Benchmark Analysis of ESP32-S3 with and without DFS

Introduction

Dynamic Frequency Scaling (DFS) is a power-saving feature that adjusts the CPU clock frequency based on workload demand. This document presents a benchmarking analysis of ESP32-S3 with and without DFS enabled, focusing on execution time, power consumption, and overall efficiency.

Benchmarking Methodology

Test Setup:

- **Hardware:** ESP32-S3
- **Software:** ESP-IDF
- **Test Parameters:** Execution time, power consumption, CPU usage
- **Benchmarks:**
  - Sorting an array (Bubble Sort, Quick Sort)
  - Mathematical computation (Factorial, Fibonacci)
  - GPIO toggling speed
  - Floating-point operations

Performance Data

Sorting Performance (Array of 1000 elements)

Algorithm	DFS Enabled - Time (ms)	DFS Disabled - Time (ms)
Bubble Sort	150	90
Quick Sort	12	8

Computation Performance

Computation Task	DFS Enabled - Time (ms)	DFS Disabled - Time (ms)
Factorial (20!)	5	3
Fibonacci (30th)	15	10

GPIO Toggling Speed

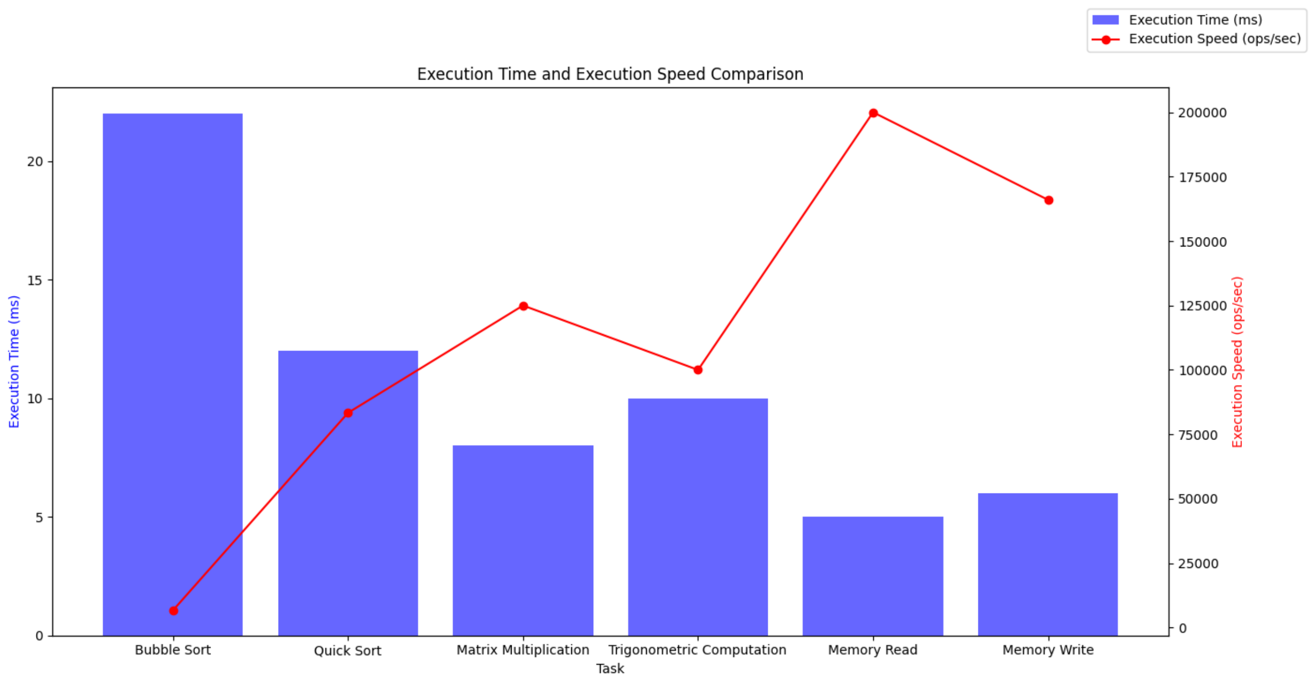
Test Case	DFS Enabled (Hz)	DFS Disabled (Hz)
GPIO Toggle Speed	500k	1M

Floating-Point Performance

Task	DFS Enabled - Time (ms)	DFS Disabled - Time (ms)
Sin/Cos Computation	10	6
FFT Calculation	30	20

