Методика технического анализа возможностей систем реального времени на различных аппаратных платформах

Евгений Таран - разработчик

при сотрудничестве с Евгением Синельниковым

ПЛК и Linux

- Отправная точка исследования – продукт «Альт Контроллер» (адаптация ОС Альт для ПЛК)
- ПЛК – специализированные вычислительные системы для управления технологическими процессами

Существующее противоречие

- Ключевая характеристика ПЛК – детерминированность отклика на критические события
- Linux – система общего назначения, не гарантирующая детерминированного поведения

Путь преодоления — PREEMPT_RT

- Установка приоритетов планировщика для realtime-процесса и irq-потоков
- Привязка процесса к изолированному ядру
- Блокировка страниц оперативной памяти выделенной процессу

Цель исследования

Цель исследования — разработка методики, которая позволит провести оценку возможности решения задач, требующих реального времени на конкретной программно-аппаратной платформе.

Другими словами, методика должна давать ответ, способна ли комбинация “Одноплатник - версия ОС - настройки ОС” обеспечить требуемый уровень скорости и надежности работы процесса

Измерительный стенд. Метрики

Латентность – интервал времени между возникновением внешнего события и началом его обработки целевой системой.

$$L_i = t_{response,i} - t_{event,i}$$

Измерительный стенд. Метрики

Джиттер – вариативность (нестабильность) временных интервалов между последовательными событиями.

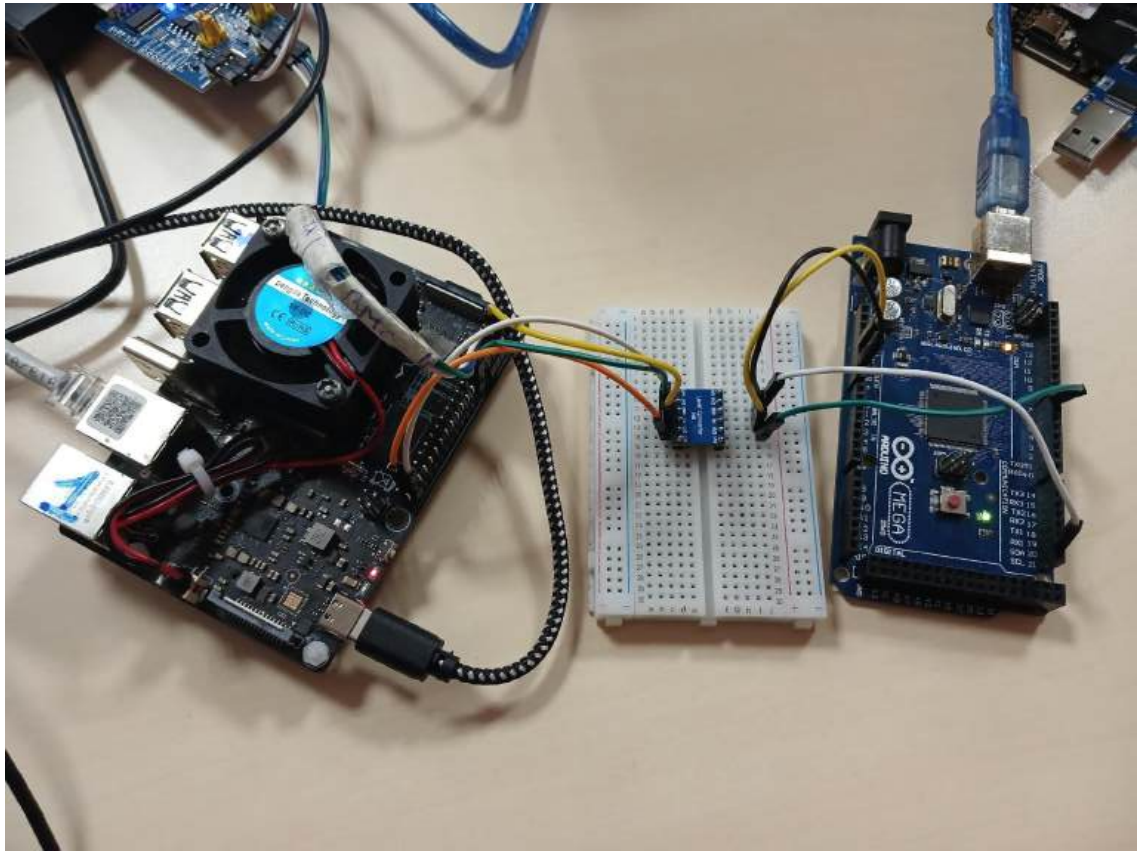
$$J_i = |L_i - \bar{L}| \qquad \sigma_J = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(L_i - \bar{L})^2}$$

