



# The Peripheral Selftesting Shield Hunting Bugs in Loopback Mode

RIOT Summit 2023

18<sup>th</sup> September, 2023

Marian Buschsieweke

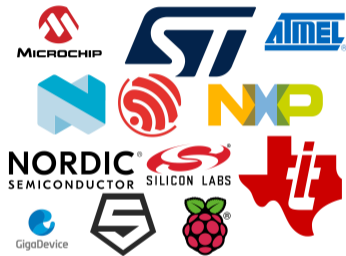
# Acknowledgments

---

This talk is largely based on:

Marian Buschsieweke, Mesut Güneş: **Automated Testing of Hardware Abstraction Layers on Microcontrollers**, *20th edition of the Fachgespräch Sensornetze (FGSN 2023)*

# Background



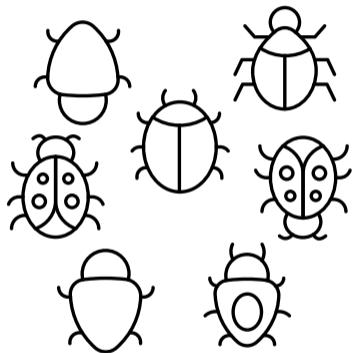
- RIOT supports many Microcontroller Unit (MCU) families by many vendors
  - for portable applications:
    - consistent APIs to access hardware required⇒ Hardware Abstraction Layers (HALs) for peripheral access
  - peripherals vastly differ between MCU vendors and even families
- ⇒ many peripheral drivers implementing the same HAL API

The company logos are registered trademarks of their respective owners. They do not indicate affiliation or endorsement and are used for illustration only.

# The Problem

- difficult to provide consistent behavior across MCUs
  - especially: corner cases & exotic features
- testing is difficult
  - software-only unit tests (not observing signals) not sufficient
  - attaching logic analyzer / scope to observe signals time consuming
  - manual effort to validate signals
- code often “fragile”
  - same peripheral + faster CPU  $\Rightarrow$  broken timing
  - different compiler optimization  $\Rightarrow$  broken timing
  - silicon bugs

$\Rightarrow$  rigorous and repeated testing required



QualityIcons from Noun Project, CC-BY-3.0

$\Rightarrow$  lots of bugs in peripheral drivers



# RIOT: Specific Challenges

---



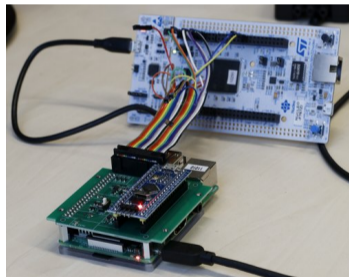
- RIOT:
    - OS for MCUs with focus on IoT
    - developed by diverse community across the globe⇒ distributed development model
  - access to tools and hardware varies
- ⇒ cannot rely on expensive testing equipment
- no single entity has access to all MCU families
- ⇒ distributed testing approach required

# Previous Work Testing HALs: PHiLIP on the HiL

Kevin Weiss et al.: **PHiLIP on the HiL: Automated Multi-platform OS Testing with External Reference Devices**, *arXiv:2107.07255 [eess.SY]*

- second MCU is connected to the device under test (DUT)
- directly interacts with the DUT
  - e.g. implements I<sup>2</sup>C peripheral to test I<sup>2</sup>C controller mode
- pros/cons:
  - + fully automated testing
  - + signal level verification of DUT
  - huge effort to setup

⇒ excellent fit for CI



Kevin Weiss et al., *arXiv:2107.07255 [eess.SY]*

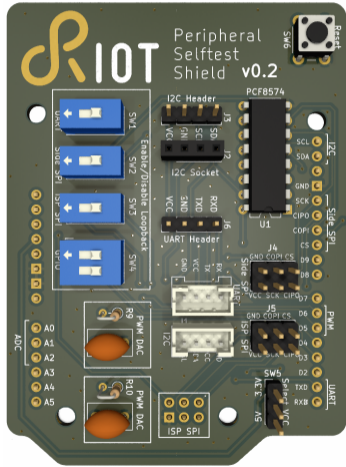
# Previous Work Testing HALs: Jumper Wires



oomlout, CC BY-SA 2.0

- loop output signal back to input signal
    - e.g. UART RXD to TXD, GPIO output to input, etc.
    - check: RX data = TX data
  - widely used (e.g. by Zephyr, RIOT, etc.)
  - pros/cons:
    - + fully automated testing
    - + really cheap
      - time to setup
      - high false positive due to wrong wiring
      - no signal level verification
- ⇒ Good fit for distributed testing approach
- But: Can we mitigate the issues?

