

POZNAŃ UNIVERSITY OF TECHNOLOGY

FACULTY OF CONTROL, ROBOTICS AND ELECTRICAL
ENGINEERING

INSTITUTE OF ROBOTICS AND MACHINE INTELLIGENCE

DIVISION OF CONTROL AND INDUSTRIAL ELECTRONICS



SERVER IoT
SENSE HAT

MOBILE AND EMBEDDED APPLICATIONS FOR
INTERNET OF THINGS

TEACHING MATERIALS FOR LABORATORY

DOMINIK ŁUCZAK, PH.D.; ADRIAN WÓJCIK, M.Sc.

DOMINIK.LUCZAK@PUT.POZNAN.PL
ADRIAN.WOJCIK@PUT.POZNAN.PL

I. GOAL

KNOWLEDGE

The aim of the course is to familiarize yourself with:

- available programming tools and libraries for the Sense Hat board,
- structure and components of Sense Hat board,

SKILLS

The aim of the course is to acquire skills in:

- operating sensors with digital interfaces using SBC (RPi),
- operating output devices (LED matrix) using SBC (RPi),
- operating input devices (joystick) using SBC (RPi),
- developing an IoT server application.

SOCIAL COMPETENCES

The aim of the course is to develop proper attitudes:

- strengthening the understanding of the role and application of network communication in IT systems and related security issues,
- strengthening understanding and awareness of the importance of non-technical aspects and effects of the engineer's activities, and the related responsibility for the decisions taken,
- proper technical communication in context of software development.
- choosing the right technology and programming tools for the given problem.

II. LABORATORY REPORT

Complete [laboratory tasks](#) as per the instructor's presentation. Work alone or in a team of two. **Keep safety rules while working!** Prepare laboratory report documenting and proving the proper execution of tasks. Editorial requirements and a report template are available on the *eKursy* platform. The report is graded in two categories: tasks execution and editorial requirements. Tasks are graded as completed (1 point) or uncompleted (0 points). Compliance with the editorial requirements is graded as a percentage. The report should be sent as a *homework* to the *eKursy* platform by Sunday, April 11, 2021 by 23:59.

III. PREPARE TO COURSE

A) KNOW THE SAFETY RULES

All information on the laboratory's safety instructions are provided in the laboratory and on Division website [1]. All inaccuracies and questions should be clarified with the instructor. It is required to be familiar with and apply to the regulations.

Attend the class prepared. Knowledge from all previous topics is mandatory.

B) INTRODUCTION TO SENSE HAT

Sense Hat is an add-on board for Raspberry Pi, created specifically for the Astro Pi competition [2]. The board allows to measure temperature, humidity, pressure and orientation, and display information using the built-in RGB LED matrix [3]. On the Raspberry Pi website, there is hardware documentation along with an electrical diagram and information about available programming tools and libraries [4]. The Sense Hat board is equipped with:

- **LSM9DS1** - integrated circuit containing 3D digital linear acceleration sensor, a 3D digital angular rate sensor, and a 3D digital magnetic sensor. The device measures linear accelerations in the range up to ± 16 g, magnetic fields up to ± 16 Gs and angular velocity up to ± 2000 dps. This sensor contains the I2C serial bus interface supporting standard and fast mode (100 kHz and 400 kHz) and the SPI interface [5].
- **HTS221** - compact relative humidity and temperature sensor. The measuring element consists of a polymer dielectric flat capacitor structure capable of detecting changes in relative humidity. It includes a built-in 16-bit analog-to-digital converter and I2C and SPI interfaces [6].
- **LPS25H** - digital barometer measuring pressure in the range of 260-1260 mbar with absolute accuracy of ± 0.2 mbar and a typical noise RMS of 0.01 mbar in high resolution mode. These pressures can be easily converted to altitude. The sensor is equipped with I2C and SPI interfaces [7].
- **ATTINY88-MUR** - 8-bit microcontroller from the AVR ATtiny family, equipped with I2C and SPI interfaces, operating at a maximum speed of 12 MHz. It is an intermediary element in controlling the LED matrix output and reading the joystick input [8].
- **CAT24C32** - 32Kb CMOS EEPROM internally organized as 4096 words of 8 bits each. It has a 32-byte page write buffer and supports the I2C interface (100 kHz / 400 kHz / 1 MHz). External address pins allow addressing of up to eight CAT24C32 devices on the same bus. This memory contains board configurations and is an essential element of extensions boards for Raspberry Pi [9].
- **LED2472G** - LED RGB matrix driver. The device is configured in 3 groups (red, green and blue) with 8 independently controlled channels. The LED current can be adjusted separately for each color in the range from 4 mA to 72 mA. This range is divided into two sub-ranges, and the current can be adjusted in each range in 64 resolution steps (6 bits per color) [10].
- **SKRHABE010** - 4-directional joystick with center monostable push-button.

