



deti 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

Índice

Definição	2
Especificações	2
CPU	2
Memória	4
Espaço de endereçamento	5
Periféricos	6
PWM	6
7.1 Introdução	6
7.2 Implementação do PWM	7
7.3 Modos do PWM	8
Características Principais	9
8.1 Interfaces	9
8.2 Fontes de Alimentação	9
8.3 Dispositivos de Memória	9
Sistemas Operativos	9
9.1 Raspberry Pi OS	10
Linguagens e ferramentas de desenvolvimento	10
Aplicações	11
Imagens do Cartão	12
Acesso Remoto e Atualizações	12
Bibliografia	13

1. Definição

Raspberry Pi 4 é capaz de fazer o mesmo que um computador. É possível configurar um sistema operativo e conectar circuitos para realizar tarefas específicas.

O Raspberry é composto por uma entrada USB tipo-C, uma porta Display Serial Interface (DSI), tem bluetooth, portas micro-HDMI, system-on-a-chip que é um microchip com todos os circuitos electrónicos e peças necessárias para um determinado sistema, num único circuito integrado (IC).

Tem GPIOs, memória RAM, uma porta Camera Serial Interface (CSI), uma entrada para cabo de áudio e vídeo, power over ethernet, duas portas USB 2.0, duas portas USB 3.0 e uma porta ethernet.

2. Especificações

Esta placa tem um processador BCM2711 com 64 bits quad-core Arm Cortex com clock de 1.5Ghz. Tem uma memória LPDDR4-3200 SDRAM de 2GB, 4GB ou 8GB, duas portas micro HDMI, portas 2-lane MIPI DSI display, 2-lane MIPI CSI camera, 4-pole stereo audio, e portas de vídeo.

Tem wireless de 2.4Ghz e 5.0 Ghz, suporta bluetooth 5.0 e BLE Gigabit Ethernet.

Relativamente à multimedia suporta gráficos 3.0, no formato H.265 ou H.264. Usa-se um micro cartão SD para guardar o sistema operativo e os dados. É alimentada a 5 volts de energia, pode ser pelo PC (porta USB tipo-C), pelos GPIOs, ou ainda por Power over Ethernet (PoE).

3. CPU

O Raspberry Pi 4 é alimentado por um processador da Broadcom (BCM2711) que é um processador com 4 cores ARM Cortex-A72 quad-core com clock a 1,5GHz.

O ARM Cortex-A72 dentro do processador BCM2711 tem um núcleo de CPU e um subsistema de memória muito melhorado.

O Cortex-A72 (Figura 1), executa a decodificação de instruções de três modos e pode emitir até cinco operações por ciclo.

Traduz as instruções ARM em micro-operações, em alguns casos, até pode realizar a fusão de micro-operações. Efectua a renomeação dos registos, deixando as micro-operações (instruções) executar quando os dados do programa estão prontos podendo ocorrer execução fora de ordem.

The Cortex-A72 executa micro-operações para 8 pipelines de execução:

1. Branch: micro-operações branch.
2. Integer 0: ALU micro-operações inteiros.
3. Integer 1: ALU micro-operações inteiros.
4. Integer Multi-Cycle: shift de inteiros na ALU, multiplicação, divisão, CRC (cyclic redundancy check) e somatório de diferenças absolutas.

5. FP/ASIMD 0: ASIMD ALU, ASIMD multiplicação de inteiros, FP conversão, FP adição, FP multiplicação, FP divisão.
6. FP/ASIMD 1: ASIMD (advanced single instruction multiple data) ALU, FP adição, FP multiplicação, FP raiz quadrada e ASIMD shift.
7. Load.
8. Store.

O Cortex-A72 pode realizar uma operação de load e uma operação de store em cada ciclo, pois tem pipelines de load e store separadas e permite a execução especulativa (branch prediction).

