



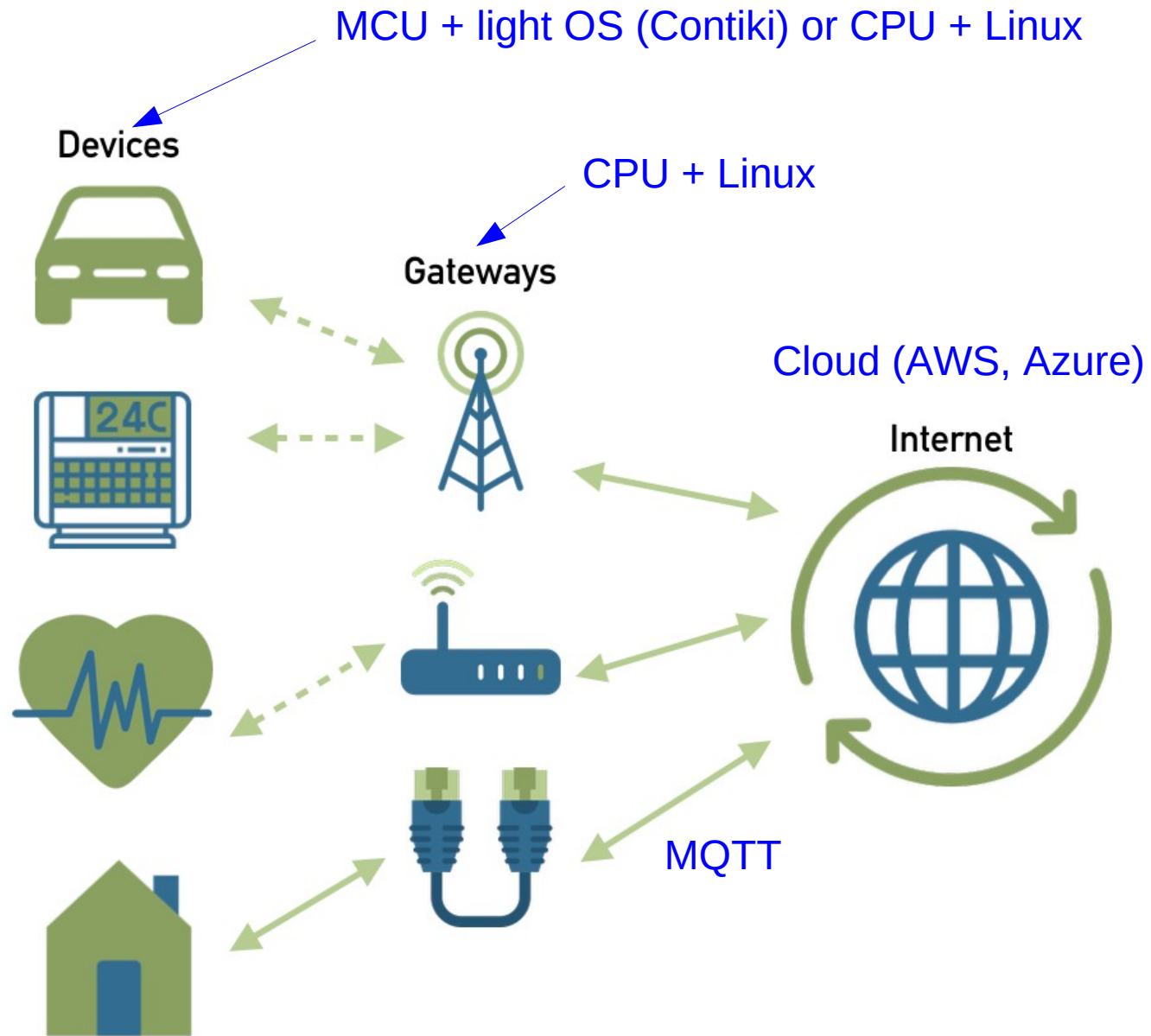
Building a **yocto** · sensor PROJECT

Pierre Ficheux (pierre.ficheux@smile.fr)