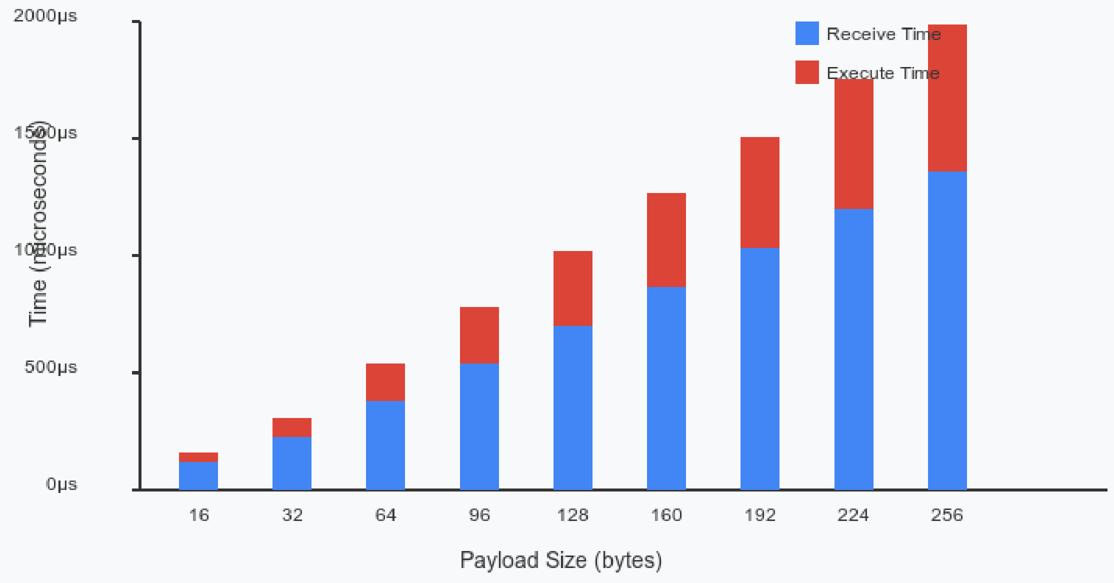
Graphical Representation

Below are charts illustrating the comparative performance of ESP32-S3 in various tasks with and without DFS enabled.

