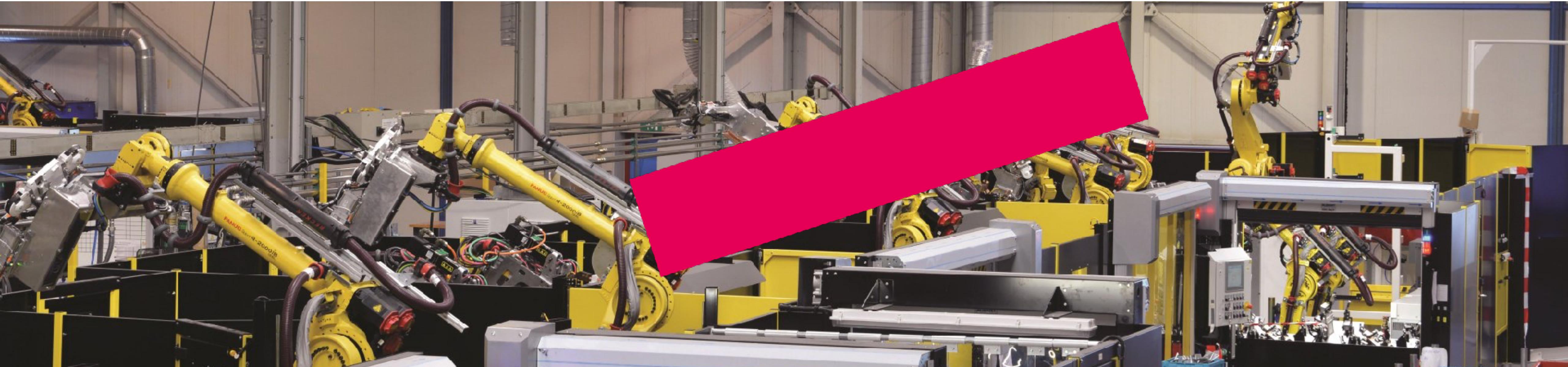


Connected Factories lecture 3/6: Protocols



Associate Degree Smart Industry
Faculty of Engineering and Automotive

johan.korten@han.nl

V1.0 April 2, 2020

Originele Leerdoelen (kneden en aanpassen)

De student heeft kennis en begrip van internationale standaarden op het gebied van gekoppelde IT systemen (Cyber Physical Systems). De student heeft kennis van *digitale identificatie* en uitwisseling van objecten als offertes, tekeningen, orders, transportinformatie en facturen conform open industrie standaarden.

De student heeft kennis en begrip van technologieën als blockchain, glasvezelverbindingen en 5G. De student heeft kennis en begrip van juridische contracten (zoals copyrights op sensor data, databankwet, privacy wetgeving, gebruiksrechten op software in equipment). De student kan zijn bedrijf adviseren welke mogelijkheden deze technologieën bieden op het gebied van integrale ketenoptimalisering.

Schedule

	Theme	
Lecture 1	Introduction	
Lecture 2	Network connections	
Lecture 3	Today: Network protocols	
Lecture 4	Interconnection	
Lecture 5	Integration	
Lecture 6	Trusting the Network	
Assessment		

