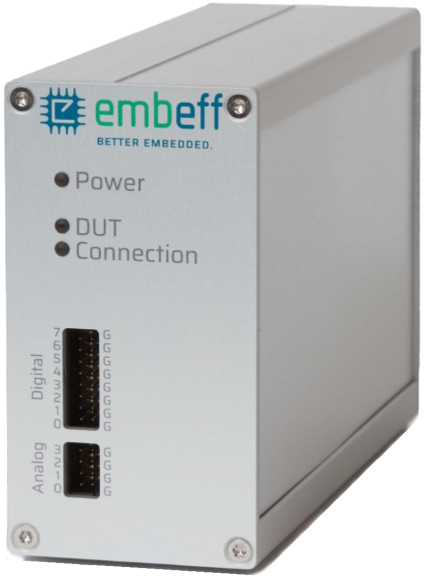
embeff

BETTER EMBEDDED.

ExecutionPlatform



- ✓ Agile Processor-in the-Loop Testing for Microcontroller.
- ✓ Unit-Tests
- ✓ HW/SW Integration-Tests
- ✓ System-Tests

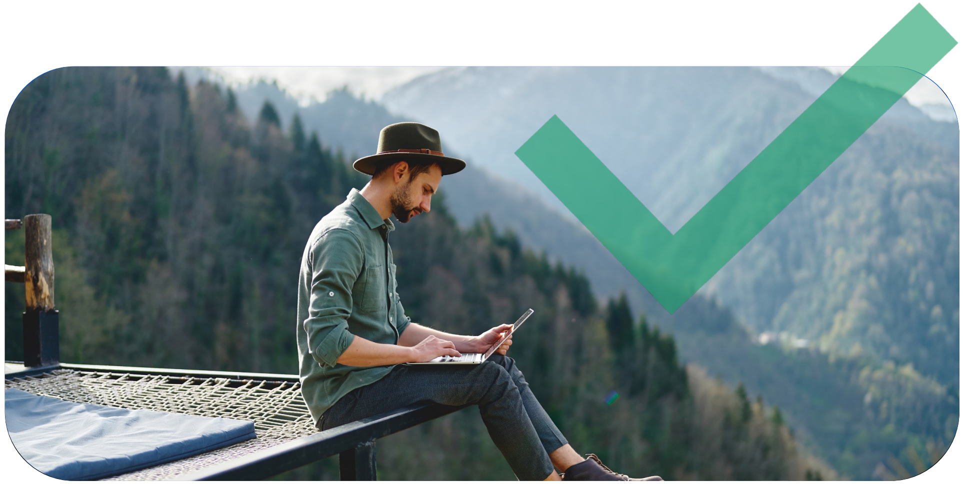
Impact for Dev-Teams

Agile testing for stable firmware

- ✓ Early tests for your firmware.
- ✓ Unit, Integration and System tests with a single device.
- ✓ Test error scenarios and ensure coverage.

Productive work – from anywhere

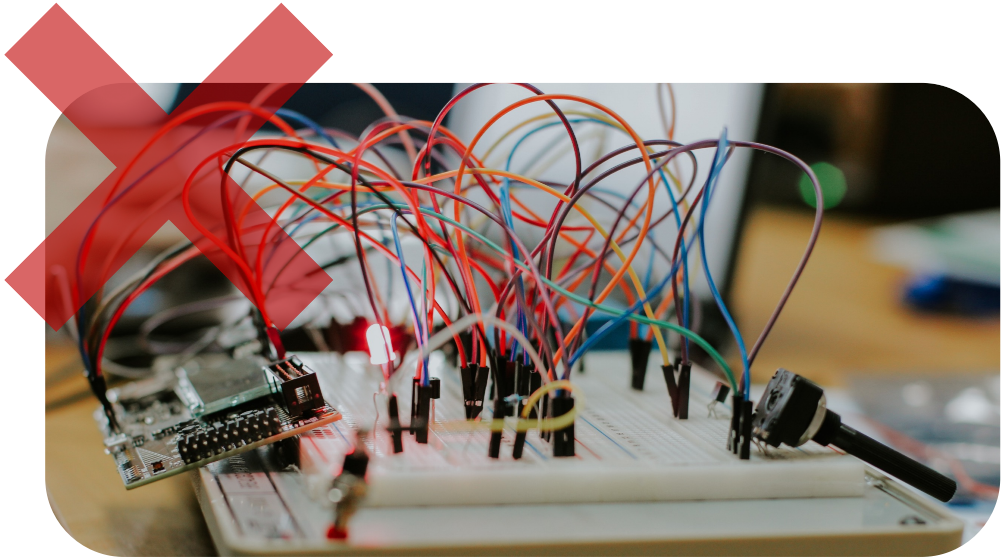
- ✓ Use it from anywhere. You only need a network connection
- ✓ All pins are visible through a Logic Analyzer. No cabling.