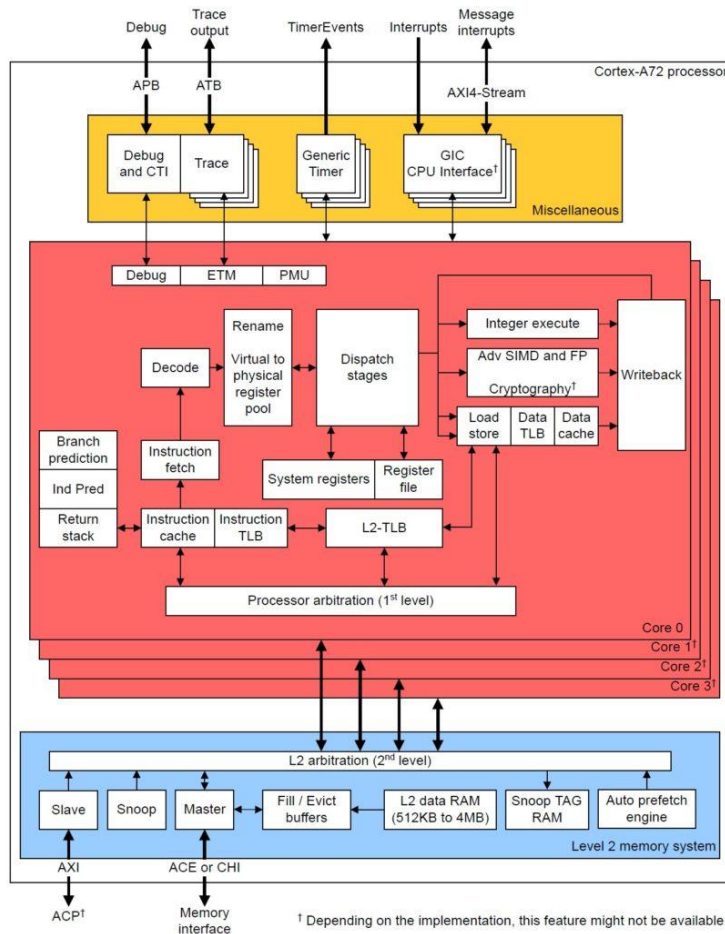


Figura 1 - Diagrama do processador Cortex-A72

4. Memória

O Cortex-A72 organiza a memória física numa hierarquia com a memória mais rápida/menor (os registos) que se encontra perto da unidade aritmética/lógica (ALU) e a memória mais lenta/maior (RAM) que está distante e fora do chip.

Os dados e instruções são lidos e escritos em chunks eficientes, tornando os dados e instruções disponíveis quando necessários pelos registos e pela ALU.

Os chunks são chamados "linhas de cache".

Devido à memória cache, os programas correm mais rapidamente quando (re)utilizam dados que estão juntos na memória (ou seja, ocupam a mesma linha cache) sendo os mais recentes acedidos. Estas noções são chamadas de "localidade espacial" e "localidade temporal".

Cada núcleo tem uma cache de instruções e uma cache de dados (Figura 2).

Os quatro núcleos partilham a cache de nível 2. Esta cache de nível 2, tem capacidade 1MB, está organizada num conjunto associativo de 16 vias e cada linha tem 64B.

A capacidade da cache de instruções nível 1 é 48KB, está organizada num conjunto associativo de 3 vias e cada linha tem 64B.

A capacidade da cache de dados nível 1 é 32KB, está organizada num conjunto associativo de 2 vias e cada linha tem 64B.

Os processadores têm uma ou mais unidades de gestão de memória (MMU) que quebram a RAM física em páginas lógicas. Este esquema é chamado "memória virtual".

A MMU traduz os endereços de programas lógicos (de loads, stores e instruction fetches) em endereços físicos da RAM.

Os buffers de tradução (TLB) são estruturas de hardware que contém a informação de tradução de endereços mais recentemente utilizada (MRU), ou seja, onde uma página lógica está localizada na memória física.

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

Figura 2 - Memória cache disposta por níveis.

