SCIENCE
PASSION
TECHNOLOGY

One Solution to Rule Them All

ATTEST as Unified Testing Solution for Programming Courses
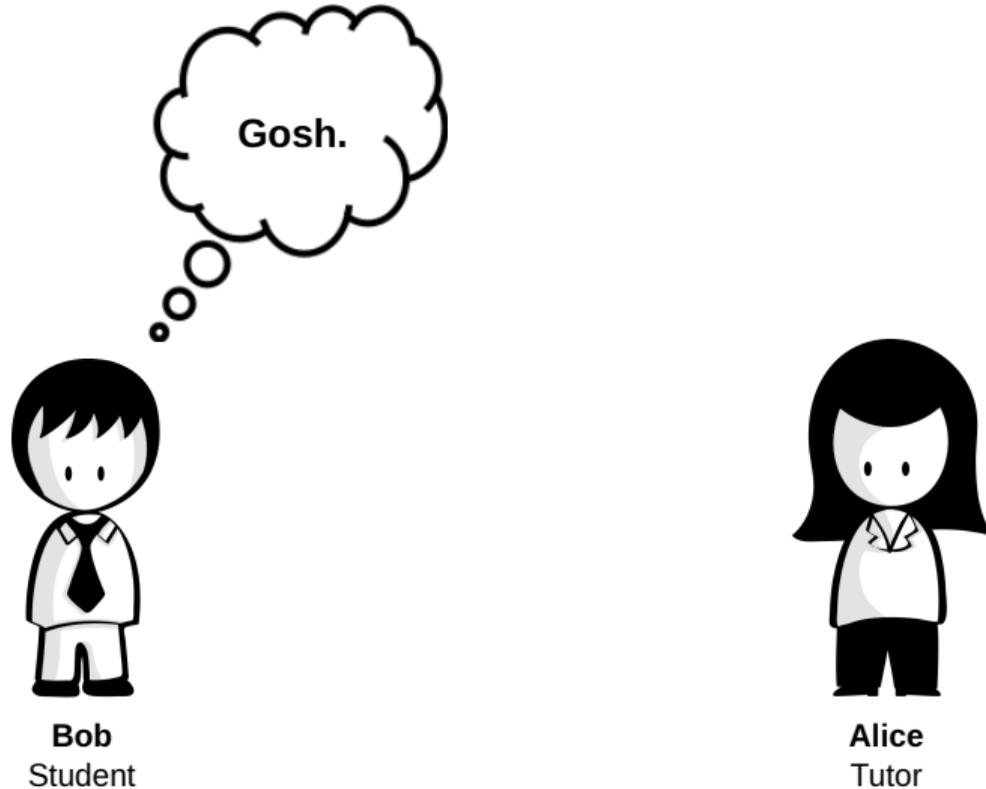
Meinhard Kissich

FGBS '23

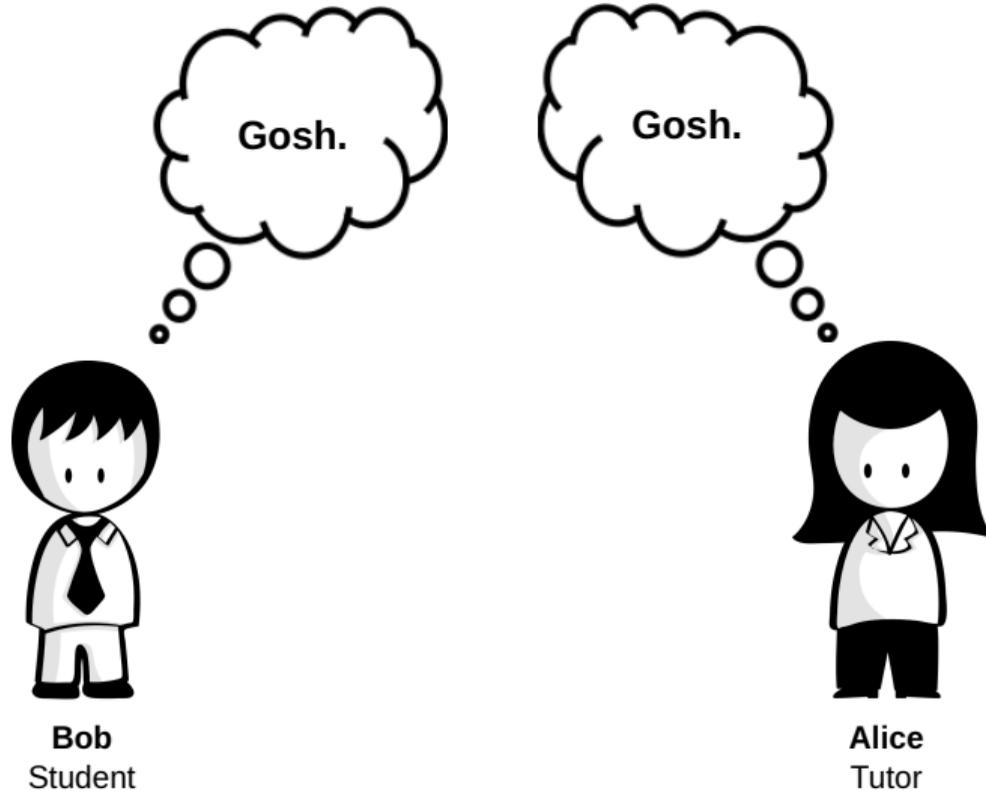Meinhard Kissich, Kristóf Kanics, Klaus Weinbauer, Tobias Scheipel, Marcel Baunach, Graz University of Technology
FGBS '23

