1.Rositsa MAKSIMOVA, 1.Krassimir KOLEV

MODERN APPROACHES WITH PYTHON IN SENSORY FUSION

1.University of Food Technologies, Technical Faculty, Plovdiv, BULGARIA

Abstract: The work presents real-time data acquisition and monitoring for environmental sensors. This paper presents data fetching and processing implemented by modern approaches of Python. A functional diagram of a materialized system and algorithmic diagrams for Python 3 implemented programs are given. The functionality of the programmed system has been checked by testing the implemented program structures and performing experiments under different environmental conditions with actual results obtained.
Keywords: environmental monitoring, sensory fusion, Raspberry Pi, PiJuice, Sense HAT, Python

1. INTRODUCTION

Data collection and accuracy are an important aspect in each type of science research. The concept of open source has been applied to the overall realization and use of a computerized system for collecting environmental data. The hardware platform Raspberry Pi 3 [2] and two expansion modules - PiJuice [5] and Sense HAT [6] are chosen to materialize such a system.

The purpose of assembling a data collection architecture is to obtain the information through a primary source - directly through the set of sensors embedded in the Sense HAT module. The information received from the system sensors is recorded in the form of tabular data (spreadsheets) which can then be processed manually or automatically by programs (scripts) programmed in the Python language. Drawing of graphs based on the collected and recorded data is done by available free Python libraries [4] designed and written by enthusiasts of the language. The freedom provided by open source and Python 3 [1] allows to realize a system that is useful for exploring different environmental conditions, not only through data that is recorded and stored locally, but also through remote wireless access via mobile devices.

2. MATERIAL AND METHODS

— Hardware overview

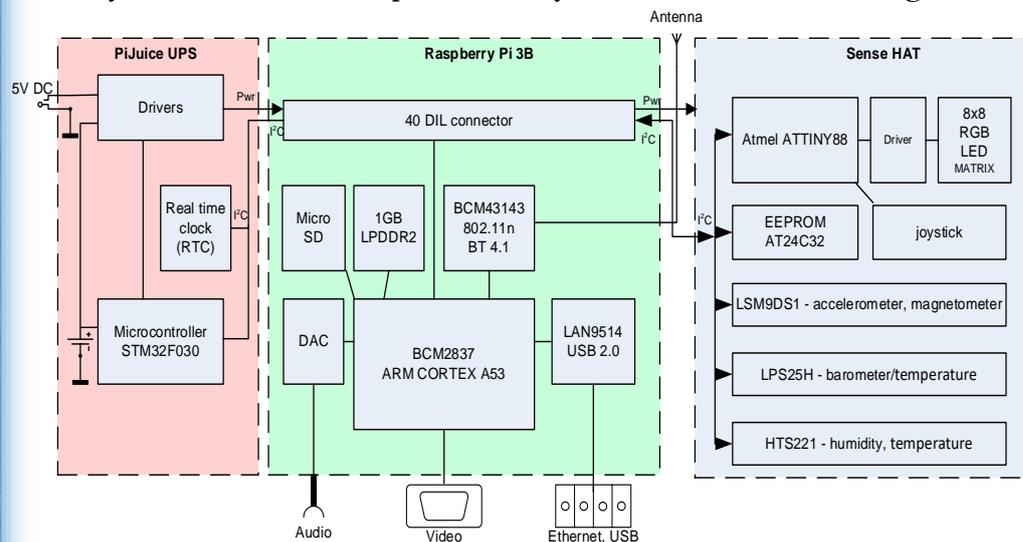The system hardware is implemented by the structure shown in Figure 1.

Figure 1. Structure diagram of the system

The hardware of the system consists of the three main modules - Raspberry Pi 3B, PiJuice UPS and Sense HAT, Drone DJI and main companion components such as smartphones, an SD (Secure Digital) card, power adapter, batteries, cables, and computer peripherals are also involved in the implementation of the system and its equipment [3].

