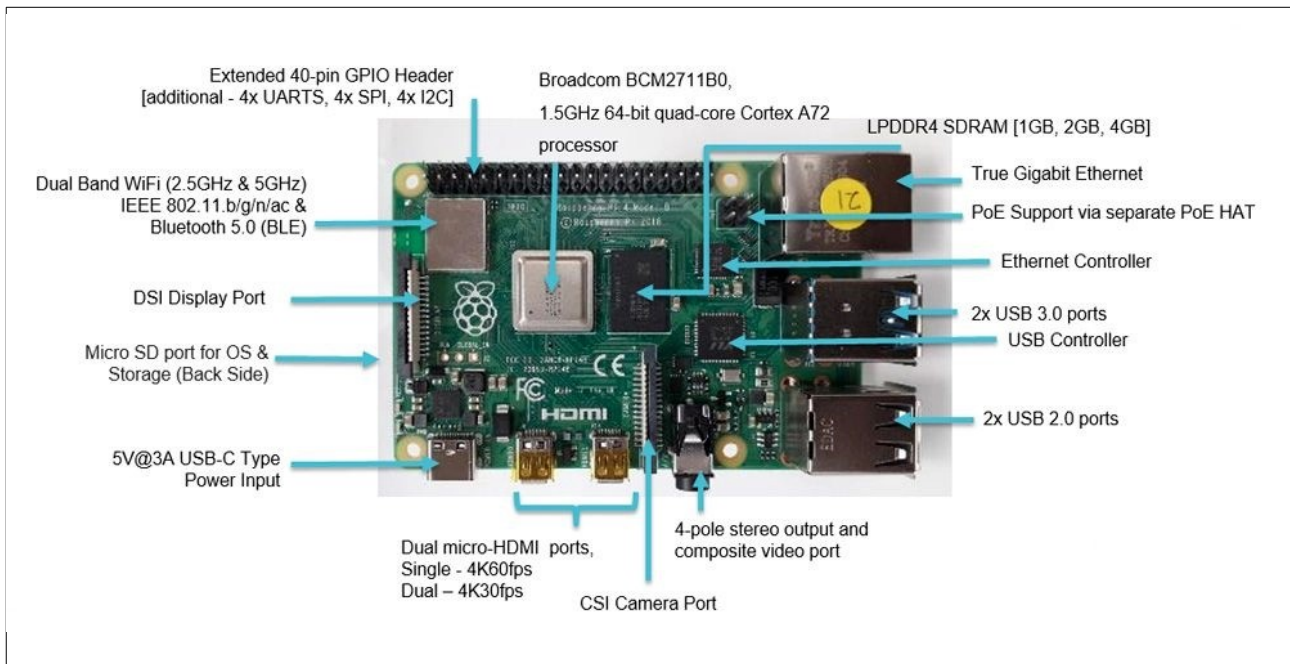


RASPBERRY PI

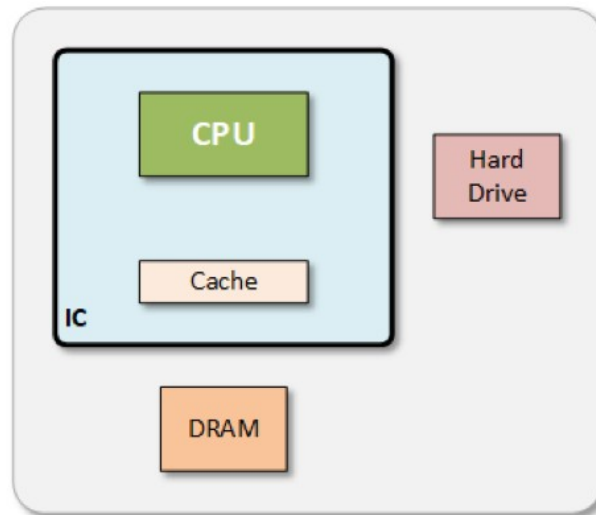


https://es.wikipedia.org/wiki/Raspberry_Pi

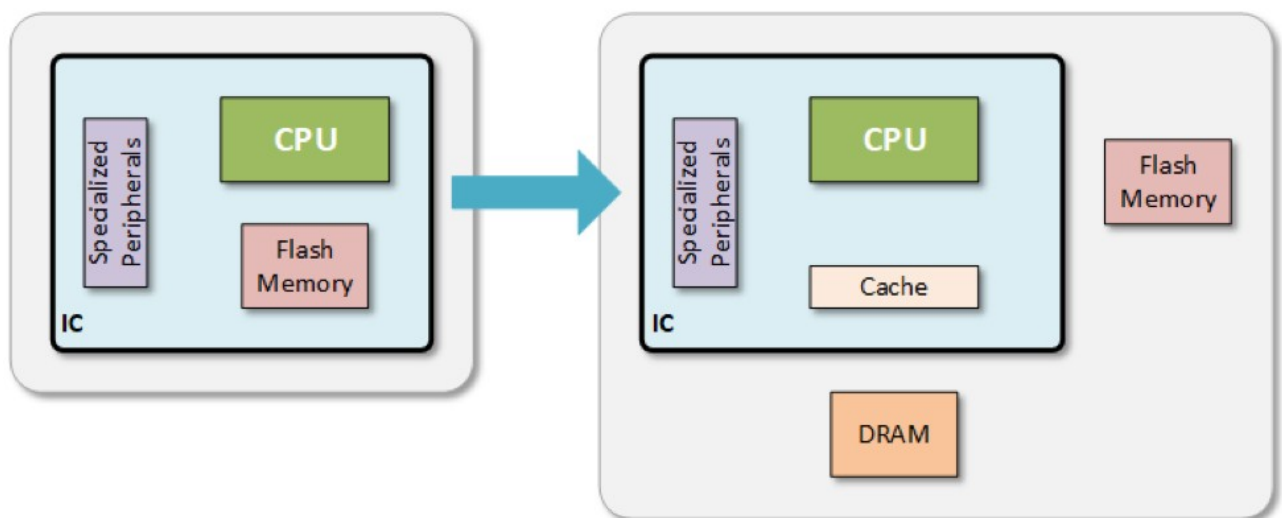
<https://www.raspberrypi.org/>

<https://www.raspberrypi.com/>

<https://www.raspberrypi.com/documentation/>



Microprocessor



Microcontrollers

¿Qué es un procesador ARM Cortex?

<https://www.profetolocka.com.ar/2021/02/01/que-es-un-procesador-arm-cortex/>

Cada vez con mas frecuencia podemos encontrar placas o kits de desarrollo que incluyen un procesador ARM Cortex. Desde una computadora de escritorio a un microcontrolador de un par de dólares, pasando por el procesador de un teléfono móvil o una Raspberry Pi todos estos dispositivos pueden contener en su interior alguna variedad de estos potentes procesadores. En este artículo haremos una introducción general, analizando algunas de sus características y cuáles son los mas accesibles para utilizar en nuestros proyectos.

ARM, el origen

[ARM Holdings](#) es una empresa británica que se dedica al diseño de procesadores. No los fabrica y comercializa como si lo hacen Intel o AMD sino que licencia sus diseños para que otras empresas los adapten o complementen según sus necesidades y los fabriquen para sus productos. Por ejemplo Qualcomm fabrica sus chips Snapdragon, que están presentes en un elevado porcentaje de dispositivos móviles a partir de un diseño de ARM y recientemente también Apple comenzó a hacer lo mismo. Por esta razón se suele decir que estos chips poseen una **arquitectura ARM**, ya que están basados en este diseño.



CISC y RISC

Los procesadores pueden clasificarse según muy distintos criterios. Uno de ellos es la **complejidad**

del set de instrucciones que poseen, que da lugar a dos categorías: **CISC y RISC**.

