

ioRaptor Webinars

**Agile Development with
Arm Mbed OS**

andreas.schmidt@thingforward.io

@aschmidt75

@thingforward



About us

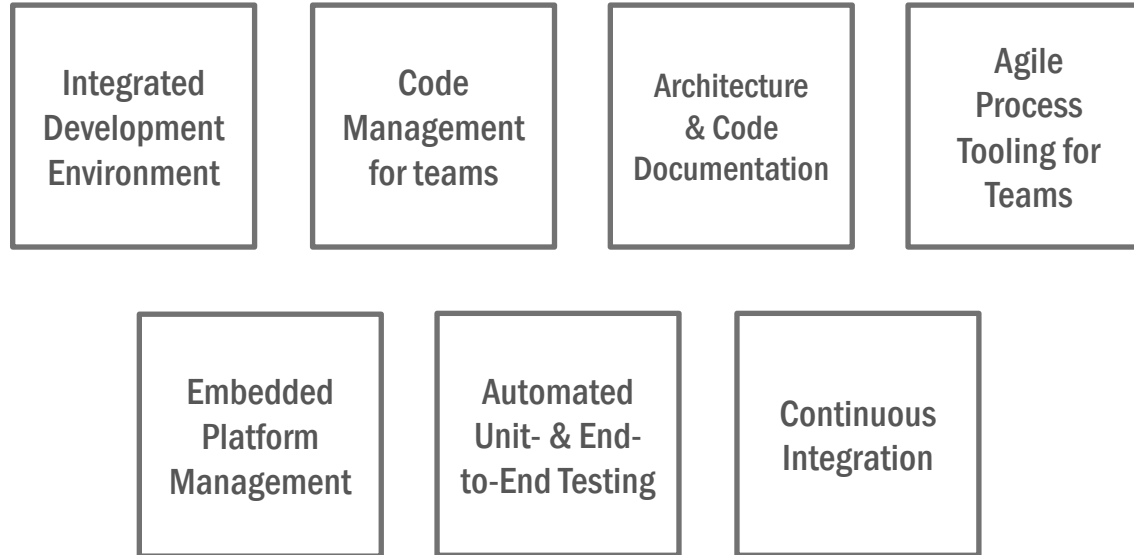


- www.thingforward.io
- Part of germany-based Cassini Group
- Topics:
 - IoT Architecture & Implementation projects
 - IoT Prototyping & Developer training workshops
 - Products for agile IoT software toolchains:
 - <https://www.thingforward.io/ioraptor/>
 - <https://slyft.io>
 - <https://thngstruction.online>

This webinar's objectives

- **Introduce to Agile Development Workflows for IoT**
- **Developing for ARM Mbed OS: options for an agile dev stack**
- **not an intro to ARM Mbed OS, but examples will look familiar to e.g. Arduino Code**

Parts of an Agile Development Workflow



Agenda

- Short intro to ARM Mbed OS and PlatformIO
- Mbed Cloud IDE
- Locally installed toolchain (mbed-cli)
- PlatformIO for Mbed & Cloud Services
- Comparison
- Q&A



Arm Mbed OS developer site

Mbed OS 5

Mbed simplifies and speeds up the creation and deployment of IoT devices based on Arm microcontrollers.

The project is being developed by Arm, its Partners and the contributions of the global Arm Mbed Developer Community.

[Get started »](#)

[Latest release »](#)



PlatformIO is an open source ecosystem for IoT development

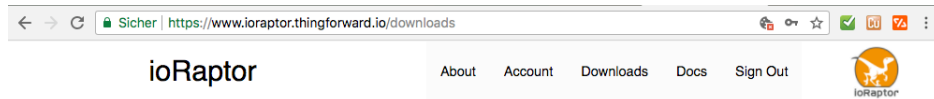
Cross-platform IDE and unified debugger. Remote unit testing and firmware updates

24 Development Platforms · **15** Frameworks · **479** Embedded Boards · **61** Project Examples · **5,212** Libraries · **21,159** Library Examples

 [Get Started](#)

[GitHub](#) · [Bintray](#) · [Twitter](#) · [Facebook](#) · [Hackaday](#) · [Forums](#)

[Release Notes](#) · [Documentation](#) · Sponsored with ❤ by [PlatformIO Plus](#)