5. Espaço de endereçamento

O espaço de endereçamento é dividido em 2 espaços principais, o full 35-bit address e o legacy master 32-bit address.

Tem um modo de periféricos low que modifica o espaço de endereçamento dos periféricos quando visto do lado do processador.

Pode-se ver na Figura 3 como estes tipos de endereçamento se interligam.

O full 35-bit address é acedido pelo legacy master e pelo CPU. Tem duas caches, uma allocating (0x4_0000_0000 até 0x4_3FFF_FFFF) e outra non-allocating (0x4_8000_0000 até 0x4_BFFF_FFFF) que armazenam o primeiro gigabyte de memória SDRAM.

O Legacy master 32 bits address é usado por um software que acede aos periféricos usando DMA (endereços 0xC000_0000 até 0xFFFF_FFFF). Se o DMA precisar de aceder à memória RAM para além dos primeiros 1GB, a janela pode ser movida usando os bits Page/Pagelite. Tem periféricos principais (0xFC00_0000 até 0x7FFF_FFFF), cache allocating (0x0000_0000 até 0x3FFF_FFFF) e non-allocating (0x8000_0000 até 0xBFFF_FFFF).

O ARM physical addresses modifica o espaço de endereçamento dos periféricos quando visto do lado do processador. Tem RAM que vai desde o endereço 0x0_0000_0000 até ao tamanho definido no ficheiro config.txt (também acontece no full), tem a seleção VideoCore, periféricos main, periféricos locais, cache allocating e non-allocating.