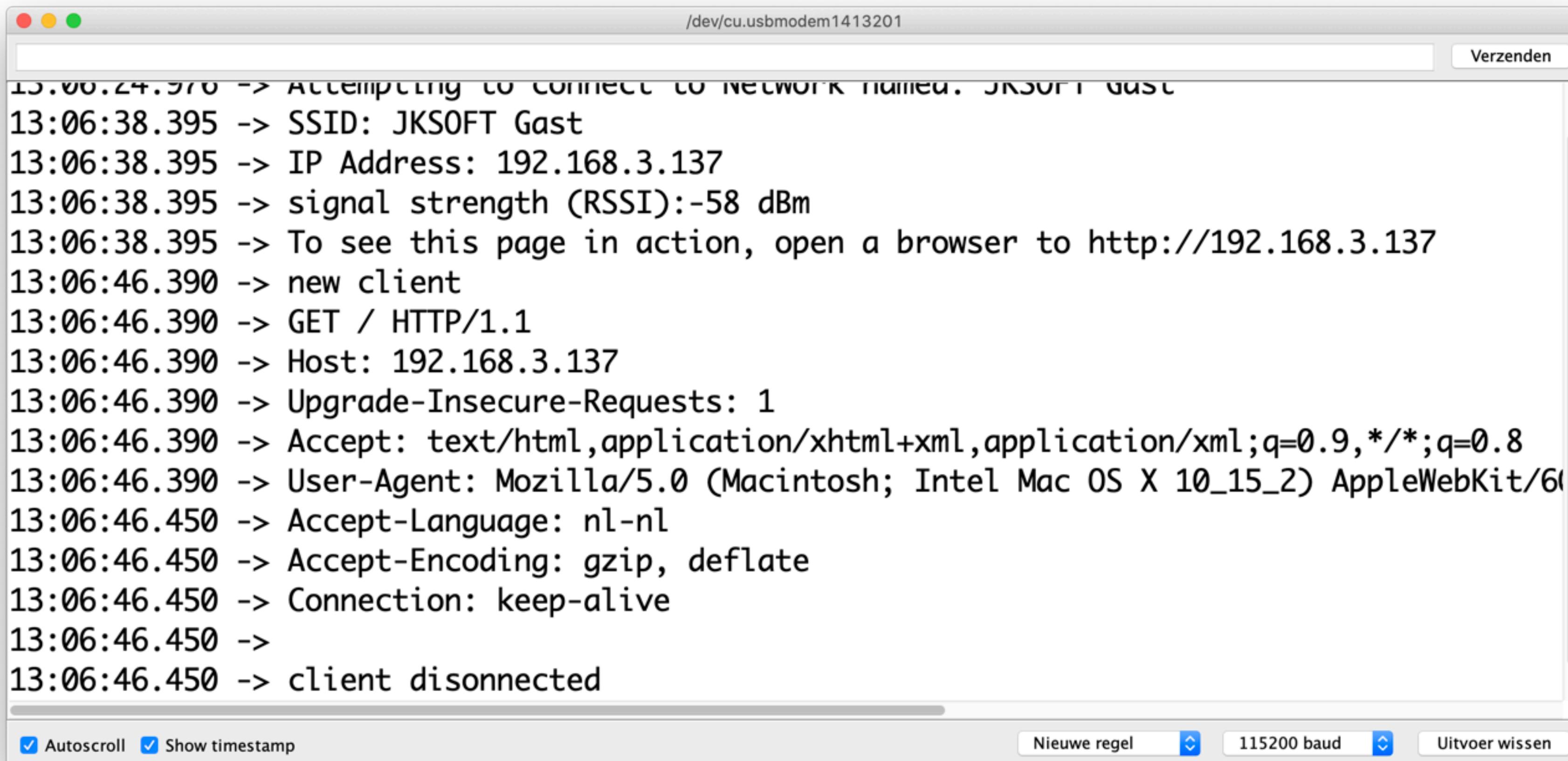
Recap: what did we learn the last two weeks...