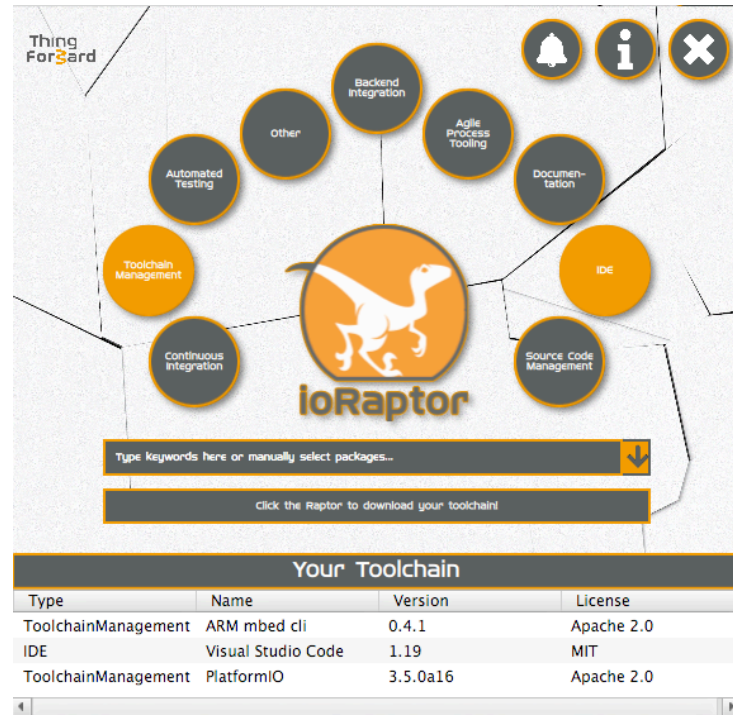
Downloads

On this page you have access to the most recent release of ioRaptor.

Current Release

The current release is 0.2.3. You can download installer files from its original location at [GitHub](#), or use the links below.

-  [ioraptor-installer-win.exe](#) installer-binary as windows executable
-  [ioraptor-installer-macos.dmg](#) installer-binary as MacOS Image
-  [ioraptor-installer-lnx.bin](#) installer-binary as executable Linux ELF 64bit



| Your Toolchain | | | |
|---------------------|--------------------|----------|------------|
| Type | Name | Version | License |
| ToolchainManagement | ARM mbed cli | 0.4.1 | Apache 2.0 |
| IDE | Visual Studio Code | 1.19 | MIT |
| ToolchainManagement | PlatformIO | 3.5.0a16 | Apache 2.0 |

The screenshot displays the Mbed Cloud IDE interface. The top bar shows the project name "Mbed" and the file path "/IoT01A_Base/main.cpp". The left sidebar, titled "Program Workspace", shows a tree view of the project files, including "My Programs", "DISCO_L475VG_IOT01-wifi_http_server", "IoT01A_Base", "easy-connect", "JSON", "mbed-http", "Classes", "http_parser", "http_parser.c", "http_parser.h", "LICENSE", "source", "LICENSE", "mbed_lib.json", "README.md", "http_router.cpp", "http_router.h", "http_wot.cpp", "http_wot.h", "main.cpp", "mbed_app.json", "wot_app.cpp", "wot_app.h", "mbed-os", "mbed-os-example-wifi", "mbed-os-tcp-server-example", and "MODSERIAL".