The entire board, due to the use of the I2C bus, involves only 6 GPIO pins of the RPi connector.

C) SENSE HAT API

On the pythonhosted.org portal, is available documentation on the application programming interface (API) for the Sense Hat board implemented in Python with simple examples for every component [11].

Another popular programming tool for Sense Hat is RTIMULib (real-time IMU library) [12]. It is a C++ and Python library that facilitates the use of an inertial measurement unit (IMU) with 9 or 10 degrees of freedom, dedicated to embedded Linux systems (especially Raspberry PI and Intel Edison).

D) SENSE HAT EMULATOR

For the desktop versions of the Raspberry Pi OS system, the Sense Hat [13] emulator is available. In fig ref fig: emu. there is a screenshot from a shared virtual machine running the Sense Hat emulator. An API in Python is available for the emulator. Using the script to communicate with a physical device requires only changing the attached library:

- Emulator: `from sense_emu import SenseHat`
- Physical device: `from sense_hat import SenseHat`

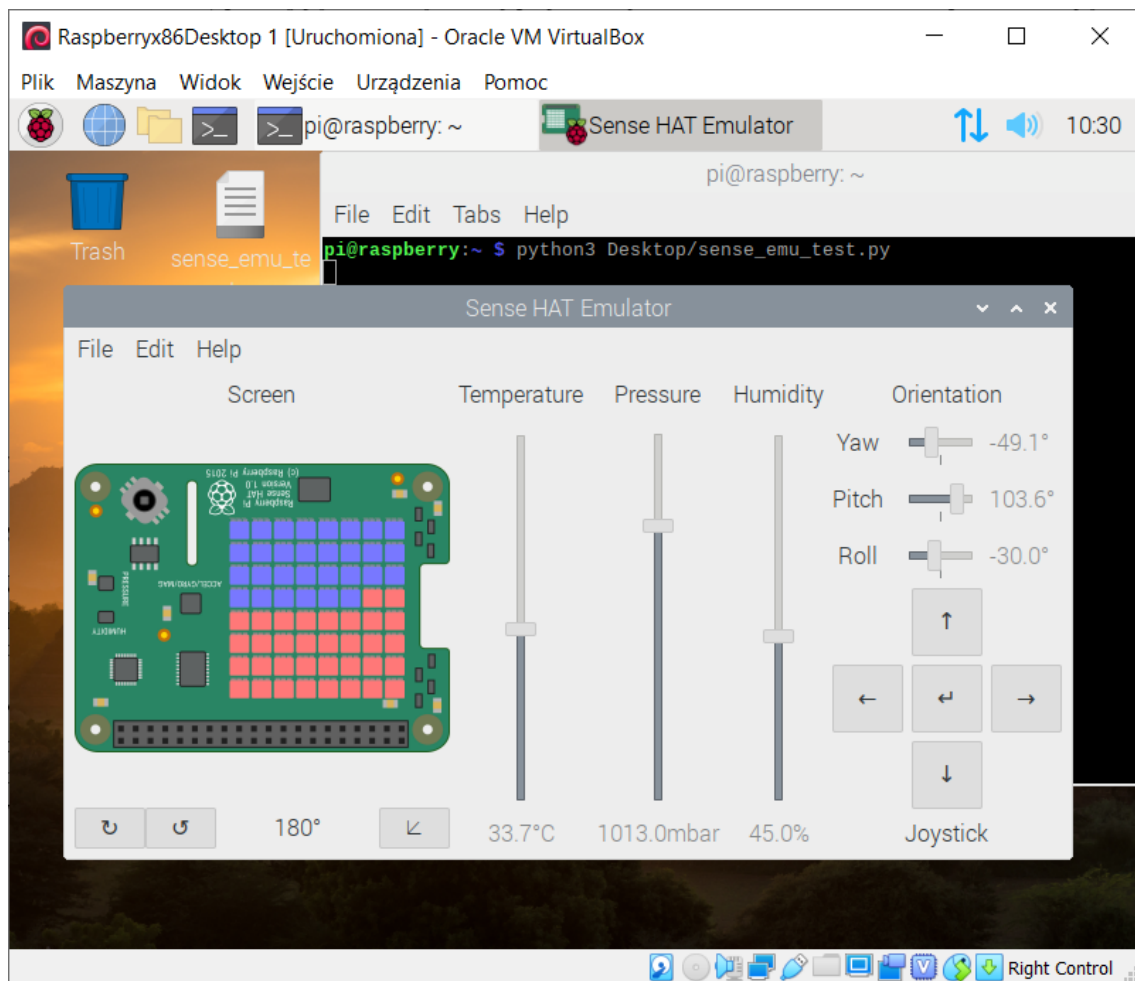


Fig. 1. Sense Hat emulator available on the virtual machine.