Измерительный стенд. Компоненты

Архитектура стенда заимствована из статьи «Choosing between Xenomai and Linux for real-time applications» и состоит из 3 компонентов:

- Измеритель на базе Arduino Mega
- Тестируемое устройство (одноплатник с GPIO)
- ПК исследователя, собирающий данные от измерителя

Измерительный стенд

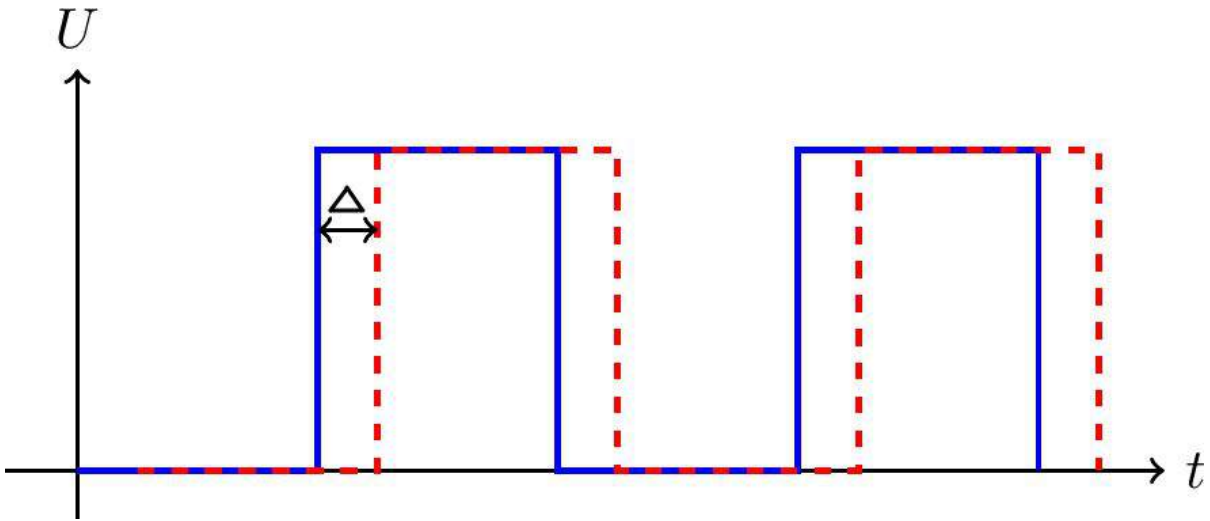


Перечень протестированных плат

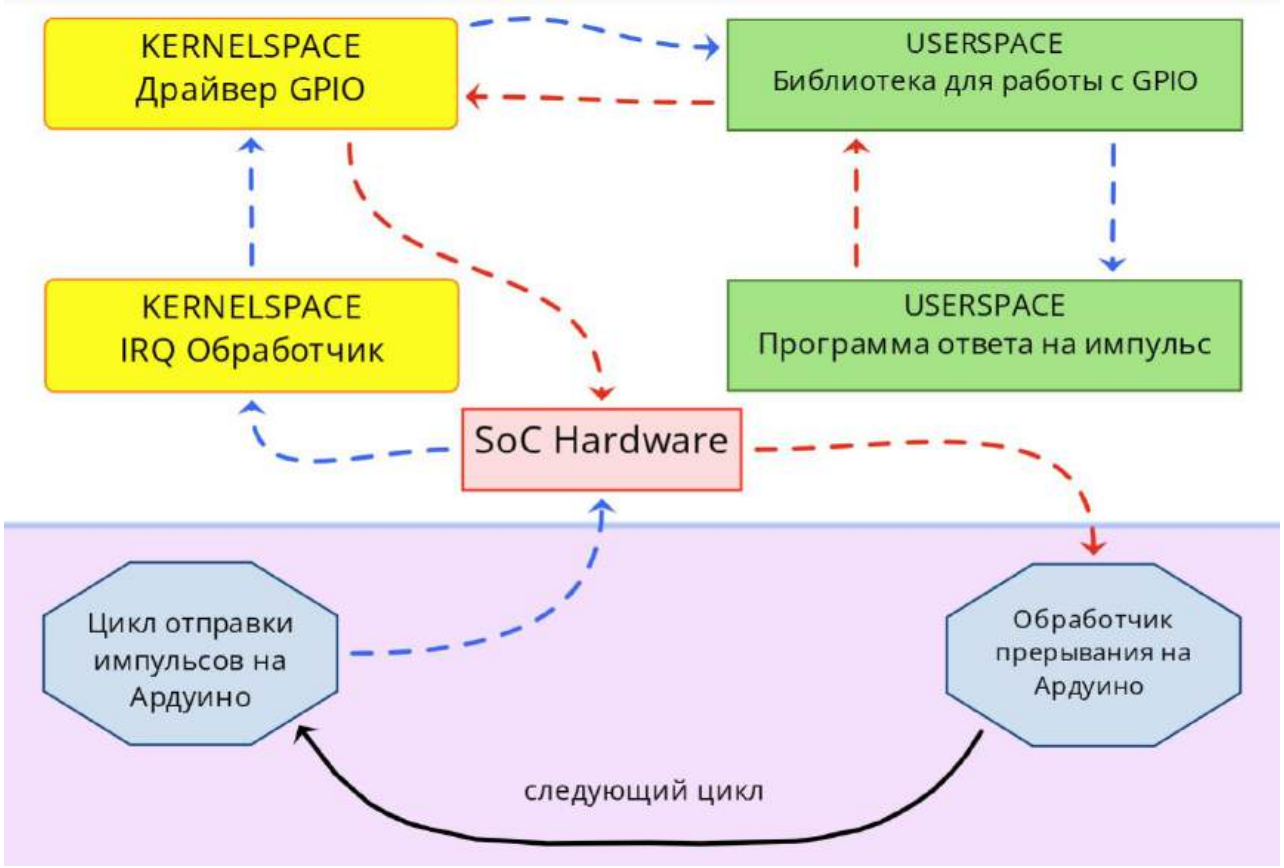
Архитектура	Одноплатник
riscv64	Starfive Visionfive 2
	Lichee RV Dock
	MangoPI MQ Pro
aarch64	Radxa Rock 5b
	RockPI 4A
	RepkaPI 4
e2k	БВЦ АРМ
loongarch	БЦВМ АРМ

Принцип измерения

- **Arduino Mega генерирует** на GPIO периодический сигнал с заданным интервалом **на линии A**
- **Тестируемая платформа считывает** изменение сигнала **на линии A** и формирует **ответный сигнал на линии B**



Тракт сигнала



Наш путь запуска realtime-процесса

Возможность	Утилита	Функция C
Приоритет IRQ-потока	chrt -f -p 99	sched_setscheduler(pid, SCHED_FIFO, &(struct sched_param){.sched_priority=99}
Приоритет GPIO-хэндлера	chrt -f -p 80	pthread_setschedparam(pthread_self(), SCHED_FIFO, &(struct sched_param){80})
Блокировка памяти процесса	—	mlockall(MCL_CURRENT | MCL_FUTURE)
Изоляция CPU-ядра	isolcpus=N (параметр ядра) + taskset -c N ./prog	sched_setaffinity(0, sizeof(cpu_set_t), &cpus)

