

Everything you ever wanted to know about the Raspberry Pi

Computer Group - Arthur Hill



Welcome

- Admin
 - Fire Alarm and Exits
 - Health & Safety
 - Trip hazards
 - Medical
 - Refreshments - £3 for the WHOLE year, please wash up after
 - Tables and Chairs – Please put away at the end of the session
- Format
 - 1 Hour presentation
 - Questions
 - One to one Helper sessions
 - Teas/Coffees and biscuits
 - Pack up at 11:30 sharp

Overview

- History of the Pi and the organisation behind it
- Major Pi Milestones
- All the Models and Hardware Architecture
- All the Chips in all the models
- Operating system versions and how to get started.
- Programming the Pi, Scratch and Python.
- Practical example

History of the Pi

- Pi story commenced in 2006 at the University of Cambridge's computer laboratory.
- Founders – Eben Upton, Rob Mullins, Jack Lang and Alan Mycroft
- Concerned that computer science candidates had no practical tinkering experience after the death of the BBC Micro in the 1980s.
- Core Mission
- To create an ultra low cost, highly accessible computing platform that encourages young people to learn programming, hardware interfacing and computer science fundamentals without fear of breaking anything expensive.

Key organisational structure

- There is a joint educational charity and commercial business structure
- The Raspberry Pi Foundation is a UK registered charity which is focussed on educational outreach,
- Then there is:
- Raspberry Pi Ltd which is the commercial subsidiary responsible for engineering, manufacturing and supply chain scaling. And,
- The Pi Hut is the primary retail outlet for Pi products and a good source of information all things Pi.
- Useful book – The Official Raspberry Pi Beginner's Guide, 6th Edition, £19.99

Major historical Milestones

- February 2012: Launch of the original Raspberry Pi Model B about £26 at the time. The initial production batch of 10,000 units sold out in minutes and crashed their servers.
- 2014: Introduction of the now standard 40-pin GPIO standard with the Model B+. This set the form factor baseline for all future models.
- 2015: Launch of the £3 Raspberry Pi Zero, creating an extreme, stripped down, miniature computer.
- 2019: Raspberry Pi 4 Model B launched with desktop class performance, dual 4K display output and USB 3.0
- 2021: Raspberry Pi design their own silicon with the RP2040 microcontroller which powers the £3 Raspberry Pi Pico.
- October 2023: Launch of Raspberry Pi 5 featuring in-house silicon I/O controller (RP1) and PCIe connectivity with over 60 million boards sold globally. £45 for the 4GB model and £60 for the 8GB model. Prices have moved on since then!

Model generations and Technical specs

Model	Release	SoC chip	CPU Specs	RAM	Key features and I/O
Pi 1 Model B	2012	Broadcom BCM2835	700 MHz single core ARM 1176JZF-S	256MB/512 MB	26 pin GPIO, Composite RCA, full HDMI, 10/100 Ethernet, SD card
Pi 2 Model B	2015	Broadcom BCM 2836	900 MHz Quad-core ARM Cortex-A7	1 GB LPDDR2	40 pin GPIO, 4x USB 2.0, 10/100 Ethernet, microSD slot
Pi 3 Model B+	2018	Broadcom BCM 2837B0	1.4 GHz Quad-core ARM Cortex-A52 (64-bit)	1 GB LPDDR2	Dual-band 2.4/5GHz WiFi, Bluetooth, GigE over USB 2.0
Pi 4 Model B	2019	Broadcom BCM2711	1.5 GHz (later 1.8 GHz) Quad-Core Cortex-A76	2 GB, 4 GB, 8 GB	True Gigabit Ethernet, 2x USB 3.0, 2x micro-HDMI (4K60), USB-C power connector

