Lab 5: Introduction to the Arduino Xbee Sensor Node

Lab: Getting familiar with sensor nodes

1 Testing the hardware/software setup

1. **(S)**: Now look at the board, what is happening?

   Solution: The sentence "Hello word" is displayed on the first line of the screen, followed by some dashes

2. **(S)**: Try pressing the buttons, what happens? Can you find the lines in the code that allow this behavior?

   Solution: The words "Welcome to Lab 5" appear on the screen. This behavior is coded with the if statements in lines 97-129

3. **(Q)**:Look at the code and in particular focus on the content of the functions setup() and loop(). Describe in detail what happens in each of them. What is initialized?

   Solution: The setup() function is used to initialize all the components that will be needed later. In this example, the I2C line (used to communicate with the OLED screen and the sensors) and the serial line (used to communicate with the computer) are initialized to allow communication over these two channels. The three buttons are initialized as inputs and the OLED screen is started and cleared. The loop() function contains the main body of the program, that will be repeated continuously. Initially, "Hello world" is printed on the screen, then either dashes or the sentence " Welcome to Lab 5" is printed on the screen depending on the status of the buttons. Finally, the sentence " Hello world!!" is printed on the serial line.

2 Local Sensing with one node

4. **(Q)**: Read the code and try to understand what is happening. What do you think will happen when you upload the code on the board?

   Solution: the code reads ambient and ir light data from the light sensor and prints it on the screen and on the serial line

5. **(S)**: Upload the code on the board (follow again the steps described in Part 1) and verify your assumptions.

   Solution: upload code and look at the oled screen

6. **(S)**: What gets printed on the serial monitor? How many samples do you get in 20 s? What is the frequency of data transmission?

Solution: the values coming from the sensor are printed on the serial line with the format " Tsl2561 (full / ir) = 628/ 138". The first value is the ambient light value, the second value is for the ir light. About 20 values in 20 seconds. Frequency is ~ 1Hz.

7. **(S)**: What do you see? What are these values? Why do you think that there are variations in the signal even when the light conditions do not change?

   Solution: The values are the full and ir light values. The variations are due to noise

8. **(S)**: Now orient the light sensor on the board towards a more intense light source and then cover it with your finger, repeat these actions a few times. How does the plot change? Write down some examples of values when the light is intense, low and medium range.

   Solution: The values increment closer to a brighter light and drop when a finger is placed on the sensor. Values are dependent on the room conditions

9. **(S)**: In the code, understand how the frequency of the data transmission in determined. Change the frequency to receive 1 sample every 5 seconds, upload the code to the board and validate your implementation by looking at the serial monitor. Now take a look at the serial plotter. How does the plot compare to the previous questions? What differences do you see in the plot when the light conditions change?

   Solution: To change the frequency, modify the value of SAMPLE_TIME:
   #define SAMPLE_TIME 5000
   The plot updates less frequently and is not able to detect rapid changes in light.

10. **(S):** Go to the folder *part2/python* and open the script *saveSerialData.py*. Check that the Arduino port name and baud rate correspond to the ones set in the Arduino IDE. What is the code doing? Open a Terminal and navigate to the *part2/python* folder. **Install** the pyserial library by writing "*pip3 install pyserial*" on the terminal. Run the script by writing "python3 saveSerialData.py" on the terminal. Check that the data is saved correctly

    Solution: Check that 10 samples are saved in a txt file.

11. **(I):** Now modify the python script to save 50 values and run it while also changing the ambient light conditions. Go to the part2/matlab folder and open the plotter.m script. Write some simple code to plot the data and compute some statistics about the ambient light. Run the script and report the data

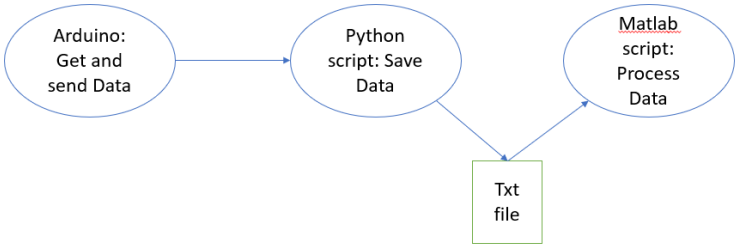
    Solution: To get 50 samples change line 12 of the saveSerialData.py script to:
    samples=50
    Matlab code to plot light data and compute some statistics
    %plot the data (you can, for example, use the function plot())
    plot(Data(:,1));
    hold on
    plot(Data(:,2));

```
legend('Full light','IR light')

%compute some statistics about the light in the room
averageFullLight= mean(Data(:,1))
averageIRLight=mean(Data(:,2))
maxFullLight=max(Data(:,1))
minFullLight=min(Data(:,1))
```

12. (Q): On a piece of paper, draw a small schematic of the information flow that you just explored with Matlab and Python. What are the advantages of using these tools compared to the Serial Plotter/Monitor?

