IoT Engineering
0: Syllabus

Slides: tmb.gr/iot-0

Overview

The *syllabus* makes sure you know what's up:

What you can expect from this course.

What is expected from you.

Hello

Thomas Amberg (@tamberg), Software Engineer.

"Prof. of Internet of Things" at FHNW since 2018.

Founder of Yaler, "secure remote access for IoT".

Organising an open IoT Meetup group in Zürich.

Email thomas.amberg@fhnw.ch

Beta alert

IoT Engineering is a relatively new course.

Content might still be incomplete.

Things will go wrong.

Found a bug? Let me know! Feedback is welcome.

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*

15 * 3 = 45 hours of lessons, including hands-on.

+ 13 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 2020 — class 5ibb1

14.09. Introduction to the Internet of Things

21.09. Microcontrollers, Sensors & Actuators

28.09. Sending Sensor Data to IoT Platforms

05.10. Internet Protocols, HTTP and CoAP

12.10. Local Connectivity with Bluetooth LE

19.10. Raspberry Pi as a Local IoT Gateway

26.10. Messaging Protocols and Data Formats

02.11. Long Range Connectivity with LoRaWAN

...

Lessons 2020 — class 5ibb1 (ff.)

09.11. Dashboards and Apps for Sensor Data

16.11. Rule Based Integration of IoT Devices

23.11. (Project week)

30.11. Voice Control for Connected Products

07.12. Raspberry Pi as an IoT Edge Device

14.12. From Prototype to Connected Product

21.12. (No class)

28.12. (No class)

04.01. Assessment

11.01. Demo Day

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, …).

Here are example assessments: FS19, HS19.

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 follow. Here's an example project.

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.

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.

Copying 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/
  - README.md → Slides, Hands-on
- 02/
  - Arduino/ESP8266_Blink/ESP8266_Blink.ino
  - …

Hands-on and project results

https://github.com/fhnw-iot-5ibb1

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.