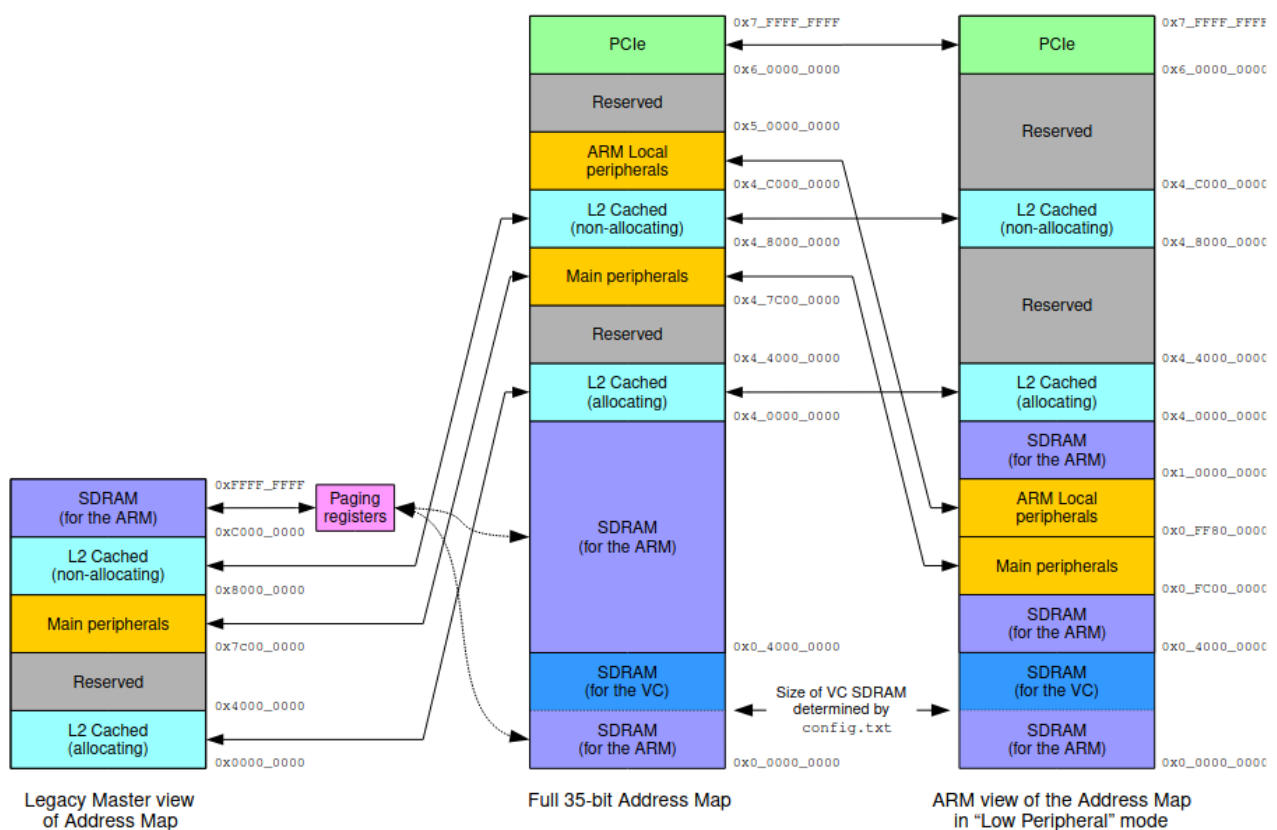


Figura 3 - Espaço de endereçamento e respetivos endereços

6. Periféricos

Os periféricos podem ser utilizados como entrada e saída controlada por software.

Os pinos GPIO podem ser comutados (multiplexados) em vários outros modos suportados por blocos periféricos dedicados tais como DPI, I2C, UART, SPI, PCM e PWM.

Os periféricos I2C, UART e SPI foram adicionados ao processador BCM2711 do Raspberry Pi 4 Model B.

Isto dá aos utilizadores muito mais flexibilidade ao anexar hardware adicional em comparação com os modelos mais antigos.

O PWM, que irá ser abordado de seguida, utiliza os GPIOs 12, 13, 18 e 19.

PIN	NAME		NAME	PIN
01	3.3V DC Power		5V DC Power	02
03	GPIO02 (SDA1, I ² C)		5V DC Power	04
05	GPIO03 (SDL1, I ² C)		Ground	06
07	GPIO04 (GPCLK0)		GPIO14 (TXD0, UART)	08
09	Ground		GPIO15 (RXD0, UART)	10
11	GPIO17		GPIO18(PWM0)	12
13	GPIO27		Ground	14
15	GPIO22		GPIO23	16
17	3.3V DC Power		GPIO24	18
19	GPIO10 (SP10_MOSI)		Ground	20
21	GPIO09 (SP10_MISO)		GPIO25	22
23	GPIO11 (SP10_CLK)		GPIO08 (SPI0_CE0_N)	24
25	Ground		GPIO07 (SPI0_CE1_N)	26
27	GPIO00 (SDA0, I ² C)		GPIO07 (SCL0, I ² C)	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12 (PWM0)	32
33	GPIO13 (PWM1)		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Figura 4 - GPIOs da placa Raspberry Pi 4 Model B, com destaque para os GPIOs usados pelo PWM.

7. PWM

7.1 Introdução

A modulação por largura de pulso de um sinal ou em fontes de alimentação envolve a modulação de sua razão cíclica para transportar qualquer informação sobre um canal de comunicação ou controlar o valor da alimentação entregue à carga.

É um processo ou técnica de modulação utilizada na maioria dos sistemas de comunicação para codificar a amplitude de um sinal diretamente numa largura de

pulso ou duração de outro sinal (geralmente um sinal portador) para transmissão.

O processador BCM2711 tem duas instâncias deste bloco como se pode ver na Figura 5. Denominam-se PWM0 (cujo endereço base de registo é 0x7E20C000) e PWM1 (cujo endereço base de registo é 0x7E20C800).

Cada controlador PWM incorpora as seguintes características:

- Dois fluxos de bits de saída independentes, cronometrados a uma frequência fixa;
- Fluxos de bits configurados individualmente para produzir PWM ou uma versão em série de uma palavra de 32 bits;
- As saídas PWM têm resoluções de saída variáveis;
- Modo Serialise configurado para ler dados de um bloco de armazenamento FIFO, que pode armazenar até 64 palavras de 32 bits;
- Ambos os modos são marcados pelo `clk_pwm` que por default é de 100 MHz, mas pode ser alterado.

