TAFITI PROJECT

MEMBERS	
NAME	**REGISTRATION NUMBER**
NASIR JACOB	ENE212-0223/2019
HANNI PENDO	ENE211-0001/2021
PETER NJUGUNA	ENE211-0006/2021
KEVIN MUTISYA	ENE211-0020/2021
MALCOLM KIVUTI	ENE211-0022/2021
ANNE WHITNEY	ENE211-0029/2021
LILIAN CHEPNG'ETICH	ENE211-0032/2021
GRACE MUGO	ENE211-0200/2021
JERRY KICHE	ENE211-0261/2021

TITLE: REPORT ON COMPARATIVE ANALYSIS OF MICROCONTROLLERS FOR INTEGRATION IN TAFITI PROJECT LEO SATELLITE

Table of Contents

LIST OF ABBREVIATIONS

CORE COMPONENTS & ARCHITECTURES

SoC – System on Chip

A single integrated circuit that includes a CPU, memory, and peripherals. Widely used in microcontrollers and embedded systems for satellites and IoT devices.

CPU – Central Processing Unit

The main processing unit that executes instructions. Present in computers, microcontrollers (e.g., ARM Cortex), and on-board satellite computers.

RAM – Random Access Memory

Temporary memory used for fast read/write operations. Used in all computing devices including satellite computers and microcontrollers.

SRAM – Static RAM

A type of RAM that retains data as long as power is supplied. Faster and more expensive than DRAM; often used in microcontrollers.

PSRAM – Pseudo Static RAM

Combines features of DRAM and SRAM, used in microcontrollers for expanded memory with easier interfacing.

CONNECTIVITY & NETWORKING

WiFi – Wireless Fidelity

Wireless communication protocol (IEEE 802.11), often used in IoT devices, and sometimes in satellite ground-station connectivity.

IoT – Internet of Things

Network of connected embedded devices, often based on microcontrollers, sensors, and wireless communications (BLE, WiFi, etc.).

BLE – Bluetooth Low Energy

A power-efficient wireless protocol for short-range communication, used in IoT sensors and devices.

GATT – Generic Attribute Profile

A BLE protocol defining how two Bluetooth Low Energy devices exchange data.

CAN bus – Controller Area Network Bus

Robust communication protocol used in automotive and satellite systems for reliable node-to-node communication.

Ethernet MAC – Media Access Control

Part of the Ethernet controller responsible for data link layer; used in microcontrollers with Ethernet connectivity.

MQTT – Message Queuing Telemetry Transport

Lightweight messaging protocol often used in IoT and satellite telemetry systems.

HTTP – Hypertext Transfer Protocol

Standard web protocol, used in embedded web servers (e.g., ESP32-based devices) and for cloud-based IoT communication.

OTA – Over-the-Air (Updates)

Remote firmware or software updates sent wirelessly to devices, used in satellites, microcontrollers, and IoT products.

TLS/SSL – Transport Layer Security / Secure Sockets Layer

Cryptographic protocols for secure data transmission, used in IoT communications and web-based satellite interfaces.

WPA – Wi-Fi Protected Access

Security protocol used to secure Wi-Fi communications, relevant to IoT and wireless microcontroller applications.

INTERFACES & PERIPHERALS

GPIO – General Purpose Input/Output

Digital pins on a microcontroller that can be programmed for input or output tasks (e.g., sensor interfacing or LED control).

PWM – Pulse Width Modulation

Technique for simulating analog voltage using digital signals. Used in motor control, LED dimming, and satellite actuators.

ADC – Analog to Digital Converter

Converts analog signals (e.g., from sensors) to digital values for microcontroller processing.

DAC – Digital to Analog Converter

Converts digital data to analog signals, used in sound generation, control signals, or analog interfacing.

UART – Universal Asynchronous Receiver/Transmitter

Serial communication protocol used to communicate with sensors, GPS, GSM, and other modules.

USART – Universal Synchronous/Asynchronous Receiver/Transmitter

Enhanced UART that supports both synchronous and asynchronous communication.

SPI – Serial Peripheral Interface

High-speed synchronous serial communication used between microcontrollers and peripherals (e.g., flash memory, displays).

PHY – Physical Layer (Transceiver)

Hardware interface for managing physical signal transmission over mediums like Ethernet or USB.

POWER MANAGEMENT

ULP – Ultra Low Power

Refers to microcontroller modes or architectures optimized for very low power consumption, important for IoT and space systems.

INDUSTRY & MANUFACTURING

OEMs – Original Equipment Manufacturers

Companies that produce hardware or components that may be rebranded or integrated into another company's system (e.g., satellite systems, IoT devices).

IDE- Integrated Development environment

An IDE is a software suite that combines the tools needed for writing, testing, and debugging code in a single user interface

SEUs: (Single Event Upsets)

Are a type of radiation-induced error that occurs when a high-energy particle (like a proton, neutron, or heavy ion) strikes a sensitive part of a microelectronic circuit and flips a bit in memory or a register.

:

I

INTRODUCTION

The Tafiti Project aims to design and simulate a functional Low Earth Orbit (LEO) satellite, focusing on practical implementation using low-cost, commercially available components. A critical aspect of this initiative involves the selection of suitable microcontrollers to power various satellite subsystems, from telemetry and data handling to power management and communication.

This report presents a comparative analysis of several affordable microcontrollers—including the ESP8266, ESP32, ATtiny85, ATmega328P, Arduino Mega, Leonardo, Uno, and Nano—evaluating their features, limitations, and applicability within the scope of the Tafiti Project. The objective is to identify microcontrollers that offer the best balance of cost, performance, power efficiency, and peripheral support suitable for educational bench models or proof-of-concept nanosatellite platforms.

By understanding the strengths and constraints of each microcontroller, this analysis provides a foundation for informed subsystem integration, guiding future development of satellite control systems that are robust, scalable, and aligned with the technical goals of the Tafiti Project.

COMPARATIVE ANALYSIS OF MICROCONTROLLERS

1. THE ESP8266

The ESP8266 was released by their manufacturers (Espressif Systems – China) in 2014. The board gained popularity in late 2014 to 2015 when inexpensive boards became available. It was mainly meant to introduce the concept of addition of WiFi to a microcontroller but at the same time limiting the price incurred for such. It therefore offered a cheap WiFi System on Chip (Soc) for IoT OEMs with its own CPU that needed no host microcontroller to run user code. After its release, it gained popularity due to its low cost, integrated WiFi, reasonable processing power and simplicity.

1.1 PIN LAYOUT

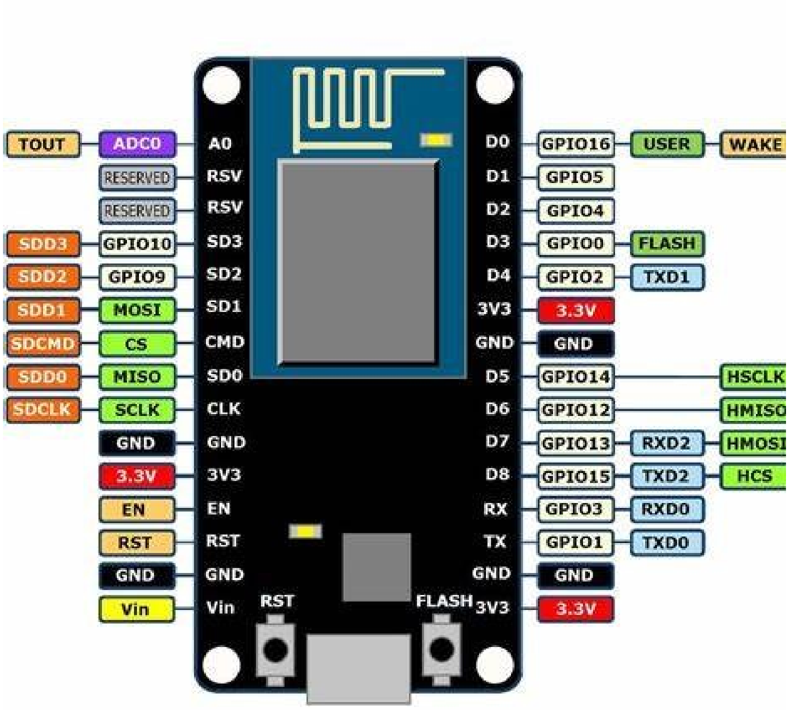


Figure 1.1 ESP8266 pin layout

1.2 FEATURES

The core chip is named **ESP8266EX**. It has the following capabilities:

a) Processor and Core

It has a 32bit Tensilica L106 microcontroller with an 80MHz clock speed that can be overclocked to 160MHz. It can achieve up to 100 DMIPS. It has a single core and an instruction cache to compensate for the speeds of the flash. It also supports RTOS via a software development kit.

b) Memory

It has a total memory of 192KB SRAM with only 160 KB available to the user. It has no internal flash memory and often requires a Serial Peripheral Interface flash chip. In the event the module comes with flash, it has a chip of 1-4MB of flash memory.

c) Wireless Connectivity

It has WiFi capabilities (IEEE 802.11) with 2.4 GHz band. It supports WPA/WPA2 security.

d) I/O and Peripherals

It has up to 17 GPIO pins depending on the variant of the chip. The GPIOs can serve multiple functions.

- It supports up to 8 PWM channels.
- It has one ADC channel.
- 2 UART interfaces.
- 2 SPI interfaces with one being a high speed master/slave.
- It can support I2C and rarely 12S.
- Has multiple hardware and software timers with a basic real time clock.

e) Power Consumption and Sleep Modes

It has an operating voltage of between 3-3.6V taking up currents of between 10 µA to 170mA depending on the power mode it is operating on. These include but is not limited to:

- Active Mode - 70 - 170 mA during WiFi transmission
- Modem Sleep - 15-20mA
- Light Sleep - 0.4mA
- Deep Sleep - 10 µA

It can be brought back online through use of timing functions or external GPIO options that are limited.

f) Security Features

It has multiple security layers like the TLS/SSL. It also has WPA/WPA2 encryption but it is vulnerable to software based attacks if not better secured by the user.

g) Development and Programming

It can support multiple programming languages like C, C++, MicroPython etc. It can also be used in development environments like the Arduino IDE which happens to be the most popular. It can also support MQTT, HTTP and Over the Air (OTA) updates.

The ESP8266 has multiple modules:

1. ESP-01 - is very small and has only 2 GPIOs.
2. ESP-12E/F - is the most popular. Has 11-17 GPIOs.
3. NodeMCU - has voltage regulation and ESP-12
4. Wemos D1 Mini - is a compact board with USB and full GPIO access.

1.3 ADVANTAGES

1. It is cheap.
2. It is easy to use and configure.
3. It is compact.
4. Has WiFi capabilities.
5. Has a low power usage.

1.4 DISADVANTAGES

1. Has less GPIO pins
2. Has no bluetooth capability
3. Has only one ADC channel.
4. Single core; it therefore is less powerful and less ideal for multitasking.
5. Has a limited memory space.
6. Has no hardware crypto/ security support.

2. THE ESP32

This microcontroller was developed due to the increasing need for industrial and commercial IoT as compared to the small scale use of the ESP8266. It was developed from around 2015 to address the shortcomings of the mother microcontroller and act as its powerful successor. The first dev. kit was released into the market towards the end of 2016. The chip gained rapid popularity in professional and hobbyist markets. This chip could:

- Handle more complex applications, e.g. audio streaming and real time control.
- Add secure communication for production IoT devices.
- Support Bluetooth that could be used for local wireless communication
- Improve multitasking and peripheral support.

This among many other functions enabled edge computing and application of the microcontroller to security sensitive applications.

2.1 ESP32 DEVELOPMENT BOARDS

A development board is a complete ready to use board built for prototyping and development. It normally includes:

1. An ESP32 module
2. A USB interface i.e. a USB to serial chip
3. Voltage Regulators that power the board via USB or an external source
4. Buttons for booting and resetting.
5. Pin headers that allow for easy connection to breadboards or peripherals.

Examples include:

ESP32 DevKit V1

It is a general purpose ESP32 development board. It is the most popular kit and best for beginners. It uses the ESP32-WROOM-32 module, has USB to Serial chip and full GPIO access. It is simple, well-supported, breadboard-friendly, is great for learning and prototyping. It however has no built in display or cameras.

Figure 2 ESP32 DevKit V1

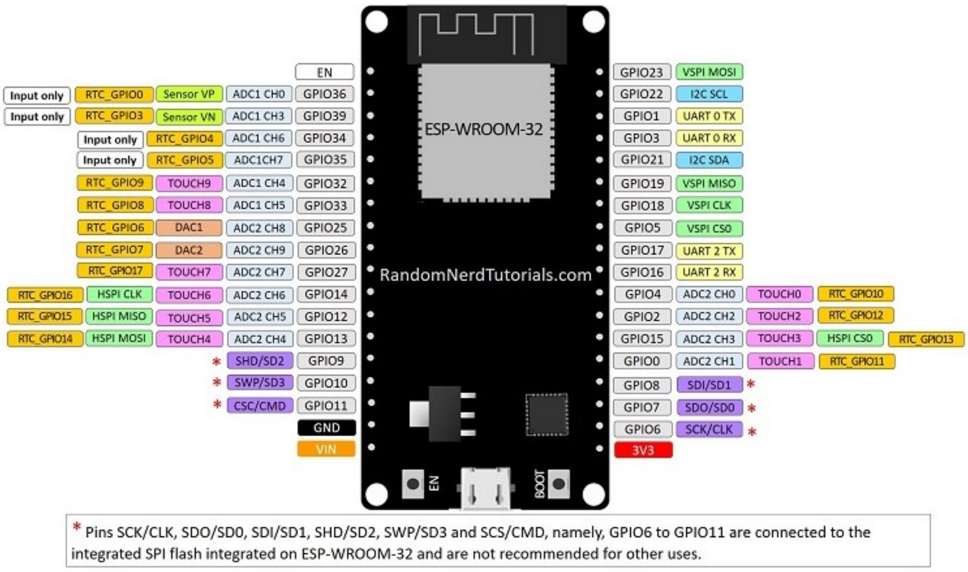


Figure 3 Pin out diagram for ESP32 DEVKIT V1 DOIT board

ESP32 OLED Kit

It includes a built in 0.96” OLED display making it a good addition for IoT applications. It has provision for connection an external antennae and a battery connector for creating battery operated projects. It uses the ESP32-WROOM-32 or WROVER module. The built in display saves GPIOs and space and is good for live feedback. It however has fewer accessible GPIOs

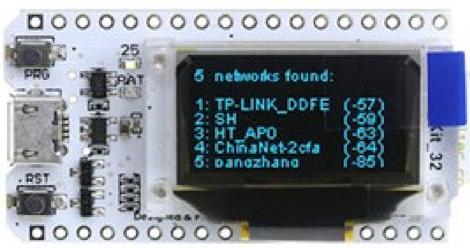


Figure 4 ESP32 OLED Kit

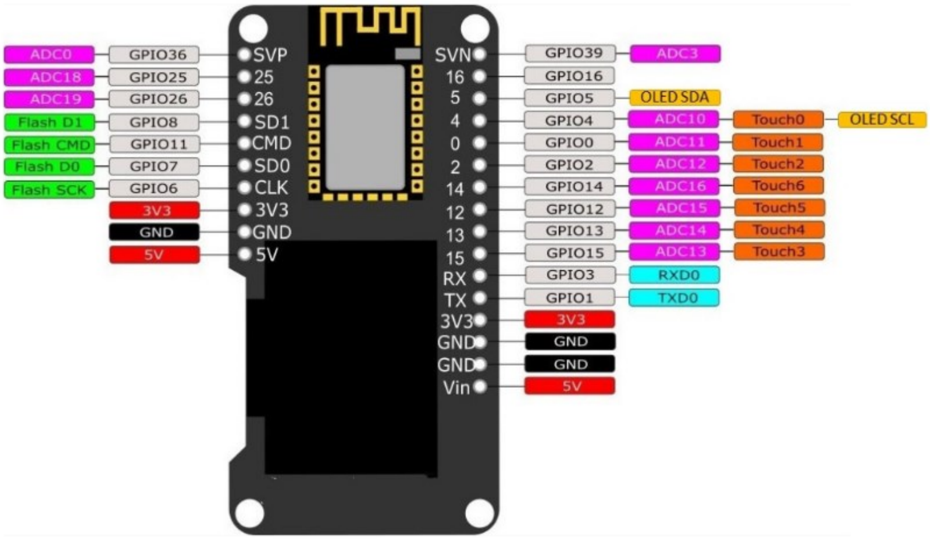


Figure 5 Pin out diagram for ESP32 OLED Kit

ESP32-CAM

It has a built in 2MP camera and a microSD card slot. It is suitable for projects that require a camera with advanced functions like image tracking and recognition. I t has no USB port and therefore requires external USB to Serial adapter. It uses ESP32-S a variant of the ESP32. It however has less GPOs due to the camera pins.

Figure 6 ESP32 CAM

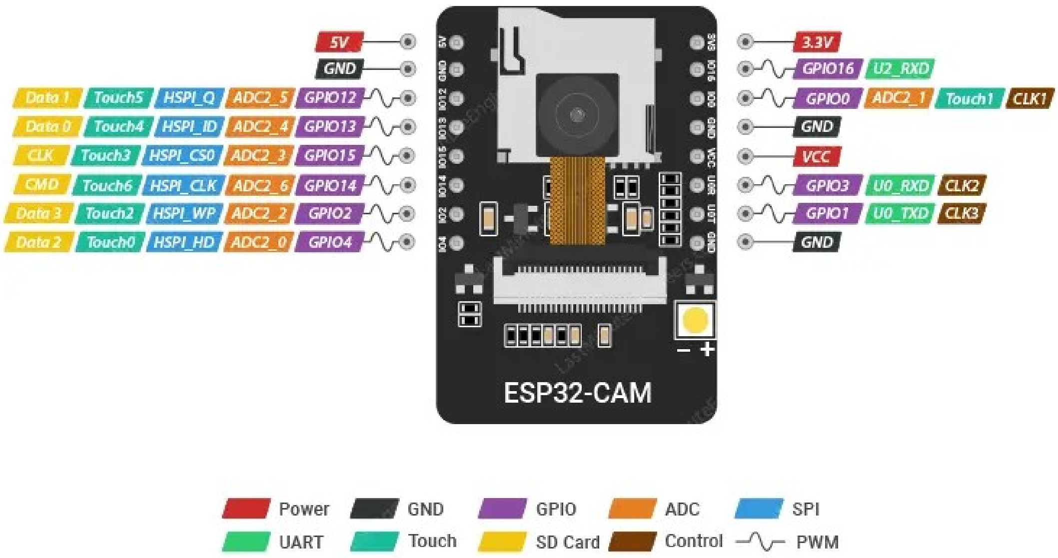


Figure 7 Pin out diagram for ESP32 CAM

ESP32 SIM800L TTGO T-Call

It allows one to connect to the internet via a SIM card data plan without the need of Wi-Fi. It has a GSM module (SIM800L) to allow cellular connectivity. It supports SMS, GPRS and voice. It is ideal for remote IoT. It can run fully off grid with battery. It however consumes more power and requires good battery management.

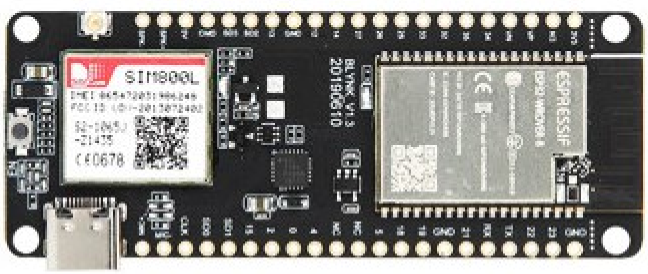


Figure 8 ESP32 SIM800L TTGO T-Call

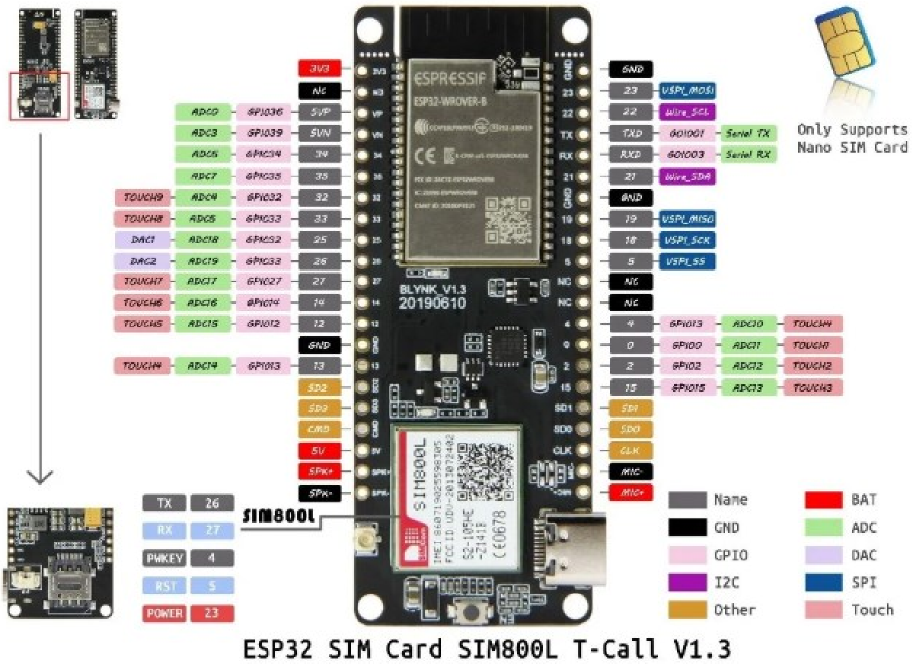


Figure 9 : Pin out Diagram for ESP32 SIM800L TTGO T-Call

M5Stack

This is a stackable modular product development toolkit based on ESP32.it includes a main controller and additional modules that can be stacked to expand the functionality of the project. It is basically a collection of modular stackable ESP32-based mini-computers and has multiple versions. It has a case, screen and battery support. It is hard to customize it at hardware level.

Figure 10 M5Stack

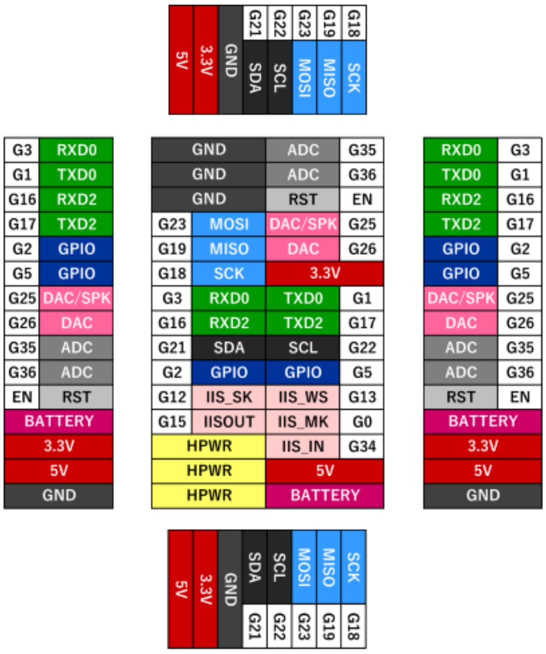


Figure 11 Pin out Diagram for M5Stack

2.2 ESP32 DEVELOPMENT BOARDS PINOUT DIAGRAM

Due to its popularity, availability and wide range of applications, the discussions below will specifically be tailored to the ESP32 DevKit V1. However, as has been highlighted earlier on, the functionalities of these pins doesn't change with the development board.

The ESP32 comes with 48 pins but only 25 are broken out to pin headers. Not every development board exposes all the pins that shall be discussed herein. It is also worth noting that the pin location and accessible can change depending on the manufacturer and the development board. The pins, however, work exactly the same regardless of the development board in question. The ESP32 has a pin multiplexing feature. This enables multiple peripherals to share a single GPIO pin. Generally, there are 40 identifiers for GPIO pins including the reserved and internal pins that may not be accessible for general use. The exact number of usable GPIOs is 34 (GPIO0- GPIO39). Of these 34, 6 are strictly for input (GPIO34- GPIO39). The remaining 6 that are not accounted of from the 40 (GPIO6-GPIO11) are tied to internal flash and are not ideal for normal GPIO usage.

In addition to those pins, there are

- Ground pins (GND).
- Power pins - 3V3 and VIN
- Enable pin (EN) - used to enable the ESP32. When HIGH, chip is enabled and when LOW chip operates at low power. The EN pin is also connected to a pushbutton switch that can pull the pin LOW and trigger a reset.

Below is table that shows what pins are best to use as inputs, outputs and which need caution while in use.

Key:

Table 1: Key for Table 2 Color Codes

Color	Meaning
Green	Pins are OK to use
Yellow	Pins are OK to use but attention needs to be paid due to their unexpected behavior especially at boot.
Red	Pins not recommended to be used as inputs or outputs.

Table 2: Table to guide on the use of different GPIO pins

GPIO	Input	Output	Notes
0	pulled up	OK	outputs PWM signal at boot, must be LOW to enter flashing mode and during programming. Must be HIGH during boot
1	TX pin	OK	debug output at boot. Used for flashing and debugging
2	OK	OK	connected to on-board LED, must be left floating or LOW to enter flashing mode and also low during boot. Is also connected to the on-board LED
3	OK	RX pin	HIGH at boot. Used for flashing and debugging
4	OK	OK	
5	OK	OK	outputs PWM signal at boot, strapping pin
6	x	x	connected to the integrated SPI flash
7	x	x	connected to the integrated SPI flash

8	x	x	connected to the integrated SPI flash
9	x	x	connected to the integrated SPI flash
10	x	x	connected to the integrated SPI flash
11	x	x	connected to the integrated SPI flash
12	OK	OK	boot fails if pulled high, strapping pin, must be LOW during boot
13	OK	OK	
14	OK	OK	outputs PWM signal at boot
15	OK	OK	outputs PWM signal at boot, strapping pin, must be HIGH during boot
16	OK	OK	
17	OK	OK	
18	OK	OK	
19	OK	OK	
21	OK	OK	
22	OK	OK	
23	OK	OK	
25	OK	OK	
26	OK	OK	
27	OK	OK	

32	OK	OK	
33	OK	OK	
34	OK		input only
35	OK		input only
36	OK		input only
39	OK		input only

2.3 FEATURES OF THE ESP32.

a. ***Processor and Architecture***

It is a 32 bit Tensilica Xtensa LX6 microcontroller with a dual core. It has a clock speed of up to 240MHz. It has the ability to allow the different cores to run at independent speeds or even be powered down. It has an Ultra-Low Power (ULP) 8 bit Coprocessor to allow sensor monitoring during deep sleep. It can execute up to 600 DMIPS for both cores.

b. ***Internal Memory***

It has 448KB of ROM for booting and core functions. It has an internal SRAM of 520 KB, shared between data RAM, instruction RAM and Cache.

It has an RTC memory that is a small and special low power memory region. It belongs to the RTC domain, a subsystem that remains powered during deep sleep. It can retain data while the main CPU is powered off. It used for low power tasks like waking up from sleep, preserving state or running ultra-low power co-processors.

It has 8 KB of SRAM in RTC, which is called RTC fast memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the light-sleep mode.

It has 8 KB of SRAM in RTC, which is called RTC slow memory and can be accessed by the co-processor during the deep-sleep mode.

c. ***External Flash and SRAM***

ESP32 can access the external QSPI flash and SRAM through high-speed caches.

- The external flash can be mapped into CPU instruction memory space and read-only memory space simultaneously.

- ✓ When external flash is mapped into CPU instruction memory space, up to 11 MB + 248 KB can be mapped at a time. Note that if more than 3 MB + 248 KB are mapped, cache performance will be reduced due to speculative reads by the CPU.
- ✓ When external flash is mapped into read-only data memory space, up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads are supported.

- External SRAM can be mapped into CPU data memory space. Up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads and writes are supported.

ESP32-WROOM-32 integrates a 4 MB SPI flash, which is connected to GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 and GPIO11. These six pins cannot be used as regular GPIOs.

Some modules like the ESP32-WROVER come with a PSRAM that can be quite useful for graphical applications that take up a lot of storage.

d. ***Wireless Connectivity***

It has Wi-Fi capabilities (IEEE 802.11) with 2.4 GHz band. It has a Wi-Fi transceiver allowing it to connect to a Wi-Fi network to access the internet (Station Mode). It can also create its own Wi-Fi wireless network (Soft Access Point mode) to which other devices can connect.

It supports classic Bluetooth (BT) and Bluetooth Low Energy (BLE 4.2/ 5.0). The BLE supports multiple advertising, scanning and GATT services. Both the Wi-Fi and BLE can be used simultaneously. The Bluetooth is mostly applicable for remote controlling purposes. The ESP32 supports

e. ***Peripherals and I/O***

- ➢ ***GPIOs***- it has up to 25 GPIO pins that can be assigned different functions by programming the appropriate registers. This is because GPIOs 0, 37 and 38 are not available on most modules (only bare chips) due to hardware constraints. There are of several kinds: digital only, analog-enabled, capacitive touch enabled etc. Analog-enabled GPIOs and Capacitive touch enabled GPIOs can be configured as digital GPIOs. Most of the digital GPIOs can be configured with internal pull-up or pull-down resistors or set to high impedance.

  - ***Input only GPIOs-*** Pins GPIO34, GPIO35, GPIO36 and GPIO39 cannot be configured as outputs. They can be used as digital or analog inputs, or for other purposes. They also lack internal pull-up and pull-down resistors, unlike the other GPIO pins.

- ***ADC channels*** - is available on 18 channels/pins. There are 2 ADC units (physical ADC core in the chip), ADC1 and ADC2. ADC1 handles 8 channels (GPIOs 32-39) while

ADC2 handles 10 channels (GPIOs 0, 2, 4, 12-15, 25-27). However, due to hardware constraints, not all of them are usable leaving room for only 15 ADC channels.

Each unit can read from multiple GPIOs in general. However, the unit can only read from one channel at a time. ADC2 shares internal resources with the Wi-Fi radio and is therefore not suited for simultaneous use with Wi-Fi.

The ADC input channels have a 12 bit resolution meaning it can detect 4096 discrete analog levels. Analog readings 0-4095 correspond to voltages 0 - 3.3V. This results in a resolution of 0.8mV per unit (3.3/4096). The ADC pins do not have a linear behavior and hence some digital values that are closer to each other will be hardly distinguishable. The ESP32 is designed to measure voltages while in sleep mode.

- ***DAC***- it has 2 8 bit channels that are used to convert digital signals to true analog voltage signal outputs. It can be used as a 'digital potentiometer' to control analog devices. These DACs have an 8-bit resolution, which means that values ranging from 0 to 256 will be converted to an analog voltage ranging from 0 to 3.3V.

The DACs 8 bit resolution may be insufficient for use in audio applications in which case an external DAC with a higher resolution is preferred.

- ***Touch Sensors***- it has 10 internal capacitive touch sensors that can be used for functions such as wakeup, UI or low power touch activation. They use capacitive touch technology. They can sense variations in anything that holds an electrical charge like the human skin. These pins can be integrated into capacitive touch and replace mechanical buttons. These internal touch sensors are connected to the following GPIOs (GPIO 0, 2, 4, 12-15, 27, 32, 33). Due to hardware constraints, only 9 are usable.
- ***SPI* -**SPI pins support high-speed communication with peripherals, making them indispensable in data-intensive applications.

The SPI uses 4 main signals:

- ✓ **MOSI-** Master Out Slave In
- ✓ **MISO-** Master In Slave Out
- ✓ **SCLK**- Serial Clock
- ✓ **CS-** Chip Select (active low)

The ESP32 has four SPI peripheral devices, called SPI0, SPI1, HSPI and VSPI. By default, the pin mapping for SPI is:

Table 3: Pin Mapping for SPI

SPI	MOSI	MISO	CLK	CS
VSPI	GPIO 23	GPIO 19	GPIO 23	GPIO 5
HSPI	GPIO 13	GPIO 12	GPIO 14	GPIO 15

GPIO6 to GPIO9, GPIO10, and GPIO11 are exposed in some ESP32 development boards. However, these pins are connected to the integrated SPI flash on the ESP-WROOM-32 chip and are not recommended for other uses.

- ***Inter-Integrated Circuit (12C)*** - has 2 I2C bus interfaces but no dedicated I2C pins. Instead, it allows for flexible pin assignment, meaning any GPIO pin can be configured as I2C SDA (data line) and SCL (clock line).

However, GPIO21 (SDA) and GPIO22 (SCL) are commonly used as the default I2C pins especially when using the Arduino IDE.

- ***UART interfaces***- has 3 hardware supported UARTs that support asynchronous communication at up to 5Mbps. The ESP32 supports up to three UART interfaces: UART0, UART1, and UART2, depending on the ESP32 board model you're using.

  - ✓ UART0 is usually reserved for communication with the serial monitor during upload and debugging. However, you can also use it for communication with other devices after uploading the code if the Serial Monitor is not needed.

UART0 pins are connected to the USB-to-Serial converter and are used for flashing and debugging. Therefore, the UART0 pins are not recommended for use.

  - ✓ UART1 and UART2: available to communicate with external devices.

Like I2C and SPI, these UART pins can be mapped to any GPIO pin on the ESP32. However, they have a default pin assignment on most board models.

  - ✓ UART2 is a safe option for connecting to UART-devices such as GPS, fingerprint sensor, distance sensor, and so on.

For most ESP32 boards the UART pin assignment is as follows:

Table 4: UART pin assignment

UART Port	TX	RX	Remarks
UART0	GPIO 1	GPIO 3	Used for Serial Monitor and uploading code; Can be assigned to other GPIOs;
UART1	GPIO 10	GPIO 9	Must be assigned to other GPIOs
UART2	GPIO 17	GPIO 16	Can be assigned to other GPIOs

About UART1 (GPIO 9 and GPIO10) – these GPIOs are connected to the ESP32 SPI flash memory, so you can't use them like that. To use UART1 to communicate with other devices, you must define different pins.

- ***Integrated Inter-chip Sound (12S) -*** on 2 pins and is used as an audio interface for microphones and DACs.
- ***PWM***- it can be assigned to any of the 25 GPIOs but only 16 can be active simultaneously. This is because the ESP32 has 16 channels capable of outputting PWM. Each channel is controlled by the LED PWM controller (LEDC). The PWM controller consists of PWM timers, the PWM operator and a dedicated capture sub-module. Each timer provides timing in synchronous or independent form, and each PWM operator generates a waveform for one PWM channel. The dedicated capture sub-module can accurately capture events with external timing.

To set a PWM signal, you need to define these parameters in the code:

- Signal's frequency;
- Duty cycle;
- PWM channel;
- GPIO where you want to output the signal.

- ***Strapping Pins –*** the ESP32 has 5 strapping pins:
    - ✓ GPIO 0 (must be LOW to enter boot mode)
    - ✓ GPIO 2 (must be floating or LOW during boot)

- ✓ GPIO 4
- ✓ GPIO 5 (must be HIGH during boot)
- ✓ GPIO 12 (must be LOW during boot)
- ✓ GPIO 15 (must be HIGH during boot)

These pins are used to put the ESP32 into BOOT mode (to run the program stored in the flash memory) or FLASH mode (to upload the program to the flash memory). Depending on the state of these pins, the ESP32 will enter BOOT mode or FLASH mode at power on.

On most development boards with built-in USB/Serial, you don't need to worry about the state of these pins, as the board puts them in the correct state for flashing or boot mode.

However, if peripherals are connected to these pins, you may encounter issues when attempting to upload new code or flash the ESP32 with new firmware, as these peripherals prevent the ESP32 from entering the correct mode.

The strapping pins function normally after reset release, but they should still be used with caution.

- ***CAN Bus***- it is supported on some modules. It is designed for robust and flexible performance in harsh environments and is useful for industrial applications.
- ***Ethernet MAC***- has been integrated but requires an external PHY chip to enable its use. This physical layer chip will enable physical layer functions and is suitable for very short distance transmission.
  - ***Hall Effect Sensor***- the ESP32 has a built-in Hall Effect sensor located beneath the metal lid of the ESP32-WROOM-32 module itself. It is used to measure magnetic fields without need to connect external modules. Being integrated into the ESP32 means that you can easily connect the sensor readings with Wi-Fi or Bluetooth functionalities, making remote monitoring and control easier.

While the onboard Hall sensor might not replace dedicated external sensors for precise applications due to its positioning and sensitivity, it still offers a range of potential uses. This includes basic magnetic field detection, triggering specific functions when a magnet is nearby, or even building simple educational projects to understand the Hall Effect.

Because the sensor is located beneath the metal lid, it is less sensitive to weak magnetic fields than standalone Hall sensors, so magnets of significant strength are usually required to obtain noticeable readings.

NB: *The ADC and DAC features are assigned to specific static pins. One can however decide which pins are UART, 12C, SPI or PWM through assigning them in the code. This is made possible by the multiplexing ability of the chip. Despite the fact that one can define pin properties on the software, one has to be careful because there are pins assigned as default.*

f. <u>Power Management and Sleep Modes</u>

The module can safely and efficiently operate within a voltage range of 2.2 -3.6V. The board therefore has a voltage regulator to keep the voltage stable at 3.3.V and can reliably provide up to 600mA of current. The output of the 3.3V regulator is broken out to the header pin labelled 3V3 which can then be used to power external circuitry. This board is typically powered by the on-board MICROB USB connector. If one would rather use a regulated 5V power supply, this can be connected through the Vin pin to directly power the ESP32 and its peripherals.

The ESP32 sleep mode is a power-saving mode. When not in use, the ESP32 can enter this mode, storing all data in RAM. At this point, all unnecessary peripherals are disabled while the RAM receives enough power to retain its data.

It takes up a current of between 5 µA and 240mA depending on the power mode it is operating in at a given moment. These include:

- **Active/Normal mode-** In this mode, all peripherals of the chip remain active.Since everything is always active in this mode (especially the WiFi module, processing core, and Bluetooth module), the chip consumes about 240 mA of power. It has also been observed that the chip draws more than 790 mA at times, particularly when both WiFi and Bluetooth are used simultaneously.
- **Modem Sleep mode** - In modem sleep mode, everything is active except for the WiFi, Bluetooth, and the radio. The CPU remains active, and the clock is configurable. In this mode, the chip consumes approximately 3 mA at slow speed and 20 mA at high speed. To keep the connection alive, Wi-Fi, Bluetooth, and the radio are woken up at predefined intervals. This is referred

to as the *Association Sleep Pattern.* During this sleep pattern, ESP32 switches between active mode and modem sleep mode. To accomplish this, the ESP32 connects to the router in station mode using the DTIM beacon mechanism. The Wi-Fi module is disabled between two DTIM beacon intervals and then automatically enabled just before the next beacon arrives. This results in power conservation. The sleeping time is determined by the router's DTIM beacon interval time, which is typically 100 ms to 1000 ms.

NB: DTIM stands for Delivery Traffic Indication Message. In this mechanism, the access point (AP)/router broadcasts beacon frames periodically. Each frame contains network-related information. It is used to announce the presence of a wireless network as well as to synchronize all connected members.

- **Light Sleep mode** - Light sleep is similar to modem sleep in that the chip follows the Association Sleep Pattern. The only difference is that in light sleep mode, the CPU, most of the RAM, and digital peripherals are clock-gated. During light sleep mode, the CPU is paused by disabling its clock pulse. The RTC and ULP-coprocessor, on the other hand, remain active. This results in a lower power consumption than the modem sleep mode, which is around 0.8 mA. Before entering light sleep mode, the ESP32 stores its internal state in RAM and resumes operation upon waking from sleep. This is referred to as Full RAM Retention.

NB: Clock gating is a popular power management technique for reducing dynamic power dissipation by removing or ignoring the clock signal when the circuit is not in use.

Clock gating reduces power consumption by pruning the clock tree. Pruning the clock disables portions of the circuitry, preventing the flip-flops in them from switching states. Since switching states consumes power, when not switched, the power consumption drops to zero.

- **Deep Sleep mode** - In deep sleep mode, the CPUs, most of the RAM, and all digital peripherals are disabled. Only the following parts of the chip remain operational:
    - ULP Coprocessor
    - RTC Controller
    - RTC Peripherals
    - RTC fast and slow memory

In deep sleep mode, the chip consumes anywhere between 0.15 mA (when the ULP coprocessor is on) and 10 µA.

During deep sleep mode, the primary CPU is turned off, whereas the Ultra-Low-Power (ULP) Coprocessor can take sensor readings and wake up the CPU as needed. This sleep pattern is referred to as the ULP sensor-monitored pattern. This is useful for designing applications where the CPU needs to be woken up by an external event, a timer, or a combination of these events, while maintaining minimal power consumption.

Along with the CPU, the main memory of the chip is also disabled. As a result, everything stored in that memory is erased and cannot be accessed.

Because RTC memory is kept active, its contents are preserved even during deep sleep and can be retrieved once the chip is woken up. This is why the chip stores Wi-Fi and Bluetooth connection data in RTC memory before entering deep sleep. When the chip wakes up from deep sleep, it performs a reset and begins program execution from the beginning.

- **Hibernation mode** - Hibernate mode is very similar to deep sleep. The only difference is that in hibernation mode, the chip disables the internal 8 MHz oscillator as well as the ULP-coprocessor, leaving only one RTC timer (on slow clock) and a few RTC GPIOs to wake the chip up. Because the RTC recovery memory is also turned off, we cannot save any data while in hibernation mode. As a result, the chip's power consumption is reduced even further; in hibernation mode, it consumes only about 2.5 µA. This mode is especially useful if you're working on a project that doesn't need to be active all the time.

g. ***<u>Security Features</u>***

Its security features are upped from the esp8266 microcontroller. It has hardware acceleration for encryption. It also has secure boot that prevents unauthorized firmware from running. Flash encryption protects contents of external flash. It supports WPA/WPA2/Enterprise encryption.

h. ***<u>Development Environment</u>***

It can support multiple programming languages like C, C++, MicroPython, JavaScript etc. It can also be used in development environments like the ESP-IDF and Arduino IDE which happen to be the most popular. It can also support MQTT, HTTP and Over the Air (OTA) updates.

2.4 ESP32 VARIANTS

The ESP32 has multiple variants that are improved versions of the original. Each of these variants has multiple modules based on the particular chip. These variants include:

ESP32 Series-

This is the original and most common variant.

