SDCI 2012

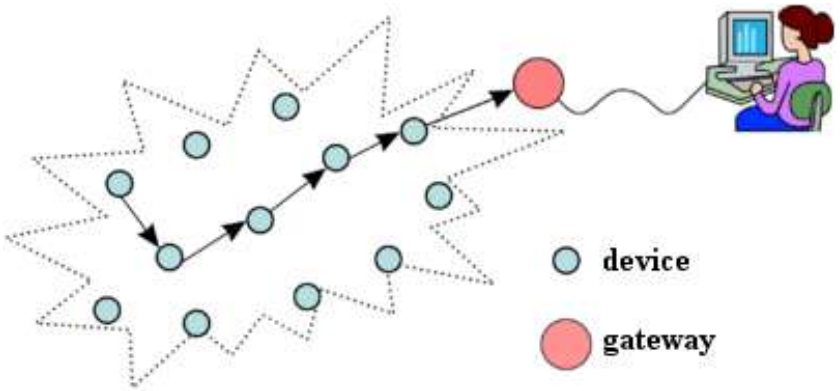
Student Project 6
Sensing Capabilites Go Wireless

Mario Caruso
Francesco Leotta
Leonardo Montecchi
Marcello Pietri

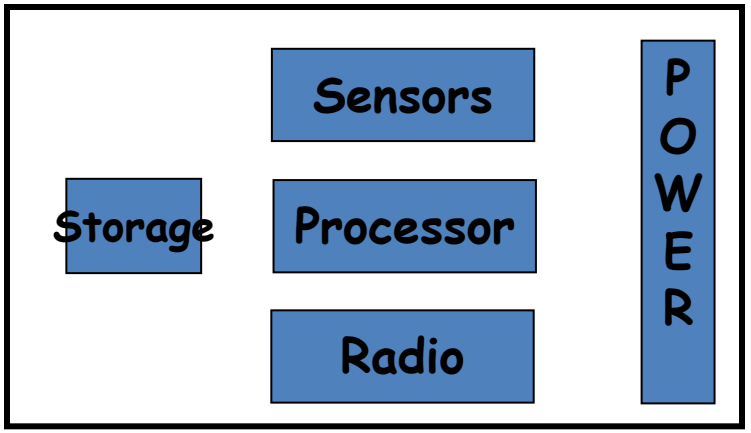
Overview

Wireless Sensor Network

Is a collection of nodes organized into a cooperative network

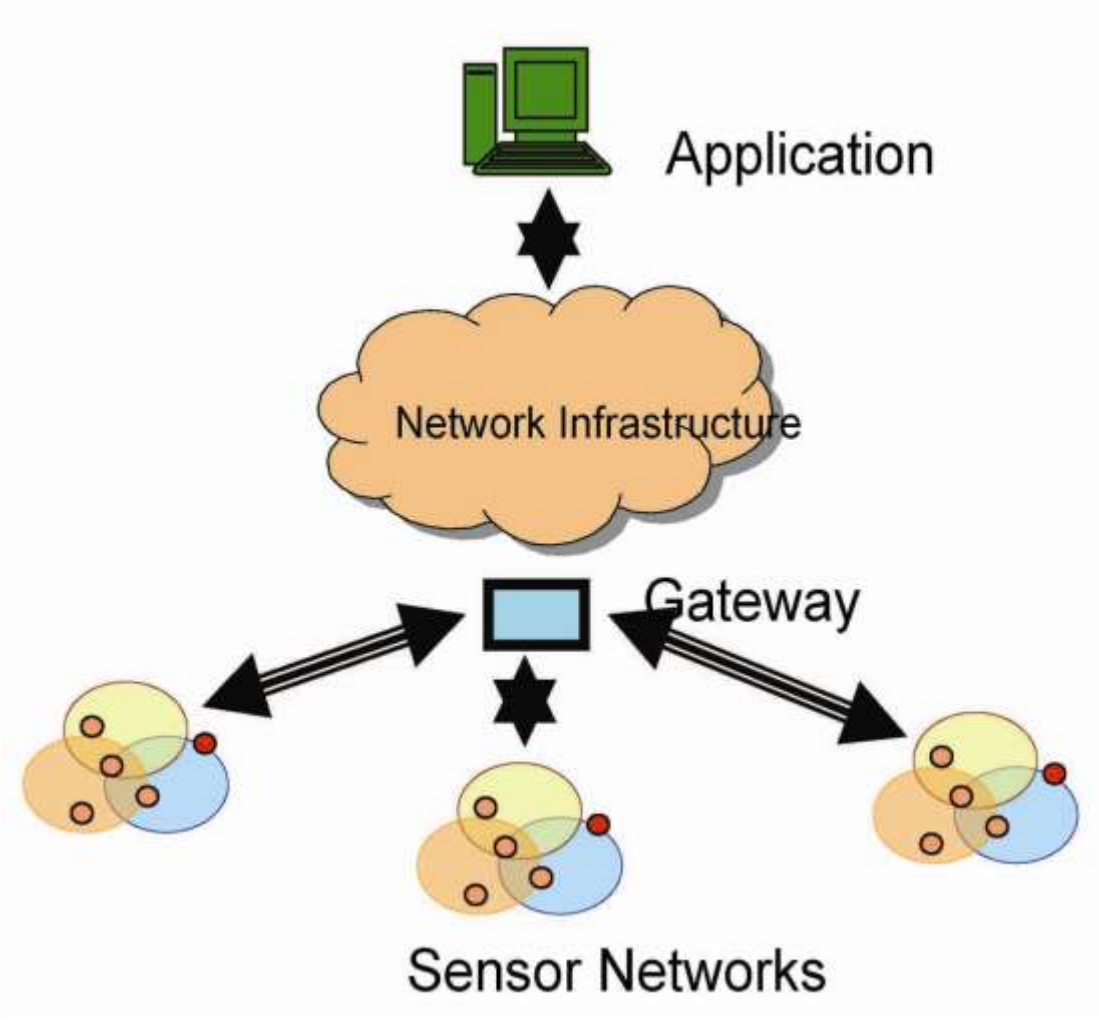


Each node consists of processing capability may contain multiple types of memory have an RF transceiver, a power source (e.g., batteries and solar cells), and accommodate various sensors.