La arquitectura **CISC** (Complex Instruction Set Computing) incluye **instrucciones muy potentes**, capaces de resolver tareas complicadas en pocos pasos, mientras que la arquitectura **RISC** (Reduced Instruction Set Computing) emplea **instrucciones mas sencillas**, que deben combinarse en un programa mas extenso para una situación mas compleja. La gran ventaja de los procesadores RISC es que consumen **menos energía**. Los primeros son mas adecuados para aplicaciones que requieren mucho poder de procesamiento, como una computadora de escritorio, mientras que los segundos encuentran su nicho de aplicación en dispositivos mas sencillos pero donde el consumo de energía es importante, como en las computadoras portátiles y sobre todo los teléfonos móviles. Los procesadores **ARM** emplean la arquitectura CISC y combinan potencia con bajo consumo, razón por la cual han triunfado en el mercado de dispositivos móviles, al punto de que casi 3 de cada 4 de estos aparatos tienen un procesador con esta arquitectura en su interior, aunque en la actualidad también están haciendo pie en las computadoras de escritorio.

Distintas familias para distintas aplicaciones

Como decía al principio, los procesadores ARM se pueden encontrar en dispositivos tan diferentes como una computadora o una placa de pocos dólares. Para cubrir un abanico tan amplio de aplicaciones (y costos) existen varias **familias** y modelos de chips ARM.

El primer chip ARM fue el ARM1, diseñado en 1985 y el primer diseño en ser comercializado fue el ARM2 al año siguiente. Los sucesivos modelos se fueron denominando mas o menos de esta forma, con un número hasta el ARM11 que apareció en el 2002. A partir de allí surgió la denominación mas moderna de **Cortex**, adoptada a partir del año 2004 hasta la actualidad, agregando una letra que identifica para que tipo de aplicaciones está destinado el diseño:

Cortex-A (Application): Para aplicaciones como teléfonos, tabletas, smart TVs o mini PCs. Son procesadores capaces de correr un sistema operativo, como Linux. Se los puede encontrar en placas como las Raspberry Pi, Orange Pi y BeagleBone.

Cortex-R (Real Time): Para aplicaciones de tiempo real como en equipamiento de redes de datos o en automóviles.

Cortex-M (Microcontroller): Aplicaciones sencillas, de bajo costo y poco consumo de energía, como IoT y dispositivos embebidos.

Cortex-X (Custom): Aplicaciones mas exigentes como computadoras de escritorio.

Esto significa que, salvo que a menos que hagamos desarrollos de aplicaciones exigentes, lo mas probable es que utilicemos en nuestros diseños algún procesador de la familia **Cortex-M**.

La familia Cortex-M

Los procesadores que emplean la arquitectura **Cortex-M** son los mas empleados para desarrollos **simples, proyectos Maker o aplicaciones educativas**. Dentro de esta subfamilia podemos encontrar las siguientes variantes, todas de 32 bits:

M0: Es el modelo mas simple y pequeño, destinado a aplicaciones de bajo costo y bajo consumo de energía.

M0+: Versión mejorada del M0, con mas potencia y menor consumo de energía.

M1: Diseñado específicamente para ser utilizado con FPGAS.

M3: Un procesador de alto rendimiento, agrega mas instrucciones para realizar tareas complejas. Incluye instrucciones de división y multiplicación y un módulo de debug que facilita la depuración del software.

M4: Es equivalente a M3 con el agregado de instrucciones de procesamiento de señales (DSP).

M7: Es un núcleo de alta performance, equivalente al M4 pero con mejor gestión de la energía. Los buses internos de datos e instrucciones son de 64 bits en vez de 32 como en los modelos anteriores.