February 2023



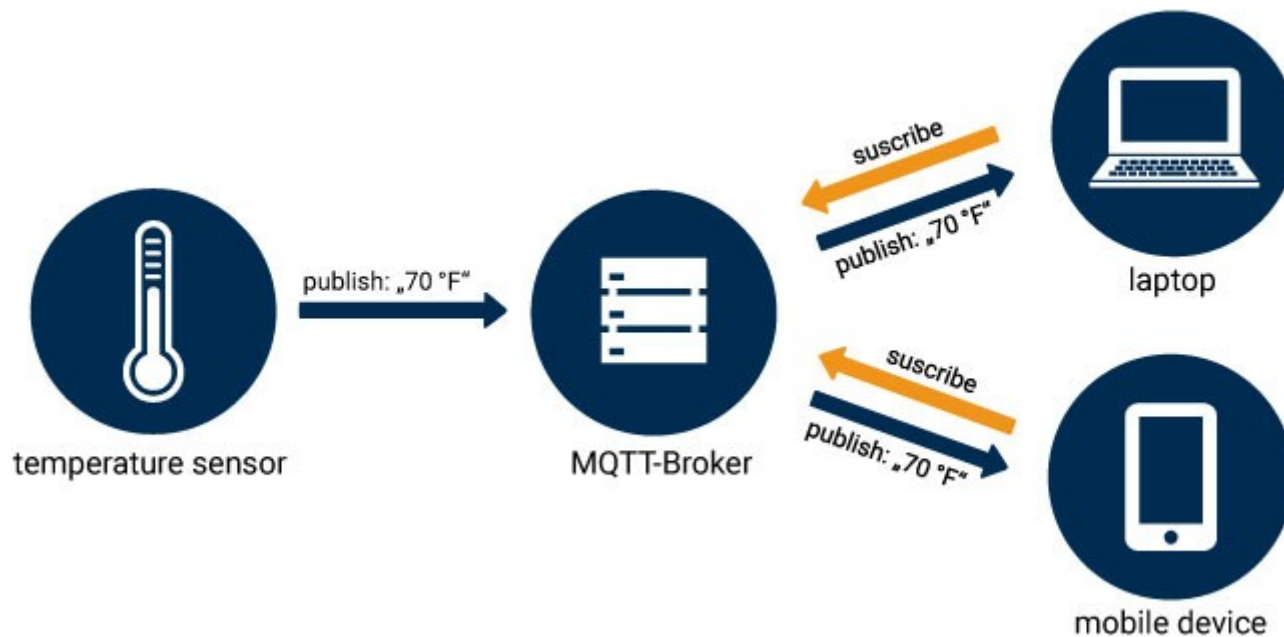
IoT ?





MQTT principle

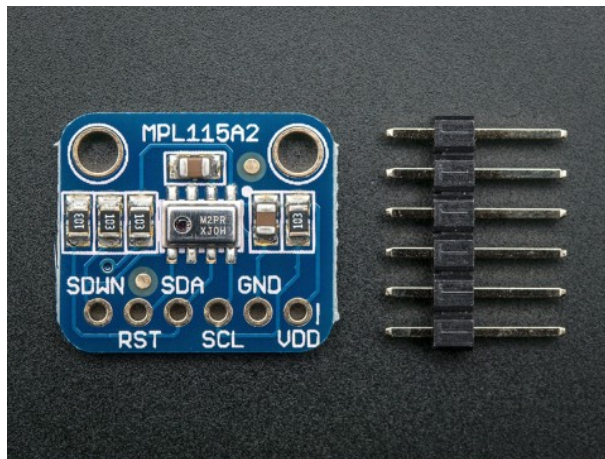
- Messaging Queue Telemetry Transport, created by IBM en 1999
- Very famous in IoT world
- TCP port 1883, SSL port 8883





Yocto IoT sensor use case

- 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 MosQiTTo 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”