WSN device schematics

Overview



•The **applications** can be very heterogeneous can be developed by everyone

•The **Network Infrastructure** connect the Gateway to the applications

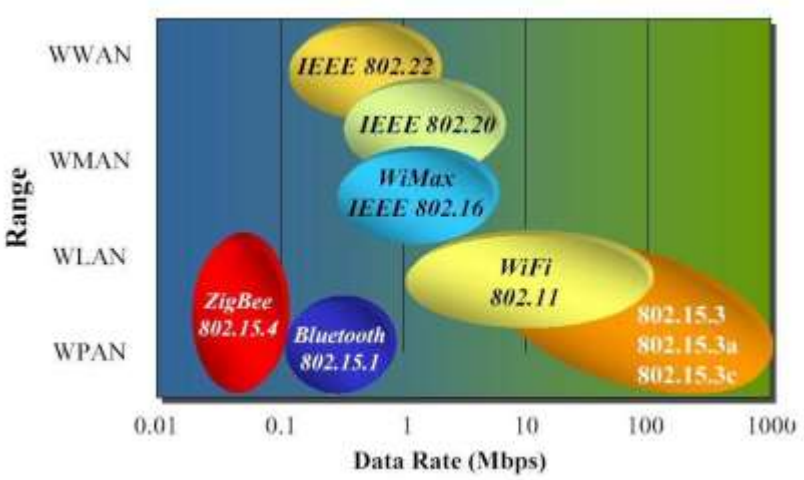
•The **Gateway** collects the data coming from the Sensor Networks

• The type and the topology of the **Sensor Networks** depends mainly on the adopted standard.
The **WSN** usually consists of large number of sensor nodes connected via Wireless; each node has capability to sense data, process the data and send it to rest of the nodes or to the Gateway.

Wireless protocols for WSN

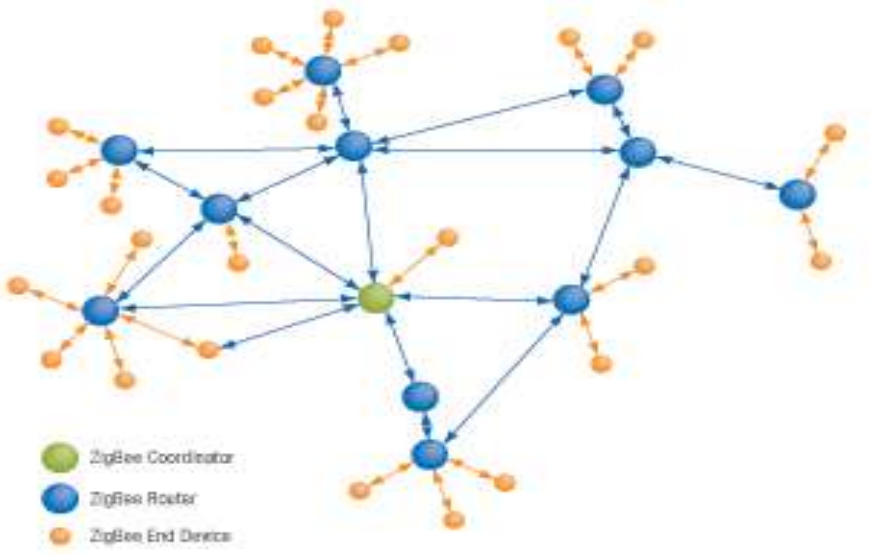
- ZigBee
  - up to hundred of devices
  - multihop
  - range up to 70m (single hop)
- Bluetooth
  - at most 7 slaves
  - range up to 30m
- 6LoWPAN
  - IPv6 over Low power Wireless Personal Area Networks
- Other
  - Wireless M-Bus, T-MAC, S-MAC

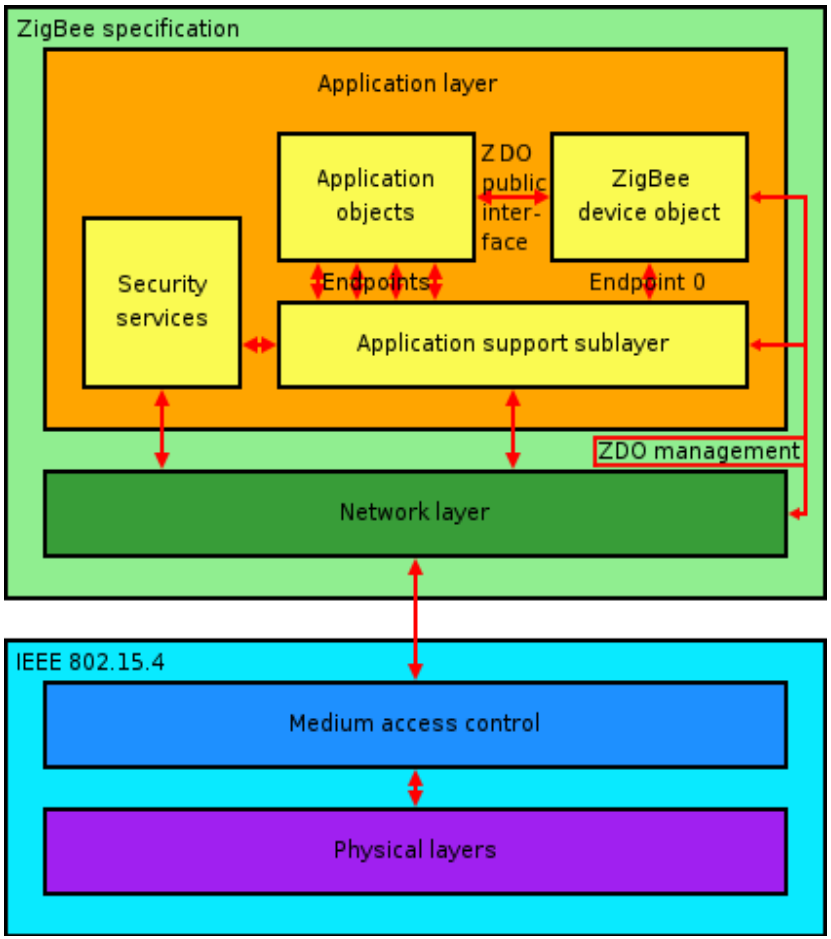
IEEE 802.15.4 is a standard which specifies the **physical layer** and **Media Access Control** for low-rate wireless personal area networks (LR-WPANs).
It is the basis for the **ZigBee**, ISA100.11a, WirelessHART, and MiWi specifications.



