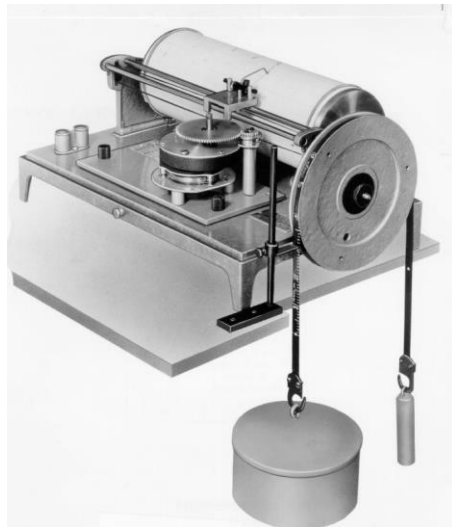


Open hardware solutions for environmental sensing

Professor Wouter Buytaert
Imperial College London

State of the art

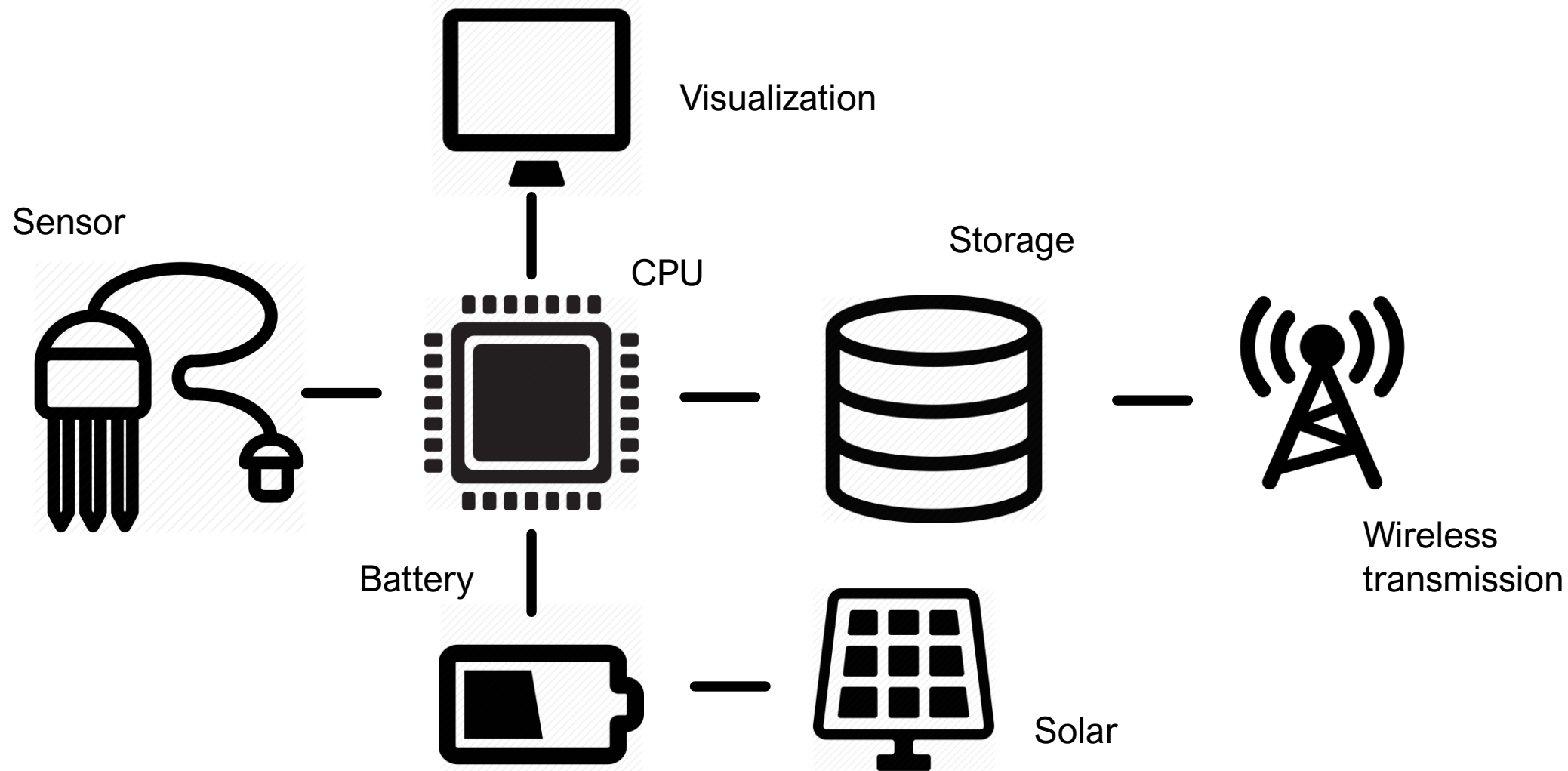