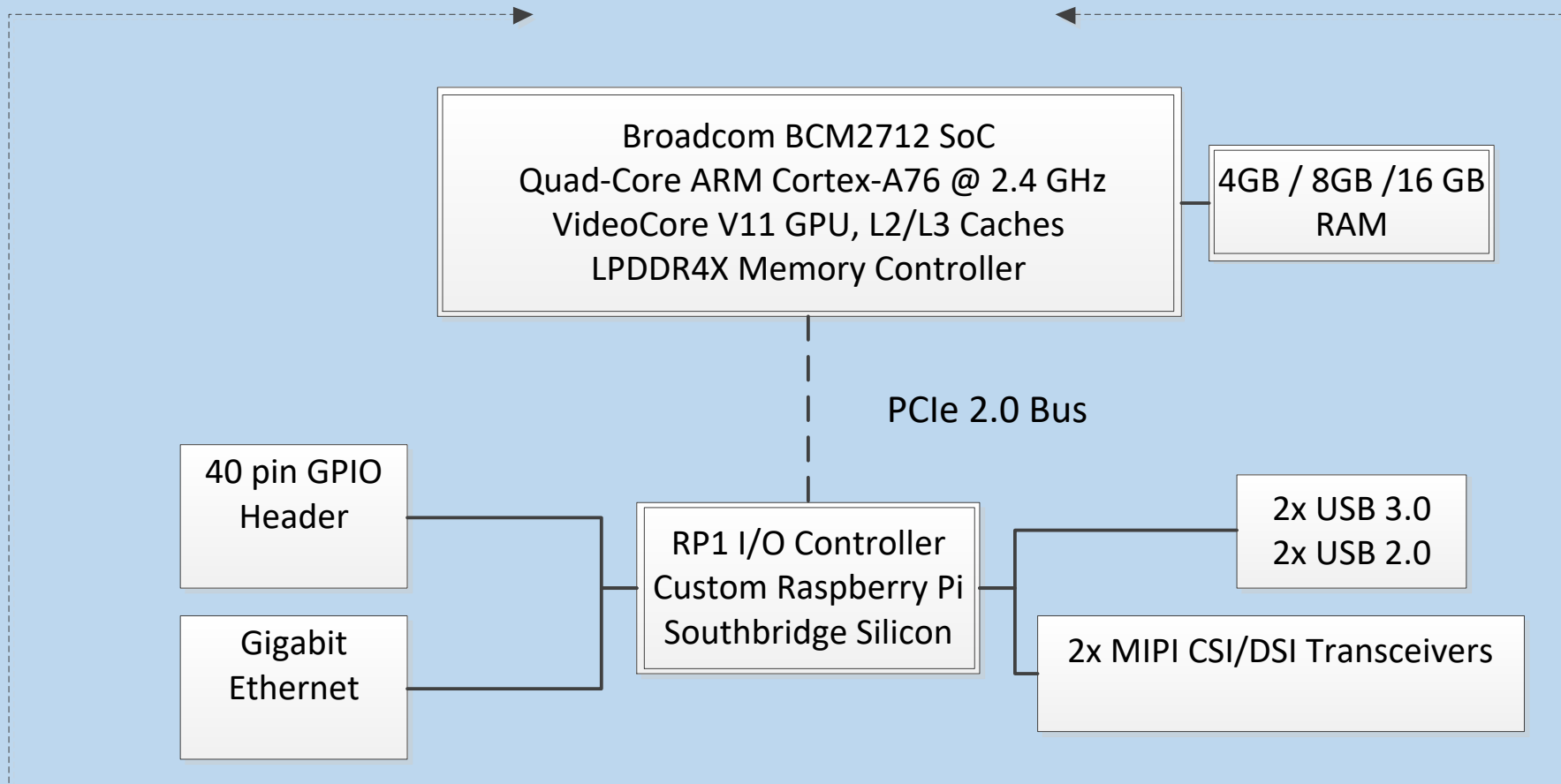
Models - continued

Model	Release	SoC chip	CPU Specs	RAM	Key features and I/O
Pi 5	2023	Broadcom BCM 2712	2.4 GHz Quad-Core ARM Cortex-A76	4 GB, 8 GB, 16 GB LPDDR4X	RP1 I/O chip, Wireless LAN, PCIe 2.0 x 1 Lane, power button, RTC, Bluetooth, Dual 4-lane MIPI
Zero 2 W	2021	Broadcom BCM2710A1	1 GHz Quad-Core ARM Cortex-A53	512 MB SIP	Ultra-compact layout, Wireless LAN, BT 4.2, micro- USB power and data
Pico / Pico W	2021	RP2040 Raspberry Pi silicon	133 MHz Dual-Core ARM Cortex-M0+	264 KB SRAM	Microcontroller, 2x PIO state machines, 26 GPIO, low power consumption
Pi 500 All-in-One	Dec 2024	Broadcom BCM 2712	2.4 GHz Quad-Core ARM Cortex-A76	8GB LPDDR4X- 4267 SDRAM	Basically it's a keyboard with a Pi 5 built into it.