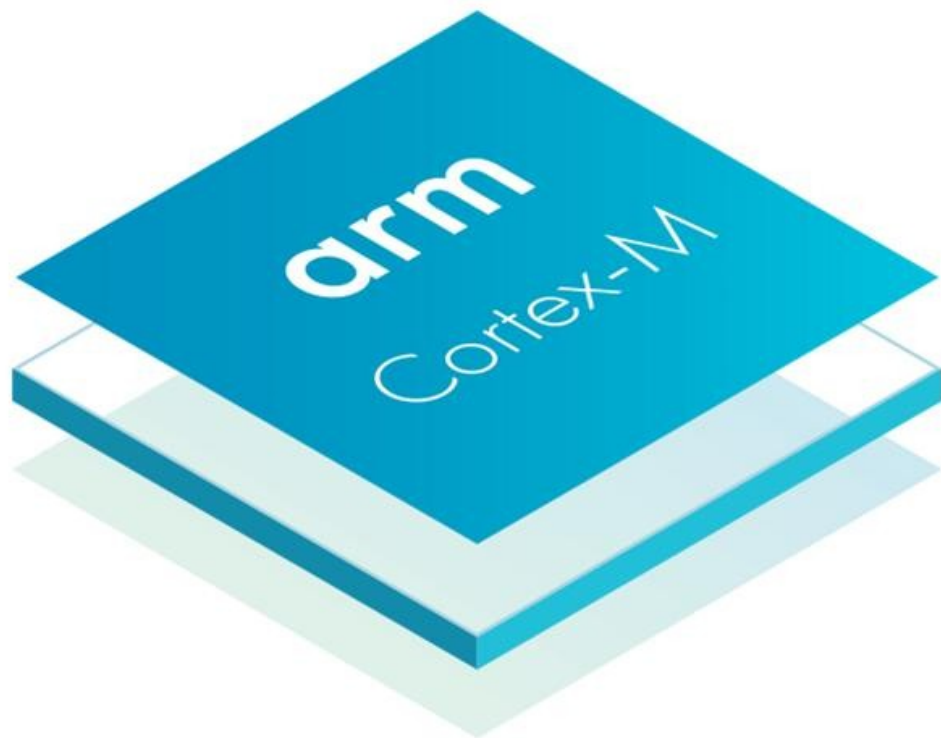
M23: Es equivalente a un m0+ con el agregado de instrucciones matemáticas y una mayor seguridad.

M33: Combina lo mejor de los modelos M4 y M23.

M35P: Es una mejora del M33 incluyendo mas funciones de seguridad.

M55: Incluye funciones que lo hacen particularmente útil en aplicaciones de Inteligencia Artificial (Machine Learning).

Estas son las características de los núcleos (cores) de ARM. Los distintos fabricantes, como Texas Instruments, ST, Microchip, Silicon Lab, Infineon (y un largo etcétera) pueden incluir mas o menos memoria, periféricos o funciones opcionales, lo que da lugar a una enorme variedad de chips y de funcionalidades para un extensísimo abanico de aplicaciones.



Módulos, placas y kits de desarrollo

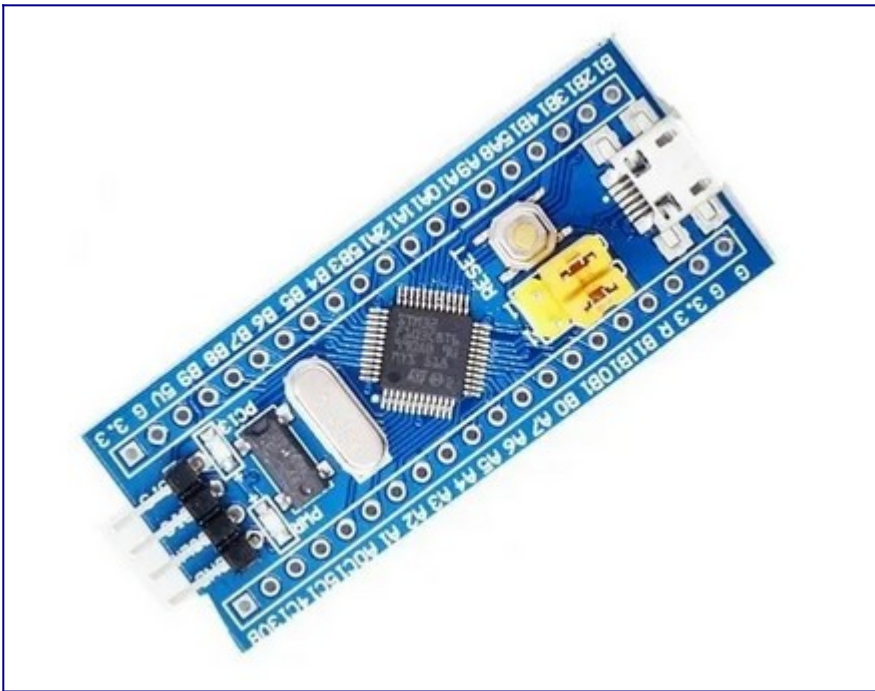
Si bien se pueden comprar los chips de los distintos fabricantes para soldarlos en nuestra propia PCB, también existen algunas placas y kits de desarrollo para hacer pruebas o proyectos sencillos. Veamos algunas de las que podemos encontrar en el mercado



