universidade de aveiro
departamento de eletrónica, telecomunicações e informática

# Raspberry Pi 4 Model B

## Architecture, Hardware and Software aspects

TP1 - Grupo 7
Lúcia Sousa 93086
Raquel Pinto 92948

03/05/2022 ASE - Arquitetura de Sistemas Embutidos

# Raspberry Pi 4

**Raspberry Pi 4** is capable of doing the same as a computer. It is possible to set up an operating system, and connect wires and circuits directly to its GPIO pins.

**COMPONENTS**

**A** USB Type-C power in

**B** DSI display port

**C** Wireless/Bluetooth

**D** Micro-HDMI 0

**E** Micro-HDMI 1

**F** System-on-Chip

**G** GPIO

**H** RAM

**I** CSI camera port

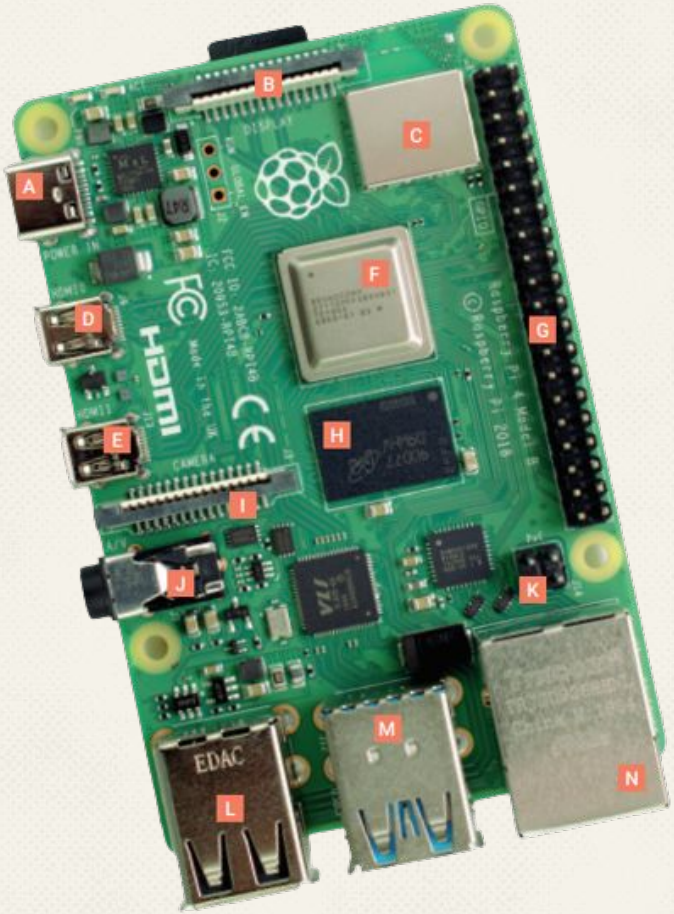
**J** 3.5mm AV

**K** PoE - Power over Ethernet

**L** 2 USB 2.0 ports

**M** 2 USB 3.0 ports

**N** Ethernet port



# Specifications

**The Processor**

Broadcom BCM2711, a 64-bit quad-core Arm Cortex-A72 clocked at 1.5GHz

**Memory**

SDRAM LPDDR4-3200 of 1GB, 2GB, 4GB or 8GB

**Video and sound**

2 × micro HDMI ports (up to 4Kp60 supported), 2-lane MIPI DSI display port, 2-lane MIPI CSI camera port, 4-pole stereo audio and composite video port

**Connectivity**

2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet, 2 × USB 3.0 ports and 2 × USB 2.0 ports

# Specifications

**Multimedia**

H.265 (4Kp60 decode);

H.264 (1080p60 decode, 1080p30 encode);

