

Lab 5

*School of Architecture, Civil and
Environmental Engineering*

EPFL, WS 2022-2023

http://disal.epfl.ch/teaching/signals_instruments_systems/

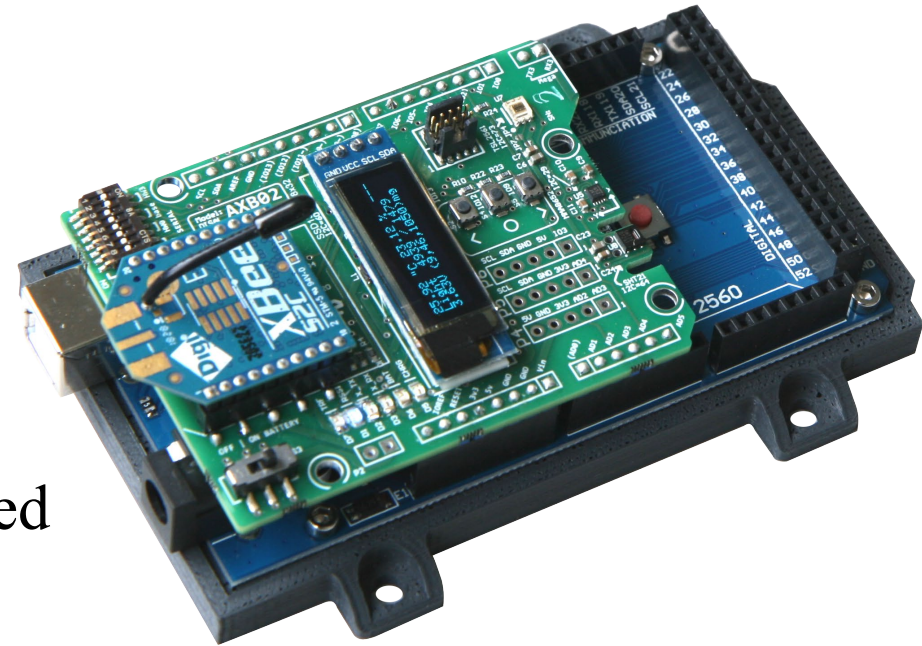
What you will learn today

- Sensor nodes
- Local sensing with one node
- Remote sensing with two nodes

- New hardware tools:
 - Arduino
 - Xbee communication

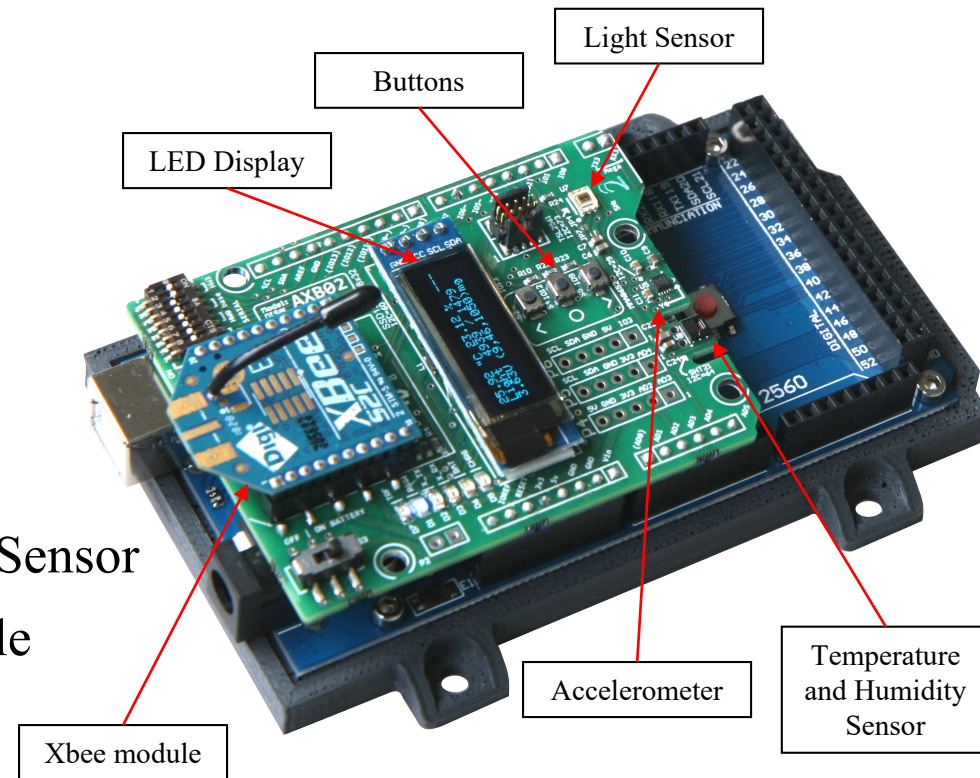
Hardware

- You will receive:
 - 2 Sensor Nodes
 - 1 Usb Cable
- Each sensor node is composed by:
 - 1 Arduino board
 - 1 Sensor shield
 - 1 Xbee module for communication



Hardware



- You will work with:
 - LED display
 - Buttons
 - Light sensor
 - Accelerometer
 - Temperature and Humidity Sensor
 - Xbee communication module



Software

- Use the Arduino IDE installed on the computers.
 - NOTE: if you are working in the EPFL computer rooms, open the IDE first and then open the files inside it
- Download and extract the .tar file with the code from Moodle

Programming an Arduino

- Connect the Arduino to your computer
- Select the correct *Board* and *Port* in the *Tools* menu
- Add the libraries you need the first time you use them (*Sketch/Include Library/ Manage Libraries*)
- Verify your code 
- Correct mistakes (if any), they will be shown in the console
- Upload code to the board 

