Interdisciplinary utilization of IT

S. Neradová, S. Zitta
University of Pardubice, Faculty of Engineering and Informatics, Pardubice, Czech Republic
sona.neradova@upce.cz

Abstract - This article describes a project realized in laboratory seminars. The aim was to practically show students utilization of information technologies in a non-technical discipline. The project involved observing and monitoring breathing of chosen plants. Recorded data were subsequently analysed into several conclusions. The goal was to design and practically build a solution which would enable to record the collected data from the sensors. The designed system contained wireless module and sensors for measuring temperature, humidity and CO_2 concentration. The collected data were transferred to the web server ThingSpeak. Data values could be studied through an internet browser and also downloaded for further analysis. The major founding is that interdisciplinary skills led to successful creation of financially undemanding functional unit, which enabled automation of the data collection. The students were given the opportunity to design an IT platform, to implement and to analyse collected data. Individual components of the project required combination of the skills from informatics, physics, biology and mathematics.

I. Introduction

Generally, education programs and concepts of all schools should preparing students for the "real life", for different tasks in different companies and organisations. Logically, the primary target of every school is to provide students the best education and knowledge of the studied disciplines. This is not an easy task, especially in such dynamic discipline like information technologies. Students have to learn many things, the education programs are focused on programming languages, networking, data security, hardware… The IT world is changing very fast, technologies, which were new when students started are often obsolete when they graduate. This is also very demanding for teachers, who should follow this rapid development and create and continuously update content of their lessons. Although teachers have to focus on the content of the study and how to overhand the knowledge to students, they shouldn't forget that it is also very important to guide students, how they can practically use their knowledge and interconnect it with other, totally different disciplines, what tasks they will face after leaving the school. And, last but not least, an interesting project and its' successful finishing can create some sort of "positive feedback loop"- motivate students for self-study of the given topic and thus deepen their knowledge.

As mentioned in the previous text, a part of the education program of any school with IT as a subject of the study should be a project, where students can prove their ability to practically use what they have learned. Students should be able:

- Understand and analyse the given problem.
- Design a solution (with possible variants) on a theoretical level.
- Choose the best variant with regard to the feasibility, delivery time and given budget.
- Practically realize the chosen variant.
- Evaluate the result, conclude what they have learned.

II. The Project

General description

The described project has been realized with students of the High school of electrical engineering (author is also teaching in this school). The aim was to show students an example, how the IT can be used in a real application for measuring and collecting non-electrical values. To demonstrate their ability to fulfill such task, students had to design a measuring system to ascertain and document development of plants in closed system of breathing gasses (oxygen and carbon dioxide) in natural light conditions with different leaf surfaces and in different stages of their development. An important part of the tasks' definition was to work with very limited financial budget to motivate students to carefully consider the price / performance ratio as it is usually required in the commercial environment.

Further processing and evaluation of the collected data was not a part of this project. However, the collected data were used in another students' project (students from another school).

- Monitoring CO_2 concentration, temperature and humidity in a closed environment with experimental plants (there were 4 plants to be observed).
- Storing the collected data for later analysis and evaluation (the evaluation of the collected data was not a part of the project for students of the electro technical school).
- Keeping the costs of the designed system as low as possible.

Analysis of the project task

At the beginning of the project, it was necessary to analyse the task and determine the particular steps. Initially, the following questions had to be answered:

- How will be the desired quantities measured?
- How long will be the interval between particular measurements?
- How will be the measured data transferred to the data storage?
- Where will be the collected data stored?
- How will be the technical layout designed?

Solution for the particular steps

Once the particular steps were defined, students could start with their realisation.

Measuring the desired quantities

Selecting the most suitable sensors was a crucial part success of the project. Students had to consider several parameters – accuracy of the measurement, digital or analog output, separate or integrated sensors and obviously the price. As there wasn't an absolute accuracy required (more important was to monitor temperature changes during day), there were sensors DHT11 and DHT22 chosen for their excellent performance vs price ratio. These sensors provide acceptable accuracy; integrate temperature and humidity measurement in one unit at low price. There were both types selected to get the opportunity to compare them in a real environment. The situation with CO_2 sensors was different: available sensors are mainly dedicated for industrial applications; their price was therefore too high for a student's project. The only sensor acceptable for the project budget was the MQ135, although this sensor has some drawbacks that students had to overcome.