OpenGL ES, 3.0 graphics

**SD card support**

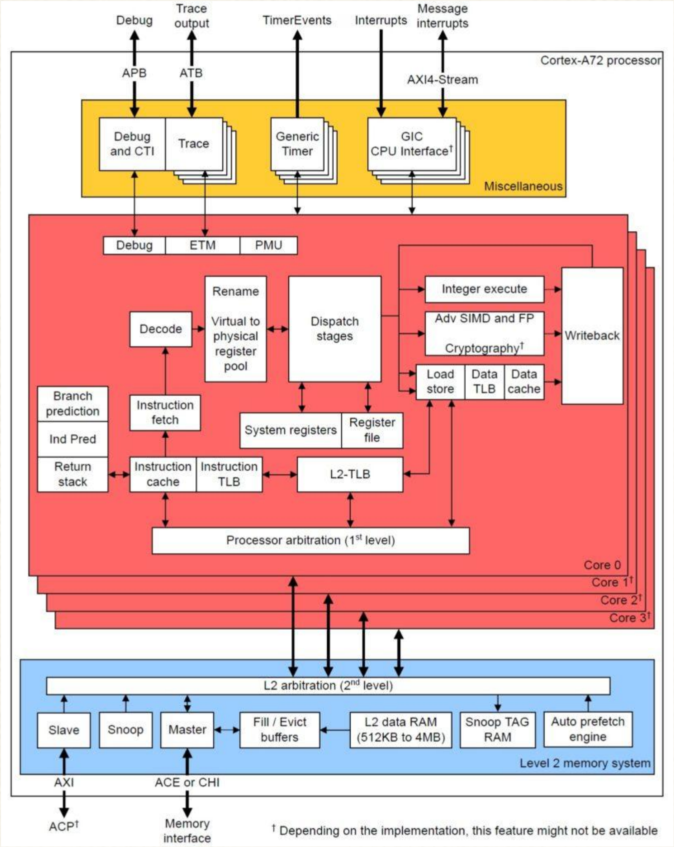
Micro SD card slot for loading operating system and data storage

**Input power**

5V DC via USB-C connector (minimum 3A1), 5V DC via GPIO header (minimum 3A1), Power over Ethernet (PoE)

# CPU

- Quad-core ARM Cortex-A72 processor clocking at 1.5GHz.
- The ARM Cortex-A72 Software Optimization Guide specifies the number of execution cycles and pipeline units for each kind of ARM instruction.
- The Cortex-A72 allows speculative execution (branch prediction).
- The Cortex-A72 can perform a load operation and a store operation every cycle because it has separate load and store pipelines.



# Memory

- The Cortex-A72 organizes physical memory into a hierarchy with the fastest/smallest memory (registers) near the arithmetic/logic unit (ALU) and the slowest/largest memory (RAM) far away and off-chip.
- Data and instructions are read (and written) in efficient chunks making data and instructions available when needed by the registers and ALU.
- The chunks are called “cache lines.”
- Thanks to cache memory, programs run faster when they (re)use data that are close together in memory (i.e., occupy the same cache line) and are the most recently accessed. These notions are called “spatial locality” and “temporal locality.”

```
   Register          Fast, but small
       |
Level 1 caches
       |
Level 2 cache
       |
     RAM             Big, but slow
```

# Memory

- Each core has an Instruction Cache (I-Cache) and Data Cache (D-Cache). The four cores share the Level 2 (L2) cache.

| FEATURE | CORTEX-A72 |
|---|---|
| L1 I-cache capacity | 48KB |
| L1 I-cache organization | 3-way set associative, 64B line |
| L1 D-cache capacity | 32KB |
| L1 D-cache organization | 2-way set associative, 64B line |
| L2 cache capacity | 1MB |
| L2 cache organization | Shared, 16-way set associative, 64B line |

- Processors have one or more memory management units (MMU) that break physical RAM into logical pages. This scheme is called “virtual memory.” The MMU translate logical program addresses (from loads, stores and instruction fetches) into physical RAM addresses.
- Translation lookaside buffers (TLB) are cache-like hardware structures that hold the most recently used (MRU) address translation information, i.e., where a logical page is located in physical memory.

```
Translation registers        Fast, but only a single mapping
          |
   Level 1 TLBs
          |
   Level 2 TLB
          |
         RAM                 Big page tables, but slow
```

# Address Space

Two main addressing schemes:

