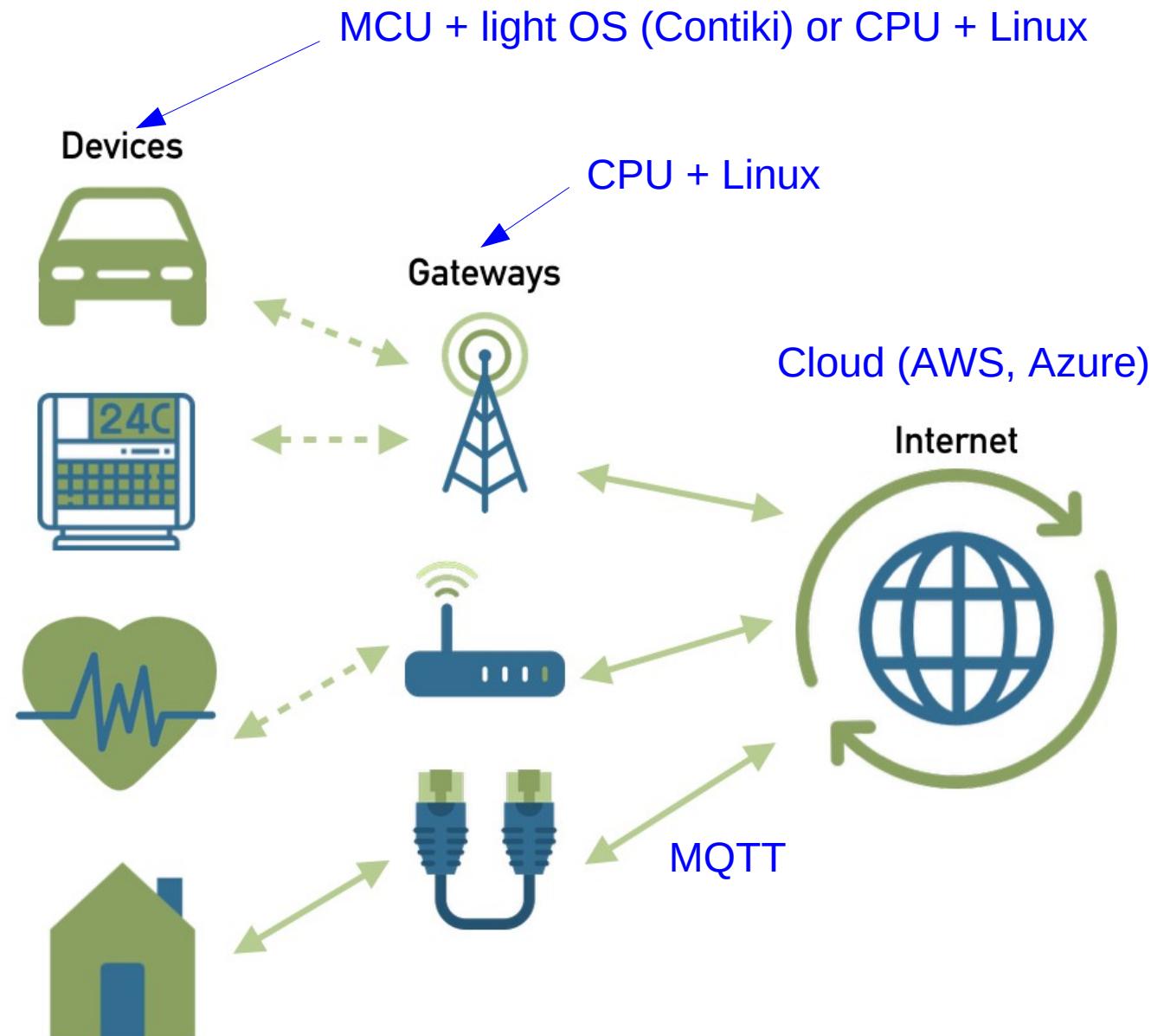




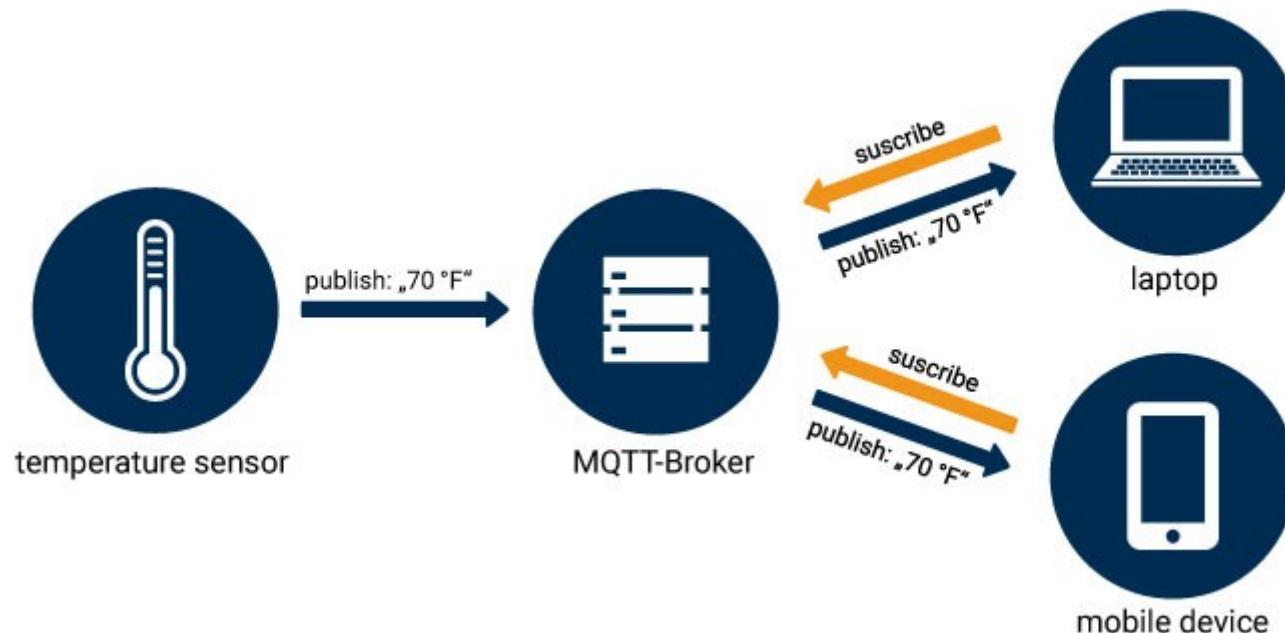
Building a **yocto** • sensor PROJECT

Pierre Ficheux (pierre.ficheux@smile.fr)

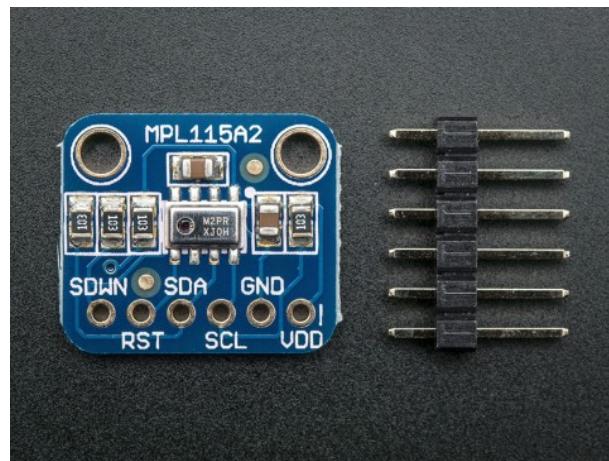
February 2023



- Messaging Queue Telemetry Transport, created by IBM en 1999
- Very famous in IoT world
- TCP port 1883, SSL port 8883