NOTE! To change the system boot mode from CLI to Desktop, run the command `sudo raspi-config` and then select the appropriate configuration in the **Boot Options** menu.

An online version of the emulator is also available, allowing to test programs directly in the browser [14].

E) JSON IN PYTHON

In Python there is available `json` API to encode and decode data - including custom, complex objects - using JSON notation [15]. In the listing 1. there is an example script for encoding the built-in data type *dictionary* (with syntax similar to JSON notation) and a simple class `EnvData` created by the user. Fig. 2. shows the result of successful script execution.

Listing 1. An example of encoding data in JSON notation with Python.

```
01. import json
02.
03. # synthetic data
04. temp_val = "20.5C"
05. press_val = "1023.0hPa"
06.
07. # use Python Dictionaries
08. print("Python_Dictionaries:")
09. press_name = "press"
10. temp_name = "temp"
11. dict_data = {temp_name : temp_val, press_name : press_val}
12. result = json.dumps(dict_data)
13. print(dict_data)
14. print(result)
15.
16. # use custom class
17. print("\nCustom_class:")
18. class EnvData:
19.     def __init__(self, temp, press):
20.         self.temp = temp
21.         self.press = press
22.
23. obj_data = EnvData(temp_val, press_val)
24. result = json.dumps(obj_data.__dict__)
25. print(obj_data)
26. print(result)
```



```
RaspberryPi_SSH_profile.tlp - pi@192.168.56.101:22 - Bitvis...
pi@raspberrypi:~/Desktop $ python3 json_dump_test.py
Python Dictionaries:
{'temp': '20.5C', 'press': '1023.0hPa'}
{"temp": "20.5C", "press": "1023.0hPa"}

Custom class:
<__main__.EnvData object at 0xb769bfcc>
{"temp": "20.5C", "press": "1023.0hPa"}
pi@raspberrypi:~/Desktop $
```

Fig. 2. Result of script from listing 1. execution.

IV. SCENARIO FOR THE CLASS

A) TEACHING RESOURCES

- Hardware
- computer,
 - Raspberry Pi single-board computer set,
 - Sense Hat add-on board,
 - microSD card,
 - microUSB 5V power supply,
 - RJ45 cable,
 - WiFi router,
 - *set of electronic components*,
 - *multimeter*,
 - *oscilloscope*,
- Software
- SSH client (e.g. Bitwise SSH Client),
 - application for writing binary disk images to external media (e.g. Win32DiskMaker),
 - text editor (Notepad++, VS Code),
 - VirtualBox application and virtual machine with Raspberry Pi OS for Desktop system,
 - Sense Hat emulator.

B) TASKS

1. Write a CLI application that allows you to read pressure, temperature and humidity sensors.
 - (a) The application should take input arguments: -p -t -h (all optional), each of them should allow selection measurement unit: hPa/mmHg, C/F, %/[number between 0-1].
 - (b) The presence of a given argument means that a given quantity should be measured and scaled to a given unit.
 - (c) The application should display the result in JSON format.
2. Write a CLI application that allows reading measurement from IMU (roll, pitch, yaw angles).
 - (a) The application should take input arguments: -r, -p, -y (optional) and -u (unit: radians or degrees),
 - (b) The presence of a given argument means that appropriate angles (roll, pitch and yaw) should be measured and scaled to a given unit.
 - (c) The application should display the result in JSON format.
3. Write the CLI application allowing you to control the LED matrix. The application should allow turning on the selected LED [x: (0-7), y: (0-7)] with the selected color (r, g, b) with input arguments.
4. Write a CLI application that allows you to read the state of the joystick.
 - (a) The application should contain a main loop, which can be left by pressing the appropriate (or any) key.
 - (b) The application should contain three software counters: counting: number of clicks on the center push-button, position in the x axis (right: increment, left: decrement) and position in the y axis (up: increment, down: decrement).
 - (c) Counters values should be written with a constant time step into a text file (e.g. joystick.dat). Apply JSON format.
 - (d) While the program is running, launch the 2nd terminal and check the contents of the file while interacting with the joystick.

