Smart Objects

Giuseppe Anastasi

Rector's Delegate for the Digital Transition

Emeritus Director, CrossLab for the Digital Transformation

Dept. of Information Engineering

University of Pisa

CROSSLAB
artefici della trasformazione digitale

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

INTERNET of THINGS

Smart Object

Real-world Object

Instrumenting Device

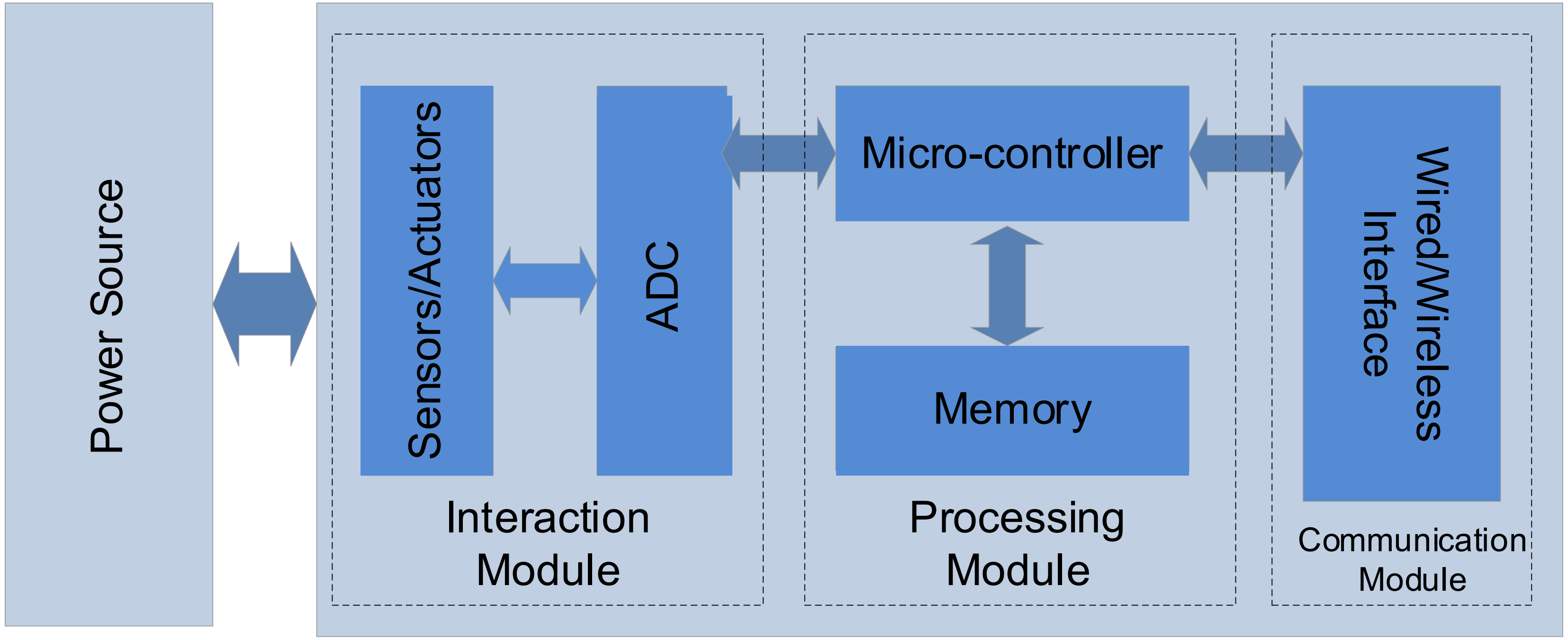
Low-cost device embedded to the object

Communication capabilities

Computing capabilities

Sensing/Actuating capabilities

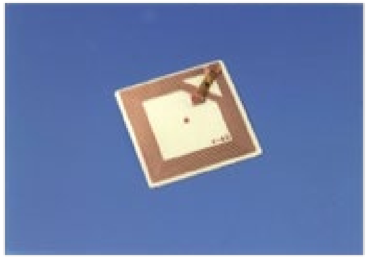
Instrumenting Device



Instrumenting Device

RFID Taggs

- Only connectivity
- Reader required for communication



CROSSLAB
artefici della trasformazione digitale

Instrumenting Device

RFID Taggs

- Only connectivity
- Reader required for communication

Sensor Actuator Node

- Sensing
- Processing
- Communication
- Actuation

Sensors: Classification

Passive Sensors	RFID
Semi-passive Sensors	Data loggers
Active Sensors	Beacons Sensor nodes

Sensor: Classification

- Temperature
- Humidity
- Light intensity
- Infrared (Presence)
- Sound (Noise)
- Accelerometer
- Compass (direction)
- Speed
- Pressure
- Seismic

Chemical Sensors

Optical Sensors

Pollution

- CO, CO_2 , NO_2 , O_3, Benzene
- PM10, PM2.5, PM1

Smart Meters

- Power/Energy Consumption
- Gas Consumption
- Water Consumption

…

Passive Tags

QR codes

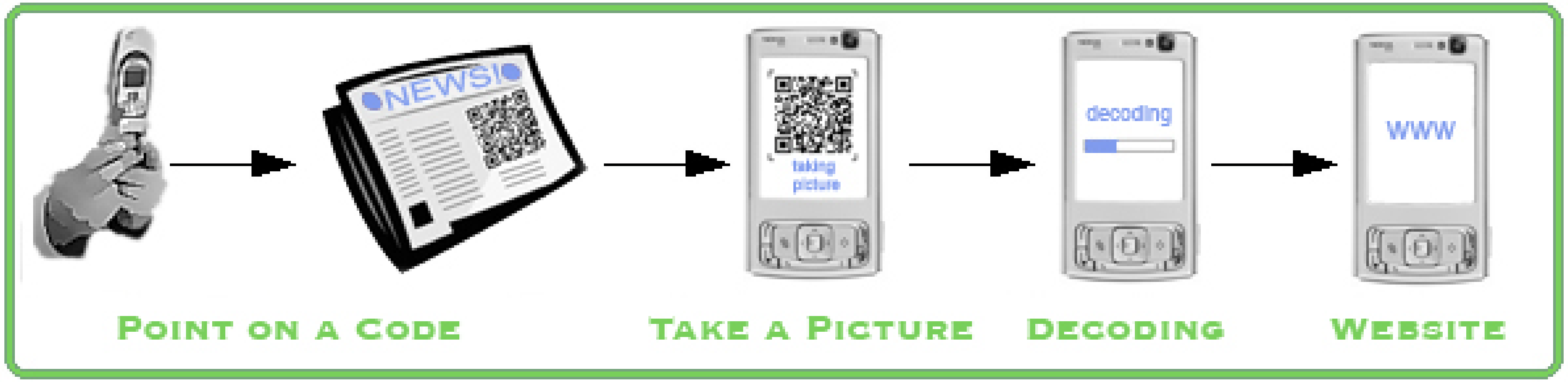
RFID

RFID

QR codes

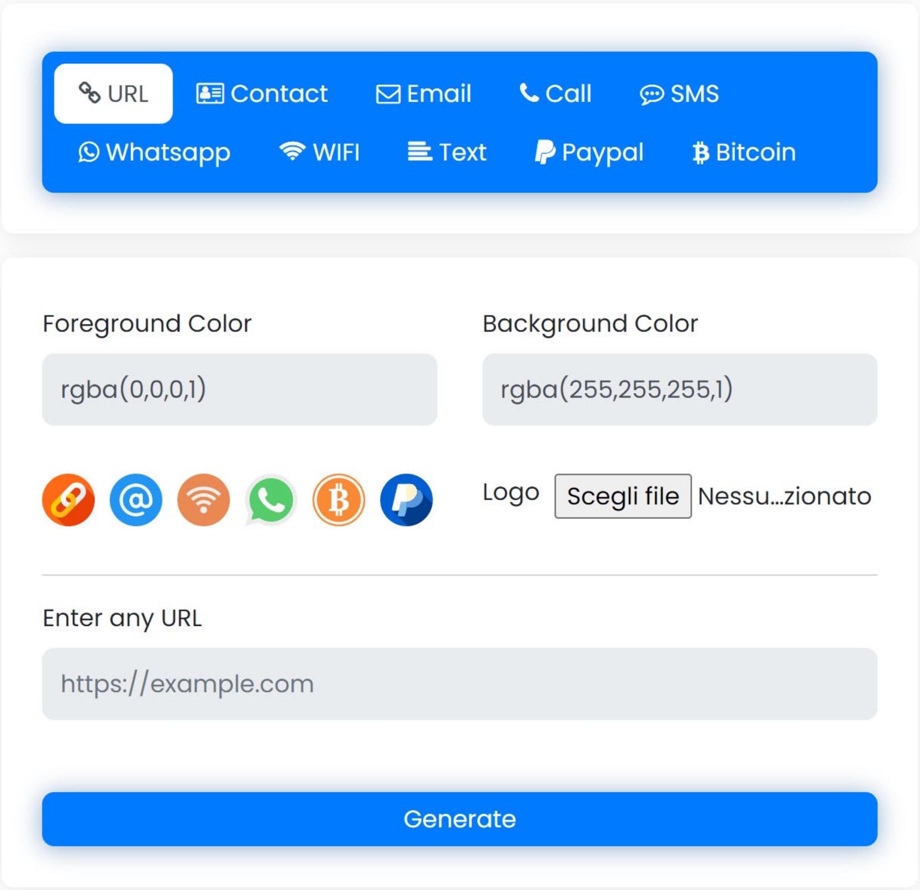
Quick Response (QR) Code

Two dimensional barcode that stores information in black and white dots (data pixels or “QR code modules”)