The expansion module Sense HAT for Raspberry Pi, which is a fundamental part of the Astro Pi mission [7], provides temperature, humidity, pressure and orientation measurements, visually outputs via its built-in 8x8 RGB LED matrix and control by a 5-button joystick. Integrated circuit based sensors can be used for different types of experiments, applications and even games. The available IMU (Inertial Measurement Unit) chip in Sense HAT includes a set of three sensors in one that capture movements - a gyroscope, an accelerometer and a magnetometer. The joystick can also be used as an opportunity for user interaction with the executed programs which is implemented in the system. The purpose of Sense HAT in the current system is to collect data from its built-in sensors. The module can be attached to the 40-pin GPIO block of RPi but in this system it is attached to the PiJuice's extension block. The functional diagram of the system is given on Figure 2.
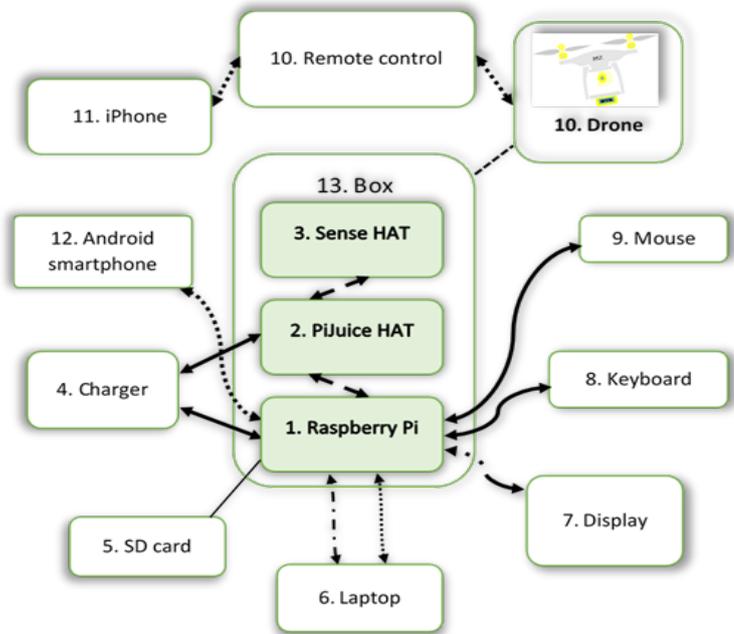
Bold font indicates the main participants in the system. Linking lines in different styles indicate the type of the connections (of information/energy exchange) between the hardware components as follows:

Figure 2. Functional diagram of the system

- Wi-Fi transceiver connection;
- Mechanical connection;
- 40-GPIO-link;
- USB or micro USB connection;
- MicroSD (Secure Digital) connection;
- Ethernet with RJ45 connection;
- HDMI-DVI connection.

The assembled state of the system is depicted in Fig. 3, where the 3-module box is presented. In the figure is shown the way that the Sense HAT LED display illuminates each time the system is being switched on.

── Software overview

We use the officially supported operating system of the Raspberry Pi Foundation the Raspbian GNU/Linux 9 (stretch) version. It contains software programs for training, programming

Figure 3. Assembled hardware of the system

and general use (Python, Scratch, Java, Mathematica and others). The great advantage of the computer RPi - the capabilities it offers along with its Raspbian operating system are communication interfaces such as the SSH (Secure Shell) protocol and VNC (Virtual Network Computing) using RFB (Remote Frame Buffer) protocol. These may allow both remote wireless access to the PC without need for a mandatory connection of RPi to a local network with the device to communicate with, as well as a wired connection. In both cases the IP address of RPi is necessary. In our system both interfaces are enabled and other third-party communication programs are used
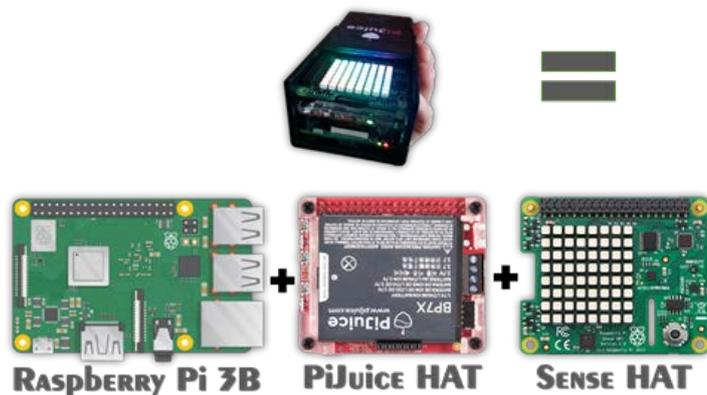
too. Except direct connection, remote connection is provided free of charge thanks to RealVNC cloud service, underwriting for educational or non-profit purposed remote access only. Cloud connections are convenient and with end-to-end encryption, which is a design framework in computer networks. In networks designed in accordance with this principle the application-specific features are located in the network termination nodes instead in intermediate nodes, such as gateways and routers that exist for the creation of the network. They are highly recommended for connection to Raspberry Pi over the Internet. There is no firewall or router reconfiguration and no need to know the IP address of the Raspberry Pi computer or provide a static one. Remote connection is available free after signing up for the official website of VNC, (www.realvnc.com/en/raspberrypi/#sign-up) [3].