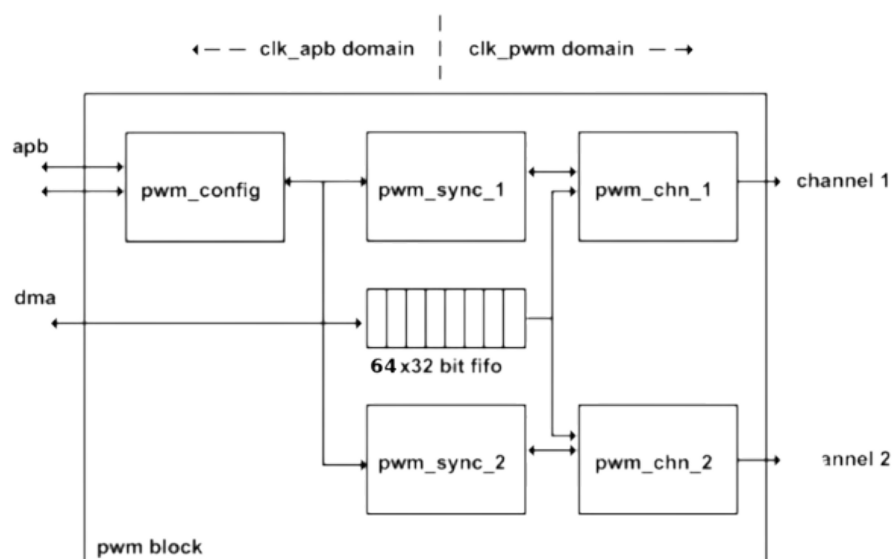


Figura 5 - Bloco PWM

7.2 Implementação do PWM

Na implementação do PWM o N representa os dados a serem enviados e o M representa o número de ciclos de relógio.

O valor representado como uma relação de N/M é transmitido ao longo de um canal, em série, no qual o valor é representado pelo ciclo de funcionamento do sinal de saída.

Para enviar o valor N/M numa sequência periódica de M ciclos, a saída é igual a 1 para N ciclos e é igual a 0 para (M-N) ciclos.

A sequência desejada deve ter uns e zeros espalhados uniformemente, para que durante um período arbitrário de tempo o duty cycle tenha uma boa aproximação do valor real.

Como se pode ver na Figura 6, inicializa-se uma variável a 0, depois vai-se atualizando o seu valor e se o valor foi maior ou igual a M subtraí-se M e retorna-se 1 senão retorna-se 0.


```

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

```

Figura 6 - Algoritmo de PWM.

7.3 Modos do PWM

Falando do modo PWM este tem 2 submodos, o submodo MSEN=0 (default) e o submodo MSEN=1.

No submodo MSEN=0 (utiliza o algoritmo anterior), tal como anteriormente, o N representa os dados a serem enviados e o M representa o número de ciclos de relógio. Os impulsos são enviados dentro deste intervalo, para que o ciclo de funcionamento resultante seja N/M . O canal envia a sua saída de uma forma contínua enquanto o data register estiver a ser usado ou se se usar um FIFO e este não estiver vazio.

No submodo MSEN=1, o canal PWM não usa o algoritmo explicado. Em vez disso, este envia dados em série com a M/S ratio, onde como se pode ver na Figura 7 o M são os dados a serem enviados, e S é o intervalo. Este submodo pode ser preferido se não se usar uma modulação com alta frequência. Este canal envia a sua saída continuamente enquanto o data register estiver a ser usado ou se se usar um FIFO e este não estiver vazio.