Industry 4.0 is all about connectivity

Same functionality can be achieved using different type of connections

Protocols

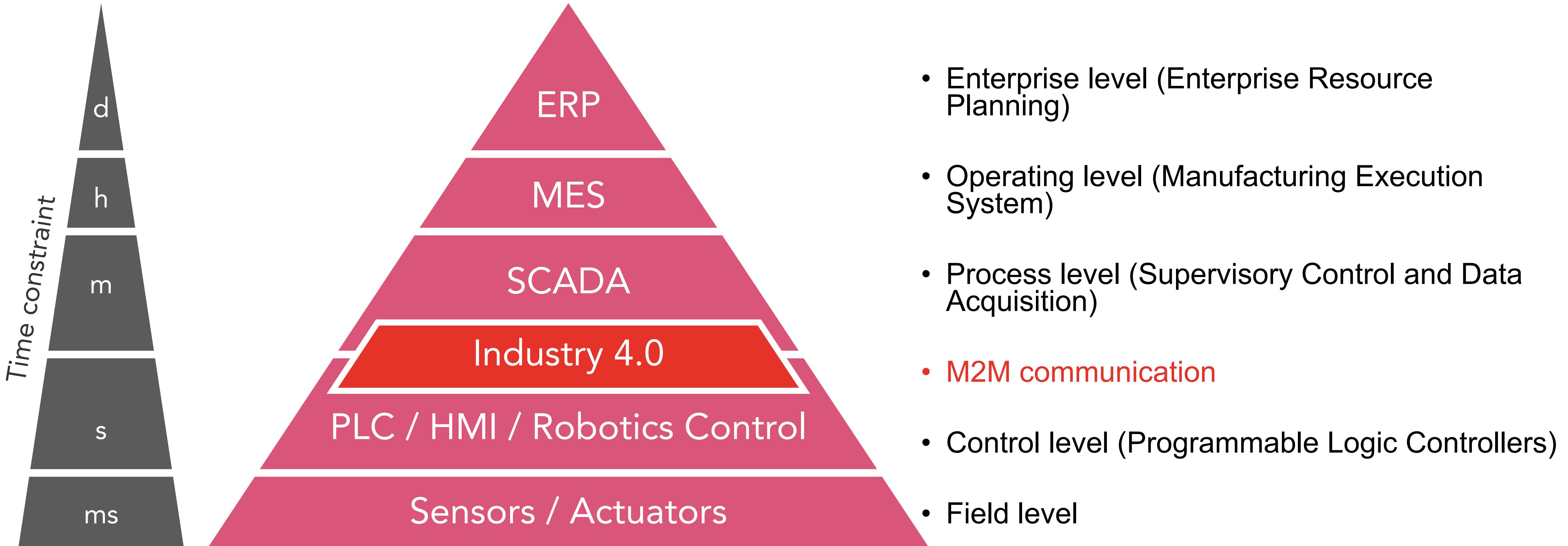
Remember what you saw in the terminal in the Arduino lab of week 1?



```
13:06:38.390 -> Attempting to connect to network named. JKSOFT Gast
13:06:38.395 -> SSID: JKSOFT Gast
13:06:38.395 -> IP Address: 192.168.3.137
13:06:38.395 -> signal strength (RSSI):-58 dBm
13:06:38.395 -> To see this page in action, open a browser to http://192.168.3.137
13:06:46.390 -> new client
13:06:46.390 -> GET / HTTP/1.1
13:06:46.390 -> Host: 192.168.3.137
13:06:46.390 -> Upgrade-Insecure-Requests: 1
13:06:46.390 -> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
13:06:46.390 -> User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_2) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15
13:06:46.450 -> Accept-Language: nl-nl
13:06:46.450 -> Accept-Encoding: gzip, deflate
13:06:46.450 -> Connection: keep-alive
13:06:46.450 ->
13:06:46.450 -> client disconnected
```

Autoscroll Show timestamp Nieuwe regel Uitvoer wissen

Industrial Connections: Automation Pyramid (IEC 62264 / IEC615112)



Model was derived from: <http://eecatalog.com/chipdesign/2018/02/22/new-class-of-embedded-design-emerges-to-support-virtualized-fog-servers/>
See also: Niedermaier, M., Hanka, T., Plaga, S., von Bodisco, A., & Merli, D. (2019). Efficient Passive ICS Device Discovery and Identification by MAC Address Correlation. *arXiv preprint arXiv:1904.04271*.
https://link.springer.com/chapter/10.1007/978-3-319-56430-2_3

Industrial Connections: basic software standards (protocols)

(Lower level) Protocol	Type / Characteristic	Standard(s) / Organization
TCP/IP	Internet protocol suite (also common for non-industrial usage)	RFC 1958
UDP	User Datagram Protocol (also common for non-industrial usage)	RFC 768