Параметры измерительной сессии

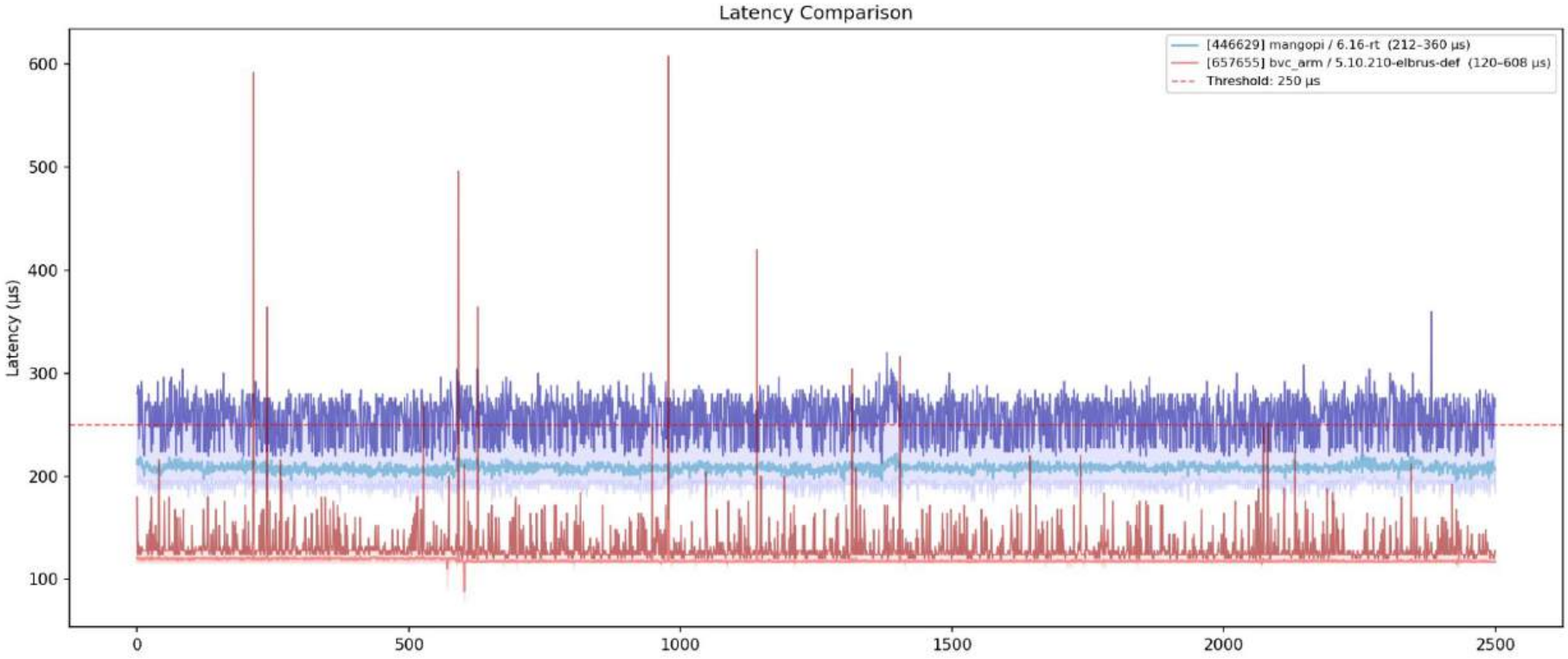
Параметр	Значение	Описание
Количество групп	2500	Групп в одной сессии
Размер группы	1500	Замеров в одной группе
Кол. худших значений	75 (5%)	Выбранные задержки
Частота переключения	300 мкс	Полупериод меандра
Порог латентности	250 мкс	Классификация превышений