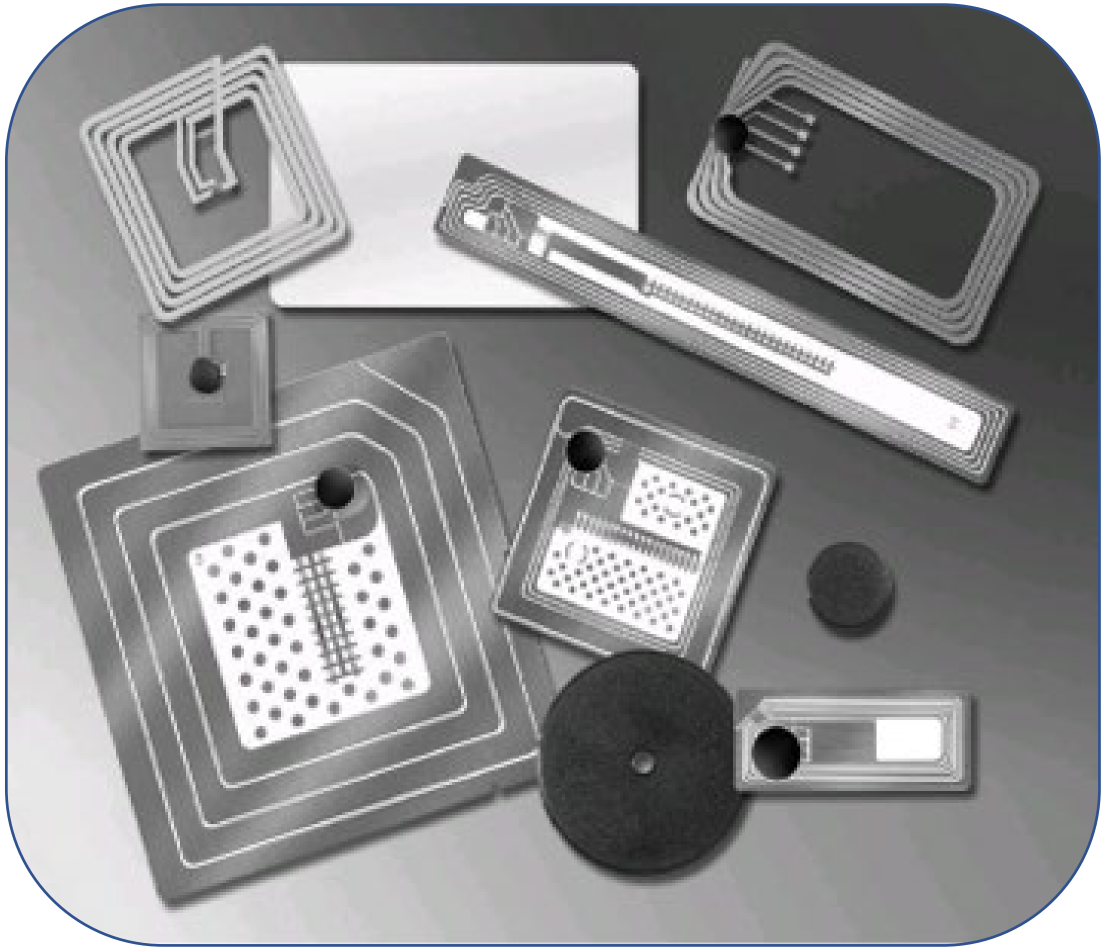
QR code generator

Many online generators available

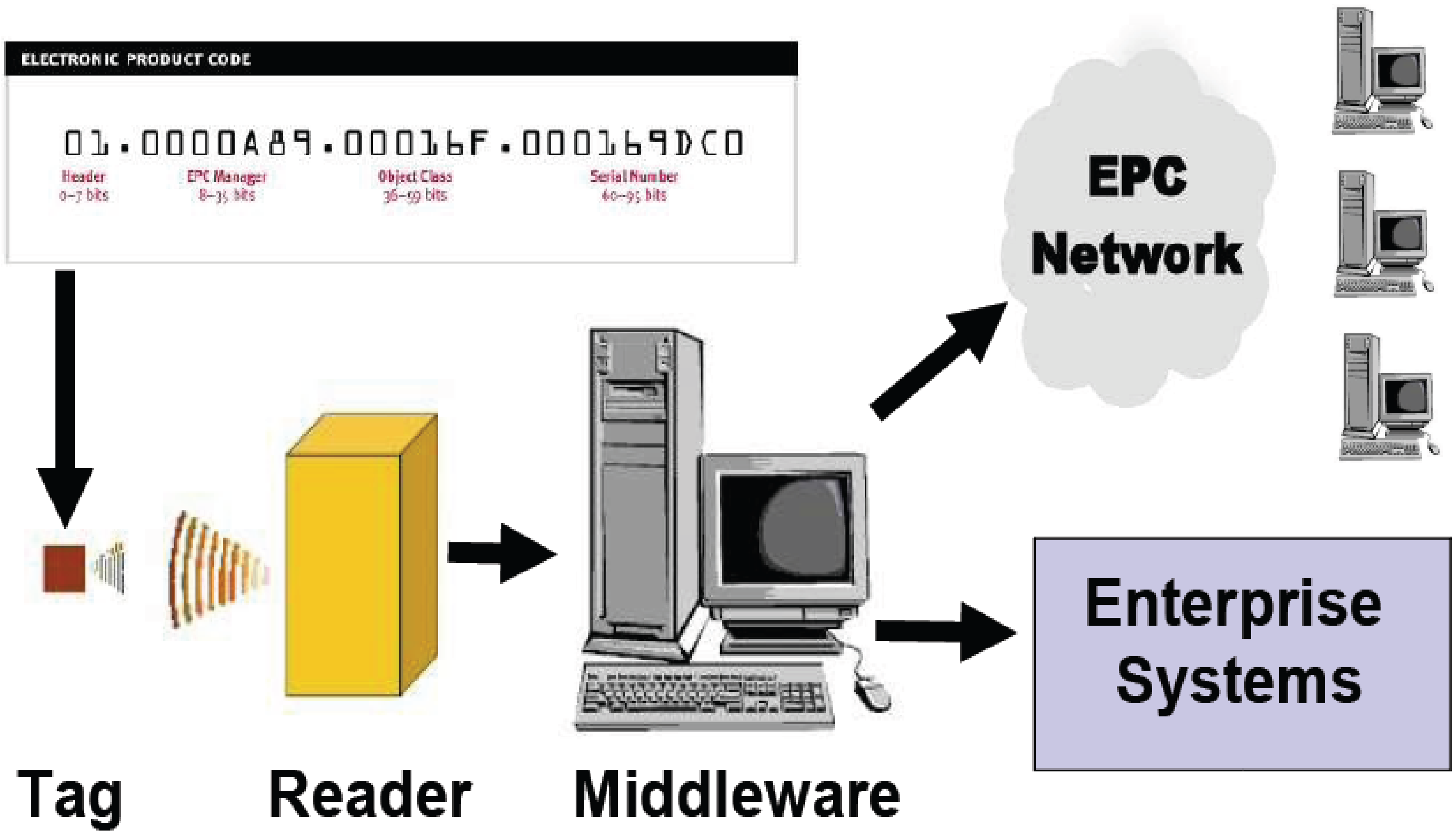


https://qurcode.com/

RFID



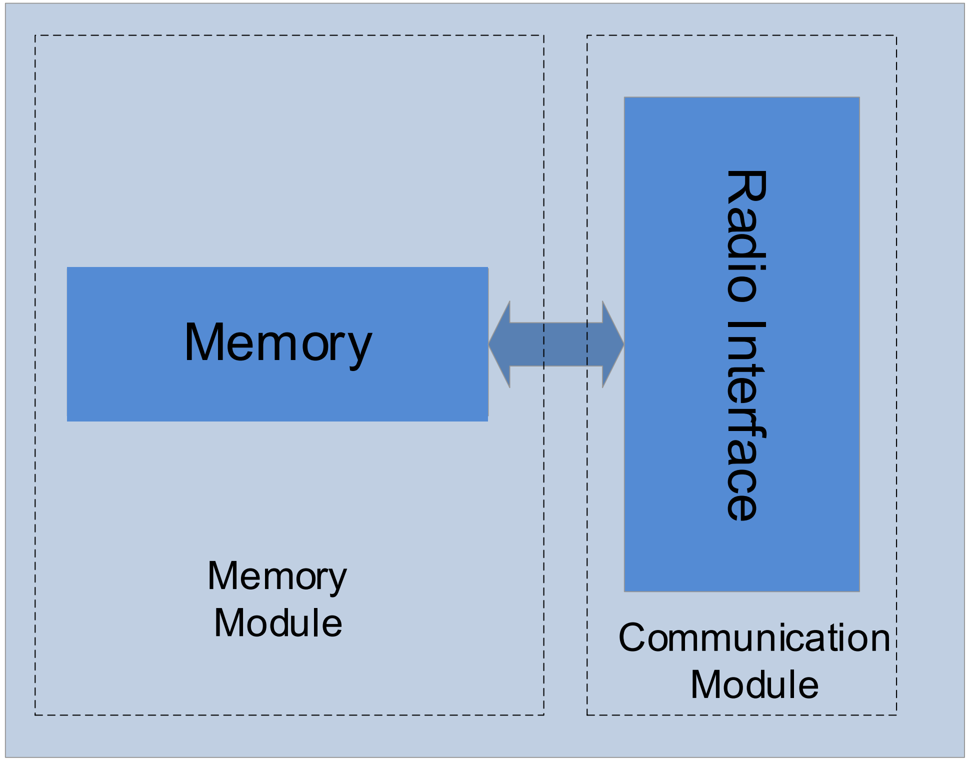
Using RFID


ELECTRONIC PRODUCT CODE
01.0000A89.00016F.000169DC0
Header
0–7 bits
EPC Manager
8–35 bits
Object Class
36–59 bits
Serial Number
60–95 bits
EPC
Network
Enterprise
Systems
Tag
Reader
Middleware

RFID Architecture

RFID Tags: only connectivity

RFID Reader required for communication



RFID Application Areas

- Supply chain management
- Baggage/Animal tagging
- Automated Access/Tolling
- Library information systems
- Parcel tracking (postal services)
- Smart cities
  - Access control
  - Ticketing (public transport, parking, etc.)
  - Parking area management
  - Augmented reality
  - ...

Supply Chain Management

RFID for Animal Tagging

RFID is used to track the movement of animals

RFID consists of an animal tag with a microchip and a reader that can scan the tag

RFID tags are attached to the animal's ear or skin and active or passive depending on the type of application

RFID technology is widely used in the livestock industry to track animals' location, health, and productivity

RFID technology is also used to track and monitor wildlife animals to gather data for conservation efforts

Automatic Access/Tolling

RFID technology enables automatic toll collection and access control

RFID tags attached to a vehicle or an individual can be scanned for automated access

RFID systems improve efficiency, reduce congestion and increase security

Toll collection and access control are faster, more accurate, and require less manpower

RFID technology is widely used in transportation systems, parking facilities and gated communities

Augumented Reality in a Smart City

Points of interest in the city (touristic point of interest, shops and public places) can be tagged through RFIDs or QR codes

Information about the tagged point are made available to the user (tourist, citizen, …)

Active Tags: Beacons

Active Tags: Beacons

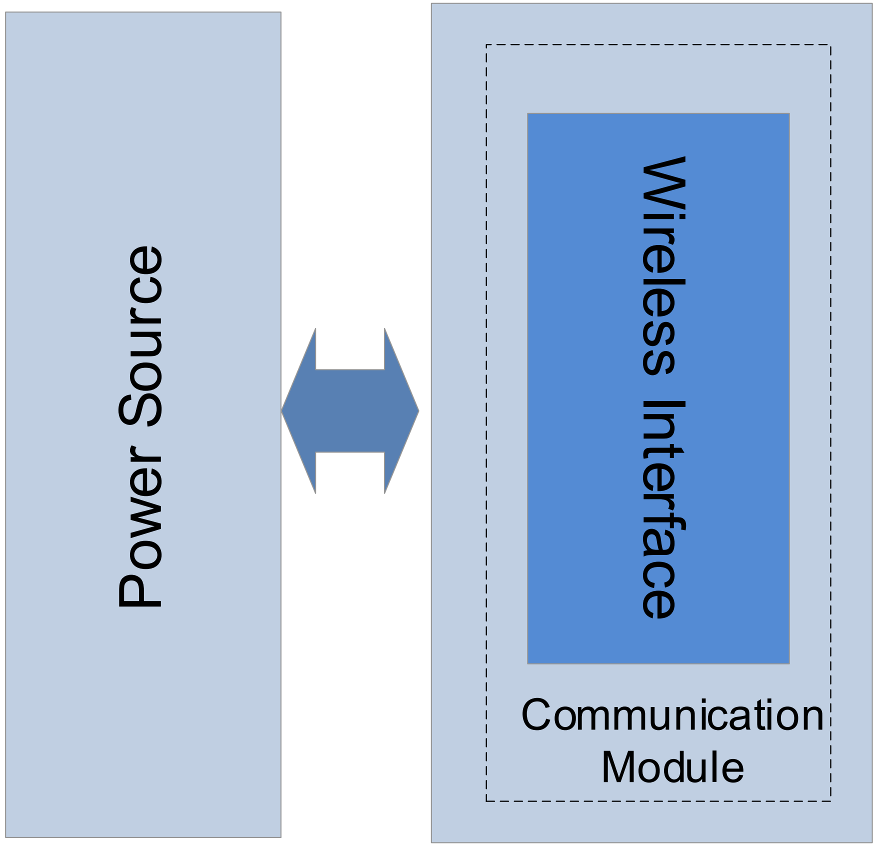
Beacons are hardware transmitters that emit periodic signals using *Bluetooth Low Energy* (BLE)

Beacons enable devices to perform actions when in proximity based on the received signal

Applications include localization in buildings, retails, tracking, indoor navigation, and smart tagging




Beacon Architecture

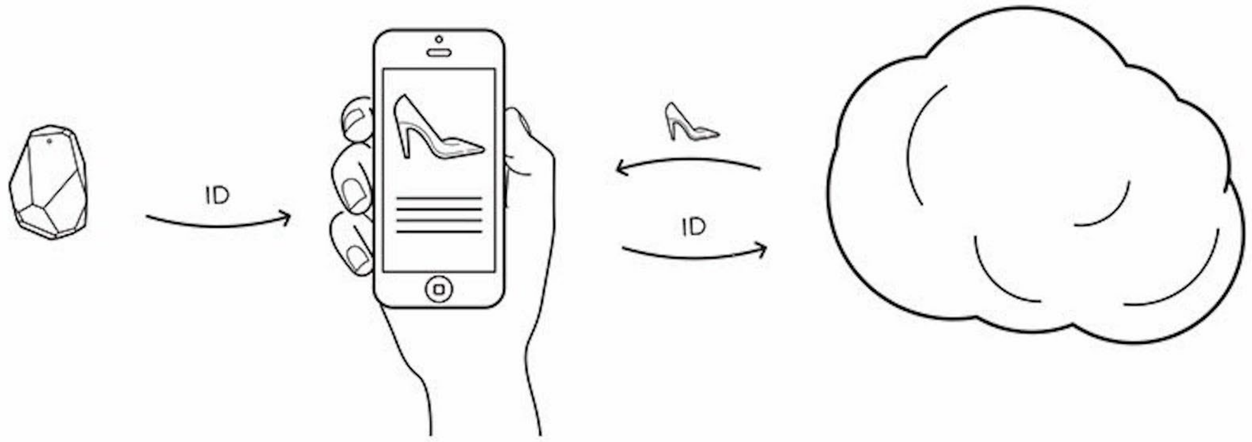


Basic Idea

Beacon uses Bluetooth Low Energy (BLE) to transmit Advertisement (ADV) messages

The ADV message is received by a smartphone and forwarded to a server (e.g., on the cloud)

The server localizes the device (user) and sends appropriate information to the smartphone



Bluetooth Low Energy (BLE)

Main goal

- providing considerably reduced power consumption and cost
- while maintaining a similar communication range

Power Consumption

- BLE has significantly lower power consumption that traditional Bluetooth
- Lifetime up to 3 years with a single coin cell battery

Cost

- BLE is 60-80% cheaper than traditional Bluetooth

Applications

- BLE is ideal for simple applications requiring periodic transfers of small data
- Traditional Bluetooth is better for more complex applications requiring consistent communication and higher throughput

How Beacons work

BLE communication mainly consists in **Advertisement (ADV) messages**

- small data packets transmitted at a regular interval by enabled devices via radio waves
- **one-way** communication

Typical parameter values for Beacons

- Broadcast interval of 100 ms
- Broadcast range of up to 100 meters

Beacon Advertisement Message

ADV messages consist of four main pieces of information

- UUID (Universal Unique IDentifier)
- Major
- Minor
- Tx Power

Beacon Advertisement Message

UUID (Universal Unique IDentifier)

- 16 byte string used to differentiate a large group of related Beacons
- In its canonical form, a UUID is represented by 32 lowercase hexadecimal digits, displayed in five groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters
- E.g., ebefd083-70a2-47c8-9837-e7b5634df524

Major

- 2-byte string distinguishing a subset of Beacons within the larger group

Minor

- 2-byte string used to identify individual Beacons

Tx Power

- Used to determine proximity (distance) from the beacon
- TX power is defined as the strength of the signal at exactly 1 meter from the device.
- This has to be calibrated and hardcoded in advance. Devices can then use this as a baseline to give a rough distance estimate.