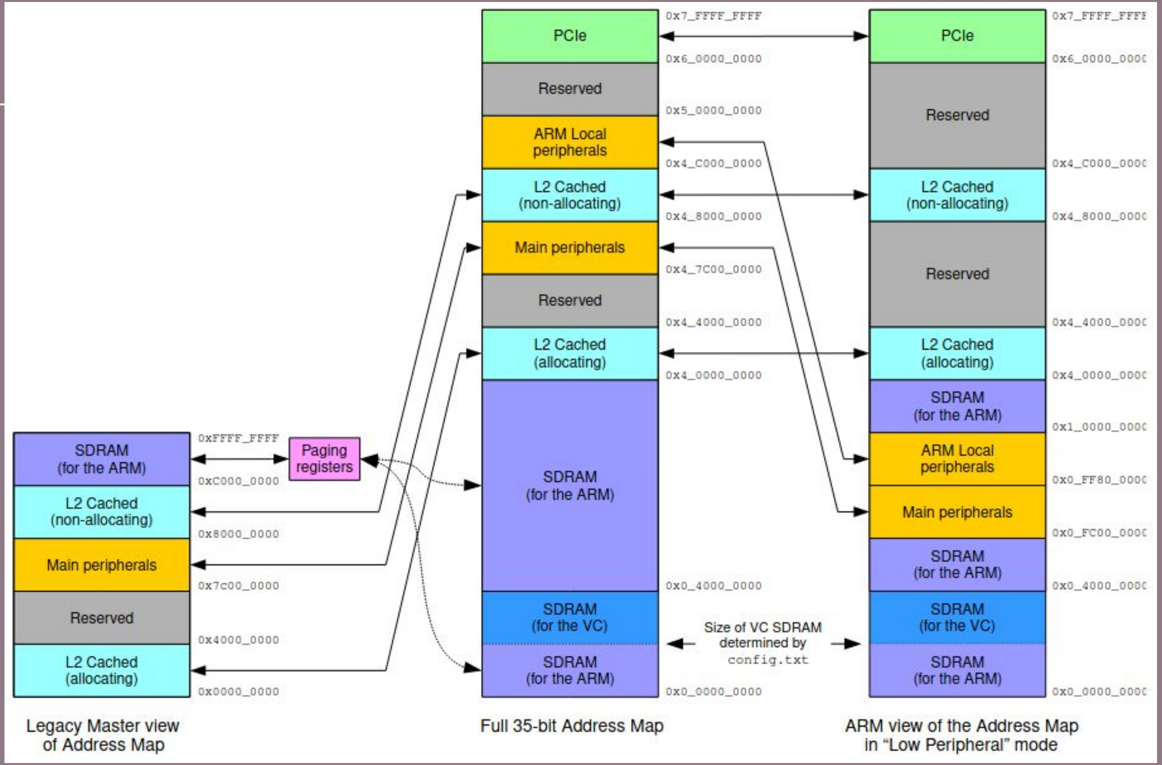
- Full 35-bit address
- Legacy Master 32-bit address

Low Peripherals mode which modifies the processor's (ARM) view of the peripheral addresses

<u>Full 35-bit address:</u> (half)

- Is accessed by Legacy Master and CPU ARM
- Have two caches (one allocating and other non-allocating)

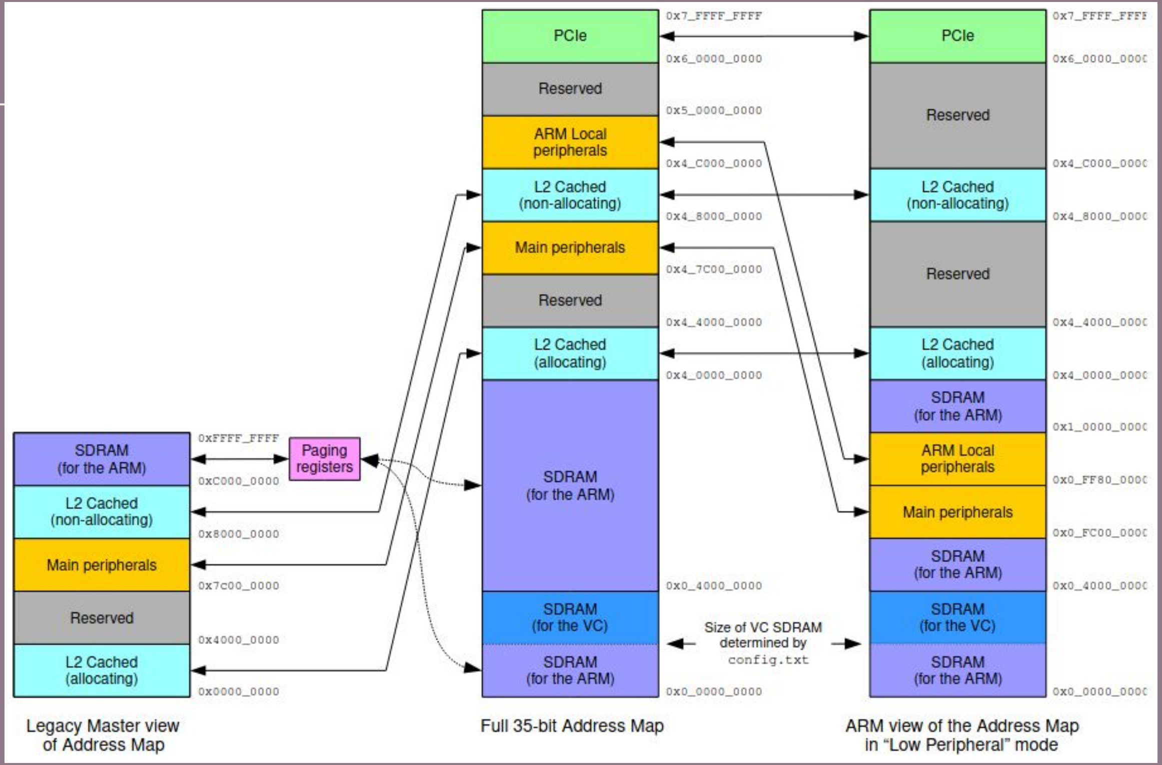
Stores the first 1GB of SDRAM.



