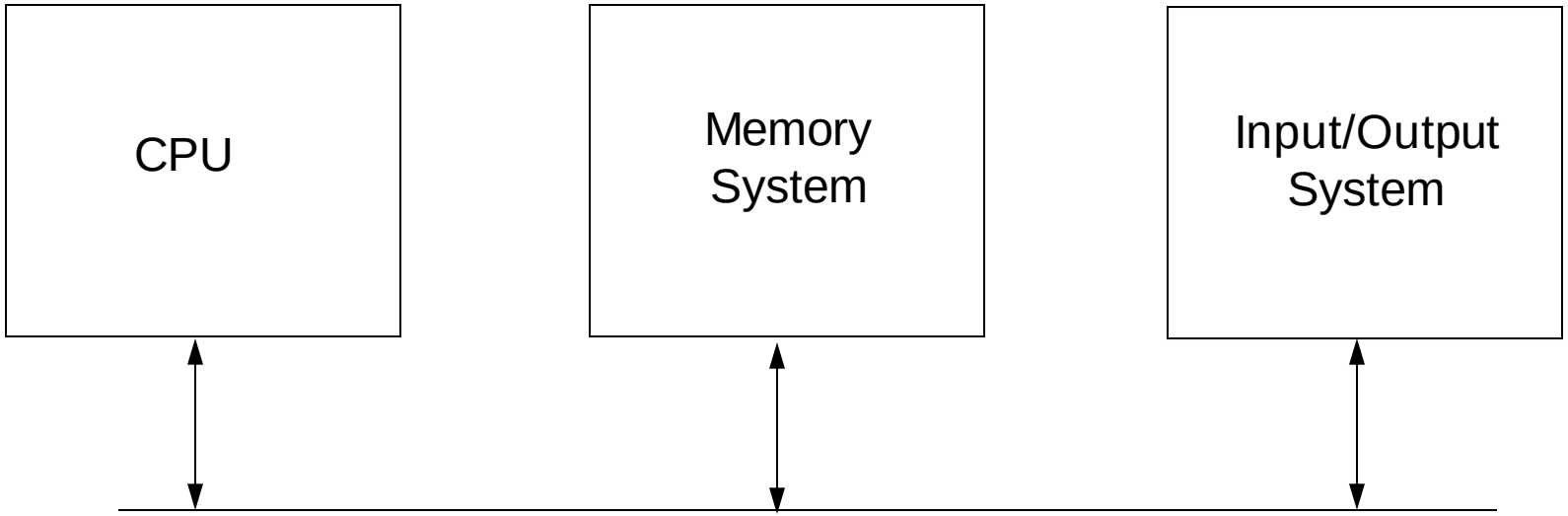
Fundamentals of Microcontrollers - Raspberry Pi

Kumar Yelamarthi

Central Michigan University

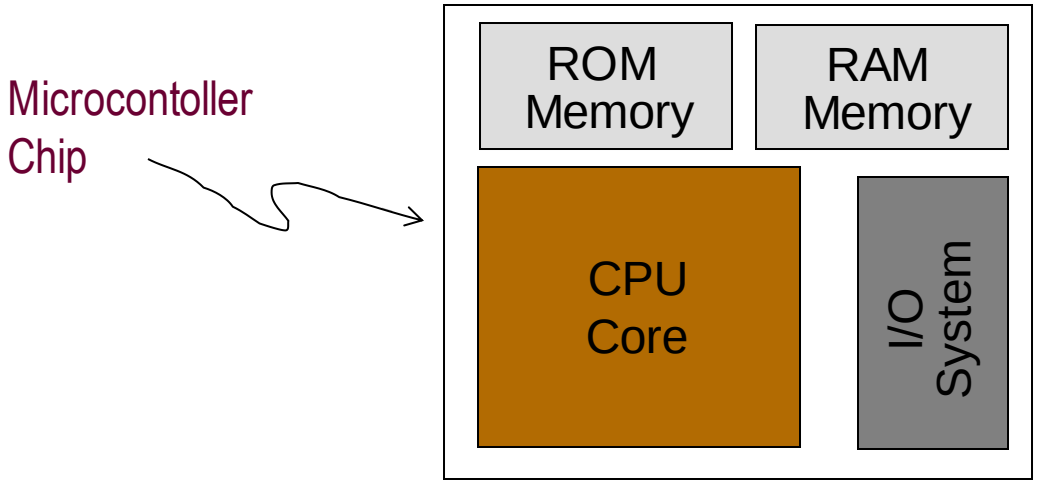
Mt Pleasant, MI

Components of a Computer System



Microcontrollers

- Microcontrollers integrate all the components of a computer system onto a single chip
- All components are optimized to perform the functions necessary to control a larger system
- Size, capability, cost, and power consumption are more important considerations
- 8 bit microcontrollers have the majority of the market right now, but 16 and 32 bit microcontrollers are available and have been gaining market share in the recent years



