Introduction to IoT: Autumn 2019

Exercise set: 3

Due on 25th September 2019 by 16:00.

Instructions: All course participants are requested to submit their exercise solutions (in English) electronically to the instructors Agustin Zuniga (agustin.zuniga at helsinki.fi) and Prof. Petteri Nurmi (petteri.nurmi at cs.helsinki.fi) by the due date. Use the following subject in your email: *IoT_week[#]_[last name_first name]_[student number], (i.e. IoT_week3_Zuniga_Agustin_12345)*

Your submission have to contain no more than **four (4)** single-spaced and numbered pages. Use font type Arial or its equivalent with size no smaller than 10 points. Include the exercise set number, your full name and student Id in the upper right corner of the first page.

In all the exercises, do not just give the answer, but also the derivation how you obtained it. Participants are encouraged to review course material to answer the problems and in some cases write computer programs to derive solutions.

Learning objective: In this set of exercises we will take a look at sensing in *IoT* environments. The exercises will help you to understand how to use sensors in IoT applications and briefly introduce how to read sensors data from smart objects using Python.

IoT Sensing and the Sensing Pipeline

Task 1 (4 pts.)

A startup company wants to use IoT for monitoring forest condition. To achieve this, the company uses two kinds of smart objects: (1) one monitoring air quality, including temperature, relative humidity, air pressure, gas and particulate matter sensors; and (2) one monitoring animal movements through infrared and sound sensors. Under normal operating conditions, sampling air quality once every ten minutes suffices as air quality does not abruptly change. However, simultaneous abrupt changes in gases (CO or NO_2) or particulate matter can reflect possible abnormal conditions and hence they are sampled every two minutes. For the second sensor, transit of animals is higher from 10 PM to 5 AM with around five animals moving around every ten minutes, while during the rest of the day transit decreases to around one animal per ten minutes.

1. Classify the sensors according to table *Ways to classify sensors* on slide 12, lecture 05: Sensing and Sensors.
2. Which duty cycle schemes do the sensors use?
3. Describe a sensing pipeline that uses the infrared and sound sensors to provide information about animal movements to the company. (a) What would be the input and output? (b) Is this classification or regression? Justify your answers to get the full points.

Task 2 (4 pts.)

Choose one IoT application and describe its sensing pipeline. Justify your answers to get the full points (refer to lecture 06: Sensing and the Sensing Pipeline).

1. What kind of intelligence would you use?
2. Which phenomena will your application sense and which sensors will it use?
3. How would you collect sensors data and ground truth labels?
4. Which machine learning/data science algorithm would you use to make it an "intelligent" application? Which kind of metric would you use to evaluate your model?

Acquiring data from smart objects' sensors

In this exercise we use a Sense HAT web-based emulator. You can access to the emulator at `https://trinket.io/python/c0587865b5`. Additionally, you can find the API Reference at `https://pythonhosted.org/sense-hat/`

Familiarise with emulator's interface by completing *Weather Logger* exercise in `https://codeclubprojects.org/en-GB/sense-hat/weather-logger/`. The fluctuation in emulated values allows us to study the effect of duty cycling in sampling.

Task 3 (2 pts.)

- Complete *Weather Logger* exercise:

  `https://codeclubprojects.org/en-GB/sense-hat/weather-logger/`.

- Set the temperature to 20°C.

- Modify *Weather Logger's* code to sense the temperature for 180 seg, while using the following duty cycling schemes: 100/3, 100/6 and 100/9.

  1. Plot the sampling curves corresponding to the three duty cycling schemes (include the plots in your submission). You can plot the values using your favourite tool. If you choose *display.py* for plotting, we recommend to use *pygal.XY()* instead of *pygal.line()*. Figure 1 shows an example of how would the plot look like.
  2. Compare the three sampling curves. How different duty cycling schemes affect sampling? (You would consider plotting the curves separately to facilitate the analysis.)

- **Hints**:

  a) Duty cycling schema is defined as $D = (T/P) * 100$. For example, if $D = 100/2$, $T = 1sec$ and $P = 2sec$. Additionally, D helps to determine the amount of samples to be read from the sensor. In the previous example, the sensor is sampled once every 2 seconds.

  b) Use the function *sleep()* to control reading rate and the *while* loop to control total sampling time.

  c) Run duty cycling schemes in parallel, which means that the three schemes will have the same value at different time, i.e. 9 sec, 18 sec, 36 sec.

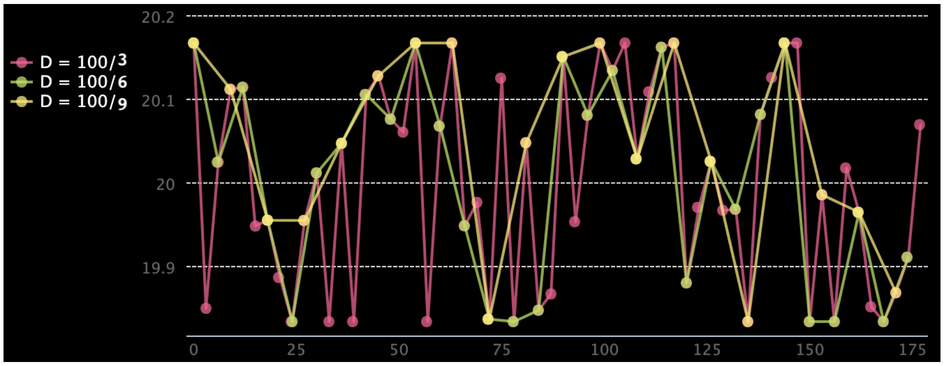

Figure 1: Output example

Bonus

Task 4 (bonus points)

Consider an air quality sensor that integrates a MOX gas sensor, a LSP particulate matter sensor, and environmental sensors measuring temperature and relative humidity. The MOX sensor requires heating to take measurements. Heating the sensor from sleep state requires 5 seconds, and it takes up to 10 seconds for the sensor to cool down. Both the values of the gas sensors and the values of the particulate matter sensor are corrected using a calibration pipeline that use external humidity and temperature as input. Correcting the measurements requires operating a microcontroller for 1 second period, during which heat is released into the sensor. What would be the optimal duty cycling period for these sensors and why?