Meinhard Kissich, Kristóf Kanics, Klaus Weinbauer, Tobias Scheipel, Marcel Baunach, Graz University of Technology
FGBS '23

Meinhard Kissich, Kristóf Kanics, Klaus Weinbauer, Tobias Scheipel, Marcel Baunach, Graz University of Technology
FGBS '23

Bob
Student

Credits [1]

Credits [1]

Meinhard Kissich, Kristóf Kanics, Klaus Weinbauer, Tobias Scheipel, Marcel Baunach, Graz University of Technology
FGBS '23

Credits [1]

Meinhard Kissich, Kristóf Kanics, Klaus Weinbauer, Tobias Scheipel, Marcel Baunach, Graz University of Technology
FGBS '23

Bob
Student

Alice
Tutor

Credits [1]

ATTEST: Automated and Thorough Testing of Embedded Software in Teaching

Meinhard Kissich
meinhard.kissich@tugraz.at
Graz University of Technology
Graz, Austria

Klaus Weinbauer
klaus.weinbauer@student.tugraz.at
Graz University of Technology
Graz, Austria

Marcel Baunach
baunach@tugraz.at
Graz University of Technology
Graz, Austria

ABSTRACT

Dependability requirements are getting increasingly stringent in embedded systems, demanding highly skilled developers. One cru-

to continue [1, 12], demanding a considerable amount of well-engineered embedded software to cope with increasing dependability requirements. In such cases, embedded Operating Systems (OSs)

Paper [2]

Meinhard Kissich, Kristóf Kanics, Klaus Weinbauer, Tobias Scheipel, Marcel Baunach, Graz University of Technology
FGBS '23

Automated and **T**horough **T**esting of **E**mbedded **S**oftware in **T**eaching