# Address Space

Legacy Master 32-bit address: (left side)

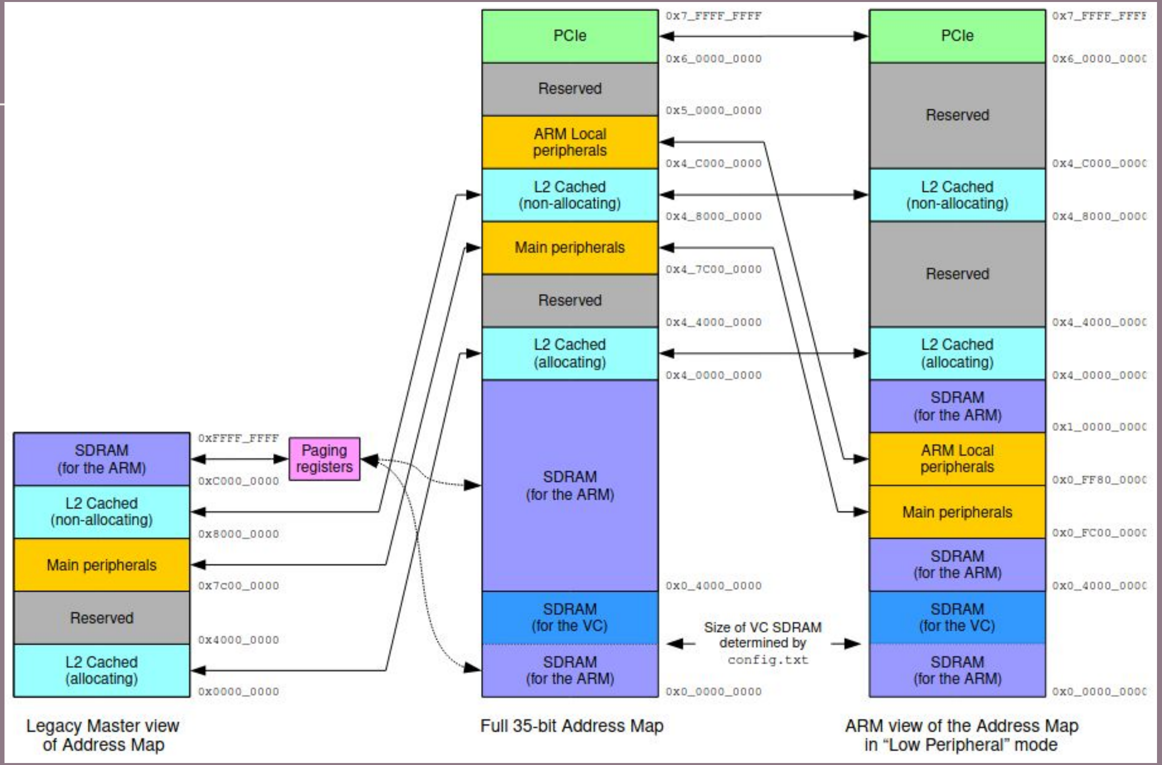
- Is used by software accessing peripherals using the DMA engines (between 0xC000_0000 and 0xFFFF_FFFF)
- If the DMA needs to access RAM above the first 1GB, this window can be moved using the **PAGE/PAGELITE** bits
- Main peripherals (0xFC00_0000 until 0x7FFF_FFFF)
- Cache allocating (0x0000_0000 until 0x3FFF_FFFF)
- Cache non-allocating (0x8000_0000 until 0xBFFF_FFFF)