For users owning Raspberry Pi without Sense HAT it is designed a special Sense HAT Emulator, which works naturally on Raspberry Pi's desktop rather than a browser and it is included in Raspbian's standard installation package in the main menu. There is also an online version of an emulator in Trinket which is developed in the United States as a tool for code sharing from any device in any browser and works instantly without the need to log in, download plugins or install additional software. To convert the code from emulator to a real physical device Sense HAT it is necessary a replacement of sense_emu with sense_hat everywhere in the source code. This usually happens when a program is transferred to a physical Sense HAT module.

When programming it is useful to keep in mind that the computer program resembles a recipe of a list of ingredients - data in variables and a list of instructions - functions or methods. Following the instructions correctly described one by one using the variables, the computer "prepares the dish" of the user. It is a good practice in programming to use an integrated development environment (IDE). There are many programming languages and they all offer something special. Python is a challenge for today's computer engineer. Its syntax is understandable and logical, allowing "programming at the speed of thought." Being a high-level language used for industrial, medical, scientific and other purposes it is ideal for both beginners and advanced. Currently there are two versions - Python 2 and Python 3, as the 3.x releases are different from 2.x and are not directly mutually compatible. Raspbian comes with a version of Python 3 so it can be started programming as soon as it is installed.

Python IDLE (Integrated Development and Learning Environment) provides REPL (Read-Evaluate-Print-Loop) so that the environment prompts for Python commands and has built-in syntax for autocompleting. Some languages like Java and C# use curly brackets to enclose a block of code which is a main feature missing in Python and instead it uses indentation to indicate the belonging of the respective block of code. Python is a cross-platform language which means you can write a program on a Windows or Mac computer, save the program and run on Raspbian in RPi. PyCharm IDE is an integrated development environment used in computer programming, specifically for Python. It offers code analysis, graphical debugger, integrated tester, integration with version control systems (VCSes) and support for web frameworks like Django.

3. RESULTS AND DISCUSSION

The configuration of the system is preceded by its programming except for a few additional settings that are made after the actual implementation of the programs. One of these is setting up the user scripts from the PiJuice GUI menu which must be preceded by the programmed scripts.

There is no real-time clock (RTC) in Raspberry Pi, so in our system the built-in RTC of the PiJuice module is configured and used. The RTC is supported by the PiJuice battery and uses the I2C which allows Raspberry Pi to communicate with many different devices at the same time.

By default, the EEPROM address in PiJuice is 0x50 and the RTC driver is automatically loaded when Raspberry Pi starts, but in our system, due to the additionally attached Sense HAT module on PiJuice, the EEPROM address is changed to 0x52 address, which requires manual loading of the driver each time the system is started. This is implemented by editing the config.txt file of Raspberry Pi. The system time of Raspberry can be set up in two ways if the Raspbian time zone is already set - when the Internet connection is automatically set up after booting and thus syncing with RTC time or manually with commands in the Terminal. After each reboot of the system, it is necessary to copy the RTC time back to the system clock. Thus even after system restart the RPi system time is read from the time of the RTC. All of the busy addresses by EEPROM, sensors and the RTC on the pair of I²C buses 0 and 1 of Raspberry Pi can be visualized by using both of the commands sudo i2cdetect -y 0 and sudo i2cdetect -y 1 in the RPi's Terminal, where UU is displayed at 0x68 for the RTC address shown in Figure 4.