Impact for Companies

Focus on your core business

- ✓ No more self-made solutions that are impossible to maintain.
- ✓ embeff trainings and support for fast onboarding.



Scales to your budget

- ✓ Use the same test system for all MCUs.
- ✓ Low invest. Flexible monthly licensing.

embeff
BETTER EMBEDDED.

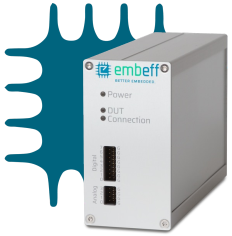
embeff ExecutionPlatform

1 Your Chip as DUT

Test system provides up to

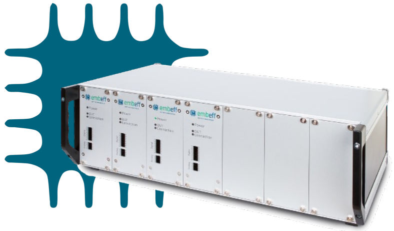
- ✓ 144 Digital I/O
- ✓ 24 Analog Outputs
- ✓ 8 Analog Inputs

2 Choose hardware for your needs



Standalone

- ✓ 1 MCU
- ✓ Use it on your desk.



Rack

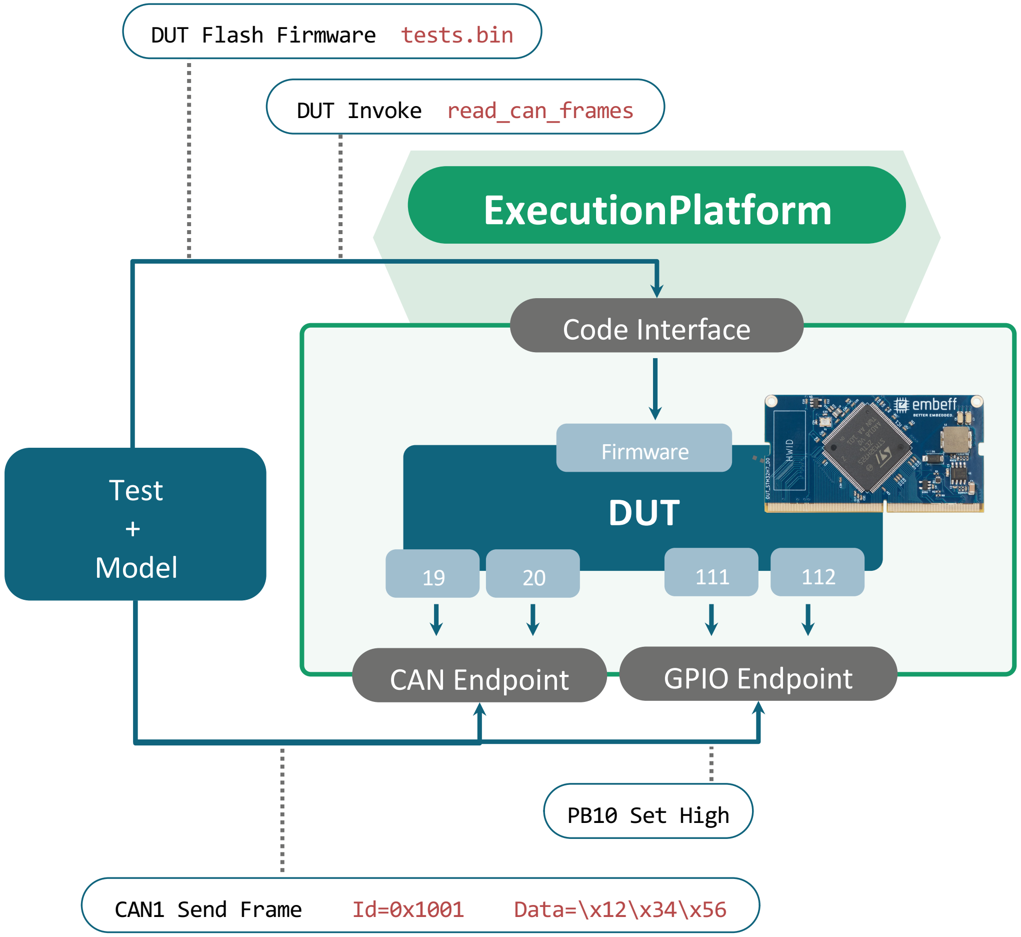
- ✓ Different MCUs
- ✓ Multiple parallel users.
- ✓ Mountable in your 19” rack.

