



Die ExecutionPlatform

Entwicklungsbegleitendes
Testen neu gedacht.

Sicherheitskritische
Software für
Mikrocontroller ist
anspruchsvoll.

Wir unterstützen Entwickler und
Tester mit einem innovativen
Testgerät bei dieser
Herausforderung.

Mit der ExecutionPlatform
können Sie
entwicklungsbegleitend
Ihre Firmware testen.



Unittest



HW/SW Integrationstest



Systemtest



- Generisches Testsystem für alle Mikrocontroller.
- Echtzeitfähige Simulation der Umgebung.
- 100% automatisiert ohne Verkabelung.

Ihr Mikrocontroller als Testobjekt

Die ExecutionPlatform ist ein generisches Testsystem, an das Ihr Mikrocontroller als Einsteckplatine angeschlossen wird. Dadurch werden Pins und Programmierinterface für das Testsystem zugänglich. Sie können die Testplatine jederzeit selbst austauschen.

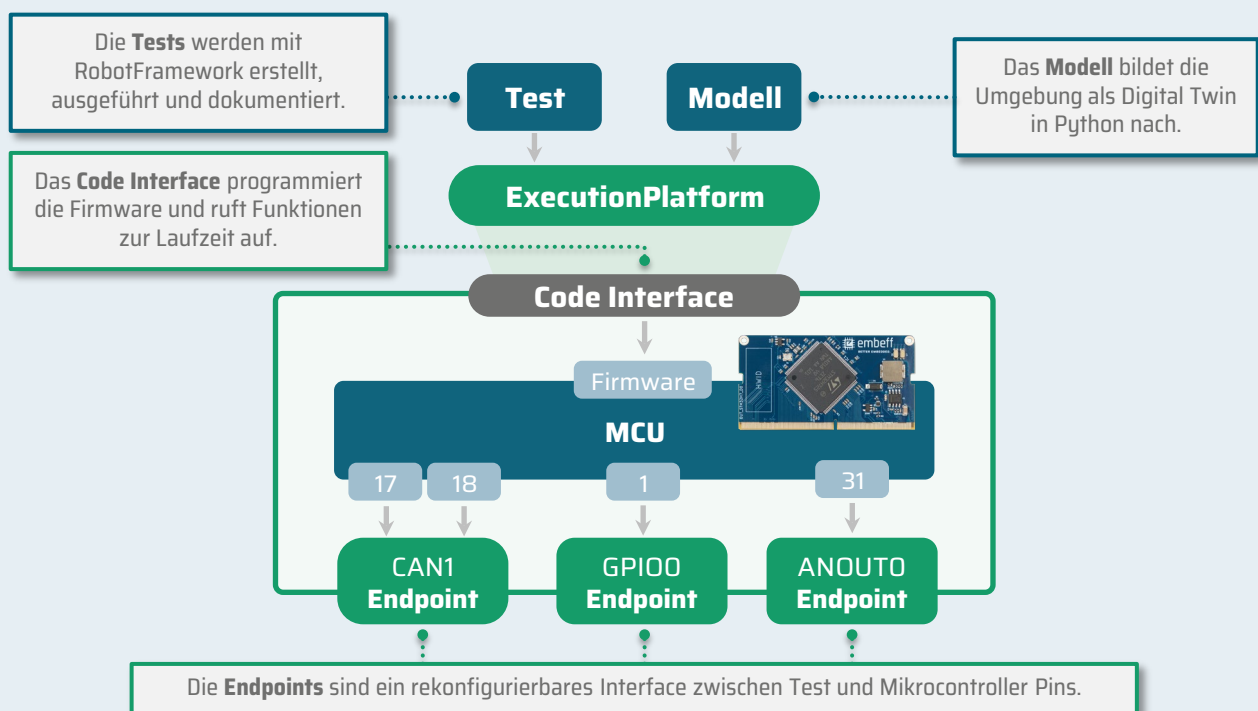
Auf Grundlage Ihrer Spezifikationen erstellen wir den Schaltplan und übernehmen Design sowie Produktion nach Ihrer Freigabe. Alternativ können Sie unsere Vorlage nutzen und die Platine selbst fertigen.



Ihre Testplatine nutzt

- bis zu 144 digitale I/O
- bis zu 24 analoge Ausgänge

Eine einheitliche Struktur für alle Tests



- **Test hat Zugriff auf alle Pins.**
- **Visualisierung beliebiger Pins mit optionalem Logic Analyzer.**
- **Eingebauter SEGGER Flasher zum Schreiben der Firmware.**

Unittest

Unittests prüfen einzelne Funktionen, Klassen oder Module.

Mit dem **Code Interface** flashen und starten Sie Ihre Unit Tests leicht auf dem Mikrocontroller. Die Ergebnisse werden komfortabel auf dem PC aufbereitet und ausgewertet.

Über eine eingebaute Benchmark-Funktion stellen Sie dabei sicher, dass das Zeitverhalten Ihren Anforderungen entspricht.