Example

Application for locating people within the University of Pisa

UUID (Universal Unique IDentifier)

- Allows to recognize that ADV messages are emitted from Beacons belonging to the University of Pisa

Major

- Identifies ADV messages emitted from Beacons inside a specific building belonging to the University of Pisa (e.g., Building A in the Engineering Campus)

Minor

- Identifies a specific room in that Building (e.g., Room A12)

Example

A beacon broadcasts the following ADV message

UUID:
ebefd083-70a2-47c8-9837-e7b5634df524

Major: 21

Minor: 17

The system realizes that it was emitted by a Beacon located at

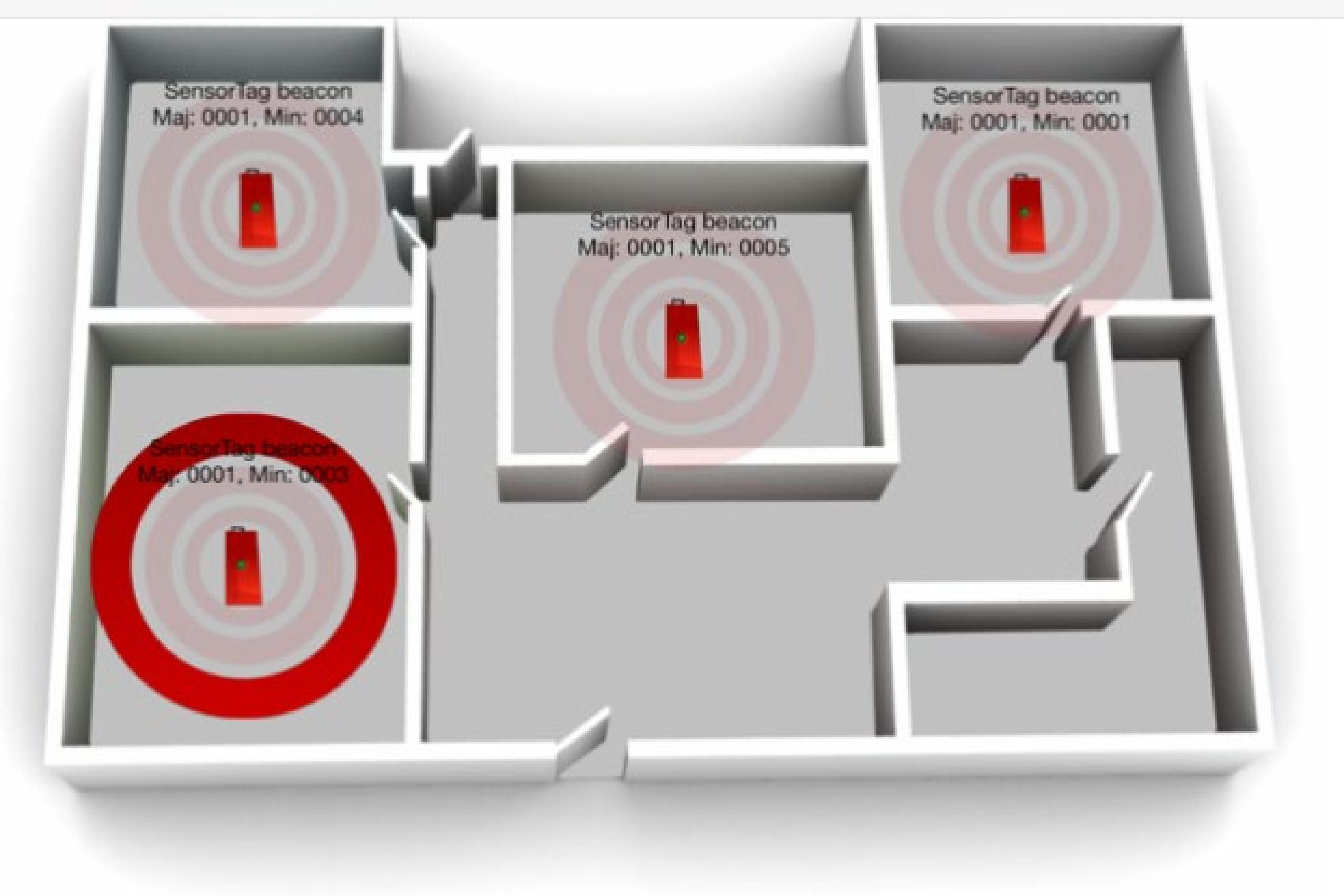
University of Pisa
(UUID)

Building A, Eng Campus
(Major)

Room A28
(Minor)

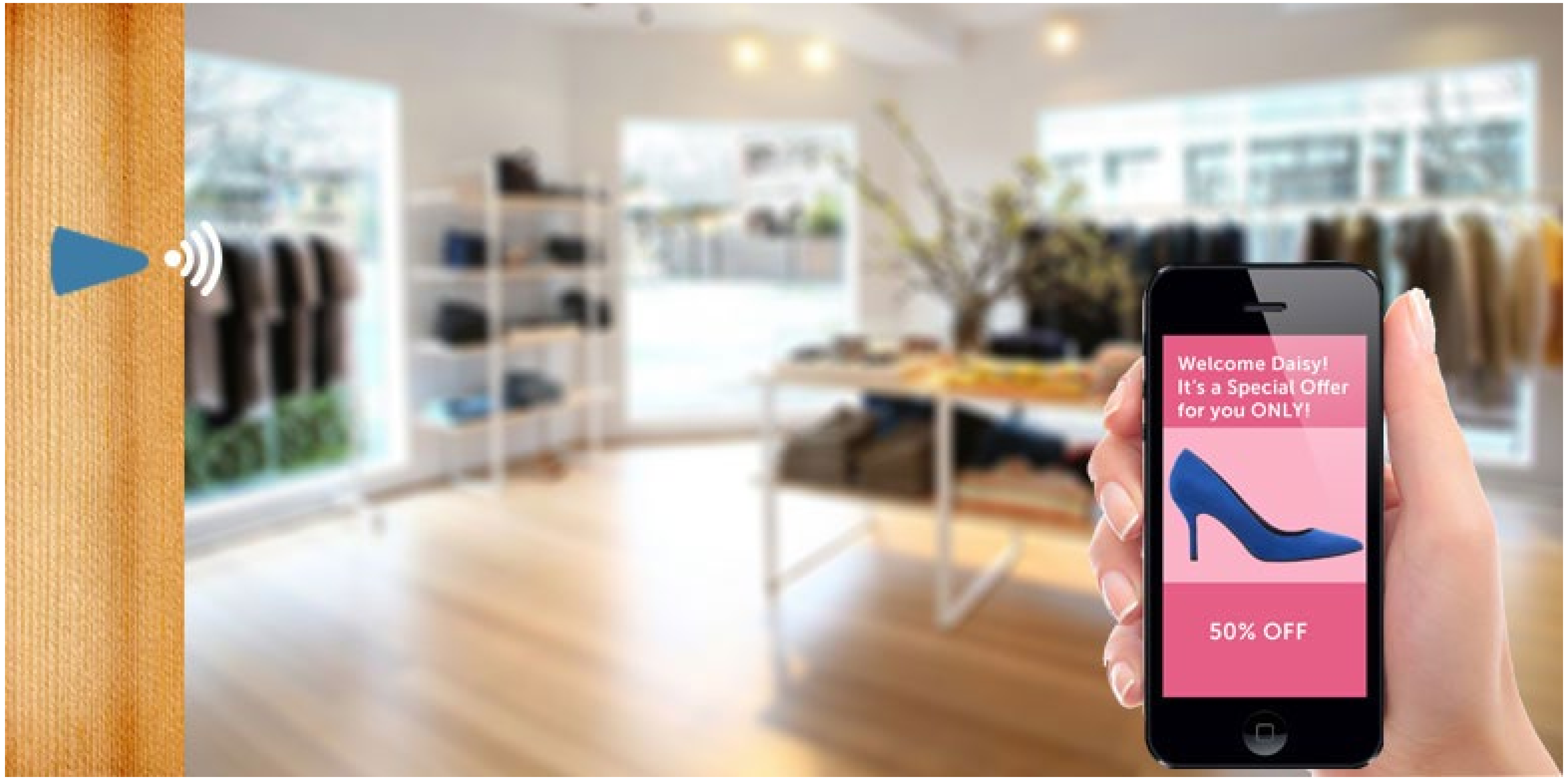
Application Areas

Localization in Buildings



Application Areas

Retails: coupons delivered via Beacons to customer phone

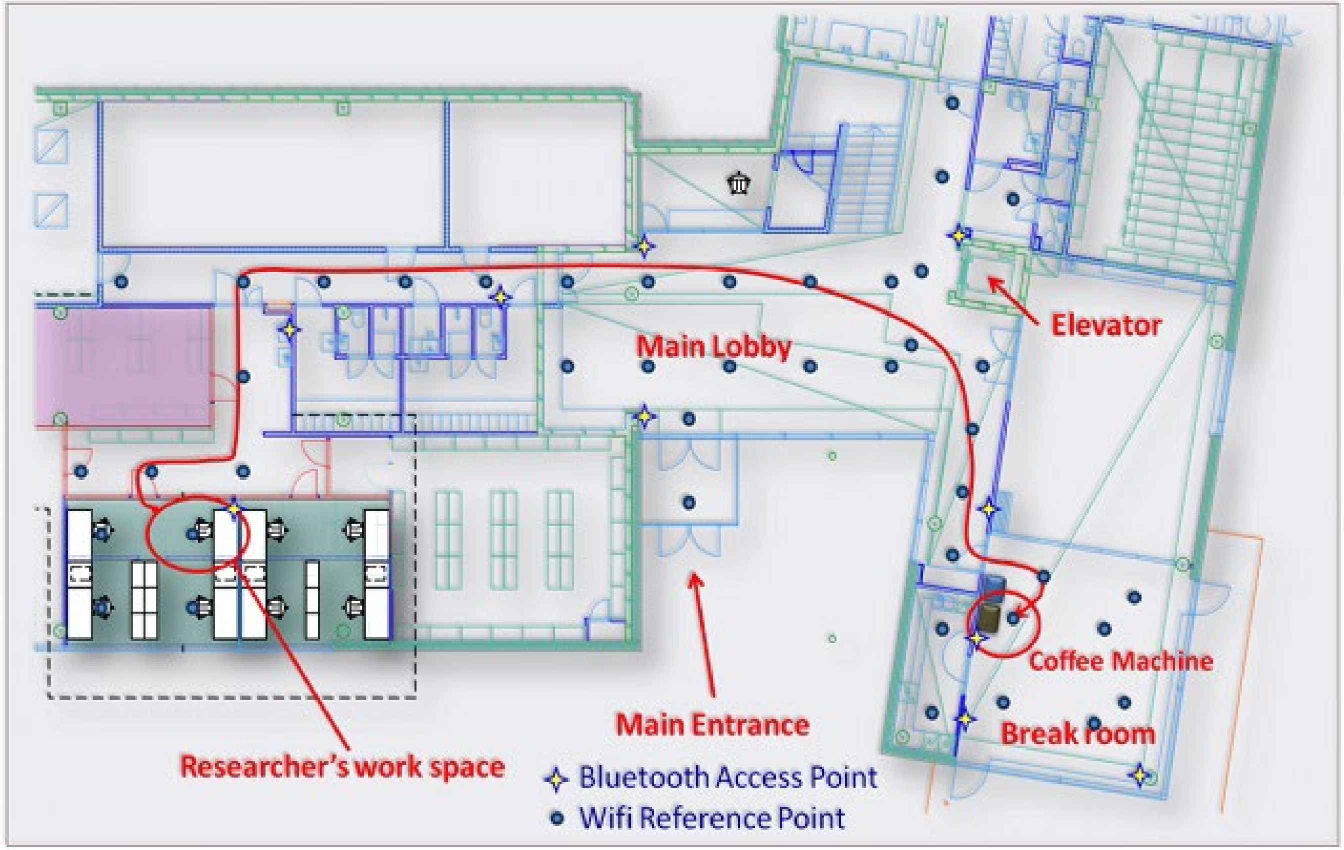


Application Areas

Tracking: locate museum artworks, shop products, or cargo containers

Application Areas

Indoor Navigation: path discover inside a building and user localization



Application Areas

Smart Tagging: allows to give more information about artworks

Semi-passive Sensors

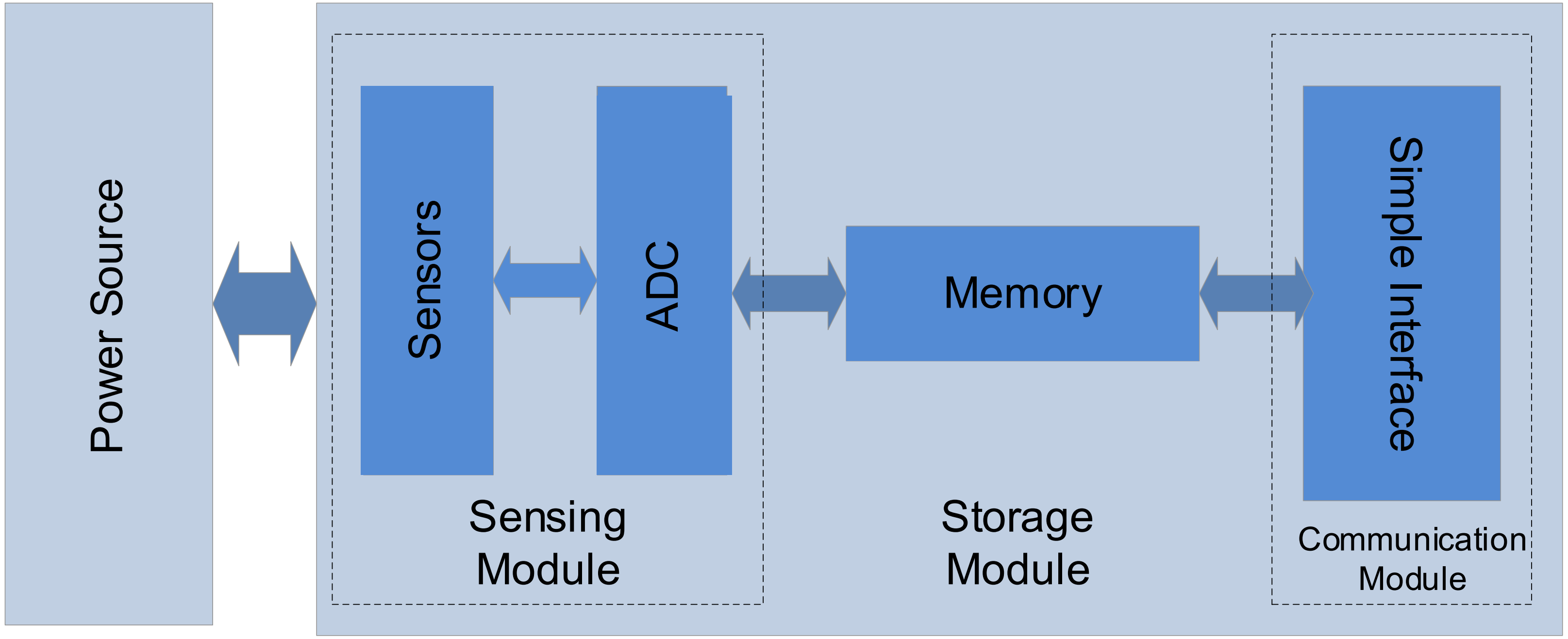
Semi-Passive Sensors

Semi-passive sensors are also called data loggers which are powered by a battery

These sensors are low-cost devices which measure and store physical quantities, such as temperature, humidity, shock, and others