Figure 4. Visualization of the pair of I²C buses of Raspberry Pi

The application software consists of several scripts alongside of a graphical user interfaced program, which combines the action of all scripting programs and provides visual management of the system. The program files are organized into special directories given in Fig. 5 as a tree structure.
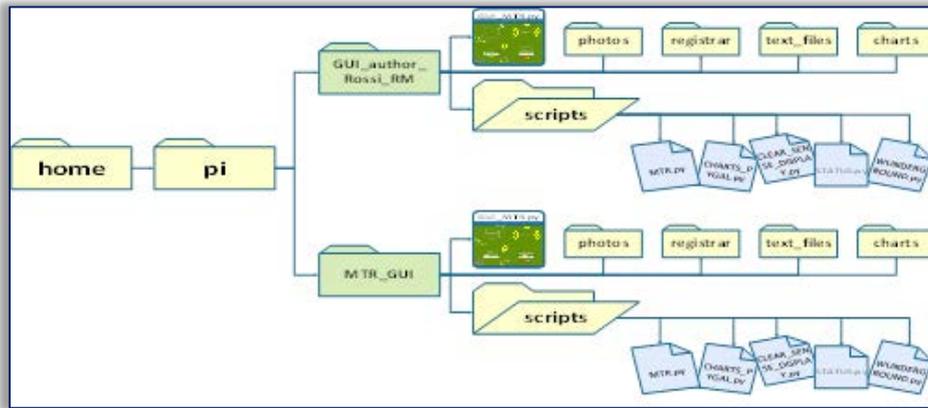

Figure 5. Directory organization

The MTR_GUI folder contains the source code using absolute or full paths, and the GUI_author_Rossi_RM folder contains the same scripts as MTR_GUI but with relative paths. All scripts are programmed in Python 3 IDLE on Raspbian and in PyCharm installed on Windows 10 Education.

The first script of our system, named as MTR.py, performs the actual data fetching by the system via Sense HAT's pressure, humidity, temperature and orientation sensors. In addition to these data, the relative altitude is calculated on the basis of the measured pressure and temperature values using a mean sea level pressure correction known as QNH in aviation and used by pilots as a secondary value for altitude information during flying. This factor is calculated and available as a value in the constantly updated information of each general airport - METAR. Each time before measurement the value of the QNH factor of the nearest airport to our location is taken via third party METAR-based mobile applications. All of the collected data is being saved as a spreadsheet in a .CSV file, which is being uniquely named each time the program is started, due to the fact that the system time is being called. Other data recorded in the file is collected by Sense HAT's get_accelerometer_raw() method, which measures the gravitational acceleration or G, acting on each of the three axes (x, y, z). For G > 1, the value of G is recorded along with the axis on which the value is measured, for

G <= 1, then G is saved as "Normal: G = 1". The purpose of this measurement by the system is to detect possible system shocks with force above the standard gravity force of G = 1.

Maintaining a reasonably sized script is a basic programming goal, as this facilitates the subsequent modification and management of the code. With Python 3 this goal is being achieved by using modules or libraries.

