Journal of Universal Computer Science, vol. 26, no. 11 (2020), 1495-1512
submitted: 2/9/20, accepted: 2/10/20, appeared: 28/10/20 *CC BY-ND 4.0*

Weather Station IoT Educational Model Using Cloud Services

Ján Molnár, Simona Kirešová, Tibor Vince, Dobroslav Kováč, Patrik Jacko
Matej Bereš, Peter Hrabovský
(Department of Theoretical and Industrial Electrical Engineering, FEI TU Košice, Slovakia
jan.molnar@tuke.sk, simona.kiresova@student.tuke.sk, tibor.vince@tuke.sk
dobroslav.kovac@tuke.sk, patrik.jacko.2@tuke.sk, matej.beres@tuke.sk
peter.hrabovsky@tuke.sk)

Abstract: IoT technology is gaining more and more popularity in practice, as it collects, processes, evaluates and stores important measured data. The IoT is used every day in the work, in the home or smart houses or in public areas. It realizes the connectivity between real world and digital world which means, that it converts physical quantities of the real world in the form of analog signals into digital numbers stored in clauds. It is essential that students gain practical experience in the design and implementation of the IoT systems during their studies. The article first describes IoT issues and communication protocols used in IoT generally are closer described. Then the design and implementation of an educational model of IoT system – Weather station with the ThingSpeak cloud support is described. The created IoT model interconnects microcontroller programming, sensors and measuring, cloud API interfaces, MATLAB scripts which are useful to analyses the stored data, Windows and Android application developing.

Keywords: ATmega2560, IoT, Education, Data Analysis, MATLAB, sensors, ThingSpeak, weather station, ESP8266.
Categories: B.1, B.4, C.2.2, H.4, J.0, K.3, L.2

1 Introduction

The current scientific and technical progress in the world offers ordinary people opportunities which they would not be able imagine in the past. Automation, the beginning of artificial intelligence or the closely related IoT technology - the Internet of Things - represent the latest technological advances. Rapid technical progress requires a quick acquaintance with new products and their implementation into everyday life. The same progress is required in the education so that graduates have the latest knowledge in the field of technical development, which they would be able to use in the practice. Due to these facts, the article is focused in the modern technology using for teaching tools.

One of the current technologies, which is becoming increasingly popular is IoT technology, which is used for the purpose of connecting and exchanging data with devices connected via the Internet. Based on received data, it performs various operations, analyzes results, compares them and evaluates them in the graphs or the other output forms. The IoT is used every day in the work, people used it in the home or smart houses or in public areas. The IoT realizes the connectivity between real world and the digital world. This connection is realized through sensors and actuators, which

main task is an external analog signal capturing. The captured signals could be processed or stored on the cloud [Garg, 19].

Clouds are currently a powerful tool associated with IoT, as it provides services stored on remote servers for anyone with access from anywhere. Cloud systems provide many services. From simple ones, where the user can upload his data to a server, which serves as a storage to which he connects from any computer, to many complex services. These include functions where the user collects data from various measuring systems, stores them on a cloud which processes and evaluates the data. The ThingSpeak is one of such platform. This open-source platform connects physical realized parts, such as a microprocessor or microcontroller included sensors, with the virtual part. The data captured by the sensor are then transmitted to the ThingSpeak platform, which analyzes, evaluates and displays results on the ThingSpeak website. The processed data are then possibly stored publicly or privately [Razali, 20]. The ThingSpeak possibilities are closely described in chapter 3.

The main goal of this article is to describe the design and implementation Weather Station IoT educational model using cloud services. This educational model aims to assist in the teaching of IoT technology through the proposed weather station.

Students gain an overview and a detailed procedure for solving problems of similar projects. At the same time, they will get acquainted with IoT applications, with their functionality and use in real projects. Of course, it should be noted that the control unit represented in this case by the ATmega2560. It is a well known alternative for students, but in the future, in the implementation of the device, the ATMega2560 may be replaced by another microcontroller such as STM32, Texas Instruments microcontrollers, NXP Semiconductors or single board computer platforms such as Raspberry Pi or other alternative.

2 IoT

2.1 IoT description

There are many organizations such IEEE, IETF, W3C, NIST, ITU, ETSI, OASIS and other with their own description or definition of Internet of Things. Possible reason, why there isn't international widely accepted definition of IoT could be the fact that IoT revolution is still not finished and no one know what there will be in the next decades.

Currently, IEEE has one project related to IoT, IEEE P2413TM which is draft standard defining an architectural framework for the Internet of Things, including descriptions of various IoT domains, definitions of IoT domain abstractions, and identification of commonalities between different IoT domains. An IEEE P2413TM is currently considering the architecture of IoT with three layers: Applications, Networking and Data, Sensing. IEEE P2413 also currently posits the extent of an IoT market and the stakeholders of IoT as represented in Figure 1 [Minerva, 15].