Semi-passive sensors can be attached to goods to track their history

Device Architecture



Application Areas

Ice cream, Frozen Products Distribution

- Temperature monitoring

Wine Transport

- Temperature monitoring
- Shock tracking

Blood Bags Transport

- Bag Identification
- Temperature tracking

…

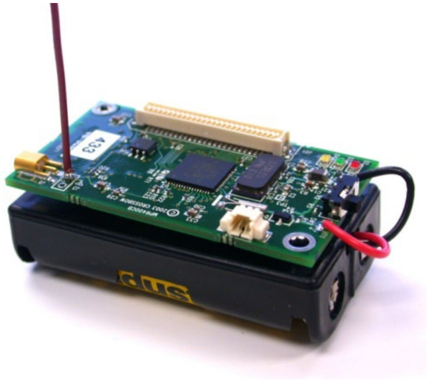
Home work

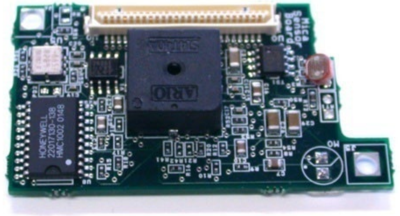
Form small groups

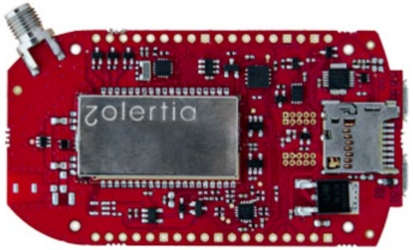
Think about possible simple applications based on Passive/active tags

- Objectives
- Requirements
- Main components
- Links between components

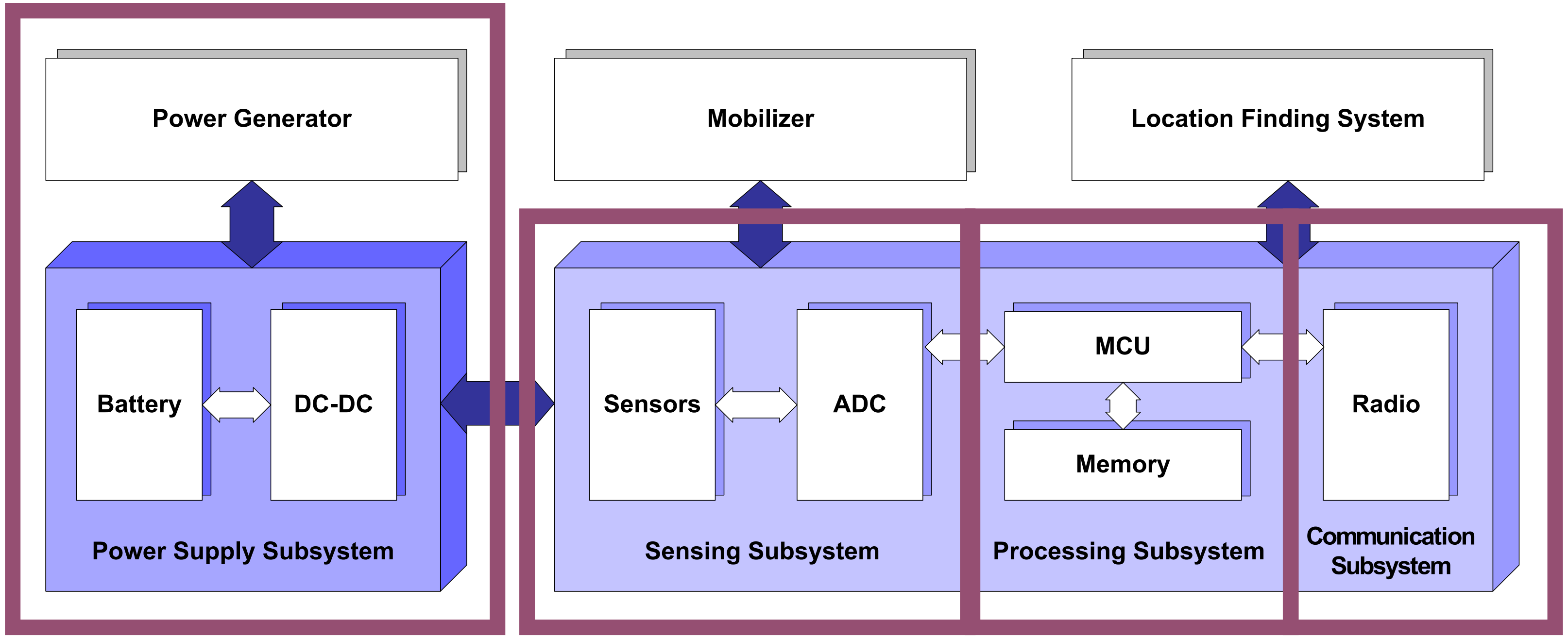
Sensor Nodes







Sensor Node Architecture



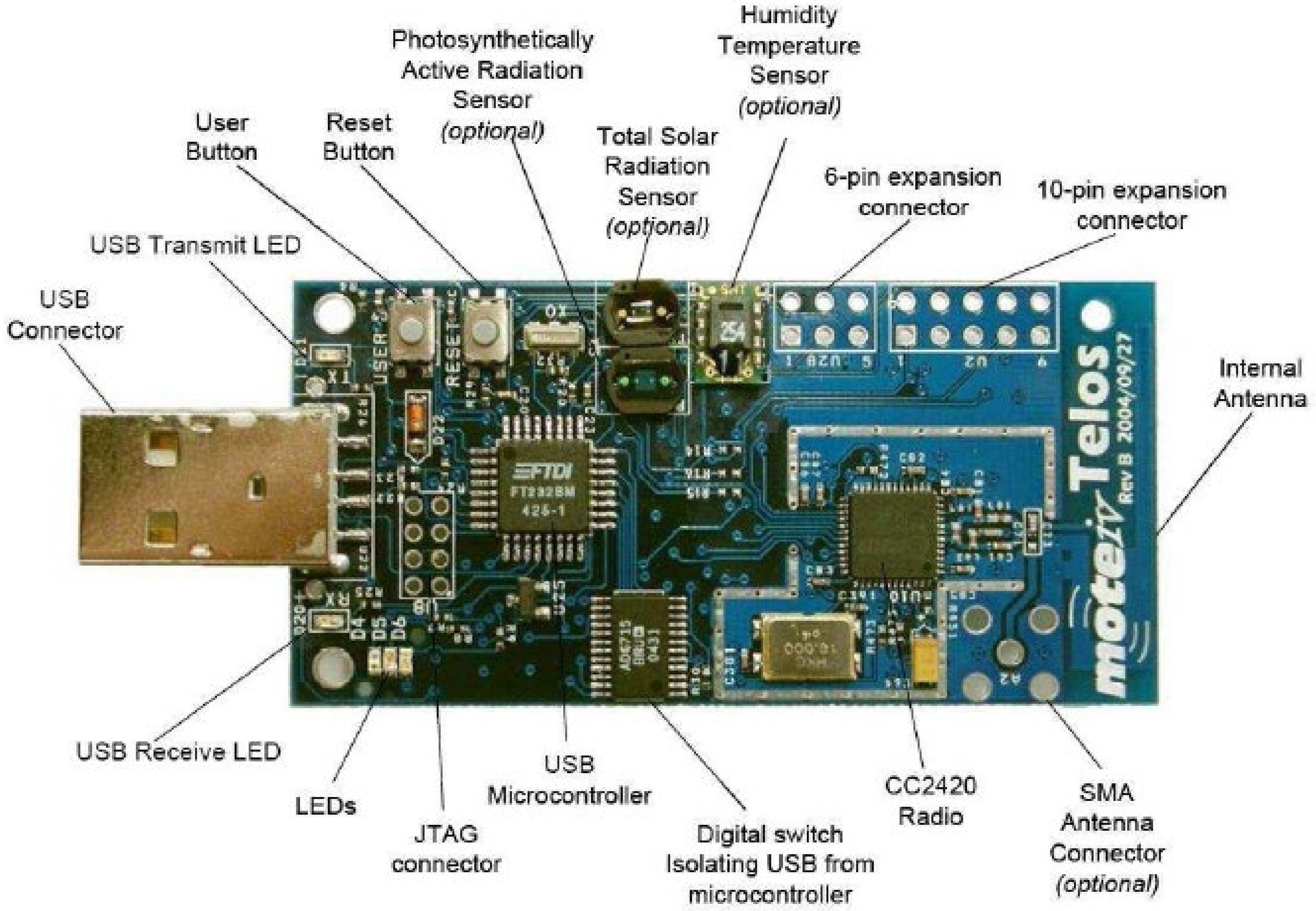
Battery powered devices
Batteries cannot be changed nor recharged

Usually negligible power consumption

Local data processing and data storage

Short range wireless communication
Radio is the most power hungry component

TelosB-Tmote Sky



Zolertia Re-MOTE

Zolertia RE-Mote sensor nodes

- ARM Cortex-M3 SoC
- 512 KB of ROM
- 32 KB of RAM
- CC 2538 radio (2.4 GHz) – IEEE 802.15.4 PHY
- Contiki OS (RIoT, OpenWSN)

Nordic nRF52840 Dongle

Microcontroller

- Arm CortexTM-M4 with floating point support
  - 64 MHz Clock Speed
  - 1 MB Flash
  - 256 KB RAM

Communication

- Bluetooth 5
- Bluetooth mesh
- IEEE 802.15.4
  - Thread
  - ZigBee
- 2.4 GHz proprietary protocols

Security

- Arm CryptoCell CC310 cryptographic accelerator

Launchpad (Texas instrument)

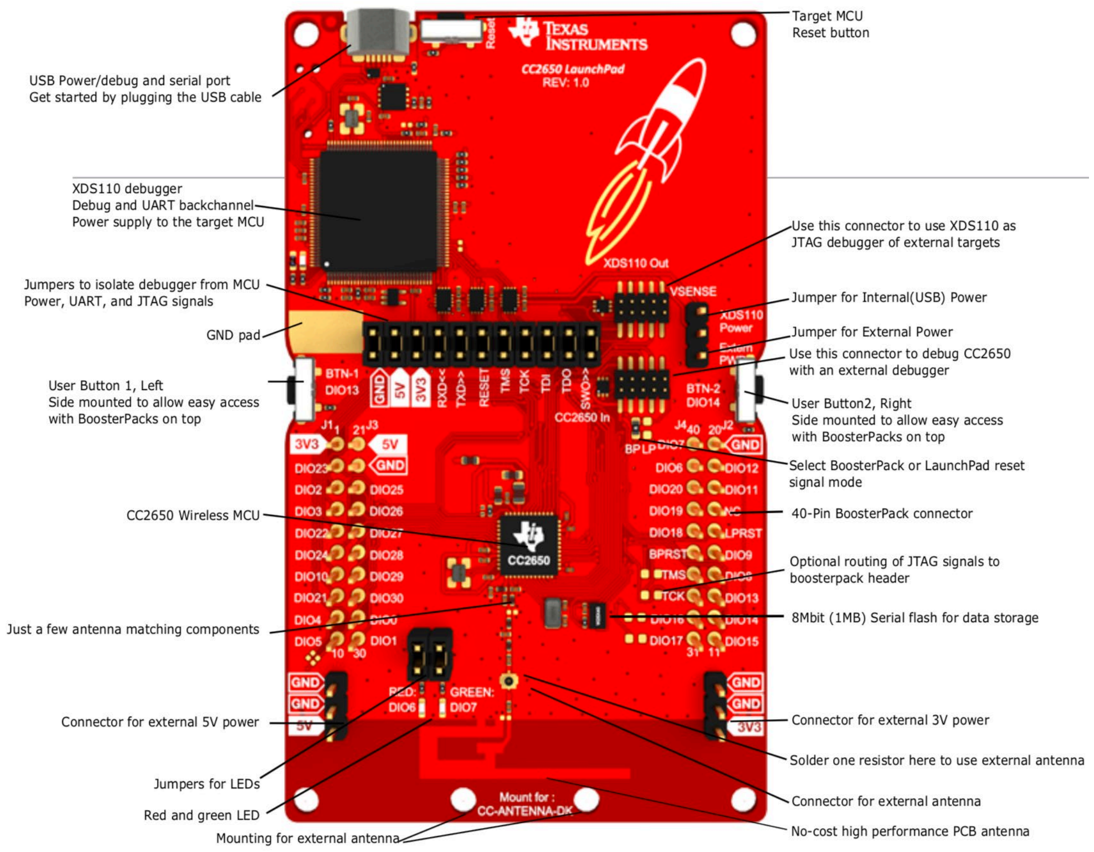
Microcontroller

- ARM Cortex-M3
- Up to 48-MHz Clock Speed
- 128KB of In-System Programmable Flash
- 8KB of SRAM for Cache
- 20KB of Ultralow-Leakage SRAM
- 2-Pin cJTAG and JTAG Debugging
- Supports Over-The-Air Upgrade (OTA)