# Address Space

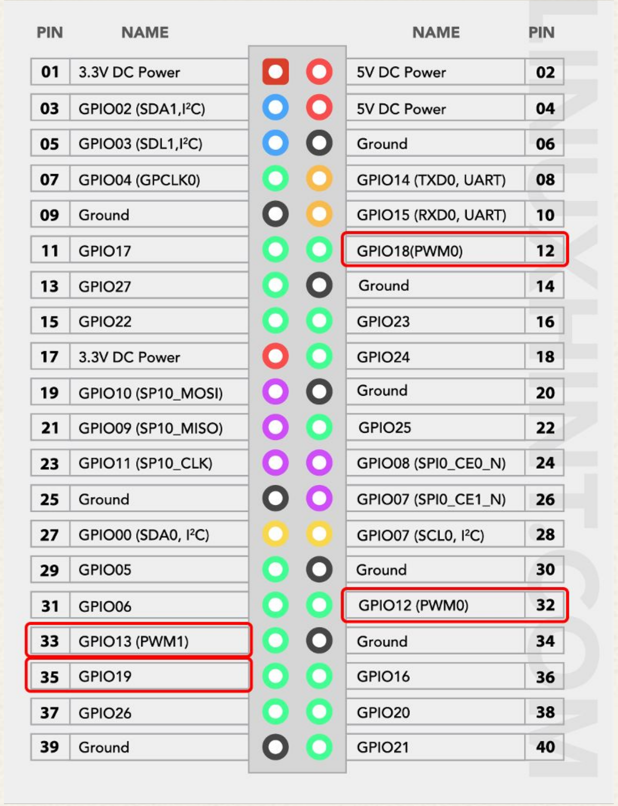
<u>ARM physical addresses:</u> (right side)

- RAM addresses: 0x0_0000_0000 until size of installed SDRAM.
- The VideoCore section of the RAM is mapped in from 0x0_4000_0000 downwards.
- Main and local ARM peripherals (0x0_FC00_0000 until 0x1_FFFF_FFFF)
- Cache allocating (0x4_0000_0000 until 0x4_3FFF_FFFF)
- Cache non-allocating (0x4_8000_0000 until 0x4_BFFF_FFFF)



# Peripherals

- Can be used as software controlled input and output.
- GPIO pins can be switched (multiplexed) into various other modes backed by dedicated peripheral blocks such as DPI, I2C, UART, SPI, PCM and PWM.
- The I2C, UART e SPI peripherals have been added to the BCM2711 processor of the Raspberry Pi 4 Model B.
- This gives users much more flexibility when attaching add-on hardware as compared to older models.