# Proposal: Replace Wires with a PCB

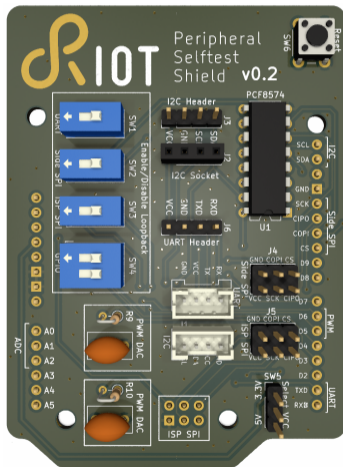


# Costs: Through-Hole Variant (“v0.2”)

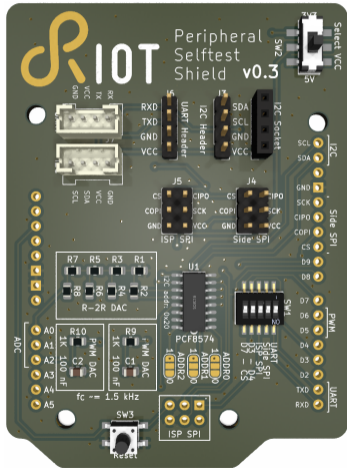
Costs for 1 pieces:

Position	Costs
PCB	\$ 2
Parts	< \$ 8
Total	< \$ 10

Note: Hand soldering doesn't scale for mass production, shipping costs ignored



# Costs: SMD Variant (“v0.3”)



Costs for 50 pieces:

Position	Costs
PCBs	\$ 19.40
SMD Parts	\$ 62.87
SMD Assembly	\$ 29.65
Mandatory THT Parts	\$ 10.56
Shipping	\$ 31.27
<b>Total</b>	<b>\$ 153.75</b>

≈ \$ 3.08 per board + ≈ 114 seconds of soldering<sup>a</sup>

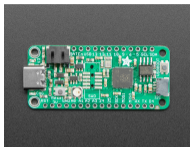
Note: Only THT parts to mate with the DUT are mandatory.

<sup>a</sup>3 seconds per pin

# Extension Board Format: Overview



SparkFun Electronics, CC BY 2.0



Adafruit Industries, CC BY-NC-SA 2.0



Quel.soler, CC BY-SA 4.0



Crash48, CC0 1.0

Extension Standard	I <sup>2</sup> C	SPI	UART	PWM	ADC	GPIO	Adoption
Arduino UNO	✓	✓	✓	✓	✓	✓	High
Arduino Mega	✓	✓	✓	✓	✓	✓	Medium
Arduino MKR	✓	✓	✓	✓	✓	✓	Low
Arduino Nano	✓	✓	✓	✓	✓	✓	Medium
Adafruit Feather	✓	✓	✓	✓	✓	✓	High
D1 Mini	✓	✓	✓	X	X	✓	Low
Microduino	✓	✓	✓	✓	✓	✓	Low
micro:bit	✓	✓	✓	✓	✓	✓	Low
Olimex UEXT	✓	✓	✓	X	X	X	Low

# Extension Board Format: Selection

MCU Family	Arduino UNO	ISP	Adafruit Feather
ATmega	Arduino UNO	✓	Feather 328P
EFM32/EFR32/EZR32	–	–	Thing Plus Matter
ESP32	–	–	HUZZAH32
ESP32S2	–	–	Feather S2
ESP32S3	–	–	Feather S3
ESP8266	–	–	HUZZAH
FE310	HiFive 1	✗	Thing Plus FE310
Kinetis	frdm-k64f	✗	–
nRF51	nRF51 DK	✓	–
nRF52	nRF52840 DK	✓	Feather nRF52840 Express
nRF9160	nRF9160 DK	✓	Thing Plus nRF9160
QN908x	QN9080-DK	✗	–
RP2040	ArduPico	✗	Feather RP2040
SAM3	Arduino Due	✓	–
SAMD21	Arduino Zero	✓	Feather M0 Express
SAMD5x	–	–	Feather M4 Express
SAML1x	–	–	Thing Plus SAMD51
STM32F4	Nucleo-F446RE	✗	Feather STM32F405
13 other STM32	Nucleo-64 or Nucleo-144	✗	–
<b>Total</b>	<b>25</b>	<b>6</b>	<b>15</b>