Анализ задержки сфокусирован на 5% наихудших значений в каждой группе измерений

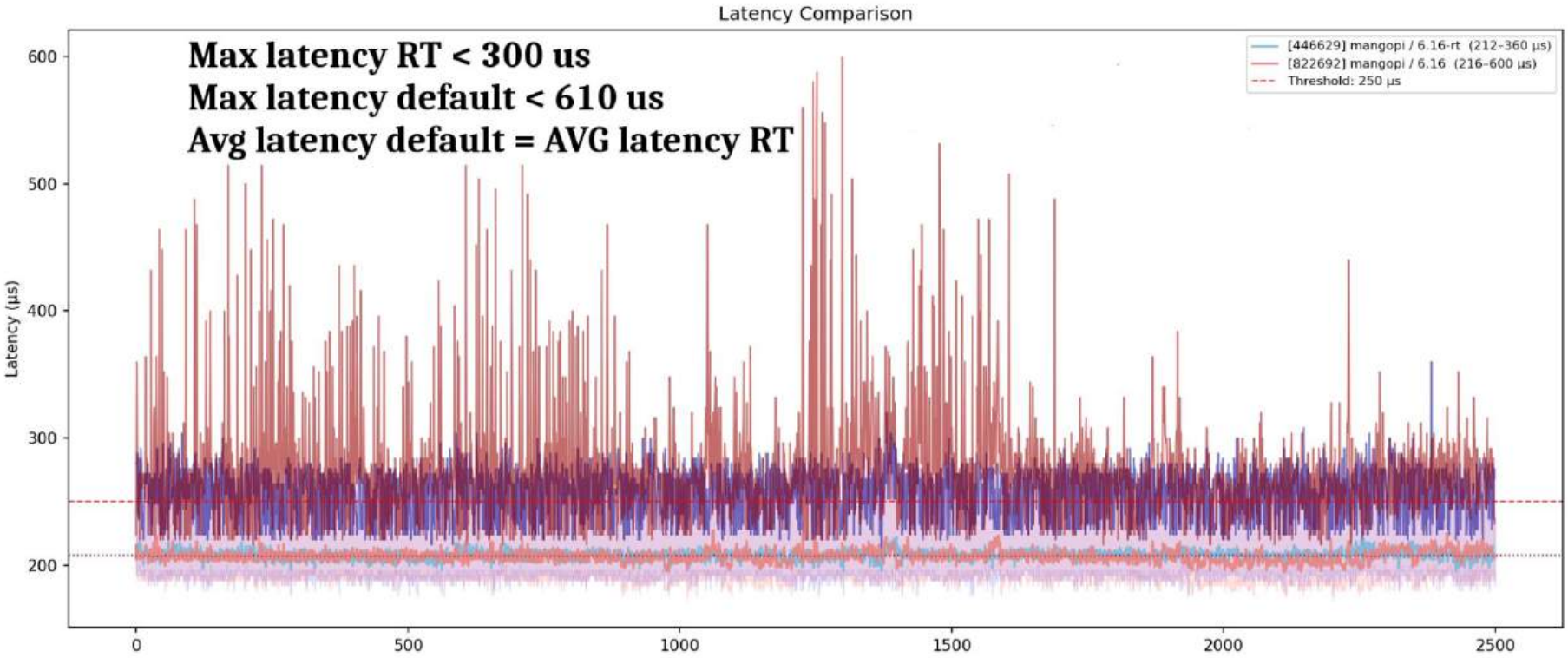
Таблица с результатами

SoC	Конфиг	Превыш.	Макс.	Ср.5%	Ср. RMS
MangoPI	default	3500	550	200	15
MangoPI	default+stress	6000	610	210	20
MangoPI	rt	1000	300	200	15
MangoPI	rt+stress	1600	330	210	15
BVC ARM	default	30	600	140	40
BVC ARM	default+stress	27	600	120	35
BVC ARM	rt	0	100	40	5
BVC ARM	rt+stress	99	150	45	5