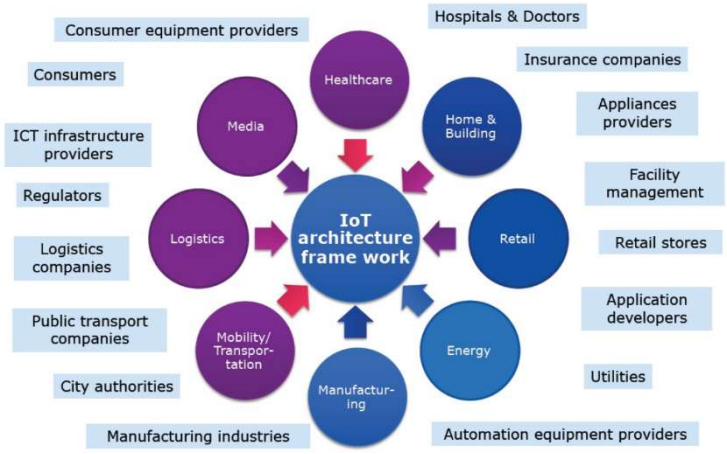

Figure 1: IoT markets and stakeholders

For instance the OASIS described Internet of Things as system where the Internet is connected to the physical world via ubiquitous sensors [Cosgrove-Sacks,2014], where OASIS describes the ubiquity of sensors as existing in every mobile, every auto, every door, every room, every part, on every parts list, every sensor in every device in every bed, chair or bracelet in every home, office, building or hospital room in every city and village on Earth.

Taking into the account existing definitions, it is possible define IoT also as a system of interconnected devices, machines, animals, people that have a unique ID and have the ability to exchange data over the Internet, without human intervention [Lörinc, 20].

2.2 IoT protocols

The IoT covers a huge range of industries and use cases that scale from a single constrained device up to massive cross-platform deployments of embedded technologies up to cloud systems connecting in real-time. Tying it all together are numerous legacy and emerging communication protocols that allow devices and servers to talk to each other in new, more interconnected ways. For this purpose, different protocols of different levels and layers with different properties are used.

It is possible to group protocols into following layers to provide some level of organization instead of existing architecture models like OSI Model [Harwood, 20]:

1. Infrastructure: 6LowPAN, IPv4/IPv6, etc.
2. Identification: uCode, IPv6, URIs, etc.
3. Communication / Transport layer: Wifi, Bluetooth, LPWAN, etc.
4. Discovery: mDNS, DNS-SD, etc.
5. Data Protocols: MQTT, CoAP, AMQP, etc.
6. Device Management: TR-069, OMA-DM, etc.
7. Semantic: JSON-LD, Web Thing Model, etc.
8. Multi-layer Frameworks: IoTivity, Weave, Homekit, etc.

Infrastructure protocols: *IPv6*, is an Internet Layer protocol for packet-switched internetworking and provides end-to-end datagram transmission across multiple IP networks. *6LoWPAN* is an acronym of IPv6 over Low power Wireless Personal Area Networks. It is an adaption layer for IPv6 over IEEE802.15.4 links. This protocol operates only in the 2.4 GHz frequency range with a 250 kbps transfer rate. The *uIP* is an open source TCP/IP stack capable of being used with tiny 8 and 16 bits microcontrollers. It was initially developed by Adam Dunkels of the "Networked Embedded Systems" group. *NanoIP*, which stands for the nano Internet Protocol, is a concept that was created to bring Internet-like networking services to embedded and sensor devices, without the overhead of TCP/IP. NanoIP was designed with minimal overheads, wireless networking, and local addressing in mind.

Data protocols: *MQTT* (Message Queuing Telemetry Transport) enables a publish/subscribe messaging model in an extremely lightweight way. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. *CoAP* (Constrained Application Protocol) is an application layer protocol that is intended for use in resource-constrained internet devices, such as WSN nodes. CoAP is designed to easily translate to HTTP for simplified integration with the web, while also meeting specialized requirements such as multicast support, very low overhead, and simplicity. AMQP (Advanced Message Queuing Protocol) is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing, reliability and security.

Communication / Transport layer: *WiFi* protocol is well known and common, so it is not described further. *IEEE 802.15.4* is a standard which specifies the physical layer and media access control for low-rate wireless personal area networks (LR-WPANs). It is maintained by the IEEE 802.15 working group. It is the basis for the ZigBee, ISA100.11a, WirelessHART, and MiWi specifications, each of which further extends the standard by developing the upper layers which are not defined in IEEE 802.15.4. *Bluetooth* works in the 2.4 GHz ISM band and uses frequency hopping. With a data rate up to 3 Mbps and maximum range of 100m. The *ZigBee* protocol uses the 802.15.4 standard and operates in the 2.4 GHz frequency range with 250 kbps. The maximum number of nodes in the network is 1024 with a range up to 200 meters. ZigBee can use 128 bit AES encryption. *WiMax* is based on the standard IEEE 802.16 and is intended for wireless metropolitan area networks. The range is different for fixed stations, where it can go up to 50 km and mobile devices with 5 to 15 km. Standards Communication / Transport layer protocols, data rate and range comparison are shown in Figure 2 [Bauer, 15].