A module can be imported by specifying its name preceded by the import keyword, and it is good practice to do it at the very beginning of the code. After importing the module names into the script, all module methods become available within the new script. Thus a script can call up the functions of another script and thus communication between two or more scripts is provided. The modules in Python can be either standard from the Python's standard functional library set included in the standard language installation package, or third-party modules (for example: sense-hat, pygal, requests etc.) written by keen users of the language. An algorithmic diagram for the logic of the MTR.py program is given in Fig. 6.
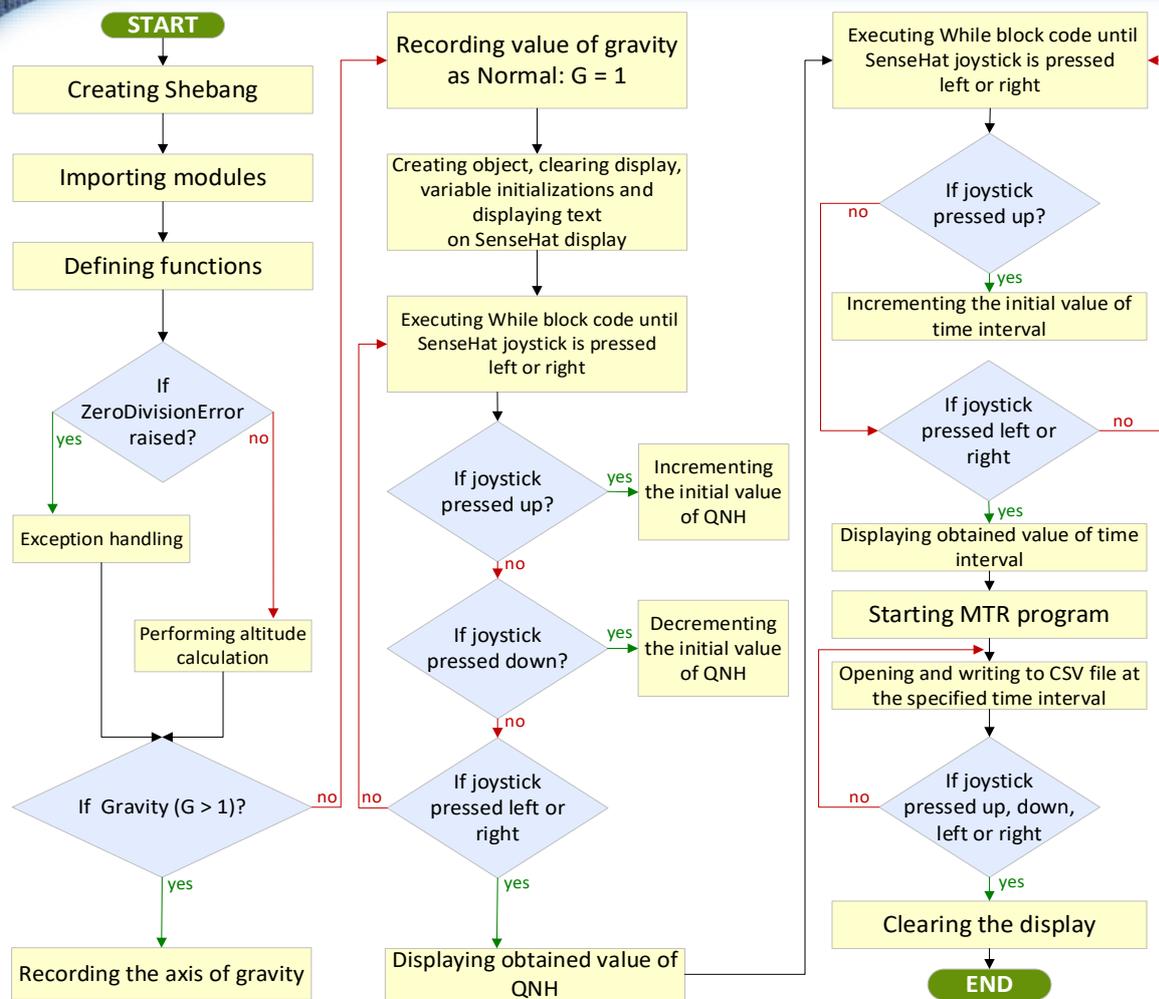
Figure 6. Algorithmic diagram of the MTR program

The modules imported in the MTR.py script are:

- ≡ import sys
- ≡ from sense_hat Import SenseHat
- ≡ import csv
- ≡ from csv import writer
- ≡ from time import sleep
- ≡ from datetime import date, datetime, timedelta

Thanks to the CHARTS_PYGAL.py program, the collected and saved data is processed converting the data from the CSV file to a graphical and text formats. The script reads the name of the CSV file as an input without the name extension and draws separate graphs of the measured values from each of the sensors - pressure, temperature and humidity alongside of the calculated altitude during the recording, common chart of all measurement and calculation values together, as well as the map of the world in which the data collection is processed. The program also converts each column of the spreadsheet from the CSV file into a separate text file to a specific directory. This is done in order to make raw data available for subsequent data processing. The third-party library pygal is used. An algorithmic diagram for the logic of the CHARTS_PYGAL program is given in Fig. 7. The modules imported in the script are:

- ≡ import sys
- ≡ import os
- ≡ import csv
- ≡ import pygal
- ≡ from pygal.maps.world import World
- ≡ from pygal.style Import Style
- ≡ from datetime import datetime