1900



1980

State of the art

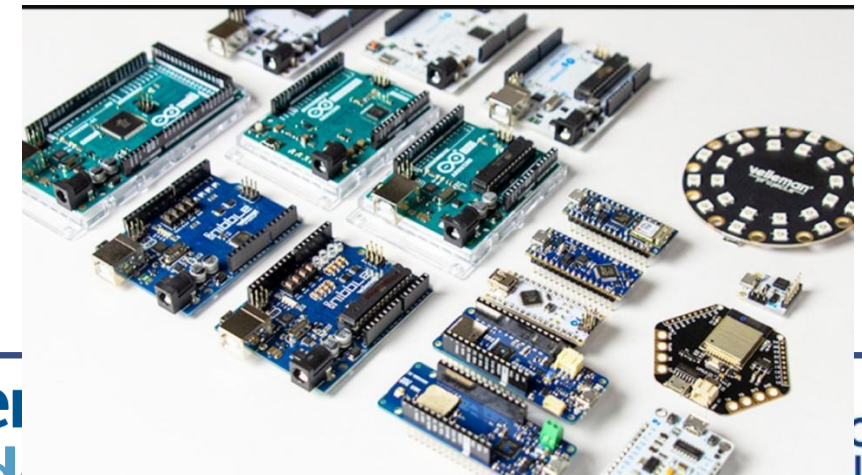
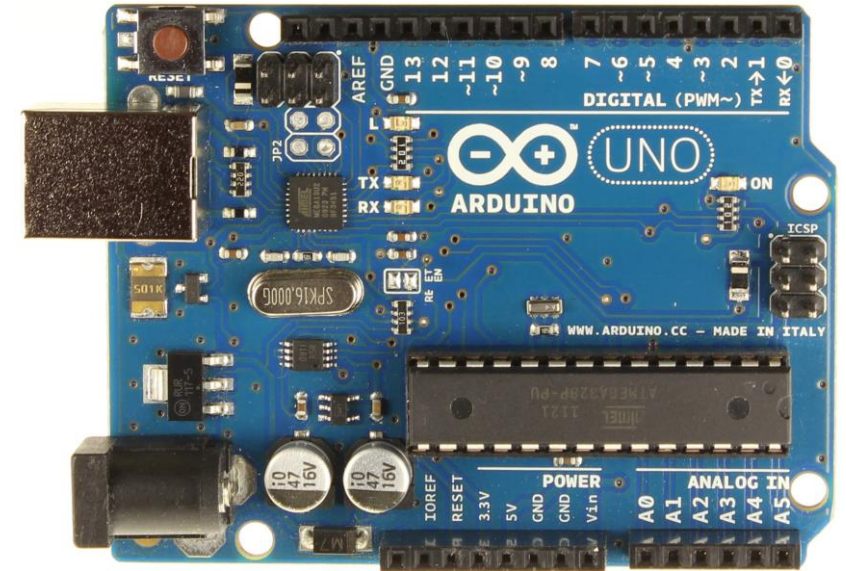


The Arduino open-hardware “ecosystem”

“Open source hardware”