Same structure for all tests

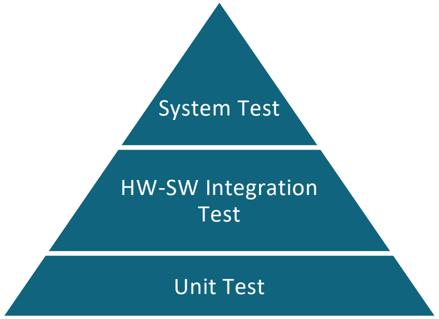
Call functions on the microcontroller via the **Code Interface**.

Endpoints are the interface between a test and a specific peripheral. Each endpoint provides specific functionality.

Use a **Model** as a digital twin to replace your environment.



```
DUT Flash Firmware   tests.bin
DUT Invoke   read_can_frames
PB10 Set High
CAN1 Send Frame    Id=0x1001    Data=\x12\x34\x56
```

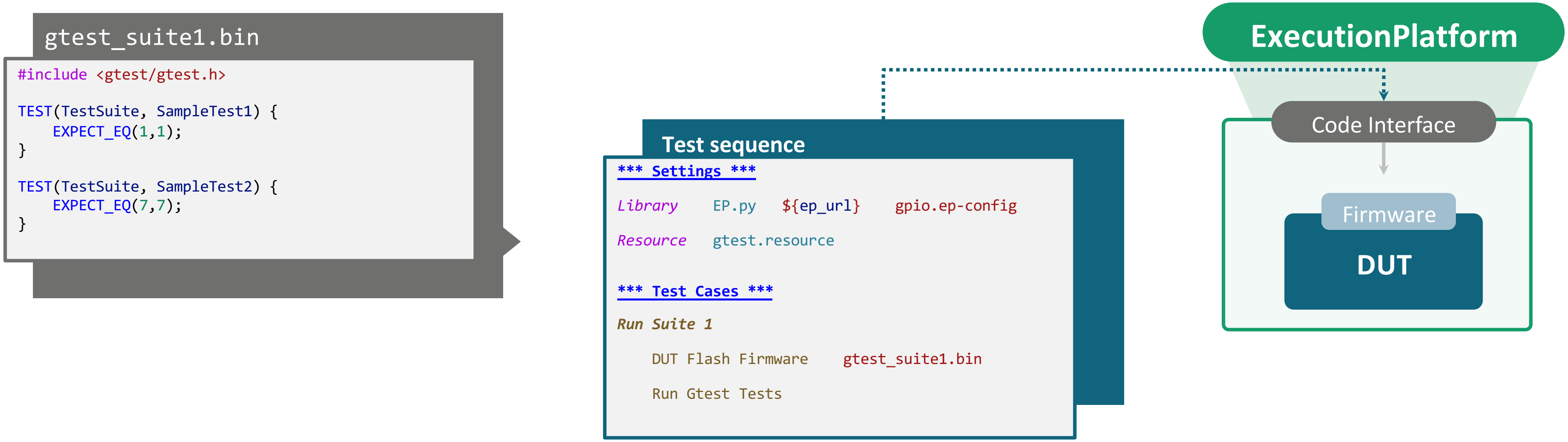


Endpoint configuration can be changed for each test.

```
GPIO:
  PB10:
    Pin: 111
    Pull: None
  PB11:
    Pin: 112
    Pull: None

CAN:
  CAN1:
    TX: 20
    RX: 19
```

embeff
BETTER EMBEDDED.

Unit-Tests



gtest_suite1.bin

```
#include <gtest/gtest.h>

TEST(TestSuite, SampleTest1) {
    EXPECT_EQ(1,1);
}

TEST(TestSuite, SampleTest2) {
    EXPECT_EQ(7,7);
}
```

Test sequence

```
*** Settings ***
Library     EP.py    ${ep_url}    gpio.ep-config
Resource    gtest.resource

*** Test Cases ***
Run Suite 1
    DUT Flash Firmware    gtest_suite1.bin