Expansion boards (HATs)

- You can buy bespoke plug-in boards called HATs, Eg
- Raspberry Pi M.2 HAT £11.50 – allows connection to SSD memory
- Raspberry Pi AI HAT+ £67.20 – allows the Pi 5 to host AI locally by including **built-in Hailo neural network AI accelerator**
- Raspberry Pi AI HAT+2 £192.00 – Adds local generative AI capability
- Raspberry Pi Sense HAT V2 £28.80 – Has a whole range of sensors you can interact with.
- Raspberry Pi TV HAT £20.60 – Receive digital terrestrial TV on your Pi.

Pi System Block Diagram – Pi 5



Pi 5 System Block Diagram

Pi 5 Layout

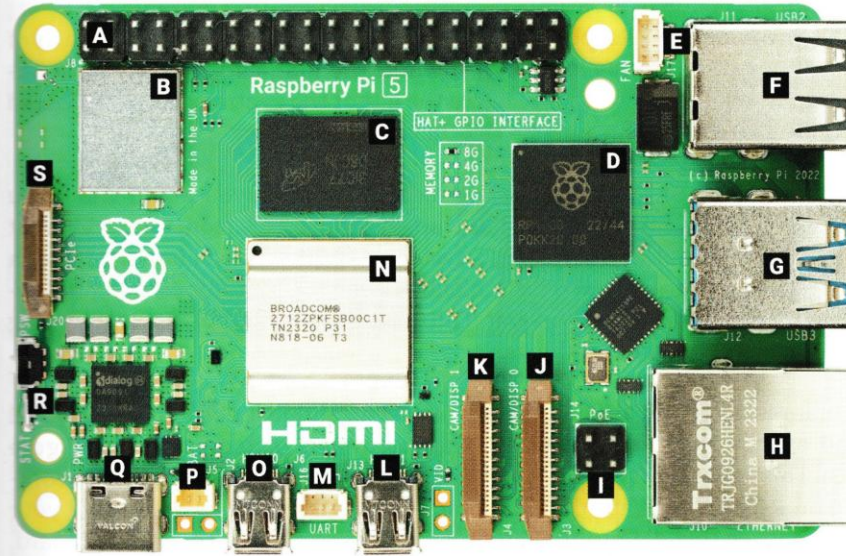
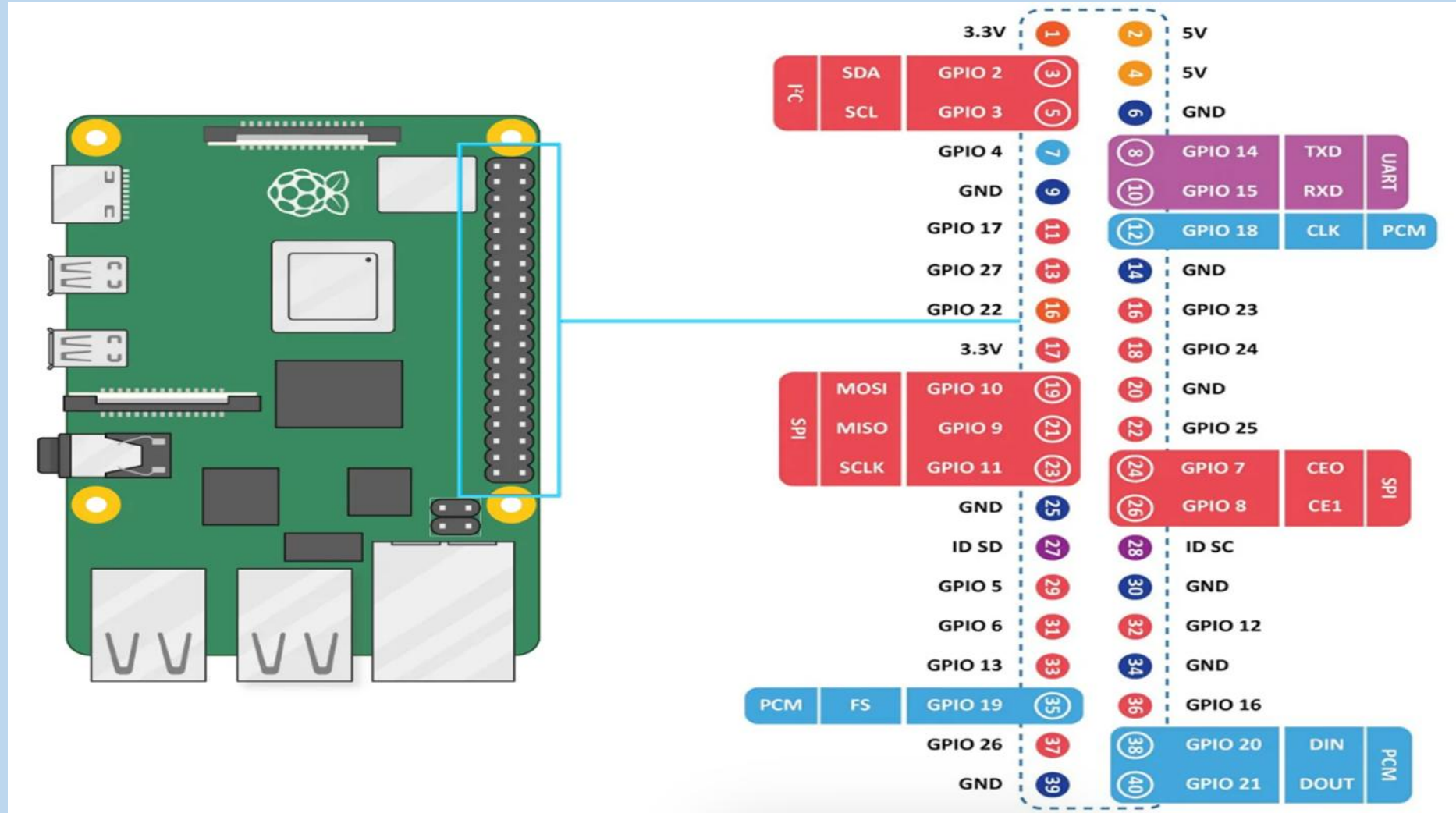