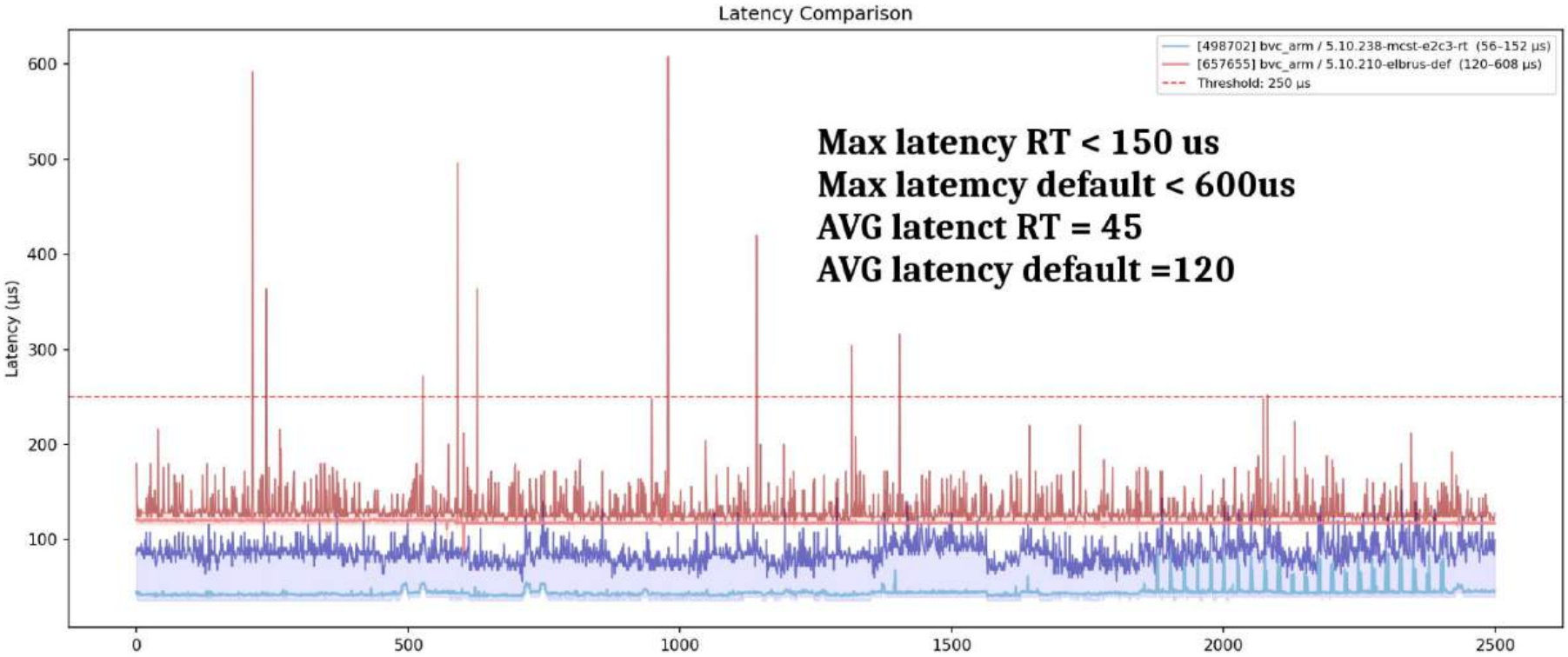
Сравнение двух плат под нагрузкой



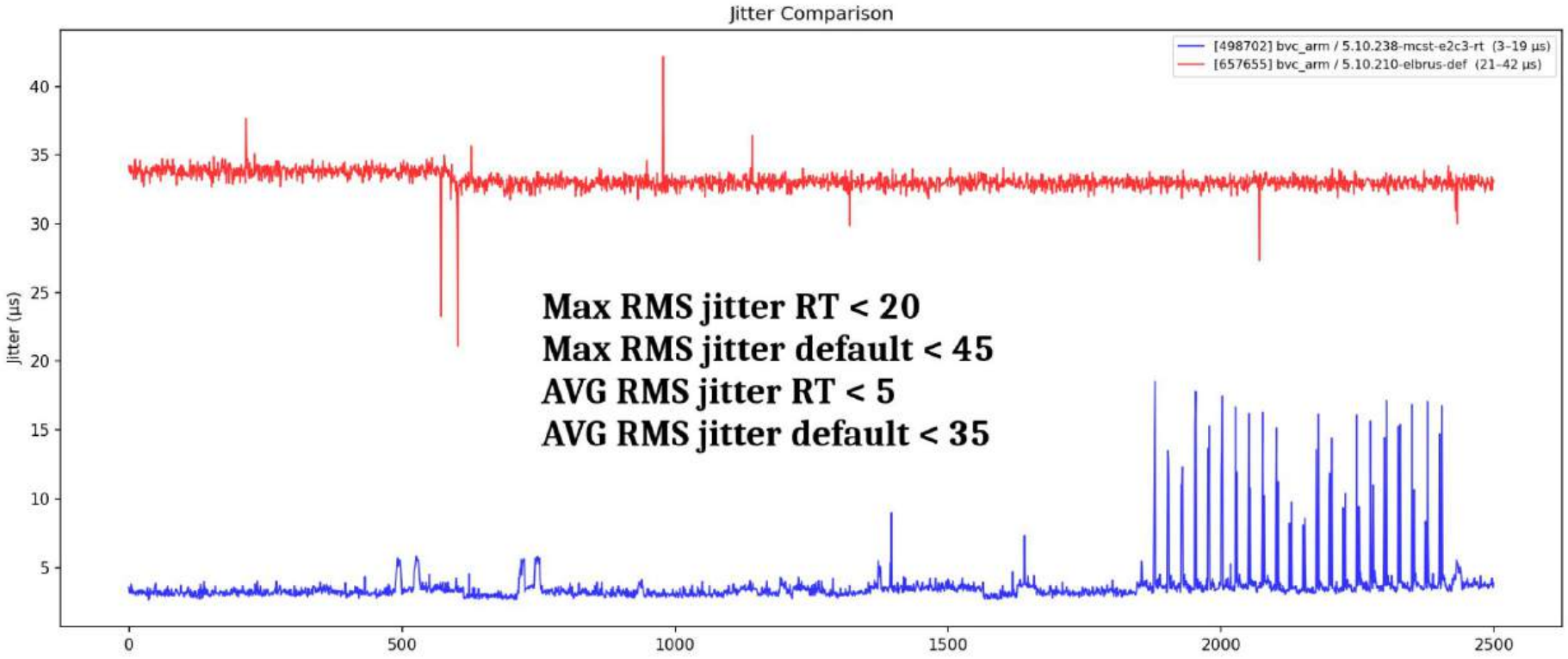
Сравнение конфигураций на MangoPI



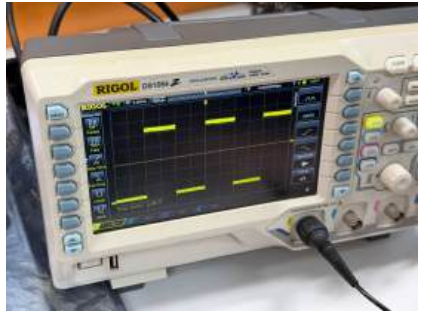
Сравнение конфигураций на BVC ARM

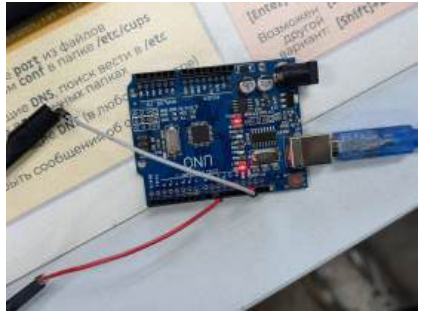


Сравнение конфигураций на BVC ARM



Интеграция в учебный процесс





Интеграция в учебный процесс

Цикл лабораторных работ по дисциплине «Системы реального времени»:

- Лабораторные по созданию генераторов импульсов на Arduino
- Лабораторная по созданию измерительной системы на базе Arduino
- Лабораторная по сравнительному анализу генераторов с помощью измерительной системы
- Исследование детерминизма Linux на Lichee RV

Итоги исследования

- Все исследуемые платы во время тестов не вышли по показателям latency за порог в 1 мс
- Подтверждена эффективность использования патча PREEMPT_RT для решения задач реального времени
- Методика позволяет провести первичную оценку соответствия исследуемой аппаратной платформы требованиям задач мягкого реального времени и может быть адаптирована под реальные прикладные задачи

Спасибо за внимание!