Microcontroller Components

CPU

- May be an adaptation of an existing microprocessor or a completely new architecture
- Instruction set is typically modified to deal with new I/O system

Memory

- On-chip NVRM to hold program - can be mask programmable ROM, OTP ROM, or EPROM/EEPROM
- On-chip RAM for variable storage - usually small amount and can double as μP's registers
- Both can be expanded using off-chip memory (extended microcontroller mode)

Microcontroller Components

I/O system - tailored to provide services usually required in embedded control applications

- Binary I/O pins (buffered, built-in pull ups, etc.)
- Timers/Counters
- Pulse Width Modulation (PWM) outputs
- Capture inputs
- AD and DA converters
- UART or other communication controllers

Miscellaneous

- Oscillator options
- Reset options
- Sleep/reduced power modes
- External interrupts
- Watchdog timers

Microcontroller Binary I/O pins

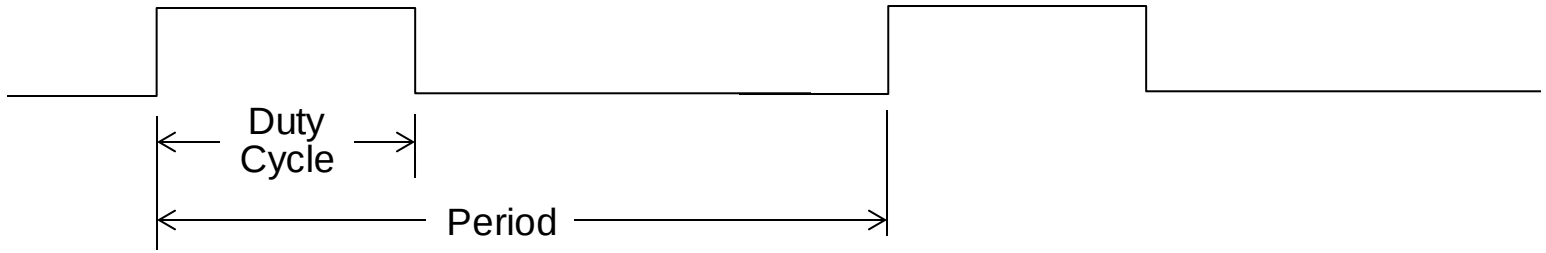
- Binary I/O pins are pins that can either write (output) a ‘1’ (Vdd) or ‘0’ (Gnd) value, or read (input) a ‘1’ or ‘0’ value
- Direction (input or output) is typically set on a per bit basis with a direction bit
- These type of I/Os are often multiplexed with other I/O functions
- Pull up (or pull down) resistors are typically included for connection to a bus
- Drive capability is usually in the medium range (20 mA - 60 mA)
- Interrupt on change features are sometimes included

Microcontroller Counter/Timers

- Timers are special registers within the microcontroller that are configured as binary counters
- Timer registers can be preloaded with a starting value and possibly have an ending value (other than overflow) specified
- Overflow or hitting the end value generates an interrupt to the processor
- Timer register clocking:
  - Internal CPU clock - timer mode
  - External clock via a dedicated I/O pin - counter mode
- Timers are the basis for PWM outputs and Capture inputs
- Timers are also sometimes used by the external communications functions (UARTS, etc.)

Microcontroller PWM Outputs

- Pulse Width Modulation (PWM) outputs are special purpose outputs that can generate a specified train of pulses
- Internal general purpose timer registers are often used to generate the pulse train
- When generating a PWM output, timers can not be used for other purposes
- Typically the period and duty cycle of the pulse train are specified
  - Period is specified in counting units of the specified timer
  - Duty cycle is specified as the actual width (in timer units) of the “on” pulse



Microcontroller Capture Inputs

- Capture inputs are used to count external events
- A general purpose timer is associated with each capture register
- The timer is incremented each time the external event occurs
- The external event can be a rising or falling edge on the capture input, or multiples there of (i.e., every 4th rising/falling edge, every 16th rising/falling edge)
- The capture (timer) register value can be read or an interrupt can be generated when it equals a specified value

Other Microcontroller I/O functions

- Some microcontrollers include an internal Analog-to-Digital converter (some with multiple channels)
- Accuracy (no. of bits) is variable, but is typically in the 8-16 bit range
- Input voltage usually in the range of 0 to 1x or 2x Vdd

- Digital-to-Analog converters are also sometimes provided
- Input is typically one or two processor data words wide
- Output is 0 to 1x or 2x Vdd
- Current drive is equivalent to other I/O ports

Other Microcontroller I/O functions (etc.)