Determining the intervals of measurement

Students have to find the balance between the amount of collected data and reliability of the measurement. As compromise there was an interval of 10 minutes chosen: it doesn't generate too large data but still provides enough data to compensate failures (caused by interferences, voltage drops etc.).

Data transfer to the data storage

There were two options considered, how to connect the sensors: wired and wireless. Students had to consider pros and cons of both solutions. Wired connection offers higher reliability, but there would be a dedicated computer (with additional interface) needed for connecting the sensors. For wireless connection, there were just Wi-Fi modules required. Potentially lower reliability of the connection could be easily compensated by certain redundancy of measured data and calculating the average. As the most suitable Wi-Fi module and interface for the sensors there was the ESP8266 chosen. Students considered also using Arduino or Raspberry, but they wouldn't offer any advantage in this application compare to the ESP8266.

Storing the collected data

Also for this particular task, students had two options: storing the collected data on a local server or using one of the publicly available cloud services. Using a local server would require certain skills for programming web accessible database and require an additional HW. Generally, it wouldn't bring any advantage comparing to a public cloud, especially because the Thingspeak cloud provides services, which were specifically designed for collecting data using IoT. Using Thingspeak automatically solved tasks like programming a database and backup the collected data. The limits of the free of charge version of this service (which is available for non-commercial projects) weren't affecting this project.

III. Technical Layout

On the basis of the previously described analysis, technical documentation was created. The following table shows the required numbers and types of particular components.

TABLE I. TECHNICAL DOCUMENTATION

Nr.	Technical documentation		
	Name of the component	***Description***	***Qty***
1	Experimental plants	Experimental plants – cress, pea	4
2	ESP8266	Wi-Fi module, interface for sensors	2
3	PSU	Power supply for ESP8266 (phone charger)	2
4	MQ135	Gas concentration sensor	2
5	DHT11	Temperature and humidity sensor	1
6	DHT22	Temperature and humidity sensor	1

Wiring diagram is shown on the next picture.

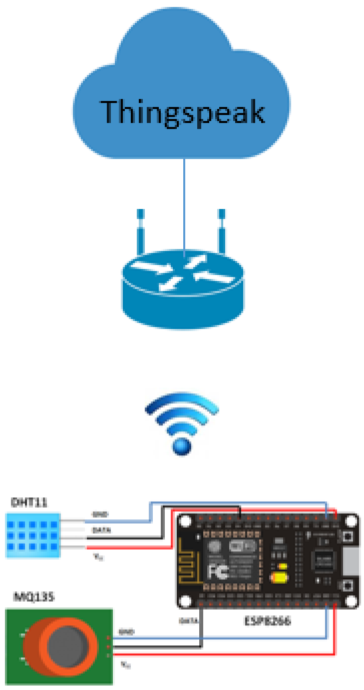

Figure 1. Schematic wiring diagram

The experimental plants were planted into pots and covered with 0,7 litre clear glasses. Along with the plants, sensors MQ135 [2] for measurement of CO_2 concentration and sensors DHT11 [3], DHT22 [4] for measurement of temperature and humidity were situated under the glasses. The sensors were connected to the module ESP8266 with short cables. The module provided reading of the entries from the sensors in 10 minute intervals. Relatively short interval of 10 minutes was chosen to enable elimination of drop-outs caused by random interference. The sensor MQ135 does not provide data of measured temperature but voltage equal to the temperature, therefore the module ESP8266 also performed the conversion. Analog input of the module ESP8266 was used to connect the sensor MQ135. Dependence of the module MQ135 on the temperature is not neutral, therefore correction constant for each of the sensors was established by measuring the voltage at 0°C and 20°C. The constant was afterwards used during the conversion. The module ESP8266 also dispatched the collected data about CO_2 concentration, temperature and humidity via local Wi-Fi network to server Thingspeak.com where the data were saved (including dates and times). After the experiment, the collected, saved data were downloaded from the server in csv format and subsequently evaluated in program MS Excel.

I. The ESP8266 Module Description

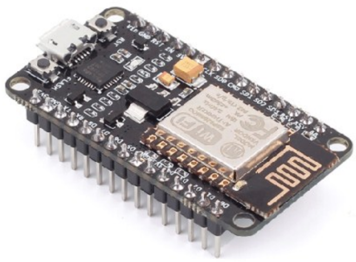

Figure 2. The ESP8266 module

The module ESP8266 is an affordable module with control microprocessor, programmable GPIO, HSPI/UART/PWM/I2C/I2S interfaces and integrated Wi-Fi unit. The module exists in multiple variations which differ with number and type of inputs / outputs and with RAM. The variation ESP8266 ESP-12E Development Board with integrated USB interface and 4MB RAM was used in this experiment. The integrated USB interface facilitates programming of the module and it can be also used for power supply of the module. The firm Espressif Systems is the producer of the module. Detailed data and technical parameters about the module are accessible at the producer's webpage https://espressif.com.