Figure 1-1 Raspberry Pi 5

- | | |
|---|---|
| A GPIO header | K CSI/DSI camera/display port 1 |
| B Wireless radio | L Micro HDMI 1 |
| C RAM | M Connector for UART serial port |
| D RP1 I/O controller chip | N System-on-chip |
| E Connector for fan | O Micro HDMI 0 |
| F USB 2.0 | P Real-time clock battery header |
| G USB 3.0 | Q USB Type-C power in |
| H Ethernet port | R Power button |
| I Power-over-Ethernet (PoE) pins | S Connector for PCI Express (PCIe) |
| J CSI/DSI camera/display port 0 | |

GPIO pinouts



Pi 5 Headers and pinouts

Raspberry Pi 5 Pinouts				
J8 GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA, I2C1)		DC Power 5v	04
05	GPIO03 (SCL, I2C1)		Ground	06
07	GPIO04		GPIO14	08
09	Ground		GPIO15	10
11	GPIO17 (SPI1 CE1)		(SPI1 CE0) GPIO18	12
13	GPIO27		Ground	14
15	GPIO22		GPIO23	16
17	3.3v DC Power		GPIO24	18
19	GPIO10 (SPI0 MOSI)		Ground	20
21	GPIO09 (SPI0 MISO)		GPIO25	22
23	GPIO11 (SPI0 SCLK)		(SPI0 CE0) GPIO08	24
25	Ground		(SPI0 CE1) GPIO07	26
27	GPIO00 (SDA0, I2C)		(SCL0, I2C) GPIO01	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19 (SPI1 MISO)		(SPI1 CE2) GPIO16	36
37	GPIO26		(SPI1 MOSI) GPIO20	38
39	Ground		(SPI1 CLK) GPIO21	40