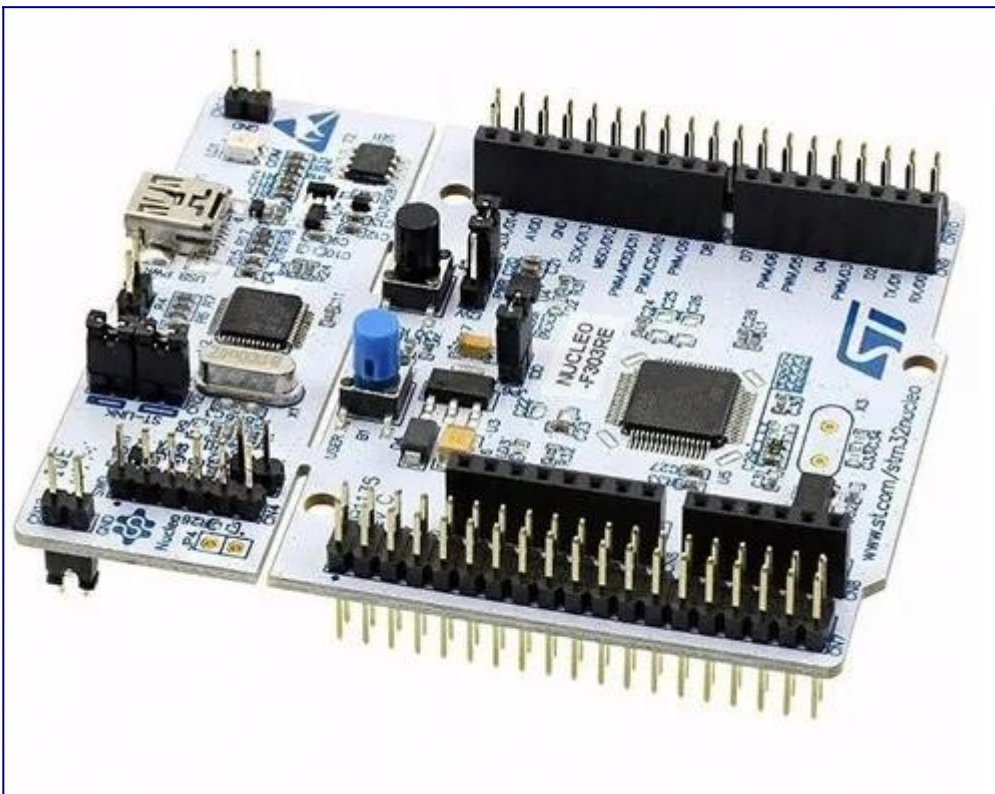
Trinket M0 de Afafruit (M0+)



Raspberry Pi Pico (M0+)



Blue Pill (M3)



STM32 Nucleo (M4)



Teensy 4.1 (M7)

¿Qué hacen los pines GPIO de la Raspberry Pi 4?

Una de las cosas más apreciadas y características de cualquier Raspberry Pi, incluyendo la nueva [Raspberry Pi 4](#), es la capacidad de utilizarlo para construir gadgets fantásticos de cualquier tipo, desde [robots](#) hasta consolas de juego retro, pasando por centros multimedia.