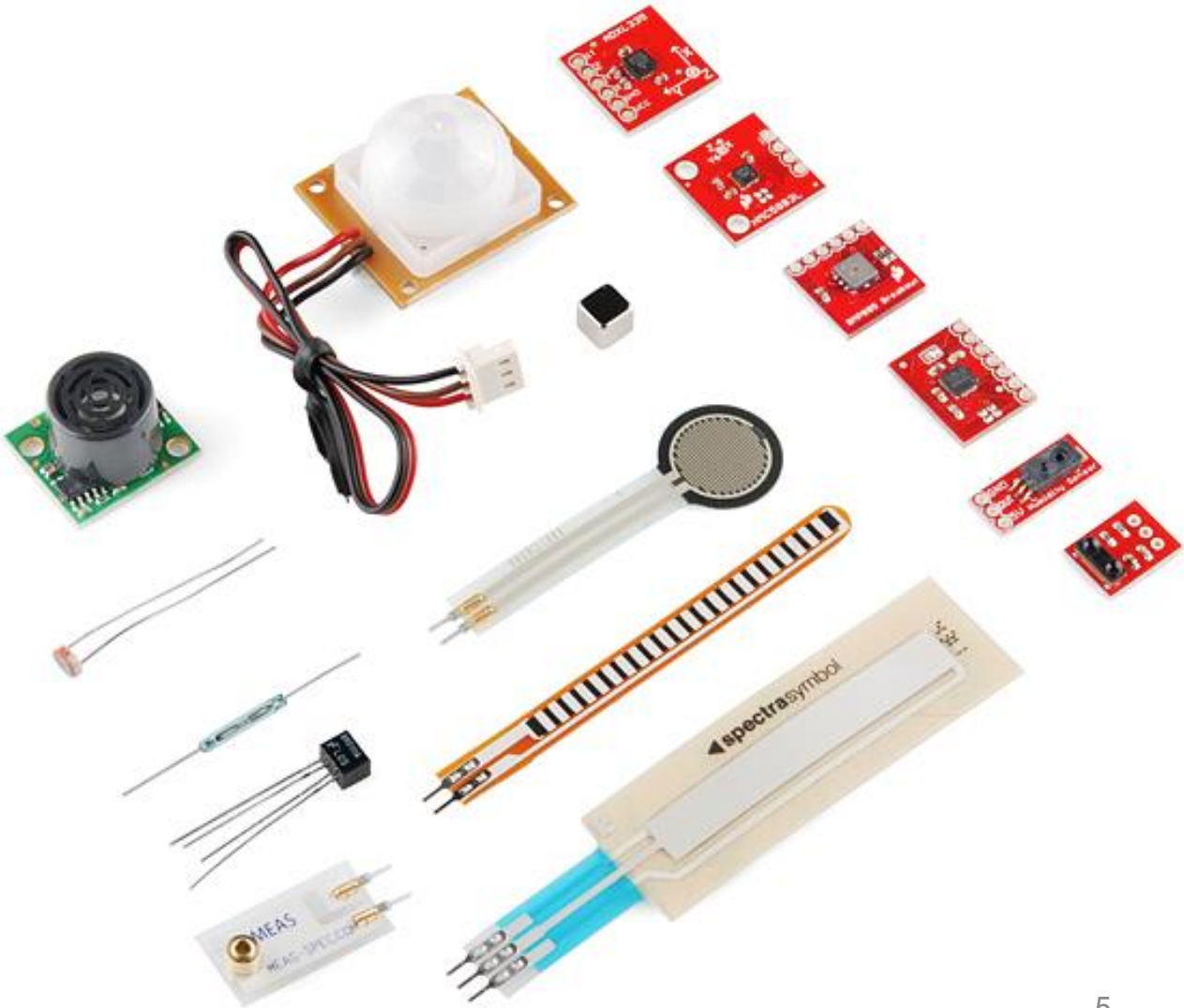
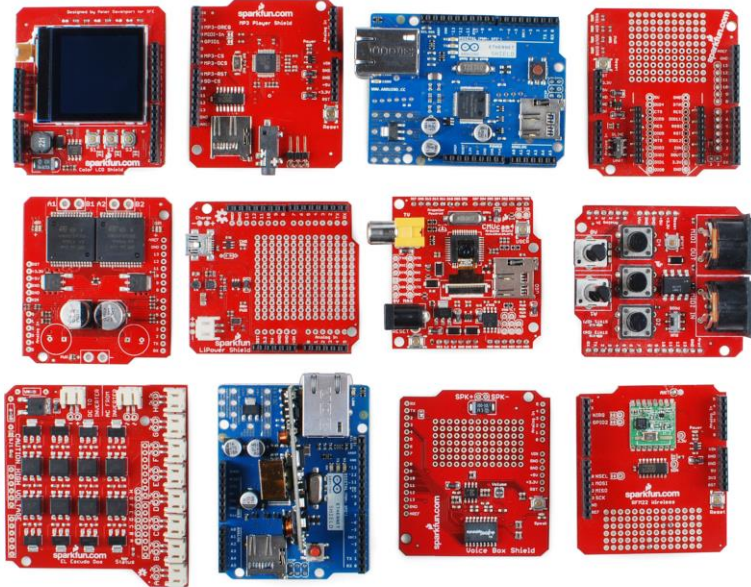
A set of hardware components and tools to facilitate the development of embedded computing systems, consisting of:

- “boards” with a microprocessor
- “shields” to add specific functionality (sensors, storage, data transmission, ...)
- An open source software development application (Arduino IDE)
- Software libraries
- Designs, tutorials, forums, and other web resources with information