- ***Main Features.***

32-bit MCU & 2.4 GHz Wi-Fi & Bluetooth/Bluetooth Les

- ESP32 embedded, two or one Xtensa® 32-bit LX6 microprocessor(s) with adjustable clock frequency, ranging from 80 MHz to 240 MHz
- +19.5 dBm output power ensures a good physical range
- Classic Bluetooth for legacy connections, also supporting L2CAP, SDP, GAP, SMP, AVDTP, AVCTP, A2DP (SNK) and AVRCP (CT)
- Support for Bluetooth Low Energy (Bluetooth LE) profiles including L2CAP, GAP, GATT, SMP, and GATT-based profiles like BluFi, SPP-like, etc
- Bluetooth Low Energy (Bluetooth LE) connects to smart phones, broadcasting low-energy beacons for easy detection
- Sleep current is less than 5 µA, making it suitable for battery-powered and wearable-electronics applications
- Peripherals include capacitive touch sensors, Hall sensor, SD card interface, Ethernet, high-speed SPI, UART, I2S and I2C
- Fully certified with integrated antenna and software stacks

MODULE	DESCRIPTION	CHIP EMBEDDED	DIMENSIONS	PINS	FLASH (MB)	PSRAM (MB)	ANETNNAE	DEVELOPMNT BOARD
ESP32-WROOM-32E	ESP32-WROOM-32E integrates ESP32-D0WD-V3, with higher stability and safety performance.	ESP32-D0WD-V3 ESP32-D0WDR2-V3	18x25.5 x3.1	38	4,8,16	N/A	PCB antenna	ESP32-DevKitC

ESP32-WROOM-32UE	ESP32-WROOM-32UE integrates ESP32-D0WD-V3, with higher stability and safety performance.	ESP32-D0WD-V3 ESP32-D0WDR2-V3	18x19.2x3.2	38	4,8,16	N/A	IPEX antenna connector	ESP32-DevKitC
ESP32-WROVER-E	ESP32-WROVER-E integrates ESP32-D0WD-V3, with higher stability and safety performance.	ESP32-D0WD-V3 ESP32-D0WDR2-V3	18x31.4x3.3	38	4,8,16	8	PCB antenna	ESP32-DevKitC ESP-WROVER-KIT ESP32-LyraT ESP32-LyraT-Mini ESP32-LyraTD-MSC ESP32-LyraTD-SYNA ESP32-Vaquita-DSPG ESP32-Korvo ESP32-Ethernet-Kit
ESP32-WRO	ESP32-WROVER-IE integrates ESP32-D0WD-	ESP32-D0WD-V3	18x31.4x3.3	38	4,8,16	8	IPEX antenna	ESP32-DevKitC

VER-IE	V3, with higher stability and safety performance	ESP32-D0WDR2-V3					connector	
ESP32-MINI-1	ESP32-MINI-1 is a highly-integrated, small-sized Wi-Fi+ Bluetooth +Bluetooth® LE MCU module that has a rich set of peripherals	ESP32-U4WDH	13.2×19×2.4	55	4 MB embedded in chip	N/A	PCB antenna	ESP32-DevKitM-1
ESP32-MINI-1U	ESP32-MINI-1U is a highly-integrated, small-sized Wi-Fi + Bluetooth® + Bluetooth® LE MCU module that has a rich set of peripherals.	ESP32-U4WDH	13.2×13.5× 2.4	55	4 MB embedded in chip	N/A	IPEX antenna connector	ESP32-DevKitM-1
ESP32-PICO-MINI-02	ESP32-PICO-MINI-02 is a small-sized powerful Wi-Fi+ Bluetooth +Bluetooth® LE MCU module. It is based on ESP32-PICO-V3-02, A System – in –Package device which integrates an 8MB SPI flash, 2MB PSRAM	ESP32-PICO-V3-02	13.2×16.6×2.4	53	8 MB embedded in chip	2 MB embedded in chip	PCB antenna	ESP32-PICO-DevKitM-2

	and a 40MHz oscillator.							
ESP32-PICO-MINI-02U	ESP32-PICO-MINI-02U is a small-sized powerful Wi-Fi+Bluetooth®+ Bluetooth® LE MCU module. It is based on ESP32-PICO-V3-02, a System-in-Package (SiP) device, which integrates an 8 MB SPI flash, 2 MB SPI Pseudo static RAM (PSRAM) and 40 MHz oscillator..	ESP32-PICO-V3-02	13.2x11.2x 2.4	53	8 MB embedded in chip	2 MB embedded in chip	IPEX antenna connector	ESP32-PICO-DevKitM-2
ESP32-PICO-V3-ZERO	ESP32-PICO-V3-ZERO is an Alexa Connect Kit (ACK) module with Espressif. ESP32-PICO-V3-ZERO is based on ESP32-PICO-V3, a System-in-Package (SiP) device. It provides complete Wi-Fi and Bluetooth® functionalities with ultra-small size, robust performance	ESP32-PICO-V3	16×23×2.3	77	4 MB embedded in chip	N/A	PCB antenna	ESP32-PICO-V3-ZERO-DevKit

	and low energy consumption.							
ESP32-WROOM-32D **ESP32-WROOM-32E**	ESP32-WROOM-32D integrates ESP32-D0WD.	ESP32-D0WD	18x25.5 x3.1	38	4,8,16	N/A	PCB antenna	ESP32-DevKitC
ESP32-WROOM-32U **ESP32-WROOM-32UE**	ESP32-WROOM-32U integrates ESP32-D0WD. It integrates a U.FL connector.	ESP32-D0WD	18x19.2 x3.2	38	4,8,16	N/A	IPEX antenna connector	ESP32-DevKitC
ESP32-SOLO-1 **ESP32-WROOM-32E**	ESP32-SOLO-1 is a powerful, generic Wi-Fi+BT+Bluetooth LE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as	ESP32-S0WD	18x25.5 x3.1	38	4	N/A	PCB antenna	ESP32-DevKitC

	voice encoding, music streaming and MP3 decoding.							
ESP32-WROOM-32 **ESP32-WROOM-32E**	SP32-WROOM-32 contains the ESP32 SoC, flash memory, high-precision discrete components, and a PCB antenna which provides outstanding RF performance in space-constrained applications.	ESP32-D0WD Q6	18x25.5 x3.1	38	4	N/A	PCB antenna	ESP32-DevKitC
ESP32-WROVER-B **ESP32-WROVER-E**	ESP32-WROVER-B is a powerful, generic WiFi-BT-Bluetooth LE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding.	ESP32-D0WD	18x31.4 x3.3	38	4,8,16	8	PCB antenna	ESP32-DevKitC ESP-WROVER-KIT ESP32-LyraTD-DSPG

ESP32-WROVER-IB **ESP32-WROVER-IE**	ESP32-WROVER-IB is a powerful, generic Wi-Fi+BT+Bluetooth LE MCU module that targets a wide variety of applications	ESP32-D0WD	18x31.4x3.3	38	4,8,16	8	IPEX antenna connector	ESP32-DevKitC
ESP32--WROOM--DA	ESP32-WROOM-DA is a powerful Wi-Fi + Bluetooth + Bluetooth LE MCU module, which has the same layout of pins as ESP32-WROOM-32E, facilitating quick and easy migration between these two modules. With two unique complementary PCB antennas in different directions on one single module,	ESP32-D0WD-V3	35.6x34.4x3.5	41	4,8,16	N/A	Two complementary PCB antennas	ESP32-DevKitC
ESP32-DU1906	ESP32-DU1906 is an industry-leading AIoT voice module powered by Espressif's	ESP32-D0WD-V3	22×42×3.5	66	8	8	PCB antenna	

	flagship chip ESP32-D0WD-V3 and Baidu's HongHu voice chip DU1906. This module integrates Wi-Fi, classic Bluetooth, Bluetooth LE, and voice processing circuits on board, providing first-class user experience for speech wake-up and interaction.							ESP32-Korvo-DU1906
ESP32-DU1906-U	ESP32-DU1906-U is an industry-leading AIoT voice module powered by Espressif's flagship chip ESP32-D0WD-V3 and Baidu's HongHu voice chip DU1906. This module integrates Wi-Fi, classic Bluetooth, Bluetooth LE, and voice processing circuits on	ESP32-D0WD-V3	22×35.5 × 3.5	66	8	8	IPEX antenna connector	ESP32-Korvo-DU1906

	board, providing first-class user experience for speech wake-up and interaction.							

ESP32-H Series

Optimized for ultra-low power consumption and is designed for IoT applications with stringent energy requirements.

- ***Main Features.***

32-bit RISC-V MCU & Bluetooth 5 (LE) & IEEE 802.15.4

- 32-bit RISC-V single-core processor that operates at up to 96 MHz
- 320 KB SRAM, 128 KB ROM, 4 KB LP Memory, and works with external flash
- 19 programmable GPIOs, with support for UART, SPI, I2C, I2S, Remote Control Peripheral, LED PWM, Full-speed USB Serial/JTAG Controller, GDMA, MCPWM
- Can be used for building Thread end devices, as well as Thread border router and Matter bridge by combining it and ESP Wi-Fi SoC

MODULE	DESCRIPTION	CHIP EMBEDDED	DIMENSIONS	PINS	FLASH	ANETNNAE	DEVELOPMNT BOARD
ESP32-H2-MINI-1	ESP32-H2-MINI-1 is a powerful, generic Bluetooth Low Energy and IEEE 802.15.4 combo module with ESP32-H2 chip at its core.	ESP32-H2FH2S ESP32-H2FH4S	13.2×16.6×2.4	53	1, 2, 4	PCB antenna	ESP32-H2-DevKitM-1

ESP32-H2-MINI-1U	ESP32-H2-MINI-1U is a powerful, generic Bluetooth Low Energy and IEEE 802.15.4 combo module with ESP32-H2 chip at its core.	ESP32-H2FH2S ESP32-H2FH4S	13.2×12.5×2.4	53	2, 4	IPEX antenna connector	ESP32-H2-DevKitM-1
ESP32-H2-WROOM-02C	The ESP32-H2-WROOM-02C module is based on the ESP32-H2 chip, and can be mounted onto the surface of a PCB board. It has 19 GPIOs and the pin supports UART, LED PWM, 12C, SPI, I2S, and RMT.	ESP32-H2FH2S ESP32-H2FH4S	18.0 × 20.0 × 3.2	29	2, 4	PCB antenna	ESP32-Module-Prog-1(R) ESP-Module-Prog-SUB-02

1. ***ESP32-C Series*** - Emphasizes cost-effectiveness and integrates Wi-Fi and Bluetooth features, ideal for simple IoT devices

- ***Main Features- ESP32-C6 Series***

32-bit RISC-V MCU & 2.4 GHz Wi-Fi 6 & Bluetooth 5 (LE) & IEEE 802.15.4

- 32-bit RISC-V single-core processor that operates at up to 160 MHz
- State-of-the-art power and RF performance
- 320 KB ROM, 512 KB SRAM, 16 KB Low-power SRAM on the chip, and works with external flash
- 30 (QFN40) or 22 (QFN32) programmable GPIOs, with support for SPI, UART, I2C, I2S, RMT, TWAI and PWM

MODULE	DESCRIPTION	CHIP EMBEDDED	DIMENSIONS	PINS	FLASH (MB)	PSRAM (MB)	ANETNNAE	DEVELOPMNT BOARD
ESP32-C6-MINI-1	ESP32-C6-MINI-1 is a ESP32-C6 based module, which supports Wi-Fi 6 in 2.4 GHz band, Bluetooth 5, Zigbee 3.0 and Thread. It's small-sized and pin-to-pin compatible with the ESP32-C3-MINI series module. With low	ESP32-C6FH4 ESP32-C6FH8	13.2×16.6 ×2.4	53	4, 8	N/A	PCB antenna	ESP32-C6-DevKitM-1

	power consumption, it is an ideal choice for a variety of IoT devices.							
ESP32-C6-MINI-1U	ESP32-C6-WROOM-1 is a ESP32-C6 based module, which supports Wi-Fi 6 in 2.4 GHz band, Bluetooth 5, Zigbee 3.0 and Thread. It's pin-to-pin compatible with the ESP32-WROOM series module. With low power consumption, it is an ideal choice for a variety of IoT devices.	ESP32-C6	18×25.5×3.2	28	4, 8, 16	N/A	PCB antenna	ESP32-C6-DevKitC-1

ESP32-C6-WROOM-1	ESP32-C6-WROOM-1 is a ESP32-C6 based module, which supports Wi-Fi 6 in 2.4 GHz band, Bluetooth 5, Zigbee 3.0 and Thread. It's pin-to-pin compatible with the ESP32-WROOM series module. With low power consumption, it is an ideal choice for a variety of IoT devices.	ESP32-C6	18×25.5×3.2	28	4, 8, 16	N/A	PCB antenna	ESP32-C6-DevKitC-1
ESP32-C6-WROOM-1U	ESP32-C6-WROOM-1U is a ESP32-C6 based module, which supports	ESP32-C6	18x19.2x3.2	28	4, 8, 16		IPEX antenna connector	

	Wi-Fi 6 in 2.4 GHz band, Bluetooth 5, Zigbee 3.0 and Thread. It's pin-to-pin compatible with the ESP32-WROOM series module. With low power consumption, it is an ideal choice for a variety of IoT devices.					N/A		ESP32-C6-DevKitC-1

➢ ***Main Features- ESP32-C61 Series***

32-bit RISC-V MCU & 2.4 GHz Wi-Fi 6 & Bluetooth 5 (LE)

- Ultra-low-power MCU with 32-bit RISC-V single-core microprocessor
- 2.4 GHz Wi-Fi 6 (802.11ax), Bluetooth® 5 (LE)
- 3.3 V flash or PSRAM in the chip's package
- 25 GPIOs
- Security features: secure boot, flash and PSRAM encryption, cryptographic accelerators, Trusted Execution Environment (TEE), and ECDSA-based Digital Signature Peripheral

MODULE	DESCRIPTION	CHIP EMBEDDED	DIMENSIONS	PINS	FLASH	PSRAM (MB)	ANETNNAE	DEVELOPMNT BOARD

					(MB)			
ESP32-C61-WROOM-1	A powerful, general-purpose 2.4G Wi-Fi 6 and Bluetooth LE module	ESP32-C61HR2	18.0 × 25.5 × 3.1	29	8	2	PCB antenna	ESP32-C61-DevKitC-1-N8R2
ESP32-C61-WROOM-1U	A powerful, general-purpose 2.4G Wi-Fi 6 and Bluetooth LE module, an ideal choice for smart homes, industrial automation, health care, consumer electronics, etc.	ESP32-C61HR2	18.0 × 25.5 × 3.1	29	8	2	IPEX Antenna Connector	ESP32-C61-DevKitC-1

➢ ***<u>Main Features- ESP32-C5 Series</u>***

32-bit RISC-V MCU & 2.4 and 5 GHz Wi-Fi 6 & Bluetooth 5 (LE) & IEEE 802.15.4

- Ultra-low-power MCU with 32-bit RISC-V single-core microprocessor
- 2.4 and 5 GHz dual-band Wi-Fi 6 (802.11ax), Bluetooth® 5 (LE), Zigbee, and Thread (802.15.4)

- Support connection to external flash and PSRAM
- 29 GPIOs, rich set of peripherals
- Security features: secure boot, flash and PSRAM encryption, cryptographic accelerators, Access Permission Management (APM) hardware block and Physical Memory Protection (PMP)

MODULE	DESCRIPTION	CHIP EMBEDDED	DIMENSIONS	PINS	FLASH (MB)	PSRAM (MB)	ANETNNAE	DEVELOPMNT BOARD
ESP32-C5-WROOM-1	A general-purpose 2.4 and 5 GHz dual-band Wi-Fi 6 (802.11ax), Bluetooth® 5 (LE), Zigbee, and Thread (802.15.4) module, ideal for smart homes, industrial automation, health care, consumer electronics, etc	ESP32-C5NR4 ESP32-C5NF4	18×27.5×3.3	29	4,8,16	4 N/A	PCB antenna	ESP32-C5-DevKitC-1
ESP32-C5-	A general-purpose 2.4 and 5	ESP32-C5NR4	18×27.5×3.3	29	8	4	IPEX antenna	

WROOM-1U	GHz dual-band Wi-Fi 6 (802.11ax), Bluetooth® 5 (LE), Zigbee, and Thread (802.15.4) module, ideal for smart homes, industrial automation, health care, consumer electronics, etc						connector	ESP32-C5-DevKitC-1

➢ ***<u>Main Features- ESP32-C3 Series</u>***

32-bit RISC-V MCU & 2.4 GHz Wi-Fi & Bluetooth 5 (LE)

- ESP32-C3 embedded, 32-bit RISC-V single-core processor, up to 160 MHz
- State-of-the-art power and RF performance
- 400KB of SRAM and 384 KB of ROM on the chip, and SPI, Dual SPI, Quad SPI, and QPI interfaces that allow connection to flash
- Reliable security features ensured by RSA-3072-based secure boot, AES-128-XTS-based flash encryption, the innovative digital signature and the HMAC peripheral, hardware acceleration support for cryptographic algorithms
- Rich set of peripheral interfaces and GPIOs, ideal for various scenarios and complex applications
- Fully certified with integrated antenna and software stacks

MODULE	DESCRIPTION	CHIP EMBEDDED	DIMENSIONS	PINS	FLASH (MB)	PSRAM (MB)	ANETNNAE	DEVELOPMNT BOARD
ESP32-C3-MINI-1	ESP32-C3-MINI-1 is a general-purpose Wi-Fi and Bluetooth LE module. This module features a rich set of peripherals and a small size, which makes it an ideal choice for smart home, industrial automation, health care, consumer electronics, etc.	ESP32-C3FH4 ESP32-C3FH4X	13.2×16.6×2.4	53	4 MB embedded in chip	N/A	PCB antenna	ESP32-C3-DevKitM-1 ESP32-C3-AWS-ExpressLink-DevKit ESP32-C3-DevKit-RUST-1
ESP32-C3-MINI-1U	ESP32-C3-MINI-1U is a powerful, generic Wi-Fi + Bluetooth LE MCU modules that have a single core CPU, a rich set of peripherals, and with an U.FL connector for external antenna. According to	ESP32-C3FH4X ESP32-C3FH4 ESP32-C3-MINI-1U-N4X	13.2×12.5×2.4	53	4 MB embedded in chip	N/A	IPEX antenna connector	ESP32-C3-DevKitM-1

	different chips integrated, ESP32-C3-MINI-1U has two variants operating in different ambient temperature. It is an ideal choice for smart homes, industrial automation, health care, consumer electronics, etc.							
ESP32-C3-WROOM-02	ESP32-C3-WROOM-02 is a general-purpose Wi-Fi and Bluetooth LE module. This module features a rich set of peripherals and pin-to-pin compatible with the ESP-WROOM-02/02D modules. It is an ideal choice for smart home, industrial automation, healthcare,	ESP32-C3	18×20×3.2	19	4	N/A	PCB antenna	ESP32-C3-DevKitC-02 ESP32-C3-Lyra

	consumer electronics, etc.							
ESP32-C3-WROOM-02U	ESP32-C3-WROOM-02U is a powerful, generic Wi-Fi + Bluetooth LE MCU modules that have a single core CPU, a rich set of peripherals, and with an U.FL connector for external antenna.	ESP32-C3	18×14.3×3.2	19	4	N/A	IPEX antenna connector	ESP32-C3-DevKitC-02

ESP32- S Series

Focuses on lower power and smaller footprint, often used in battery-powered applications.

➢ ***<u>Main Features-ESP32-S3 Series</u>***

32-bit MCU & 2.4 GHz Wi-Fi & Bluetooth 5 (LE)

- Xtensa® 32-bit LX7 dual-core processor that operates at up to 240 MHz
- 512 KB of SRAM and 384 KB of ROM on the chip, and SPI, Dual SPI, Quad SPI, Octal SPI, QPI, and OPI interfaces that allow connection to flash and external RAM
- Additional support for vector instructions in the MCU, which provides acceleration for neural network computing and signal processing workloads
- Peripherals include 45 programmable GPIOs, SPI, I2S, I2C, PWM, RMT, ADC, DAC and UART, SD/MMC host and TWAI™
- Reliable security features ensured by RSA-based secure boot, AES-XTS-based flash encryption, the innovative digital signature and the HMAC peripheral, "World Controller"
- Fully certified with integrated antenna and software stacks

MODULE	DESCRIPTION	CHIP EMBEDDED	DIMENSIONS	PINS	FLASH (MB)	PSRAM (MB)	ANETNNAE	DEVELOPMNT BOARD
ESP32-S3-WROOM-1	ESP32-S3-WROOM-1 is a powerful, generic Wi-Fi + Bluetooth LE MCU module that has a Dual core CPU, a rich set of peripherals, and provides acceleration for neural network computing and signal processing workloads. It is an ideal choice for a wide variety of application scenarios related to AI +	ESP32-S3 ESP32-S3R2 ESP32-S3R8	18×25.5×3.1	41	4, 8, 16	N/A 2 8	PCB antenna	ESP32-DevKitC ESP32-S3-DevKitC-1 ESP32-S3-BOX-3 ESP32-S3-BOX ESP32-S3-EYE ESP32-S3-Korvo-1 ESP32-S3-Korvo-2 ESP32-S3-LCD-EV-Board

	Internet of Things (AIoT), such as wake word detection and speech commands recognition, face detection and recognition, smart home, smart appliance, smart control panel, smart speaker etc.							
ESP32-S3-WROOM-1U	ESP32-S3-WROOM-1U is a powerful, generic Wi-Fi + Bluetooth LE MCU module that has a Dual core CPU, a rich set of periphera	ESP32-S3 ESP32-S3R2 ESP32-S3R8	18×19.2×3.2	41	4, 8, 16	0 2 8	IPEX antenna connector	

	ls, and provides acceleration for neural network computing and signal processing workloads. It is an ideal choice for a wide variety of application scenarios related to AI + Internet of Things (AIoT).							ESP32-S3-DevKitC-1
ESP32-S3-WROOM-2	ESP32-S3-WROOM-2 is based on ESP32-S3R16V, with flash memory of Octal 32 MB and PSRAM memory of 16 MB. It provides	ESP32-S3R16V ESP32-S3R8V (EOL)	18×25.5×3.1	41	32	16	PCB antenna	ESP32-S3-DevKitC-1

	acceleration for neural network computing and signal processing workloads. They are an ideal choice for a wide variety of application scenarios related to AI and Artificial Intelligence of Things (AIoT), such as wake word detection and speech commands recognition, face detection and recognition, smart home,							

	smart appliances, smart control panel, smart speaker, etc.							
ESP32-S3-MINI-1	ESP32-S3-MINI-1 is a powerful, generic Wi-Fi + Bluetooth LE MCU module that features a rich set of peripherals, yet an optimized size. It is an ideal choice for a wide variety of application scenarios related to Internet of Things (IoT), such as embedded systems,	ESP32-S3FN8 ESP32-S3FH4R2	15.4×20.5×2.4	65	8 MB 4MB embedded in chip	N/A 2	PCB antenna	ESP32-S3-DevKitM-1 ESP32-S3-USB-OTG ESP32-S3-USB-Bridge

	smart home, wearable electronics, etc.							
ESP32-S3-MINI-1U	ESP32-S3-MINI-1U is a powerful, generic Wi-Fi + Bluetooth LE MCU module that features a rich set of peripherals, yet an optimized size. It is an ideal choice for a wide variety of application scenarios related to Internet of Things (IoT), such as embedded systems, smart home, wearable	ESP32-S3FN8 ESP32-S3FH4R2	15.4×15.4×2.4	65	8 MB 4MB embeded in chip	N/A 2	IPEX antenna connector	ESP32-S3-DevKitM-1

	electronics, etc.							

➢ ***Main Features-ESP32-S2 Series***

32-bit MCU & 2.4 GHz Wi-Fi

- ESP32-S2 embedded, Xtensa® single-core 32-bit LX7 microprocessor, up to 240 MHz
- Ultra-low-power performance: fine-grained clock gating, dynamic voltage and frequency scaling
- Security features: eFuse, flash encryption, secure boot, signature verification, integrated AES, SHA and RSA algorithms
- Peripherals include 43 GPIOs, 1 full-speed USB OTG interface, SPI, I2S, UART, I2C, LED PWM, LCD interface, camera interface, ADC, DAC, touch sensor, temperature sensor
- Availability of common cloud connectivity agents and common product features shortens the time to market
- Fully certified with integrated antenna and software stacks

MODULE	DESCRIPTION	CHIP EMBEDDED	DIMENSIONS	PINS	FLASH (MB)	PSRAM (MB)	ANETNNAE	DEVELOPMNT BOARD
ESP32-S2-MINI-2	ESP32-S2-MINI-2 is a powerful, generic Wi-Fi MCU modules, which integrates ESP32-S2 ECO1 with improved RF	ESP32-S2FH4 ESP32-S2FN4R2	15.4×20×2.4	65	4 MB embedded in chip	N/A 2	PCB antenna	ESP32-S2-DevKitM-1

	performance.							
ESP32-S2-MINI-2U	ESP32-S2-MINI-2U is a powerful, generic Wi-Fi MCU modules, which integrates ESP32-S2 ECO1 with improved RF performance.	ESP32-S2FH4 ESP32-S2FN4R2	15.4×15.4×2.4	65	4 MB embedded in chip	N/A 2	IPEX antenna connector	ESP32-S2-DevKitM-1
ESP32-S2-SOLO-2U	ESP32-S2-SOLO-2U is a powerful, generic Wi-Fi MCU modules, which integrates ESP32-S2 ECO1 with improved RF performance.	ESP32-S2 ESP32-S2R2	18×19.2×3.2	41	4	N/A 2	IPEX antenna connector	ESP32-S2-DevKitC-1
ESP32-S2-MINI-1	ESP32-S2-MINI-1 is a	ESP32-S2FH4 ESP32-	15.4×20×2.4	65	4 MB embed	N/A 2	PCB antenna	

ESP32-S2-MINI-2	powerful, generic Wi-Fi MCU modules that have a rich set of peripherals. It's an ideal choice for a wide variety of application scenarios relating to Internet of Things (IoT), wearable electronics and smart home. At the core of this module is ESP32-S2FH4 / ESP32-S2FN4R2, an Xtensa® 32-bit LX7 CPU that operates	S2FN4R2			ded in chip			ESP32-S2-DevKitM-1

	at up to 240 MHz.							
ESP32-S2-MINI-1U **ESP32-S2-MINI-2U**	ESP32-S2-MINI-1U is a powerful, generic Wi-Fi MCU modules that have a rich set of peripherals. It's an ideal choice for a wide variety of application scenarios relating to Internet of Things (IoT), wearable electronics and smart home. At the core of this module is ESP32-S2FH4 / ESP32-S2FN4R2, an	ESP32-S2FH4 ESP32-S2FN4R2	15.4×15.4×2.4	65	4 MB embedded in chip	N/A 2	IPEX antenna connector	ESP32-S2-DevKitM-1

	Xtensa® 32-bit LX7 CPU that operates at up to 240 MHz.							
ESP32-S2-SOLO **ESP32-S2-SOLO-2**	ESP32-S2-SOLO is a powerful, generic Wi-Fi MCU modules that have a rich set of peripherals and pin-to-pin compatible with the ESP32-WROOM modules. Its an ideal choice for a wide variety of application scenarios relating to Internet of Things (IoT), wearable	ESP32-S2 ESP32-S2R2	18×25.5×3.1	40	4,8,16	N/A 2	PCB antenna	ESP32-S2-DevKitC-1

	electronics and smart home.							
ESP32-S2-SOLO-U **ESP32-S2-MINI-2U** **ESP32-S2-SOLO-2U**	ESP32-S2-SOLO-U is a powerful, generic Wi-Fi MCU modules that have a rich set of peripherals and pin-to-pin compatible with the ESP32-WROOM modules. Its an ideal choice for a wide variety of application scenarios relating to Internet of Things (IoT), wearable electronics and	ESP32-S2 ESP32-S2R2	18×19.2×3.2	40	4,8,16	N/A 2	IPEX antenna connector	ESP32-S2-DevKitC-1

	smart home.							
ESP32-S2-WROVER **ESP32-S2-MINI-2U** **ESP32-S2-SOLO-2U**	ESP32-S2-WROVER integrates ESP32-S2. It is a powerful, generic Wi-Fi MCU module that has a rich set of peripherals.	ESP32-S2	18x31x3.3	42	4,8,16	2	PCB antenna	ESP32-S2-Kaluga-1 ESP32-S2-HMI-DevKit-1 ESP32-S2-Saola-1
ESP32-S2-WROVER-I **ESP32-S2-MINI-2U** **ESP32-S2-SOLO-2U**	ESP32-S2-WROVER-I integrates ESP32-S2. It is a powerful, generic Wi-Fi MCU module that has a rich set of peripherals.	ESP32-S2	18x31x3.3	42	4,8,16	2	IPEX antenna connector	ESP32-S2-Saola-1 ESP32-S2-Kaluga-1
ESP32-S2-WROOM **ESP32-S2-**	ESP32-S2-WROOM integrates ESP32-S2. It is a powerful, generic Wi-Fi	ESP32-S2	18x31x3.3	42	4,8,16	N/A	PCB antenna	ESP32-S2-Saola-1

MINI-2U **ESP32-S2-SOLO-2U**	MCU module that has a rich set of peripherals.							
ESP32-S2-WROOM-I	ESP32-S2-WROOM-I integrates ESP32-S2. It is a powerful, generic Wi-Fi MCU module that has a rich set of peripherals.	ESP32-S2	18x31x3.3	42	4,8,16	N/A	IPEX antenna connector	ESP32-S2-Saola-1

2.5 ADVANTAGES

1. Has a dual core processor hence can handle multitasking better.
2. Has more GPIOs with support for ADC, DAC, capacitive touch, PWM, SPI, I2C, I2S UART etc.
3. Has bluetooth and WiFi support.
4. Has a larger memory capacity.
5. Has good hardware security features.
6. Has multiple low power modes.

2.6 DISADVANTAGES

1. Has a high power consumption.
2. Can be complex in use for beginners.
3. Is slightly expensive.
4. Is wider in physical sense and hence can be a problem in compact designs.
5. It has no built in display driver unless connected via a Serial Peripheral Interface.
6. The WiFi and Bluetooth share a common radio. When loaded, the microcontroller may end up experiencing a performance drop due to the internal conflict.
7. There is limited ADC accuracy and linearity especially when the WiFi is active. This is because the ADC2 unit and WiFi share internal resources. This may lead to a slack in performance and drop in accuracy when the two are used simultaneously.
8. It requires good power supply decoupling as it has sensitive analog and radio frequency performance.

2.7 APPLICATIONS OF THE MICROCONTROLLERS TO VARIOUS SATELLITE SUBSYSTEMS.

1. *Controller (On board computer - OBC)*

The ESP32 is preferred due to its higher processing ability, multitasking and built-in peripheral support. It is however not radiation hardened and may need reset logic for reliability.

2. *Communication System*

The ESP32 would be preferred due to its Bluetooth and WiFi capabilities. This will however only be useful for short range telemetry and data dump for ground testing. It has the capability of handling the bandwidth required by the hobby satellite but it would be better if it would be complemented by other modules especially for RF applications.

3. *Electrical Power System (EPS)*

ESP32 would be better used due to its multiple GPIOs, ADC capabilities and low power modes. It can handle power monitoring with ADCs, control relays and switches. Deep sleep can be employed for power saving. It however needs efficient power management.

4. *Attitude Determination and Control System (ADCS)*

ESP32 can be used with external coprocessors for a more precise orientation control. It can be used with IMUs (Inertial Measurement Units), magnetometers, sun sensors and others. Its dual core allows for parallel sensor reading and processing.

5. *Payload*

The ESP32 has a camera interface that can handle simple image capture and sensor interfacing. It can not, however, handle large images due to its limited image processing capacity and memory.

6. *Structure and Integration*

This mostly involves mechanical design and is not necessarily controlled by a microcontroller.

7. *Telemetry command and Data Handling*

The ESP32 is better as it can handle larger amounts of data better due to its dual core and faster speeds. It has more storage capacity and multiple channels for interfacing with subsystems. It does have non-volatile storage that can be used for storing telemetry logs without need of power supply. In addition, it has RTOS support that allows scheduling of tasks and reliable data management.

2.8 CONCLUSION

All factors considered, the ESP32 is undoubtedly more useful and wide in its scope of application. This can even simply be based on the various improvements made by the manufacturers to address the shortcomings of the ESP8266 microcontroller. This has been demonstrated by the massive capabilities of the ESP32 microcontroller.

Taking into account all the advantages and disadvantages of the presented microcontrollers as applied in the various subsystems, the ESP32 would be suited to serve the ***Telemetry Command and Data Handling Subsystem*** as it can handle a large chunk of data from various peripherals, and so at a good speed. The only limitation it presents is the power it consumes while in use which is indeed a worthy trade off.

3. ATTINY85

The ATtiny85 microcontroller is designed with a compact structure that combines efficient performance and power-saving capabilities. Known for its small size, this 8-bit microcontroller can manage various functions with minimal components, making it popular for both small-scale hobby projects and more demanding industrial applications. It has a program memory of 8K, allowing you to store applications efficiently while saving space. The ATTINY85 also includes multiple power-saving modes, enabling it to run efficiently on battery-operated devices. Its flexible range of input and output options further supports versatile usage, making it adaptable for a wide array of embedded system tasks. This compact, cost-effective design allows it to deliver solid performance without compromising on essential microcontroller features, offering an easy-to-use solution for developers and experimenters alike.

3.1 FEATURES OF ATTINY85 MICROCONTROLLER

1. Compact and Affordable: The ATTINY85 is known for its small size and affordability, making it a great choice for both hobby projects and professional applications. It provides essential features without taking up much space or stretching your budget.

2. Adequate Memory: With 8K of program memory, the ATTINY85 offers enough space for many applications. This allows you to store code and execute a range of tasks without needing additional components or storage.

3. Power-Saving Modes: The ATTINY85 includes several power-saving modes, which help reduce power consumption in battery-operated devices. These modes allow it to run efficiently, prolonging battery life for mobile or remote projects. These power-saving modes include:

i. **Idle Mode**: In Idle mode, the CPU stops executing instructions, but all the peripherals like timers, SPI, and ADC can keep running. This mode is useful when the microcontroller is waiting for a peripheral event (like a timer overflow or data reception) but doesn't need to actively process data. Since the system clock is still running, the ATtiny85 can wake up very quickly from this mode.

ii. **ADC Noise Reduction Mode:** This mode is designed to improve the accuracy of analog-to-digital conversions. It shuts down the CPU and most digital peripherals, which reduces electrical noise that can interfere with analog signals. Only the ADC and asynchronous interrupts remain active. It's ideal for applications requiring precise analog readings without interference from other system components.

iii. **Power-down Mode:** Power-down mode is the most energy-efficient sleep state. In this mode, the CPU and all internal peripherals are completely turned off. The only components that can wake the microcontroller are external interrupts (e.g., a button press), pin change interrupts, or the watchdog timer. This mode is perfect for long-term sleeping when the ATtiny85 doesn't need to do anything for extended periods.

iv. **Power-save Mode:** This mode is similar to Power-down, but it keeps the asynchronous timer (Timer/Counter2) running. This allows the device to maintain a real-time clock or time-based wake-up functionality without needing an external timer circuit. It is ideal for applications that require periodic activity, such as data logging or timed sensor checks.

v. **Standby Mode:** Standby mode is like Power-down mode but with the main system oscillator kept running. This allows the ATtiny85 to wake up more quickly, as it doesn't need to wait for the oscillator to start. It's useful when low power consumption is still important, but a faster wake-up time is also needed, such as in time-sensitive applications that use external crystals.

4. Flexible I/O Options: Equipped with multiple input/output options, the ATTINY85 can handle various tasks and interact with different peripherals. This flexibility makes it adaptable for many types of embedded systems.

5. Built-in Watchdog Timer: The ATTINY85 comes with a watchdog timer, which helps prevent the microcontroller from hanging or freezing. This feature increases reliability, especially for applications that need consistent and stable performance.

3.2 ADVANTAGES

i. **High Performance and Low Power:** The ATTINY85 operates as a high-performance microcontroller while using low power. Its compact 8-bit AVR design allows it to handle tasks efficiently without draining much energy, making it suitable for various applications.

ii. **Advanced RISC Architecture:** With its RISC architecture, the ATTINY85 can execute 120 powerful instructions, most within a single clock cycle. It includes 32 x 8 general-purpose working registers, allowing for quick and smooth operations and fully static operation.

iii. **Reliable Memory Options:** The microcontroller offers 8K bytes of in-system programmable flash memory, supporting up to 10,000 write/erase cycles. It also includes 512 bytes of in-system programmable EEPROM (up to 100,000 write/erase cycles) and 512 bytes of internal SRAM, making it suitable for a variety of data storage needs.

iv. **Strong Data Protection:** The ATTINY85 includes a programming lock to protect your stored data and code, adding an extra layer of data security in memory.

v. **Versatile Peripheral Features:** Equipped with an 8-bit timer/counter with a prescaler, two PWM channels, and a high-speed timer, the ATTINY85 handles timing and pulse-width modulation tasks smoothly. It also supports a universal serial interface (USI) with a start condition detector and has a 10-bit ADC for handling analog signals.

vi. **Enhanced ADC Options:** The 10-bit ADC offers four single-ended channels and two differential pairs with programmable gain, making it suitable for analog signal measurement. Temperature measurement is also possible, adding flexibility to its capabilities.

vii. **Built-in Watchdog Timer:** The ATTINY85 includes a watchdog timer with a separate on-chip oscillator to keep it running consistently, helping prevent system errors from causing it to freeze or stall.

viii. **On-chip Analog Comparator:** With an on-chip analog comparator, the ATTINY85 can quickly compare analog input signals, which is useful for applications requiring signal analysis or comparisons.

ix. **Special Microcontroller Features:** The ATTINY85 offers debug WIRE for easy debugging, in-system programming via SPI port, and both external and internal interrupt sources. It also includes low-power idle modes, ADC noise reduction, and power-down modes to manage power efficiently.

x. **Enhanced Power and Voltage Support:** The enhanced power-on reset circuit and programmable brown-out detection circuit ensure reliability in varying power conditions. An internal calibrated oscillator provides stable operation without needing an external oscillator.

xi. **Programmable I/O Lines:** With six programmable input/output lines, the ATTINY85 provides flexibility in how you connect it to other devices or components, allowing more versatile setups.

xii. **Wide Operating Voltage Range:** Operating between 2.7 to 5.5V, the ATTINY85 can handle a variety of voltage inputs, making it compatible with different power sources.

xiii. **Speed and Temperature Compatibility:** The microcontroller performs efficiently across a range of speeds, reaching up to 20 MHz at higher voltages. It also operates within an industrial temperature range, from -40°C to 85°C, making it suitable for diverse environments.

xiv. **Low Power Consumption:** In active mode, the ATTINY85 draws just 300 µA at 1 MHz, 1.8V. In power-down mode, it reduces consumption to 0.1 µA at 1.8V, making it an excellent choice for battery-powered devices and power-sensitive applications.

3.3 DISADVANTAGES

i. **Limited I/O Pins:** The ATtiny85 comes in an 8-pin package, out of which only 6 pins are available for general-purpose input/output (GPIO). If you use one of those pins for resetting (which is the default behavior of pin PB5), you're left with just 5 usable I/O pins. This significantly limits the number of external components (like sensors, LEDs, or actuators) that can be connected directly to the chip. Although creative multiplexing or communication protocols like I2C can help expand its functionality, the limited number of pins remains a significant constraint in more complex designs.

ii. **No Hardware UART:** Unlike more advanced microcontrollers (like the ATmega328P used in Arduino Uno), the ATtiny85 does not have a built-in UART (Universal Asynchronous Receiver/Transmitter) for serial communication. While it is possible to implement UART through software, often called "SoftwareSerial", it's not as reliable or efficient, especially at higher baud rates. This limitation makes debugging or communicating with other serial devices more challenging and less robust.

iii. **No Native USB Support:** The ATtiny85 does not come with built-in USB support. However, versions like Digispark boards use a custom bootloader to emulate USB, allowing you to program or use the microcontroller as a basic USB device. This workaround is functional but limited. It can only simulate certain USB devices and doesn't provide full USB stack support. If your project requires native USB connectivity (like HID, MIDI, or mass storage), the ATtiny85 would not be an ideal choice without significant compromises.

iv. **Small Memory Capacity:** The microcontroller has only 8 KB of flash memory for code, 512 bytes of SRAM, and 512 bytes of EEPROM. This is sufficient for simple programs but quickly becomes a limitation as your project grows in complexity. Libraries, especially those with graphical or network functionality, often take up more memory than the ATtiny85 can handle. Developers need to write highly optimized, minimal code and be cautious with data structures and variable usage.

v. **No On-Chip Debugging Tools:** Unlike higher-end microcontrollers that come with dedicated debugging interfaces (such as JTAG or SWD), the ATtiny85 lacks native debugging hardware. This makes it harder to step through code, set breakpoints, or inspect memory during runtime. Debugging is usually done via blinking LEDs, serial prints (if SoftwareSerial is implemented), or simulation tools. For beginners or complex development workflows, this limitation can make troubleshooting time-consuming and less precise.

vi. **Limited Library and Peripheral Support:** While the ATtiny85 can be programmed through the Arduino IDE, not all Arduino libraries are compatible due to its smaller memory, lack of certain hardware features (like UART or advanced timers), and architectural differences. Some functions and libraries may need to be rewritten or trimmed to fit within the constraints. This can be an obstacle, especially for users relying on plug-and-play code or advanced peripherals like servo controllers, displays, or networking modules.

3.4 PROGRAMMING ATTINY85 MICROCONTROLLER

Programming the ATTINY85 involves several simple steps to get your code ready and running on the device. Below is a quick guide on the programming of this microcontroller:

1. First, outline what tasks you want the microcontroller to perform. Listing these tasks will give you a clear direction as you develop your code.
2. Next, write the functions or commands that the microcontroller will need to perform the tasks you listed. Using an Integrated Development Environment (IDE) can make this process easier.
3. After writing the functions, compile your code in the IDE to check for any errors. This step ensures that the code will run smoothly when transferred to the microcontroller.
4. Once compiled, the IDE will produce a HEX file, which is the format needed for the ATTINY85 to understand and execute the code. This file contains the machine instructions that will be stored on the microcontroller.
5. Select a programming device, like an SPI programmer, which will connect your computer to the ATTINY85. This programmer allows you to transfer the HEX file directly to the microcontroller’s memory.
6. Using the programmer’s software, select the HEX file you generated and upload it to the ATTINY85. This step installs your code on the device, preparing it to execute as soon as it’s powered.
7. Finally, disconnect the programmer and connect any peripherals your application needs. Once powered, the ATTINY85 will begin executing the code you've loaded, completing the setup process.

3.5 APPLICATIONS

1. **Development Boards and Hobby Projects:** The ATTINY85 is widely used in development boards and DIY electronics projects due to its compact size and flexibility. It's a great choice for anyone looking to create custom gadgets or automate simple tasks at home.

2. **Industrial Control Systems:** In industrial settings, the ATTINY85 can manage control systems where space and power efficiency are needed. Its stability and reliability make it suitable for basic control tasks in machinery or automated processes.

3. **Power Regulation and SMPS Systems:** The ATTINY85 is commonly used in power regulation and switch-mode power supply (SMPS) systems. Its ability to control and monitor power makes it ideal for devices that require consistent power management.

4. **Analog Signal Measurement and Manipulation:** With its built-in ADC (analog-to-digital converter), the ATTINY85 can measure and manipulate analog signals. This feature is useful in applications that require data collection from sensors or other analog sources.

5. **Embedded Systems in Appliances:** The ATTINY85's compact design allows it to be embedded in appliances like coffee machines and vending machines, where it can manage basic operations or interface with user controls.

6. **Display Units:** The microcontroller can also be used to drive display units, controlling LEDs or small screens in devices where you need a simple display without complex hardware requirements.

7. **Peripheral Interface Systems:** The ATTINY85 is often used as an interface between different peripherals, allowing various components in a system to communicate effectively. This application is common in both consumer electronics and custom-designed circuits.

4. ATMEGA328P

The ATmega328P is a highly popular 8-bit AVR microcontroller manufactured by Microchip Technology. It's widely known as the microcontroller at the heart of the Arduino Uno development board, which has made it a favorite in education, DIY projects, and prototyping. The chip offers a great balance between power, memory, and features, making it suitable for a variety of applications. Whether you're building a simple blinking LED project or a more complex sensor-based system, the ATmega328P provides a reliable and versatile platform for development. With support for digital and analog inputs, communication interfaces, and power-saving modes, it's ideal for both battery-powered and plug-in devices.

4.1 FEATURES

1. **User-Friendly for Beginners and Professionals**: The ATmega328P is easy to use, especially through the Arduino platform. It requires minimal setup and is supported by thousands of tutorials and online examples. Students can start coding and testing their ideas without deep technical knowledge of hardware.

2. **Plenty of Memory for Embedded Projects**: With 32 KB of flash memory for storing program code, 2 KB of SRAM for running variables and buffers, and 1 KB of EEPROM for storing permanent data, the ATmega328P provides enough memory for most small-to-medium embedded applications.

3. **Wide Range of Communication Interfaces**: The ATmega328P supports UART, SPI, and I2C interfaces, allowing it to communicate with various external devices like sensors, displays, memory cards, wireless modules, and other microcontrollers.

4. **Versatile Input and Output Pins**: It offers up to 23 digital I/O pins, 6 analog input channels, and 6 PWM outputs, allowing the chip to interact with LEDs, motors, sensors, and many other components used in real-world embedded systems.

5. **Built-in Timers and PWM Support**: The microcontroller includes two 8-bit timers and one 16-bit timer for precise time-based operations. These are useful for scheduling tasks, generating delays, or creating accurate PWM signals for motor control and dimming LEDs.

6. **Energy-Efficient Operation with Sleep Modes**: The ATmega328P supports multiple sleep modes like Idle, Power-down, Power-save, and ADC Noise Reduction, which make it suitable for low-power and battery-powered applications. It can wake up on interrupts, ensuring efficiency without sacrificing performance.

7. **Reliable in Harsh Environments**: It operates across a wide temperature range (-40°C to 85°C) and voltage range (1.8V to 5.5V), making it robust enough for use in industrial and outdoor applications.