    Run Gtest Tests
```

- ✓ Easy On-Target unit test execution.
- ✓ Support for GoogleTest, Unity, Catch2.
- ✓ Generate Coverage Reports.

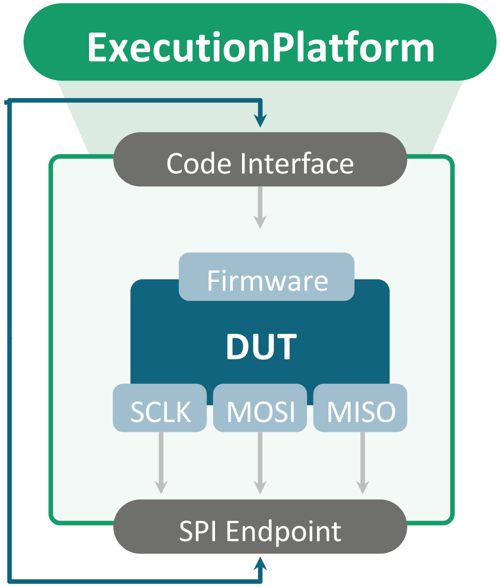
Integration-Tests

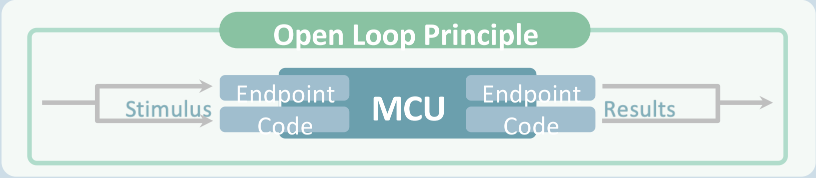
example_spi_crc.bin

```c
#include <ep/core.h>
#include <hardware/spi.h>

// External SPI sensor returns the temperature as a 16-bit value mCelcius.
// A third byte is appended that contains a checksum (CRC).
//
int32_t read_temperature_mC() {
    uint8_t rxBuf[3]{};
    spi_read_blocking(spi0, 0x00, rxBuf, 3);

    uint16_t temp_mC = (rxBuf[1] << 0) | (rxBuf[0] << 8);
    uint8_t crc_expected = crc8(rxBuf, sizeof(uint16_t));
    uint8_t crc_seen = rxBuf[2];

    if (crc_seen != crc_expected) { return -1; }
    return temp_mC;
}

int32_t ep_read_temperature_mC(uint8_t const *, int) {
    // Wrap generic EP invoke interface to specific function.
    return read_temperature_mC();
}

int ep_app_main() {
    spi_init(spi0, 100e3);
    ep_register("read_temperature_mC", &ep_read_temperature_mC);
    return ep_process_loop();
}
```