Most microcontrollers include built-in serial communications capabilities

- Universal Asynchronous Receiver/Transmitter (UART)
- Serial Peripheral Interface (SPI®)
- I^2C

Most microcontrollers also include the following capabilities for dealing with real-time systems:

- Optional oscillator sources such as external clock, external crystal, or simple RC timing circuit
- Watchdog timers that will interrupt the CPU after a certain period of time if the CPU does not reset it - useful for recovering from software malfunctions
- Power-down or sleep modes from which wake-up occurs on interrupts (from watchdog timer, timer/counters, capture inputs, external interrupts, etc.)

What is Raspberry Pi

- Created by Raspberry Pi Foundation
- A credit-card sized computer that plugs into your TV and a keyboard
- Pi-1 Model B was released in 2012, Model A in early 2013,….
- Pi-3 Model B was released in 2016
- It’s a capable little PC which can be used for many of the things that your desktop PC does

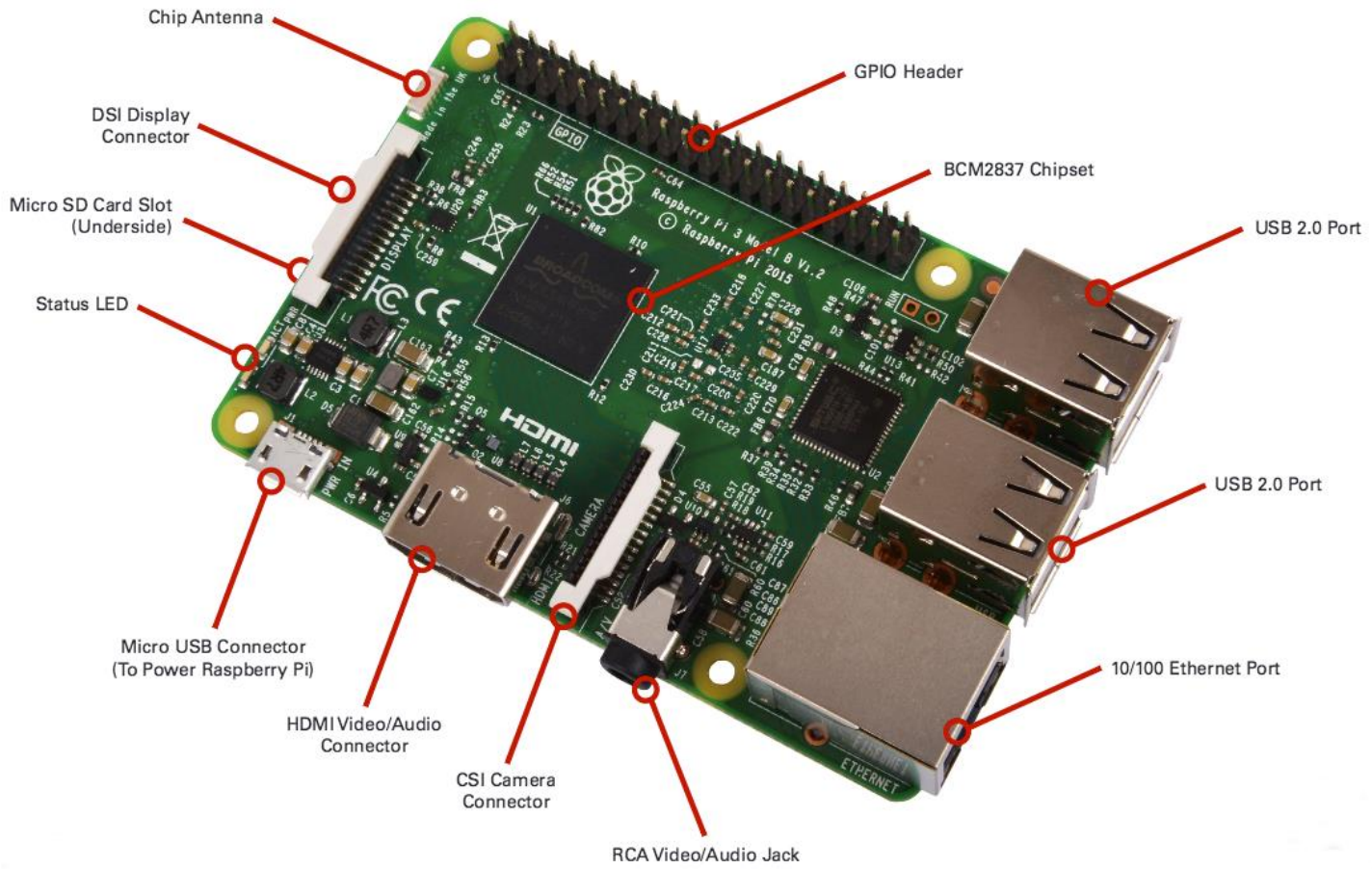
Why was Raspberry Pi Made?