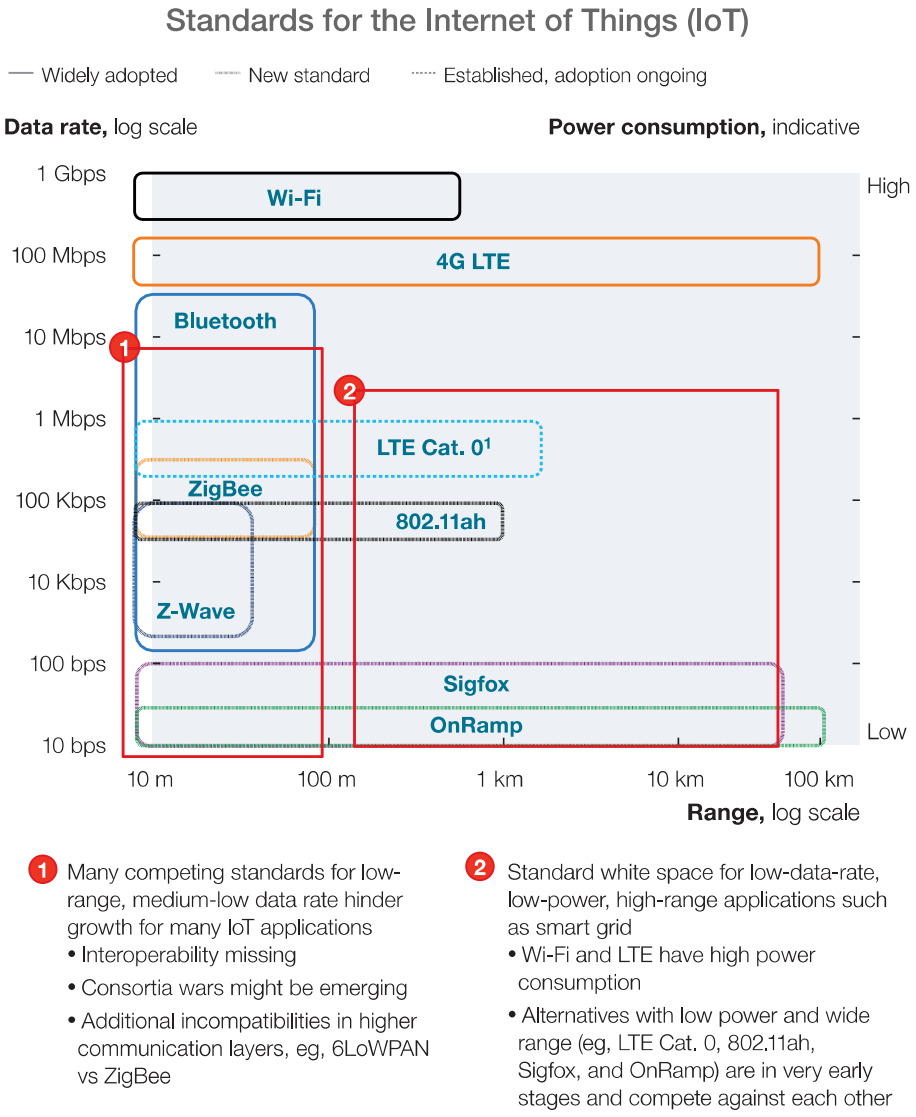

Figure 2: IoT protocols standards data rate and range comparison

The core requirements related to IoT protocol stack include low-power communication protocols and IP-based, secure and reliable communication stacks. The standardization bodies are addressing the issues related to interoperability of protocol stacks, the transparency of open standards, interfaces, and architectures for IoT. This initiative has led towards the creation of IoT-specific protocol stack, since there exists a plethora of wireless protocols, such as ZigBee, IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN), WirelessHART, IEEE 802.15.4e, ISA100.11a.

The overview of IoT standards in network layer architecture is shown in Figure 3 [Vermesan, 13].

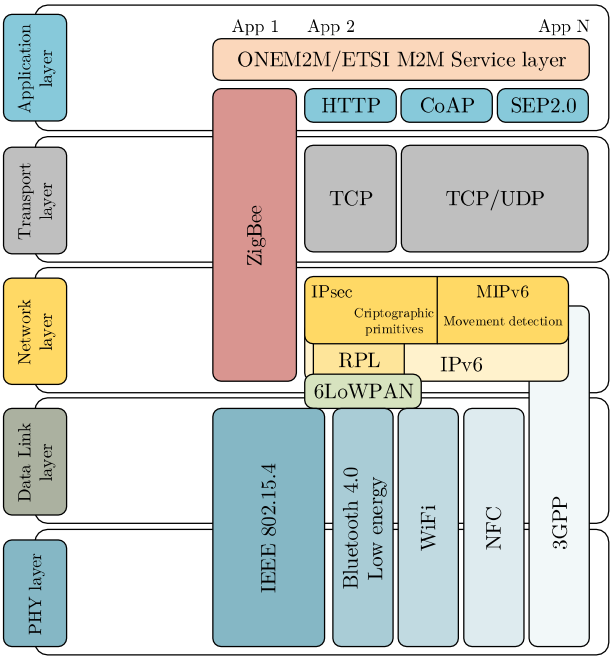

Figure 3: The overview of IoT standards in network layer architecture

3 Thingspeak

ThingSpeak is an IoT analytics platform service that allows developers to aggregate, visualize, and analyze live data streams in the cloud. It is an open-source IoT application programming interface primarily intended for the data collection and storage. There are various add-ons allowing further data processing and usage. One of the most useful add-ons is, for instance the MATLAB Analysis add-on. It allows write and execute MATLAB code to perform data preprocessing, visualizations and analyses. Using the MATLAB script, it analyses the stored data and writes the result to another channel, which can be further visualized. This feature is very valuable, especially in education process where students combine and utilize knowledge from other subjects in their IoT projects. There are also MATLAB Visualizations add-ins used for this visualization. ThingSpeak also includes the Plugins add-on, which is intended for graphical visualization of measured data [MathWorks, 20]. Another very interesting are communication features, for instance ThingTweet add-on allowing to link a channel to a Twitter account and allows scheduled or conditional tweets (alerts) or Twilio service allowing send alert SMS or emails. The TimeControl add-on is designed to automatically launch applications (one-time or periodic). This add-on allows user to

run the following applications: MATLAB Analysis, ThingHTTP, ThingTweet and TalkBack. The *React* add-on is designed for conditional application launch, which responds to predetermined conditions. The TalkBack add-in allows any device connected to the ThingSpeak server to act according to preset commands arranged in a queue. ThingHTTP enables communication between devices, web pages and servers without the need to implement a device-level protocol [ThingSpeak, 19].