Meinhard Kissich, Kristóf Kanics, Klaus Weinbauer, Tobias Scheipel, Marcel Baunach, Graz University of Technology
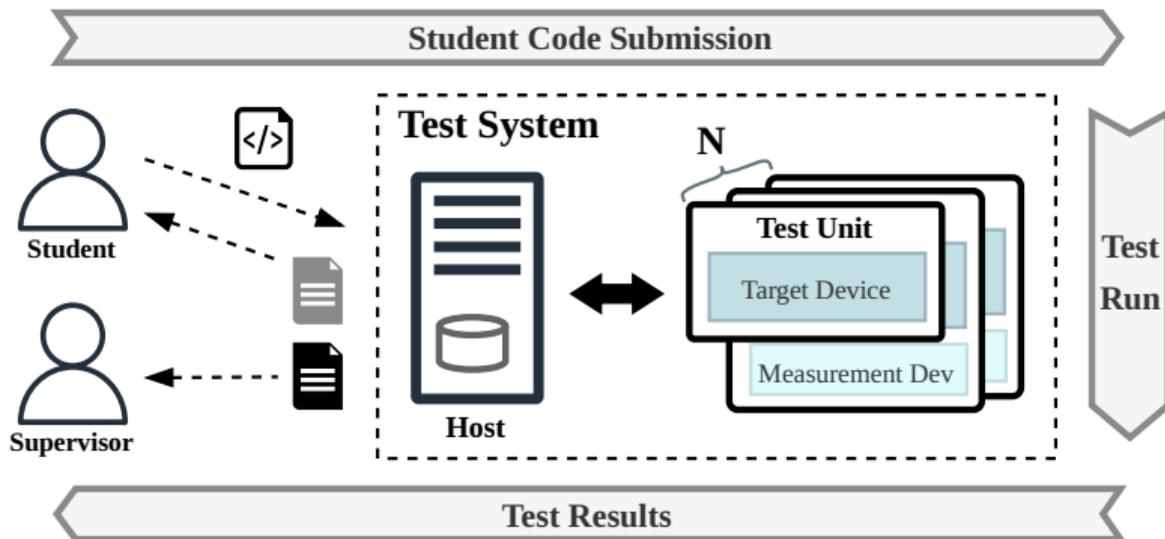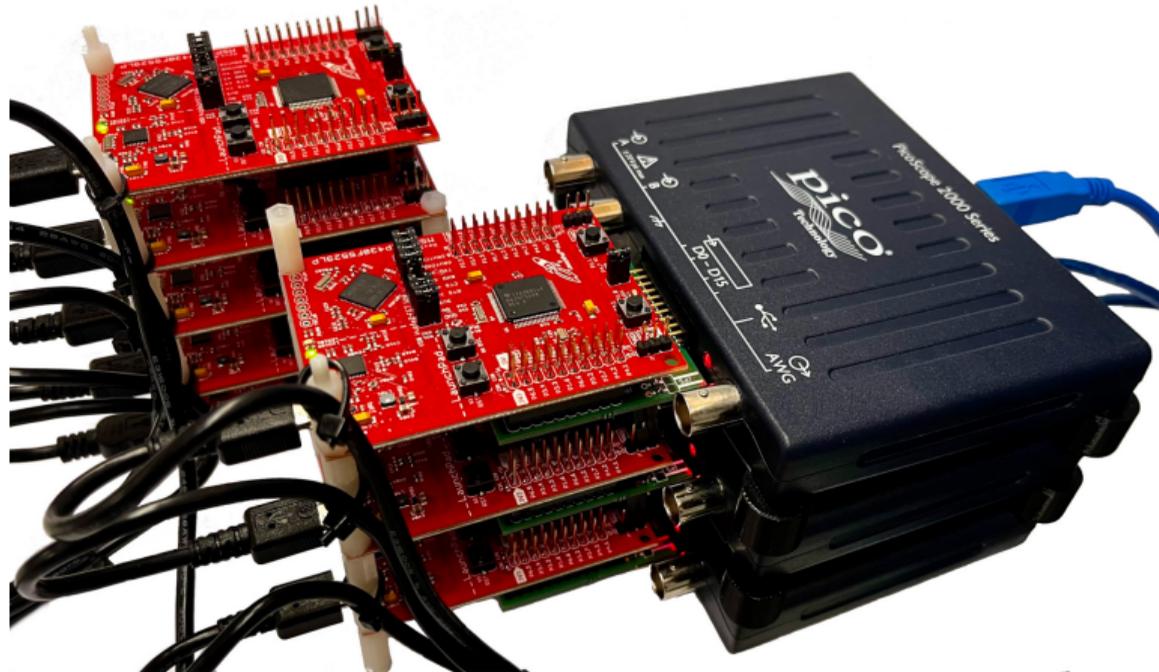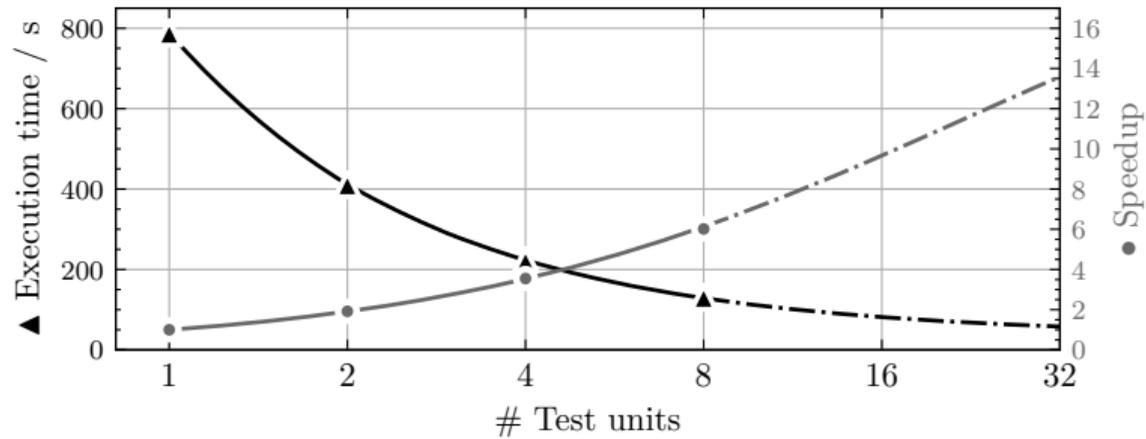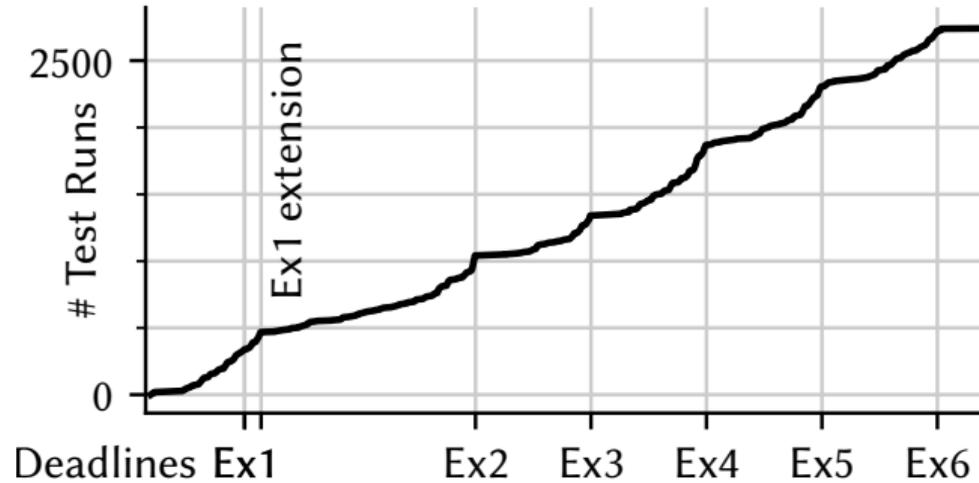
Course Details