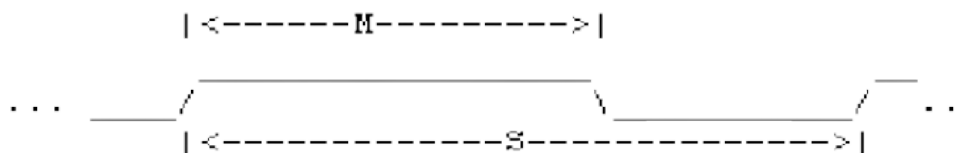


Figura 7 - Ciclo de relógio no submodo MSEN=1.

Exemplo para o submodo MSEN=0

Um valor representado como uma relação de N/M pode ser transmitido ao longo de um canal de série com modulação de largura de pulso (PWM), no qual o valor é representado pelo ciclo de trabalho do sinal de saída. Para enviar o valor N/M dentro de uma sequência periódica de ciclos M, a saída deve ser 1 para N ciclos e 0 para (M-N) ciclos. A sequência desejada deve ter 1s e 0s espalhados o mais uniformemente possível, para que durante qualquer período arbitrário de tempo o ciclo de funcionamento atinja a aproximação mais próxima do valor.

Isto pode ser mostrado na tabela seguinte onde 4/8 é modulado (N=4, M=8). Seguindo o algoritmo acima abordado na Figura 6.

Bad	0	0	0	0	1	1	1	1	0	0	0	0
Fair	0	0	1	1	0	0	1	1	0	0	1	1
Good	0	1	0	1	0	1	0	1	0	1	0	1

Figura 8 - Tabela da modulação 4/8.

8. Características Principais

8.1 Interfaces

Raspberry Pi 4 Model B aceita as muitas interfaces como câmera, SSH, VNC, SPI, I2C, serial, 1-wire, remote gpio e utiliza uma variedade de dispositivos de entrada/saída baseados em protocolos tais como HDMI, USB, e Ethernet para comunicar com o mundo exterior.

8.2 Fontes de Alimentação

Esta placa pode ser alimentada pela porta micro USB de 5V a 2A, através da GPIO ou ainda por portas USB.

8.3 Dispositivos de Memória

Tem uma memória LPDDR4-3200 SDRAM de 2GB, 4GB ou 8GB usando um cartão MicroSDHC para guardar o sistema operativo e os dados.

9. Sistemas Operativos

Existem vários sistemas operativos para o Raspberry Pi 4 Model B.

O mais usado é o Raspberry Pi OS (Figura 9), mas existem muitos outros como o DietPi (Figura 10) usado para servidores, o Batocera (Figura 11) usado para gaming e o LibreElec (Figura 12) usado para media-center que diferencia dos outros porque apenas executa o kodi.