GPIO pins explained

- The Raspberry Pi 5 packs **40 GPIO pins** that speak a simple language: 3.3V logic. That's your golden rule. Feed these pins 5V, and you'll smell the magic smoke.
- Here's what you're working with:
- **Pin Categories:**
- **Power pins:** 2x 5V (pins 2, 4) and 2x 3.3V (pins 1, 17)
- **Ground pins:** 8 total (pins 6, 9, 14, 20, 25, 30, 34, 39)
- **GPIO pins:** 26 programmable pins for your projects
- The Pi 5's GPIO pins are **multi-talented**. Pin 3 It's GPIO2, but also SDA1 for I2C communication. Pin 5 GPIO3, doubling as SCL1. This flexibility means you can run multiple protocols without running out of pins.
- **Voltage specs that matter:**
- Logic HIGH: 3.3V
- Logic LOW: 0V
- **Max current per pin: 16mA**
- **Total GPIO current: 50mA combined**

Numbering: Two Schemes Used on Raspberry Pi GPIO Pins

Understanding GPIO pin numbering is essential when working with Raspberry Pi GPIO pins. The Raspberry Pi supports two schemes—BCM vs physical pin layout—each serving different purposes, that can be used to refer to the pins:

Physical Pin Numbering (Board Numbering)

The physical pin numbering refers to the actual location of the pins on the 40-pin header of the Raspberry Pi, counting sequentially from Pin 1 (top left) to Pin 40 (bottom right). This numbering scheme is based purely on the physical layout of the pins.

- Example:

- Pin 1: 3.3V Power
- Pin 2: 5V Power
- Pin 6: Ground (GND)
- Pin 7: GPIO 4

Physical numbering is useful for wiring components when you're following a visual guide or schematic, as it reflects the physical position of the pins on the header, which is especially helpful for beginners.

BCM (Broadcom SOC Channel) Numbering

BCM numbering refers to the Broadcom chip-specific GPIO numbers assigned to each pin based on the Raspberry Pi's internal SOC (System on Chip). These GPIO numbers correspond to the actual functional channels on the Broadcom processor.

- Example:

- BCM GPIO 2: Pin 3 (used for I2C SDA).
- BCM GPIO 3: Pin 5 (used for I2C SCL).
- BCM GPIO 18: Pin 12 (used for PWM).

BCM numbering is typically used in software when writing scripts or programs in languages like Python (with libraries like ``RPi.GPIO`` or ``gpiozero``). The Raspberry Pi's internal code and libraries recognize pins by their BCM numbers rather than their physical layout — a method clearly outlined in the [official GPIO usage guide](#).

More on pins

The two numbering schemes exist to address different needs:

- Physical Pin Numbering: This is simple and intuitive when physically wiring components.
- BCM Numbering: This is more precise for programming, as it reflects the actual hardware configuration and functions tied to specific GPIO channels.

Example to Compare:

- Pin 3 (Physical): Corresponds to GPIO 2 in BCM numbering.
- Pin 5 (Physical): Corresponds to GPIO 3 in BCM numbering.

Power Pins

These pins supply power to external devices or circuits connected to the Raspberry Pi.

- 5V Pins (Pin 2 and Pin 4): Provide 5V directly from the Raspberry Pi's power supply, useful for powering components like relays or higher-power devices.
- 3.3V Pins (Pin 1 and Pin 17): Deliver a regulated 3.3V, suitable for lower-power devices such as sensors.
- Ground (GND) Pins: These pins are essential for completing electrical circuits with external components and are clearly labeled in any Raspberry Pi GPIO guide.

Standard GPIO Pins

- Digital Input/Output: These GPIO functions allow pins to be programmed as either input or output, depending on the task. They can read signals from external devices (input) or send signals to control them (output).

- Example Use: We can read the state of a button (input) or control an LED (output).

GPIO pins on the Raspberry Pi 5 include pins like GPIO 2, GPIO 3, etc., and can be referred to by their BCM numbers or physical pin numbers.

More on pins

UART Pins (Universal Asynchronous Receiver-Transmitter)

These pins are used for serial communication with other devices like microcontrollers or even other Raspberry Pi boards.

- TXD (Transmit): Pin 8 (BCM GPIO 14).
- RXD (Receive): Pin 10 (BCM GPIO 15).

Ideal for communication between the Raspberry Pi and another device using a serial protocol, such as a GPS module or a wireless communication chip, which are common examples featured in a Raspberry Pi GPIO tutorial.