Communication

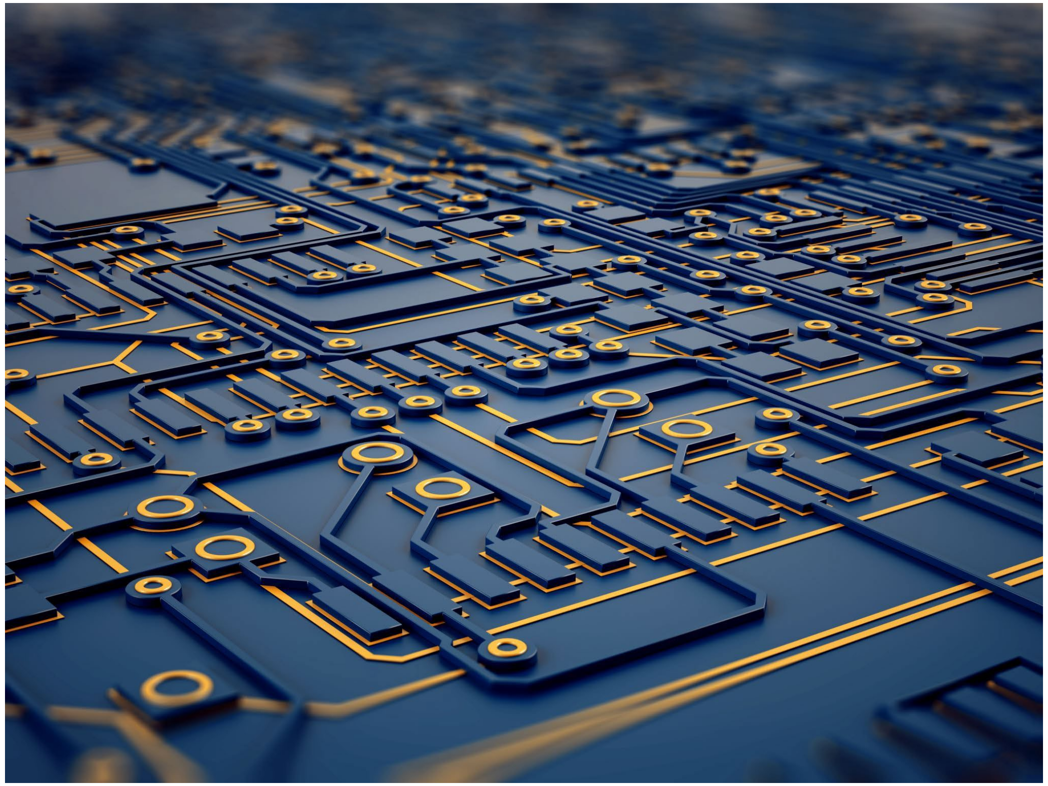
- 2.4-GHz RF Transceiver
- Compatible With Bluetooth Low Energy (BLE) 4.2 Specification
- IEEE 802.15.4 PHY and MAC

CC2650 Launchpad Overview



https://www.ti.com/lit/ds/symlink/cc2650.pdf?ts=1633430232781&ref_url=https%253A%252F%252Fwww.google.com%252F

Operating Systems for Sensor Nodes



Limited Resources

- Limited computational and storage capabilities
- Limited communication capabilities
- Limited energy budget

Activities driven by interaction with environment

- Message arrival, sensor acquisition
- Concurrency Management
  - Event arrival and data processing are concurrent activities
  - Potential bugs must be managed (e.g., race conditions)

Management

- Over-the-air programming

Other requirements

- Applications may have soft real-time requirements

TinyOS

- Specifically targeted to sensor nodes
- Component-based architecture
- Event-based concurrency
- Split-phase operation

http://www.tinyos.net/

TinyOS

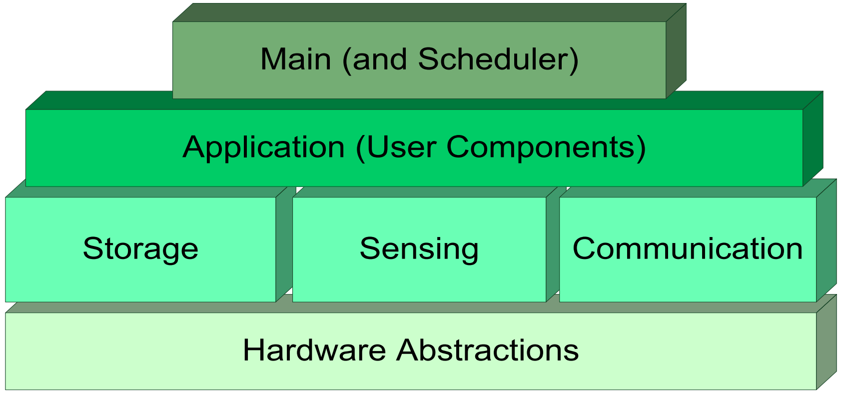
Component-based architecture

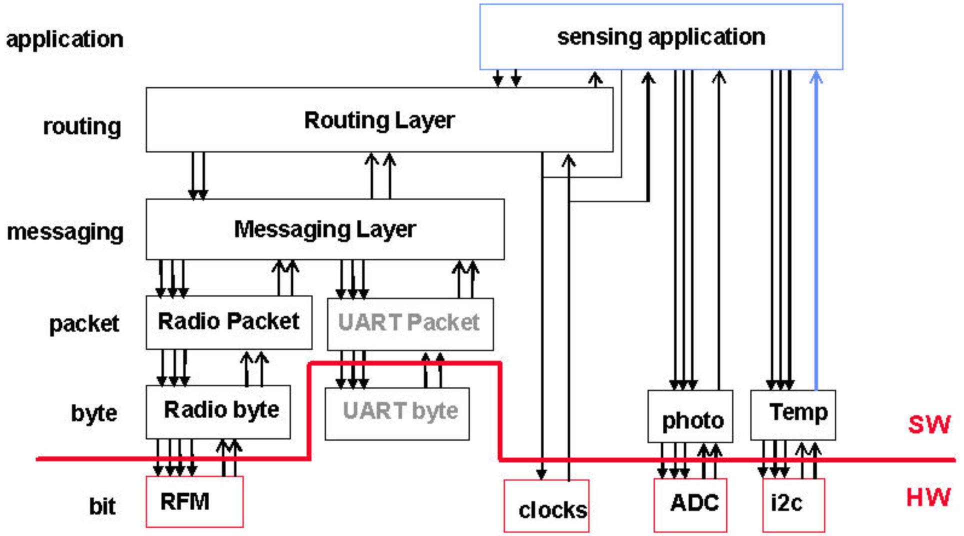
Components

- Software modules
- Hardware modules
- The distinction is invisible to developers

Unused Services

- Excluded from the application

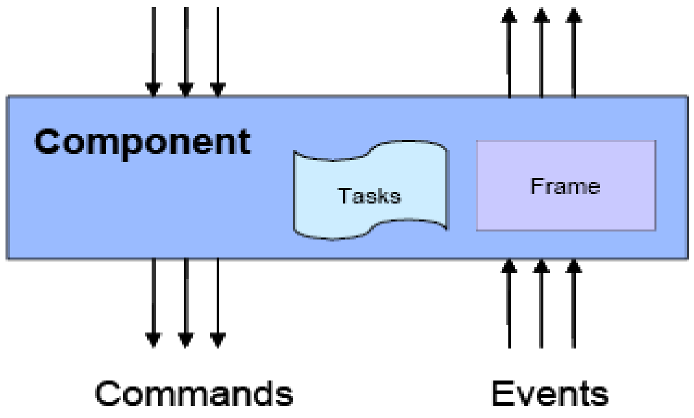




TinyOS computation model

Component

- computing entity
- Interface (commands and events)
- frame (private variables)



Computing abstractions

- command
  - service request to a component
  - non-blocking
- event
  - ✓command completion
  - ✓message
  - ✓interrupt
- task
  - ✓context of execution (function)
  - ✓run to completion
  - ✓preemption by event

TinyOS development env

nesC language

- extension to the C language
- definition of interfaces
- abstraction between definition and composition of components

nesC compiler and OS source

- composition of the component graph (at compilation time)
- TinyOS computational model (additional checks)

TOSSIM simulator

- same code runs in actual nodes and simulator
- flexible models for radio and sensors
- graphical interface