Arduino code structure

- `setup()` → called once at startup or reset, used for initialization and configuration
- `loop()` → main body of the program. Keeps looping



```
Classic_Blink_LED | Arduino 1.0.5-r2
File Edit Sketch Tools Help
[Icons] Verify
Classic_Blink_LED $
const int LED = 13;

void setup ()
{
  pinMode(LED, OUTPUT);
}

void loop ()
{
  digitalWrite(LED, HIGH);
  delay(1000);
  digitalWrite(LED, LOW);
  delay(1000);
}

Done compiling.

Binary sketch size: 1,076 bytes (of a 32,256 byte maximum)
```

Arduino communication

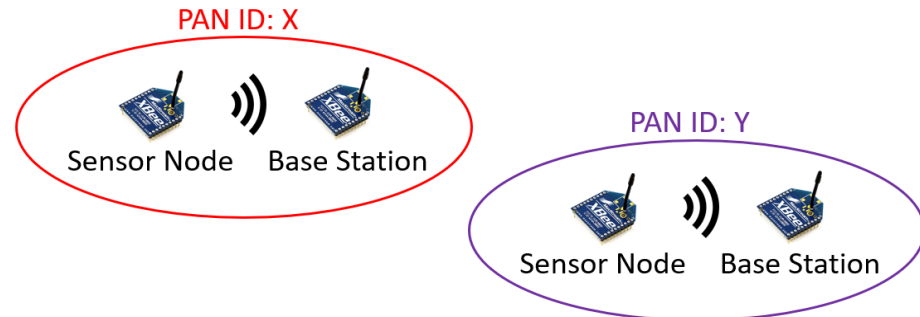
- Serial line communication via USB cable between the board and the computer (*Serial.print("Hello world/n");*)
- Baud rate (set to 9600 for this lab)
- Serial monitor and Serial plotter tools
- I2C communication between microcontroller and sensors (already implemented)

Xbee basics

- Low cost and easy-to-use modules for short-range communication
- Zigbee and 802.15.4 compliant
- Two nodes:
 - One used as transmitter (power it with the battery)
 - One used as a base station (connect it to the computer)
- Transmitter node: reads data from sensors and sends it to base station
- Base station: receives data from transmitter and sends it to the computer

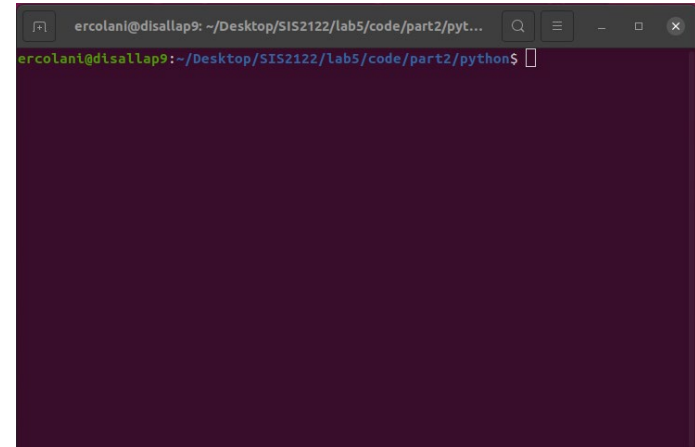
Xbee programming

- In this lab, the Serial3 line is used to communicate from the board to the Xbee.
- In part 3, set the same PAN ID in the transmitter and in the receiver to create a network (**use the number of your computer as PAN ID**)



Python scripts

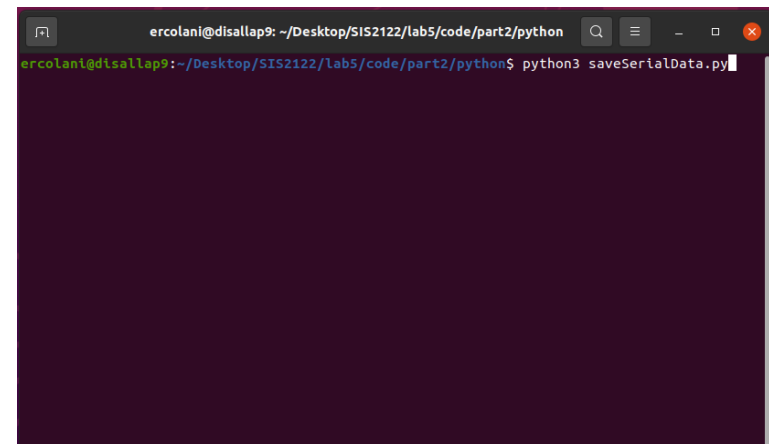
- We will use Python scripts to save the data on the computer.
- Open the terminal
- Run this command to install a missing library:
 - *pip3 install pyserial*

A terminal window with a dark purple background. The title bar shows the user 'ercolani@disallap9' and the current directory path. The terminal prompt is 'ercolani@disallap9: ~/Desktop/SIS2122/Lab5/code/part2/python\$' with a cursor at the end.

```
ercolani@disallap9: ~/Desktop/SIS2122/Lab5/code/part2/python$
```

Python scripts

- On the terminal, navigate to the python directory of the lab(in either parts 2 or 3)
- Use the “cd” and “ls” commands to navigate to the appropriate directory
- Launch the python scripts by running:
 - *python3 nameOfScript.py*



```
ercolani@disallap9: ~/Desktop/SIS2122/lab5/code/part2/python
ercolani@disallap9:~/Desktop/SIS2122/lab5/code/part2/python$ python3 saveSerialData.py
```

At the end of the lab

- Upload the code in the folder *maintenance* on both boards
- Ensure that the battery switch is off

Please fill out the feedback form for Lab 5!

Thank you!