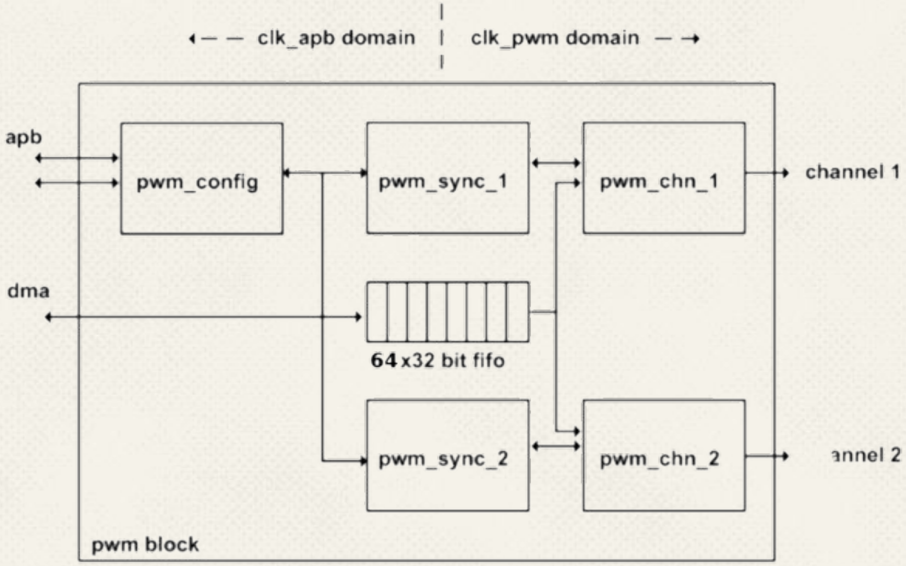
# PWM

**Pulse-width modulation (PWM)** is a modulation process or technique used in most communication systems for encoding the amplitude of a signal right into a pulse width or duration of another signal, usually a carrier signal, for transmission.

The BCM2711 device has two instances of this block, named PWM0 (0x7E20C000) and PWM1 (0x7E20C800).

Each PWM controller incorporates the following features:

- Two independent output bit-streams, clocked at a fixed frequency;
- Bit-streams configured individually to output either PWM or a serialised version of a 32-bit word;
- PWM outputs have variable output resolutions;
- Serialise mode configured to read data from a FIFO storage block, which can store up to sixty-four 32-bit words;
- Both modes clocked by clk_pwm which is nominally 100MHz, but can be varied by the clock manager.



# PWM Implementation

- A value represented as a ratio of N/M can be transmitted along a serial channel with pwm, in which the value is represented by the duty cycle of the output signal.
- **To send value N/M within a periodic sequence of M cycles, output should be 1 for N cycles** and **0 for (M-N) cycles**.
- The desired sequence should have 1s and 0s spread out as evenly as possible, so that during any arbitrary period of time the duty cycle achieves the closest approximation of the value.

ALGORITHM

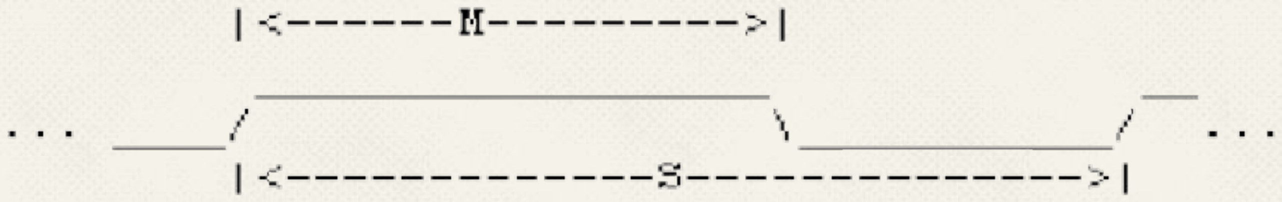
```
1. Set context = 0
2. context = context + N
3. if (context >= M)
       context = context - M
       send 1
   else
       send 0
4. Repeat from step 2
```

**N** = data to be sent

**M** = number of clock cycles used to send the data

# Modes PWM

- Mode **MSEN=0** (default mode), data to be sent is interpreted as the value N. The number of clock cycles (range) used to send data is the value M of the algorithm. Pulses are sent within this range so that the resulting duty cycle is N/M. The channel sends its output continuously as long as the data register is used or the FIFO is used and it is not empty.
- Mode **MSEN=1**, the PWM channel does not use the algorithm, instead it sends serial data with the M/S ratio. M is the data to be sent, and S is the range. This mode may be preferred if high frequency modulation is not required. The channel sends its output continuously as long as the data register is used or the FIFO is used and it is not empty.



# Key Features

**External Interfaces**

Camera — enable the Raspberry Pi Camera Module.

SSH — allow remote access to your Raspberry Pi from another computer using SSH.

VNC — allow remote access to the Raspberry Pi Desktop from another computer using VNC.

SPI — enable the SPI GPIO pins.

I2C — enable the I2C GPIO pins.

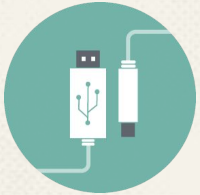
Serial — enable the Serial (Rx, Tx) GPIO pins.

1-Wire — enable the 1-Wire GPIO pin.

Remote GPIO — allow access to your Raspberry Pi’s GPIO pins from another computer.

Input/output devices based on protocols such as HDMI, USB, and Ethernet.

