Exercise 13: Communication with External Hardware

The aim of today's exercise is to learn the basics for communicating with external devices.

Exercise 13.1: (Preparation)

1. Write a script (see `01_doc_strings.py`) which prints the docstrings of the following builtin functions on the screen: `bin`, `hex`, `oct`, `ord`, `chr`, `int`.

   Background: Often, e.g. in task 3, you have to reinterpret data (bytes) (`'a'` $\triangleq$ `97` $\triangleq$ `0x61` $\triangleq$ `0b1100001`)

   Note: Use the `.__doc__` attribute in each case.

Exercise 13.2: Socket Communication (Chat Example)

1. Get an overview of `02_chat_client.py` and `02_chat_server.py`. In the client script, add the method calls `s.send(...)` and `s.recv(1024)` in the appropriate places. Start both scripts (server first!) and check the functionality.
2. Use the functions: `socket.gethostname()` and `socket.gethostbyname(...)` to find out your local IP address. In cooperation with your neighbor, modify client and server script to allow communication between both machines.

The following exercises were originally used int the "classroom" lecture with the Arduino based robot and a digital multimeter physically available. In the online version of the python course this is not possible. The result of your code thus cannot be tried out. However, it might still serve as a startig point for own projects.

Exercise 13.3: Communication with Arduino

1. Familiarize yourself with the structure and content of the `arduino_lib` module. Also refer to the C++ code for the Arduino in the `external_code` directory.
2. Expand `03_light_on.py` such that the LED lights for 10 seconds.
3. Make the LED light from interactive mode (IPython shell).
4. Extend `03_robo.py` such that the robot moves forward a bit, honks briefly, sends a reading to the computer, and finally moves back.

Exercise 13.4: Communication with Multimeter

1. Extend `04_measurement.py` to read and display a voltage measurement.