- Original application

- Embedded operating system from scratch (SmartOS [3])

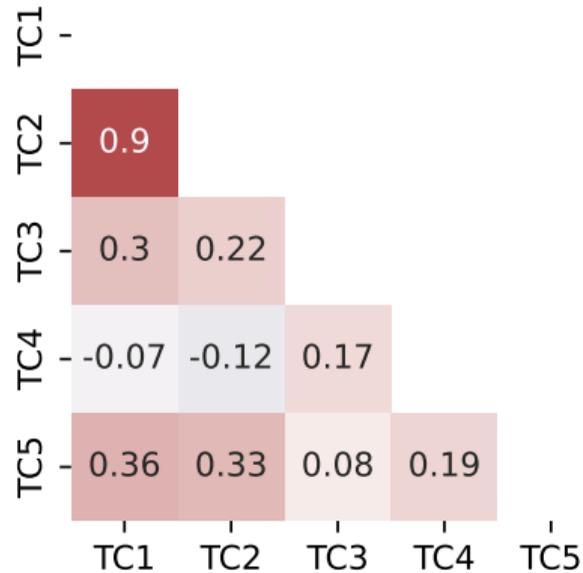- 40 to 70 participants (48 last semester)
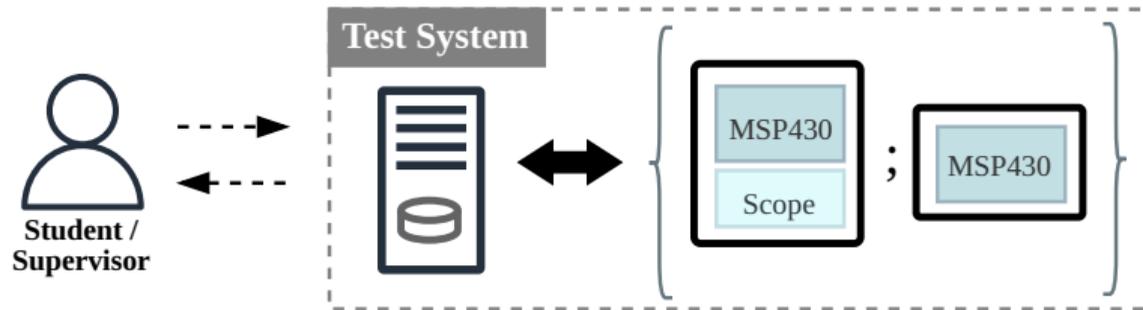
- 6 exercises