Solution: The biggest advantage of using Matlab and Python is the versatility they provide. The Serial Plotter/Monitor have basic functionalities, but with Matlab and Python we can write code to exactly match our needs and we can use them for the whole data management pipeline, from acquisition to post-processing.



13. (I): Go to the loop() function and add a few lines of code to first check if the light value is lower than the threshold you selected and then send a warning message on the serial line when this happens. Upload your code to the board, open the serial monitor and check that, when your finger is placed on the sensor, the warning is triggered correctly.

Solution: Define a threshold variable and then check if the ambient light (full) is lower than the threshold:

```
if(full<threshold){
    Serial.print("WARNING\n");
  }
```

14. (S): Upload the code to the board and describe what happens. Which sensors are used?

Solution: Temperature and humidity sensor, light sensor and accelerometer are used. The values from these sensors are printed on the screen.

15. (I): Go to the loop() function. Add a few lines of code to allow the user to receive data from a sensor that they can select using buttons. You should print the value that the user is requesting by writing to the serial line and you should change the value of the variable btn to display on the screen which sensor data is sent back to the computer.
Change the value of the variable btn according to this schema:

- btn=”T” when the board is sending temperature and humidity values
- btn=”G” when the board is sending the accelerometer values
- btn=”L” when the board is sending light values

Verify your implementation by opening the serial monitor and checking that, when you press a button, the corresponding sensor value is sent on the serial line.

Solution:

```
if (digitalRead(BTN_BACK) == false)
  {
    btn="T";
    Serial.print("T: " + String(t)+ "C  H: "+ String(h)+" %\n");

  }else if (digitalRead(BTN_OK) == false)
  {
    btn="L";
    Serial.print(format("Light %5u /%5u\n",  full, ir));

  }else if (digitalRead(BTN_FWD) == false){
    btn="G";
    Serial.print(String("(")+String (accel.axis.x)+","+String (accel.axis.y)+","+String (accel.axis.z)+")mg\n");
  }
```

3 Remote Sensing with two nodes

16. (Q): Look at the loop() function. Can you explain what is happening?

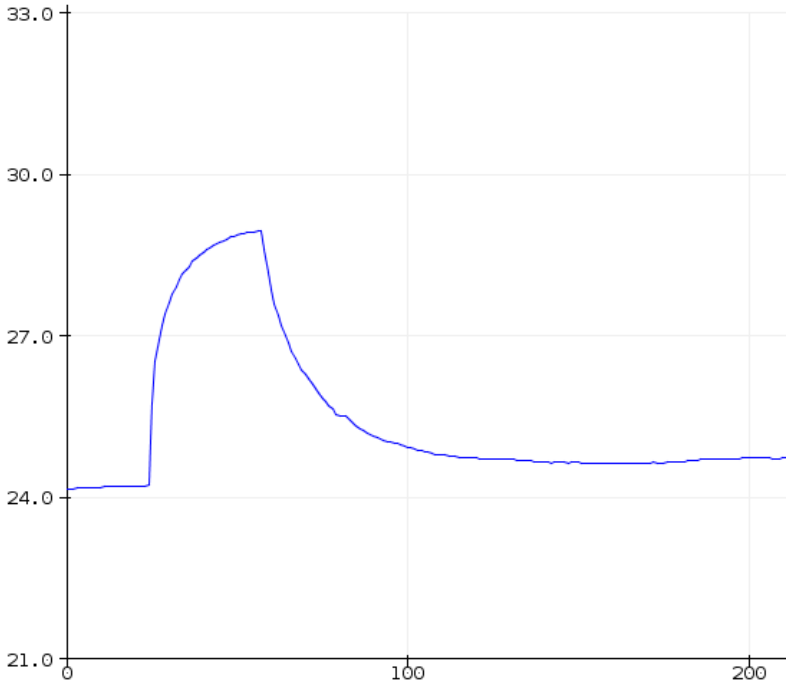
Solution: The loop() function checks if there are values on Serial3, reads them and writes them on Serial. This allows to print on the serial monitor the values that are coming from the transmitting Xbee.