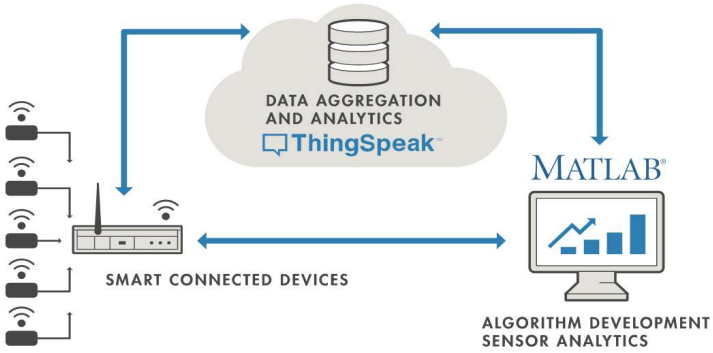

Figure 4: Method of data storage and processing using ThingSpeak

4 IoT Weather station - hardware part

Most efficient way to teach students is to use real models, where can be tested gained knowledge and different case scenario. This is really true especially in area of IoT. For this purpose, educational model of the IoT Weather station ("Meteostanica" in Slovak) has been developed. This and future chapters describe the design and implementation of IoT education model – Weather Station with implemented additional services extending the Weather Station usage possibilities.

The IoT education model is based on the components well known by students from various subjects from their study at the university, so they can focus for implementing IoT capabilities to make all defined services working as required. The weather station is based on microcontroller ATmega2560, development board Arduino Mega 2560 rev3. For monitoring the environment conditions use sensors: DHT11 for monitoring temperature and humidity and BMP180 for atmospheric pressure measurement. The DHT11 sensor has a temperature range from 0°C to 50°C and humidity range of 20% ~ 90% RH. Temperature measurement accuracy is ± 2.0°C and humidity measurement accuracy is ± 5.0 %. The sensor power supply can be from 3V to 5.5V. Sensor response is less than 5s [D-Robotics, 10]. The pressure sensor BMP180 measures an absolute pressure in the range of 30 kPa to 110 kPa (0.3 bar to 1.1 bar). The BMP180 is connected to the I^2C bus and the interface sends the pressure values as a 16-bit value. Power supply of this sensor is from 1.8V to 3.6V and current consumption is 0.3uA [Bosch, 15]. The sensors are shown in Figure 5. As a local output 20x4 LCD display (Hitachi HD44780) with an I^2C interface is used.

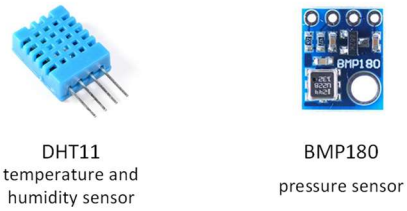

Figure 5: Weather Station sensors

For the communication with remote devices and servers IoT microcontroller ESP8266 [Wang, 17] is used. Hardware components are shown in Figure 6.

Figure 6: Hardware components: ATmega2560, ESP8266 and LCD display

The measured data are shown on the LCD display and also uploaded into the cloud via ThingSpeak API interface by ESP8266. Block scheme of the Weather Station IoT model is shown in Figure 7.

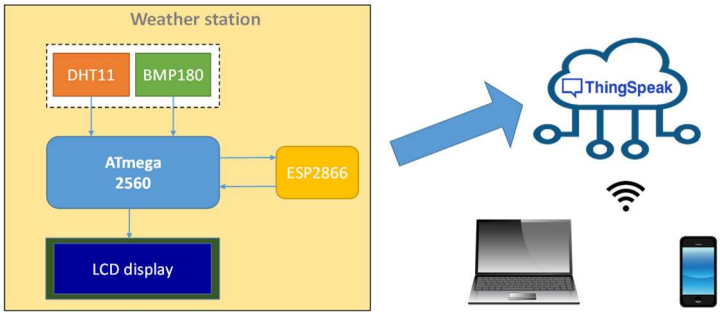

Figure 7: Weather Station block scheme

The assembled and functional educational model of the IoT Weather Station is shown in Figure 8. The weather station model does not have waterproof cover, because it is an educational model intended to use in interior. On the other hand, the sensors are located in outside of the cover box, so the read values are not affected by other components, for instance measuring temperature by heat from the Arduino and ESP.

Figure 8: Weather Station educational model

5 IoT Weather station - software part

Software solution consists of several parts:

- software for the Arduino MEGA 2560 build in the Arduino IDE
- ThingSpeak server scripts and configuration
- project's web site
- additional support application for Windows and Android Smartphone

5.1 ATmega2560 control program

Software for the ATmega2560 is responsible for the measuring the environmental parameters and control ESP8266 communication process. First, the process of *Initialization* is required. During the initial process, used pins are setup for input or output and communication with peripherals: LCD display, ESP8266, Serial port is initialized and attempt to connect to the configured WiFi is made.