Industrial Connections: more specific standards

High level Protocol	Type / Characteristic	Standard(s) / Organization
MQTT	Message Queuing Telemetry Transport for M2M communication	ISO/IEC PRF 20922
CIP	Common Industrial Protocol	ODVA

Modbus: varieties using different protocols

Modbus RTU

Modbus ASCII

Modbus TCP(/IP)

Modbus over TCP/IP

Modbus over UDP

Modbus Plus (Modbus+, MB+ or MBP)

Pemex Modbus

Enron Modbus

See: <http://www.modbus.org>

▲ PROFINET: consequences of different protocols

- TCP/IP (Standard) reaction time ± 100 ms
- RT (Real-Time) cycle times up to 10 ms
- IRT (Isochronous Real-Time) cycles times of less than 1 ms

PROFIsafe (PROFIBUS safety or PROFINET safety) (IEC 61508) (**Verder uitwerken**)

See: <http://www.modbus.org>; https://en.wikipedia.org/wiki/Isochronous_timing;

Cisco modules

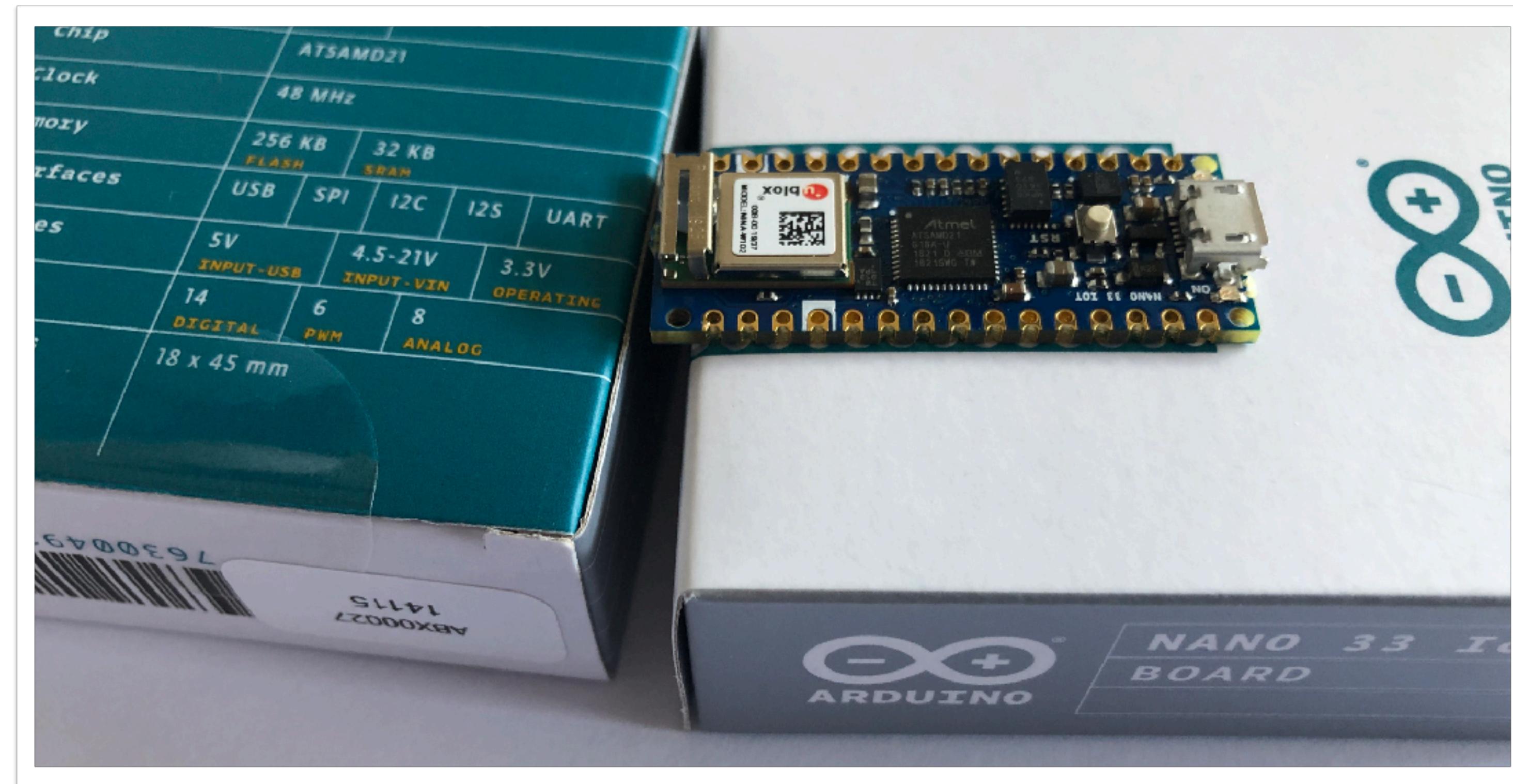
Remember, course URL: <https://www.netacad.com/portal/learning>