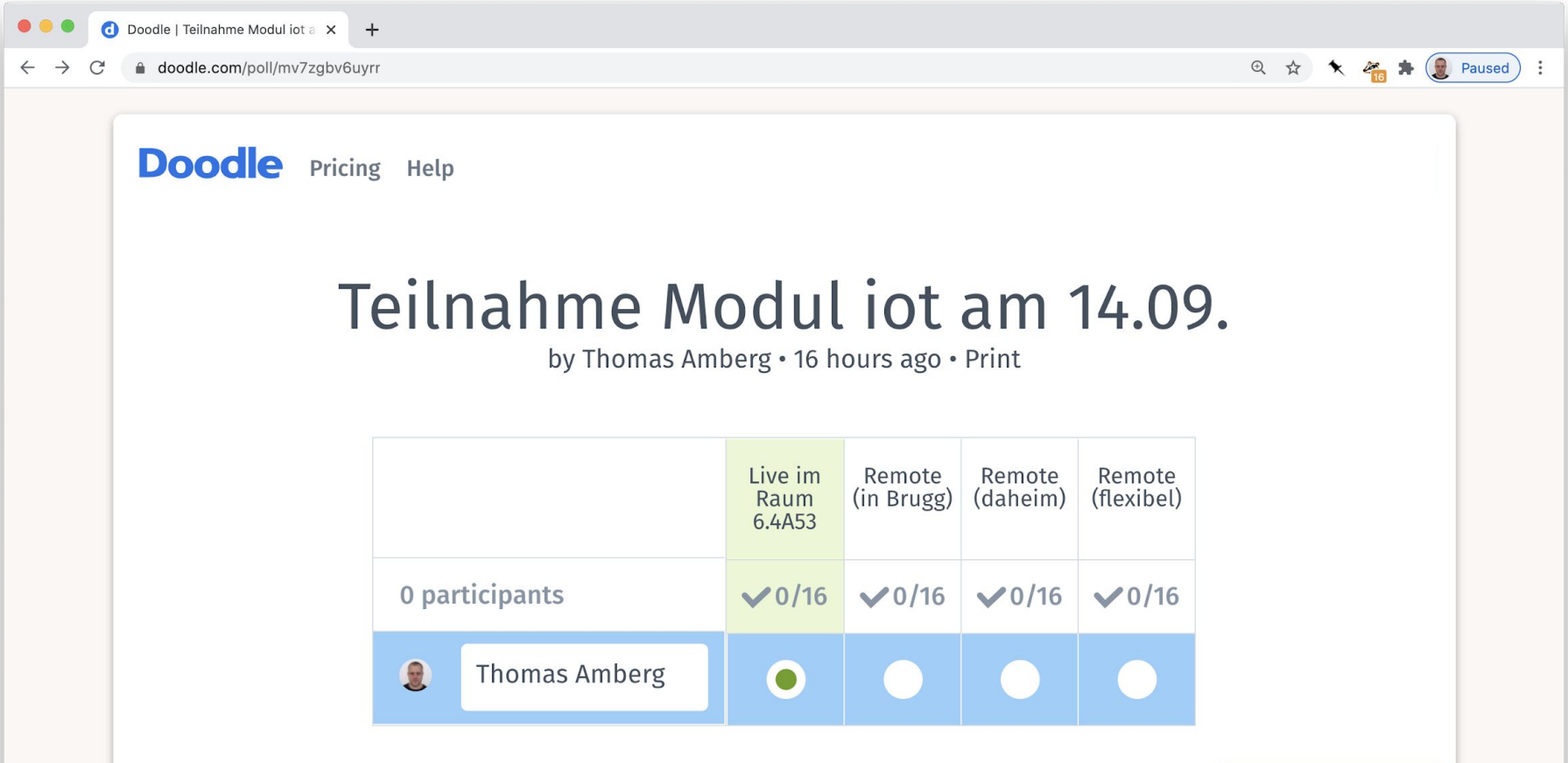
Communication

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.

Attendance

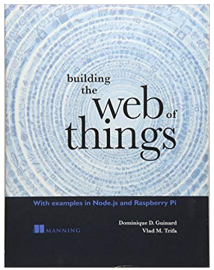

Doodle | Teilnahme Modul iot a
doodle.com/poll/mv7zgbv6uyrr
Paused
Doodle
Pricing
Help
Teilnahme Modul iot am 14.09.
by Thomas Amberg • 16 hours ago • Print
Live im Raum 6.4A53
Remote (in Brugg)
Remote (daheim)
Remote (flexibel)
0 participants
0/16
0/16
0/16
0/16
Thomas Amberg

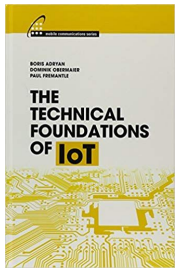
Books on IoT

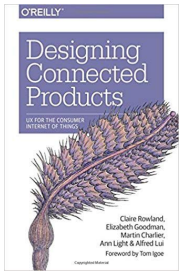
A book is not required for this course.

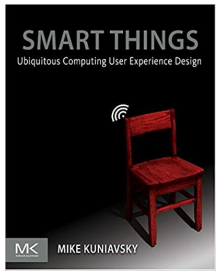
We will read individual articles on demand.

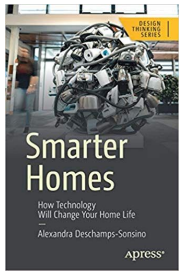
Our Wiki has a list of books on a range of topics.

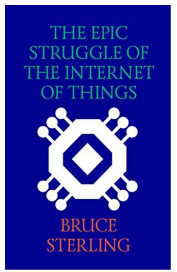


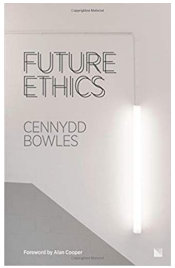


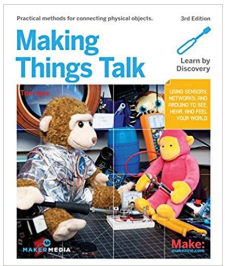


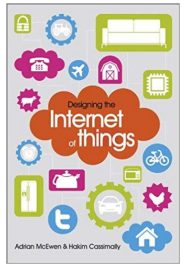












Tools

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

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

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

Code version control with *git*.