4.2 ADVANTAGES

i. **Beginner-Friendly and Well-Supported**
One of the biggest strengths of the ATmega328P is how easy it is to get started with. Through the Arduino IDE, users can program it in simple C++-like code without needing to understand complex register-level programming. The large Arduino community ensures a wealth of libraries, sample codes, and project guides that make development much smoother for students and hobbyists.

ii. **Adequate Memory for Practical Projects**
With 32 KB of flash memory, most typical embedded programs can be written without running out of space. This includes room for sensor-reading, control logic, and even simple user interfaces. The 2 KB of SRAM allows for enough variables, sensor readings, and buffers for tasks like serial communication, while the 1 KB of EEPROM is perfect for storing settings or logs even when the device loses power.

iii. **Multiple Digital and Analog Interfaces**
The ATmega328P includes a mix of digital and analog I/O, making it suitable for a wide variety of applications. It can read sensor values using its 10-bit ADC, output analog-like signals with PWM, and control devices such as relays, LEDs, and buzzers. This flexibility allows the microcontroller to be at the core of many different systems, from data loggers to control panels.

iv. **Strong Communication Capabilities**
With SPI, I2C, and UART support, it is easy to connect the ATmega328P to external memory, displays, sensors, or wireless modules like the HC-05 Bluetooth or ESP8266 Wi-Fi. These communication features make it perfect for applications that require networking or real-time data sharing.

v. **Integrated Timers and PWM Outputs**
Timers are essential for creating delays, measuring time, or generating square waves. The ATmega328P has three hardware timers that support multiple features including Pulse Width Modulation (PWM), which is used for applications such as motor control, sound generation, or dimming LEDs.

vi. **Power-Efficient and Battery-Friendly**
The microcontroller supports various power-saving modes that let it sleep when idle. For example, in Power-down mode, it draws almost no current and wakes up only through external or watchdog interrupts. This is especially useful in battery-operated systems where conserving power is essential, such as remote sensors or wearable devices.

vii. **Stable and Durable Operation**
The ATmega328P is built to handle a wide range of conditions. It includes built-in

safety features like a watchdog timer that resets the system if it hangs, a brown-out detector that prevents errors during voltage drops, and a programmable oscillator for consistent timing.

viii. **Cost-Effective and Easily Available**
Due to its widespread use, the ATmega328P is both affordable and easy to find. It's included in many beginner kits and development boards, such as the Arduino Uno, which adds USB programming and power regulation features to make it even more accessible.

ix. **Highly Documented and Supported**
There’s a vast amount of official documentation, user guides, datasheets, and third-party tutorials available. This makes it easy for learners and developers to understand the chip, debug their projects, and extend functionality with confidence.

4.3 DISADVANTAGES

i. **Limited RAM**
With only 2 KB of SRAM, the ATmega328P cannot handle large data buffers or memory-heavy applications. Programs need to be memory-efficient.

ii. **No Built-in USB Support**
The chip does not have native USB connectivity, unlike the ATmega32U4. To communicate over USB, it requires an external USB-to-serial converter.

iii. **No Built-in Wireless Communication**
Unlike newer microcontrollers (like the ESP32), it doesn’t have Wi-Fi or Bluetooth. You must connect external modules, which adds complexity and cost.

iv. **8-bit Architecture**
While it's suitable for many control tasks, it’s not good for high-precision calculations or applications that require floating-point operations or large number processing.

v. **No Hardware Multitasking**
It can only execute one instruction at a time in a single-threaded loop. Advanced real-time multitasking is not possible without external software structures or RTOS-like designs.

vi. **Slower Compared to Modern Chips**
With a top speed of 20 MHz, it is slower than newer 32-bit chips, limiting its use in high-performance embedded systems.

4.4 PROGRAMMING ATMEGA328P MICROCONTROLLER

Programming the ATmega328P is straightforward, especially using the Arduino platform:

1. **Plan the Application**: List the tasks the microcontroller will perform (e.g., read sensor, blink LED).
2. **Write the Code**: Use the Arduino IDE to write and test the code. Use built-in functions and libraries.
3. **Compile the Code**: The IDE checks for errors and converts the code into a HEX file.
4. **Upload the Code**: Use a USB cable (if using Arduino Uno) or a programmer (like USBasp or Arduino as ISP) to upload the code to the microcontroller.
5. **Run the Program**: Once uploaded and powered, the ATmega328P starts executing the code and interacting with peripherals.

4.5 ATMEGA328P APPLICATIONS

1. **Arduino Development Boards**: The most common use, powering Arduino Uno and Nano.
2. **Educational Projects**: Used in classrooms to teach programming and electronics.
3. **Automation Systems**: Controls lighting, fans, alarms, or smart devices.
4. **Data Loggers**: Collects data from sensors for analysis or storage.
5. **Robotics**: Manages motors, sensors, and decision-making in robots.
6. **Consumer Electronics**: Embedded in home gadgets, appliances, and toys.
7. **Wearables and Portable Devices**: Used in battery-efficient designs due to its power-saving modes.

4.6 CONCLUSION

In designing a hobby satellite system, especially a bench model that simulates a real satellite's subsystems, it's important to select microcontrollers based on the specific roles they will play. A typical satellite system includes modules such as a central flight controller, sensor interfaces, communication links, power management, and possibly actuators (like motors or servos). Both the ATmega328P and ATtiny85 are 8-bit microcontrollers, but they are quite different in terms of capability and are best suited for different tasks within the system.

1. THE MAIN CONTROL SYSTEM (ATMEGA328P)

The **ATmega328P** is best suited to act as the **central control unit** or "brain" of the satellite system. This is because it has more memory (32 KB of flash and 2 KB of SRAM), allowing it to handle more complex logic, such as reading sensor data, executing commands, logging information, and communicating with the ground station. It supports **multiple communication protocols**, like **UART, SPI, and I2C**, making it compatible with a wide range of sensors and modules, such as GPS, gyroscopes, IMUs, and radios.

In addition, the ATmega328P offers a greater number of I/O pins (23 in total), which is essential when interfacing with multiple components. It also features built-in **timers and PWM outputs**, which are useful for simulating or controlling actuators like motors or servo-based antenna alignment systems. Because this chip is used in **Arduino Uno** boards, it is supported by the Arduino IDE and thousands of libraries and examples online. This makes development, debugging, and integration much easier, especially for students working on a prototype or learning project.

For example, in our bench model, the ATmega328P can manage the main logic—such as reading telemetry from sensors, making decisions, and simulating a data transmission to a base station via serial or radio. It can also manage time-based tasks like data logging and checking system status periodically.

2. SUPPORT ROLES (ATTINY85)

The **ATtiny85**, on the other hand, is a much smaller and more limited microcontroller. It has **only 8 KB of flash** and **512 bytes of SRAM**, with just **6 I/O pins**. While this makes it unsuitable for a central controller role, it is **perfect for handling small, specific tasks** as a **support microcontroller**. Its small size and low power consumption make it ideal for modules that do not require many peripherals or complex logic.

For instance, the ATtiny85 could be used to control a **power management module**, monitoring battery voltage using its ADC, switching between power sources, or turning on indicators (like LEDs) to show power status. Another good use case is as a dedicated **sensor interface** for a single analog sensor, such as a temperature sensor or light sensor, where it simply reads the value and reports it to the main controller. Because the ATtiny85 supports sleep modes and uses very little power in standby, it's also ideal for **low-power background tasks**.

Even though the ATtiny85 does not have a built-in UART, it can communicate using **SPI or I2C**. In our bench model, we could use I2C to let the ATtiny85 send periodic battery status updates to the ATmega328P, or use its PWM output to drive a buzzer or status LED based on simple thresholds.

COMPARISON AND RECOMMENDATION

In summary, the **ATmega328P** is the better choice for the **core control system** of the satellite bench model because it has more memory, more I/O pins, stronger communication options, and greater flexibility for multitasking. It is the ideal candidate for handling sensor data processing, actuator control, data logging, and communication with a PC or base station.

Meanwhile, the **ATtiny85** shines in **supporting roles**, where minimal input/output and low power consumption are important. It is perfect for dedicated monitoring tasks, handling individual components, or acting as a secondary controller for modules like power systems or simple alert mechanisms.

By using both microcontrollers together, we can divide tasks efficiently: let the **ATmega328P focus on system logic and communication**, while the **ATtiny85 manages background tasks and peripheral monitoring**. This separation of roles leads to a more organized, efficient, and modular satellite system design.

5. ARDUINO MEGA 2560

The Arduino Mega 2560 Rev3 is a high-performance development board designed for advanced and multi-interface applications. It is built around the ATmega2560 microcontroller, making it suitable for complex embedded systems and satellite subsystem simulations where multiple peripherals must be monitored and controlled simultaneously.

5.1 PIN LAYOUT

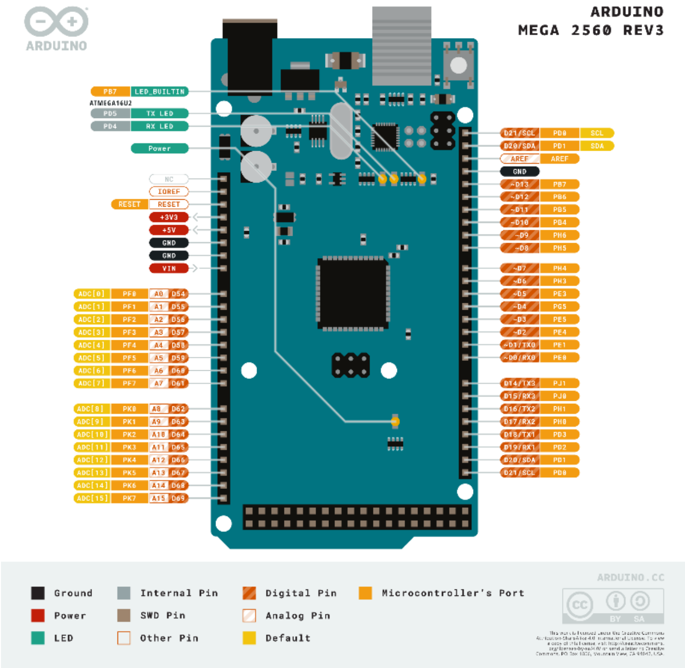


Figure 12 Arduino Mega 2580 pin layout

5.2 FEATURES

- **Processor:**
  Equipped with the ATmega2560, an 8-bit AVR RISC-based microcontroller operating at 16 MHz, the Mega 2560 delivers stable performance for control-intensive tasks. The 16 MHz crystal oscillator provides accurate timing, essential for time-sensitive satellite applications.
- **Memory:**
  - **Flash Memory:** Offers 256 KB of program memory, with 8 KB reserved for the bootloader, enabling deployment of large firmware with robust functionality.
  - **SRAM:** Includes 8 KB of static RAM, sufficient for handling moderate data buffers, temporary sensor readings, and telemetry queues.

- **EEPROM:** Provides 4 KB of non-volatile memory for storing persistent settings, calibration data, or logs.

- **Power Supply and Voltage Tolerance:**
    - Operates at a logic level of 5V and is compatible with many sensors and actuators.
    - Accepts input voltage from 7–12V via barrel jack or VIN pin. The onboard voltage regulator manages internal voltage levels safely.
    - The 3.3V pin supplies up to 50 mA, useful for powering low-current 3.3V devices.
    - Each I/O pin can safely provide or sink up to 20 mA, with an absolute maximum of 40 mA, making it suitable for direct LED drive or signal control without the need for transistors in low-power applications.
- **I/O Capabilities:**
    - 54 digital I/O pins, with 15 capable of PWM output, allow the board to interface with a large number of digital sensors, switches, relays, or actuators.
    - 16 analog input pins support 10-bit resolution ADC, suitable for monitoring analog signals like temperature, voltage levels, or sensor outputs.
    - PWM pins can be configured for power control of devices like motors, LEDs, and DC/DC converters in EPS systems.
- **Communication Interfaces:**
    - Features 4 hardware UARTs (Serial, Serial1, Serial2, Serial3), ideal for communicating with multiple serial devices such as radios, GPS modules, or telemetry units.
    - Includes 1 I²C (TWI) interface for sensors and peripherals with addressable communication.
    - 1 SPI interface for high-speed data communication with SD cards, shift registers, or radio modules.
- **USB Connectivity:**
  Uses an ATmega16U2 chip as a USB-to-serial converter, enabling easy programming and serial monitoring from a PC. The separate USB controller ensures stable operation without interrupting MCU processing.
- **Physical Dimensions and Weight:**
  Measures 101.52 mm × 53.3 mm, and weighs approximately 37 grams, making it relatively large but mechanically stable for mounting in development testbeds.

5.3 ADVANTAGES

1. **Extensive I/O Capability**

With **54 digital pins** (15 PWM) and **16 analog inputs**, it offers flexibility for interfacing with multiple subsystems, which is ideal for satellite payloads with complex wiring and parallel sensor arrays.

2. **Multiple Hardware Serial Ports (4 UARTs)**

Enables **simultaneous communication with multiple peripherals**, such as GPS, telemetry radios, and sensor modules—crucial for On-Board Computers (OBCs) or Telemetry, Command, and Data Handling units.

3. **Large Program and Data Memory**

With **256 KB of flash** and **8 KB of SRAM**, it can support larger firmware with structured task management and buffer space for sensor data or command queuing.

4. **Modular Development Support**

Compatible with many Arduino **shields** and **sensor modules**, the board can be rapidly adapted for educational and prototyping use in aerospace or research environments.

5. **Robust Power Handling**

It is designed to operate reliably with **up to 20 mA per I/O pin**, and can source 3.3V at 50 mA, which is suitable for many external sensor systems without requiring additional regulators.

5.4 DISADVANTAGES

1. **Large and Heavier**

At over **100 mm in length and 37 g**, the Mega is physically bulky, making it less suitable for compact payloads or dense satellite subsystems where PCB space is limited. However, using only the microcontroller chip allows for custom PCB design, enabling significant reductions in physical size, weight, and power consumption.

2. **Higher Power Consumption**

The multiple onboard chips (ATmega2560 + ATmega16U2 for USB) contribute to **increased idle current**, which must be accounted for in EPS design and may drain batteries faster.

3. **5V Logic Only**

Not natively compatible with modern 3.3V sensors without logic level shifting, which introduces complexity and potential signal integrity issues.

4. **No Native USB or HID Support**

Cannot function as a USB keyboard, mouse, or native COM port emulator without additional firmware; less useful in PC simulation or human-interface contexts.

5. **Limited Processing Speed**

Despite its memory, the **16 MHz 8-bit architecture** lacks advanced processing features (e.g., floating-point unit), limiting performance in real-time control or image processing.

6. ARDUINO LEONARDO

The Arduino Leonardo is a compact and cost-effective microcontroller board ideal for simpler or USB-interfaced systems. It is powered by the ATmega32U4 microcontroller, which includes built-in USB capability, allowing the board to emulate USB peripherals such as a mouse, keyboard, or virtual COM port.

6.1 PIN LAYOUT

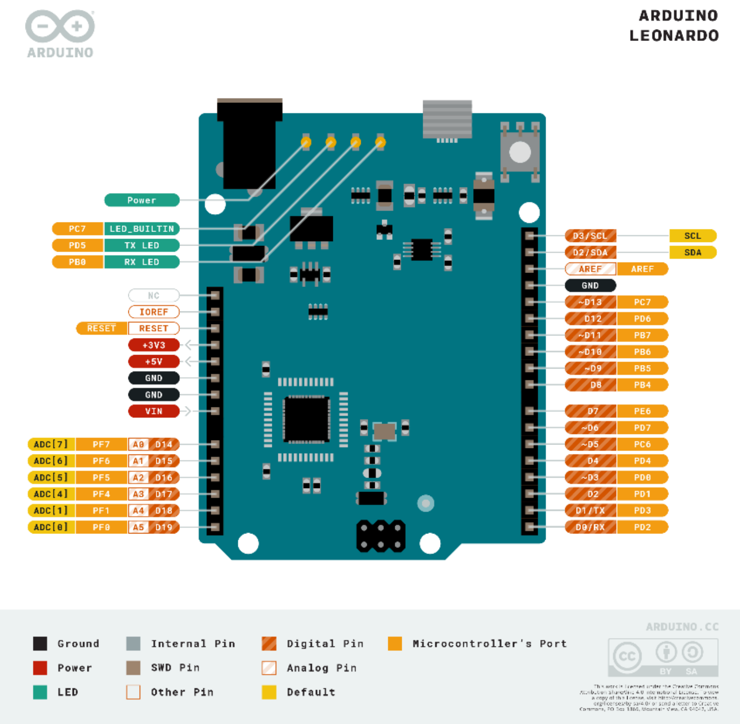


Figure 13 Arduino Leonardo pin layout

6.2 FEATURES

- **Processor:**
  Incorporates the ATmega32U4, a versatile 8-bit AVR MCU running at 16 MHz. It supports full USB device functionality, eliminating the need for a separate USB-to-serial chip and reducing board complexity and power consumption during ground-based testing.

- **Memory:**
  - **Flash Memory:** Contains 32 KB of program memory, with 4 KB reserved for the bootloader, sufficient for smaller or task-specific firmware routines.

- **SRAM:** Offers 2.5 KB of RAM, limiting the amount of simultaneous data it can buffer but enough for light telemetry or control applications.
- **EEPROM:** Features 1 KB of onboard EEPROM for storing calibration constants, telemetry logs, or system settings between power cycles.

- **Power Supply and Voltage Tolerance:**
  - Operates at 5V logic, compatible with legacy 5V sensors and actuators.
  - Accepts input voltages from 7–12V via barrel jack or VIN pin, regulated down for safe operation.
  - Each digital pin can source or sink up to 40 mA, allowing simple peripherals like status LEDs or low-power drivers to be connected directly.
- **I/O Capabilities:**
  - Provides 20 digital I/O pins, including 7 PWM-capable outputs, making it suitable for basic control of actuators and dimmable components.
  - Has 12 analog input pins, connected to a 10-bit ADC for reading voltages from sensors, potentiometers, or monitoring circuits.
- **Communication Interfaces:**
  - Includes 1 hardware UART (Serial1) for communication with serial devices such as GPS or radios.
  - Supports I²C (TWI) via digital pins 2 (SDA) and 3 (SCL) for addressable devices.
  - SPI communication is available through the dedicated ICSP header, freeing up digital pins for other tasks.
- **USB Connectivity:**
  The standout feature is its native USB 2.0 interface, allowing the Leonardo to appear as multiple USB devices simultaneously (e.g., mouse + keyboard + serial). This feature makes it especially well-suited for ground-testing, human-interfacing, or simulation tools.
- **Physical Dimensions and Weight:**
  Measures 68.6 mm × 53.3 mm, and weighs approximately 20 grams, offering a smaller footprint compared to the Mega while retaining enough headers for moderate peripheral interfacing.

6.3 ADVANTAGES

1. **Compact Size and Lightweight**

It is more space-efficient and easier to integrate into compact satellite modules or small payload enclosures.

2. **Built-in USB Capability**

The native USB support allows the board to act as a USB keyboard, mouse, joystick, or virtual COM port, which is useful for ground testing, PC-based simulations, or in applications where operator input is needed.

3. **Simplified Hardware Design**

No external USB-to-serial chip reduces component count and potential failure points, making the board slightly more power-efficient and less electrically noisy.

4. **Sufficient Analog Input Capability for Basic EPS or Sensor Tasks**

Offers 12 analog inputs, allowing for decent coverage in small-scale EPS monitoring or multi-sensor experiments.

5. **Low Cost and High Availability**

It is cheaper, making it attractive for budget-conscious educational projects or single-subsystem testing.

6.4 DISADVANTAGES

1. **Limited Serial Communication Interfaces**

Only one hardware UART (Serial1) restricts the number of serial devices it can communicate with simultaneously, making it unsuitable for complex systems involving multiple radios, GPS, or UART-based sensors.

2. **Smaller Memory Size**

With only 32 KB flash and 2.5 KB SRAM, it becomes constrained in applications involving multiple tasks, logging, or sensor data buffering.

3. **Fewer I/O and PWM Channels**

Compared to the Mega, its 20 digital pins and 7 PWM outputs limit the number of actuators or sensors it can drive/control directly.

4. **No Independent USB Interface**

While it can emulate USB devices, this USB port is also used for programming and debugging, which can cause functional overlaps and inconvenience in embedded setups.

5. **Still Uses 5V Logic**

Like the Mega, it operates at 5V logic, which limits compatibility with modern 3.3V sensor modules without using level shifters.

6. **Not Ideal for Centralized Subsystem Control**

Better suited for dedicated or single-function roles, such as USB interface simulation or one-sensor payload control, rather than managing integrated subsystems.

6.5 Subsystem Suitability in Hobby Satellite Development

1. Electrical Power System (EPS)

The Mega is well suited for EPS control due to its multiple analog inputs and PWM outputs, enabling monitoring of voltage and current across multiple power buses and control of power switches. The 16 ADC channels and 15 PWM outputs make it possible to implement features like MPPT (maximum power point tracking), battery charging, and fault protection.

The Leonardo can handle basic EPS tasks such as monitoring a small number of analog sensors and toggling relays or switches. Its 12 ADC channels are sufficient for simple EPS tasks, and its USB support is useful during ground simulation and testing phases.

2. On-Board Computer (OBC)

The Arduino Mega is favorable for use as an On-Board Computer in hobbyist satellites. Its flash memory and RAM, along with multiple UART ports, allow it to interface simultaneously with subsystems such as EPS, ADCS, telemetry radios, and payload sensors. Its watchdog timer and brown-out detection provide basic reliability for long-duration tasks, and similar microcontroller chips have been used in actual satellite missions like ArduSat.

3. Attitude Determination and Control System (ADCS)

The Mega supports multiple PWM and ADC channels for driving actuators (like magnetorquers) and reading from sun sensors or magnetometers. Its multiple serial ports are beneficial for interfacing with Internal Measurement Units or gyroscopes.

4. Payload Management

Both boards can be used for basic payload control, such as data acquisition from environmental sensors or logging to an SD card. The Mega's extra memory and I/O make it more suitable for interfacing with multiple sensors or SD storage simultaneously, while the Leonardo is adequate for simple sensor experiments or single-device payloads.

Neither board is capable of handling high-throughput imaging payloads or heavy real-time processing, but both are suitable for low-data-rate scientific payloads.

5. Telemetry, Command, and Data Handling (TC&DH)

The Mega's multiple UARTs allow it to manage communication with multiple radios, while concurrently handling commands and internal subsystem telemetry. Its memory and pin count allow for implementation of packet parsing, storage, and simple error correction logic.

The Leonardo is more restricted, with only one UART. It can still function in basic telemetry applications but is not ideal for systems needing multiple serial connections.

6. Communication System

While neither board contains an onboard RF front-end, both can interface with communication modules (e.g., LoRa, UHF, Bluetooth) via UART or SPI. The Mega's expanded I/O and multiple communication buses make it preferable for managing multiple devices or implementing serial protocols.

6.6 Cost and Availability

Both the Arduino Mega 2560 and Arduino Leonardo boards are widely available and affordable, making them accessible for hobby satellite development. Their standalone microcontroller chips are also available for more compact, power-efficient, and custom PCB designs, which are essential for systems such as in a satellite payload.

- Arduino Mega 2560 Board: Typically costs Ksh. 2,100 – 2,700
- ATmega2560 Microcontroller Chip (for Mega): Approximately Ksh. 1400
- Arduino Leonardo Board: Priced lower, at Ksh. 1,500 – 2,100
- ATmega32U4 Microcontroller Chip (for Leonardo): Approximately Ksh. 600

For prototyping or ground simulation, the full boards are cost-effective and easier to implement.

6.7 CONCLUSION AND RECOMMENDATION

The comparison between the Arduino Mega 2560 and Arduino Leonardo highlights that while both are valuable microcontrollers in hobby satellite development, they can serve different roles based on their technical capabilities.

The **Arduino Mega 2560** is recommended for use in the **On-Board Computer (OBC)** subsystem of a hobby satellite project. Its architecture offers critical features that align well with the functional requirements of an OBC.

- **Multiple Serial Interfaces:** The Mega provides four hardware UARTs, allowing it to communicate simultaneously with multiple serial devices such as telemetry radios, GPS modules, IMUs, and ground-test interfaces. This reduces the need for software-based serial emulation, which can introduce delays and data loss under high communication loads.
- **Ample Memory and I/O Capability:** With 256 KB of flash memory and 8 KB of SRAM, the Mega can accommodate more complex firmware, including command parsing, data packet handling, and error detection routines. Its 54 digital I/O pins and 16 analog inputs make it highly flexible for interfacing with a wide range of satellite subsystems and sensors.

7. ARDUINO UNO

7.1 ARDUINO UNO R3

Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

7.1.1 DIAGRAM

Fig 5. Arduino UNO R3 diagram

7.1.2 FEATURES

Here you will find the technical specifications for the Arduino UNO R3.

Table 5 Arduino UNO R3 features

Board	
Name	Arduino UNO R3
SKU	A000066

Microcontroller	
ATmega328P	
USB connector	
USB-B	
Pins	
Built-in LED Pin	13
Digital I/O Pins	14
Analog input pins	6
PWM pins	6
Communication	
UART	Yes
I2C	Yes
SPI	Yes
Power	
I/O Voltage	5V

Input voltage (nominal)	7-12V
DC Current per I/O Pin	20 mA
Power Supply Connector	Barrel Plug
Clock speed	
Main Processor	ATmega328P 16 MHz
USB-Serial Processor	ATmega16U2 16 MHz
Memory	
ATmega328P	2KB SRAM, 32KB FLASH, 1KB EEPROM
Dimensions	
Weight	25 g
Width	53.4 mm
Length	68.6 mm

More features;

1. Replaceable chip - The ATmega328P can easily be replaced, as it is not soldered to the board.
2. EEPROM **-** The ATmega328P also features 1kb of EEPROM, a memory which is not erased when powered off.
3. Battery Connector - The Arduino UNO features a barrel plug connector, that works great with a standard 9V battery.

7.1.3 Communication

Arduino can be used to communicate with a computer, another Arduino board or other microcontrollers. The ATmega328P microcontroller provides UART TTL (5V) serial communication which can be done using digital pin 0 (Rx) and digital pin 1 (Tx). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The ATmega16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1)

7.1.4 Pin out

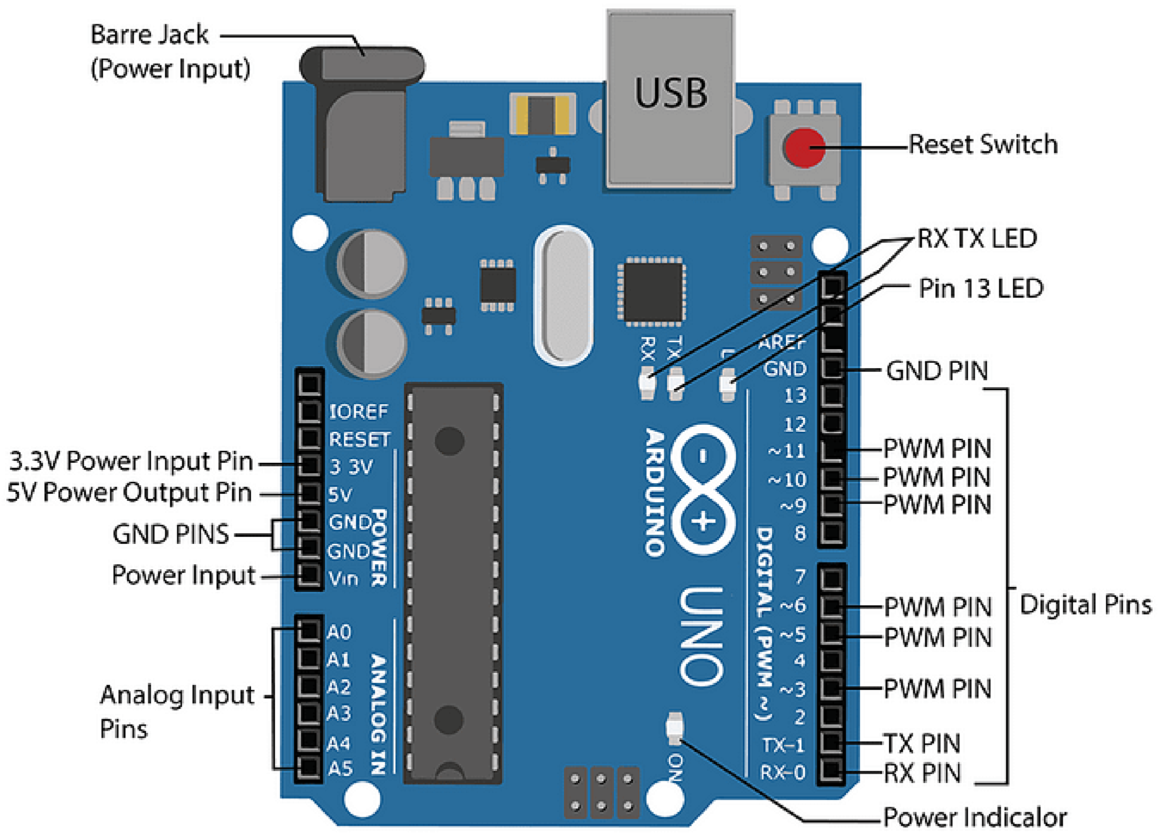


Figure 14 Arduino Uno R3 pin diagram

The Arduino Uno R3 pin diagram is shown. It comprises 14-digit I/O pins. From these pins, 6-pins can be utilized like PWM outputs. This board includes 14 digital input/output pins, Analog inputs-6, a USB connection, quartz crystal-16 MHz, a power jack, resonator-16Mhz, an ICSP header an RST button.

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

1. Serial Pins 0 (Rx) and 1 (Tx) - Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.

2. External Interrupt Pins 2 and 3 - These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

3. PWM Pins 3, 5, 6, 9 and 11 - These pins provide an 8-bit PWM output by using analogWrite() function.

4. SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK) - These pins are used for SPI communication.

5. In-built LED Pin 13 - This pin is connected with a built-in LED, when pin 13 is HIGH — LED is on and when pin 13 is LOW, its off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

1. TWI: A4 or SDA pin and A5 or SCL pin - Support TWI communication using the Wire library.

There are a couple of other pins on the board:

1. AREF. Reference voltage for the analog inputs. Used with analogReference().

2. Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

7.1.5 Power

The Arduino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector. The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

1. Vin. The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

2. 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3. 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA. • GND. Ground pins.

4. IOREF. This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V

7.1.6 PROGRAMMING

To program an Arduino Uno R3, you'll need the Arduino Integrated Development Environment (IDE), connect the board to your computer via USB, and write code in the IDE. After that, you can compile and upload the code to the Arduino.

Steps for Programming Arduino Uno R3:

1. Install the Arduino IDE: Download and install the Arduino IDE from the official Arduino website.

2. Connect the Arduino Uno R3: Connect the board to your computer using a USB cable.

3. Launch the Arduino IDE: Open the IDE software.

4. Select the board: In the IDE, go to "Tools" > "Board" and choose "Arduino Uno".

5. Select the port: Go to "Tools" > "Port" and select the appropriate serial port for your Arduino.

6. Write or open a sketch: You can create a new sketch (program) or open one of the examples provided by the IDE.

7. Compile and upload: Click the "Verify/Compile" button (checkmark) to compile the sketch and then the "Upload" button (right arrow) to upload the code to the Arduino.

8. Monitor the output (optional): You can use the serial monitor to view output from your Arduino.

7.1.7 Other features

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

7.1.8 Price

Arduino Uno R3 is available in the Kenyan market with some prices listed below:

Table 6 Arduino Uno R3 prices

Vendor	**Price (Ksh)**
Jumia Kenya	2280
Nerokas	2000
jiji	2100
Novatech	1600
Sagana electronics	1600
Universal Tech	1800

7.1.9 Advantages

Beginner-Friendly

Very easy to learn and use — ideal for students and hobbyists.

Tons of tutorials, guides, and forums available.

Simple and intuitive programming via the Arduino IDE.

Wide Hardware Compatibility

Compatible with a vast range of sensors, modules, and shields.

Follows the standard Arduino pin layout, allowing for easy hardware expansion (e.g., GPS, LoRa, camera modules, SD cards, etc.).

Extensive Community Support

One of the most supported boards online.

Thousands of open-source libraries and sample codes are available.

You can easily find help for debugging and extending functionality.

Stable and Reliable

Based on the ATmega328P microcontroller, which is proven, stable, and robust.

Simple power regulation circuitry — can be powered via USB or external supply.

Good for Prototyping

Through-hole components make it easy to solder or troubleshoot.

Can be integrated into small prototypes and embedded systems.

Cheap enough to allow for multiple iterations or backups.

Well-Suited for Demonstrations

Great for simulating sensor integration, data logging, and basic satellite subsystems (e.g., onboard control, mock telemetry).

Ideal for your use case of non-flight educational demonstration of a nanosatellite.

Cross-Platform Development

Can be programmed using Windows, macOS, Linux. Supported by other environments like PlatformIO or Atmel Studio for advanced users.

7.1.10 Disadvantages

Limited Processing Power

8-bit AVR architecture (ATmega328P) with a clock speed of 16 MHz.

Not suitable for high-speed data processing, such as real-time video/image processing or running complex algorithms.

Low Memory

Memory Type	Capacity
Flash	32 KB (only ~28 KB usable)
SRAM	2 KB
EEPROM	1 KB

Severely limits data storage, especially for sensor arrays, camera buffers, or trend analysis tasks.

You must offload data quickly to SD cards or external memory modules.

Poor Camera Support

Does not support camera modules directly (e.g., no DVP/MIPI CSI interface).

Can only use low-resolution serial cameras (e.g., OV7670, VC0706) with reduced frame rate and quality.

Cannot process or store large image/video data onboard.

No Native Communication Modules

No built-in Wi-Fi, Bluetooth, or LoRa — must rely on shields or external modules, which add cost, power consumption, and bulk.

Power Inefficient

Not designed for low-power operation, which is critical in satellite systems.

Basic power management options; lacks sleep modes as advanced as ARM-based boards.

Limited Analog and Digital I/O

Only 6 analog inputs and 14 digital I/O pins (6 PWM).

May require multiplexers or expanders for a sensor-rich satellite model.

No Real-Time Operating System (RTOS) Support

Cannot easily implement multitasking, real-time scheduling, or complex system management without custom programming tricks.

No Debugging Interface

No native support for hardware debugging (e.g., breakpoints, stepping through code).

Only basic Serial Monitor debugging available.

7.1.11 RELEVANCE TO SATELLITE PROJECT

While the Uno R3 is perfect for simulating basic satellite subsystems, it's not powerful enough for onboard image processing, advanced telemetry, or real-time control. However, it's a great educational tool for demonstrating the core principles of microcontroller-based control in satellite systems.

7.2 ARDUINO UNO R4

The Arduino UNO R4 is a microcontroller board that represents a significant upgrade from previous UNO revisions. It features a 32-bit ARM Cortex-M4 processor, offering increased performance and memory compared to the ATmega328P used in the UNO R3. The UNO R4 comes in two main versions: the UNO R4 Minima and the UNO R4 WiFi.

- **UNO R4 Minima:** This version focuses on a more basic and lightweight design, while still offering the performance benefits of the 32-bit processor.

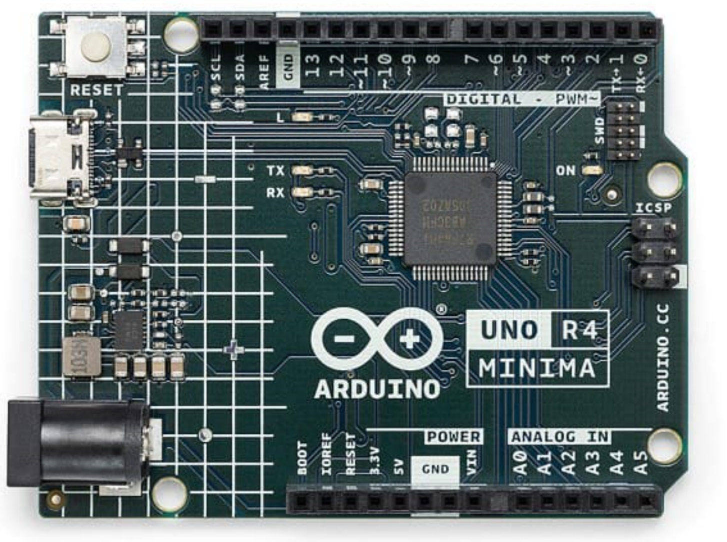


Figure 15 The Arduino UNO R4 MINIMA

- **UNO R4 WiFi**: This version adds Wi-Fi and Bluetooth connectivity, making it suitable for projects requiring network capabilities. It also features a 12x8 LED matrix and a Qwiic connector for easier prototyping and integration.

Figure 16 The Arduino UNO R4 WIFI

7.2.1 Key features

Arduino Uno R4 models compares with Arduino Uno R3 in the following ways:

Processor & Performance

R3: 8-bit ATmega328P @ 16 MHz

R4 Minima & WiFi: 32-bit ARM Cortex-M4 @ 48 MHz (Renesas RA4M1)

Faster, more powerful, supports complex operations and real-time tasks

Memory

R3: 32 KB Flash, 2 KB SRAM

R4s: 256 KB Flash, 32 KB SRAM. 8× more Flash, 16× more RAM — better for logging, buffering, larger programs

Analog Precision

R3: 10-bit ADC

R4s: 14-bit ADC

Higher resolution sensor readings for precise measurements

Built-in Features

Only available in R4 Minima & WiFi:

- ✓ Real-Time Clock (RTC)
- ✓ DAC (Digital-to-Analog Converter)
- ✓ OP AMP
- ✓ CAN Bus

Advanced peripheral support for control and communication systems

Wireless Communication

R3: None built-in

R4 Minima: None built-in

R4 WiFi: Wi-Fi + Bluetooth (via ESP32-S3)

Ideal for telemetry simulation and wireless data transfer

Extra Sensors (R4 WiFi Only)

Built-in 6-axis IMU (accelerometer + gyroscope)

Useful for simulating orientation and motion in nanosatellite models

USB & Power

R3: USB-B, 7–12V input

R4s: USB-C, accepts 6–24V input. Modern, flexible, and easier to power in demos

Debugging Support

R3: No hardware debugging

R4s: SWD debugging available. Better development and troubleshooting experience

Form Factor & Compatibility

All models use the same shield-compatible layout. No change in size or pin headers — smooth upgrade path

7.2.2 Price

Arduino uno R4 price are as follows;

Table 7Arduino uno R4 prices

Vendor	Uno R4 minima	Uno R4 WiFi
Jumia	4990	5827
Nerokas	2200	6000
Pixel electric	5000	6000

7.2.3 Advantages of arduino uno r4 (minima & wifi)

1. Modern 32-bit ARM Cortex-M4 processor provides faster and more efficient performance than the 8-bit R3.
2. Increased memory: 256 KB Flash and 32 KB SRAM allow larger programs, smoother data handling, and better sensor management.
3. Higher ADC resolution (14-bit) improves the precision of analog sensor readings.
4. Built-in Real-Time Clock (RTC), DAC, OP AMP, and CAN Bus offer powerful control and communication features.
5. R4 WiFi includes Wi-Fi, Bluetooth, and a 6-axis IMU — ideal for wireless data transmission and motion simulation.
6. USB-C connection for modern power and data handling.
7. Compatible with existing R3 shields, keeping hardware design flexible.
8. Supports SWD debugging, making development and troubleshooting more professional.

7.2.4 Disadvantages of arduino uno r4 (minima & wifi)

1. More expensive than Uno R3 or Nano, especially the WiFi version.
2. Still lacks native camera support — not suitable for onboard image capture or processing.
3. Wireless communication only available in the WiFi version — Minima has none.
4. Some R3 shields/libraries may not be fully compatible, especially low-level libraries.
5. Slightly steeper learning curve for beginners due to advanced features.
6. Higher power consumption under full load, especially in the R4 WiFi with networking active.

7.2.5 Relevance to satellite project

The Arduino Uno R4 is highly relevant for a nanosatellite bench model due to its increased processing power, enhanced analog capabilities, and built-in peripherals that can simulate various subsystems. It handles multiple sensor inputs with greater precision and supports real-time system behavior using its onboard RTC.

The R4 WiFi model is especially suited for telemetry simulation thanks to built-in wireless communication and the IMU, which can mimic orientation changes. While it still lacks camera support and isn't space-grade, the Uno R4 is an excellent platform for demonstrating satellite-like functionality, processing environmental data, and simulating satellite modes more effectively than the older R3.

8. ARDUINO NANO

Arduino Nano is one type of microcontroller board, and it is designed by Arduino.cc. This microcontroller is based on Atmega168 or Atmega328p. It is fairly similar to Arduino Uno board but when it comes to pin-configuration and features, this nano board has replaced Arduino Uno due to small in size. This microcontroller is also used in Arduino UNO. It is a small size board and also flexible with a wide variety of applications.

8.1 DIAGRAM

Figure 17 Arduino Nano diagram

8.2 FEATURES

- ATmega328P Microcontroller is from 8-bit AVR family
- Operating voltage is 5V
- Input voltage (Vin) is 7V to 12V
- Input/Output Pins are 22
- Analog i/p pins are 6 from A0 to A5
- Digital pins are 14
- Power consumption is 19 mA
- I/O pins DC Current is 40 mA

- Flash memory is 32 KB
- SRAM is 2 KB
- EEPROM is 1 KB
- CLK speed is 16 MHz
- Weight-7g
- Size of the printed circuit board is 18 X 45mm
- Supports three communications like SPI, IIC, & USART

8.3 COMMUNICATION

The communication of an Arduino Nano board can be done using different sources like using an additional Arduino board, a computer, otherwise using microcontrollers. The microcontroller using in Nano board (ATmega328) offers serial communication (UART TTL). This can be accessible at digital pins like TX, and RX. The Arduino software comprises of a serial monitor to allow easy textual information to transmit and receive from the board.

The TX & RX LEDs on the Nano board will blink whenever information is being sent out through the FTDI & USB link in the direction of the computer. The library-like SoftwareSerial allows serial communication on any of the digital pins on the board. The microcontroller also supports SPI & I2C (TWI) communication.

8.4 PIN OUT

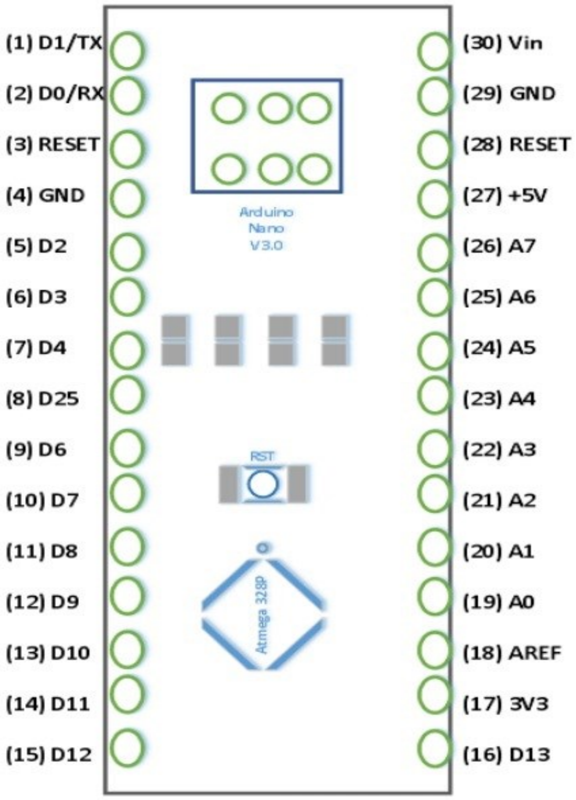


Figure 18 Arduino Nano pin layout

RST Pin(Reset): This pin is used to reset the microcontroller

Analog Pins (A0-A7): These pins are used to calculate the analog voltage of the board within the range of 0V to 5V

I/O Pins (Digital Pins from D0 – D13): These pins are used as an i/p otherwise o/p pins. 0V & 5V

Serial Pins (Tx, Rx): These pins are used to transmit & receive TTL serial data.

External Interrupts (2, 3): These pins are used to activate an interrupt.

PWM (3, 5, 6, 9, 11): These pins are used to provide 8-bit of PWM output.