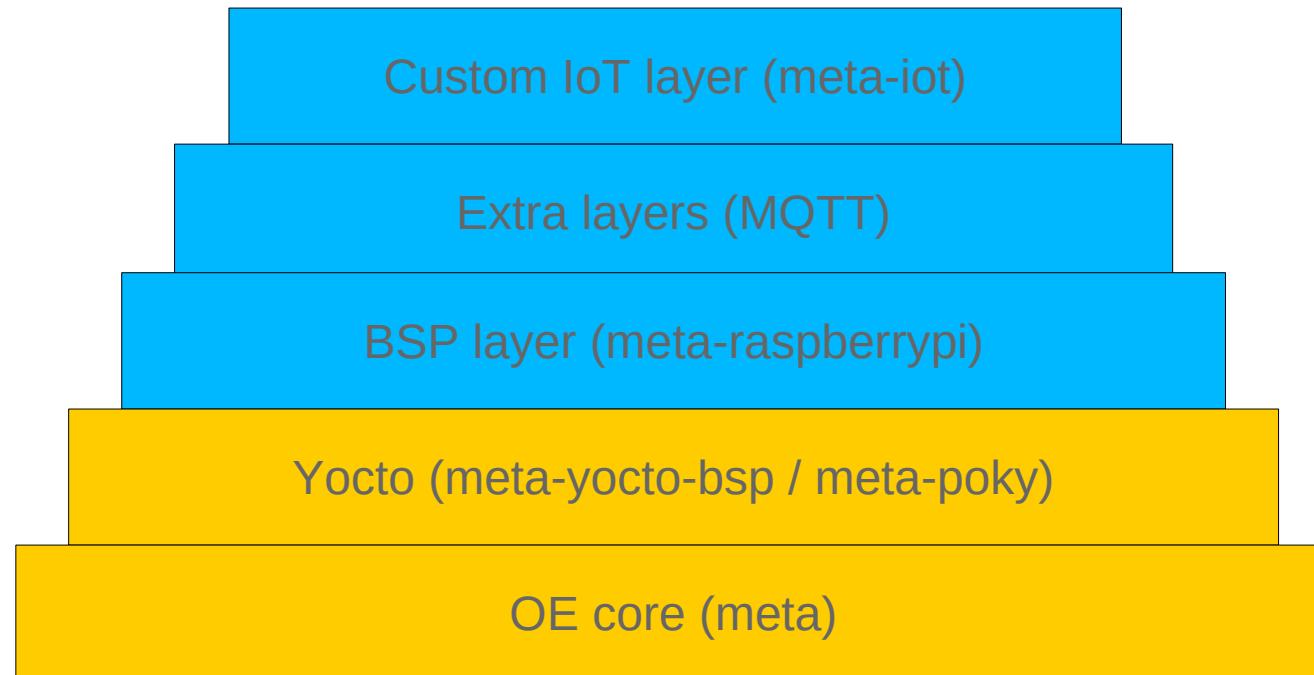


- Temperature sensor demonstration
 - Raspberry Pi board
 - I2C sensor (MPL115A2)
 - Wi-Fi or Ethernet connection
 - MQTT protocol
- Sensor is simulated (no I2C) if QEMU target





Yocto layers principle



- █ Extra layers
- █ Standard layers

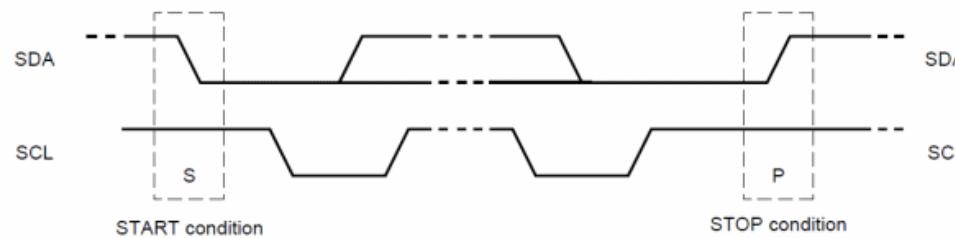


- We use “core-image-base” image (Wi-Fi support)
- IoT layer will contain
 - Dedicated recipe for MPL115A2 (.bb)
 - Mostly extended recipes (.bbappend) for the rest
 - I2C device tree support
 - I2C drivers
 - Network configuration
- Testing could be done with local.conf
- Finally, we create a dedicated image “iot-sensor-image”