⇒ widest support: Arduino UNO Shields



# Testing Approach

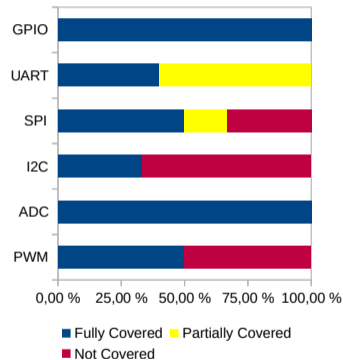


Unknown Author, Public Domain

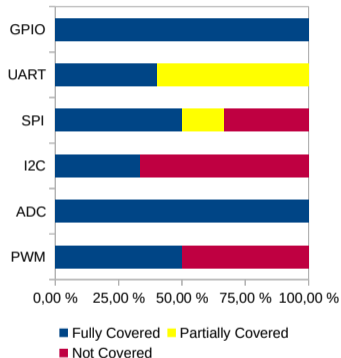
Peripheral	Tested By	Shield Required?
GPIO	loop-back (two pins connected)	✓
	I <sup>2</sup> C GPIO extender	✓
UART	loop-back (TXD to RXD)	✓
	timer to estimate symbol rate	X
SPI	loop-back (serial out to serial in)	✓
	$\overline{CS}$ connected to GPIO pin	✓
	SCK connected to GPIO extender pin	✓
	timer to estimate clock frequency	X
I <sup>2</sup> C	I <sup>2</sup> C GPIO extender	✓
	GPIO extender pin connected to GPIO pin	✓
ADC	connected to PWM DAC	✓
	connected to 4 bit R-2R DAC	✓
PWM	connected to ADC with low-pass filter	✓

# Test Coverage (1/3)

Peripheral	Mode / Aspect	Covered By Test?
GPIO	Floating Input	✓
	Push-Pull Output	✓
	Input with Pull-Up	✓
	Input with Pull-Down	✓
	Open-Drain	✓
	Open-Drain with Pull-Up	✓
	Interrupts	✓
UART	Data Integrity	✓
	Symbol Rate	(✓)
	Stop Bits	✓
	Parity Bit	(✓)
	Power Off Behavior	(✓)



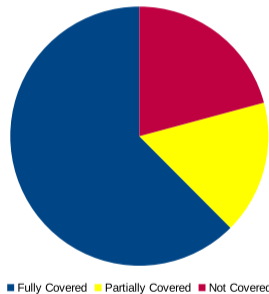
# Test Coverage (2/3)



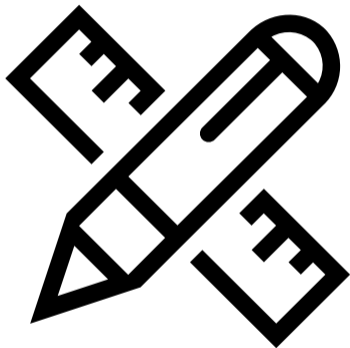
Peripheral	Mode / Aspect	Covered By Test?
SPI	Data Integrity	✓
	Bit Order	✗
	Clock Frequency	(✓)
	Clock Polarity	✓
	Clock Phase	✗
	$\overline{CS}$ Signaling	✓
I <sup>2</sup> C	Data Integrity	✓
	Clock Frequency	✗
	Clock Stretching	✗
ADC	Accuracy	✓
PWM	Duty Cycle	✓
	Pulse Width Modulation (PWM) Frequency	✗

# Test Coverage (3/3)

Peripheral	Covered	Partially Covered	Not Covered
GPIO	7	0	0
UART	2	3	0
SPI	3	1	2
I <sup>2</sup> C	1	0	2
ADC	1	0	0
PWM	1	0	1
<b>Total</b>	<b>15</b>	<b>4</b>	<b>5</b>



# Testing App Design



Tami Nova from Noun Project

- single test application
  - ⇒ fast test cycle
  - ⇒ detects resource conflicts between peripheral drivers
  - ⇒ downside: all periph drivers & test app need to fit ROM
- save ROM by
  - short, concise messages
  - reuse messages (e.g. starting test for %s)
  - print line numbers instead of descriptions on failed test
  - ⇒ much smaller than a description
  - ⇒ comments in the source code can still provide context
- soft dependency (`FEATURES_OPTIONAL`) on peripherals
  - test app useful even if subset of peripherals provided