- Fertige Unterstützung für GoogleTest, Unity, Catch2.
- Keine Ausführung auf PC mit anderem Compiler nötig.
- Zeitverhalten für kritische Pfade dauerhaft prüfen.

gtest_suite1.bin

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

TEST(SafeSensor, calculate_temp_0C) {
    ASSERT_EQ(SafeSensor::calc_from_r(1000), 0);
}
```

Testsequenz

*** Test Cases ***

Run Suite 1

DUT Flash Firmware gtest_suite1.bin

Run Gtest Tests

Hardware/Software Integrationstest

test_can.bin

```
#include <ep/core.h>
#include "hal_can.hpp"

int32_t send_frame(uint8_t const* payload, int len) {
    can_msg msg;
    msg.id = 0x100;
    msg.dlc = len;
    for (int i=0; i<len; ++i) { msg.data[i] = payload[i]; }
    return hal_can_send(msg) ? 0 : -1;
}
```

Testsequenz

*** Test Cases ***

Send Frame Returns Error When Twice Not Acked
CAN1 Start Bitrate=500kbit/s
CAN1 Cause Bus Error On Next Received Frame
... Count=2 BusError=NoAck
\${res} = DUT Invoke send_frame abc
Should Be Equal As Integers \${res} -1

- Reproduzierbarkeit durch Automation garantiert.
- Fehlerzustände leicht prüfen.
- Coverage für Treiber/HAL nachweisen.

HW/SW Integrationstests prüfen das Zusammenspiel von hardwareabhängigem Code (Treiber/HAL) und der physischen Pin-Ebene des Mikrocontrollers.

Mit **Endpoints** starten Sie Vorgänge auf den Pins oder werten durch den Mikrocontroller gestartete Vorgänge aus. Das kann beispielsweise das Setzen/Lesen eines GPIO Pins oder das Senden/Empfangen eines CAN-Frames sein.

Dabei verwenden Tests das **Code Interface** um eine Funktion auf dem Mikrocontroller auszuführen und damit das zu testende Verhalten an den Pins auszulösen. Mit dem passenden Endpoint wird dieses Verhalten anschließend automatisch geprüft.

Umgekehrt kann über das Code Interface auch das korrekte Reagieren des Mikrocontrollers auf durch einen Endpoint erzeugtes Pin-Verhalten geprüft werden.

Für jede Peripherie gibt es eigene Endpoints mit umfangreichen Funktionen. Der Funktionsumfang ist dabei so ausgelegt, dass sämtliche Fehlerszenarien ausgelöst oder abgeprüft werden können.

Systemtest

Traditionelle Systemtests verwenden komplexe HiL-Systeme, die das Produkt an den Außengrenzen der Leiterplatte testen.

Die ExecutionPlatform bietet eine innovative und praxisnahe Alternative – und das bereits ohne fertige Leiterplatte in der frühen Entwicklungsphase.

Ein **Modell** simuliert die Umgebung des Mikrocontrollers an den Pins. Dabei können **Endpoints** einmalige oder regelmäßige Ereignisse an den Pins auslösen. Mit Callbacks reagiert das Modell auf Ereignisse, die vom Mikrocontroller initiiert werden.

pt1000_model.py

```
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")
```

- Entwicklungsbegleitende Systemtests.
- Digital Twin der Umgebung über Modell.
- Echtzeitfähigkeit garantiert.

Testsequenz

*** 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 Temp in Celsius From Frame \${frames[1]}

Should Be True 45 <= \${temp} <= 55

Check That Failure LED on GPIO0 was never enabled

| Ihre Vorteile



- Generisches System ersetzt aufwendige Eigenbauten.
- Flexible Skalierung bei minimalen Investitionskosten.
- Ausgezeichnete Dokumentation und schneller Support.

| Schnelles Onboarding

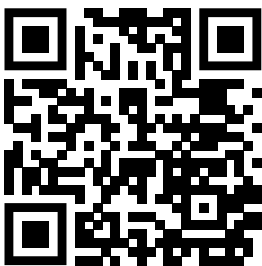
Unsere **Trainings** vermitteln praxisnahes Wissen über entwicklungsbegleitendes Testen in Mikrocontroller-Projekten. Dazu gehören Unit-, Integrations- und Systemtests und deren spezifische Bedeutung im Embedded-Kontext.

Alle vorgestellten Prinzipien und Techniken sind allgemeingültig und können als Ausgangspunkt für eigene Testkonzepte verwendet werden. Die Übungen finden an echter Hardware statt, unterstützt durch die ExecutionPlatform als Testgerät.

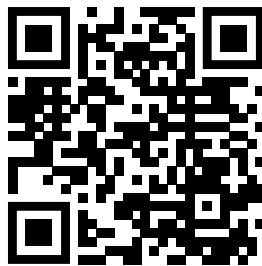
Nach dem Training besitzen Teilnehmer fundierte Kenntnisse über Testverfahren für Mikrocontroller-Projekte und können agile Tests eigenständig in ihre Projekte integrieren. Sie erlernen den Einsatz von gängigen Testframeworks und entwickeln ein Verständnis für effizientes Testvorgehen.

In unseren **Workshops** erarbeiten wir auf der Grundlage dieser Trainings gemeinsam Konzepte für den Test Ihrer Projekte.

| Mehr erfahren



Videos



Training

Sie wollen die ExecutionPlatform ausprobieren - direkt und ohne etwas zu installieren?

Dafür bietet unsere browser-basierte Demo auf embeff.com/ep-demo viele vorbereitete Beispiele.