Test sequence

```
*** Settings ***
Library           EP.py    ${ep_url}      spi.ep-config
Suite Setup       DUT Flash Firmware      example_spi_crc.bin

*** Test Cases ***
Value 20.000 With Correct CRC
    SPI1 Set Response Bytes     \x4E\x20\x6D
    ${temperature_mC} =     Dut Invoke     read_temperature_mC
    Should Be Equal As Integers     ${temperature_mC}     20000


Value 20.000 With CORRUPT CRC
    SPI1 Set Response Bytes     \x4E\x20\xEE
    ${temperature_mC} =     Dut Invoke     read_temperature_mC
    Should Be Equal As Integers     ${temperature_mC}     -1
```





embeff
BETTER EMBEDDED.

Model your environment

- ✓ **Use Callbacks**
- ✓ **Connect Endpoints**
- ✓ **Hard real time**

- Robot sequence loads model.
- Model runs in parallel on ExecutionPlatform.

- Use all endpoint functionality.
- React to endpoint events with callbacks.

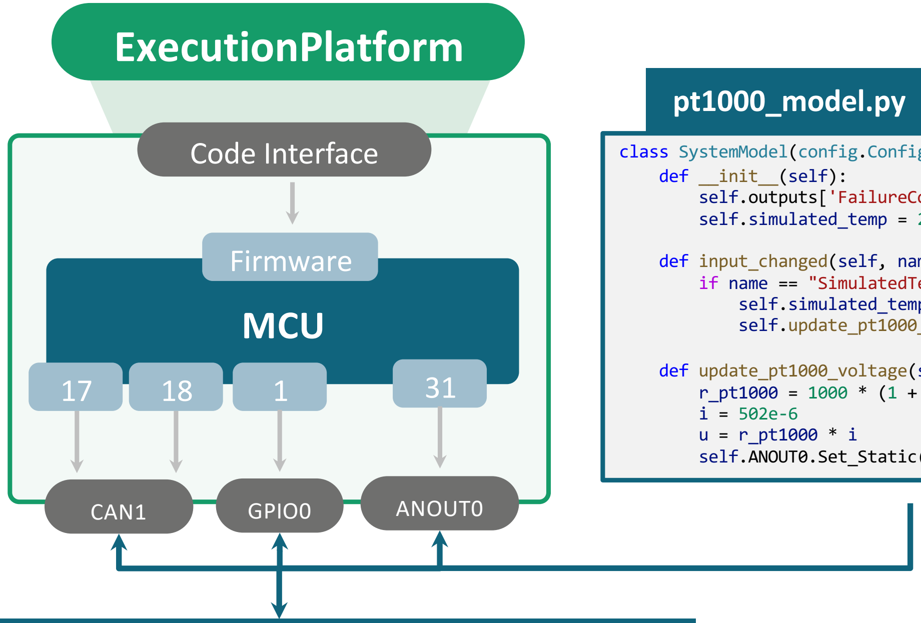
```
# example_model.py
class SystemModel():
    def gpio0_on_rising(self):
        self.anout0.set_static(3.0)

    def gpio0_on_falling(self):
        self.anout0.set_static(1.5)

    def periodic_100ms(self):
        byte0 = 0x11 if gpio1.read() else 0x22
        self.can1.send_frame(id=0x200, data=[byte0])

    def start(self):
        self.gpio0.callback_on_rising(self.gpio0_on_rising)
        self.gpio0.callback_on_falling(self.gpio0_on_falling)
        self.timer.add_periodic(0.1, self.periodic_100ms)
```

embeff
BETTER EMBEDDED.