La mayoría de los sensores, motores, luces y otros periféricos que hacen posible estos proyectos se conectan a los pines GPIO de la Pi. Cada modelo, desde el Pi B+ en adelante tenía 40 pines GPIO, para ser precisos, de hecho, el Pi Zero y el Zero W tenían 40 agujeros para soldar los pines o cables.

Esta guía trata sobre las nuevas características de la Raspberry Pi, que todavía viene con 40 pines GPIO, pero que tiene algunas nuevas conexiones adicionales como I2C, SPI y UART.

Para que tus proyectos se hagan realidad es necesario que conozcas el pinout del Pi GPIO de la Raspberry Pi, que es el mapa y la explicación de lo que cada pin puede hacer. Mientras que algunos pines proporcionan electricidad, otros son GND (tierra) y otros se conectan a diferentes tipos de interfaces.

Pines GPIO (General Purpose Input Output)

El GPIO es el aspecto más simple, pero más accesible de Raspberry Pi. Los pines GPIO son digitales, lo que significa que sólo pueden tener dos estados, apagado o encendido.

Pueden tener una dirección para recibir o enviar corriente (entrada, salida respectivamente) y todo esto es totalmente controlable por lenguajes de programación como Python, JavaScript, node-RED y otros.

Los pines trabajan con una tensión de 3,3 V y un consumo máximo de corriente de 16 mA. Esto significa que podemos suministrar energía de forma segura desde un solo pin GPIO a través de una resistencia, uno o dos LEDs.



Para alimentar cualquier otra cosa que requiera más corriente, como un motor de corriente continua, es necesario utilizar componentes externos para evitar dañar el GPIO.

Para controlar GPIO con Python, primero debes importar una librería de código escrito previamente. El más común y difundido es el Rpi.GPIO, utilizado para crear miles de proyectos desde los primeros días de la Raspberry Pi.

Más recientemente, se ha introducido otra biblioteca, llamada GPIO Zero, que ofrece una experiencia más fácil para aquellos que son nuevos en Python y en la electrónica básica. Ambas bibliotecas están preinstaladas en el sistema operativo Raspbian.

En cuanto a la nomenclatura de los pines GPIO, hay varias maneras de identificarlos; el método más simple y obvio es definitivamente el que se refiere a su ubicación física.

Empezando por la parte superior izquierda del GPIO, y con esto nos referimos al pin más cercano a la ranura de la micro SD, tenemos el pin físico 1 que proporciona alimentación 3v3.

A la derecha de ese pin se encuentra el pin físico de 2 que proporciona 5v de potencia. El número de pines seguirá creciendo a medida que descendas por las columnas, como habrás entendido a la izquierda encontraremos todos los pines con números impares y a la derecha los pares.

Hay otro método utilizado, y es la numeración de pines de Broadcom (BCM) (numeración de pines GPIO) que, sin embargo, para el usuario medio puede parecer caótica, con GPIO17, 22 y 27 siguiéndose unos a otros con lo que parecería una pequeña reflexión sobre la numeración lógica.

En realidad, el mapeo de alfileres BCM se refiere a los alfileres GPIO que estaban conectados directamente al SoC del Raspberry Pi. Básicamente, tenemos conexiones directas al cerebro de nuestra Pi para conectar sensores y componentes para usar en nuestros diseños.

La mayoría de los tutoriales de Raspberry Pi que verás utilizan este esquema de numeración, también porque es el que cuenta con el apoyo oficial de la Fundación Raspberry Pi. Por lo tanto, si quieres entrar en este mundo, es bueno que aprendas este esquema de numeración tan pronto como sea posible porque, con el tiempo, se convertirá en el estándar que vas a usar. También ten en cuenta que los números de pin BCM y GPIO se refieren al mismo esquema. Por ejemplo, GPIO17 es lo mismo que BCM17.

Por último, algunos pines GPIO también tienen funciones secundarias que les permiten interactuar con diferentes tipos de dispositivos que utilizan los protocolos I2C, SPI o UART; por ejemplo, los pines GPIO3 y GPIO4 también son pines SDA y SCL I2C utilizados para conectar dispositivos a través del protocolo I2C.

