Institute of Technical Informatics ■ Inffeldgasse 16/I, 8010 Graz, Austria ■ *office@iti.tugraz.at*

Automated and Thorough Testing of Embedded Software in Teaching

Meinhard Kissich, Klaus Weinbauer, Marcel Baunach

Institute of Technical Informatics, Graz University of Technology
{meinhard.kissich, baunach}@tugraz.at, klaus.weinbauer@student.tugraz.at

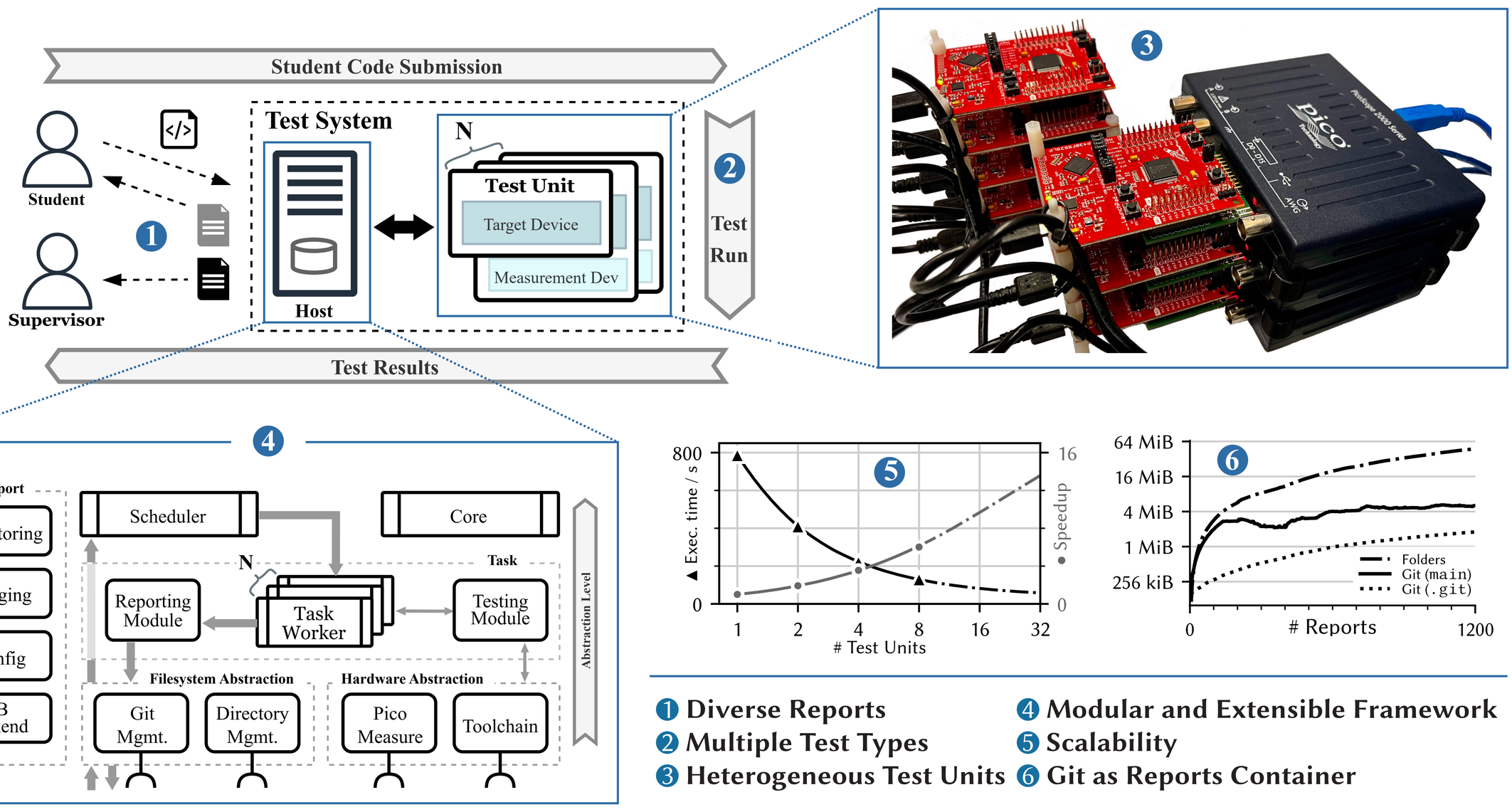
Abstract

Dependability requirements are getting increasingly stringent in embedded systems, demanding highly skilled developers. One crucial point in building up expertise is getting precise feedback in programming courses at university to recognize flaws and learn from mistakes. Depending on the assignment and learning outcome, the assessment may include testing for the implementation's completeness, correctness, performance, and robustness. A timely and in-depth review for a large number of course participants relies on test automation. However, embedded software often includes hardware-dependent code that can only be executed on the target device. Thus, we provide an open-source and remote hardware-in-the-loop testing solution with pre-defined test cases for embedded software particularly designed