Wireless protocols, ZigBee

ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on an IEEE 802 standard for PAN. It is intended to be simpler and less expensive than other WPANs, such as Bluetooth. It is targeted at radio-frequency (RF) applications that require a **low data rate, long battery life**, and ***secure networking***.





Application fields

Environmental Observation:
Sensor networks can be used to monitor environmental changes.
E.g. to monitor air and water pollution, fire dectection in the forest, rainfall observation in agriculture, …
Sensor nodes could be randomly deployed in unknown and hostile areas and relay the exact origin of a pollutant.

Healthcare:
Sensors can be used in biomedical applications to improve the quality of the provided care. Sensors are implanted in the human body to monitor medical problems like cancer and help patients maintain their health.

Building Monitoring:
Sensors can also be used in large buildings or factories monitoring climate changes. Thermostats and temperature sensor nodes are deployed all over the building's area. In addition, sensors could be used to monitor vibration that could damage the structure of a building.

Available Technologies

- Currently, there is no common WSN platform
  - Some platforms have broader user base and developer communities
- Several commercial platforms on the market:
  - Libelium Waspmote (the one that has been used for this project)
  - Coalesenses iSense
  - Arduino
  - Netduino (Arduino variant that uses .NET framework)
  - AIRcable SMD
  - Intel WISP (combines WNS capabilities with RFID tags)
  - MicaZ motes
  - SUN Spot (SUN Microsystems)

Custom platforms

- For some companies / research labs it may be more convenient to build their own devices
  - It allows to reduce costs (boards and sensors are cheap and easy to build)
  - It allows to obtain smaller devices
  - Usually, the RF module is an exception, as it is considerably more expensive and complex to build it
- Large number of specific platforms, developed by companies or academic institutions, e.g.:
  - Glacsweb (University of Southampton), aimed at monitoring glacier behavior
  - WSN430 (INSA de Lyon/INRIA)

Operating Systems for WSN

- Quite different from the ones used in traditional embedded systems, because of:
  - Power, processing and storage constraints, power efficiency, mobility, concurrency
- Multithread-driven
  - Traditional embedded systems approach adapted to WSN
  - Each task is given a CPU slot and all tasks concur to access resources
  - RETOS, MANTIS-OS …
- Event-driven
  - Responds to events such as an incoming data packet or a sensor reading, by calling an event handler and allocating CPU and resources to solve the event in hands to completion
  - TinyOS, SOS, Contiki, Yatos …

Languages and libraries

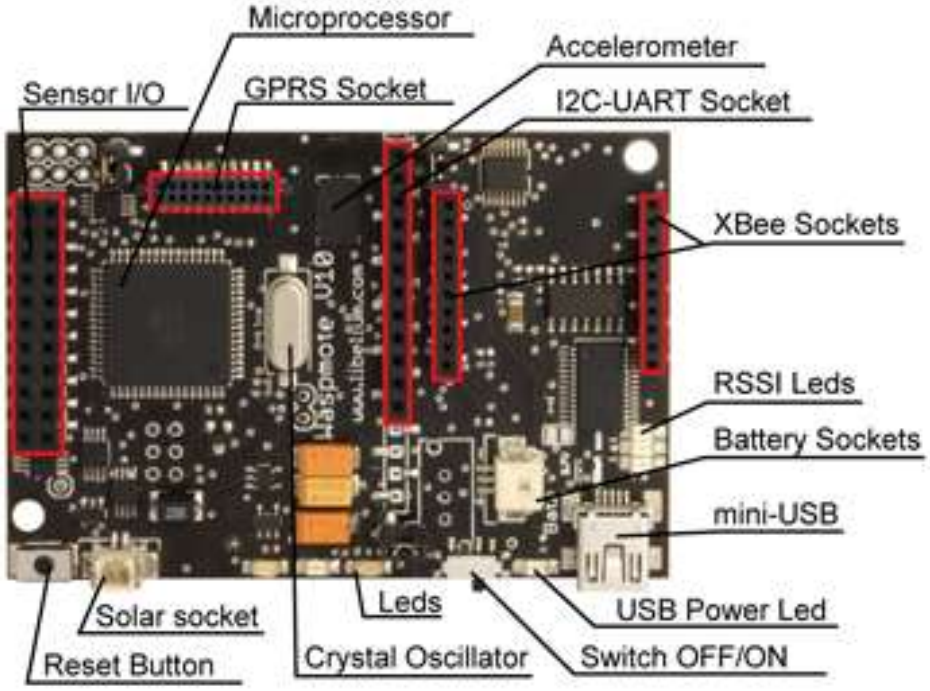
- Most basic choice:
  - Program the WSN devices in the microcontroller assembly of C
- However,
  - Specific toolkits are available to ease the programming process and achieve the hardware independence on the software
  - nesC
    - An extension of the C language designed for developing TinyOS applications
  - Protothreads
    - Lightweight stackless threads designed for embedded systems or WSN nodes

Open Issues

- Scalability
  - Networking coordinators have limited resources thus supporting a limited number of devices
- Power consumption
  - Low consumption components and keeping the nodes in stand-by as long as possible
  - Sun, wind, vibration and bio energy are some of the possible solutions already under study.
- Bandwidth
  - Limited bandwidth also due to constraints on power consumptions
- Heterogeneity
  - Several different platform, specific components, specific programming languages and frameworks
- Security
  - Issues due to the nature of communication
  - Specific security attacks

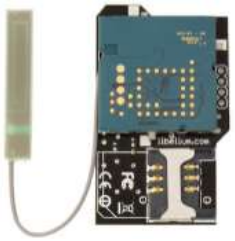
Libelium Waspmote platform

- Waspmote Board
  - Main component of the platform
- Contains
  - Microprocessor
  - Memory
  - Data storage
- Supports the connection of other modules
  - Sensor boards
  - Communication devices
  - Batteries



Libelium Waspmote: Communication

- Different radio communication technologies supported:
  - 802.15.4/ZigBee
  - Bluetooth
  - GSM/GPRS
  - RFID
  - WiFi
- Can also be combined on the same device
  - Multifrequency sensor networks
  - Hybrid technology sensor networks
- Waspmote Gateway
  - allows ordinary PCs to join the wireless sensor and communicate with Waspmotes
  - Connects through USB serial port