Many programme libraries and even actual firmware for the module ESP8266 are accessible. Its programmes can be written in multiple programming languages. In this experiment, firmware NodeMCU 1.5.4.1 was uploaded to the module, and scripting language Lua was used for programming. Programmes in the language Lua can be written in any text editor. In this project, a specialized editor LuaEdit was used (free-downloadable e.g. at http://luaedit.sourceforge.net) which provides expanded options for editing and tuning of programmes written in the Lua language [5]. The firmware and the actual service programme for communication with the sensors, the temperature calculation and the data dispatch were uploaded to the module via interface ESPlorer (see http://esp8266.ru) which is also freely available, and is specially designed for the ESP8266 modules [6].

II. The MQ135 Module Description

The module MQ135 is sensor-based module designed for measurement of gas concentration. The gas sensor, operative repeater LN393 for output amplification, analogue voltage output and digital output with TTL level are attached to the module. The voltage on the analogue output is directly proportional to the gas concentration. The conversional characteristics between the gas concentration / voltage and additional parameters are stated in the producer's datasheet.

Figure 3. The module MQ135

The DHT11 and DHT22 modules description

The DHT11 and DHT22 modules are designed for temperature and humidity measurement. There is one data output accessible for the connection to the control microprocessor. Also, two power supply pins are present. The difference between the DHT11 and DHT22 types occurs in the range of measurement and the sensitivity. The DHT11 type measures temperature in the range 0 – 50 degrees Celsius with accuracy ±1 °C and humidity in the range 20 – 90 % with accuracy ±4 %. The advanced module DHT22 enables to measure the temperature in the range -40 to +80 degrees Celsius with accuracy ±0,5 °C and the air humidity in the range 0-100 % with accuracy ±2 %. Parameters of the modules are stated in the producer's datasheet.

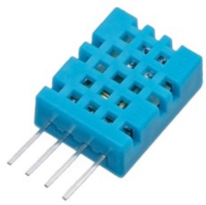

Figure 4. The DHT11 and DHT22 modules for temperature and humidity measurement

III. The Thingspeak Server

In this project, the Thingspeak.com server was used as the data storage. This server does not offer only the storage and basic data visualisation but also an analysis via MATLAB programme, warning dispatch, Twitter messages dispatch and other functions. Some of the functions are available only in the paid version, nevertheless the server is for non-commercial use (with limited number of entries 3 million / year) free of charge.

To use the server, it is necessary to create a user account. After logging in to the account, it is possible to create so called channels where the data are saved. The channels or more precisely the saved data can be tagged and their development can be observed via charts. Data from 8 devices is the maximum which is possible to store in one channel. The saved data are downloadable in csv format at any time. It is also possible to upload data to the server (again in the csv format). The channels can be established as private, thereafter it is necessary to log in for the access to them. The second option is to establish the channel as public, thereafter the view of the channel is freely accessible. When creating a channel, a unique 16 digit alphanumeric chain, so called API-Key, is generated. This chain serves for identification of the channel and it is necessary to enter it when programming the log in procedure of the device connected to the Thingspeak.com server.

Extensive documentation, tutorials, examples of the programming code (including links to extern servers like GitHub or Instructables) and community forum, where it is possible to ask other users and search for solutions, are at disposal at the server.

Following pictures show the data, which were stored on the Thingspeak server.