The Weather Station itself does not have any user input controls, as shown in Figure 8. Any configuration changes like WiFi SSID or the password can be reconfigured through the serial port commands of the Arduino Mega 2560. Arduino libraries "*dht.h*" for the DHT11 sensor and library "*bmp180.h*" for the BMP180 sensor are used to retrieve data from the sensors. On the regular basis every 30s are data measured and uploaded to the ThingSpeak server, if the connection is established. Communication between ESP8266 and Arduino is based on *WiFiEsp.h* library.

ThingSpeak server support updates every 15s shortest time for the free account. On the other hand, values like air temperature, humidity and air pressure are generally not changing rapidly in seconds so time cycle of 30s is sufficient. Measured data are also displayed on local LCD display, where *LiquidCristal_I2C.h* library is used.

5.2 ThingSpeak and MathWorks implementation

ThingSpeak itself and its services were already described in chapter 3. For the weather station model two channels were created: *Meteostanica* ("Weather Station" in Slovak) and *Meteostanica Analysis*. The channel info is shown in Figure 9.

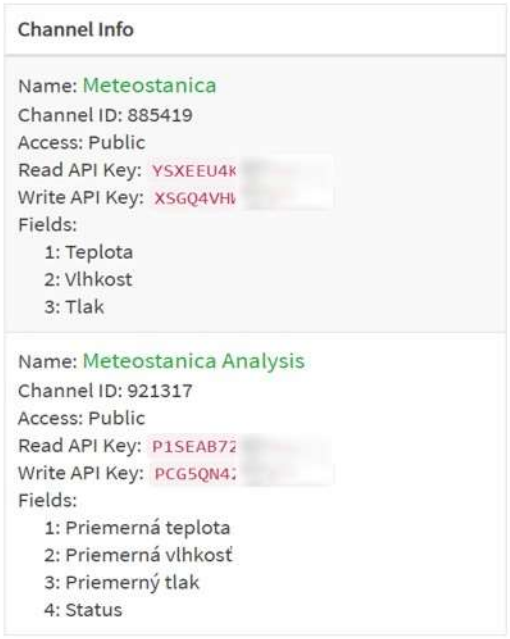

Figure 9: SpeakThing defined channel info

The *Meteostanica* channel contains three fields used for data storage: Field 1 – *Teplota* (temperature in Slovak), Field 2 – *Vlhkosť* (humidity in Slovak) and Field 3 - *Tlak* (pressure in Slovak). All these fields are entered by Arduino via ESP8266. The *Meteostanica Analysis* channel was created for the analysis of the data, specifically calculation of the average values over the specific time period as well as checking if the weather station is active. The *Meteostanica Analysis* channel contains four fields: Field 1 – *Priemerná teplota* (average temperature in Slovak), Field 2 – *Priemerná vlhkosť* (average humidity in Slovak), Field 3 – *Priemerný tlak* (average pressure in Slovak) and Field 4 *Status*. SpeakThing support maximum 8 fields per channel.

The weather station connects to the channels with API keys (Figure 9). The Write API key is used for writing data into the channel. This key is used in the Arduino program to send data to *Meteostanica channel* as well as in the MATLAB Analysis script that calculates the average values and writes them to the *Meteostanica Analysis* channel. The Read API key is used for reading data from the channel. It is used by all MATLAB Analysis and MATLAB Visualization scripts [Tomčíková 19].

For the educational purpose there is a MATLAB Analysis script called *Average values*. The script reads data from Meteostanica channel, sent in the last 10 minutes (educational purpose). Data are read from fields 1, 2 and 3. Values of each measured

quantity are saved in its own vector. Then the average of each vector is calculated and the values are saved into *avgTemperature, avgHumidity* and *avgPressure* variables.

Figure 10: Example of humidity measured value (Left) and calculated average values (Right)

There are 7 scripts for the visualization of data: *Correlation of temperature, humidity and pressure*, *Correlation of temperature and humidity*, *Correlation of temperature and pressure*, *Correlation of humidity and pressure*, *Variation of temperature data*, *Variation of humidity data*, and *Variation of pressure data*.

The *Correlation of temperature, humidity and pressure* script reads last 500 measured values of temperature, humidity and pressure and saves it in their own vectors. Function *scatter3()* prints a 3D graph with the correlation of those quantities.