Libelium Waspmote: Sensor boards

- Sensor boards: boards supporting different kinds of sensors
- Gases
  - CO, CO2, H2, CH4…
  - Control of city pollution, fires…

- Events
  - Temperature, pressure, liquid level luminosity, vibration
  - Security (presence of persons) emergencies (fires, floodings)

- Smart Parking
  - 3 axis magnetic field

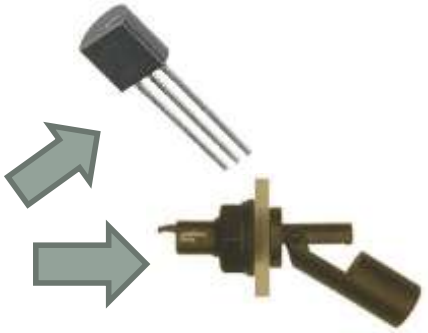
- Radiation

Project Description (1/2)

- A WSN based **temperature and liquid level** monitoring application

- A three node WSN
  - Two wireless motes **Libelium Waspmote 802.15.4-PRO SMA 5dBi** each provided with:
    - A Xbee communication module

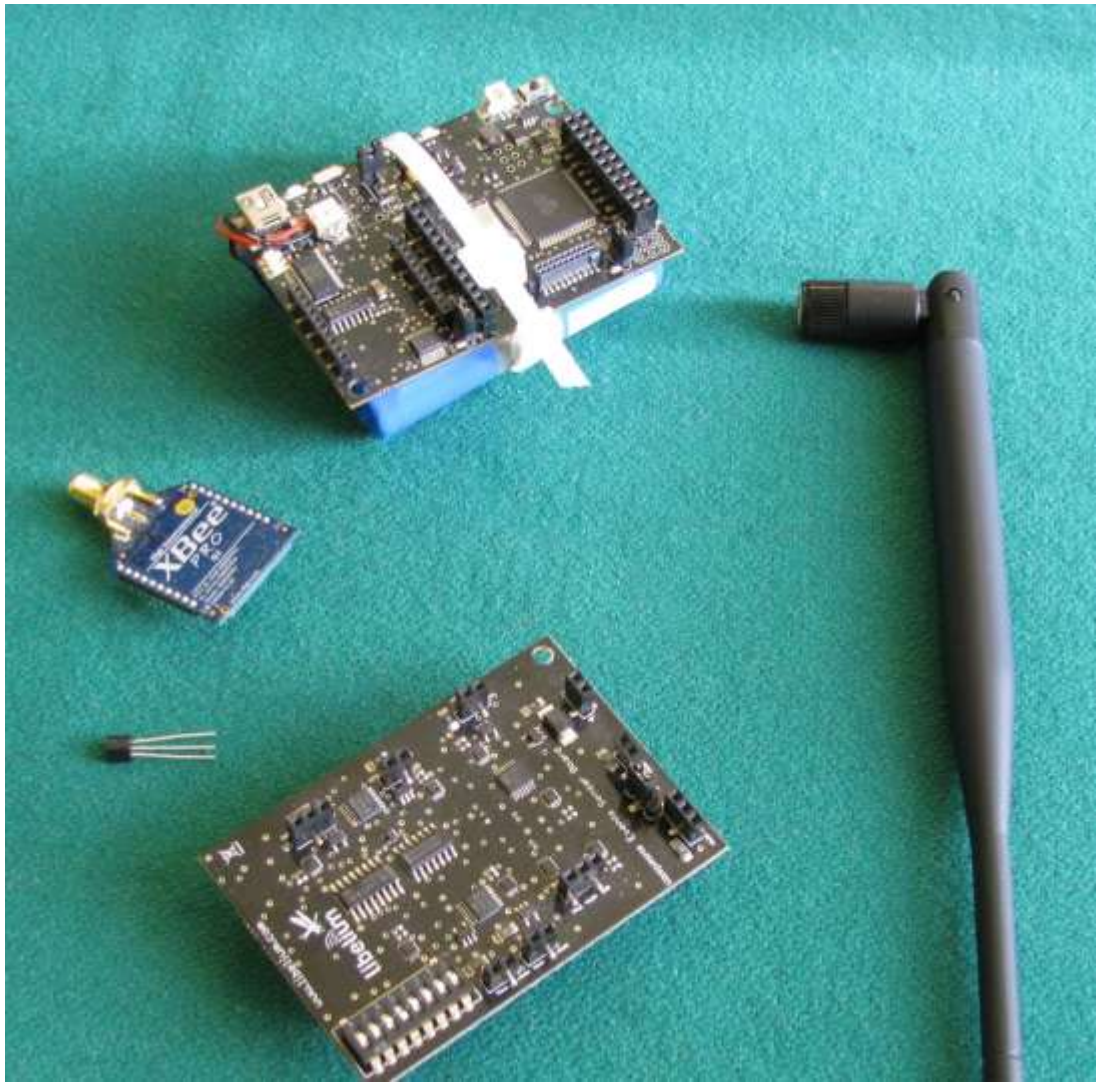
    - A **Libelium Events** sensor board
      - Both sensor boards equipped with a **temperature sensor**
      - A single sensor board equipped with a **liquid level sensor**