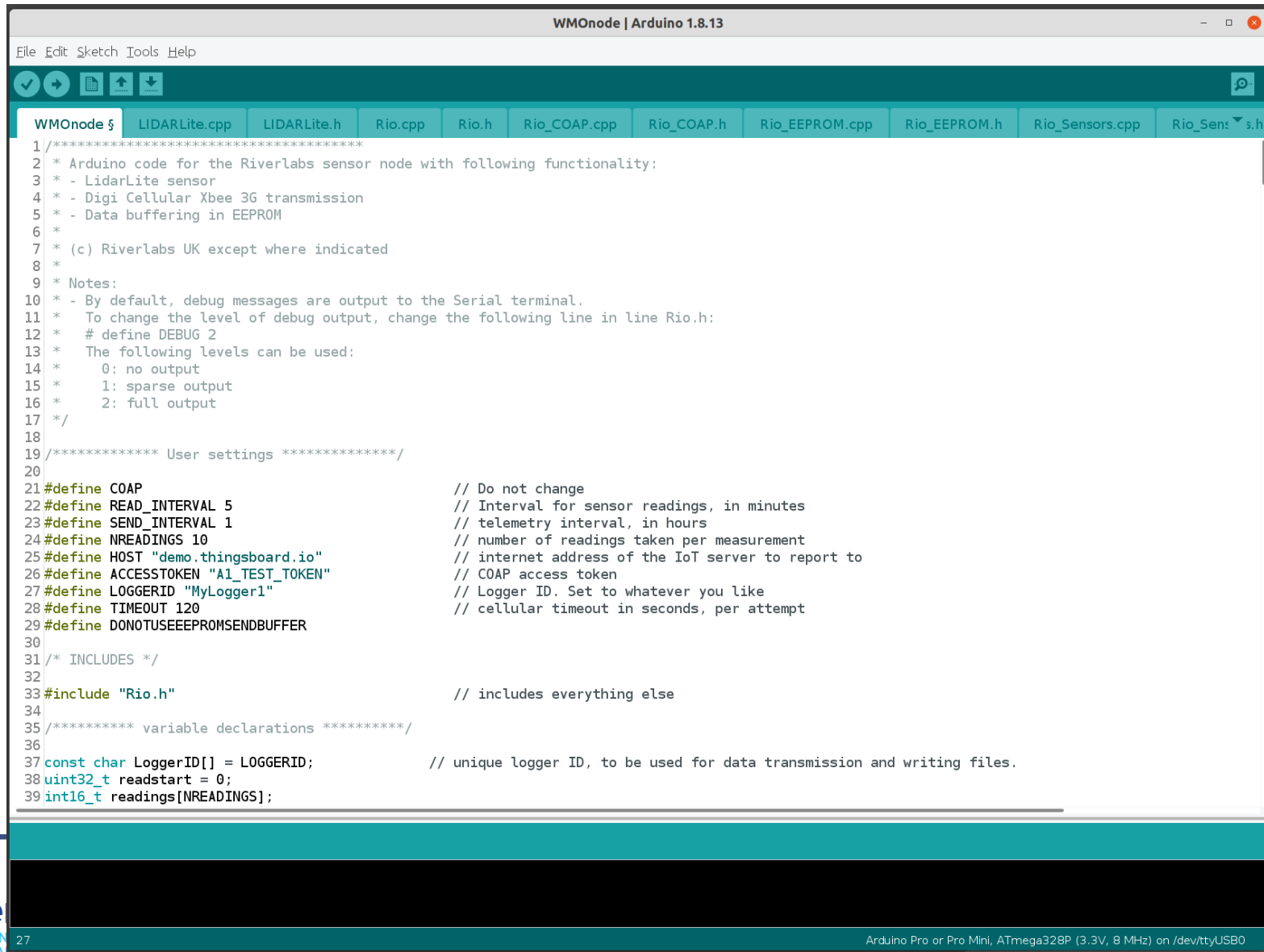


<https://www.arduino.cc/en/Guide/>

A vast toolbox of “shields”, sensors, and other components



The Arduino IDE



```
WModem | Arduino 1.8.13
File Edit Sketch Tools Help
WModem $ LIDARLite.cpp LIDARLite.h Rio.cpp Rio.h Rio_COAP.cpp Rio_COAP.h Rio_EEPROM.cpp Rio_EEPROM.h Rio_Sensors.cpp Rio_Sens s.h
1/*****
2 * Arduino code for the Riverlabs sensor node with following functionality:
3 * - LidarLite sensor
4 * - Digi Cellular Xbee 3G transmission
5 * - Data buffering in EEPROM
6 *
7 * (c) Riverlabs UK except where indicated
8 *
9 * Notes:
10 * - By default, debug messages are output to the Serial terminal.
11 * To change the level of debug output, change the following line in line Rio.h:
12 * # define DEBUG 2
13 * The following levels can be used:
14 * 0: no output
15 * 1: sparse output
16 * 2: full output
17 */
18
19/***** User settings *****/
20
21#define COAP // Do not change
22#define READ_INTERVAL 5 // Interval for sensor readings, in minutes
23#define SEND_INTERVAL 1 // telemetry interval, in hours
24#define NREADINGS 10 // number of readings taken per measurement
25#define HOST "demo.thingsboard.io" // internet address of the IoT server to report to
26#define ACCESS_TOKEN "A1_TEST_TOKEN" // COAP access token
27#define LOGGER_ID "MyLogger1" // Logger ID. Set to whatever you like
28#define TIMEOUT 120 // cellular timeout in seconds, per attempt
29#define DONT_USE_EEPROM_SENDBUFFER
30
31/* INCLUDES */
32
33#include "Rio.h" // includes everything else
34
35/***** variable declarations *****/
36
37const char LoggerID[] = LOGGER_ID; // unique logger ID, to be used for data transmission and writing files.
38uint32_t readstart = 0;
39int16_t readings[NREADINGS];
27
Hydrological Programme
Arduino Pro or Pro Mini, ATmega328P (3.3V, 8 MHz) on /dev/ttyUSB0
```

<https://www.arduino.cc/en/software>

Integration with open source logging systems

Arduino libraries:

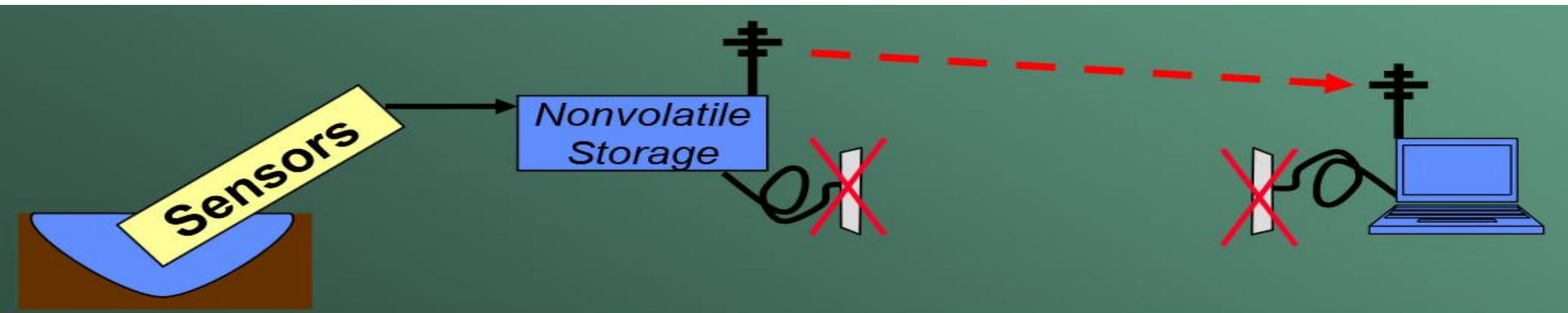
- Xbee-Arduino: <https://github.com/andrewrapp/xbee-arduino>
- Xbee-Arduino (3G compatible): <https://github.com/ICHydro/xbee-arduino>
- LoRa: <https://www.airspayce.com/mikem/arduino/RadioHead/>
- Arduino BLE library: <https://www.arduino.cc/en/Reference/ArduinoBLE>

Other open libraries and toolboxes

- Particle.io: <https://www.particle.io/>
- Nordic nRF5 SDK: www.nordicsemi.com/Software-and-tools/Software/nRF5-SDK-for-Thread-and-Zigbee

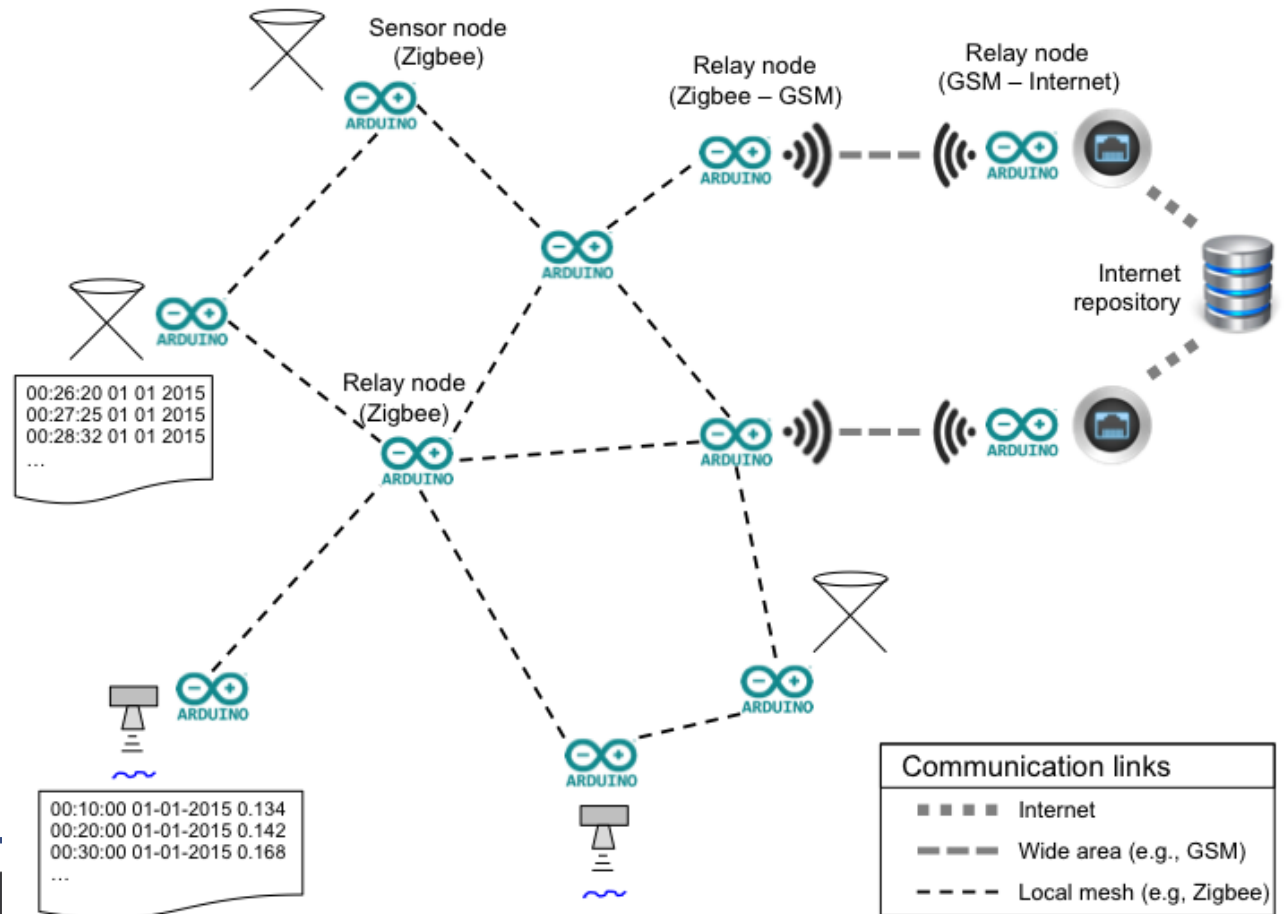
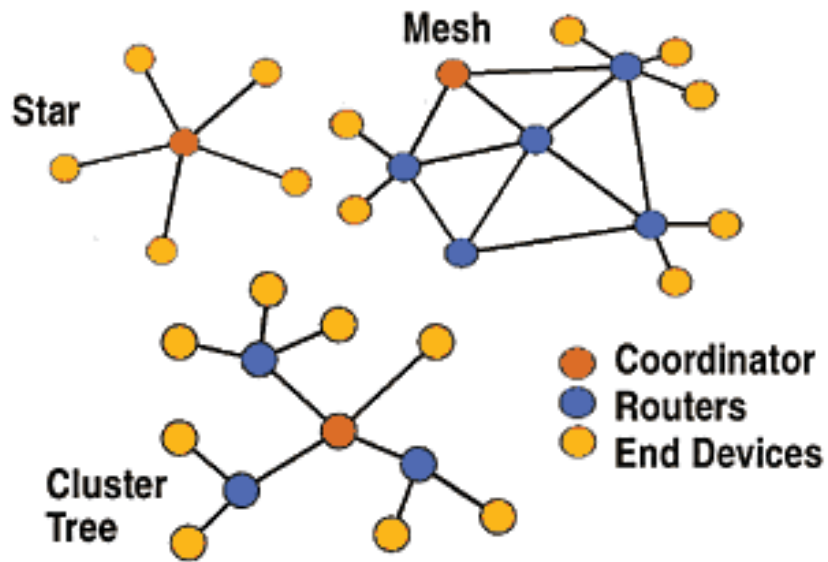
Networking and telemetry