The *Correlation of temperature and humidity, Correlation of temperature and pressure* and *Correlation of humidity an pressure* scripts is analogous to the *Correlation of temperature, humidity and pressure* script, the only difference is that the graphs with the correlation are printed by the MATLAB function *scatter(),* which is a 2D counterpart to the function *scatter3.*

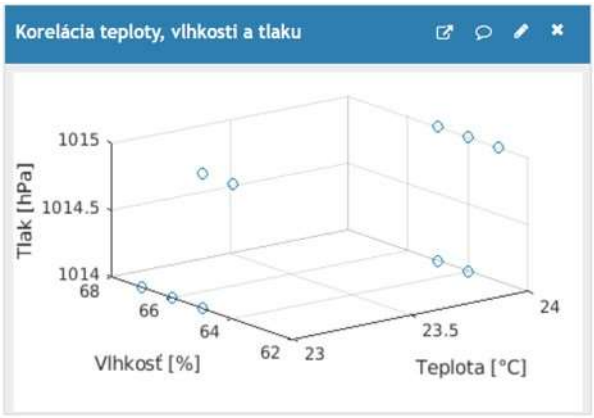

Figure 11: Correlation of temperature, humidity and pressure - Example of the MATLAB visualization

The *Variation of temperature data, Variation of humidity data* and *Variation of pressure* data scripts are analogous to each other. The *Variation of temperature data* script saves the values of temperature, measured over the period of last 10 hours and prints the graph via the MATLAB function *histogram().* MATLAB Visualization scripts are not activated via *TimeControl.* The scripts are activated automatically when the web page of the weather station or the ThingSpeak channels of the weather station are loaded. An example of the histogram graphical representation is shown in Figure 12.

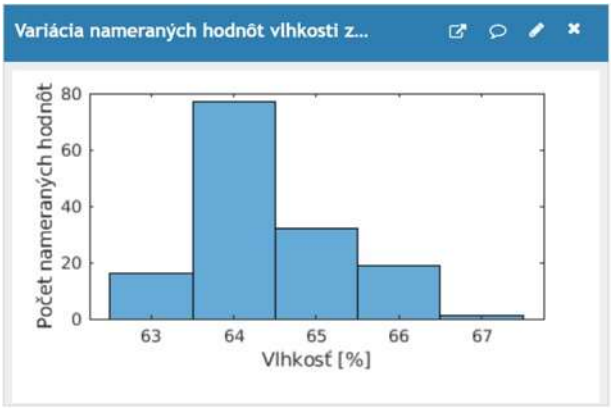

Figure 12: Humidity histogram - Example of the MATLAB visualization

ThingSpeak allows the data to be represented numerically or graphically through various widgets. Both ways of data presentation are shown in Figure 13.

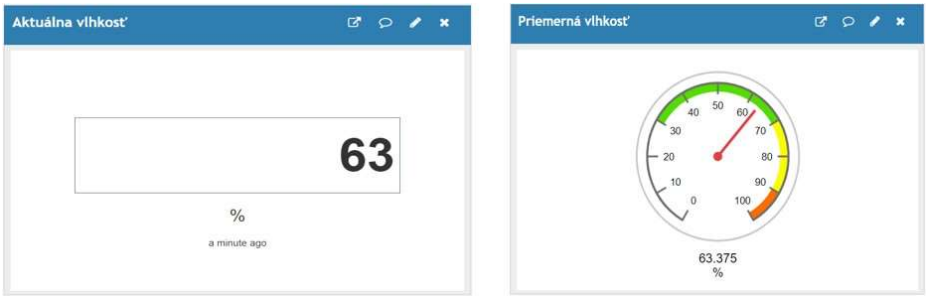

Figure 13: Example of numerical (left) and graphical (right) data presentation widgets

5.3 Weather Station data web presentation

Graphical representation of the measured and analyzed data is directly available on the ThingSpeak weather station channels:

- *Meteostanica* (https://thingspeak.com/channels/885419)
- *Meteostanica Analysis* (https://thingspeak.com/channels/921317)

Since there are two different channels, the charts cannot be found in one place using ThingSpeak web services. Therefore, it was necessary to create a website where graphs from both channels were accessible in one place. For the sake of clarity, the charts are sorted into the following sections: temperature, humidity, pressure and correlation. In addition, there is a section – *Channel* containing links to both channels on the ThingSpeak server, weather station status and location information. Location information is shown in Figure 14. Since all the data are available on the ThingSpeak server, there was no need to implement the website using the Wi-Fi module ESP8266.

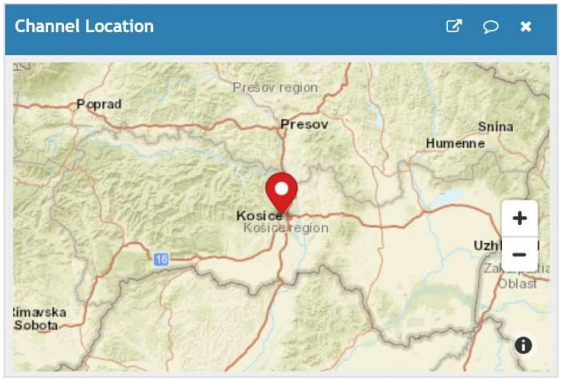

Figure 14: Location information section

The web page code was published on GitHub using GitHub Pages, where default url is: https://sk342hg.github.io/meteostanica. To gain more public-friendly name, Freenom services were used. Freenom is free domain provider, where the URL has been registered: https://meteostanica.ml/. This domain was provided for free and is used to redirect the website from the project GitHub address

5.4 Additional applications

For the easier configuration, Windows application "*Meteostanica control*" in C# has been developed. Using the application, the user can easily change the WiFi connection settings for the ESP8266 and also reset the Weather Station by button. Screenshot of the *Meteostanica Control* is shown in Figure 16.

For simpler and easier reading measured and analyzed data on mobile devices, dedicated Android application was created.