# Key Features



### Power Supplies

Micro USB Port (5V at 2A)

Via the GPIO

Via the USB Ports

### Memory Devices

Memory

2, 4, or 8GB LPDDR4-3200 SDRAM

Storage

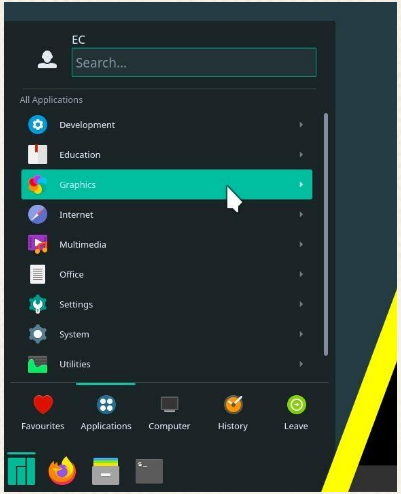
MicroSDHC slot

# Operating systems

- Raspberry Pi OS → used for desktop
- DietPi → used for server
- Batocera → used for gaming
- LibreElec (executes kodi)

  ↓

  used for media-center



# Raspberry Pi OS

- The Raspberry Pi OS is a free operating system based on Debian.
  - Many Debian packages (35,000) are available on Raspberry Pi.
  - Focus on stability and performance.
  - Preinstalled applications like Chromium, Minecraft Pi, Scratch, and LibreOffice.
  - APT Packet Manager helps installing more packages.
  - 64 bit version still in test.

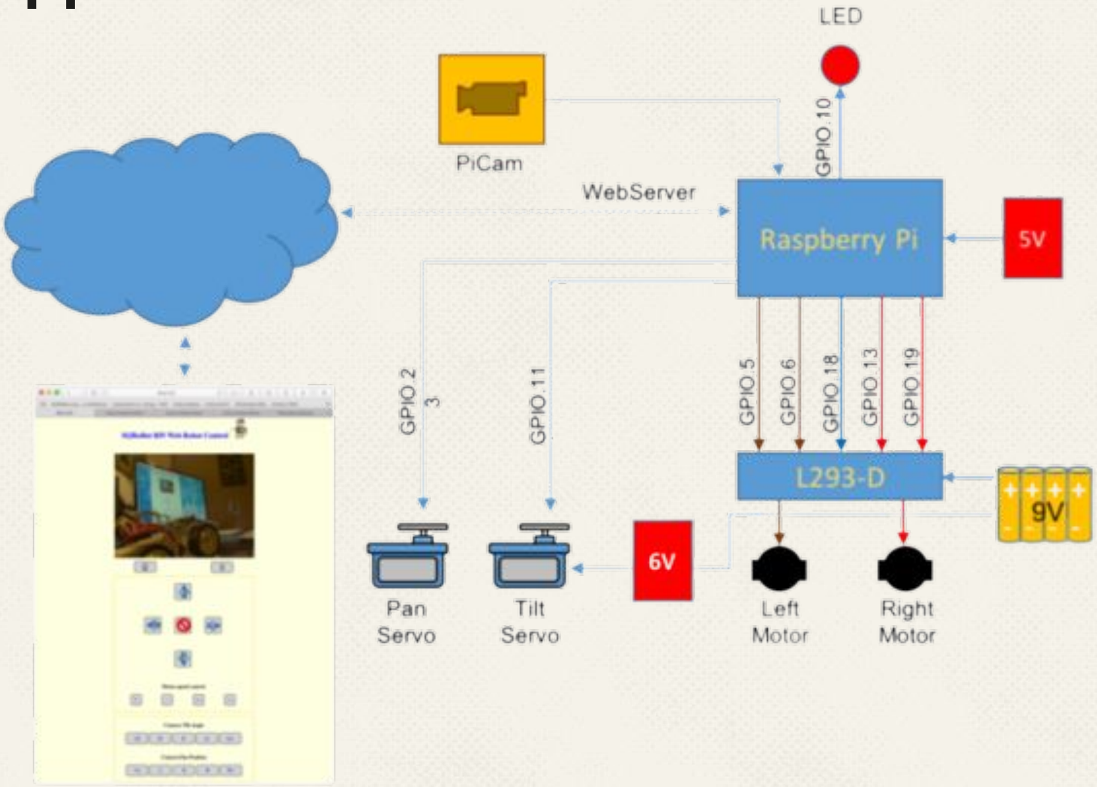
- Today Raspberry Pi OS comes with the PIXEL desktop (based on LXDE), which makes the Raspberry Pi usable as a daily desktop computer.
- It is compatible, reliable, and adaptable to most beginner projects. Raspberry Pi OS works well on any Raspberry Pi model.