For education

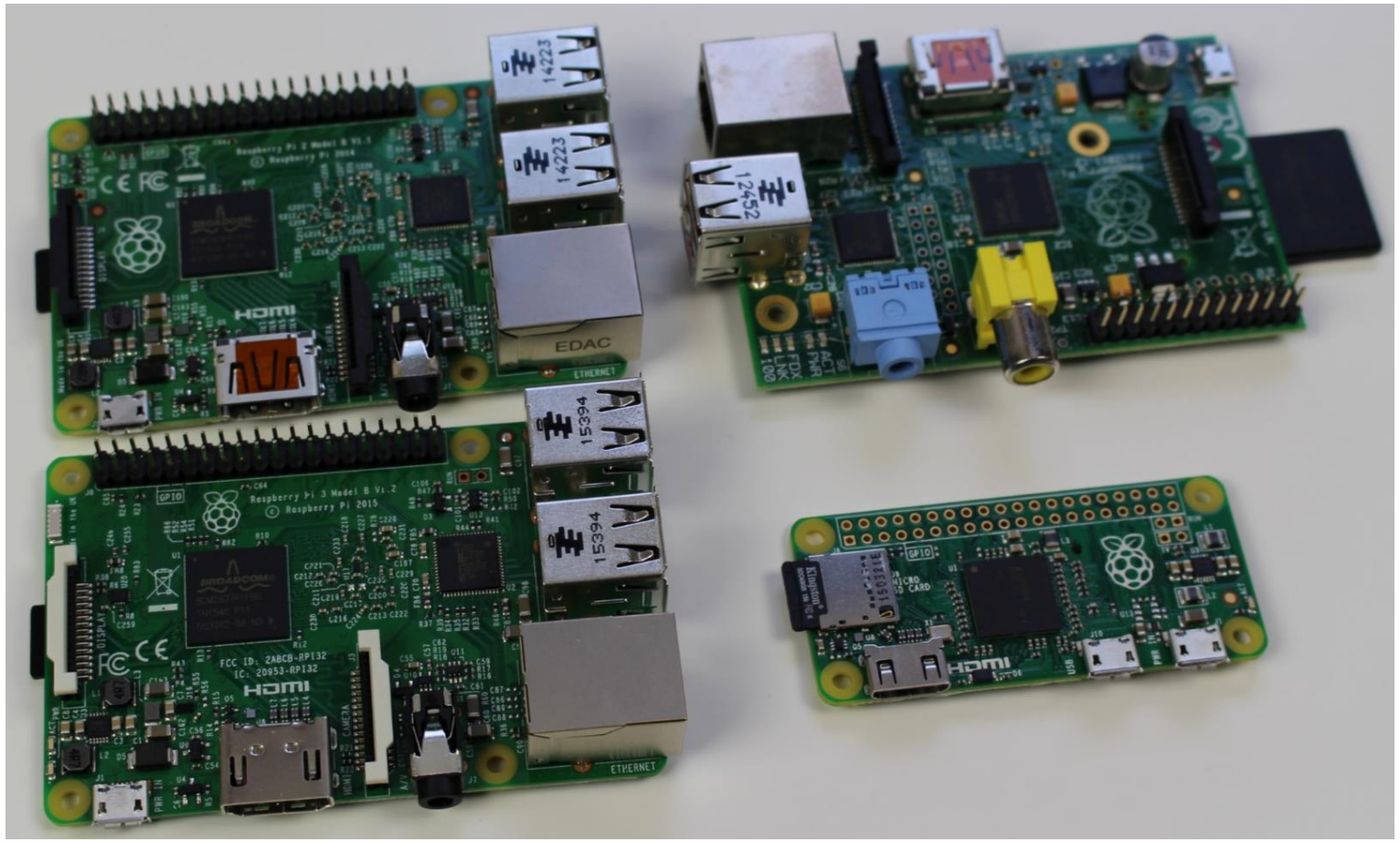
Original measure of success: more CS students

But education-related efforts are rapidly spreading downward, aiming toward young children