The App consists of eleven screens / views. Screen1 – home screen, contains 10 buttons showing specific data and one button titled "*Webstránka*" which opens the website http://meteostanica.ml/ in the system browser. By clicking on other buttons, the user lunches the assigned screen. Button *Aktualna_teplota* shows the graph and numerical representation of the measured temperature. Button *Priemerna_teplota* shows graph and numerical representation of the average temperature. Button *Aktualna_vlhkost* shows graph and numerical representation of the measured humidity.

Button *Priemerna_vlhkost* shows graph and numerical representation of the average humidity. Button *Aktualny_tlak* shows graph and numerical representation of the actual measured pressure. Button *Priemerny_tlak* shows graph and numerical representation of the average pressure. Button *Korelacia* shows visualized correlation between the measured quantities. Button *Meteostanica_status* contains graph and text status updates of the weather station channel. Button *Informacie* contains information about the project as well as the location of the weather station.

Figure 15: Screenshot of the Weather Station public webpage on https://meteostanica.ml/

Figure 16: Screenshot of the Meteostanica Control application

6 Conclusion

The article described the design and implementation of the Weather station IoT educational model. The weather station is based on the development board Arduino Mega 2560. It senses three variables – temperature and humidity with DHT11 sensor and pressure with BMP180 sensor. It sends measured data via Wi-Fi module ESP8266 to ThingSpeak server and show the data locally on the LCD display.

The ThingSpeak server stores data in the *Meteostanica* channel, where it also plots graphs. This channel is also used for MATLAB visualization, namely correlations between variables and variation of measured values of variables (displayed by histogram). The second channel – *Meteostanica Analysis* – is used for MATLAB data analysis, namely the recording of average values of measured variables and finding the current state of the weather station. The analysis is run by *TimeControl* app. Graphs, visualizations and numerical also graphical representations of both channels are displayed on the website http://meteostanica.ml/ and Android application. For the configuration purpose C# application was created titled *Meteostanica Control* that configure weather station Wi-Fi network via serial port.

For the students is very important to link knowledge from different fields of study. Using this IoT Weather Station education model, there are linked issues of the programming microcontroller (Arduino Mega), measurement and sensors, communication with peripherals (LCD, ESP8266), usage of public cloud services – ThingSpeak, using application programming interface (API) of remote server, Matlab scripts, webpage design, windows application developing, Android Application developing and together to achieve complex system based on IoT technology.

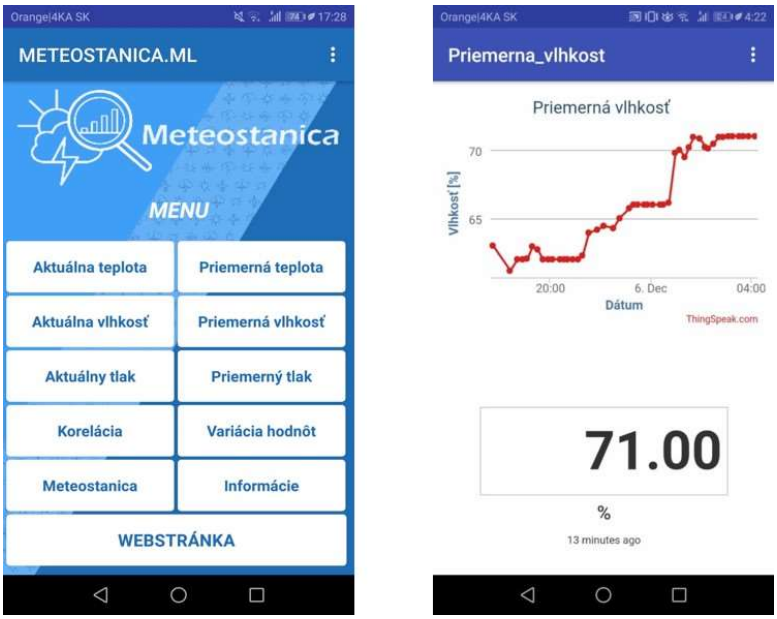

Figure 17: Android application home screen (left) and average humidity (right)

7 Future Work

On actual stage, the IoT Weather Station educational model was focused to interconnect different issues in one complex system as mentioned in the Conclusion. For the communication purpose, it uses the HTTP protocol only for sending data to the ThingSpeak cloud. In near future implementation of different IoT protocols are planned. As a first step MQTT implementation is planned. Also, multiple IoT devices similar to actual Weather Station is planned to implement device-to-device communication. New devices should be built on more sophisticated microcontrollers as mentioned in the Introduction. In later stage implementation of non-WiFi communication is planned so the students gain experiences with communication like ZigBee.

References

[Alulema, 18] Alulema, D., Criado, J., & Iribarne, L. (2018). A Cross-Device Architecture for Modelling Authentication Features in IoT Applications. J. UCS, 24(12), 1758-1775.

[Apiecionek, 15] Apiecionek, Ł., Czerniak, J. M., Dobrosielski, W. T.: "Quality of services method as a DDoS protection tool." Intelligent Systems' 2014. Springer, Cham, 2015. 225-234.

[Apiecionek, 19] Apiecionek, L., Großmann, M., & Krieger, U. R. (2019). Harmonizing IoT-Architectures with Advanced Security Features-A Survey and Case Study. J. UCS, 25(6), 571-590.

[AVSYSTEM, 19] AVSYSTEM, „IoT Standards and Protocols Guide — Protocols of the Internet of Things,“ 2019. https://www.avsystem.com/blog/iot-protocols-and-standards/

[Bauer, 15] Bauer, H., Patel, M., & Veira, J. (2015). Internet of Things: Opportunities and challenges for semiconductor companies. Article by McKinsey'soctober.