System-Tests

firmware.bin

```c
int main() {
    hal_gpio_init();
    hal_can_setup();
    hal_adc_init();

    for (;;) {
        double r_pt1000 = measure_pin31();
        uint8_t temp_C = calculate_temp_from_r(r_pt1000);
        can_msg msg { 0x100, 1};
        msg.data[0] = temp_C;
        bool send_ok = hal_can_send(msg);

        if (send_ok) {
            hal_gpio_set(Pin1, false);
        } else {
            hal_gpio_set(Pin1, true);
        }
        sleep_ms(900);
    }
}
```



pt1000_model.py

```python
class SystemModel(config.Configuration):
    def __init__(self):
        self.outputs['FailureCount'] = 0
        self.simulated_temp = 20.0

    def input_changed(self, name, value):
        if name == "SimulatedTemp_C":
            self.simulated_temp = value
            self.update_pt1000_voltage()

    def update_pt1000_voltage(self):
        r_pt1000 = 1000 * (1 + 3.9083E-03 * self.simulated_temp)
        i = 502e-6
        u = r_pt1000 * i
        self.ANOUT0.Set_Static(f"{u}V")
```

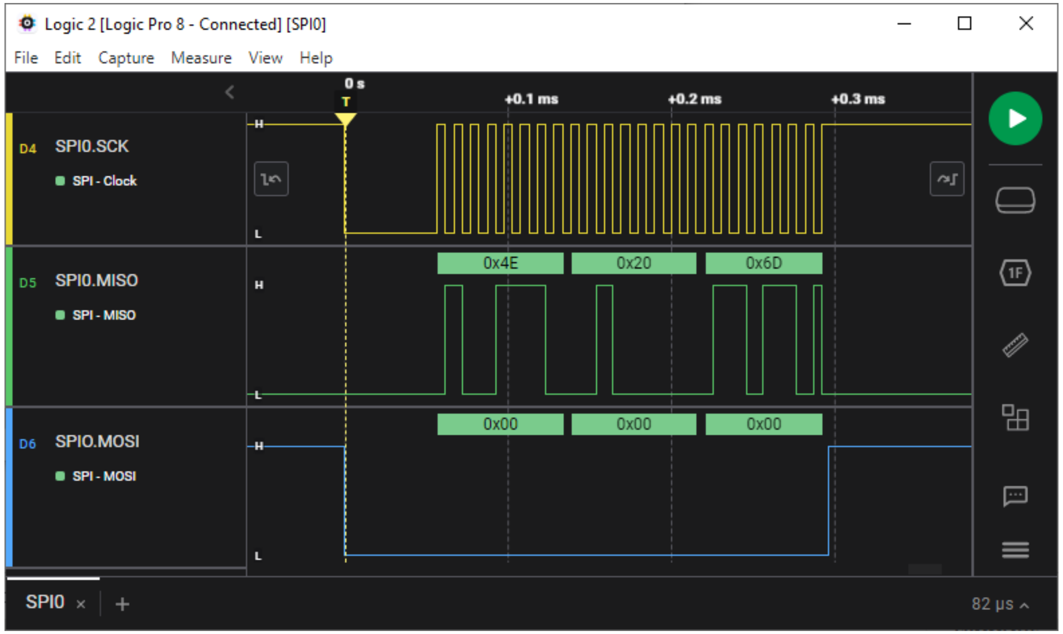
- The firmware reads a PT1000 analog voltage and sends the temperature value every second in a CAN frame.
- The model simulates the temperature voltage based on the test input.
- The test checks whether the firmware sends a simulated temperature value correctly via CAN.