# Demo Time



Magnus, CC BY-SA 2.0

# Demo: Critical Failure

---

Console output:

```
main(): This is RIOT! (Version: 2023.10-devel-262-g6193c-peripheral-selftest)
self-testing peripheral drivers
=====
CRITICAL FAILURE in tests/periph/selftest_shield/main.c:984
```

Failing source code:

```
981     /* the GPIO extender is used by the I2C test and the ADC test, so only
982     * initialize it once here */
983     if (IS_USED(MODULE_PCF857X)) {
984         ASSERT(pcf857x_init(&egpios, &params) == PCF857X_OK);
985     }
```

⇒ I<sup>2</sup>C GPIO extender failed to initialize ⇒ issue with I<sup>2</sup>C driver or pin mapping

# Demo: Non-Critical Failure

---

Console output:

```
make BOARD=arduino-due flash test -C tests/periph/selftest_shield
main(): This is RIOT! (Version: 2023.10-devel-262-g6193c-peripheral-selftest)
[...]
self-testing peripheral drivers
=====
Starting test for GPIO at tests/periph/selftest_shield/main.c:283
[OK]
[...]
Starting test for GPIO-IRQ at tests/periph/selftest_shield/main.c:438
FAILURE in tests/periph/selftest_shield/main.c:478
FAILURE in tests/periph/selftest_shield/main.c:482
FAILURE in tests/periph/selftest_shield/main.c:487
FAILURE in tests/periph/selftest_shield/main.c:492
[FAILED]
[...]

SOME TESTS FAILED
```



# Demo: Pass

---

```
make BOARD=nrf52840dk flash test -C tests/periph/selftest_shield
make: Entering directory '/home/maribu/Repos/software/RIOT/peripheral-selftest/
tests/periph/selftest_shield'
Building application "tests_selftest_shield" for "nrf52840dk" with MCU "nrf52".
[...]
START
main(): This is RIOT! (Version: 2023.10-devel-262-g6193c-peripheral-selftest)
self-testing peripheral drivers
=====
Starting test for GPIO at tests/periph/selftest_shield/main.c:283
[OK]
Starting test for GPIO at tests/periph/selftest_shield/main.c:305
[OK]
[...]
Starting test for SPI at tests/periph/selftest_shield/main.c:763
[OK]

ALL TESTS SUCCEEDED
```

# Summary

- cheap PCB
    - $\approx$  \$ 3 @ 50 pieces
    - < \$ 10 @ 1 piece
  - fully automated testing convering 19 / 24 aspects
    - super fast test cycle
    - already found real world bugs
  - easy and quick setup
- ⇒ testing no longer a pain
- ⇒ hopefully lots of bugs get found and fixed!



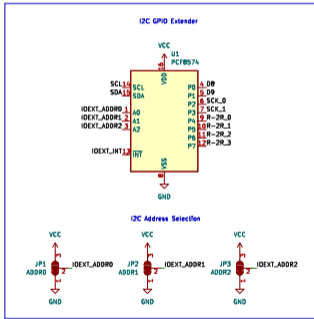
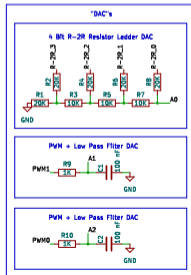
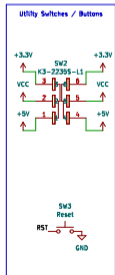
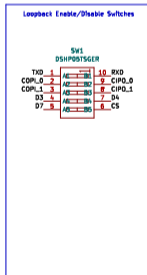
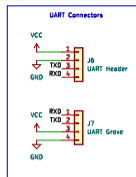
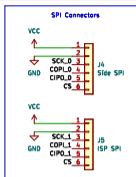
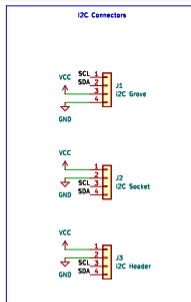
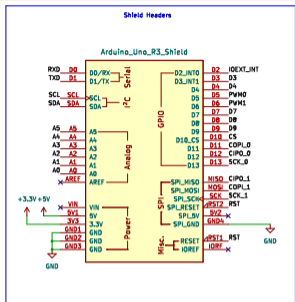
Eko Purnomo from Noun Project, CC-BY-3.0

# Handing Out Free PCBs



ideogram.ai

- handing out:
  - 40 pcs of the Peripheral Selftesting Shields
  - SMD-Variant, “v0.3”, optional parts not populated
- eligible:
  - active contributors to RIOT
  - one PCB per person





# SR-IOT

## Peripheral Selftest Shield v0.3