# Languages and development tools

- Scratch, Python, HTML5, JavaScript, JQuery, Java, C, C++, Perl, Erlang
- IDE that are normally used → Visual Studio Code, Eclipse
- Most used IDEs for Raspberry PI
  - Geany (Powerful and lightweight development environment for Raspberry Pi, supports 50 languages)
  - BlueJ for java
  - Thonny is simple and ideal for Python
  - Code::Blocks for C and C++
  - Lazarus (have advantages in compiling and executing code)

# Applications

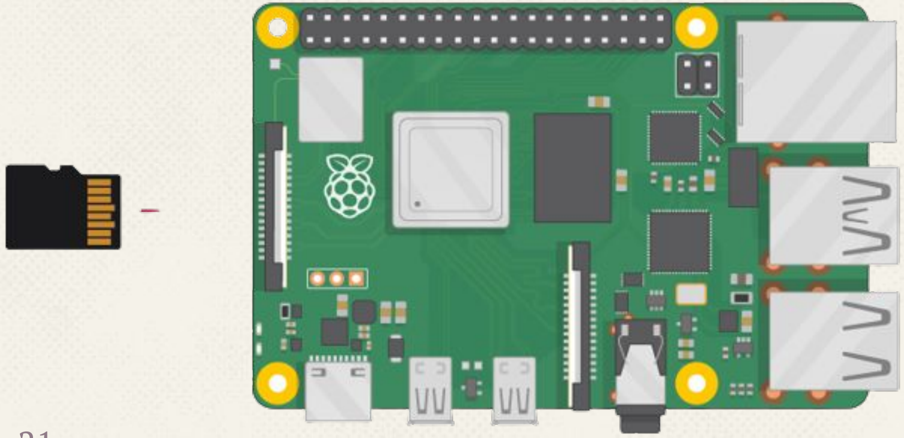
- Desktop PC
- Wireless print server
- Media Usage
- Game Servers
- Retro Gaming Machine
- Robot Controller
- Stop Motion Camera
- Time-lapse Camera
- FM Radio Station
- Web Servers




Controlling a Raspberry Pi robot via the Internet

# Card Images

- The first thing to consider is the storage you are using on Raspberry Pi.
- A solid backup is mandatory.
- Create an image on Windows, Linux or macOS.
- Copy an image to a SD card.



# Remote Access and Upgrades

- Sometimes you need to access a Raspberry Pi without connecting it to a monitor.
- It is possible to connect to the Raspberry Pi from another machine. But to do so, it requires knowing the IP address.
- Any device connected to a Local Area Network is assigned an IP address. It is possible to connect to the Raspberry Pi from another machine using SSH or VNC.
- **How to Find Raspberry Pi's IP Address?**
  - Router devices list
  - Resolving raspberrypi.local with mDNS
  - nmap command
  - Getting IPv6 addresses by pinging from a second device
  - Getting the IP address of a Raspberry Pi using a smartphone

# Bibliography


- http://sandsoftwaresound.net/raspberry-pi-4-arm-cortex-a72-processor/
- https://datasheets.raspberrypi.com/bcm2711/bcm2711-peripherals.pdf?fbclid=IwAR1mYG5Zi8yi7mNeEFxqEq3dTnofJRHTeyt2W0lWJnY6G6LrsfWFgrbvop8
- https://linuxhint.com/gpio-pinout-raspberry-pi/
- https://projects.raspberrypi.org/en/projects/raspberry-pi-using/9
- https://thepihut.com/blogs/raspberry-pi-tutorials/how-do-i-power-my-raspberry-pi
- https://en.wikipedia.org/wiki/Raspberry_Pi
- https://raspberrytips.com/best-os-for-raspberry-pi/
- https://www.educba.com/uses-of-raspberry-pi/
- https://www.raspberrypiportugal.pt/controlando-um-raspberry-pi-robot-pela-internet/
- https://www.raspberrypi.com/documentation/computers/remote-access.html