I2C Pins (Inter-Integrated Circuit)

The I2C protocol on Raspberry Pi 5 GPIO is used for communication with multiple low-speed devices over two shared lines.

- SDA (Data Line): Pin 3 (BCM GPIO 2).
- SCL (Clock Line): Pin 5 (BCM GPIO 3).

Frequently used to connect the Raspberry Pi to sensors like temperature, humidity, or pressure sensors.

SPI Pins (Serial Peripheral Interface)

The Raspberry Pi GPIO pinout includes several SPI pins, which support high-speed communication with displays, sensors, and SD cards.

- MOSI (Master Out Slave In): Pin 19 (BCM GPIO 10).
- MISO (Master In Slave Out): Pin 21 (BCM GPIO 9).
- SCLK (Serial Clock): Pin 23 (BCM GPIO 11).
- CE0 (Chip Enable 0): Pin 24 (BCM GPIO 8).
- CE1 (Chip Enable 1): Pin 26 (BCM GPIO 7).

Commonly used for high-speed data communication with devices like TFT displays or high-speed sensors.

Yet more on pins

PWM Pins (Pulse Width Modulation)

Pulse Width Modulation (PWM) is a technique used to control the speed of motors or dim LEDs by varying the duty cycle of the signal — a fundamental method explained in the [Raspberry Pi documentation on PWM](#).

- PWM0: Pin 12 (BCM GPIO 18).
- PWM1: Pin 33 (BCM GPIO 13).

These pins are often used to control motor speed, LED brightness, or servos—showcasing the versatility of GPIO functions in robotics.

EEPROM Identification Pin

- ID_SD (Data): Pin 27 (BCM GPIO 0).
- ID_SC (Clock): Pin 28 (BCM GPIO 1).

Among Raspberry Pi 5 GPIO pins, the EEPROM ID pins help identify add-on boards, allowing the Raspberry Pi to automatically configure settings when an accessory board is attached.

Pinout Diagrams on Raspberry Pi 5

To get an overview of the Raspberry Pi GPIO pin layout directly from your Raspberry Pi, we can use the built-in pinout command. This is a simple command-line tool that displays a visual representation of the GPIO pins, their numbers, and functions.

Any Raspberry Pi GPIO guide or Raspberry Pi 5 pin diagram resource will recommend running the following command for onboard reference.

pinout

Pi Operating Systems

- The Pi is pretty flexible when it comes to operating systems. It can run a diverse range of Linux distributions, special purpose operating systems environments and bare-metal micro-executables.
- Raspberry Pi OS (the official OS)
This is based on Debian 64-bit, optimised for the Pi. It features the PIXEL desktop environment, Pre-installed LibreOffice, Python IDEs (Thonny) and native configuration tools (raspi-config). Current release is called Trixie.
- Ubuntu (Desktop and Server)
Canonical provides 64-bit Ubuntu builds for the Pi. Ideal for cloud native development, robotics frameworks using ROS (Robot OS).

Pi OS continued

- Specialised and Microcontroller OS

DietPi: Ultra-lightweight Debian build for maximum efficiency

Home Assistant OS: Dedicated IoT smart home automation host

MicroPython / FreeRTOS: Bare-metal execution environments for Pi Pico.

- Media and Retro Gaming OS

LibreELEC: Minimalist OS running Kodi for 4K media streaming

Retro Pie / Batocera: Retro gaming emulation suites using EmulationStation and RetroArch interfaces.

Programming the Pi using Scratch 3

- This is a block based programming language used to teach kids how to program. You program by dragging and dropping blocks together.