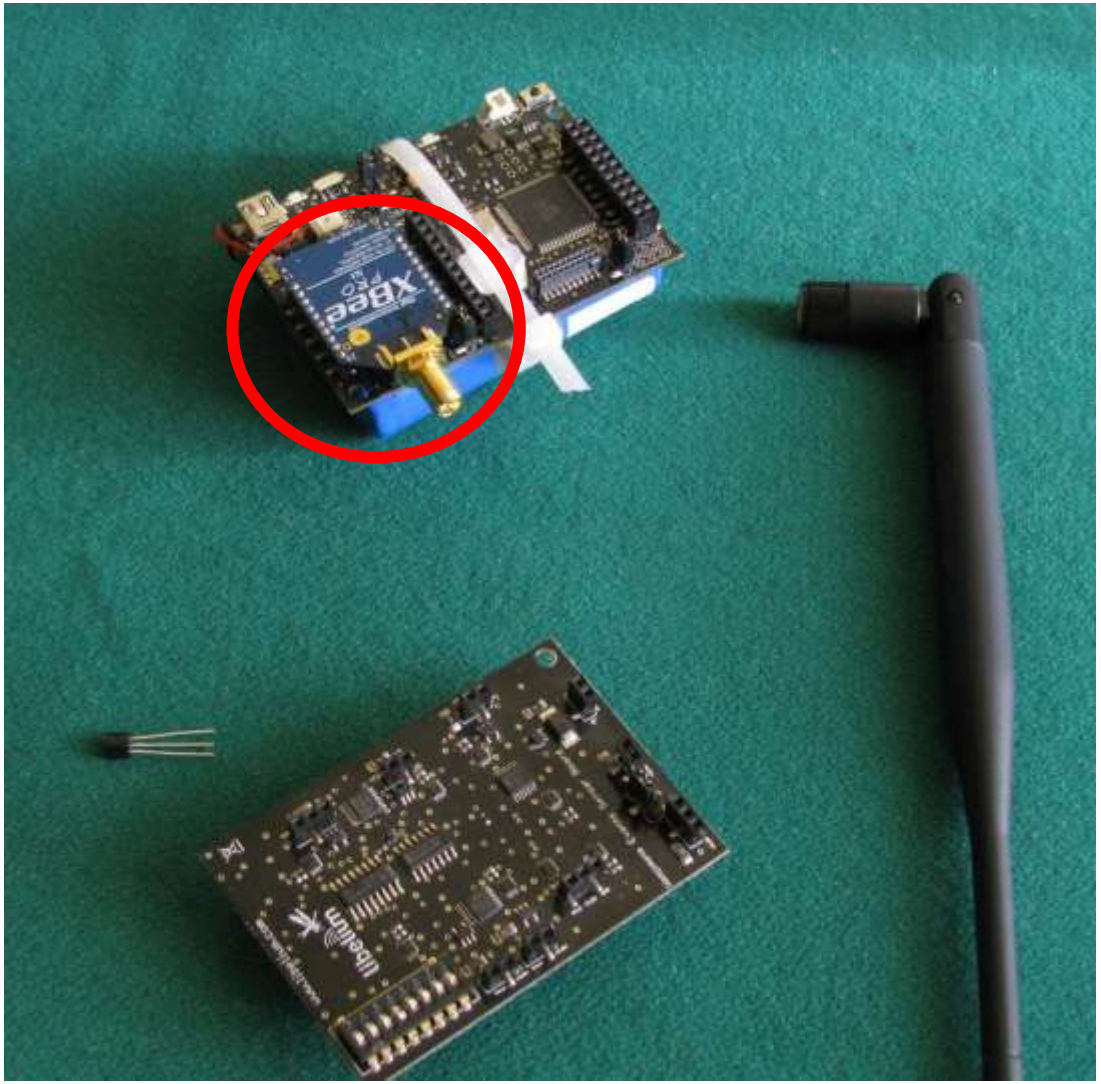
    - A battery pack

  - A wireless gateway **Libelium Waspmote Gateway 802.15.4-PRO SMA 5dBi** provided with an Xbee communication module