Raspberry Pi 3 – Model B



Raspberry Pi Model Comparison

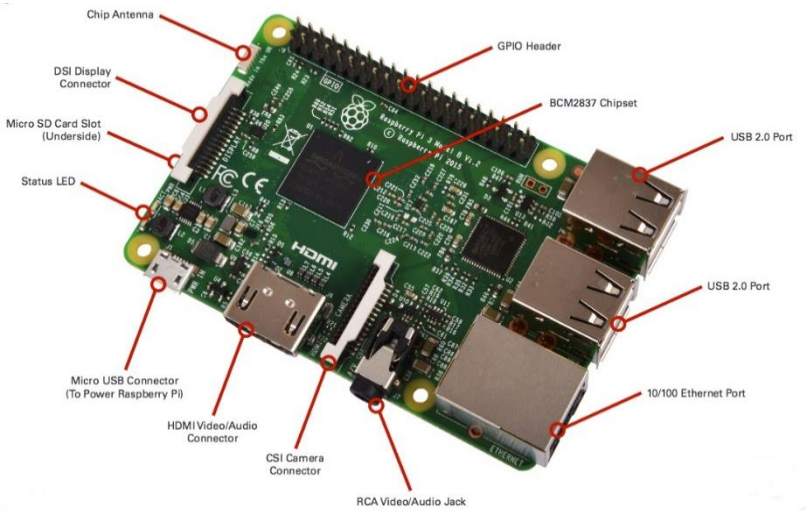


Raspberry Pi Model Comparison

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	$35	$5	$35	$35

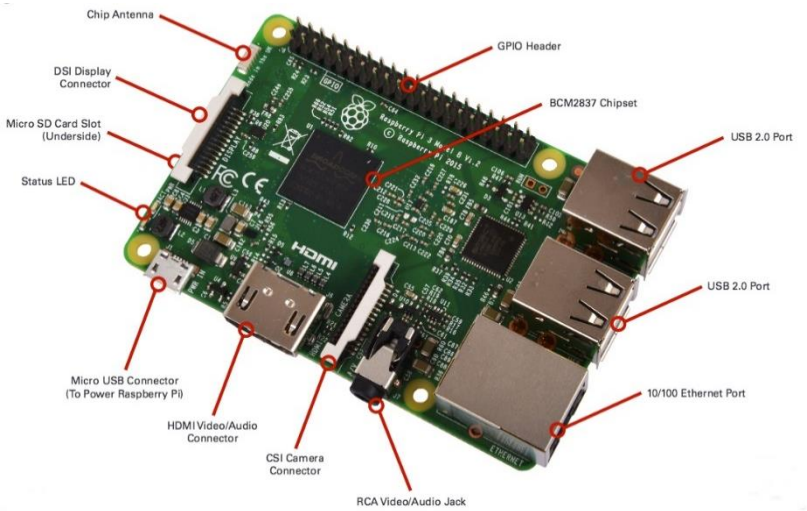
Raspberry Pi - Hardware

- Processor and Memory
  - Broadcom 2835 System On Chip
    - 32 bit ARM RISC CPU core (not x86 compatible)
    - Videocore IV GPU
  - ARM11 hardware, ARM6 architecture
  - Default clock is 1200Mhz. Overclocking not permitted in Rpi-3
  - GPU is basically as powerful as what was on the original Xbox
- Mass Storage: micro SD card
  - Any SD card
  - Kernel boots from SD card
  - Easy to have root FS on other device



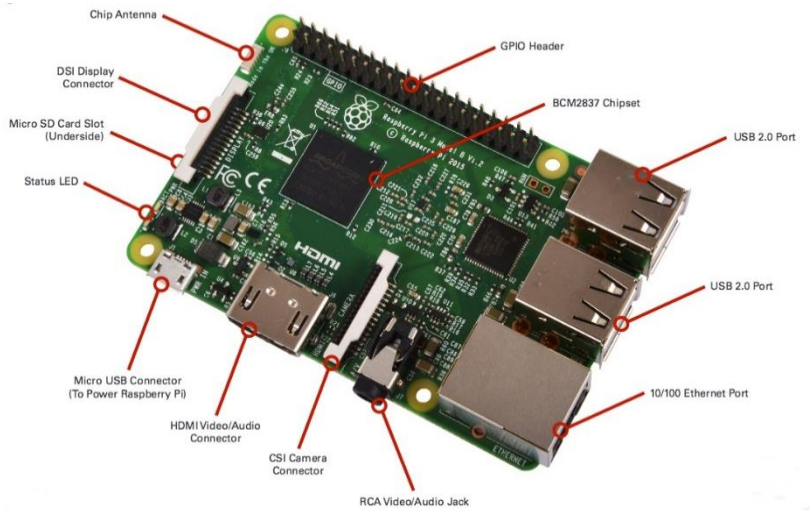
Raspberry Pi - Hardware

- Video
  - HDMI or (digital) DVI via cheap adaptor/cable
  - Composite NTSC/PAL via RCA
  - Wide range of resolutions
    - 640×350 to 1920×1200
- Audio
  - Via HDMI
  - Via 3.5mm stereo jack
  - Stereo analog
  - Output only