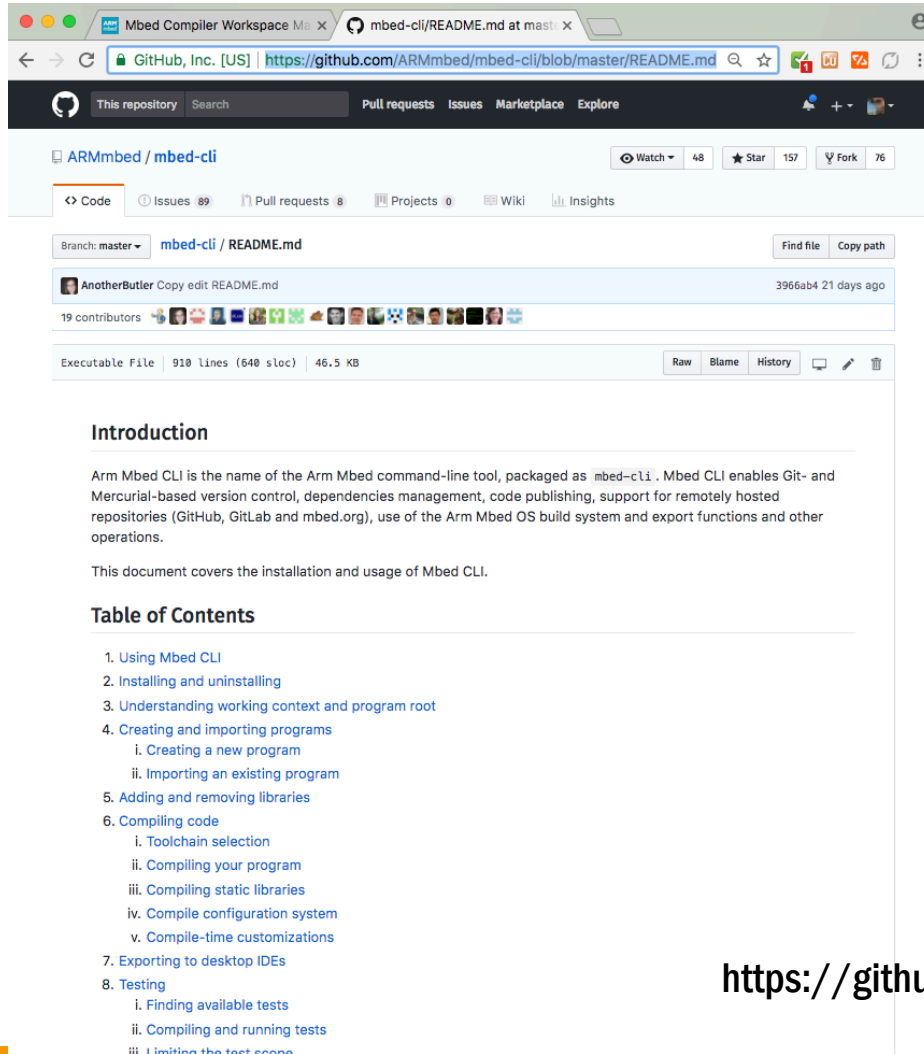
The main editor area shows the code for "main.cpp":

```

1 #include "mbed.h"
2 #include "easy-connect.h"
3 #include "TCPServer.h"
4 #include "http_router.h"
5 #include "http_wot.h"
6
7 Serial pc(SERIAL_TX, SERIAL_RX);
8
9 #define URIPATH_PREFIX "/"wot"
10 #define HTTP_PORT 80
11
12
13 int main() {
14     pc.baud(115200);
15
16     NetworkInterface* network = easy_connect(true);
17     if (!network) {
18         printf("Connecting to the network failed... See serial output.\r\n");
19         return 1;
20     }
21
22     TCPServer* server = new TCPServer();
23
24     nsapi_error_t ret;
25
26     ret = server->open(network);
27     if (ret != NSAPI_ERROR_OK) {
28         printf("error(1): %i\n", ret);
29         return ret;
30     }
31
32     ret = server->bind(network->get_ip_address(), HTTP_PORT);
    
```

Below the editor, the "Compile output for program: IoT01A_Base" section is visible, showing a table with columns: Description, Error Number, Resource, In Folder, and Location. The table is currently empty.

The bottom status bar shows "Ready." and "21.03.18".



The screenshot shows the GitHub repository page for `ARMmbed/mbed-cli`. The repository has 48 watches, 157 stars, and 76 forks. The `README.md` file is selected, showing it was last edited by `AnotherButler` 21 days ago. The file is an executable, 46.5 KB in size, and contains 910 lines of code. The `Introduction` section states that the Arm Mbed CLI is a command-line tool for managing dependencies, publishing code, and building projects. The `Table of Contents` lists the following sections:

1. Using Mbed CLI
2. Installing and uninstalling
3. Understanding working context and program root
4. Creating and importing programs
 - i. Creating a new program
 - ii. Importing an existing program
5. Adding and removing libraries
6. Compiling code
 - i. Toolchain selection
 - ii. Compiling your program
 - iii. Compiling static libraries
 - iv. Compile configuration system
 - v. Compile-time customizations
7. Exporting to desktop IDEs
8. Testing
 - i. Finding available tests
 - ii. Compiling and running tests
 - iii. Limiting the test scope

<https://github.com/ARMmbed/mbed-cli/blob/master/README.md>

Docker-based build environment

- Follow instructions on <https://github.com/thingforward/docker-mbed-cli-gcc-arm>
- `mkdir demo && cd demo`
- `mbed new --program demo`
- `mbed target DISCO_L475VG_IOT01A`
- `mbed toolchain gcc_arm`
- `mbed compile`

Building project mbed (DISCO_L475VG_IOT01A, GCC_ARM)

Scan: .