SPI (10, 11, 12, & 13): These pins are used for supporting SPI communication.

Inbuilt LED (13): This pin is used to activate the LED.

IIC (A4, A5): These pins are used for supporting TWI communication.

AREF: This pin is used to give reference voltage to the input voltage

8.5 POWER

Power Pin (Vin, 3.3V, 5V, GND): These pins are power pins

- Vin is the input voltage of the board, and it is used when an external power source is used from 7V to 12V.
- 5V is the regulated power supply voltage of the nano board and it is used to give the supply to the board as well as components.
- 3.3V is the minimum voltage which is generated from the voltage regulator on the board.
- GND is the ground pin of the board

8.6 PROGRAMMING

The programming of an Arduino nano can be done using the Arduino software. Click the Tools option and select the nano board. Microcontroller ATmega328 over the Nano board comes with preprogrammed with a boot loader. This boot loader lets to upload new code without using an exterior hardware programmer. The communication of this can be done with the STK500 protocol. Here the boot loader can also be avoided & the microcontroller program can be done using the header of in-circuit serial programming or ICSP with an Arduino ISP.

8.7 COMPARISON BETWEEN ARDUINO UNO AND ARDUINO NANO

The Arduino Nano board is similar to an Arduino UNO board including similar microcontroller like Atmega328p. Thus they can share a similar program. The main difference between these two is the size. Because Arduino Uno size is double to nano board. So Uno boards use more space on the system. The programming of UNO can be done with a USB cable whereas Nano uses the mini USB cable. A comparison between Arduino Uno and Nano is shown below.

Table 8 comparison between Arduino Uno and Nano

Specifications	Arduino Uno	Arduino Nano
Processor	ATmega328P	ATmega328P
Input Voltage	5V / 7-12V	5V / 7-12V
Speed of CPU	16 MHz	16 MHz
Analog I/O	6 / 0	8 / 0
Digital IO/PWM	14 / 6	14 / 6
EEPROM / SRAM [kB]	1 / 2	1 / 2
Flash	32	32
USB	Regular	Mini
USART	1	1

8.8 PRICE

Arduino Nano is available in the Kenyan market with some prices listed below:

Table 9 Arduino Nano prices

Vendor	Price (Ksh)
Jumia Kenya	1800
Nerokas	1300
Pixel electric	1000
Novatech	1200
Sagana electronics	1200

8.9 ADVANTAGES

Compact Size

Extremely small and lightweight — ideal for projects where space is limited (e.g., nanosatellite enclosures).

Breadboard-friendly for prototyping and circuit testing.

Affordable

Often cheaper than Uno R3 and R4 versions.

Widely available in Kenya and globally.

Same Core as Uno R3

Uses the same ATmega328P microcontroller as the Uno R3.

Same programming language, libraries, and pin functions.

USB Power + Easy Programming

Powered and programmed via Mini USB or Micro USB (depending on version).

Can be programmed via Arduino IDE with no special tools required.

Sufficient I/O for Simple Systems

22 I/O pins (14 digital + 8 analog inputs) — more analog pins than Uno.

Supports PWM, SPI, I2C, and UART for sensor communication.

Lower Power Consumption

Lower idle current compared to Uno, especially in sleep mode.

Ideal for battery-powered or solar-based systems in demo setups.

Compatible with Most Arduino Libraries

No need to rewrite code from Uno to Nano — portable across projects.

Most example sketches and sensors work out of the box.

Great for Basic Satellite Subsystem Demos

Perfect for simulating:

- ✓ Sensor data collection (soil moisture, temp, gas sensors)

- ✓ Basic data logging to SD card
- ✓ LED status indicators
- ✓ Communication protocols like I2C/SPI

8.10 DISADVANTAGES

Limited Processing Power

Same 8-bit ATmega328P as the Uno R3, running at 16 MHz.

Cannot handle complex computations like real-time image processing or multitasking.

Low Memory

Type	Size
Flash	32 KB (approx. 28 KB usable)
SRAM	2 KB
EEPROM	1 KB

Too limited for large data buffers, image storage, or trend analysis over time.

Requires frequent data offloading (e.g., to SD card).

USB Port Fragility

Uses Mini USB or Micro USB, which is more fragile and less durable than USB-B or USB-C.

USB connectors may loosen or break with repeated use.

No Native Debugging Interface

Just like the Uno R3, Nano lacks hardware debugging features (e.g., breakpoints, stepping through code).

Relies on Serial Monitor for debugging, which can be limiting for real-time systems.

No Native Wireless

Does not include Wi-Fi, Bluetooth, or LoRa modules.

Must be paired with external modules (e.g., ESP8266, HC-05, LoRa), which takes up space and power.

Basic Power Management

Power-efficient at idle but lacks advanced low-power modes found in ARM-based boards.

No built-in battery charging circuitry (must use external modules for LiPo power).

No Mounting Holes

Most Nano boards lack mounting holes — harder to fix in enclosures unless you create custom brackets.

Slight Pin Confusion

Some I/O pins are shared or multifunctional (e.g., D0/D1 used for UART).

Care must be taken not to conflict pin functions when using multiple peripherals.

8.11 RELEVANCE TO SATELLITE PROJECT

The Arduino Nano is well-suited for simulating basic satellite subsystems in an educational or bench-model context. While it lacks native camera integration and is not capable of processing or storing image data, it can effectively simulate sensor readings and system responses. Wireless communication is not built-in, but can be achieved using external modules like LoRa, ESP8266, or Bluetooth. Data storage is limited by onboard memory, but logging to an external SD card module is possible for sensor outputs. Real-time control is constrained by the limited processing speed and memory, making it unsuitable for multitasking or autonomous decisions, but ideal for scripted sensor-response demonstrations. Its small size and simplicity make it excellent for educational use, quick prototyping, and demonstrating satellite-like behavior in a classroom or exhibition setting.

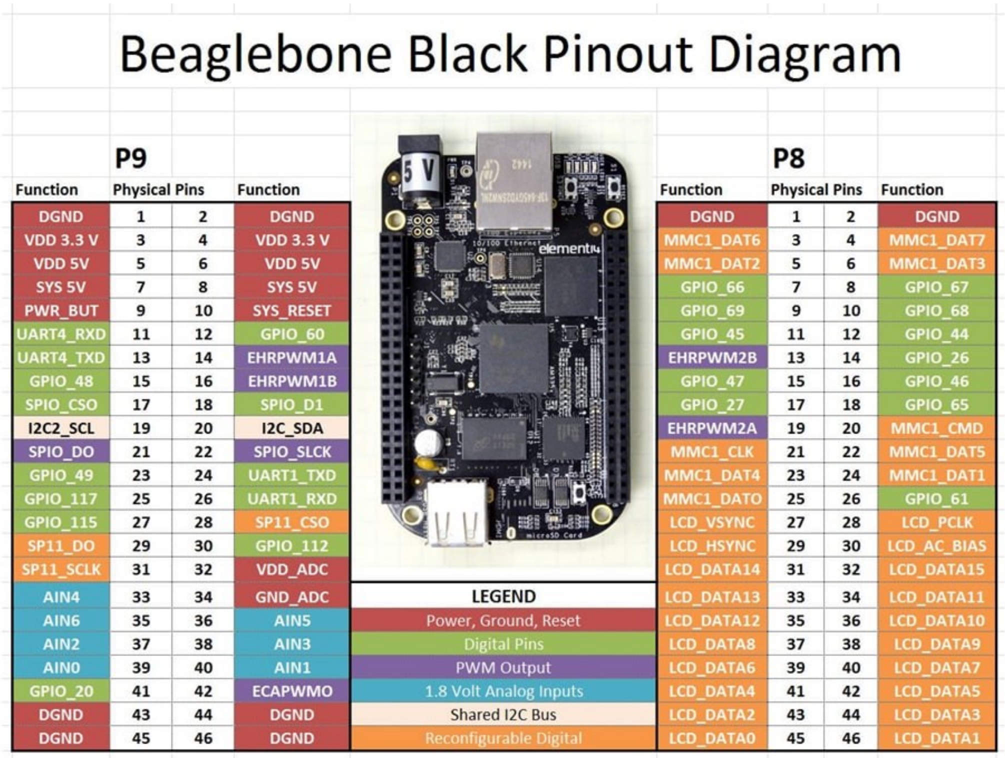
9. BEAGLEBONE BLACK

Beagle Bone Black has been introduced as a low-cost developer platform supporting the ARM-A8 microprocessor development architecture. This mini-computer enables Linux to boot in under 10 seconds and boots on the Sitara AM335x ARM Cortex-A8 processor for developers in under 5 minutes with just a single USB cable.

In terms of specifications, this 8.6 x 5.3 cm motherboard is equipped with a 1GHz ARM Cortex-A8 processor (made by Texas Instruments) supporting graphics acceleration, a 46-pin connector for peripheral connections, 2GB of integrated flash memory and 512MB of DDR3 memory. The BeagleBone Black is also equipped with USB ports, Ethernet connections and HDMI supporting a resolution of 1280 x 1024 pixels.

BeagleBone Black uses a 5V DC power input that can be supplied from the USB port (USB cable included) or connected to adapters with the corresponding voltage level.

9.1 PIN LAYOUT



P9

Function	Physical Pins		Function
DGND	1	2	DGND
VDD 3.3 V	3	4	VDD 3.3 V
VDD 5V	5	6	VDD 5V
SYS 5V	7	8	SYS 5V
PWR_BUT	9	10	SYS_RESET
UART4_RXD	11	12	GPIO_60
UART4_TXD	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
SPIO_CSO	17	18	SPIO_D1
I2C2_SCL	19	20	I2C_SDA
SPIO_DO	21	22	SPIO_SLCK
GPIO_49	23	24	UART1_TXD
GPIO_117	25	26	UART1_RXD
GPIO_115	27	28	SP11_CSO
SP11_DO	29	30	GPIO_112
SP11_SCLK	31	32	VDD_ADC
AIN4	33	34	GND_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPWMO
DGND	43	44	DGND
DGND	45	46	DGND

P8

Function	Physical Pins		Function
DGND	1	2	DGND
MMC1_DAT6	3	4	MMC1_DAT7
MMC1_DAT2	5	6	MMC1_DAT3
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
EHRPWM2B	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
EHRPWM2A	19	20	MMC1_CMD
MMC1_CLK	21	22	MMC1_DAT5
MMC1_DAT4	23	24	MMC1_DAT1
MMC1_DATO	25	26	GPIO_61
LCD_VSYNC	27	28	LCD_PCLK
LCD_HSYNC	29	30	LCD_AC_BIAS
LCD_DATA14	31	32	LCD_DATA15
LCD_DATA13	33	34	LCD_DATA11
LCD_DATA12	35	36	LCD_DATA10
LCD_DATA8	37	38	LCD_DATA9
LCD_DATA6	39	40	LCD_DATA7
LCD_DATA4	41	42	LCD_DATA5
LCD_DATA2	43	44	LCD_DATA3
LCD_DATA0	45	46	LCD_DATA1

Figure 19 BeagleBone Black pinout diagram

9.2 DIAGRAM

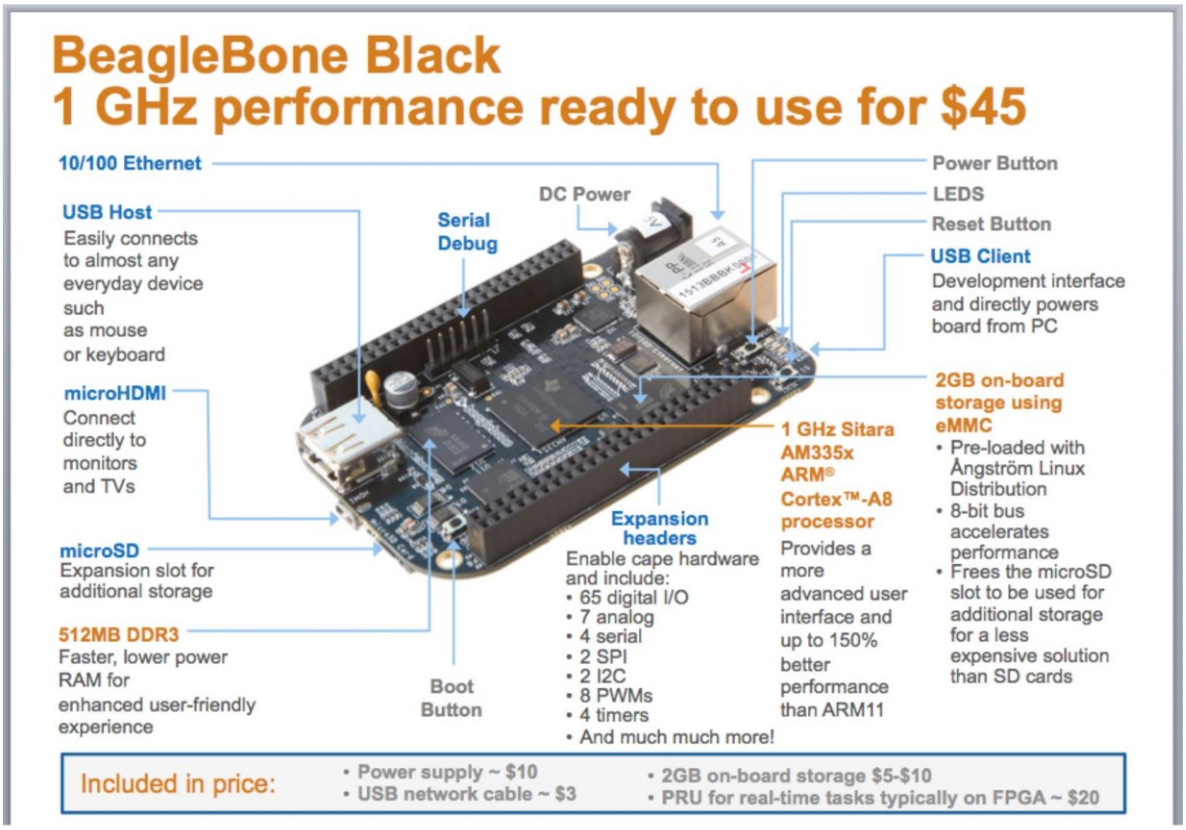


Figure 20 BeagleBone Black diagram

9.3 FEATURES

Table 10 BeagleBone Black features

Specification	Details
Processor	AM335x 1GHz ARM Cortex-A8
RAM	512MB DDR3
Storage	4GB eMMC on-board, microSD card support
Operating System	Debian Linux (default), supports custom OS
Connectivity	USB, Ethernet, HDMI, UART, SPI, I2C, GPIO
Power Consumption	2.5W typical
I/O Pins	2x46 headers (GPIO, PWM, ADC, SPI, etc.)

PROS & CONS

Table 11 BeagleBone Black pros and cons

Pros	Cons
High processing power	Not radiation-hardened (COTS component)
Supports full Linux OS	Higher power consumption
Multiple I/O interfaces	Complex for small subsystems
Excellent for payload processing or AI onboard	Requires shielding in space

9.4 COMMUNICATION

The BeagleBone Black supports multiple communication interfaces essential for nanosatellite systems. It includes UART (serial) ports for GPS or telemetry modules, I²C and SPI buses for connecting to sensors and other peripherals, and USB (host and device modes) for data transfer or peripheral connections like flash drives. It also features a 10/100 Ethernet port for high-speed networking and supports CAN bus for robust industrial communication. These interfaces are accessible through its P8 and P9 header pins, making the board highly versatile for various satellite subsystems.

9.5 PROGRAMMING

The BeagleBone Black (BBB) supports flexible and powerful programming options thanks to its built-in Linux operating system (usually Debian or Ubuntu). Users can write code directly on the board using high-level languages like Python, C, C++, JavaScript (Node.js), or even Shell scripting. It comes with Cloud9 IDE pre-installed, allowing users to code in a web-based environment through a browser by simply connecting the BBB via USB.

For hardware control, programmers can use libraries like Adafruit BBIO or libpruio to manage GPIO, PWM, UART, I²C, and SPI interfaces. It also supports the BoneScript library, similar to JavaScript, for easy control of hardware pins. Additionally, developers can write low-level code for PRUs (Programmable Real-time Units) to achieve real-time performance where needed. Programming is typically done via SSH access, remote desktop, or USB mass storage, making development highly accessible.

This versatility makes the BeagleBone Black an excellent platform for both real-time embedded control and high-level application development in nanosatellite systems

9.6 PRICE

1. Nerokas- ksh. 8000
2. Jumia- Ksh. 13,924

9.7 APPLICATIONS OF BEAGLEBONE BLACK

Used as an embedded computer in industry

- With low cost, easy to connect to peripheral devices, meeting industrial standards,
- Works well in temperature range from -40C to +85C, suitable for 3D printer or CNC projects

Do Robot projects

9.8 RECOMMENDED USE:

- Payload data processing
- Image or signal analysis
- Backup On-Board Computer (OBC)

9.9 RELEVANCE OF THE BEAGLE BONE BLACK TO NANOSAT PROJECT

The Beagle Bone Black is highly relevant to nanosatellite projects due to its powerful processing capabilities and rich I/O features. It is equipped with a 1GHz ARM Cortex-A8 processor and 512MB of RAM, making it suitable for data-intensive tasks like image processing, attitude control algorithms, and real-time data analysis. With onboard USB, Ethernet, HDMI, and GPIOs, it supports multiple sensor interfaces, external storage, and communication modules (e.g., GSM, GPS, RF). Its support for Linux (Ubuntu, Debian, etc.) allows for high-level programming, multitasking, and easier integration of open-source tools. The board's expandability via capes and its ability to run real-time applications using PRUs (Programmable Real-time Units) make it ideal for handling autonomous control systems within a satellite. Despite its relatively high power consumption, the Beagle Bone Black is perfect for ground simulation, payload control, or onboard data handling, where complex processing and flexibility are more important than ultra-low power operation.

10. PIC16F877A MICROCONTROLLER

10.1 PINOUT

- 40-pin DIP (Dual Inline Package)
- 5 Ports:
  - PORTA (6-bit), PORTB–PORTD (8-bit), PORTE (3-bit)
- Pins support analog, digital, and interrupt-on-change

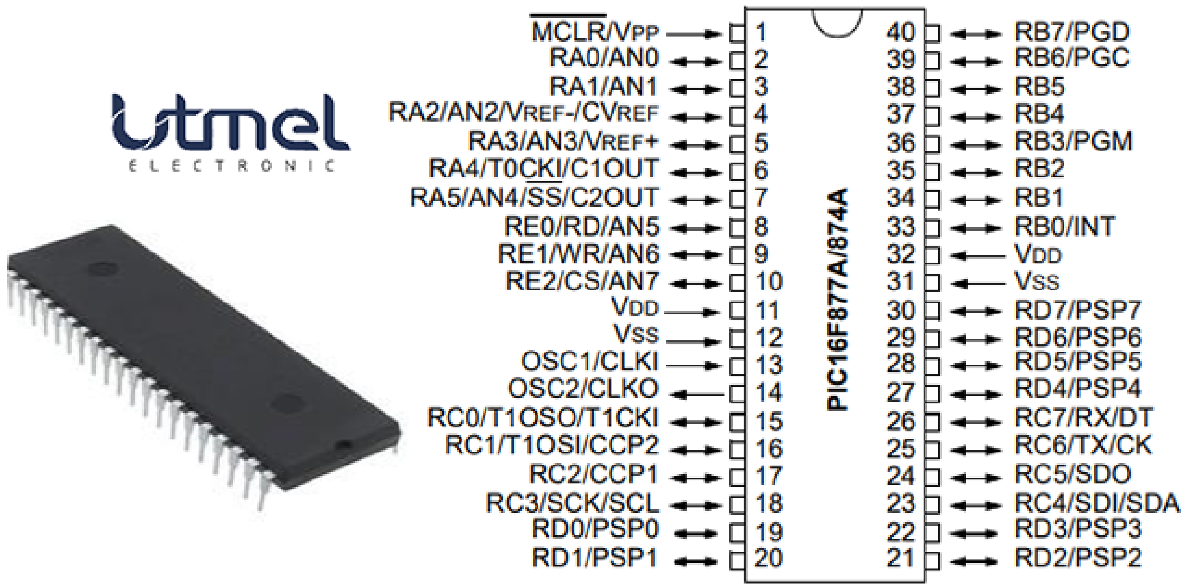


Figure 21 PIC16F877A PIN OUT DIAGRAM

Table 12 PIC16F877A PIN OUT

Pin Number	Pin Name	Description
1	MCLR/Vpp	MCLR is used during programming, mostly connected to programmers like PicKit
2	RA0/AN0	Analog pin 0 or 0 th pin of PORTA
3	RA1/AN1	Analog pin 1 or 1 st pin of PORTA
4	RA2/AN2/Vref-	Analog pin 2 or 2 nd pin of PORTA
5	RA3/AN3/Vref+	Analog pin 3 or 3 rd pin of PORTA

6	RA4/T0CKI/C1out	4 th pin of PORTA
7	RA5/AN4/SS/C2out	Analog pin 4 or 5 th pin of PORTA
8	RE0/RD/AN5	Analog pin 5 or 0 th pin of PORTE
9	RE1/WR/AN6	Analog pin 6 or 1 st pin of PORTE
10	RE2/CS/AN7	7 th pin of PORTE
11	Vdd	Positive supply for logic and /O pins
12	Vss	Ground reference for logic and VO pins
13	OSC1/CLKI	External Oscillator /clock input pin
14	OSC2/CLKO	External Oscillator /clock output pin
15	RC0/T1OSO/T1CKI	0 th pin of PORT C
16	RC1/T1OSI/CCP2	1 st pin of POCTC or Timer/PWM pin
17	RC2/CCP1	2 nd pin of POCTC or Timer/PWM pin
18	RC3/SCK/SCL	3 rd pin of POCTC
19	RD0/PSP0	0 th pin of POCTD
20	RD1/PSPI	1 st pin of POCTD
21	RD2/PSP2	2 nd pin of POCTD
22	RD3/PSP3	3 rd pin of POCTD
23	RC4/SDI/SDA	4 th pin of POCTC or Serial Data in pin
24	RC5/SDO	5 th pin of POCTC or Serial Data Out pin
25	RC6/Tx/CK	6 th pin of POCTC or Transmitter pin of Microcontroller
26	RC7/Rx/DT	7 th pin of POCTC or Receiver pin of Microcontroller
27	RD4/PSP4	4 th pin of POCTD
28	RD5/PSP5	5 th pin of POCTD

29	RD6/PSP6	6 th pin of POCTD
30	RD7/PSP7	7 th pin of POCTD
31	Vss	Ground reference for logic and VO pins
32	Vdd	Positive supply for logic and /O pins
33	RB0/INT	0 th pin of POCTB or External Interrupt pin
34	RB1	1 st pin of POCTB
35	RB2	2 nd pin of POCTB
36	RB3/PGM	3 rd pin of POCTB or connected to programmer
37	RB4	4 th pin of POCTB
38	RB5	5 th pin of POCTB
39	RB6/PGC	6 th pin of POCTB or connected to programmer
40	RB7/PGD	7 th pin of POCTB or connected to programmer

10.2 FEATURES

- Architecture: 8-bit RISC
- Clock Speed: Up to 20 MHz
- Program Memory: 14 KB Flash
- RAM: 368 Bytes
- EEPROM: 256 Bytes
- Operating Voltage: 2V – 5.5V
- I/O Pins: 33 (5 ports: A–E)
- Timers: 3 timers (Timer0, Timer1, Timer2)
- Watchdog Timer: Built-in for safety
- Interrupts: 15 total interrupt sources

10.3 COMMUNICATION

The PIC16F877A microcontroller supports essential communication protocols that are useful in nanosatellite subsystems. It features a USART (Universal Synchronous/Asynchronous Receiver/Transmitter) module for serial communication, often used with devices like GPS modules, RS-232 interfaces, and telemetry units. The serial transmit pin is RC6/TX (pin 25) and the receive pin is RC7/RX (pin 26). For high-speed sensor communication, it includes an SPI (Serial Peripheral Interface) that uses RC3/SCK (pin 18) for the clock signal, RC4/SDI (pin 23) for input data (MISO), and RC5/SDO (pin 24) for output data (MOSI). A separate digital I/O pin such as RA5 (pin 7) is commonly used as the Chip Select (CS) line for SPI. The I^2C (Inter-Integrated Circuit) interface supports communication with multiple slower peripherals using just two lines: RC3/SCL (pin 18) for the clock and RC4/SDA (pin 23) for data. This is ideal for connecting to temperature sensors, EEPROMs, or real-time clocks. Additionally, the PIC16F877A includes a Parallel Slave Port (using PORTD: pins RD0 to RD7, pins 19–22 and 27–30) which enables 8-bit parallel data transfer, useful in applications requiring faster throughput or wide data buses. Unlike the PIC18F4550, the PIC16F877A does not have built-in USB or Ethernet, so such features would require external interface modules.

10.4 POWER

- Operating Voltage: 2 – 5.5V
- Low power consumption
- Sleep Mode: For power saving
- Can run from battery or regulated DC supply

10.5 PROGRAMMING

- Languages: Assembly, Embedded C
- Programmer Required: e.g., PICkit 2 or 3
- Software:
  - MPLAB X IDE (Microchip)
  - MPLAB XC8 Compiler
- No operating system (bare-metal programming)

10.6 OTHER FEATURES

- ADC: 10-bit, 8 channels (for sensors)
- PWM: For controlling motors or servos
- Comparator Module: Analog signal comparison
- EEPROM: Non-volatile memory for saving data during power loss
- Timers with Prescalers: Good for real-time control

10.7 PRICE

- Approximate Cost: $3 – $6 USD (KES 400 – 800)

10.8 ADVANTAGES

- Cheap and readily available
- Easy to learn and use
- Great for real-time control applications
- Low power and space-efficient
- Excellent for simple I/O management and sensor interfacing

10.9 DISADVANTAGES

- No USB or native Ethernet
- No OS support
- Low memory and processing power
- Not suitable for complex tasks like image processing or multitasking

10.10 PIC16F877 APPLICATIONS

- Multiple DIY Projects
- Very good choice if you are learning PIC
- Projects requiring Multiple I/O interfaces and communications
- Replacement for Arduino Module

- Ideal for more advanced level A/D applications in automotive, industrial, appliances, and consumer applications.

10.11 RELEVANCE TO NANOSATELLITE PROJECT

- **Ideal For**:
    - Sensor interfacing (temperature, magnetometer, gyroscope)
    - Real-time control of actuators (reaction wheels, antenna motors)
    - Power management tasks (battery switching, regulation)
    - Communication with higher-level controller (like BeagleBone)
- Excellent choice for subsystems needing low power and high reliability

11. PIC18F4550 MICROCONTROLLER

11.1 PINOUT

- 40-pin Dual Inline Package (DIP) or 44-pin QFN/TQFP
- PORTA–PORTE:
    - Configurable as analog or digital I/O
    - USB uses specific pins (D+ and D− on RC4 and RC5)
- Each pin can source/sink 25 mA (good for LEDs/small loads)

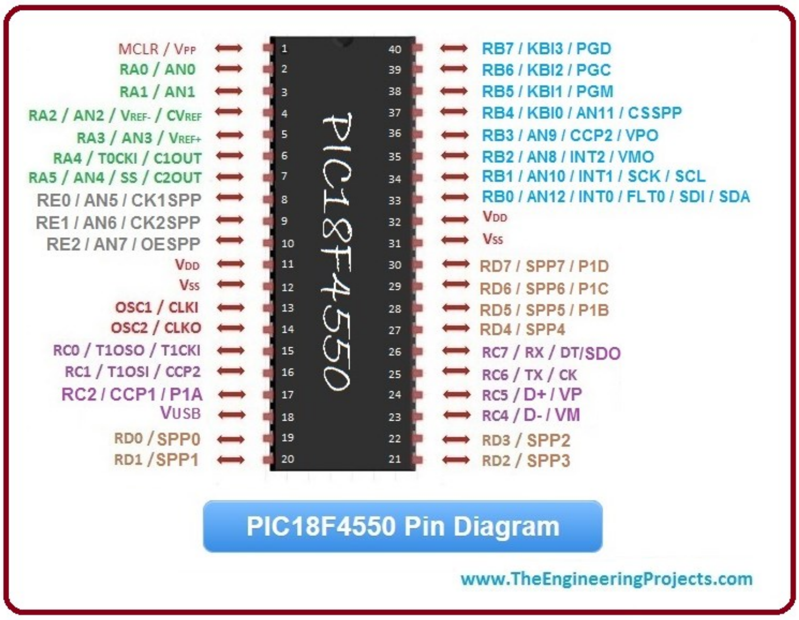


Figure 22 PIC18F4550 pin diagram

11.2 FEATURES

- Architecture: 8-bit RISC
- Clock Speed: Up to 48 MHz
- Program Memory: 32 KB Flash
- RAM: 2 KB SRAM
- EEPROM: 256 Bytes
- Operating Voltage: 2.0V – 5.5V
- I/O Pins: 35 (on 40-pin DIP)

- Timers: 4 timers (Timer0–Timer3)
- Watchdog Timer: Present
- Oscillator Options: Internal and external oscillator support

11.3 COMMUNICATION

The PIC18F4550 microcontroller supports multiple communication protocols essential for nanosatellite systems. A standout feature is its USB 2.0 interface (Full-Speed at 12 Mbps), enabling device or host modes via RC4 (pin 23, USB D–) and RC5 (pin 24, USB D+). This allows direct PC connection for programming or data transfer to USB peripherals—useful in ground testing and diagnostics.

For serial communication, it offers a USART module using RC6/TX (pin 25) and RC7/RX (pin 26), ideal for connecting GPS, GSM, or telemetry modules. It also includes a high-speed SPI interface with RC3/SCK (pin 18), RC4/SDI (pin 23, MISO), and RC5/SDO (pin 24, MOSI), plus a user-defined Chip Select (e.g., RA5/pin 7)—commonly used for sensors, flash memory, or SD cards.

For low-speed peripherals, the I^2C bus uses RC3/SCL (pin 18) and RC4/SDA (pin 23), enabling multiple device communication over just two lines, ideal for EEPROMs, gyroscopes, or RTCs. While it lacks built-in Ethernet, its USB support compensates well. Altogether, these communication options make the PIC18F4550 highly adaptable for modular nanosatellite subsystems.

11.4 POWER

The PIC18F4550 operates within a voltage range of 2.0V to 5.5V, making it flexible for various power sources, including direct USB-powered operation at 5V. It features power-saving modes such as Sleep, which shuts down most internal functions to conserve energy, and Idle, useful for USB suspend mode during low activity. These modes help reduce power consumption, which is critical for energy-limited systems like nanosatellites, ensuring longer operational life and thermal efficiency in space environments.

11.5 PROGRAMMING

- Languages: Assembly, Embedded C
- Tools:
    - MPLAB X IDE

  - XC8 Compiler
  - USB bootloader (no external programmer needed) or PICkit 2/3
- Supports In-Circuit Serial Programming (ICSP)

11.6 OTHER FEATURES

- 10-bit ADC (13 channels)
- PWM outputs
- USB support (Device/Host mode)
- Enhanced PWM (ECCP) Module
- Comparator & Voltage Reference Modules
- Enhanced Flash endurance – 100k erase/write cycles

11.7 PRICE

- Approximate Cost: $4 – $8 USD (KES 500 – 1000)

11.8 ADVANTAGES

- USB support for data transfer and programming
- Large Flash and RAM compared to PIC16 series
- Suitable for medium-complexity tasks
- Built-in ADCs for sensors
- Multiple communication protocols
- Can act as a USB device or host

11.9 DISADVANTAGES

- Still 8-bit, so limited processing power
- Not ideal for multitasking or heavy data processing
- No built-in Ethernet or wireless communication
- Higher power than ultra-low-power MCUs

11.10 RELEVANCE TO NANOSATELLITE PROJECT

- **Ideal For**:
  - USB interfacing with ground station or data loggers
  - Command and data handling subsystems
  - Interfacing sensors (thermal, magnetic, etc.)
  - Mid-level task control
  - Communication with Beagle Bone (via serial or USB)
- Better than PIC16F877A for subsystems requiring USB, more memory, and faster response

11.11 CONCLUSION

BeagleBone Black, PIC16F877A, and PIC18F4550—each offer unique advantages that are valuable to nanosatellite systems. The BeagleBone Black provides powerful processing and Linux-based flexibility, making it suitable for tasks that require complex computation, such as payload data handling or onboard processing. The PIC18F4550, with its USB support and multiple communication protocols (USART, SPI, I²C), is ideal for peripheral interfacing and efficient data transfer. Meanwhile, the PIC16F877A, known for its simplicity, low power consumption, and robust peripheral features, fits well in control and monitoring tasks. Collectively, these microcontrollers demonstrate how different hardware platforms can be strategically used to address various functional needs in a nanosatellite, from real-time control to communication and data processing.

12. RASPBERRY PI PICO

The remarkable reduction in the cost and complexity of CubeSat and Nanosatellite development has effectively lowered the barrier to entry for space exploration, enabling a broader spectrum of participants, such as universities, research centres, and even individual enthusiasts, to design and launch their own satellite missions. This fundamental shift in the space paradigm directly influences the viability of employing Commercial Off-The-Shelf (COTS) components, which were traditionally deemed unsuitable for the rigorous demands of space environments.

At the core of any CubeSat or nanosatellite mission lies the On-Board Computer (OBC), serving as the central intelligence that monitors and controls all satellite operations. The OBC is tasked with critical functions including receiving, validating, and distributing commands to various subsystems, detecting and recovering from anomalies—particularly those induced by space radiation—maintaining precise spacecraft timekeeping, and processing data collected by payloads.

This report undertakes a comprehensive, data-driven comparative analysis of two widely adopted microcontrollers: the Raspberry Pi Pico and the STM32F103C8T6. The evaluation focuses on their technical specifications, economic advantages, and their inherent suitability and limitations when subjected to the harsh conditions of the space environment.

12.1 PIN LAYOUT

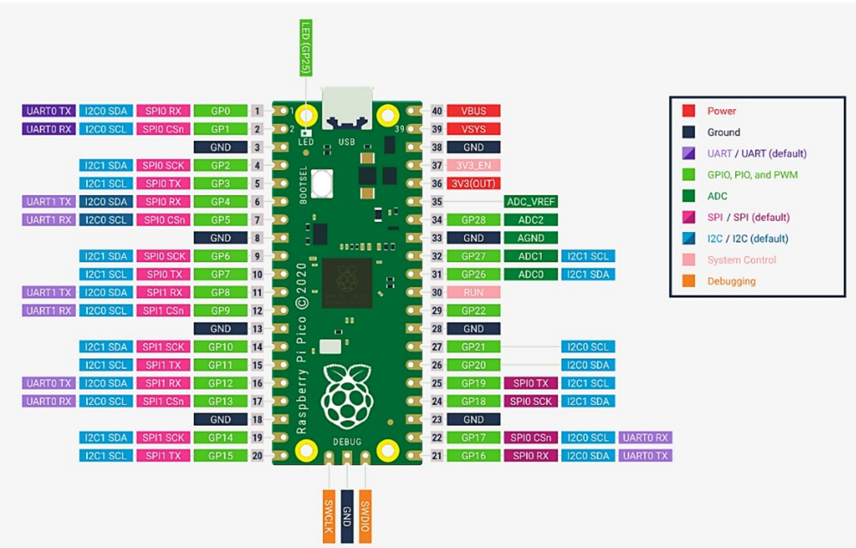


Figure 23 The pinout for a Raspberry Pi Pico Rev3 Board

12.2 PERFOMANCE AND TECHNICAL SPECIFICATIONS

The original Raspberry Pi Pico, built around the RP2040 silicon platform, features a dual-core Arm Cortex-M0+ processor operating at a clock speed of 133MHz.This symmetric dual-core architecture provides a solid foundation for various embedded tasks. The newer Raspberry Pi Pico 2 and Pico 2W, which are based on the more advanced RP2350 microcontroller. These variants deliver a significant performance upgrade, incorporating either dual Arm Cortex-M33 cores or dual open-hardware Hazard3 RISC-V cores, both clocked at 150MHz. The Arm Cortex-M33 core, notably, includes a Floating-Point Unit (FPU) and has been observed to be 2x as fast as the M0+ of the RP2040, enhancing its capacity for more complex mathematical computations. This flexibility allows for initial prototyping and development on the more economical original Pico, with the option to transition to a higher-performance variant as mission requirements evolve, thereby reducing the need for a complete redesign and mitigating development risk.

On memory, the original RP2040 is equipped with 264KB of on-chip SRAM and 2MB of on-board QSPI Flash Memory. The RP2350 doubles this to 520KB of SRAM and 4MB of QSPI Flash Memory. The notable increase in flash memory is good in applications using CircuitPython or MicroPython as it provides more space for storing files and application code.

Both the RP2040 and RP2350 have rich peripheral handling capabilities for various applications. Both have 26 General Purpose Input-Output (GPIO) pins (3-Analogue inputs-RP2040, 4 Analogue Inputs- RP2350), 2 UARTs, 2 SPI controllers, 2 I^2C Controllers and USB 1.1. The RP2040 has 16PWM channels and 8 PIO (Programmable Input/Output) State machines. Whereas, the RP2350 has 24PWM channels and 12 PIO state machines. The Pico 2W variant specifically integrates single-band 2.4GHz wireless interfaces (802.11n) using the Infineon CYW43439 chip, connected via SPI to the RP2350, offering Wi-Fi and Bluetooth Low Energy (BLE) capabilities.

The power consumption of the RP2040 in the active mode is ~90mA (at 133MHz). In Idle or Sleep Mode the current draw is in the micro-ampere range (specific modes can go down to ~100uA)

13. STM32F103C8T6

13.1 PIN LAYOUT

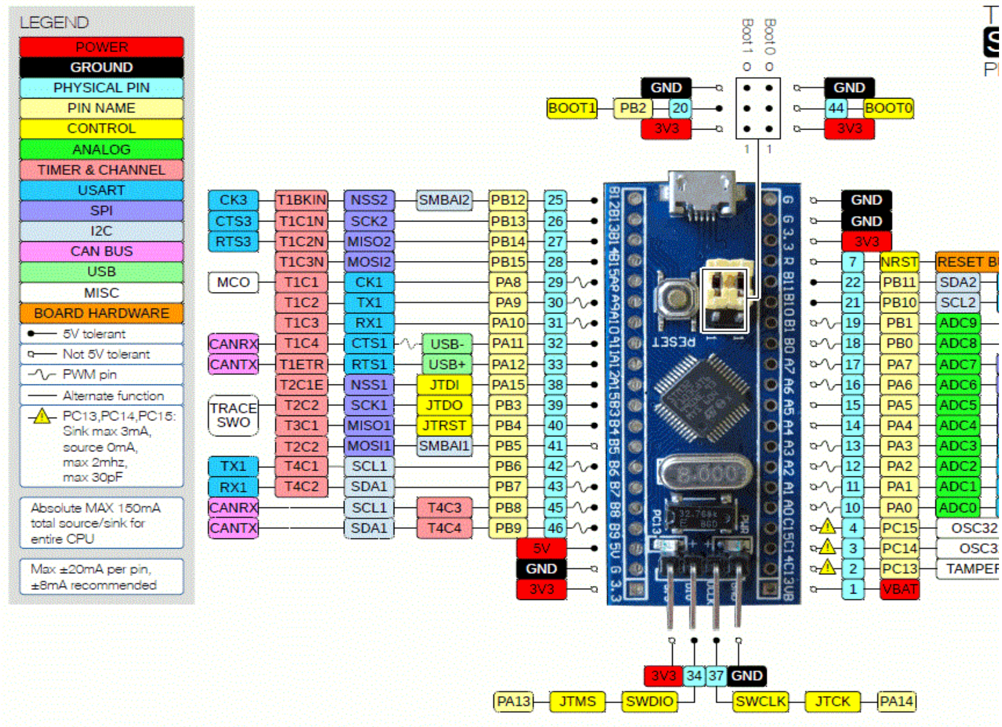


Figure 24 STM32F103C8T6 (Bluepill) pinout

13.2 FEATURES

The STM32F103C8T6, a prominent member of STMicroelectronics' STM32F103xx medium-density performance line. It incorporates a single-core Arm Cortex-M3 32-bit RISC core. This CPU operates at a maximum frequency of 72MHz, delivering a performance of 1.25 DMIPS/MHz (Dhrystone 2.1) with zero wait state memory access. The core also features single-cycle multiplication and hardware division, which enhances its efficiency for various embedded control applications.

For memory, the STM32F103C8T6 includes 64KB of Flash memory, with some variants in the broader STM32F103xx family offering up to 128KB, and 20KB of SRAM.

The peripheral set and I/O capabilities of the STM32F103C8T6 are more compared to the Raspberry Pi Pico. It has 37 general-purpose I/O pins. These I/Os are highly flexible, being mappable on 16 external interrupt vectors, and notably, almost all are 5V-tolerant, which offers broad compatibility with a wide array of external sensors and actuators.

Communication interfaces are comprehensive, including CANbus, 2 I2C interfaces, IrDA, LINbus, 2 SPI interfaces,3 USARTs and a USB 2.0 full-speed interface. Additional integrated peripherals include a 7-channel DMA controller (supporting timers, ADC, SPIs, I2Cs, and

USARTs), motor control PWM, Power-On Reset (POR), Power-Down Reset (PDR), Programmable Voltage Detector (PVD), and a robust set of seven timers. These timers comprise three 16-bit general-purpose timers, one 16-bit motor control PWM timer with dead-time generation and emergency stop, two watchdog timers (independent and window), and a 24-bit SysTick timer. The microcontroller also integrates two 12-bit, 1µs Analog-to-Digital Converters (ADCs) with up to 16 channels, a 0-3.6V conversion range, dual-sample and hold capability, and an integrated temperature sensor.

The power consumption/current draw of the STM32F103C8T6 is ~36mA (at 72MHz) in the active mode, ~24mA in Run Mode, ~5mA in Sleep Mode, ~50uA in Stop Mode and ~2uA in Standby Mode

13.3 COMPARISON BETWEEN RPI PICO AND STM32F103C8T6

Table 13 Comparative Technical Specifications (Raspberry Pi Pico vs. STM32F103C8T6)

Feature	Raspberry Pi Pico (RP2040)	Raspberry Pi Pico 2/2W (RP2350)	STM32F103C8T6
CPU Core	Dual-core Arm Cortex-M0+	Dual Arm Cortex-M33 or RISC-V	Single-core Arm Cortex-M3
Clock Speed (MHz)	133	150	72
FPU	No	Yes (Cortex-M33)	No
On-chip SRAM (KB)	264	520	20
On-board Flash (MB)	2	4	0.064 (64KB)
Total GPIO Pins	26	26	37
Analog Inputs	3	4	10 (12-bit ADC)
UART Controllers	2	2	3
SPI Controllers	2	2	2
I2C Controllers	2	2	2
PWM Channels	16	24	7 (general-purpose, motor control)
USB Controller	1.1 (Host/Device)	1.1 (Host/Device)	2.0 Full-Speed

CAN Interface	No	No	Yes (2.0B Active)
PIO State Machines	8 (2 blocks)	12 (3 blocks)	No
Wireless (Wi-Fi/BLE)	No	Yes (Pico 2W only)	No
Input Voltage Range (V DC)	1.8–5.5	1.8–5.5	2.0–3.6
Operating Temp. Range (°C)	-20 to +85	-20 to +85	-40 to +85 (up to +105 for family)
Dimensions (LxWxH, mm)	51 x 21 x 3.9	51 x 21 x 3.9	54 x 22 (board)
Weight (g)	3	4	9 (board)

ECONOMIC VIABILITY

Both the Raspberry Pi Pico and STM32F103C8T6 fall into the Commercial Off-The-Shelf category with remarkably similar pricing. The Raspberry Pi family offers several options within a narrow price range, with the original Pico costing between Ksh 520 and 1,040, the newer Pico 2 priced at Ksh 650, and the Wi-Fi enabled Pico W available for Ksh 780. The STM32F103C8T6 presents comparable costs, with the standalone microcontroller chip priced at Ksh 665 and the popular Blue Pill development boards ranging from Ksh 1,040 to 1,235.

This analysis reveals that both microcontroller options cost between Ksh 520 and 1,300, making them equally affordable for initial component acquisition. The minimal price difference means that neither platform offers a significant economic advantage over the other in terms of direct purchase costs. A true economic comparison for CubeSat applications must therefore consider broader factors including development time, environmental hardening requirements, and long-term reliability costs.