Figura 9 - Raspberry Pi OS

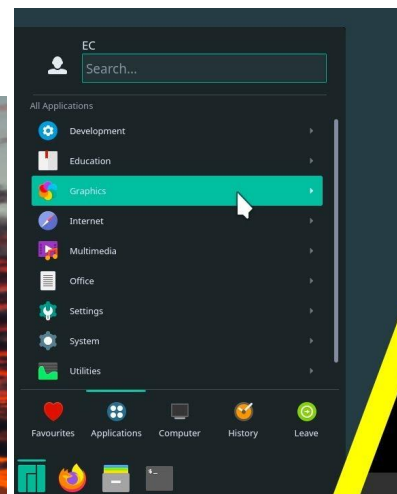


Figura 10 - DietPi

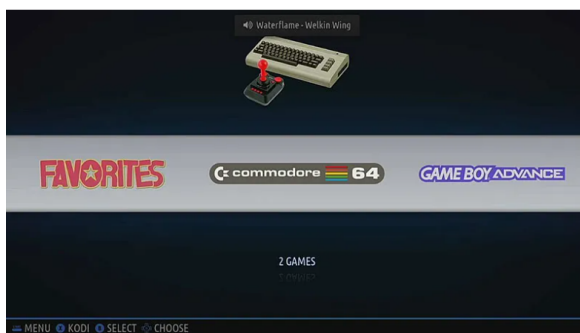


Figura 11 - Batocera

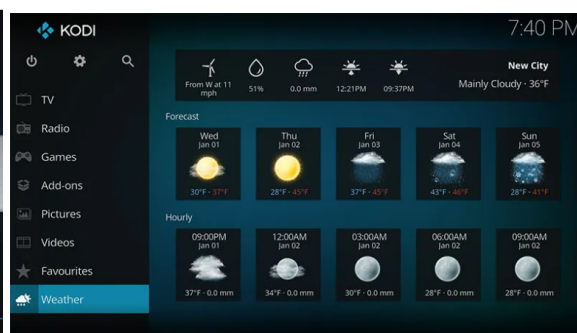


Figura 12 - LibreElec

9.1 Raspberry Pi OS

Relativamente ao Raspberry Pi OS este é um sistema debian gratuito com muitos pacotes debian. Foca-se na estabilidade e desempenho. Tem pré-instalações como o Chromium, Minecraft Pi, Scratch e o LibreOffice. Para instalar mais pacotes existe o APT packet manager.

Hoje em dia, este sistema operativo vem com o desktop PIXEL que faz com que se possa usar a Raspberry Pi como um desktop do nosso PC.

Este sistema é compatível, fiável e adaptável para a maioria dos projetos, trabalhando em todos os modelos Raspberry Pi.

10. Linguagens e ferramentas de desenvolvimento

Pode-se programar esta placa em várias linguagens como o Scratch, Python, HTML5, JavaScript, JQuery, Java, C/C++, Perl, Erlang, sendo o Python mais

adequada para utilizadores de uso geral.

Como ambiente de desenvolvimento pode-se usar os IDEs comuns no dia a dia de um programador como o Visual Studio Code ou o Eclipse.

No contexto da Raspberry os mais usados são o Geany que é bastante poderoso e leve, suportando 50 linguagens de programação, o BlueJ usado para a linguagem Java, o Thonny que é simples e fácil de se usar para a linguagem Python, o Code::Blocks usado para C/C++ e por fim o Lazarus que oferece vantagens na compilação e execução do código.

11. Aplicações

O Raspberry Pi 4 Model B pelas suas características, tem muitas aplicações no mundo real como:

- Desktop de um PC;
- Servidor para imprimir por wireless;
- Meios de comunicação;
- Servidores de jogos;
- Máquinas de retro gaming;
- Controlar robôs;
- Time-lapse camera;
- Stop motion camera;
- FM Radio Station;
- Web Servers.