- Automatic data transmission
- Warning systems
- “Smart” sensor networks
- ...



Network characteristics

Topology: Local vs wide networks



Source: fierceelectronics.com

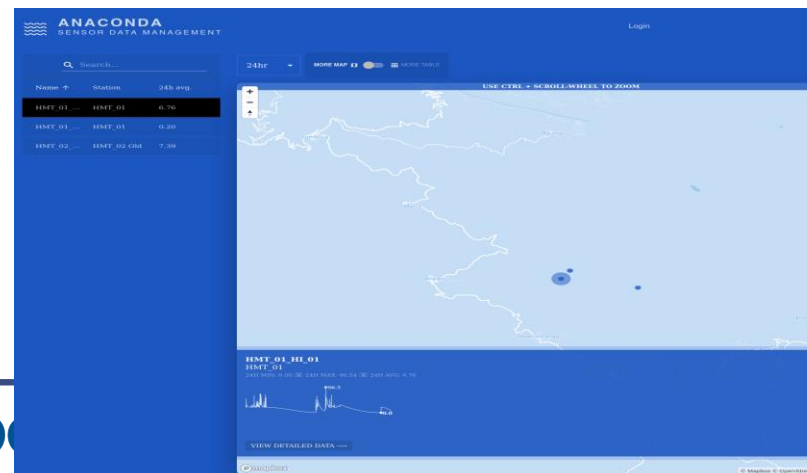
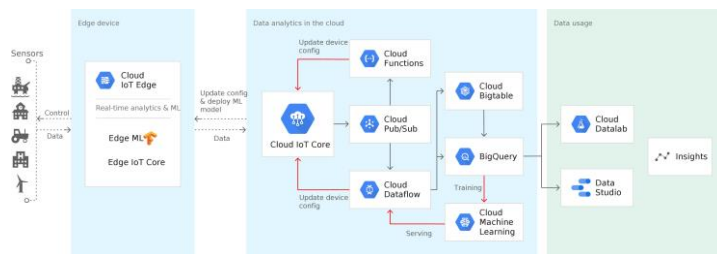
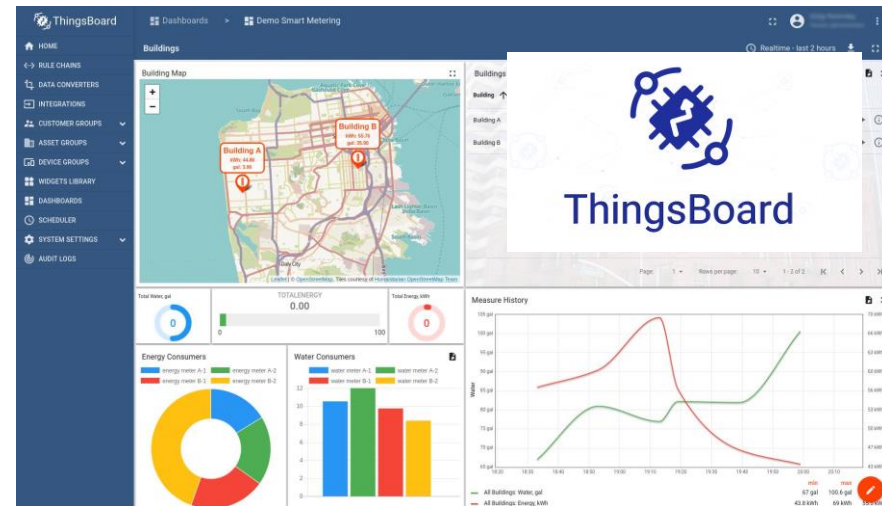
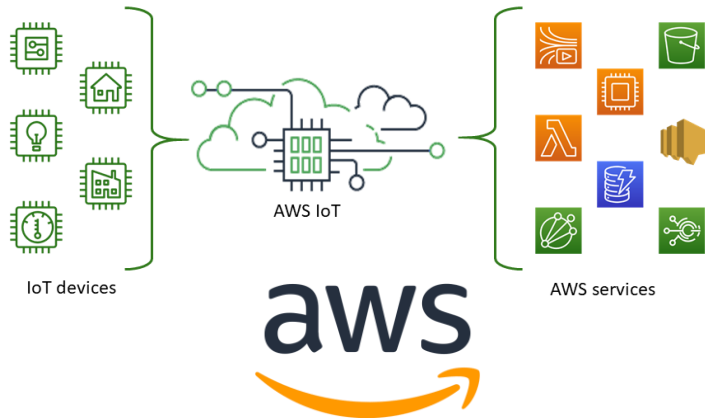
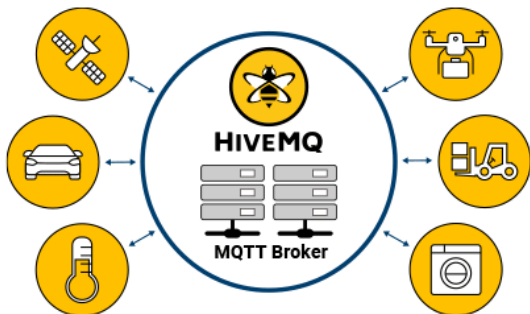
Data transmission