The cost landscape changes dramatically when comparing these COTS components to space-grade alternatives. Traditional space-qualified microcontrollers command premium prices that are orders of magnitude higher than their commercial counterparts. The VORAGO VA10820, a radiation-hardened Cortex-M0 processor designed specifically for space applications, costs between Ksh 65,000 and 169,000 for the basic chip. Specialized variants can reach extraordinary prices of Ksh 1,575,000 to 2,600,000, while high-performance processors from the Gaisler LEON series are priced on a quote-only basis, reflecting their ultra-premium positioning.

SUITABILITY FOR HARSH SPACE ENVIRONMENT

Space Radiation Challenges

Space contains high-energy particles that damage electronic components over time. Two main problems affect microcontrollers in space:

Total Ionizing Dose (TID) causes gradual damage that builds up over months or years. This leads to higher power consumption, timing changes, and reduced performance. In Low Earth Orbit, commercial parts may fail within a year due to radiation levels of 4-40 krad per year.

Single Event Effects (SEE) happen when a single high-energy particle hits the microcontroller. This can cause temporary errors in memory or logic, or permanent damage through destructive short circuits that can destroy the device. These include: They include:

- **Single Event Upsets (SEU):** Non-destructive, temporary bit flips in memory cells or logic states.
- **Single Event Latch-up (SEL):** A destructive event that can create a low-impedance path (short circuit) within the device, potentially leading to permanent damage or destruction if not mitigated.

Raspberry Pi Pico (RP2040)

Has a radiation tolerance up to 50 krad, which is much better than expected for a commercial part. The main processor and input/output functions remained stable during testing. However, the onboard flash memory and SD card components failed under radiation, creating system interruptions. Replacing the flash memory with radiation-tolerant alternatives like MRAM could significantly improve reliability.

STM32F103C8T6

Has a radiation tolerance of up to 50 krad, with a slight clock speed reduction as radiation dose increased, eventually failing at 107 krad primarily due to flash memory failure. In space, the processor may run at about 85% of its programmed speed due to radiation effects. STMicroelectronics produces specialized radiation-hardened versions of STM32 microcontrollers specifically for space applications, providing 50 krad immunity without requiring additional qualification testing.

Mitigation Requirements

Commercial microcontrollers need additional protection systems to work reliably in space. These include physical shielding, backup systems, specialized radiation-tolerant memory, watchdog circuits, and sophisticated error-correction software. While the microcontroller itself costs only a few dollars, these protection systems add significant weight, complexity, and cost to the overall design.

Economic Reality

The initial low cost of commercial microcontrollers becomes misleading when all necessary protection systems are included. For short missions with lower radiation exposure, commercial parts with minimal protection may be cost-effective. However, for longer missions requiring high reliability, the total system cost including all mitigation measures can make purpose-built radiation-hardened components more economical despite their much higher initial price.

Microcontroller Radiation Performance and Cost Comparison:

Microcontroller Model	CPU Core	Typical TID Tolerance (krad(Si))	SEL Immunity (LETth in MeV-cm²/mg)	Key Radiation Hardening Features	Approximate Unit Cost (Ksh)
Raspberry Pi Pico (RP2040)	Dual-core Arm Cortex-M0+	~50 krad(Si) (observed)	Not specified for COTS, potential vulnerability	COTS, requires external mitigation for flash/SD card	Ksh 520 - 1,040
STM32F103C8T6	Single-core Arm Cortex-M3	~50 krad(Si) (observed for ARM Cortex-M)	Not specified for COTS, potential vulnerability	COTS, requires external mitigation for flash	Ksh 421 - 1,234
VORAGO VA10820	Arm Cortex-M0	>300	Latch-up immune	Rad-hard, DICE latches, TMR, EDAC, specific packaging	Ksh 65,000 - 2,600,000
Gaisler GR716B	LEON3FT SPARC V8	100	>118	Radiation-hardened, fault-tolerant, EDAC, hermetic package	Price on request
Gaisler GR740	Quad-core LEON4FT SPARV8	300	>125	Radiation-hardened, fault-tolerant,	Price on request

				ceramic/plastic package	

Table 10. Microcontroller Radiation Performance and Cost Comparison

13.4 RECOMMENDATION

Use Cases and Subsystem Suitability

Individual Use

When used individually, the Raspberry Pi Pico (RP2040 or RP2350) is suitable for**:**

i. Payload Control: For non-critical scientific instruments, cameras, or sensor arrays where data processing is required. Its dual-core nature and PIO can handle complex timing and parallel operations.

ii. On-board Data Processing: Due to its higher clock speed and larger RAM than STM32F103, it's better suited for light image processing, data compression, or machine learning inference (edge computing) for experimental payloads.

iii. Communication Interface: Can serve as a robust interface for S-band or X-band transceivers, handling modulation/demodulation if the data rates are within its capability.

It is less suitable for critical flight software ADCS (Attitude Determination and Control System) primary control due to its radiation susceptibility and reliance on external flash.

When used individually the, STM32F103C8T6 is suitable for:

i. Power Management Unit (PMU): Due to its low power modes and integrated ADCs, it can effectively monitor battery voltage, current, and control power switching to various subsystems.

ii. Attitude Determination and Control System (ADCS) - Secondary/Backup: Can handle sensor interfaces (magnetometers, gyroscopes, sun sensors) and basic attitude algorithms. Not recommended as the sole ADCS controller unless significant rad-hardening is implemented.

iii. Command and Data Handling (CDH) - Distributed Nodes: As an intelligent peripheral controller or a watchdog for other subsystems.

iv. Temperature Monitoring and Thermal Control: Its ADCs and GPIOs are suitable for reading temperature sensors and controlling heaters

The STM32F103C8T6 is less suitable for high-throughput data processing, primary flight software for critical missions without heavy redundancy.

Used Together (Heterogeneous Architecture):

Using both microcontrollers together in a heterogeneous architecture can leverage their strengths and mitigate individual weaknesses, offering a more robust and flexible system compared to using either alone.

Primary/Secondary Controller Setup:

i. STM32F103C8T6 as a "Safe Mode" Controller / Watchdog: Its lower power consumption and slightly better (though still COTS-level) observed radiation tolerance could make it a good candidate for a critical, low-power "safe mode" controller. It can monitor the health of the RP2040, trigger resets, power cycles, or switch to a minimal operational mode if the RP2040 encounters issues due to radiation.

ii. Raspberry Pi Pico (RP2040) as the "Main" Payload Processor/High-Performance CDH: The RP2040 can handle the computationally intensive tasks of the main payload (e.g., image processing, advanced algorithms), high-rate data acquisition, and primary communication protocols. It offloads these tasks from the STM32, which can then focus on critical "housekeeping" functions.

Distributed Subsystem Control:

i. STM32F103C8T6 for PMU and ADCS Front-end: One STM32 could manage the power subsystem, monitoring power lines and controlling switches. Another could interface directly with ADCS sensors and actuators (magnetorquers) for basic control, reporting data to the RP2040.

ii. Raspberry Pi Pico for Central CDH and Payload Interface: The RP2040 acts as the central command and data handling unit, orchestrating communication between the STM32-controlled subsystems and the ground station, processing scientific data from the payload, and executing higher-level mission commands. Its USB and PIO capabilities make it versatile for interfacing with various specialized sensors or transceivers.

Redundancy and Fault Tolerance:

i. By using both, designers can implement *cross-strapping* or *redundancy schemes*. If one fails, the other can attempt to take over critical functions (e.g., safe mode, communication).

ii. The STM32 could act as a hardware watchdog for the RP2040, and vice-versa, or a more complex health monitoring system could be implemented.

iii. The RP2040's higher processing power allows for more sophisticated software-based error detection and correction (EDAC) for its own internal memory and potentially for data received from other subsystems.

Why this combination is better and more economical than using only one or a rad-hard MCU:

i. Cost-Effectiveness: The combined cost of multiple Pico and STM32 boards is still orders of magnitude lower than a single rad-hard microcontroller. This allows for increased redundancy and distributed intelligence without breaking the budget.

ii. Performance vs. Reliability Trade-off: The RP2040 provides the necessary processing power for modern CubeSat payloads and data handling, while the STM32 can handle more robust, lower-power critical functions. This allows for a balance between cutting-edge capabilities and fundamental system survival.

iii. Flexibility and Modularity: Designing with distinct microcontrollers for different subsystems promotes modularity. Issues in one subsystem are less likely to bring down the entire system, and development can proceed in parallel.

iv. Risk Mitigation: While neither is rad-hard, using them together with intelligent power management, watchdogs, and software fault tolerance can significantly improve the overall system resilience against radiation-induced failures compared to a single COTS device. The STM32 can provide a basic "limp mode" capability if the more complex RP2040 fails

13.5 COMMUNICATION CHALLENGES FOR CUBESAT APPLICATIONS

1. Processing Power Limitations

Insufficient Data Rate Handling

The STM32F103C8T6 microcontroller, based on a single-core 72 MHz architecture, lacks the computational capacity to handle high-data-rate communication protocols efficiently. Its limited processing bandwidth creates difficulty when simultaneously managing communication tasks and onboard data processing. These constraints are especially critical in real-time scenarios, where the microcontroller cannot perform complex modulation or demodulation while executing other time-sensitive operations.

Mathematical Processing Limitations: The absence of a hardware floating-point unit (FPU), forcing the system to rely on slow software-based floating-point computations. This inadequacy severely impacts digital signal processing (DSP) capabilities, making it impractical to implement advanced modulation schemes. Complex algorithms, such as forward error correction and encryption, place heavy demands on the CPU, consuming excessive cycles and introducing latency that further hampers communication responsiveness.

2. Memory Constraints

Severe RAM Limitations: With only 20 KB of SRAM, the STM32F103C8T6 cannot allocate sufficient memory for essential communication buffers or full protocol stacks. Modern communication protocols typically require significantly more memory for queuing and handling packet data, making them incompatible with the microcontroller's limited resources.

Flash Memory Restrictions

The 64 KB flash memory imposes additional limitations by restricting the size and complexity of the firmware that can be stored. Communication libraries with comprehensive protocol support often exceed this capacity, forcing developers to make difficult trade-offs between features. Furthermore, the limited flash space complicates over-the-air firmware updates, which are critical for long-duration missions requiring post-launch software enhancements.

3. Modern Communication Interface Gaps

Missing Critical Interfaces

The STM32F103C8T6 does not feature native support for modern wireless standards such as Wi-Fi or Bluetooth, necessitating external modules that increase system complexity and integration overhead. Additionally, the microcontroller lacks high-speed peripheral interfaces, such as full USB 2.0 and Ethernet, which are often required for advanced communication modules. Compared to alternatives like the Raspberry Pi Pico, which offers more capable USB implementations, the STM32 falls short in interface support.

Advanced Protocol Support Issues

Implementing modern communication standards and technologies such as software-defined radio (SDR) is infeasible due to the microcontroller's limited processing power. It cannot accommodate adaptive protocols that require real-time switching or dynamic resource allocation. Additionally, its limited capacity for secure computation restricts the implementation of modern encryption and authentication protocols, critical for mission data security.

4. Advanced Modulation and Signal Processing

Digital Signal Processing Inadequacy

The STM32F103C8T6 lacks the resources to execute complex modulation schemes such as QAM or OFDM and cannot implement sophisticated error correction algorithms. Real-time signal quality analysis is also infeasible, which undermines the ability to dynamically adjust system parameters based on link conditions. Adaptive communication systems, which are increasingly essential in space applications, are beyond the microcontroller's capabilities.

Software-Defined Radio Limitations

The microcontroller is unsuitable for software-defined radio applications due to its inadequate computational performance and memory. It cannot support frequency agility, generate complex waveforms, or operate on multiple frequency bands simultaneously. These limitations prevent the implementation of advanced, flexible communication systems tailored to dynamic mission requirements.

Summary on critical limitations:

The STM32F103C8T6 is fundamentally constrained in several key areas:

1. It lacks the computational capability to support modern, high-data-rate communication.
2. It has insufficient memory to implement full-featured communication protocol stacks.
3. It cannot perform real-time digital signal processing or error correction.
4. It is incapable of supporting software-defined radio functionality.
5. It does not natively interface with modern high-speed communication hardware

To overcome these challenges, a hybrid architecture is recommended. The STM32F103C8T6 should be relegated to support functions such as radio power management, basic UART or SPI communication, and watchdog monitoring. Meanwhile, a more capable microcontroller like the Raspberry Pi Pico can serve as the primary communication controller.

This approach leverages the strengths of both systems:

- The STM32 handles low-level interfacing, power control, and auxiliary communication tasks.
- The Pi Pico is responsible for computationally intensive communication processing, including protocol handling, modulation/demodulation, and error correction.
- The combined system balances reliability with performance, providing a robust solution for CubeSat communications in low-Earth orbit and beyond.

14. ARDUINO DUE

The Arduino Due, also known as the 32-Bit Arduino pioneer, was launched in 2012, represented a paradigm shift for the Arduino platform, moving from the 8-bit AVR architecture to a 32-bit ARM Cortex-M3 core.

The increase of 32-bit microcontrollers (MCUs) has revolutionized embedded system design, offering a significant leap in processing power, memory capacity, and peripheral sophistication over their 8-bit predecessors. This advancement has enabled more complex applications and has democratized access to powerful computing for hobbyists, educators, and professionals alike.

The Arduino Due and the Adafruit Feather M4 Express stand out as popular representatives of this trend, each catering to different design philosophies and application niches. The Arduino Due based on the Atmel (now Microchip) SAM3X8E, was Arduino's pioneering foray into the 32-bit ARM world. The Adafruit Feather M4 Express, featuring the Microchip ATSAMD51, embodies a more modern approach with a focus on performance, Python integration, and a compact, extensible form factor.

The allure of Commercial-Off-The-Shelf (COTS) components, including these MCUs, for space applications, especially in the blooming CubeSat sector, is undeniable due to reduced costs and faster development cycles. However, this approach necessitates a thorough understanding of the components' capabilities and, more critically, their vulnerabilities in the unforgiving space environment. This review aims to provide that understanding for the Due and Feather M4 Express.

14.1 DIAGRAM

Figure 25 Arduino Due MCU

14.2 PIN LAYOUT

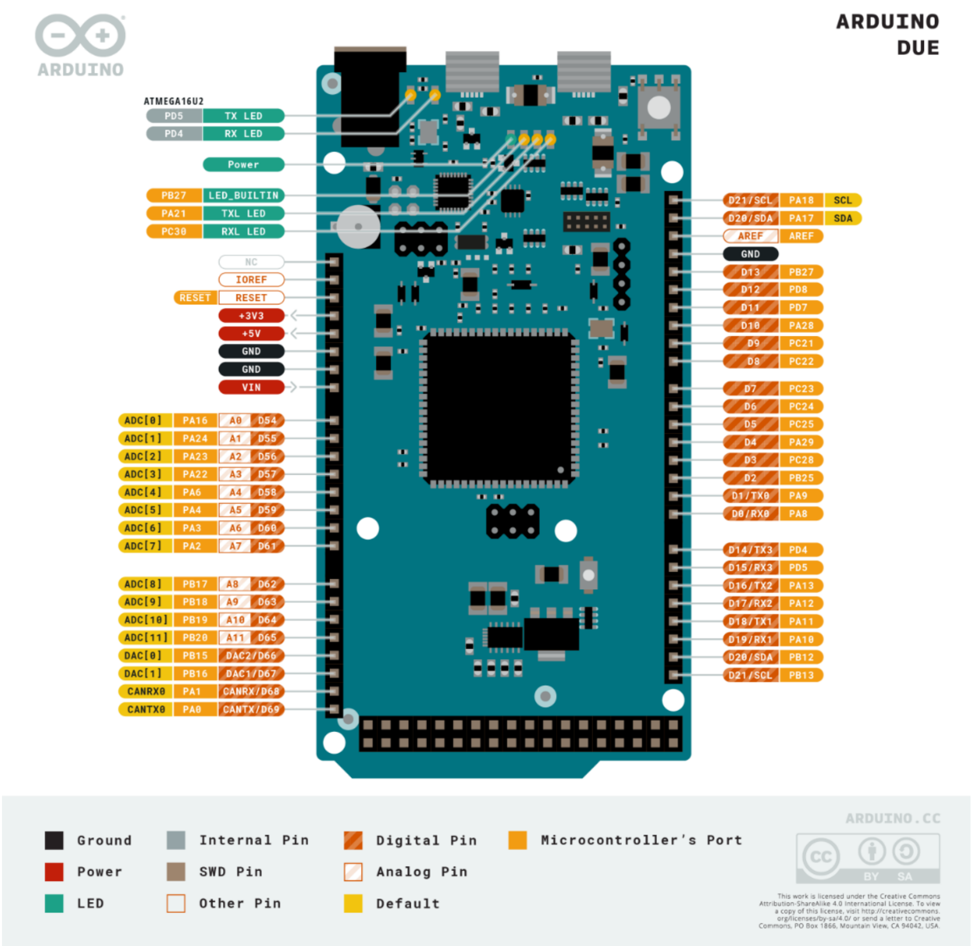


Figure 26 Arduino Due pin layout

14.3 CORE ARCHITECTURE AND DETAILED SPECIFICATIONS

The Arduino Due is powered by the Atmel SAM3X8E microcontroller.

- **Processor:** Atmel SAM3X8E, featuring an ARM Cortex-M3 core. The Cortex-M3 is a 32-bit RISC processor implementing the ARMv7-M architecture, known for its balance of performance and energy efficiency. It includes a 3-stage pipeline and Harvard architecture (separate instruction and data buses).
- **Clock Speed:** 84MHz. This is derived from a PLL (Phase Locked Loop) driven by an external crystal.
- **Memory:**
  - **Flash Memory:** 512 KB, organized in two 256 KB banks. This allows for Read-While-Write (RWW) operations, potentially enabling bootloader updates or data logging without halting program execution.

    - **SRAM:** 96KB split into two contiguous banks: 64 KB (SRAM0) and 32 KB (SRAM1). This segmented SRAM can be accessed concurrently by different bus masters (e.g., CPU and DMA).
- **Operating Voltage:** 3.3V. A critical distinction from many 8-bit Arduinos, as applying 5V to I/O pins can permanently damage the SAM3X8E.
- **Digital I/O Pins:** 54, with extensive multiplexing for peripherals. 12 pins support Pulse Width Modulation (PWM).
- **Analog Capabilities:**
    - **Analog Input Pins (ADC):** 12 channels, multiplexed to a single 12-bit Analog-to-Digital Converter (ADC) capable of up to 1Msps.
    - **Analog Output Pins (DAC):** 2 channels, connected to a 12-bit Digital-to-Analog Converter (DAC), a rare feature on Arduino boards.
- **Communication Peripherals:**
    - 4 UARTs (hardware serial ports)
    - 2 TWIs (I2C compatible interfaces)
    - 1 SPI header (with advanced SPI features like chip select control)
    - 1 CAN interface (compliant with CAN 2.0A and 2.0B)
    - USB: One native USB OTG capable port (programming and host/device functionality) and one programming port (interfaced via an ATMega16U2).
- **DMA Controller:** The SAM3X8E includes a Central DMA (DMAC) controller, allowing peripherals to transfer data to/from memory without CPU intervention, significantly improving throughput for data-intensive tasks.
- **Other Features:** JTAG header for advanced debugging, erase button (to clear flash), reset button.

14.4 SOFTWARE ECOSYSTEM AND DEVELOPMENT NUANCES

Arduino Due is primarily programmed via the Arduino IDE using C++, the Due benefits from the Arduino abstraction layer. However, developers seeking maximum performance or access to all SAM3X8E features often delve into direct register manipulation or utilize Microchip's (formerly Atmel's) Software Framework (ASF).

- **Library Compatibility:** While many core Arduino libraries were ported, the 3.3V logic and different underlying hardware mean that libraries directly manipulating AVR registers or assuming 5V logic are incompatible or require significant modification.

- **Interrupts and Timers:** The ARM Cortex-M3 offers a more sophisticated Nested Vectored Interrupt Controller (NVIC) and more capable timers than AVRs, but leveraging these fully might require stepping outside standard Arduino functions.
- **Real-Time Operation**: While not an RTOS-centric platform out-of-the-box, the SAM3X8E can run RTOSs like FreeRTOS, ChibiOS, enabling more deterministic real-time applications.

14.5 TYPICAL APPLICATIONS AND DESIGN PHILOSOPHY

The Due targets projects outgrowing the capabilities of 8-bit Arduinos:

- Complex robotics requiring multiple sensor inputs and motor controls.
- Basic digital audio processing leveraging the DACs and ADC.
- Automotive projects utilizing the CAN bus.
- Projects needing high I/O counts or multiple simultaneous serial communications.

14.6 ADVANTAGES

- **High I/O Count:** Unmatched by most development boards in its class.
- **Dual DACs:** Useful for analog signal generation.
- **CAN Bus:** Integrated CAN controller is a significant advantage for automotive or industrial networking.
- **Mature Cortex-M3 Core:** Well-understood architecture with good compiler and toolchain support.
- **DMA:** Enables efficient data handling.

14.7 DISADVANTAGES

- **Strict 3.3V I/O:** Requires careful level shifting for 5V systems.
- **Power Consumption:** Higher than many contemporary MCUs, especially newer Cortex-M0+ or M4 designs. Sleep modes are available but the platform isn't optimized for ultra-low power.
- **Floating Point:** No hardware Floating Point Unit (FPU); floating-point operations are emulated in software, impacting performance.

- **Form Factor:** Large Arduino Mega-like footprint is not ideal for space-constrained designs.
- **Age:** While still capable, the SAM3X8E is an older design compared to newer ARM Cortex-M generations.

15. ADAFRUIT FEATHER M4 EXPRESS

The Adafruit Feather M4 Express, based on the Microchip ATSAMD51J19A, is a flagship of Adafruit's Feather ecosystem, emphasizing high performance, ease of use with CircuitPython, and extensibility through FeatherWings.

15.1 PIN LAYOUT

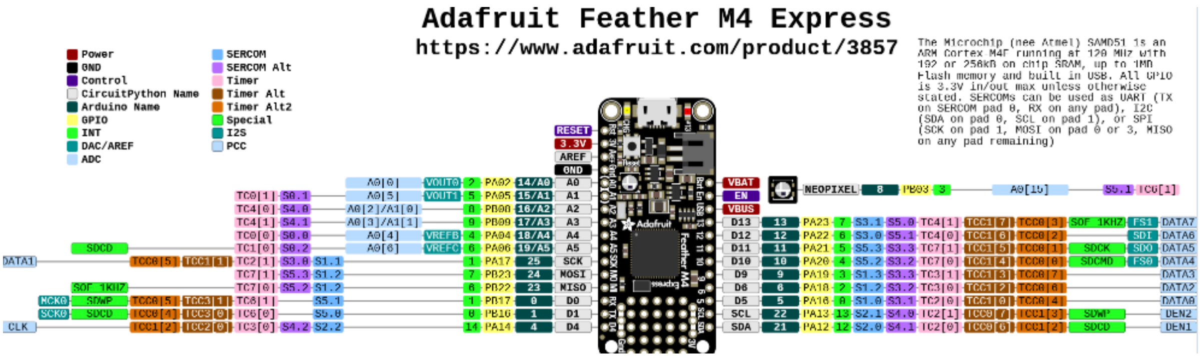


Figure 27 Adafruit Feather M4 Express pinout

15.2 CORE ARCHITECTURE AND DETAILED SPECIFICATIONS

The Feather M4 Express utilizes the Microchip ATSAMD51J19A MCU.

- **Processor:** Microchip ATSAMD51J19A, featuring an ARM Cortex-M4F core. The Cortex-M4F includes a single-precision hardware Floating Point Unit (FPU) and DSP instructions, dramatically accelerating mathematical computations.
- **Clock Speed:** Typically 120 MHz, but the ATSAMD51 can be clocked up to 200 MHz in some configurations (though Adafruit's board runs it at 120 MHz for stability and power).
- **Memory:**
    - **Flash Memory (Internal):** 512 KB for program code. The SAMD51 features ECC (Error Correcting Code) on its flash memory.
    - **SRAM:** 192 KB, a substantial amount for complex applications and data buffering.
    - **External QSPI Flash**: 2 MB connected via a Quad Serial Peripheral Interface. This is primarily used by CircuitPython for storing the Python runtime, libraries, and user scripts as a small filesystem, but can also be used for general data logging.
- **Operating Voltage:** 3.3V logic and power.

- **Digital I/O Pins:** 21 GPIO pins. These are highly multiplexed via the SERCOM (Serial Communication Interface) system.
- **Analog Capabilities:**
    - **Analog Input Pins (ADC):** 6 pins multiplexed to two 12-bit ADCs (though typically one is used at a time for Arduino compatibility), capable of up to 1Msps.
    - **Analog Output Pins (DAC):** 1 pin connected to a 10-bit DAC.
- **Communication Peripherals (SERCOM):** The ATSAMD51 features multiple (typically 6-8 depending on package) highly configurable SERCOM peripherals. Each SERCOM can be independently configured as an I2C, SPI, or UART interface, offering great flexibility in peripheral assignment to pins.
    - Native USB (device and host capabilities).
    - I2S (Inter-IC Sound) interface for digital audio.
    - PDEC (Position Decoder) for quadrature encoder input.
- **DMA Controller:** A powerful DMA controller with multiple channels to offload data transfers.
- **Other Features:** Built-in LiPo battery charging circuitry, status NeoPixel RGB LED, SWD (Serial Wire Debug) port for advanced debugging.

15.3 SOFTWARE ECOSYSTEM AND DEVELOPMENT NUANCES

The Feather M4 Express shines with its dual support for:

- **Arduino IDE:** Programmed in C/C++ using Adafruit's SAMD core, which provides access to many ATSAMD51 features. Performance-critical applications often use this route.
- **CircuitPython:** A fork of MicroPython optimized for ease of use on Adafruit (and other) boards. It presents the QSPI flash as a USB drive, allowing for drag-and-drop code updates and rapid iteration. This significantly lowers the barrier to entry for Python developers.
- **RTOS Support:** The ATSAMD51 is well-suited for running RTOSs like FreeRTOS, and Adafruit provides examples.
- **Low-Level Access:** For advanced users, Microchip's Harmony framework or direct register programming can unlock the full potential of the ATSAMD51.

15.4 TYPICAL APPLICATIONS AND DESIGN PHILOSOPHY

The Feather M4 Express is designed for:

- IoT projects requiring connectivity, processing power, and low-power capabilities.
- Wearable technology and portable devices benefiting from the small form factor and battery management.
- Machine learning inference at the edge (e.g., TensorFlow Lite for Microcontrollers).
- Projects involving digital signal processing, audio synthesis, or sensor fusion thanks to the FPU and DSP instructions.
- Rapid prototyping and educational purposes due to CircuitPython's accessibility.

15.5 ADVANTAGES

- **Powerful Cortex-M4F Core:** FPU and DSP instructions provide significant performance gains for specific tasks.
- **Ample SRAM (192KB):** Facilitates complex algorithms and larger data sets.
- **QSPI Flash:** Excellent for file storage, configuration data, or large CircuitPython applications.
- **CircuitPython:** Superb for rapid development and ease of use.
- **Integrated LiPo Charging:** Simplifies portable project design.
- **Compact Feather Form Factor & Ecosystem:** Small size and wide range of plug-and-play FeatherWing add-ons.
- **Advanced Peripherals:** Flexible SERCOMs, I2C, and PDEC.
- **Power Efficiency:** The SAMD51 offers more granular power control and lower active power consumption relative to its performance compared to the SAM3X.

15.6 DISADVANTAGES

- **Fewer Total I/O Pins:** Compared to the Due, the 21 GPIOs can be limiting for I/O-heavy applications without expanders.
- **Single DAC (10-bit):** Less analog output capability than the Due.

- **Complexity:** The ATSAMD51's rich peripheral set and clocking system can be complex to configure at a low level.
- **QSPI as System Drive:** While flexible, reliance on external QSPI for CircuitPython's core operation means the system is inoperable if this flash fails.

15.7 IN-DEPTH COMPARATIVE ANALYSIS: ARDUINO DUE VS. ADAFRUIT FEATHER M4 EXPRESS

Table 14 comparative analysis: arduino due vs. Adafruit feather m4 express

Feature	**Arduino Due (SAM3X8E)**	**Adafruit Feather M4 Express (ATSAMD51J19A)**	**Notes**
Processor Core	ARM Cortex-M3 @ 84 MHz	ARM Cortex-M4F @ 120 MHz	M4F includes single-precision FPU & DSP instructions, drastically outperforming M3 in math/signal processing. M4F also has slightly deeper pipeline.
SRAM	96 KB (2 banks)	192 KB	Feather M4's larger contiguous SRAM is crucial for RTOS, complex algorithms, large network buffers. Due's banked SRAM allows some concurrency.
Flash (Internal)	512 KB (2 banks, RWW)	512 KB (with ECC)	Due's RWW is good for firmware updates. Feather's ECC offers better

			data integrity for program code.
Flash (External)	None	2 MB QSPI	Feather's QSPI is a major asset for data logging, CircuitPython environment, and large asset storage.
Floating Point Unit	No (Software Emulation)	Yes (Hardware Single-Precision)	Feather M4 is orders of magnitude faster for float operations (e.g., sensor fusion, FFTs, filtering).
Digital I/O	54	21	Due offers significantly more direct I/O. Feather relies on its ecosystem or I/O expanders for more connections.
Analog Inputs (ADC)	12 channels (12-bit, 1 Msps)	6 channels (12-bit, 1 Msps, dual ADCs available)	Due offers more raw analog input channels. Feather's SAMD51 ADC is quite capable and flexible.
Analog Outputs (DAC)	2 channels (12-bit)	1 channel (10-bit)	Due provides superior analog output capabilities.
Special Peripherals	CAN Controller, Ethernet (on variants not Due)	I2S, PDEC, Flexible SERCOMs	Due's CAN is unique. Feather's SERCOMs offer versatile I2C/SPI/UART configurations; I2S is good for audio.

Power Management	Basic sleep modes, generally higher consumption	Advanced sleep modes, power gating, lower active current	ATSAMD51 is designed for better power efficiency with more granular control. Feather board adds LiPo charging.
Form Factor	Arduino Mega size (101.5 x 53.3mm)	Feather size (51 x 23 mm)	Feather is vastly smaller and lighter, crucial for wearables, portables, and space/mass constrained systems.
Primary Software	Arduino IDE (C++), ASF	Arduino IDE (C++), CircuitPython, Microchip Harmony	Feather's CircuitPython enables rapid development. Both support C/C++ for performance.
Debugging	JTAG, Serial	SWD, Serial	Both offer robust hardware debugging. SWD uses fewer pins than JTAG.
Community & Ecosystem	Large, mature Arduino community. Less SAM3X specific.	Strong Adafruit/CircuitPython community, rich FeatherWing ecosystem.	Adafruit provides excellent libraries and support for its hardware.

15.8 SUITABILITY AND LIMITATIONS IN SATELLITE PROJECTS

The use of COTS MCUs in space missions, especially CubeSats, is driven by pragmatism but fraught with risks associated with the space environment.

The Space Environment Challenges

- **Radiation:**
  - **Total Ionizing Dose (TID):** Accumulation of radiation dose over mission lifetime, leading to parametric degradation (e.g., increased leakage current, threshold voltage shifts) and eventual functional failure.
  - **Single Event Effects (SEEs):** Caused by individual high-energy particles.
    - **Single Event Upset (SEU):** Bit-flips in memory cells (SRAM, registers and configuration bits in FPGAs/MCUs). Can corrupt data or program flow.
    - **Single Event Functional Interrupt (SEFI):** A soft error that causes the device to enter an unexpected state, often requiring a reset or power cycle to recover.
    - **Single Event Latch-up (SEL):** A potentially destructive high-current state triggered in CMOS structures, requiring immediate power cycling to prevent permanent damage. COTS MCUs are often prone to SEL.
    - **Single Event Gate Rupture (SEGR) / Single Event Burnout (SEB)**: Destructive events in power MOSFETs, less common in MCUs themselves but relevant for power systems.
- **Temperature Extremes & Cycling:** Satellites in LEO can experience temperature swings from -40°C to +85°C or wider. MCUs must operate reliably, and repeated cycling can cause mechanical stress.
- **Vacuum:** Outgassing of materials can contaminate sensitive optical surfaces or cause issues with high voltage components. Packaged MCUs are generally sealed, but PCBs and other materials are a concern.
- **Power Constraints:** Solar power and battery capacity are limited, mandating highly efficient power usage and robust sleep modes.
- **Reliability & Uptime:** Mission failure is costly. Systems must be robust, with fault detection, isolation, and recovery (FDIR) mechanisms.

Arduino Due (SAM3X8E) in Satellite Projects

- **Potential Suitability:**
  - **High I/O for Payloads:** If a mission requires interfacing with a large number of simple sensors or actuators directly, the Due's pin count is an advantage.
  - **CAN Bus for Intra-Satellite Communications:** Useful if a CAN bus is adopted for the satellite's internal data network.
  - **Familiarity for some teams:** The Arduino ecosystem can lower the entry barrier for software development if performance is not paramount.
- **Significant Limitations & Risks:**
  - **Radiation Susceptibility:** The SAM3X8E, being an older design on a larger process node, is highly susceptible to SEUs in its 96KB SRAM and registers. SEL risk is also present and needs characterization. TID tolerance is likely low.
  - **Power Consumption:** Higher active and sleep currents compared to more modern MCUs make it less suitable for power-starved missions.
  - **Lack of FPU:** Any calculations requiring floating-point (e.g., ADCS algorithms) will be slow.
  - **Form Factor & Mass:** The standard Due board is too large and heavy. A custom PCB with the SAM3X8E would be necessary, but the chip itself is also relatively large.
  - **Limited Error Mitigation Features:** Lacks internal ECC on SRAM. RWW on Flash is useful but doesn't protect against Flash SEUs directly.

Adafruit Feather M4 Express (ATSAMD51J19A) in Satellite Projects

- **Potential Suitability:**
  - **OBC/ADCS Processing**: The Cortex-M4F core with its FPU and DSP instructions is well-suited for attitude determination and control (e.g., running Kalman filters, processing IMU/star tracker data), and complex payload data processing.
  - **Data Logging & Storage:** The 2MB QSPI flash is invaluable for telemetry storage, scientific data buffering, or even holding larger mission software segments.
  - **Power Efficiency:** The ATSAMD51 offers better power management features (e.g., granular clock gating, multiple sleep modes) allowing for lower average power consumption if carefully managed.

- **Compactness:** The ATSAMD51 chip itself allows for very dense custom board designs, fitting well within CubeSat volume constraints.
- **ECC on Internal Flash:** Provides some protection against SEUs corrupting the program code.

- **Significant Limitations & Risks:**
  - **Radiation Susceptibility (SRAM & Peripherals):** The 192KB SRAM is highly vulnerable to SEUs. The complex peripherals and clocking system of the ATSAMD51 mean many configuration registers are also susceptible to SEUs, potentially leading to intricate SEFIs. SEL risk must be assessed.
  - **QSPI Flash Reliability in Space**: While useful, the radiation tolerance (TID and SEE) of generic QSPI flash chips used on development boards is often unknown and can be poor. Errors in QSPI could corrupt the CircuitPython environment or stored data. Careful selection of space-grade or radiation-tolerant QSPI flash would be ideal but adds cost and complexity.
  - **Software Complexity:** While CircuitPython aids rapid prototyping, flight software would likely be C/C++ with an RTOS or bare-metal for determinism and control. Managing the ATSAMD51's complexity (numerous clocks, peripherals, DMA) in a high-reliability context is demanding.
  - **Thermal Performance:** High clock speeds (120MHz+) can lead to significant heat generation in a vacuum where convective cooling is absent. Thermal analysis and mitigation (heatsinks, thermal pathways to chassis) are critical.

Essential Mitigation Strategies for COTS MCUs in Space

Regardless of the choice, deploying COTS MCUs necessitates a robust FDIR strategy:

- **Hardware Watchdogs:** External watchdog timers are crucial to recover from SEFI-induced hangs. Internal watchdogs are a first line of defense.
- **Redundancy:**
  - **Component Redundancy:** Using multiple MCUs (e.g., two or three in a voting or hot/cold spare configuration) for critical functions.
  - **Data Redundancy:** Storing critical data with checksums, CRCs, or Reed-**Solomon codes.**
- **Software Fault Tolerance:**

- **Memory Scrubbing:** Periodically reading and rewriting SRAM (if ECC is not present) or critical data areas to correct accumulated SEUs (requires knowing good values or using error codes).
- **Defensive Programming:** Robust error handling, sanity checks, and safe-state fall-backs.
- **Instruction/Task-Level TMR:** Triple Modular Redundancy for critical calculations or code sections, though this has high overhead.

- **Radiation Shielding:** Localized shielding (e.g., tantalum spots over sensitive chips) can reduce TID and somewhat lower SEE rates, but adds mass and may generate secondary particles.
- **Latch-up Protection:** Current limiting and fast power cycling circuitry to detect and mitigate SELs.
- **Thorough Testing:**
    - **Functional Testing:** Across the full expected temperature range in a thermal vacuum chamber.
    - **Radiation Beam Testing:** Exposing components to proton and heavy-ion beams to characterize SEE cross-sections, SEL thresholds, and TID tolerance. This is expensive but provides invaluable data.
- **System-Level Design:** Ensuring a robust power distribution system, careful clock management, and reliable inter-component communication.

15.9 PRICES

The Arduino Due is locally available ranging from Ksh.4000- 5000 whereas the Adafruit feather M4 Express isn't readily available and would have to be shipped from abroad. Its price tag is around $23 non-inclusive of shipping costs.

15.10 CONCLUSION

The Arduino Due and Adafruit Feather M4 Express are powerful 32-bit microcontrollers offering distinct advantages for terrestrial embedded applications. The Due excels with its sheer number of I/O pins and established Arduino ecosystem, making it suitable for complex interfacing tasks. The Feather M4 Express, with its faster Cortex-M4F core, FPU, integrated QSPI flash, and strong CircuitPython support, offers a compelling package for computationally intensive tasks, IoT, and rapid prototyping in a compact, power-conscious form factor.

For satellite projects, particularly cost-sensitive CubeSats, both platforms present the allure of COTS accessibility and performance. The Feather M4 Express appears slightly more advantageous due to its more modern core (Cortex-M4F with FPU), better potential power efficiency, and onboard QSPI flash for data storage. Its FPU is particularly beneficial for ADCS and complex sensor processing. The Arduino Due's main advantage would be its high I/O count if a mission absolutely required numerous direct connections without expanders.

However, neither is inherently designed for the harsh space environment. Their use necessitates a thorough understanding of their limitations, especially concerning radiation effects and temperature cycling. Significant effort in implementing hardware and software fault tolerance mechanisms, along with potential shielding, is mandatory. While COTS MCUs like these are making space more accessible, mission success relies heavily on rigorous testing, robust system design, and realistic expectations of their performance and reliability in orbit. Future research should continue to focus on characterizing COTS MCU behaviour in simulated space environments and developing standardized mitigation techniques.

16. THE TEENSY MICROCONTROLLERS

The teensy microcontroller series is a family of small powerful, microcontrollers which are suitable for fast embedded computing.

These microcontrollers are exceptionally good when it comes to audio interfacing.

They are programmed using Arduino IDE with the **Teensyduino** plugin.

16.1 DIAGRAM

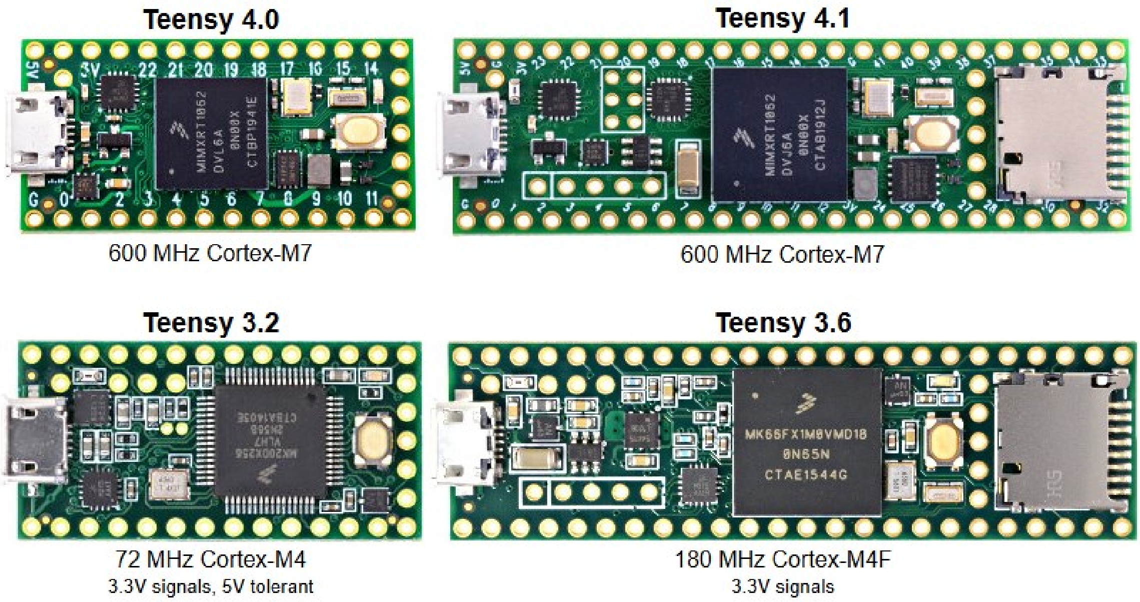


Figure 28 Teensy 4.0, 4.1, 3.2 and 3.6 diagrams

16.2 TEENSY 4.0 AND 4.1

16.2.1PIN LAYOUT

The following is the pin diagrams;

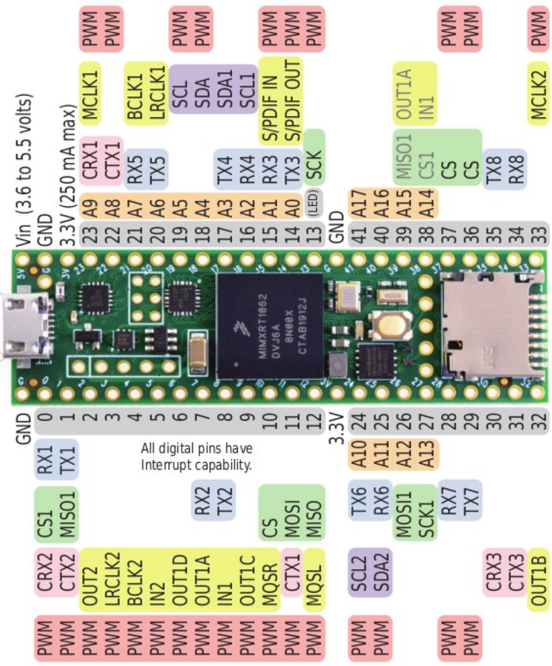


Figure 29 Teensy 4.1 pin layout

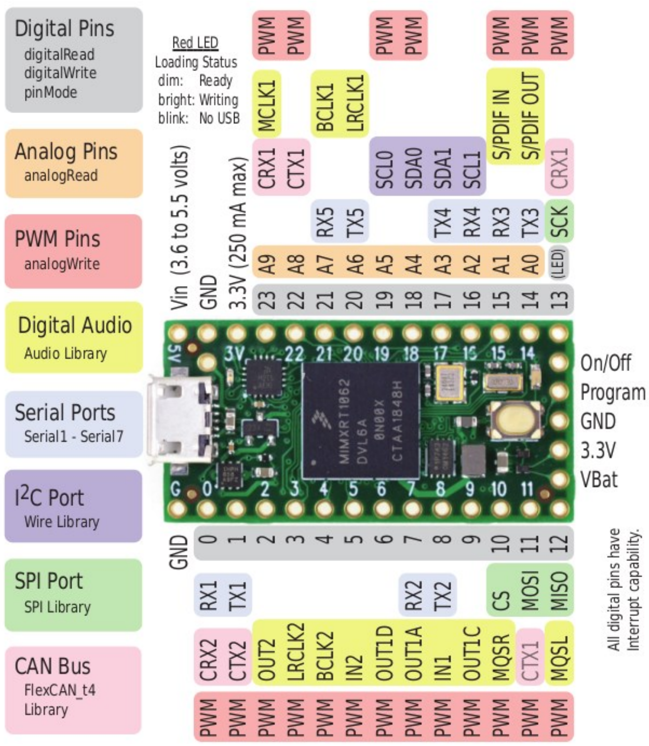


Figure 30 Teensy 4.0 pin layout

16.2.2FEATURES

The most powerful models are the teensy 4.0 and 4.1 with the following specs;

Teensy 4.0

Processor: ARM Cortex-M7 @ 600 MHz

Flash: 2 MB

RAM: 1 MB (512K tightly coupled + 512K general)

GPIO: 40 pins (31 digital I/O, 14 analog inputs)

Interfaces: 3 UART, 3 SPI, 2 I2C, CAN, USB Host/Device

Size: 1.4" × 0.7"

Teensy 4.1

Processor: Same as 4.0 (Cortex-M7 @ 600 MHz)

Flash: 8 MB (external QSPI)

RAM: 1 MB + optional PSRAM

GPIO: 55+ pins, including Ethernet, microSD slot, and more serial interfaces

Expansion: 2 memory expansion sockets (PSRAM/Flash)

Size: 2.4" × 0.7"

Special features

Audio Library & Shield: Supports 44.1 kHz 16-bit audio with DSP filters, FFT, mixers, etc.