Scan: mbed

Scan: env

Elf2Bin: mbed

| Module | .text | .data | .bss |
|------------------|-------|-------|------|
| [fill] | 149 | 0 | 15 |
| [lib]/c.a | 37012 | 2240 | 56 |
| [lib]/gcc.a | 3152 | 0 | 0 |
| [lib]/misc | 296 | 16 | 28 |
| main.o | 180 | 4 | 136 |
| mbed-os/drivers | 1181 | 4 | 100 |
| mbed-os/hal | 1449 | 4 | 66 |
| mbed-os/platform | 2932 | 0 | 354 |
| mbed-os/rtos | 9102 | 168 | 5989 |
| mbed-os/targets | 11271 | 8 | 904 |
| Subtotals | 66724 | 2444 | 7648 |

Total Static RAM memory (data + bss): 10092 bytes

Total Flash memory (text + data): 69168 bytes

Image: ./BUILD/DISCO_L475VG_IOT01A/gcc_arm/mbed.bin

```
c018@CC-C02RN139FVH6-c018 ~/work/embedded/mbed/pio pio init -b disco_l475vg_iot01a --ide vscode
```

The current working directory /Users/cassini/work/embedded/mbed/pio will be used for project.
You can specify another project directory via
'platformio init -d %PATH_TO_THE_PROJECT_DIR%' command.

The next files/directories have been created in /Users/cassini/work/embedded/mbed/pio

platformio.ini - Project Configuration File

src - Put your source files here

lib - Put here project specific (private) libraries

Project has been successfully initialized!

Useful commands:

'platformio run' - process/build project from the current directory

'platformio run --target upload' or 'platformio run -t upload' - upload firmware to embedded board

'platformio run --target clean' - clean project (remove compiled files)

'platformio run --help' - additional information

```
c018@CC-C02RN139FVH6-c018 ~/work/embedded/mbed/pio ls -al
```

```
total 24
drwxr-xr-x  9 c018  staff   288 21 Mär 18:05 .
drwxr-xr-x  8 c018  staff   256 21 Mär 18:04 ..
-rw-r--r--  1 c018  staff    84 21 Mär 18:05 .gitignore
drwxr-xr-x  4 c018  staff   128 21 Mär 18:05 .pioenvs
-rw-r--r--  1 c018  staff  1553 21 Mär 18:05 .travis.yml
drwxr-xr-x  4 c018  staff   128 21 Mär 18:05 .vscode
drwxr-xr-x  3 c018  staff    96 21 Mär 18:05 lib
-rw-r--r--  1 c018  staff   454 21 Mär 18:05 platformio.ini
drwxr-xr-x  2 c018  staff    64 21 Mär 18:05 src
```

```
c018@CC-C02RN139FVH6-c018 ~/work/embedded/mbed/pio cat platformio.ini
```

```
; PlatformIO Project Configuration File
;
; Build options: build flags, source filter
; Upload options: custom upload port, speed and extra flags
; Library options: dependencies, extra library storages
; Advanced options: extra scripting
;
; Please visit documentation for the other options and examples
; http://docs.platformio.org/page/projectconf.html
```

```
[env:disco_l475vg_iot01a]
platform = ststm32
board = disco_l475vg_iot01a
framework = mbed
```

```

update          update installed development platforms
c018@CC-C02RN139FVH6-c018 ~/work/embedded/mbed/pio ➤ pio platforms frameworks mbed
mbed ~ mbed
=====
The mbed framework The mbed SDK has been designed to provide enough hardware abstraction to be
jects. It is built on the low-level ARM CMSIS APIs, allowing you to code down to the metal if r
kbook of hundreds of reusable peripheral and module libraries have been built on top of the SDK

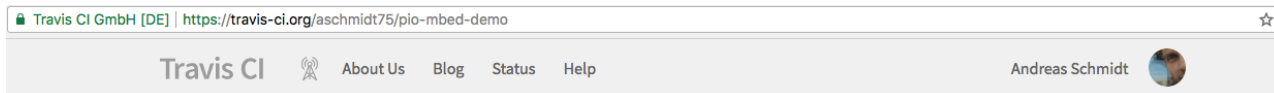
Home: https://platformio.org/frameworks/mbed