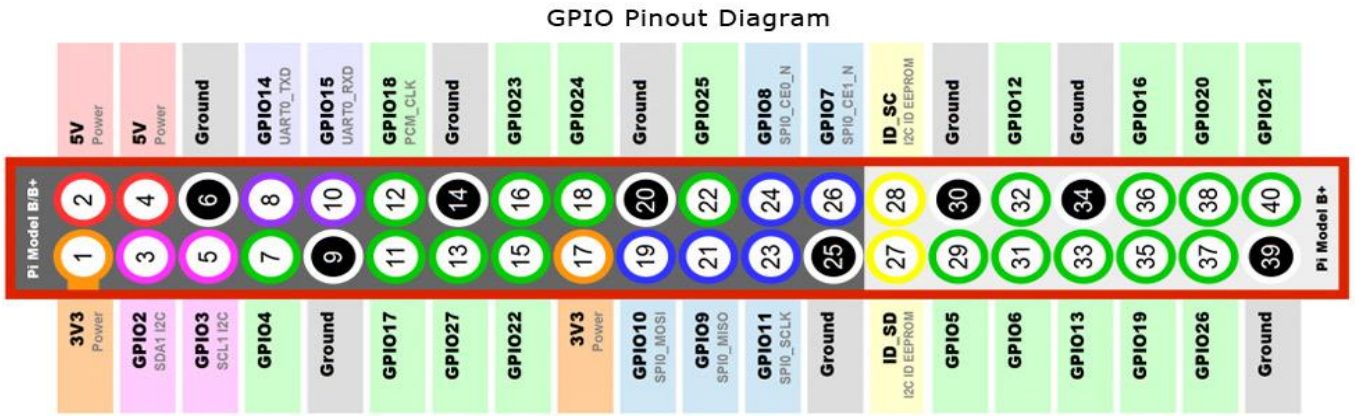
Raspberry Pi - Hardware

- Networking
  - 10/100Mbps via RJ45 on model B
  - Wireless – Wifi
  - Bluetooth LE 4.0
- USB
  - Four USB sockets on RPi model B, single on model A
  - Expandable via regular or powered hubs
- Power
  - Primary power via microUSB plug
    - A 2.5A cell charger works well, but a 1A cell charger might be enough to start the processor
  - Pi-1 requires only 0.8A current
  - PC USB port does not work – Why?

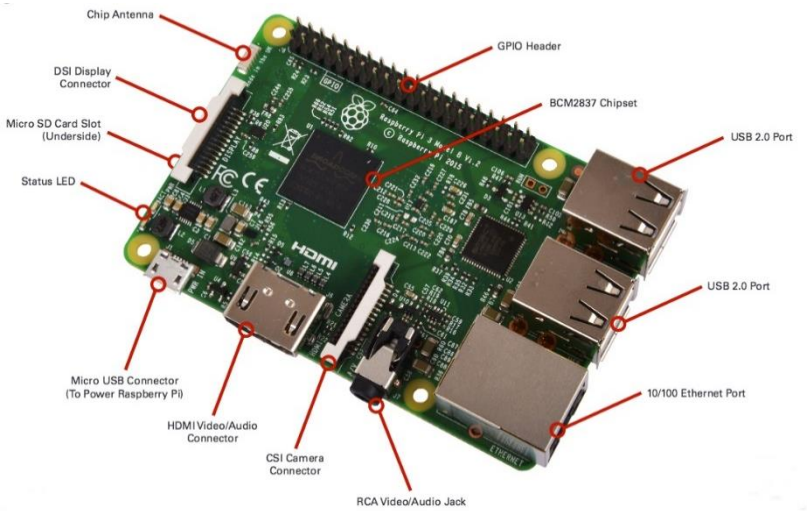


Raspberry Pi - Hardware

- General Purpose I/O (GPIO)
  - 3.3 volt logic via 40 pin header
    - NOT 5 volt or short tolerant
  - Parallel I/O pins
  - UART (Linux console support)
  - I2C, SPI for peripherals
  - No A/D, so no analog input
  - Can do software-based PWM
  - Libraries exist for interacting with the GPIO through several programming languages



- More I/O
  - Display Serial Interface (DSI) LCD panel support
  - Camera Serial Interface (CSI) camera support
  - JTAG
  - Additional GPIO via other headers