Real-Time Performance: Low-latency I/O, fast interrupts, and deterministic control.

Overclockable: Can be safely pushed to 720 MHz in some applications.

Power: Operates at 3.3V logic; draws more current than typical Arduino boards due to high speed.

16.2.3ADVANTAGES

Why and where to use Teensy 4.0 or 4.1 microcontrollers in a LEO Satellite:

1. The teensy microcontrollers are faster than a lot of microcontrollers including many Arduinos. With the speed of 600MHz, compared to the 16MHz speed of Arduino Uno, Arduino Nano and Arduino Mega. This makes them suitable for more complex computations and UHF communication.
2. They are equipped with useful Audio and digital signal processing (DSP) hardware which are helpful in vibration analysis, monitoring different sounds and signal processing. They are also suitable for data compression for transmission.
3. They have rich I/O and communication options. Teensy microcontrollers support UART, SPI, I2C, CAN, USB, PWM and ADC. A very important option is the SPI option which is fully supported and can be used for interfacing with long range communication (LoRa), which is an important communication aspect we can use for long distance communication such as in our LEO satellite.
4. Teensy microcontrollers support SD cards and this can be helpful when logging sensor data, audio files and images.
5. They are also small in size making them lighter for space and are cheap and affordable.

16.2.4LIMITATIONS

1. It's not space-rated. The teensy microcontroller is vulnerable to radiations and high space temperatures and hence not usually suitable for many space operations.
2. It does not have a built-in Bluetooth or Wi-Fi hence, in order to incorporate these capabilities, other modules are connected which increase the power drawn and complexity.
3. At 600MHz operation, it consumes more power hence may need custom power regulation.

In conclusion, due to it's various features such as high speeds and advanced communication features, I can recommend the use of Teensy microcontrollers for communication, including receiving data from telemetry health sensors and Ground sensors, compressing the data and interfacing with communication modules like LoRa or LoRaWAN for transmission of the data over a wide area from space to the earth or back.

16.3 TEENSY 3.2

16.3.1PIN LAYOUT

The teensy 3.2 is an older version of the teensy microcontrollers. It has the following pin diagram;

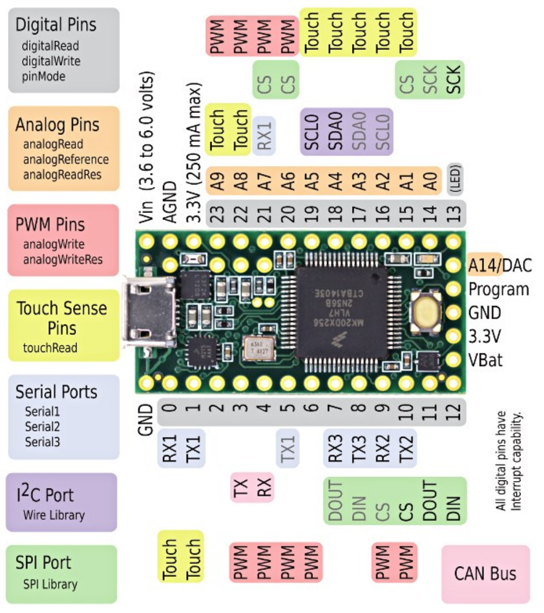


Fig 23. Teensy 3.2 pin layout

FEATURES

Microcontroller: NXP MK20DX256VLH7 (ARM Cortex-M4)

CPU Speed: 72 MHz (overclockable to 96 MHz)

Flash Memory: 256 KB

RAM: 64 KB

EEPROM: 2 KB

Operating Voltage: 3.3V (with 5V-tolerant inputs)

Input Voltage Range: 3.6V to 6.0V

USB: Micro USB (12 Mbit/sec, supports USB HID, MIDI, Serial, etc.)

Dimensions: 1.4" x 0.7" (35.56 mm x 17.78 mm)

I/O and Peripherals

Digital I/O Pins: 34

PWM Outputs: 12

Analog Inputs: 21

Analog Output: 1 (DAC)

UART (Serial Ports): 3

SPI Ports: 1

I2C Ports: 2

CAN Bus: 1

I2S Audio Interface: 1

DMA Channels: 16

Real-Time Clock (RTC): Yes

Touch Sensing Inputs: Available

16.3.2 ADVANTAGES

Why and where to use Teensy 3.2 microcontroller in a LEO Satellite:

1. **Moderately Fast Processing Power**
   The Teensy 3.2 operates at 72 MHz (overclockable to 96 MHz), which is significantly faster than many traditional Arduino boards (e.g., Arduino Uno, Mega at 16 MHz). While not as fast as Teensy 4.x, this speed is still sufficient for mid-level tasks like environmental sensor data processing, telemetry formatting, and basic communications such as UART or SPI interfacing with LoRa modules.

2. **Built-in Audio and DSP Capabilities**
   Teensy 3.2 includes a 12-bit DAC and supports an audio library for basic signal processing tasks. While not as powerful as the Teensy 4.x in DSP, it still supports applications like sound detection, basic vibration analysis, or simple audio compression tasks — useful for payload diagnostics or acoustic monitoring.

3. **Rich I/O and Communication Options**
   Teensy 3.2 supports multiple UARTs, SPI, I2C, PWM, ADC, and even CAN Bus — making it very flexible for LEO satellite subsystems. SPI is especially critical for interfacing with long-range communication modules like LoRa, a key asset for low-power, long-distance communication between satellite and ground.

4. **Support for SD Cards**
   Teensy 3.2 can interface with SD cards via SPI, enabling local logging of telemetry data, environmental sensor readings, or captured images. This is useful for storing data during communication blackouts or scheduled downlink windows.

5. **Compact and Lightweight**
   At only 35.56 mm x 17.78 mm, and weighing very little, Teensy 3.2 is ideal for space-constrained systems like CubeSats. It is also very cost-effective compared to more space-rated or high-end computing modules.

16.3.3 LIMITATIONS

1. **Not Space-Rated**
   Teensy 3.2 is a commercial-grade MCU not hardened against radiation or extreme temperature fluctuations, which poses a risk in the harsh environment of space. Proper shielding and error correction would be necessary for space missions.
2. **No Built-in Wireless Communication**
   Lacking onboard Bluetooth or Wi-Fi, it requires external modules to add these capabilities, increasing system complexity and power usage — a challenge in power-limited space systems.
3. **Limited Processing Power and RAM**
   With only 72 MHz CPU speed and 64 KB RAM, Teensy 3.2 is not ideal for compute-heavy operations like real-time image processing, advanced compression, or ML inference, which may be required for future LEO satellite missions.

16.3.4 CONCLUSION

Teensy 3.2 is a practical choice for specific LEO satellite subsystems such as **sensor interfacing**, **basic data preprocessing**, and **communication module control** (e.g., LoRa telemetry). Its balance of size, power efficiency, and I/O capabilities make it well-suited for **non-critical onboard control and data acquisition tasks**. However, for more intensive computation or higher fault tolerance, **Teensy 4.0 or 4.1** would be better suited.

17. RASPBERRY PI ZERO 2 W MSP430 (TI) AND TIVA C SERIES LAUNCHPAD MICROCONTROLLERS

Modern satellite systems — especially CubeSats and small-scale platforms — require a mix of high processing power, ultra-low power efficiency, and real-time control. To meet these diverse demands, a hybrid architecture leveraging multiple microcontrollers is often deployed. This design ensures fault tolerance, modular development, and optimal resource allocation. In this context, **Raspberry Pi Zero 2 W**, **MSP430**, and **Tiva C Series LaunchPad** microcontrollers each serve specialized roles within the satellite subsystem:

Raspberry Pi Zero 2 W – ***Onboard Computer (OBC) & Payload Processing***

As a compact Linux-based single-board computer, the Raspberry Pi Zero 2 W handles **data-intensive tasks**, such as **image processing**, **machine learning**, and **high-level communication protocols**. Its quad-core 64-bit ARM Cortex-A53 processor makes it ideal for processing payload data from sensors like cameras or spectrometers. Although not real-time or radiation-hardened, it is well-suited for **non-critical tasks** that benefit from the flexibility and computing power of a general-purpose OS.

MSP430 (Texas Instruments) – ***Power Management Unit (PMU) & Sensor Interface***

The MSP430 microcontroller excels in **ultra-low power consumption**, making it the best choice for always-on operations such as **voltage/current monitoring**, **temperature sensing**, and **basic fault detection**. Operating efficiently even in deep-sleep modes, the MSP430 ensures **continuous environmental monitoring** while conserving energy—a crucial feature for long-duration missions with tight power budgets.

Tiva C Series LaunchPad – ***Attitude Determination & Control System (ADCS) & Real-Time Coordination***

The Tiva C Series, based on the ARM Cortex-M4F architecture, provides a robust platform for **real-time processing**, essential for **attitude control**, **reaction wheel management**, and **intra-satellite communication**. With support for FreeRTOS, floating-point math, and advanced interfaces like CAN and SPI, the Tiva C operates as the **deterministic control layer** of the satellite, orchestrating time-sensitive subsystems.

17.1 RASPBERRY PI ZERO 2 W

17.1.1DIAGRAM

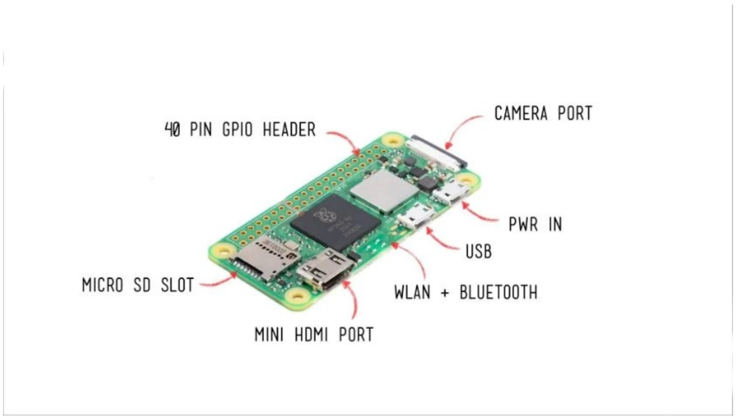


Figure 31 Raspberry Pi Zero 2 W diagram

17.1.2SPECIFICATIONS

- 1GHz quad-core 64-bit Arm Cortex-A53 CPU
- 512MB SDRAM
- 2.4GHz 802.11 b/g/n wireless LAN
- Bluetooth 4.2, Bluetooth Low Energy (BLE), onboard antenna
- Mini HDMI port and micro–USB On-The-Go (OTG) port
- microSD card slot
- CSI-2 camera connector
- HAT-compatible 40-pin header footprint (unpopulated)
- H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30)
- OpenGL ES 1.1, 2.0 graphics
- Micro USB power
- Composite video and reset pins via solder test points
- 65mm x 30mm

Table 15 Raspberry Pi Zero 2 W specifications

Feature	Specification	Satellite Role
CPU	1GHz ARM1176 (Single-core)	Onboard Computer (OBC)
Memory	512MB RAM	Payload Data Processing
Wireless	Wi-Fi 802.11n, Bluetooth 4.1	Ground Debugging (proximity)
Power	5V, ~350mA active	Non-critical (high-power tasks)
OS	Linux (Raspberry Pi OS)	Runs Python/C++ for imaging/AI

17.1.3ADVANTAGES

- High compute power and Linux ecosystem.
- Good for image processing, ML, and Python scripting.
- Built-in Wi-Fi/Bluetooth.

17.1.4DISADVANTAGES

- Not real-time (no hardware-level timing guarantees).
- High power consumption (~300-400mA).
- Vulnerable to space radiation and temperature fluctuations.
- SD card corruption risk in high-vibration or radiation environments.

17.1.5Alternatives:

Table 16 Raspberry Pi Zero 2 W alternatives

Alternative	Benefit
BeagleBone Black Industrial	Industrial-grade Linux SBC, better I/O protection, supports PRU real-time units.
Raspberry Pi Pico (RP2040)	Lower power, better for real-time tasks, but no Linux.
Teensy 4.1	600MHz, real-time capable with low-latency I/O.
STM32F7/H7	Higher reliability, RTOS support, more rugged for control tasks.
Adafruit Feather M4	ARM Cortex-M4, low power and modular sensor support.
Arduino Due	ARM Cortex-M3, better than Uno/Nano for real-time performance.

17.2 MSP430 (TI)

17.2.1DIAGRAM

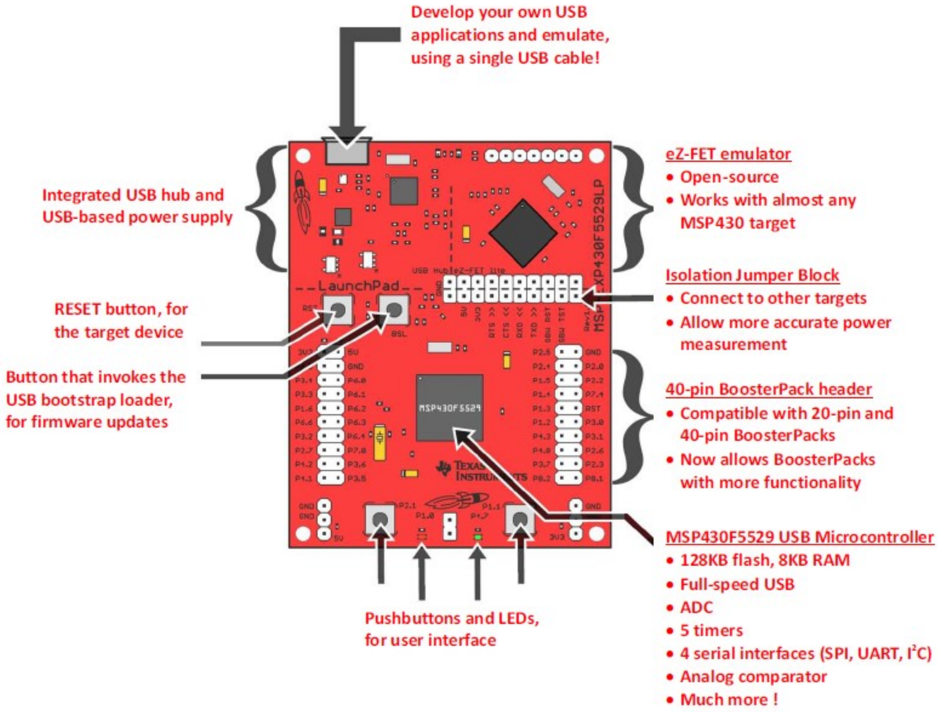


Figure 32 MSP430 DIAGRAM

17.2.2SPECIFICATIONS

- **Architecture:** 16-bit RISC (Harvard architecture)
- **Core:** MSP430FRxx (FRAM variants available)
- **Clock Speed:** Up to 25 MHz
- **Memory:** 2 KB – 256 KB Flash/FRAM, 128 B – 32 KB RAM
- **Power Modes:** 5 ultra-low power modes (Sleep current < 0.1 µA)
- **Peripherals:** ADC, DAC, timers, UART, SPI, I2C, watchdog
- **Radiation Resistance:** Some variants offer radiation tolerance (e.g., MSP430FR5969)
- **Programming Languages:** C, Assembly
- **IDE/Tools:** Code Composer Studio (TI), Energia (Arduino-like), IAR Embedded Workbench
- **Strength:** Extremely low power, ideal for always-on monitoring
- **Weakness:** Limited processing power and RAM for complex tasks

Table 17MSP430 specifications

Feature	Specification	Satellite Role
Architecture	16-bit RISC (MSP430FRxx for FRAM)	Power Management Unit (PMU)
Power	0.1µA (sleep), ~100µA/MHz (active)	Always-on sensor monitoring
Radiation	Some variants radiation-hardened	Reliable in harsh environments
Memory	2KB–256KB Flash	Small RTOS or bare-metal code

17.2.3MSP430 MODELS

			10–24				
			16–48			– 3	
			48–83			– 7	
			12–32			– 3	
			32–83			– 7	

			– 4	

8–3.6 V			

operation with aggressive sleep modes (LPM0–LPM4).

GPIO & Connectivity Comparison Table

(SPI, UART & I2C)			
	✕	✕	✕

17.2.4 ADVANTAGES

- Extremely low power consumption (ideal for always-on sensor nodes).
- Fast wake-up and sleep cycles.
- Some radiation-hardened versions exist.
- Stable analog performance.

17.2.5 DISADVANTAGES

- Very limited computing power and memory.
- Fewer peripheral options than modern MCUs.
- Smaller developer ecosystem compared to ARM-based chips.

Pros:

1. **Low Power Consumption**: Critical for power-limited spacecraft.
2. **Fast Wake Time**: Useful for event-driven architectures.
3. **Radiation Tolerance (with Caveats)**:
    - FRAM models (FRxx) offer better resistance to **SEUs** than Flash.
    - Lower complexity reduces susceptibility.

4. **Simple Architecture**: Useful for hardened or fault-tolerant design.
5. **TI's Space-Grade MSP430F2618S-HT**: Available for Hi-Rel systems.

Cons:

1. **No Built-In Radiation Hardening (except space-grade variants)**:
   - Requires TMR (Triple Modular Redundancy) or watchdogs for reliability.
2. **16-bit CPU Limitations**:
   - May be underpowered for heavy OBC tasks (e.g., attitude control algorithms).
3. **Lack of Native CAN/RS-485/SpaceWire**:
   - Requires external transceivers or controllers.
4. **Memory Limitations**:
   - Larger OBCs or payloads (e.g., cameras, telemetry logs) may exceed onboard FRAM/Flash.
5. **No MMU or OS Support**:
   - Real-time OS like FreeRTOS can be run, but no MMU or virtualization support.

Best Fit Use Cases in a Satellite

Role	Feasibility	Notes
OBC	Moderate	Good for CubeSats with simple tasks. FRAM-based models recommended.
Telemetry Subsystem	High	Ideal due to low power, ADC support, and serial/I2C interfaces.
Power Management Unit (PMU)	High	ADCs, timers, and low power make it ideal.
Attitude Control	Low	Lacks DSP/FPU needed for control algorithms.
Communications Handling	Moderate	Needs external RF modules; SPI/UART manageable.
Thermal Monitoring	High	Works well with analog sensors, thermistors.

17.2.7 ALTERNATIVES:

Table 18 MSP430 alternatives

Alternative	Benefit
Ambiq Apollo4	Cortex-M4F ultra-low-power MCU with better performance.
STM32L4	Low-power STM32 series with richer features than MSP430.
ATtiny85/84	Compact, ultra-low-power 8-bit alternative for simple I/O tasks.
Arduino Nano Every	Affordable and low-power for basic tasks.
Adafruit Feather (M0)	Low-power Cortex-M0+ with great modular sensor support.

17.3 TIVA C SERIES LAUNCHPAD

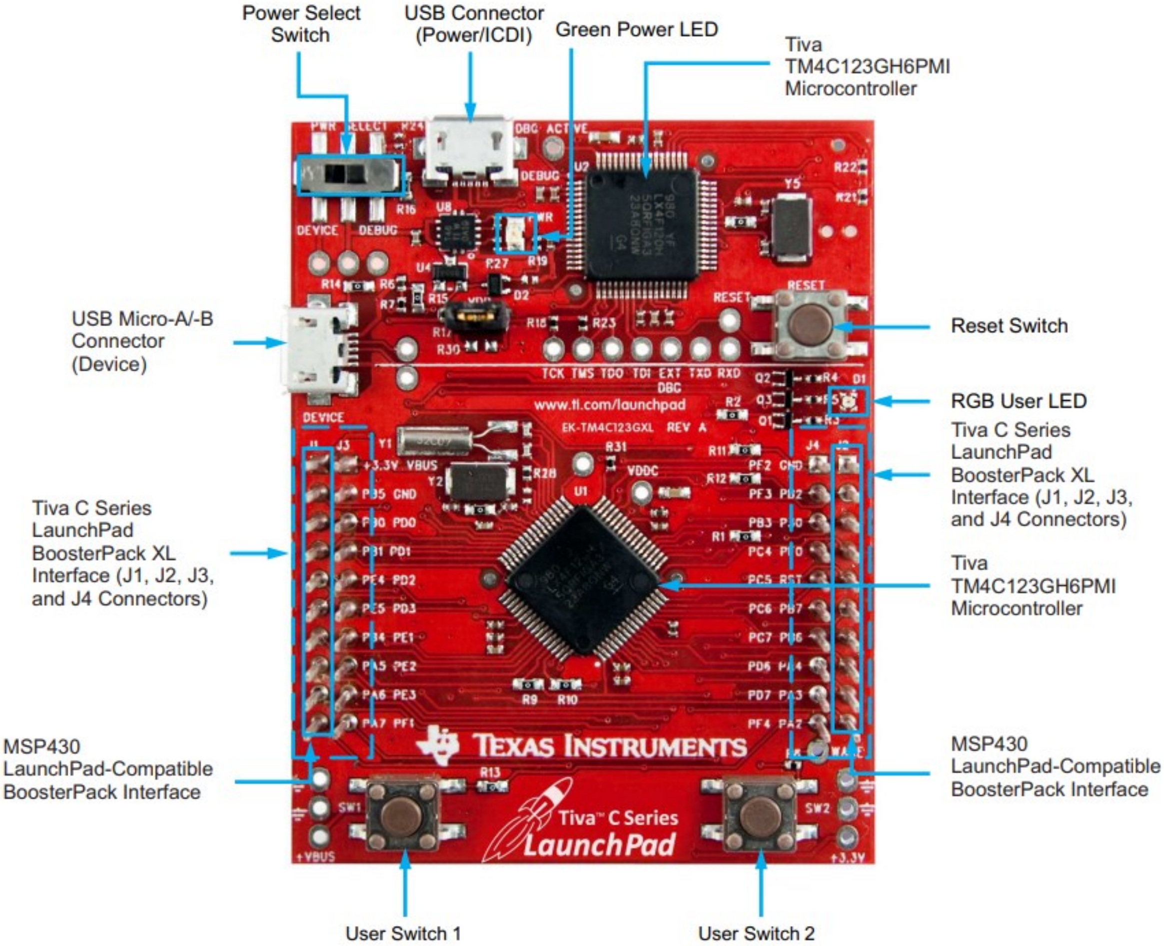


Figure 33 TIVA C SERIES LAUNCHPAD DIAGRAM

17.3.1SPECIFICATIONS

- **Architecture:** 32-bit ARM Cortex-M4F (Floating Point Unit + DSP)
- **Clock Speed:** Up to 80 MHz (TM4C123), 120 MHz (TM4C129)
- **Memory:** 256 KB – 1 MB Flash, 32 KB – 256 KB SRAM
- **Interfaces:** UART, SPI, I2C, USB, CAN, Ethernet, PWM, ADC
- **Real-Time Support:** FreeRTOS, TI-RTOS (deterministic multitasking)
- **Power Consumption:** ~1 mA/MHz (active)
- **Programming Languages:** C, C++
- **IDE/Tools:** Code Composer Studio (TI), Keil µVision, IAR Embedded Workbench
- **Strength:** Real-time control, advanced I/O, suitable for ADCS and inter-microcontroller coordination

- **Weakness:** Higher power consumption than MSP430, more complex to develop

Table 19 TIVA C SERIES LAUNCHPAD specifications

Feature	Specification	Satellite Role
CPU	80 – 120MHz Cortex-M4F (FPU/DSP)	Attitude Control (ADCs)
Memory	256KB – 1MB Flash	Real-time control loops
Interfaces	CAN, Ethernet, USB	Inter-subsystem communication
Power	~1mA/MHz	Balanced performance/power

17.3.2 ADVANTAGES

- Cortex-M4F core supports DSP and floating-point ops.
- Real-time capability with multiple I/O protocols.
- Good balance between performance and power.
- Ideal for control systems and subsystem coordination.

17.3.3 DISADVANTAGES

- Less mainstream support compared to STM32 or ESP32.
- Slightly higher power draw than newer Cortex-M devices.
- Toolchain (Code Composer Studio) can be less flexible than open-source alternatives.

17.3.4ALTERNATIVES:

Alternative	Benefit
STM32F4/F7/H7 Series	Rich ecosystem, real-time ready, industry standard for embedded systems.
Teensy 4.0/4.1	Excellent speed (600 MHz), compact form factor, great RT control.
Arduino Due	Cortex-M3 with plenty of I/O and better community support.
PIC32MX/PIC32MZ	Real-time capable and stable performance with legacy support.
BeagleBone Black	Good for hybrid real-time + Linux-based subsystems.

Table 17. TIVA C SERIES LAUNCHPAD alternatives

COMPARISON OF RASPBERRY PI ZERO 2 W, MSP430 AND TIVA C

Recommended Sensors & Communication Modules

2.1. Sensors

Sensor	Purpose	Best MCU	Interface
BME280	Temp/pressure/humidity	MSP430	I2C/SPI
MPU-6050	Gyroscope/accelerometer (ADCS)	Tiva C	I2C
HMC5883L	Magnetometer (orientation)	Tiva C	I2C
Raspberry Pi Camera	Earth imaging	RPi Zero W	CSI-2
INA219	Power monitoring	MSP430	I2C

Table 18. Recommended Sensors & Communication Modules for RASPBERRY PI ZERO 2 W, MSP430 AND TIVA C

2.2. Communication Modules

Module	Protocol	Best MCU	Use Case
RFM95W (LoRa)	LoRa	Tiva C	Long-range telemetry
AX.25 Modem	UHF/VHF	RPi Zero 2 W	Amateur radio downlink
CAN Bus (SN65HVD23)	CAN	Tiva C	Reliable intra-satellite comms

Table 19. Communication Modules for RASPBERRY PI ZERO 2 W, MSP430 AND TIVA C

3. Programming and Integration

A detailed breakdown of the **Raspberry Pi Zero 2 W**, **MSP430**, and **Tiva C Series LaunchPad** focusing on their *programming languages*, *architecture*, *IDEs/software*, and *efficiency* in satellite applications:

3.1. Raspberry Pi Zero 2 W

Category	Details
Architecture	64-bit Quad-Core ARM Cortex-A53 (1GHz)
OS	Raspberry Pi OS (Debian-based Linux)
Programming Languages	Python, C/C++, Bash, Rust, Go
IDEs & Tools	VS Code (via SSH or remote), Thonny, Geany, nano/vim

Software Support	OpenCV, TensorFlow Lite, GStreamer, ROS, WiringPi, pigpio, I2C-tools
Efficiency in Satellite Applications	
- Power Usage	~0.6W–1.5W depending on load
- Real-time Capability	No true real-time OS; soft real-time only
- Best for	Payload processing, image compression, AI inference, ground debugging
- Limitations	Not radiation-hardened, sensitive to SEUs, requires robust power management and watchdogs

Table 20. Raspberry Pi Zero 2 W programming

3.2. MSP430 (Texas Instruments)

Category	**Details**
Architecture	16-bit RISC (Ultra-low-power, some FRAM variants)
Programming Languages	C, Assembly
IDEs & Tools	Code Composer Studio (CCS), IAR Embedded Workbench
Software Support	Energia (Arduino-like), MSP430 DriverLib, Grace GUI
Efficiency in Satellite Applications	

- Power Usage	Sleep: 0.1 µA, Active: ~100 µA/MHz
- Real-time Capability	Basic bare-metal or small RTOS
- Best for	Sensor polling, power management, watchdog supervision
- Radiation Suitability	Some variants more tolerant (FRAM > Flash), used in CubeSat PMUs
- Limitations	Limited computational power, small RAM/Flash, no FPU

Table 21. MSP430 programming

3.3. Tiva C Series (ARM Cortex-M4)

Category	**Details**
Architecture	32-bit ARM Cortex-M4F (with FPU and DSP)
Programming Languages	C, C++
IDEs & Tools	Code Composer Studio (CCS), Keil µVision, IAR Embedded Workbench
Software Support	TI-RTOS, FreeRTOS, TivaWare, CMSIS
Efficiency in Satellite Applications	

- Power Usage	~1–2 mA/MHz active, very efficient for Cortex-M
- Real-time Capability	Excellent – hardware deterministic with RTOS
- Best for	Real-time attitude control, reaction wheel control, CAN Bus
- Radiation Suitability	Not radiation-hardened, needs shielding or watchdogs
- Limitations	Requires tighter thermal and EMI controls in space environments

Table 22. Tiva C Series programming

Features Summary Table

Feature	**RPi Zero 2 W**	**MSP430**	**Tiva C Series**
CPU Arch	ARM Cortex-A53	16-bit RISC	ARM Cortex-M4F
OS	Linux (Debian)	None / Bare-metal	RTOS (TI-RTOS/FreeRTOS)
Main Languages	Python, C/C++, Bash	C, Assembly	C, C++
IDE	VS Code, Thonny	Code Composer Studio	CCS, Keil, IAR
Real-time Operation	✕ (soft only)	✓ (bare-metal)	✓✓ (RTOS with FPU)

Power Efficiency	Moderate (0.6–1.5W)	Excellent (µA range)	Very good (mA/MHz)
Best For	AI, Imaging, Downlink	Sensing, PMU, Sleep operations	ADCS, Control loops
Radiation Tolerance	Low	Medium (FRAM models)	Low–Medium (needs shielding)
Satellite Role	Payload Processing	Power & Sensor Monitor	ADCS & Intra-Sat Control

Table 23. Summary of features for RASPBERRY PI ZERO 2 W, MSP430 AND TIVA C

5. Prices and Shop Locations

Raspberry PI Zero 2 W
https://shop.ivyliam.com/product/raspberry-pi-zero-2-w/
Ksh 3,800

MSP430 (TI)
https://www.jumia.co.ke/generic-launchpad-development-board-ti-msp430-msp-exp430g2-277243100.html
Ksh 3,805

Tiva C Series LaunchPad
https://store.nerokas.co.ke/SKU-3330
Ksh 3,200

Table 24. RASPBERRY PI ZERO 2 W, MSP430 AND TIVA C prices

18. RASPBERRY PI 1

The Raspberry Pi 1, launched in 2012 by the Raspberry Pi Foundation, was a seminal device in the single-board computer (SBC) revolution. This series includes the original Model A and Model B, followed by improved Model A+ and Model B+ variants. While all share the same core Broadcom BCM2835 SoC, differences in RAM, connectivity, form factor, and power consumption distinguish them. Primarily conceived as an affordable educational tool to promote computer science learning in schools, its low cost, open nature, and versatile General Purpose Input /Output (GPIO) pins quickly saw its adoption by hobbyists, makers, and researchers for a wide array of projects. While often colloquially referred to as a "microcontroller" due to its embedded project usage, the Raspberry Pi 1 is a fully-fledged computer capable of running a Linux operating system, offering significantly more processing power and software flexibility than traditional microcontrollers like Arduinos.

This review will detail the specific design specifications and capabilities of each Raspberry Pi 1 model, analyse their respective advantages and disadvantages, and critically assess their suitability for various satellite subsystems, a domain where Commercial-Off-The-Shelf (COTS) components are increasingly considered for cost reduction in non-critical or research-oriented missions.

18.1 DIAGRAMS AND SCHEMATICS

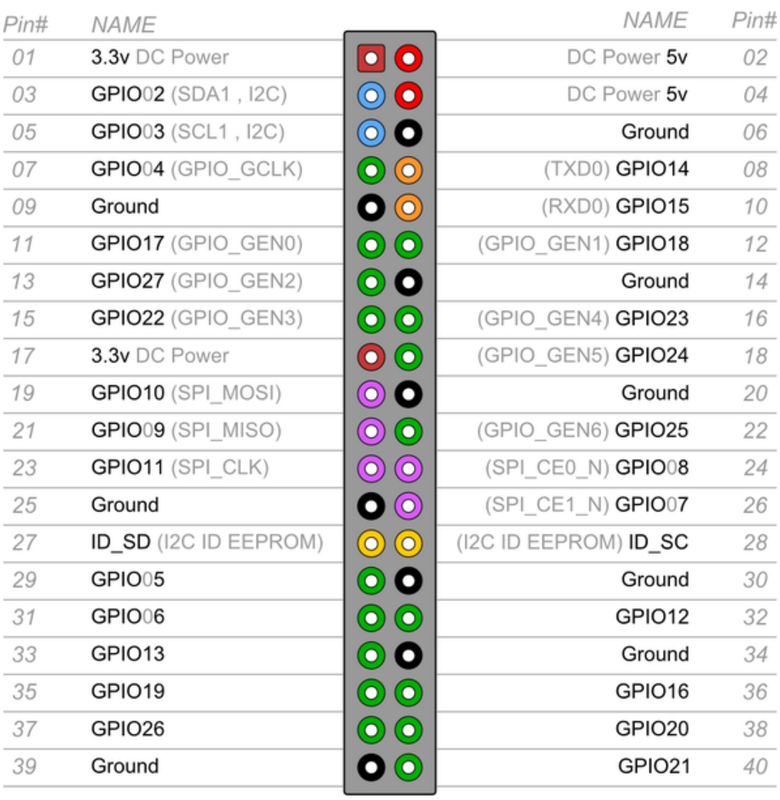


Figure 34 : Raspberry Pi 1 pin out (Model A+ and B+ 40 pin header)

Figure 35 : Raspberry pi 1 (Model A/B 26 pin header

Figure 36 Raspberry Pi 1 board layout (Model A+)

18.2 SPECIFICATIONS, CAPABILITIES, AND CHARACTERISTIC

Core Common Architecture (All Raspberry Pi 1 Models)

- **System on Chip (SoC):** Broadcom BCM2835. This integrates:
  - **CPU:** A single-core ARM1176JZF-S processor typically clocked at 700MHz. This ARMv6 architecture was pivotal for running full Linux distributions.
  - **GPU:** A Broadcom Video Core IV GPU, capable of Blu-ray quality H.264 playback at 1080p30 and supporting OpenGL ES 2.0.
- **Operating System Support:** Primarily designed to run Linux distributions, with "Raspbian" (a Debian derivative optimized for Raspberry Pi hardware) being the officially supported OS. Other options include Arch Linux ARM, RISC OS.
- **Video Output:** HDMI output is standard. Composite video output is also available (via RCA on earlier models, TRRS jack on later ones).
- **Audio Output:** Analog audio via a 3.5mm jack and digital audio via HDMI.
- **Power Input:** 5V DC via Micro USB connector.
- **Fundamental Capabilities (derived from common core):**
  - Ability to run a desktop-like graphical environment (though performance-limited).
  - Execution of applications written in high-level languages (Python, C/C++, Perl, Java etc.).
  - Basic networking (model-dependent or via USB dongles).
  - Interfacing with external electronics through GPIO pins.

Model-Specific Specifications and Characteristics

3.2.1. Raspberry Pi Model A

- **RAM:** 256MB SDRAM (shared between CPU and GPU).
- **USB Ports:** 1x USB 2.0 port.
- **Ethernet:** None.
- **GPIO:** 26-pin header (P1).
- **Storage:** Full-size SD card slot.
- **Power Consumption:** Lowest of the initial models, typically around 1-1.5W under load.
- **Dimensions:** 85.60 mm × 56.5 mm.

- **Characteristics:** Designed as the lowest-cost variant. Its minimal peripheral set targeted applications where power and cost were paramount over connectivity. The single USB port and lack of Ethernet severely limited its out-of-the-box connectivity.

3.2.2. Raspberry Pi Model B (Original and Revision 2.0)

- **RAM:**
  - Original Model B: 256MB SDRAM (shared).
  - Model B Revision 2.0 (released October 2012): Upgraded to 512MB SDRAM (shared). This upgrade significantly enhanced usability.
- **USB Ports:** 2x USB 2.0 ports.
- **Ethernet:** 10/100 Mbps Ethernet port. Notably, the Ethernet controller (Microchip LAN9512) was connected via the internal USB bus, meaning USB and Ethernet traffic shared bandwidth, a known performance bottleneck.
- **GPIO:** 26-pin header (P1). Revision 2.0 boards added an unpopulated 8-pin P5 header providing access to four more GPIO lines.
- **Storage:** Full-size SD card slot.
- **Power Consumption:** Higher than Model A, typically around 2.5-3.5W under load.
- **Dimensions:** 85.60 mm × 56.5 mm.
- **Characteristics:** The "standard" initial model, offering more connectivity (Ethernet, dual USB) and, with Rev 2.0, more usable RAM. This made it more suitable for general-purpose tinkering and projects requiring network access.

3.2.3. Raspberry Pi Model A+

- **RAM:** 256MB SDRAM (shared). Some sources suggest BCM2835 batches used could support 512MB, but 256MB was the advertised standard.
- **USB Ports:** 1x USB 2.0 port.
- **Ethernet:** None.
- **GPIO:** Expanded to a 40-pin header, with the first 26 pins maintaining compatibility with Model A/B.
- **Storage:** Switched to a MicroSD card slot.
- **Power Consumption:** Lowest of all Pi 1 models, typically around 0.7-1.2W, due to more efficient power circuitry.
- **Dimensions:** Smaller form factor: 65 mm × 56.5 mm.

- **Characteristics:** An evolution of Model A, focusing on reduced size, lower power consumption, and an expanded GPIO header. It also featured improved audio circuitry. The move to MicroSD was in line with market trends.

3.2.4. Raspberry Pi Model B+

- **RAM:** 512MB SDRAM (shared).
- **USB Ports:** Increased to 4x USB 2.0 ports. Power delivery to USB ports was also improved.
- **Ethernet:** 10/100 Mbps Ethernet port (still using a USB-connected controller, Microchip LAN9514, an evolution of the LAN9512).
- **GPIO:** Expanded to a 40-pin header, with the first 26 pins maintaining compatibility.
- **Storage:** Switched to a MicroSD card slot.
- **Power Consumption:** More efficient than the original Model B, typically around 2-3W under load, despite more USB ports.
- **Dimensions:** Similar footprint to Model B (85 mm × 56.5 mm), but with a neater port layout and four mounting holes.
- **Characteristics:** A significant refinement of the Model B. It offered much-improved I/O with four USB ports, the expanded GPIO, better power management, and a more practical layout. This became the de standard Pi 1 for more complex projects prior to the Pi 2.

18.3 ADVANTAGES AND DISADVANTAGES

General Advantages (Applicable to all Raspberry Pi 1 Models)

- **Full Linux Operating System:** Provided a familiar, powerful, and highly extensible software environment with access to vast repositories of open-source tools and libraries.
- **Versatile GPIO Interface:** Enabled direct interaction with external hardware, sensors, and actuators, bridging software with the physical world. The 40-pin header on A+/B+ was a notable improvement.
- **Strong Community Support:** A massive global community rapidly emerged, producing extensive documentation, tutorials, software, and troubleshooting support.
- **Multimedia Capabilities:** The VideoCore IV GPU offered respectable multimedia decoding for its class.

General Disadvantages (Critical for Satellite Applications)

- **Lack of Radiation Hardening:** As COTS consumer electronics, they possess no inherent protection against space radiation (Single Event Upsets - SEUs, Single Event Latch-ups - SELs, Total Ionizing Dose - TID effects), making them highly prone to malfunction or permanent damage in orbit.
- **SD Card Reliability:** The reliance on SD/MicroSD cards for boot and primary storage is a major point of failure. These cards are not designed for the write endurance, temperature extremes, or radiation environment of space, and are prone to data corruption.
- **Non-Real-Time Operating System:** Standard Linux distributions are not Real-Time Operating Systems (RTOS). This makes achieving deterministic timing for critical control loops (e.g., in ADCS or EPS) extremely challenging without specialized kernels (e.g., PREEMPT_RT patch), which add complexity and may not be fully validated for the BCM2835 (Jones, 2014).
- **Limited Processing Performance:** The single-core 700MHz ARM1176JZF-S CPU struggled with computationally intensive tasks, heavy multitasking, or modern complex software.
- **Thermal Management in Vacuum:** Consumer electronics rely on convective cooling. In vacuum, heat dissipation becomes a significant engineering challenge requiring dedicated conductive thermal pathways not present on the bare boards.
- **Commercial-Grade Components & Environmental Tolerance:** Connectors, solder joints, and components are not specified or tested for the vibration, shock, outgassing, and extreme temperature cycles of launch and space operation.
- **Power Consumption (Relative to Space-Grade Microcontrollers):** While low for a computer, their power draw (0.7W to 3.5W) is considerably higher than dedicated, low-power space-grade microcontrollers.

Model-Specific Advantages/Disadvantages

- **Model A/A+:**
    - ***Advantage:*** Lower power consumption is appealing for severely power-constrained scenarios. Model A+ had the smallest form factor.
    - ***Disadvantage:*** Limited RAM (256MB) is highly restrictive. Single USB port and no Ethernet severely limit connectivity without add-ons.
- **Model B/B+:**
    - ***Advantage:*** More RAM (512MB on Rev2.0 B and B+) provides better performance. Integrated Ethernet and more USB ports (especially the 4 on B+) offer superior connectivity. Model B+ had improved power circuitry and layout.
    - ***Disadvantage:*** Higher power consumption than A/A+. Shared USB/Ethernet bus on B/B+ can be a bottleneck.

18.4 SUITABILITY IN SATELLITE SUBSYSTEMS

The assessment of suitability is based on the general advantages and disadvantages of the Raspberry pi 1 series. Similarly, the different subsystems are considered.

Telemetry, Tracking, and Command (TT&C)

- **Role:** Handles vital two-way communication with ground control, processing commands, and transmitting satellite health and payload data.
- **Raspberry Pi 1 Model Suitability:**
    - While any model could technically interface with a radio transceiver via GPIO (UART, SPI), the risk of SD card corruption leading to loss of command reception or telemetry transmission is mission-ending.
    - The non-RTOS nature could lead to missed or mistimed commands.
    - Radiation-induced errors in the CPU or RAM during critical TT&C operations are catastrophic. No specific model offers a meaningful advantage given these fundamental flaws.

On-Board Computer (OBC) / Command and Data Handling (C&DH)

- **Role:** The central "brain" of the satellite, managing operations, scheduling tasks, processing payload data, storing data, and coordinating other subsystems.
- **Raspberry Pi 1 Model Suitability:**
    - The OBC is arguably the most critical subsystem. All the general disadvantages (radiation, SD card reliance, lack of RTOS, thermal) apply with extreme prejudice.
    - The 512MB RAM of the Model B (Rev 2.0) and Model B+ offers slightly more headroom for software than the 256MB models (A, A+), but this is insignificant compared to the overwhelming reliability risks.
    - For highly experimental, non-critical secondary data logging on a CubeSat with accepted high failure risk, a Model A+ (for low power) or B+ (for more connectivity to sensors) *might* be considered.

Electrical Power System (EPS) Control

- **Role:** Manages power generation (solar panels), storage (batteries), and distribution; monitors voltages, currents, and temperatures; controls charging and switching.
- **Raspberry Pi 1 Model Suitability:**
    - EPS is life-critical for the satellite. An RPi1's unreliability makes it an unacceptable choice for managing power distribution or battery charging. The non-RTOS nature is problematic for precise charging algorithms or load shedding responses.
    - Model A+ offers the lowest power draw, which is a minor consideration against the backdrop of unreliability.