```

```
c018@CC-C02RN139FVH6-c018 ~/work/embedded/mbed/pio$ pio run
[Wed Mar 21 18:12:06 2018] Processing disco_l475vg_iot01a (platform: ststm32; board: disco_l475vg_iot01a; framework: mbed)
-----
Verbose mode can be enabled via '-v, --verbose' option
PLATFORM: ST STM32 > ST DISCO-L475VG-IOT01A
SYSTEM: STM32L475VGT6 80MHz 128KB RAM (1MB Flash)
DEBUG: CURRENT(stlink) ON-BOARD(stlink) EXTERNAL(blackmagic, jlink)
Library Dependency Finder -> http://bit.ly/configure-pio-ldf
LDF MODES: FINDER(chain) COMPATIBILITY(light)
Collected 14 compatible libraries
Scanning dependencies...
No dependencies
Compiling .pioenvs/disco_l475vg_iot01a/FrameworkMbedCore/drivers/AnalogIn.cpp.o
Compiling .pioenvs/disco_l475vg_iot01a/FrameworkMbedCore/drivers/BusIn.cpp.o
Compiling .pioenvs/disco_l475vg_iot01a/FrameworkMbedCore/drivers/BusInOut.cpp.o
Compiling .pioenvs/disco_l475vg_iot01a/FrameworkMbedCore/drivers/BusOut.cpp.o
```

```
/Users/cassini/.platformio/packages/framework-mbed/targets/TARGET_STM/TARGET_STM32L4/device/stm32l4xx_ll_spi.h
ter will break strict-aliasing rules [-Wstrict-aliasing]
*((__IO uint16_t *)&SPIx->DR) = TxData;
^
Compiling .pioenvs/disco_l475vg_iot01a/FrameworkMbedCore/targets/TARGET_STM/us_ticker_32b.c.o
Compiling .pioenvs/disco_l475vg_iot01a/src/main.cpp.o
Generating LD script .pioenvs/disco_l475vg_iot01a/STM32L475XX.ld.link_script.ld
Linking .pioenvs/disco_l475vg_iot01a/firmware.elf
Calculating size .pioenvs/disco_l475vg_iot01a/firmware.elf
Building .pioenvs/disco_l475vg_iot01a/firmware.bin
text      data      bss      dec      hex filename
40180      2508      1508      44196     aca4 .pioenvs/disco_l475vg_iot01a/firmware.elf
===== [SUCCESS] Took 46.32 seconds =====
```

Build via Travis-ci + PlatformIO



aschmidt75 / pio-mbed-demo build failing

Current Branches Build History Pull Requests

More options

✓ master fixed env

✓ #5 passed

Restart build

Commit f50ed21

Ran for 2 min 29 sec

Compare ab7731e..f50ed21

12 minutes ago

Branch master

Andreas Schmidt authored and committed

Job log

View config

```
1 Worker information
6 mode of '/usr/local/clang-5.0.0/bin' changed from 0777 (rwxrwxrwx) to 0775 (rwxrwxr-x)
7 Build system information
404
405 removed '/etc/apt/sources.list.d/basho_riak.list'
406 Network availability confirmed.
407 127.0.0.1 localhost
408 ::1 ip6-localhost ip6-loopback
409 fe80::0 ip6-localnet
410 ff00::0 ip6-mcastprefix
411 #500x1 ip6-allnodes
```


Comparison

| | Mbed Cloud IDE | Mbed-cli local toolchain | PlatformIO |
|--|----------------|--------------------------|------------|
| Ease of use (“Getting started”) | ++ | 0 | + |
| Required Experience Level (Dev+Ops) | low | high | medium |
| Control of build chain | - | ++ | + (SCONS) |
| Control of code repositories | 0 | ++ | ++ |
| CI/CD integration options | - | 0 | ++ |

Thank you!

andreas.schmidt@thingforward.io
@aschmidt75

www.thingforward.io
@thingforward

Andreas Schmidt, Digital Incubation & Growth GmbH

Alle Angaben basieren auf dem derzeitigen Kenntnisstand. Änderungen vorbehalten.

Dieses Dokument von Cassini ist ausschließlich für den Adressaten bzw. Auftraggeber bestimmt. Es bleibt bis zu einer ausdrücklichen Übertragung von Nutzungsrechten Eigentum von Cassini.

Jede Bearbeitung, Verwertung, Vervielfältigung und/oder gewerbsmäßige Verbreitung des Werkes ist nur mit Einverständnis von Cassini zulässig.

© 2018 Digital Incubation & Growth GmbH / Cassini Group. All rights reserved.

PlatformIO is © Ivan Kravets, PlatformIO.org

ARM Mbed © ARM Limited