Setting up Raspberry Pi

- SD card (Minimum size 8Gb)
- HDMI to HDMI / DVI lead
- Keyboard and mouse (USB 2.0)
- Ethernet network cable (optional)
- Power adapter (micro USB power 2.4A at 5V)
- Audio lead (If you are not using HDMI)

HDMI connector

HDMI to DVI lead

Setting up Raspberry Pi

Raspberry Pi OS

Raspberry Pi is a full blown computer

Runs Linux

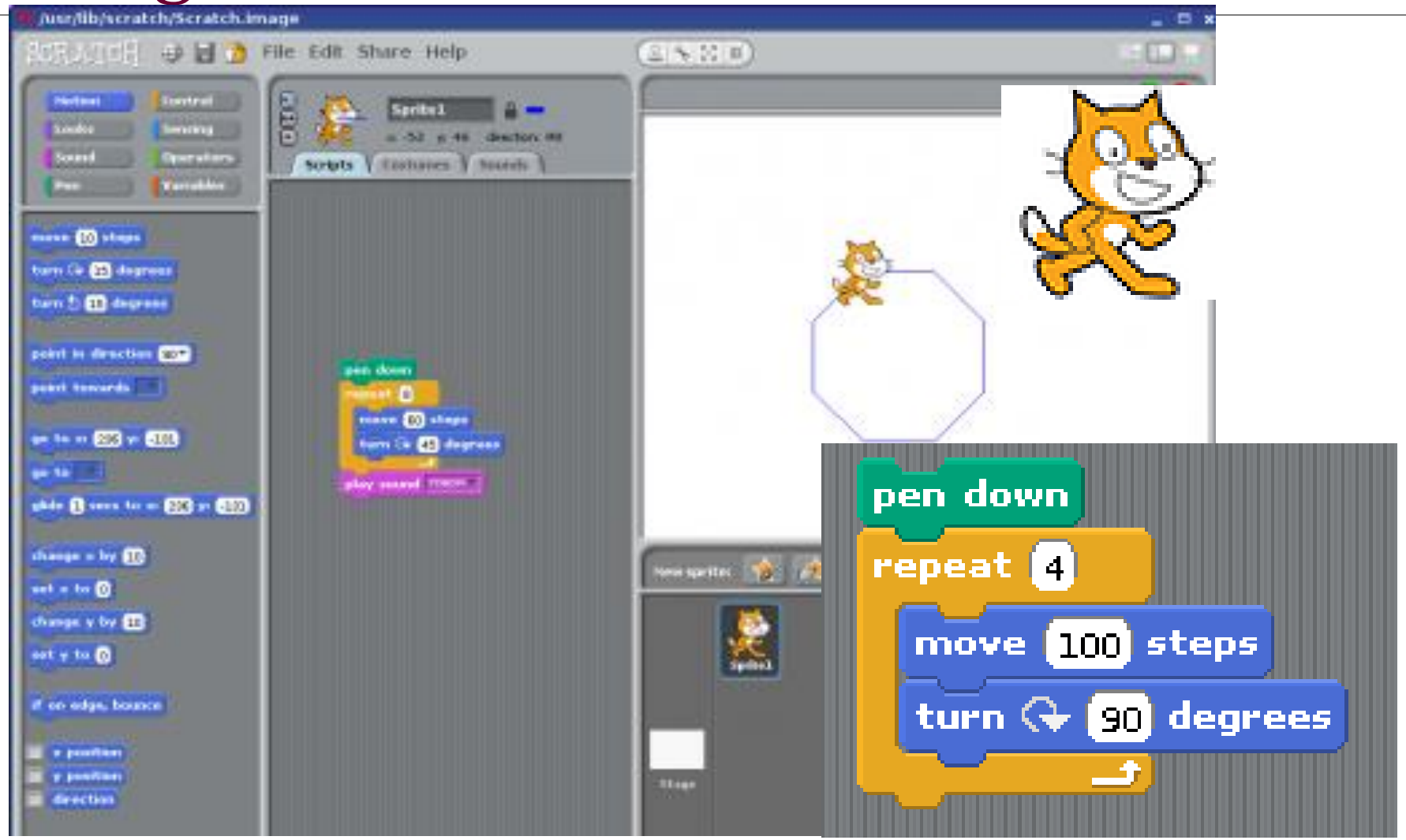
Supported OS

- Raspbian (Debian optimized for Rpi)
- Debian (non-optimized)
- Arch Linux
- RISC OS
- NetBSD
- Openelec – an XBMC media center
- Pidora – Fedora remix

Programming

- By default, supporting Python as the educational language
- Any language which will compile for ARMv6 can be used with the Raspberry Pi
- For primary age SCRATCH game maker is bundled
- Raspbian contains Java SE platform