[Bosch, 15] Bosch, BMP180 Digital pressure sensor, datasheet 2015, https://datasheet4u.com/datasheet-parts/BMP180-datasheet.php?id=770150

[D-Robotics, 10] D-Robotics, DHT11 Humidity & Temperature Sensor, datasheet 2010, https://datasheet4u.com/datasheet-parts/DHT11-datasheet.php?id=785590

[Dziak 19] J. Dziak, „Model of factory with alternative sources used to reduce power supply.“ In: JIEE Časopis priemyselnej elektrotechniky, journal of industrial electrical engineering. Roč. 3, č. 2 (2019), s. 46-50. ISSN 2454-0900

[Gardašević, 17] Gardašević, G., Veletić, M., Maletić, N., Vasiljević, D., Radusinović, I., Tomović, S., & Radonjić, M. (2017). The IoT architectural framework, design issues and application domains. Wireless personal communications, 92(1), 127-148.

[Garg, 19] Garg, H., & Dave, M. (2019, April). Securing iot devices and securelyconnecting the dots using rest api and middleware. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) (pp. 1-6). IEEE.

[Halachan, 18] P. Halachan, M. Guzan: „Usage of Python programming language in 3D modeling“ In: Journal of Industrial Electrical Engineering. Roč. 2, č. 1 (2018), s. 28-32. - ISSN 2454-0900

[Harwood, 20] T. Harwood, „IoT Standards and Protocols,“ 2020. https://www.postscapes.com/internet-of-things-protocols/.

[Kováčová 18] I. Kováčová „Autonomic transport LEGO robot with optical and ultrasonic sensors“ . In: JIEE Časopis priemyselnej elektrotechniky. Roč. 2, č. 1 (2018), s. 22-27. - ISSN 2454-0900

[Kováčová 18] I. Kováčová, „Simple digital voltmeter and signal generator realized by PC“ In: JIEE Časopis priemyselnej elektrotechniky = = journal of industrial electrical engineering. Roč. 2, č. 4 (2018), s. 1-7. - ISSN 2454-0900

[Lörinc, 20] Lörinc, T (2020): Desing and implementation of IoT simulator, Diploma thesis, Technical university in Kosice, Supervisor: Jan Molnar, 2020

[Martać, 20]Martać, R., Milivojević, N., Despotović-Zrakić, M., Bogdanović, Z., & Barać, D. (2020). Enhancing Large Dam Safety Using IoT Technologies: A Case of a Smart Dam. Journal of Universal Computer Science, 26(5), 583-603.

[MathWorks, 20] MathWorks: ThingSpeak, 2020, https://nl.mathworks.com/help/thingspeak/index.html.

[Medveď, 17] Medveď, D., Zbojovský, J., Pavlík, M., Ilenin, S., Čonka, Z., Kanálik, M., ... & Kolcun, M. (2017). Additional modification of thermomagnetic properties of objects of low relative permeability in electromagnetic field. Acta Physica Polonica A, 131(4), 1138-1140.

[Minerva, 15] Minerva, R., Biru, A., & Rotondi, D. (2015). Towards a definition of the Internet of Things (IoT). IEEE Internet Initiative, 1(1), 1-86.

[Pavlík, 19] Pavlík, M. (2019). Compare of shielding effectiveness for building materials. Przeglad Elektrotechniczny, 95(5), 137-140. ISSN 0033-2097

[Razali,20] Razali, M. A. A., Kassim, M., Sulaiman, N. A., & Saaidin, S. (2020, June). A ThingSpeak IoT on Real Time Room Condition Monitoring System. In 2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS) (pp. 206-211). IEEE.

[Teicher, 18] J. Teicher, „The little-known story of the first IoT device," IBM, 2018, https://www.ibm.com/blogs/industries/little-known-story-first-iot-device/

[Theera-Umpon, 18] Theera-Umpon, N., Han, K. H., Bae, W. S., Lee, S., & Van Huy Pham. (2018). Verifying Secure Authentication Protocol for Communication between IoT-based Medical Devices. J. UCS, 24(9), 1258-1270.

[ThingSpeak, 19] ThingSpeak, Learn More About ThingSpeak. 2019, https://thingspeak.com/pages/learn_more

[Tomčíková 19] I. Tomčíková, „Complete response of underdamped second-order circuits with sinusoidal input" . In: JIEE Časopis priemyselnej elektrotechniky: journal of industrial electrical engineering. Roč. 3, č. 3 (2019), s. 31-36 [online]. - ISSN 2454-0900

[Vermesan, 13] Vermesan, O., & Friess, P. (Eds.). (2013). Internet of things: converging technologies for smart environments and integrated ecosystems. River publishers.

[Wang, 17] Ch. Wang, Esp8266 Series Modules User Manual V1.1, 2017, http://wiki.ai-thinker.com/_media/esp8266/esp8266_series_modules_user_manual_v1.1.pdf

[Zagirnyak, 14] Zagirnyak, M., Mamchur, D., & Kalinov, A. (2014, March). Induction motor diagnostic system based on spectra analysis of current and instantaneous power signals. In IEEE SOUTHEASTCON 2014 (pp. 1-7). IEEE.