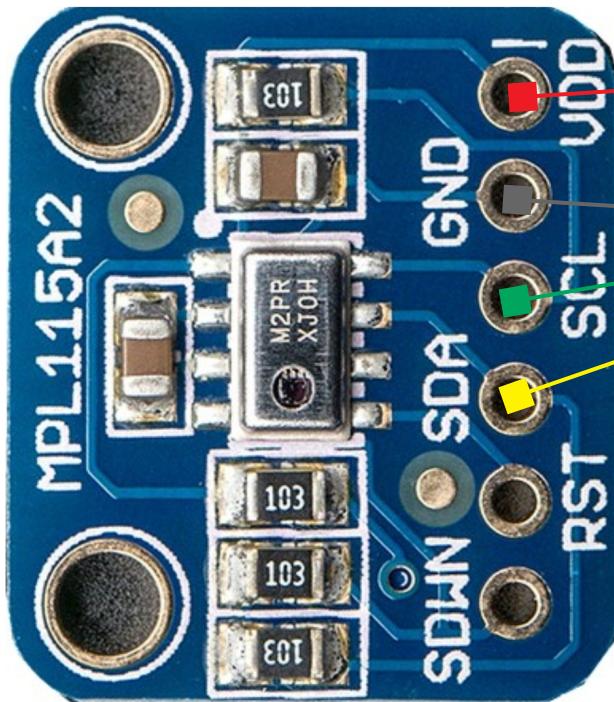
- Install MosQuiTTo Ubuntu package
- Use public broker is “test.mosquitto.org”
- Subscribe (and so receive the message)
`$ mosquitto_sub -h test.mosquitto.org -t my-topic`
- Publish message with a “topic”
`$ mosquitto_pub -h test.mosquitto.org -t my-topic -m "hello world"`
- Provided by Intel in “meta-openembedded”
`IMAGE_INSTALL:append = " mosquitto-clients"`
- The Linux PC can be used a as local broker (for testing purpose)

- I2C was designed by Philips in 1982
- Master/slave serial bus
- Two signals
 - SDA (1 bit data)
 - SCL (clock)
- Start / stop conditions
- Slave address is on 7 bits
- One or several control/data registers
- Chaining is simple





Connecting the sensor to the Pi



Raspberry Pi J8 Header (Model B+)			
GPIO#	NAME	NAME	GPIO#
3	3.3 VDC Power	-	5.0 VDC Power
8	GPIO 8 SDA1 (I2C)	D	2
9	GPIO 9 SCL1 (I2C)	5	4
7	GPIO 7 GPCLK0	Ground	6
0	GPIO 0	8	8
2	GPIO 2	10	10
3	GPIO 3	12	12
12	3.3 VDC Power	14	14
13	GPIO 12 MOSI (SPI)	16	16
14	GPIO 13 MISO (SPI)	1	1
21	GPIO 14 SCLK (SPI)	4	4
22	Ground	5	5
23	SDA0 (I2C ID EEPROM)	6	6
24	GPIO 21 GPCLK1	10	10
25	GPIO 22 GPCLK2	11	11
39	GPIO 23 PWM1	SCL0 (I2C ID EEPROM)	26
37	GPIO 24 PCM_FS/PWM1	27	27
35	GPIO 25	28	28
33	Ground	29	29
31	39	PCM_DIN	30
30	27	PCM_DOUT	32
29	28	36	34
28	29	38	38
27	30	40	40
26	31	Ground	36
24	32	Ground	34
23	33	Ground	32
22	34	Ground	30
21	35	Ground	28
20	36	Ground	26
19	37	Ground	24
18	38	Ground	22
17	39	Ground	20
16	40	Ground	18
15	5.0 VDC Power	21	15
16	5.0 VDC Power	22	16
17	Ground	23	1
18	Ground	24	1
19	Ground	25	1
20	Ground	26	1
21	Ground	27	1
22	Ground	28	1
23	Ground	29	1
24	Ground	30	1
25	Ground	31	1
26	Ground	32	1
27	Ground	33	1
28	Ground	34	1
29	Ground	35	1
30	Ground	36	1
31	Ground	37	1
32	Ground	38	1
33	Ground	39	1
34	Ground	40	1



I2C Yocto support for the Pi

- User space access from /sys and /dev
 - /sys/class/i2c-adapter/i2c-1
 - /dev/i2c-1
- We must load the following modules:

```
# modprobe i2c-dev
# modprobe i2c-bcm2708
```
- Updating config.txt with a .bbappend

```
dtparam=i2c_arm=on
```
- Auto-loading “i2c-dev” in a kernel .bbappend

```
KERNEL_MODULE_AUTOLOAD += "i2c-dev"
```
- The “i2c-tools” package can be helpful

```
IMAGE_INSTALL:append = " i2c-tools"
```



- Specific recipe is necessary
- User space control program (in C)
- SysvInit script started at boot time (simpler than systemd)
`IMAGE_INSTALL:append = " mpl115a2"`
- Use “update-rc.d” class (service started at runlevel 99)
 - `INITSCRIPT_NAME = "mpl115a2"`
 - `INITSCRIPT_PARAMS = "defaults 99"`
- The script reads the temperature and publish it with MQTT



The QEMU version

- Same principle
- No I2C bus support (simulated sensor value)
- Use the “qemu-sensor” recipe instead of “mpl115a2”