Homework for next session:

- Les 4. Slimme oplossingen t/m 4.5,
 - 4.6 Niet
-
- Les 5. Slimme apparaten 5.1, 5.3, 5.5 - 5.9
 - 5.2, 5.4 Niet, Opdracht slimme stad niet.

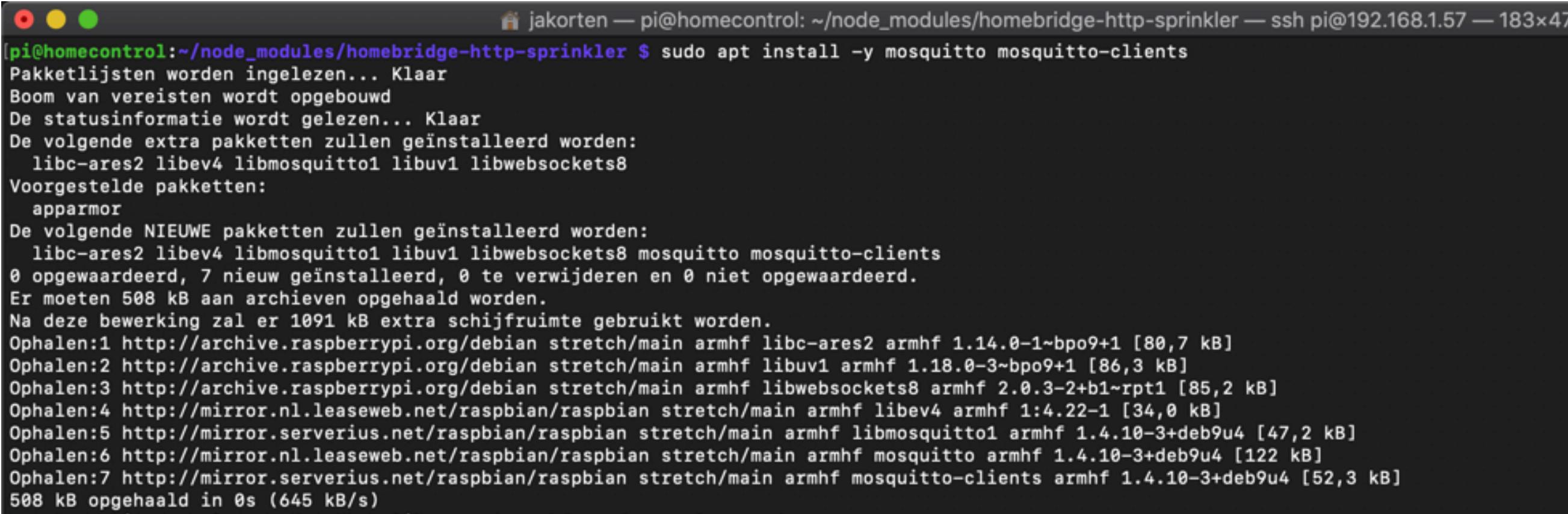
Towards your own IoT / MQTT device.

Practical: connecting your Nano 33 IoT Board to your Phone network.