Communication (Payload Data Downlink / Inter-Satellite Links)

- **Role:** Transmitting payload science data (often high volume) or establishing links with other satellites.
- **Raspberry Pi 1 Model Suitability:**
    - The CPU performance of all RPi1 models is insufficient for complex Software Defined Radio (SDR) tasks, error correction coding, or high-bandwidth modulation/demodulation.

- For very low-data-rate, non-critical tasks like basic beaconing with a simple transceiver module (e.g., LoRa, via UART/SPI), any model *could* technically drive the module. Models B/B+ with Ethernet could simplify ground testing and integration with IP-based radio front-ends.

Ground Stations

- **Role:** Facilitates communication with the satellite from Earth, including antenna tracking, signal reception/demodulation, command uplink, and data display/archival.
- **RPi 1 Model Suitability:**
    - Here, the harsh space environment is absent. The low cost and Linux environment are advantageous.
    - **Model B+:** Its 512MB RAM, 4 USB ports (for keyboard, mouse, SDR dongle, other peripherals), and Ethernet make it the most suitable Pi 1 for amateur or educational ground station controllers, telemetry decoders, or simple data servers.
    - **Model B (Rev 2.0):** A viable, but less convenient, alternative due to fewer USB ports.
    - **Model A/A+:** Severely limited by single USB port and no Ethernet, making them impractical for most ground station setups without significant additional hub hardware.
    - Even here, performance for demanding SDR processing or handling multiple high-data-rate streams would be a limitation.

Attitude Determination and Control System (ADCS)

- **Role:** Determines and controls the satellite's orientation using sensors (sun sensors, magnetometers, gyros, star trackers) and actuators (reaction wheels, magnetorquers). Requires precise, real-time calculations and control loops.
- **Raspberry Pi 1 Model Suitability:**
    - ADCS is a highly critical, real-time control system. The non-RTOS nature of Linux on the RPi1, its limited processing power for complex navigation and control algorithms (e.g., Kalman filtering), and its inherent unreliability make it entirely inappropriate. Failure of the ADCS can lead to loss of communication, power

generation, or mission objectives. The 40-pin GPIO on A+/B+ offers more sensor connectivity points, but the core system is inadequate for the task.

18.5 CONCLUSION

The Raspberry Pi 1 family (Models A, B, A+, B+) was a landmark achievement in terrestrial low-cost computing, fostering a generation of makers and learners. However, when evaluated against the stringent requirements of satellite subsystems, the models exhibit fundamental deficiencies that render them less suited for critical on-orbit applications. The lack of radiation hardening, reliance on unreliable SD card storage, the non-real-time nature of their typical operating system, limited processing power, and commercial-grade environmental tolerances are disqualifying factors for roles demanding high reliability and robust performance in the space environment.

19. RASPBERRY PI 2

The Raspberry Pi 2 Model B is a compact, cost-effective, and versatile single-board computer developed by the Raspberry Pi Foundation. It is equipped with a 900 MHz quad-core ARM Cortex-A7 processor, 1 GB of LPDDR2 RAM, and a 40-pin GPIO header, providing a strong foundation for running general-purpose software, controlling hardware interfaces, and performing moderate levels of computation. Originally designed to promote computer science education, the Raspberry Pi 2 has become widely adopted in hobbyist projects, including robotics, IoT, and space-related experiments.

For a hobby satellite project, the Pi 2 offers several compelling capabilities, especially for processing and data management tasks. However, deploying it in space introduces engineering challenges related to its environmental durability, power efficiency, and real-time control limitations, which must be weighed carefully.

Below is an image of the Raspberry pi 2 board:

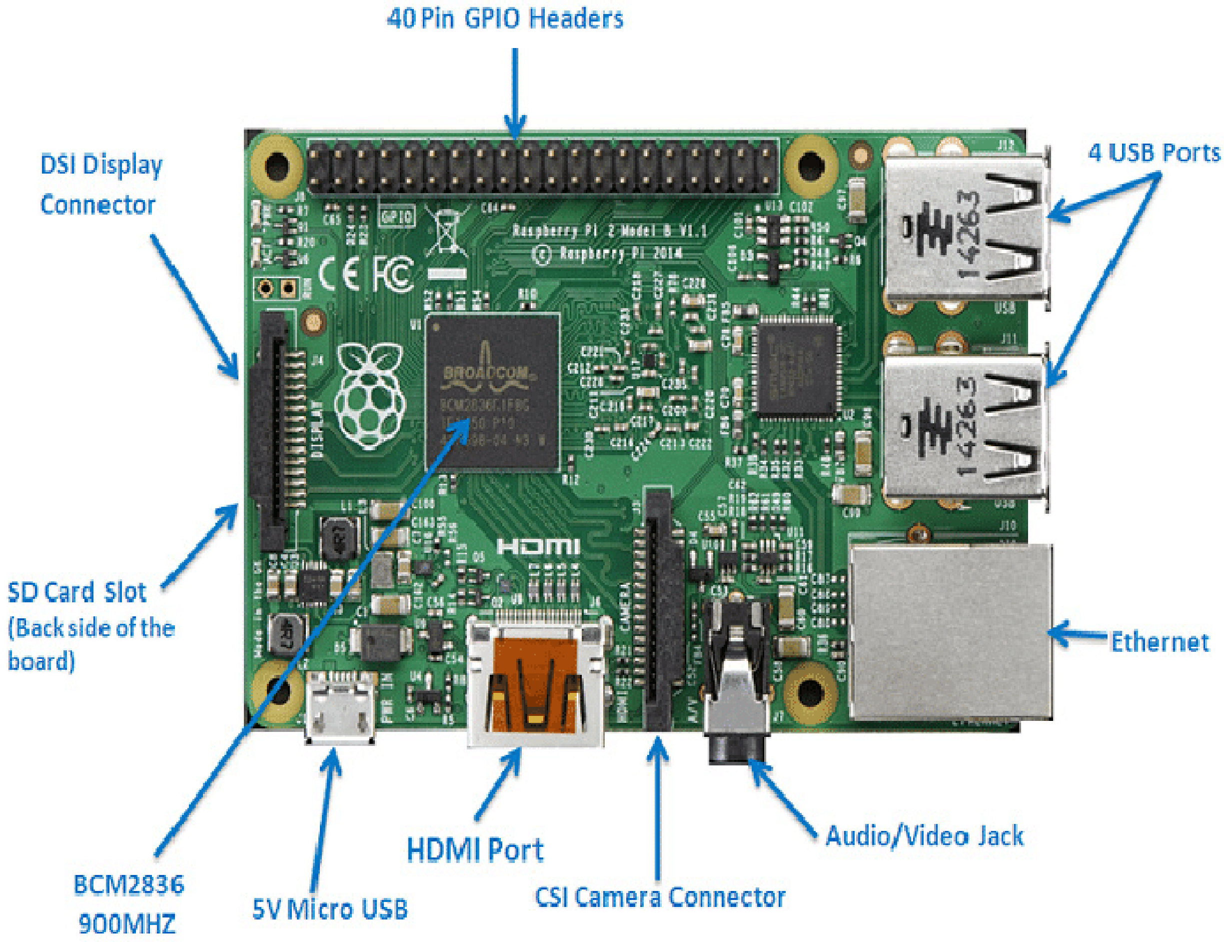


Figure 37 Raspberry pi 2

19.1 KEY FEATURES

- **Quad-Core ARM Processor**: The 900 MHz Cortex-A7 processor can run multitasking applications and handle moderate computational loads. This makes it capable of performing edge processing tasks such as image filtering, compression, or telemetry formatting.
- **1 GB RAM**: With 1 GB of RAM, the Pi 2 can manage larger data buffers, run multiple processes simultaneously, and support more demanding software packages compared to microcontrollers.
- **Linux Operating System Support**: The Pi 2 can run full Linux distributions, like Raspberry Pi OS (formerly Raspbian), Ubuntu MATE, or specialized OS images, allowing use of high-level programming languages (Python, C/C++, Java), libraries (OpenCV, NumPy), and software packages for networking and data processing.
- **GPIO and Peripheral Support**: The 40-pin GPIO header allows easy connection to external sensors, actuators, and communication modules (e.g., GPS, gyroscopes, RF transmitters), giving users direct control over hardware.
- **Camera and Display Interfaces**: With a CSI (Camera Serial Interface) port and DSI (Display Serial Interface), it can interface with the Raspberry Pi Camera Module for onboard imaging, and output data to a screen for debugging or ground-based visualization.

19.2 ADVANTAGES

1. **High Processing Power**: Compared to traditional microcontrollers, the Raspberry Pi 2 offers significantly more processing power. This allows it to run image-processing algorithms, compress large amounts of sensor data, or even perform light machine learning inference tasks, which are often useful in satellite-based Earth observation or autonomous onboard decision-making.
2. **Full Linux Environment**: Running a full OS enables developers to use familiar tools and languages, set up cron jobs, perform parallel processing, and use open-source libraries. It also supports multitasking and high-level abstraction, which simplifies development and debugging.
3. **Camera and Imaging Capability**: The Pi 2 can connect to a high-definition camera module (e.g., 5MP Pi Camera) and process or store images locally. This is especially useful in CubeSat or CanSat missions that require visual monitoring or Earth imaging.

4. **Rich Peripheral Ecosystem**: With USB, UART, I2C, and SPI support, it can interface with many types of sensors and communication modules, making it a flexible platform for experimentation and integration.
5. **Large Developer Community**: There is extensive online support, tutorials, libraries, and community examples available, which greatly reduces the development time and simplifies troubleshooting.

19.3 DISADVANTAGES

1. **Not Radiation-Hardened**: Since Raspberry Pi 2 is a commercial off-the-shelf (COTS) product, it lacks any radiation tolerance. In space, cosmic rays can cause bit-flips or logic errors, especially in RAM and storage, which may crash or corrupt the system.
2. **High Power Consumption**: Drawing about 700–900 mA at 5V, the Pi 2 is relatively power-hungry for a small satellite system. This increases the demand on the power subsystem, which is usually limited in both solar panel area and battery size.
3. **Thermal Management Concerns**: The Raspberry Pi 2 can overheat under heavy load or in poorly ventilated enclosures. In orbit, the absence of convection cooling makes thermal control difficult, and overheating could affect reliability or lifespan.
4. **No Real-Time Operation**: Linux is not a real-time operating system (RTOS), so tasks like motor control or precise sensor sampling cannot be guaranteed to occur at fixed intervals. This limits its usefulness for flight-critical timing-sensitive tasks.
5. **Storage Fragility**: The microSD card used for booting and storage is vulnerable to corruption from sudden power loss or radiation. If the card fails, the system may become completely unbootable unless redundancy is built in.

19.4 BEST USE CASES IN A HOBBY SATELLITE

- **Ground-Based Testing and Development**: Before deploying to space, the Raspberry Pi 2 is excellent for developing and testing satellite software on the ground. Its Linux OS and GPIO interface make it ideal for simulating sensor inputs and outputs, testing communication protocols, and visualizing telemetry data.
- **Onboard Imaging and Data Collection**: The Pi 2 can be used to control a camera module for capturing images during flight. It can also apply basic image processing (like resizing, compressing, or adding timestamps) before storing the data or sending it to a ground station.

- **Telemetry Formatting and Storage**: Another practical use is handling telemetry data from various sensors. The Pi 2 can aggregate, timestamp, and format this data into packets suitable for RF downlink, or temporarily store it in a buffer.
- **Secondary (Non-Critical) Onboard Computer**: In space, it's better suited as a backup or auxiliary processor that performs non-critical tasks, such as running experiments, collecting environmental data, or logging mission metrics — while a more robust microcontroller handles vital controls.
- **Post-Processing Tasks in a High-Altitude Balloon or Suborbital Mission**: For hobbyists testing payloads on weather balloons or suborbital rockets, the Raspberry Pi 2 offers powerful computing for data processing once the payload is recovered.

19.5 COMPARISON WITH OTHER RASPBERRY PI MODELS

1. **Performance**
   The Raspberry Pi 4 is the most powerful, with a 1.5 GHz Cortex-A72 CPU and up to 8 GB RAM, great for multitasking, data processing, and AI. However, it requires more power and cooling. The Pi 3 offers decent performance with a 1.2 GHz CPU and built-in wireless, making it a reliable option for general tasks and onboard computation. Pi 2 is a solid mid-range choice with a 900 MHz quad-core processor. It lacks wireless support but handles basic processing tasks well.
   The Pi Zero 2 W is compact but surprisingly capable, performance is close to the Pi 3, suitable for lightweight applications like telemetry or simple automation.
2. **Power Consumption**
   Pi Zero 2 W and Pi 2 consume less power, making them better suited for power-limited environments like balloon or satellite payloads. Pi 4 is the most power-hungry and generates heat, so it's better for ground-based or short-duration systems.
3. **Connectivity**
   Pi 3, Pi 4, and Zero 2 W have built-in Wi-Fi and Bluetooth, helpful for wireless data links. Pi 2 lacks these features, so USB dongles are needed for wireless communication.
4. **Size and Weight**
   Pi Zero 2 W is the most compact and lightest, ideal for small payloads or tight enclosures. Pi 2, 3, and 4 are larger but offer easier access to ports and more expansion options.
5. **Cost**
   Pi Zero 2 W is the most affordable, followed by Pi 2, then Pi 3. Pi 4 is more expensive, especially with higher RAM.
6. **GPIO and Storage**

All models use the same 40-pin GPIO layout, which ensures hardware compatibility. All boot from microSD cards, but Pi 4 also supports USB booting.

19.6 CONCLUSION

The Raspberry Pi 2 Model B is a flexible and affordable computing platform with several features that make it appealing for student-led or hobbyist satellite missions. While it is not recommended for core flight control or critical operations in space due to its lack of radiation protection and real-time capabilities, it is highly effective as a supporting onboard processor for tasks like data logging, imaging, or communications prep. Used in conjunction with a low-power, real-time microcontroller such as an STM32, it can help offload high-level computation and significantly expand the functionality of a small satellite system.

20. RASPBERRY PI 3

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT.

- The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced, wireless LAN compliance testing, improving both cost and time to market.
- The Raspberry Pi 3 Model B+ maintains the same mechanical footprint as both the Raspberry Pi 2 Model B and the Raspberry Pi 3 Model B.

Figure 38 Raspberry Pi 3

20.1 Specifications

- Processor: Broadcom BCM2837B0, Quad-Core Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz **Clockspeed**

- Memory: 1GB LPDDR2 SDRAM

- Connectivity:

1. 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
2. Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
3. 4 USB 2.0 ports with up to 1.2A output

- Access: Extended 40-pin GPIO header

- Video & sound:

1. Full-size HDMI
2. CSI camera port for connecting a Raspberry Pi camera
3. DSI display port for connecting a Raspberry Pi touchscreen display
4. 4-pole stereo output and composite video port

- Multimedia: H.264, MPEG-4 decode(1080p30); H.264 encode(1080p30); OpenGL ES 1.1,2.0 graphics

- **SD card support:** Micro SD port for loading your operating system and storing data
- **Input power:**
1. 5V/2.5A DC power input
2. 5V DC via GPIO header
3. Power-over-Ethernet (PoE) support (requires separate PoE HAT)
- **Environment:** Operating temperature, 0-50°C

20.2 Pin-out diagram

A Raspberry Pi 3 board has 40 pins on it. Among these pins, we have four power pins on the Raspberry Pi, two of which are 5v pins and another two are 3.3v pins. The 5v power pins are connected directly to the Raspberry Pi's power input and we can use these pins to run low power applications.

Then there are the ground pins. There are eight ground pins and all of these are connected to each other; you can use any of these ground pins for your projects. That leaves us with 28 GPIO pins, labeled starting from GPIO 0 and going up to GPIO 27

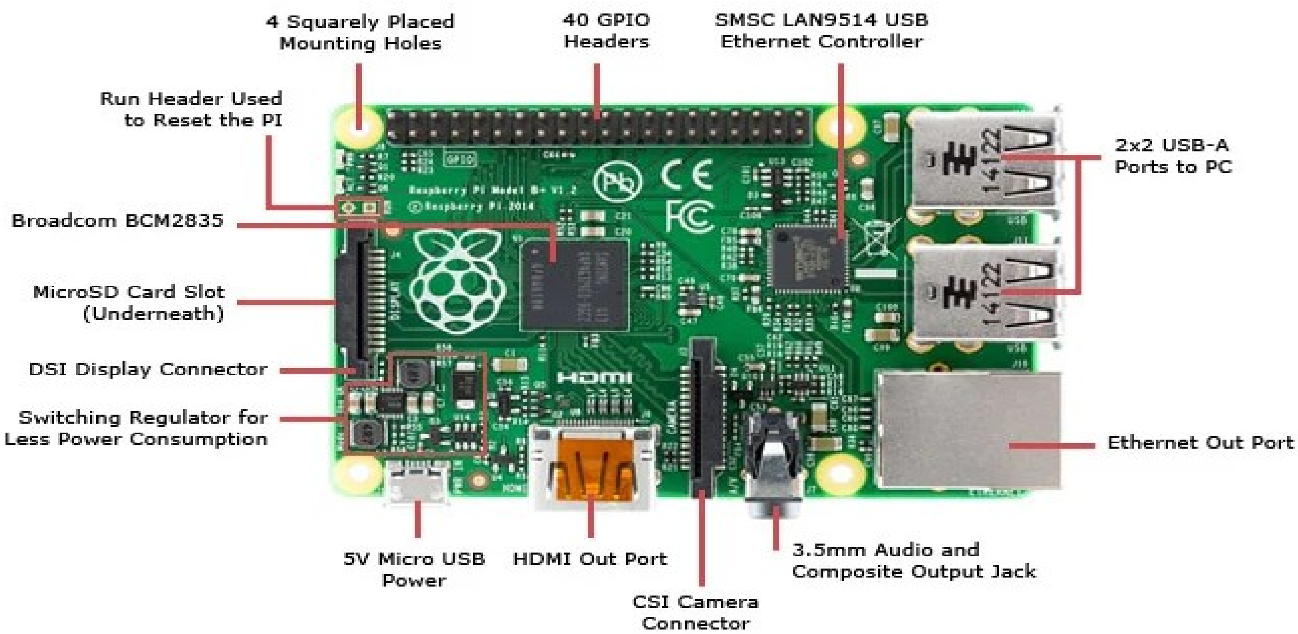


Figure 39 Raspberry Pi 3 features

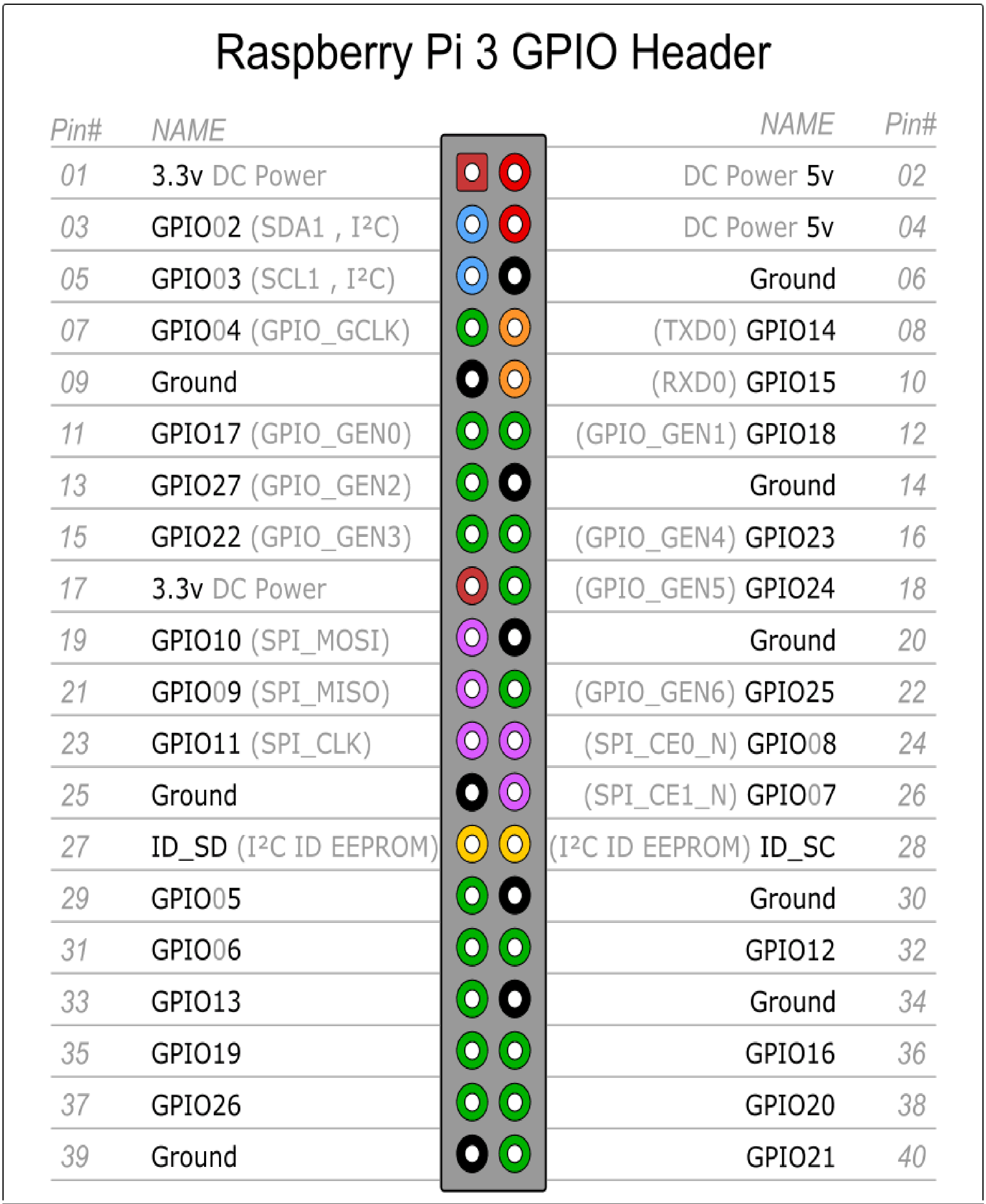


Figure 40 Raspberry pi 3 Pin out

20.3 Communication

The Raspberry Pi 3 Model B+ offers a rich variety of communication interfaces that make it suitable for a wide range of embedded and IoT applications, including satellite system prototypes. One of the most significant communication features is the **40-pin GPIO (General Purpose Input/Output) header**, which supports multiple serial protocols. It includes dedicated pins for **UART (Universal Asynchronous Receiver/Transmitter)** used for serial communication with modules like GPS, GSM, and debug consoles. The board also supports **I²C (Inter-Integrated Circuit)** and **SPI (Serial Peripheral Interface)**, which are essential for communicating with sensors, memory chips, real-time clocks, and other microcontrollers. These protocols enable the Raspberry Pi to gather and transmit data from various onboard or external devices efficiently.

In terms of wireless communication, the Raspberry Pi 3 Model B+ is equipped with **dual-band Wi-Fi (2.4 GHz and 5 GHz)** that adheres to the 802.11 b/g/n/ac standards. This makes it possible to connect the Pi to wireless networks for internet access or to other devices over a LAN. The Wi-Fi capability is particularly valuable for sending telemetry data or remotely monitoring systems during satellite ground testing or lab simulations. Additionally, the board supports **Bluetooth 4.2 and Bluetooth Low Energy (BLE)**. BLE is especially useful for low-power, short-range communication with mobile devices, wearable sensors, or nearby embedded systems during configuration and testing. Bluetooth also enables wireless debugging or data transfer in areas without LAN infrastructure.

For wired communication, the Raspberry Pi 3 Model B+ includes a **Gigabit Ethernet port**, though it is limited to USB 2.0 speeds (~300 Mbps). The Ethernet connection provides a stable and fast data link that is ideal for development environments, remote monitoring setups, or as part of a ground station in satellite projects. Furthermore, the board includes **four USB 2.0 ports**, which support peripheral devices such as USB-to-serial adapters, storage drives, cameras, and communication dongles. These USB ports enhance the Pi's flexibility in connecting to other systems or external interfaces that may be required in a satellite development environment.

In addition to data communication, the Raspberry Pi 3 Model B+ also features a **full-size HDMI port**, **CSI (Camera Serial Interface)**, and **DSI (Display Serial Interface)**. These ports are mainly used for multimedia communication and are helpful for development, debugging, or creating real-time visual interfaces. For instance, the HDMI port can be used to connect a display for monitoring sensor outputs or configuring the Pi without needing remote access. The CSI and DSI ports allow connection to Raspberry Pi-compatible camera and touchscreen display modules, respectively, which are useful in projects that require imaging or user interaction.

Lastly, the Pi 3 Model B+ supports **Power over Ethernet (PoE)** through a 4-pin PoE header. When paired with a PoE HAT (Hardware Attached on Top), the board can receive both power and network data through a single Ethernet cable, simplifying installation in remote or constrained environments. This feature is particularly advantageous for ground-based satellite test setups where multiple devices need centralized power and data lines.

20.4 Power

Powering the Raspberry Pi 3 B+ is easy: just plug any 5V/2.5A USB power supply into the microUSB port. There's no power button, so the RPi will begin to boot as soon as power is applied. To turn it off, simply shut down the Pi 3 B+, then remove power. The four built-in USB ports can even output up to 1.2A, enabling you to connect more power-hungry USB devices.

The top side is painted with metal shielding, instead of plastic in the earlier models, that acts as a heat sink and drains the excessive amount of heat if the board is subjected to the high temperature or pressure

20.5 Programming

The Raspberry Pi 3 Model B+ supports flexible and powerful programming options, making it ideal for embedded systems and satellite-related projects. It functions like a mini-computer and can be programmed in a variety of high-level and low-level languages, with extensive hardware control.

Operating System (OS):

- The Raspberry Pi typically runs Raspberry Pi OS (formerly Raspbian), a Debian-based Linux distribution.
- Other supported OSs include Ubuntu, Windows IoT Core, and lightweight real-time systems (for more control).
- OS is installed on a microSD card, which also stores your programs and files.

Programming Languages:

- Python – The most popular choice due to simplicity and strong GPIO support.
- C/C++ – For low-level hardware control and performance-critical tasks.
- Java, Node.js, Scratch, and Shell scripts are also supported.
- Bash/Shell – Useful for automation and script-based control in Linux.

GPIO Programming:

- You can control digital I/O pins, UART, I²C, SPI, PWM, etc., directly via code.
- Python libraries like RPi.GPIO, gpiozero, or pigpio allow you to:
    - Read sensor data.
    - Control actuators, LEDs, motors.

- Send/receive data from communication modules.

To get started with the **Raspberry Pi 3 Model B+**, you'll need to set it up physically, install an operating system, and run your first program.

Step 1: What You Need

Item	Description
Raspberry Pi 3 Model B+	The main board
microSD card (16GB or more)	Stores the OS and files (Class 10 recommended)
Power supply (5V 2.5A)	Official Raspberry Pi power supply or equivalent
HDMI cable + monitor	For initial setup and interface
USB keyboard and mouse	For setup if not using SSH or remote
Internet (Wi-Fi or Ethernet)	For downloading software updates
Breadboard + jumper wires	For simple hardware tests
LED + 330Ω resistor	For GPIO test program

Step 2: Install the Operating System

1. **Download Raspberry Pi OS**:
    - Download Raspberry Pi Imager for your platform (Windows/macOS/Linux).
2. **Install OS on microSD Card**:
    - Insert your microSD card into your computer.
    - Open Raspberry Pi Imager.
    - Choose OS (e.g., Raspberry Pi OS Lite or Desktop).
    - Choose the SD card.
    - Click **Write**.
3. **Boot Raspberry Pi**:
    - Insert the SD card into the Pi.

- Connect HDMI, keyboard, mouse, and power.
- It should boot into Raspberry Pi OS.
- Configure Wi-Fi, password, and regional settings if prompted.

Step 3: Update & Setup

Step 4: Write Your First Program (LED Blink)

Wiring

- Connect an LED's anode (+) to **GPIO17 (Pin 11)**.
- Connect the cathode to one end of a **330Ω resistor**.
- Connect the other end of the resistor to **GND (Pin 6)**.

Next Steps

You can now:

- Connect **sensors** (like temperature, motion).
- Control **motors** or **relays**.
- Interface with **camera modules**.
- Build **IoT** or **space payload** applications.

20.6 PRICE

Nerokas - Ksh.10,000.00

20.7 ADVANTAGES

Powerful Yet Compact:

- Quad-core ARM Cortex-A53 processor @ 1.4GHz.
- Small form factor (credit card-sized), ideal for space-constrained applications like nanosatellites.

Built-in Wireless Communication:

- Integrated Wi-Fi (802.11 b/g/n/ac) and Bluetooth 4.2 / BLE — no need for external dongles.

GPIO & Peripheral Support:

- 40 GPIO pins supporting I^2C, SPI, UART, PWM — great for sensor and actuator interfacing.

OS and Software Flexibility:

- Can run full Linux distributions, supporting many programming languages and tools.
- Enables advanced software like Python, ROS (for robotics), OpenCV, etc.

Internet & Network Support:

- Has Ethernet port and can support PoE (Power over Ethernet) with an optional HAT.

Multiple USB Ports:

- Four USB 2.0 ports for connecting external devices like storage, keyboard, mouse, GSM module, etc.

Multimedia Support:

- HDMI output, camera interface, and GPU support (useful for image processing tasks).

20.8 DISADVANTAGES

Not Industrial-Grade:

- Not designed for extreme environments (e.g., temperature, radiation, vibration in orbit).
- Limited durability compared to radiation-hardened systems.

Power Sensitivity:

- Requires stable 5V/2.5A power supply.
- Sudden power loss can corrupt the SD card, causing system failure.

Uses SD Card for Storage:

- SD cards are slower and less reliable than SSDs or onboard flash storage.

Limited Security Features:

- Lacks secure boot and hardware encryption compared to some microcontrollers or industrial SBCs.

Not Real-Time:

- Linux-based systems are not deterministic, which can be a challenge for real-time control unless you use RT-kernels.

No Built-in ADC:

- It lacks Analog-to-Digital Converters — external ADCs are needed to read analog sensors.

High Power Consumption (for Embedded):

- Draws more power than simpler microcontrollers like PIC, AVR, or ARM Cortex-M based systems.

20.9 RELEVANCE TO NANO-SATELLITE PROJECT

1. Payload Controller and Data Processor

- The Raspberry Pi 3 Model B+ can **collect, store, and process data** from payload instruments such as:
  - Earth observation cameras via CSI interface.
  - Environmental sensors (temperature, radiation, magnetic field).
  - Custom scientific instruments e.g., spectrometers, microgravity experiments.
- It allows for **real-time data processing**, image pre-processing (cropping, compression), and intelligent decision-making using Python or C++.

ii. Image and Video Acquisition for EO Missions

- Supports **Pi Camera Modules**, which are lightweight and integrate directly through the CSI interface.
- The Pi can capture high-resolution images and **compress or analyze them onboard**, reducing the need to transmit full raw data, saving bandwidth.

iii. Data Storage and Logging

- The Pi can log payload data locally on the **microSD card** or a connected USB flash drive.
- Data is timestamped, categorized, and queued for transmission via the communication subsystem.
- It can implement redundancy and backups in storage for critical experiments.

iv. Interface with Other Subsystems

- Uses GPIO, I^2C, SPI, and UART to communicate with sensors, actuators, or the On-Board Computer (OBC).

21. RASPBERRY PI 4

The Raspberry Pi 4 Model B, launched in June 2019 by the Raspberry Pi Foundation, marked a significant advancement in the single-board computer (SBC) lineup. It introduced substantial improvements over its predecessors, including a more powerful processor, increased memory options, enhanced connectivity, and support for dual 4K displays. These enhancements expanded its applicability beyond educational purposes to encompass industrial, commercial, and research domains, including space applications.

This document provides an in-depth analysis of the Raspberry Pi 4's specifications, capabilities, advantages, and limitations, particularly concerning its suitability for various satellite subsystems.

21.1 DIAGRAMS AND SCHEMATICS

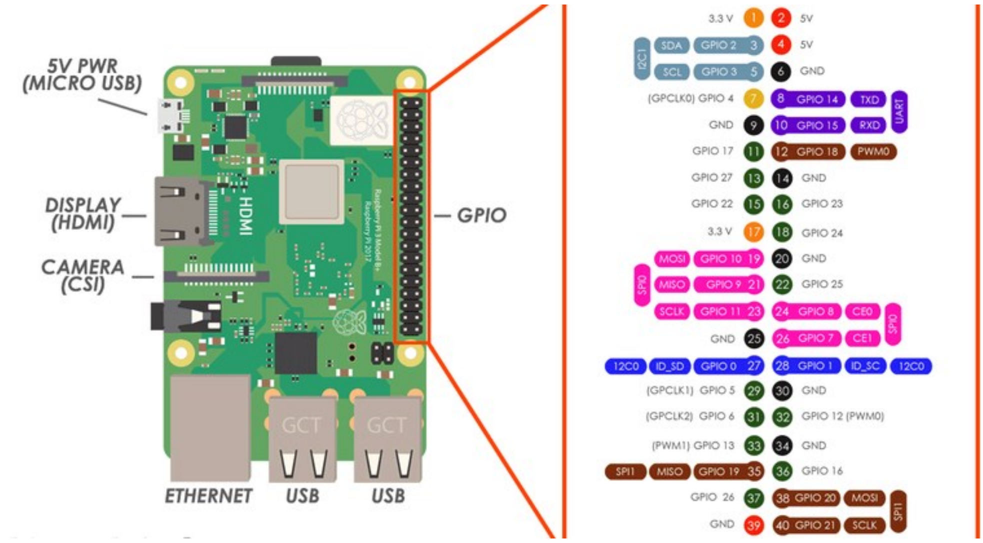


Figure 41 Raspberry Pi 4 GPIO Pinout (40-pin header)

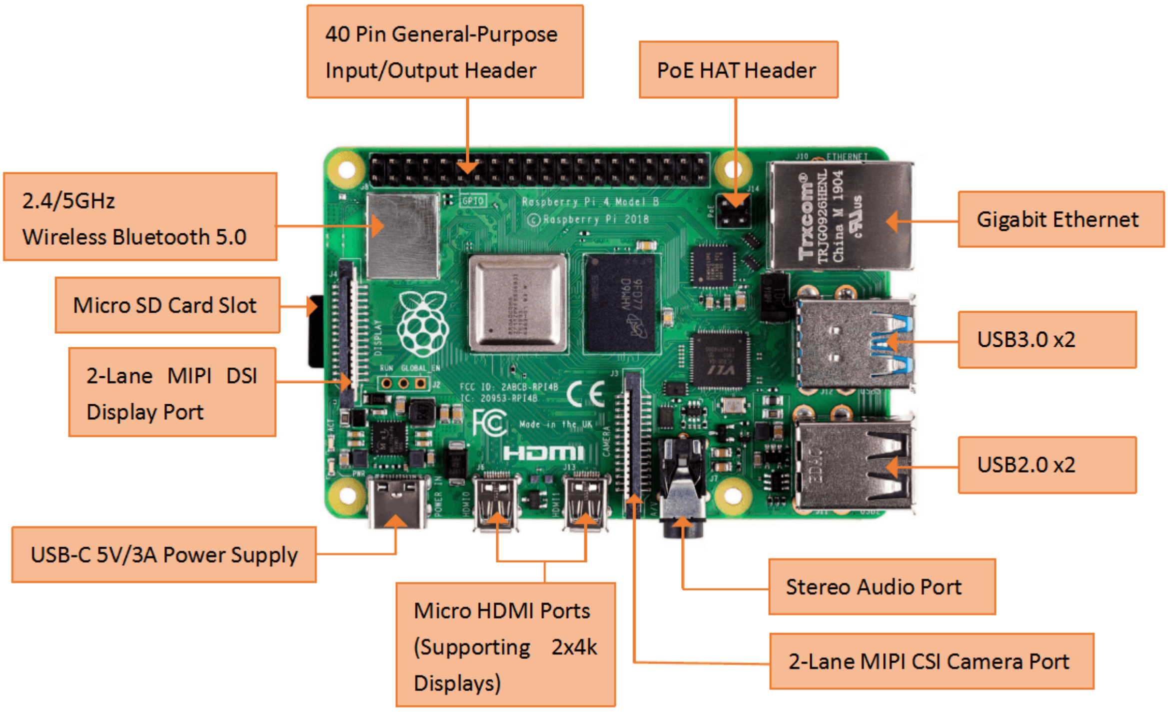


Figure 42 Raspberry Pi 4 Board Layout

21.2 SPECIFICATIONS, CAPABILITIES, AND CHARACTERISTICS

Core Architecture

- System on Chip (SoC): Broadcom BCM2711
  - CPU: Quad-core ARM Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
  - GPU: Broadcom VideoCore VI
- Memory: Available in 1GB, 2GB, 4GB, or 8GB LPDDR4-3200 SDRAM
- Storage: MicroSD card slot for OS and data storage
- Networking:
  - Ethernet: Gigabit Ethernet
  - Wireless: 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN
  - Bluetooth: Bluetooth 5.0, BLE
- USB Ports:
  - 2 × USB 3.0 ports
  - 2 × USB 2.0 ports
- Video & Sound:
  - 2 × micro-HDMI ports (up to 4Kp60 supported)
  - MIPI DSI display port

- MIPI CSI camera port
  - 4-pole stereo audio and composite video port
- GPIO: 40-pin GPIO header, fully backward-compatible with previous boards
- Power: 5V DC via USB-C connector (minimum 3A)
- Operating System Support: Raspberry Pi OS (32-bit and 64-bit), Ubuntu, and other Linux distributions

Multimedia Capabilities

- Video Decoding:
  - H.265 (4Kp60 decode)
  - H.264 (1080p60 decode, 1080p30 encode)
- Dual Display Support: Supports dual monitors via two micro-HDMI ports

Power Consideration

Power consumption is a critical consideration in a power-constrained satellite. Empirical measurements show the Pi 4 is far more power-consuming than earlier Pi models. With no peripherals attached, the Pi 4 draws about **2.5 W at idle** (≈0.5 A at 5 V) and can peak around **6.8–7 W** under a worst-case synthetic load. (By comparison, a Raspberry Pi 3B+ draws on the order of 1.5–2 W idle, and the tiny Pi Zero 2 W can idle around 0.6 W.

The Pi 4 does not have a built-in sleep or deep-standby mode like a microcontroller; it remains fully powered unless externally switched off. In practice, the board is either running or completely powered down. Satellites usually must provide a regulated 5 V supply and fuse the input for safety. The official spec is 5 V/3 A, but under typical loads the Pi 4 will draw less (e.g. ~0.6 A idle, up to ~1.4 A heavy CPU). Care must be taken to handle surge currents during boot and any faults.

- **Idle:** ≈2.5 W (0.5 A at 5 V).
- **Under load:** up to ≈6.8 W (1.3–1.4 A at 5 V).
- **Standby:** effectively none (no low-power mode without custom hardware).

21.3 ADVANTAGES AND DISADVANTAGES

Advantages

- Enhanced Performance: The quad-core Cortex-A72 processor and increased RAM options provide significant performance improvements over previous models.
- Improved Connectivity: Inclusion of USB 3.0 ports, Gigabit Ethernet, and dual-band Wi-Fi enhances data transfer and networking capabilities.
- Dual 4K Display Support: Ability to drive two 4K displays simultaneously expands its use in multimedia applications.
- Backward Compatibility: Maintains compatibility with accessories and GPIO pins from earlier models.

Disadvantages

- Lack of Radiation Hardening: Components are not designed to withstand space radiation, posing risks of malfunction in orbit.
- Thermal Management Challenges: Higher performance leads to increased heat generation, requiring effective thermal management solutions, especially in vacuum conditions.
- Power Consumption: Higher power requirements may be challenging for power-constrained satellite systems.
- Non-Real-Time Operating System: Standard Linux distributions are not real-time, which may not be suitable for time-critical applications without additional configurations.

21.4 SUITABILITY IN SATELLITE SUBSYSTEMS

Telemetry, Tracking, and Command (TT&C)

While the Raspberry Pi 4's enhanced networking capabilities could be advantageous, the lack of radiation hardening and real-time processing may compromise reliability in TT&C applications.

On-Board Computer (OBC) / Command and Data Handling (C&DH)

The increased processing power and memory make it suitable for handling complex tasks; however, concerns about reliability in space environments persist.

Electrical Power System (EPS) Control

Higher power consumption and non-real-time OS may not be ideal for managing power systems that require precise control and low power usage.

Communication (Payload Data Downlink / Inter-Satellite Links)

USB 3.0 and Gigabit Ethernet support high data rates, beneficial for payload data handling. Nonetheless, environmental vulnerabilities must be addressed.

Ground Stations

In ground-based applications, the Raspberry Pi 4's features are highly beneficial, offering a cost-effective solution for data processing and communication tasks.

Attitude Determination and Control System (ADCS)

The lack of real-time processing and environmental robustness makes it less suitable for critical ADCS functions.

21.5 CONCLUSION

The Raspberry Pi 4 Model B represents a leap in performance and features within the Raspberry Pi lineup. Its enhanced processing capabilities, memory options, and connectivity make it a versatile tool for various applications. However, for satellite subsystems, especially those requiring high reliability and resilience to harsh environmental conditions, the Raspberry Pi 4's limitations, such as lack of radiation hardening and real-time processing, must be considered. It may be best suited for non-critical functions or ground-based support systems where its advantages can be fully leveraged without compromising mission integrity.

22. RASPBERRY PI 5

The Raspberry Pi 5 is a small, affordable, and powerful single-board computer developed by the Raspberry Pi Foundation. It is the latest and most advanced version of the Raspberry Pi family.