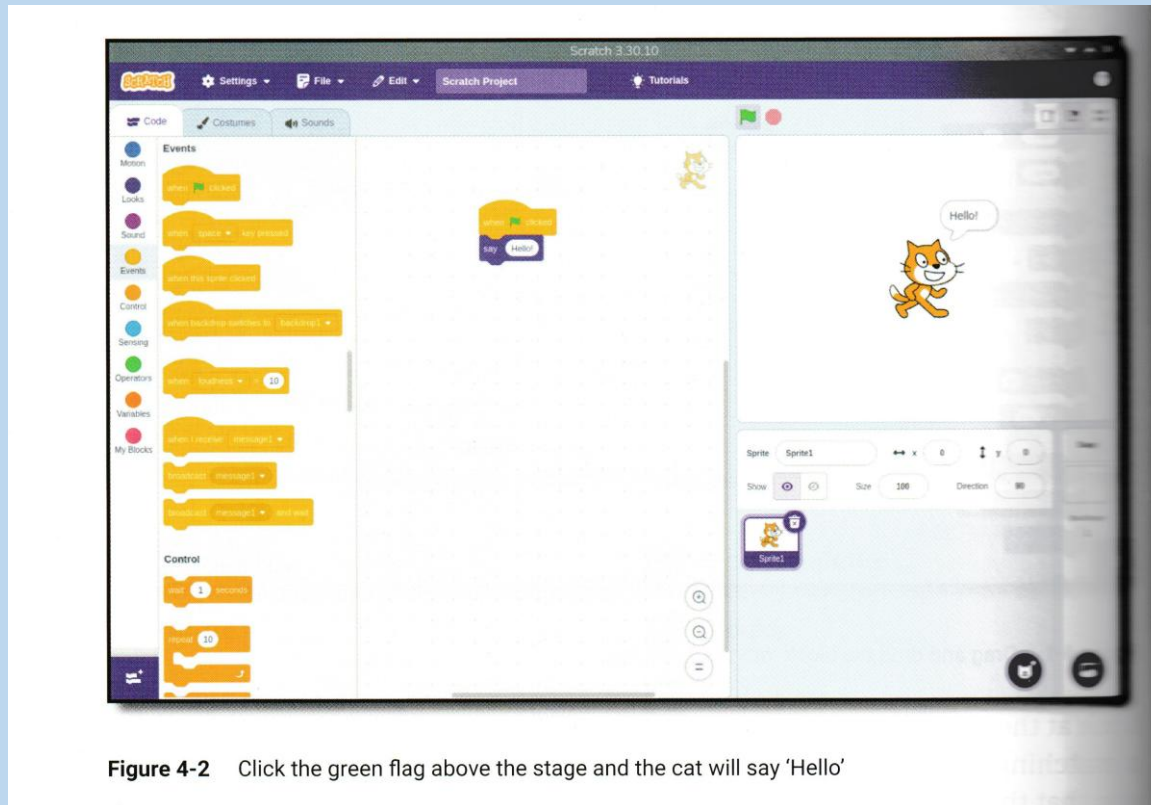


Figure 4-2 Click the green flag above the stage and the cat will say 'Hello'

Programming the Pi using Python

- This is a fully featured powerful programming language which can be used to create industry strength applications.
 - `#!/usr/bin/env python3`
 - `# Raspberry Pi Environmental & Thermal Monitor Demo`
 - `import time`
 - `from gpiozero import LED, Button, CPUTemperature`
 - `alert_led = LED(17)`
 - `user_button = Button(27, pull_up=True)`
 - `cpu = CPUTemperature()`
 - `TEMP_THRESHOLD = 60.0 # °C`
 - `def manual_override():`

Python part2

- `print("\n[INTERRUPT] Manual override button pressed! Flashing LED...")`
- `alert_led.blink(on_time=0.2, off_time=0.2, n=5)`
- `user_button.when_pressed = manual_override`
- `try:`
- `while True:`
- `current_temp = cpu.temperature`
- `print(f"[{time.strftime('%H:%M:%S')}] CPU Temp: {current_temp:.1f}°C")`
- `if current_temp > TEMP_THRESHOLD:`
- `alert_led.on()`
- `else:`
- `alert_led.off()`
- `time.sleep(2.0)`
- `except KeyboardInterrupt:`
- `alert_led.off()`

Enterprise Applications

- The Pi has evolved into a major force in commercial and industrial engineering:
- Compute Modules: compact form factor without the usual ports, designed to be soldered or socketed into custom carrier boards for industrial automation, edge AI gateways and digital signage.
- Industrial IoT Deployment: Its low power consumption makes it ideal for remote environmental telemetry, smart agricultural controllers and factory automation systems.

Conclusion

- Originally developed as a low cost education and learning aid, the Pi has developed into a very flexible computing platform.
- It is well supported with modular software support, vast community libraries and real-time GPIO hardware control.
- The Raspberry Pi is an unmatched platform for rapid prototyping to full scale deployment.
- You could realistically use a Pi 5 as a very serious desktop replacement.