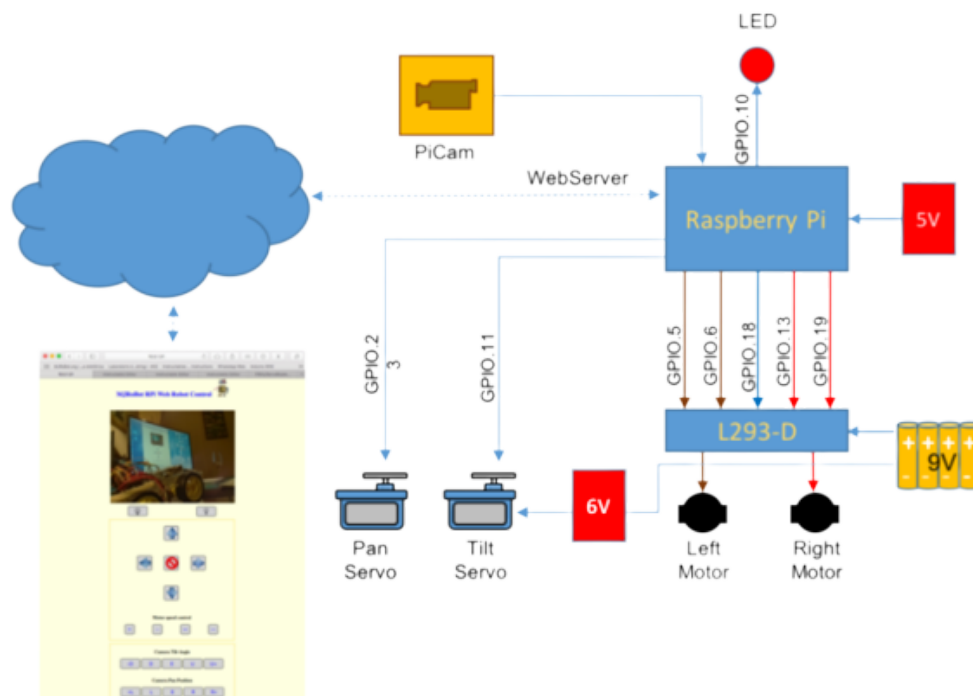


Figura 13 - Raspberry Pi a ser usado para muitas aplicações

12. Imagens do Cartão

A primeira coisa a considerar é o armazenamento que estamos a utilizar no Raspberry Pi.

É necessário fazer uma cópia de segurança.

Criar uma imagem para o cartão SD pode ser feito facilmente em Windows, Linux ou macOS.

Após criada a imagem esta é copiada para o cartão SD.

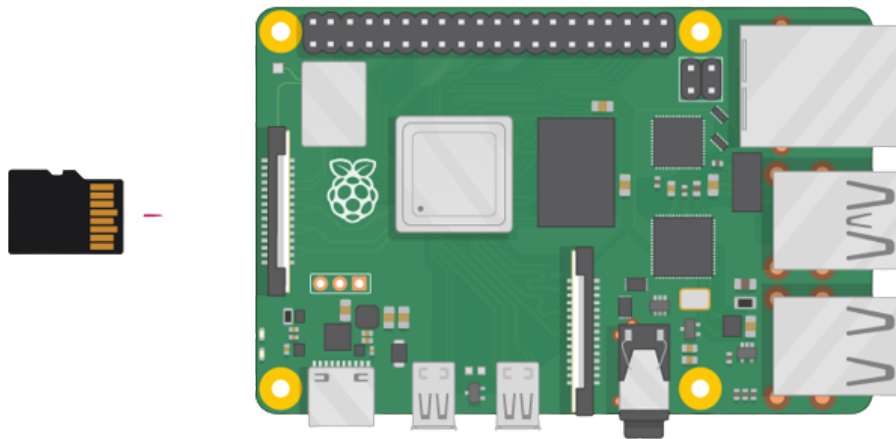


Figura 14 - Cartão SD numa Raspberry Pi 4 Model B

13. Acesso Remoto e Atualizações

Por vezes é necessário aceder a um Raspberry Pi 4 Model B sem o ligar a um monitor.

É possível ligar-se a esta placa a partir de outra máquina, mas para o fazer, é necessário conhecer o seu endereço IP.

A qualquer dispositivo ligado a uma Rede de Área Local é atribuído um endereço IP. Assim é possível ligar-se à placa através de uma máquina usando ssh ou vnc.

Há várias hipóteses para encontrar o endereço IP do Raspberry Pi 4 Model B:

1. Lista de dispositivos de router;
2. Resolver o raspberrypi.local com mDNS;
3. Comando nmap;
4. Obtenção de endereços IPv6 pingando a partir de um segundo dispositivo;
5. Utilizando um smartphone.

Tendo efetuado a ligação por SSH ou VNC com o Raspberry Pi 4 Model B é possível então fazer atualizações remotamente.

14. Bibliografia

- <http://sandsoftwaresound.net/raspberry-pi-4-arm-cortex-a72-processor/>
- <https://datasheets.raspberrypi.com/bcm2711/bcm2711-peripherals.pdf?fbclid=IwAR1mYG5Zi8yi7mNeEFxqEq3dTnofJRHTeyt2W0IWJnY6G6LrsfWFgrbvop8>
- <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>