for teaching in university courses. This paper defines and elaborates on the requirements, gives an insight into design decisions, and evaluates the test system on metrics of our *Real-Time Operating Systems* course.

Requirements

- **Educational Requirements**
  - ↳ **Support building up expertise** through precise and timely feedback
  - ↳ **In-depth feedback reports on submissions** completeness, correctness, performance, robustness
  - ↳ **Large number of participants**
- **Architectural Requirements**
  - ↳ Embedded software: testing on **real hardware**
  - ↳ Dynamic & static tests: behavior, timing, sizes
  - ↳ Well-known interface: student ↔ test system
  - ↳ Modular, scalable, parallelized execution, ...

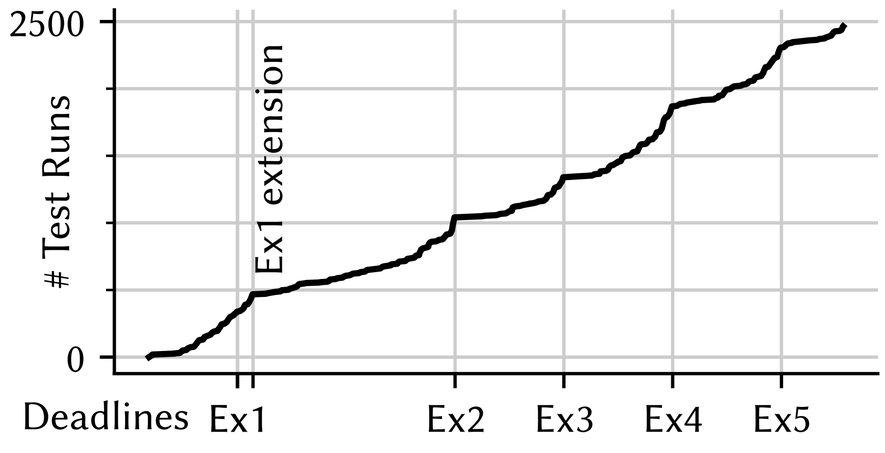
Design & Evaluation

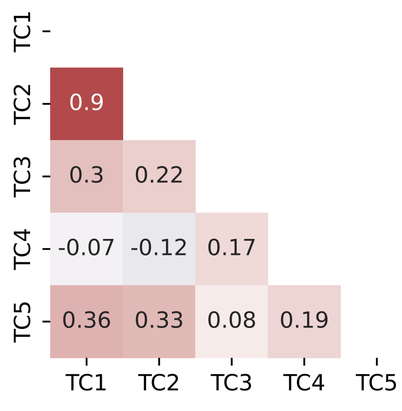


1. **Diverse Reports**
2. **Multiple Test Types**
3. **Heterogeneous Test Units**
4. **Modular and Extensible Framework**
5. **Scalability**
6. **Git as Reports Container**

Current Usage Data & Future Work

Real-Time Operating Systems Laboratory, TU Graz, 48 participants, 6 exercises





- **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

EAS

iti.tugraz.at/attest