Testing

- Test on the actual target hardware

- No simulation models

- 13 to 57 test cases (depending on course progress)

- A *lot* of test runs

Test Runs

Test Case Optimization

Test System

Student /
Supervisor

MSP430

Scope

;

MSP430

Static Tests	**Message Tests**	**Probe Tests**
Binary size *Coding style*	Correctness tests	Timing checks Performance benchmarks

Exercise 1

Id	Test Case	Deploy Process	Result
1	size	Completed	158872.0
101	sleep 100 ms	Completed	99741.725
102	getCurrentTime overhead	Completed	3.8285
103	hello world timing	Completed	1
104	TA0R getCurrentTime	Completed	1
105	miss interrupt	Completed	1
106	close to TA0R OV	Completed	1
107	multiple close and miss	Completed	1

📄 **README.md**

Test Results for Group 1

Tested Commit: 02cbe8ce
Commit Time: 15.06.2023 13:11:03
Commit Message: 62 - 5baa8d8

Exercise 0

Test Case 10 ✅

Result: 38.7965

Test Case 11 ✅

Result: 1.178

Test Case 12 ✅

Result: 521.9935

Test Case 101 ✅

Result: 99741.725

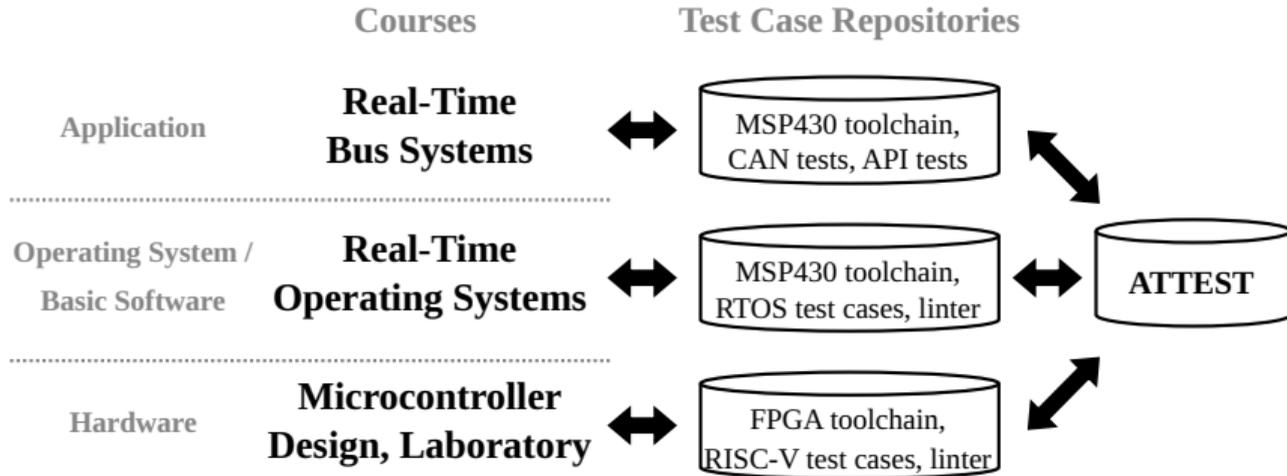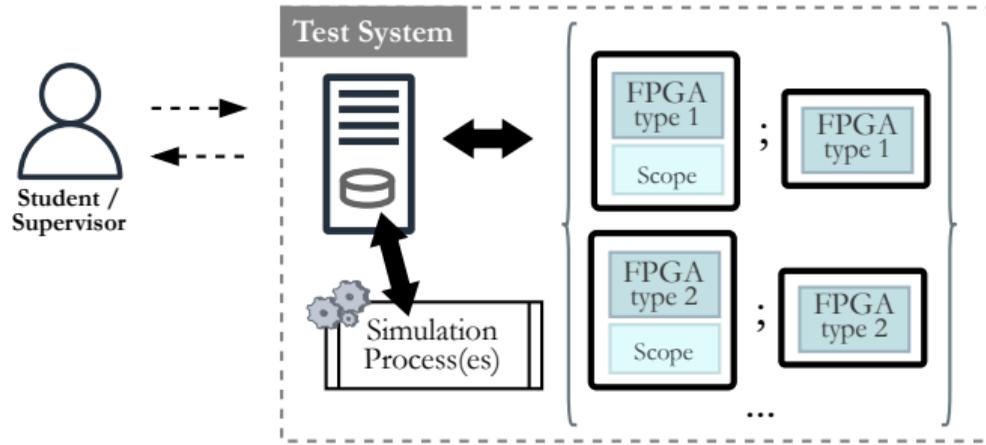**Test Case 102** ✅