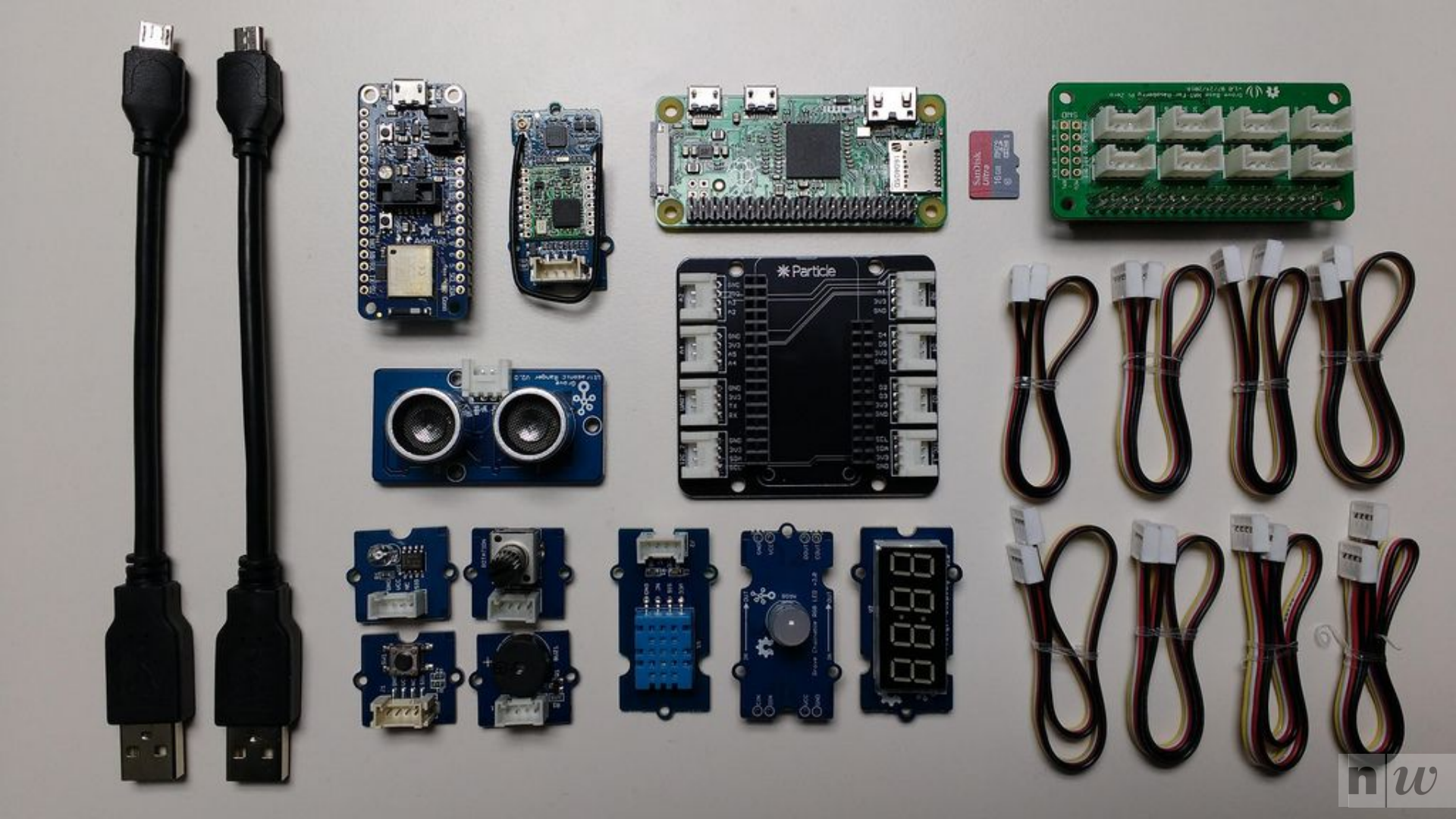
Simple tools, no "magic" => deep understanding.

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.


Particle

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 or questions?

Write me on https://fhnw-iot.slack.com/

Or email thomas.amberg@fhnw.ch

Thanks for your time.