Para utilizar estos pines con estos protocolos, debes habilitar las interfaces utilizando la aplicación Raspberry Pi Configuration disponible en el sistema operativo Raspbian, en el menú Preferencias.

I2C, SPI y UART: ¿Cuál usar?

En los siguientes párrafos veremos en detalle cuáles son las diferencias entre I2C, SPI y UART, pero si te preguntas cuál es el pin más apropiado para conectar un dispositivo en particular, la respuesta es simple: siempre revisa la hoja de datos, donde sin duda aparecerá reflejado.

Por ejemplo, una pantalla de LEDs pequeña puede requerir el SPI y otra puede requerir el protocolo I2C (UART es utilizado por muy pocos dispositivos). Si lees la documentación suministrada con un producto (siempre que tengas uno), normalmente te dirá qué pines de Pi debes utilizar.

Si eres un usuario de Raspberry Pi 4 entonces debes saber que ahora los pines SPI, I2C y UART disponibles son muchos más que en los modelos anteriores. Estas interfaces adicionales se activan superponiendo la estructura de árbol del dispositivo y pueden proporcionar cuatro conexiones SPI, I2C y UART adicionales.

I2C – Circuito Integrado Interno

El I2C es un protocolo serie de dos hilos de baja velocidad que se utiliza para conectar dispositivos que utilizan el estándar I2C. Los dispositivos que utilizan este estándar tienen una relación maestro-esclavo.

Puede haber varios maestros, pero recuerda que cada dispositivo esclavo requiere una dirección única, obtenida del fabricante NXP, anteriormente conocido como Philips Semiconductors.

Esto significa que podemos hablar con múltiples dispositivos a través de una única conexión I2C porque cada dispositivo es único y detectable a través de comandos de Linux como `i2cdetect`.

Como mencionamos anteriormente, el I2C tiene dos conexiones: SDA y SCL. ¿Cómo funcionan? Enviando datos desde y hacia la conexión SDA, con velocidad controlada por el pin SCL. El I2C es una forma rápida y sencilla de conectar una gama bastante heterogénea de componentes a su proyecto, desde pantallas LCD/OLED hasta sensores de temperatura, pasando por convertidores analógicos a digitales para su uso con fotoresistores y muchos más.

Aunque es un poco más difícil de entender que los pines GPIO estándar, el conocimiento adquirido al aprender I2C será de gran ayuda cuando entienda cómo conectar sensores más precisos para su uso en diferentes tareas.

La Raspberry Pi tiene dos conexiones I2C en GPIO2 y GPIO3 (SDA y SCL) y son I2C0 (maestro), mientras que los pines físicos 27 y 28 son pines I2C que permiten a la Pi comunicarse con el HAT (Hardware Atado en la Parte Superior) compatible añadido a la placa.

SPI – Interfaz periférica en serie

SPI es un segundo protocolo para conectar dispositivos compatibles disponibles en tu Raspberry Pi . Funciona de forma similar al I2C, ya que existe una relación maestro-esclavo entre Raspberry Pi y los dispositivos conectados.

SPI se utiliza generalmente para enviar datos a cortas distancias entre microcontroladores y componentes tales como registros de cambio, sensores e incluso una tarjeta SD. Los datos se sincronizan usando un reloj (SCLK en GPIO11) desde el maestro (el Raspberry Pi) y los datos se envían desde el Pi al componente SPI conectado usando el pin MOSI (GPIO10).

MOSI significa Master Out Slave In. De lo contrario, si el componente tiene que responder a nuestra Pi, entonces enviará los datos usando el pin MISO (GPIO9) que significa Master In Slave Out.

UART – Receptor / Transmisor Asíncrono Universal

También conocidos como «Serial», los pines UART (transmisor GPIO14 y receptor GPIO15) proporcionan acceso a la consola/terminal para una instalación sin cabeza, lo que significa que puede conectarse a la Pi sin necesidad de un teclado o dispositivo señalador.