Репозиторий
с исходниками
и артефактами
исследования

Подвал презентации…

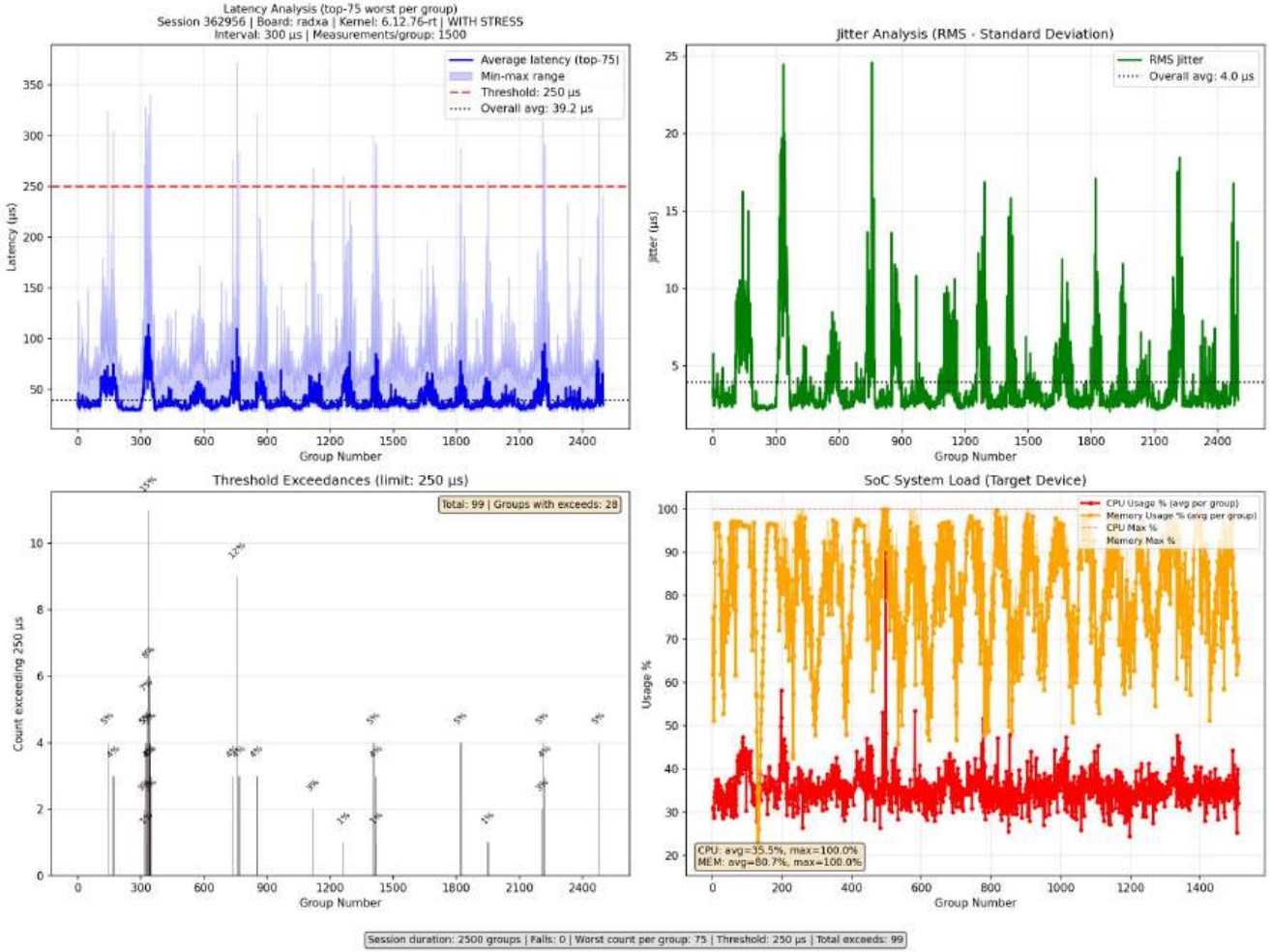

Latency Analysis (top-75 worst per group)
Session 362956 | Board: radxa | Kernel: 6.12.76-rt | WITH STRESS
Interval: 300 µs | Measurements/group: 1500
Average latency (top-75)
Min-max range
Threshold: 250 µs
Overall avg: 39.2 µs
Latency (µs)
Group Number
Jitter Analysis (RMS - Standard Deviation)
RMS Jitter
Overall avg: 4.0 µs
Jitter (µs)
Group Number
Threshold Exceedances (limit: 250 µs)
Total: 99 | Groups with exceeds: 28
Count exceeding 250 µs
Group Number
SoC System Load (Target Device)
CPU Usage % (avg per group)
Memory Usage % (avg per group)
CPU Max %
Memory Max %
CPU: avg=35.5%, max=100.0%
MEM: avg=80.7%, max=100.0%
Usage %
Group Number
Session duration: 2500 groups | Fails: 0 | Worst count per group: 75 | Threshold: 250 µs | Total exceeds: 99

Таблица с результатами

SoC	Конфиг	Превыш.	Макс.	Ср.5%	Ср.RMS
Lichee	6.12	45812	335	249	17
Lichee	6.12+stress	95198	575	252	18
Lichee	6.12-rt	109726	365	255	18
Lichee	6.12-rt+stress	104533	475	255	18
Radxa	6.12	0	150	29	2.5
Radxa	6.12+stress	3647	350	66	11.2
Radxa	6.12-rt	0	125	36	4
Radxa	6.12-rt+stress	99	360	39	4
VF2	6.18	5033	350	164	26
VF2	6.18+stress	71240	600	185	36
VF2	6.18-rt	3469	310	172	28
VF2	6.16-rt+stress	42934	520	149	27

Задачи исследования

1) Построение легко адаптируемого и воспроизводимого тестового стенда, применимого для различных технических решений
2) Выбор метрик оценки системы с точки зрения задач реального времени
3) Агрегация полученных данных и их анализ