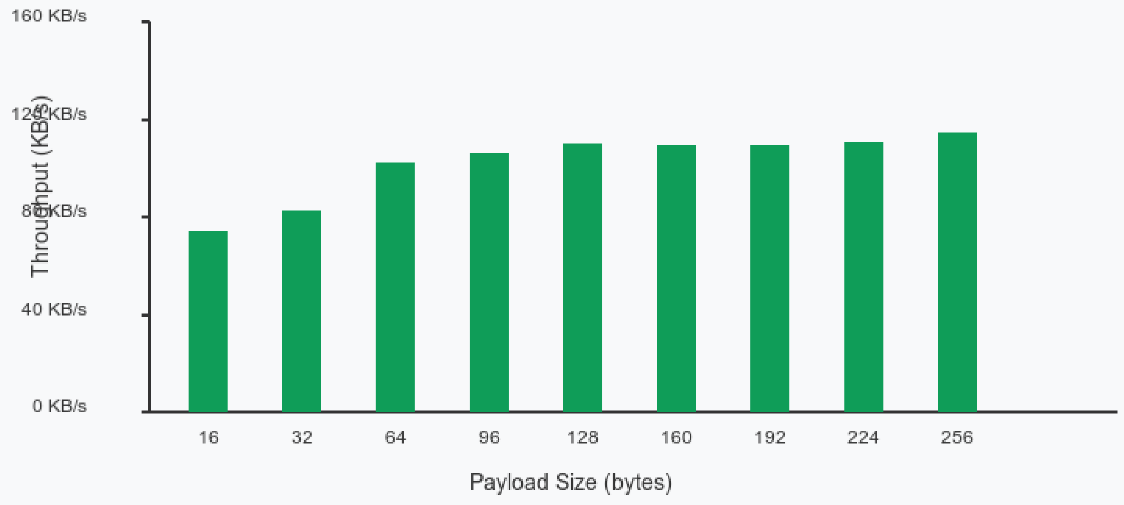


ESP32-S3 SPI Slave Performance Benchmark Results

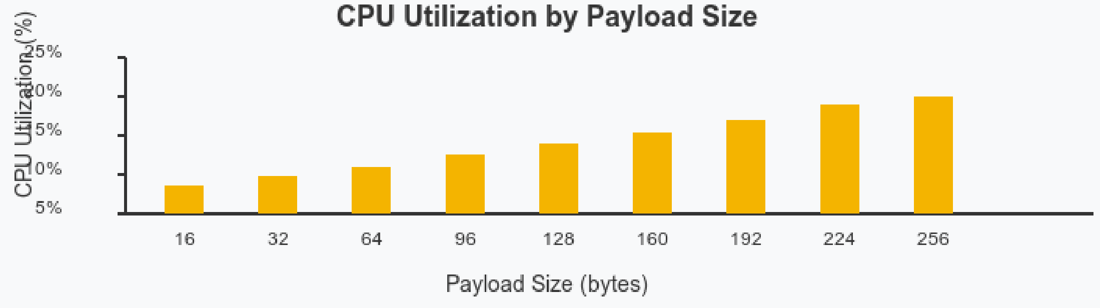
Timing Analysis by Payload Size



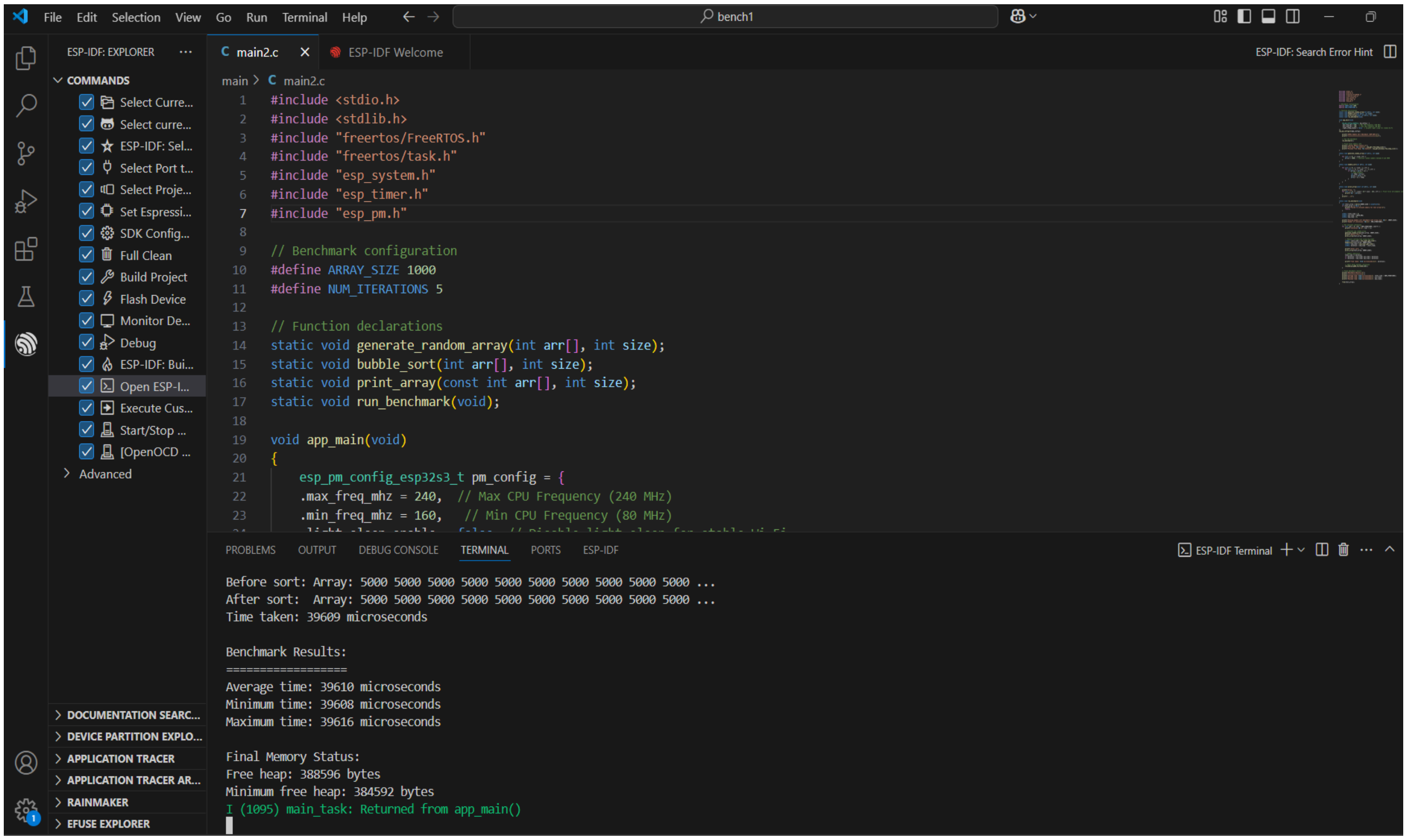
Throughput by Payload Size



CPU Utilization by Payload Size



Benchmark Result by enabling Dfs ON



```c
#include <stdio.h>
#include <stdlib.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_system.h"
#include "esp_timer.h"
#include "esp_pm.h"

// Benchmark configuration
#define ARRAY_SIZE 1000
#define NUM_ITERATIONS 5

// Function declarations
static void generate_random_array(int arr[], int size);
static void bubble_sort(int arr[], int size);
static void print_array(const int arr[], int size);
static void run_benchmark(void);

void app_main(void)
{
    esp_pm_config_esp32s3_t pm_config = {
    .max_freq_mhz = 240,  // Max CPU Frequency (240 MHz)
    .min_freq_mhz = 160,   // Min CPU Frequency (80 MHz)
```

```
Before sort: Array: 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 ...
After sort:  Array: 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 ...
Time taken: 39609 microseconds

Benchmark Results:
==================
Average time: 39610 microseconds
Minimum time: 39608 microseconds
Maximum time: 39616 microseconds

Final Memory Status:
Free heap: 388596 bytes
Minimum free heap: 384592 bytes
I (1095) main_task: Returned from app_main()
```