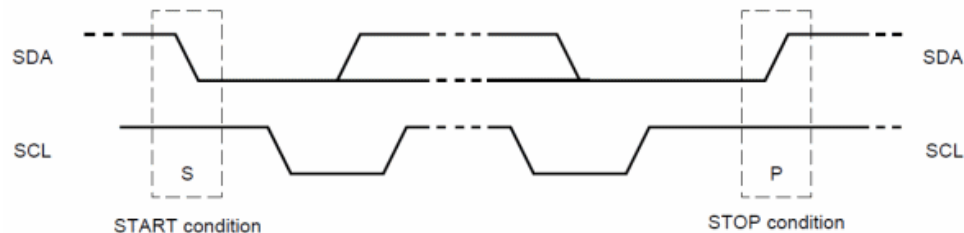
```
$ mosquitto_pub -h test.mosquitto.org -t my-topic -m "hello world"
```
- Provided by Intel in “meta-openembedded”

```
IMAGE_INSTALL:append = " mosquitto-clients"
```
- The Linux PC can be used a as local broker (for testing purpose)



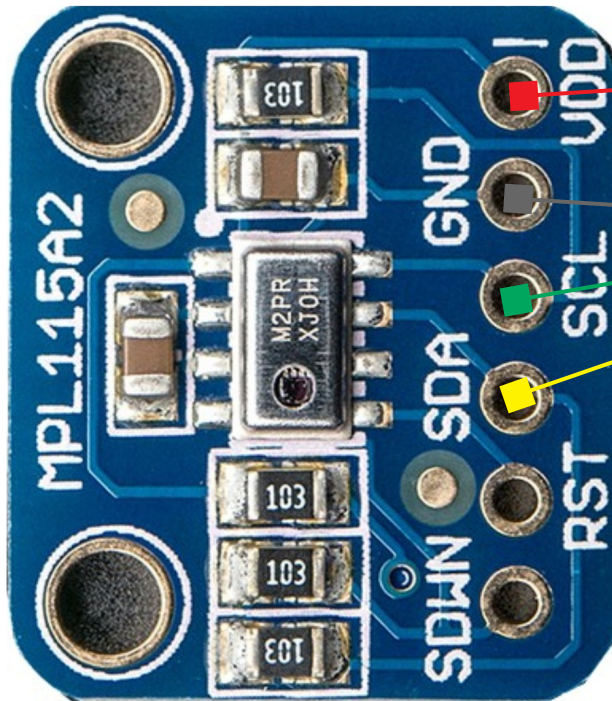
What is I2C ?

- 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 VDC Power	1		2	5.0 VDC Power		
8	GPIO 8 SDA1 (I2C)	3		4	5.0 VDC Power		
9	GPIO 9 SCL1 (I2C)	5		6	Ground		
7	GPIO 7 GPCLK0	7		8	GPIO 15 TxD (RS232)	15	
	Ground	9		10	GPIO 16 RxD (RS232)	16	
0	GPIO 0	11		12	GPIO 1 PCM_CLK/PWM0	1	
2	GPIO 2	13		14	Ground		
3	GPIO 3	15		16	GPIO 4	4	
	3.3 VDC Power	17		18	GPIO 5	5	
12	GPIO 12 MOSI (SPI)	19		20	Ground		
13	GPIO 13 MISO (SPI)	21		22	GPIO 6	6	
14	GPIO 14 SCLK (SPI)	23		24	GPIO 10 CE0 (SPI)	10	
	Ground	25		26	GPIO 11 CE1 (SPI)	11	
	SDA0 (I2C ID EEPROM)	27		28	SCL0 (I2C ID EEPROM)		
21	GPIO 21 GPCLK1	29		30	Ground		
22	GPIO 22 GPCLK2	31		32	GPIO 26 PWM0	26	
23	GPIO 23 PWM1	33		34	Ground		
24	GPIO 24 PCM_FS/PWM1	35		36	GPIO 27	27	
25	GPIO 25	37		38	GPIO 28 PCM_DIN	28	
	Ground	39		40	GPIO 29 PCM_DOUT	29	

<http://www.pi4j.com>



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"`



Handling MPL115A2

- Specific recipe is necessary
- User space control program (in C)
- SysvInit script started at boot time (simpler than systemd)
`IMAGE_INSTALL:append = " mp1115a2"`
- Use “update-rc.d” class (service started at runlevel 99)
 - `INITSCRIPT_NAME = "mp1115a2"`
 - `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”