Each script can be run through a specially created text files with the .DESKTOP extension on the RPi's desktop. However almost every operating system includes some kind of a graphical user interface (GUI) to allow users to input data and see the results. Graphical programming handles a specific set of terms. For example, the primary area of the GUI is called a window, and the area in the window is frame. The frame may contain various widgets such as text boxes, buttons, sliders etc. that the program uses to interact with users. Each widget has its own set of properties that determine how it is displayed in the window and what features it performs. The implementation of various functions is preceded by the concept of event handling. Thus the graphical application does not have a sequence of tasks but instead relies on a set of functions (methods) that are executed individually in response to a particular event.
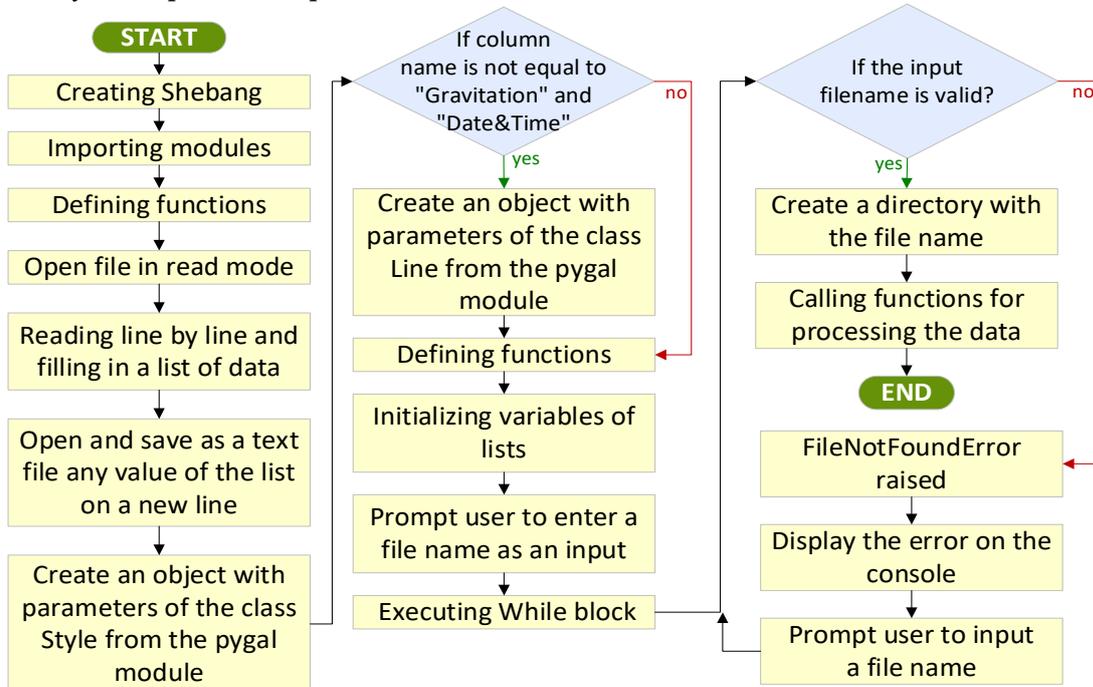
Figure 7. Algorithmic diagram of the CHARTS_PYGAL program

In order for the program to work it is necessary to create separate functions that Python calls when an event occurs. The standard tkinter library of Python is one of the most popular for GUI programming. Since Python includes the standard tkinter library in its default library set, it is typically used to create graphical Python programs in Raspberry Pi too. The GUI_MTR program (Fig. 8) in our system is written in Python 3 using the third-party guizero library, which is a private case of the tkinter library.

The GUI_MTR window is mentally divided into 4 sections where the individual scripts are located and performing. Thus the MTR program is located and started form the first section, whereby the value of the QNH and the time interval are being set by sliders. The third