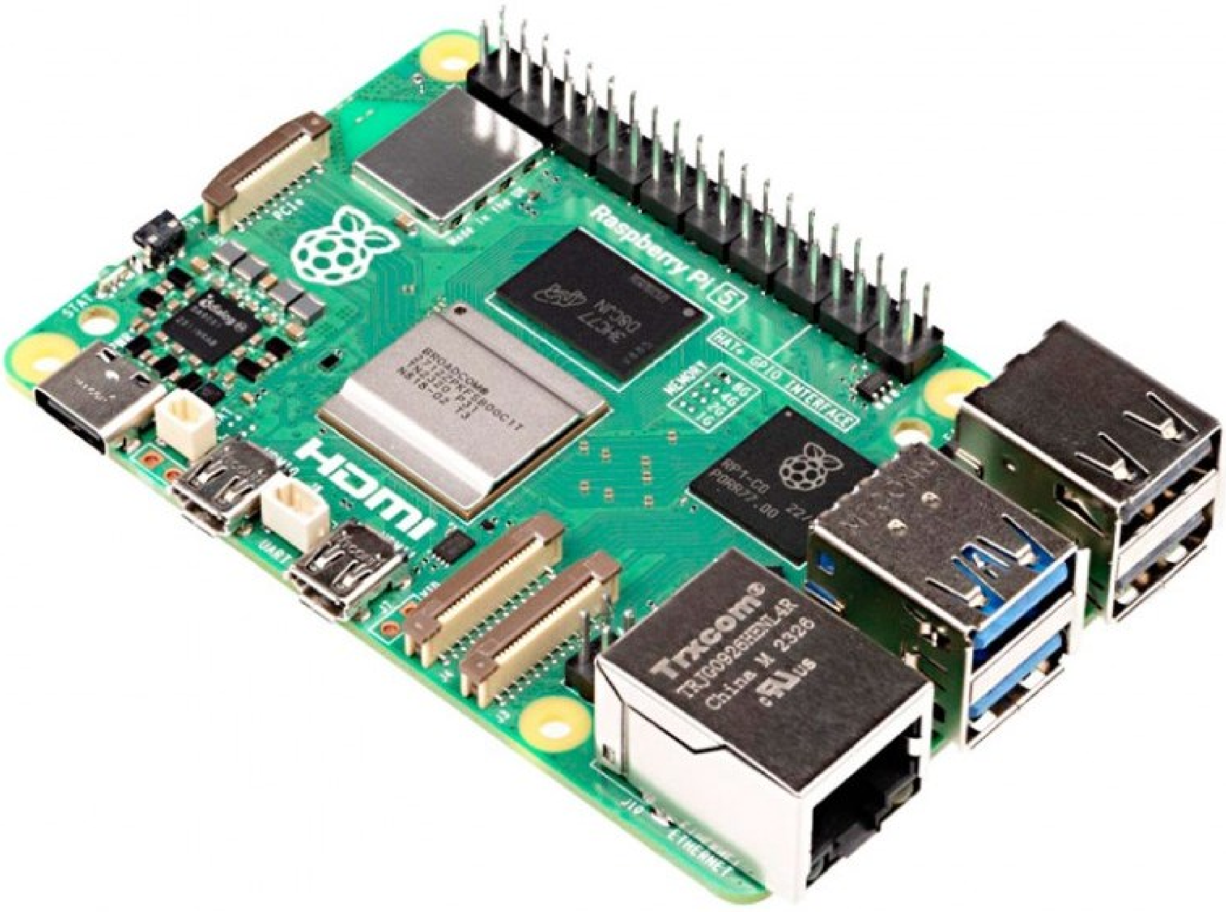
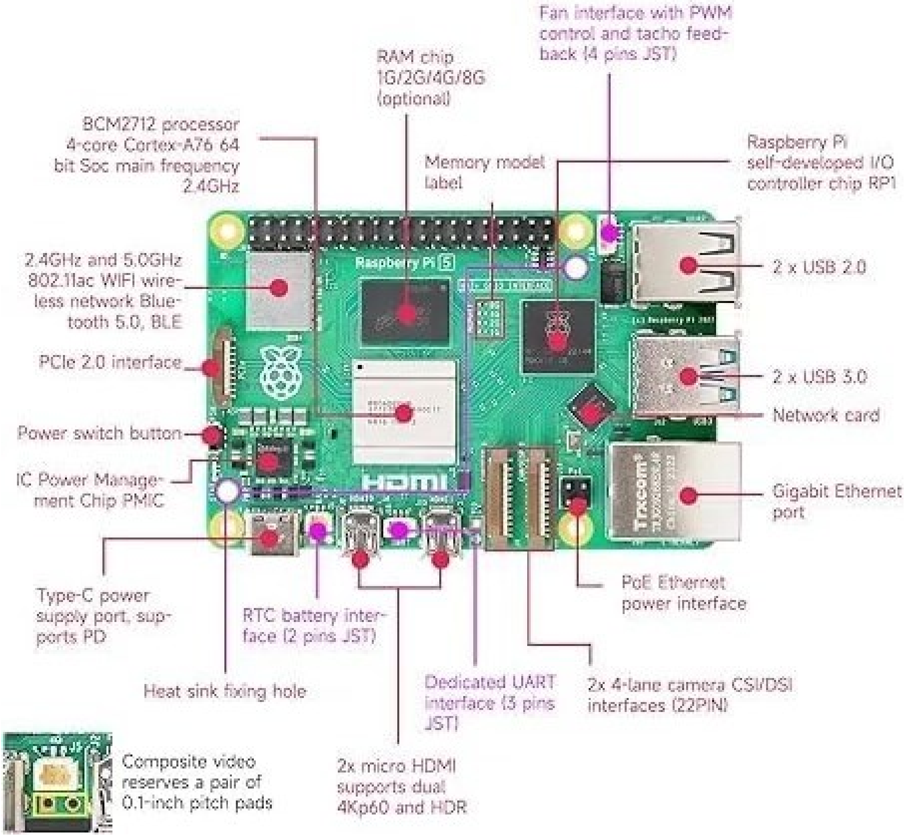


Figure 43 Raspberry pi 5

22.1 FEATURES



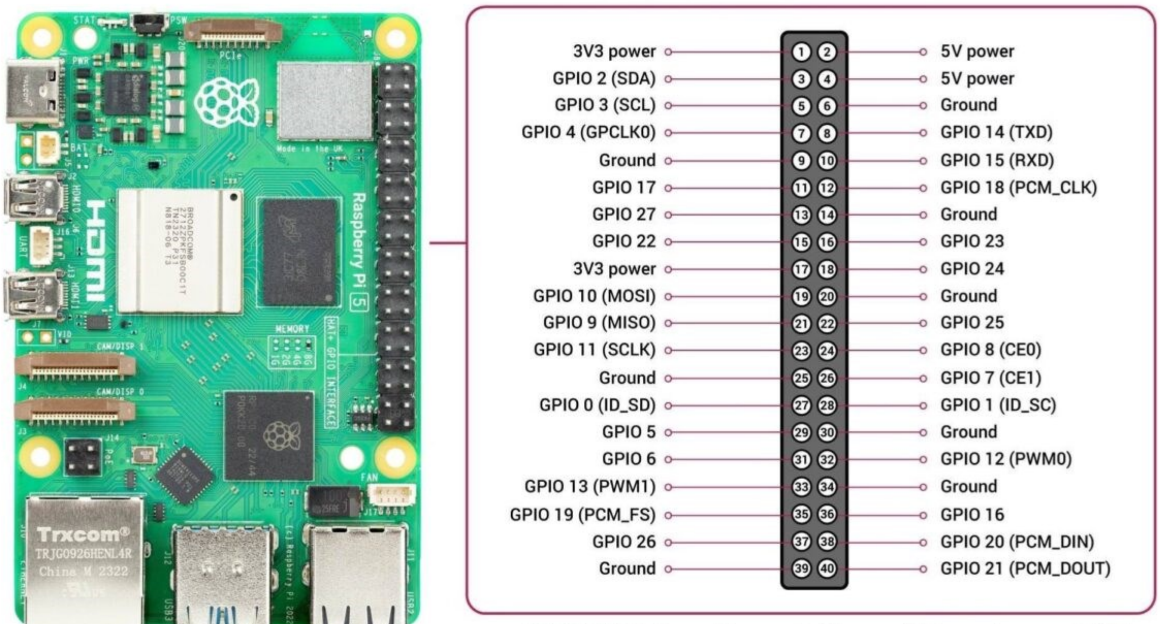
1. Processor - Quad-core ARM Cortex-A76 @ 2.4GHz which is up to 3x faster than Raspberry Pi 4
2. RAM Options – 2GB, 4GB, 8GB or 16GB LPDDR4X RAM
3. Storage - MicroSD card slot and aPCIe 2.0 x1 (via FPC connector) which supports NVMe SSDs
4. Graphics & Display - VideoCore VII GPU and two micro-HDMI ports (up to 4K@60Hz each)
5. USB Ports - 2 USB 3.0 and 2 USB 2.0
6. Networking - Gigabit (1Gb/s) Ethernet RJ45 with PoE+ support
   2.4/5GHz dual-band 802.11ac Wi-Fi 5 (300Mb/s)
   Bluetooth 5, Bluetooth Low Energy (BLE)
7. Camera & Display Interfaces- 2 CSI (camera) connectors and 2 DSI (display) connectors
8. Power Supply - USB-C, 5V/5A (25W recommended). It has a power button included
9. GPIO & Expansion - Standard 40-pin GPIO header
   PCIe expansion support
   Real-Time Clock (RTC) support (external battery needed)
10. Cooling - Improved thermal management. Supports active cooling (official fan & case available) and a heat sink slot

22.2 COMMUNICATION

Raspberry Pi 5 interacts with external systems using various interfaces and protocols:

1. Wired Communication:
    - GPIO Pins (40-pin header): For custom digital/analog I/O with sensors, switches, LEDs, and more.
    - I2C, SPI, UART: Serial communication protocols via GPIO for peripherals like IMUs, EEPROMs, GPS, etc.
    - USB 3.0 / 2.0 Ports: Connect peripherals like storage devices, keyboards, cameras, or microcontrollers.
    - HDMI Ports: Send video/audio to external displays.
    - Gigabit Ethernet: For high-speed, stable network communication.
    - PCIe Interface: Attach high-speed peripherals like SSDs, radios, or custom boards via an adapter.
2. Wireless Communication:
    - Wi-Fi 5 (802.11ac): Dual-band for internet or LAN-based data transfer.
    - Bluetooth 5.0 / BLE: Connect to low-power devices like sensors, controllers, or smartphones.
3. Other Interfaces:
    - microSD Slot: Load OS and read/write data.
    - Camera & Display Ports (MIPI): Use CSI/DSI interfaces for image capture or GUI display.

22.3 PIN OUT



40 GPIO Pins Description of Raspberry Pi 5

The Raspberry Pi 5 has a 40-pin GPIO header, which is the same GPIO header from Raspberry Pi 4. You can use the GPIO header to connect a variety of sensors, actuators, and other electronic devices to the Raspberry Pi. The table below shows the GPIO pinout for the Raspberry Pi 5.

Pin	Name	Function
1	3.3V	3.3V power supply
2	5V	5V power supply
3	GPIO 2	I2C1 SDA (Data Line)
4	5V	5V power supply
5	GPIO 3	I2C1 SCL (Clock Line)
6	Ground	Ground
7	GPIO 4	General-purpose input/output
8	GPIO 14	UART TXD (Transmit Data)
9	Ground	Ground
10	GPIO 15	UART RXD (Receive Data)
11	GPIO 17	General-purpose input/output
12	GPIO 18	PCM_CLK (Pulse Code Modulation Clock)
13	GPIO 27	General-purpose input/output
14	Ground	Ground
15	GPIO 22	General-purpose input/output
16	GPIO 23	General-purpose input/output
17	3.3V	3.3V power supply
18	GPIO 24	General-purpose input/output
19	GPIO 10	SPI0 MOSI (Master Out, Slave In)
20	Ground	Ground
21	GPIO 9	SPI0 MISO (Master In, Slave Out)
22	GPIO 25	General-purpose input/output
23	GPIO 11	SPI0 SCLK (Clock Line)

24	GPIO 8	SPI0 CE0 (Chip Enable 0)
25	Ground	Ground
26	GPIO 7	SPI0 CE1 (Chip Enable 1)
27	GPIO 0	I2C EEPROM Data Line / ID_SD
28	GPIO 1	I2C EEPROM Clock Line / ID_SC
29	GPIO 5	General-purpose input/output
30	Ground	Ground
31	GPIO 6	General-purpose input/output
32	GPIO 12	PWM (Pulse Width Modulation)
33	GPIO 13	PWM (Pulse Width Modulation)
34	Ground	Ground
35	GPIO 19	PCM_FS (Frame Sync for Pulse Code Modulation)
36	GPIO 16	General-purpose input/output
37	GPIO 26	General-purpose input/output
38	GPIO 20	PCM_DIN (Data In for Pulse Code Modulation)
39	Ground	Ground
40	GPIO 21	PCM_DOUT (Data Out for Pulse Code Modulation)

22.4 POWER

1. Powering the Raspberry Pi 5 with USB-C power supply. Raspberry Pi sells its official 5.1V 5A 27W USB-C PSU which is made specifically for the Raspberry Pi 5 (well, and the Pi 500, CM5, etc) which has a “requirement” of 5A so as to not limit the USB ports current to 600mA by default.
2. Using GPIO Header. This is an advanced method and is not Recommended for Beginners. It involves supplying 5V and GND directly to the GPIO pins as Pin 2 or 4: +5V and Pin 6 or 9:

GND. This method has no protection circuitry hence risky if the voltage isn't regulated precisely to 5V.

3. Power over Ethernet (PoE+) - Requires the Raspberry Pi PoE+ HAT (sold separately). PoE+ HAT is a Power over Ethernet add-on board for Raspberry Pi that utilizes the PoE+ standard to deliver both power and data over a single Ethernet cable. It allows for power delivery from a PoE+ switch or injector to the Raspberry Pi, instead of relying on a separate power supply.

22.5 PROGRAMMING

Getting started with Raspberry Pi 5 you'll require:

1. Raspberry Pi Board (e.g., Pi 5)
2. microSD Card (minimum 8GB, Class 10 recommended)
3. Power Supply (USB-C, 5V/3A or higher)
4. Display (HDMI-compatible monitor or TV)
5. HDMI Cable (micro-HDMI to standard HDMI)
6. USB Keyboard and Mouse
7. Internet Connection (via Ethernet or Wi-Fi)

Installing the Operating System

1. Download Raspberry Pi Imager: Available for Windows, macOS, and Linux from the official website.
2. Insert microSD Card: Connect it to your computer using a card reader.
3. Launch Raspberry Pi Imager:
    - Choose OS: Select "Raspberry Pi OS (32-bit)" or another preferred OS.
    - Choose Storage: Select your connected microSD card.
    - Write: Click "Write" to install the OS onto the card.
4. Insert microSD Card into Pi: Once imaging is complete, safely eject the card and insert it into your Raspberry Pi.

First Boot and Configuration

1. Connect Peripherals: Attach your keyboard, mouse, and monitor to the Raspberry Pi.
2. Power Up: Connect the power supply to boot the Pi.
3. Initial Setup:
    - Set Locale: Choose your country, language, and time zone.
    - Change Password: Replace the default password for security.
    - Connect to Wi-Fi: If not using Ethernet, select your network and enter the password.
    - Update Software: Allow the system to check for and install updates.

You can use Raspberry Pi 5 in two modes ie Desktop and Headless Setup

- Desktop Mode: Utilize the Raspberry Pi with a connected monitor, keyboard, and mouse for a traditional desktop experience.
- Headless Mode: Operate the Pi without a monitor or keyboard by enabling SSH or VNC, allowing remote access over the network.

Enabling Interfaces

To expand the Pi's capabilities:

1. Open Raspberry Pi Configuration: Found under Preferences in the main menu.
2. Navigate to Interfaces Tab:
   - Enable SSH: Allows remote terminal access.
   - Enable VNC: Provides remote desktop access.
   - Enable SPI, I2C, Serial: For hardware interfacing with various sensors and modules.

With your Raspberry Pi set up, you can now explore programming
The Pi 5 supports almost any language. The most popular options are;

1. Python (Best for Beginners) - Pre-installed on Raspberry Pi OS. Great for GPIO, robotics, sensors, and AI.
2. C/C++ (High Performance). Ideal for hardware control (e.g., custom drivers). Use WiringPi (deprecated but alternatives like libgpiod exist).
3. JavaScript (Node.js) - For web apps/IoT (using Node.js + onoff for GPIO).
4. Bash scripting (for automation).
5. Java/Kotlin (Android Things compatibility).
6. Rust (for embedded safety).

22.6 OTHER FEATURES

Raspberry Pi 5 does not have a built-in 3.5mm audio jack like older models.

To get Audio output use;

1. HDMI Audio (Default) - If your monitor or TV has speakers, just connect via HDMI. Audio will come out through the screen.
2. Use a USB sound card or plug in USB speakers. The Pi will detect it automatically.
3. Bluetooth headphones or speakers - Pair the device in Bluetooth settings. Set it as the default audio output.

22.7 PRICE

Vendor	Price (Ksh.)
Nerokas	19500
Jumia	20300
Pixel Electric	19500

22.8 ADVANTAGES

1. Faster Performance
   - Powered by a 2.4GHz quad-core Cortex-A76 CPU, which is 2–3× faster than Pi 4.
   - Great for multitasking, running software smoothly, and handling heavier workloads.

2. Better Graphics
   - VideoCore VII GPU supports dual 4K displays at 60Hz, Vulkan 1.2, and OpenGL ES 3.1.
   - Ideal for video streaming, digital signage, and light gaming.

3. Improved Connectivity
    - 2× USB 3.0 ports for faster data transfer.
    - Gigabit Ethernet and Bluetooth 5.0 for better network and device connections.
    - Dual-band Wi-Fi for strong internet access.

4. Expandability
    - PCIe 2.0 interface lets you add fast SSDs, AI accelerators, and more (via adapter).
    - 2× MIPI camera/display connectors for advanced camera setups or extra displays.

5. Enhanced Power Management
    - USB-C power input (5V/5A) supports Power Delivery for stable performance.
    - Has a real-time clock (RTC) with external battery support and a power button—both new features.

6. Better Memory
    - Uses LPDDR4X-4267 RAM (faster and more efficient).
    - Multiple RAM options (4GB or 8GB).

7. Software Compatibility
    - Works with Raspberry Pi OS and other Linux distros.
    - Supports programming in Python, C, C++, Java, etc.

22.9 DISADVANTAGES

1. No Audio Jack
    - Unlike older models (like Pi 4), Pi 5 doesn't have a 3.5mm audio jack.
    - You need to use HDMI, USB sound cards, or Bluetooth for audio.

2. Higher Power Requirements
    - Needs a 5V/5A USB-C power supply.
    - Many older or basic power adapters/power banks may not work well.

3. No Built-in eMMC Storage
    - Still relies on a microSD card for storage (slower than internal storage).
    - You need to buy and manage external storage for better speed

4. More Heat
    - Faster processor generates more heat.
    - A cooling fan or heatsink is recommended, especially during heavy tasks.

5. Not Beginner-Friendly for Some Tasks
    - Features like PCIe, RTC, and MIPI connectors are powerful but require technical skills to use.

22.10 COMPARISON WITH OTHER RASPBERRY PI MODELS

Feature	RPi 1 B+	RPi 2 B	RPi 3 B+	RPi 4 B	RPi 5
Release Year	2014	2015	2018	2019	2023
CPU	700MHz Cortex-A7 (Single)	900MHz Quad-core Cortex-A7	1.4GHz Quad-core Cortex-A53	1.5GHz Quad-core Cortex-A72	2.4GHz Quad-core Cortex-A76
GPU	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore VI	VideoCore VII (Vulkan 1.2)
RAM	512MB LPDDR2	1GB LPDDR2	1GB LPDDR2	2GB / 4GB / 8GB LPDDR4	4GB or 8GB LPDDR4X-4267
Storage	microSD	microSD	microSD	microSD, USB boot	microSD, PCIe SSD (via adapter)
USB Ports	4 × USB 2.0	4 × USB 2.0	4 × USB 2.0	2 × USB 3.0, 2 × USB 2.0	2 × USB 3.0, 2 × USB 2.0
Ethernet	10/100 Mbps	10/100 Mbps	10/100 Mbps	Gigabit (shared with USB)	True Gigabit + PoE+ support
Wi-Fi	✗	✗	802.11n (Wi-Fi 4)	802.11ac (Wi-Fi 5)	802.11ac (Wi-Fi 5, dual-band)
Bluetooth	✗	✗	Bluetooth 4.2	Bluetooth 5.0	Bluetooth 5.0 / BLE
HDMI Output	1 × HDMI	1 × HDMI	1 × HDMI	2 × micro-HDMI (4Kp30)	2 × micro-HDMI (4Kp60 with HDR)
Audio Output	3.5mm + HDMI	3.5mm + HDMI	3.5mm + HDMI	3.5mm + HDMI	HDMI / USB / Bluetooth only
Camera Interface	1 × MIPI CSI	1 × MIPI CSI	1 × MIPI CSI	1 × MIPI CSI	2 × 4-lane MIPI transceivers
Display Interface	1 × DSI	1 × DSI	1 × DSI	1 × DSI	2 × 4-lane MIPI transceivers
PCIe Support	✗	✗	✗	✗	✓ PCIe 2.0 x1 (via FFC adapter)
RTC (Real-Time Clock)	✗	✗	✗	✗	✓ (with external battery)
Power Supply Type	5V/2A micro-USB	5V/2A micro-USB	5V/2.5A micro-USB	5V/3A USB-C	5V/5A USB-C with Power Delivery
Cooling Required	Low	Low	Low	Medium	High – active cooling needed
GPIO Header	26 pins	40 pins	40 pins	40 pins	40 ins

22.11 RELEVANCE TO SATELLITE PROJECT

Raspberry Pi 5 is much relevant to the satellite project because of;

1. High Processing Power	Cortex-A76 CPU enables advanced data processing, image handling, or AI models on-board.
2. GPIO + Serial Protocols (I2C/SPI/UART)	Easily connect satellite subsystems like attitude sensors, GPS modules, or power systems.
3. Low Size, Weight, and Power (SWaP)	Compact design and efficient power use make it suitable for CubeSat/nanosat structures.
4. Camera Interface (MIPI CSI)	Capture Earth images or payload visuals directly.
5. Linux OS Support	Full OS allows multitasking, logging, real-time health monitoring, and remote updates.
6. Wireless (for Ground Testing)	Use Wi-Fi/Bluetooth for simulation, telemetry, or debugging during ground-based prototyping.
7. Storage + PCIe Option	High-speed SSD or flash memory for storing payload data or logs.
8. Community and Tools	Rich documentation and global community simplify prototyping and testing.

Raspberry pi 5 best fits in On Board computer subsystem because of powerful data processing, data storage, task scheduling, subsystems coordination, health monitoring and software flexibility.

Raspberry Pi 5 can also best fit in payload subsystem for satellites whose primary mission involves imaging, sensors and/or computation.

23 RECOMMENDATIONS: MICROCONTROLLER SUITABILITY SUMMARY

Table 20 Suitability of the different microcontrollers

Microcontroller	Power Efficiency	Compute Performance	Radiation Resilience	Ecosystem/Support	Best Fit Role
MSP430	☆☆☆☆☆	☆☆	☆☆☆☆ (FRx variants)	☆☆☆	Sensor polling, PMU
Tiva C Series	☆☆☆	☆☆☆☆	☆☆	☆☆☆	ADCS, real-time control
RPi Zero W	☆	☆☆☆☆	☆	☆☆☆☆	Payload processing, imaging
STM32F4/F7/H7	☆☆☆☆	☆☆☆☆	☆☆	☆☆☆☆☆	General control, ADCS, telemetry
Teensy 4.1	☆☆	☆☆☆☆☆	☆☆	☆☆☆☆	High-speed data/control
Arduino Due	☆☆	☆☆☆	☆☆	☆☆☆	Mid-level control systems
Raspberry Pi Pico	☆☆☆☆	☆☆	☆☆	☆☆☆☆	Peripheral/sensor control
ESP32/ESP8266	☆☆	☆☆☆	☆	☆☆☆☆	Ground link, testing/debug only
BeagleBone Black	☆☆	☆☆☆☆	☆☆	☆☆☆☆	Payload/processing
Adafruit Feather M4	☆☆☆☆	☆☆☆	☆☆	☆☆☆☆	Wireless sensing node
ATtiny Series	☆☆☆☆☆	☆	☆☆	☆☆	Redundant backup, switches
ATmega328/2560	☆☆☆	☆☆	☆☆	☆☆☆☆	Simple control/logic

PIC16F877A	☆☆☆	☆☆	☆☆	☆☆☆	Legacy tasks, I/O logic

Table 21 Different microcontrollers with their suitable Sections

MCU	Key Specs	Price Range	Subsystem Use	Specific Payload Examples or Use Cases
STM32F4 Series	ARM Cortex-M4, 168 MHz, FPU, 1MB Flash	\$5–\$10	OBC, Payload, ADCS	Image processing payloads, spectrometer control, material science sensor interfacing, real-time telemetry collection
ATmega328P	8-bit AVR, 20 MHz, 32KB Flash	\$2–\$5	EPS, Simple Payloads	Temp & voltage monitoring, radiation sensor interface, low-data biological payload logging
MSP430 Series	16-bit, ultra-low-power	\$2–\$6	EPS, Housekeeping	Environmental monitoring, energy harvesting experiments
ATmega2560	8-bit AVR, 256KB Flash, 54 I/O pins	\$8–\$12	OBC, Payload	Multiple sensor interfacing, beaconing, housekeeping arrays
Raspberry Pi Pico	Dual-core Cortex-M0+, 133 MHz, 264KB RAM	\$4	ADCS, Payload, OBC	Magnetometer fusion, basic imaging payloads, solar test logging
ESP32	Dual-core Xtensa, Wi-Fi, BT, up to 240 MHz	\$4–\$8	Telemetry, OBC, Payload, EPS	Comms experiments, IoST relays, environmental telemetry
SAMD21	ARM Cortex-M0+, 48 MHz	\$3–\$6	Housekeeping, EPS	Power telemetry, thermal degradation tests
Teensy 4.0 / 4.1	Cortex-M7, 600 MHz, up to 1MB RAM	\$20–\$30	Payload, ADCS, OBC	SDR control, onboard AI, real-time radiation analysis
nRF52840	ARM Cortex-M4, BLE, low-power	\$5–\$10	COMM (short-range), EPS	Wireless mesh for sensor data, remote panel telemetry

MCU	Key Specs	Price Range	Subsystem Use	Specific Payload Examples or Use Cases
TI Tiva C Series	ARM Cortex-M4, 80 MHz	$5–$10	OBC, Payload Control	Spectrometer control, payload scheduling
Raspberry Pi 4	Quad-core Cortex-A72, 1.5 GHz, up to 8GB RAM	$35–$75	Payload, OBC, Ground Testing	High-resolution image compression, AI inference, Linux-based scripting, SDR/FFT experiments
Raspberry Pi 5	Quad-core Cortex-A76, 2.4 GHz, up to 8GB RAM, PCIe, USB 3.0, Gigabit Ethernet	$60–$90	Payload, Imaging, OBC	Advanced AI payloads, deep learning models, onboard 4K image processing, real-time camera handling, neural inference logging
ATtiny Series	8-bit AVR, low-power	<$3	Redundant backup, switches	Simple logic tasks, sleep-mode sensors
PIC16F877A	8-bit PIC, 20 MHz, 368B RAM	$3–$6	Legacy logic, EPS	GPS interfacing, telemetry tasks, analog signal handling

23. EXISTING LEO SATELLITE PROJECTS AND THE MICROCONTROLLERS USED

To obtain the best functionality for our Leo satellite, we had to gather as much information as we could obtain from the documentation of different projects done by students across the world. This will help us make informed decisions.

The following are some of the documented Leo satellite projects we were able to gather some useful information on the different microcontrollers used.

PROJECTS AND THEIR MICROCONTROLLER SELECTIONS

NaSPUoN project

A project by NanoSatellite Platform for the University of Nairobi (NaSPUoN) Team.

The cameras investigated for this project are the Arducam 5MP Mini Camera, USB 3MP Camera and Raspberry Pi High Quality Camera. The camera chosen is the Raspberry Pi High Quality Camera.

The initial design choice was to use a **Raspberry Pi 4** for the **on-board computer**. However, mainly due to **size restrictions** in the 1U CubeSat specification, the **Raspberry Pi Zero** was the component that was settled upon.

Even though the Raspberry Pi Zero is less computationally powerful and has less interfaces compared to the Raspberry Pi 4.

The specifications of the Raspberry Pi Zero are as follows:

- Single-core BCM2835 at 1GHz
- 512MB RAM
- Small Size
- Typical Power draw of 0.75W – this is lower than that of the Raspberry Pi 4
- Can be powered using the GPIO header – the Raspberry Pi 4 cannot

Interfaces available:

- 1 UART *{there are 2 UART’s but only one can be used at a time since they are on the same pins}
- 2 SPI buses (total of 5 CS pins)
- 1 I2C bus *{2 I2C’s but only one is usable the other being a special-purpose I2C}
- 1 CSI header (only on Raspberry Pi v1.3)

For the **EPS**, an **STM 32** was used but the exact microcontroller is not specified in the documentation

The interfacing block diagram is shown below;

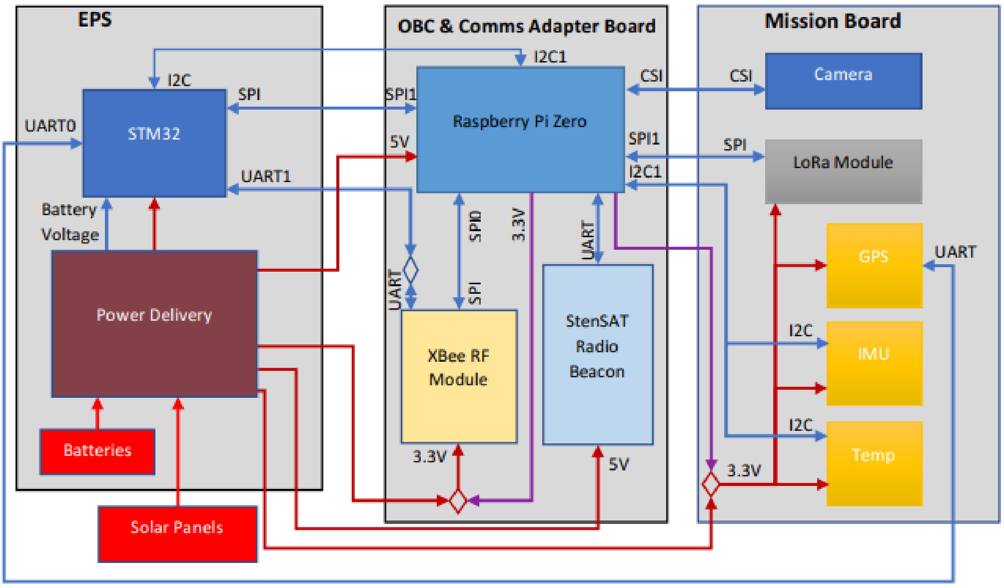


Fig 1. Subsystem interfacing block diagram

Ardusat

ArduSat is an Arduino based nanosatellite, based on the CubeSat standard. It contains a set of Arduino boards and sensors. The general public will be allowed to use these Arduinos and sensors for their own creative purposes while they are in space.

The ArduSat project currently consists in two identical satellites: ArduSat-1 and ArduSat-X.

Category	Specifications
General Architecture	**1U CubeSat** : the satellites implements the standard 10×10×10 cm basic CubeSat architecture.
Computing features	**Arduino-based** : The ArduSat is equipped with 16 processor nodes (ATmega328P) and 1 supervisor node (ATmega2561). The processor nodes are dedicated to the computing of the experiments (each on one node), the supervisor uploads the code to the processor nodes.
Sensors	The Arduino processors may sample data from the following sensors : • one digital 3-axis **magnetometer** (MAG3110) • one digital 3-axis **gyroscope** (ITG-3200) • one 3-axis **accelerometer** (ADXL345)

	• one **infrared temperature sensor** with a wide sensing range (MLX90614) • four digital **temperature sensors** (TMP102) : 2 in the payload, 2 on the bottomplate • two **luminosity sensor** (TSL2561) covering both infrared and visible light : 1 on the bottomplate camera, 1 on the bottomplate slit • two **geiger counter** tubes (LND 716) • one optical **spectrometer** (Spectruino) • one 1.3MP **camera** (C439)
Coding	The experiments for ArduSat are developed in **C/C++** for AVR/Arduino, using the ArduSatSDK.
Communication	ArduSat is equipped with a **half-duplex UHF transceiver**, operating in the 435–438 MHz amateur radio satellite band. ArduSat-1 : 437.325 MHz 9k6 MSK CCSDS downlink • ArduSat-X : 437.345 MHz 9k6 MSK CCSDS downlink Both satellites have a Morse beacon (FM-modulated 800 Hz tones) that is transmitted at 20 WPM every two or three minutes on 437.000 MHz. The beacon will be structured in the following format. • ArduSat-1 beacon: Battery voltage (uint16_t), RX_counter (number of received valid data packets, uint32_t), TX_counter (number of sent valid data packets, uint32_t), "WG9XFC-1” • ArduSat-X beacon: Battery voltage (uint16_t), RX_counter (number of received valid data packets, uint32_t), TX_counter (number of sent valid data packets, uint32_t), "WG9XFC-X"

Table 1. Ardusat specifications

ESTCube-1 & -2 (Estonian Student Satellite-1 & -2)

ESTCube-1 & -2 were two Estonian student CubeSat projects of the University of Tartu, which started in the summer of 2008. The objective was to get students involved in space projects. Another goal was to foster the development of Estonian space and high-tech industry by training experts and disseminating knowledge about space technologies.

EPS

The EPS was controlled through an **ATMega1280** 8 bit AVR microcontroller from Atmel (can be found in Arduino Mega 1280). The processor had been tested before and deemed suitable for conditions similar to the current mission.

COMMAND AND DATA HANDLING SUBSYSTEM

The CDHS had to

1. store data from the CAM (Camera subsystem) in the form of multiple binary images of up to 600 KB each.

2. store housekeeping data of all subsystems.

3. compile the beacon and data packets for downlink.

The **CDHS** contained two **STM32F103** ARM processors, one of which was turned on. The two processors gave the possibility to activate the second one if the first was defective, to make sure that the satellite remained operational. The CDHS had three data interface types with other ESTCube-1 & -2 modules: SPI (Serial Peripheral Interface), I^2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver/Transmitter) which were organized into several logical buses. The FreeRTOS operating system was used to fulfil requirements of real-time operations, processing capacity and memory footprint.

RF COMMUNICATIONS

Use of a VHF/UHF system for uplink (1.2 kbit/s) and downlink (9.6 kbit/s) data transmissions. Both uplink and downlink used AX.25 unnumbered information frames as a transport protocol. In addition, a CW beacon was used.

Quarter wave monopoles were used as antennas for both uplink and downlink. A power amplifier on the satellite provided up to 500 mW for data downlink, and a preamplifier provided 18 dB amplification for uplink. The transmit power of the CW beacon was 100 mW. Due to the shared RF chain the beacon and the primary downlink couldn't be transmitted simultaneously. The Texas Instruments **MSP430F2418** MCU was used on the COM.

CAMERA PAYLOAD

A robust independent camera module with on-board image processing, based on the **ARM Cortex-M3** microcontroller and fast static random access memory, had been developed and characterized for the requirements of the ESTCube-1 mission.

GASPACS (Get Away Special Passive Attitude Control Satellite)

The **Get Away Special Passive Attitude Control Satellite (GASPACS)** was a 1U CubeSat technology demonstration mission designed to test inflatable structures in space.

For the first time ever, a Raspberry Pi was used as a satellite's on-board computer. GASPACS' Raspberry Pi Zero handles all of the satellite's computing, running on software written entirely by the Software Team.

Attached to the **Raspberry Pi Zero W** is a **Raspberry Pi Camera**, which is used to capture an image of the deployed AeroBoom. The board also powers the burn wire system, which is used to deploy the AeroBoom. Also on the electrical board is a **DFRobot Beetle microcontroller** that acts as a **Watchdog,** ensuring that the Raspberry Pi is functioning nominally. The Pi sends a signal to the Watchdog at 0.25 Hz. If the Watchdog does not receive a signal for more than five seconds, the Watchdog will power cycle the Pi in hopes that the Pi will return to normality after reboot.

Challenge; Three days into the mission, GASPACS lost charging on the Y-axis solar panels. As a result of this, GASPACS became very power negative, meaning it used much more power than it could generate. This was reflected in GASPACS's ontime. GASPACS would typically be turned off and charging for around six hours, and then be on for about an hour.

Swayam – College of Engineering, Pune (India)

On Board Computer

- **ARM7TDMI** based Microcontroller
- Foreground background interrupt driven system
- Responsible for data handling and housekeeping
- Custom network layer protocol 'COEP Satellite Protocol'
- 2 GB On board SD Card Storage

ESTCube-2 – University of Tartu (Estonia)

The **EPS** main board performs numerous functions related to power management and is the core of the EPS. It is responsible for power distribution, voltage conversion, and monitoring different current, voltage, and temperature levels. The system has a dedicated STMicroelectronics **STM32L4series** microcontroller (MCU), external ferroelectric random-access memory (FRAM) for storing data, external analog-to-digital converters (ADCs) to perform analog measurements,3.3 V housekeeping which is isolated from the rest of the satellite, and a communication interface with other subsystems.

The **primary communication (PCOM)** system is tuned at the radio amateur 70 cm band with a planned frequency of 435.8 MHz and 9600 to19200 baud variable data rate. Its processing unit is an STMi-croelectronics **STM32L4 series** MCU, and RF connectivity is achieved with a Silicon Labs Si4463 transceiver.

SCOM is primarily used as a secondary receiver subsystem and uses a radio amateur 2 m band. The data processing unit is a Silicon Labs **EZR32WG330 series MCU** with built-in RF transceiver module. For transmission, the subsystem can change its carrier frequency to the 70 cm band and use PCOM's RF path to transmit the signal as a backup transmitter

The **OBC** handles the operations of all subsystems and payloads, runs the AOCS algorithms and stores housekeeping and telemetry data. The central computer of the satellite is a

STM32F7 series MCU. The MCU features 512 kB of static random-access memory (SRAM)and 2 MB flash memory.

There were issues with the STM MCU which are discussed in this document later in this document.

E-st@r – Polytechnic University of Turin (Italy)

OBC (On-Board Computer): The OBC is based on an off-the-shelf processing unit developed by Pumpkin Inc. It consists of a microprocessor of Texas Instruments (**MSP430**, 16 bit), and works on the SALVO real-time operating system.

Communication; . A commercial transceiver (Radiometrix) is employed and integrated on the onboard shelf contained electronics, equipped also with a **PIC16** that accomplishes the modem function.

AcubeSAT – Aristotle University of Thessaloniki (Greece)

The hosted subsystems are functionally isolated and feature an **ARM Cortex-M7**, radiation-tolerant microcontroller each.

TJREVERB (Thomas Jefferson High School, USA)

The **Onboard microcontroller** used was a **Raspberry Pi Zero**.

M-Cubed – University of Michigan (USA)

On-board control is provided by a Taskit Stamp9G20 microcontroller running RTLinux.

Project	Subsystem	Microcontroller Used	Reasons
NaSPUoN	On-Board Computer (OBC)	Raspberry Pi Zero (BCM2835, 1GHz, 512MB RAM)	Chosen over Pi 4 due to **size and power constraints**; suitable for camera processing
	Camera / Payload	Raspberry Pi High Quality Camera	Connected via CSI to Pi Zero; higher resolutio for imaging
	Electrical Power System (EPS)	STM32 (specific model not given)	STM32 used for EPS—likely low-power series for energy efficiency
GASPACS	OBC	Raspberry Pi Zero	Simple Linux-based controller for student mission
	Watchdog	ATmega32U4 (DFRobot Beetle)	Simple 8-bit MCU used for fault recovery and resets
TJREVERB	OBC	Raspberry Pi Zero	Used in student CubeSat for mission computi
ESTCube-1	CDHS	STM32F103 (Cortex-M3)	Dual-redundant STM32 for robust data handli with FreeRTOS
	EPS	ATmega1280	Power management using 8-bit AVR
	Camera Payload	STM32F103 (Cortex-M3)	Used for interfacing with CMOS camera and data handling
	COM	MSP430F2418	Low-power controller for communication module
ESTCube-2	OBC	STM32F7	Powerful ARM Cortex-M7 for central control
	EPS	STM32L4	Low-power microcontroller with advanced AD features
	SCOM	EZR32WG330	MCU with built-in RF capabilities for simplifie radio subsystem
ArduSat	Supervisor / Watchdog	ATmega2561	Used for monitoring and fallback operations
E-st@r	COM (Modem)	PIC16	Lightweight 8-bit MCU for simple modulation tasks

Table 2. Summary of microcontrollers choices for different subsystems in Leo projects

CHALLENGES

Some of the documented challenges with using different microcontrollers are listed below;

RASPBERRY PI 4.

The NaSPUoN project had decided to use Raspberry Pi 4 but opted for **Raspberry Pi Zero** mainly due to **size restrictions** in the 1U CubeSat specification despite Raspberry Pi Zero being less computationally powerful and having less interfaces compared to the **Raspberry Pi 4.**

STM32

The following were the limitations given from using STM32 MCU in the ESTCube-1 & -2 projects

The STM32 HAL is a high-level driver framework provided by STMicroelectronics to simplify hardware development across different STM32 microcontroller families. While useful for beginners and rapid development, it has several important limitations and potential pitfalls, especially in performance- and reliability-critical applications like satellite systems.

Below are key concerns explained clearly:

1. **No Programmatic Peripheral Access by Index**
   The STM32 HAL does not support accessing peripherals (e.g., USART1, USART2, etc.) via index in a generic way.
   Developers cannot write scalable or loop-based code to handle multiple peripherals dynamically. Each peripheral must be handled explicitly with its unique HAL handle (e.g., &huart1, &huart2).
2. **Model-Specific Handling is Manual**
   The HAL does not completely abstract hardware differences across STM32 families (like F1, F3, F4, etc.).
   Developers must account for family-specific quirks or register layouts manually, reducing code portability.
3. **Deadlocks in Interrupt Handlers**
   Certain HAL functions (e.g., HAL_UART_Transmit) use blocking or polling behavior. If called within Interrupt Service Routines (ISRs), they may cause the system to hang.
   Example: A HAL delay or wait in an ISR may never complete if interrupts are disabled or delayed, leading to deadlock.
4. **Incompatibility with Compiler Optimizations**
   HAL code may rely on specific timing or memory access behavior. When compiler optimizations like -O2 or -O3 are used, this behavior can break due to reordering or inlining.
   Developers may face unexpected bugs unless memory barriers or volatile variables are properly used.
5. **High Code and Memory Overhead**
   The HAL generates a large amount of boilerplate and overhead code.
   In resource-constrained systems, such as CubeSats or IoT devices, this can be problematic due to limited Flash/RAM.

24. CONCLUSION

In selecting an optimal microcontroller for the Tafiti Project LEO satellite, the primary considerations included processing power, energy efficiency, peripheral support, cost-effectiveness, and space suitability. After a comprehensive comparative analysis of over a dozen microcontrollers and development boards—including ESP32, ATmega328P, STM32F4, Teensy, and Raspberry Pi Pico—it became evident that no single microcontroller could serve all subsystems optimally. As such, the final selection strategy involved pairing complementary microcontrollers, each tailored to specific subsystem demands.

The ESP32 emerged as a robust candidate due to its dual-core architecture, integrated Wi-Fi/Bluetooth, and low power sleep modes. Its versatility and communication features make it highly suited for telemetry, remote command processing, and experimental payloads. Furthermore, its development ecosystem, support for OTA updates, and multiple variants allowed for flexibility depending on whether communication, camera support, or compact design was required.

For power-constrained subsystems such as the Electrical Power System (EPS) and basic environmental monitoring, ATmega328P and MSP430 were favorable choices. Their ultra-low power consumption, wide availability, and community support make them ideal for housekeeping functions that don’t demand high computational power. In the case of simpler logic, the ATtiny series can serve as backup controllers or watchdogs.

In contrast, for data-intensive operations such as image processing, neural inference, or SDR signal handling, more powerful platforms like the Teensy 4.1, Raspberry Pi 4, and Raspberry Pi 5 were evaluated. The Raspberry Pi 5, in particular, with its PCIe support, USB 3.0 interface, and Quad-core Cortex-A76 processor, opens opportunities for AI-based applications, real-time payload management, and high-resolution imaging. However, due to its power requirements and potential for thermal issues, it is best suited for ground testing or highly specialized onboard functions with adequate energy budgets.

While boards like the STM32F103C8T6 and the PIC microcontrollers offer reliable performance and are well-documented in the aerospace industry, they often require additional configuration or custom toolchains that may slow down development for a student-level project. Moreover, their peripheral integration tends to be less straightforward compared to modern development boards like the ESP32 or Raspberry Pi Pico.

The final proposed configuration for the satellite includes;

1. **Raspberry Pi 4**

The Raspberry Pi 4 was chosen as the core controller for the payload, onboard computer (OBC), and communication subsystems. With its quad-core Cortex-A72 processor and up to 8 GB of RAM, the RPi 4 provides the necessary processing throughput for high-resolution data collection, processing, and communication-intensive tasks. Its Linux support and rich set of peripherals make it ideal for handling mission-critical operations that require multitasking, local storage, and real-time control of onboard experiments or communication relays.

2. **ESP32**

The ESP32 –WROOM - 32 will serve as the controller for the telemetry subsystem, where its built-in Wi-Fi/Bluetooth, low power modes, and real-time processing capabilities are highly advantageous. It will be responsible for collecting system health data and transmitting it efficiently to the onboard computer and subsequently to the ground station. Additionally, the ESP32 will offer supplementary control for the EPS, enabling intelligent power usage tracking, switching, and battery state management during flight.

3. **MSP430**

Given its ultra-low power consumption and reliability, the MSP430 will act as a redundant or complementary microcontroller, particularly for the EPS. Should the ESP32 become overburdened or require isolation for real-time power control tasks, the MSP430 can autonomously manage critical energy management functions. This adds an extra layer of redundancy to the power system, which is vital in space missions.

4. **Raspberry Pi 5**

On the ground, the Raspberry Pi 5 will be deployed at the ground station. Its vastly improved I/O capabilities, PCIe support, USB 3.0, and enhanced CPU performance make it an excellent choice for tasks such as receiving downlinked data, processing telemetry, controlling uplinks, and visualizing satellite data in near-real-time. Its compatibility with modern SDR tools and AI frameworks will allow for expanded experimentation and flexibility in ground operations.

In conclusion, our microcontroller selection strategy balances performance, power, cost, and development convenience. By leveraging the strengths of each platform in their optimal roles, the Tafiti Project ensures that all satellite subsystems are adequately supported, robustly integrated, and primed for success in both simulation and deployment environments.