Result: 3.8285

Test Case 103 ✅

Result: 1.0

▼ Program Output

```
TESTCASE BEGIN
Hello World!
Hello World!
Hello World!
Hello World!
```

TU Graz

Unified Testing Solution

15

	Courses	**Test Case Repositories**
Application	**Real-Time Bus Systems** ⟷	MSP430 toolchain, CAN tests, API tests
Operating System / Basic Software	**Real-Time Operating Systems** ⟷	MSP430 toolchain, RTOS test cases, linter	⟷ **ATTEST**
Hardware	**Microcontroller Design, Laboratory** ⟷	FPGA toolchain, RISC-V test cases, linter

Static Tests	Sim Tests	Message Tests	Probe Tests
Coding style	—————RISC-V ISA test programs—————		
	————Correctness tests————		
			Timing analysis
	—————Smoke tests—————		
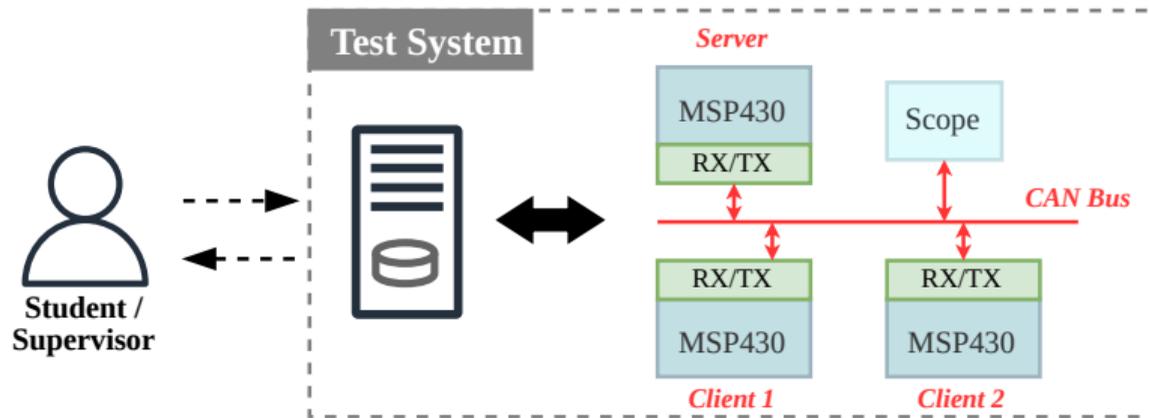
Static Tests	Message Tests	Probe Tests
Coding style	Correctness tests	
		Timing analysis

Expected Outcomes

⊕ Adopt to further courses

⊕ Reduce maintenance effort

⊕ Lower extraneous cognitive load

⊕ Improve learning experience

Future Work

- **Deadline Optimization:** relieve crowded weeks

- **Test Case Granularity:** check test correlation

- **Workload Balancing:** re-balance exercises

- **Device Monitoring:** track flash life time

- **Extensions:** calibration, optimized priorities, ...

- **Improvements:** consider student feedback

Questions?

Meinhard Kissich

Kristóf Kanics

Klaus Weinbauer

Tobias Scheipel

Marcel Baunach

Backup Slides

Name	Description	Approach
(RQ1)	host system independent	Docker, Python
(RQ2)	diverse test types	{message, probe, size} tests
(RQ3)	secure and well-known interface	Git
(RQ4)	modular and extensible framework	modular, abstraction layers
(RQ5)	scalable by parallelization	worker pattern
(RQ6)	heterogeneous test units	state awareness, scheduling
(RQ7)	customized reports	report generation
(RQ8)	testing of the test system	testing suite
(F1)	predictive test units maintenance	flash cycle counter
(F2)	usage supervision	scheduling logic
(F3)	runtime configurability	config variables, config files
(F4)	easy test unit installation	automatic test unit detection

Bibliography I

[1] SAP AppHaus, SAP SE, **Scenes,** [Online] `http://experience.sap.com/designservices/scenes` (Sept. 2023), 2019.

[2] M. Kissich *et al.*, **ATTEST: Automated and Thorough Testing of Embedded Software in Teaching,** ECSEE, 2023.

[3] T. Scheipel *et al.*, **SmartOS: An OS Architecture for Sustainable Embedded Systems,** Tagungsband des FG-BS Frühjahrstreffens, 2022.