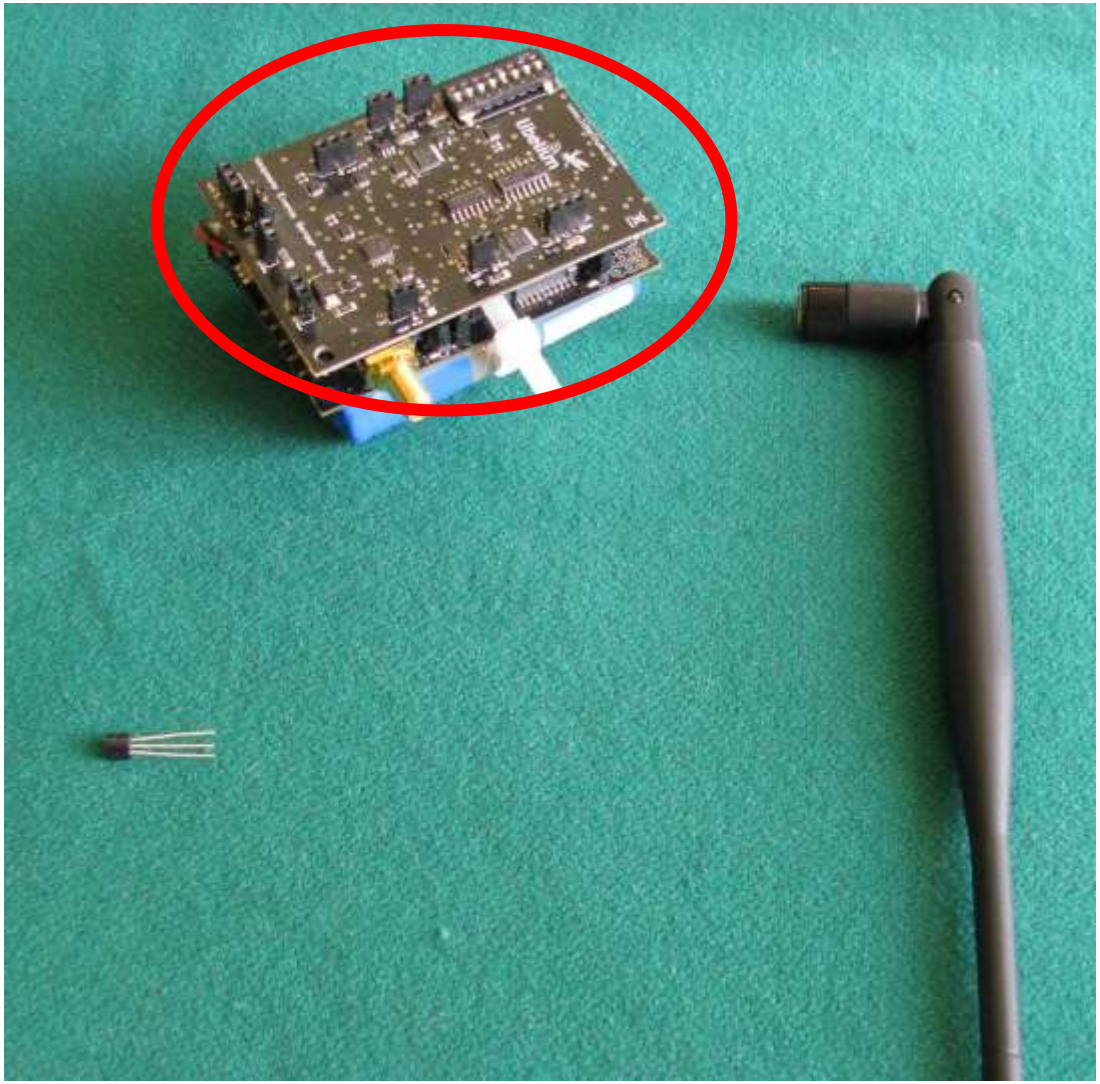
Assembling the Mote



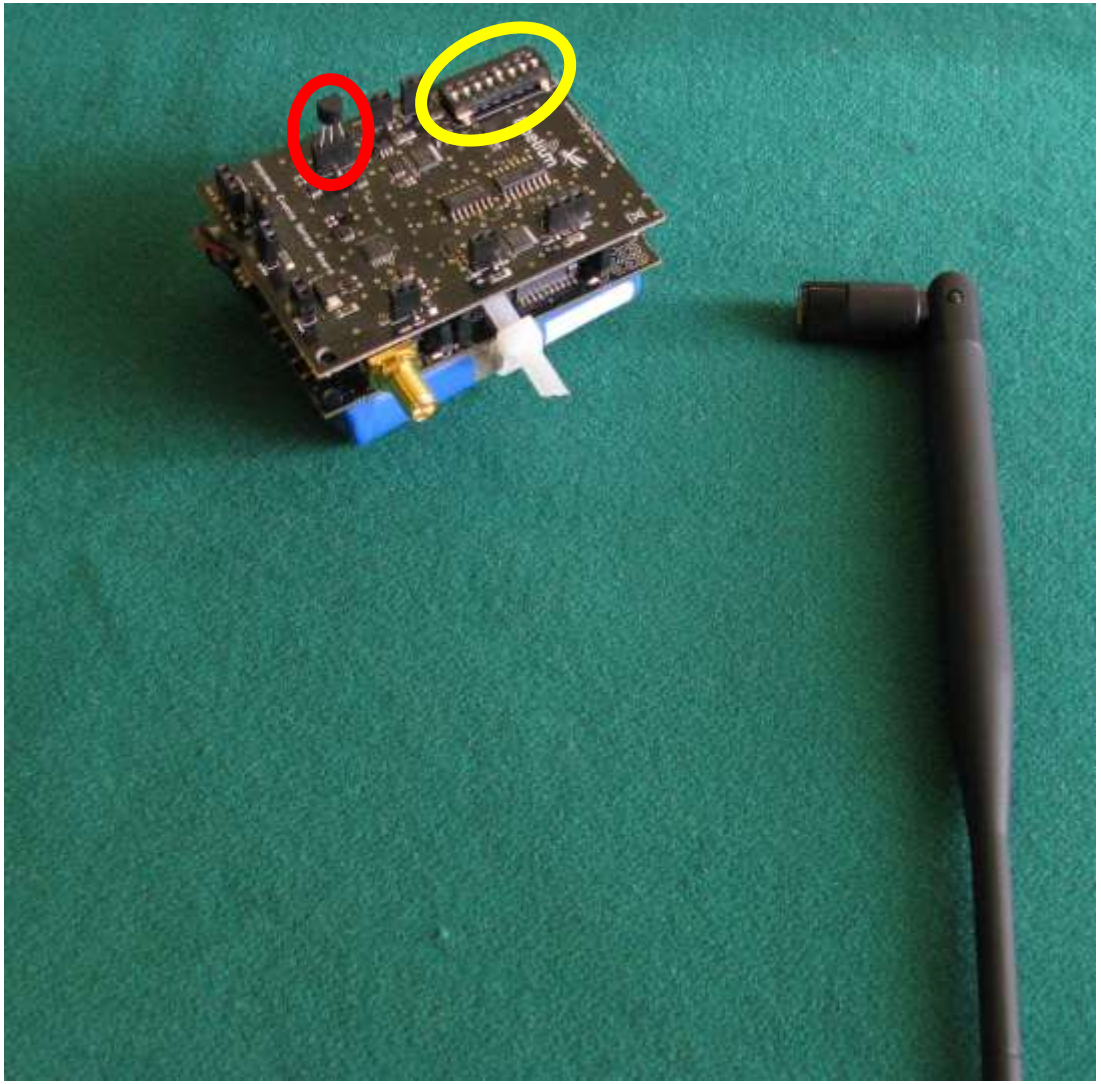
Assembling the Mote (XBee Module)