Normalmente, la forma más fácil de realizar una configuración sin cabeza en Raspberry Pi es simplemente controlar la Pi a través de una red o una conexión USB directa (en el caso de Pi zero).

Sin embargo, si no hay conexión de red, puedes tener un control sin cabeza en la Pi a través de un cable serie o USB conectado a una tarjeta serie desde un ordenador que ejecute una consola.

UART es un protocolo excepcionalmente fiable y garantiza el acceso a una Raspberry Pi sin necesidad de equipo adicional. Sólo tienes que recordar habilitar la consola serie en la aplicación Raspberry Pi Configuration. Dicho esto, es poco probable que quieras o necesites hacer esto, pero el soporte de UART está presente si lo necesitas en algún momento.

Tierra (gnd)

La tierra es comúnmente llamada GND, o gnd – pero todas ellas significan lo mismo. GND es donde se pueden medir todos los voltajes y donde se completa un circuito eléctrico.

Es nuestro punto cero y al conectar un componente, como un LED, a una fuente de alimentación y a tierra, el componente se convierte en parte del circuito y la corriente fluye a través del LED produciendo luz.

Al construir cualquier circuito, siempre es aconsejable hacer las conexiones a tierra antes de aplicar cualquier potencia, ya que esto evitará cualquier problema con los componentes sensibles.

Raspberry Pi tiene ocho conexiones a tierra a lo largo del GPIO y cada uno de estos pines gnd se conecta a una sola conexión a tierra.

Por lo tanto, la elección de la clavija a utilizar está exclusivamente relacionada con las preferencias personales o la comodidad a la hora de conectar los componentes.

5v

Los pines de 5v permiten el acceso directo a la fuente de alimentación de 5v desde el adaptador principal, que es menos potente que la que utiliza la propia Raspberry Pi.

Una Pi puede ser alimentada directamente desde estos pines y también puede alimentar otros dispositivos de 5v. Necesitas ser consciente de lo que estás haciendo cuando vas a usar estos pines directamente, así que ten cuidado y verifica los voltajes antes de hacer una conexión, porque ellos pasan por alto las funciones de seguridad, tales como el regulador de voltaje y el fusible que están ahí para proteger a la Pi.

Así que ten cuidado con lo que haces, porque si evitas esto último con un voltaje más alto, puedes hacer que tu Pi quede inutilizable.

3v3

La clavija de 3v está ahí para proporcionar una fuente de alimentación estable de 3,3V para los componentes y probar los LEDs. De hecho, es raro que consideres usar este pin en una construcción, pero tiene un uso especial. Al conectar un LED al GPIO, primero debes asegurarte de que el LED esté correctamente conectado y que se encienda.

Conectando la pata larga del LED, el ánodo a la clavija de 3.3v a través de una resistencia y la pata más corta, el cátodo a una de las clavijas de tierra (gnd), podemos comprobar que nuestro LED se enciende y funciona. Esto elimina un error de hardware del proyecto y nos permite empezar a construirlo con confianza.

Arrancar Raspi 4 from USB

<https://hardzone.es/tutoriales/montaje/raspberry-pi-4-ssd-pendrive-usb/>

<https://www.youtube.com/watch?v=c4fb7Y6mfAo>

Windows 10 como punto de acceso

https://support.microsoft.com/es-es/windows/usar-su-windows-pc-como-zona-de-acceso-m%C3%B3vil-c89b0fad-72d5-41e8-f7ea-406ad9036b85#WindowsVersion=Windows_10

<https://www.redeszone.net/tutoriales/redes-wifi/crear-punto-acceso-wifi-windows/>

Ruta estática windows 10

https://docs.microsoft.com/es-es/windows-server/administration/windows-commands/route_ws2008

Raspy como emisor FM

<https://circuitdigest.com/microcontroller-projects/raspberry-pi-fm-transmitter>

Jueguitos y esas cosas

<https://www.adslzone.net/reportajes/videojuegos/sistemas-operativos-emuladores-raspberry-pi/>