Using your Pi as an MQTT broker

In case you haven't done that yet...



```
[pi@homecontrol:~/node_modules/homebridge-http-sprinkler $ sudo apt install -y mosquitto mosquitto-clients
Pakketlijsten worden ingelezen... Klaar
Boom van vereisten wordt opgebouwd
De statusinformatie wordt gelezen... Klaar
De volgende extra pakketten zullen geïnstalleerd worden:
 libc-ares2 libev4 libmosquitto1 libuv1 libwebsockets8
Voorgestelde pakketten:
 apparmor
De volgende NIEUWE pakketten zullen geïnstalleerd worden:
 libc-ares2 libev4 libmosquitto1 libuv1 libwebsockets8 mosquitto mosquitto-clients
0 opgewaardeerd, 7 nieuw geïnstalleerd, 0 te verwijderen en 0 niet opgewaardeerd.
Er moeten 508 kB aan archieven opgehaald worden.
Na deze bewerking zal er 1091 kB extra schijfruimte gebruikt worden.
Ophalen:1 http://archive.raspberrypi.org/debian stretch/main armhf libc-ares2 armhf 1.14.0-1~bpo9+1 [80,7 kB]
Ophalen:2 http://archive.raspberrypi.org/debian stretch/main armhf libuv1 armhf 1.18.0-3~bpo9+1 [86,3 kB]
Ophalen:3 http://archive.raspberrypi.org/debian stretch/main armhf libwebsockets8 armhf 2.0.3-2+b1~rpt1 [85,2 kB]
Ophalen:4 http://mirror.nl.leaseweb.net/raspbian/raspbian stretch/main armhf libev4 armhf 1:4.22-1 [34,0 kB]
Ophalen:5 http://mirror.serverius.net/raspbian/raspbian stretch/main armhf libmosquitto1 armhf 1.4.10-3+deb9u4 [47,2 kB]
Ophalen:6 http://mirror.nl.leaseweb.net/raspbian/raspbian stretch/main armhf mosquitto armhf 1.4.10-3+deb9u4 [122 kB]
Ophalen:7 http://mirror.serverius.net/raspbian/raspbian stretch/main armhf mosquitto-clients armhf 1.4.10-3+deb9u4 [52,3 kB]
508 kB opgehaald in 0s (645 kB/s)
```

Update your packages:
sudo apt update

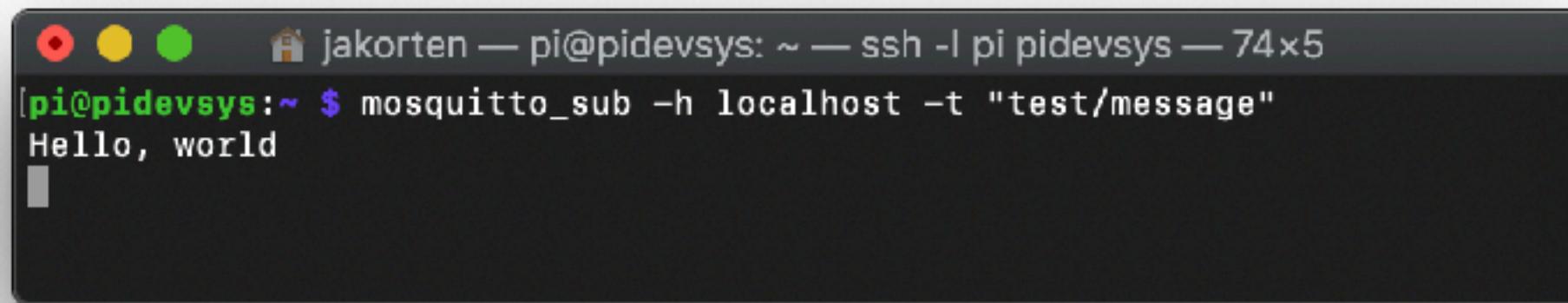
Install Mosquitto:
sudo apt install -y mosquitto mosquitto-clients