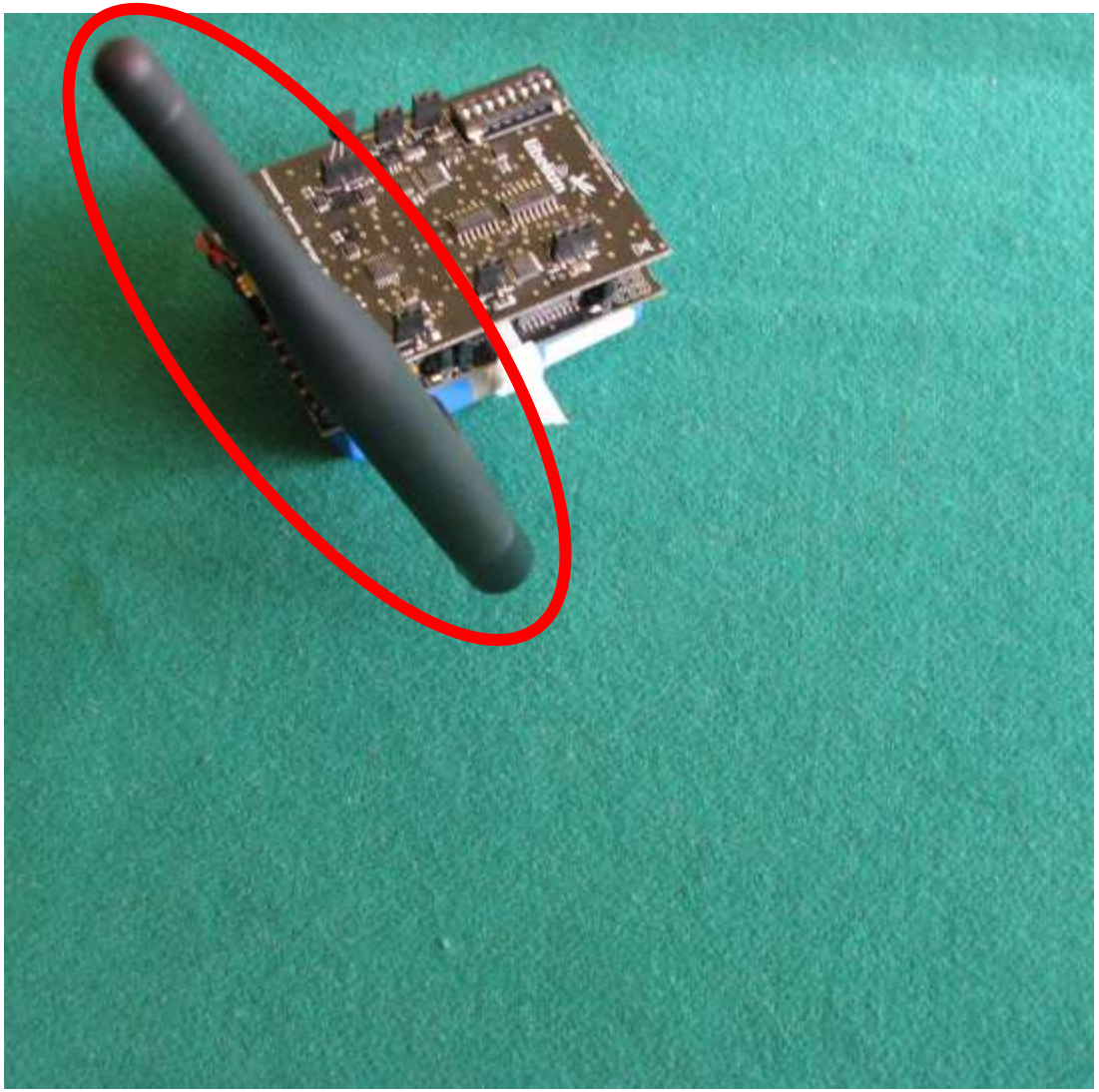
Assembling the Mote (Sensor Board)



Assembling the Mote (Temp. Sensor)



Assembling the Mote (Antenna)

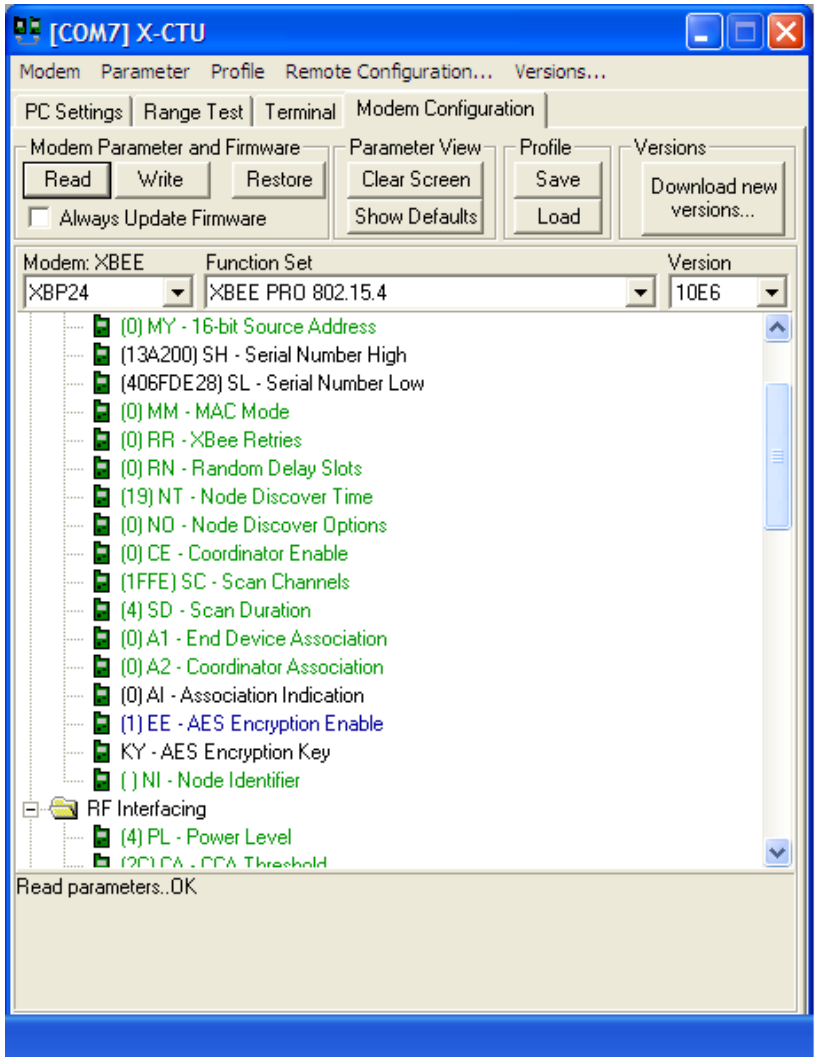


Project Description (2/2)