17. (S): Keep the remote node on and connect the base station to the computer. Open the serial monitor (if the serial monitor does not open, try to select the correct Port of the receiver board from the Tools menu). It might take up to 30 seconds for the communication to start. What do you observe? Which sensor is used for data transmission? What is the frequency of transmission?

Solution: Temperature values coming from the transmitter board are received by the base station and printed on the serial line with a frequency of 1Hz

18. (S): Open the serial plotter and observe the data. Let the board send data for about 30 seconds, then place your finger on the temperature sensor and wait for about 30 seconds. Remove your fingers and wait for 60 seconds. How does the data plot vary? Is the sensor responsive? Is it noisy? How does it compare to the response time of the light sensor used in Part 2?

Solution: Temperature increases when a finger is placed on the sensor and decreases when the finger is lifted. However, the sensor is less responsive than the light sensor and it takes more time to adjust to the right temperature.



19. (S): In the transmitter.ino file, change the value of the SAMPLE_FREQ variable to allow faster sampling, upload the code to the remote node and repeat the experiment described in **S15** (pay attention to upload the code to the remote node and not to the base station board!!). Do you observe any differences in the plot? What would happen if you decreased the sampling frequency?

Solution: The response time is faster, and the sampled values transmitted to the base station vary faster. If the sampling frequency was slower, we would not be able to capture changes in the temperature values as precisely.

20. (S): Modify the script to collect 50 values of temperature. Launch the script and while modifying the temperature conditions for the sensor on the transmitter. Check that the data is saved correctly.

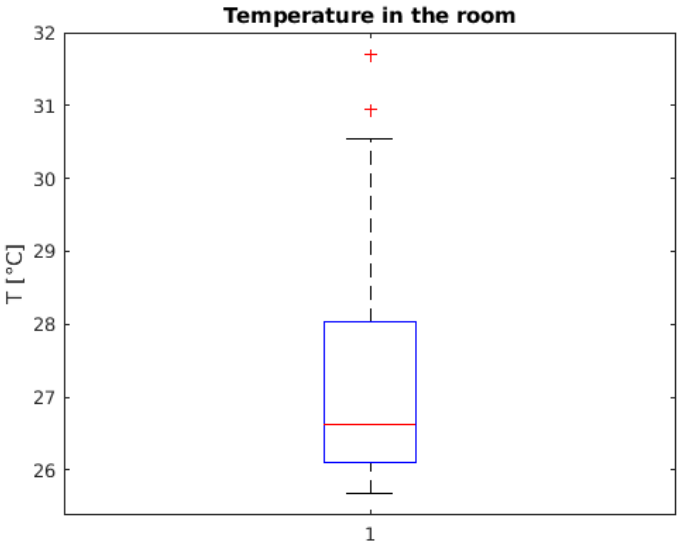
Solution: Change `samples` *to 50 and check that the data is saved correctly to a txt file*

21. (I): Open the script *cleanData.py* in *part3/python* and complete the code to check that all your data follows the expected format. You can then run the script and check that the data is cleaned and saved correctly to a new file.

Solution: rex=re.compile("^[0-9][0-9].[0-9][0-9]$")

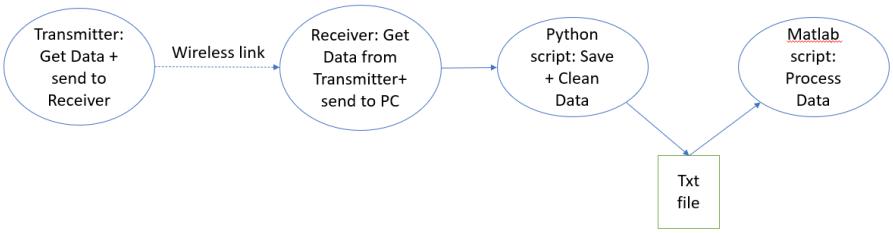
22. (S): open the script *plotter.m* in *part3/matlab*. Use visualization tools (such as boxplots) to draw conclusions on the temperature condition reported by your remote sensor. Report your findings.

Solution: boxplot(Data)



23. (Q): Draw the data flow in this scenario on a piece of paper. Can you imagine a scenario where this data flow could be useful?

Solution: This could be useful in many environmental monitoring applications, for example monitoring temperature in an experimental field over several hours or days.



IMPORTANT: What to do before handing the boards back to the TAs

Before handing the boards back to a TA, go to the folder *maintenance* and upload the code you find inside on both boards. This code reads the battery voltage and prints it on the OLED screen. This way, we can quickly turn the boards on and see if they need to be recharged. Thank you for your help!