Auto start Mosquitto on boot up:
sudo systemctl enable mosquitto.service

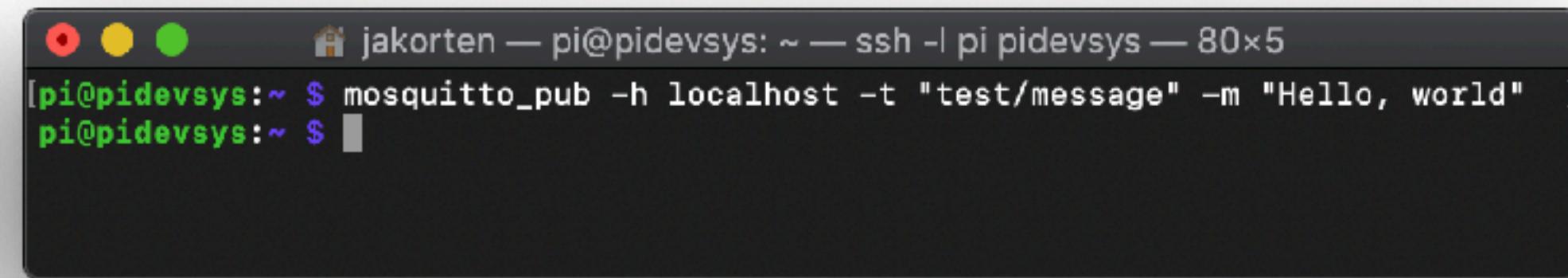
See also: <https://appcodelabs.com/introduction-to-iot-build-an-mqtt-server-using-raspberry-pi>

Test MQTT on your Pi

Open two terminals:



```
[pi@pidevsys:~ $ mosquitto_sub -h localhost -t "test/message"
Hello, world]
```



```
[pi@pidevsys:~ $ mosquitto_pub -h localhost -t "test/message" -m "Hello, world"
pi@pidevsys:~ $ ]
```

Terminal 1:

mosquitto_sub -h localhost -t "test/message"

Terminal 2:

mosquitto_pub -h localhost -t "test/message" -m "Hello, world"

Test MQTT on your Pi

Open another (local) terminal:

The image displays three terminal windows. The top-left window is on a Pi (raspberrypi OS), showing the command `mosquitto_sub -h localhost -t "test/message"` and the output "Hello, world" followed by "Hello from remote". The top-right window is also on the Pi, showing the command `mosquitto_pub -h localhost -t "test/message" -m "Hello, world"`. The bottom window is on a Macbook (zsh shell), showing the command `mosquitto_pub -h pidevsys -t "test/message" -m "Hello from remote"`.

```
[pi@pidevsys:~ $ mosquitto_sub -h localhost -t "test/message"
Hello, world
Hello from remote
[pi@pidevsys:~ $ mosquitto_pub -h localhost -t "test/message" -m "Hello, world"
pi@pidevsys:~ $ ]
[jakorten — zsh — 104x5
jakorten@MBP-van-JA-2221 ~ % mosquitto_pub -h pidevsys -t "test/message" -m "Hello from remote"
[jakorten@MBP-van-JA-2221 ~ %
[
```

Terminal 1:

mosquitto_sub -h localhost -t "test/message"

Terminal 2:

mosquitto_pub -h localhost -t "test/message" -m "Hello, world"

Local (Macbook) Terminal 3:

mosquitto_pub -h pidevsys -t "test/message" -m "Hello from remote"

Installing Mosquitto on Windows:

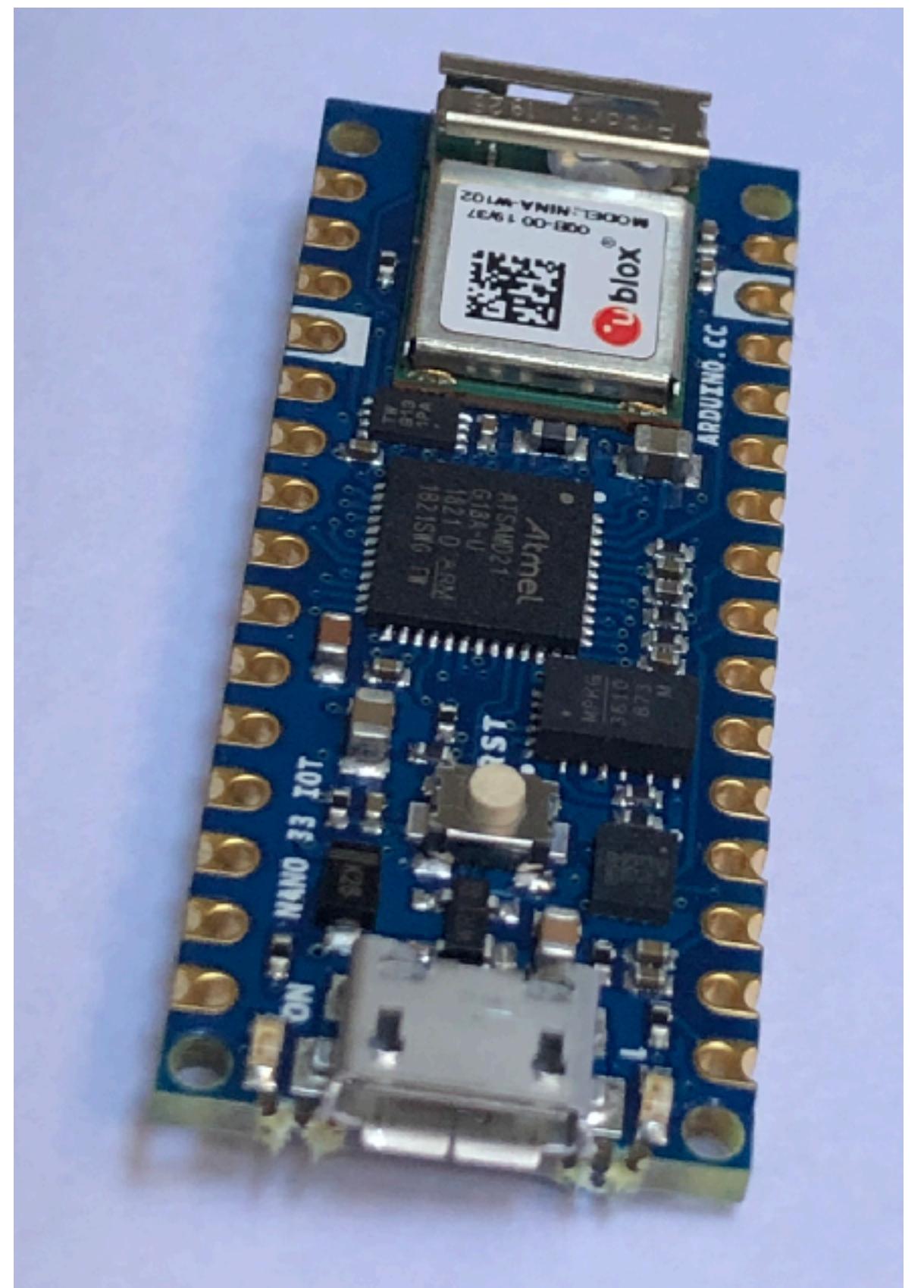
<http://www.steves-internet-guide.com/install-mosquitto-broker/>

Towards your own IoT / MQTT device.

During online session we will do this live, but you can also do it for yourself.

- Turn the module into a MQTT IoT client.
- Try to switch an LED using MQTT.

Video: <https://youtu.be/9z2gyEKuEG8>



Arduino Menu cheat sheet... (not entirely visible in videos)



Reflection Arduino Lab

- If you think back of lesson 1 about the Sensor and Actuator Reference Architecture (SARA), what architecture would fit in best with the MQTT solution (monolithic, silo or horizontal layers), and why?

If you look in the Arduino example you also see the following code:

char pubTopic[] = "arduino/ledState";

- What would that be useful for?
- How would one be able to use it?



“