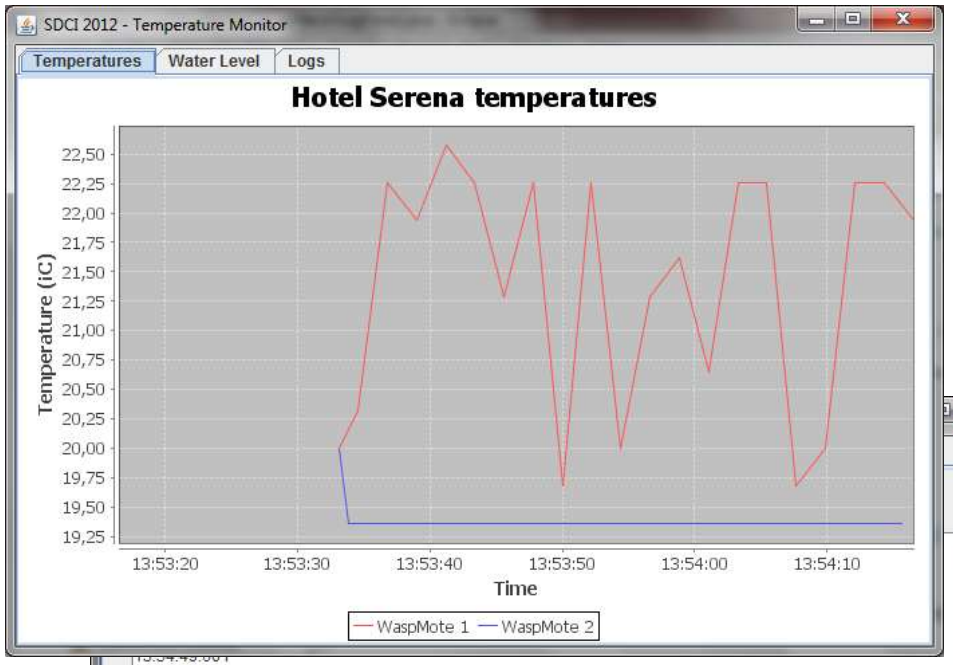
- Xbee Network features:
  - PAN (Personal Area Network) ID 0x1234
  - Channel 0x0D
  - AES-128 Encryption (Password SDCI2012MCFLLMMP)
    - Encryption is totally transparent to the final application and provided by the Xbee layer
    - Mote messages are parsed from the monitoring application as plain text received using a serial port

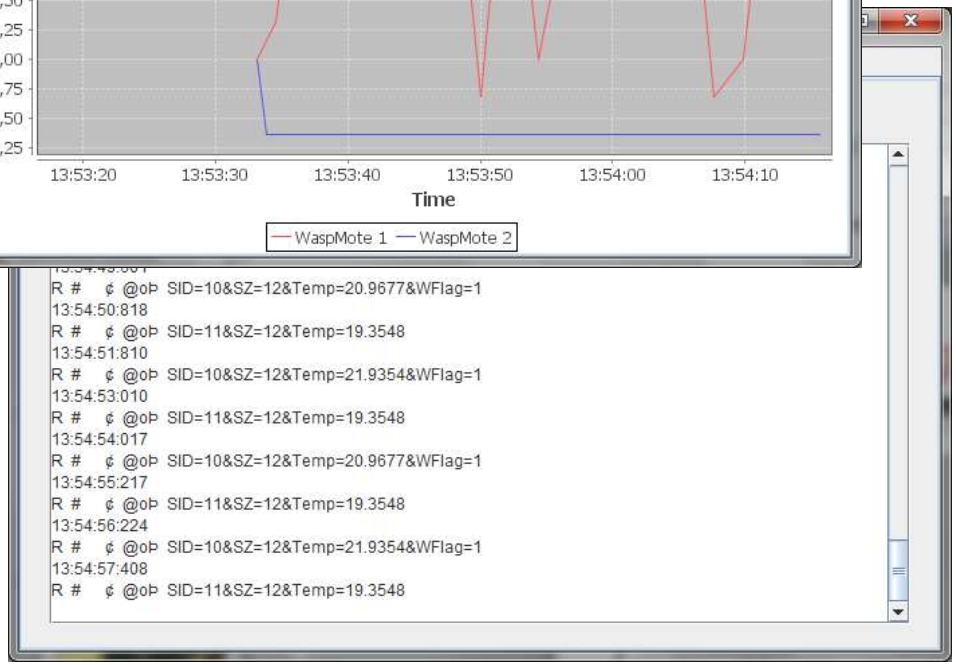
- Each mote sends to the gateway, with a 2 seconds interval, a string containing:
  - Application defined SensorID
  - Application defined ZoneID
  - Temperature value (float)
  - A boolean value denoting the state of the liquid level sensor (if available)
  - Example: ***SID=10&SZ=12&Temp=19.6774&WFlag=0***

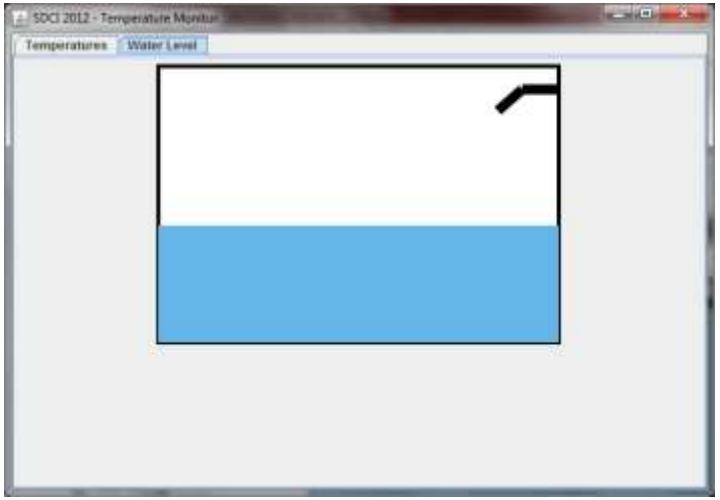
Development KIT

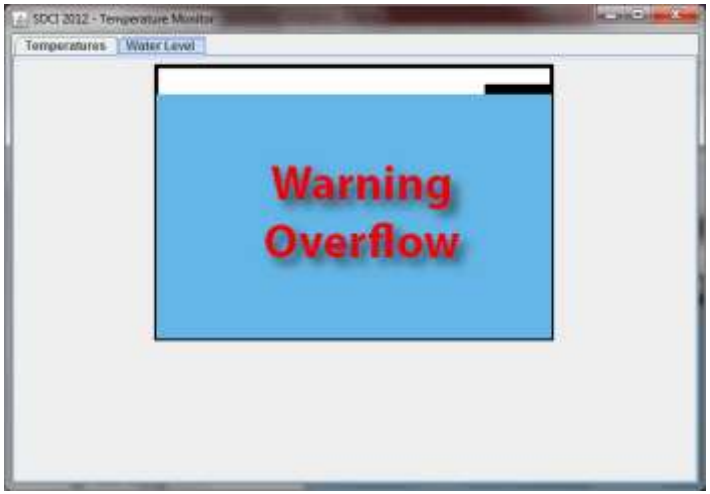


Java Monitoring Application









Thank You!!!