Figure 8. Main window of the GUI_MTR program

section is dedicated to the CHARTS_PYGAL graphics drawing program which has a dropdown menu for selection of the country where the data acquisition takes place thanks to the pygal-maps-world library. Currently the country of Bulgaria is selected. Below the dropdown menu there is a text box where you enter the name of the CSV file without its extension in which the data is stored. After starting the program from the specified button, the program automatically creates a directory containing text files, graphics and map of the world with .TXT, .PNG, and .SVG extensions. Environmental data obtained by MTR system are shown in Fig. 9.
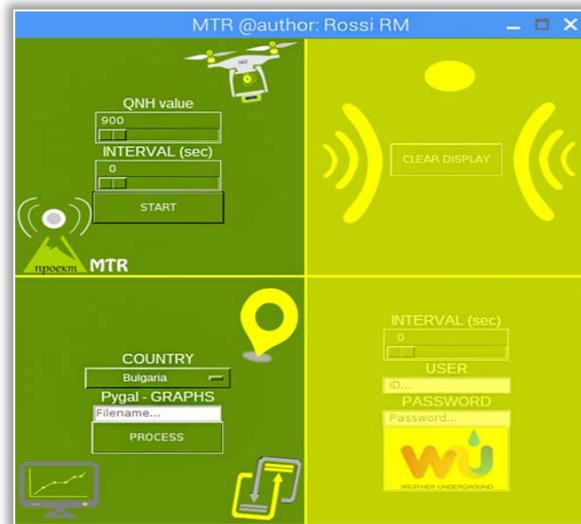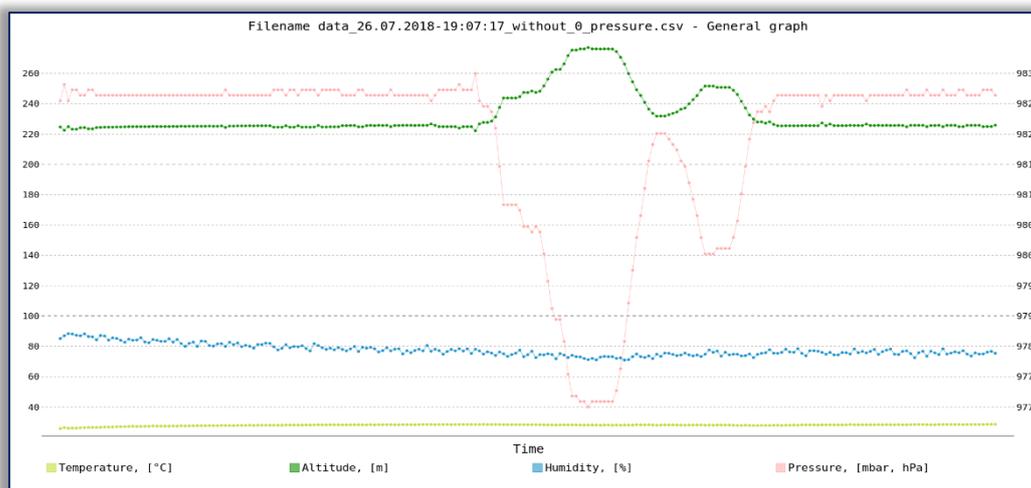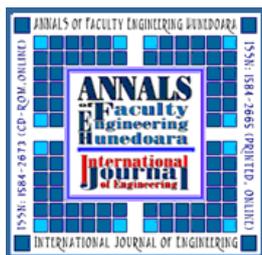
Figure 9. Environmental data obtained by MTR system

4. CONCLUSIONS

The assembled hardware of the mentioned data acquisition computer system fetching a whole range of direct sensory information about the immediate environment can be used to conduct personal and scientific experiments in various scientific fields, due to the fact that all processes are automated by programs in Python 3. In case a specific set of sensor data is required, the Python language easily provides the flexibility of the code for subsequent modifications. The ability for remote access from each Internet connected node of the world provides the freedom and the sense of IoT with no need for a local area network. The advantage of our system over other ready systems is the ability and flexibility of its programming code. The fact that this system can be adapted to different needs allows it to be used not only in the world of computer technology but also in interdisciplinary science fields where various fundamental natural processes can be monitored through the available sensors. Main application of developed system is to be used for monitoring and data acquisition with environmental sensors. The presented paper provides results of the master thesis afterwards published as a book of Rositsa Maksimova with the scientific coordination of Assoc. Prof. Krassimir Kolev.

Note: This paper is based on the paper presented at IIZS 2019 – The 9th International Conference on Industrial Engineering and Environmental Protection, organized by Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, in Zrenjanin, SERBIA, in 03–04 October, 2019.

References
[1] Blum R., C. Bresnahan, Teach Yourself Python Programming for Raspberry Pi in 24 Hours, SAMS, USA, 2016.
[2] Charleston T., RASPBERRY PI 3:2016 USER GUIDE FOR BEGINNERS, CreateSpace Independent Publishing Platform, 2016.
[3] Maksimova R., Open source computer system for data acquisition – MTR, Saypres OOD, Plovdiv, 2019. (in Bulgarian)
[4] Mounier F., pygal Documentation Release 2.0.0, pygal.org, 2016.
[5] PiJuice HAT - A Portable Power Platform For Every Raspberry Pi – Pi Supply, website: https://uk.pi-supply.com/products/pijuice-standardBest
[6] Sense HAT - Raspberry Pi, website: https://www.raspberrypi.org/products/sense-hat/
[7] The European Astro Pi Challenge, website: https://astro-pi.org