

IoT Engineering

o: Syllabus

CC BY-SA, Thomas Amberg, FHNW
(unless noted otherwise)

Today

$\frac{1}{3}$ syllabus,

$\frac{2}{3}$ lesson 1.

Slides, code & hands-on: tmb.gr/iot-0



Hello

Thomas Amberg ([@tamberg](#)), Software Engineer.

New at FHNW, as a "Prof. of Internet of Things".

Founder of [Yaler](#), "secure remote access for IoT".

Organising an open [IoT Meetup](#) group in Zürich.

thomas.amberg@fhnw.ch

Beta alert

This is the first ever *IoT Engineering* class.

Content might still be incomplete.

Things will go wrong.

Found a bug? Let me know! Feedback is welcome. **n|w**

Language

Everything written by me will be in English.

You can write German or English.

In class we speak German.

Gar kein Englisch? Kontaktieren Sie mich.

Programming Language

On microcontroller devices we will use (Arduino) C.

On Raspberry Pi, backend and client, you choose:

Java*, Javascript or Python (for your own code).

Examples will be in Javascript with Node.js.

*) Bluetooth libraries might not be available.

Baseline

Which modules did you finish already?

Which languages can you write code in?

Which semester are you in right now?

Reply here*: tmb.gr/iot-baseline

*) Redirects to a Google Form, no login required.

Module *iot*

14 * 3 = 42 hours of lessons, including hands-on.

+ 16 hours of private study (reading or video).

+ 32 hours (per person) IoT team project.

=> 90 hours per person.

=> 3 ECTS credits.

Learning targets

Understanding IoT systems and their fundamental concepts, including the acquisition, transport and visualisation of sensor measurements, as well as integration with 3rd-party systems or services.

Developing the software part, without electronics*, of an end-to-end IoT system based on IoT platforms.

*) But including embedded programming.

Lessons 2019 — class 4ia

- | | | | |
|--------|--|--------|--------------------------------------|
| 18.02. | Introduction to the Internet of Things | 18.03. | Local Connectivity with Bluetooth LE |
| 25.02. | Microcontrollers, Sensors & Actuators | 25.03. | Raspberry Pi as a Local IoT Gateway |
| 04.03. | Sending Sensor Data to IoT Platforms | 01.04. | Messaging Protocols and Data Formats |
| 11.03. | Internet Protocols, HTTP and CoAP | 08.04. | Long Range Connectivity with LoRaWAN |

...

Lessons 2019 — class 4ia (ff.)

...

15.04. Dashboards and Apps
for Sensor Data

22.04. (No class)

29.04. Rule Based Integra-
tion of IoT Devices

06.05. (Project week)

13.05. Voice Control for
Connected Products

20.05. Raspberry Pi as an
IoT Edge Device

27.05. Assessment

03.06. Demo Day

10.06. (No class)

Learning target assessment

A mandatory, written assessment of 90 minutes.

A graded team project, due on *Demo Day*, 0 am.

Counting 50% each for the overall performance.

The final grade will be rounded to one-tenth.

There is no Modulschlussprüfung (MSP).

Assessment

90 minutes, closed book, written assessment.

1 A4 sheet of handwritten notes allowed.

No other material (slides, books, ...).

No communication (phone, ...).

There is no example yet, you're the first class.

Team project

2 person teams, building an IoT system.

32 hours of work per person, 1 prototype.

10' presentation of the project at *Demo Day*.

Project source code and setup steps on GitHub.

Both team members are able to explain the project.

Details on the following slides.

Team project code

GitHub repo with the following parts:

- 1) Embedded code / microcontroller firmware.
- 2) Glue Code used on the gateway or "in the cloud".
- 3) App or Web UI code, or IoT platform setup steps.

GitHub repo URL will be provided.

Team project presentation

- 1) Use-case.
 - 2) Reference model.
 - 3) Short, one slide interface docs.
 - 4) Issues you faced, how you solved them.
- + Live demo of the end-to-end IoT system prototype.

Slides to be submitted as PDF.

Team project prototype demo

Working end-to-end prototype, "device to cloud".

- 1) Sensor input on a IoT device triggers an event.
- 2) The event / measurement shows up online.
- 3) The event triggers actuator output*.

*) Same or separate device, details are up to you.

Plagiarism

Unfortunately has to be mentioned, sanctions apply.

From [Betrug und Plagiate bei Leistungsnachweisen](#):

"Wer in Arbeiten im Rahmen des Studiums Eigen- und Fremdleistung nicht unterscheidet, wer plagiiert, macht sich strafbar." - M. Meyer

Using 3rd-party code? Make it clear, check license. **n|w**

Lessons

You will need a laptop with admin rights.

There will be quite some [hardware](#) involved.

Content of slides and hands-on will be assessed.

Slides come as PDF, with many links, to learn more.

Hands-on sessions

"Be excellent to each other", asking / helping is OK.

Google ([DDG.co](#), ...) error messages to fix issues.

Coping blindly does not lead to new insight.

Reading other people's code helps a lot.

Slides, code & hands-on materials

<http://tmb.gr/iot> →

<https://github.com/tamberg/fhnw-iot>

01/

hello.c

README.md → Slides, Hands-on

02/

...

Hands-on and project results

<https://github.com/fhnw-iot-4ia>

fhnw-iot-work-01

Repo template w/ link

fhnw-iot-work-01-USER

Repo fork per user

README.md

Hands-on exercise

my_result.ino

"Private", tutor & user

Why GitHub? Professional tool and reliable backup.

Why a repo per lesson? Easier than updating forks.

Communication

<https://fhnw-iot.slack.com/>

- | | |
|-----------|------------------------------|
| #general | Questions and announcements. |
| #random | Off-topic, random posts. |
| #arduino | Arduino questions. |
| # ... | More channels. |
| • tamberg | Direct messages to a person. |

Slack App is recommended, mobile or desktop.

Books on IoT

A book is not required for this course.

We will read individual articles on demand.

Our [Wiki](#) has [a list of books](#) on a range of topics.



Tools

Terminal (MacOS) or *cmd* (Windows).

Text editor, e.g. *nano* or **VS Code**.

C (via Arduino), Java, JS, Python.

Code version control with *git*.

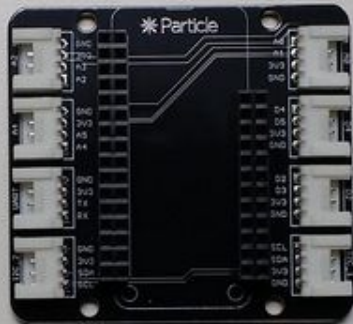
Simple tools, no "magic" => deep understanding.

Hardware

The course is based on the following [hardware](#):

Raspberry Pi Zero W	Linux, I/O, Wi-Fi, BLE
Feather Huzzah ESP8266	Microcontroller, Wi-Fi
Feather nRF52840 Express	Microcontroller, BLE
FeatherWing RFM95W	Extension, LoRaWAN
Grove Sensors & Actuators	Plug & play

Why? See [IoT Hardware for CS bachelor students](#).





Motivation

I'm highly motivated to provide the best experience.

Hardware takes a lot of trial and error to master.

If something does not work, try again, twice.

It's worth the effort, IoT is here to stay :)

Feedback?

Find me on <https://fhnw-iot.slack.com/>

Or email thomas.amberg@fhnw.ch

Slides, code & hands-on: tmb.gr/iot-0