Continki OS

Limited Footprint

Event-driven Kernel

Portability

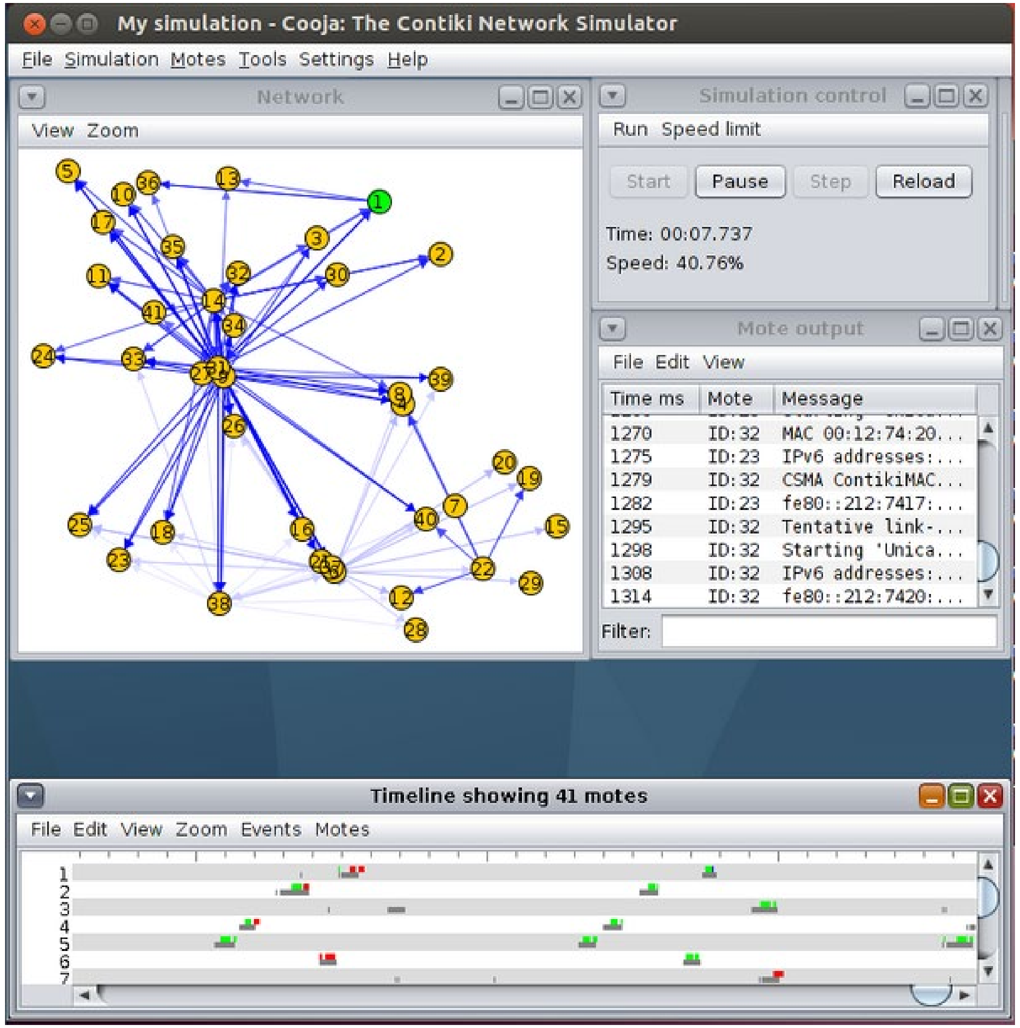
Many different platform supported

Tmote Sky, Zolertia, RedBee, and many others

C Programming

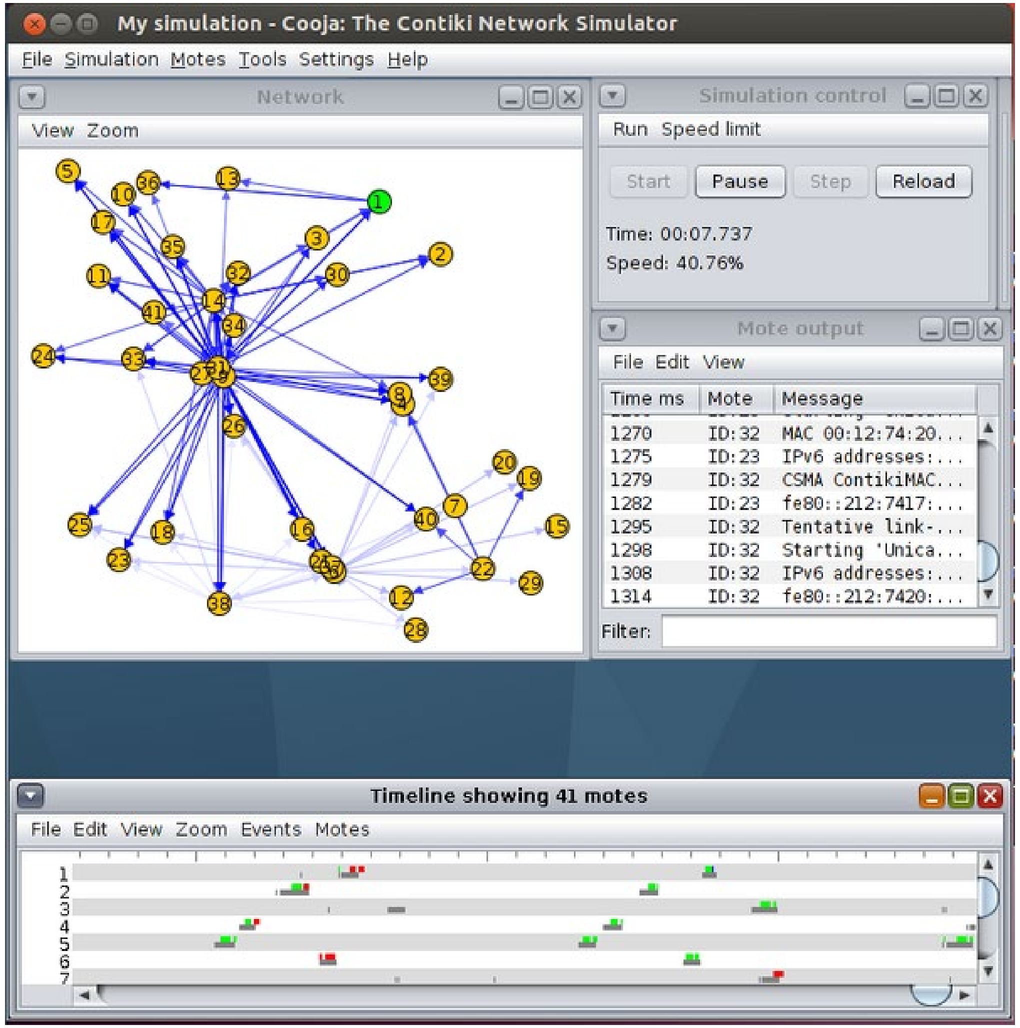
Academic and Industrial support

Texas Instruments, Atmel, Cisco, ENEA, ETH Zurich, Redwire, RWTH Aachen University, Oxford University, SAP, Sensinode, Swedish Institute of Computer Science, ST Microelectronics, Zolertia, and many other



Continki OS

- Protothread (optional multi-threading)
- Dynamic Memory Allocation
- TCP/IP stack (uIP)
  - Both IPV4 and IPv6
- Power profiling
- Dynamic loading and over-the-air programming
- IPsec
- On-node database Antelope
- Coffee file system
- …



Continki OS

Prototread example

Stop-and-wait sender

```
PROCESS_THREAD(reliable_sender, …) {
  PROCESS_THREAD_BEGIN();

  do {
    PROCESS_WAIT_UNTIL(data_to_send());
    send(pkt);
    timer_start();
    PROCESS_WAIT_UNTIL((ack_received() || timer_expired());
  } while (!ack_received());

  PROCESS_THREAD_END();
}
```

Contiki-NG

Operating System for the next IoT generation

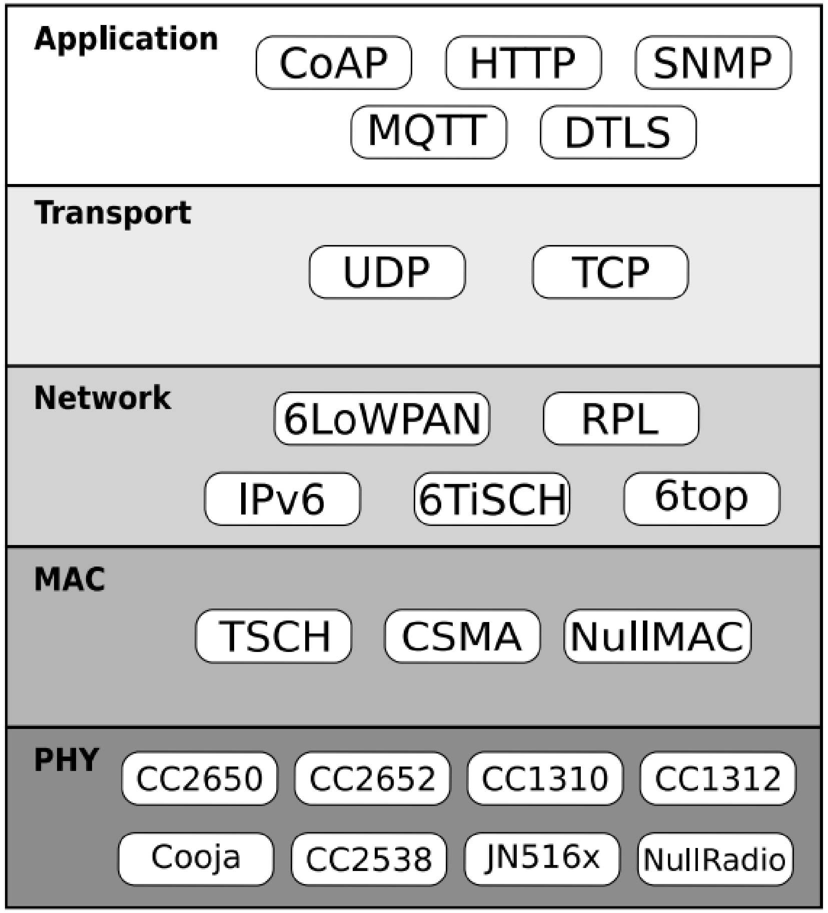
Contains an RFC-compliant, low-power IPv6 communication stack, enabling Internet connectivity

Runs on a variety of platforms based on energy-efficient architectures

- ARM Cortex-M3/M4
- Texas Instruments MSP430

The code footprint is on the order of a 100 kB, and the memory usage can be configured to be as low as 10 kB

The source code is available as open source license



George Oikonomou, Simon Duquennoy, Atis Elsts, Joakim Eriksson, Yasuyuki Tanaka, Nicolas Tsiftes, "The Contiki-NG open source operating system for next generation IoT devices", SoftwareX, 18, 2022, https://doi.org/10.1016/j.softx.2022.101089.