Programming – SCRATCH



scratch is free from MIT

Programming – Python

polygon.py

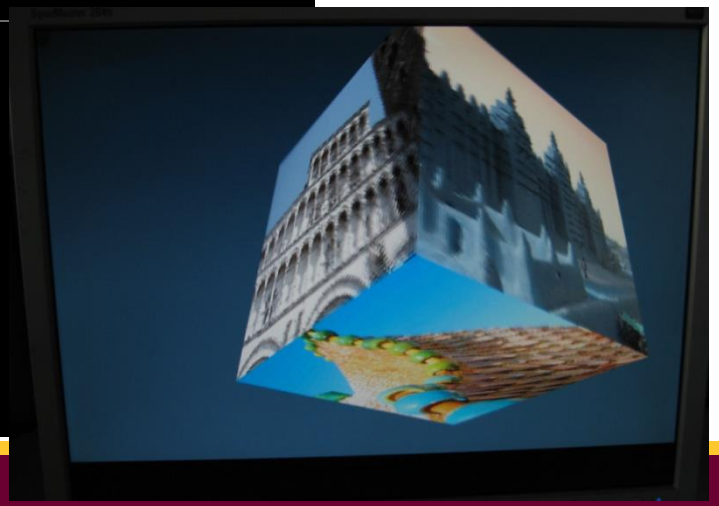
```
1  # polygon.py
2  # draws polygons
3
4  import turtle
5  def polygon(length, sides):
6      for i in range (sides):
7          turtle.fd(length)
8          turtle.left(360/sides)
9  # main
10 print("Let's draw a polygon.")
11 how_many = int(input("How many sides would you like?"))
12 how_big = int(input("How long do you want the sides?"))
13 polygon(how_big, how_many)
14 input("Press a key to quit.")
```

Programming - OpenGL

Raspberry Pi incorporates a powerful graphics accelerator – OpenGL

- Examples, including Quake 3 at
  - https://github.com/raspberrypi/quake3

```
   I A  triangle.c (Modified)(c)  static void redraw_sc  Row 359  Col 48   6:10  Ctrl-K H for help
static void redraw_scene(CUBE_STATE_T *state)
{
   // Start with a clear screen
   glClear( GL_COLOR_BUFFER_BIT );
   glMatrixMode(GL_MODELVIEW);

   glEnable(GL_TEXTURE_2D);
   glTexEnvx(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);

   glBindTexture(GL_TEXTURE_2D, state->tex[0]); // bind texture
   glRotatef(270.f, 0.f, 0.f, 1.f ); // front face normal along z axis
   glDrawArrays( GL_TRIANGLE_STRIP, 0, 4);

   // same pattern for other 5 faces - rotation chosen to make image orientation 'nice'
   glBindTexture(GL_TEXTURE_2D, state->tex[1]);
   glRotatef(90.f, 0.f, 0.f, 1.f ); // back face normal along z axis
   glDrawArrays( GL_TRIANGLE_STRIP, 4, 4);

   glBindTexture(GL_TEXTURE_2D, state->tex[2]);
   glRotatef(90.f, 1.f, 0.f, 0.f ); // left face normal along x axis
   glDrawArrays( GL_TRIANGLE_STRIP, 8, 4);

   glBindTexture(GL_TEXTURE_2D, state->tex[3]);
   glRotatef(90.f, 1.f, 0.f, 0.f ); // right face normal along x axis
   glDrawArrays( GL_TRIANGLE_STRIP, 12, 4);

   glBindTexture(GL_TEXTURE_2D, state->tex[4]);
   glRotatef(270.f, 0.f, 1.f, 0.f ); // top face normal along y axis
   glDrawArrays( GL_TRIANGLE_STRIP, 16, 4);

   glTexEnvx(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);

   glBindTexture(GL_TEXTURE_2D, state->tex[5]);
   glRotatef(90.f, 0.f, 1.f, 0.f ); // bottom face normal along y axis
   glDrawArrays( GL_TRIANGLE_STRIP, 20, 4);

   glDisable(GL_TEXTURE_2D);

   eglSwapBuffers(state->display, state->surface);
}
```



Some Other Things

Can run XMBC Home Theater PC software

- Turn your TV into a Smart TV! – I Have been doing this since 2014

Emulators for tons of old platforms (C64, Atari, NES, etc.)

Games – OpenTTD, OpenArena, more…

Has its own app store

- http://store.raspberrypi.com/

Projects Using Raspberry Pi

Raspberry Pi Cluster

- http://www.southampton.ac.uk/~sjc/raspberrypi/pi_supercomputer_southampton.htm
- http://www.youtube.com/watch?v=Jq5nrHz9I94