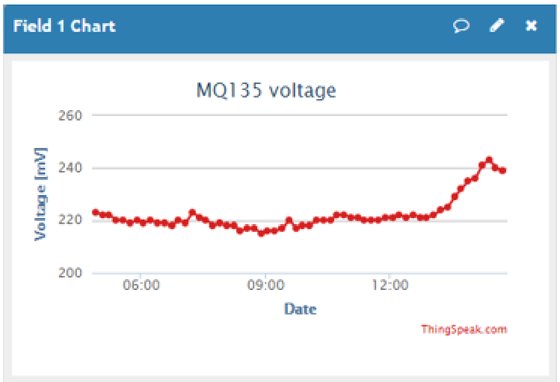

Figure 5. Output voltage

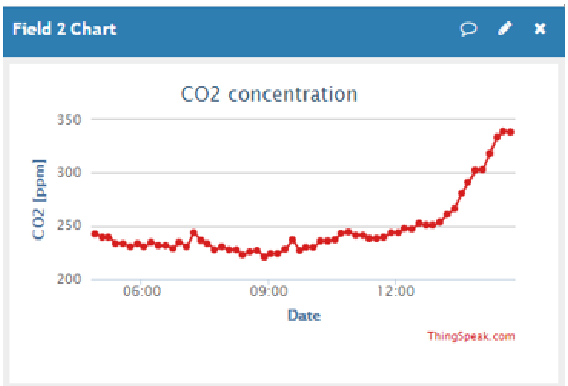

Figure 6. Calculated CO_2 concentration

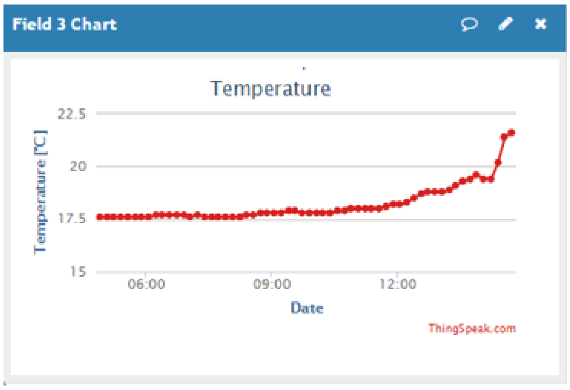

Figure 7. Measured temperature

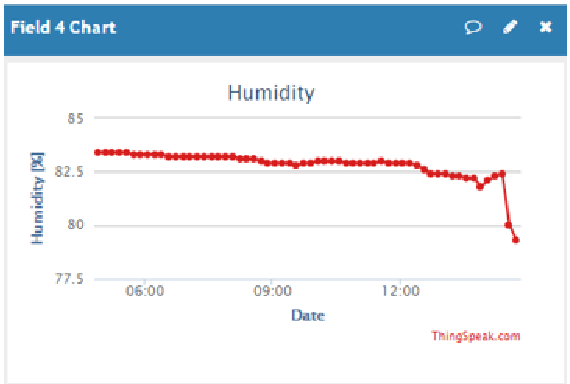

Figure 8. Measured humidity

IV. Conclusion

The goal of the project was fulfilled. The students designed and tested solution, which enabled collecting data from the sensors. Students have learned how to analyse the given task, split it into particular steps and consider possible variants of the solution for each particular step. They have learned that the solution should be the best compromise between the technically ideal solution and the practically available resources.

The students were acquainted with settings of hardware, software installation and data storage at the Thingspeak server. During the project, the students learned how to connect the real word with the digital one. They tried the measuring of physical quantities via sensors in practise. The mutual cooperation between the students during solving the project leads to the cognition that interconnection between different disciplines is very important.

Acknowledgment

This work and contribution is supported by the project of the student grant competition of the University of Pardubice, Faculty of Electrical Engineering and Informatics "Usage of ambient intelligence in the area of information systems and their communication".

References

[1] ThingSpeak, "IoT Analytics – ThingSpeak," (jan 2017) [online]. Available: https://thingspeak.com/

[2] Olimex, ”SNS-MQ135,” (jan 2017) [online]. Available:goo.gl/prBtGw

[3] DHT11 Humidity & Temperature Sensor, ”Micropik.com,” (jan 2017) [online]. http://www.micropik.com/PDF/dht11.pdf

[4] Aosong(Guangzhou) Electronics, ”Digital-output relative humidity & temperature sensor/module. Adafruit,” (jan 2017) [online]. Available: https://cdn-shop.adafruit.com/datasheets/DHT22.pdf

[5] LuaEdit, ”LuaEdit - Free IDE/Debugger/Editor for Lua,” (jan 2017) [online]. Available: http://luaedit.sourceforge.net/

[6] ESP8266, ” esp8266 – Сообщество разработчиков,” (jan 2017) [online]. Available: https://esp8266.ru/

[7] Auxins, Cytokinins, and Gibberellins, ” Boundless learning,” (jan 2017) [online]. Available: goo.gl/1ZttYo