Application Areas

Environmental Monitoring

- Temperature, Humidity, Light Intensity, …

Presence Detection

- Home security systems
- Energy efficiency in buildings

Location Detection

- Anti-theft systems

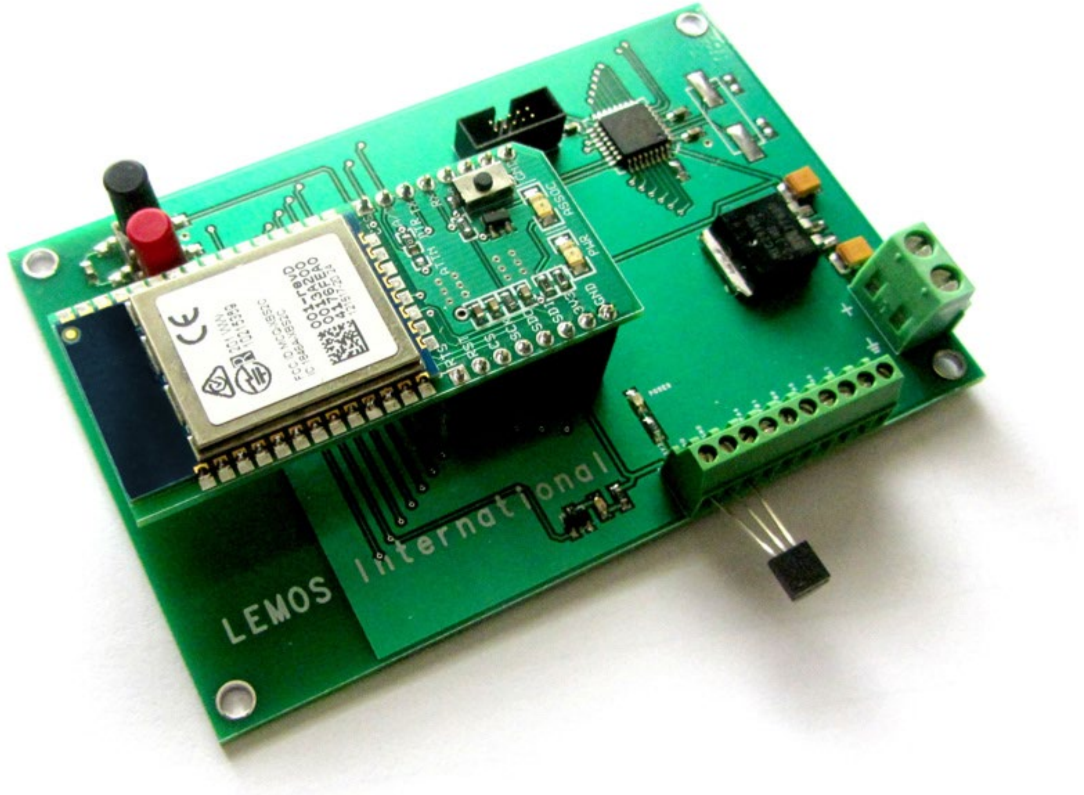
Activity Detection

- Fall detection
- Athlete monitoring

Health monitoring

- Physiological parameters

…



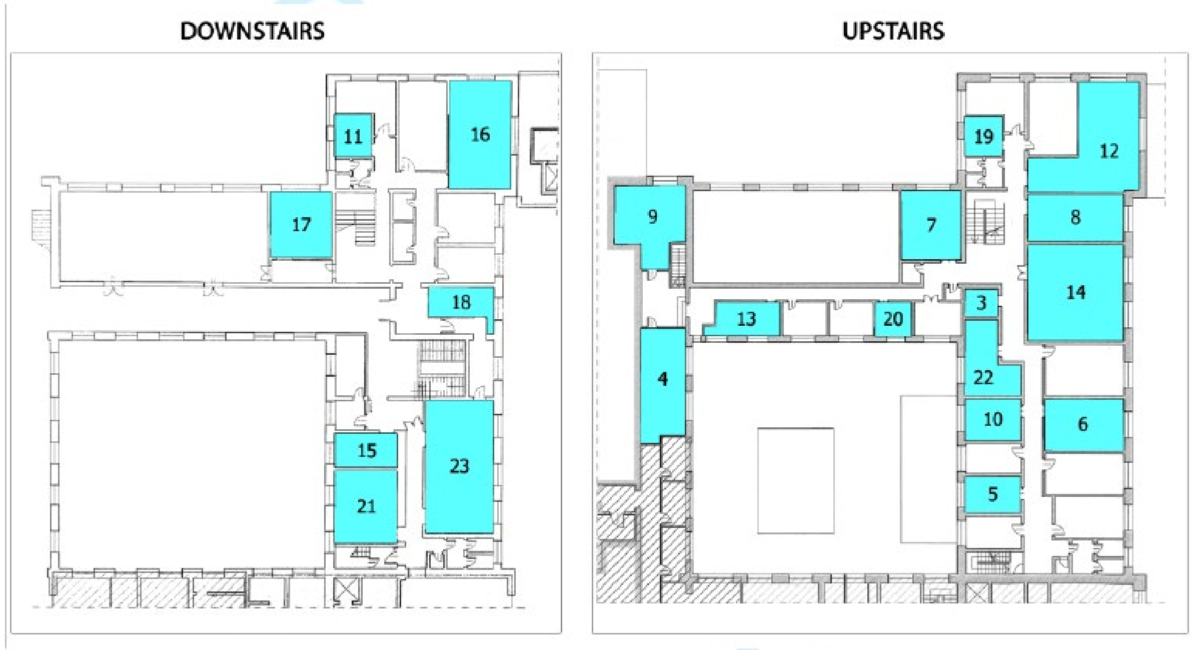
Scientific IoT Testbed

23 nodes deployed in a 2-floor building

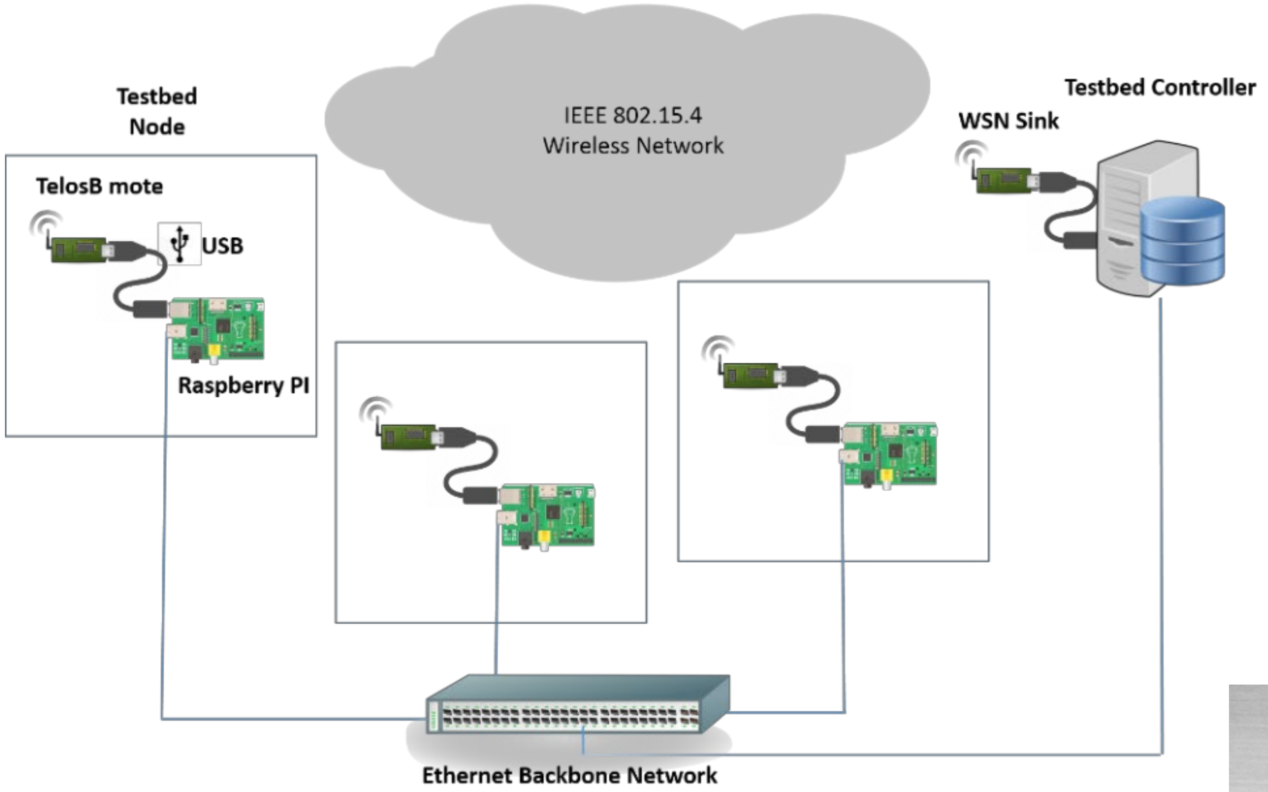
WiFi networks in the area

Zolertia RE-Mote sensor nodes

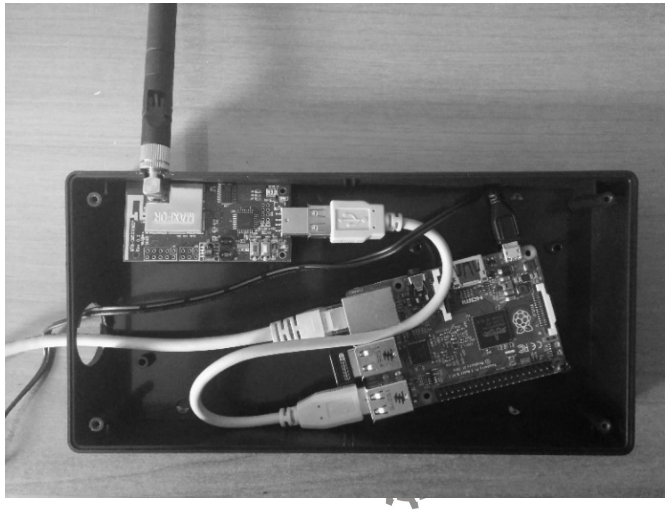
- ARM Cortex-M3 SoC
- CC 2538 radio (2.4 GHz) – IEEE 802.15.4 PHY
- Contiki OS



Scientific IoT Testbed



- ARM Cortex-A7 CPU
- Different Linux distributions
- USB connection provides power and serial communication to the Re-Mote



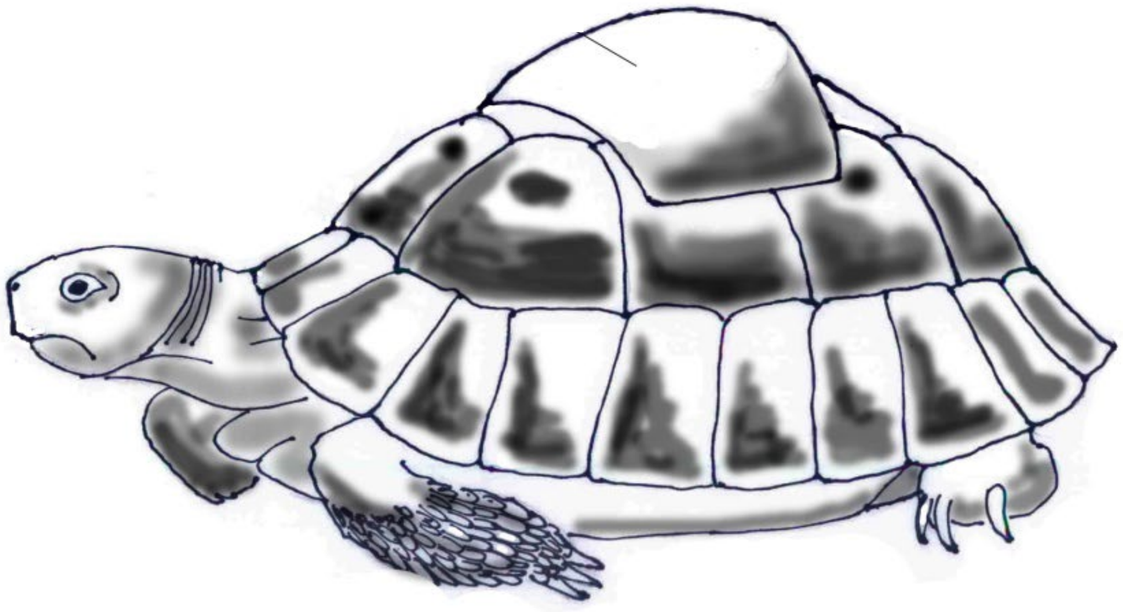
Very Special Instrumenting Device

Very Special Sensor Node

Smart-Phone Sensing Apps

- **Personal Sensing**
  - Personal tracking
  - SoundSense (environmental noise level)
  - …
- **E-health**
  - Fall detection, Activity detection
  - Gait Analysis, Sleep Quality Analysis, …
  - Hearth rate monitoring
  - Well-being support
- **Social sensing**
  - where do people live and work
  - how they commute from home to work
  - average time spent waiting for buses
  - how public and private places are used
  - what residents typically do
- **Participatory sensing**
  - information about the environment
  - weather
  - noise pollution
  - urban mobility
  - Congestion
  - …
- **Others**

An example of smart animal



Tortoise@

Project of the University of Pisa

- Dept. of Computer Science
- Dept. of Information Engineering
- Patent

Goal

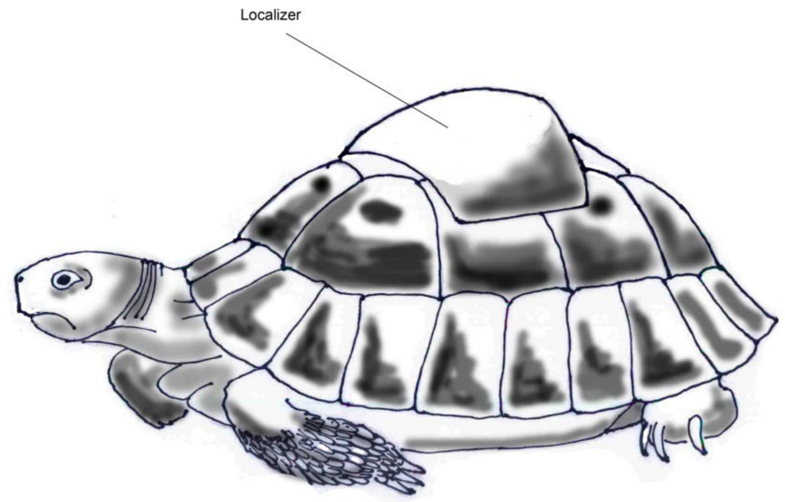
protection of wild tortoise populations

Solution

early localization of the nesting sites

Implementation

- sensor-based system installed on top of the carapace
- localizes the tortoise during its deposition phase
- transmits its geographic coordinates to a remote control center in real time



Tortoise@: Challenges

Deposition pattern recognition

- Deposition occurs only under some environmental conditions
- Typical movements due to nest excavation phase

Weight limitations

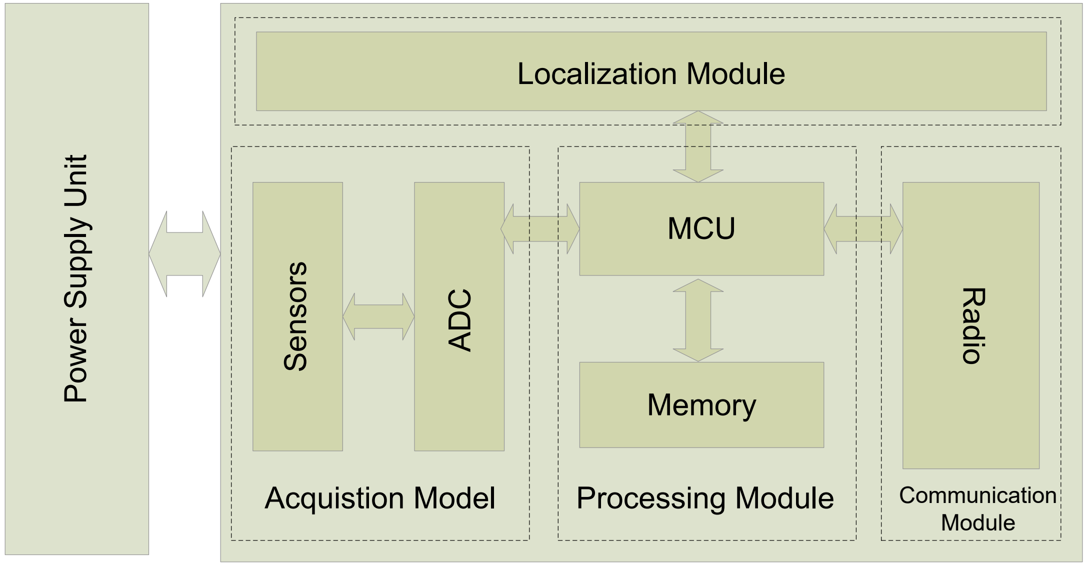
Form factor

Energy limitations

Power management strategies

Tortoise@: System Architecture

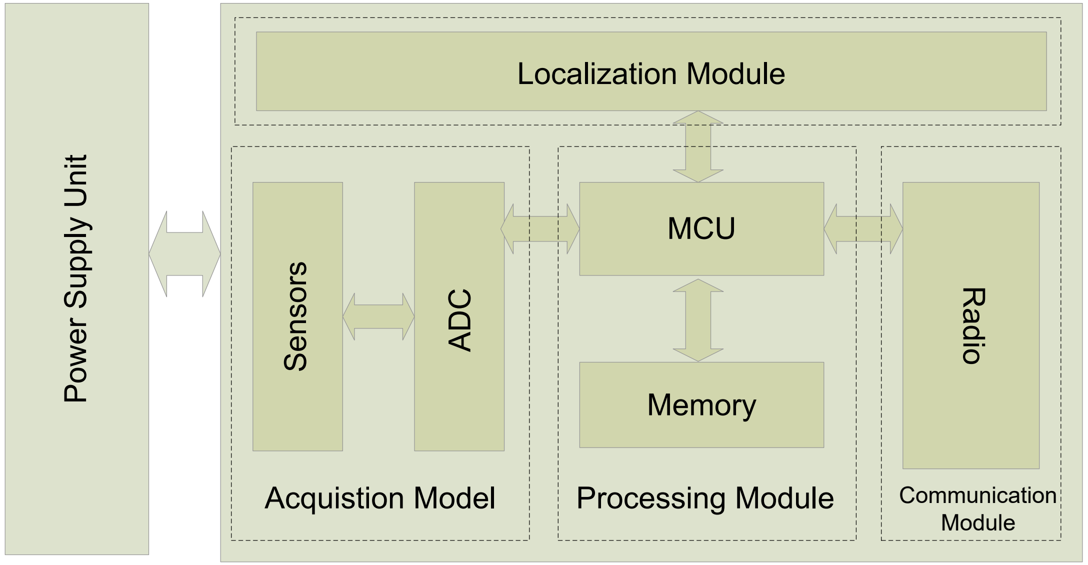
- Acquisition Module
- Localization Module
- Processing Module
- Communication Module
- Power Supply Unit



Tortoise@: System Architecture

Acquisition Module

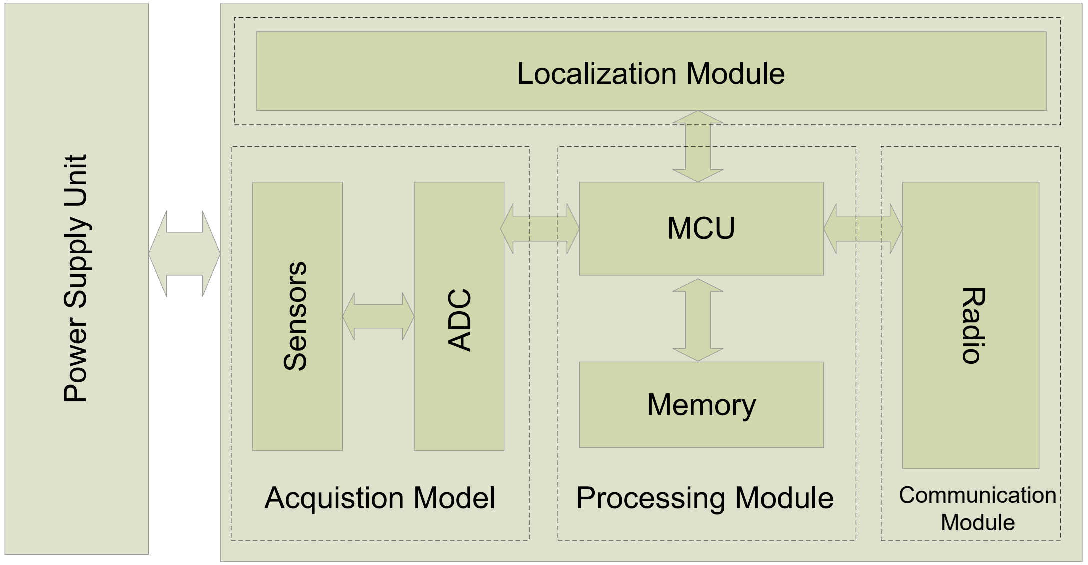
- Monitors external conditions and tortoise's movements
- Environmental sensors
  - ✓temperature
  - ✓light intensity
  - ✓humidity
- 3-axis accelerometer, compass