Test sequence

```
*** Settings ***
Library        ${ep_app_dir}/EP.py    ${ep_url}    redalert.ep-config
Suite Setup    DUT Flash Firmware     firmware.bin

*** Test Cases ***
Simulate Change to 50° and Check Value In CAN Frame
    System Start Model     pt1000_model.py
    System Set Input     SimulatedTemp_C     ${50.0}
    Sleep    1.0s
    ${frames} =     CAN1 Get Frames
    ${temp} =     Get Temperature in Celsius From Frame     ${frames[1]}
    Should Be True     45 <= ${temp} <= 55
    Check That Failure LED on GPIO0 was never enabled
```

embeff
BETTER EMBEDDED.

Understand, what is happening on the MCU



Logic 2 [Logic Pro 8 - Connected] [SPI0]

View any of your DUT pins by connecting an optional Logic Analyzer. You choose which signals should be shown at runtime. No re-wiring necessary.

embeff
BETTER EMBEDDED.

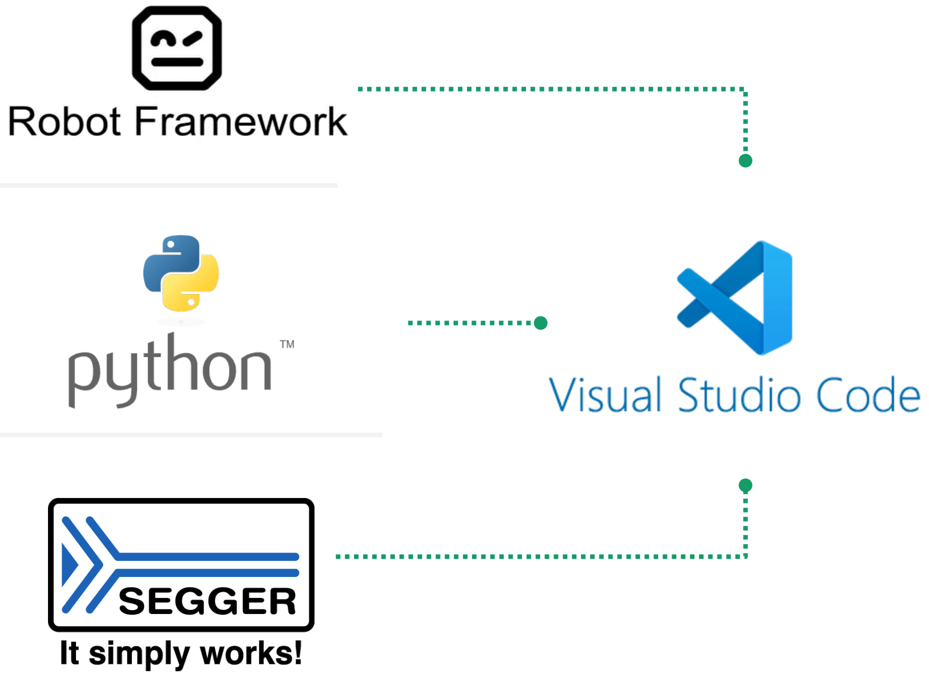
Focus on usability.

We use state-of-the-art technology.

Write and execute tests.
Reporting.

Model your environment as a digital twin.

Built-In Flasher.
Debugging for Cortex-M chips.



- ✓ Ready-to-use integration.
- ✓ Examples.
- ✓ Code-Completion and great usability.

How to get your ExecutionPlatform.

- ✓ Tell us which chip you need.
- ✓ Choose package

- ✓ We provide quote.
- ✓ You order.

- ✓ You decide which pins to use.
- ✓ Review.

- ✓ Production.
- ✓ Shipment.

- ✓ Connect to your network.
- ✓ Use ready-to-run examples.

Packages

Unit	Integration	System
✓ Code Interface ✓ GPIO+Analyzer Endpoint ✓ Software Updates & Hardware-Upgrades ✓ Top E-Mail Support	✓ All from "Unit". ✓ All Endpoints	✓ All from "Integration" ✓ Models for digital twins.

Email us sales@embeff.com

Call us +49-451-16088690

Start demo

embeff
BETTER EMBEDDED.