A variety of transmission protocols exist. Some networking technologies (e.g. LoRa) have their own transmission protocols (e.g. LoRaWAN), while others (e.g., cellular) can be used with several different protocols. Some of the most common transmission protocols include:

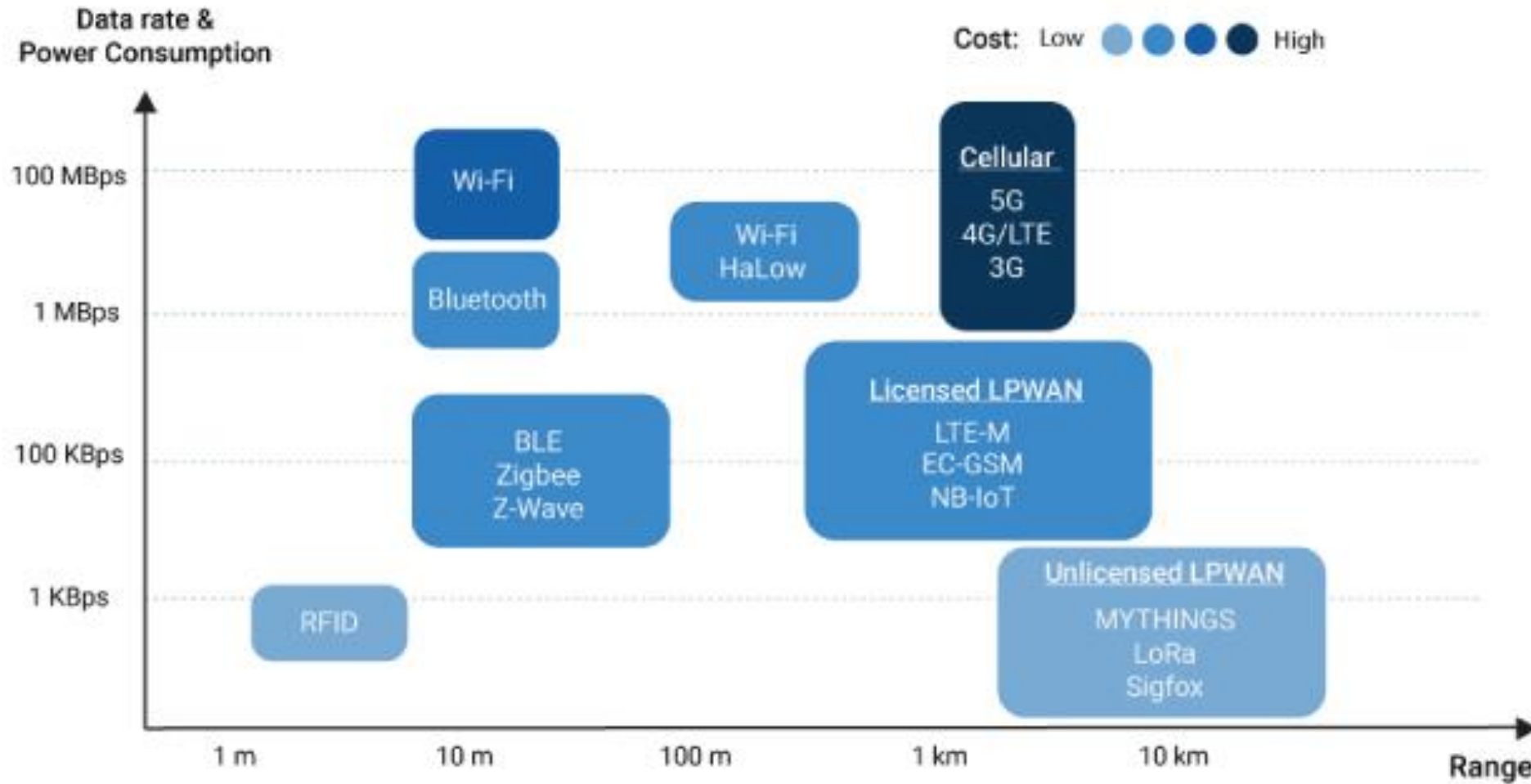
- LoRaWAN
- Thread / 6LoWPAN
- Zigbee
- BLE
- SMS
- Internet protocols:
 - HTTP REST
 - MQTT
 - CoAP