Tortoise@: System Architecture

Localization Module

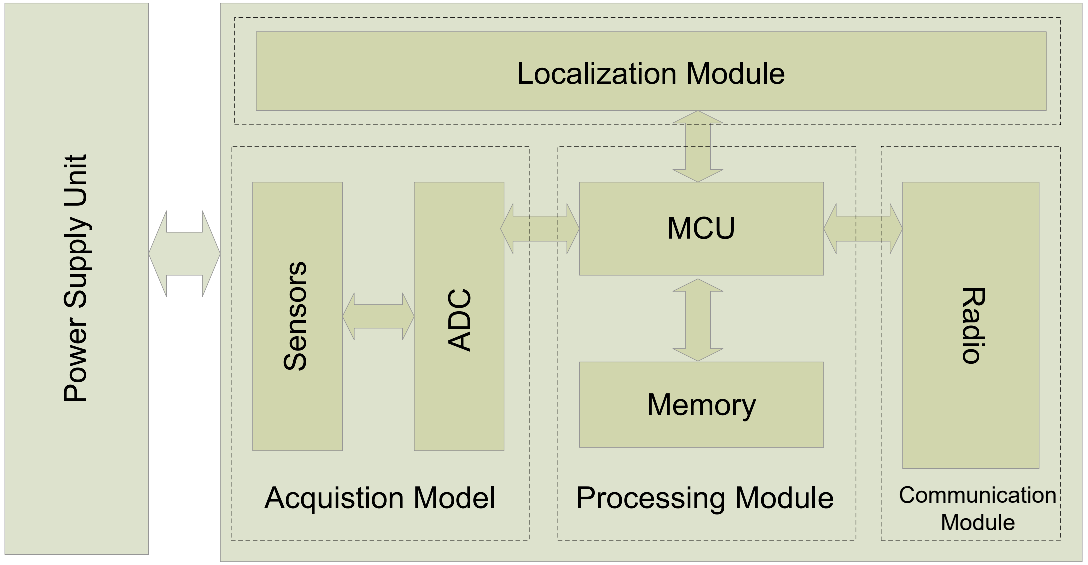
- Geographic localization system
  - e.g. GPS receiver
- Allows to localize the monitored tortoise



Tortoise@: System Architecture

Processing Module

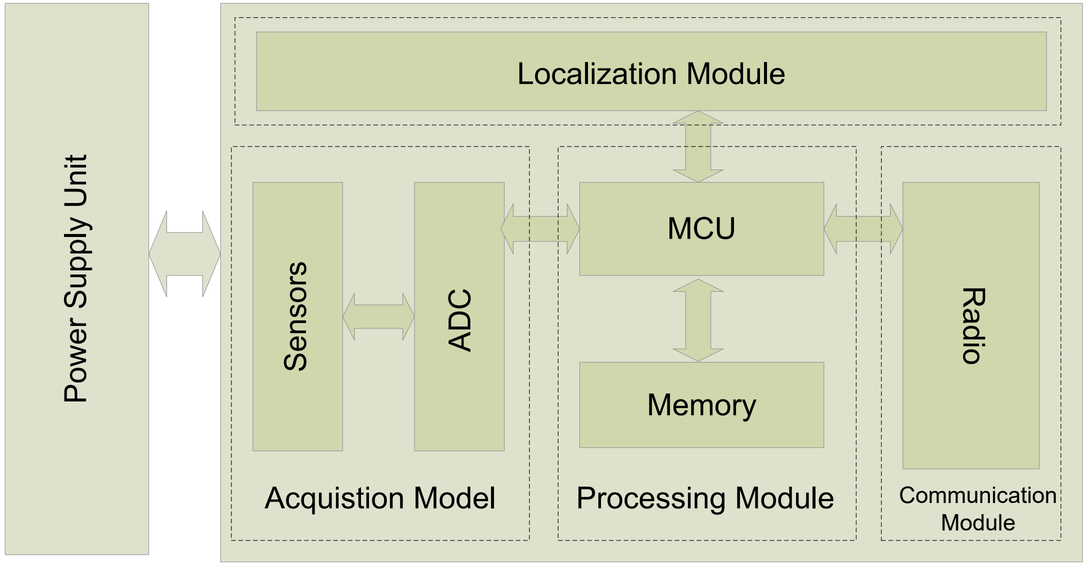
- micro-controller and (RAM + Flash) memory
- stores and processes data received from the sensors and/or the localization module



Tortoise@: System Architecture

Communication Module

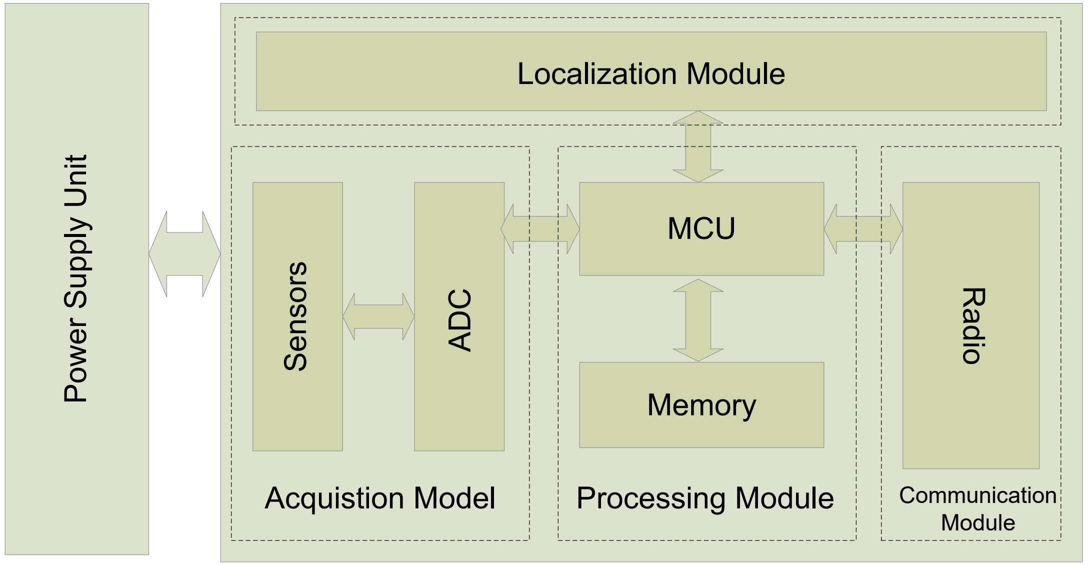
- long-range radio to communicate in real time the tortoise's location and motion direction
- reported by the localization module and compass



Tortoise@: System Architecture

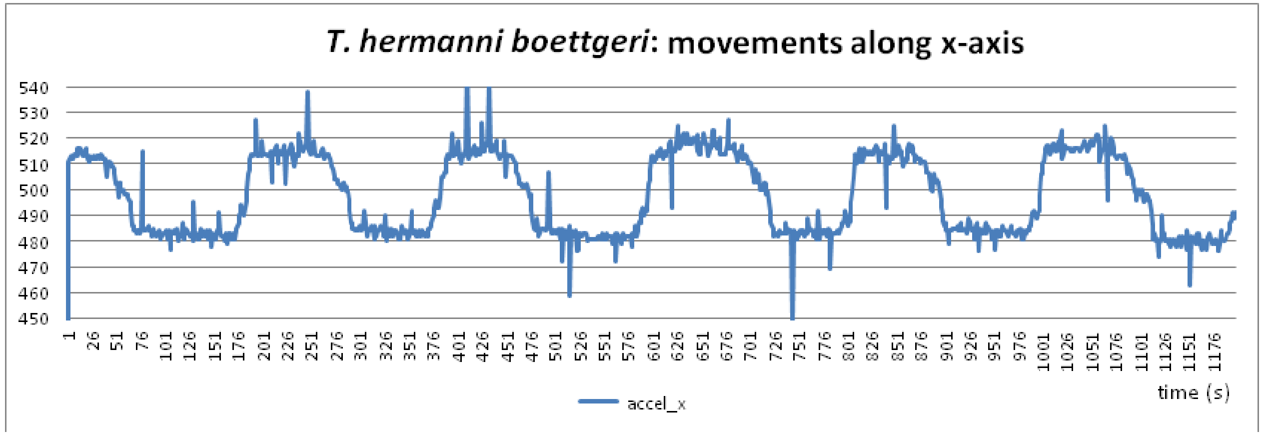
Power Supply Unit

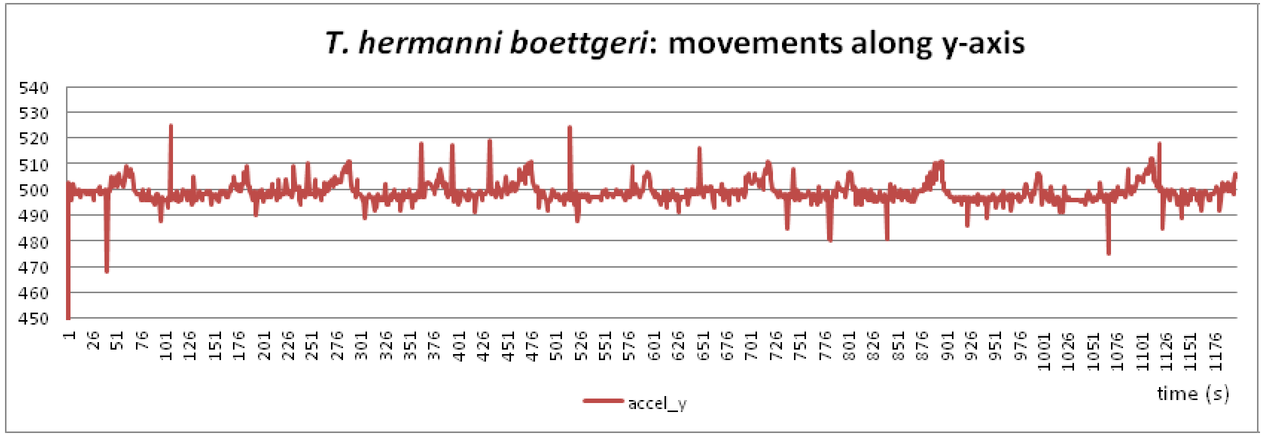
- consists of one or more batteries
- feds all the system components
- must have a lifetime of some months
- Power management required



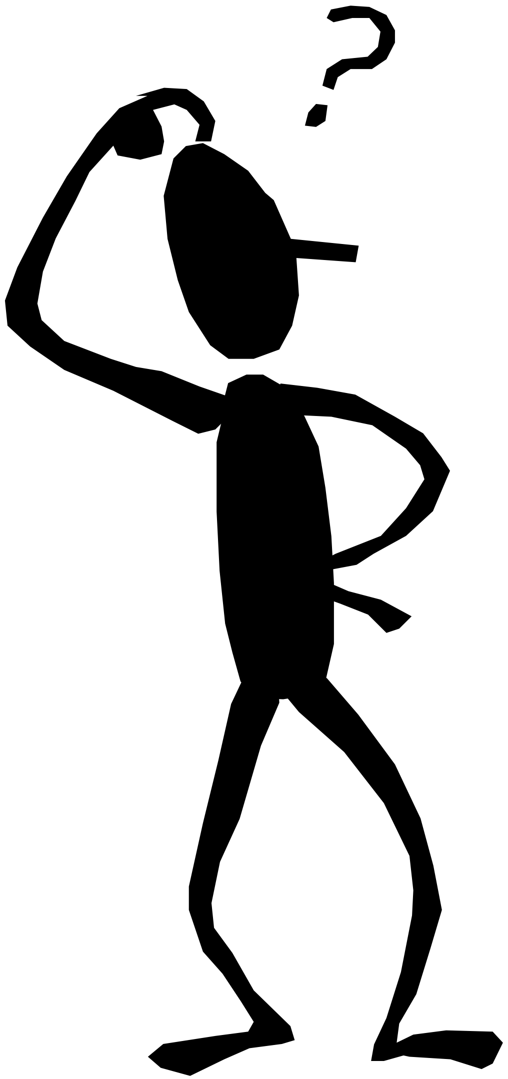
Tortoise@: experimental data

Typical movements of a *T. hermanni boettgeri* tortoise during the nest excavation phase, along the x-axis (top side) and y-axis (bottom side)





Questions



Practical Activity

Think of a realistic application based on

- Passive tags (QR codes, RFIDs)
- Active Tags (Beacons)
- Data Loggers
- Sensors/Sensor Nodes
- Smart Phone

Identify the application requirements

- Application Domain, Potential Users
- Goals, User Desires
- Requirements

Define the overall system

- Services
- Overall architeture
- Components
- Functionalities provided by each component
- ...