5. Write a PHP script that allows you to view the measurements and content of the joystick.dat file.
 - (a) Create a PHP script that will call programs from tasks 1, 2 and 4.
 - (b) Display the result in the browser in the form of JSON format (i.e. aggregate objects).
6. Write a PHP script that allows you to set the color of the selected LEDs:
 - (a) Create an HTML form with 8x8 text inputs - see: *PHP instruction*.
 - (b) Use the POST method to send information to a PHP script.
 - (c) Control the appropriate LEDs based on the data from the form with program from task 3.
7. (*) This additional tasks requires a physical device. Modify the demo in:
`/usr/src/sense-hat/examples/RTIMULib/RTIMULibDrive11` so that:
 - (a) Information about the initialization result was redirected to the „rtimulibdrive11.log” file
 - (b) The next measurement results should be saved (overwritten) to the „rtimulibdrive11.dat” file in JSON format.
 - (c) Write a PHP script that allows you to display „rtimulibdrive11.dat”.
 - (d) Add input arguments to the program allowing you to choose which measurements are performed and saved to a file.

REFERENCES

1. *Regulations, health and safety instructions* [online]. [N.d.] [visited on 2019-09-30]. Available from: <http://zsep.cie.put.poznan.pl/materialy-dydaktyczne/MD/Regulations-health-and-safety-instructions/>.
2. *Home* [online]. [N.d.] [visited on 2020-03-25]. Available from: <https://astro-pi.org/>. Library Catalog: astro-pi.org.
3. *Getting started with the Sense HAT - What is a Sense HAT? / Raspberry Pi Projects* [online]. [N.d.] [visited on 2020-03-25]. Available from: <https://projects.raspberrypi.org/en/projects/getting-started-with-the-sense-hat/2>.
4. *Sense HAT - Raspberry Pi Documentation* [online]. [N.d.] [visited on 2020-03-23]. Available from: <https://www.raspberrypi.org/documentation/hardware/sense-hat/>.
5. *LSM9DS1* [online]. [N.d.] [visited on 2020-03-23]. Available from: <https://www.st.com/en/mems-and-sensors/lsm9ds1.html>. Library Catalog: www.st.com.
6. *HTS221* [online]. [N.d.] [visited on 2020-03-23]. Available from: <https://www.st.com/en/mems-and-sensors/hts221.html>. Library Catalog: www.st.com.
7. *Pololu - LPS25H Pressure/Altitude Sensor Carrier with Voltage Regulator* [online]. [N.d.] [visited on 2020-03-27]. Available from: <https://www.pololu.com/product/2724>. Library Catalog: www.pololu.com.
8. *ATTINY88-MUR Microchip Technology / Układy scalone (IC) / DigiKey* [online]. [N.d.] [visited on 2020-03-23]. Available from: <https://www.digikey.pl/product-detail/pl/microchip-technology/ATTINY88-MUR/ATTINY88-MURCT-ND/3789345>.
9. *CAT24C32: EEPROM Serial 32-Kb I²C* [online]. [N.d.] [visited on 2020-03-23]. Available from: <https://www.onsemi.com/products/memory/eeeprom-memory/cat24c32>.
10. *LED2472G* [online]. [N.d.] [visited on 2020-03-23]. Available from: <https://www.st.com/en/power-management/led2472g.html>. Library Catalog: www.st.com.

-
11. *API Reference - Sense HAT* [online]. [N.d.] [visited on 2020-03-23]. Available from: <https://pythonhosted.org/sense-hat/api/>.
 12. *RPi-Distro/RTIMULib* [online]. RPi-Distro, 2020 [visited on 2020-03-25]. Available from: <https://github.com/RPi-Distro/RTIMULib>. original-date: 2015-06-11T08:51:00Z.
 13. *Sense HAT API — Sense-Emu 1.0 documentation* [online]. [N.d.] [visited on 2020-03-25]. Available from: <https://sense-emu.readthedocs.io/en/v1.0/api.html>.
 14. *Trinket: run code anywhere* [online]. [N.d.] [visited on 2020-03-25]. Available from: <https://trinket.io/sense-hat>. Library Catalog: trinket.io.
 15. *json — JSON encoder and decoder — Python 3.8.2 documentation* [online]. [N.d.] [visited on 2020-03-30]. Available from: <https://docs.python.org/3/library/json.html>.