embeff

BETTER EMBEDDED.

Use Case: Remote Work

Productive work for Arm Cortex-M Microcontroller. From anyhwere.

Challenge

How can you enable efficient remote work for embedded teams working with Arm Cortex-M microcontroller?

Microcontroller development traditionally requires direct access to various devices. In addition to the embedded board, this also includes at least the debugger and laboratory power supply. A typical workplace also requires an oscilloscope, logic analyzer and dongles for special interfaces (CAN, UART, etc.).

Duplicating this environment for home offices and other places is financially and logistically difficult. Furthermore, duplicates also remain tied to one location. **Many companies therefore try to set up the devices centrally and make them accessible remotely via VPN connections.**

Network-compatible debuggers are permanently connected to an embedded board and connected to the company network. This allows your developers to access the microcontroller and run software on it.

This approach is reasonable feasible for some applications. However, the **microcontroller pins remain inaccessible.** Developers are therefore still unable to carry out the majority of hardware-related tasks. Driver development, for example, requires direct pin access with devices such as logic analyzers and interface devices such as a CAN dongle.

Solution

The embeff ExecutionPlatform enables efficient remote work for embedded teams for the first time.

With this device, **all microcontroller pins are fully accessible from anywhere**.

arm

SEGGER

It simply works!

Use Case Features:

- Compact device with integrated Microcontroller and SEGGER J-Link.
- Work on pins and test HAL/driver without additional devices.
- Observe all pins without wiring.

Standardized setup instead of cable clutter guarantees reliable usage.

- You tell us the microcontroller you are using. Together we define the basic schematic and integrate your chip. This makes **your specific chip** and all its pins available for use in the ExecutionPlatform. Instead of different setups and wiring for microcontroller, you get the same devices and a appearance that is both clean and robust.
- You only need to connect the ExecutionPlatform to your network.
- Each device contains a **J-Link debugger** from SEGGER.
- In manual mode, developers can work directly on the microcontroller via remote debugging from anywhere - for example with SEGGER Ozone.
- In automatic mode, your developers write small test sequences. These sequences automatically program the microcontroller with the appropriate firmware and execute individual functions in the code. The integrated user administration supports working with different users on the same ExecutionPlatform.

Working with the microcontroller pins

Embedded systems communicate with their environment via signals and interfaces. Your developers write hardware-related code for this, such as drivers or a hardware abstraction layer (HAL). This code is almost always tied to a specific microcontroller and cannot be executed or simulated in any other way.

To test such software parts, developers have to run the code on the microcontroller and test the resulting behavior on the pins.

This traditionally requires either passive measuring (oscilloscope, logic analyzer) or active receiver/sender (serial interface, CAN dongle, a specific SPI sensor) to be connected to the pins. **This dependance on many external devices limits the possibilities of meaningful remote work.**

The **ExecutionPlatform revolutionizes such scenarios:** It fully integrates measuring devices and active receiver/sender on a digital level. They can be configured and used flexibly. You do not have to connect any external devices.

Our **Endpoints** are created dynamically and can both listen and operate on the physical microcontroller pins.

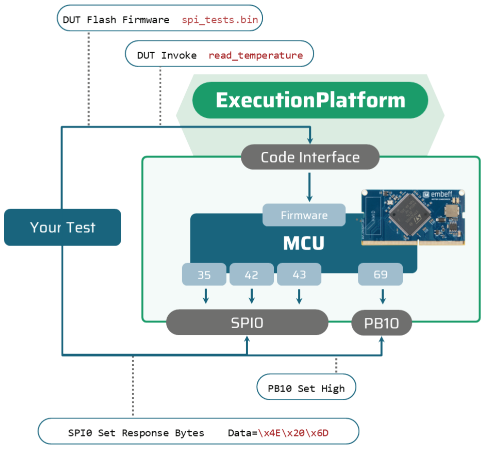
Example

Your developer has to create code to read a SPI-connected temperature sensor. Instead of going to the physical board and connecting a specific sensor, it is sufficient and much quicker to configure a SPI-Slave Endpoint on the desired pins.

Using the test sequence, your developer now specifies the data with which the Endpoint should respond on MISO to requests from the microcontroller (SCK). All write accesses to MOSI are automatically recorded and are also available for evaluation in the test.

This **approach completely eliminates the need to manually wire and connect external devices** to the pins. Your developers can work on hardware-related code from anywhere.

Example Endpoint configuration for SPI and a single GPIO. All pins are shown on the external logic analyzer.



DUT Flash Firmware spi_tests.bin
DUT Invoke read_temperature
ExecutionPlatform
Code Interface
Your Test
Firmware
MCU
35
42
43
69
SPIO
PB10
PB10 Set High
SPI0 Set Response Bytes Data=\x4E\x20\x6D

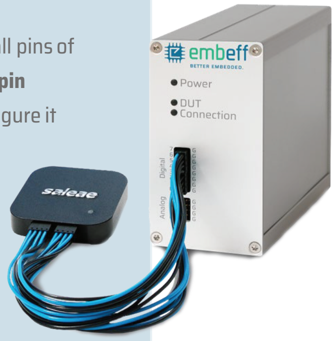
```
SPI:
  SPI0:
    SCK: 35
    MISO: 42
    MOSI: 43

GPIO:
  PB10:
    Pin: 69
    Pull: None

Analyzer:
  Digital:
    0: PB10.Pin
    4: SPI0.SCK
    5: SPI0.MISO
    6: SPI0.MOSI
```

Monitor all pins without wiring

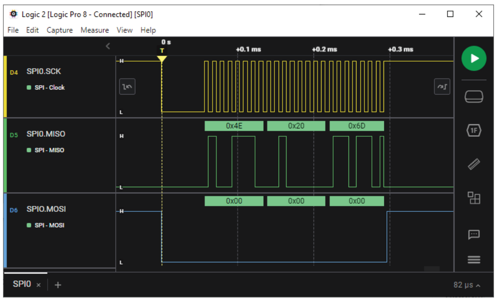
You may connect an optional logic analyzer to the ExecutionPlatform.

The unique feature: The device has physical access to all pins of your microcontroller, so it can **display up to 8 arbitrary pin signals on the logic analyzer**. All you need to do is configure it accordingly at runtime. The inconvenience of reconnecting different cables is completely eliminated. This allows your developers to conveniently keep an eye on pin behavior – powerful for identifying complicated problems – from anywhere.

ExecutionPlatform connected to Saleae Logic Pro 8.



Captured SPI signals for a single test.



Result

With the network-based ExecutionPlatform, your embedded teams can work with microcontroller from anywhere. And more efficiently than at the developer's desk in the company. As **all operations and tests on the pins are fully automated,** the manual and time-consuming steps involved in traditional development and testing are eliminated. Developers do not have to carry out any wiring. There is no need to connect logic analyzers, protocol dongles and any other external devices.

Instead, your developers write easy-to-read test sequences and execute them within seconds.

Distributed teams can work together on complex problems and develop innovative solutions.

Try it from your browser

Do you want to see for yourself how efficient remote work looks like for a **STM32 microcontroller**?

Then get your personal test environment with a single click on embeff.com/ep-demo

This environment consists of a browser-based version of Visual Studio Code which is connected to one of our demo boxes. It is ready-to-use with test examples for HAL code. You can observe the pin behavior with an integrated logic analyzer.

embeff GmbH
Willy-Brandt-Allee 31 b · D-23554 Lübeck · Germany
Tel. +49 451 16088690 · Fax +49 451 16088699
info@embeff.com · www.embeff.com

embeff
BETTER EMBEDDED.