Data management systems and solutions



Wireless networking technologies



Source: behrtech.com

Wireless networking technologies

1. LPWANs

- Licenced spectrum: LTE-M, NB-IoT, ...
- Unlicenced spectrum: LoRa, Sigfox, ...
- Mostly star-like networks
- Commonly used in IoT applications
- Deployment still ongoing

2. Cellular (3G, 4G, 5G)

- Readily available
- Star network
- High power consumption
- High data transmission capacity

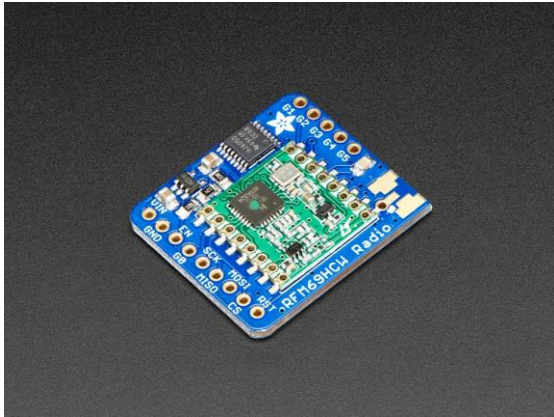
3. Zigbee and other mesh protocols

- Zigbee, Z-Wave, Thread, ...
- short-range, low power
- Self-organising network

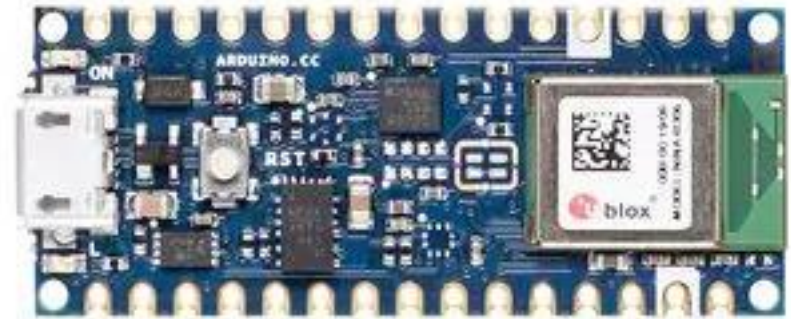
4. Bluetooth and Bluetooth Low Energy (BLE)

- Consumer IoT
- Bluetooth Mesh specification (2017)
- Low power, short distance

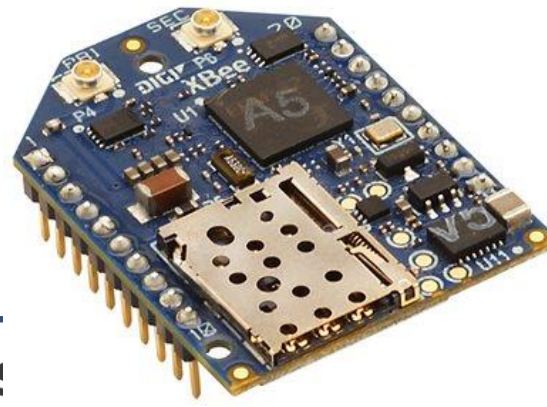
Wireless networking technologies



Source: Adafruit.com



Source: Arduino.cc



Art

London

age



International Centre
Source: digi.com